

**PENERAPAN ALGORITME *FIREFLY* UNTUK OPTIMASI
METODE *BACKPROPAGATION* DALAM IDENTIFIKASI JENIS
PENYAKIT TANAMAN PADI**

SKRIPSI

Disusun Oleh :

Muhammad Hafizh Atthoriq

2011102441184



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

**PENERAPAN ALGORITMA *FIREFLY* UNTUK OPTIMASI METODE
BACKPROPAGATION DALAM IDENTIFIKASI JENIS PENYAKIT
TANAMAN PADI**

SKRIPSI

Diajukan Sebagai Salah Satu Persyaratan Untuk Memenuhi Gelar Sarjana Fakultas Sains Dan
Teknologi Universitas Muhammadiyah Kalimantan Timur

Disusun Oleh :

Muhammad Hafizh Atthoriq

2011102441184



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

LEMBAR PERSETUJUAN

**PENERAPAN ALGORITME *FIREFLY* UNTUK OPTIMASI METODE
BACKPROPAGATION DALAM IDENTIFIKASI JENIS PENYAKIT
TANAMAN PADI**

SKRIPSI

Diajukan oleh:

**Muhammad Hafizh
Atthoriq 2011102441184**

**Disetujui untuk
diujikan Pada
tanggal 17 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

**PENERAPAN ALGORITME *FIREFLY* UNTUK OPTIMASI METODE
BACKPROPAGATION DALAM IDENTIFIKASI JENIS PENYAKIT
TANAMAN PADI**

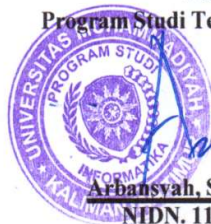
SKRIPSI

Diajukan oleh:
Muhammad Hafizh Atthoriq
2011102441184

Diseminarkan dan Diujikan
Pada tanggal 10 Juli 2024

Penguji I	Penguji II
 <u>Abdul Rahim, S.Kom., M.Cs</u> NIDN. 0009047901	 <u>Fendy Yulianto, S.Kom., M.Kom</u> NIDN. 1102079402

Mengetahui,
Ketua
Program Studi Teknik Informatika




Arbansyah, S.Kom., M.TI
NIDN. 1118619203

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Hafizh Atthoriq
NIM : 2011102441184
Program Studi : S1 Teknik Informatika
Judul Penelitian : Penerapan Algoritme *Firefly* untuk optimasi metode *Backpropagation* dalam identifikasi jenis Penyakit tanaman Padi

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, 10 Juli 2024

Yang membuat pernyataan



Muhammad Hafizh Atthoriq
2011102441184

ABSTRAK

Padi (*Oryza sativa*) merupakan sumber karbohidrat utama bagi masyarakat Indonesia dan mengalami peningkatan konsumsi seiring dengan pertumbuhan populasi. Untuk memenuhi kebutuhan ini, perluasan area tanam dan penerapan teknologi pertanian modern menjadi kunci dalam meningkatkan produksi padi nasional. Meskipun demikian, padi rentan terhadap berbagai penyakit yang dapat menurunkan produksi dan kualitas hasil pertanian. Penyakit tanaman padi seperti blast, hawar daun, busuk pelepah, tungro, bercak coklat, dan bercak bergaris menjadi ancaman serius yang memerlukan penanganan cepat dan tepat. Penelitian ini mengusulkan penggunaan sistem klasifikasi untuk menganalisis penyakit pada tanaman padi menggunakan Jaringan Syaraf Tiruan (JST) dengan metode Backpropagation. Namun, metode Backpropagation memiliki kelemahan, seperti waktu konvergensi yang lama dan sensitivitas terhadap kondisi awal bobot yang sering terjebak dalam minima lokal. Untuk mengatasi kelemahan ini, Firefly Algorithm (FA) digunakan sebagai teknik optimasi guna meningkatkan performa Backpropagation. Hasil pengujian metode BPP menunjukkan akurasi sebesar 43%. Sementara kombinasi metode BPP-FA menunjukkan akurasi 90%. Dengan demikian, kombinasi metode BPP-FA diharapkan dapat memberikan efektivitas dalam meningkatkan akurasi deteksi penyakit.

Kata kunci : Padi, Klasifikasi, *Backpropagation*, *Firefly*, penyakit

ABSTRACT

Rice (*Oryza sativa*) serves as the primary carbohydrate source for the Indonesian population, with its consumption rising in tandem with population growth. To satisfy the growing demand, it is crucial to expand the cultivation area and adopt modern agricultural technologies to increase national rice production. However, rice is vulnerable to a variety of diseases that can diminish yield and quality. Diseases such as blast, leaf blight, sheath blight, tungro, brown spot, and narrow brown spot present significant threats that necessitate prompt and accurate intervention. This study suggests utilizing a classification system to diagnose rice plant diseases through an Artificial Neural Network (ANN) employing the Backpropagation method. The Backpropagation method, however, has certain limitations, including lengthy convergence times and sensitivity to initial weight settings, which can lead to entrapment in local minima. To overcome these limitations, the Firefly Algorithm (FA) is used as an optimization technique to improve Backpropagation performance. Evaluation of the BPP method alone showed an accuracy of 43%, whereas the combined BPP-FA methods achieved an accuracy of 90%. Therefore, the combined BPP-FA approach is anticipated to significantly enhance the accuracy of disease detection.

Keyword: Rice, Classification, Backpropagation, Firefly, Disease

PRAKATA

Alhamdulillahirabbil'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 Abdul Syahid dan Ibu Ekawati Syahrial yang selalu mendo'akan dan memberikan semangat dan dukungan yang tak terhingga kepada penulis.
3. Bapak Ir. Sarjito, M.T.,Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur.
4. Bapak Arbansyah, S.Kom., M.TI selaku Kepala Program Studi Teknik Informatika
5. Bapak Fendy Yulianto, S.Kom., M.Kom selaku Dosen Pembimbing Skripsi yang senantiasa memberikan bantuan kepada penulis sehingga penulisan skripsi dapat selesai tepat waktu.

Samarinda, 10 Juli 2024

Penyusun



Muhammad Hafizh AtthorIQ

DAFTAR ISI

LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
PERYATAAN KEASLIAN PENELITIAN	v
ABSTRAK	vi
ABSTRACT	vii
PRAKATA	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
BAB II METODE PENELITIAN	4
2.1 Objek Penelitian	4
2.2 Data Penelitian	5
2.3 Jaringan Saraf Tiruan	5
2.3.1 Jaringan Lapisan Tunggal	5
2.3.2 Jaringan Banyak Lapisan	5
2.3.3 Jaringan Lapisan Kompetitif	6
2.4 K-Fold Cross Validation	6
2.5 Backpropagation	6
2.6 Multi-Class	7
2.7 Firefly	8
2.8 Alat dan Bahan	8
2.9 Evaluasi	9
2.9.1 Akurasi	9
2.9.2 Presisi	9
2.9.3 Recall	9

2.9.4 F1-score	10
2.10 Prosedur Penelitian	10
BAB III HASIL ANALISIS DAN PEMBAHASAN	12
3.1 Alur Flowchart	12
3.2 Implementasi Program	15
3.2.1 Pembagian data latih dan data uji	15
3.2.2 Normalisasi Data	15
3.2.3 K-Fold Cross Validation	15
3.2.4 Menginisialisasi bobot dan bias	17
3.2.5 Proses Optimasi Menggunakan Firefly	17
3.2.6 Epoch dan Loss	18
3.2.7 Forward Propagation dan Backward Propagation	19
3.2.8 Perbandingan Data Uji Dan Data Klasifikasi	20
3.2.9 Evaluasi	21
3.3 Hasil Pengujian	21
3.3.1 Parameter awal Backpropagation	21
3.3.2 Rasio Data	21
3.3.3 K-Fold	22
3.3.4 Learning Rate	22
3.3.5 Epoch	23
3.3.6 Hidden	23
3.3.7 Parameter Akhir Backpropagation	24
3.3.8 Parameter Awal Firefly Optimization	25
3.3.9 Jumlah Firefly	25
3.3.10 Iterasi Firefly	25
3.3.11 Parameter Akhir Firefly Optimization	26
3.3.12 Hasil Perbandingan	26
3.4 Hasil	27
BAB IV PENUTUP	28
4.1 Kesimpulan	28
4.2 Saran	28
Daftar Pustaka	29
LAMPIRAN	32

DAFTAR GAMBAR

2.1 Alur K-Fold.....	6
2.2 Alur Prosedur Penelitian	10
2.3 Alur Flowchart	12
3.1 Pengujian Pembagian Data	15
3.2 Pengujian Normalisasi Data.....	15
3.3 Pengujian Train Index K-Fold.....	16
3.4 Pengujian Test Index K-Fold	16
3.5 Pengujian Bobot dan bias sebelum optimasi.....	17
3.6 Pengujian Bobot dan bias setelah optimasi.....	18
3.7 Pengujian Nilai Keatraktifan Firefly.....	18
3.8 Pengujian Epoch dan Loss	19
3.9 Pengujian Forward Propagation.....	19
3.10 Pengujian Backward Propagation	20
3.11 Pengujian perbandingan Data	20
3.12 Hasil Pengujian Evaluasi	21
3.13 Gambar Grafik Pengujian Data.....	22
3.14 Grafik Pengujian K-Fold.....	22
3.15 Grafik Pengujian Learning Rate	23
3.16 Grafik Pengujian Epoch.....	23
3.17 Grafik Pengujian Hidden Layer 1	24
3.18 Grafik Pengujian Hidden Layer 2	24
3.19 Grafik Pengujian Jumlah Kunang-Kunang	25
3.20 Grafik Pengujian Iterasi Kunang-Kunang.....	26

DAFTAR TABEL

2.1. Data Penyakit Padi.....	4
2.2 Alat Penelitian.....	8
2.3 Bahan Penelitian	9
3.1 Parameter awal Backpropagation.....	21
3.2 Parameter Akhir Backpropagation	24
3.3 Parameter Awal Firefly Optimization	25
3.4 Parameter Akhir Firefly Optimization	26

DAFTAR LAMPIRAN

L1 Kuesioner Penelitian	32
L2 Surat Izin Penelitian	33
L3 Lembar Konsultasi	35
L4 Wawancara Dengan Petani.....	36
L5 Riwayat Hidup.....	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Padi (*Oryza sativa*), sebagai sumber karbohidrat utama bagi Masyarakat Indonesia, mengalami peningkatan konsumsi seiring dengan pertumbuhan populasi. Untuk memenuhi kebutuhan, perluasan area tanam dan penerapan teknologi pertanian modern menjadi kunci dalam meningkatkan produksi padi nasional (Christiawan *et al.*, 2023). Padi merupakan makanan utama yang diandalkan oleh masyarakat Indonesia karena kaya akan nutrisi sesuai dengan kebutuhan tubuh manusia (Hawar *et al.*, 2023).

Namun, di balik upaya peningkatan produksi terdapat berbagai tantangan yang harus dihadapi salah satu masalah yang timbul adalah serangan hama dan kondisi cuaca yang berubah-ubah, karena masalah ini penyakit tanaman padi muncul. Penyakit pada tanaman padi merupakan ancaman yang dapat menyebabkan penurunan produksi padi dan kualitas dalam pertanian (Nugroho and Nurraharjo, 2023). Ada beberapa penyakit yang menyerang pada tanaman padi seperti penyakit blast, penyakit hawar daun (*kresek*), penyakit busuk pelepah, penyakit tungro (*kerdil*), penyakit bercak coklat, penyakit bercak bergaris dan lain-lain (Sulistiyanto, Saputri and Noviyanti, 2022).

Untuk mengatasi penyakit pada tanaman padi, maka dibutuhkan konsep untuk mengatasi permasalahan tersebut. Konsep yang digunakan adalah sistem untuk menganalisis penyakit pada tanaman padi, Sistem untuk menganalisis penyakit pada tanaman padi adalah dengan konsep klasifikasi (Purwadi and Nasyuha, 2022). Klasifikasi dapat membantu masyarakat petani untuk mengetahui penyakit yang ada pada tanaman padi untuk mencegah penurunan produksi dan kerugian (Nugroho and Nurraharjo, 2023).

Salah satu metode yang efektif untuk melakukan proses klasifikasi adalah Jaringan Syaraf Tiruan, Jaringan Syaraf Tiruan merupakan sistem pengolahan informasi yang menyerupai struktur jaringan syaraf biologis. Jaringan Syaraf Tiruan ditentukan oleh tiga elemen utama, yaitu pola koneksi antar *neuron*, teknik penyetelan bobot penghubung, dan fungsi aktivasi (Ikhsan, Armansyah and Tamba, 2022). Jaringan Syaraf Tiruan mempunyai beberapa metode diantaranya *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), *Naïve Bayes*, dan *Backpropagation* (Gunawan and Putra, 2021).

Untuk menentukan jenis penyakit pada tanaman padi memerlukan pemahaman seperti penelitian yang dilakukan sebelumnya oleh Fakhri Habib Hawari, dengan metode *Convolutional* yang dilakukan pada tahun 2022, mengklasifikasi penyakit pada padi. Nilai akurasi yang di dapat sebesar 85% untuk data training, 86% untuk data *testing*, dan 95% untuk data *validation* dengan jumlah 10 *epoch* (Hawari *et al.*, 2022). Untuk penelitian selanjutnya dilakukan oleh Ery Murniyasih, dilakukan pada tahun 2020, dengan metode *Learning Vector Quantization* untuk Identifikasi Penyakit Padi Berdasarkan Bentuk Bercak Daun. Nilai akurasi yang di dapat sebesar 75% dan mempunyai Tingkat error 25%. Dengan ukuran piksel 95x35, *Learning rate* 0.02, EPS 0.001, pengurangan *alpha* 0.5, dan epoch sebanyak 5 kali (Murniyasih *et al.*, 2020).

