BAB II

METODE PENELITIAN

2.1 Objek Penelitian

Penyakit pada tanaman padi dapat menyebabkan penurunan signifikan dalam kualitas dan kuantitas hasil pertanian. Ketika tanaman padi terkena penyakit dan serangan hama, petani cenderung segera menggunakan pestisida atau metode penanggulangan lainnya, yang tidak selalu sesuai dengan jenis penyakit serta hama pada tanaman padi. Jenis penyakit pada daun tanaman padi bervariasi, dan para ahli dapat mengidentifikasi mereka dengan baik (Jinan and Hayadi, 2022).

Solusi untuk mengatasi penyakit tersebut dapat diberikan setelah jenis penyakit teridentifikasi. Namun, orang yang belum terbiasa mungkin tidak dapat mengenali jenis penyakit pada daun tanaman padi, yang bisa mengakibatkan kesalahan dalam identifikasi dan penanganannya (Saputra *et al.*, 2021). Bahkan para ahli pun bisa melakukan kesalahan identifikasi dalam kondisi tertentu. Untuk mengatasi masalah ini, dapat dilakukan klasifikasi menggunakan jaringan saraf tiruan dengan algoritme *Backpropagation* dan optimisasi menggunakan *Ant Colony Optimization*.

Adapun jenis-jenis penyakit tanaman padi yang digunakan dapat dilihat pada Tabel 2.1

Tabel 2.1 Data Penyakit Tanaman Padi

No.	Penyakit	Gejala
1.	Tungro	• Daun kuning kemerah – merahan • Daun muda menjadi belang/garus hijau pucat • Bulir bercak cokelat dan beratnya kurang dibanding normal • Kerdil • Jumlah anakan sedikit berkurang • Pertumbuhan akar tidak sempurna
2.	Blas	• Bercak berbentuk belah ketupat • Bercak cokelat kehitaman pada batang • Batang mudah patah • Malai hampa atau tidak berisi
3.	Kresek	• Daun berwana putih kekuningan • Layu • Batang berwarna cokelat • Daun garis memanjang atau oval
4.	Bercak daun cokelat	• Bercak muda berbentuk bulat kecil • Bercak berwarna cokelat gelap • Bercak tua berukuran lebih besar (0,4 – 1 cm x -,1 – 02 cm) • Bercak berwarna kuning di sekelilingnya • Bulir berwarna cokelat kehitaman
5.	Bercak cokelat sempit	• Pada daun dan pelepah daun terdapat bercak cokelat yang sempit

		• Varietas yang tahan bercak berukuran 0,2 – 1 cm x 0,1 cm, berwarna cokelat gelap • Varietas bercak lebih besar & berwarna cokelat terang
6.	Bercak Garis	• Muncul garis yang kebasah – basahan diantara urat daun • Garis memanjang dan menjadi cokelat dengan lingkaran kuning di sekelilingnya • Berlendir • Lendir yang kering berbentuk butiran kecil pada garis luka
7.	Hangus Palsu	• Bulir padi menjadi gumpalan spora yang ukurannya sampai 1 cm • Gumpalan spora menjadi hijau gelap • Daun yang menguning menjadi kering
8.	Kerdil Hampa	• Daun jadi kasar, tidak teratur • Bulir padi hanya sedikit yang berisi • Daun menguning dan terpilin • Tanaman menjadi kerdil
9.	Kerdil Rumput	• Tanaman padi sangat kerdil • Daun berwarna kuning / tetap hijau • Tetap berbunga, tapi bulir padi tidak berisi
10.	Busuk Batang	• Pelepah daun terlihat bercak basah berbentuk bulat • Bercak pada bagian tengah berwarna abu – abu & bagian tepi berwarna cokelat • Pembusukan batangnya dari pangkal hingga atas
11.	Kerdil Kuning	• Warna daun dari kuning kehijauan ke kuning keputihan • Bulir padi hampa • Kerdil / pendek

2.2 Data Penelitian

Data untuk penelitian ini dikumpulkan melalui serangkaian observasi langsung pada lokasi yang diusulkan untuk klasifikasi jenis penyakit tanaman padi, dengan total 1094 data. Proses pengumpulan data melibatkan evaluasi terhadap 11 penyakit tanaman padi yang menjadi pertimbangan, sebagaimana terperinci dalam Tabel 2.1

