

**KLASIFIKASI JENIS PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *BACKPROPAGATION* DENGAN OPTIMASI ALGORITME
BEE COLONY OPTIMIZATION (BCO)**

SKRIPSI

Diajukan oleh:

Azelina Zahra Riadini

2011102441195



**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
SAMARINDA
JULI 2024**

**KLASIFIKASI JENIS PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *BACKPROPAGATION* DENGAN OPTIMASI ALGORITME
BEE COLONY OPTIMIZATION (BCO)**

SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur

Diajukan oleh:
Azelina Zahra Riadini
2011102441195



**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
SAMARINDA
JULI 2024**

LEMBAR PERSETUJUAN

**KLASIFIKASI JENIS PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *BACKPROPAGATION* DENGAN OPTIMASI ALGORITME
BEE COLONY OPTIMIZATION (BCO)**

SKRIPSI

**Diajukan oleh:
Azelina Zahra Riadini
2011102441195**

**Disetujui untuk diujikan
Pada tanggal 16 Mei 2024**

Dosen Pembimbing



**Fendy Yulianto, S.Kom., M.Kom
NIDN. 1102079402**

**Mengetahui,
Koordinator Skripsi**



**Abdul Rahim, S.Kom., M.Cs
NIDN. 0009047901**

LEMBAR PENGESAHAN

**KLASIFIKASI JENIS PENYAKIT TANAMAN PADI MENGGUNAKAN
METODE *BACKPROPAGATION* DENGAN OPTIMASI ALGORITME
BEE COLONY OPTIMIZATION (BCO)**

SKRIPSI

**Diajukan oleh:
Azelina Zahra Riadini
2011102441195**


**Diseminarkan dan Diujikan
Pada tanggal 16 Juli 2024**

Penguji I



**Abdul Rahim, S.Kom., M.Cs
NIDN. 0009047901**

Penguji II



**Endy Yulianto, S.Kom., M.Kom
NIDN. 1102079402**

**Mengetahui,
Ketua**

Program Studi Teknik Informatika



**Arbansyah, S.Kom., M.T.I
NIDN. 1118019203**

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Azclina Zahra Riadini

NIM : 2011102441195

Program Studi : S1 Teknik Informatika

Judul Penelitian : Klasifikasi Jenis Penyakit Tanaman Padi Menggunakan Metode *Backpropagation* dengan Optimasi Algoritme *Bee Colony Optimization* (BCO)

Menyatakan bahwa **skripsi** yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam skripsi saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Samarinda, 16 Juli 2024

Yang membuat pernyataan



Azelina Zahra Riadini
2011102441195

ABSTRAK

Pada sektor pertanian khususnya budidaya padi, memiliki peran yang sangat penting dalam perekonomian Indonesia. Produksi padi sering terganggu oleh hama dan penyakit sehingga memerlukan sistem klasifikasi untuk deteksi dini. Penelitian ini menggunakan metode *Backpropagation* dan optimasi algoritme *Bee Colony Optimization* (BCO) untuk klasifikasi penyakit pada tanaman padi. Dengan menggunakan 1094 sampel data, $K\text{-Fold} = 2$, *Learning Rate* 0,01, *Epoch* 1000, dua lapisan tersembunyi (masing-masing 78 dan 50 neuron), 18 koloni lebah, dan 5 iterasi serta pembagian data 90% untuk pelatihan dan 10% untuk pengujian. Hasil penelitian menunjukkan akurasi klasifikasi sebesar 94% dari kombinasi *Backpropagation* dan *Bee Colony Optimization* dalam meningkatkan kinerja dan memberikan solusi yang lebih baik untuk klasifikasi penyakit pada tanaman padi.

Kata Kunci: Pertanian, Padi, Klasifikasi, *Backpropagation*, *Bee Colony Optimization*

ABSTRACT

The agricultural sector, especially rice cultivation, has a very important role in the Indonesian economy. Rice production is often disturbed by pests and diseases that require a classification system for early detection. This research uses the Backpropagation method and Bee Colony Optimization (BCO) algorithm optimization for disease classification in rice plants. Using 1094 data samples, K-Fold = 2, Learning Rate 0.01, Epoch 1000, two hidden layers (78 and 50 neurons respectively), 18 bee colonies, and 5 iterations as well as 90% data for training and 10% for testing. The results showed a classification accuracy of 94% from the combination of Backpropagation and Bee Colony Optimization in improving performance and providing a better solution for disease classification in rice plants.

Keywords: Agriculture, Rice, Classification, Backpropagation, Bee Colony Optimization

PRAKATA

Alhamdulillahrabbi'l'alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat, taufik, serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW, yang telah membawa penulis dari alam kegelapan menuju cahaya Islam.

Penyusunan skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur. Proses penyelesaian skripsi ini tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak, baik secara moral maupun material. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberi kerajinan ini jalan dan bantuan kepada hamba.
2. Kedua orang tua penulis, Bapak Hadi Santoso dan Ibu Sri Hartini yang selalu mendo'akan dan memberikan semangat dan dukungan yang tak terhingga kepada penulis.
3. Saudara-saudara penulis Angger Purwadharma dan Bayu Dwi Dharma yang menjadi penyemangat penulis.
4. Bapak Fendy Yulianto, S.Kom., M.Kom selaku Dosen Pembimbing Skripsi yang senantiasa memberikan bantuan kepada penulis sehingga penulisan skripsi dapat selesai tepat waktu.
5. Bapak Arbansyah, S.Kom., M.TI selaku Kepala Program Studi Teknik Informatika
6. Bapak Ir. Sarjito, M.T.,Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur.
7. Bapak Dr. Muhammad Musiyam, M.T selaku Rektor Muhammadiyah Kalimantan Timur.
8. Staff Dosen dan Tendik UMKT yang sudah banyak membantu penulis dalam menimba ilmu selama di bangku perkuliahan.
9. Sahabat penulis Widiya Feri, Lika Saqowati, Najia Raihana, Della Puspita, Sri Mar'ati Sholikhah dan Highness Mailani Putri yang selalu menemani dan memberi dukungan penuh kepada penulis.

Samarinda, 16 Juli 2024

Penyusun



Azelina Zahra Riadini

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN PENELITIAN.....	iii
ABSTRAK	v
<i>ABSTRACT</i>	vi
PRAKATA	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
1.5 Batasan Masalah.....	2
BAB II METODE PENELITIAN	3
2.1 Objek Penelitian	3
2.2 Data Penelitian	4
2.3 Jaringan Syaraf Tiruan	4
2.4 Klasifikasi.....	5
2.5 <i>Multiclass</i>	5
2.5.1 <i>One vs One</i>	5
2.5.2 <i>One vs All</i>	5
2.6 <i>K-Fold</i>	5
2.7 <i>Backpropagation</i>	6
2.8 <i>Bee Colony Optimization</i>	9
2.9 Alat dan Bahan	10
2.10 Evaluasi Kinerja	11
2.10.1 Akurasi	11
2.10.2 Presisi	11
2.10.3 Recall.....	11

2.10.4 F-1 Score	12
2.11 Diagram Alur Penelitian.....	12
2.12 Alur Metode	13
BAB III HASIL DAN PEMBAHASAN.....	19
3.1 Hasil Implementasi Program.....	19
3.1.1 Pembagian Data.....	19
3.1.2 Normalisasi Data	19
3.1.3 Index <i>K-Fold</i>	19
3.1.4 Inisialisasi Bobot <i>Backpropagation</i>	21
3.1.5 Bobot dan Bias Optimasi.....	22
3.1.6 Perhitungan Kesalahan	22
3.1.7 <i>Forward Propagation</i>	23
3.1.8 <i>Backward Propagation</i>	23
3.1.9 Data Kelas, Aktual, Klasifikasi	24
3.1.10 Evaluasi	24
3.2 Hasil Pengujian	25
3.2.1 Parameter Awal <i>Backpropagation</i>	25
3.2.2 Rasio Pembagian Data.....	25
3.2.3 <i>K-Fold</i>	26
3.2.4 <i>Learning Rate</i>	26
3.2.5 <i>Epoch</i>	27
3.2.6 <i>Hidden Layer 1 & 2</i>	27
3.2.7 Parameter Akhir <i>Backpropagation</i>	28
3.2.8 Parameter Awal <i>Bee Colony Optimization</i>	28
3.2.9 Koloni.....	29
3.2.10 Iterasi.....	29
3.2.11 Parameter Akhir <i>Bee Colony Optimization</i>	30
3.2.12 Perbandingan Akurasi	30
3.3 Pembahasan.....	30
BAB IV PENUTUP	32
4.1 Kesimpulan.....	32
4.2 Saran.....	32
DAFTAR RUJUKAN	33
LAMPIRAN	36
DAFTAR RIWAYAT HIDUP	44

DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Penyakit dan Gejala.....	3
Tabel 2.2 Alat.....	10
Tabel 2.3 Bahan.....	11
Tabel 2.4 Perhitungan Bobot Sinyal <i>Input</i>	14
Tabel 2.5 Perhitungan <i>Output</i> pada <i>Hidden Layer</i>	14
Tabel 2.6 Perhitungan Bobot Sinyal <i>Input</i>	15
Tabel 2.7 Perhitungan Faktor δ Unit <i>Output</i>	15
Tabel 2.8 Perhitungan Perbaikan pada Bobot	15
Tabel 2.9 Perhitungan Faktor δ Unit Tersembunyi	16
Tabel 2.10 perhitungan hasil bobot δ_j	16
Tabel 2.11 Perbaikan pada Bobot v_{ij}	16
Tabel 2.12 Perbaikan pada Bobot v_{ij}	17
Tabel 2.13 Perbandingan Sumber Makanan.....	17
Tabel 2.14 Perhitungan Setiap Solusi Lebah Baru.....	18
Tabel 2.15 Solusi Baru	18
Tabel 3.1 Parameter Awal <i>Backpropagation</i>	25
Tabel 3.2 Parameter Akhir <i>Backpropagation</i>	28
Tabel 3.3 Parameter Awal <i>Bee Colony Optimization</i>	28
Tabel 3.4 Parameter Akhir <i>Bee Colony Optimization</i>	30

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 K-Fold	6
Gambar 2.2 Alur Penelitian.....	12
Gambar 2.3 Alur Metode.....	13
Gambar 3.1 Jumlah Data	19
Gambar 3.2 Normalisasi Data	19
Gambar 3.3 Train K-Fold.....	20
Gambar 3.4 Test K-Fold.....	21
Gambar 3.5 Bobot dan Bias <i>Backpropagation</i>	21
Gambar 3.6 Bobot dan Bias Optimasi.....	22
Gambar 3.7 Nilai Kesalahan (loss).....	22
Gambar 3.8 Forward Propagation	23
Gambar 3.9 Backward Propagation.....	24
Gambar 3.10 Data Kelas, Aktual, Klasifikasi	24
Gambar 3.11 Hasil Evaluasi	25
Gambar 3.12 Rasio Pembagian Data.....	25
Gambar 3.13 K-Fold.....	26
Gambar 3.14 <i>Learning Rate</i>	26
Gambar 3.15 <i>Epoch</i>	27
Gambar 3.16 <i>Hidden Layer 1</i>	27
Gambar 3.17 <i>Hidden Layer 2</i>	28
Gambar 3.18 Koloni.....	29
Gambar 3.19 Iterasi	29
Gambar 3.20 Perbandingan Akurasi	30

DAFTAR LAMPIRAN

Lampiran	Halaman
Lampiran 1 Surat Izin Penelitian	36
Lampiran 2 Balasan Surat Izin Penelitian	37
Lampiran 3 Kartu Bimbingan.....	40
Lampiran 4 Wawancara dan Kuesioner Petani.....	42
Lampiran 5 Lembar Kuesioner.....	43
Lampiran 6 Riwayat Hidup	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sektor pertanian memiliki peran yang sangat penting dalam pembangunan ekonomi negara di mana mayoritas penduduk Indonesia bekerja sebagai petani dengan fokus utama pada budidaya padi (Syafitri, Widi Nurcahyo & Sumijan, 2020; Saputra *et al.*, 2021). Padi adalah salah satu jenis tanaman yang dibudidayakan atau dieksploitasi terutama untuk menjaga kehidupan manusia, karena tanaman ini dapat menghasilkan beras dan merupakan makanan utama bagi sebagian besar orang di Indonesia (Jeraman, Faizah & Koryanto, 2023).

Produksi padi di Indonesia mengalami penurunan sepanjang tahun, salah satu faktor yang menyebabkan penurunannya adalah hama dan penyakit. Konversi lahan sawah serta kondisi lingkungan seperti iklim tropis yang lembab dengan curah hujan yang tinggi dapat mempercepat perkembangan hama maupun penyakit tanaman yang berdampak pada penurunan produksi padi. (Sulistyaningsih & Muhlis, 2022; Sari *et al.*, 2023). Beberapa penyakit yang sering menyerang padi seperti: penyakit tungro, penyakit blast, penyakit bercak daun coklat, penyakit hawar pelepah, hawar daun, busuk pelepah, penyakit bercak coklat sempit, penyakit bercak daun coklat dan lain – lain (Sethy *et al.*, 2020).

Beberapa cara untuk mengatasi penyakit pada tanaman padi meliputi pengamatan visual terhadap gejala ataupun menggunakan sistem (Julianto, Sunyoto & Wibowo, 2022). Beberapa sistem yang dapat digunakan untuk menganalisis penyakit yang menyerang tanaman padi seperti penggunaan *clustering*, *association*, klasifikasi dan lain-lain (Herviany *et al.*, 2021). Klasifikasi merupakan salah satu konsep untuk mendeteksi penyakit yang menyerang tanaman padi, dan pada proses klasifikasi memiliki metode yang dapat digunakan seperti Jaringan Syaraf Tiruan, *Machine Learning*, dan lain sebagainya (Rizka Yudana, Suyanto & Nasiri, 2023).

Dalam menentukan jenis penyakit dibutuhkan sebuah pengetahuan untuk dapat mengidentifikasinya. Beberapa penelitian sebelumnya yakni klasifikasi jenis penyakit pada daun tanaman tomat menggunakan metode *Artificial Neural Network* dan *Backpropagation* dengan tiga kelas yaitu *bacterial spot*, *yellow leaf curl virus*, dan daun sehat menghasilkan akurasi sebesar 78% (Putri, 2021). Penelitian lainnya yaitu klasifikasi penyakit daun padi menggunakan metode *Convolutional Neural Network*, didapatkan rata-rata akurasi sebesar 98% (Khoiruddin, Junaidi & Saputra, 2022). Ini menunjukkan bahwa Jaringan Syaraf Tiruan bekerja dengan baik dalam melakukan klasifikasi.

Jaringan Syaraf Tiruan (JST) adalah model handal untuk menggambarkan hubungan *Input-Output* kompleks, dengan kemudahan penggunaan, ketangguhan terhadap *input*, eksekusi cepat, dan kemampuan menginisialisasi sistem kompleks (Satria, 2020). Jaringan Syaraf Tiruan mampu memproses *input* ke *output* dengan mengenali pola, yang mirip dengan cara kerja neuron manusia. Karena kemampuan ini, Jaringan Syaraf Tiruan cocok untuk kasus klasifikasi (Indra Borman, Ahmad & Rahmanto, 2022). Dalam Jaringan Syaraf Tiruan terdapat beberapa metode yang dapat digunakan antara lain, *Perceptron*, *Learning Vector Quantization*, Kohonen, *Backpropagation*, dan *Multi Layer Perceptron* (Marito Tondang *et al.*, 2022).