Backpropagation merupakan jaringan *multilayer* karena memiliki banyak lapisan. Lapisan output berasal dari lapisan tersembunyi karena pola input digunakan sebagai pola pelatihan (Norhikmah and Rumini, 2020). Pada penelitian sebelumnya oleh Jaka Permadi, dilakukan pada tahun 2021, dengan metode *K-Nearest Neighbor* dan *Backpropagation Neural Network* dalam prediksi resiko diabetes tahap awal. Metode *Backpropagation* lebih baik dari pada metode *K-Nearest Neighbor*. Dengan nilai akurasi yang di dapat sebesar 90%, presisi 90%, dan *recall* 90% oleh metode *Backpropagation*. Sedangkan

metode KNN hanya mendapatkan nilai akurasi sebesar 83.75%, presisi sebesar 85.5497% dan recall sebesar 83.75% (Permadi, Rhomadhona and Aprianti, 2021).

Algoritme *Backpropagation* memiliki beberapa kelemahan seperti membutuhkan waktu lama untuk konvergensi dalam proses *training*, cukup sensitif dengan kondisi awal *weight* dan relatif sering terjebak pada lokal minimal yang bisa menggagalkan proses *training* (Bohari and Teknik, 2022). Maka untuk mengatasi kekurangan yang terdapat di *Backpropagation* di butuhkan teknik optimasi dengan mengubah fungsi pembelajaran untuk memperbaiki performa *Backpropagation*. Dengan menggunakan teknik optimasi tersebut, *Backpropagation* dapat memperoleh solusi yang lebih optimal dan akurat (Ridho *et al.*, 2023).

Terdapat beragam metode optimasi yang dapat di terapkan yaitu seperti *Particle Swarm Optimization* (PSO), *Genetic Algorithm* (GA), dan *Firefly Algorithm* (FA) yang dapat dimasukkan ketika proses pelatihan jaringan (Zurkarnain, Damayanti and Winarko, 2021). Algoritma *Firefly* adalah algoritma yang terinspirasi oleh perilaku gerombolan kunang-kunang dan telah menjadi salah satu algoritma paling populer yang digunakan untuk berbagai masalah optimasi. Algoritma ini termasuk baik dan memiliki kinerja yang tangguh dalam menyelesaikan masalah (Dewi, 2022).

Penggunaan metode *Firefly Algorithm* sebagai teknik optimasi dipilih karena kemungkinannya untuk mencapai nilai optimal dengan lebih cepat daripada algoritma Genetika dan PSO. Algoritma ini menawarkan potensi efisiensi yang lebih tinggi dalam proses pencarian solusi optimal dalam berbagai konteks optimasi (Sistem *et al.*, 2021). Pada penelitian sebelumnya yang di lakukan oleh ZHAO Xinhao, dilakukan pada tahun 2023, dengan metode *Firefly Optimization Algorithm* untuk melihat kondisi kesehatan pada baterai *Lithium*. Menunjukkan bahwa rentang fluktuasi kesalahan dapat berkurang 0,049% (Xinhao and Liang, 2023).

Penelitian ini bertujuan menerapkan algoritma optimasi *Firefly* dan metode *Backpropagation* pada Jaringan Saraf Tiruan (JST) guna mendeteksi penyakit pada tanaman padi. Penelitian ini juga mengevaluasi sejauh mana efektivitas algoritma *Firefly* dalam meningkatkan akurasi deteksi penyakit pada tanaman padi.

1.2 Rumusan masalah

Rumusan masalah dalam penelitian ini (i) Bagaimana hasil klasifikasi penyakit tanaman padi yang diperoleh melalui penggunaan metode *Backpropagation* dalam proses klasifikasi, (ii) Bagaimana dampak penerapan algoritma optimasi *Firefly* dalam memperbaiki pengoptimalan bobot pada algoritma *Backpropagation* untuk menentukan klasifikasi jenis penyakit pada tanaman padi?

1.3 Tujuan Penelitian

Tujuan penelitian dalam penelitian ini (i) Mengidentifikasi jenis penyakit padi dan mendapatkan hasil identifikasinya dengan bantuan metode *Backpropagation*, (ii) Untuk mengetahui seberapa pengaruh algoritma optimasi *Firefly* dalam mengoptimalkan bobot pada metode *Backpropagation*?

1.4 Manfaat Penelitian

Manfaat penelitian dalam penelitian ini (i) Dapat membantu mempercepat proses klasifikasi penyakit pada tanaman padi, (ii) Dengan adanya algoritma optimasi *Firefly* dan metode *Backpropagation*, algoritma ini dapat membantu meningkatkan akurasi dalam mengklasifikasi jenis penyakit pada tanaman padi.

1.5 Batasan Masalah

Batasan Masalah dalam penelitian ini (i) Penelitian ini fokus pada penggunaan algoritma *Firefly* untuk mengoptimalkan metode *Backpropagation*, (ii) Data yang digunakan didapat dari petani melewati wawancara mengenai penyakit pada tanaman padi.

BAB II

METODE PENELITIAN

2.1 Objek Penelitian

Salah satu penghambat dalam pertanian padi adalah hama dan penyakit yang menyerang pada tanaman padi. Namun pengetahuan para petani padi tentang hama dan penyakit serta pengendaliannya masih kurang. Sementara jumlah pakar tentang hama dan penyakit padi terbatas (Sholikhah, Kurniadi and Riansyah, 2021). Maka dari itu salah satu ilmu dalam bidang teknologi yang dapat menjadi solusi pada masalah tersebut yaitu ilmu Jaringan Syaraf Tiruan.

Pada penelitian ini Jaringan Syaraf Tiruan digunakan untuk mengklasifikasi penyakit pada tanaman padi. Untuk menghasilkan nilai klasifikasi yang bagus dipilihlah metode *Backpropagation* dengan algoritma optimasi *Firefly*. Data tanaman padi di ambil dari kuisioner kepada petani padi berdasarkan gejala-gejala yang terdapat di kota Samarinda. Adapun jenis-jenis penyakit dan gejala pada penyakit tanaman padi dapat dilihat pada Table 2.1.

Table 2.1 Data Penyakit Padi

Penyakit	Gejala
1. Tungro	<ul style="list-style-type: none"> - Daun kuning kemerah – merahan - Daun muda menjadi belang/garis hijau pucat - Bulir bercak cokelat dan beratnya kurang dibanding normal - Kerdil - Jumlah anakan sedikit berkurang
2. Blas	<ul style="list-style-type: none"> - Pertumbuhan akar tidak sempurna - Bercak berbentuk belah ketupat - Bercak cokelat kehitaman pada batang - Batang mudah patah
3. Kresek	<ul style="list-style-type: none"> - Malai hampa atau tidak berisi - 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
5. Bercak cokelat sempit	<ul style="list-style-type: none"> - Bulir berwarna cokelat kehitaman - 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	<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
-

2.2 Data Penelitian

Dalam penelitian ini, data diperoleh dari kuisisioner yang di ambil oleh beberapa petani di Samarinda. Data yang di ambil berjumlah 1000 data, terdiri dari jawaban dan tanggapan terhadap petani. Data ini berisi 11 penyakit dan gejala yang terdapat pada tanaman padi seperti pada contoh Tabel 2.1.

Untuk mendapatkan data yang akurat, dalam penelitian ini menggunakan Teknik wawancara kepada petani yang ada di Samarinda, Kec. Samarinda Utara, Lempake, Kec. Tenggarong Seberang, Kabupaten Kutai Kartanegara. Untuk mengamati kondisi tanaman padi dilakukan Observasi. Observasi dilakukan agar bisa mendapatkan informasi terkait penyakit yang ada pada tanaman padi melalui gejala-gejala yang ada. Dengan ada nya proses wawancara dengan petani, diharapkan dapat membentuk dasar yang kuat untuk klasifikasi penyakit pada tanaman padi menggunakan metode *Backpropagation* dengan algoritma optimasi *Firefly*.

2.3 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (*Artificial Neural Network*), adalah sistem pengolahan informasi yang dirancang untuk meniru cara kerja otak manusia dalam menyelesaikan masalah dengan memperbarui bobot koneksi sinopsisnya selama proses pembelajaran (Damanik *et al.*, 2022).

Jaringan Syaraf Tiruan sering kali menggunakan arsitektur jaringan yang digunakan secara luas dalam berbagai aplikasi. Berikut beberapa contoh dari arsitektur tersebut (Indrawan and Pandu Kusuma, 2021).

2.3.1 Jaringan Lapisan Tunggal

Jaringan ini terdiri dari satu lapisan masukan dan satu lapisan keluaran, dimana *neuron* pada lapisan masukan terhubung langsung ke *neuron* pada lapisan keluaran. Prosesnya langsung dari masukan ke keluaran tanpa melalui *hidden layer*.

2.3.2 Jaringan Banyak Lapisan

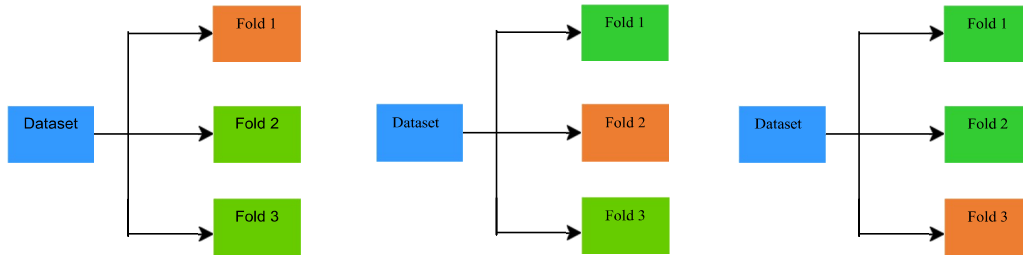
Jaringan dengan banyak lapisan memiliki karakteristik khusus yang terdiri dari tiga jenis lapisan, yaitu lapisan masukan, lapisan keluaran, dan *hidden layer*. Jaringan ini mampu menyelesaikan masalah yang lebih spesifik dibandingkan dengan jaringan yang hanya memiliki satu lapisan. Namun, proses pelatihan pada jenis jaringan ini cenderung lambat, dan menghabiskan waktu yang lebih lama.

2.3.3 Jaringan Lapisan Kompetitif

Pada fase ini, *neuron* dalam jaringan akan terhubung dan bersaing menjadi aktif dengan merespon pola input yang mirip.

2.4 K-Fold Cross Validation

K-Fold Cross Validation adalah metode di mana kumpulan data dibagi menjadi K bagian atau *fold*, dan setiap *fold* secara bergantian digunakan sebagai set pengujian pada berbagai titik dalam proses evaluasi model (Peryanto, Yudhana and Umar, 2020). Setiap kelompok secara bergantian digunakan sebagai data uji, sementara kelompok lainnya berfungsi sebagai data latih. Evaluasi kinerja algoritme dilakukan dengan menggunakan setiap kelompok sebagai data uji secara bergiliran, sementara data yang tersisa digunakan sebagai data latih (Adiba, Nur Risal and Tahir, 2023). Mari menggunakan validasi *2-fold* ($K = 2$). Kumpulan data dibagi menjadi 2 bagian *fold*. Pada iterasi pertama, bagian pertama digunakan untuk menguji model, sementara bagian lainnya digunakan untuk melatih model. Pada iterasi kedua, bagian kedua digunakan untuk data uji, sementara bagian lainnya digunakan untuk data latih. Contoh alur *k-fold* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur K-Fold

2.5 Backpropagation

Backpropagation adalah algoritma pembelajaran dalam Jaringan Saraf Tiruan. *Backpropagation* adalah proses pembelajaran dimana bobot-bobot dalam Jaringan Saraf Tiruan disesuaikan mundur berdasarkan nilai kesalahan yang terjadi selama proses pembelajaran (Ikhsan, Armansyah and Tamba, 2022). Langkah awal yang harus dilakukan adalah mengaktifkan *neuron* dengan menggunakan fungsi aktivasi serta dapat diubah-ubah seperti sigmoid berikut (Putri, 2021).

$$y = f(x) = \frac{1}{1+e^{-\sigma x}} \quad (1)$$

$$f'(x) = \sigma f(x)[1 - f(x)] \quad (2)$$

atau menggunakan tangen sigmoid

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

$$f'(x) = [1 + f(x)][1 - f(x)] \quad (4)$$

Langkah berikutnya adalah melakukan perhitungan feedforward dengan rumus sebagai berikut.

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (5)$$

Setiap data nilai masukan ($Z_j, j = 1, \dots, p$) yang dijumlahkan termasuk biasanya. Kemudian, hasil dari data tersembunyi dihitung menggunakan fungsi aktivasi $z_j = f(z_{in_j})$ langkah berikutnya adalah mengirimkan keluaran dengan rumus sebagai berikut.

$$y_{in_k} = w_{0k} + \sum_{i=1}^n z_j w_{jk} \quad (6)$$

Dimana setiap data nilai luaran ($y_k, k = 1, \dots, m$) yang di jumlahkan dengan bobot pada masukan sehabis itu dari data luaran menggunakan aktivasi $y_k = f(y_{in_k})$ yang selanjutnya di kirim ke seluruh luaran. (Zhao *et al.*, 2021)

Untuk melakukan penyesuaian bobot dan bias pada setiap keluaran, menggunakan cara sebagai berikut.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (7)$$

Untuk melakukan penyesuaian bobot dan bias pada setiap input, menggunakan cara sebagai berikut.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (8)$$

Untuk melakukan propagasi error dengan mempertimbangkan nilai vektor selisih yang dievaluasi melalui perulangan mundur $\tau = \tau - 1, \tau - 2, \dots, 1$ adalah

$$\delta_k^i(i) = [\sum_{l=1}^{\tau+1} \delta_k^{r+1}(l) W^{r+1}(li)] \frac{\partial f^\tau}{\partial n_k^\tau(i)} \quad (9)$$

Dalam algoritma proses *Backpropagation of error* terdapat beberapa langkah, yang pertama setiap nilai keluaran yang dihasilkan ($y_k, k = 1, 2, \dots, m$) di bandingkan dengan nilai target (t_k). Kemudian menentukan nilai bobot yang menghubungkan *hidden note*, *output note*, dan bias dengan menghitung nilai koreksi. Langkah selanjutnya adalah menghitung nilai bobot pada jalur yang dilaluinya, kemudian dikalikan dengan nilai *error* dari semua node pada lapisan tersembunyi saat telah mencapai lapisan luar. Nilai dari *hidden note* akan dijumlahkan setelah menerima setiap nilai.

