

BAB III

HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Program

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

3.1.1 Pembagian Data

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

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

Gambar 3.1 Jumlah Data

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

3.1.2 Normalisasi Data

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

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

Gambar 3.2 Normalisasi Data

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

3.1.3 Index K-Fold

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

```

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

```

Gambar 3.3 Train K-Fold

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

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

Gambar 3.4 Test K-Fold

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

3.1.4 Inisialisasi Bobot *Backpropagation*

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

```

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

```

Gambar 3.5 Bobot dan Bias *Backpropagation*

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

3.1.5 Bobot dan Bias Optimasi

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

```
Optimized Weights and Biases:
new_w1: [[-2.61359178 -4.64569694 -5.32675357 -1.8020603 0.17924981 2.80802257
 2.16599169 -4.71074133 -2.62904358 -2.7785822 -0.24774264 3.59135545
 2.82509833 -0.56886304 2.36921212 2.75548983 -4.09624396 -2.26039238
-3.31633996 3.4621162 1.93174138 1.62075455 2.31152526 -2.49592217
-0.24915995 5.7634762 -0.60167989 -2.29551749 -1.25810857 2.76052284
-0.80269471 2.41524985 -3.97590289 -2.87626682 -3.0782073 -3.04459014
2.75756043 0.01593445 3.05336095 3.59164856 -2.81159396 4.56481203
4.2867424 4.89694042 -2.36731379 3.4128215 4.04823112 3.43397569
-1.82616935 2.45931815 -2.68722392 3.16567149 -5.04286433 -0.80479122
-2.79487261 4.47077285 2.78623534 -1.58493471 -2.5467437 1.97965101
-1.33638997 3.97505511 -2.79355018 -1.25493549 4.38597167 -6.2659309
4.57906369 1.2192929 0.30563865 -4.61705971 -2.29667449 3.73911787
4.62231095 2.61377362 2.78252712 3.55540125 -4.12519123 -5.33603844]]
new_b1: [[-2.04230743 -5.29365144 5.33736262 -1.80377947 -2.49039437 0.42286542
-2.8668917 1.04983479 2.08909172 -3.7566014 -2.16699235 -4.56890299
1.95864001 -2.52367366 -3.20349733 -2.74001814 1.21066912 -2.10913922
-3.28563607 1.98566698 -2.13323079 -2.16278873 -2.38495868 -1.8377596
-2.53023331 -7.12332612 -1.97055274 -2.02734067 -1.02010516 0.93333215
-2.55993231 -2.36674846 -4.4401978 2.15087013 -2.2201707 -2.96215844
-3.5865209 -2.53735491 2.34778994 -4.04600486 0.10045585 -5.61444418
-2.15695553 4.23859899 -1.09976594 -4.17238169 1.45707273 2.58224791
-1.51258566 0.85900669 2.06897424 -4.38475159 2.08672354 -2.65560117
-2.86760033 3.85909502 1.29040736 -1.05633313 -1.68734498 -0.5122844
-0.78062931 1.26429678 2.20856162 -1.67720608 3.97556051 6.81531907
-5.27678202 -2.93325855 -1.93785993 5.23882091 -1.94473813 -4.12545177
4.78728667 -3.29183008 1.18827376 2.67168421 -0.66851014 -5.87685706]]
new_w2: [[-0.56856557 -0.49596179 0.1578991 ... -1.26537839 -0.17210606
-0.81687983]
[-0.25727811 0.57094075 -0.30119605 ... -2.67048606 0.55498618
0.580032 ]
[ 0.61006708 0.2123116 -2.91947061 ... -1.36469902 1.27625878
-0.44646102]
```

Gambar 3.6 Bobot dan Bias Optimasi

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

3.1.6 Perhitungan Kesalahan

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

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

Gambar 3.7 Nilai Kesalahan (loss)

