

**PENERAPAN ALGORITMA *BACKPROPAGATION* DALAM
KLASIFIKASI PENYAKIT RETINOPATI DIABETIK
MENGUNAKAN *PARTICLE SWARM OPTIMIZATION***

SKRIPSI

BABY ANDINI

191401010



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

**PENERAPAN ALGORITMA BACKPROPAGATION DALAM KLASIFIKASI PENYAKIT
RETINOPATI DIABETIK MENGGUNAKAN PARTICLE SWARM OPTIMIZATION**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer

BABY ANDINI

191401010



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

Persetujuan

Judul : PENERAPAN ALGORITMA *BACKPROPAGATION*
DALAM KLASIFIKASI PENYAKIT RETINOPATI
DIABETIK MENGGUNAKAN *PARTICLE SWARM*
OPTIMIZATION

Kategori : SKRIPSI

Nama : BABY ANDINI

Nomor Induk Mahasiswa : 191401010

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Departemen : ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 05 Januari 2024

Komisi Pembimbing :
Pembimbing 2

Dian Rachmawati S.Si, M.Kom
NIP. 198307232009122004

Pembimbing 1

Amer Sharif S.Si, M.Kom
NIP. 196910212021011001

Diketahui/disetujui oleh
Program Studi S-1 Ilmu Komputer

Ketua,

Dr Amalia, S.T., M.T.
NIP. 197812212014042001

PERNYATAAN

**PENERAPAN ALGORITMA *BACKPROPAGATION* DALAM KLASIFIKASI
PENYAKIT RETINOPATI DIABETIK MENGGUNAKAN *PARTICLE SWARM*
*OPTIMIZATION***

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 05 Januari 2024



Baby Andini

191401010

PENGHARGAAN

Bissmiillahirrahmannirrahiim. Alhamdulillahirabbil'alamin, puji dan syukur atas segala berkat dan Rahmat Allah SWT yang telah memberikan kemudahan, kesempatan, dan pertolongan-Nya sehingga penulis dapat menyelesaikan tugas akhir ini sebagai syarat untuk mendapatkan gelar Sarjana Komputer di Program Studi S-1 Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Dengan segala kerendahan hati, penulis ingin menyampaikan rasa terimakasih yang sebesar-besarnya kepada seluruh pihak yang bertanggung jawab serta memberikan dukungan, dan do'a kepada penulis dalam proses pengerjaan tugas akhir ini, antara lain:

1. Bapak Dr. Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Amer Sharif, S.Si, M.Kom selaku Dosen Pembimbing I yang senantiasa memberikan bimbingan, saran, nasihat, serta motivasi kepada penulis selama melakukan pengerjaan tugas akhir.
5. Ibu Dian Rachmawati S.Si, M.Kom selaku Dosen Pembimbing II yang senantiasa memberikan bimbingan, saran, serta motivasi kepada penulis selama melakukan pengerjaan tugas akhir.
6. Seluruh dosen dan pegawai di Program Studi S-1 Ilmu Komputer Fasilkom-TI Universitas Sumatera Utara.
7. Kedua orang tua tercinta, Ayahanda Dr. Budi Utomo SP.MP. dan Ibunda Rahimah Andayani S. yang selalu memberikan semangat dan dukungan baik secara moral dan materil, memberikan do'a, perhatian, kesabaran dan kasih sayang yang berlimpah kepada penulis.

8. Adik-adik tersayang Satria Jeverton, Shirley Utami, Aziz Nugroho, Wiwit Novira, dan Chika Aqila yang selalu menghibur dan memberikan kasih sayang kepada penulis dalam proses pengerjaan tugas akhir.
9. Senior yang selalu memberikan dukungan, motivasi, serta semangat kepada penulis terutama Muhammad Ridho dan senior lainnya yaitu Ansya Purnama Agatha, dan Muhammad Fachrozy.
11. Sahabat-sahabat seperjuangan “Lambe Turah” yaitu Kiki Safrina, Elfina Hutagalung, Nabillah Syahwina Batubara, dan Tarmana Ebenezer Sinuhaji yang senantiasa memberikan hiburan, semangat dan dukungan kepada penulis.
12. Sahabat-sahabat seperjuangan “Nocturnal” yaitu Puti Melur Maulinda, Siti Zuhairah, dan Dwi Utami Putri yang senantiasa memberikan semangat, dukungan serta kerjasamanya selama perkuliahan kepada penulis.
13. Sahabat-sahabat seperjuangan “Toxic” yaitu Anggita Mutiara Sari Siregar, Elfina Hutagalung, Adellia Balqis Nasution, Kiki Safrina, Christine Natalia Nainggolan, Tarmana Ebenezer Sinuhaji, Yoshepine Ignacia, Nabillah Syahwina Batubara, Dwivia Wahyu Amanda, Niza Syahfitri, yang senantiasa memberikan semangat dan dukungan kepada penulis.
13. Teman-teman stambuk 2019 khususnya Kom A yang telah memberikan kerjasama dan saling membantu semasa perkuliahan.
14. Semua pihak yang telah membantu penulis dalam penyelesaian skripsi ini yang tidak dapat disebutkan satu per satu.

Semoga Allah SWT membalas semua amal kebaikan dan senantiasa dilimpahkan kasih sayang, berkah dan rahmat-Nya kepada semua pihak yang telah memberikan bantuan, motivasi, semangat, perhatian, serta dukungan kepada penulis dalam menyelesaikan skripsi ini.

Medan, 05 Januari 2024

Penulis,



Baby Andini

ABSTRAK

Retinopati Diabetik merupakan penyakit yang diakibatkan dari diabetes melitus yang dapat menyebabkan kematian. Penyakit ini dapat mengganggu penglihatan sehingga dapat mengalami kebutaan apabila tidak segera ditangani. Diperlukan teknologi informasi untuk melakukan pencegahan penyakit Retinopati Diabetik. Pengklasifikasian pada data Retinopati Diabetik dilakukan perbandingan menggunakan algoritma *Backpropagation* dan algoritma *Backpropagation* yang dioptimasi *Particle Swarm Optimization*. Pada penelitian ini digunakan 383 data yang telah dilakukan preprocessing data dari 1.151 data Retinopati Diabetik yang bersumber dari *UCI Machine Learning*. Hasil akurasi dari klasifikasi data mining Retinopati Diabetik sebanyak 1.151 dataset menggunakan algoritma *Backpropagation* mendapatkan nilai 61% sedangkan algoritma *Backpropagation* yang dioptimasi dengan *Particle Swarm Optimization* mencapai 55%. Dan hasil akurasi dari klasifikasi data mining Retinopati Diabetik yang telah dilakukannya preprocessing data sebanyak 383 dataset menggunakan algoritma *Backpropagation* mendapatkan nilai 70% sedangkan algoritma *Backpropagation* yang dioptimasi dengan *Particle Swarm Optimization* mencapai 65%.

Kata Kunci: *Data Mining*, Klasifikasi, Retinopati Diabetik, *Backpropagation*, *Particle Swarm Optimization*.

APPLICATION OF THE BACKPROPAGATION ALGORITHM IN THE CLASSIFICATION OF DIABETIC RETINOPATHY USING PARTICLE SWARM OPTIMIZATION

ABSTRACT

Diabetic Retinopathy is a disease resulting from diabetes mellitus which can cause death. This disease can interfere with vision that can make a blindness too if it's not treated immediately. Information technology is needed to prevent Diabetic Retinopathy. The classification of Diabetic Retinopathy data was compared using the Backpropagation algorithm and the Backpropagation algorithm with optimized by Particle Swarm Optimization. In this study, 383 data were used which had been preprocessing data from 1.151 Diabetic Retinopathy data sourced from UCI Machine Learning. The accuracy results of the Diabetic Retinopathy data mining classification of 1.151 dataset using the Backpropagation algorithm achieved a value of 61%, while the Backpropagation algorithm optimized with Particle Swarm Optimization reached 55%. And the accuracy results of the Diabetic Retinopathy data mining classification that has been preprocessing data as much as 383 dataset using the Backpropagation algorithm achieved a value of 70%, while the Backpropagation algorithm optimized with Particle Swarm Optimization reached 65%.

Keywords: Data Mining, Classification, Diabetic Retinopathy, Backpropagation, Particle Swarm Optimization.

DAFTAR ISI

	Halaman
Persetujuan	ii
Pernyataan	iii
Penghargaan	iv
Abstrak	vi
<i>Abstract</i>	vii
Daftar Isi	viii
Daftar Tabel	xi
Daftar Gambar	xiii
Daftar Lampiran	xvi
 BAB 1 PENDAHULUAN	 1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	3
1.7. Sistematika Penulisan	4
 BAB 2 LANDASAN TEORI	 5
2.1. <i>Data Mining</i>	5
2.1.1. Fungsi <i>Data Mining</i>	5
2.1.2. Tujuan <i>Data Mining</i>	6
2.2. Klasifikasi	6
2.3. Jaringan Syaraf Tiruan (JST)	7
2.3.1. <i>Backpropagation – Neural Network</i>	7
2.3.1.1. Perhitungan <i>Backpropagation-Neural Network</i>	11

2.4. <i>Particle Swarm Optimization (PSO)</i>	16
2.5. Retinopati Diabetik	17
2.5.1. Klasifikasi Retinopati Diabetik	18
2.6. <i>Confusion Matrix</i>	19
2.9. Penelitian Relevan	20
 BAB 3 ANALISIS DAN PERANCANGAN	 22
3.1. Analisis Sistem	22
3.1.1. Analisis Masalah	22
3.1.2. Analisis Persyaratan	23
3.1.3. Analisis Proses	24
3.2. Pemodelan Sistem	24
3.2.1. Diagram Umum	25
3.2.2. <i>Use Case Diagram</i>	26
3.2.3. <i>Activity Diagram</i>	26
3.2.3.1. <i>Activity Diagram</i> pada <i>Backpropagation</i>	26
3.2.3.1. <i>Activity Diagram</i> pada <i>Backpropagation</i> dengan <i>Particle Swarm Optimization</i>	27
3.2.4. <i>Sequence Diagram</i>	28
3.2.5. <i>Flowchart</i> (Diagram Alir)	29
3.2.5.1. <i>Flowchart</i> Sistem	29
3.2.5.2. <i>Flowchart</i> Algoritma <i>Backpropagation</i>	31
3.2.5.2. <i>Flowchart</i> Algoritma <i>Particle Swarm Optimization</i>	31
3.2.6. Perancangan <i>User Interface</i>	32
 BAB 4 IMPLEMENTASI DAN PENGUJIAN	 36
4.1 Implementasi	36
4.1.1. Halaman Utama	37
4.1.2. Halaman <i>Backpropagation</i>	38
4.1.3. Halaman <i>Particle Swarm Optimization</i>	38

4.1.4. Halaman Dataset	39
4.1.5. Halaman Hasil	40
4.2. Pengolahan Dataset Sistem	41
4.2.1. Transformasi Data	41
4.2.2. Data <i>Reduction</i>	42
4.3. Pengujian Sistem	43
4.3.1. Dataset <i>Original</i> (1.151 Data)	44
4.3.1.1. Hasil Pengujian Algoritma <i>Backpropagation</i>	44
4.3.1.2. Hasil Pengujian Algoritma <i>Particle Swarm Optimization dan Backpropagation</i>	47
4.3.2. Dataset <i>Reduction</i> (383 Data)	49
4.3.2.1. Hasil Pengujian Algoritma <i>Backpropagation</i>	49
4.3.2.2. Hasil Pengujian Algoritma <i>Particle Swarm Optimization dan Backpropagation</i>	52
4.3. Hasil Perbandingan Pengujian Algoritma	54
 BAB 5 KESIMPULAN DAN SARAN	 55
5.1. Kesimpulan	55
5.2. Saran	55
 DAFTAR PUSTAKA	 57

DAFTAR TABEL

2.1. Tabel Contoh Data	12
2.2. Tabel Bobot Awal V_{ij} Contoh Data	13
2.3. Tabel Bobot Awal W_{ij} Contoh Data	13
2.4. Tabel Contoh Data Uji	15
2.5. Tabel Contoh Data Bobot Baru V_{ij}	15
2.6. Tabel Contoh Data Bobot Baru W_{ij}	15
2.7. Tabel Hasil Contoh Data Uji	16
4.1. Tabel Keterangan Dataset Retinopati Diabetik	36
4.2. Tabel Hasil Akurasi <i>Backpropagation</i>	45
4.3. Tabel Hasil Akurasi <i>Backpropagation + PSO</i>	47
4.4. Tabel Hasil Akurasi <i>Backpropagation</i>	50
4.5. Tabel Hasil Akurasi <i>Backpropagation + PSO</i>	52
4.6. Tabel Hasil Akurasi Algoritma	54

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Proses <i>Data Mining</i>	6
Gambar 2.2. Sel Syaraf Manusia	7
Gambar 2.3. Arsitektur <i>Backpropagation</i>	8
Gambar 2.4. Contoh Arsitektur <i>Backpropagation</i>	12
Gambar 2.5. Retinopati Diabetik	18
Gambar 2.6. <i>Confusion Matrix</i>	19
Gambar 3.1. <i>Diagram Ishikawa</i> Analisis Masalah	23
Gambar 3.2. Diagram Umum Sistem	25
Gambar 3.3. <i>Use case diagram</i>	26

Gambar 3.4.	<i>Activity diagram pada Backpropagation</i>	27
Gambar 3.5.	<i>Activity diagram pada Backpropagation dengan Particle Swarm Optimization</i>	28
Gambar 3.6.	<i>Sequence Diagram</i>	29
Gambar 3.7.	<i>Flowchart Sistem</i>	30
Gambar 3.8.	<i>Flowchart Backpropagation</i>	31
Gambar 3.9.	<i>Flowchart Particle Swarm Optimization</i>	32
Gambar 3.10.	Halaman Utama	33
Gambar 3.11.	Halaman Dataset	34
Gambar 3.12.	Halaman Hasil	35
Gambar 4.1.	Halaman Utama	37
Gambar 4.2.	Halaman <i>Backpropagation</i>	38
Gambar 4.3.	Halaman <i>Particle Swarm Optimization</i>	39
Gambar 4.4.	Halaman dataset	40
Gambar 4.5.	Halaman hasil	41
Gambar 4.6.	File .csv (Sebelum)	42
Gambar 4.7.	File .xls (Sesudah)	42
Gambar 4.8.	Dataset sebelum transformasi	43
Gambar 4.9.	Dataset sesudah transformasi	43
Gambar 4.10.	<i>Confusion Matrix Backpropagation</i>	45
Gambar 4.11.	Hasil Perhitungan <i>Confusion Matrix Backpropagation</i>	45
Gambar 4.12.	<i>Confusion Matrix Backpropagation+PSO</i>	47
Gambar 4.13.	Hasil Perhitungan <i>Confusion Matrix Backpropagation+PSO</i>	47
Gambar 4.14.	<i>Confusion Matrix Algoritma Backpropagation</i>	49
Gambar 4.15.	<i>Confusion Matrix Algoritma PSO + Backpropagation</i>	52

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 <i>Curiculumm Vitae</i>	A-1

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Penyakit Retinopati Diabetes merupakan penyakit yang diakibatkan dari diabetes melitus. Retinopati Diabetes termasuk penyakit yang menyebabkan kematian. Penyebab terjadinya Retinopati Diabetes ini adalah tingginya kadar glukosa dalam darah atau memecahnya pembuluh darah di retina. Umumnya penyakit ini akan mengganggu penglihatan mata dan dapat mengakibatkan kebutaan jika tidak cepat ditangani. Dalam hal ini, maka diperlukan teknologi informasi yang dapat melakukan deteksi dini untuk mencegah penyakit Retinopati Diabetik (Subarkah et al., 2021).

Data mining merupakan sebuah alur ekstraksi informasi yang bertujuan untuk menemukan pola penting (*pattern recognition*) didalam suatu tumpukan data sehingga menjadi pengetahuan (*knowledge discovery*). Beberapa fungsi data mining yaitu deskripsi, estimasi, prediksi, klustering, klasifikasi dan asosiasi. Pada kasus mendiagnosa penyakit Retinopati Diabetik ini digunakan fungsi klasifikasi untuk menentukan pasien yang mengidap Retinopati Diabetik.

Klasifikasi bertujuan untuk melihat kelas objek yang belum diketahui labelnya dengan cara menemukan model yang menggambarkan dan membedakan kelas data (Hizham, 2018). Terdapat beberapa algoritma yang dapat digunakan dalam Data Mining Klasifikasi, contohnya algoritma *K- Nearest Neighbor*, *Neural Network*, *Naive Bayes*, *Decision Tree*, dan *Logistic Regresion* (Khoiruzzaman et al., 2021).

Jaringan Syaraf Tiruan (JST) merupakan algoritma komputasi yang cara kerjanya sama seperti neuron atau sistem jaringan saraf otak manusia (Norhikmah dan Rumini, 2019). Algoritma *Backpropagation* merupakan algoritma pembelajaran dari Jaringan Syaraf Tiruan (JST). Algoritma *Backpropagation* adalah *supervised learning* yang terdiri dari banyak lapisan. Terdapat tiga fase dalam melakukan *Backpropagation training*, yaitu *feedforward*, *backpropagation*, dan penyesuaian bobot (Nurdiansyah et al., 2018). *Backpropagation* salah satu algoritma yang kerap dipakai sebagai pengklasifikasian. Algoritma *Backpropagation* memiliki kemampuan dalam memecahkan masalah yang berskala besar dan mampu mengenali pola yang rumit dengan membangun metode terlatih yang mengeluarkan kinerja yang baik. Namun, hambatan dari algoritma *Backpropagation* harus memiliki data

training besar dan pengoptimasian agar dapat menghasilkan hasil yang baik. Untuk melengkapi permasalahan tersebut, maka algoritma *Backpropagation* dapat diatasi dengan salah satu algoritma optimasi yaitu algoritma *Particle Swarm Optimization*.

Particle Swarm Optimization (PSO) merupakan algoritma optimasi yang berguna untuk pengambilan keputusan atau mengoptimalkan perancangan jalur. PSO ditemukan oleh Eberhart dan Kennedy pada tahun 1995 yang terinspirasi dari perilaku sekumpulan burung (Purnama et al., 2021). Algoritma yang mudah diimplementasikan, sederhana, dan efisien ini menjadi salah satu kelebihan dari algoritma optimasi *Particle Swarm Optimization* (PSO) (Arifin et al., 2019). Dalam penelitian ini, Algoritma *Backpropagation* akan dikombinasi dengan metode optimasi *Particle Swarm Optimization* (PSO) yang berfungsi untuk memaksimalkan hasil pengklasifikasian dengan dataset yang berjumlah besar dan dapat meningkatkan nilai akurasi dalam mengidentifikasi penyakit Retinopati Diabetik.

Penelitian yang dilakukan oleh Tandrio et al (2018) dengan judul “*Classification of Wine Types Based on Composition Using Backpropagation And Particle Swarm Optimization*” menyatakan bahwa *Neural Network* berbasis PSO menghasilkan hasil lebih baik daripada tidak menggunakan PSO. NNPSO dapat mengklasifikasikan semua jenis *wine* sedangkan NNBP hanya dapat mengklasifikasikan satu jenis *wine* saja.

Berdasarkan permasalahan tersebut, maka penulis mengimplementasikan algoritma *Backpropagation* dengan menggunakan algoritma optimasi *Particle Swarm Optimization* dalam klasifikasi penyakit Retinopati Diabetik.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang sebelumnya, maka rumusan masalah yang akan diambil dalam penelitian ini adalah bagaimana mengimplementasikan algoritma *Backpropagation* pada penyakit Retinopati Diabetik menggunakan metode optimasi *Particle Swarm Optimization*.

1.3. Batasan Masalah

Penelitian ini memiliki batasan masalah sebagai berikut :

1. Menggunakan algoritma *Backpropagation* untuk melakukan klasifikasi data mining.
2. Menggunakan algoritma optimasi *Particle Swarm Optimization*.
3. Membandingkan hasil akurasi klasifikasi dari algoritma *Backpropagation* dengan hasil akurasi algoritma *Backpropagation + Particle Swarm Optimization*.
4. Data klasifikasi yang digunakan ialah data penelitian penyakit Retinopati Diabetik yang berasal dari *UCI Machine Learning Repository*.
5. Menggunakan bahasa pemrograman *Python* dengan GUI *Hyper Text Markup Language* (HTML).