Untuk setiap output, *error* akan dihitung untuk setiap nilai dan dibandingkan dengan target yang diinginkan. Karena untuk melakukan penyesuaian nilai bobot yang menghubungkan antara lapisan tersembunyi dan lapisan luar, nilai *error* tersebut digunakan sebagai dasar. Dalam *Backpropagation*, bentuk fungsi error (MSE) sebagai berikut

$$E = \frac{1}{2} \sum_{k=1}^{\tau} (t_k - y_k^{(\tau)})^2 \quad (10)$$

2.6 Multi-Class

Multi-Class merupakan Kumpulan data atau yang biasa di sebut *dataset* yang terdiri dari atribut, dan kelas. Dataset dengan karakteristik *Multi-Class* merupakan dataset yang memiliki kelas klasifikasi lebih dari 2 dan biasa di sebut *polynomial* (Handayani *et al.*, 2023). Dalam situasi klasifikasi multi kelas, terdapat lebih dari satu *hyperplane* yang dibentuk. Salah satu metode pendekatan yang digunakan adalah *One Versus All*.

Metode pendekatan *One Versus All* adalah metode yang membandingkan satu kelas dengan semua kelas lainnya, sesuai dengan namanya (Sastypratiwi, Muhardi and Noveanto, 2022). *One Versus One* adalah metode klasifikasi biner dibangun oleh setiap pasangan kelas. Pendekatan sederhana untuk menggabungkan semua hasil dari metode klasifikasi biner pada *One Versus One* yaitu menggunakan majority voting dimana contoh uji akan diklasifikasikan ke dalam kelas yang memiliki *vote* tertinggi (Meidianingsih, Eka and Meganingtyas, no date).

2.7 Firefly

Algoritma kunang-kunang atau *Firefly Algorithm* adalah sebuah algoritma metaheuristik yang terinspirasi dari perilaku berkedip kunang-kunang. Fokus utama perilaku ini adalah menarik kunang-kunang lain. Algoritma ini di kembangkan oleh Dr Xin She Yang pada tahun 2007 (Simamora, Hajar and Fernandes, 2019). Berikut adalah rumus nya

Berdasarkan pada algoritma *Firefly*, fungsi objektif didapatkan dari rumus berikut.

$$\text{maks } f(x), x = (x_1, \dots, x_d)^T \quad (1)$$

Algoritma *Firefly* juga harus menentukan intensitas cahaya dari setiap kunang-kunang. Untuk menentukan intensitas cahaya di dapatkan dengan rumus berikut.

$$I(x) = f(x) \quad (2)$$

Setelah memperoleh intensitas cahaya, Langkah selanjutnya adalah menentukan jarak antara kunang-kunang. Jarak antara kunang-kunang dihitung menggunakan rumus berikut.

$$r_{1,2} = \sqrt{(m_1^1 - m_2^1)^2 + (m_1^2 - m_2^2)^2} \quad (3)$$

Bukan hanya jarak aja, tetapi algoritma *Firefly* juga butuh nilai daya tarik yang di miliki setiap kunang-kunang. Nilai daya tarik di dapatkan dengan rumus.

$$\beta = \beta_0 e^{-\gamma r^2} \quad (4)$$

Pergerakan kunang-kunang menjadi penentu dalam algoritma *Firefly*. Kunang-kunang dengan intensitas cahaya yang lebih rendah akan bergerak menuju kunang-kunang dengan intensitas cahaya yang lebih tinggi. Perpindahan *Firefly* di dapatkan dengan rumus.

$$m_{i_baru}^k = m_{i_lama}^k + \beta_0 e^{-\gamma r^2} (m_{i_lama}^k - m_{i_lama}^k) + a(\text{rand} - \frac{1}{2}) \quad (5)$$

2.8 Alat dan Bahan

Dalam penelitian ini membutuhkan berbagai alat dan bahan untuk mendukung penelitian. Alat dan bahan yang di gunakan sebagai berikut.

a. Alat

Alat penelitian adalah alat perangkat fisik atau bisa juga berbentuk digital yang kegunaanya untuk mengumpulkan data, dan melakukan percobaan yang sesuai. Tabel alat yang di gunakan untuk mengklasifikasikan jenis penyakit pada tanaman padi menggunakan metode *Backpropagation* dengan optimasi *Firefly* dapat dilihat pada Table 2.2.

Table 2.2 Alat Penelitian

Komponen-komponen	Spesifikasi
Laptop	MSI KATANA GF66 11UC
Processor	intel Core i5 11400H Gen 11

RAM	16 GB
SSD	512 GB + 1 TB
<i>Operation System</i>	<i>Windows 11</i>
<i>Text Editor</i>	<i>Google Colab</i>
<i>Software Perhitungan Manual</i>	<i>Microsoft Excel</i>

b. Bahan

Bahan penelitian adalah materi yang digunakan dalam percobaan, dan analisis untuk mendukung tujuan riset. Tabel bahan yang digunakan untuk mengklasifikasikan jenis penyakit pada tanaman padi menggunakan metode *Backpropagation* dengan optimasi *Firefly* dapat dilihat pada Table 2.3.

Table 2.3 Bahan Penelitian

Bahan-bahan	Diperoleh dari
Data gejala penyakit pada tanaman padi	Di ambil dari pertanyaan dari petani
Bahasa Pemrograman	<i>Python</i>

2.9 Evaluasi

Evaluasi merupakan langkah untuk memastikan bahwa model dapat menggeneralisasi pola yang telah dipelajari dari data latihan ke data baru dengan tepat (Nurhikam *et al.*, 2024). Kemampuan klasifikasi di ukur dari nilai akurasi, *recall*, presisi, *f1-score*.

2.9.1 Akurasi

Akurasi digunakan untuk mengetahui seberapa baik sistem dapat mengklasifikasi data dengan benar (Nasution, Limbong and Ramadhan, 2020) (Sari and Suryono, 2024).

$$\text{Akurasi} = \frac{TP + TN}{(\text{Total numbers of sampel})}$$

2.9.2 Presisi

Presisi digunakan untuk mengetahui seberapa akurat suatu model dapat mengidentifikasi suatu sentimen (Sari and Suryono, 2024).

$$\text{Presisi} = \frac{TP}{TP + FP}$$

2.9.3 Recall

Recall digunakan untuk mengetahui seberapa baik model menemukan dan mengenali suatu sentimen (Sari and Suryono, 2024).

$$\text{Recall} = \frac{TP}{TP + FN}$$

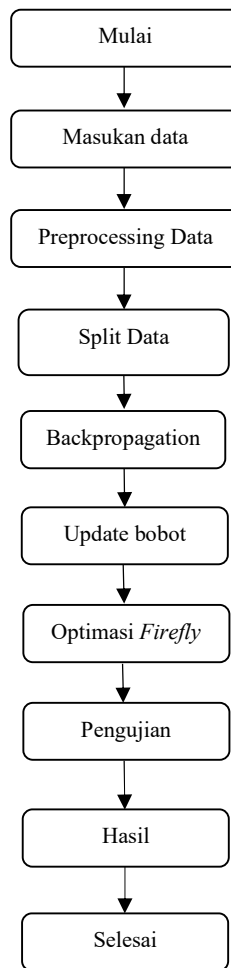
2.9.4 F1-score

F1-score digunakan untuk mengetahui perbandingan rata-rata presisi dan *recall* (Athallah, Azhar and Munarko, 2020).

$$F1 - score = 2 \times \frac{\text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}}$$

2.10 Prosedur Penelitian

Sesuai dengan tujuan penelitian ini yaitu adalah mendeteksi penyakit pada tanaman padi. Adapun prosedur dalam membangunnya :



Gambar 2.2 Alur Prosedur Penelitian

Keterangan Diagram alir penelitian :

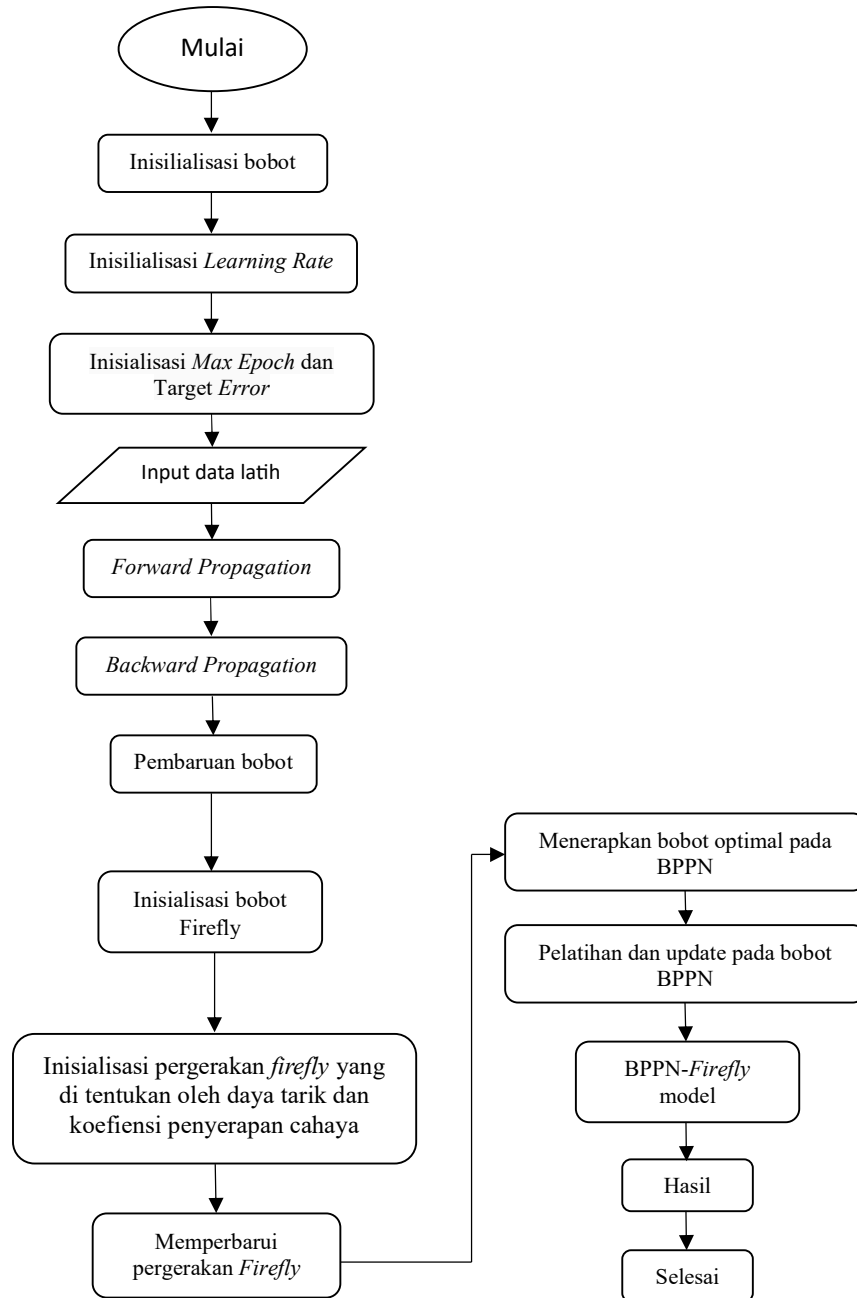
1. Di mulai dengan memasukkan data.
2. Preprocessing data. Beberapa tahapan di lakukan dalam preprocessing di antara nya pembersihan data, transformasi data, normalisasi, dan lain-lain.
3. Data *Splitting*. Membagi data menjadi data latih dan data uji.
4. *K-Fold Cross Validation*. Adalah metode data dibagi menjadi “K” nilai fold.
5. Backpropagation. Melakukan pengujian algoritma Backpropagation untuk melihat hasil kinerja yang di peroleh.
6. Update bobot. Setelah pengujian maka dilakukan pembaruan bobot untuk meningkatkan akurasi dan mengurangi kesalahan model.
7. Optimasi. Langkah model Jaringan Syaraf Tiruan yang telah di latih oleh metode *Backpropagation* lalu dioptimasi dengan algoritma *Firefly*.
8. Evaluasi. Proses untuk memastikan bahwa model dapat menggeneralisasi pola yang telah dipelajari dari data latihan ke data baru dengan tepat. Hasil pengujian yang akan di dapatkan adalah nilai akurasi, *recall*, presisi, *f1-score*

BAB III

HASIL ANALISIS DAN PEMBAHASAN

3.1 Alur Flowchart

Flowchart sistem menjelaskan alur dari simbol-simbol yang digunakan untuk menggambarkan proses kerja sistem pada klasifikasi penyakit pada tanaman padi



Gambar 2.3 Alur Flowchart

Keterangan flowchart penelitian :