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

3.1.7 Forward Propagation

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

```
Forward propagation for data index 0:
Input: [[-1.38060615]]
Layer 1 activation (a1): [[8.30320559e-01 6.78952072e-02 9.44770854e-02 7.20993607e-01
4.33436165e-02 1.01702407e-01 7.51611560e-01 6.33778324e-01
1.76086429e-01 1.42128977e-02 1.65805104e-03 1.31113129e-01
3.02710235e-04 1.72851955e-04 8.30781042e-01 4.30850890e-02
7.50604914e-01 9.90408079e-01 6.81226661e-01 7.41709263e-03
8.16700385e-01 1.39627428e-01 6.68382403e-01 9.16890522e-01
2.89214887e-03 1.03620815e-03 3.25898548e-02 2.98208110e-03
2.47373528e-02 6.52023525e-01 8.97731121e-01 1.00273215e-02
9.98616767e-01 1.31398422e-01 7.41737042e-02 1.62097168e-02
7.60399712e-04 9.92462807e-01 8.46057380e-02 9.99699097e-01
8.77856708e-01 2.70225808e-02 1.96886292e-02 4.86310447e-01
6.79094301e-01 4.25284608e-02 5.13022985e-04 8.51130190e-01
9.99907670e-01 3.60996061e-02 9.99340954e-01 9.99750757e-01
8.09311693e-01 9.57435922e-01 9.85558439e-01 5.95591103e-02
9.94664887e-01 8.26344739e-01 9.60212879e-01 8.83435248e-02
1.40828559e-02 8.70550300e-01 4.06887438e-05 9.9923557e-01
4.44219648e-03 1.66493757e-03 8.54548375e-02 9.97407446e-01
9.69814014e-01 8.47840323e-01 9.99081420e-01 9.50948141e-01
4.30263134e-04 6.81847488e-02 9.50869819e-01 8.04632232e-02
4.44940581e-02 1.55213702e-01]]
Layer 2 activation (a2): [[1.23066116e-21 2.27974351e-08 2.89950718e-07 1.31991614e-08
1.96715919e-04 3.87613315e-06 1.92542644e-04 2.29299155e-05
1.09670588e-10 1.97449266e-10 1.04834775e-06 5.71555752e-05
1.48410234e-24 1.59715444e-07 4.47358634e-06 6.26229467e-05
9.99992607e-01 1.04308741e-03 4.53105629e-16 8.11882100e-06
1.00000000e+00 3.64894495e-04 8.03726221e-19 1.46996542e-03
9.98381335e-01 2.73446423e-03 1.00000000e+00 9.34353377e-04
1.20204236e-06 9.99999999e-01 8.48614364e-07 9.99999771e-01
1.19769039e-07 3.97311675e-10 4.50474571e-04 3.71277536e-04
3.63989373e-04 1.00000000e+00 2.36420711e-10 1.45291809e-11
4.49336963e-21 1.28708180e-05 1.00781166e-04 6.74515760e-06
9.99999999e-01 5.88460026e-23 9.99999995e-01 8.07509542e-19
5.98416177e-09 3.90600093e-04]]
```

Gambar 3.8 *Forward Propagation*

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

3.1.8 Backward Propagation

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

```

Backward propagation for data index 0:
Updated W1: [[-1.43864819  0.1824416  1.51033818 -2.71575098  1.59180275 -0.10831773
-1.48981224 -2.32775094 -0.59895734  3.88439829  2.55351601  4.1815977
 3.01889126  3.25952105 -3.61996847  3.06683057 -1.59578309 -3.39361076
-1.79399646  2.30730633 -3.01579764  2.66280095 -1.34658445 -4.08204221
 3.63165893  2.97455515  3.54240387  3.12676418  1.63744504 -2.66535117
-4.69242441  2.49283706 -2.87265478 -0.24321871  1.98245399  1.7295836
 3.02354215 -2.49757747  2.22879949 -3.60757514 -2.6563975  1.31387137
 2.79392406 -1.01603703 -2.69059897  1.01555038  2.92866365 -2.13316721
-3.59653385  2.49420905 -3.08954942 -3.36532112 -1.72531469 -1.81686916
-2.03092493  1.6083958 -2.72485438 -1.89430298 -2.69004575  5.94462086
 2.37544193 -4.71025776  3.63077972 -3.48549919  2.17366733  2.56356832
 4.50576152 -3.29016756 -3.94439291 -2.44626475 -3.27993727 -1.7548241
 2.9669648  3.20979749 -4.01401254  1.32575815  4.19059926  3.40750349]]
Updated b1: [[-0.39830559 -2.36759978 -0.17497371 -2.79998626 -0.89663248 -2.32799431
-0.94961755 -2.66524592 -2.37001461  1.12353349 -2.87505311  3.88198572
-3.93453193 -4.16278739 -3.40657722  1.13354737 -1.1013066 -0.04804397
-1.71738714 -1.71104214 -2.66947787  1.85789053 -1.15822405 -3.23486292
-0.82896852 -2.76446135  1.50004275 -1.49531746 -1.41372596 -3.05185351
-4.30612408 -1.1507377  2.61594287 -2.2244393  0.212711 -1.71792814
-3.00658474  1.43216895  0.69574114  3.12778091 -1.69514996 -1.76974914
-0.05052034 -1.45751887 -2.96504354 -1.71204731 -3.5313457 -1.20157029
 4.32465417  0.15881414  3.05860663  3.65064933 -0.9364361  0.60486737
 1.41919069 -0.53881872  1.46614514 -1.05534612 -0.53028197  5.87314853
-0.96906442 -4.5971768 -5.09684156  4.66679224 -2.41117575 -2.85702313
 3.85024262  1.41009027 -1.97592654 -1.65956609  2.46346109  0.54186048
-3.65447339  1.81655207 -2.57886657 -0.60572038  2.7186815  3.01013859]]