Backpropagation adalah algoritma pembelajaran yang digunakan dalam Jaringan Syaraf Tiruan dengan beberapa lapisan, yang bertujuan untuk mengubah bobot koneksi antara neuron-neuron di lapisan tersembunyi. (Elvaningsih *et al.*, 2021). *Backpropagation* melatih Jaringan Syaraf untuk mengenali pola dan merespon masukan mirip, tapi tidak identik, dengan pola pelatihan (Hasanati *et al.*, 2020). Selain memiliki kelebihan, algoritme *Backpropagation* juga memiliki kelemahan kecepatan konvergensi yang lambat, ketidakstabilan pembelajaran yang dapat menyebabkan hasil yang diperoleh terjebak pada minimum lokal dan sering dipengaruhi oleh pemilihan parameter bobot awal yang acak (Dwi Suhendra & Chandra Saputra, 2020).

Dikarenakan adanya kekurangan pada *Backpropagation* maka diperlukan suatu metode optimasi untuk mengatasi masalah pembaruan bobot. Beberapa algoritme optimasi yang biasa digunakan yaitu *Adaboost*, *Particle Swarm Optimization*, *Bagging* dan *Genetic Algorithm* (Yunial, 2020). Salah satu algoritme optimasi lainnya yaitu *Bee Colony Optimization*. *Bee Colony Optimization* adalah salah satu algoritme kecerdasan populasi yang meniru perilaku pencarian makanan lebah madu untuk mendorong pencarian menuju solusi optimasi (Huang and Chuang, 2020).

Berdasarkan uraian yang telah disampaikan, penelitian ini akan menggunakan algoritme *Backpropagation* untuk melakukan klasifikasi jenis penyakit pada tanaman padi. Selain itu digunakan juga algoritme *Bee Colony Optimization* untuk mengoptimasi parameter bobot pada metode *Backpropagation* yang nantinya akan membuat hasil klasifikasi jenis penyakit pada tanaman padi menjadi lebih baik.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah (i) Bagaimana proses klasifikasi menggunakan metode *Backpropagation* dan *Bee Colony Optimization* untuk mengklasifikasi jenis penyakit pada tanaman padi?, (ii) Apakah terdapat perubahan terkait hasil evaluasi sebelum dan sesudah menggunakan tambahan metode optimasi *Bee Colony Optimization*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah (i) Mengetahui proses klasifikasi penyakit pada daun padi menggunakan metode *Backpropagation* dan *Bee Colony Optimization*, (ii) Mengetahui apakah ada perubahan hasil evaluasi sebelum dan sesudah menggunakan tambahan metode *optimasi Bee Colony Optimization*.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah (i) Dapat membantu meningkatkan akurasi klasifikasi jenis penyakit pada tanaman padi, (ii) Dapat meningkatkan efisiensi klasifikasi jenis penyakit pada daun padi. Hal ini dapat membantu petani menghemat waktu dan sumber daya dalam proses identifikasi penyakit

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah (i) Objek pada penelitian ini yaitu tanaman padi, (ii) Data yang digunakan diperoleh dari petani melalui kusioner yang berisi gejala dan penyakit tanaman padi, (iii) Klasifikasi penyakit tanaman padi ini menggunakan algoritme *Backpropagation*, (iv) Metode optimasi yang digunakan yaitu *Bee Colony Optimization*.

BAB II

METODE PENELITIAN

2.1 Objek Penelitian

Penyakit pada tanaman biasanya disebabkan oleh jamur dan bakteri yang dapat berdampak negatif terhadap produksi dan kualitas tanaman. Penyakit tanaman padi dapat dikenali dari perubahan fisik pada permukaan daun (Sitompul, Okprana and Prasetyo, 2022). Dalam proses budidaya, banyak tanaman padi yang rentan terhadap serangan penyakit serta hama. Pada kebanyakan kasus, ketika tanaman padi diserang oleh hama dan penyakit petani akan langsung mengaplikasikan pestisida atau memberikan penanganan yang sering kali tidak tepat dengan penyakit dan hama yang menyerang tanaman padi. (Jinan and Hayadi, 2022).

Pada penelitian ini berfokus pada Jaringan Syaraf Tiruan terkait klasifikasi penyakit pada tanaman padi dengan menggunakan metode *Backpropagation* dan optimasi *Bee Colony Optimization*. Sistem ini akan mengolah data dan informasi tentang gejala dan penyakit pada tanaman padi. Adapun jenis-jenis penyakit tanaman padi beserta gejala yang digunakan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penyakit dan Gejala

No	Penyakit	Gejala
1.	Tungro	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- Bercak berbentuk belah ketupat- Bercak cokelat kehitaman pada batang- Batang mudah patah- Malai hampa atau tidak berisi
3.	Kresek	<ul style="list-style-type: none">- Daun berwarna putih kekuningan- Layu- Batang berwarna cokelat
4.	Bercak daun cokelat	<ul style="list-style-type: none">- Daun garis memanjang atau oval- 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 sempit cokelat	<ul style="list-style-type: none">- 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

No	Penyakit	Gejala
6.	Bercak Garis	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Bulir padi menjadi gumpalan spora yang ukurannya sampai 1 cm - Gumpalan spora menjadi hijau gelap
8.	Kerdil Hampa	<ul style="list-style-type: none"> - Daun yang menguning menjadi kering - Daun jadi kasar, tidak teratur - Bulir padi hanya sedikit yang berisi - Daun menguning dan terpilin
9.	Kerdil Rumput	<ul style="list-style-type: none"> - Tanaman menjadi kerdil - Tanaman padi sangat kerdil - Daun berwarna kuning / tetap hijau
10.	Busuk Batang	<ul style="list-style-type: none"> - Tetap berbunga, tapi bulir padi tidak berisi - Pelepah daun terlihat bercak basah berbentuk bulat - Bercak pada bagian tengah berwarna abu – abu & bagian tepi berwarna cokelat
11.	Kerdil Kuning	<ul style="list-style-type: none"> - Pembusukan batangnya dari pangkal hingga atas - Warna daun dari kuning kehijauan ke kuning keputihan - Bulir padi hampa - Kerdil / pendek

2.2 Data Penelitian

Data yang digunakan pada penelitian ini ialah gejala penyakit pada tanaman padi di lingkungan Samarinda. Data tersebut berjumlah 1094 yang didapatkan dengan memberikan kuesioner berisi gejala pada penyakit tertentu kepada petani yang disesuaikan dengan gejala penyakit sesuai literatur. Terdapat 11 penyakit yang menyerang tanaman padi beserta gejalanya seperti yang terlampir pada Tabel 2.1. Teknik pengumpulan data agar mendapat informasi yang akurat pada penelitian ini ialah dengan melakukan wawancara, observasi, dan dokumentasi.

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. 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 *Bee Colony Optimization*.

2.3 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan adalah permodelan data yang sangat efektif yang mampu merepresentasikan dan menangkap hubungan yang kompleks antara *input* dan *output*. Hal ini disebabkan oleh kemampuannya untuk menyelesaikan berbagai masalah dengan relatif mudah, ketahanannya terhadap variasi data *input*, dan kecepatan dalam mengeksekusi dan menginisialisasi sistem yang kompleks (Satria, 2020).

Kemampuan Jaringan Syaraf Tiruan (JST) sangat efektif, seperti yang terlihat dari beberapa aplikasinya. Jaringan Syaraf Tiruan sangat cocok untuk digunakan dalam Klasifikasi, Optimasi, Asosiasi, dan *Self Organizing* (Fitrianingsih Hasan, Kusri and Al Fatta, 2019). Konsep Jaringan Syaraf Tiruan melibatkan pemrosesan sinyal *input* untuk menghasilkan *output* melalui neuron yang tersusun dalam tiga lapisan: lapisan *input*, lapisan tersembunyi, dan lapisan *output* (Hasanati *et al.*, 2020).

2.4 Klasifikasi

Klasifikasi adalah suatu pembelajaran terawasi yang dapat digunakan untuk menganalisis data dan membangun model yang memisahkan data ke dalam kelas-kelas berdasarkan kriteria tertentu (Cervantes *et al.*, 2020). Klasifikasi sebagian besar digunakan untuk memprediksi kelas pada sebuah label, dengan mengklasifikasikan data (membangun model) menurut sekumpulan nilai pelatihan (label kelas) untuk mengklasifikasikan atribut tertentu (Tangkelayuk & Mailoa, 2022).

2.5 Multiclass

Klasifikasi multi-kelas adalah metode klasifikasi untuk mendeteksi satu atau lebih kelas dari sebuah sampel dan setiap sampel hanya dapat dilabeli dengan satu kelas (Ningsih *et al.*, 2024).

2.5.1 One vs One

One-Versus-One merupakan sebuah metode klasifikasi yang bekerja dengan cara membandingkan dua kelas yang berbeda satu sama lain (Kartika Delimayanti *et al.*, 2021).

Contoh :

Terdapat kelas X, Y, dan Z

- Model 1: Melatih untuk membedakan Kelas X versus Kelas Y.
- Model 2: Melatih untuk membedakan Kelas X versus Kelas Z.
- Model 3: Melatih untuk membedakan Kelas Y versus Kelas Z.

2.5.2 One vs All

One-Versus-All adalah sebuah metode klasifikasi yang membandingkan satu kelas dengan semua kelas lain secara bersamaan (Kartika Delimayanti *et al.*, 2021).

Contoh :

Terdapat kelas X, Y, dan Z

- Model 1: Melatih untuk membedakan Kelas X (positif) versus Kelas Y dan Z (negatif).
- Model 2: Melatih untuk membedakan Kelas Y (positif) versus Kelas X dan Z (negatif).
- Model 3: Melatih untuk membedakan Kelas Z (positif) versus Kelas X dan Y (negatif).

2.6 K-Fold

Pengujian *Cross Validation K-Fold* digunakan untuk mengevaluasi kinerja metode algoritme (Adiba, Akram Nur Risal & Tahir, 2023). *K-Fold* digunakan untuk membagi data menjadi data uji

dan data latih dengan tujuan mengurangi kemungkinan terjadinya pembagian data yang menguntungkan, seperti yang bisa terjadi saat menggunakan metode split data sederhana (Pradema Sanjaya, Pribadi & Prastya, 2022). Konsep *K-Fold* dapat dilihat pada Gambar 2.1.

Iterasi 1	Test	Train	Train	Train	Train
Iterasi 2	Train	Test	Train	Train	Train
Iterasi 3	Train	Train	Test	Train	Train
Iterasi 4	Train	Train	Train	Test	Train
Iterasi 5	Train	Train	Train	Train	Test

Gambar 2.1 *K-Fold*

2.7 Backpropagation

Backpropagation merupakan algoritme pembelajaran terawasi yang biasa digunakan oleh *perceptron* berlapis untuk mengoptimalkan bobot dengan menghubungkan neuron-neuron di lapisan tersembunyi. Algoritme ini menggunakan kesalahan *output* untuk mengubah nilai bobot-bobot tersebut dalam arah mundur (*backward*) (Hutabarat et al., 2021).

Untuk mendapatkan kesalahan *output* ini, langkah perambatan maju (*forward propagation*) harus dilakukan terlebih dahulu. Selama perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang bisa didiferensiasikan, seperti sigmoid (Jayanti, Lumbanbatu & Ramadani, 2021).

Backpropagation terdiri dari 3 fase, yaitu *feedforward*, *backward* dan penyesuaian bobot. Berikut rumus – rumus yang digunakan pada fase tersebut (Rolimarch Pratama & Darmawan, 2021). Fase *feedforward* setiap unit *input* menerima sinyal dan mengirimkannya ke unit tersembunyi, yang kemudian menghitung bobot sinyal *input* tersebut (z_j) digunakan rumus pada Persamaan 2.1.

$$z_{in_j} = v_{0j} + \sum_i x_i v_{ij} \quad (2.1)$$

Keterangan :

z_{in_j} = Nilai unit tersembunyi ke- j

x_i = Unit *input* ke- i untuk sebuah unit *input*, sinyal *input* dan sinyal *output* adalah sama yaitu x

v_{0j} = Bobot pada unit di layer ke tersembunyi ke- j

v_{ij} = Bobot pada unit ke- i di layer tersembunyi ke- j

Menghitung *output* pada *hidden layer* menggunakan fungsi aktivasi digunakan rumus pada Persamaan 2.2.

$$z_{out} = \frac{1}{1 + \exp(z_{in_j})} \quad (2.2)$$

Keterangan :

z_{out} = *Output* akhir pada *hidden layer*

z_{in_j} = *Ouput* dari *input* untuk unit ke-j

Menghitung bobot sinyal *input* di unit *output* Y_k digunakan rumus pada Persamaan 2.3 dan 2.4.

$$y_{in_k} = w_{0k} + \sum_i z_j w \quad (2.3)$$

$$y_k = f(y_{in_k}) \quad (2.4)$$

Keterangan :

Y_{in_k} = Nilai *output* tersembunyi ke-k

Y_k = Nilai unit *output* ke-k menggunakan fungsi aktivasi

W_{0k} = Bobot unit tersembunyi ke unit *ouput* ke-k

W_{jk} = Bobot unit tersembunyi ke-j ke unit *output* ke-k

Fase *Backward* menghitung faktor δ unit *output* berdasarkan kesalahan di setiap unit *output* digunakan rumus pada Persamaan 2.5.

$$\delta_k = (t_k - y_k) f^1(y_{in_k}) \quad (2.5)$$

Keterangan :

δ_k = Nilai *error* unit *output*

t_k = Nilai target *output*

Menghitung perbaikan pada bobot w_{jk} yang menghubungkan *hidden layer* dan *input layer* digunakan rumus pada Persamaan 2.6.

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2.6)$$

Keterangan :

δ_k = Nilai *error* unit *output*

α = *Learning Rate*

Δw_{jk} = Perubahan bobot unit tersembunyi ke-i ke unit *output* ke-k

z_j = Luaran di *hidden layer*

Menghitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j digunakan rumus pada Persamaan 2.7.

$$\delta_{in_j} = \delta_k \times w_{jk} \quad (2.7)$$

Keterangan :

δ_{in_j} = Nilai *error* pada *hidden layer*

δ_k = Nilai *error* unit *output*

w_{jk} = Bobot unit tersembunyi ke-i ke unit *output* ke-k

Menghitung hasil bobot δ_j di *hidden layer* digunakan rumus pada Persamaan 2.8

$$\delta_j = (\delta_{in_j} \times z_{out} (1 - z_{out})) \quad (2.8)$$

Keterangan :

δ_j = Nilai *error* unit tersembunyi

z_{out} = *Output* akhir pada *hidden layer*

δ_{in_j} = Nilai *error* pada *hidden layer*

Menghitung perbaikan pada bobot v_{ij} digunakan rumus pada Persamaan 2.9

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.9)$$

Keterangan :

α = *Learning Rate*

δ_j = Nilai *error* unit tersembunyi

Δv_{ij} = Perubahan bobot unit *input* ke-i ke unit tersembunyi ke-j

x_i = Unit *input* ke-i untuk sebuah unit *input*, sinyal *input* dan sinyal *output* adalah sama yaitu x

Penyesuaian bobot di setiap unit *output* digunakan rumus pada Persamaan 2.10

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.10)$$

Keterangan :

$w_{jk}(\text{baru})$ = Bobot setelah diperbarui

$w_{jk}(\text{lama})$ = Bobot sebelum diperbarui

Δw_{jk} = Perubahan pada bobot w_{jk}

2.8 Bee Colony Optimization

Bee Colony Optimization merupakan algoritme kecerdasan yang meniru perilaku koloni lebah dalam mencari makan dengan cerdas. Dalam algoritme ini, setiap sumber madu dianggap sebagai solusi yang layak untuk masalah tersebut, di mana lebah-lebah dibagi menjadi *employee bee*, *onlooker bee*, dan *scout bee* sesuai dengan pembagian kerjanya dan dengan tiga tahap pencarian dalam algoritme tersebut. Lebah-lebah tersebut melakukan aktivitas yang berbeda sesuai dengan pembagian kerja masing-masing dan saling berbagi informasi untuk menyelesaikan proses pencarian masalah (Yang & Liu, 2023).