Untuk memperoleh informasi yang akurat dan relevan, penelitian ini juga melibatkan wawancara dengan petani di Lempake, Kec. Samarinda Utara, Kota Samarinda, Bukit Raya, Kec. Tenggarong Seberang, Kabupaten Kutai Kartanegara, Jl. Penghijauan, Bukuan, Kec. Palaran, Kota Samarinda, Jln, tani aman, Loa Janan Ulu, Kec, Loa Janan, Kab. Kutai Kartanegara. Observasi dilakukan untuk menilai kondisi fisik dan lingkungan di sekitar setiap lokasi yang dipertimbangkan. Keseluruhan proses pengumpulan data ini membentuk dasar yang kokoh untuk klasifikasi jenis penyakit pada tanaman padi menggunakan algoritme *Backpropagation* dan optimasi *Ant Colony Optimization*.

2.3 Klasifikasi

Klasifikasi adalah upaya menemukan pola atau fungsi yang menjelaskan dan memisahkan kelas data, serta digunakan untuk mengelompokkan data yang belum diklasifikasikan (Setio, Saputro and Winarno, 2020). Tujuannya adalah guna melatih model dengan menggunakan data latih untuk mengidentifikasi jenis penyakit tanaman padi yang akan diuji (Rizal, Wijaya and Hidayat, 2020).

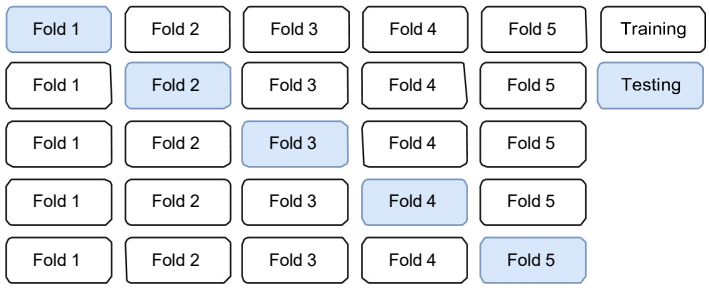
Terdapat beberapa teknik yang dapat digunakan pada klasifikasi, yaitu *regresi logistik, random forest, neural networks, dan Gradient Boosting Machines* (GBM). *Neural networks* atau Jaringan Saraf Tiruan merupakan salah satu teknik yang dapat digunakan untuk melakukan klasifikasi terhadap sekumpulan objek. Jaringan Saraf Tiruan yang digunakan adalah *Backpropagation* (Rizal, Wijaya and Hidayat, 2020).

2.4 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan merupakan salah satu cabang AI (*Artificial Intelligence*). Jaringan Saraf Tiruan adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem otak manusia dalam menerima suatu informasi dan menyelesaikan masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya (Rahmadani and Pardede, 2021). Arsitektur Jaringan Saraf Tiruan terdiri dari neuron yang berada di lapisan input, tersembunyi, dan output yang saling berhubungan (Alkhairi, Damanik and Windarto, 2019). Salah satu metode yang digunakan dalam Jaringan Syaraf Tiruan adalah algoritme *Backpropagation*.

2.5 K-Fold Cross Validation

K-Fold Cross metode yang membagi data menjadi k bagian yang ukurannya sama atau mendekati sama, memungkinkan setiap bagian data dievaluasi secara bergantian. Dengan pembagian ini, prediksi data dapat dilakukan lebih cepat dibandingkan jika data tidak dibagi terlebih dahulu. Akurasi model diuji menggunakan data uji dari setiap fold, dan proses ini berlanjut hingga semua fold digunakan. Akurasi dari setiap fold kemudian dijumlahkan dan dirata-rata dengan membaginya dengan jumlah k(Rangga, Nasution and Hayaty, 2019). Dapat dilihat cara kerja *K-Fold* pada Gambar 2.1