1. Di mulai dengan memasukkan data.
2. Inisialisasi bobot. Pada tahap ini, bobot-bobot yang menghubungkan neuron-neuron dalam jaringan diberikan nilai awal secara acak. Inisialisasi bobot yang tepat sangat penting untuk memastikan konvergensi algoritma pelatihan yang efisien seperti propagasi mundur dan menghindari terjebak pada minimum lokal.
3. Inisialisasi *learning rate*. proses pelatihan jaringan saraf yang menentukan nilai awal kecepatan pembelajaran. Kecepatan pembelajaran menentukan seberapa besar bobot jaringan disesuaikan pada setiap iterasi pelatihan. Nilai *learning rate* yang benar sangat penting untuk mencapai konvergensi yang cepat dan stabil serta menghindari osilasi dan proses pembelajaran yang lambat.
4. Inisialisasi *Max Epoch* menentukan jumlah maksimum iterasi atau siklus pelatihan yang dilakukan algoritma. Hal ini dimaksudkan untuk mencegah pelatihan yang terlalu lama atau tidak efisien. Inisialisasi *Target Error*, sebaliknya, adalah fase yang menentukan nilai kesalahan minimum yang diperlukan sebagai kriteria untuk menghentikan pelatihan. Proses pelatihan berhenti ketika nilai *Error* jaringan mencapai atau turun di bawah nilai target ini, untuk memastikan bahwa model mencapai tingkat akurasi yang diinginkan.
5. Input data pelatihan adalah fase yang menyediakan data yang digunakan untuk melatih jaringan saraf tiruan. Data pelatihan terdiri dari pasangan masukan dan keluaran yang diketahui yang digunakan untuk mengajarkan model cara memetakan masukan ke keluaran yang benar. Kualitas dan keterwakilan data pelatihan penting untuk memastikan bahwa jaringan dapat mempelajari dan menggeneralisasi data baru yang belum pernah dilihat sebelumnya dengan benar.
6. *Forward Propagation* merupakan fase dalam proses pelatihan jaringan saraf dimana data masukan diproses oleh jaringan untuk menghasilkan keluaran yang diprediksi. Pada tahap ini, setiap *neuron* pada lapisan tersembunyi dan lapisan keluaran menghitung nilai aktivasi berdasarkan bobot yang ada dan fungsi aktivasi yang digunakan. Keluaran *Forward Propagation* kemudian dibandingkan dengan keluaran yang diharapkan untuk menghitung *error*, yang digunakan pada fase *Backward Propagation*.
7. *Backward Propagation* adalah fase dalam proses pelatihan jaringan saraf di mana kesalahan yang dihasilkan selama *Forward Propagation* disebarkan kembali melalui jaringan. Fase ini menghitung gradien kesalahan untuk setiap bobot dalam jaringan dan memperbaiki bobot menggunakan algoritme optimasi seperti penurunan gradien. Proses ini bertujuan untuk meminimalkan kesalahan prediksi dengan mengatur bobot sehingga jaringan mendekati keluaran yang diharapkan.
8. Pembaruan bobot adalah fase dalam proses pelatihan jaringan saraf di mana bobot jaringan disesuaikan berdasarkan gradien kesalahan yang dihitung selama propagasi mundur. Proses ini menggunakan *learning rate* yang telah ditentukan untuk menyesuaikan nilai bobot guna mengurangi kesalahan prediksi. Pembaruan bobot yang efektif membantu jaringan syaraf belajar dan menemukan solusi optimal.
9. Inisialisasi bobot *Firefly* merupakan langkah awal dalam melakukan optimasi bobot Jaringan Syaraf Tiruan menggunakan algoritma *Firefly*. Pada titik ini, setiap kunang-kunang diinisialisasi dengan nilai bobot acak yang berbeda. Berat masing-masing kunang-kunang mewakili solusi yang mungkin, dan algoritma kunang-kunang digunakan untuk menemukan solusi optimal melalui interaksi dan pergerakan kunang-kunang berdasarkan intensitas cahaya dan daya tariknya. Tujuan dari inisialisasi ini adalah untuk memberikan solusi awal yang berbeda yang akan dioptimalkan selama proses pelatihan.
10. Inisialisasi gerakan *Firefly* merupakan salah satu tahapan dalam algoritma *Firefly*, dimana setiap kunang-kunang bergerak berdasarkan daya tariknya terhadap kunang-kunang lain dan koefisien serapan cahayanya. Gaya tarik menarik antara dua kunang-kunang dipengaruhi oleh intensitas

cahaya kunang-kunang yang berkaitan dengan kualitas larutan yang diwakili oleh berat kunang-kunang. Koefisien serapan optik menentukan seberapa cepat gaya tarik menarik berkurang seiring bertambahnya jarak. Tujuan fase ini adalah memandu kunang-kunang menuju solusi optimal, dengan mempertimbangkan interaksi dan jaraknya.

11. Memperbarui pergerakan *Firefly* merupakan langkah dalam algoritma *Firefly* yang menyesuaikan posisi masing-masing kunang-kunang berdasarkan daya tariknya terhadap kunang-kunang lain yang lebih terang dan koefisien serapan cahayanya. Kunang-kunang bergerak menuju kunang-kunang yang lebih terang, dengan memperhatikan jarak antara mereka dan intensitas cahaya yang dipancarkan. Proses ini memungkinkan *Firefly* menjelajahi ruang solusi secara efektif dan mencapai solusi optimal.
12. Penerapan bobot optimal pada BPPN merupakan langkah penerapan bobot yang dioptimalkan oleh algoritma *Firefly* pada jaringan syaraf tiruan *Backpropagation*. Bobot optimal ini menggantikan bobot awal dan memungkinkan jaringan saraf memulai pelatihan dalam kondisi yang lebih baik. Tujuan dari proses ini adalah untuk meningkatkan efisiensi dan akurasi pelatihan serta mempercepat konvergensi menuju solusi optimal.
13. Latihan beban dan *update* BPPN merupakan tahap pelatihan jaringan syaraf tiruan dengan menggunakan data latih menggunakan *Backpropagation*. Pada fase ini, jaringan memperbarui bobot secara berulang berdasarkan gradien kesalahan yang dihitung. Proses ini berlanjut hingga jaringan mencapai konvergensi. Hingga kesalahan prediksi mencapai level minimum yang diinginkan atau jumlah iterasi maksimum tercapai.
14. Model *BPPN-Firefly* menggabungkan algoritma jaringan syaraf tiruan *Backpropagation* (BPPN) dengan algoritma *Firefly* yang digunakan untuk mengoptimalkan bobot jaringan syaraf tiruan. Dalam model ini, algoritma *Firefly* terlebih dahulu menginisialisasi dan mengoptimalkan bobot awal jaringan, kemudian BPPN menggunakan bobot tersebut untuk proses pelatihan selanjutnya. Tujuan dari model ini adalah untuk memanfaatkan kekuatan kedua algoritma untuk meningkatkan akurasi dan efisiensi pelatihan.
15. Hasil *BPPN-Firefly* merupakan keluaran akhir dari algoritma jaringan syaraf tiruan *Backpropagation* (BPPN) yang digabungkan dengan algoritma *Firefly* setelah proses pelatihan selesai. Hasil ini mencakup model jaringan syaraf tiruan dengan bobot optimal yang dapat membuat prediksi sangat akurat. Hasil ini digunakan untuk mengevaluasi kinerja model pada data pengujian dan untuk memastikan bahwa model tersebut dapat diandalkan untuk tugas klasifikasi atau prediksi yang diinginkan.

3.2 Implementasi Program

Implementasi program adalah proses penerapan program yang sudah dirancang sebelumnya. Tujuannya untuk mencapai hasil yang telah ditetapkan.

3.2.1 Pembagian data latih dan data uji

Pembagian data latih dan data uji adalah proses yang dimana model yang dibangun dapat generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya. Berikut hasil dari pembagian data latih dan data uji.

```
Jumlah data latih: 875
Jumlah data uji: 219
```

Gambar 3.1 Pengujian Pembagian Data

Berdasarkan Gambar 3.1 menunjukkan Jumlah data latih yang berjumlah 875 dan data uji yang berjumlah 219 dari total data yang berjumlah 1094 data. Pembagian data dilakukan dengan memasukkan parameter *test_size* 0.2, artinya 20% dari total data dijadikan data uji, sedangkan 80% di gunakan untuk data latih.

3.2.2 Normalisasi Data

Pada tahap ini langkah dalam prapemrosesan data yang membantu meningkatkan kinerja pengujian normalisasi data.

```
Data setelah normalisasi:
      T
0 -1.380606
1 -1.380606
2 -1.380606
3 -1.380606
4 -1.380606
```

Gambar 3.2 Pengujian Normalisasi Data

Berdasarkan Gambar 3.2 menunjukkan data setelah proses normalisasi. Hasil normalisasi yang menunjukkan nilai yang sama bisa disebabkan oleh keseragaman data awal atau masalah dalam proses normalisasi itu sendiri.

3.2.3 K-Fold Cross Validation

Dalam setiap perulangan *K-Fold Cross Validation*, data dibagi menjadi data latih (*Train Index*) dan data uji (*Text Index*). Pada pengujian ini, nilai K yang digunakan adalah 2, dengan membagi data menjadi 2 *Fold* dan mengulangi proses pelatihan serta evaluasi. Teknik ini memastikan setiap data digunakan sebagai data latih dan data uji, sehingga nilai evaluasi konsisten. Berikut hasil dari Pengujian *Index K-Fold*.

```

Train Index K-fold:
[ 1 4 7 9 10 15 18 20 21 22 24 25 28 29
 30 32 36 37 38 39 42 43 44 48 51 52 53 55
 63 64 67 70 71 77 79 83 86 87 93 95 96 97
 98 100 102 105 109 112 113 115 116 122 123 124 127 128
129 130 131 136 137 138 140 141 143 144 145 149 150 152
153 155 157 158 162 163 164 166 167 168 169 170 171 173
174 176 179 182 183 184 188 193 196 197 198 199 203 206
208 209 210 211 212 213 217 219 221 222 225 227 229 230
231 232 234 235 237 240 243 249 251 252 253 256 260 261
263 266 269 270 271 272 275 278 279 284 285 287 288 290
292 295 296 300 302 303 305 308 312 313 315 316 317 319
322 324 325 326 327 328 330 332 333 334 337 338 342 343
346 351 352 353 356 357 359 360 361 362 365 369 373 376
377 378 379 380 384 391 392 394 396 397 398 399 400 401
402 405 406 407 409 410 413 416 417 420 421 423 424 425
427 431 438 440 441 442 444 445 448 450 451 454 456 458
460 461 462 465 467 468 469 470 475 476 477 485 488 489
490 492 497 498 499 500 501 503 504 505 506 508 510 511
513 514 515 517 518 519 520 522 524 526 532 533 536 539
540 542 543 544 545 547 549 551 556 557 558 562 563 564
565 566 568 569 570 572 573 575 580 583 585 586 588 590
591 592 595 596 598 601 603 605 606 608 609 610 611 614
615 617 618 620 621 622 624 626 627 628 630 633 634 636
637 642 643 645 646 647 648 649 652 654 656 657 660 662
663 665 666 668 669 670 675 679 681 682 687 689 690 693
694 695 696 703 706 708 709 712 714 715 716 717 722 723
724 726 727 728 729 732 733 738 739 740 742 743 745 746
747 749 750 751 752 753 755 756 758 760 765 770 772 777
778 781 784 787 788 791 792 800 801 805 806 807 808 810
811 816 819 821 822 824 825 829 832 834 835 836 837 841
843 845 846 847 849 850 852 853 855 861 864 865 866 868
869 871 880 881 882 883 884 886 889 891 892 895 896 898
900 901 903 904 905 907 908 909 910 911 914 915 919 922
924 925 927 928 931 932 933 938 939 940 944 946 948 951
952 953 954 957 960 962 963 964 968 969 970 975 977 986
987 989 992 999 1001 1003 1004 1006 1008 1010 1011 1012 1013 1014
1015 1017 1018 1020 1021 1023 1026 1030 1031 1034 1035 1036 1044 1046
1048 1050 1052 1053 1054 1055 1056 1057 1058 1060 1061 1063 1064 1065
1069 1070 1072 1074 1076 1078 1080 1081 1082 1083 1084 1087 1090 1091
1092]