Berikut adalah tahapan yang dilakukan pada *Bee Colony Optimization* (Arfiani, Yuliansyah & Suratin, 2022).

1. Tentukan jumlah awal solusi potensial atau sumber makanan (x_{ij})
2. Mengevaluasi kualitas sumber makanan (f_i) dalam populasi, karena semakin banyak nektar pada sumber makanan, semakin besar kemungkinan lebah pencari akan memilihnya
3. Fase *Employee Bee* dengan menghitung (v_{ij}) yakni tetangga terdekat dari (x_{ij})
4. Fase *Onlooker Bee* yang akan mengevaluasi kembali dan menggabung sumber makanan dengan nilai probabilitas terbaik pada setiap cluster, dan menghasilkan sumber baru yang disebut dengan (x_{ij}). Langkah ini diulangi, namun sumber terbaik akan tetap disimpan di memori sebagai sumber terbaik pada iterasi tersebut
5. Pada fase *Scout Bee*, sumber makanan baru akan diproduksi selama jumlah sumber makanan belum mencapai batas siklus maksimum. Fase ini menghubungkan iterasi pertama dengan iterasi kedua dan seterusnya.
6. Sampai sumber yang paling cocok ditemukan, pengumpulan sumber akan dibandingkan dengan probabilitas kompatibilitas.

$$x_{ij} = l_{jk} + rand(0,1)(u_{jk} - l_{jk}) \quad (2.11)$$

Keterangan :

- SN = Jumlah sumber makanan
- D = Jumlah dimensi data
- K = Jumlah cluster
- k = {1,2 ..., K}
- i = {1,2, ..., SN}
- x_{ij} = i-sumber makanan awal dalam dimensi ke-j
- l_{jk} = nilai yang lebih rendah dari setiap nilai dalam dimensi-j untuk setiap cluster
- u_{jk} = nilai teratas setiap dimensi-j untuk setiap cluster
- Rand(0,1) = menghasilkan bilangan acak dengan distribusi biasa 0 samai 1

Memulai siklus iterasi sama dengan 1 hingga jumlah siklus maksimum. Untuk setiap solusi penghitungan lebah yang digunakan cara baru:

$$v_{ij} = z_{ij} + \phi_{ij}(z_{ij} - z_{kj}) \quad (2.12)$$

Keterangan :

- v_{ij} = kandidat posisi makanan baru berdasarkan posisi lama di memori.

ϕ_{ij} = angka acak antara $\{-1,1\}$

Indeks acak dalam persamaan berfungsi untuk mengatur produksi sumber makanan di sekitar tetangga dan membandingkan dua posisi sumber makanan yang ditemukan oleh lebah. Langkah selanjutnya ialah menghitung nilai dan memilih sumber makanan terbaik di setiap cluster.

Setiap lebah pengamat akan menentukan solusi z_i berdasarkan p_i . Selanjutnya, hitung nilai v_i sesuai dengan langkah lebah pekerja. Membandingkan dua sumber makanan yang tersedia berdasarkan hasil probabilitas, solusi terbaik pada iterasi ini akan disimpan.

Berdasarkan hasil probabilitas, solusi terburuk yang tersisa ditemukan oleh lebah pengamat dan akan digantikan oleh solusi baru yang dibangkitkan secara acak.

$$Z_j^i = Z_{min}^j + rand(0,1)(Z_{max}^i - Z_{min}^i) \quad (2.13)$$

Keterangan :

Z^i = sumber makanan meninggalkan lebat pengamat dan $\{1,2,\dots\}$.

Ulangi siklus tersebut hingga jumlah siklus maksimum.

2.9 Alat dan Bahan

Pada penelitian ini akan menggunakan beberapa alat dan bahan yang tercantum dalam Tabel 2.2 dan 2.3

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 *Bee Colony Optimization* dapat dilihat pada Tabel 2.2

Tabel 2.2 Alat

Nama Alat	Keterangan
Perangkat	Laptop Asus X430FN_S430FN
Prosesor	Intel Core i5, RAM 12 GB, Windows 11
Bahasa Pemrograman	Python
Text Editor	Visual Studio Code, Google Colab
Perhitungan Manual	Microsoft Excel
Pembuat Laporan	Microsoft Word

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

Nama Bahan	Keterangan
Data gejala dan penyakit tanaman padi	Diperoleh dari artikel dan buku
Hasil kuesioner penyakit tanaman padi	Diperoleh dari petani

2.10 Evaluasi Kinerja

Tahap ini bertujuan untuk mengevaluasi kinerja algoritme yang telah dikembangkan. Pengujian dilakukan dengan mengukur tingkat akurasi klasifikasi algoritme yang menunjukkan seberapa dekat hasil pengujian algoritme dengan nilai sebenarnya (Herdiansah *et al.*, 2022).

2.10.1 Akurasi

Akurasi menunjukkan seberapa baik sistem dapat mengklasifikasikan data dengan benar. Nilai akurasi adalah rasio data yang diklasifikasikan dengan benar terhadap jumlah keseluruhan data. (Agustina, Magdalena & Pratiwi, 2022).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (2.14)$$

Keterangan :

True Positive (TP) : Jika data yang diprediksi bernilai positif dan sesuai dengan nilai actual (positif).

True Negative (TN) : Jika benar antara prediksi negatif dan aktualnya negatif

False Positive (FP) : Jika data yang diprediksi tidak sesuai dengan nilai actual.

False Negative (FN) : Jika yang diprediksi bernilai negatif dan aktualnya positif.

2.10.2 Presisi

Presisi adalah perbandingan nilai jumlah kategori data yang diklasifikasikan secara benar dengan jumlah total kategori data yang diklasifikasikan secara benar. (Agustina, Magdalena & Pratiwi, 2022).

$$Presisi = \frac{TP}{FP + TP} * 100\% \quad (2.15)$$

Keterangan :

True Positive (TP) : Jika data yang diprediksi bernilai positif dan sesuai dengan nilai actual (positif).

False Positive (FP) : Jika data yang diprediksi tidak sesuai dengan nilai actual.

2.10.3 Recall

Recall merupakan perbandingan antara item relevan yang dipilih dengan total jumlah keseluruhan item relevan yang ada (Argina, 2020).

$$Recall = \frac{TP}{FN + TP} * 100\% \quad (2.16)$$

Keterangan :

True Positive (TP) : Jika data yang diprediksi bernilai positif dan sesuai dengan nilai aktual (positif).

False Negative (FN) : Jika yang diprediksi bernilai negatif dan aktualnya positif.

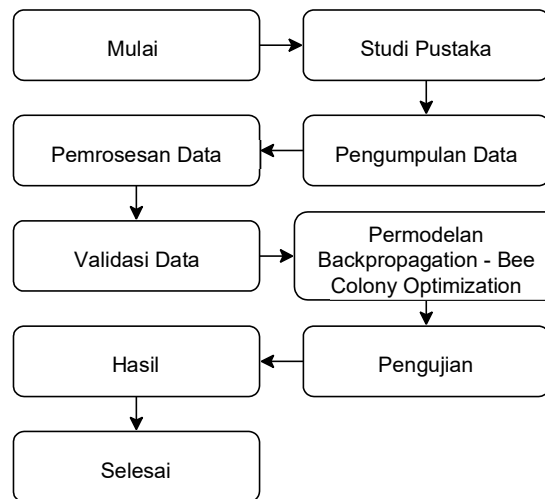
2.10.4 F-1 Score

F-1 Score merupakan metrik evaluasi temu kembali dengan menggabungkan *recall* dan *precision* (Habibi & Winar Cahyo, 2020).

$$F - 1 \text{ Score} = 2 \frac{(\text{Presisi} \times \text{Recall})}{(\text{Presisi} + \text{Recall})} \quad (2.17)$$

2.11 Diagram Alur Penelitian

Dalam penelitian ini dilakukan dalam beberapa tahapan yang dapat dilihat pada diagram alur penelitian pada Gambar 2.2



Gambar 2.2 Alur Penelitian

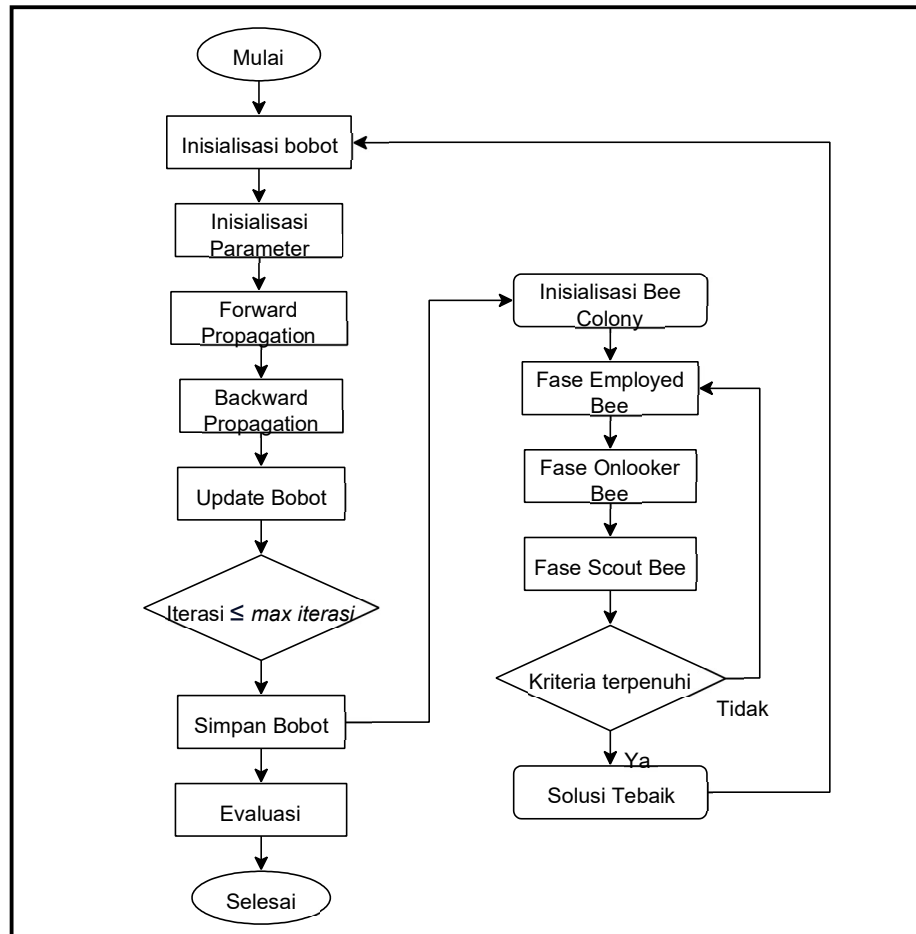
Berdasarkan diagram alur penelitian pada Gambar 2.2 tahapan yang dilakukan pada penelitian ini dapat dijelaskan sebagai berikut :

1. Memulai penelitian dengan melakukan studi pustaka untuk mengumpulkan dan mempelajari literatur atau referensi yang berkaitan dengan topik penelitian, seperti artikel, buku, jurnal, dan sumber-sumber lain.
2. Pada tahapan selanjutnya yaitu mengumpulkan data yang diperlukan untuk penelitian dengan melakukan wawancara dan memberikan kusioner kepada petani.
3. Setelah pengumpulan data, dilakukan pemrosesan data yang bertujuan untuk membersihkan data dengan menghilangkan noise atau kesalahan sehingga siap untuk di analisis lebih lanjut

4. Kemudian dilakukan validasi keakuratan dan keabsahan terhadap data yang telah melewati pemrosean data oleh pakar terkait
5. Selanjutnya, menerapkan algoritme *Backpropagation* dan *Bee Colony Optimization* untuk menghasilkan model yang sesuai dengan penelitian.
6. Pada tahap pengujian data, yaitu menguji hasil yang didapatkan dari permodelan data untuk mendapatkan hasil yang lebih baik
7. Setelah data diuji, hasil pengujian akan ditampilkan, hasil ini dapat berupa laporan, visualisasi data, dan *output* model yang telah diuji.

2.12 Alur Metode

Langkah Langkah dalam mengklasifikasi data dengan menggunakan algoritme *Backpropagation* dan *Bee Colony Optimization* dapat dilihat pada Gambar 2.3



Gambar 2.3 Alur Metode

Berdasarkan alur metode pada Gambar 2.3 tahapan yang dilakukan dapat dijelaskan sebagai berikut :

1. Proses dimulai dengan memuat data yang diperlukan ke dalam sistem yaitu data penyakit yang didapat dari petani

Tabel 2.6 Perhitungan Bobot Sinyal *Input*

y in	y out
0,0770	0,5192
0,0925	0,5231
0,0791	0,5198
0,0662	0,5165
0,0837	0,5209

4. *Backward Propagation* merupakan proses untuk menghitung kesalahan antara *output* yang dihasilkan dan *output* yang diharapkan, kemudian disebarkan kembali ke lapisan sebelumnya untuk memperbaiki bobot

Menghitung faktor δ unit *output* berdasarkan kesalahan di setiap unit *output* menggunakan Persamaan 2.5.

$$\delta_k = (0 - 0,5051) \times (0,5051 \times (1 - 0,5021))$$

$$\delta_k = -0,1263$$

Hasil lengkap perhitungan faktor δ unit *output* berdasarkan kesalahan di setiap unit *output* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Perhitungan Faktor δ Unit *Output*

dk
-0,1296
0,1439
0,1698
-0,1290
0,1944

Menghitung perbaikan pada bobot w_{jk} yang menghubungkan *hidden layer* dan *input layer* menggunakan Persamaan 2.6.

$$\Delta w_{jk} = 1 \times -0,1263 \times (0,7644 + 0,5884 + 0,5913 + 0,5001)$$

$$\Delta w_{jk} = -0,3086$$

Hasil lengkap perhitungan perbaikan pada bobot w_{jk} yang menghubungkan *hidden layer* dan *input layer* dapat dilihat pada Tabel 2.8.

Tabel 2.8 Perhitungan Perbaikan pada Bobot

deltaW
-0,4113
0,5604
0,5630
-0,3289
0,6613

Menghitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j menggunakan Persamaan 2.7.

$$\delta_{in_j} = -1,1263 \times 0,0185$$

$$\delta_{in_j} = -0,0023$$

Hasil lengkap perhitungan faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j dapat dilihat pada Tabel 2.9.