Gambar 2.1 K-fold

2.6 Backpropagation

Algoritme *Backpropagation* pertama kali diajukan oleh Seppo Linnainmaa mahasiswa Finlandia pada tahun 1970. Algoritme klasifikasi *Backpropagation* menggunakan pendekatan untuk meminimalkan fungsi kesalahan dengan asumsi bahwa ekspresi dalam fungsi kesalahan dapat dihitung turunan (Thoriq, 2022). Kemampuan Jaringan Saraf Tiruan *Backpropagation* untuk mempelajari data yang kompleks, multidimensi, dan pemetaan *non-linear* dari data yang besar telah terbukti sangat baik. Oleh karena itu, algoritme ini menjadi salah satu pilihan utama dalam penggunaannya (Satria, 2020).

Algoritme ini dapat menangani berbagai masalah dengan baik. *Backpropagation* merupakan algoritme pembelajaran yang sering digunakan dalam perceptron dengan beberapa lapisan, yang memungkinkan penyesuaian bobot yang terhubung dengan neuron di lapisan tersembunyi. Jaringan Saraf Tiruan *Backpropagation* memiliki konsep pembelajaran yang relatif lebih mudah dipahami dibandingkan dengan metode lainnya (Natalia Napitupulu *et al.*, 2023).

Proses *Backpropagation* terbagi menjadi dua tahap, yaitu *forward-pass dan backward-pass*. Pada tahap *forward-pass*, dilakukan perhitungan untuk *hidden layer* menggunakan rumus berikut ini:

$$H_i = \sum(1_i \times W_{i)} + \sum b_i \tag{2.1}$$

$$\text{Out}\, H_i = \frac{1}{(1+e^{-h1)}} \tag{2.2}$$

$$O_1 = \sum(\text{Out}\, H_i \times W_i) + \sum b_i \tag{2.3}$$

Hasil pada tahap ini adalah arsitektur model dengan bobot-bobot pada *hidden layer*. Selanjutnya dilakukan penghitungan nilai error dengan rumus sebagai berikut :

$$E_i = \frac{1}{2}(t_i - \text{out}\, o_i)^2 \tag{2.4}$$

Dimana (H) merupakan *Hidden Layer*, (i) merupakan *input*, (w) merupakan *weight*/beban layer dan (b) merupakan nilai bias. Setelah dilakukan proses *forward-pass*, selanjutnya dilakukan proses *backward-pass* untuk mengupdate nilai bobot dari masing-masing *hidden layer* menggunakan rumus:

$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial w} \tag{2.5}$$

$$b_{new} = b_{old} - \alpha \frac{\partial E}{\partial b} \tag{2.6}$$

Dimana (E) merupakan *Error*, (w) merupakan *weight* dan (b) merupakan bias. Sehingga setelah seluruh bobot dari *hidden layer* berhasil diperbarui maka model dari *Backpropagation* dapat digunakan (Firzatullah, 2021).

2.7 Multi-class

Multi-class adalah istilah yang digunakan dalam pembelajaran mesin untuk mengklasifikasikan data ke dalam lebih dari dua kategori. Dalam klasifikasi biner, data hanya diklasifikasikan ke dalam dua kategori, seperti "ya" atau "tidak". Sebaliknya, klasifikasi *multi-class* memungkinkan klasifikasi data ke dalam beberapa kategori yang lebih banyak (Shu *et al.*, 2020).

Teknik klasifikasi *multi class* yang umum digunakan yaitu OvA (*One-vs-All*) adalah Teknik ini melatih model klasifikasi biner terpisah untuk setiap kelas, di mana setiap model melatih Jaringan Saraf Tiruan untuk memisahkan kelasnya dari semua kelas lain dan OvO (*One-vs-One*) adalah Teknik ini melatih model klasifikasi biner terpisah untuk setiap pasangan kelas (Shu *et al.*, 2020).

2.7.1 OvO (One-vs-One)

Metode ini bekerja dengan cara memisahkan setiap kelas dari kelas lainnya dalam dataset, mengubah permasalahan klasifikasi *multi-label* menjadi serangkaian permasalahan klasifikasi biner. Model klasifikasi dilatih sebanyak jumlah kelas dalam dataset, dengan setiap model memperlakukan satu kelas sebagai label positif dan kelas lainnya sebagai label negatif (Taufiqurrahman, Al Faraby and Purbolaksono, 2021).