```

Gambar 3.3 Pengujian Train Index K-Fold

Berdasarkan gambar 3.3 menunjukkan *train index* dari data yang digunakan sebagai data latih pada *k-fold*. Indeks ini merujuk pada baris-baris data yang digunakan untuk melatih model pada setiap *fold*. Berikut contoh pada *train index*. Pada *fold* pertama, data latih mencakup indeks 1, 4, 7, 9 dan seterusnya hingga indeks 1092.

```

Test Index k-fold:
[ 0 2 3 5 6 8 11 12 13 14 16 17 19 23
 26 27 31 33 34 35 40 41 45 46 47 49 50 54
 56 57 58 59 60 61 62 65 66 68 69 72 73 74
 75 76 78 80 81 82 84 85 88 89 90 91 92 94
 99 101 103 104 106 107 108 110 111 114 117 118 119 120
121 125 126 132 133 134 135 139 142 146 147 148 151 154
156 159 160 161 165 172 175 177 178 180 181 185 186 187
189 190 191 192 194 195 200 201 202 204 205 207 214 215
216 218 220 223 224 226 228 233 236 238 239 241 242 244
245 246 247 248 250 254 255 257 258 259 262 264 265 267
268 273 274 276 277 280 281 282 283 286 289 291 293 294
297 298 299 301 304 306 307 309 310 311 314 318 320 321
323 329 331 335 336 339 340 341 344 345 347 348 349 350
354 355 358 363 364 366 367 368 370 371 372 374 375 381
382 383 385 386 387 388 389 390 393 395 403 404 408 411
412 414 415 418 419 422 426 428 429 430 432 433 434 435
436 437 439 443 446 447 449 452 453 455 457 459 463 464
466 471 472 473 474 478 479 480 481 482 483 484 486 487
491 493 494 495 496 502 507 509 512 516 521 523 525 527
528 529 530 531 534 535 537 538 541 546 548 550 552 553
554 555 559 560 561 567 571 574 576 577 578 579 581 582
584 587 589 593 594 597 599 600 602 604 607 612 613 616
619 623 625 629 631 632 635 638 639 640 641 644 650 651
653 655 658 659 661 664 667 671 672 673 674 676 677 678
680 683 684 685 686 688 691 692 697 698 699 700 701 702
704 705 707 710 711 713 718 719 720 721 725 730 731 734
735 736 737 741 744 748 754 757 759 761 762 763 764 766
767 768 769 771 773 774 775 776 779 780 782 783 785 786
789 790 793 794 795 796 797 798 799 802 803 804 809 812
813 814 815 817 818 820 823 826 827 828 830 831 833 838
839 840 842 844 848 851 854 856 857 858 859 860 862 863
867 870 872 873 874 875 876 877 878 879 885 887 888 890
893 894 897 899 902 906 912 913 916 917 918 920 921 923
926 929 930 934 935 936 937 941 942 943 945 947 949 950
955 956 958 959 961 965 966 967 971 972 973 974 976 978
979 980 981 982 983 984 985 988 990 991 993 994 995 996
997 998 1000 1002 1005 1007 1009 1016 1019 1022 1024 1025 1027 1028
1029 1032 1033 1037 1038 1039 1040 1041 1042 1043 1045 1047 1049 1051
1059 1062 1066 1067 1068 1071 1073 1075 1077 1079 1085 1086 1088 1089
1093]

```

Gambar 3.4 Pengujian Test Index K-Fold

Berdasarkan gambar 3.4 menunjukkan indeks yang tidak termasuk dalam test index pada iterasi tersebut akan digunakan sebagai data uji yaitu indeks nilai 2,3,5,6 dan seterusnya hingga indeks 1093. Data uji menilai kemampuan model dalam mengklasifikasikan data baru, sehingga memberikan gambaran yang lebih jelas mengenai kinerja keseluruhan model.

3.2.4 Menginisialisasi bobot dan bias

Menginisialisasi bobot dan bias dengan tepat sangat penting dalam pelatihan. Pemilihan metode inisialisasi yang sesuai dengan arsitektur jaringan dan jenis fungsi aktivitas yang digunakan dapat mempengaruhi kinerja dan efisiensi model secara signifikan.

Bobot dan bias sebelum optimasi:

```

W1: [[ -2.86361331 -2.34181424 -2.89092042  2.84164395 -2.33495535 -2.31528158
 2.30135127  2.32035404 -2.31225695  2.77778823 -2.33188357 -2.32480159
 2.83669175 -2.88078735 -2.87001468  2.32582575 -2.32671242  2.33621678
 2.85289087  2.32807234]]
b1: [[ 1.90484616 -1.2572877  1.91543419  2.4547279 -1.20254566 -1.08717706
-0.74584085 -0.83023791 -1.12169172  2.32813354 -1.22268329 -1.08139828
 2.45029961  1.83642921  1.8241937 -0.81188226 -1.16864153 -0.8553951
 2.44940802 -0.86003486]]
W2: [[ 0.42276463 -2.90593607 -3.36554013  1.40867829  1.62915742]
 [ 1.85322222 -1.73506476 -1.50061174 -2.39184047  2.21534666]
 [ 0.19058994 -2.93264122 -3.33083857  1.41602533  1.61661144]
 [ -2.77494493  1.00417259 -1.24749747  3.7293457 -2.24773887]
 [ 1.8457326 -1.82553978 -1.49351239 -2.35952133  2.22656645]
 [ 1.8122843 -1.81160153 -1.49537341 -2.21116909  2.22584439]
 [ -1.71456854  2.24253425  2.17133197  1.63029292 -2.16294916]
 [ -1.7021034  2.26584242  2.22764912  1.59167943 -2.11838704]
 [ 1.81584325 -1.83947386 -1.53256532 -2.27855608  2.21823795]
 [ -2.7161819  1.0348223 -0.83821355  3.62660029 -2.29146412]
 [ 1.8368583 -1.75407915 -1.4644055 -2.36265076  2.21372224]
 [ 1.82249872 -1.86353959 -1.48864007 -2.22064631  2.24513673]
 [ -2.72611766  0.98726785 -1.30334736  3.72484883 -2.26507925]
 [ 0.7108349 -2.98223062 -3.30218378  1.38637892  1.66762696]
 [ 0.70809071 -2.96572927 -3.26475461  1.37733343  1.66244087]
 [ -1.64976061  2.26381874  2.21413933  1.59979957 -2.1667058 ]
 [ 1.82731573 -1.74722719 -1.4757494 -2.26999496  2.208755 ]
 [ -1.64383806  2.27582352  2.24360166  1.58026338 -2.14296091]
 [ -2.81523721  1.00354887 -1.2906383  3.71331917 -2.30791472]
 [ -1.63491033  2.2699391  2.24300018  1.58107936 -2.11744599]]
b2: [[ -1.81747526 -1.5555643 -1.59481  1.75455237 -0.38613145]]
W3: [[ 5.58462122  7.15992323  4.79999652 -7.70019508 -6.33426554
-1.94607631 -1.5385076 -1.18711291 -0.83246226 -0.42296735
-0.48159571]
 [ -1.2637718 -1.5391371 -2.30474735 -2.41157501 -6.78469444
-10.7879454  0.63497805  11.99362506  7.88445831  2.4917071
 1.22111097]
 [ 0.18105222 -0.08021844 -0.42930104 -0.75823288 -1.58259078
-1.99112683 -7.0676251 -12.44087749  0.02726384  9.48724584
13.42657188]
 [ -15.61528542 -1.97993746  8.4901378  1.48364693  3.6854798
 4.77860984  3.37699042 -0.38966453  0.87478298  0.16961723
-1.72466206]
 [ 3.91009223  1.83142836  2.11591414  13.22393961  1.41055412
-10.39331952 -6.43653223 -1.87722671 -1.80701439 -1.53094224
-1.33357209]]
b3: [[ 3.29183269  0.11487854 -6.1075618 -3.54402872  2.83120635  4.52778601
 3.21818927 -1.10394495 -1.05589407 -1.31382964 -1.81758335]]

```

Gambar 3.5 Pengujian Bobot dan bias sebelum optimasi

Berdasarkan Gambar 3.5 terdapat beberapa layer, masing-masing dengan bobot (*weights*) dan bias (*biases*). Terdapat tiga layer utama yang ditunjukkan oleh W1, b1, W2, b2, W3 dan b3. Inisialisasi penting dalam pelatihan Jaringan Syaraf Tiruan karena nilai awal dari bobot dan bias dapat mempengaruhi kinerja akhir dari model yang dilatih.

3.2.5 Proses Optimasi Menggunakan Firefly

Pada tahap optimasi nilai yang ditampilkan adalah keaktraktifan kunang-kunang (*Firefly*), bobot dan bias yang telah di optimasi.

```

Bobot dan bias setelah optimasi:
w1: [[-2.87869027 -2.36913891 -2.91908965 2.87906877 -2.36199901 -2.34325107
2.31439175 2.32986858 -2.33975267 2.81360582 -2.35920789 -2.35251079
2.87381792 -2.92358002 -2.91034692 2.33816111 -2.35447847 2.34671755
2.88953516 2.33665518]]
b1: [[ 1.78197313 -1.26716089 1.78769151 2.49802437 -1.20993936 -1.09133792
-0.70707236 -0.78889845 -1.12709568 2.36996611 -1.23139647 -1.08460596
2.49342333 1.70942916 1.70093988 -0.77061636 -1.175362 -0.81269905
2.49201134 -0.81809877]]
w2: [[ 0.42166778 -3.01172697 -3.3138754 1.42139829 1.67460286]
[ 1.88656116 -1.75993053 -1.4974536 -2.4316858 2.26633256]
[ 0.18941784 -3.03908473 -3.27838505 1.42883074 1.66214923]
[-2.81933965 0.94799678 -1.3090581 3.81960386 -2.25662416]
[ 1.87866771 -1.85157694 -1.49016842 -2.39761688 2.27864129]
[ 1.8441998 -1.84022079 -1.49160699 -2.24575765 2.27998618]
[-1.72894712 2.25950076 2.09136517 1.64174069 -2.20007491]
[-1.715549 2.28499097 2.14635808 1.60216032 -2.15456771]
[ 1.84799059 -1.86730568 -1.52892738 -2.31434183 2.2715795 ]
[-2.76054576 0.97973494 -0.89991328 3.71323249 -2.30235353]
[ 1.86988679 -1.77968053 -1.46112937 -2.40148889 2.26527748]
[ 1.8544552 -1.89229492 -1.48485331 -2.25486923 2.29961333]
[-2.77045777 0.93113079 -1.36491416 3.81501355 -2.27399686]
[ 0.71003883 -3.08830435 -3.25238693 1.39897217 1.71454647]
[ 0.70736022 -3.07153685 -3.21541525 1.38987331 1.70946004]
[-1.66339675 2.28261134 2.13297202 1.61041501 -2.20326621]
[ 1.85994529 -1.77400323 -1.47228562 -2.30709314 2.26140161]
[-1.65700396 2.29572122 2.16171833 1.5904009 -2.17901695]
[-2.8599082 0.94740779 -1.35219695 3.8032374 -2.31707585]
[-1.64803343 2.2898489 2.16120621 1.59122804 -2.15328861]]
b2: [[[-1.82350539 -1.61981919 -1.65565828 1.76990123 -0.36445479]]]
w3: [[ 5.68835605 7.44869807 5.00851447 -8.15436395 -6.47626797
-1.94698621 -1.54066312 -1.18728516 -0.83313796 -0.42376964
-0.48173638]
[-1.26377183 -1.54025419 -2.36760709 -2.41213232 -7.10032053
-11.64852963 0.60796908 12.68138351 8.34907343 2.73058513
1.09761284]
[ 0.18105222 -0.08036003 -0.43439998 -0.75826738 -1.58431664
-1.99171256 -7.47377039 -13.39554944 -0.06902606 9.61966746
14.75884403]
[-16.42884097 -2.18512441 9.11959358 1.39514814 3.79031647
5.10539562 3.55925394 -0.56638504 0.82393533 0.3857095
-1.84928661]
[ 4.01072282 1.82540422 2.22616739 13.89589907 1.51830021
-11.05587206 -6.75101028 -1.87819049 -1.81074744 -1.53351131
-1.33384085]]
b3: [[ 3.39246259 0.10349255 -6.49270923 -3.72014976 2.93120974 4.85454677
3.40018531 -1.28073225 -1.10695652 -1.0980266 -1.94227227]]

```

Gambar 3.6 Pengujian Bobot dan bias setelah optimasi

Log Nilai Keatraktifan:			
Iteration	Firefly	Attractiveness	
0	0	0	0.200000
1	0	0	0.200000
2	0	0	0.200000
3	0	0	0.200000
4	0	0	0.200000
..
248	8	4	0.200000
249	8	4	0.971600
250	8	4	0.971533
251	8	4	0.974920
252	8	4	0.200000

[253 rows x 3 columns]

Gambar 3.7 Pengujian Nilai Keatraktifan Firefly

Berdasarkan Gambar 3.6 menampilkan hasil dari bobot dan bias yang telah di optimasi menggunakan algoritme *Firefly*, sedangkan pada Gambar 3.7 menampilkan keatraktifan kunang-kunang dari algoritme *Firefly*.

3.2.6 Epoch dan Loss

Pada tahap ini menunjukkan nilai loss di *Epoch* selama proses pelatihan. Berikut adalah pengujian *Epoch* dan *loss*.

```

Epoch dan loss
Epoch 0, Loss: 2.8549959115592904

Epoch dan loss
Epoch 100, Loss: 1.4394959053138086

Epoch dan loss
Epoch 200, Loss: 1.0955708190304654

Epoch dan loss
Epoch 300, Loss: 0.9067436189863152

Epoch dan loss
Epoch 400, Loss: 0.7937491013618913

```

Gambar 3.8 Pengujian Epoch dan Loss

Berdasarkan Gambar 3.8 menunjukkan nilai dari pelatihan sebanyak 500 *Epoch*. Penurunan nilai *loss* dari 0 hingga 400 menunjukkan bahwa model mengalami proses pembelajaran yang baik selama pelatihan.

3.2.7 Forward Propagation dan Backward Propagation

Pada tahap ini menunjukkan proses dari *Forward Propagation* dari data indeks 0. Dimana proses input syaraf melewati beberapa lapisan hingga menghasilkan output terakhir.

```

Forward propagation for data index 0:
Input: [[-1.38060615]]
Layer 1 activation (a1): [[0.99684721 0.88118363 0.99703469 0.18590585 0.88605377 0.89508507
0.01979493 0.01788881 0.89121199 0.18027058 0.88347328 0.89690369
0.18630699 0.99681382 0.99672769 0.01800926 0.88847378 0.01708047
0.18282876 0.01722363]]
Layer 2 activation (a2): [[9.99930811e-01 2.98282013e-11 1.95556582e-11 1.45978834e-02
9.99999992e-01]]
Output: [[9.70836362e-01 2.91372567e-02 6.10644336e-06 1.96270225e-05
3.54733783e-07 7.93499969e-10 2.01443346e-08 3.27189097e-08
6.05573230e-08 1.20626534e-07 5.78315873e-08]]

```

Gambar 3.9 Pengujian Forward Propagation

Berdasarkan Gambar 3.9 menunjukkan input diteruskan melalui setiap lapisan dalam jaringan saraf tiruan, menghasilkan nilai aktivasi di setiap lapisan, hingga menghasilkan output.

