

KLASIFIKASI HAMA DAN PENYAKIT TANAMAN JAGUNG DENGAN MENGGUNAKAN NEURAL NETWORK BERBASIS ALGORITMA GENETIKA

Rais¹

email: rais.hojawa@gmail.com

¹Politeknik Harapan Bersama

Jalan Mataram No 9 Kota Tegal 52142, Indonesia Telp (0283) 352000

Abstrak

Kerusakan akibat hama penyakit tanaman jagung menyebabkan kerugian hasil panen bagi para petani. Pengelompokan hama penyakit tanaman jagung sangatlah penting bagi para petani agar dengan mudah petani mengenal hama penyakit tanaman jagung. Metode *Neural Network* digunakan sebagai *Artificial Intelligence* untuk mengklasifikasi hama penyakit tanaman jagung, sedangkan algoritma genetika digunakan untuk optimasi parameter *Neural Network* seperti jumlah *hidden layer* dan *learning rate* agar akurasi yang dihasilkan bisa lebih bagus. Selain itu, penelitian ini bertujuan untuk membandingkan antara metode kombinasi *Neural Network* dengan *Neural Network* yang dioptimasi dengan algoritma genetika. Metode evaluasi uji coba yang digunakan adalah metode *10 fold cross validation*. Hasil uji coba *10 fold cross validation* menunjukkan bahwa metode *Neural Network* yang optimasi parameternya menggunakan algoritma genetika menghasilkan rata-rata akurasi yang cukup tinggi yaitu 97,20%, lebih baik dari metode *Neural Network* yang menghasilkan rata-rata akurasi 96,60%.

Kata Kunci : *Algoritma genetika, Backpropagation Neural Network, Cross Validation*

1. Pendahuluan

Kerusakan akibat hama penyakit jagung pernah dilaporkan mencapai 26,5% [1]. Untuk mengatasi kehilangan tersebut perlu adanya usaha untuk menekan perkembangan hama penyakit tersebut. Sekitar 70 jenis serangga hama [2] dan 100 macam penyakit telah dilaporkan menyerang tanaman jagung [3].

Pengklasifikasian hama dan penyakit tanaman jagung salah satu cara untuk meningkatkan akurasi diagnosa menggunakan hasil-hasil dari gejala, tanda yang ada pada tanaman jagung.

Penelitian tentang hama penyakit jagung sudah pernah dilakukan sebelumnya. Edi Munanda dkk tahun 2013 mengatakan bahwa salah satu metode yang dapat digunakan untuk mendiagnosa penyakit tanaman jagung adalah sistem pakar dan *fuzzy MCDM* bahan penelitian adalah 9 penyakit dan 27 gejala sehingga melalui proses perhitungan *Fuzzy MCDM* mendapatkan hasil persentase penyakit bulai memiliki 15,208 % dan karat daun memiliki 14,791%. [4]

Menurut Deuk Hee Park, dkk (2012) mengatakan bahwa salah satu model klasifikasi yang digunakan dalam data mining adalah *Neural Network* [5].

Algoritma Genetika merupakan salah satu metode optimasi yang handal sehingga dapat digunakan untuk menentukan nilai parameter kontrol yang optimal untuk suatu proses tertentu [6]

Untuk mendapatkan nilai parameter pada *Neural Network*, GA yang merupakan salah satu algoritma terbaik untuk optimasi, digunakan sebagai model untuk mendapatkan nilai parameter NN yang terbaik

Tujuan dari penelitian ini adalah diperolehnya model untuk menerapkan *Genetic Algorithm* (GA) pada *Neural Network* (NN) sehingga dapat mengoptimasi dalam pemilihan nilai parameter awal NN yaitu *learning rate* dan *momentum* untuk mengklasifikasi hama penyakit tanaman jagung sehingga tingkat akurasi yang dihasilkan menjadi lebih baik lagi.

2. Metode Penelitian

Peneliti ini menggunakan jenis penelitian eksperimen dataset yang digunakan adalah hasil kuesioner di OPT Kecamatan Banjar Harjo Kabupaten Brebes dengan 25 variabel pertanyaan terkait dengan jawaban ya bernilai 1 (satu) dan tidak bernilai 0 (nol) dan menghasilkan 5 label atau hasil diagnosa.