Contoh pada *OvO (One-vsOne)* :

1. Kelas 1 : Hijau vs Biru
2. Kelas 2 : Hijau vs Merah
3. Kelas 3 : Biru vs Merah (Kelen, Baso and Korespondensi, 2023).

2.7.2 OvA (One-vs-All)

OvA adalah sebuah alternatif model dari *OvO* yang bertujuan untuk mengurangi klasifikasi multikelas menjadi biner yaitu membandingkan setiap kelas dengan yang lainnya dalam dua kelas (Clémençon and Vogel, 2020).

Contoh pada OvA (One-vs-All) :

1. Kelas 1 : [Hijau] vs [Merah,Biru]
2. Kelas 2 : [Biru] vs [Hijau,Merah]
3. Kelas 3 : [Merah] vs [Biru,Hijau] (Kelen, Baso and Korespondensi, 2023).

2.8 Ant Colony Optimization

Ant Colony Optimization (ACO) adalah salah satu metode dalam kecerdasan buatan yang digunakan untuk mengoptimalkan suatu sistem. Nama metode ini berasal dari kata-kata "*ant*" yang berarti semut, "*colony*" yang berarti kumpulan atau koloni, dan "*optimization*" yang berarti optimasi. Semut dikenal sebagai hewan yang cerdas karena mereka dapat menemukan jalur terpendek dan tercepat menuju sumber makanan mereka. Perilaku semut ini menjadi inspirasi bagi pengembangan Algoritme *Ant Colony Optimization* (Cahyono *et al.*, no date).

Lumer dan Faieta memperkenalkan algoritme ACO, yang terinspirasi oleh perilaku semut mayat dalam menyortir larva semut. Prinsip dasar pengumpulan dan penyortiran larva semut digunakan sebagai dasar dalam algoritme ini. ACO menyediakan partisi yang relevan dari data tanpa memerlukan pengetahuan awal tentang pusat klister (Silalahi, Fathiah and Supriyo, 2019). Terdapat semut agen yang melakukan perpindahan secara acak pada *grid* dua dimensi dimana dalam*grid* tersebut terdapat objek yg tersebar secara acak, dan ukuran *grid* tergantung pada jumlah objek (Cahyono et al., 2019).

Untuk menerapkan algoritme *Ant Colony Optimization* ke dalam *Backpropagation* dengan mengoptimalkan bobotnya, beberapa penyusunan perlu dilakukan untuk memadukan penggunaan fungsi *Ant Colony Optimization* dengan *Backpropagation*. Perubahan ini tetap memanfaatkan fungsi-fungsi dasar yang ada pada *Ant Colony Optimization*. Beberapa aspek yang perlu modifikasi mencakup :

1. Dalam penggunaan ACO untuk mengoptimalkan bobot pada ANN, tidak ada konsep panjang jarak yang dilalui oleh semut dari titik i ke titik j. Hal ini disebabkan oleh sifat *Arsitektur Neural Network* yang memiliki jarak yang tidak pasti antara node i dan node j. Oleh karena itu, ketika semut memilih probabilitas jalur dari i ke j, hanya intensitas feromon yang dipertimbangkan. Rumus untuk probabilitas pemilihan jalur dari i ke j diubah menjadi:

$$P_{ij} = \frac{c_{ij}\alpha}{\sum c_{ij}\alpha} \quad (2.7)$$

2. Rumus untuk penguapan feromon tetap sama, dengan variabel ρ mewakili laju penguapan feromon dan Δr_{ij} menyatakan total jumlah feromon yang ditambahkan ke jalur oleh semut. Konstanta Q dalam Δr_{ij} menggambarkan jumlah feromon yang ditinggalkan oleh semut pada jalur tertentu, sedangkan L digunakan untuk menggantikan error.
3. Semut menambahkan feromon ke jalur yang mereka tempuh berdasarkan dari kualitas solusi. Feromon ditambahkan dengan rumus :

$$\Delta r_{ij} = \frac{\mathrm{Q}}{\mathrm{Error}} \quad (2.8)$$

Δr_{ij} = $Perubahan\ feromon\ yang\ ditambahkan\ oleh\ semut\ ke\ jalur\ i\ ke\ j$
$Q = Konstanta\ jumlah\ feromon\ yang\ ditinggalkan\ semut$
Error = Cost error dari solusi yang ditemukan oleh semut