Tabel 2.9 Perhitungan Faktor δ Unit Tersembunyi

dlnJ1	dlnJ2	dlnJ3	dlnJ4
-0,0074	-0,0092	-0,0258	-0,0228
0,0082	0,0102	0,0286	0,0254
0,0097	0,0121	0,0338	0,0299
-0,0074	-0,0092	-0,0257	-0,0227
0,0111	0,0138	0,0387	0,0343

Menghitung hasil bobot δ_j di *hidden layer* menggunakan Persamaan 2.8.

$$\delta_j = -0,0023 \times 0,7644 \times (1 - 0,7644)$$

$$\delta_j = -0,0004$$

Hasil lengkap perhitungan hasil bobot δ_j di *hidden layer* dapat dilihat pada Tabel 2.10.

Tabel 2.10 perhitungan hasil bobot δ_j

dj1	dj2	dj3	dj4
-0,0012	-0,0011	-0,0030	-0,0051
0,0001	0,0003	0,0013	0,0002
0,0016	0,0011	0,0068	0,0024
-0,0017	-0,0023	-0,0054	-0,0047
0,0014	0,0003	0,0093	0,0011

Menghitung perbaikan pada bobot v_{ij} menggunakan Persamaan 2.9.

$$\Delta v_{ij} = 1 \times -0,0004 \times 0$$

$$\Delta v_{ij} = 0$$

Hasil lengkap perhitungan perbaikan pada bobot v_{ij} dapat dilihat pada Tabel 2.11.

Tabel 2.11 Perbaikan pada Bobot v_{ij}

Δv_{10}	Δv_{20}	Δv_{30}	Δv_{40}	Δv_{11}	Δv_{21}	Δv_{31}	Δv_{41}
0,0000	-0,0011	-0,0030	-0,0051	-0,0012	0,0000	0,0000	0,0000
0,0000	0,0000	0,0000	0,0000	0,0000	0,0003	0,0013	0,0002
0,0016	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
0,0000	-0,0023	0,0000	-0,0047	-0,0017	0,0000	0,0000	0,0000
0,0014	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Δv_{12}	Δv_{22}	Δv_{32}	Δv_{42}	Δv_{13}	Δv_{23}	Δv_{33}	Δv_{43}
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
0,0001	0,0003	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
0,0000	0,0000	0,0000	0,0024	0,0016	0,0011	0,0000	0,0000
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
0,0000	0,0000	0,0093	0,0000	0,0000	0,0000	0,0093	0,0011

5. Update bobot memperbarui bobot model berdasarkan kesalahan perhitungan untuk meminimalkan kesalahan pada iterasi berikutnya.

Menghitung perbaikan pada bobot v_{ij} menggunakan Persamaan 2.10.

$$W_{jk}(\text{baru}) = 0,0185 + -0,3086$$

$$W_{jk}(\text{baru}) = -0,2901$$

Hasil lengkap perhitungan perbaikan pada bobot v_{ij} dapat dilihat pada Tabel 2.12.

Tabel 2.12 Perbaikan pada Bobot v_{ij}

Wbaru
-0,3541
0,6316
0,7620
-0,1527
0,7994

6. Mengulang proses pelatihan dengan algoritme *Backpropagation* hingga mencapai jumlah iterasi maksimum.
7. Algoritme *Bee Colony Optimization* dimulai dengan menginisialisasi populasi lebah, di mana *employee bee* mencari dan berbagi informasi sumber makanan, *onlooker bee* memilih sumber makanan berdasarkan probabilitas, kemudian solusi terbaik dari sumber makanan ditemukan, dan *scout bee* secara acak mencari sumber makanan baru untuk dieksplorasi lebih lanjut.
8. Membandingkan sumber makanan yang telah dikumpulkan dengan probabilitas kompatibilitas menggunakan Persamaan 2.11.

Jika nilai absolut

$$31,720 + 0,3 \times (31,720 - 12,906) < 1 = 1$$

Jika nilai absolut

$$31,720 + 0,3 \times (31,720 - 12,906) > 100 = 100$$

Jika tidak, hasilnya adalah absolut dari $31,720 + 0,3 \times (31,720 - 12,906)$

$$x_{ij} = 37,3644$$

Hasil lengkap perhitungan perbandingan sumber makanan yang telah dikumpulkan dengan probabilitas kompatibilitas dilihat pada Tabel 2.13.

Tabel 2.13 Perbandingan Sumber Makanan

w1	w2	w3	w4	w5
37,36442	35,15211	36,0647769	11,52489	40
1,536388	16,32753	36,4999232	28,1874	35
64,16316	59,47177	32,1907857	22,11779	37

Menghitung setiap solusi lebah baru menggunakan persamaan 2.12.

Jika nilai absolut

$$40 + 0,3 x (40 - 15 < 1 = 1$$

Jika nilai absolut

$$40 + 0,3 x (40 - 35 > 100) = 100$$

Jika tidak, hasilnya adalah absolut dari $40 + 0,3 x (40 - 35)$

$$v_{ij} = 41$$

Hasil lengkap perhitungan setiap solusi lebah baru dilihat pada Tabel 2.14.

Tabel 2.14 Perhitungan Setiap Solusi Lebah Baru

w1	w2	w3	w4	w5
48,112835	40,7995	35,934233	6,526131	41
29,776996	5,24458	38,654492	31,22221	34
100	100	57,9434143	39,81202	67

Berdasarkan hasil probabilitas, solusi terburuk yang tersisa adalah lebah *onlooker* dan diganti dengan solusi baru yang diperoleh secara acak menggunakan Persamaan 2.13.

$$Z_j^i = 1 + 0,9 x 88$$

$$Z_j^i = 80,2$$

Hasil lengkap perhitungan Berdasarkan hasil probabilitas, solusi terburuk yang tersisa adalah lebah *onlooker* dan diganti dengan solusi baru yang diperoleh secara acak dilihat pada Tabel 2.15

Tabel 2.15 Solusi Baru

w1	w2	w3	w4	w5
4,2	17	7	29,5	46
15,4	10,9	4,8	10,9	7,1
13	60,4	20,8	90,1	35,2

9. Memeriksa apakah kriteria penghentian telah terpenuhi, misalnya apakah solusi terbaik telah ditemukan atau iterasi maksimum telah tercapai.
10. Mengulang proses *Backpropagation* menggunakan solusi terbaik atau bobot yang didapatkan dari proses *Bee Colony Optimization*.
11. Mengevaluasi kinerja model dengan menggunakan metrik evaluasi seperti akurasi, presisi, dan *recall* yang didapat dari permodelan *Backpropagation* dan *Bee Colony Optimization*, selesai.

BAB III

HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Program

Berikut merupakan tampilan hasil dari program yang telah dirancang untuk mengklasifikasikan jenis penyakit pada tanaman padi

3.1.1 Pembagian Data

Data yang digunakan dalam pemodelan ini dibagi menjadi 90% data latih dan 10% data uji dapat dilihat pada Gambar 3.1

```
Jumlah data latih: 984
Jumlah data uji: 110
```

Gambar 3.1 Jumlah Data

Pada Gambar 3.1 merupakan tampilan jumlah data yang digunakan yaitu 984 untuk data latih dan 110 untuk data uji.

3.1.2 Normalisasi Data

Pada pemodelan dilakukan normalisasi data sehingga hasilnya dapat dilihat pada Gambar 3.2

```
Data setelah normalisasi:
[[-1.38060615]
 [-1.38060615]
 [-1.38060615]
 ...
 [ 1.67190624]
 [ 1.67190624]
 [ 1.67190624]]
```

Gambar 3.2 Normalisasi Data

Gambar 3.2 menunjukkan hasil setelah dinormalisasi dengan menggunakan StandardScaler, setiap fitur dalam dataset disesuaikan sehingga memiliki rata-rata 0 dan standar deviasi 1. Normalisasi ini menyamakan skala semua fitur, yang meningkatkan kinerja model dengan menghilangkan perbedaan skala yang dapat mempengaruhi hasil klasifikasi. Data yang dinormalisasi ditampilkan dalam bentuk array dengan nilai di sekitar 0, termasuk nilai negatif dan positif yang mencerminkan variasi standar. Proses ini mengurangi bias yang disebabkan oleh perbedaan skala, memungkinkan model untuk belajar secara lebih efektif dan melakukan klasifikasi yang lebih akurat.

3.1.3 Index K-Fold

Pada setiap iterasi, data dibagi menjadi dua bagian: data latih (train index) dan data uji (test index). Dalam proses ini $K = 2$, yang berarti data dibagi menjadi dua subset yang digunakan secara bergantian sebagai data latih dan data uji.


```

inisialisasi k-fold cross-validation
train Index K-fold:
 1  4  8 11 13 14 15 16 17 18 19 20 21 22
26 27 32 34 35 36 37 38 40 42 45 46 47 48
50 52 53 57 61 62 64 68 74 75 80 85 87 89
91 93 95 98 99 102 103 105 111 112 114 115 116 117
119 121 122 123 124 126 127 128 129 130 134 138 142 143
146 147 149 150 151 152 153 154 156 157 159 160 161 162
166 167 170 171 175 176 179 180 183 186 187 188 189 190
191 194 195 197 200 201 202 203 205 206 207 216 217 219
222 224 225 226 229 230 232 233 236 240 241 242 245 246
252 253 255 257 258 262 263 267 268 269 271 272 276 278
279 282 283 284 287 288 293 295 297 301 303 304 313 315
317 320 322 324 325 330 335 337 339 340 341 343 345 347
348 349 353 356 358 364 368 369 372 373 374 375 376 378
379 384 385 386 387 391 392 393 395 396 397 399 400 401
402 403 406 407 410 412 414 415 416 417 418 419 421 426
431 434 437 438 441 443 446 452 455 456 459 460 461 463
466 469 470 471 472 473 474 476 480 484 487 488 489 492
496 498 500 502 503 505 508 509 510 511 512 515 517 520
521 524 537 540 544 546 547 550 553 555 556 559 562 563
564 565 569 571 573 574 576 577 579 580 582 586 592 600
603 606 607 608 609 610 612 619 621 623 624 625 627 632
633 635 636 637 638 639 640 641 642 644 645 646 647 648
651 653 654 655 658 659 662 663 664 666 669 671 672 674
676 679 680 681 683 684 685 686 687 690 691 695 698 699
702 705 706 709 710 713 717 719 720 721 724 725 727 729
731 733 734 735 736 738 740 742 744 747 748 749 751 759
763 766 767 768 769 770 772 775 776 777 778 779 782 783
785 790 791 794 795 796 797 800 801 802 804 805 806 807
809 812 815 817 818 819 820 821 822 824 825 826 829 831
833 834 835 836 838 839 840 841 843 847 850 854 856 857
859 860 862 863 865 870 871 873 877 878 879 881 883 884
886 887 888 889 891 894 897 898 899 901 907 910 911 913
915 916 917 919 921 922 924 926 927 929 933 935 936 938
940 941 942 947 952 953 955 957 960 965 967 968 969 971
972 975 976 977 978 979 980 981 982 983 984 985 987 988
989 991 995 996 997 999 1000 1003 1006 1008 1011 1013 1016 1017
1018 1021 1023 1024 1025 1027 1028 1029 1030 1031 1032 1035 1038 1041
1044 1045 1046 1048 1052 1059 1062 1064 1065 1066 1067 1069 1070 1072
1073 1074 1075 1077 1078 1079 1080 1081 1082 1084 1086 1087 1089 1092
1093]

```

Gambar 3.3 Train K-Fold

Gambar 3.3 menampilkan indeks data yang digunakan sebagai data pelatihan pada K-lipatan pertama. Indeks-indeks ini mewakili deretan data yang digunakan untuk melatih model di setiap lipatan. Contohnya, pada lipatan pertama data yang dilatih meliputi indeks 1, 4, 8, 11, dan seterusnya hingga indeks 1093.

```

Test Index k-fold:
[ 0  2  3  5  6  7  9 10 12 23 24 25 28 29
 30 31 33 39 41 43 44 49 51 54 55 56 58 59
 60 63 65 66 67 69 70 71 72 73 76 77 78 79
 81 82 83 84 86 88 90 92 94 96 97 100 101 104
106 107 108 109 110 113 118 120 125 131 132 133 135 136
137 139 140 141 144 145 148 155 158 163 164 165 168 169
172 173 174 177 178 181 182 184 185 192 193 196 198 199
204 208 209 210 211 212 213 214 215 218 220 221 223 227
228 231 234 235 237 238 239 243 244 247 248 249 250 251
254 256 259 260 261 264 265 266 270 273 274 275 277 280
281 285 286 289 290 291 292 294 296 298 299 300 302 305
306 307 308 309 310 311 312 314 316 318 319 321 323 326
327 328 329 331 332 333 334 336 338 342 344 346 350 351
352 354 355 357 359 360 361 362 363 365 366 367 370 371
377 380 381 382 383 388 389 390 394 398 404 405 408 409
411 413 420 422 423 424 425 427 428 429 430 432 433 435
436 439 440 442 444 445 447 448 449 450 451 453 454 457
458 462 464 465 467 468 475 477 478 479 481 482 483 485
486 490 491 493 494 495 497 499 501 504 506 507 513 514
516 518 519 522 523 525 526 527 528 529 530 531 532 533
534 535 536 538 539 541 542 543 545 548 549 551 552 554
557 558 560 561 566 567 568 570 572 575 578 581 583 584
585 587 588 589 590 591 593 594 595 596 597 598 599 601
602 604 605 611 613 614 615 616 617 618 620 622 626 628
629 630 631 634 643 649 650 652 656 657 660 661 665 667
668 670 673 675 677 678 682 688 689 692 693 694 696 697
700 701 703 704 707 708 711 712 714 715 716 718 722 723
726 728 730 732 737 739 741 743 745 746 750 752 753 754
755 756 757 758 760 761 762 764 765 771 773 774 780 781
784 786 787 788 789 792 793 798 799 803 808 810 811 813
814 816 823 827 828 830 832 837 842 844 845 846 848 849
851 852 853 855 858 861 864 866 867 868 869 872 874 875
876 880 882 885 890 892 893 895 896 900 902 903 904 905
906 908 909 912 914 918 920 923 925 928 930 931 932 934
937 939 943 944 945 946 948 949 950 951 954 956 958 959
961 962 963 964 966 970 973 974 986 990 992 993 994 998
1001 1002 1004 1005 1007 1009 1010 1012 1014 1015 1019 1020 1022 1026
1033 1034 1036 1037 1039 1040 1042 1043 1047 1049 1050 1051 1053 1054
1055 1056 1057 1058 1060 1061 1063 1068 1071 1076 1083 1085 1088 1090
1091]

```

Gambar 3.4 Test K-Fold

Gambar 3.4 menampilkan data yang tidak termasuk dalam indeks latih digunakan sebagai data uji. Misalnya, indeks seperti 1, 3, 5, 7, 9 hingga 1091 digunakan sebagai data uji untuk mengevaluasi kinerja model dalam mengklasifikasikan data baru.

