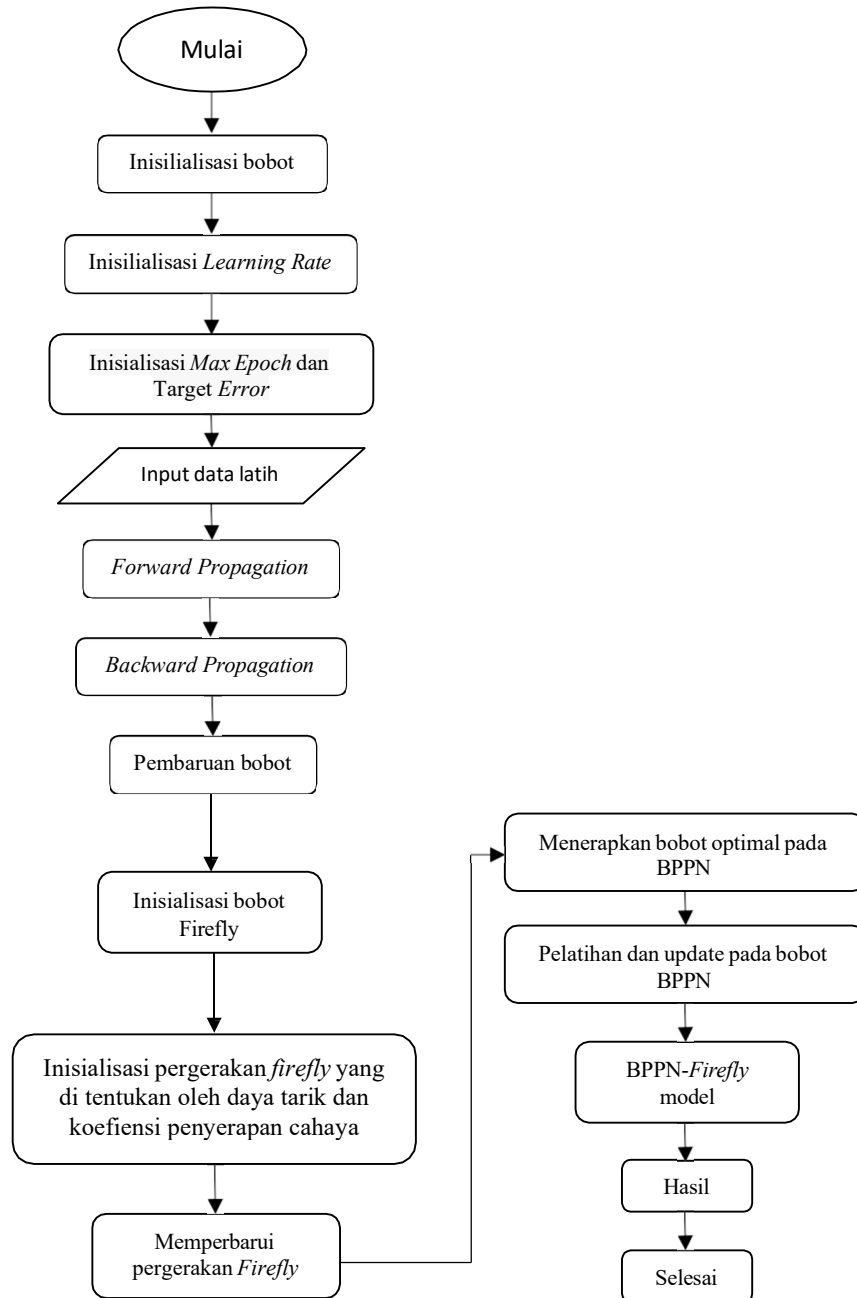


BAB III HASIL ANALISIS DAN PEMBAHASAN

3.1 Alur Flowchart

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



Gambar 2.3 Alur Flowchart

Keterangan flowchart penelitian :

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

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

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

3.2 Implementasi Program

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

3.2.1 Pembagian data latih dan data uji

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

```
Jumlah data latih: 875
```

```
Jumlah data uji: 219
```

Gambar 3.1 Pengujian Pembagian Data

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

3.2.2 Normalisasi Data

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

```
Data setelah normalisasi:
```

```
T
```

```
0 -1.380606
```

```
1 -1.380606
```

```
2 -1.380606
```

```
3 -1.380606
```

```
4 -1.380606
```