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan dari penelitian ini adalah untuk mengimplementasikan dan melihat tingkat akurasi dari algoritma *Backpropagation* yang telah di optimasi dengan *Particle Swarm Optimization* pada klasifikasi penyakit Retinopati Diabetik.

1.5. Manfaat Penelitian

Berdasarkan tujuan penelitian, manfaat dari penelitian adalah untuk mengetahui kinerja dari algoritma *Backpropagation* dalam melakukan klasifikasi penyakit Retinopati Diabetik dengan metode optimasi *Particle Swarm Optimization*.

1.6. Metodologi Penelitian

Tahapan-tahapan metode penelitian ini sebagai berikut :

1. Studi Literatur

Pada tahap ini penulis melakukan pencarian referensi terkait seluruh keperluan penelitian berupa e-book, jurnal, artikel ilmiah, makalah, skripsi maupun situs internet yang berkaitan dengan Jaringan Syaraf Tiruan (JST), algoritma *Backpropagation*, penyakit Retinopati Diabetik, *Particle Swarm Optimization*, dan Data Mining.

2. Analisis dan Perancangan

Penulis akan melakukan analisa dan perancangan yang diperlukan dalam penelitian,

yaitu diagram Ishikawa, *use case diagram*, *activity diagram*, *sequence diagram*, *flowchart*, dan *interface design*.

3. Implementasi Sistem

Penulis akan mengimplementasikan analisis dan rancangan sebelumnya dengan algoritma *Backpropagation*, *Particle Swarm Optimization*, dan Data Mining menggunakan bahasa pemrograman *Python*.

4. Pengujian Sistem

Pada tahap ini penulis akan melakukan uji coba hasil kinerja dari algoritma *Backpropagation* dan *Backpropagation* dengan *Particle Swarm Optimization*.

5. Dokumentasi Sistem

Penulis akan melakukan tahap dokumentasi dengan mengumpulkan proses awal hingga akhir untuk penyusunan laporan dengan aturan penulisan penelitian (skripsi).

1.7. Sistematika Penulisan

Sistematika dalam penulisan skripsi ini terdiri dari beberapa bagian utama sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini merupakan pemaparan dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Pada bab ini merupakan gambaran teori dari penulisan skripsi untuk memahami konsep penelitian yang akan dirancang.

BAB 3 ANALISIS DAN PERANCANGAN

Pada bab ini memaparkan analisis perancangan dari kerja sistem, seperti perancangan *flowchart*, diagram umum, dan perancangan *interface*.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Pada bab ini merupakan hasil dari pembangunan sistem skripsi yang diteliti dan hasil dari kinerja algoritma beserta penjelasannya.

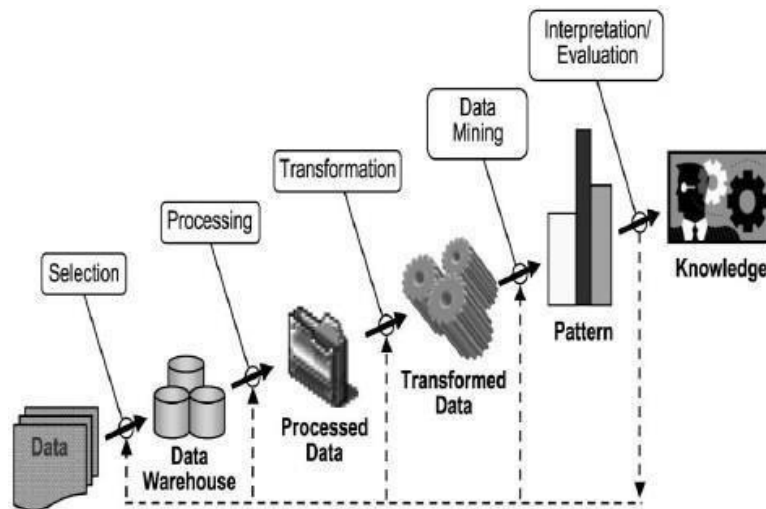
BAB 5 KESIMPULAN DAN SARAN

Pada bab ini merupakan hasil kesimpulan dan saran dari seluruh kinerja algoritma yang telah di aplikasikan yang dapat digunakan dan dikembangkan pada penelitian yang mendatang.

BAB 2

LANDASAN TEORI

2.1. Data Mining



Gambar 2.1. Proses *Data Mining*

Data mining merupakan sebuah proses ekstraksi informasi yang bertujuan untuk menemukan pola penting (*pattern recognition*) didalam suatu tumpukan data sehingga menjadi pengetahuan (*knowledge discovery*). Pada bidang sains, teknik, dan bisnis sudah menjadi hal biasa dalam memerlukan kumpulan data yang besar. Data mining dapat diterapkan dalam bidang kesehatan, analisis pasar, pendidikan, rekayasa manufaktur, *Costumer Relationship Management* (CRM), *fraud detection*, *instrusion detection*, deteksi kebohongan, segmentasi pelanggan, bank/keuangan, pengawasan perusahaan, analisis riset, investigasi kriminal, bioinformatika, dan lain-lain.

Data yang besar tentunya berisi informasi yang kompleks. Data mining adalah inovasi baru, yang dapat menampung jumlah data yang besar. Data mining dapat diibaratkan dengan koperasi usaha manusia dan komputer. Data Mining menggunakan teknik statistik, matematika, kecerdasan buatan, *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang berguna dan relevan dari berbagai data dalam jumlah besar.

2.1.1. Fungsi Data Mining

Beberapa fungsi data mining yaitu deskripsi, estimasi, prediksi, klastering, klasifikasi dan asosiasi. Berikut adalah penjelasannya (Liantoni, 2022) :

1. Deskripsi, menggambarkan pola pada data.
2. Estimasi, variabel bersifat numerik.
3. Prediksi, memperkirakan informasi yang bersifat prediktif.
4. Klustering, pembagian data kedalam kelompok yang memiliki jenis yang sama.
5. Klasifikasi, merupakan pengelompokkan fitur ke dalam kelas yang sesuai.
6. Asosiasi, menentukan pola berdasarkan hubungan yang saling berkaitan.

2.1.2. Tujuan Data Mining

Adapun tujuan data mining dibagi menjadi dua bagian, yaitu :

1. Prediktif

Bertujuan sebagai meramal nilai dari suatu atribut yang dilandaskan dari atribut lainnya. Atribut ramalan disebut dengan target atau variabel tidak bebas. Sementara itu, atribut lain disebut dengan *explanatory* atau variabel bebas.

2. Deskriptif

Bertujuan untuk membangun pola-pola seperti korelasi, trend, cluster, trayektori, dan anomali yang diringkas kedalam kelompok suatu data. Data mining deskriptif sering digunakan sebagai penyelidikan yang memerlukan teknik *postprocessing* untuk validasi dan hasil.

2.2. Klasifikasi

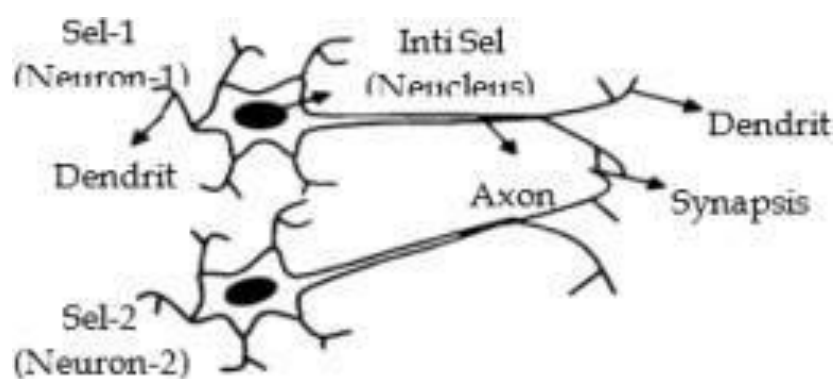
Klasifikasi pada data mining termasuk model supervised learning. Klasifikasi bertujuan untuk melihat kelas objek yang belum diketahui labelnya dengan cara menemukan model yang menggambarkan dan membedakan kelas data (Hizham, 2018).

Proses klasifikasi ini sudah menyediakan dan diketahui classnya. Cara melakukan klasifikasi dengan menelaah atribut lalu dikelompokkan kedalam class (Liantoni, 2022). Terdapat beberapa algoritma yang dapat digunakan dalam DataMining Klasifikasi, contohnya algoritma *K-Nearest Neighbor*, *Neural Network*, *Naive Bayes*, *Decision Tree*, dan *Logistic Regresion* (Khoiruzzaman et al., 2021).

2.3. Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (JST) atau yang sering dikenal sebagai *Artificial Neural Network (ANN)* pertama kali diciptakan pada tahun 1943 oleh *neurophysiologist* Warren McCulloch dan *logician* Walter Pitts. Jaringan Syaraf Tiruan (JST) banyak digunakan dalam memecahkan berbagai masalah dalam ilmu kedokteran, finansial, teknik, fisika, dan geologi. (Yani, 2005).

Jaringan Syaraf Tiruan (JST) merupakan algoritma yang terinspirasi dari sel syaraf biologi, yaitu algoritma komputasi yang cara kerjanya sama seperti neuron atau sistem jaringan saraf otak manusia (Norhikmah dan Rumini, 2019).



Gambar 2.2. Sel Syaraf Manusia

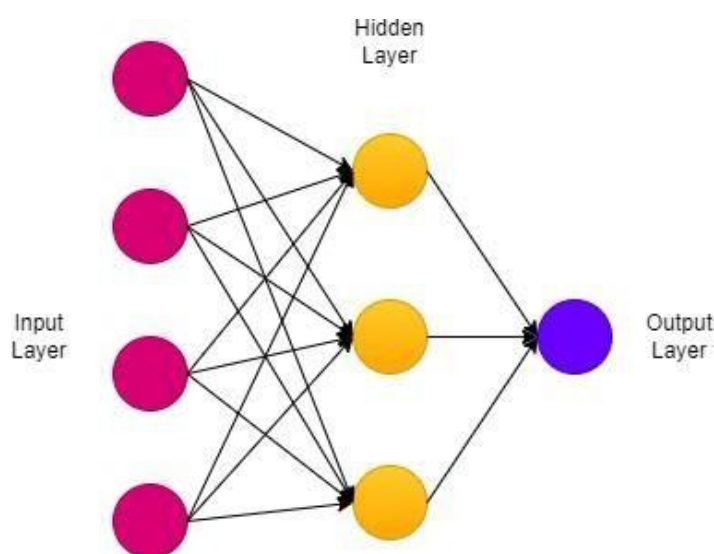
Gambar 2.2. merupakan sel syaraf manusia yang dapat digambarkan pada Jaringan Syaraf Tiruan (JST). Badan sel dari sel syaraf dapat digambarkan sebagai node atau unit, dendrit sebagai input, axon sebagai output, dan synapsis merupakan gambaran untuk nilai bobot (Khamdani, 2020).

2.3.1. Backpropagation – Neural Network

Algoritma *Backpropagation* adalah *supervised learning* yang terdiri dari banyak lapisan. Algoritma *Backpropagation* disebut *supervised learning* karena kemampuannya dalam mempelajari gambaran hasil output dari input, sehingga algoritma dapat memprediksi hasil outputnya. Terdapat tiga langkah dalam melakukan *Backpropagation training*, yaitu *feedforward*, *backpropagation*, dan penyesuaian bobot

(Nurdiansyah et al., 2018).

Algoritma *Backpropagation* memiliki 3 arsitektur yang merupakan kumpulan dari neuron-neuron yang tersusun dalam lapisan (*layers*), yaitu *input layer*, *hidden layer*, dan *output layer*. Skema Arsitektur *Backpropagation* dapat dilihat pada **Gambar 2.3.** :



Gambar 2.3. Arsitektur *Backpropagation*

Ketiga Arsitektur *Backpropagation* akan dijelaskan sebagai berikut :

1. *Input Layer* merupakan layer input data yang akan dimasukkan kedalam system dan akan diproses ke layer selanjutnya.
2. *Hidden Layer* merupakan layer antara input dan output yang berfungsi untuk mengolah dan memproses data dari input layer.
3. *Output Layer* merupakan hasil keluaran dari proses hidden layer yang masuk dari input layer.

Adapun langkah-langkah perhitungan algoritma *Backpropagation* sebagai berikut (Nurdiansyah et al., 2019):

Langkah 0 :

 Inisialisasi bobot dengan bilangan acak terkecil.

Langkah 1 :

Jika kondisi berhenti bernilai salah, maka lakukan langkah 2 – 9.

Langkah 2 :

Untuk setiap pasangan *training*, lakukanlah langkah 3 – 8.

Fase 1 : Feedforward

Langkah 3 :

Setiap unit *input* (X_i , $i = 1, \dots, n$) menerima sinyal *input* x_i dan diteruskan ke sinyal seluruh lapisan di atasnya (*hidden unit*).

Langkah 4 :

Setiap *hidden unit* (Z_j , $j = 1, \dots, p$) jumlahkanlah bobot sinyal *input* :

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

gunakan fungsi aktivasi berikut untuk menghitung sinyal *output*,

$$z_j = f(z_{in_j})$$

dan mengirimkan sinyal kepada semua unit pada lapisan di atasnya (*output unit*).

Keterangan :

z_{in_j} = Unit *hidden* j

v_{0j} = Bias pada unit *hidden* j

x_i = *Input* unit i

v_{ij} = unit *input* i pada *hidden layer* j

z_j = unit *hidden* dari j menggunakan fungsi aktivasi sigmoid

Langkah 5 :

Setiap unit *output* (Y_k , $k = 1, \dots, m$) jumlahkanlah bobot sinyal *output* :

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

dan aplikasikan fungsi aktivasi untuk menghitung sinyal *output*,

$$y_k = f(y_{in_k})$$

Keterangan :

y_{in_k} = nilai *output* k

w_{0k} = Bias pada *output* unit k

w_{jk} = unit *hidden* j ke satuan *output* k

y_k = nilai unit *output* k menggunakan fungsi aktivasi sigmoid

Fase 2 : Backpropagation

Langkah 6 :

Setiap unit *output* (Y_k , $k = 1, \dots, m$) menerima pola target sesuai pola *training input*. Hitung informasi *error* :

$$\delta_k = (t_k - y_k)f'(y_{in_k})$$

hitung koreksi bobot (digunakan untuk memperbaharui w_{jk}),

$$\Delta w_{jk} = \alpha \delta_k z_j$$

hitung koreksi bias (digunakan untuk memperbaharui w_{0k})

$$\Delta w_{0k} = \alpha \delta_k$$

dan mengirimkan δ_k ke unit lapisan dibawahnya.

Keterangan :

δ_k = nilai *error output*

t_k = nilai target *output*

α = *learning rate*

Δw_{jk} = perubahan bobot unit *hidden* j menjadi unit *output* k

Langkah : 7

Setiap *hidden unit* (Z_j , $j = 1, \dots, p$) menjumlahkan delta *input* (dari unit pada lapisan atasnya),

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

dikalikan dengan turunan fungsi aktivasi untuk menghitung informasi *error*,

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

hitung koreksi bobot (digunakan untuk memperbaharui v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i$$

hitung koreksi bias (digunakan untuk memperbaharui v_{0j})

$$\Delta v_{0j} = \alpha \delta_j$$

Keterangan :

δ_j = nilai *error* unit *hidden*

Δv_{ij} = perubahan bobot unit *input* i ke unit *hidden* j

Fase 3 : Penyesuaian Bobot dan Bias

Langkah : 8

Setiap unit output (Y_k , $k = 1, \dots, m$) memperbaharui bias dan bobot ($j = 0, \dots, p$) :

$$w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk}$$

Setiap unit *hidden* (Z_j , $j = 1, \dots, p$) memperbaharui bobot dan bias ($i = 0, \dots, n$) :

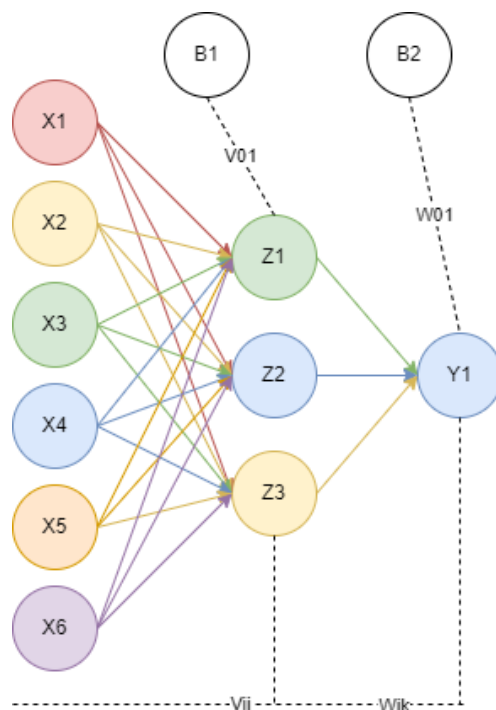
$$v_{ij}(\text{new}) = v_{ij}(\text{old}) + \Delta v_{ij}$$

Langkah : 9

Selesai.

2.3.1.1. Perhitungan *Backpropagation* - Neural Network

Pada **Gambar 2.4.** berikut memiliki beberapa neuron, yaitu 6 input neuron, 3 *hidden layer* neuron, dan 1 output neuron yang akan dilakukan perhitungan secara acak.



Gambar 2.4 Contoh Arsitektur *Backpropagation*

a. Pelatihan *Backpropagation* - Neural Network

1. Sebagai contoh dataset acak yang dibuat penulis x_1 sampai dengan x_6 sebagai nilai inputan atau *input layer*, dan *class* merupakan keluaran atau *output layer* dari dataset acak tersebut.