3.1.4 Inisialisasi Bobot *Backpropagation*

Dalam *Backpropagation*, langkah awal yang sangat penting adalah inisialisasi bobot (*weights*) dan bias. Inisialisasi ini memungkinkan pelatihan sebuah Jaringan Syaraf Tiruan (*Neural Network*) untuk dimulai dengan benar.

```

Menentukan struktur dan parameter dari BPNN
Menginisialisasi nilai bobot dan bias dari Neural Network
w1_initial: [[ 0.23255695  0.04772174  0.66387384  0.41729439 -0.38748507 -0.68590824
 0.88434333  0.76863848 -0.53964072  0.55190385 -0.52393381  0.82361526
-0.49706707 -0.28817176  0.66038088 -0.99617502 -0.36886633 -0.35076664
-0.06050512  0.92459463 -0.26582126  0.7569381 -0.6735838  0.59372495
-0.11108869 -0.83663471  0.45096685  0.87303309 -0.92815485  0.90980798
 0.01293251 -0.42060402 -0.21081602 -0.77608461 -0.47225416 -0.99517125
-0.93855035  0.28559445  0.88387384  0.75057908  0.70315688  0.59832063
 0.78505934  0.81619412 -0.95781168 -0.30035393  0.7930372 -0.11065529
 0.43348938  0.59342244  0.24969776 -0.73170277  0.56320831  0.29767517
 0.36387007 -0.2454185  0.53473094  0.28657481  0.39815593 -0.38643569
-0.42666732  0.31983113  0.13782081  0.4637614  0.92488895 -0.57724433
-0.91866646 -0.29825063 -0.1285298 -0.13880253 -0.00501565 -0.76855278
-0.27761075  0.30481097 -0.95504654 -0.4558613 -0.71405949 -0.20532347]]
b1_initial: [[ 3.20157490e-01 -7.20531818e-01  8.40391152e-01  8.61750265e-01
 8.37716138e-01  9.37340042e-01  1.51891327e-01  4.49953174e-02
 1.30551109e-03 -5.42157854e-02  9.15218621e-01 -2.99057277e-01
 9.28356534e-01  9.53097104e-02  9.47802651e-01 -2.06293975e-01
 3.35134395e-01  7.64357218e-03 -5.21475370e-01 -4.63683801e-04
 5.90082159e-01 -6.06880968e-02 -6.20847835e-01 -3.00693590e-01

```

Gambar 3.5 Bobot dan Bias *Backpropagation*

Gambar 3.5 menampilkan inialisasi bobot (W_1, W_2, W_3) dan bias (b_1, b_2, b_3) pada Jaringan Syaraf Tiruan *Backpropagation* menghubungkan antar neuron dan menentukan nilai awal untuk setiap lapisan.

3.1.5 Bobot dan Bias Optimasi

Bobot dan bias *Backpropagation* yang telah di optimasi oleh pemodelan ditampilkan dalam Gambar 3.6.

```
Optimized Weights and Biases:
new_W1: [[-2.61359178 -4.64569694 -5.32675357 -1.8020603  0.17924981  2.80802257
 2.16599169 -4.71074133 -2.62904358 -2.7785822  -0.24774264  3.59135545
 2.82509833 -0.56886304  2.36921212  2.75548983 -4.09624396 -2.26039238
 -3.31633996  3.4621162  1.93174138  1.62075455  2.31152526 -2.49592217
 -0.24915995  5.7634762  -0.60167989 -2.29551749 -1.25810857  2.76052284
 -0.80269471  2.41524985 -3.97590289 -2.87626682 -3.0782073  -3.04459014
 2.75756043  0.01593445  3.05336095  3.59164856 -2.81159396  4.56481203
 4.2867424  4.89694042 -2.36731379  3.4128215  4.04823112  3.43397569
 -1.82616935  2.45931815 -2.68722392  3.16567149 -5.04286433 -0.80479122
 -2.79487261  4.47077285  2.78623534 -1.58493471 -2.5467437  1.97965101
 -1.33638997  3.97505511 -2.79355018 -1.25493549  4.38597167 -6.2659309
 4.57906369  1.2192929  0.30563865 -4.61705971 -2.29667449  3.7391787
 4.62231095  2.61377362  2.78252712  3.55540125 -4.12519123 -5.33603844]]
new_b1: [[-2.042230743 -5.29365144  5.33736262 -1.80377947 -2.49039437  0.42286542
-2.8668917  1.04983479  2.08909172 -3.7566014 -2.16699235 -4.56890299
 1.95864001 -2.52367366 -3.20349733 -2.74001814  1.21066912 -2.10913922
 -3.28563607  1.98566698 -2.13323079 -2.16278873 -2.38495868 -1.8377596
 -2.53023331 -7.12332612 -1.97055274 -2.02734067 -1.02010516  0.93333215
 -2.55993231 -2.36674846 -4.4401978  2.15087013 -2.2201707 -2.96215844
 -3.5865209 -2.53735491  2.34778994 -4.04600486  0.10045585 -5.61444418
 -2.15695553  4.23859899 -1.09976594 -4.17238169  1.45707273  2.58224791
 -1.51258566  0.85980669  2.06897424 -4.38475159  2.08672354 -2.65560117
 -2.86760033  3.85909502  1.29040736 -1.05633313 -1.68734498 -0.5122844
 -0.78062931  1.26429678  2.20856162 -1.67720608  3.97556051  6.81531907
 -5.27678202 -2.93325855 -1.93785993  5.23882091 -1.94473813 -4.12545177
 4.78728667 -3.29183008  1.18827376  2.67168421 -0.66851014 -5.87685706]]
new_W2: [[-0.56856557 -0.49596179  0.1578991 ... -1.26537839 -0.17210606
-0.81687983]
[-0.25727811  0.57094075 -0.30119605 ... -2.67048606  0.55498618
 0.580032 ]
[ 0.61006708  0.2123116 -2.91947061 ... -1.36469902  1.27625878
 -0.44646102]
```

Gambar 3.6 Bobot dan Bias Optimasi

Gambar 3.6 menampilkan hasil optimasi bobot (W_1, W_2, W_3) dan bias (b_1, b_2, b_3) pada Jaringan Syaraf Tiruan, yang digunakan untuk meminimalisasi kesalahan klasifikasi melalui penyesuaian model terhadap data pelatihan, demi meningkatkan akurasi klasifikasi.

3.1.6 Perhitungan Kesalahan

Nilai kesalahan (*loss*) yang diperoleh dari permodelan *Backpropagation* dan *Bee Colony Optimization* ditampilkan pada Gambar 3.7

```
Epoch 0, Loss: 6.971480311517462
Epoch 100, Loss: 3.5468439240424914
Epoch 200, Loss: 2.6874158636461596
Epoch 300, Loss: 3.0070231369828724
Epoch 400, Loss: 1.3713201519801521
Epoch 500, Loss: 2.1972709977615197
Epoch 600, Loss: 0.9094261607307884
Epoch 700, Loss: 1.027388734222912
Epoch 800, Loss: 0.9172765499861493
Epoch 900, Loss: 0.3595208858213698
```

Gambar 3.7 Nilai Kesalahan (loss)

Dalam Gambar 3.7 terlihat kurva "loss" (kesalahan) yang berangsur menurun dengan sedikit naik turun selama pemodelan. Hal ini menunjukkan bahwa model sedang dalam proses belajar dan perbaikan untuk mencapai performa klasifikasi data yang lebih optimal.

3.1.7 Forward Propagation

Proses *Forward Propagation* dengan inputan -1,38060615 menghasilkan luaran seperti yang ditampilkan pada Gambar 3.8

```
Forward propagation for data index 0:
Input: [[-1.38060615]]
Layer 1 activation (a1): [[8.30320559e-01 6.78952072e-02 9.44770854e-02 7.20993607e-01
4.33436165e-02 1.01702407e-01 7.51611560e-01 6.33778324e-01
1.76086429e-01 1.42128977e-02 1.65805104e-03 1.31113129e-01
3.02710235e-04 1.72851955e-04 8.30781042e-01 4.30850890e-02
7.50604914e-01 9.90408079e-01 6.81226661e-01 7.41709263e-03
8.16700385e-01 1.39627428e-01 6.68382403e-01 9.16890522e-01
2.89214887e-03 1.03620815e-03 3.25898548e-02 2.98208110e-03
2.47373528e-02 6.52023525e-01 8.97731121e-01 1.00273215e-02
9.98616767e-01 1.31398422e-01 7.41737042e-02 1.62097168e-02
7.60399712e-04 9.92462807e-01 8.46057380e-02 9.99699097e-01
8.77856708e-01 2.70225808e-02 1.96886292e-02 4.86310447e-01
6.79094301e-01 4.25284608e-02 5.13022985e-04 8.51130190e-01
9.99907670e-01 3.60996061e-02 9.99340954e-01 9.99750757e-01
8.09311693e-01 9.57435922e-01 9.85558439e-01 5.95591103e-02
9.94664887e-01 8.26344739e-01 9.60212879e-01 8.83435248e-02
1.40828559e-02 8.70550300e-01 4.06887438e-05 9.99923557e-01
4.44219648e-03 1.66493757e-03 8.54548375e-02 9.97407446e-01
9.69814014e-01 8.47840323e-01 9.99081420e-01 9.50948141e-01
4.30263134e-04 6.81847488e-02 9.50869819e-01 8.04632232e-02
4.44940581e-02 1.55213702e-01]]
Layer 2 activation (a2): [[1.23066116e-21 2.27974351e-08 2.89950718e-07 1.31991614e-08
1.96715919e-04 3.87613315e-06 1.92542644e-04 2.29299155e-05
1.09670588e-10 1.97449266e-10 1.04834775e-06 5.71555752e-05
1.48410234e-24 1.59715444e-07 4.47358634e-06 6.26229467e-05
9.99992607e-01 1.04308741e-03 4.53105629e-16 8.11882100e-06
1.00000000e+00 3.64894495e-04 8.03726221e-19 1.46996542e-03
9.98381335e-01 2.73446423e-03 1.00000000e+00 9.34353377e-04
1.20204236e-06 9.99999999e-01 8.48614364e-07 9.99999771e-01
1.19769039e-07 3.97311675e-10 4.50474571e-04 3.71277536e-04
3.63989373e-04 1.00000000e+00 2.36420711e-10 1.45291809e-11
4.49336963e-21 1.28708180e-05 1.00781166e-04 6.74515760e-06
9.99999999e-01 5.88460026e-23 9.99999995e-01 8.07509542e-19
5.98416177e-09 3.90600093e-04]]
```

Gambar 3.8 Forward Propagation

Pada Gambar 3.8 menampilkan keluaran dari proses *forward propagation Backpropagation*, di mana data *input* diolah melalui *hidden layer* 1 dan 2 untuk menghasilkan *output* akhir.

3.1.8 Backward Propagation

Proses *Backward Propagation* pada indeks ke 0 menghasilkan luaran seperti yang ditampilkan pada Gambar 3.10

```

Backward propagation for data index 0:
Updated w1: [[-1.43864819  0.1824416  1.51033818 -2.71575098  1.59180275 -0.10831773
-1.48981224 -2.32775094 -0.59895734  3.88439829  2.55351601  4.1815977
3.01889126  3.25952105 -3.61996847  3.06683057 -1.59578309 -3.39361076
-1.79399646  2.30730633 -3.01579764  2.66280095 -1.34658445 -4.08204221
3.63165893  2.97455515  3.54240387  3.12676418  1.63744504 -2.66535117
-4.69242441  2.49283706 -2.87265478 -0.24321871  1.98245399  1.7295836
3.02354215 -2.49757747  2.22879949 -3.60757514 -2.6563975  1.31387137
2.79392406 -1.01603703 -2.69059897  1.01555038  2.92866365 -2.13316721
-3.59653385  2.49420905 -3.08954942 -3.36532112 -1.72531469 -1.81686916
-2.03092493  1.6083958 -2.72485438 -1.89430298 -2.69004575  5.94462086
2.37544193 -4.71025776  3.63077972 -3.48549919  2.17366733  2.56356832
4.50576152 -3.29016756 -3.94439291 -2.44626475 -3.27993727 -1.7548241
2.9669648  3.20979749 -4.01401254  1.32575815  4.19059926  3.40750349]]
Updated b1: [[-0.39830559 -2.36759978 -0.17497371 -2.79998626 -0.89663248 -2.32799431
-0.94961755 -2.66524592 -2.37001461  1.12353349 -2.87505311  3.88198572
-3.93453193 -4.16278739 -3.40657722  1.13354737 -1.1013066 -0.04804397
-1.71738714 -1.71104214 -2.66947787  1.85789053 -1.15822405 -3.23486292
-0.82896852 -2.76446135  1.50004275 -1.49531746 -1.41372596 -3.05185351
-4.30612408 -1.1507377  2.61594287 -2.2244393  0.212711 -1.71792814
-3.00658474  1.43216895  0.69574114  3.12778091 -1.69514996 -1.76974914
-0.05052034 -1.45751887 -2.96504354 -1.71204731 -3.5313457 -1.20157029
4.32465417  0.15881414  3.05860663  3.65064933 -0.9364361  0.60486737
1.41919069 -0.53881872  1.46614514 -1.05534612 -0.53028197  5.87314853
-0.96906442 -4.5971768 -5.09684156  4.66679224 -2.41117575 -2.85702313
3.85024262  1.41009027 -1.97592654 -1.65956609  2.46346109  0.54186048
-3.65447339  1.81655207 -2.57886657 -0.60572038  2.7186815  3.01013859]]

```

Gambar 3.9 Backward Propagation

Gambar 3.9 menampilkan *output* dari proses *Backward* pada Jaringan Syaraf Tiruan untuk indeks 0, yang digunakan untuk memperbarui bobot dan bias berdasarkan kesalahan. Bobot dan bias diperbarui untuk mengurangi kesalahan dan meningkatkan akurasi melalui metode *Backpropagation*, memastikan jaringan dapat belajar dari data dan meningkatkan klasifikasi berdasarkan *input* yang diberikan.

3.1.9 Data Kelas, Aktual, Klasifikasi

Data kelas/label, aktual, dan klasifikasi yang dihasilkan oleh permodelan dapat dilihat pada Gambar 3.10

T	Actual	Classification
1	Blas	Kresek
9	Busuk Batang	Busuk Batang
5	Bercak Garis	Bercak Garis
0	Tungro	Kresek
7	Kerdil Hampa	Kerdil Hampa
5	Bercak Garis	Bercak Garis
1	Blas	Kresek
0	Tungro	Kresek
3	Bercak Daun Cokelat	Bercak Daun Cokelat
3	Bercak Daun Cokelat	Bercak Daun Cokelat

Gambar 3.10 Data Kelas, Aktual, Klasifikasi

Gambar 3.10 menampilkan tabel dengan tiga kolom dan 109 baris, yang berisi informasi kelas ("T"), nilai aktual ("Actual"), dan nilai yang diklasifikasikan ("Classification") oleh model dengan menampilkan nama jenis penyakit tanaman padi.

3.1.10 Evaluasi

Nilai evaluasi yang didapat dari permodelan ditampilkan bentuk matrik evaluasi akurasi, presisi dan *recall* yang dapat dilihat pada Gambar 3.11

Average Accuracy: 0.9670932358318098
 Average Precision: 0.9340909090909091
 Average Recall: 0.9545454545454546

Gambar 3.11 Hasil Evaluasi

Gambar 3.11 menampilkan hasil evaluasi dari pemodelan *Backpropagation* dan *Bee Colony Optimization* dengan akurasi sebesar 96,7%, presisi 93,4%, dan *recall* 95,4%.

3.2 Hasil Pengujian

Pada Algoritme *Backpropagation*, terdapat tiga parameter utama, yaitu *Learning Rate* (LR), maksimum iterasi (epoch), dan *hidden layer*. Parameter tersebut akan diuji untuk mendapatkan nilai optimal, pengujian juga dilakukan terhadap pembagian rasio data, *K-Fold*, jumlah koloni dan iterasi sehingga didapatkan hasil akhir yang lebih baik.