Sampel penelitian ini adalah 100 petani tanaman jagung di OPT Kecamatan Banjar Harjo Kabupaten Brebes dengan variabel dan label perumpamaan seperti dibawah ini.

- X1 : Terdapat bekas gerogotan pada biji jagung yang baru ditanam
 X2 : Tanaman jagung mati
 X3 : Terdapat bekas gerogotan atau gigitan dari batang hingga bakal buah
 X4 : Terdapat lubang kecil pada daun jagung
 X5 : Tanaman mengalami gangguan pertumbuhan
 X6 : Terdapat bercak kecil, oval kebasahan
 X7 : Adanya bekas gigitan pada tongkol jagung
 X8 : Warna dan bentuk daun jagung tidak normal
 X9 : Adanya lubang pada biji jagung yang baru ditanam
 X10 : Terdapat lubang gorokan pada batang tanaman jagung
 X11 : Tanaman terlihat kerdil dan tidak berproduksi
 X12 : Banyak tongkol jagung yang rusak
 X13 : Terdapat bekas gigitan pada daun jagung
 X14 : Permukaan daun atas maupun bawah terdapat warna putih seperti tepung
 X15 : Warna daun menguning kemudian coklat memanjang kemerahan memanjang sejajar tulang daun jagung
 X16 : Terdapat larva di permukaan bawah daun
 X17 : Daun berserakan ditanah
 X18 : Tongkol berubah bentuk dan isi
 X19 : Daun berbentuk runcing, kecil, dan kaku
 X20 : Pertumbuhan batang terhambat
 X21 : Terdapat bercak-bercak berukuran kecil, berbentuk bulat sampai lonjong, dan berwarna kuning yang ditengahnya dikelilingi warna coklat
 X22 : Buah jagung tidak maksimal atau kecil
 X23 : Daun tanaman jagung mengalami klorosis(menguning)
 X24 : Tanaman jagung mengering

X25 : Terdapat bercak memanjang berbentuk elips, menjadi bercak nekrotik (kering) yang luas (hawar), berwarna hijau keabu-abuan atau coklat

Sedang output data dari penelitian ini adalah hama dan penyakit tanaman jagung dengan perumpamaan seperti dibawah ini :

Y1 : Bulai

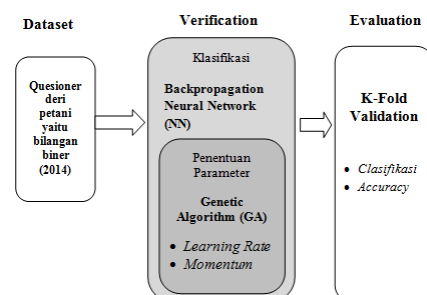
Y2 : Kutu Daun

Y3 : Ulat Grayak

Y4 : Bercak Daun

Y5 : Tikus

Metode yang diusulkan adalah penggunaan *Neural Network* (NN) *Backpropagation* untuk klasifikasi hama penyakit tanaman jagung. Untuk meningkatkan tingkat akurasi klasifikasi hama penyakit tanaman jagung maka menerapkan algoritma genetika (GA). Penerapan *Genetic Algorithm* (GA) pada penelitian ini adalah digunakan untuk penentuan dalam pencarian nilai parameter yaitu *learning rate* dan *momentum* pada *Neural Network* sehingga mendapatkan nilai parameter yang lebih optimal, sehingga dapat meningkatkan tingkat akurasi klasifikasi pada model yang akan diterapkan. Dataset yang digunakan untuk model adalah data hasil questioner dari petani yang berbentuk bilangan biner yaitu 1 dan 0 yang akan digunakan kedalam model *Neural Network*. Adapun untuk metode yang diusulkan seperti tampak pada gambar 1 dibawah ini.