Gambar 3.2 Pengujian Normalisasi Data

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

3.2.3 K-Fold Cross Validation

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

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

3.2.4 Menginisialisasi bobot dan bias

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

Bobot dan bias sebelum optimasi:

```

w1: [[-2.86361331 -2.34181424 -2.89092042 2.84164395 -2.33495535 -2.31528158
2.30135127 2.32035404 -2.31225695 2.77778823 -2.33188357 -2.32480159
2.83669175 -2.88078735 -2.87001468 2.32582575 -2.32671242 2.33621678
2.85289087 2.32807234]]
b1: [[ 1.90484616 -1.2572877 1.91543419 2.4547279 -1.20254566 -1.08717706
-0.74584085 -0.83023791 -1.12169172 2.32813354 -1.22268329 -1.08139828
2.45029961 1.83642921 1.8241937 -0.81188226 -1.16864153 -0.8553951
2.44940802 -0.86003486]]
w2: [[ 0.42276463 -2.90593607 -3.36554013 1.40867829 1.62915742]
[ 1.85322222 -1.73506476 -1.50061174 -2.39184047 2.21534666]
[ 0.19058994 -2.93264122 -3.33083857 1.41602533 1.61661144]
[ -2.77494493 1.00417259 -1.24749747 3.7293457 -2.24773887]
[ 1.8457326 -1.82553978 -1.49351239 -2.35952133 2.22656645]
[ 1.8122843 -1.81160153 -1.49537341 -2.21116909 2.22584439]
[ -1.71456854 2.24253425 2.17133197 1.63029292 -2.16294916]
[ -1.7021034 2.26584242 2.22764912 1.59167943 -2.11838704]
[ 1.81584325 -1.83947386 -1.53256532 -2.27855608 2.21823795]
[ -2.7161819 1.0348223 -0.83821355 3.62660029 -2.29146412]
[ 1.8368583 -1.75407915 -1.4644055 -2.36265076 2.21372224]
[ 1.82249872 -1.86353959 -1.48864007 -2.22064631 2.24513673]
[ -2.72611766 0.98726785 -1.30334736 3.72484883 -2.26507925]
[ 0.7108349 -2.98223062 -3.30218378 1.38637892 1.66762696]
[ 0.70809071 -2.96572927 -3.26475461 1.37733343 1.66244087]
[ -1.64976061 2.26381874 2.21413933 1.59979957 -2.1667058 ]
[ 1.82731573 -1.74722719 -1.4757494 -2.26999496 2.208755 ]
[ -1.64383806 2.27582352 2.24360166 1.58026338 -2.14296091]
[ -2.81523721 1.00354887 -1.2906383 3.71331917 -2.30791472]
[ -1.63491033 2.2699391 2.24300018 1.58107936 -2.11744599]]
b2: [[-1.81747526 -1.5555643 -1.59481 1.75455237 -0.38613145]]
w3: [[ 5.58462122 7.15992323 4.79999652 -7.70019508 -6.33426554
-1.94607631 -1.5385076 -1.18711291 -0.83246226 -0.42296735
-0.48159571]
[ -1.2637718 -1.5391371 -2.30474735 -2.41157501 -6.78469444
-10.7879454 0.63497805 11.99362506 7.88445831 2.4917071
1.22111097]
[ 0.18105222 -0.08021844 -0.42930104 -0.75823288 -1.58259078
-1.99112683 -7.0676251 -12.44087749 0.02726384 9.48724584
13.42657188]
[ -15.61528542 -1.97993746 8.4901378 1.48364693 3.6854798
4.77860984 3.37699042 -0.38966453 0.87478298 0.16961723
-1.72466206]
[ 3.91009223 1.83142836 2.11591414 13.22393961 1.41055412
-10.39331952 -6.43653223 -1.87722671 -1.80701439 -1.53094224
-1.33357209]]
b3: [[ 3.29183269 0.11487854 -6.1075618 -3.54402872 2.83120635 4.52778601
3.21818927 -1.10394495 -1.05589407 -1.31382964 -1.81758335]]