Tabel 2.1 Contoh Data

Data	X_1	X_2	X_3	X_4	X_5	X_6	Class
1	13	3	16	0	5	1	0
2	15	6	8	1	5	2	1
3	5	9	1	1	3	3	1

2. Memberikan nilai bobot secara acak (0-1)
 - Inisialisasi jaringan *input layer* dan *hidden layer* (V_{ij})

Tabel 2.2 Bobot Awal V_{ij} Contoh Data

V_{ij}	$V_{(bias)}$	V_{21}	V_{31}	V_{41}	V_{51}	V_{61}	V_{71}
1	0.31	0.12	0.111	0.14	0.23	0.168	0.136

- Inisiasi jaringan *hidden layer* dengan *output layer* (W_{jk})

Tabel 2.3 Bobot Awal W_{ij} Contoh Data

W_{jk}	$W_{(bias)}$	W_{11}
1	0.541	0.538

3. *Learning rate* = 0.5
Minimum error = 0.01
Maximum epoch = 1

4. Perhitungan *Forward*

- Menghitung setiap node *hidden layer* (z_{inj} , $j=1$)

$$\begin{aligned}
 z_{in1} &= 1 \times (0.31) + 13 \times (0.12) + 3 \times (0.111) + 16 \times (0.14) + 0 \times \\
 &\quad (0.23) + 5 \times (0.168) + 2 \times (0.136) \\
 &= 0.31 + 1.56 + 0.333 + 2.24 + 0 + 0.84 + 0.272 \\
 &= 5.555
 \end{aligned}$$

- Menghitung nilai *hidden layer* dengan fungsi aktivasi sigmoid

$$z_1 = f(z_{in1}) = \frac{1}{1 + e^{-(5.555)}} = 0.996$$

- Menghitung $y_{ink}(y_k, k=1)$ pada lapisan luaran

$$y_{in1} = 1 \times (0.54) + 0.996 \times (0.538) = 1.076$$

- Menghitung nilai $y_k(y_k, k=1)$ dengan fungsi aktivasi sigmoid

$$y_1 = f(y_{in1}) = \frac{1}{1 + e^{-(1.076)}} = 0.745$$

5. Perhitungan *Backpropagation*

- Menghitung δ error pada *output layer* $y_k(k=1)$

$$\delta_1 = (t_k - y_k) f'(y_{ink}) = (t_1 - y_1) y_1 (1 - y_1)$$

$$= (1-0.745) \times 0.745 (1-0.745) = 0.0484$$

- Menghitung koreksi bobot W_{jk} dan W_{0k} (bias)

$$\begin{aligned}\Delta W_{01} &= \alpha \delta z_j \\ &= 0.5 \times 0.0484 \times 1 = 0.0242\end{aligned}$$

$$\begin{aligned}\Delta W_{11} &= \alpha \delta_k z_j \\ &= 0.5 \times 0.0484 \times 0.0996 = 0.0241\end{aligned}$$

- Menghitung unit atas dari *hidden layer* z_j ($j=1$) yaitu *input layer*

$$\begin{aligned}\delta_{in1} &= \sum_{k=1}^m \delta_k W_{jk} \\ &= 0.0484 \times 0.538 = 0.026\end{aligned}$$

- Menghitung informasi *error*

$$\begin{aligned}\delta_1 &= \delta_{in1} f'(z_{in1}) \\ &= 0.026 \times 0.996 \times (1 - 0.996) \\ &= 0.026 \times 0.996 \times 0.004 \\ &= 0.0001\end{aligned}$$

- Menghitung pembaruan V_{ij}

$$\begin{aligned}\Delta v_{01} &= 0.5 \times 0.0001 \times 1 = 0.0005 \\ \Delta v_{11} &= 0.5 \times 0.0001 \times 13 = 0.00065 \\ \Delta v_{21} &= 0.5 \times 0.0001 \times 3 = 0.00015 \\ \Delta v_{31} &= 0.5 \times 0.0001 \times 16 = 0.0008 \\ \Delta v_{41} &= 0.5 \times 0.0001 \times 0 = 0 \\ \Delta v_{51} &= 0.5 \times 0.0001 \times 5 = 0.00025 \\ \Delta v_{61} &= 0.5 \times 0.0001 \times 1 = 0.0005\end{aligned}$$

6. Perhitungan pembaruan bobot

- Unit *hidden layer* y_k ($k=1$) memperbarui bias dan bobot ($j=0$)

$$W_{jk}(\text{new}) = W_{jk}(\text{old}) + \Delta W_{jk}$$

$$W_{01} = W_{01}(\text{old}) + \Delta W_{01} = 0.541 + 0.0242 = 0.5652$$

$$W_{11} = W_{11}(\text{old}) + \Delta W_{11} = 0.538 + 0.0241 = 0.5651$$

- Unit *hidden layer* z_k ($j=1$) memperbarui bias dan bobot ($i=0$)

$$V_{ij}(\text{new}) = V_{ij}(\text{old}) + \Delta V_{ij}$$

$$\begin{aligned}
 V01 &= 0.31 + 0.0005 = 0.3105 \\
 V11 &= 0.12 + 0.00065 = 0.12065 \\
 V21 &= 0.111 + 0.00015 = 0.11115 \\
 V31 &= 0.14 + 0.0008 = 0.1408 \\
 V41 &= 0.23 + 0 = 0.23 \\
 V51 &= 0.168 + 0.00025 = 0.16825 \\
 V61 &= 0.136 + 0.0005 = 0.1365
 \end{aligned}$$

b. Pengujian *Backpropagation*

1. Input data uji secara acak

Tabel 2.4 Contoh Data Uji

Data	X1	X2	X3	X4	X5	Class
1	3	6	9	16	8	?

2. Melakukan inisialisasi bobot awal dengan menggunakan bobot baru

Tabel 2.5 Contoh Data Bobot Baru Vij

V01	V11	V21	V31	V41	V51
0.168	0.169	0.153	0.155	0.123	0.111

Tabel 2.6 Contoh Data Bobot Baru Wij

W01	W11
0.369	0.681

3. Menghitung nilai Z_{inj} dari *hidden layer*

$$\begin{aligned}
 Z_{inj} &= 1 \times (0.168) + 3 \times (0.169) + 6 \times (0.153) + 9 \times (0.155) \\
 &\quad + 16 \times (0.123) + 8 \times (0.111) \\
 &= 0.168 + 0.507 + 0.918 + 1.395 + 1.968 + 0.888 = 5.844
 \end{aligned}$$

4. Menghitung nilai *output* dari *hidden layer* dengan fungsi aktivasi sigmoid

$$Z1 = f(z_{in1}) = \frac{1}{1 + e^{-(5.884)}}$$

5. Menghitung nilai Y_{ink} (y_k , $k=1$) pada *output layer*

$$Y_{inj} = 1 \times (0.369) + 0.997 \times (0.681) \\ = 0.369 + 0.679 = 1.048$$

6. Menghitung nilai *output* y_k (y_k , $k=1$) dengan fungsi aktivasi sigmoid

$$Y_1 = f(y_{in1}) = \frac{1}{1 + e^{-(1.048)}} = 0.740$$

Maka hasil output adalah 0.740 yang dimana 0.740 lebih besar ($>$) dari 0.5, maka *class* yang memenuhi adalah **1**

Tabel 2.7 Hasil Contoh Data Uji

Data	X1	X2	X3	X4	X5	Class
1	3	6	9	16	8	1

2.4. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan algoritma optimasi yang berguna untuk pengambilan keputusan atau mengoptimalkan perencanaan jalur. Eberhart dan Kennedy pada tahun 1995 merupakan nememu *PSO* yang terinspirasi dari perilaku sekumpulan burung (Purnama et al., 2021).

Algoritma *Particle Swarm Optimization (PSO)* memiliki keunggulan di antara algoritma lain dalam melakukan optimasi karena dapat bekerja lebih laju juga memiliki tingkat konvergensi yang konstan meski dengan data masalah banyak sekalipun sehingga algoritma ini dapat dikatakan algoritma yang efisien (Muzakkir et al., 2014).

Adapun algoritma *Particle Swarm Optimization (PSO)* sebagai berikut (Surantha et al., 2021) :

1. Menginisialisasi kecepatan dan populasi partikel secara *random*
2. Mulai dengan iterasi baru
3. Menentukan *p-best* tiap partikel dan mengganti *p-best* yang lebih baik dari sebelumnya
4. Menentukan nilai *g-best*

5. Update tiap kecepatan partikel menggunakan persamaan dari :

$$V_{id}^{t+1} = WV_{id}^t + c_1 \cdot r_1 \cdot pbest_{id}^t - x_{id}^t + c_2 \cdot r_2 \cdot (p_{gd}^t - x_{id}^t)$$

Keterangan :

V_{id}^{t+1}	= Kecepatan partikel i dari iterasi t
W	= Bobot inersia
c_1	= Partikel yang mengikuti <i>p-best</i>
c_2	= Partikel yang mengikuti <i>g-best</i>
r_1 dan r_2	= Bilangan acak (dari 0 sampai 1)
$pbest_{id}^t$	= Posisi terbaik dari partikel i
p_{gd}^t	= Partikel terbaik pada kelompok/kawanan
x_{id}^t	= Posisi partikel i

6. *Update* posisi pada tiap partikel :

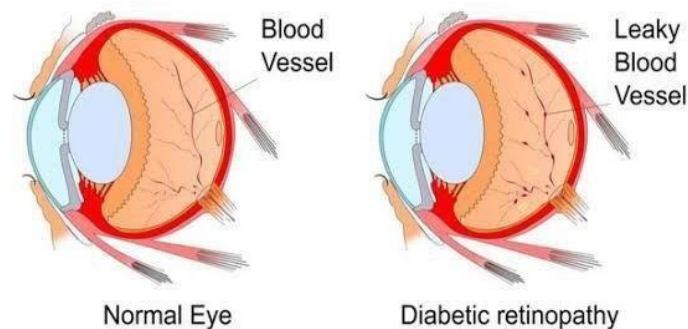
$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$$

7. Iterasi berhenti jika terminasi terpenuhi. Lakukan Langkah kembali ketahap 3 apabila terminasi tidak terpenuhi.

2.5. Retinopati Diabetik

Penyakit Retinopati Diabetes merupakan penyakit yang diakibatkan dari diabetes melitus. Retinopati Diabetes termasuk penyakit yang menyebabkan kematian. Penyebab terjadinya Retinopati Diabetes ini adalah tingginya kadar glukosa dalam darah atau memecahnya pembuluh darah di retina. Umumnya penyakit ini akan mengganggu penglihatan mata dan dapat mengakibatkan kebutaan jika tidak cepat ditangani (Subarkah et al., 2021).

Diabetic Retinopathy



Gambar 2.5 Retinopati Diabetik

Dapat dilihat pada **Gambar 2.5** bahwa seseorang yang mengidap penyakit Retinopati Diabetik akan mengalami kerusakan pada pembuluh darah.

Berdasarkan data yang dirangkum pada penelitian di beberapa daerah, Indonesia mendapati 42,6% yang mengidap diabetik retinopati dan 10% diantaranya mengalami kebutaan dan akan meningkat hingga tahun 2030 (Subarkah, 2020).

2.5.1. Klasifikasi Retinopati Diabetik

Menurut Persatuan Dokter Spesialis Mata Indonesia 2018, Diabetik Retinopati dapat diklasifikasikan menjadi dua yang merupakan tanda awal penyakit, yaitu :

1. *Non-Proliferative Diabetic Retinopathy* (NPDR)

a. Ringan

Minimal terdeteksi 1 mikroaneurisma.

b. Sedang

Terdapat mikroaneurisma, pendarahan, eksudat keras, *cotton wool spots* (CWS), *beading vena*, penyempitan lumen arteri, serta adanya *Intraretinal Microvascular Abnormalities* (IRMA).

c. Berat

Dapat disebut dengan proliferasif, ditemukan kelainan pada NPDR dan ditemukan pendarahan di empat kuadran retina, venous beading di dua kuadran, dan IRMA

di satu kuadran retina.

d. Sangat Berat

Memiliki kemungkinan mencapai 45% berkembang menjadi diabetik retinopati *proliferated* dalam jangka waktu 1 tahun.

2. *Proliferative Diabetic Retinopathy* (PDR)

a. Awal

Tumbuhnya pembuluh darah baru pada diskus dalam jarak 1DD atau neovaskularisasi di daerah lain.

b. Resiko Tinggi

NVD lebih besar atau lebih kecil dari $\frac{1}{4}$ dengan pendarahan *vitreous*.

c. *Advanced*

PDR resiko tinggi ditambah adanya ablasi retina traksional yang melibatkan makula.

2.6. Confusion Matrix

Confusion Matrix adalah metode pengevaluasian yang bekerja untuk menghitung kinerja selama proses klasifikasi. *Confusion Matrix* mempunyai 4 tabel yang tiap kotaknya berbeda-beda dari nilai prediksi dan nilai aktual yang terdiri dari *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.6 Confusion Matrix

Berikut adalah penjelasan **Gambar 2.6** dari empat tabel *Confusion Matrix* (Karimi, 2021) :

1. *True Positive* (TP)
Nilai aktual positif dan model prediksi nilai positif.
2. *True Negative* (TN)
Kesalahan Tipe 1, atau juga dikenal hasil prediksi positif, dan salah.
3. *False Positive* (FP)
Kesalahan Tipe 2, atau juga dikenal hasil prediksi Anda negatif, dan hasilnya juga salah.
4. *False Negative* (FN)
Nilai sebenarnya negatif dan model prediksi nilai negatif.

Confusion Matrix dapat menghitung *Performance Metrics* seperti *accuracy*, *precision*, *recall*, dan *f-measure* yang akan dijelaskan (Karimi, 2021) :

1. *Accuracy*

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. *Precision*

$$precision = \frac{TP}{TP + FP}$$

3. *Recall*

$$recall = \frac{TP}{TP + FN}$$

4. *F-Measure*

$$fmeasure = 2 \times \frac{precision \cdot recall}{precision + recall}$$

2.9. Penelitian Relevan

Berikut merupakan beberapa penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan oleh penulis adalah sebagai berikut :

1. Pada penelitian terdahulu yang dilakukan oleh Purnama et al (2021) dengan judul penelitian “Deteksi Peluang Gagal Bayar Calon Debitur Menggunakan Algoritma *Particle Swarm Optimization* (PSO) untuk Meningkatkan Kinerja Manajemen Risiko pada Koperasi Simpan Pinjam ABC” menyatakan bahwa algoritma Regresi Logistik menggunakan *Particle Swarm Optimization* mampu menghasilkan akurasi sebesar 99.1%.
2. Penelitian yang dilakukan oleh Tandrio et al (2018) dengan judul “*Classification of Wine Types Based on Composition Using Backpropagation And Particle Swarm Optimization*” menyatakan bahwa *Neural Network* berbasis PSO menghasilkan hasil lebih baik daripada tidak menggunakan PSO. NNPSO dapat mengklasifikasikan semua jenis *wine* sedangkan NNBP hanya dapat mengklasifikasikan satu jenis *wine* saja.
3. Penelitian terdahulu yang dilakukan oleh Humam et al (2019) yang berjudul “*The Application Of Particle Swarm Optimization Using Neural Network To Optimize Classification Of Employee Performance Assessment*” menyatakan bahwa eksperimen tersebut *Neural Network* dengan optimasi Algoritma *Particle Swarm Optimization* (PSO) menghasilkan akurasi yang lebih baik daripada jaringan struktural tanpa optimasi.
4. Penelitian yang dilakukan oleh Khoiruzzaman et al., (2021) dengan judul penelitian “Klasifikasi Peminjaman Buku Menggunakan *Neural Network Backpropagation*” menyatakan bahwa hasil pengujian menggunakan *Neural Network Backpropagation* mampu melakukan tingkat akurasi 95.5%.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

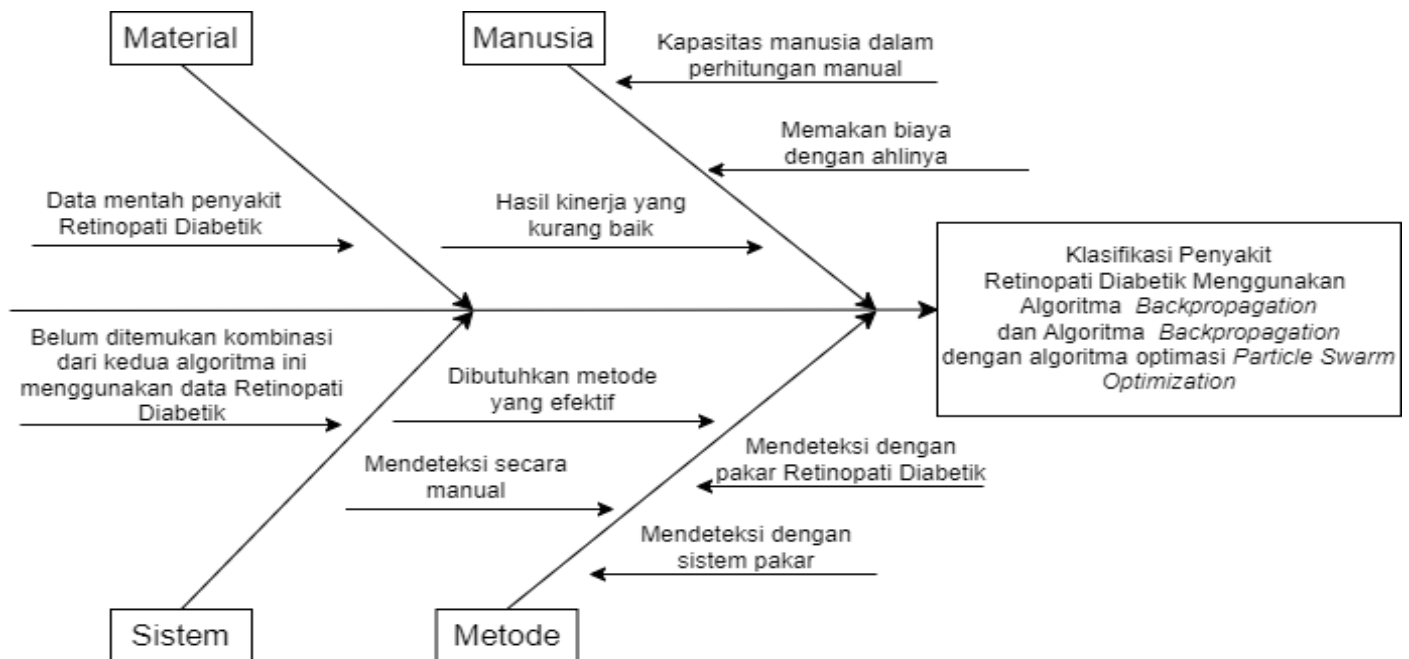
3.1. Analisis Sistem

Analisis sistem merupakan pendeskripsian awal dari pengembangan sistem yang akan dibangun. Analisis sistem bertujuan untuk memaparkan secara rinci komponen-komponen sistem yang dibutuhkan dengan pemaparan yang lebih sederhana agar mudah dipahami dalam melakukan pengembangan sistem. Pada umumnya, analisis sistem memiliki tiga tahapan utama, yaitu analisis masalah, analisis kebutuhan, dan analisis proses.

3.1.1. Analisis Masalah

Analisis masalah merupakan langkah utama yang harus dilakukan untuk membangun perancangan sistem. Pada langkah ini akan dilakukan identifikasi sebab akibat sistem dibangun sehingga dapat menghasilkan hasil yang tepat sasaran dan bekerja dengan baik. Permasalahan pada penelitian ini adalah menggunakan klasifikasi pada data penyakit Diabetik Retinopati untuk membandingkan hasil kerja algoritma *Backpropagation* jika algoritma tersebut *stand alone* dengan kerja algoritma *Backpropagation* yang dioptimasi algoritma *Particle Swarm Optimization*. Digunakan *Confusion Matrix* untuk melihat hasil akurasi terbaik dari kedua algoritma yang digunakan.

Pada penelitian ini, untuk melakukan identifikasi masalah pada sistem tersebut digunakan Diagram Ishikawa (*Fishbone Diagram*). Diagram Ishikawa berbentuk seperti tulang-tulang anggota tubuh ikan yang merupakan pendeskripsian diagram berisi sebab dan akibat dari permasalahan penelitian. Diagram Ishikawa terdiri dari *material*, *method*, *machine*, dan *man* yang akan dijelaskan pada gambar **Gambar 3.1**.



Gambar 3.1. Diagram Ishikawa Analisis Masalah

3.1.2. Analisis Persyaratan

Analisis persyaratan dilakukan sebagai pengidentifikasian serta penjelasan dari seluruh fungsi yang ada pada sistem. Analisis kebutuhan terdiri dari dua bagian, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

1. Persyaratan Fungsional

Kebutuhan fungsional adalah pendeskripsian seluruh bagian-bagian komponen yang harus ada pada sistem. Persyaratan fungsional yang dirumuskan pada sistem ini yakni:

- a. Sistem dapat menampilkan *dataset* dari penyakit Diabetik Retinopati.
- b. Sistem dapat menampilkan hasil kinerja algoritma *Backpropagation* dan algoritma optimasi *Particle Swarm Optimization* menggunakan *Confusion Matrix* dari data penyakit Diabetik Retinopati.

2. Persyaratan Non-Fungsional

Persyaratan non-fungsional diperlukan untuk merumuskan ukuran sebuah sistem yang akan digunakan sebagai penentu kinerja sistem sudah sesuai atau tidak. Persyaratan

non-fungsional pada sistem ini, antara lain:

a. Efisiensi

Sistem dirancang sesederhana mungkin agar mudah dipahami dan digunakan.

b. Desain

Sistem dibangun harus memiliki tampilan yang semenarik rupa untuk digunakan bagi *user*.

c. Kinerja

Sistem yang dibangun harus memiliki kinerja yang baik meskipun menggunakan spesifikasi komputer yang berbeda dari peneliti.