Gambar 1. Metode yang diusulkan

3. Hasil dan Pembahasan

Untuk mendapatkan model yang sesuai dengan yang diharapkan, pada tahapan eksperimen dilakukan dengan melakukan dua tahapan, yaitu eksperimen dengan

menggunakan *Neural Network* (NN) dan eksperimen dengan menggunakan *Neural Network* (NN) berbasis algoritma genetika (GA)

Dalam menentukan jumlah *training cycles* dilakukan beberapa percobaan untuk mendapatkan jumlah *training cycles* terbaik, pada eksperimen ini digunakan nilai *training cycles* dari 100 – 1000 dengan *hidden layer* secara *default* dalam eksperimen ini menggunakan nilai parameter 0.3 (*default*) *learning rate* dan 0.2 (*default*) untuk *momentum*. Hasil yang didapatkan seperti pada tabel dibawah ini.

Tabel 1. Tabel penentuan jumlah *training cycles*

<i>training cycles</i>	<i>learning rate</i>	<i>momentum</i>	<i>accuracy</i>
100	0.3	0.2	95.20%+/-3.25%
200	0.3	0.2	95.80%+/-3.63%
300	0.3	0.2	95.80%+/-3.63%
400	0.3	0.2	95.80%+/-3.63%
500	0.3	0.2	96.20%+/-3.63%
600	0.3	0.2	96.20%+/-3.63%
700	0.3	0.2	96.20%+/-3.63%
800	0.3	0.2	96.55%+/-3.63%
900	0.3	0.2	96.20%+/-3.63%
1000	0.3	0.2	96.20%+/-3.63%

Dalam menentukan jumlah *neuron* pada *hidden layer* dilakukan beberapa percobaan untuk mendapatkan yang terbaik, pada eksperimen ini digunakan 800 *training cycles*. Secara *default* dalam eksperimen ini menggunakan nilai parameter 0.3 *learning rate* dan 0.2 untuk *momentum*. Hasil yang didapatkan dalam percobaan yang telah dilakukan seperti pada tabel dibawah.

Tabel 2. Tabel penentuan jumlah *neuron* pada *hidden layer*

<i>Neuron</i>	<i>training cycles</i>	<i>learning rate</i>	<i>momentum</i>	<i>accuracy</i>
1	800	0.3	0.2	75.00%+/- 7.11%
2	800	0.3	0.2	93.80%+/- 5.62%
3	800	0.3	0.2	95.00%+/- 3.61%
4	800	0.3	0.2	95.80%+/- 3.74%
5	800	0.3	0.2	96.20%+/- 3.40%
6	800	0.3	0.2	95.00%+/- 3.38%
7	800	0.3	0.2	95.40%+/- 3.58%
8	800	0.3	0.2	96.00%+/- 3.79%
9	800	0.3	0.2	95.60%+/- 3.56%
10	800	0.3	0.2	96.00%+/- 3.79%
11	800	0.3	0.2	95.60%+/- 3.56%
12	800	0.3	0.2	95.80%+/- 3.74%
13	800	0.3	0.2	95.09%+/- 3.23%
14	800	0.3	0.2	96.55%+/- 3.63%
15	800	0.3	0.2	95.82%+/- 3.95%
16	800	0.3	0.2	96.55%+/- 3.63%
17	800	0.3	0.2	96.36%+/- 3.80%
18	800	0.3	0.2	96.18%+/- 3.66%
19	800	0.3	0.2	96.20%+/- 3.63%
20	800	0.3	0.2	96.00%+/- 3.58%

Nilai *learning rate* ditentukan dengan cara melakukan dengan uji coba memasukkan nilai dengan *range* 0.1 sampai dengan 0.9. Nilai *training cycles* ditentukan sama seperti dari percobaan sebelumnya yaitu 800, sedangkan 0.2 digunakan untuk nilai *momentum*. Berikut ini adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai *learning rate* seperti pada tabel 3 dibawah ini:

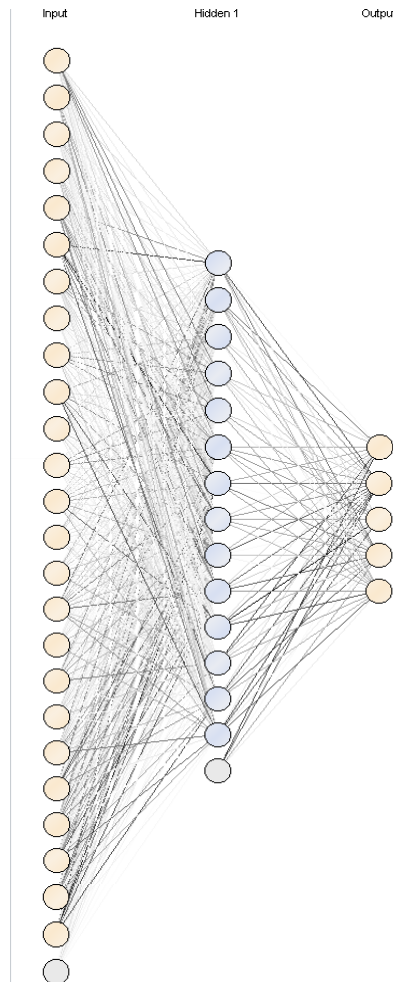
Tabel 3. Eksperimen Penentuan nilai *learning rate*

<i>training cycles</i>	<i>Momentum</i>	<i>learning rate</i>	<i>accuracy</i>
800	0.2	0.1	96.20%+/- 3.63%
800	0.2	0.2	96.20%+/- 3.63%
800	0.2	0.3	96.55%+/- 3.63%
800	0.2	0.4	96.36%+/- 3.80%
800	0.2	0.5	96.36%+/- 3.80%
800	0.2	0.6	96.36%+/- 3.80%
800	0.2	0.7	96.36%+/- 3.80%
800	0.2	0.8	96.00%+/- 3.79%
800	0.2	0.9	96.36%+/- 3.80%

Nilai *momentum* ditentukan dengan cara melakukan dengan uji coba memasukkan nilai dengan *range* 0.1 sampai dengan 0.9. Nilai *training cycles* 800 dan *learning rate* 0.3 dipilih berdasarkan percobaan sebelumnya. Berikut ini adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai *momentum* seperti pada tabel 4 berikut ini :

Tabel 4. Eksperimen Penentuan nilai *momentum*

<i>learning rate</i>	<i>momentum</i>	<i>Accuracy</i>
0.3	0.1	96.55%+/- 3.63%
0.3	0.2	96.55%+/- 3.63%
0.3	0.3	95.82%+/- 3.66%
0.3	0.4	96.36%+/- 3.80%
0.3	0.5	96.36%+/- 3.80%
0.3	0.6	96.36%+/- 3.80%
0.3	0.7	96.36%+/- 3.80%
0.3	0.8	96.36%+/- 3.80%
0.3	0.9	96.60%+/- 3.58%



Gambar 2 Arsitektur model *Neural Network*

Dengan menggunakan Fungsi aktivasi *Sigmoid* dalam menentukan pemberian bobot tiap neuron dimana range nilainya antara -1 s.d 1 dan *training cycles* = 800, maka nilai akhir yang dihasilkan setiap *weight* (bobot) pada *hidden layer* model algoritma *Neural Network* tampak seperti pada pada tabel 5 dibawah ini:

Tabel 5. nilai bobot *hidden layer*

Node	Hidden Layer (Sigmoid)													
(Sigmoid)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
X1	-1.233	-0.581	-1.158	3.626	4.019	3.254	-0.765	0.357	1.202	0.115	-0.510	-0.542	0.181	-0.308
X2	-1.205	-2.289	-0.143	-0.930	-0.639	-0.063	0.417	-2.287	-0.699	-0.000	-0.200	-0.359	-0.883	1.877
X3	-0.770	-0.440	-1.104	1.506	1.518	0.396	-0.602	0.391	1.193	0.453	-0.414	-0.515	0.453	-0.918
X4	0.638	-0.311	0.778	-0.175	-1.007	-1.500	1.299	-0.312	1.644	-0.692	1.345	-0.503	0.392	-0.074
X5	3.405	0.147	0.595	-1.767	-1.768	-0.148	2.200	3.396	-2.249	2.835	0.764	0.031	2.429	3.774
X6	-0.307	-0.524	-0.673	-0.053	-0.602	-0.909	7.287	0.090	-0.033	4.210	-0.241	-0.661	4.792	-0.008
X7	-1.074	-0.566	-1.306	3.469	3.955	3.780	-1.540	-0.176	1.457	-1.107	-1.038	-0.578	-0.596	-1.180
X8	-0.618	-0.470	-0.177	0.086	-0.440	-1.230	0.468	0.827	0.240	0.207	-0.052	-0.503	0.556	0.757
X9	-1.740	-0.571	-2.092	3.782	4.272	3.685	-1.099	0.195	1.299	0.006	-0.942	-0.528	0.069	-1.672
X10	0.696	-0.267	2.840	0.173	-1.247	-1.703	3.920	-0.512	3.927	-0.767	5.465	-0.461	0.711	-0.313
X11	1.805	1.050	0.653	-0.296	-0.019	0.477	-0.432	1.942	-0.365	1.524	0.116	1.219	-0.057	0.545
X12	-0.716	-0.378	-0.446	3.562	3.811	3.520	-0.657	-0.173	1.232	-0.811	-0.412	-0.564	-0.267	-0.529
X13	1.335	-0.312	1.705	-0.916	-2.094	-2.416	2.701	-0.764	2.411	-0.790	5.104	-0.521	0.617	-0.385
X14	1.777	1.008	1.213	-0.069	0.128	0.809	-0.857	0.561	0.682	-0.898	0.181	1.142	-1.248	0.046
X15	0.246	2.195	0.976	0.278	1.099	1.829	-0.319	1.965	0.827	1.091	0.160	2.638	-0.326	-0.022
X16	4.816	-1.039	0.030	-2.175	-1.838	-2.449	-4.572	-1.657	-3.355	-4.738	0.226	-1.262	-3.119	3.642
X17	0.861	-0.303	0.704	0.240	-0.572	-0.253	0.807	-0.125	2.191	-0.682	1.174	-0.400	0.261	1.515
X18	0.414	3.378	1.321	-0.338	0.425	2.687	-2.543	3.292	-0.226	2.265	0.188	0.912	-0.688	0.198
X19	0.323	1.812	0.478	0.321	0.868	1.325	-0.447	1.444	0.784	0.554	0.107	1.980	-0.050	-0.113
X20	3.013	0.983	0.245	-1.532	-0.617	1.214	0.570	3.315	-2.561	1.738	-1.191	0.943	2.240	4.355
X21	-4.218	-0.553	-0.459	-0.107	-0.681	-1.488	5.373	4.550	-0.183	3.465	-0.228	-0.621	1.463	0.727
X22	1.572	3.394	1.984	-0.625	0.549	3.010	-0.442	3.605	-0.889	2.988	0.214	3.803	-0.387	-0.023
X23	3.072	-0.572	-0.550	-1.181	-0.764	-0.587	2.781	2.954	-0.795	5.150	-0.768	-0.412	2.961	4.806
X24	4.245	0.600	0.344	-0.465	-0.107	0.775	-4.878	-3.321	-1.317	-3.235	-0.485	0.502	-3.390	2.983
X25	-5.211	-0.584	-0.749	-0.399	-0.890	-3.375	6.957	4.888	-0.803	5.540	-0.424	-0.579	5.712	0.422
Bias	-0.132	0.312	-0.030	-0.266	0.263	0.979	0.221	-0.172	-0.922	0.513	-0.182	0.358	0.058	0.198

Adapun nilai bobot pada *output layer* yang dihasilkan dapat dilihat pada tabel 6. dibawah ini:

Tabel 6. Nilai Bobot akhir untuk *output layer*

Node	Output (Sigmoid)				
	Bulai	Kutu Daun	Ulat Grayak	Bercak Daun	Tikus
1	-0.256	5.635	-1.158	-4.112	-1.663
2	4.440	-3.719	-0.751	-2.484	-2.265
3	0.800	-2.532	0.006	-2.831	-0.455
4	-1.997	-1.443	-1.841	-0.650	2.319
5	0.008	-1.467	-3.099	-0.423	3.026
6	2.547	-2.879	-3.122	-2.775	2.370
7	-2.602	-4.453	-0.117	4.072	-3.040
8	1.421	-3.180	-2.735	2.686	-2.269
9	-2.507	-5.342	-0.152	-1.830	-0.316
10	1.330	-5.069	-1.544	5.127	-3.207
11	-0.900	-4.763	8.077	-2.426	-5.614
12	6.050	-4.772	-0.874	-2.370	-2.983
13	-1.175	-5.668	-1.217	5.772	-2.661
14	-0.747	6.125	-5.209	0.644	-5.502
Threshold	-5.953	-5.296	0.178	-2.980	-0.651