```

Gambar 3.5 Pengujian Bobot dan bias sebelum optimasi

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

3.2.5 Proses Optimasi Menggunakan Firefly

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

```

Bobot dan bias setelah optimasi:
W1: [[ -2.87869027 -2.36913891 -2.91908965  2.87906877 -2.36199901 -2.34325107
  2.31439175  2.32986858 -2.33975267  2.81360582 -2.35920789 -2.35251079
  2.87381792 -2.92358002 -2.91034692  2.33816111 -2.35447847  2.34671755
  2.88953516  2.33665518]]
b1: [[ [ 1.78197313 -1.26716089  1.78769151  2.49802437 -1.20993936 -1.09133792
 -0.70707236 -0.78889845 -1.12709568  2.36996611 -1.23139647 -1.08460596
  2.49342333  1.70942916  1.70093988 -0.77061636 -1.175362  -0.81269905
  2.49201134 -0.81809877]]
W2: [[ [ 0.42166778 -3.01172697 -3.3138754  1.42139829  1.67460286]
 [ 1.88656116 -1.75993053 -1.4974536 -2.4316858  2.26633256]
 [ 0.18941784 -3.03908473 -3.27838505  1.42883074  1.66214923]
 [-2.81933965  0.94799678 -1.3090581  3.81960386 -2.25662416]
 [ 1.87866771 -1.85157694 -1.49016842 -2.39761688  2.27864129]
 [ 1.8441998 -1.84022079 -1.49160699 -2.24575765  2.27998618]
 [-1.72894712  2.25950076  2.09136517  1.64174069 -2.20007491]
 [-1.715549  2.28499097  2.14635808  1.60216032 -2.15456771]
 [ 1.84799059 -1.86730568 -1.52892738 -2.31434183  2.2715795 ]
 [-2.76054576  0.97973494 -0.89991328  3.71323249 -2.30235353]
 [ 1.86988679 -1.77968053 -1.46112937 -2.40148889  2.26527748]
 [ 1.8544552 -1.89229492 -1.48485331 -2.25486923  2.29961333]
 [-2.77045777  0.93113079 -1.36491416  3.81501355 -2.27399686]
 [ 0.71003883 -3.08830435 -3.25238693  1.39897217  1.71454647]
 [ 0.70736022 -3.07153685 -3.21541525  1.38987331  1.70946004]
 [-1.66339675  2.28261134  2.13297202  1.61041501 -2.20326621]
 [ 1.85994529 -1.77400323 -1.47228562 -2.30709314  2.26140161]
 [-1.65700396  2.29572122  2.16171833  1.5904009 -2.17901695]
 [-2.8599082  0.94740779 -1.35219695  3.8032374 -2.31707585]
 [-1.64803343  2.2898489  2.16120621  1.59122804 -2.15328861]]
b2: [[ [-1.82350539 -1.61981919 -1.65565828  1.76990123 -0.36445479]]
W3: [[ [ 5.68835605  7.44869807  5.00851447 -8.15436395 -6.47626797
 -1.94698621 -1.54066312 -1.18728516 -0.83313796 -0.42376964
 -0.48173638]
 [ -1.26377183 -1.54025419 -2.36760709 -2.41213232 -7.10032053
 -11.64852963  0.60796908  12.68138351  8.34907343  2.73058513
  1.09761284]
 [ 0.18105222 -0.08036003 -0.43439998 -0.75826738 -1.58431664
 -1.99171256 -7.47377039 -13.39554944 -0.06902606  9.61966746
 14.75884403]
 [-16.42884097 -2.18512441  9.11959358  1.39514814  3.79031647
 5.10539562  3.55925394 -0.56638504  0.82393533  0.3857095
 -1.84928661]
 [ 4.01072282  1.82540422  2.22616739  13.89589907  1.51830021
 -11.05587206 -6.75101028 -1.87819049 -1.81074744 -1.53351131
 -1.33384085]]
b3: [[ [ 3.39246259  0.10349255 -6.49270923 -3.72014976  2.93120974  4.85454677
 3.40018531 -1.28073225 -1.10695652 -1.0980266 -1.94227227]]

```

Gambar 3.6 Pengujian Bobot dan bias setelah optimasi

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

[253 rows x 3 columns]