3.2.1 Parameter Awal *Backpropagation*

Parameter awal *Backpropagation*, rasio pembagian data, dan k-fold yang digunakan pada pemodelan ini didapatkan dari artikel yang terkait dengan penelitian, berikut merupakan nilai parameter, rasio, dan *K-Fold* yang digunakan.

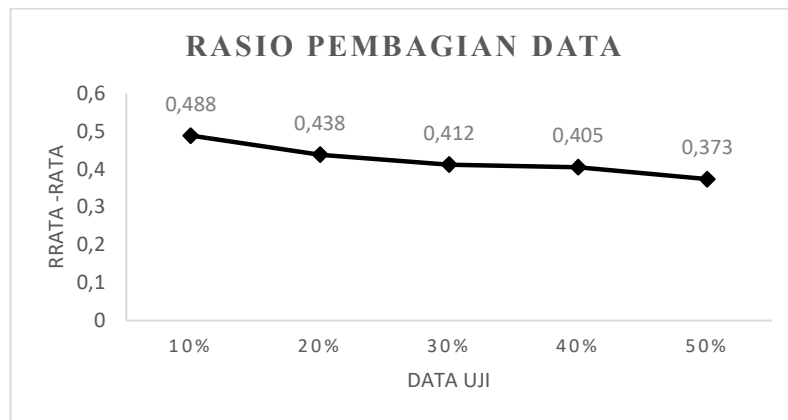
Tabel 3.1 Parameter Awal *Backpropagation*

Split Data	K-Fold	LR	Hidden 1	Hidden 2	Epoch	Akurasi	Presisi	Recall	F-1 Score
0,1	3	0,01	100	50	1000	42%	29%	42%	31%

Tabel 3.1 menunjukkan parameter awal yang digunakan dengan rasio data *training* 90% dan data *testing* 10%, *K-Fold*=3, *epoch* 1000, *learning rate* 0,01, *hidden layer* 1 = 100, *hidden layer* 2 = 50, didapatkan hasil akurasi 42%, presisi 29%, *recall* 42%, dan F-1 Score 31%.

3.2.2 Rasio Pembagian Data

Untuk mengawali proses uji coba, dilakukan pengujian pada rasio pembagian data yang berbeda, yakni 10%, 20%, 30%, 40%, dan 50%. Pada tahap pengujian ini, nilai parameter yang digunakan sesuai dengan Tabel 3.1.

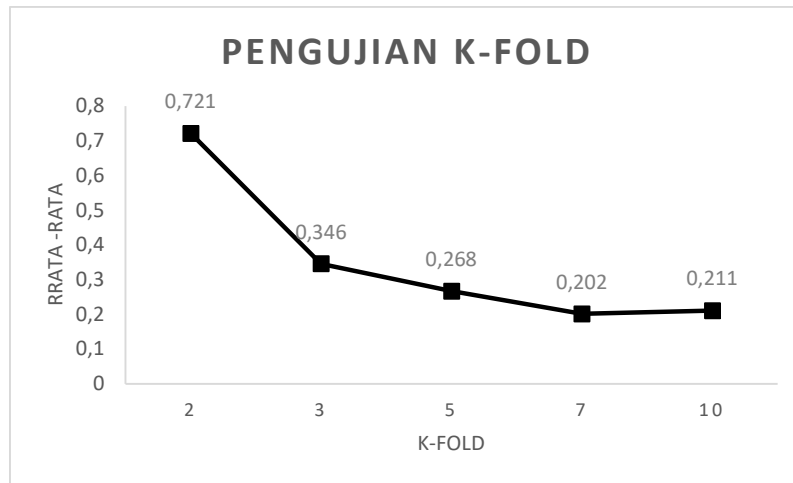


Gambar 3.12 Rasio Pembagian Data

Berdasarkan Gambar 3.12 yang memiliki akurasi tinggi adalah rasio pembagian data 10% untuk data *testing* dan 90% untuk data *training* dengan akurasi 0,488 atau 49%, maka untuk pengujian berikutnya menggunakan rasio pembagian data 90:10.

3.2.3 K-Fold

Pengujian dilakukan menggunakan $K=2,3,5,7,10$. Setiap nilai K dilakukan pengujian sebanyak 5 kali menunjukkan bahwa akurasi model menurun seiring dengan peningkatan nilai K .

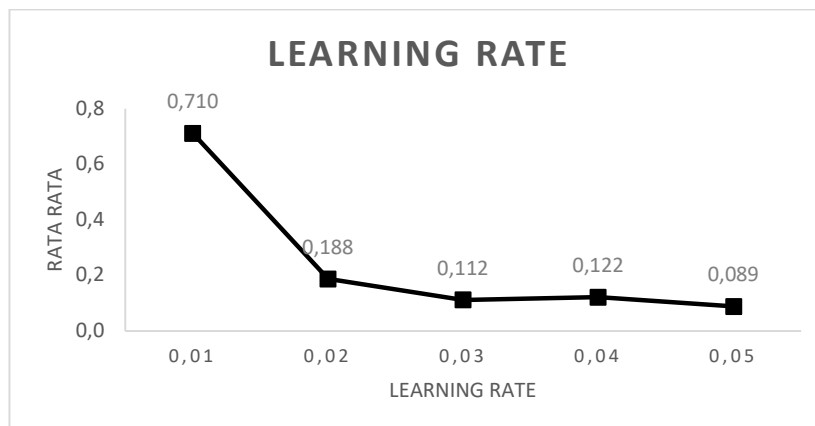


Gambar 3.13 K-Fold

Berdasarkan Gambar 3.13 pengujian K-Fold didapatkan nilai akurasi tertinggi terletak pada $K=2$ sebesar 0,721 atau 72%, sehingga nilai K yang digunakan pada pengujian selanjutnya adalah 2.

3.2.4 Learning Rate

Pengujian ini digunakan *learning rate* berkisar antara 0,01-0,05. Hasil dari pengujian *learning rate* dapat dilihat pada Gambar 3.14.

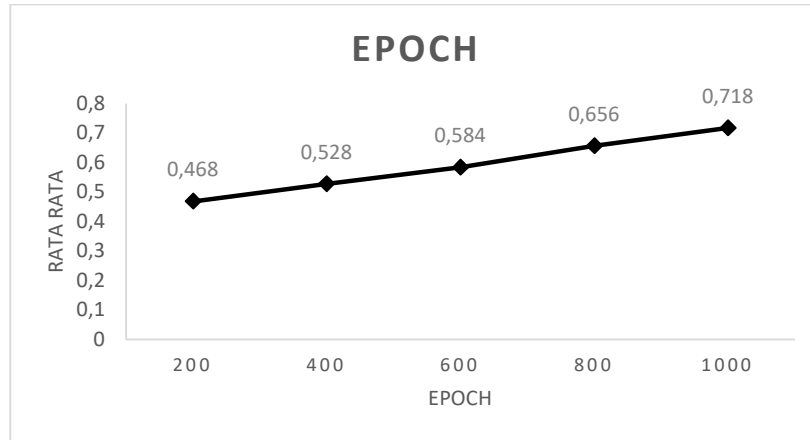


Gambar 3.14 Learning Rate

Berdasarkan Gambar 3.14 hasil tertinggi yang didapatkan pada pengujian di atas yaiu dengan nilai *learning rate* 0.01, menghasilkan akurasi sebesar 0,710 atau 71%. Hasil tersebut akan digunakan untuk membuat model pengujian selanjutnya.

3.2.5 Epoch

Pengujian dilakukan untuk mengetahui pengaruh jumlah langkah pelatihan (*epoch*) yang digunakan dalam proses pelatihan model *neural network*. Untuk membandingkan hasilnya, pengujian dilakukan dengan menggunakan jumlah *epoch* yang berbeda, yaitu 200, 400, 600, 800, dan 1000.

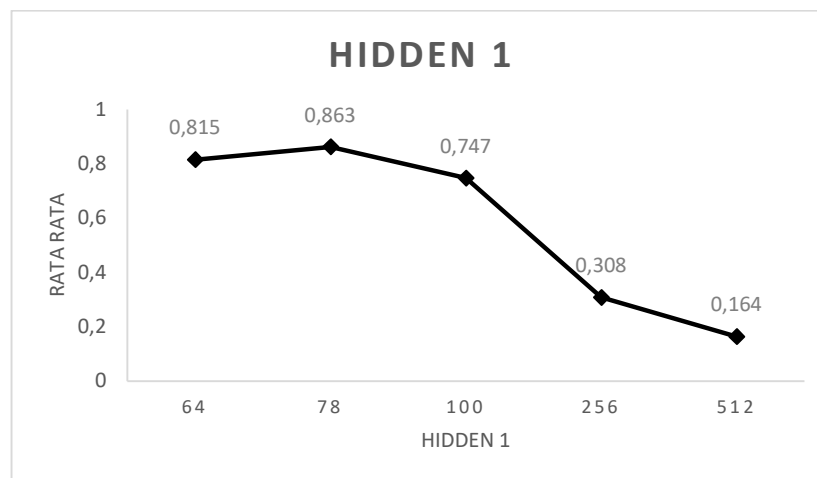


Gambar 3.15 Epoch

Hasil pengujian pada Gambar 3.15 menunjukkan bahwa semakin tinggi jumlah *epoch*, semakin baik tingkat akurasi yang dicapai dengan akurasi tertinggi pada *epoch* 1000 sebesar 71%.

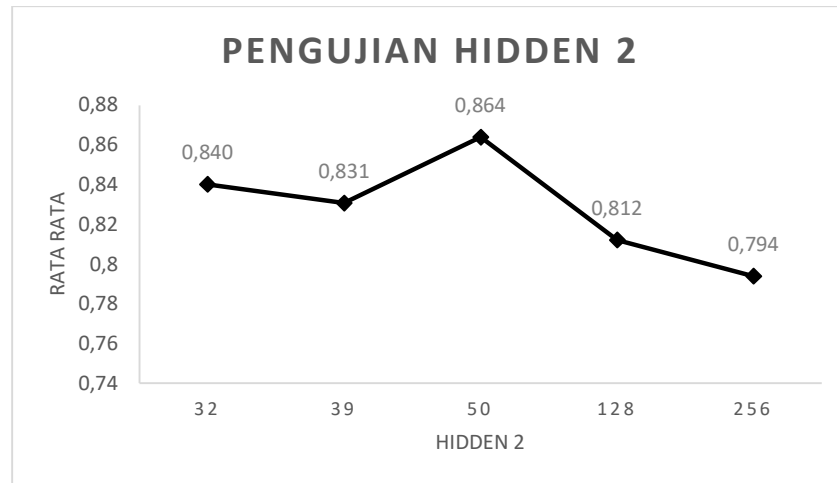
3.2.6 Hidden Layer 1 & 2

Pada pengujian ini dilakukan pada *hidden layer* 1 dan 2, pada *hidden layer* 1 digunakan *neuron* 64, 78, 100, 256, dan 512. *Hidden layer* 2 digunakan *neuron* 32, 39, 50, 128, dan 256.



Gambar 3.16 Hidden Layer 1

Berdasarkan Gambar 3.16 didapatkan bahwa akurasi tertinggi dicapai dengan menggunakan 78 *neuron* di *hidden layer* 1, yaitu sebesar 0,863 atau 86%. Oleh karena itu, jumlah *neuron* optimal untuk *hidden layer* 1 adalah 78.



Gambar 3.17 *Hidden Layer 2*

Dari gambar 3.17 diketahui bahwa akurasi tertinggi dicapai dengan menggunakan 50 *neuron* pada lapisan *hidden layer* ke-2, yaitu sebesar 0,864 atau 86%. Dengan demikian, jumlah *neuron* yang optimal untuk lapisan *hidden layer* ke-2 adalah 50.

3.2.7 Parameter Akhir *Backpropagation*

Setelah dilakukan pengujian terhadap parameter *Backpropagation*, rasio data dan K-Fold didapatkan hasil seperti Tabel 3.2

Tabel 3.2 Parameter Akhir *Backpropagation*

Split Data	K-Fold	LR	Hidden 1	Hidden 2	<i>Epoch</i>	Akurasi	Presisi	Recall	F-1 Score
0,1	2	0,01	78	50	1000	85%	81%	86%	83%

Hasil akhir yang didapat dari pengujian yaitu, pembagian rasio data sebesar 0,1, *K-Fold*=2, *learning rate* 0,01, *hidden layer* 1 78 *neuron*, *hidden layer* 2 50 *neuron*, *epoch* 1000 dengan akurasi 85%, presisi 81%, *recall* 86%, F-1 Score 83%.

3.2.8 Parameter Awal *Bee Colony Optimization*

Setelah didapatkan parameter akhir *Backpropagation*, rasio pembagian data, K-fold terbaik, kemudian ditambahkan parameter *Bee Colony Optimization* yaitu jumlah koloni, dan iterasi untuk pemodelan *Backpropagation* dan *Bee Colony Optimization*

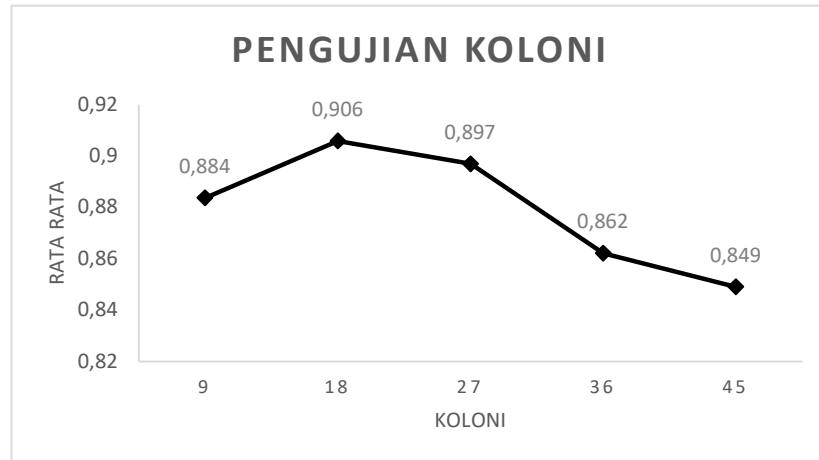
Tabel 3.3 Parameter Awal *Bee Colony Optimization*

Colony	Iterasi	Akurasi	Presisi	Recall	F-1 Score
9	5	90%	87%	91%	88%

Berdasarkan Tabel 3.3 jumlah koloni yang digunakan yaitu 9, dan iterasi 5 mendapatkan hasil akurasi 90%. Untuk didapatkan hasil yang lebih baik maka dilakukan pengujian terhadap 2 parameter *Bee Colony Optimization*.

3.2.9 Koloni

Pada pengujian dilakukan dengan beberapa jumlah koloni yaitu 9, 19, 27, 36, 45 dan didapatkan hasil yang ditampilkan pada Gambar 3.18.