Pada eksperimen ini, model arsitektur *Neural Network* yang digunakan adalah model yang didapatkan pada eksperimen sebelumnya dalam mendapatkan nilai *accuracy terbaik* yaitu seperti pada tabel 7 dibawah ini:

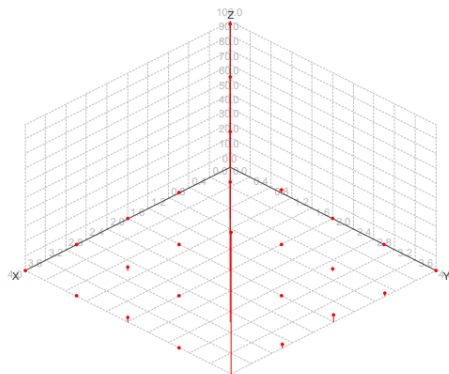
Tabel 7. Arsitektur Model NN

<i>Input (neuron)</i>	25
<i>Output (neuron)</i>	5
<i>Training Cycles</i>	800
<i>Hidden Layer / neurom</i>	1 /14 neuron
<i>Learning rate</i>	0.3
<i>Momentum</i>	0.9
<i>Accuracy</i>	96.60% +/- 3.58%

Tabel 8. Detail akurasi *testing Training Cycle* 800, *learning rate* 0,3 dan *momentum* 0,9

accuracy: 96.60% +/- 3.58% (mikro: 96.60%)						
	true Bulai	true Kutu Daun	true Ulat Grayak	true Bercak Daun	true Tikus	class precision
pred. Bulai	98	0	0	0	0	100.00%
pred. Kutu Daun	2	97	0	2	0	96.04%
pred. Ulat Grayak	0	0	95	0	3	96.94%
pred. Bercak Daun	0	1	0	96	0	98.97%
pred. Tikus	0	2	5	2	97	91.51%
class recall	98.00%	97.00%	95.00%	96.00%	97.00%	%

Adapun plotview untuk hasil *testing training cycle* 800, *learning rate* 0,3 dan *momentum* 0,9 adalah sebagai berikut :



Gambar 3. Confusion Matrix Plot View hasil *Neural Network*

Adapun parameter GA yang telah ditentukan adalah seperti pada tabel 9 dibawah ini, parameter ditentukan oleh peneliti sesuai dengan keinginan untuk mendapatkan model GA yang sesuai.

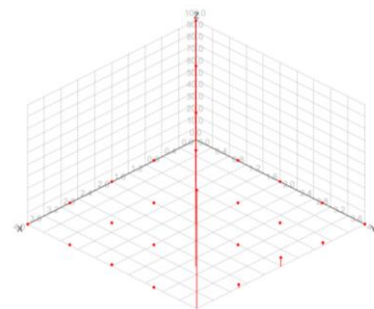
Tabel 9. Penentuan parameter *Genetic Algorithm* (GA)

Max generations	50
Population size	5
Mutation type	Gaussian_mutation
Selection type	Roulette wheel
Crossover prob	0.9
Neural Net.learning_rate	= 0.6266590432200198
Neural Net.momentum	= 0.8711847643359119
accuracy	= 97.20% +/- 1.83%

Tabel 10. Detail akurasi Kombinasi nilai parameter optimalisasi *Genetic Algorithm* (GA)

accuracy: 97.20% +/- 1.83% (mikro: 97.20%)						
	true Bulai	true Kutu Daun	true Ulat Grayak	true Bercak Daun	true Tikus	class precision
pred. Bulai	100	0	0	0	0	100.00%
pred. Kutu Daun	0	97	0	1	0	98.98%
pred. Ulat Grayak	0	0	93	0	1	98.94%
pred. Bercak Daun	0	1	0	97	0	98.98%
pred. Tikus	0	2	7	2	99	90.00%
class recall	100.00%	97.00%	93.00%	97.00%	99.00%	