3.1.3. Analisis Proses

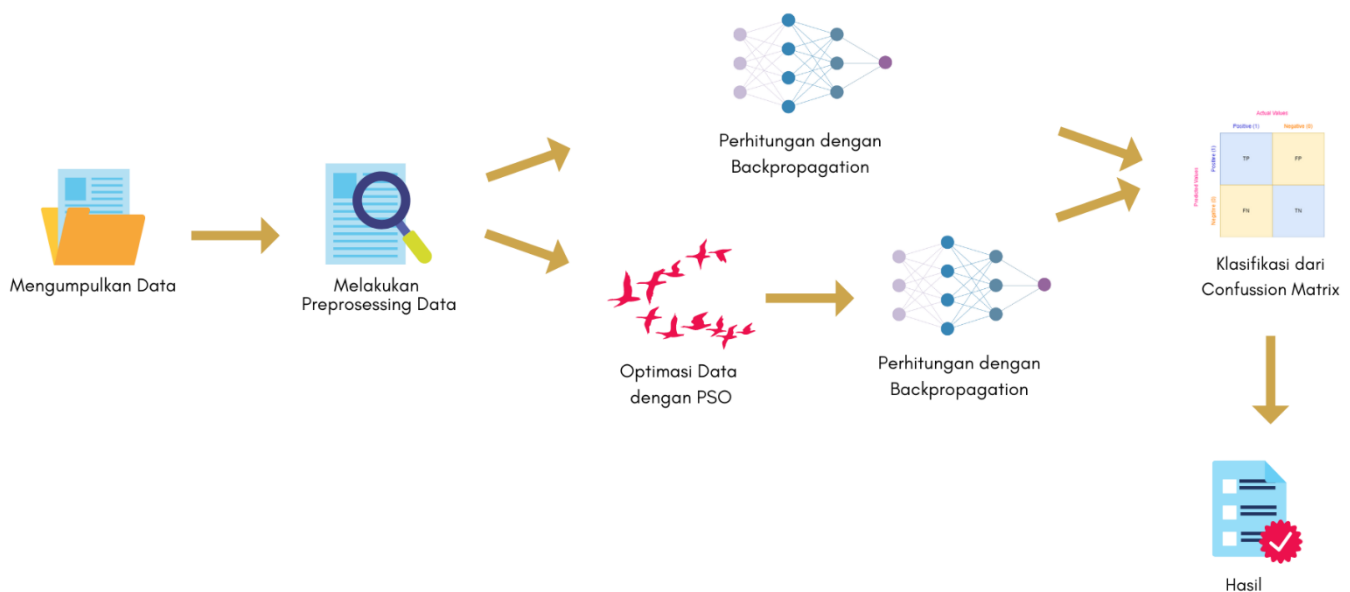
Pada penelitian yang akan dirancang ini, terdapat beberapa tahapan untuk membandingkan hasil kinerja dari algoritma *Backpropagation* dan algoritma *Backpropagation* yang dioptimasi dengan algoritma *Particle Swarm Optimization*. Dimulai dari mengumpulkan dataset yang telah dipilih oleh peneliti, yaitu Diabetik Retinopati yang berasal dari *UCI Machine Learning*. Data tersebut akan memasuki tahap *preprocessing* terlebih dahulu agar menghindari kerusakan data seperti *missing value*. Setelah selesai melakukan tahap *preprocessing*, dilakukan tahapan *feature selection* pada data menggunakan algoritma optimasi *Particle Swarm Optimization*. Kemudian data tersebut akan dipisah menjadi data *training* (data latih) dan data *testing* (data pengujian), yang dimana data *training* (data latih) akan dilatih menggunakan algoritma *Backpropagation* dan *Particle Swarm Optimization* dan melakukan data *testing* (data pengujian) yang hasilnya akan ditampilkan oleh *Confusion Matrix*.

3.2. Pemodelan Sistem

Pemodelan sistem (*modeling*) bertujuan untuk menyederhanakan elemen-elemen yang berperan penting dalam sistem yang akan dirancang agar mudah dipahami. Pada pemodelan sistem ini menggunakan Diagram Umum, *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

3.2.1. Diagram Umum

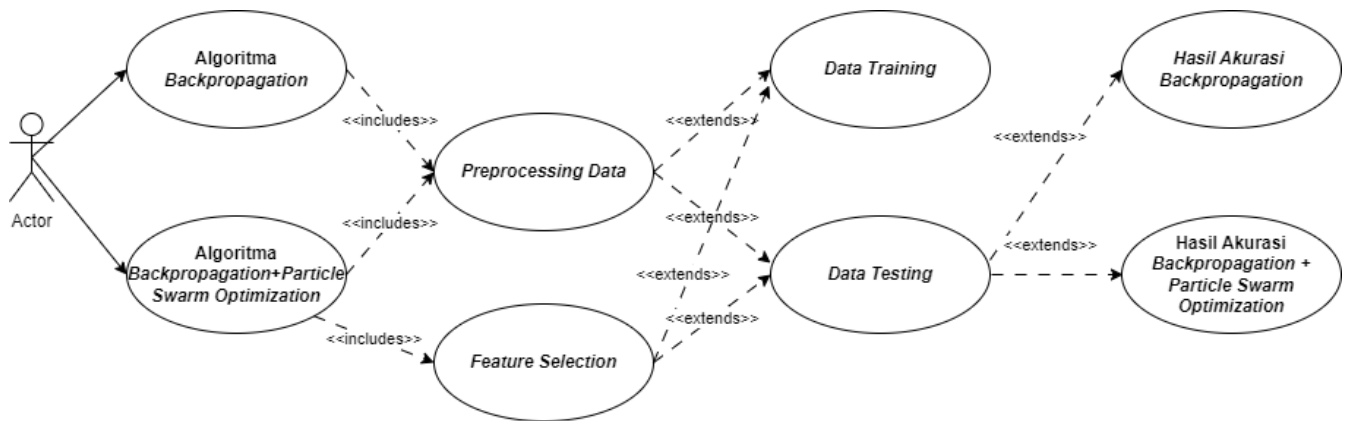
Diagram umum adalah suatu gambaran yang berguna untuk menjelaskan suatu kerangka atau gambaran sistem yang akan dirancang. Diagram umum sistem dapat dilihat pada **Gambar 3.2**.



Gambar 3.2 Diagram Umum Sistem

Gambar 3.2 dapat dijelaskan proses gambaran kerja sistem yang dimulai dengan mengumpulkan data dari penyakit Retinopati Diabetik dan melakukan tahap *preprocessing* data agar menjadi data yang lebih efisien. Setelah selesai melakukan *preprocessing* data, selanjutnya dataset tersebut akan dioptimasi menggunakan algoritma *Particle Swarm Optimization* dan hasil optimasi tersebut akan memasuki tahap perhitungan dari algoritma *Backpropagation*. Setelah selesai melakukan perhitungan *Backpropagation*, *Confusion Matrix* akan menghitung hasil kinerja dari klasifikasi dataset dan mengeluarkan hasil yang berupa nilai *Accuracy*, *Precision*, *Recall*, dan *F-Measure*.

3.2.2. Use Case Diagram



Untuk memudahkan proses peng analisis komponen digunakanlah *Use Case Diagram* yang bertujuan untuk menggambarkan interaksi antara *user* dengan sistem. Berikut *Use Case Diagram* pada sistem yang dapat dilihat pada **Gambar 3.3**.

Gambar 3.3. *Use case diagram*

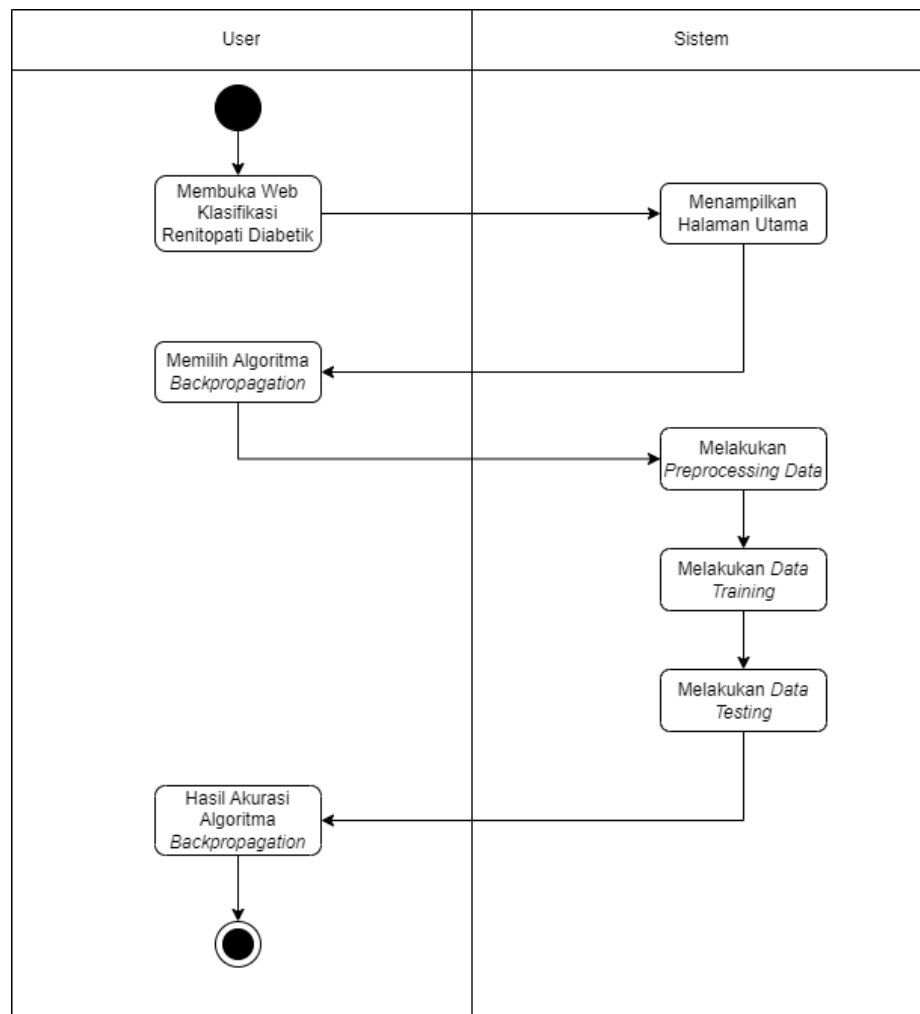
Pada **Gambar 3.3** *Use Case Diagram* ini dapat disimpulkan terdapat seorang *actor* yang berperan sebagai *user*. *User* dapat memilih menginginkan algoritma yang digunakan seperti algoritma *Backpropagation* saja atau algoritma *Backpropagation* dengan algoritma optimasi *Particle Swarm Optimization*. Kemudian algoritma tersebut memasuki tahap *preprocessing data*. Setelah melakukan *preprocessing data*, dilakukan *feature selection* apabila menggunakan algoritma optimasi *Particle Swarm Optimization* dan dilanjutkan dengan tahap data *training* dan data *testing*, lalu menampilkan hasil kinerja pengklasifikasian Retinopati Diabetik.

3.2.3. Activity Diagram

Activity Diagram adalah urutan penggambaran alur aktivitas dari proses kerja *user* dengan sistem dari awal digunakan hingga akhir digunakan. *Activity Diagram* pada sistem yang akan dirancang adalah *Activity Diagram* pada *Backpropagation* dan *Activity Diagram* pada *Backpropagation* dengan *Particle Swarm Optimization*.

3.2.3.1. Activity Diagram pada Backpropagation

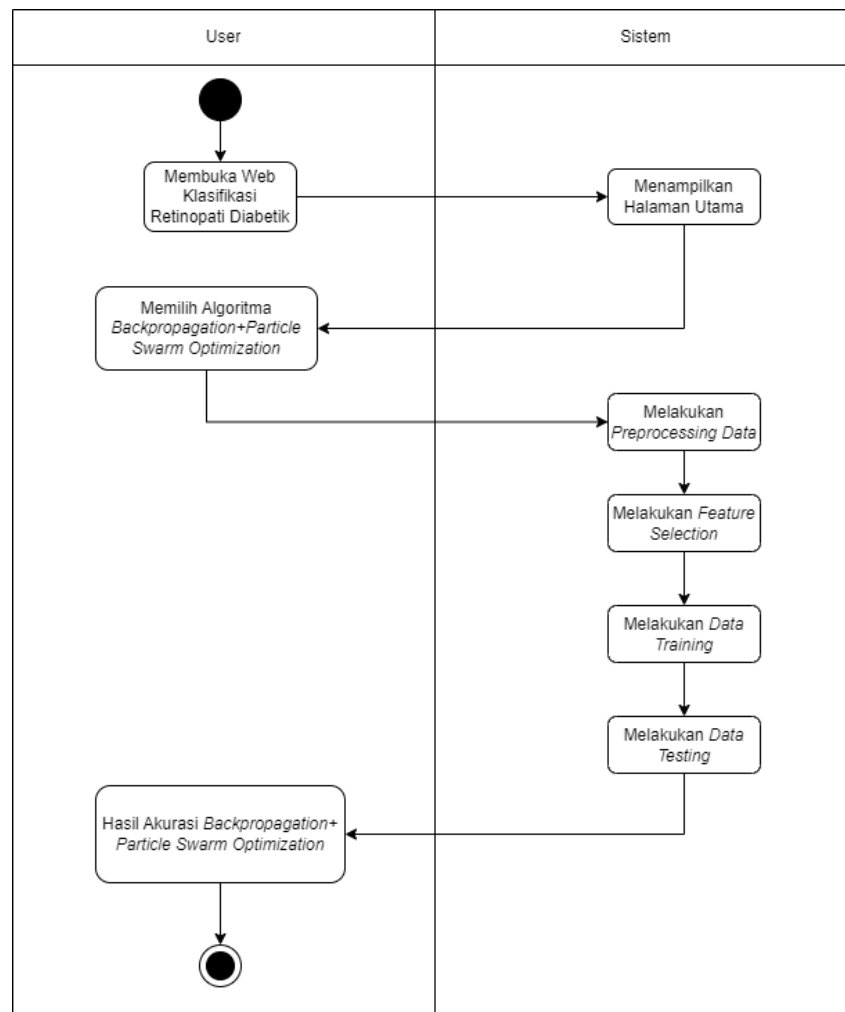
Activity Diagram menggunakan algoritma *Backpropagation* ditunjukkan pada **Gambar 3.4.**



Gambar 3.4. Activity diagram pada *Backpropagation*

3.2.3.2. Activity Diagram pada Backpropagation dengan Particle Swarm Optimization

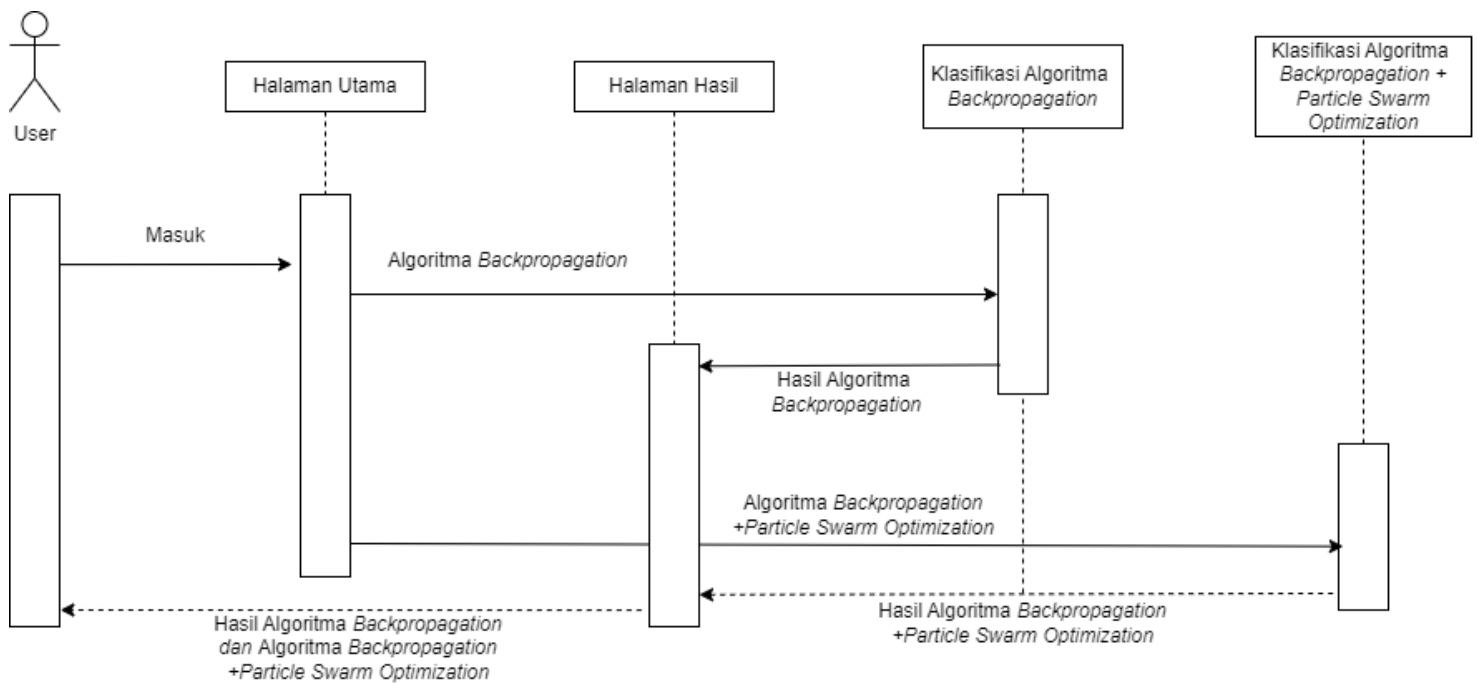
Activity Diagram menggunakan algoritma *Backpropagation* dengan *Particle Swarm Optimization* ditunjukkan pada **Gambar 3.5.**



Gambar 3.5. Activity diagram pada Backpropagation dengan Particle Swarm Optimization

3.2.4 Sequence Diagram

Sequence Diagram merupakan diagram yang mengacu pada waktu. Diagram ini akan menjelaskan alur sistem yang akan dirancang secara berurut berdasarkan waktu yang rinci. Pada **Gambar 3.6** merupakan sequence diagram dari sistem.



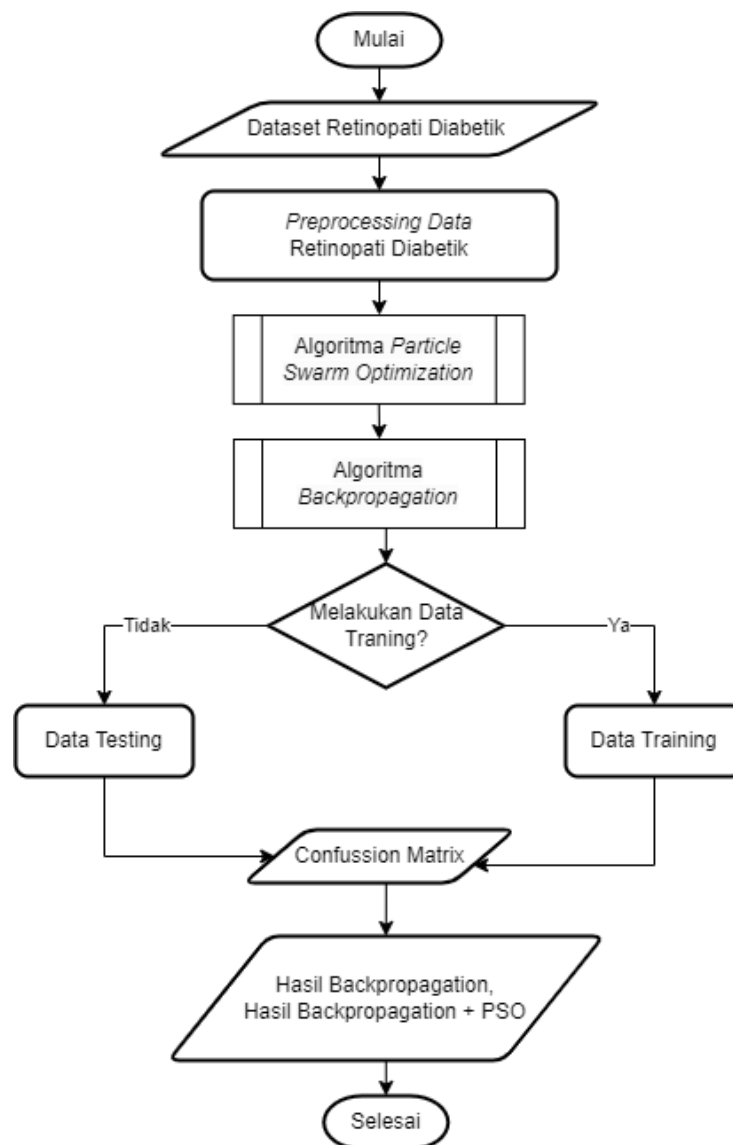
Gambar 3.6. *Sequence Diagram.*

3.2.5 Flowchart (Diagram Alir)

Flowchart atau diagram alir adalah sebuah bagan yang berfungsi untuk mempelajari langkah-langkah maupun tujuan dalam melakukan suatu proses dalam program.