```

Gambar 3.9 Backward Propagation

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

3.1.9 Data Kelas, Aktual, Klasifikasi

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

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

Gambar 3.10 Data Kelas, Aktual, Klasifikasi

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

3.1.10 Evaluasi

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

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

Gambar 3.11 Hasil Evaluasi

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

3.2 Hasil Pengujian

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

3.2.1 Parameter Awal *Backpropagation*

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

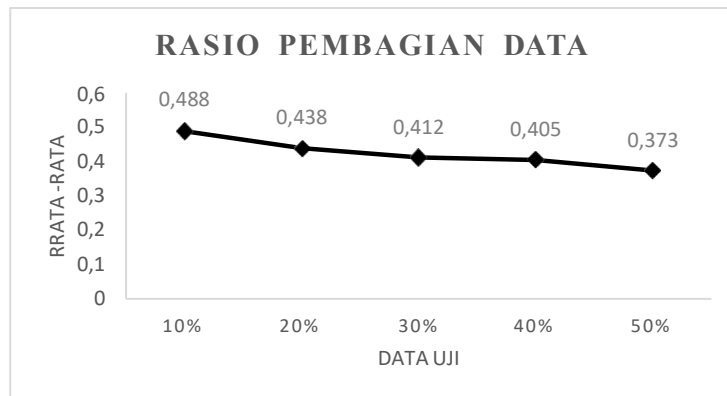
Tabel 3.1 Parameter Awal *Backpropagation*