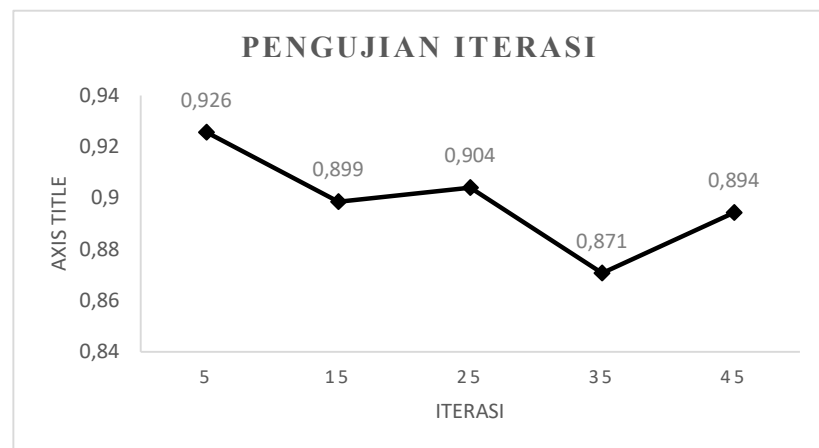


Gambar 3.18 Koloni

Berdasarkan Gambar 3.18 pengujian koloni didapatkan hasil akurasi tertinggi pada jumlah koloni 18 dengan akurasi sebesar 0,906 atau 91%.

3.2.10 Iterasi

Pada pengujian ini dilakukan sebanyak 5 nilai iterasi yaitu 5, 15, 25, 35, dan 45. Hasil pengujian dapat dilihat pada Gambar 3.19.



Gambar 3.19 Iterasi

Gambar 3.19 menunjukkan hasil pengujian iterasi didapatkan hasil tertinggi pada iterasi 5 dengan akurasi sebesar 0,926 atau 93%.

3.2.11 Parameter Akhir *Bee Colony Optimization*

Setelah dilakukan pengujian terhadap parameter *Bee Colony Optimization* yaitu jumlah koloni dan iterasi didapatkan hasil terbaik yang ditampilkan pada Tabel 3.4

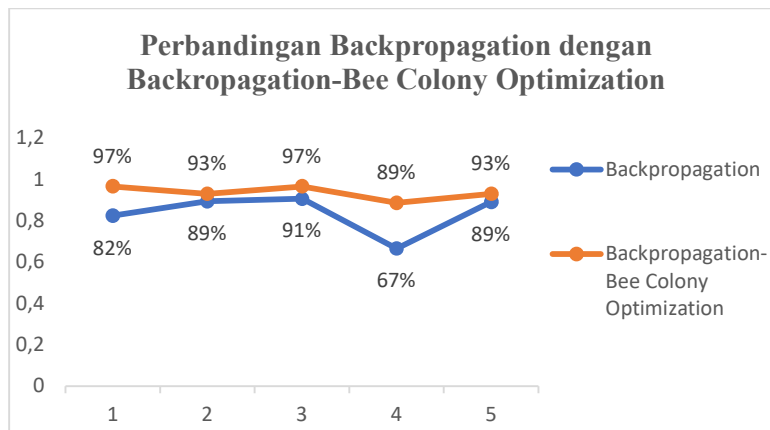
Tabel 3.4 Parameter Akhir *Bee Colony Optimization*

Colony	Iterasi	Akurasi	Presisi	Recall	F-1 Score
18	5	94%	91%	94%	92%

Tabel 3.4 menunjukkan parameter *Bee Colony Optimization* terbaik yang dihasilkan dari pengujian yaitu jumlah koloni 18 dan iterasi 5 mendapat akurasi sebesar 94%

3.2.12 Perbandingan Akurasi

Untuk mengetahui perbandingan akurasi terhadap metode *Backpropagation* dengan *Backpropagation – Bee Colony Optimization* program dijalankan sebanyak 5 kali pada masing – masing metode dan hasilnya dapat dilihat pada Gambar 3.20.



Gambar 3.20 Perbandingan Akurasi

Gambar 3.20 menunjukkan bahwa akurasi yang didapatkan dengan metode *Backpropagation-Bee Colony Optimization* selalu lebih tinggi dibandingkan dengan penggunaan metode *Backpropagation*.

3.3 Pembahasan

Penelitian ini menggunakan data penyakit tanaman padi sebanyak 1094 data yang diperoleh dengan melakukan wawancara dan memberikan kuesioner yang berisi nama dan gejala penyakit tanaman padi. Untuk mengklasifikasikan penyakit pada tanaman padi, dibuatlah model klasifikasi menggunakan metode *Backpropagation*. Tahapan yang dilakukan pada metode ini meliputi inialisasi bobot dan parameter, *Forward Propagation*, *Backward Propagation*, dan pembaruan bobot.

Parameter yang digunakan dalam *Backpropagation* meliputi *Learning Rate*, *Epoch*, dan *Hidden Layer*. Setelah melakukan pengujian sebanyak lima kali untuk setiap parameter, hasil terbaik dicapai dengan rasio data pengujian 10%, *K-Fold 2*, *Learning Rate* 0,01, *Epoch* 1000, serta *Hidden Layer* pertama dengan 78 neuron dan *Hidden Layer* kedua dengan 50 neuron. Dengan parameter tersebut, model mencapai akurasi rata-rata sebesar 84%. Selanjutnya, metode ini dioptimasi menggunakan

algoritma *Bee Colony Optimization* dengan jumlah koloni sebanyak 18 dan iterasi sebanyak 5, akurasi meningkat hingga 94%

BAB IV

PENUTUP

4.1 Kesimpulan

Pada penelitian klasifikasi jenis penyakit tanaman padi menggunakan metode *Backpropagation* dengan optimasi algoritme *Bee Colony Optimization* ini dapat disimpulkan sebagai berikut :

1. Metode klasifikasi penyakit pada tanaman padi dengan menggunakan *Backpropagation* berhasil mencapai akurasi sebesar 84%. Proses ini melibatkan beberapa tahap penting, yaitu inialisasi bobot dan parameter, *Feedforward Propagation* untuk menghitung *output* berdasarkan *input*, *Backward Propagation* untuk memperbarui bobot guna meminimalisir *error*, serta *update* bobot. Selain itu, penentuan nilai bobot *Backpropagation* dapat dioptimalkan dengan menggunakan *Bee Colony Optimization*. Proses ini meliputi inialisasi populasi, evaluasi solusi, pencarian solusi baru oleh *employee bee*, *onlooker bee*, dan *scout bee* hingga didapatkan solusi terbaik. Hasilnya adalah bobot awal yang lebih baik, sehingga meningkatkan kinerja dan efektivitas model dalam klasifikasi penyakit pada tanaman padi.
2. Terdapat perubahan cukup signifikan dalam nilai akurasi klasifikasi sebelum dan sesudah menggunakan tambahan metode optimasi *Bee Colony Optimization* (BCO). Akurasi yang didapat dengan menggunakan metode *Backpropagation* adalah 84%, setelah menerapkan optimasi *Bee Colony Optimization*, akurasi meningkat menjadi 94%. Ini menunjukkan bahwa penggunaan *Bee Colony Optimization* secara efektif meningkatkan kinerja model klasifikasi penyakit pada tanaman padi.

4.2 Saran

Penulis menyarankan untuk penelitian selanjutnya atau pengembangan dari studi ini adalah mempertimbangkan penambahan jenis penyakit yang diamati, mengeksplorasi metode yang menggunakan citra sebagai data masukan, atau menerapkan metode optimasi lainnya untuk meningkatkan kinerja dan relevansi hasil penelitian.

DAFTAR RUJUKAN

- Adiba, F., Akram Nur Risal, A. and Tahir, M. (2023) 'Implementasi Algoritme Backpropagation untuk Klasifikasi Kualitas Susu Sapi', *Jurnal MediaTIK: Jurnal Media Pendidikan Teknik Informatika dan Komputer*, 6(2). Available at: <https://fdc.nal.usda.gov/>.
- Agustina, R., Magdalena, R. and Pratiwi, N.K.C. (2022) 'Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16', *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(2), p. 446. Available at: <https://doi.org/10.26760/elkomika.v10i2.446>.
- Arfiani, I., Yuliansyah, H. and Suratin, M.D. (2022) 'Implementasi Bee Colony Optimization Pada Pemilihan Centroid (Klaster Pusat) Dalam Algoritme K-Means', *Building of Informatics, Technology and Science (BITS)*, 3(4), pp. 756–763. Available at: <https://doi.org/10.47065/bits.v3i4.1446>.
- Argina, A.M. (2020) 'Penerapan Metode Klasifikasi K-Nearest Neighbor pada Dataset Penderita Penyakit Diabetes', *Indonesian Journal of Data and Science*, 1(2), pp. 29–33.
- Cervantes, J. *et al.* (2020) 'A comprehensive survey on support vector machine classification: Applications, challenges and trends', *Neurocomputing*, 408, pp. 189–215. Available at: <https://doi.org/10.1016/j.neucom.2019.10.118>.
- Dwi Suhendra, C. and Chandra Saputra, A. (2020) 'PENENTUAN PARAMETER LEARNING RATE SELAMA PEMBELAJARAN JARINGAN SYARAF TIRUAN BACKPROPAGATION MENGGUNAKAN ALGORITME GENETIKA', *JURNAL TEKNOLOGI INFORMASI*, 14(2).
- Elvaningsih, H. *et al.* (2021) 'Prediksi Stok Obat Menggunakan Metode Backpropagation (Studi Kasus: Puskesmas Dumai Barat)', in *Seminar Nasional Sains dan Teknologi Informasi (SENSASI)*, pp. 228–232. Available at: <http://prosiding.seminar-id.com/index.php/sensasi/issue/archivePage|228>.
- Fitrianingsih Hasan, N., Kusriani, K. and Al Fatta, H. (2019) 'Analisis Arsitektur Jaringan Syaraf Tiruan Untuk Peramalan Penjualan Air Minum Dalam Kemasan', *JURTI*, 3(1).
- Habibi, M. and Winar Cahyo, P. (2020) 'Journal Classification Based on Abstract Using Cosine Similarity and Support Vector Machine', *JISKA*, 4(3), pp. 185–192.
- Hasanati, Z. *et al.* (2020) *Kajian Implementasi Jaringan Syaraf Tiruan Metode Backpropagation Untuk Deteksi Bau*, *JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)*. Available at: <http://journal.isas.or.id/index.php/JACOST>.
- Herdiansah, A. *et al.* (2022) 'Klasifikasi Citra Daun Herbal Dengan Menggunakan Backpropagation Neural Networks Berdasarkan Ekstraksi Ciri Bentuk', *JURIKOM (Jurnal Riset Komputer)*, 9(2), p. 388. Available at: <https://doi.org/10.30865/jurikom.v9i2.4066>.
- Herviany, M. *et al.* (2021) 'Perbandingan Algoritme K-Means dan K-Medoids untuk Pengelompokan Daerah Rawan Tanah Longsor di Provinsi Jawa Barat', *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 1, pp. 34–40.
- Huang, H.C. and Chuang, C.C. (2020) 'Artificial bee colony optimization algorithm incorporated with fuzzy theory for real-time machine learning control of articulated robotic manipulators', *IEEE Access*, 8, pp. 192481–192492. Available at: <https://doi.org/10.1109/ACCESS.2020.3032715>.

- Hutabarat, D. *et al.* (2021) ‘Penerapan Algoritme Backpropagation dalam Memprediksi Hasil Panen Tanaman Sayuran’, *BIOS : Jurnal Teknologi Informasi dan Rekayasa Komputer*, 2(1), pp. 21–29. Available at: <https://doi.org/10.37148/bios.v2i1.18>.
- Indra Borman, R., Ahmad, I. and Rahmanto, Y. (2022) ‘Klasifikasi Citra Tanaman Perdu Liar Berkhasiat Obat Menggunakan Jaringan Syaraf Tiruan Radial Basis Function’, *Bulletin of Informatics and Data Science*, 1(1). Available at: <https://ejournal.pdsi.or.id/index.php/bids/index>.
- Jayanti, K., Lumbanbatu, K. and Ramadani, S. (2021) ‘Memprediksi Jumlah Siswa Baru Menggunakan Metode Backpropagation (STUDI KASUS: SMK HARAPAN BANGSA KUALA)’, *JUKI : Jurnal Komputer dan Informatika*, 3(1).
- Jeraman, F., Faizah, N. and Koryanto, L. (2023) ‘Sistem Pakar Diagnosa Penyakit pada Tanaman Padi Kecamatan Satarmese Kabupaten Manggarai Provinsi Nusa Tenggara Timur Berbasis Web dengan Metode Forward Chaining’, *Computer Journal*, 1(1), pp. 73–81. Available at: <https://doi.org/10.58477/cj.v1i1.66>.
- Jinan, A. and Hayadi, B.H. (2022) ‘Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron)’, *Journal of Computer and Engineering Science*, 1(2).
- Julianto, A., Sunyoto, A. and Wibowo, F.W. (2022) ‘OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI (OPTIMIZATION OF CONVOLUTIONAL NEURAL NETWORK HYPERPARAMETERS FOR CLASSIFICATION OF RICE PLANT DISEASES)’, *TEKNIMEDIA*, 3(2). Available at: <https://www.kaggle.com/tedisetiady/leaf-rice-dis->.
- Kartika Delimayanti, M. *et al.* (2021) ‘Pemanfaatan Metode Multiclass-SVM pada Model Klasifikasi Pesan Bencana Banjir di Twitter’, *Edu Komputika Journal*, 8(1). Available at: <http://journal.unnes.ac.id/sju/index.php/edukom>.
- Khoiruddin, M., Junaidi, A. and Saputra, W.A. (2022) ‘Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network’, *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, 2(1), pp. 37–45. Available at: <https://www.kaggle.com/tedisetiady/leaf-rice-disease->.
- Marito Tondang, S., Satria Tambunan, H. and Susiani, S. (2022) ‘Penerapan Jaringan Saraf Tiruan dalam Memprediksi Impor Garam Menurut Negara Asal Menggunakan Metode Back-propagation’, *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, 1(3), pp. 2828–9099. Available at: <https://doi.org/10.55123/jomlai.v1i3.919>.
- Ningsih, L. *et al.* (2024) ‘Perbandingan Kinerja Algoritme Klasifikasi Status Mutu Air’, *Indonesian Journal of Multidisciplinary on Social and Technology*, 2(1), pp. 72–76. Available at: <https://doi.org/10.31004/ijmst.v2i1.298>.
- Pradema Sanjaya, U., Pribadi, T. and Prastya, I.W.D. (2022) ‘Klasifikasi Dana Hibah Usaha Mikro Kecil Dan Menengah dengan Metode Naïve Bayes’, *Indonesian Journal of Computer Science*, 11(3).
- Putri, A.W. (2021) ‘IMPLEMENTASI ARTIFICIAL NEURAL NETWORK (ANN) BACKPROPAGATION UNTUK KLASIFIKASI JENIS PENYAKIT PADA DAUN TANAMAN TOMAT’, *Jurnal Ilmiah Matematika*, 09(02).