4. Pembaruan global feromon dari solusi terbaik tidak diimplementasikan secara eksplisit. Pembaruan feromon masih menggunakan rumus yang sama dengan rumus pembaruhan feromon:

$$r_{ij} = (1 - \rho) \times r_{ij} + \Delta r_{ij} \quad (2.9)$$

Parameter-parameter seperti jumlah semut, jumlah iterasi, alpha, beta, rho, dan Q ditentukan ketika akan menggunakan model *Ant Colony Optimization* dengan memanggil *class Ant Colony Optimization* yang sudah dibangun. Karena panjang jarak yang dilalui semut untuk optimasi bobot *Backpropagation* tidak diketahui dan tidak dihitung, maka beta tidak digunakan meski nilainya disimpan di dalam model *Ant Colony Optimization* (Artanti, 2023).

2.9 Alat dan Bahan

Alat dan bahan yang diperlukan untuk penelitian ini mencakup komponen perangkat lunak dan keras, serta sumber daya data dan informasi tambahan yang akan digunakan selama pelaksanaan penelitian.

a. Alat

Alat penelitian merujuk pada segala jenis perangkat, instrumen, atau metode yang digunakan untuk mengumpulkan data atau informasi dalam suatu penelitian ilmiah. Alat yang digunakan pada penelitian klasifikasi jenis penyakit tanaman padi pada algoritme *Backpropagation* dan optimasi *Ant Colony Optimization* dapat dilihat pada Tabel 2.2

Tabel 2.2 Alat Penelitian

Nama Komponen	Spesifikasi
Device	LAPTOP-AEF91NDV
Prosesor	12th Gen Intel(R) Core (TM) i5-12500H 3.10 GHz
Memory (RAM)	8,00 GB (7,71 GB usable)
Harddisk	477 GB
Sistem Operasi	Windows 11
Bahasa Pemrograman	Python
Text Editor	Google Colab

Perhitungan Manual	Microsoft Excel

b. Bahan

Bahan penelitian tertuju pada segala jenis materi atau sumber yang digunakan dalam suatu penelitian. Bahan yang digunakan dapat dilihat pada Tabel 2.3 :

Tabel 2.3 Bahan Penelitian

Nama Komponen	Spesifikasi
Data gejala penyakit tanaman padi	Diproleh melalui jurnal dan buku.
Hasil kuesioner penyakit tanaman padi	Diperoleh melalui petani.

2.10 Evaluasi

Evaluasi kinerja model *Machine Learning* merupakan aspek krusial dalam memastikan keakuratan dan keandalannya. Di antara berbagai matriks evaluasi, presisi dan *recall* menjadi dua indikator penting yang sering digunakan, khususnya dalam konteks klasifikasi biner. Memahami makna dan hubungan keduanya menjadi kunci dalam menginterpretasikan hasil evaluasi dan memilih model terbaik untuk suatu tugas tertentu.

2.10.1 Akurasi

Akurasi mengukur seberapa tepat suatu sistem dalam mengklasifikasikan data dengan benar. Nilai akurasi adalah rasio antara jumlah data yang diklasifikasikan dengan benar dan total jumlah data yang ada (Agustina et al., 2022).

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} * 100\% \quad (2.10)$$

Keterangan :

1. *True Positive* (TP) : Jika data yang diprediksi bernilai positif dan sesuai dengan nilai actual (positif)
2. *True Negative* (TN) : Jika benar antara prediksi negatif dan aktualnya negatif
3. *False Positive* (FP) : Jika data yang diprediksi tidak sesuai dengan nilai aktual
4. *False Negative* (FN) : Jika yang diprediksi bernilai negatif dan aktualnya positif

2.10.2 Presisi

Presisi merupakan rasio antara jumlah data yang benar-benar terklasifikasi dalam kategori tertentu dengan total keseluruhan data yang diklasifikasikan dalam kategori tersebut (Agustina et al., 2022).