Pada tahap ini ini menunjukkan proses dari *Backward pagation* dari data indeks 0. Proses ini melibatkan pembaruan bobot dan bias dari lapisa-lapisan jaringan berdasarkan kesalahan yang dihitung selama fase *Forward propagation*.


```

Backward propagation for data index 0:
Updated W1: [[ -2.8786899 -2.36915991 -2.9190893 2.87911646 -2.36201897 -2.34326846
2.31439437 2.3298709 -2.33977118 2.81365108 -2.35922829 -2.35252798
2.87386563 -2.92357966 -2.91034654 2.33816346 -2.35449733 2.34671975
2.88958203 2.3366574 ]]
Updated b1: [[ 1.78197287 -1.26714568 1.78769126 2.49798983 -1.2099249 -1.09132532
-0.70707426 -0.78890013 -1.12708227 2.36993333 -1.2313817 -1.0845935
2.49338878 1.70942889 1.70093961 -0.77061806 -1.17534835 -0.81270064
2.4919774 -0.81810038]]
Updated W2: [[ 0.42166774 -3.01172697 -3.3138754 1.4213387 1.67460286]
[ 1.88656113 -1.75993053 -1.4974536 -2.43173847 2.26633256]
[ 0.1894178 -3.03908473 -3.27838505 1.42877114 1.66214923]
[ -2.81933966 0.94799678 -1.3090581 3.81959275 -2.25662416]
[ 1.87866768 -1.85157694 -1.49016842 -2.39766984 2.27864129]
[ 1.84419977 -1.84022079 -1.49160699 -2.24581115 2.27998618]
[ -1.72894712 2.25950076 2.09136517 1.64173951 -2.20007491]
[ -1.71554901 2.28499097 2.14635808 1.60215925 -2.15456771]
[ 1.84799056 -1.86730568 -1.52892738 -2.31439511 2.2715795 ]
[ -2.76054576 0.97973494 -0.89991328 3.71322171 -2.30235353]
[ 1.86988676 -1.77968053 -1.46112937 -2.40154169 2.26527748]
[ 1.85445517 -1.89229492 -1.48485331 -2.25492285 2.29961333]
[ -2.77045777 0.93113079 -1.36491416 3.81500242 -2.27399686]
[ 0.7100388 -3.08830435 -3.25238693 1.39891259 1.71454647]
[ 0.70736019 -3.07153685 -3.21541525 1.38981373 1.70946004]
[ -1.66339675 2.28261134 2.13297202 1.61041394 -2.20326621]
[ 1.85994526 -1.77400323 -1.47228562 -2.30714625 2.26140161]
[ -1.65700396 2.29572122 2.16171833 1.59039988 -2.17901695]
[ -2.8599082 0.94740779 -1.35219695 3.80322647 -2.31707585]
[ -1.64803344 2.2898489 2.16120621 1.59122701 -2.15328861]]
Updated b2: [[ -1.82350543 -1.61981919 -1.65565828 1.76984145 -0.36445479]]
Updated W3: [[ 5.68864766 7.44840672 5.0085144 -8.15436415 -6.47626797
-1.94698621 -1.54066312 -1.18728516 -0.83313796 -0.42376964
-0.48173638]
[ -1.26377183 -1.54025419 -2.36760709 -2.41213232 -7.10032053
-11.64852963 0.60796908 12.68138351 8.34907343 2.73058513
1.09761284]
[ 0.18105222 -0.08036003 -0.43439998 -0.75826738 -1.58431664
-1.99171256 -7.47377039 -13.39554944 -0.06902606 9.61966746
14.75884403]
[ -16.42883671 -2.18512866 9.11959358 1.39514814 3.79031647
5.10539562 3.55925394 -0.56638504 0.82393533 0.3857095
-1.84928661]
[ 4.01101445 1.82511284 2.22616733 13.89589887 1.51830021
-11.05587206 -6.75101028 -1.87819049 -1.81074744 -1.53351132
-1.33384085]]
Updated b3: [[ 3.39275422 0.10320118 -6.49270929 -3.72014996 2.93120973 4.85454677
3.40018531 -1.28073225 -1.10695652 -1.0980266 -1.94227227]]

```

Gambar 3.10 Pengujian Backward Propagation

Berdasarkan Gambar 3.10 menunjukkan perubahan dalam parameter jaringan Syaraf setelah *backward pagation*. Yang mencerminkan jaringan saraf untuk belajar dri data yang diberikan pada indeks 0.

3.2.8 Perbandingan Data Uji Dan Data Klasifikasi

Pada tahap ini menunjukkan hasil dari proses klasifikasi menggunakan data uji. Berikut adalah hasil dari perbandingannya.

Data Uji	Klasifikasi	Klasifikasi
480	6	6
90	0	0
255	0	0
323	8	8
487	8	8
..
277	2	2
27	3	3
89	0	0
449	2	2
73	9	9

Gambar 3.11 Pengujian perbandingan Data

Berdasarkan Gambar 3.11 menunjukkan hasil dari klasifikasi dari model terhadap data uji, dengan hasil yang tampaknya akurat berdasarkan tabel yang ditampilkan.

3.2.9 Evaluasi

Average Accuracy: 0.9003656307129799
 Average Precision: 0.8825960136120564
 Average Recall: 0.9090909090909092
 Average F1-Score: 0.8903945111492281

Gambar 3.12 Hasil Pengujian Evaluasi

Berdasarkan Gambar 3.12 Secara keseluruhan nilai-nilai metrik yang ditampilkan menunjukkan bahwa model memiliki kinerja yang baik, dengan nilai akurasi 0.9003656307129799, presisi 0.8825960136120564, recall 0.9090909090909092, dan F1-Score 0.8903945111492281.

3.3 Hasil Pengujian

Dalam bab ini, menjelaskan langkah-langkah implementasi algoritme *Firefly* untuk mengoptimalkan metode *Backpropagation*. Pembahasan mencakup mulai dari tahap persiapan data, perancangan model, implementasi algoritme, hingga evaluasi kinerja model.

3.3.1 Parameter awal Backpropagation

Pengujian pada parameter awal dilakukan untuk mengetahui parameter yang optimal dalam model algoritme *Backpropagation*. Berikut adalah parameter awal algoritme *Backpropagation*.

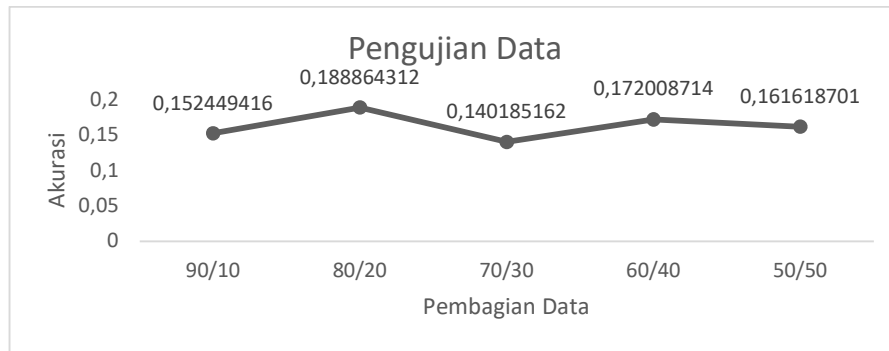
Tabel 3.1 Parameter awal Backpropagation

Rasio	K-Fold	Learning Rate	Epoch	Hidden 1	Hidden2	Akurasi
10%	5	0,03	500	10	5	0,171845335

Berdasarkan pada Tabel 3.1 menunjukkan parameter awal yang akan digunakan untuk pengujian yaitu rasio data dengan nilai 10%, nilai *K-Fold* 5, nilai *Learning rate* 0,03, nilai *Epoch* 500, *hidden 1* dengan nilai 10 dan, *hidden 2* dengan nilai 5 dengan akurasi 0,171845335.

3.3.2 Rasio Data

Rasio data digunakan untuk membagi data menjadi data latih dan data uji. Berikut hasil dari pengujian rasio data.

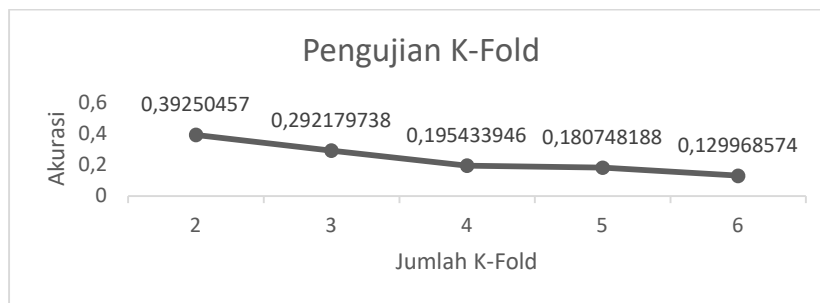


Gambar 3.13 Gambar Grafik Penguujian Data

Berdasarkan Gambar 3.13 grafik penguujian rasio data mendapatkan nilai akurasi tertinggi pada rasio 20% sebesar 0,188864312. Sehingga nilai penguujian rasio terbaik adalah 20%.

3.3.3 K-Fold

K-fold Cross Validation adalah metode evaluasi yang digunakan untuk membandingkan algoritme pelatihan. Prosesnya melibatkan pembagian dataset menjadi beberapa bagian atau lipatan sebanyak K. Berikut hasil dari penguujian *K-Fold*.

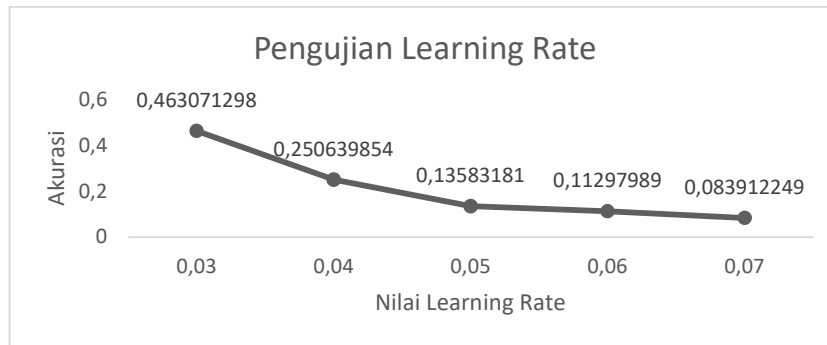


Gambar 3.14 Grafik Penguujian K-Fold

Berdasarkan Gambar 3.14 menunjukkan grafik penguujian *K-Fold* mendapatkan nilai akurasi tertinggi pada jumlah *K-Fold* 2 sebesar 0,39250457. Sehingga nilai penguujian pada *K-Fold* terbaik adalah 2.

3.3.4 Learning Rate

Learning rate adalah parameter yang menentukan besarnya langkah pembaruan bobot model selama pelatihan. Berikut hasil dari penguujian *Learning rate*.

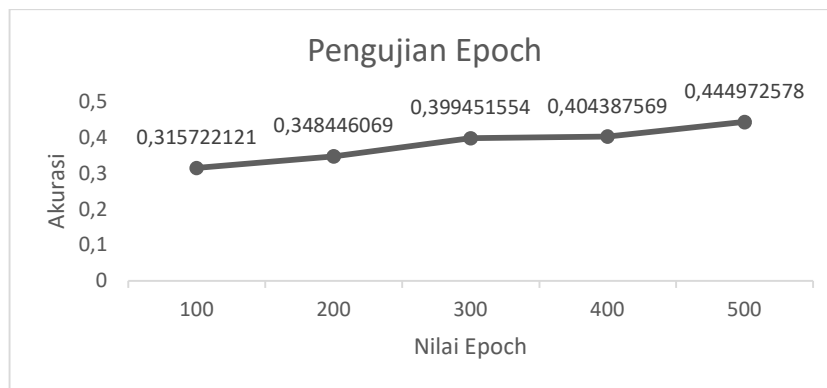


Gambar 3.15 Grafik Pengujian Learning Rate

Berdasarkan Gambar 3.15 menunjukkan grafik pengujian *Learning rate* mendapatkan hasil akurasi tertinggi pada nilai *Learning rate* 0,03 sebesar 0,463071298. Sehingga nilai pengujian pada *Learning rate* terbaik adalah 0,03.

3.3.5 Epoch

Epoch adalah parameter menggambarkan satu putaran lengkap ketika model mempelajari seluruh dataset. Berikut hasil dari pengujian *Epoch*.

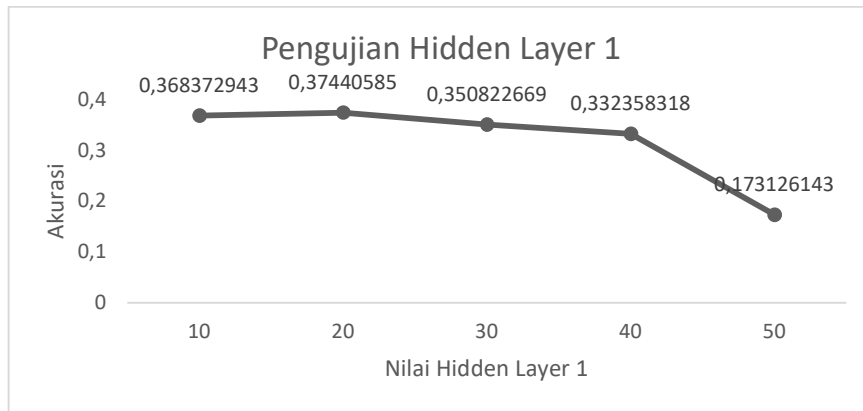


Gambar 3.16 Grafik Pengujian Epoch

Berdasarkan Gambar 3.16 menunjukkan grafik pengujian *Epoch* mendapatkan hasil akurasi tertinggi pada nilai *Epoch* 500 sebesar 0,444972578. Sehingga nilai pengujian pada *Epoch* terbaik adalah 500.