3.2.5.1 Flowchart Sistem

Berikut **Gambar 3.6** merupakan diagram alir dari perancangan sistem:

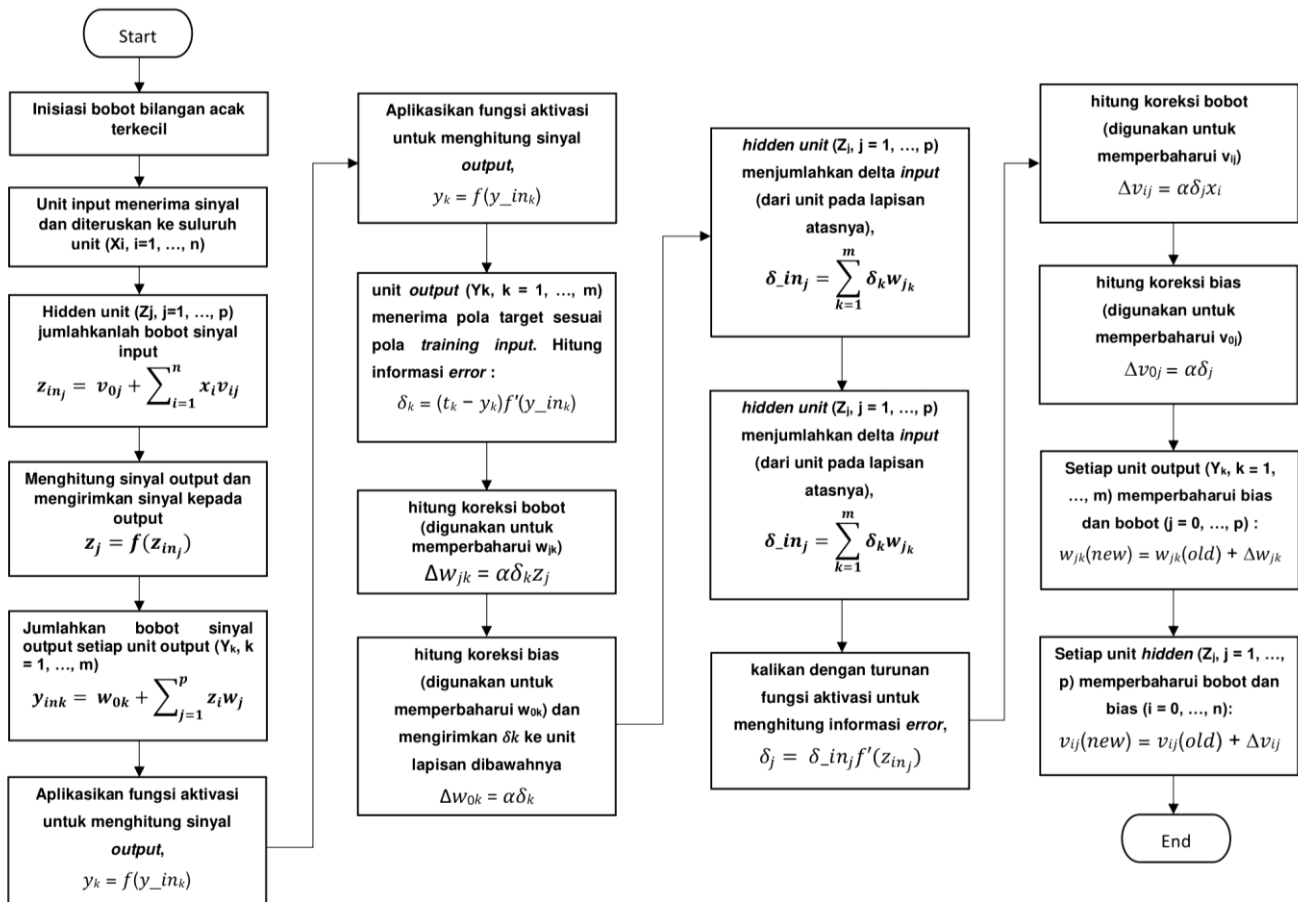


Gambar 3.7. *Flowchart Sistem.*

Pada *flowchart* sistem **Gambar 3.7** dimulai dengan menginput dataset retinopati diabetik yang telah disediakan, setelah itu sistem akan bekerja untuk melakukan preprocessing data retinopati diabetik seperti data validation dan data discrization, dilanjutkan dengan proses perhitungan menggunakan algoritma *Particle Swarm Optimization* dan algoritma *Backpropagation*. Setelah itu, sistem akan memasuki decision apakah data retinipati diabetik melakukan data training atau tidak, data yang terpilih akan melakukan proses perhitungan dan menampilkan hasil dari *Confussion*

Matrix yang berisi Accuracy, Precision, Recall, dan F-Measure.

3.2.5.2 Flowchart Algoritma Backpropagation

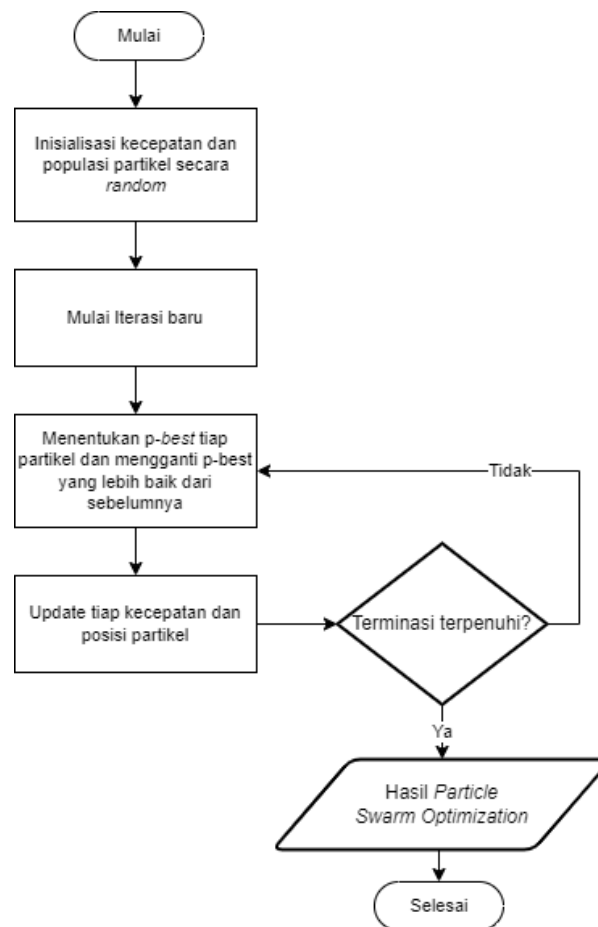


Gambar 3.8. Flowchart Backpropagation

Flowchart dari algoritma Backpropagation dapat dilihat pada **Gambar 3.8** sebagai berikut:

3.2.5.3 Flowchart Algoritma Particle Swarm Optimization

Flowchart dari algoritma Particle Swarm Optimization dapat dilihat pada **Gambar 3.9** sebagai berikut:



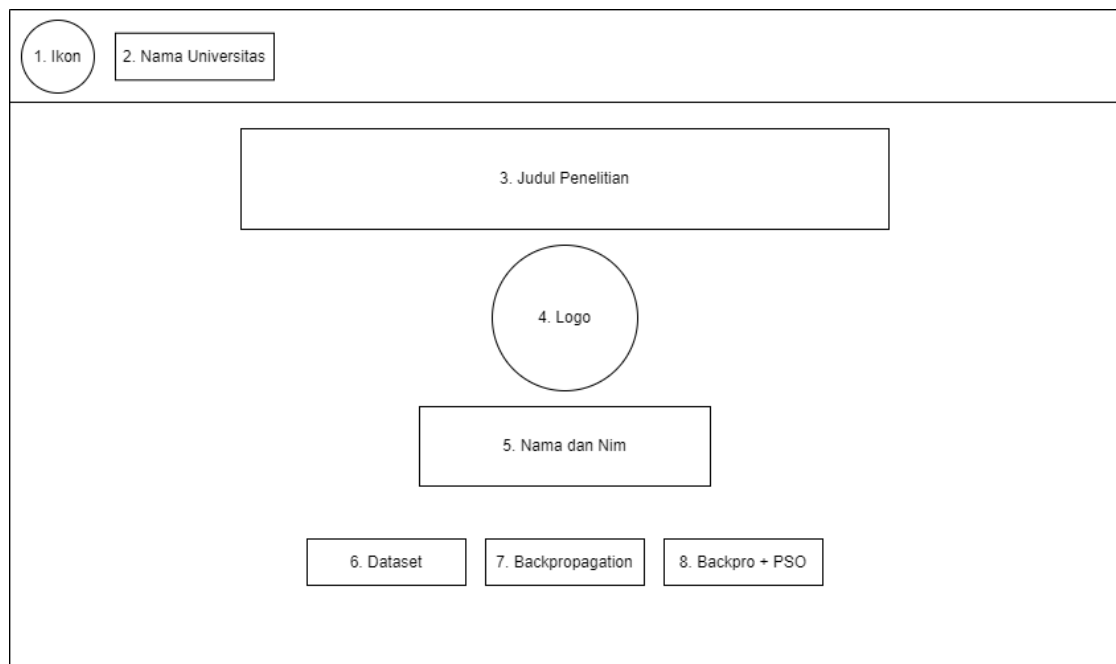
Gambar 3.9. *Flowchart Particle Swarm Optimization.*

3.2.6 Perancangan Interface

Perancangan *Interface* atau Perancangan antar muka adalah gambaran tampilan sistem yang akan dirancang pada sistem yang akan dibuat dengan *user friendly* agar memudahkan pengguna dalam menjalankan sistem.

a. Halaman Utama

Halaman utama ini merupakan halaman yang pertama kali ditampilkan sistem. Pada **Gambar 3.8.** merupakan rancangan dari halaman utama penelitian.



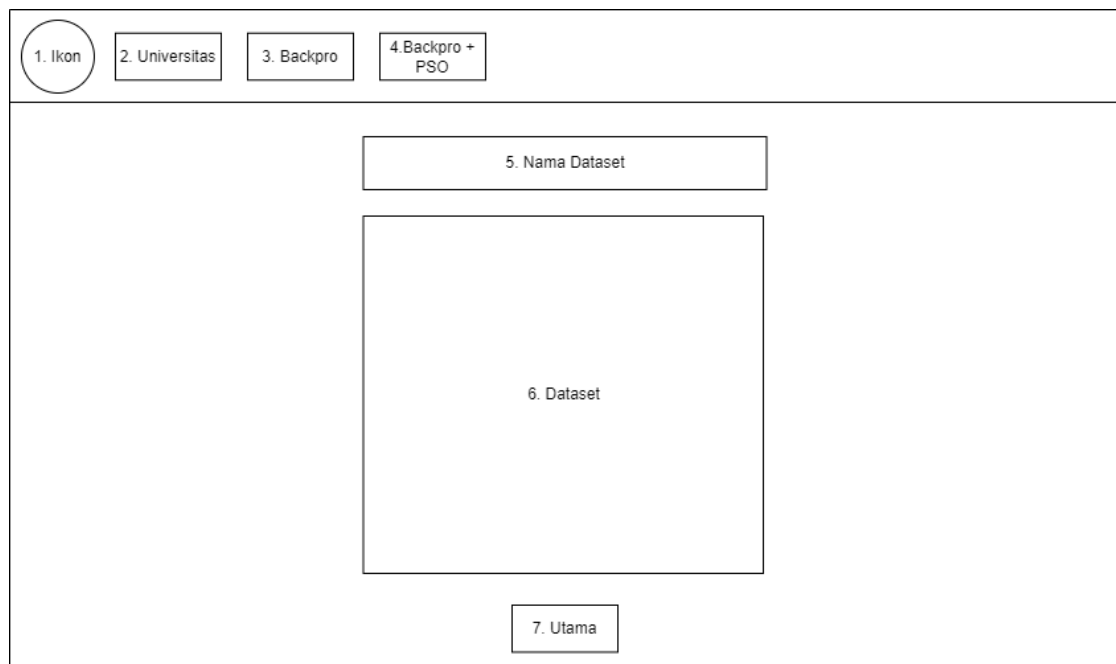
Gambar 3.10. Halaman Utama.

Keterangan gambar:

- | | |
|-----------|---|
| 1. Icon | berfungsi untuk menampilkan logo universitas |
| 2. Text | berfungsi untuk menampilkan <i>font</i> universitas |
| 3. Text | berfungsi untuk menampilkan judul penelitian |
| 4. Icon | berfungsi untuk menampilkan logo universitas |
| 5. Text | berfungsi untuk menampilkan nama dan nim peneliti |
| 6. Button | berfungsi untuk mengakses halaman <i>dataset</i> Retinopati Diabetik |
| 7. Button | berfungsi untuk mengakses halaman perhitungan algoritma <i>Backpropagation</i> |
| 8. Button | berfungsi untuk mengakses halaman perhitungan algoritma <i>Backpropagation</i> + algoritma <i>Particle Swarm Optimization</i> |

b. Halaman *Dataset*

Ketika *user* memilih *button* dataset, maka sistem akan berpindah ke halaman dataset dan akan menampilkan data dari Retinopati Diabetik. Pada **Gambar 3.9.** merupakan rancangan dari halaman dataset penelitian.



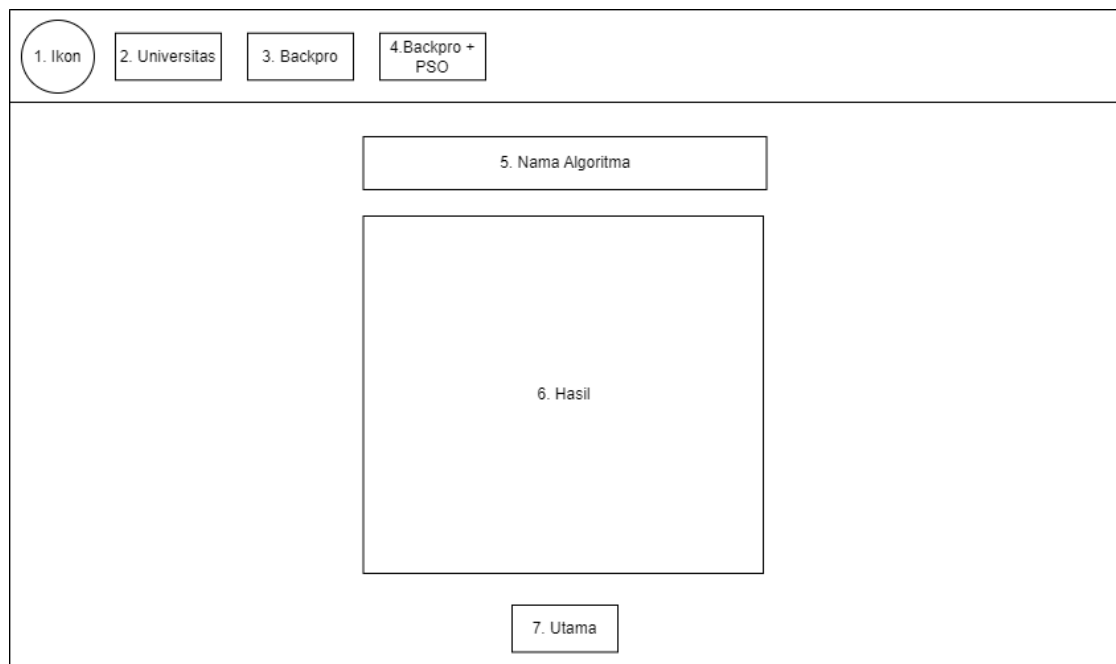
Gambar 3.11. Halaman *Dataset*.

Keterangan gambar:

1. Icon berfungsi untuk menampilkan logo universitas
2. Text berfungsi untuk menampilkan *font* universitas
3. Button berfungsi untuk mengakses halaman perhitungan algoritma *Backpropagation*
4. Button mengakses halaman perhitungan algoritma *Backpropagation* + algoritma *Particle Swarm Optimization*
5. Text berfungsi untuk menampilkan nama *dataset*
6. Label berfungsi untuk menampilkan *dataset*
7. Button berfungsi untuk mengakses halaman utama

c. Halaman Hasil

Ketika user memilih button *Backpropagation* atau *Backpropagation + Particle Swarm Optimization*, maka sistem akan menampilkan halaman dari algoritma yang dipilih. Halaman ini berisi hasil perhitungan dari kerja algoritma yang akan ditampilkan dalam bentuk *Confusion Matrix*. Pada **Gambar 3.10.** merupakan rancangan dari halaman dataset penelitian.



Gambar 3.12. Halaman Hasil .

Keterangan gambar:

1. Icon berfungsi untuk menampilkan logo universitas
2. Text berfungsi untuk menampilkan *font* universitas
3. Button berfungsi untuk mengakses halaman perhitungan algoritma *Backpropagation*
4. Button mengakses halaman perhitungan algoritma *Backpropagation* + algoritma *Particle Swarm Optimization*
5. Text berfungsi untuk menampilkan nama algoritma
6. Label berfungsi untuk menampilkan hasil perhitungan
7. Button berfungsi untuk mengakses halaman utama

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Pada bab ini, selanjutnya akan dijelaskan mengenai hasil dari implementasi sistem yang telah selesai dirancang. Pelaksanaan penelitian ini menggunakan bahasa pemrograman *Python 3.10* dan *Visual Studio Code* Versi 1.52.1.

Terdapat 1.151 dataset dengan 21 atribut deteksi penyakit Retinopati Diabetik yang berasal dari *UCI Machine Learning Repository* oleh Balint Antal, Andras Hajdu.

Tabel 4.1 Keterangan Dataset Retinopati Diabetik

No.	Atribut	Keterangan
1	id	Nomor identitas
2	0	Hasil penilaian kualitas.
3	1	Hasil pra-skrining (Diabetik Retinopati tidak parah/ Diabetik Retinopati parah).
4	2-7	Jumlah MA (pendektesian mikroaneurisma) yang terdeteksi pada tingkat kepercayaan $\alpha = 0,5, \dots, 1.0$, ulangan.
5	8-16	Banyaknya piksel eksudat pada tingkat kepercayaan $\alpha = 0,1, \dots, 1.0$, ulangan.
6	17	Jarak euclidean dari pusat makula dan pusat cakram optik.
7	18	Hasil klasifikasi berbasis AM (Modulasi Amplitudo)/FM (Modulasi Frekuensi) (Diabetik Retinopati tidak parah/ Diabetik Retinopati

		parah).
8	Class	Hasil Deteksi Penyakit (Ya/Tidak).

Adapun beberapa tampilan yang dibangun pada sistem ini yaitu Halaman Utama, Halaman Dataset, Halaman *Backpropagation*, Halaman *Particle Swarm Optimization*, Halaman Hasil *Backpropagation*, dan Halaman Hasil *Particle Swarm Optimization*.

4.1.1. Halaman Utama

Saat sistem pertama kali dijalankan, maka tampilan yang akan keluar adalah Halaman Utama. Halaman utama ini berisikan pengenalan penulis yang berupa judul skripsi, nama penulis, NIM penulis, logo universitas, serta *button* dataset, *button Backpropagation*, dan *button Backpropagation+PSO*. Pada sisi atas terdapat pula *navbar* yang berisikan *button* menuju ke halaman *Backpropagation* dan halaman *Particle Swarm Optimization*. Pada **Gambar 4.1** merupakan tampilan halaman utama yang telah dijelaskan.



Gambar 4.1. Halaman Utama

4.1.2. Halaman Backpropagation

Berikutnya adalah Halaman *Backpropagation* yang menampilkan informasi seperti pengertian dari *Backpropagation* beserta rumusannya. Pada **Gambar 4.2** tampilan halaman *Backpropagation* yang telah dijelaskan.

BACKPROPAGATION - NEURAL NETWORK

Algoritma Backpropagation memiliki 3 arsitektur yang merupakan kumpulan dari neuron-neuron yang tersusun dalam lapisan (layers), yaitu input layer, hidden layer, dan output layer. Adapun langkah-langkah perhitungan algoritma Backpropagation sebagai berikut :

Langkah 0 :
Inisialisasi bobot dengan bilangan acak terkecil.

Langkah 1 :
Jika kondisi berhenti bernilai salah, maka lakukan langkah 2 – 9.

Langkah 2 :
Untuk setiap pasangan *training*, lakukanlah langkah 3 – 8.

Fase 1 : Feedforward

Langkah 3 :
Setiap unit *input* ($X_i, i = 1, \dots, n$) menerima sinyal *input* x_i dan diteruskan ke sinyal seluruh lapisan diatasnya (*hidden unit*).