Gambar 3.7 Pengujian Nilai Keatraktifan Firefly

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

3.2.6 Epoch dan Loss

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

```

Epoch dan loss
Epoch 0, Loss: 2.8549959115592904

Epoch dan loss
Epoch 100, Loss: 1.4394959053138086

Epoch dan loss
Epoch 200, Loss: 1.0955708190304654

Epoch dan loss
Epoch 300, Loss: 0.9067436189863152

Epoch dan loss
Epoch 400, Loss: 0.7937491013618913

```

Gambar 3.8 Pengujian Epoch dan Loss

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

3.2.7 Forward Propagation dan Backward Propagation

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

```

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

```

Gambar 3.9 Pengujian Forward Propagation

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

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


```

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

```

Gambar 3.10 Pengujian Backward Propagation

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

3.2.8 Perbandingan Data Uji Dan Data Klasifikasi

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

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

Gambar 3.11 Pengujian perbandingan Data

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

3.2.9 Evaluasi

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

Gambar 3.12 Hasil Pengujian Evaluasi

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

3.3 Hasil Pengujian

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

3.3.1 Parameter awal Backpropagation

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

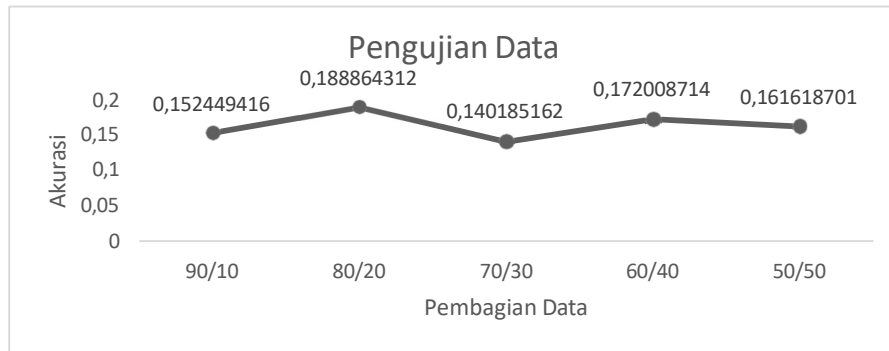
Tabel 3.1 Parameter awal Backpropagation