Adapun plotview untuk hasil *testing* yang dihasilkan dari kombinasi dengan *Genetic Algorithm* (GA) antara lain menghasilkan *training cycle* 485, *learning rate* 0.6266590432200198 dan *momentum* 0.8711847643359119 adalah sebagai berikut :



Gambar 4. Confusion Matrix Plot View hasil kombinasi *Genetic Algorithm* (GA)

Berdasarkan dari analisa pengujian antara model *Neural Network* dengan *Neural Network* berbasis *Genetic Algorithm* maka dapat dirangkumkan hasilnya pada tabel 11 berikut:

Tabel 11. Hasil analisa dan komparasi NN dengan NN berbasis GA

	Neural Network (NN)	Neural Network (NN) – Genetic Algorithm (GA)
Learning rate	0.3	0.6266590432200198
Momentum	0.9	0.8711847643359119
Accuracy	96.60% +/- 3.58%	97.20% +/- 1.83%

Berdasarkan hasil analisa dalam eksperimen yang telah dilakukan terlihat bahwa dengan adanya optimalisasi penentuan parameter *Neural Network* (NN) dengan menggunakan algoritma genetika (GA) menjadikan nilai *accuracy* menjadi lebih baik, yaitu adanya peningkatan nilai *accuracy* sebesar **0.6%** dari **96.60%** menjadi **97.20%**. Dengan demikian dengan

optimalisasi algoritma genetika terhadap *Neural Network* terjadi sebuah peningkatan *accuracy* yaitu dengan adanya peningkatan nilai *accuracy*

4. Kesimpulan

Hasil penelitian menunjukkan Model yang dibentuk oleh *neural network* berbasis algoritma Genetika menghasilkan akurasi yang lebih baik dibandingkan dengan *neural network* dengan tanpa dioptimasi. Hasil optimasi ini sangat penting dalam penentuan nilai parameter yang paling optimal, sehingga menghasilkan tingkat akurasi yang tinggi.

Peningkatan dapat dilihat dari peningkatan nilai akurasi untuk model *Neural Network* dengan nilai *accuracy* yang didapatkan sebesar 96.60%, setelah dioptimasi nilai akurasi algoritma *Neural Network* berbasis Algoritma Genetika nilai *accuracy* yang didapatkan sebesar 97.20% lebih besar dibandingkan dengan sebelum di optimalisasi terbukti dengan adanya peningkatan nilai *accuracy* sebesar 0.6%. Berdasarkan perubahan yang terjadi pada nilai *accuracy* tersebut, dengan demikian dapat disimpulkan bahwa penerapan teknik optimasi dengan Algoritma Genetika dapat mempermudah dalam mencari nilai parameter optimal yang dapat meningkatkan nilai akurasi pada algoritma *Neural Network* sehingga dapat digunakan bagi para pengambil kebijakan untuk dapat menentukan pengambilan keputusan dalam melakukan klasifikasi hama penyakit tanaman jagung

5. Daftar Pustaka

- [1] Sudjono, M.S. 1988. Penyakit Jagung dan Pengendaliannya. Dalam Subandi, M. Syam, dan A. Widjono. 1988. Jagung. Puslitbangtan Bogor. Hal.205-241
- [2] Ortega, C.A. 1987. Insect pests of maize. A Guide for Field Identification. CIMMYT Mexico. Pp.106.
- [3] Shurtleff, M.C. 1980. Compendium of Corn Diseases. Second Edition. The American Phythological Society. USA. Pp.105.

- [4] Edi Munanda, N. P. (2013). Perancangan Sistem Pakar Untuk Mendiagnosa Penyakit Tanaman Jagung Menggunakan FUZZY MCDM. Jurnal Litek, 113-117.
- [5] Deuk Hee Park, H. K. (2012). A Literature Review and Classification of Recommender Systemes Research. Expert System with Applications, 10060.
- [6] Mehmet Bilgili, Besir Sahin, Abdulkadir Yasara & Erdogan Simseka, "Electric energy demands of Turkey in residential and industrial sectors", Renewable and Sustainable Energy Reviews 16, (2012), 404– 414