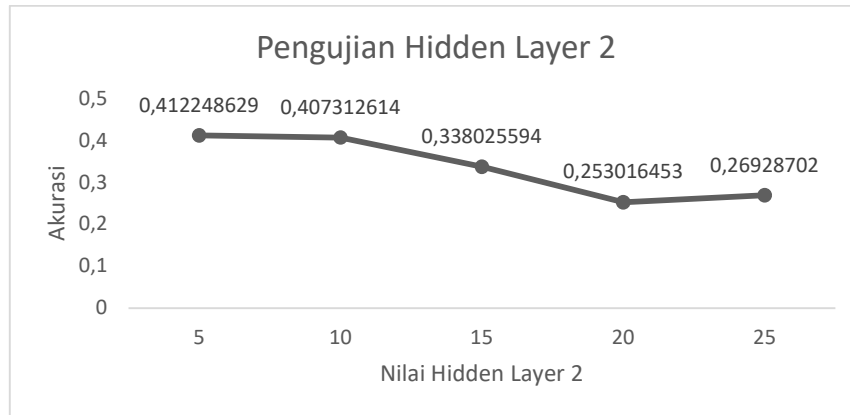
3.3.6 Hidden

Hidden layer adalah lapisan tersembunyi dari neural network yang terletak di antara lapisan *input layer* dan lapisan *output layer*. Berikut hasil dari pengujian *Hidden layer 1*.



Gambar 3.17 Grafik Pengujian Hidden Layer 1

Berdasarkan Gambar 3.17 menunjukkan grafik pengujian *Hidden layer 1* mendapatkan hasil akurasi tertinggi pada nilai sebesar 0,37440585. Sehingga nilai pengujian pada *Hidden layer 1* terbaik adalah 20.



Gambar 3.18 Grafik Pengujian Hidden Layer 2

Berdasarkan Gambar 3.18 menunjukkan grafik pengujian *Hidden layer 2* mendapatkan hasil akurasi tertinggi pada nilai sebesar 0,412248629. Sehingga nilai pengujian pada *Hidden layer 2* terbaik adalah 5.

3.3.7 Parameter Akhir Backpropagation

Parameter akhir dilakukan untuk mengetahui parameter yang terbaik setelah melakukan pengujian dari parameter awal dalam model algoritme *Backpropagation*.

Tabel 3.2 Parameter Akhir Backpropagation

Rasio	K-Fold	Learning Rate	Epoch	Hidden 1	Hidden2	Akurasi
20%	2	0,03	500	20	5	0,436380256

Berdasarkan pada Tabel 3.2 menunjukkan parameter akhir yang akan digunakan untuk pengujian yaitu rasio data dengan nilai 20%, nilai *K-Fold* 2, nilai *Learning rate* 0,03,

nilai *Epoch* 500, *Hidden* 1 dengan nilai 20 dan, *Hidden* 2 dengan nilai 5 dengan akurasi 0,436380256.

3.3.8 Parameter Awal Firefly Optimization

Pengujian pada parameter awal dilakukan untuk mengetahui parameter yang optimal dalam model algoritme *Firefly Optimization*.

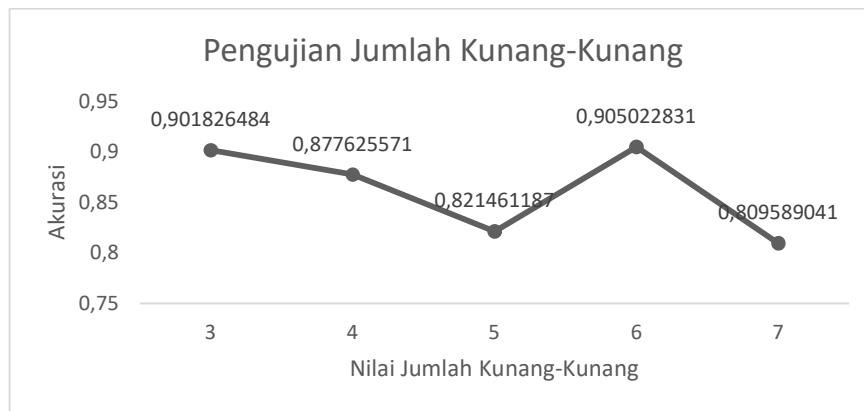
Tabel 3.3 Parameter Awal Firefly Optimization

<i>Firefly</i> (Kunang-kunang)	Iterasi	Akurasi
7	10	0,800913242

Berdasarkan pada Tabel 3.3 menunjukkan parameter awal yang akan digunakan untuk pengujian yaitu dengan nilai *Firefly* yang berjumlah 7, dan nilai iterasi yang berjumlah 10 dengan akurasi 0,800913242.

3.3.9 Jumlah Firefly

Jumlah *Firefly* adalah banyaknya individu (atau solusi) dalam populasi yang digunakan untuk eksplorasi ruang pencarian. Berikut hasil dari pengujian jumlah *Firefly*.

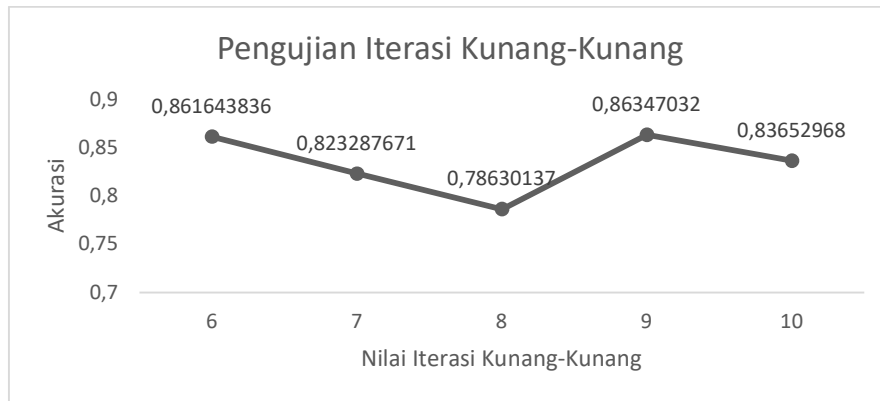


Gambar 3.19 Grafik Pengujian Jumlah Kunang-Kunang

Berdasarkan Gambar 3.19 menunjukkan grafik pengujian Jumlah kunang-kunang mendapatkan hasil akurasi tertinggi pada nilai sebesar 0,905022831. Sehingga nilai pengujian jumlah kunang-kunang terbaik adalah 6.

3.3.10 Iterasi Firefly

Iterasi *Firefly* adalah proses yang berulang, melibatkan evaluasi dan pembaruan posisi kunang-kunang untuk secara bertahap menemukan solusi terbaik melalui interaksi berdasarkan intensitas cahaya.



Gambar 3.20 Grafik Pengujian Iterasi Kunang-Kunang

Berdasarkan Gambar 3.20 menunjukkan grafik pengujian Iterasi kunang-kunang mendapatkan hasil akurasi tertinggi pada nilai sebesar 0,86347032. Sehingga nilai pengujian jumlah kunang-kunang terbaik adalah 9.

3.3.11 Parameter Akhir Firefly Optimization

Parameter akhir dilakukan untuk mengetahui parameter yang terbaik setelah melakukan pengujian dari parameter awal dalam model algoritme *Firefly Optimization*. Berikut adalah parameter akhir algoritme *Firefly Optimization*.

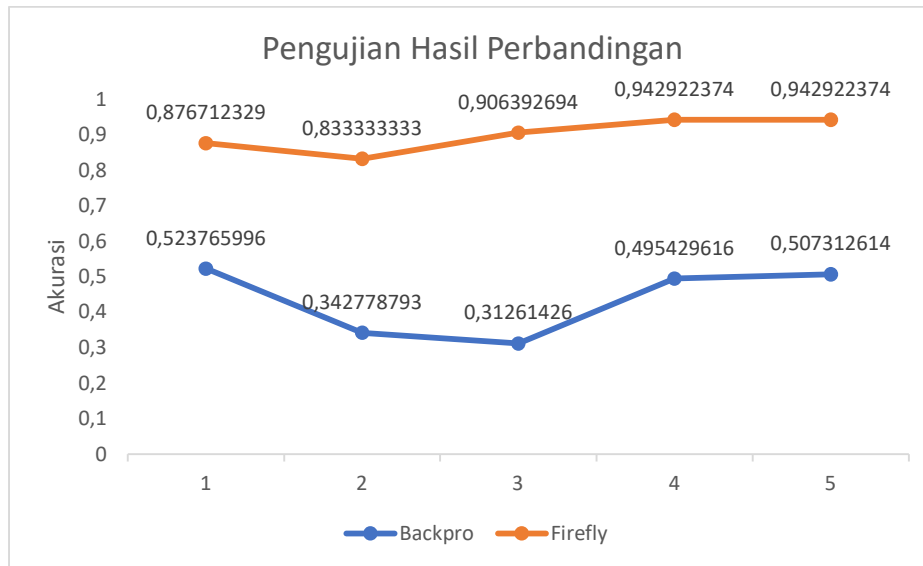
Tabel 3.4 Parameter Akhir Firefly Optimization

<i>Firefly</i> (Kunang-kunang)	Iterasi	Akurasi
6	9	0,90045662

Berdasarkan pada Tabel 3.4 menunjukkan parameter akhir yang akan digunakan untuk pengujian yaitu dengan nilai *Firefly* yang berjumlah 6, dan nilai iterasi yang berjumlah 9 dengan akurasi 0,90045662.

3.3.12 Hasil Perbandingan

Hasil perbandingan dilakukan untuk mengetahui perbedaan antara metode *Backpropagation* dan *Backpropagation* dengan optimasi *Firefly*.



Gambar 3.21 Hasil Perbandingan

Berdasarkan Gambar 3.21 menunjukkan grafik hasil perbandingan antara *Backpropagation* dan *Backpropagation* dengan optimasi *Firefly*. Perbandingan menunjukkan hasil *Backpropagation* dengan optimasi *Firefly* menghasilkan akurasi yang lebih baik dibandingkan *Backpropagation* tanpa optimasi.

3.4 Hasil

Hasil pada kedua metode dilakukan sebanyak 5 kali pengujian, kemudian rata-rata nilai pada pengujian dihitung. *Backpropagation* dengan optimasi *Firefly* menghasilkan nilai akurasi sebesar 90%, sementara *Backpropagation* tanpa optimasi hanya mendapatkan nilai akurasi sebesar 43%. Dapat disimpulkan bahwa dengan optimasi *Firefly* berhasil meningkatkan nilai akurasi sebesar 47%.

BAB IV PENUTUP

4.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan tentang Penerapan algoritme *Firefly* untuk optimasi metode *Backpropagation* dalam identifikasi jenis Penyakit tanaman Padi, maka dapat diambil kesimpulan sebagai berikut :

- 1) Hasil metode *Backpropagation* menghasilkan nilai akurasi sebesar 43% dalam menentukan jenis penyakit pada tanaman padi. Dalam metode *Backpropagation* nilai akurasi bergantung pada parameter yang digunakan seperti *learning rate*, *k-fold*, rasio data, *epoch*, dan *hidden layer*.
- 2) Penerapan optimasi *Firefly* dalam memperbaiki pengoptimalan bobot pada *Backpropagation* memberikan dampak yang positif. Optimasi *Firefly* dapat membantu kinerja *Backpropagation* dengan mengoptimalkan bobot, sehingga proses *training* menjadi efisien. Hal ini menghasilkan peningkatan pada nilai akurasi dalam klasifikasi penyakit pada tanaman padi.

4.2 Saran

Berdasarkan hasil penelitian yang didapat, perlu dikembangkan agar hasil dalam mengklasifikasikan lebih akurat sehingga dapat membantu kebutuhan pengguna. Pada penelitian ini menggunakan program *Backpropagation* dengan optimasi *Firefly* dengan menghasilkan nilai yang tinggi. Selanjutnya dapat diteliti bagaimana implementasi *Backpropagation* untuk pengklasifikasian dengan optimasi yang berbeda.