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

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

3.2.2 Rasio Pembagian Data

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

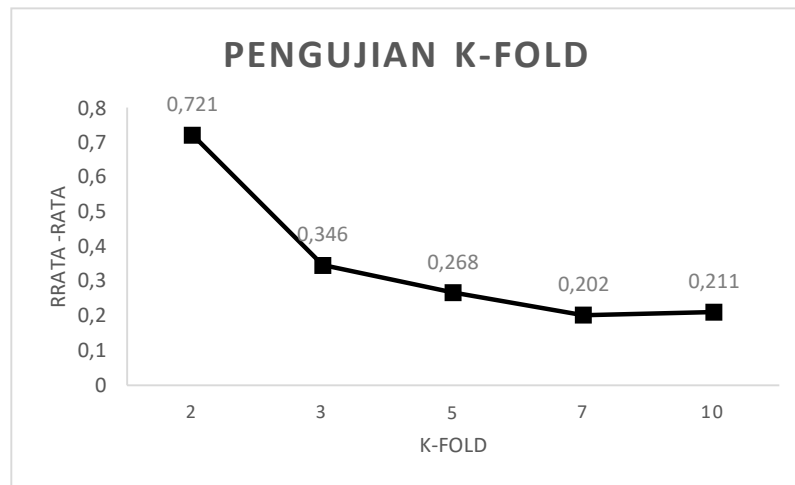


Gambar 3.12 Rasio Pembagian Data

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

3.2.3 K-Fold

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

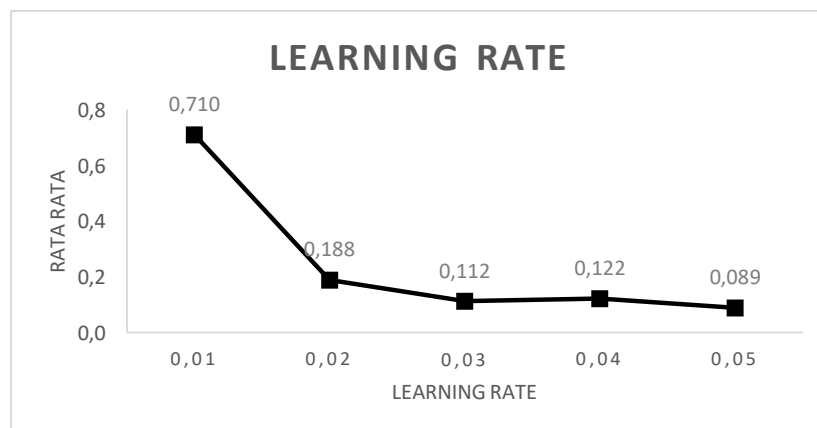


Gambar 3.13 K-Fold

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

3.2.4 Learning Rate

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

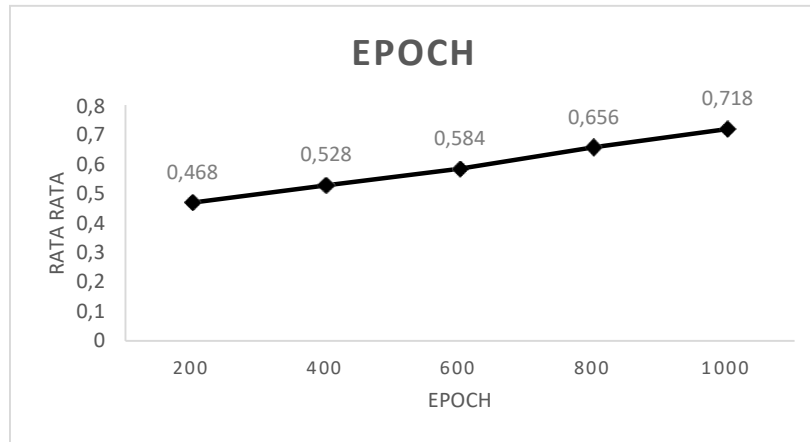


Gambar 3.14 Learning Rate

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

3.2.5 Epoch

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

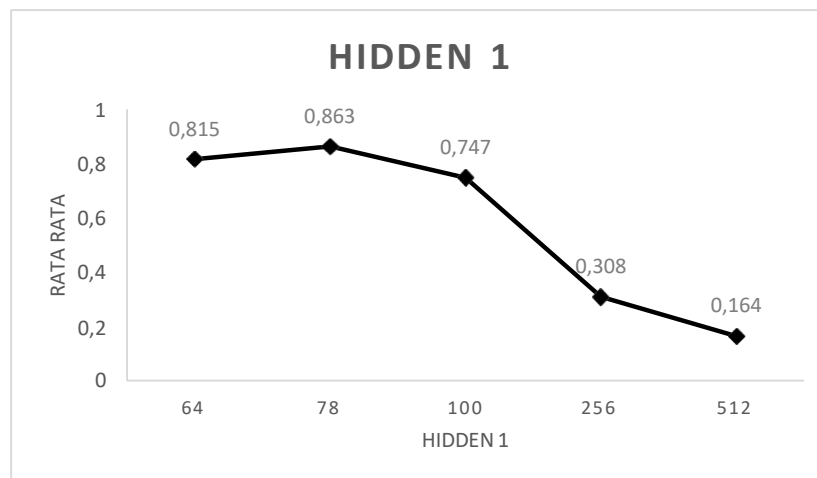


Gambar 3.15 Epoch

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

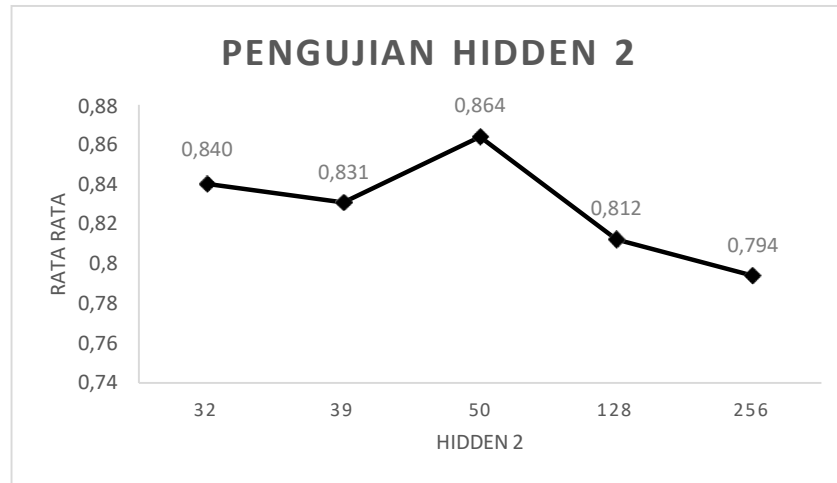
3.2.6 Hidden Layer 1 & 2

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



Gambar 3.16 Hidden Layer 1

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



Gambar 3.17 *Hidden Layer 2*

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

3.2.7 Parameter Akhir *Backpropagation*

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

Tabel 3.2 Parameter Akhir *Backpropagation*

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