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

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

3.3.2 Rasio Data

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

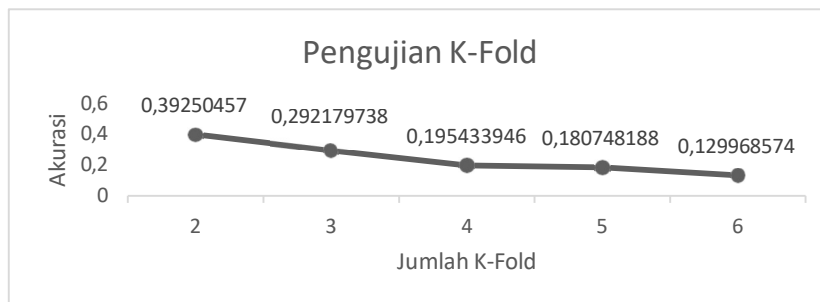


Gambar 3.13 Gambar Grafik Penguujian Data

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

3.3.3 K-Fold

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

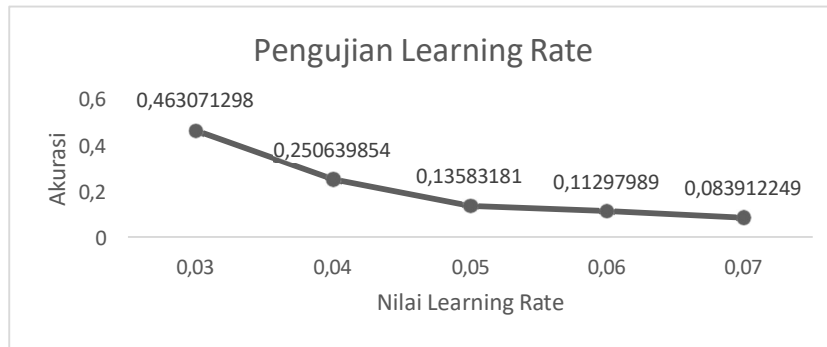


Gambar 3.14 Grafik Penguujian K-Fold

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

3.3.4 Learning Rate

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

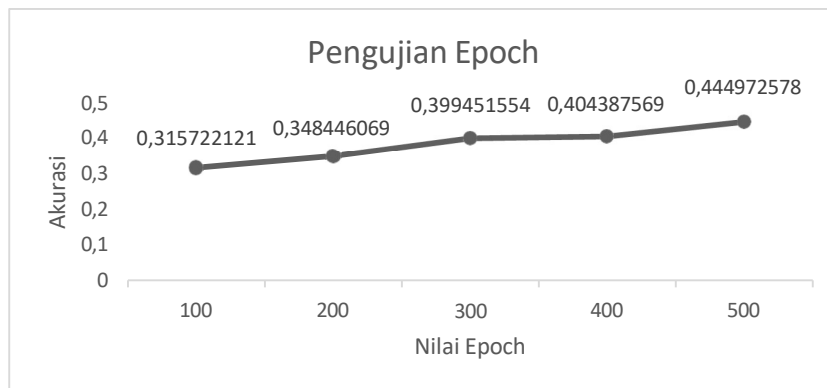


Gambar 3.15 Grafik Pengujian Learning Rate

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

3.3.5 Epoch

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

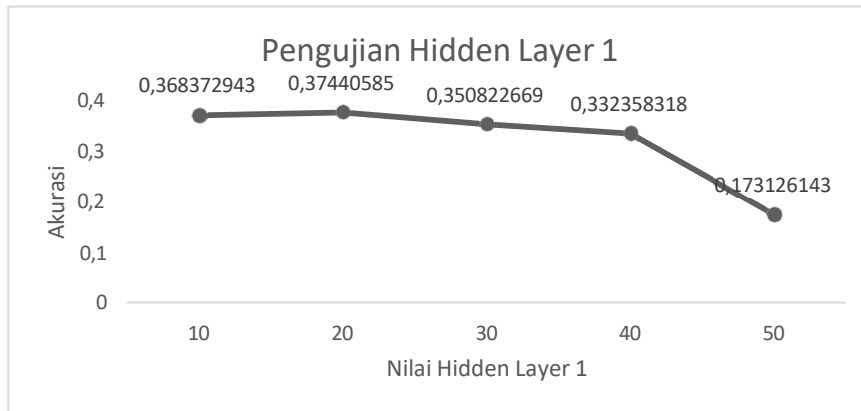


Gambar 3.16 Grafik Pengujian Epoch

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

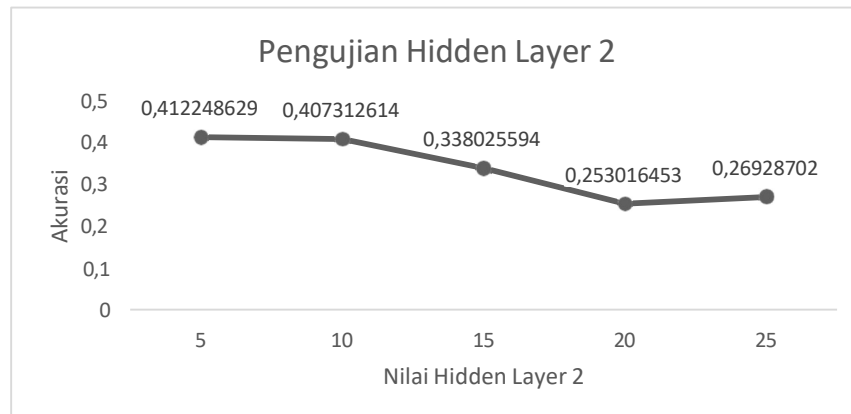
3.3.6 Hidden

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



Gambar 3.17 Grafik Pengujian Hidden Layer 1

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



Gambar 3.18 Grafik Pengujian Hidden Layer 2

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

3.3.7 Parameter Akhir Backpropagation

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

Tabel 3.2 Parameter Akhir Backpropagation