Langkah 4 :
Setiap *hidden unit* ($Z_j, j = 1, \dots, p$) jumlahkanlah bobot sinyal *input* :

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

gunakan fungsi aktivasi berikut untuk menghitung sinyal *output*,

$$z_j = f(z_{in_j})$$

dan mengirimkan sinyal kepada semua *unit* pada lapisan diatasnya (*output unit*).

Keterangan :

z_{in_j} = Unit *hidden* j
 v_{0j} = Bias pada unit *hidden* j
 x_i = *Input* unit i
 v_{ij} = unit *input* i pada *hidden layer* j

Gambar 4.2. Halaman *Backpropagation*

4.1.3. Halaman Particle Swarm Optimization

Berikut ini merupakan Halaman *Particle Swarm Optimization* yang menampilkan informasi seperti pengertian dari *Particle Swarm Optimization* beserta rumusannya. Pada **Gambar 4.3** tampilan halaman *Particle Swarm Optimization* yang telah ditunjukkan.

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan algoritma optimasi yang berguna untuk pengambilan keputusan atau mengoptimalkan perencanaan jalur. PSO ditemukan oleh Eberhart dan Kennedy pada tahun 1995 yang terinspirasi dari perilaku sekumpulan burung. Adapun algoritma Particle Swarm Optimization (PSO) sebagai berikut :

1. Menginisialisasi kecepatan dan populasi partikel secara random
2. Mulai dengan iterasi baru
3. Menentukan $p\text{-best}$ tiap partikel dan mengganti $p\text{-best}$ yang lebih baik dari sebelumnya
4. Menentukan nilai $g\text{-best}$
5. Update tiap kecepatan partikel menggunakan persamaan dari :

$$V_{id}^{t+1} = W V_{id}^t + c_1 \cdot r_1 \cdot (pbest_{id}^t - x_{id}^t) + c_2 \cdot r_2 \cdot (p_{gd}^t - x_{id}^t)$$

Keterangan :

V_{id}^{t+1} = Kecepatan partikel i dari iterasi t
 W = Bobot inersia
 c_1 = Partikel yang mengikuti $p\text{-best}$
 c_2 = Partikel yang mengikuti $g\text{-best}$
 r_1 dan r_2 = Bilangan acak (dari 0 sampai 1)
 $pbest_{id}^t$ = Posisi terbaik dari partikel i
 p_{gd}^t = Partikel terbaik pada kelompok/kawanan
 x_{id}^t = Posisi partikel i

6. Update posisi pada tiap partikel :

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$$

7. Iterasi berhenti jika terminasi terpenuhi. Lakukan Langkah kembali ketahap3 apabila terminasi tidak terpenuhi.

Gambar 4.3. Halaman *Particle Swarm Optimization*

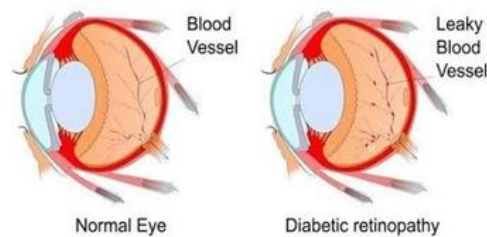
4.1.4. Halaman Dataset

Halaman dataset ini akan menampilkan penjelasan dan sumber data Retinopati Diabetik yang akan dikelola. Nantinya data ini dapat di *download* oleh *user* dalam bentuk *file .xls*. Pada **Gambar 4.4** tampilan halaman dataset yang telah ditunjukkan.

RETINOPATI DIABETIK

Penyakit Retinopati Diabetes merupakan penyakit yang diakibatkan dari diabetes melitus. Retinopati Diabetes termasuk penyakit yang menyebabkan kematian. Penyebab terjadinya Retinopati Diabetes ini adalah tingginya kadar glukosa dalam darah atau memecahnya pembuluh darah di retina. Umumnya penyakit ini akan mengganggu penglihatan mata dan dapat mengakibatkan kebutaan jika tidak cepat ditangani.

Diabetic Retinopathy



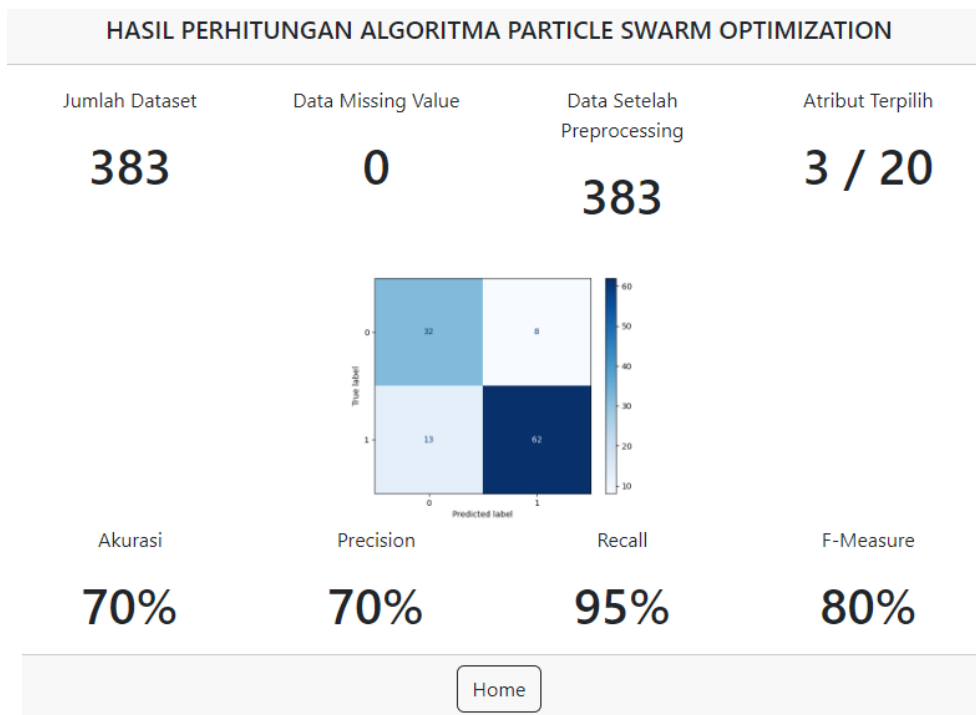
Dapat dilihat pada Gambar diatas bahwa seseorang yang mengidap penyakit Retinopati Diabetik akan mengalami kerusakan pada pembuluh darah. Berdasarkan data yang dirangkum pada penelitian di beberapa daerah, Indonesia mendapati 42,6% yang mengidap diabetik retinopati dan 10% diantaranya mengalami kebutaan dan akan meningkat hingga tahun 2030.

[Link Download File Retinopati Diabetik.csv](#)

Gambar 4.4. Halaman dataset

4.1.5. Halaman Hasil

Halaman terakhir dari sistem ini adalah halaman hasil. Halaman hasil terbagi atas 2 halaman, yaitu halaman hasil *Backpropagation* dan halaman hasil *Backpropagation+PSO*. Halaman ini akan menampilkan hasil akhir dari dataset yang di uji menggunakan *confusion matrix* seperti *accuracy*, *recall*, *precision*, dan *F-Measure* dari algoritma *Backpropagation* dan *Backpropagation+PSO*. Pada **Gambar 4.5** merupakan contoh tampilan halaman hasil perhitungan dari algoritma *Backpropagation*.



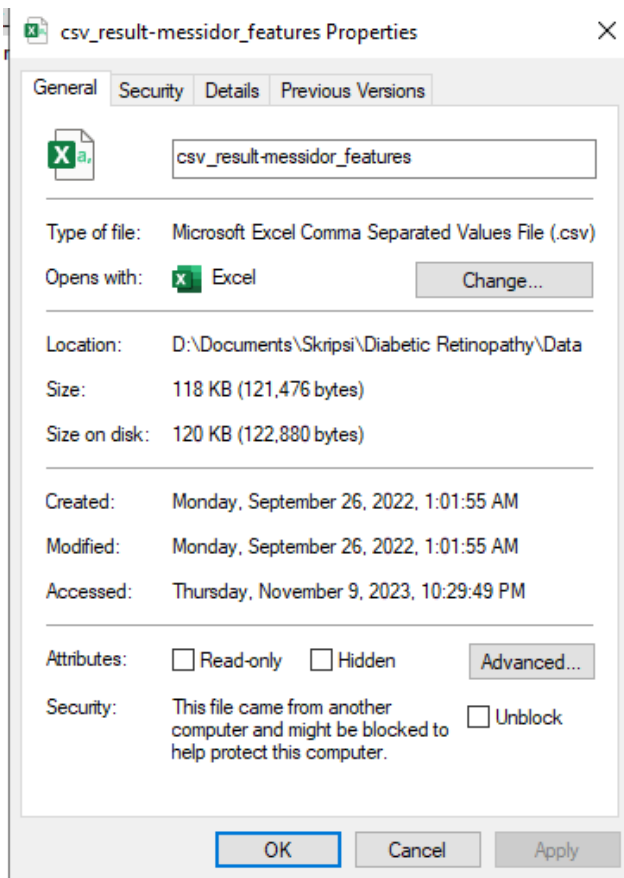
Gambar 4.5. Halaman hasil

4.2. Pengolahan Dataset Sistem

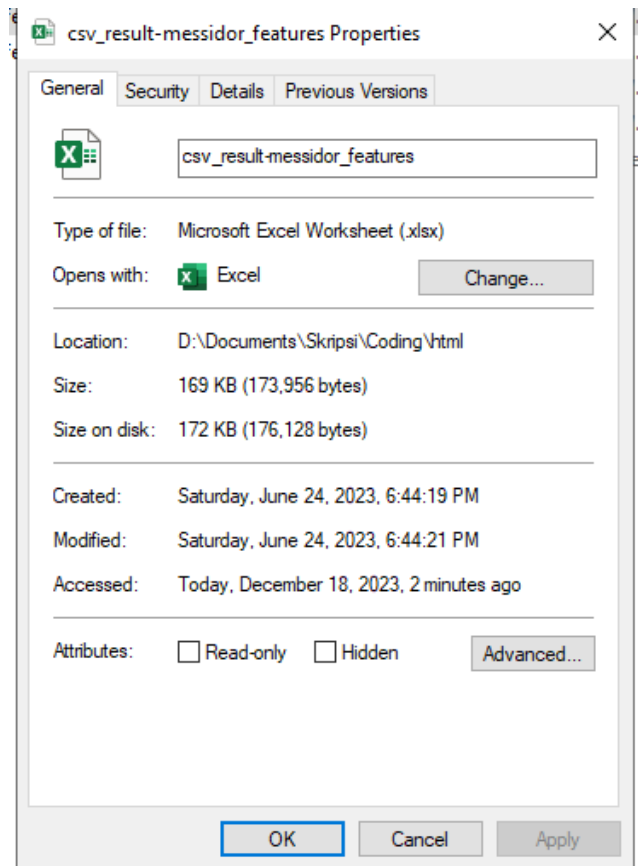
Pengolahan pada dataset akan memasuki beberapa tahapan yang berguna untuk memudahkan dalam melakukan proses klasifikasi, yaitu melakukan transformasi pada dataset, melakukan *feature selection*, membagi *training data* dan *testing data*, melakukan klasifikasi dengan Algoritma *Backpropagation* dan menghitung akurasi dari algoritma tersebut.

4.2.1 Transformasi Data

Untuk memudahkan pengelolaan dataset Retinopati Diabetik yang memiliki format awal .csv terlebih dahulu digantikan dengan ekstensi file menjadi .xls seperti pada contoh **Gambar 4.6** dan **Gambar 4.7**.



Gambar 4.6 File .csv (Sebelum)



Gambar 4.7 File .xls (Sesudah)

4.2.2 Data Reduction

Data *Reduction* bertujuan untuk mengurangi jumlah data tanpa mengganggu hasil analisis data. Pada **Gambar 4.8** dan **Gambar 4.9** dilakukan pengurangan data dari 1.152 dataset menjadi 383 dataset Retinopati Diabetik. Hal ini dikarenakan rendahnya hasil akurasi sehingga dilakukan proses pemilihan data manual dengan cara menghapus data yang bernilai 0 atau nol, maka jumlah penyortiran data berjumlah 383 data. Namun, penulis akan tetap melakukan eksekusi 1.151 yang bertujuan untuk melihat hasil perbandingannya.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U																				
1	id		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Class																			
2	1	1	1	1	22	22	22	19	18	14	49.8958	17.776	5.27092	0.77176	0.01863	0.00686	0.00392	0.00392	0.4869	0.10003	1	0																			
3	2	1	1	1	24	24	22	18	16	13	57.7099	23.8	3.32542	0.23419	0.0039	0.0039	0.0039	0.0039	0.52091	0.14441	0	0																			
4	3	1	1	1	62	60	59	54	47	33	55.8314	27.9939	12.6875	4.85228	1.39389	0.37325	0.04182	0.00774	0.5309	0.12855	0	1																			
5	4	1	1	1	55	53	53	50	43	31	40.4672	18.446	9.1189	3.07943	0.84026	0.27243	0.00765	0.00153	0.48328	0.11479	0	0																			
6	5	1	1	1	44	44	44	41	39	27	18.0263	8.57071	0.41038		0	0	0	0	0.47594	0.12357	0	1																			
7	6	1	1	1	44	43	41	41	37	29	28.3564	6.93564	2.30577	0.32372		0	0	0	0	0.50283	0.12674	0	1																		
8	7	1	0	29	29	29	27	25	16	15.4484	9.11382	1.63349		0	0	0	0	0	0.54174	0.13958	0	1																			
9	8	1	1	6	6	6	6	2	1	20.6796	9.49779	1.22366	0.15038		0	0	0	0	0.57632	0.07107	1	0																			
10	9	1	1	1	22	21	18	15	13	10	66.6919	23.5455	6.15112	0.49637		0	0	0	0.50007	0.11679	0	1																			
11	10	1	1	1	79	75	73	71	64	47	22.1418	10.0544	0.87463	0.09978	0.02339		0	0	0.56096	0.10913	0	1																			
12	11	1	1	1	45	45	45	43	40	32	84.3584	50.9775	17.2937	1.97442		0	0	0	0.54601	0.11238	0	0																			
13	12	1	0	25	25	25	23	22	18	22.48	13.95	0.43623	0.11612		0	0	0	0	0.55168	0.13966	1	0																			
14	13	1	1	1	70	69	65	63	63	50	10.5601	3.10836	0.62551	0.28796	0.10399	0.0048		0	0.5344	0.08959	0	1																			
15	14	1	1	1	48	43	39	32	27	18	23.0128	6.73758	2.4039	0.18924	0.01144		0	0	0.50155	0.13829	1	1																			
16	15	1	1	1	94	93	92	89	86	77	8.61082	1.98132	0.40118	0.0661		0	0	0	0.54128	0.12451	0	0																			
17	16	1	1	1	20	18	16	15	13	9	65.1137	33.1248	8.78538	0.67354	0.05181	0.00293	0.00098	0.00098	0.56946	0.08994	1	0																			
18	17	1	1	1	105	95	81	66	46	32	123.053	70.571	37.4099	19.9373	14.7867	6.11491	2.34574	1.00224	0.52446	0.13425	1	1																			
19	18	1	1	1	25	25	24	23	22	19	17.0341	9.97694	1.06724	0.48483	0.46779	0.3067	0.18898	0.13011	0.552	0.10843	0	0																			
20	19	1	1	1	64	64	63	58	55	40	19.6735	6.06487	0.90734	0.08011		0	0	0	0.55118	0.09859	0	0																			
21	20	1	0	46	41	39	32	23	15	115.534	21.2933	9.66574	2.27668	0.3294	0.186	0.11846	0.0717	0.54047	0.10495	1	1																				
22	21	1	1	1	37	37	37	34	31	23	61.3576	35.1659	8.11403	1.20487	0.1785	0.01077		0	0.47819	0.11079	0	0																			
23	22	1	1	1	19	17	15	12	12	7	179.704	34.6782	13.019	1.04516	0.023	0.005	0.002		0.47043	0.09401	1	1																			
24	23	1	0	37	34	31	30	28	24	8.81823	3.16154	1.90092	1.52473	1.29287	0.16583		0	0	0.53822	0.09827	0	1																			
25	24	1	1	1	10	10	9	9	9	6	72.9389	20.2854	9.79322	0.91627	0.04081		0	0	0.52893	0.10816	1	0																			
csv result-messidor features																																									

Gambar 4.8 Dataset sebelum transformasi

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U																				
1	id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Class																				
2	1	1	1	1	22	22	22	19	18	14	49.8958	17.776	5.27092	0.77176	0.01863	0.00686	0.00392	0.00392	0.4869	0.10003	1	0																			
3	2	1	1	1	24	24	22	18	16	13	57.7099	23.8	3.32542	0.23419	0.0039	0.0039	0.0039	0.52091	0.14441	0	0																				
4	3	1	1	1	62	60	59	54	47	33	55.8314	27.9939	12.6875	4.85228	1.39389	0.37325	0.04182	0.00774	0.5309	0.12855	0	1																			
5	4	1	1	1	55	53	53	50	43	31	40.4672	18.446	9.1189	3.07943	0.84026	0.27243	0.00765	0.00153	0.48328	0.11479	0	0																			
6	5	1	1	1	20	18	16	15	13	9	65.1137	33.1248	8.78538	0.67354	0.05181	0.00293	0.00098	0.00098	0.56946	0.08994	1	0																			
7	6	1	1	1	105	95	81	66	46	32	123.053	70.571	37.4099	19.9373	14.7867	6.11491	2.34574	1.00224	0.52446	0.13425	1	1																			
8	7	1	1	1	25	25	24	23	22	19	17.0341	9.97694	1.06724	0.48483	0.46779	0.3067	0.18898	0.13011	0.552	0.10843	0	0																			
9	8	1	0	46	41	39	32	23	15	115.534	21.2933	9.66574	2.27668	0.3294	0.186	0.11846	0.0717	0.54047	0.10495	1	1																				
10	9	1	1	5	5	5	5	4	3	133.054	6.89089	2.71837	0.1699	0.01794	0.01161	0.00317	0.00106	0.58863	0.10975	1	1																				
11	10	1	1	40	38	33	25	20	12	73.0827	23.1213	13.0936	5.43738	3.84529	2.02878	0.51857	0.10715	0.52711	0.10513	1	1																				
12	11	1	1	99	98	68	53	42	27	298.069	50.2697	33.6937	10.9892	6.12444	1.74896	0.35844	0.18128	0.55648	0.11742	1	1																				
13	12	1	1	1	103	89	83	71	60	38	11.0251	3.76234	0.01559	0.01091	0.00312	0.00312	0.00156	0.00156	0.48857	0.13409	0	1																			
14	13	1	1	1	107	98	77	63	47	28	199.429	44.8209	25.5349	7.70897	0.24247	0.01145	0.00312	0.00208	0.49131	0.1384	1	1																			
15	14	1	1	1	43	42	42	39	37	27	39.211	18.2755	3.93916	0.5842	0.01821	0.0091	0.00607	0.00455	0.47098	0.12139	0	0																			
16	15	1	0	16	16	16	13	10	8	45.5458	22.4931	4.82899	1.67217	0.43302	0.26387	0.02996	0.01063	0.51083	0.12372	0	1																				
17	16	1	1	1	44	43	42	41	39	27	173.757	38.2955	21.4309	8.54797	3.13531	1.043	0.17138	0.08096	0.52022	0.1146	1	1																			
18	17	1	1	1	7	7	7	7	7	4	37.2389	17.4445	2.88848	0.0783	0.03438	0.00859	0.00573	0.00382	0.49992	0.14896	0	0																			
19	18	1	1	1	15	15	14	12	10	9	70.4139	40.1634	11.1613	1.59832	0.31716	0.01157	0.00578	0.00096	0.42417	0.1205	1	0																			
20	19	1	1	1	19	19	18	16	12	11	32.665	13.683	4.32582	0.41333	0.10749	0.04104	0.00391	0.00391	0.50517	0.10456	0	0																			
21	20	1	1	1	107	97	89	73	61	39	46.8983	26.6253	11.2907	8.11985	5.25833	3.75899	2.16976	1.3302	0.51187	0.09752	0	1																			
22	21	1	1	1	22	19	19	15	12	9	70.1923	24.9713	4.4037	0.71367	0.39467	0.20756	0.08998	0.03988	0.51867	0.09816	1	1																			
23	22	1	1	1	15	15	14	14	11	9	120.038	55.1413	9.17627	1.02097	0.00769	0.00481	0.00481	0.00385	0.51462	0.08556	0	0																			
24	23	1	1	1	56	55	46	41	37	28	6.96079	2.74732	1.54412	0.63141	0.20181	0.16053	0.13301	0.11772	0.52287	0.10243	1	1																			
25	24	1	1	1	14	14	14	14	12	9	125.732	48.659	16.775	1.69486	0.00595	0.00595	0.00595	0.00595	0.55084	0.11107	1	0																			
Sheet1																						⌵		⌵		⌵		⌵		⌵		⌵		⌵		⌵		⌵		⌵	

Gambar 4.9 Dataset sesudah transformasi

Data *Reduction* dilakukan untuk mempermudah pengelolaan data serta untuk meningkatkan hasil akurasi dataset Retinopati Diabetik pada bahasa pemrograman.