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (2.11)$$

Keterangan:

1. *True Positive* (TP) : Jika data yang diprediksi bernilai positif dan sesuai dengan nilai actual (positif).
2. *False Positive* (FP) : Jika data yang diprediksi tidak sesuai dengan nilai aktual.

2.10.3 Recall

Recall mengukur seberapa baik model dalam mengidentifikasi semua *instance* yang relevan dalam sebuah dataset. Dengan kata lain, *recall* mengukur kemampuan model untuk menemukan semua sampel positif yang benar dalam data. Nilai *recall* yang tinggi menunjukkan bahwa model berhasil menangkap sebagian besar atau semua sampel positif (Azhari, Situmorang and Rosnelly, 2021).

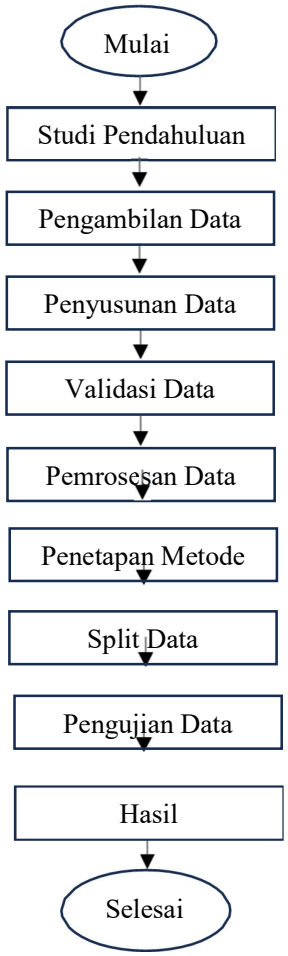
$$Recall = \frac{TP}{FN+TP} * 100\% \quad (2.12)$$

Keterangan

1. *True Positive* (TP) : Jika data yang diprediksi bernilai positif dan sesuai dengan nilai actual (positif).
2. *False Negative* (FN) : Jika yang diprediksi bernilai negatif dan aktualnya positif.

2.11 Diagram Alir Penelitian

Diagram alir penelitian adalah serangkaian langkah-langkah sistematis yang dilakukan untuk mencapai tujuan penelitian. Diagram alir penelitian dapat dilihat pada Gambar 2.2



Gambar 2.2 Diagram Alir

Keterangan diagram alir sistem diatas adalah sebagai berikut :

1. Di mulai dari studi pendahualan, yaitu tahap awal dalam suatu penelitian yang bertujuan untuk mengumpulkan informasi dasar dan memahami konteks penelitian.
2. Proses mengumpulkan informasi pada petani untuk menentukan gejala dan jenis penyakit padi. Proses ini dilakukan dengan cara wawancara dan mengisi kuesioner.
3. Penyusunan data yang telah dikumpulkan sehingga lebih mudah di pahami. Proses ini melibatkan beberapa langkah penting, seperti pengelompokan data berdasarkan kategori tertentu, dan penataan data dalam format yang sesuai.
4. Verifikasi data untuk memastikan bahwa data tersebut akurat, konsisten, lengkap, dan sesuai dengan aturan atau standar yang telah ditetapkan. Proses ini sangat penting untuk memastikan kualitas data yang digunakan dalam klasifikasi.
5. Pemrosesan data adalah serangkaian tindakan yang dilakukan untuk mengolah data mentah menjadi informasi yang bermakna dan dapat digunakan.
6. Menentukan metode yang akan digunakan dalam penelitian. Proses ini melibatkan pemilihan teknik, alat, prosedur, dan langkah-langkah yang dianggap paling efektif dan efisien untuk mencapai tujuan yang diinginkan pada penelitian.
7. *Split data* adalah proses membagi dataset menjadi set pelatihan (*training set*) dan set pengujian (*testing set*).
8. Pengujian data adalah proses evaluasi algoritme untuk menilai kinerjanya dan memastikan akurasi prediksinya menggunakan data yang belum pernah dilihat oleh model sebelumnya.
9. Mendapatkan hasil dari klasifikasi pada data penyakit padi, selesai.