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

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

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

3.3.8 Parameter Awal Firefly Optimization

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

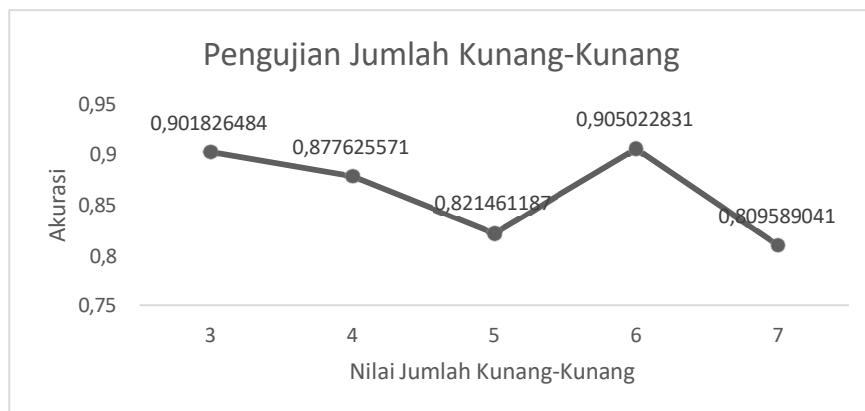
Tabel 3.3 Parameter Awal Firefly Optimization

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

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

3.3.9 Jumlah Firefly

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

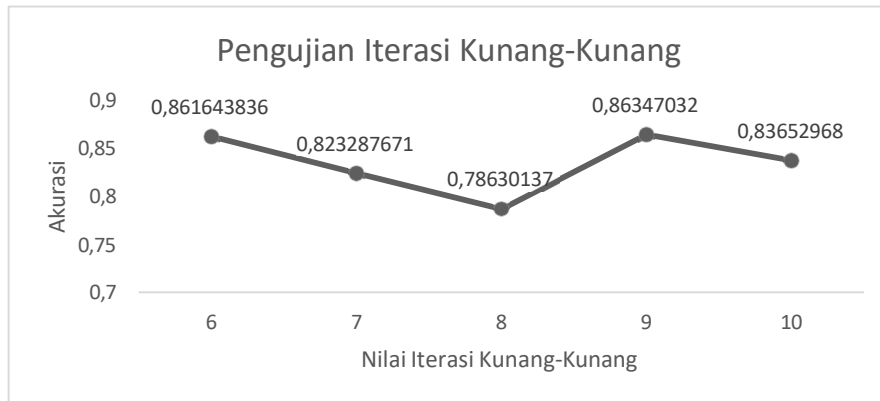


Gambar 3.19 Grafik Pengujian Jumlah Kunang-Kunang

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

3.3.10 Iterasi Firefly

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



Gambar 3.20 Grafik Pengujian Iterasi Kunang-Kunang

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

3.3.11 Parameter Akhir Firefly Optimization

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

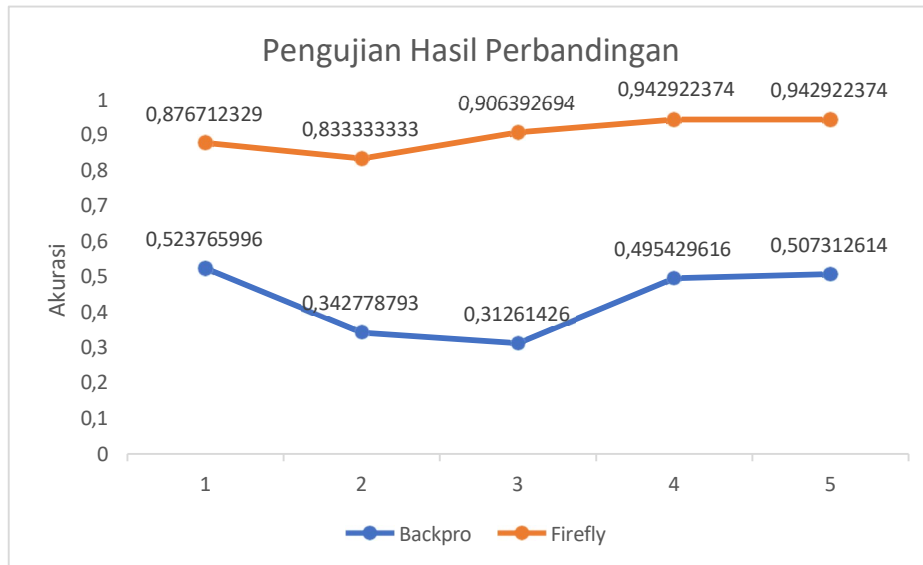
Tabel 3.4 Parameter Akhir Firefly Optimization

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

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

3.3.12 Hasil Perbandingan

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



Gambar 3.21 Hasil Perbandingan

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

3.4 Hasil

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