Daftar Pustaka

- Adiba, F., Nur Risal, A.A. and Tahir, M. (2023) 'Implementasi Algoritma Backpropagation untuk Klasifikasi Kualitas Susu Sapi', *Jurnal MediaTIK*, 6(2), p. 42. Available at: <https://doi.org/10.26858/jmtik.v6i2.46013>.
- Athailah, M., Azhar, Y. and Munarko, Y. (2020) 'Perbandingan Metode Klasifikasi Berita Hoaks Berbahasa Indonesia Berbasis Pembelajaran Mesin', *Jurnal Repositor*, 2(5), pp. 675–682. Available at: <https://doi.org/10.22219/repositor.v2i5.692>.
- Bohari, A.R. and Teknik, F. (2022) 'Meningkatkan Kinerja Backpropagation Neural Network Menggunakan Algoritma Adaptif', 3(1), pp. 1–6.
- Christiawan, G.Y. *et al.* (2023) 'Penerapan Metode Convolutional Neural Network (CNN) Dalam Mengklasifikasikan Penyakit Daun Tanaman Padi Informasi Artikel Abstrak', *Jurnal of Information and Technology*, 65146(204), pp. 295–305.
- Damanik, B. *et al.* (2022) 'Jurnal Teknologi , Kesehatan Dan Ilmu Sosial', 4(2), pp. 328–338.
- Dewi, M.M. (2022) 'Optimasi Pearson Correlation untuk Sistem Rekomendasi menggunakan Algoritma Firefly', 9(1), pp. 1–5.
- Gunawan, V.A. and Putra, L.S.A. (2021) 'Comparison of American Sign Language Use Identification using Multi-Class SVM Classification, Backpropagation Neural Network, K - Nearest Neighbor and Naive Bayes', *Teknik*, 42(2), pp. 137–148. Available at: <https://doi.org/10.14710/teknik.v42i2.36929>.
- Handayani, R.N. *et al.* (2023) 'OPTIMASI SUPPORT VECTOR MACHINE MENGGUNAKAN PARTICLE SWARM OPTIMIZATION', 5(1), pp. 120–126.
- Hawar, D.A.N. *et al.* (2023) 'Digital Daun Padi Menggunakan Metode', 5(2), pp. 334–346.
- Hawari, F.H. *et al.* (2022) 'KLASIFIKASI PENYAKIT PADI MENGGUNAKAN ALGORITMA CNN (CONVOLUTIONAL NEURAL NETWORK)', 4(2), pp. 184–189.
- Ikhsan, M., Armansyah, A. and Tamba, A.A. (2022) 'Implementasi Jaringan Syaraf Tiruan Backpropagation Pada Klasifikasi Grade Teh Hitam', *Jurnal Sistem Komputer dan Informatika (JSON)*, 4(2), p. 387. Available at: <https://doi.org/10.30865/json.v4i2.5312>.
- Indrawan, A.M. and Pandu Kusuma, A. (2021) 'Analisis Algoritma Jaringan Syaraf Tiruan Dengan Metode Backpropagation Dalam Mendeteksi Keahlian Mahasiswa Program Studi Teknik Informatika Universitas Islam Balitar', *Jurnal Mnemonic*, 5(1), pp. 9–13. Available at: <https://doi.org/10.36040/mnemonic.v5i1.4272>.
- Meidianingsih, Q., Eka, D. and Meganingtyas, W. (no date) 'ANALISIS PERBANDINGAN PERFORMA METODE ENSEMBLE DALAM MENANGANI IMBALANCED MULTI-CLASS', pp. 13–21.
- Murniyasih, E. *et al.* (2020) 'PENERAPAN METODE LEARNING VECTOR QUANTIZATION', 6(1).
- Nasution, S.L., Limbong, C.H. and Ramadhan, D.A. (2020) 'PENGARUH KUALITAS PRODUK, CITRA MEREK, KEPERCAYAAN, KEMUDAHAN, dan HARGA TERHADAP KEPUTUSAN PEMBELIAN PADA E-COMMERCE SHOPEE (Survei pada Mahasiswa S1 Fakultas Ekonomi

- Jurusan Manajemen Universitas Labuhan Batu), *Ecobisma (Jurnal Ekonomi, Bisnis Dan Manajemen)*, 7(1), pp. 43–53. Available at: <https://doi.org/10.36987/ecobi.v7i1.1528>.
- Norhikmah, N. and Rumini, R. (2020) ‘Klasifikasi Peminjaman Buku Menggunakan Neural Network Backpropagation’, *Sistemasi*, 9(1), p. 1. Available at: <https://doi.org/10.32520/stmsi.v9i1.562>.
- Nugroho, M.S. and Nurraharjo, E. (2023) ‘KLASIFIKASI HAMA TANAMAN PADI BERDASARKAN CITRA DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK’, *Jurnal Pendidikan Biologi dan Sains*, 4(1), pp. 88–100.
- Nurhikam, A.S. *et al.* (2024) ‘Klasifikasi Tanaman Obat Berdasarkan Citra Daun Menggunakan Backpropagation Neural Networks Classification of Medicinal Plants Based on Leaf Image Using Backpropagation Neural Networks’, (1), pp. 1–11. Available at: <https://doi.org/10.26798/jiko.v8i1.945>.
- Permadi, J., Rhomadhona, H. and Aprianti, W. (2021) ‘Perbandingan K-Nearest Neighbor Dan Backpropagation Neural Network Dalam Prediksi Resiko Diabetes Tahap Awal’, *Kumpulan jurnal Ilmu Komputer (KLIK)*, 08(3), pp. 352–365.
- Peryanto, A., Yudhana, A. and Umar, R. (2020) ‘Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation’, *Journal of Applied Informatics and Computing*, 4(1), pp. 45–51. Available at: <https://doi.org/10.30871/jaic.v4i1.2017>.
- Purwadi, P. and Nasyuha, A.H. (2022) ‘Implementasi Teorema Bayes Untuk Diagnosa Penyakit Hawar Daun Bakteri (Kresek) Dan Penyakit Blas Tanaman Padi’, *JURIKOM (Jurnal Riset Komputer)*, 9(4), p. 777. Available at: <https://doi.org/10.30865/jurikom.v9i4.4350>.
- Putri, A.W. (2021) ‘Implementasi Artificial Neural Network (ANN) Backpropagation Untuk Klasifikasi Jenis Penyakit Pada Daun Tanaman Tomat’, *MATHunesa: Jurnal Ilmiah Matematika*, 9(2), pp. 344–350. Available at: <https://doi.org/10.26740/mathunesa.v9n2.p344-350>.
- Ridho, I.I. *et al.* (2023) ‘Optimasi Fungsi Pembelajaran Jaringan Saraf Tiruan dalam Meningkatkan Akurasi pada Prediksi Ekspor Kopi Menurut Negara Tujuan Utama’, 4(4). Available at: <https://doi.org/10.47065/bits.v4i4.3240>.
- Sari, P.K. and Suryono, R.R. (2024) ‘Komparasi Algoritma Support Vector Machine Dan Random Forest Untuk Analisis Sentimen Metaverse’, *Jurnal Mnemonic*, 7(1), pp. 31–39. Available at: <https://doi.org/10.36040/mnemonic.v7i1.8977>.
- Sastypratiwi, H., Muhandi, H. and Noveanto, M. (2022) ‘Klasifikasi Emosi Pada Lirik Lagu Menggunakan Algoritma Multiclass SVM dengan Tuning Hyperparameter PSO’, 6, pp. 2279–2286. Available at: <https://doi.org/10.30865/mib.v6i4.4609>.
- Sholikhah, S., Kurniadi, D. and Riansyah, A. (2021) ‘Sistem Pakar Menggunakan Metode Forward Chaining untuk Diagnosa Hama dan Penyakit Tanaman Padi’, 2(2), pp. 103–110.
- Simamora, Y., Hajar, I. and Fernandes, A. (2019) ‘Penerapan Algoritma Kunang – Kunang (Firefly Algorithm) untuk Optimasi Rekonfigurasi Jaringan Distribusi Radial’, *Energi & Kelistrikan*, 11(2), pp. 71–79. Available at: <https://doi.org/10.33322/energi.v11i2.498>.
- Sistem, R. *et al.* (2021) ‘JURNAL RESTI Optimasi Parameter Support Vector Machine Berbasis Algoritma Firefly’, 5(10), pp. 904–910.
- Sulistiyanto, S., Saputri, T.A. and Noviyanti, N. (2022) ‘Deteksi Dini Hama dan Penyakit Padi Menggunakan Metode Certainty Factor’, *JURIKOM (Jurnal Riset Komputer)*, 9(1), p. 48. Available at:

<https://doi.org/10.30865/jurikom.v9i1.3778>.

Xinhao, Z. and Liang, X.U. (2023) ‘储能测试与评价 改进的萤火虫算法优化反向传播神经网络动力电池健康状态估计 Improved firefly optimization algorithm to optimize back propagation neural network for state of health estimation of power lithium ion batteries’. Available at: <https://doi.org/10.19799/j.cnki.2095-4239.2022.0668>.

Zhao, Q. *et al.* (2021) ‘Stepped generalized predictive control of test tank temperature based on backpropagation neural network’, *Alexandria Engineering Journal*, 60(1), pp. 357–364. Available at: <https://doi.org/10.1016/j.aej.2020.08.032>.

Zurkarnain, D.Z., Damayanti, A. and Winarko, E. (2021) ‘Hybrid Jaringan Saraf Tiruan Backpropagation dengan Firefly Algorithm dan Simulated Annealing untuk Peramalan Curah Hujan di Surabaya’, 3(1), pp. 56–70.

LAMPIRAN

Lampiran 1 Kuesioner Penelitian

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 putih	1	0	1	1	1	1
G3	Bulir bercak coklat 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 coklat kehijauan pada batang	1	1	1	0	1	1
G9	Batang mudah patah	1	1	1	1	1	1
G10	Mata hampan atau tidak berisi	1	1	1	1	1	1

P3 KRESEK

Kode	Gejala	D1	D2	D3	D4	D5	D6
G11	Daun berwarna putih kekuningan	1	1	1	0	1	1
G12	Layu	1	0	1	0	0	1
G13	Batang berwarna coklat	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 coklat gelap	0	1	0	1	1	1
G17	Bercak tua berukuran lebih besar (0,4 - 1 cm x 1 - 0,2 cm)	0	1	1	1	1	0
G18	Bercak berwarna kuning di sekelilingnya	1	1	1	1	0	1
G19	Bulir berwarna coklat kehijauan	1	1	1	1	1	1

P5 BERCAK COKELAT SEMPIIT

Kode	Gejala	D1	D2	D3	D4	D5	D6
G12	Layu	1	1	0	1	1	1
G20	Pada daun dan petalok daun terdapat bercak coklat yang sempit	1	1	0	1	1	1
G21	Varietas yang tahan bercak berukuran 0,2 - 1 cm x 0,1 cm, berwarna coklat gelap	1	0	1	1	1	1
G22	Varietas bercak lebih besar & berwarna coklat 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 coklat 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 terjalim	1	1	1	1	1	1

P9 Kerdil Rumpit

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 kepudutan	1	1	1	0	1	1

Maka terdapat
Sangat Menghina
Menghina



Lampiran 2 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



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>SKMKT</i>	Diterima Tgl : <i>26 04 2024</i> No. Agenda : <i>137</i>
No. Surat : <i>003.0-15/FS.1/D3/C/2024</i>	<input type="checkbox"/> Sangat Segera <input type="checkbox"/> Segera <input type="checkbox"/> Rahasia
Tgl. Surat : <i>25 APRIL 2024</i>	
Perihal : <i>PERMOHONAN PENCAMBIAN DATA</i>	
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"/> Buatlah 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>Sda. Wawan</i> <i>[Signature]</i> <i>26/4.24</i>

Lampiran 3 Lembar Konsultasi

KARTU KENDALI BIMBINGAN SKRIPSI

Nama Mahasiswa : Muhammad Hafizh Athoriq
 NIM : 2011102441184
 Nama Dosen Pembimbing : Fendy Yulianto, S.Kom., M.Kom.
 Judul Penelitian : Penerapan Algoritma firefly untuk optimasi metode Backpropagation dalam identifikasi jenis Penyakit tanaman Padi

15	20 Juni 2024	Menambahkan diagram alir pada Bab 2, dan flowchart di pindahin ke Bab 3	
16	25 Juni 2024	Menambahkan hasil pengujian, dan implementasi program pada Bab 3. Pada Bab 4 di isi dengan kesimpulan dan saran	

Dosen Pembimbing

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

Mengetahui

 Program Studi

 S.Kom., M.TI
 NIDN. 118019203

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	3 Februari 2024	Konsultasi Judul Penelitian, data, metode dan optimasi yang digunakan	
2	7 Februari 2024	Konsultasi penulisan bab 1	
3	19 Februari 2024	Revisi penulisan bab 1 terkait penulisan kalimat dan paragraf yang salah dan kurang	
4	26 Februari 2024	Perbaiki kalimat pada pendahuluan, rumusan masalah di tambahkan, tujuan dan manfaat	
5	13 Maret 2024	Revisi penulisan dan penambahan pada Bab 2 terkait penulisan rumus, poin poin, dan objek penelitian	
6	19 Maret 2024	Konsultasi data yang akan digunakan	
7	4 April 2024	Konsultasi data yang diperoleh dari wawancara	
8	21 April 2024	Data yang di dapat masih kurang dan perlu ditambahkan lagi	
9	30 April 2024	Konsultasi penulisan Bab 1, dan Bab 2 ditambaka sub bab baru dan flowchart penelitian	
10	6 Mei 2024	Konsultasi cara perhitungan manual excel untuk permodelan Backpropagation	
11	4 Juni 2024	Konsultasi Bab 2 untuk memperbaiki dan menambahkan teori tentang One Versus All dan One Versus One, multiclass, dan evaluasi	
12	7 Juni 2024	Konsultasi permodelan Backpropagation dan menambahkan optimasi Firefly untuk metode backpropagation pada permodelan	
13	10 Juni 2024	Konsultasi perhitungan manual optimasi Firefly pada excel untuk permodelan Backpropagation	
14	15 Juni 2024	Menambahkan teori tentang K-Fold cross validation pada Bab 2	

Lampiran 4 Wawancara Dengan Petani





Muhammad Hafizh Atthoriq, dilahirkan pada tanggal 21 September 2002 di Samarinda, Kalimantan Timur. Penulis adalah anak pertama dari pasangan Bapak Abdul Syahid dan Ibu Ekawati Syahrial. Penulis memulai Pendidikan di PAUD Al-Qalam Kecamatan Samarinda Utara selama 1 tahun dan lulus pada tahun 2008. Kemudian melanjutkan Pendidikan di SD Muhammadiyah 1 Samarinda Kota dari kelas 1 sampai dengan kelas 6 dan lulus pada tahun 2014.

Penulis melanjutkan Pendidikan ke jenjang Sekolah Menengah Pertama di SMP IT MADINA Sungai Pinang selama 3 tahun dan lulus pada tahun 2017. Selanjutnya penulis melanjutkan Sekolah Menengah Kejuruan di SMK Negeri 7 Samarinda Kota 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.