- Rizka Yudana, F., Suyanto, M. and Nasiri, A. (2023) 'Model Klasifikasi Untuk Menentukan Kesiapan Kerja Mahasiswa Dan Kelulusan Tepat Waktu Dengan Metode Machine Learning', *IJITECH: Indonesian Journal of Information Technology*, 1(1). Available at: <https://doi.org/10.37680/ijitech.v1i1.xx>.
- Rolimarch Pratama, E. and Darmawan, J.B.B. (2021) 'KLASIFIKASI STATUS GIZI BALITA MENGGUNAKAN JARINGAN SYARAF TIRUAN BACKPROPAGATION', in *Riset dan Teknologi Terapan (RITEKTRA)*.
- Saputra, R.A. *et al.* (2021) 'Penerapan Algoritme Convolutional Neural Network Dan Arsitektur MobileNet Pada Aplikasi Deteksi Penyakit Daun Padi', *JURNAL SWABUMI*, 9(2), pp. 184–188. Available at: <https://archive.ics.uci.edu/ml/datasets/Rice>.
- Sari, N. *et al.* (2023) 'Pengabdian kepada Masyarakat: Pengenalan Penyakit Tanaman Padi dan Teknik Pengendaliannya di Desa Bentok Darat, Bati-bati, Kalimantan Selatan', *Lambung Inovasi: Jurnal Pengabdian kepada Masyarakat*, 8(2), pp. 232–243. Available at: <https://doi.org/10.36312/linov.v8i1.1130>.
- Satria, W. (2020) 'JARINGAN SYARAF TIRUAN BACKPROPAGATION UNTUK PERAMALAN PENJUALAN PRODUK (STUDI KASUS DI METRO ELECTRONIC DAN FURNITURE)', *Djtechno: Journal of Information Technology Research*, 1(1).
- Sethy, P.K. *et al.* (2020) 'Image Processing Techniques for Diagnosing Rice Plant Disease: A Survey', in *Procedia Computer Science*. Elsevier B.V., pp. 516–530. Available at: <https://doi.org/10.1016/j.procs.2020.03.308>.
- Sitompul, P., Okprana, H. and Prasetyo, A. (2022) 'Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201', *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, 1(2), pp. 2828–9099. Available at: <https://doi.org/10.55123/jomlai.v1i2.889>.
- Sulistyaningsih, S. and Muhlis, A. (2022) 'Pengendalian Hama Penyakit Pada Tanaman Padi Dengan Penggunaan Pestisida Alami Di Desa Mlandingan Wetan Kecamatan Mlandingan Kabupaten Situbondo', *Jurnal Media Abdimas*, 1(3). Available at: <https://doi.org/10.37817/mediaabdimas.v1i3>.
- Syafitri, W., Widi Nurcahyo, G. and Sumijan, S. (2020) 'Sistem Pakar Diagnosis Penyakit Tanaman Padi Menggunakan Metode Forward Chaining (Studi Kasus Di Balai Benih Induk Padi Dharmasraya)', *JURNAL SAINS DAN INFORMAIKA*, 6, pp. 31–38. Available at: <https://doi.org/10.22216/jsi.v6i1.5212>.
- Tangkelayuk, A. and Mailoa, E. (2022) 'Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree', *Jurnal Teknik Informatika dan Sistem Informasi*, 9(2), pp. 1109–1119. Available at: <http://jurnal.mdp.ac.id>.
- Yang, Y. and Liu, D. (2023) 'A Hybrid Discrete Artificial Bee Colony Algorithm for Imaging Satellite Mission Planning', *IEEE Access* [Preprint]. Available at: <https://doi.org/10.1109/ACCESS.2023.3269066>.
- Yunial, A.H. (2020) 'Analisis Optimasi Algoritme Klasifikasi Support Vector Machine, Decision Trees, dan Neural Network Menggunakan Adaboost dan Bagging', *Jurnal Informatika Universitas Pamulang*, 5(3), p. 247. Available at: <https://doi.org/10.32493/informatika.v5i3.6609>

LAMPIRAN

Lampiran 1 Surat Izin Penelitian



UMKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi

Telp. 0541-748511 Fax. 0541-766832

Website <http://informatika.umkt.ac.id>

email: informatika@umkt.ac.id



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 003-015/FST.1/D.3/C/2024

Lampiran : -

Perihal : **Permohonan Pengambilan Data**

Kepada Yth.
Kepala Dinas Pertanian Kota Samarinda
di -

Tempat

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Aamiin.

Sehubungan untuk memenuhi Tugas Akhir/Skripsi Tahun Akademik 2023/2024, maka dengan ini kami bermaksud untuk melakukan pengambilan data di Dinas Pertanian Kota Samarinda. Adapun data yang diminta yaitu validasi data penyakit tanaman padi, dengan nama mahasiswa sebagai berikut:

No	Nama	NIM
1	Azelina Zahra Riadini	2011102441195
2	Highness Mailani Putri	2011102441158
3	Muhammad Hafizh Atthoriq	2011102441184
4	Sri Mar'ati Sholikhah	2011102441199
5	Zulkarnaen	2011102441206

Demikian surat permohonan ini dibuat. Atas perhatiannya dan kerjasamanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 16 Syawal 1445 H
25 April 2024 M

Ketua Program Studi S1 Teknik Informatika



Arbansyah, S.Kom., M.TI
NIDN. 1118019203

Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda

Lampiran 2 Balasan Surat Izin Penelitian



PEMERINTAH KOTA SAMARINDA
DINAS KETAHANAN PANGAN DAN PERTANIAN

Jl. Biola No. 1 No. Telp. (0541) 743114 Fax. (0541) 746827
 S A M A R I N D A

LEMBAR DISPOSISI	
Surat Dari : <i>EMKI</i> No. Surat : <i>003.0-15/FS.1/D3/C/2024</i> Tgl. Surat : <i>25 APRIL 2024</i> Perihal : <i>PERMOHONAN PENCAMBIHAN DATA</i>	Diterima Tgl : <i>26 04 2024</i> No. Agenda : <i>137</i> <input type="checkbox"/> Sangat Segera <input type="checkbox"/> Segera <input type="checkbox"/> Rahasia
Diteruskan Kepada Sdr : <input type="checkbox"/> Sekretaris <input type="checkbox"/> Kabid SAPRAS <input type="checkbox"/> Kabid Ketersediaan & Distribusi Pangan <input type="checkbox"/> Kabid Konsumsi & Keamanan Pangan <input type="checkbox"/> Kabid Penyuluh <input type="checkbox"/> Kabid Peternakan & Kesehatan Hewan <input checked="" type="checkbox"/> Kabid Tanaman Pangan, Hortikultura dan Perkebunan	Dengan Hormat Harap : <input checked="" type="checkbox"/> Tindak Lanjut <input type="checkbox"/> Proses Sesuai Prosedur <input type="checkbox"/> Koordinasi / Konfirmasi <input type="checkbox"/> Konsultasi Dengan Kadis <input type="checkbox"/> Siapkan Telaahan Staf <input type="checkbox"/> Siapkan Draf / Konsep / Jawaban <input type="checkbox"/> Kordinir Kegiatan Ini <input type="checkbox"/> Dampingi Kadis <input type="checkbox"/> Buat Surat Perintah Tugas (SPT) <input type="checkbox"/> Sosialisasikan Kepada Seluruh Staf <input type="checkbox"/> Bahan Evaluasi dan Monitoring <input type="checkbox"/> Mewakili Kadis
Catatan : <i>Di bantu.</i> Kadis <i>[Signature]</i> <i>26/4/24</i>	Catatan : Sekretaris, Kabid <i>Tolong di bantu data tsb</i> <i>Sdr. Wawan</i> <i>[Signature]</i> <i>26/4.24</i>

P1 TUNGRO

Kode	Gejala	D1	D2	D3	D4	D5	D6
G1	Daun kuning kemerah merahan	1	0	1	1	0	1
G2	Daun muda menjadi belang/garis hijau pucat	1	0	1	1	1	1
G3	Bulir bercak cokelat dan beratnya kurang dibanding normal	1	0	1	1	1	1
G4	Kerdil	1	1	1	1	1	1
G5	Jumlah anakan sedikit berkurang	1	1	1	0	1	0
G6	Pertumbuhan akar tidak sempurna	1	1	0	1	0	0

P2 BLAST

Kode	Gejala	D1	D2	D3	D4	D5	D6
G7	Bercak berbentuk belah ketupat	1	0	1	0	0	1
G8	Bercak cokelat kehitaman pada batang	1	1	1	0	1	1
G9	Batang mudah patah	1	1	1	1	1	1
G10	Malai hampa atau tidak berisi	1	1	1	1	1	1

P3 KRESEK

Kode	Gejala	D1	D2	D3	D4	D5	D6
G11	Daun bewarna putih kekuningan	1	1	1	0	1	1
G12	Layu	1	0	1	0	0	1
G13	Batang bewarna cokelat	1	0	1	1	1	1
G14	Daun garis memanjang atau oval	1	1	1	1	1	1

P4 BERCAK DAUN COKELAT

Kode	Gejala	D1	D2	D3	D4	D5	D6
G15	Bercak muda berbentuk bulat kecil	0	1	0	0	1	1
G16	Bercak berwarna cokelat gelap	0	1	0	1	1	1
G17	Bercak tua berukuran lebih besar (0,4 – 1 cm x -1 – 02 cm)	0	1	1	1	1	0
G18	Bercak berwarna kuning di sekelilingnya	1	1	1	1	0	1
G19	Bulir berwarna cokelat kehitaman	1	1	1	1	1	1

P5 BERCAK COKELAT SEMPIT

Kode	Gejala	D1	D2	D3	D4	D5	D6
G12	layu	1	1	0	1	1	1
G20	Pada daun dan pelepah daun terdapat bercak cokelat yang sempit	1	1	0	1	1	1
G21	Varietas yang tahan bercak berukuran 0,2 – 1 cm x 0,1 cm, berwarna cokelat gelap	1	0	1	1	1	1
G22	Varietas bercak lebih besar & berwarna cokelat terang	1	1	1	0	1	0

P6 BERCAK GARIS

Kode	Gejala	D1	D2	D3	D4	D5	D6
G23	Muncul garis yang kebasah – basahan diantara urat daun	1	1	0	1	0	1
G24	Garis memanjang dan menjadi cokelat dengan lingkaran kuning di sekelilingnya	1	1	0	1	1	1
G25	Berlendir	0	0	1	0	1	1

G26	Lendir yang kering berbentuk butiran kecil pada garis luka	0	1	1	0	1	0
-----	--	---	---	---	---	---	---

P7 HANGUS PALSU

Kode	Gejala	D1	D2	D3	D4	D5	D6
G12	Layu	0	0	1	1	0	1
G27	Bulir padi menjadi gumpalan spora yang ukurannya sampai 1 cm	1	0	1	0	1	1
G28	Gumpalan spora menjadi hijau gelap	1	1	1	0	1	1
G29	Daun yang menguning menjadi kering	1	1	1	1	1	1

P8 KERDIL HAMPA

Kode	Gejala	D1	D2	D3	D4	D5	D6
G4	Kerdil	1	0	0	1	0	1
G30	Daun jadi kasar, tidak teratur	1	1	0	1	0	1
G31	Bulir padi hanya sedikit yang berisi	1	1	1	1	1	1
G32	Daun menguning dan terpinil	1	1	1	1	1	1

P9 KERDIL RUMPUT

Kode	Gejala	D1	D2	D3	D4	D5	D6
G4	Kerdil	0	1	1	0	1	1
G22	Varietas bercak lebih besar & berwarna coklat terang	1	1	0	0	1	1
G34	Daun berwarna kuning / tetap hijau	1	1	1	1	1	1
G33	Tetap berbunga, tapi bulir padi tidak berisi	1	1	0	1	1	0

P10 BUSUK BATANG

Kode	Gejala	D1	D2	D3	D4	D5	D6
G12	Layu	0	1	1	1	1	1
G36	Pelepah daun terlihat bercak basah berbentuk bulat	1	1	0	0	1	1
G37	Bercak pada bagian tengah berwarna abu-abu & bagian tepi berwarna coklat	1	1	1	1	1	1
G38	Pembusukan batangnya dari pangkal hingga atas	1	1	1	1	1	0

P11 KERDIL KUNING

Kode	Gejala	D1	D2	D3	D4	D5	D6
G4	Kerdil	1	1	0	1	0	1
G5	Jumlah anakan sedikit berkurang	1	1	1	1	0	1
G35	Bulir padi tidak berisi	1	1	1	1	1	1
G39	Warna daun dari kuning kehijauan ke kuning keputihan	1	1	1	0	1	1

Muga kerdil
Sangat muga
Muga



Lampiran 3 Kartu Bimbingan

KARTU KENDALI BIMBINGAN SKRIPSI

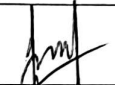






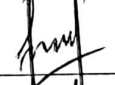

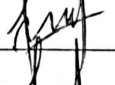




Nama Mahasiswa : Azelina Zahra Riadini



NIM : 2011102441195

Nama Dosen Pembimbing : Fendy Yulianto, S.Kom., M.Kom.

Judul Penelitian : Klasifikasi Jenis Penyakit Tanaman Padi Menggunakan Metode

Backpropagation dengan Optimasi Algoritma Bee Colony Optimization (BCO)

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	3 Februari 2024	Konsultasi pertama mengenai, data, metode yang akan digunakan, dan judul	
2	7 Februari 2024	Konsultasi mengenai penulisan Bab 1	
3	19 Februari 2024	Revisi penulisan kalimat dan paragraf Bab 1	
4	26 Februari 2024	Revisi kalimat dan paragraf pada Bab 1	
5	13 maret 2024	Revisi Bab 2 terkait objek penelitian dan dasar teori	
6	19 Maret 2024	Konsultasi data yang akan digunakan untuk penelitian	
7	4 April 2024	Revisi bab 2 dan menambahkan dasar teori	
8	21 April 2024	Menambahkan data yang kurang	
9	30 April 2024	Revisi Bab 1 dan Bab 2 dan menambahkan flowchart	
10	6 Mei 2024	Konsultasi perhitungan manual excel Backpropagation	
11	14 Mei 2024	Memperbaiki flowchart dan menambahkan teori tentang multiclass OVO, OVA dan evaluasi	
12	7 Juni 2024	Konsultasi terkait Program Backpropagation	
13	10 juni 2024	Konsultasi perhitungan manual excel Backpropagation dan metode optimasi	
14	15 juni 2024	Menambahkan sub-bab K-fold cross validation pada Bab 2	

15	20 Juni 2024	Menambahkan diagram alir pada bab 3	
16	25 Juni 2024	Menambahkan pengujian pada Bab 3 dan kesimpulan pada Bab 4	

Mengetahui

Dosen Pembimbing



Fendy Yulianto, S.Kom., M.Kom.

NIDN. 1102079402

Ketua Program Studi



Arbansyah, S.Kom, M.Ti

NIDN. 118019203

Lampiran 4 Wawancara dan Kuesioner Petani



Lampiran 6 Riwayat Hidup

DAFTAR RIWAYAT HIDUP



Azelina Zahra Riadini, dilahirkan pada tanggal 5 Agustus 2002 di Samarinda, Kalimantan Timur. Penulis adalah anak ketiga dari tiga bersaudara dari pasangan Bapak Hadi Santoso dan Ibu Sri Hartini. Penulis memulai Pendidikan di TK Enggang Putih Samarinda selama 1 tahun dan lulus pada tahun 2008. Kemudian melanjutkan Pendidikan di SDN 033 Samarinda dan lulus pada tahun 2014.

Penulis melanjutkan Pendidikan ke jenjang Sekolah Menengah Pertama di SMP Negeri 1 Samarinda selama 3 tahun dan lulus pada tahun 2017. Selanjutnya penulis melanjutkan Sekolah Menengah Atas di SMA Negeri 3 Samarinda selama 3 tahun dan lulus pada tahun 2020. Pada tahun yang sama, penulis mendaftar kuliah dan diterima di Universitas Muhammadiyah Kalimantan Timur, Jurusan Teknik Informatika.