4.3. Pengujian Sistem

Setelah melewati tahap pengolahan data, selanjutnya adalah pengujian sistem. Pengujian sistem merupakan tahap akhir dari implementasi sistem yang penulis

rancang. Pengujian sistem dapat dikatakan berhasil dan layak digunakan apabila sistem dapat berjalan sesuai dengan kebutuhan dan tujuan rancangannya. Hasil pengujian sistem ini akan menampilkan 2 data berbeda, yaitu data *original* sebanyak 1.151 dataset dan data yang telah di *preprocessing* yaitu data *reduction* sebanyak 383 dataset. Berikut merupakan hasil dari pengujian algoritma *Backpropagation* dan *Particle Swarm Optimization*.

4.3.1 Dataset *Original* (1.151 Data)

Hasil pengujian menggunakan algoritma *Backpropagation* dan *Particle Swarm Optimization* menggunakan klasifikasi *Data Mining* dilaksanakan dengan membagikan 70% data *training* dan 30% data *testing*. Data Retinopati Diabetik berjumlah 1.151 data dengan 806 data *training* dan 346 data *testing*. Berikut merupakan perhitungan pembagian dari data *training* dan data *testing*.

- a. Menghitung Data *Testing* 30%

$$\text{Data Testing} = \frac{30 \times 1.152}{100} = \frac{34.560}{100} = 345,6 = 346$$

Maka hasil data *testing* 30% dari 1.151 data adalah **346**

- b. Menghitung Data *Training* 70%

$$\text{Data Training} - \text{Data Testing} = 1.152 - 346 = 806$$

Maka hasil data *training* 70% dari 1.151 data adalah **806**

4.3.1.1 Hasil Pengujian Algoritma *Backpropagation*

Pada dataset original dalam pengimplementasian data Retinopati Diabetik pada Algoritma *Backpropagation* sebanyak 1.151 akan dilakukan pengulangan sebanyak 20 iterasi seperti pada **Tabel 4.2**.

Tabel 4.2 Hasil Akurasi *Backpropagation*

Iterasi	Eksekusi	Confusion Matrix BPNN + PSO
1	1	61
	2	61
	3	61
	4	61
	5	61
	6	61
	7	61
	8	61
	9	61
	10	61
2	1	61
	2	61
	3	61
	4	61
	5	61
	6	61
	7	61
	8	61
	9	61
	10	61

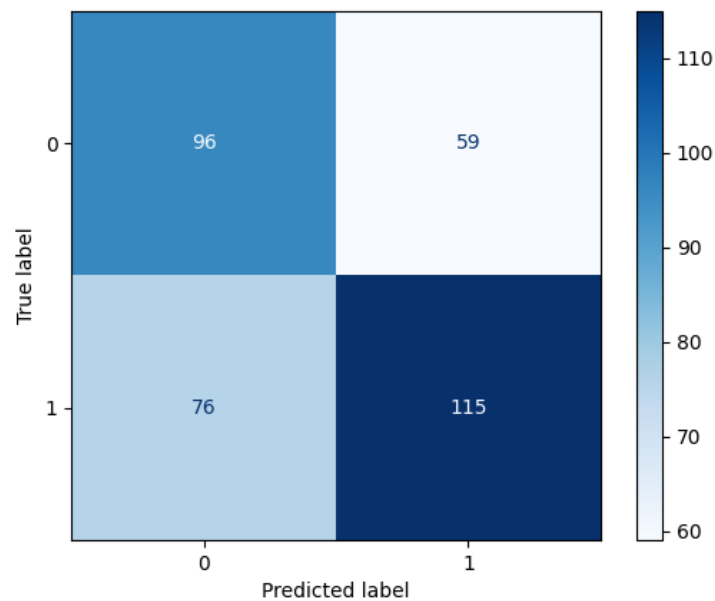
Nilai Min = 61

Nilai Max = 61

$$\begin{aligned}
 \text{Rata-Rata} &= \frac{61+61+61+61+61+61+61+61+61+61+61+61+61+61+61+61+61+61+61+61}{20} \\
 &= \frac{1.200}{20} = \mathbf{61\%}
 \end{aligned}$$

Pada **Tabel 4.2** merupakan perhitungan *Confusion Matrix* dari hasil perhitungan sistem algoritma *Backpropagation* sebelum melakukan *preprocessing* data dan **Gambar 4.10** adalah hasil *Confusion Matrix* dari sistem. Nilai rata-rata akurasi pada

data Retinopati Diabetik adalah sebesar 61%. Dapat disimpulkan bahwa algoritma Backpropagation memiliki nilai yang tetap dan tidak berubah-ubah.



Gambar 4.10. *Confusion Matrix Algoritma Backpropagation*

1. *Accuracy*

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{96 + 115}{96 + 115 + 59 + 76} = 61\%$$

2. *Precision*

$$\frac{TP}{TP + FP} = \frac{96}{96 + 59} = 66\%$$

3. *Recall*

$$\frac{TP}{TP + FN} = \frac{96}{96 + 76} = 60\%$$

4. *F-Measure*

$$2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{0,6 \times 0,5}{0,6 + 0,5} = 0,63 = 63\%$$

4.3.1.2 Hasil Pengujian Algoritma *Particle Swarm Optimization* dan *Backpropagation*

Pada dataset original dalam pengimplementasian data Retinopati Diabetik sebanyak 1.152 akan dilakukan pengulangan sebanyak 20 iterasi seperti pada **Tabel 4.3**.

Tabel 4.3 Hasil Akurasi *Backpropagation* + *PSO*

Iterasi	Eksekusi	Confusion Matrix BPNN + PSO
1	1	55
	2	50
	3	60
	4	57
	5	58
	6	68
	7	60
	8	66
	9	58
	10	59
2	1	58
	2	52
	3	56
	4	50
	5	58
	6	57
	7	59
	8	63
	9	59
	10	61

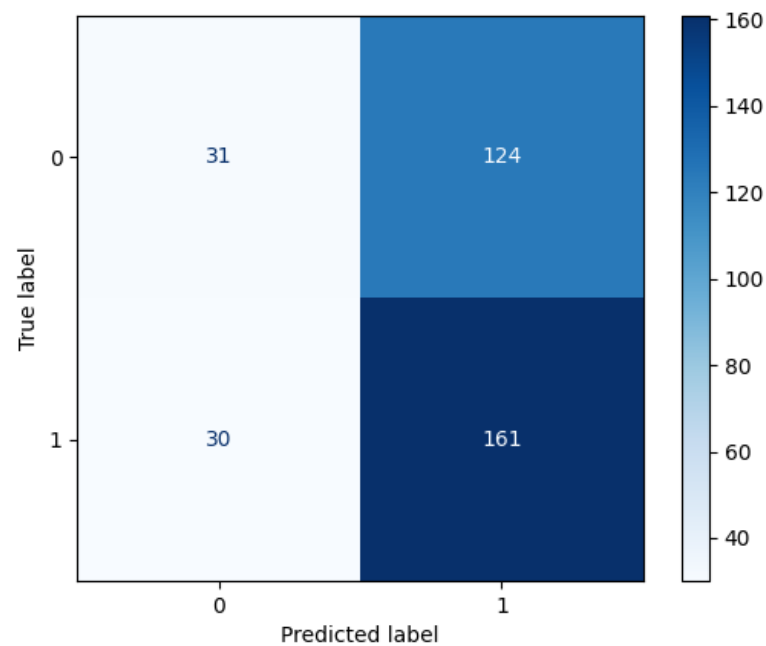
Nilai Min = 68

Nilai Max = 50

Rata-Rata = $\frac{55+50+60+57+58+68+60+66+58+59+58+52+56+50+58+57+59+63+59+61}{20}$

$$= \frac{1.106}{20} = 55.2 = \mathbf{55\%}$$

Pada **Gambar 4.12** merupakan perhitungan *Confusion Matrix* dari hasil perhitungan sistem algoritma *Backpropagation* dengan *Particle Swarm Optimization* sebelum melakukan *preprocessing* data dan **Gambar 4.13** adalah hasil *Confusion Matrix* dari sistem. Nilai rata-rata akurasi pada data Retinopati Diabetik adalah sebesar 55%.



Gambar 4.12 *Confusion Matrix Backpropagation+PSO*

1. *Accuracy*

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{31 + 161}{31 + 161 + 124 + 30} = 55\%$$

2. *Precision*

$$\frac{TP}{TP + FP} = \frac{31}{31 + 124} = 56\%$$

3. *Recall*

$$\frac{TP}{TP + FN} = \frac{31}{31 + 161} = 84\%$$

4. *F-Measure*

$$2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{0.56 \times 0.84}{0.56 + 0.84} = 68\%$$

4.3.2 Dataset Reduction (383 Data)

Pada dataset original memiliki hasil akurasi 61% pada algoritma *Backpropagation* dan 55% pada algoritma *Backpropagation* dengan *Particle Swarm Optimization*. Dikarenakan hasil akurasi penelitian rendah, maka penulis melakukan proses reduksi data dengan menghapus data yang bernilai 0 atau nol sehingga jumlah dataset yang telah direduksi sebesar 383 data. Hasil pengujian menggunakan algoritma *Backpropagation* dan *Particle Swarm Optimization* menggunakan klasifikasi *Data Mining* dilaksanakan dengan membagikan 70% data *training* dan 30% data *testing*. Data Retinopati Diabetik berjumlah 383 data dengan 268 data *training* dan 115 data *testing*. Berikut merupakan perhitungan pembagian dari data *training* dan data *testing*.

- a. Menghitung Data *Testing* 30%

$$\text{Data Testing} = \frac{30 \times 383}{100} = \frac{11.490}{100} = 114,9 = 115$$

Maka hasil data *testing* 30% dari 383 data adalah **115**

- b. Menghitung Data *Training* 70%

$$\text{Data Training} - \text{Data Testing} = 383 - 115 = 268$$

Maka hasil data *training* 70% dari 383 data adalah **268**

4.3.2.1 Hasil Pengujian Algoritma *Backpropagation*

Pada dataset original dalam pengimplementasian data Retinopati Diabetik pada Algoritma *Backpropagation* sebanyak 383 akan dilakukan pengulangan sebanyak 20 iterasi seperti pada **Tabel 4.4**.

Tabel 4.4 Hasil Akurasi *Backpropagation*

Iterasi	Eksekusi	Confusion Matrix BPNN + PSO
1	1	70
	2	70
	3	70
	4	70
	5	70
	6	70
	7	70
	8	70
	9	70
	10	70
2	1	70
	2	70
	3	70
	4	70
	5	70
	6	70
	7	70
	8	70
	9	70
	10	70

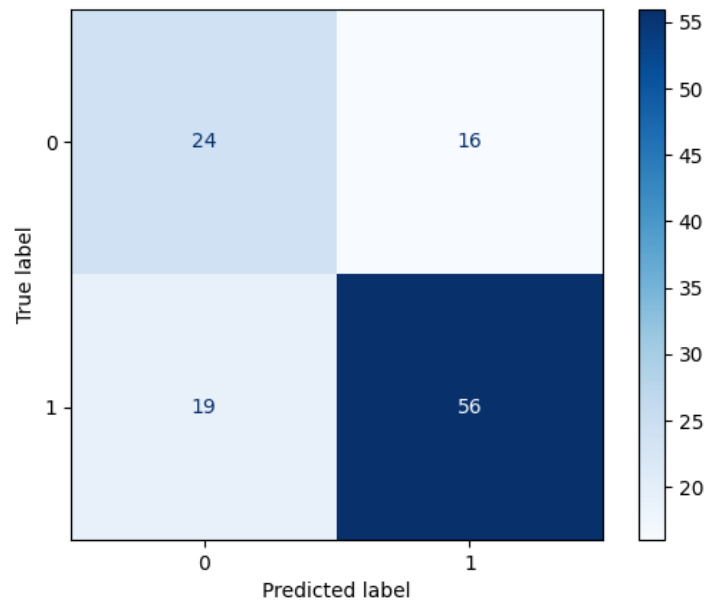
Nilai Min = 70

Nilai Max = 70

$$\begin{aligned}
 \text{Rata-Rata} &= \frac{70+70+70+70+70+70+70+70+70+70+70+70+70+70+70+70+70+70+70+70}{20} \\
 &= \frac{1.400}{20} = \mathbf{70\%}
 \end{aligned}$$

Pada **Tabel 4.4** merupakan perhitungan *Confusion Matrix* dari hasil perhitungan sistem algoritma *Backpropagation* sebelum melakukan *preprocessing* data dan **Gambar 4.14** adalah hasil *Confusion Matrix* dari sistem. Nilai rata-rata akurasi pada

data Retinopati Diabetik adalah sebesar 70%. Dapat disimpulkan bahwa algoritma Backpropagation memiliki nilai yang tetap dan tidak berubah-ubah.



Gambar 4.14. *Confusion Matrix Algoritma Backpropagation*

1. *Accuracy*

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{24 + 56}{24 + 56 + 16 + 19} = 70\%$$

2. *Precision*

$$\frac{TP}{TP + FP} = \frac{24}{24 + 16} = 78\%$$

3. *Recall*

$$\frac{TP}{TP + FN} = \frac{24}{24 + 19} = 75\%$$

4. *F-Measure*

$$2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{0.78 \times 0.75}{0.78 + 0.75} = 76\%$$

4.3.2.2 Hasil Pengujian Algoritma *Particle Swarm Optimization* dan *Backpropagation*

Hasil pengujian menggunakan algoritma *Backpropagation* dan *Feature Selection Particle Swarm Optimization* menggunakan klasifikasi *Data Mining* dilaksanakan dengan membagikan 70% data *training* dan 30% data *testing*. Data Retinopati Diabetik berjumlah 383 data dengan 268 data *training* dan 115 data *testing*.

Saat sistem dijalankan, pada percobaan algoritma *Backpropagation* dan *Feature Selection Particle Swarm Optimization* ini menghasilkan berbagai hasil akurasi sehingga memerlukan beberapa iterasi untuk mendapatkan hasil yang paling tertinggi. Pada penelitian ini dilakukan iterasi sebanyak 80 kali untuk mencapai nilai akurasi peneliti yang mencapai sedikitnya 80%.

Tabel 4.5 Hasil Akurasi *Backpropagation* + *PSO*

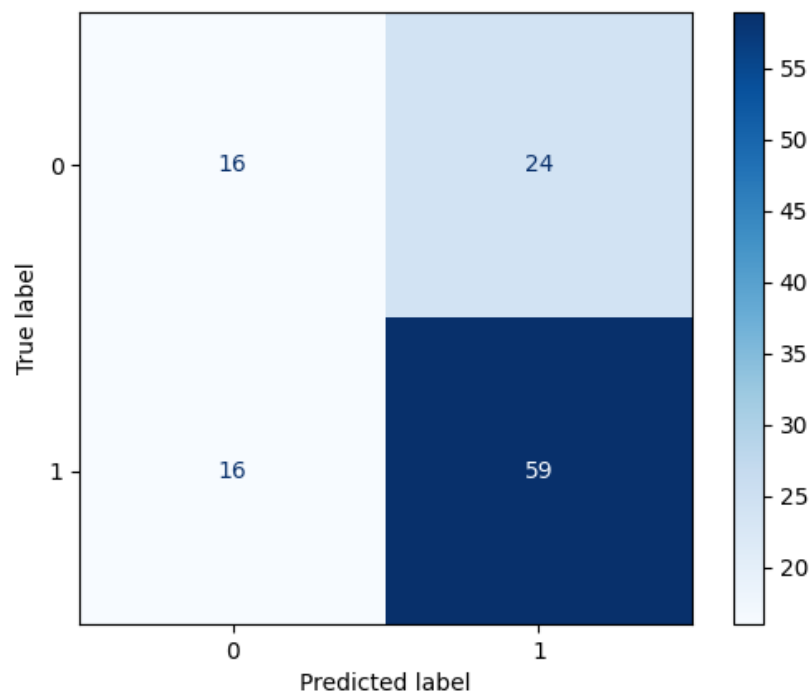
Iterasi	Eksekusi	Confusion Matrix BPNN + PSO
1	1	63
	2	66
	3	63
	4	64
	5	62
	6	67
	7	57
	8	67
	9	66
	10	63
2	1	63
	2	57
	3	64
	4	65
	5	61
	6	70
	7	70
	8	59
	9	75
	10	71

Nilai Min = 75

Nilai Max = 57

$$\begin{aligned} \text{Rata-Rata} &= \frac{55+50+60+57+58+68+60+66+58+59+58+52+56+50+58+57+59+63+59+61}{20} \\ &= \frac{1.293}{20} = 64.65 = \mathbf{65\%} \end{aligned}$$

Pada **Tabel 4.5** merupakan berbagai hasil eksekusi algoritma *Backpropagation* dan *Particle Swarm Optimization* dari sistem. Nilai rata-rata dari 20 iterasi yaitu sebesar 65%. Pada **Gambar 4.15** merupakan perhitungan *Confusion Matrix* dari hasil perhitungan sistem algoritma *Backpropagation* setelah melakukan *preprocessing* data.



Gambar 4.15. *Confusion Matrix* Algoritma *PSO + Backpropagation*

1. *Accuracy*

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{16 + 59}{16 + 59 + 24 + 16} = 0,65 = \mathbf{65\%}$$

2. *Precision*

$$\frac{TP}{TP + FP} = \frac{16}{16 + 24} = 69\%$$

3. *Recall*

$$\frac{TP}{TP + FN} = \frac{16}{16 + 16} = 84\%$$

4. *F-Measure*

$$2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \times \frac{0.69 \times 0.84}{0.69 + 0.84} = 76\%$$

4.4. Hasil Perbandingan Pengujian Algoritma

Setelah melakukan perhitungan dari pengujian algoritma *Backpropagation* dan *Backpropagation + Particle Swarm Optimization* menggunakan data Retinopati Diabetik, tahap selanjutnya adalah membandingkan hasil dari kedua dataset tersebut. Berikut **Tabel 4.6** merupakan hasil akurasi dari 2 dataset klasifikasi Retinopati Diabetik.

Tabel 4.6 Hasil Akurasi Algoritma

Dataset	Akurasi Algoritma <i>Backpropagation</i>	Akurasi Algoritma <i>Backpropagation + Particle Swarm Optimization</i>
1.151	61%	55%
383	70%	65%

Berdasarkan **Tabel 4.3** perbandingan yang sudah dibuat, dalam melakukan klasifikasi pada penyakit Retinopati Diabetik sebanyak 1.151 data memperlihatkan hasil akurasi algoritma *Backpropagation* sebesar 61% sedangkan algoritma *Backpropagation* dengan optimasi *Particle Swarm Optimization* sebesar 55%. Pada data Retinopati Diabetik yang telah dilakukan reduksi sebesar 383 data memperlihatkan hasil akurasi *Backpropagation* sebesar 70% dan hasil akurasi algoritma *Backpropagation* dengan *Particle Swarm Optimization* sebesar 65%. Dari tabel tersebut, dapat disimpulkan bahwa dengan adanya algoritma optimisasi *Particle Swarm Optimization* tidak mampu meningkatkan akurasi pada klasifikasi data Retinopati Diabetik menggunakan algoritma *Backpropagation*.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Bersadarkan hasil penerapan dan pengujian yang telah dilakukan dalam penelitian penyakit retinopati diabetik menggunakan algoritma *Backpropagation* dan *Particle Swarm Optimization*, dapat disimpulkan bahwa:

1. Pada uji validasi kinerja klasifikasi Retinopati Diabetik yang menggunakan 1.151 data dari algoritma *Backpropagation* menghasilkan nilai akurasi sebesar 61% dan algoritma *Backpropagation* + *PSO* mendapatkan nilai akurasi sebesar 55%.
2. Pada uji validasi kinerja klasifikasi Retinopati Diabetik yang menggunakan data yang telah di reduksi yaitu sebesar 383 data dari algoritma *Backpropagation* menghasilkan nilai akurasi sebesar 70% dan algoritma *Backpropagation* + *PSO* mendapatkan nilai akurasi sebesar 65%
3. Pada data klasifikasi penyakit Retinopati Diabetik setelah dilakukan *preprocessing* data (*data reduction*), algoritma *Backpropagation* memiliki nilai akurasi yang lebih tinggi dibandingkan dengan algoritma *Backpropagation* yang menggunakan algoritma optimasi *Particle Swarm Optimization*.