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

3.2.8 Parameter Awal *Bee Colony Optimization*

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

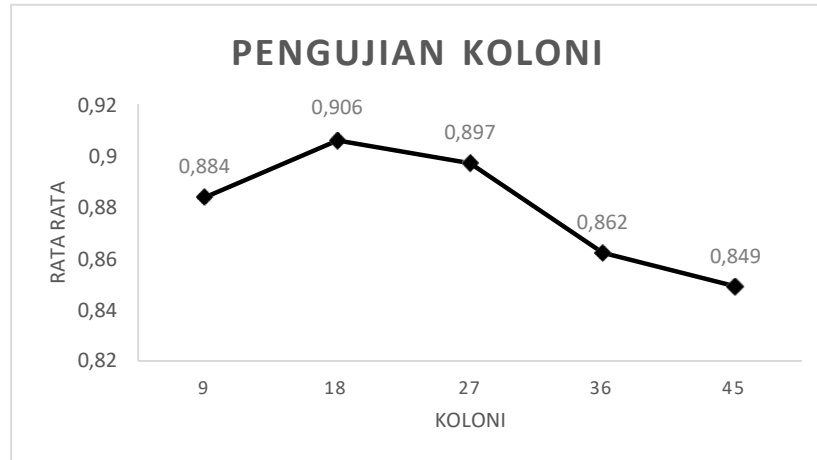
Tabel 3.3 Parameter Awal *Bee Colony Optimization*

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

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

3.2.9 Koloni

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

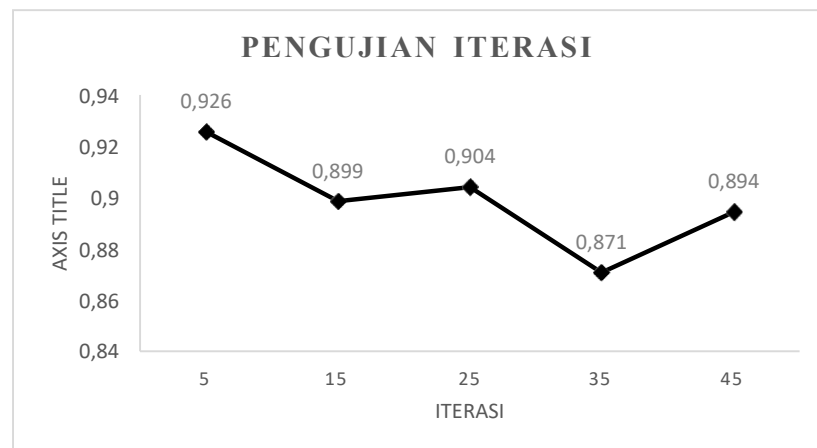


Gambar 3.18 Koloni

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

3.2.10 Iterasi

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



Gambar 3.19 Iterasi

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

3.2.11 Parameter Akhir *Bee Colony Optimization*

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

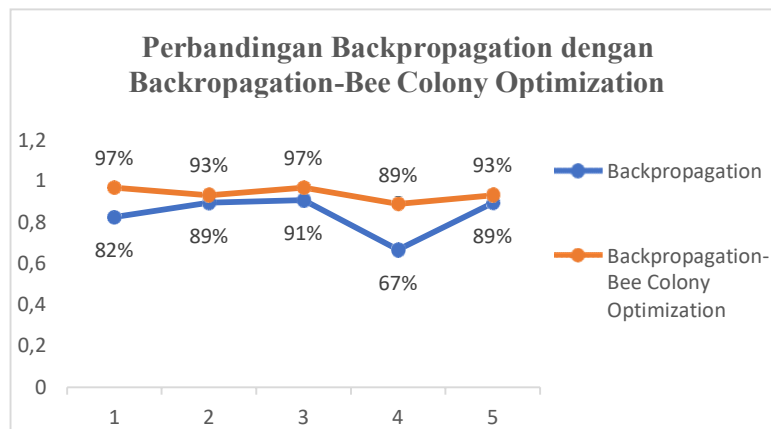
Tabel 3.4 Parameter Akhir *Bee Colony Optimization*

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

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

3.2.12 Perbandingan Akurasi

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



Gambar 3.20 Perbandingan Akurasi

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

3.3 Pembahasan

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

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

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