5.2. Saran

Berikut merupakan saran penelitian yang dapat dipertimbangkan untuk kedepannya adalah:

1. Pada penelitian selanjutnya diharapkan klasifikasi data penyakit Retinopati Diabetik juga dilakukan pada penelitian menggunakan algoritma optimasi *Particle Swarm Optimization* dengan algoritma Jaringan Syaraf Tiruan (JST) lainnya, seperti *Convolutional Neural Network (CNN)*
2. Pada penelitian yang akan datang diharapkan menggunakan algoritma optimasi lainnya pada pengklasifikasian penyakit Retinopati Diabetik, seperti algoritma *Backpropagation* dengan algoritma optimasi *Glowworm Swarm Optimization*.
3. Pada penelitian mendatang diharapkan menggunakan algoritma *Backpropagation* dengan optimasi *Particle Swarm Optimization* dengan data klasifikasi penyakit yang berbeda, seperti kardiovaskular, diabetes mellitus, dan lain sebagainya .
4. Pada penelitian yang akan datang diharapkan menggunakan algoritma *Backpropagation* dengan algoritma optimasi *Particle Swarm Optimization* dengan data klasifikasi penyakit Retinopati Diabetik selain dari *UCI Machine Learning*.

DAFTAR PUSTAKA

- Antal, B., & Hadju, A. (2014). An ensemble-based system for automatic screening of diabetic retinopathy. *University of Debrecen*.
- Arifin, T., & Herliana, A. (2018). Optimasi Metode Klasifikasi dengan Menggunakan Particle Swarm Optimization untuk Identifikasi Penyakit Diabetes Retinopathy. *Jurnal Ilmu Komputer dan Informatika*.
- Diana, Indrajit, R. E., & Dazki, E. (2022). Komparasi Algoritma Naive Bayes, Logistic Regression, dan Support Vector Machine pada Klasifikasi File Application Package Kit Android Malware. *Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*.
- Efendi, A. B., & Mashuri, A. A. (2021). Penentuan Standar Kelayakan Sertifikasi Guru Menggunakan Metode Particle Swarm Optimization (PSO) dan Algoritma Klasifikasi Neural Network (NN). *JURNAL INFORMATIKA UPGRIS Vol. 7, No. 1*.
- Fausett, L. (1994). *Fundamental of Neural Network: Architectures, Algorithms, And Applications*. NJ: Prentice-Hall: Englewood Cliffs.
- Hardoni, A., Rini, D. P., & Sukemi. (2021). Integrasi SMOTE pada Naive Bayes dan Logistic Regresion Berbasis Particle Swarm Optimization untuk Prediksi Cacat Perangkat Lunak. *Jurnal Media Informatika Budidarma*.
- Hizham, F. A. (2018). *Implementasi Metode Backpropagation Neural Network (BNN) dalam Sistem Klasifikasi Ketepatan Waktu Kelulusan Mahasiswa (Studi Kasus: Program Studi Sistem Informasi Universitas Jember)*. Universitas Jember.
- Humam, M., Somantri, O., Abdillah, M. B., Romadhon, S. A., Khambali, M., & Rahim, R. (2019). The application of particle swarm optimization using neural network to optimize classification of employee performance assessment. *Journal of Physics: Conference Series*.
- Indrasetianingsih, A., Fitriani, F., & Junita, P. (2021). Klasifikasi Indeks Pembangunan Gender Di Indonesia Tahun 2020 Menggunakan Supervised Machine Learning Algorithms. *Institut Teknologi Sepuluh Nopember*.
- Irnanda, K. F., Windarto, A. P., & Damanik, I. S. (2022). Optimasi Particle Swarm Optimization Pada Peningkatan Prediksi dengan Metode Backpropagation Menggunakan Software RapidMiner. *JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 1*.
- Kantardzic, M. (2020). *Data Mining Concepts, Models, Methods, and Algorithms*. Canada: IEEE Press.
- Khamdani, M. A. (2020). Klasifikasi Penyakit Diabetes Retina Menggunakan Algoritma Backpropagation. *Jurusan Matematika*.

- Kurniadi, R. P., Saedudin, R. R., & Widartha, V. P. (n.d.). Perbandingan Akurasi Algoritma K-Nearest Neighbor dan Logistic Regression untuk Klasifikasi Penyakit Diabetes.
- Marchi, D. L., & Mitchell, L. (2019). *Hands-On Neural Network: Learn how to build and train your first neural network model using Python*. Birmingham: Packt Publishing.
- Mercangos, B. A. (2021). *Applying Particle Swarm Optimization*. USA: International Series in Operations Research & Management Science.
- Muzakkir, I., Syukur, A., & Dewi, I. N. (2014). PENINGKATAN AKURASI ALGORITMA BACKPROPAGATION DENGAN SELEKSI FITUR PARTICLE SWARM OPTIMIZATION DALAM PREDIKSI PELANGGAN TELEKOMUNIKASI YANG HILANG. *Jurnal Pseudocode, Volume 1 Nomor 1*.
- Nikentari, N., Kurniawan, H., Ritha, N., & Kurniawan, D. (2018). OPTIMASI JARINGAN SYARAF TIRUAN BACKPROPAGATION DENGAN PARTICLE SWARM OPTIMIZATION UNTUK PREDIKSI PASANG SURUT AIR LAUT. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*.
- Nikmatun, I. A. (2019). Implementasi Data Mining untuk Klasifikasi Masa Studi Mahasiswa Menggunakan Algoritma K-Nearest Neighbor. *Jurnal SIMETRIS*.
- Norhikmah, & Rumini. (n.d.). Klasifikasi Peminjaman Buku Menggunakan Neural Network Backpropagation. *SISTEMASI : Jurnal Sistem Informasi*.
- Nurdiansyah, Y., Media, D., & Hizam, F. A. (2019). Implementation of Backpropagation Neural Network Method in Classification System of Timeliness of Graduation. *ICSTI*.
- Nurdiansyah, Y., Media, D., & Hizam, F. A. (2018). Implementation of Backpropagation Neural Network Method in Classification System of Timeliness of Graduation. *ICSTI*.
- Parsopoulos, K. E., & Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence Advance and Application*. United States of America: IGI GLOBAL.
- Ramli, Yuniarti, D., & Goejantoro, R. (2013). Perbandingan Metode Klasifikasi Regresi Logistik Dengan Jaringan Saraf Tiruan. *Jurnal EKSPONENSIAL*.
- Samosir, R. O., Wilandari, Y., & Yasin, H. (2015). Perbandingan Metode Klasifikasi Regresi Logistik Biner dan Radial Function Network pada Berat Bayi Lahir Rendah. *JURNAL GAUSSIAN*.
- Santosa, B. (n.d.). Tutorial Particle Swarm Optimization. *ACADEMIA*.
- Surantha, N., Lesmana, T. F., & Isa, S. M. (2021). Sleep stage classification using extreme learning machine and particle swarm optimization for healthcare big data. *Journal Big Data*.
- Tandrio, K. H., Nuzry, K. A., Ardi, Y. P., & Prasetyo, H. (2018). Classification of Wine Types Based on Composition Using Backpropagation And Particle Swarm Optimization.

ITSMART: Jurnal Ilmiah Teknologi dan Informasi.

- Tyasnurita, R., & Pamungkas, A. Y. (2020). Deteksi Diabetik Retinopati menggunakan Regresi Logistik. *ILKOM Jurnal Ilmiah*.
- Wati, D. A., & Rochman, Y. A. (2013). Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO). *Jurnal Rekayasa Sistem Industri Vol. 2, No.1*.
- Windarto, A. P., Nasution, D., Tambunan, A. W., Hasibuan, M. S., Siregar, M. N., Lubis, M. R., . . . Nofriansyah, D. (2020). *Jaringan Syaraf Tiruan: Algoritma Prediksi dan Implementasi*. Medan: Yayasan Kita Menulis.
- Yani, E. (2005). *Pengantar Jaringan Syaraf Tiruan*. Indonesia: MateriKuliah.Com.
- Zajmi, L., Ahmed, F. Y., & Jaharadak, A. A. (2018). Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks. *Hindawi*.

LISTING PROGRAM

```

import numpy as np #libr math
import pandas as pd #libr data
import seaborn as sns #libr grafik
import matplotlib.pyplot as plt #libr menghasilkan gambar, berkaitan dg
seaborn
import pyswarms as ps #libr pso

from sklearn import metrics #libr klasifikasi
from flask import Flask, app, render_template #libr koneksi ke web
from sklearn.neural_network import MLPClassifier #libr bpnn
from sklearn.metrics import ConfusionMatrixDisplay #libr confusion matrix
from sklearn.metrics import classification_report #libr klasifikasi
from sklearn.model_selection import train_test_split
from pyswarms.utils.functions import single_obj as fx

app = Flask(__name__) #pemanggilan web
app.secret_key = 'baby' #nama class

@app.route('/')
def utama():
    return render_template('home.html')

@app.route('/beranda')
def beranda():
    return render_template('home.html')

@app.route('/data')
def data():
    return render_template('data.html')

#BPNN
@app.route('/bpnn')
def bpnn():
    return render_template('bpnn.html')

@app.route('/hasil_bpnn')
def hasil_bpnn():
    # Dataset
    dataset = pd.read_excel("D:\Documents\Skripsi\Coding\html\csv_result-
messidor_features.xlsx") #baca dataset
    # preprocessing
    data = len(dataset) #input jumlah dataset
    dataset = dataset.dropna(axis = 0, how ='any')

```

```

datamv = len(dataset) #input jumlah missing value
x = np.array(dataset.drop(['Class'], axis=1))
y = np.array(dataset['Class'])

# Metode Neural Network
metode = MLPClassifier(hidden_layer_sizes=(6,), random_state=42)

# Backpropagation (BPNN)
X_train, X_test, y_train, y_test = train_test_split(x,y,train_size=.70,
random_state=42)
modelbpnn = metode.fit(X_train, y_train)
hasilbpnn_Xtest = modelbpnn.predict(X_test)

# Confusion Matrix (BPNN)
cm_bpnn = metrics.confusion_matrix(y_test, hasilbpnn_Xtest)
print(classification_report(y_test, hasilbpnn_Xtest))
# Confusion Matrix
cm = metrics.confusion_matrix(y_test, hasilbpnn_Xtest)
plt.figure(1)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,)
disp = disp.plot(cmap="Blues", values_format='')
plt.savefig('D:\Documents\Skrripsi\Coding\html\static\Hasil Confusion
Matrix Backpropagation.png')
plt.close()

# Nilai Confusion Matrix (BPNN)
acc_bpnn = int(np.around(metrics.accuracy_score(y_test,
hasilbpnn_Xtest)*100))
pre_bpnn = int(np.around(metrics.precision_score(y_test,
hasilbpnn_Xtest)*100))
rec_bpnn = int(np.around(metrics.recall_score(y_test,
hasilbpnn_Xtest)*100))
fm_bpnn = int(np.around(metrics.f1_score(y_test, hasilbpnn_Xtest)*100))

# Output
print (classification_report(y_test, hasilbpnn_Xtest))
print('Hasil Confusion Matrix Backpropagation : \n', cm_bpnn)
print('Akurasi NN : ', acc_bpnn)
print('Precision NN: ', pre_bpnn)
print('Recall NN: ', rec_bpnn)
print('F-Measure NN: ', fm_bpnn)
print('Jumlah Dataset : ', data)
print('Jumlah Dataset setelah penanganan missing value: ', datamv)

```

```

        return render_template('hasil_bpnn.html', fm_bpnn=fm_bpnn,
rec_bpnn=rec_bpnn, pre_bpnn=pre_bpnn,
acc_bpnn=acc_bpnn,data=data,datamv=datamv)

# PSO
@app.route('/pso')
def pso():
    return render_template('pso.html')

@app.route('/hasil_pso')
def hasil_pso():
    # Dataset
    dataset = pd.read_excel("D:\Documents\Skripsi\Coding\html\csv_result-
messidor_features.xlsx") #baca dataset
    # Preprocessing
    data = len(dataset) #input jumlah dataset
    dataset = dataset.dropna(axis = 0, how = 'any')

    datamv = len(dataset) #input jumlah missing value
    x = np.array(dataset.drop(['Class'], axis=1))
    y = np.array(dataset['Class'])

    # Metode Neural Network
    metode = MLPClassifier(hidden_layer_sizes=(6,), random_state=42)

    # Particle Swarm Optimization
    def f_per_particle(m, alpha):
        total_features = x.shape[1]
        if np.count_nonzero(m) == 0:
            X_subset = x
        else:
            X_subset = x[:,m==1]
        X_train, X_test, y_train, y_test =
train_test_split(X_subset,y,train_size=.70, random_state=42)
        metode.fit(X_train, y_train)
        P = (metode.predict(X_test) == y_test).mean()
        j = (alpha * (1.0 - P) + (1.0 - alpha) * (1 - (X_subset.shape[1] /
total_features)))
        return j

    def f(x, alpha=0.88):
        n_particles = x.shape[0]
        j = [f_per_particle(x[i], alpha) for i in range(n_particles)]

```

```

        datamv = len(dataset) #input jumlah missing value

        return render_template('hasil_bpnn.html', fm_bpnn=fm_bpnn,
rec_bpnn=rec_bpnn, pre_bpnn=pre_bpnn,
acc_bpnn=acc_bpnn,data=data,datamv=datamv)

# PSO
@app.route('/pso')
def pso():
    return render_template('pso.html')

@app.route('/hasil_pso')
def hasil_pso():
    # Dataset
    dataset = pd.read_excel("D:\Documents\Skripsi\Coding\html\csv_result-
messidor_features.xlsx") #baca dataset
    # Preprocessing
    data = len(dataset) #input jumlah dataset
    dataset = dataset.dropna(axis = 0, how = 'any')

    datamv = len(dataset) #input jumlah missing value
    x = np.array(dataset.drop(['Class'], axis=1))
    y = np.array(dataset['Class'])

    # Metode Neural Network
    metode = MLPClassifier(hidden_layer_sizes=(6,), random_state=42)

    # Particle Swarm Optimization
    def f_per_particle(m, alpha):
        total_features = x.shape[1]
        if np.count_nonzero(m) == 0:
            X_subset = x
        else:
            X_subset = x[:,m==1]
        X_train, X_test, y_train, y_test =
train_test_split(X_subset,y,train_size=.70, random_state=42)
        metode.fit(X_train, y_train)
        P = (metode.predict(X_test) == y_test).mean()
        j = (alpha * (1.0 - P) + (1.0 - alpha) * (1 - (X_subset.shape[1] /
total_features)))
        return j

```

```

def f(x, alpha=0.88):
    n_particles = x.shape[0]
    j = [f_per_particle(x[i], alpha) for i in range(n_particles)]
    return np.array(j)

#Initialize swarm, arbitrary
options = {'c1': 0.5, 'c2': 0.5, 'w':0.9, 'k': 1 , 'p':2}
dimensions = x.shape[1] # dimensions should be the number of features
optimizer = ps.discrete.BinaryPSO(n_particles=2,
dimensions=dimensions,options=options)
#Optimization
cost, pos = optimizer.optimize(fx.sphere, iters=20)

# Backpropagation - PSO (BPNN - PSO)
fiturSeleksiX = x[:,pos==1]
X_train, X_test, y_train, y_test =
train_test_split(fiturSeleksiX,y,train_size=.70, random_state=42)
model_pso = metode.fit(X_train, y_train)
hasilpso_Xtest = model_pso.predict(X_test) # Menentukan hasil prediksi
y_pred

# Confusion Matrix (BPNN)
cm_pso = metrics.confusion_matrix(y_test, hasilpso_Xtest)
print(classification_report(y_test, hasilpso_Xtest))
# Confusion Matrix
cm = metrics.confusion_matrix(y_test, hasilpso_Xtest)
plt.figure(1)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,)
disp = disp.plot(cmap="Blues", values_format='')
plt.savefig('D:\Documents\Skrupsi\Coding\html\static\Hasil Confusion
Matrix PSO.png')
plt.close()

# Nilai Confusion Matrix (NN - PSO)
acc_pso = int(np.around(metrics.accuracy_score(y_test,
hasilpso_Xtest)*100))
pre_pso = int(np.around(metrics.precision_score(y_test,
hasilpso_Xtest)*100))
rec_pso = int(np.around(metrics.recall_score(y_test,
hasilpso_Xtest)*100))
fm_pso = int(np.around(metrics.f1_score(y_test, hasilpso_Xtest)*100))

# Menampilkan Atribut Terpilih (NN - PSO)

```



```

# Menampilkan Atribut Terpilih (NN - PSO)
atribut =
["id","1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16",
"17","18","Class"]
atributArr = np.array(atribut)
atributSel = atributArr[pos==1]
atributTerpilih = str(sum((pos == 1)*1))
atributTotal = str(len(pos))
dataTraining = len(str(X_train))
dataTesting = str(len(X_test))
dataMissing = data - datamv

# Output
print (classification_report(y_test, hasilpso_Xtest))
print('Fitur Terseleksi : \n' + str(sum((pos == 1)*1)) + '/' +
str(len(pos)))
print('Hasil Confusion Matrix Backpropagation & Particle Swarm
Optimization : \n', cm_pso)
print('Akurasi PSO : ', acc_pso)
print('Precision PSO : ', pre_pso)
print('Recall PSO : ', rec_pso)
print('F-Measure PSO : ', fm_pso)
print('Jumlah Dataset : ', data)
print('Jumlah Dataset setelah penanganan missing value: ', datamv)
print('Atribut Yang Terpilih :', atributTerpilih)
print('Atribut Yang Terpilih :', atributSel)

return render_template('hasil_pso.html', atributSel=atributSel,
fm_pso=fm_pso, rec_pso=rec_pso, pre_pso=pre_pso,
acc_pso=acc_pso,data=data,datamv=datamv, atributTerpilih=atributTerpilih,
atributTotal=atributTotal, dataTraining=dataTraining,
dataTesting=dataTesting, dataMissing=dataMissing)

app.run(debug=True)

```

DAFTAR RIWAYAT HIDUP
CURRICULUM VITAE

DATA PRIBADI

Nama Lengkap : Baby Andini
 Tempat, Tanggal Lahir : Bogor, 16 Agustus 2001
 Jenis Kelamin : Perempuan
 Agama : Islam
 Alamat : Komplek Tasbi 2 Blok VI No.96 LK X Medan
 E-mail : bebyandini16@gmail.com



KEMAMPUAN

Bahasa Pemrograman : Python, C++
 Bahasa Markup : HTML, PHP
 Bahasa Desain dan Framework : CSS, Bootstrap
 DBMS : MySQL
 Software : Microsoft Office, VSCode

RIWAYAT PENDIDIKAN

2019 - 2024 S-1 Ilmu Komputer Universitas Sumatera Utara
 2016 - 2019 SMA Negeri 4 Medan
 2013 - 2016 SMP Negeri 1 Medan
 2008 - 2013 SD Namira Islamic School

PENGALAMAN

Praktik Kerja Lapangan <i>Telkom Medan Centrum</i>	2022
Master of Ceremony Computer Science Anniversary 20	2021
Master of Ceremony Penerimaan Mahasiswa Baru S-1 Ilmu Komputer	2021
Panitia Penerimaan Mahasiswa Baru 2021 <i>Anggota Hubungan Masyarakat</i>	2020
Panitia Penerimaan Mahasiswa Baru 2020 <i>Anggota Hubungan Masyarakat</i>	2020
Panitia Computer Science Anniversary 18 <i>Anggota Hubungan Masyarakat</i>	2019
