

MODEL PREDIKSI BERBASIS NEURAL NETWORK UNTUK PENGUJIAN PERANGKAT LUNAK METODE BLACK-BOX

Zulkifli

Jurusan Teknik Informatika STMIK Teknokrat Bandar Lampung
Jl. Z.A. Pagar Alam No 9-11 Kedaton Bandar Lampung Indonesia Telp (0721)-702022
e-mail : zulkiflist34@yahoo.co.id

Abstrak

Dalam Penelitian ini, pendekatan yang digunakan algoritma neural network untuk memprediksi akurasi pengujian perangkat lunak metode black-box. pengujian perangkat lunak metode black-box merupakan pendekatan pengujian dimana datates berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir, dan teknik yang digunakan yaitu equivalence partitioning.

Teknik dari penelitian ini, sistem informasi akademik menjadi test case, test case ini kemudian dilakukan pengujian black-box, dari pengujian black-box didapat dataset, kemudian dataset ini dilakukan pengukuran tingkat akurasi dalam hal memprediksi output realitas dan output prediction, selanjutnya tahapan terkahir dilakukan perhitungan error, RMSE dari output realitas dan output prediction.

Hasil dari penelitian ini didapat model tingkat akurasi prediksi, yaitu: 85%(4 hidden layer, epoch=900, learning rate=0,1) , 99%(4 hidden layer, epoch=1000, learning rate=0,1), dan 80%(5 hidden layer, epoch=1000, learning rate=0,1), dan model desain training neural network yang paling akurat adalah dengan 4 hidden layer, epoch=1000, learning rate=0,1 dengan tingkat akurasi 99%.

Kata Kunci : Nerural Network, Black-Box, Equivalence Partitioning

1. PENDAHULUAN

Pengujian adalah suatu proses pengekseskuan program yang bertujuan untuk menemukan kesalahan(Berard, 1994). Pengujian sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian dinyatakan sukses jika berhasil memperbaiki kesalahan tersebut. Selain itu, pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya. Sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi(Fournier, Cs, 2009).

Pengujian *black-box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya: fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses database eksternal, kesalahan performa, kesalahan inialisasi dan terminasi(B. B. Agarwad, 2010).

Belum adanya model prediksi tingkat akurasi berbasis *algoritma neural network* untuk pengujian perangkat lunak metode *black-box*, serta apakah *algoritma neural network* dapat diterapkan untuk memprediksi tingkat akurasi pengujian perangkat lunak metode *black-box*, dan ini merupakan ruang lingkup permasalahan dalam penelitian ini.

Pada penelitian ini, akan mengacu pada penelitian yang dilakukan oleh Vanmali pada tahun 2002, dalam penelitiannya Vanmali melakukan penelitian untuk pengujian perangkat lunak metode *white-box*, pendekatan yang digunakan adalah *algoritma neural network*. Dari latar belakang diatas peniliti akan membuat "model prediksi akurasi berbasis *algoritma neural network* pengujian perangkat lunak dengan metode *black-box*".

2. TINJAUAN PUSTAKA

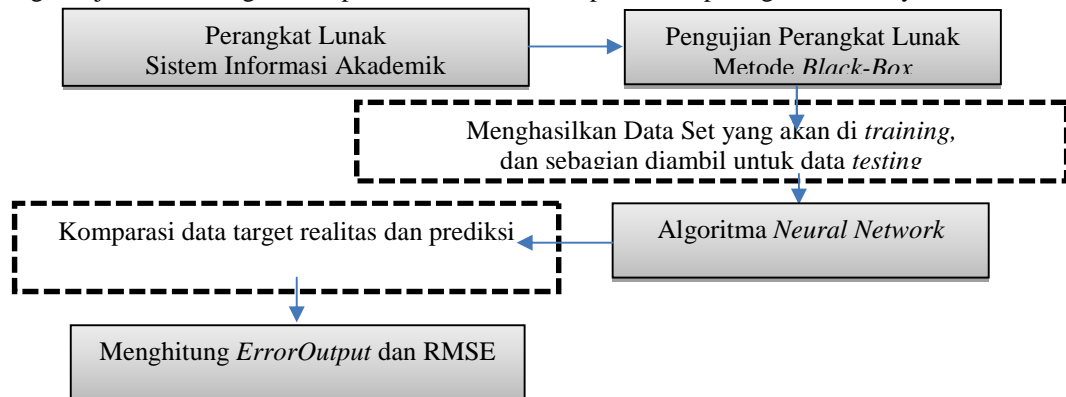
Penelitian yang berhubungan dengan penelitian ini adalah:

- Effective Black-Box Testing with Genetic Algorithms*(Mark Last, Cs, 2002)
Penelitian ini membahas bagaimana pengujian perangkat lunak dengan metode *black-box* secara efektif dilakukan oleh *algoritma genetika*. Pengujian kasus *black-box* diidentifikasi berdasarkan input yang di mapping kedalam output.kumpulan uji kasus diantaranya memiliki probabilitas yang tinggi. Algoritma Genetik mencoba mengekspos setiap *error* yang ditemukan.
- Black-Box Test Reduction Using Input-Output Analysis* (Patrick J, 2000)
Penelitian ini membahas tentang pengujian reduksi dengan menggunakan input-output perangkat lunak sebagai kategori yang akan dicari kesalahannya dengan metode *black-box*. Dan dalam penelitian ini memperkenalkan sebuah pendekatan untuk menguji pengurangan yang menggunakan otomatis input-output analisis untuk mengidentifikasi hubungan antara program input dan output.
- Using a Neural Network in the Software Testing Process*(Vanmali, 2002)
Dalam penelitian ini melakukan pengujian perangkat lunak dengan memastikan kesesuaian aplikasi untuk spesifikasinya, dimanana dibutuhkan tes "*oracle*" untuk menentukan apakah *test case* yang diberikan

mengeluarkan kesalahan atau tidak. Dan pada penelitian ini menggunakan metode *white-box* untuk pengujian perangkat lunak.

3. METODE PENELITIAN

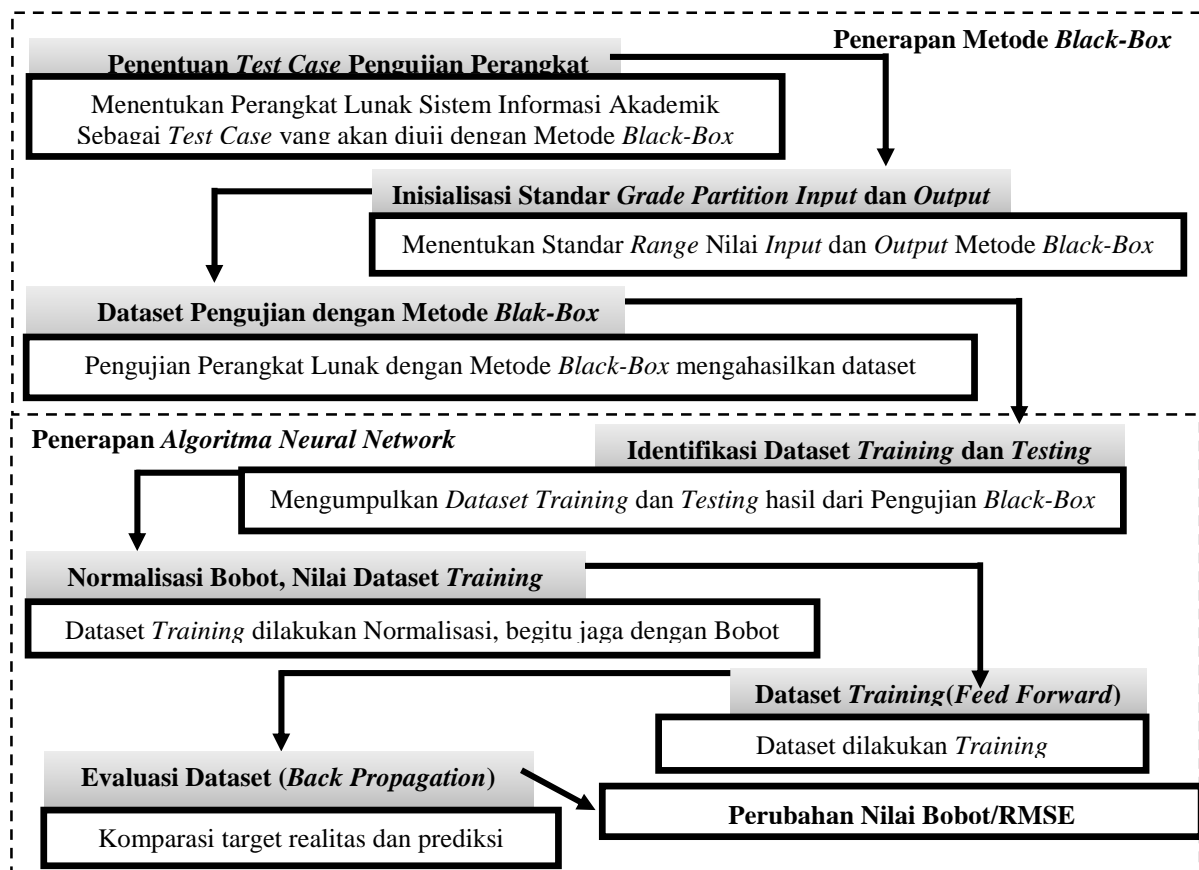
Penelitian ini menggunakan metode eksperimen, dengan tahapan sebagai berikut: perancangan penelitian, teknik analisis. Penelitian ini akan membuat suatu model prediksi tingkat akurasi berbasis *algoritma neural network* pengujian perangkat lunak metode *black-box*, yaitu: dengan terlebih dahulu *test case* akan uji dengan pengujian perangkat lunak metode *black-box*, hal ini dilakukan untuk mendapatkan dataset, kemudian dataset ini akan di *training* dan sebagian dijadikan data *testing* dalam penerapan ke *algoritma neural network*. Berikut yang menjadi metodologi dalam penelitian ini adalah dapat dilihat pada gambar 3.1., yaitu:



Gambar 3.1 Metodologi penelitian

3.1 PERANCANGAN PENELITIAN

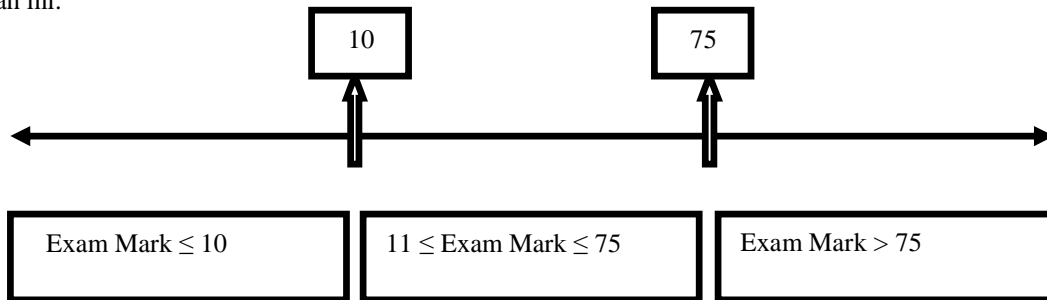
Tahapan ini dapat dilihat pada gambar 3.2, Pada tahapan pertama diawali dengan Penentuan *test case* Perangkat Lunak, *Inisialisasi Standart Grade Partition Input Dan Output*, Dokumentasi Pengujian Dengan Metode *Blak-Box*, Identifikasi *Dataset Training* dan *Testing*, Normalisasi Bobot, Nilai *Dataset Training*, *Dataset Training(Feed Forward)* Evaluasi *Dataset(Back Propagation)*, Perubahan Nilai Bobot.



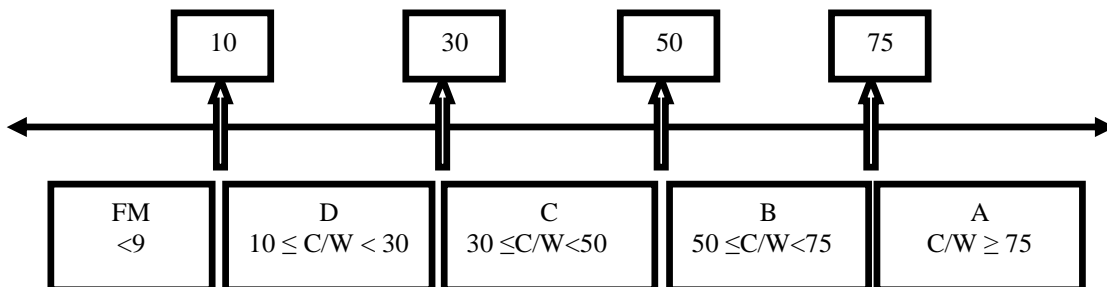
Gambar 3.2 Perancangan penelitian

3.1.1 Inisialisasi Standar Grade Partition Input dan Output

Pada penelitian ini untuk value dataset yang akan di implementasikan berdasarkan *partition error*, menurut Kelvin tahun 1998, untuk analisa *error* untuk *partition input* dan *output* dapat dilihat pada gambar dibawah ini:



Gambar 3.3 Equivalence partitions for Exam mark input



Gambar 3.4 Equivalence partitions for grade outputs

3.1.2 Dataset Pengujian dengan Metode *Blak-Box*

Berikutnya akan dilakukan dokumentasi pengujian perangkat lunak metode *Black-Box*, pada tahapan ini akan diungkapkan grade value yang ditemukan kesalahan pada setiap form dibagi menjadi lima model kesalahan, diantaranya: kesalahan pada **Fungsi**, **Struktur Data**, **Interface**, **Inisialisasi**, dan **Performance**. adapun *score grade value* kesalahan yang ditemukan pada setiap formnya, yaitu:

Form Ke-1 Pengujian Ke-1	
Test Case Value Input (Error)	Score Error
Input (Exam Mark Fungsi)	10
Total Error(as Calculated)	10
Partiton Tested (Of Exam Mark)	$10 \leq C/W < 30$
Expected Output	D

Dari pengujian perangkat lunak metode *black-box*, didapat dataset seperti yang terlihat pada tabel 3.1, dan dataset ini akan menjadi data yang akan diprediksi akurasinya dengan *algoritma neural network*:

Tabel 3.1 Dataset Hasil Pengujian Perangkat Lunak Metode *Black-Box*

No	Form	Pengujian Ke-	Tester	Jumlah Error	Equivalence Partition	Defect	Status Pengujian
1	1	1	1	1	10	1	Correct
2	1	2	1	2	20	1	Correct
3	1	3	1	3	30	1	Correct
4	1	4	1	4	40	1	Correct
198	40	3	1	0	0	0	InCorrect
199	40	4	1	0	0	0	InCorrect
200	40	5	1	0	0	0	InCorrect

3.1.3 Normalisasi Bobot, Nilai Dataset *Training*

Menurut Siang, JJ pada tahun 2005, rumus yang digunakan untuk proses pengubahan data testing asli menjadi data rangenya menjadi 0,1 dan 0,9 karena fungsi aktivasi yang digunakan adalah fungsi sigmoid yang nilai fungsi tidap pernah mencapai 0 ataupun 1(Siang, 2005). Rumusnya :

$$X' = \frac{0.8 * (X - a)}{b - a} + 0.1$$

Tabel 3.2 Dataset Hasil Normalisasi

Form	Pengujian Ke-	Tester	Jumlah Error	Equivalence Partition	Defect	Status Pengujian
1	1	0,12	0,12	0,3	0,12	Correct
1	2	0,12	0,14	0,5	0,12	Correct
1	3	0,12	0,16	0,7	0,12	Correct
1	4	0,12	0,18	0,9	0,12	Correct
1	5	0,12	0,1	0,1	0,1	InCorrect
40	1	0,1	0,1	0,1	0,1	Correct
40	2	0,1	0,1	0,1	0,1	Correct

Tabel 3.3 Tabel Nilai Unit Output Y_k

Y_k							
Z_{in1-1}	0,3394	Z_{in2-2}	0,5384	Z_{in3-1}	0,2104	Z_{in4-1}	0,6388
Z_{in1-2}	0,442	Z_{in2-2}	0,4574	Z_{in3-2}	0,0498	Z_{in4-2}	0,7774
Z_{in1-3}	0,5446	Z_{in2-3}	0,3764	Z_{in3-3}	-0,1108	Z_{in4-3}	0,9608
$Z_{in1-104}$	0,5446	$Z_{in2-104}$	0,3764	$Z_{in3-104}$	-0,1108	$Z_{in4-104}$	1,0616
$Z_{in1-105}$	0,442	$Z_{in2-105}$	0,4574	$Z_{in3-105}$	0,0498	$Z_{in4-105}$	0,8782

3.1.4 Perubahan Nilai Bobot

Perubahan Bobot nilai bobot *hidden layer* , nilainya dapat dilihat dibawah ini:

$$V_{ji}(\text{baru}) = V_{ji} + \Delta_{ji}$$

Tabel 3.4 Perubahan Nilai Bobot

V_{1-1}	0,0000276	V_{1-2}	0,0000144	V_{1-3}	0,000096	V_{1-4}	0,000012
V_{2-1}	0,0000276	V_{2-2}	0,0000168	V_{2-3}	0,00016	V_{2-4}	0,000012
V_{3-1}	0,0000276	V_{3-2}	0,0000192	V_{3-3}	0,000224	V_{3-4}	0,000012
V_{4-1}	0,0000276	V_{4-2}	0,0000216	V_{4-3}	0,000288	V_{4-4}	0,000012
V_{5-1}	0,0000276	V_{5-2}	0,000012	V_{5-3}	0,000032	V_{5-4}	0,00001
V_{6-1}	0,0000276	V_{6-2}	0,000012	V_{6-3}	0,000032	V_{6-4}	0,00001
V_{7-1}	0,0000276	V_{7-2}	0,000012	V_{7-3}	0,000032	V_{7-4}	0,00001

4. HASIL

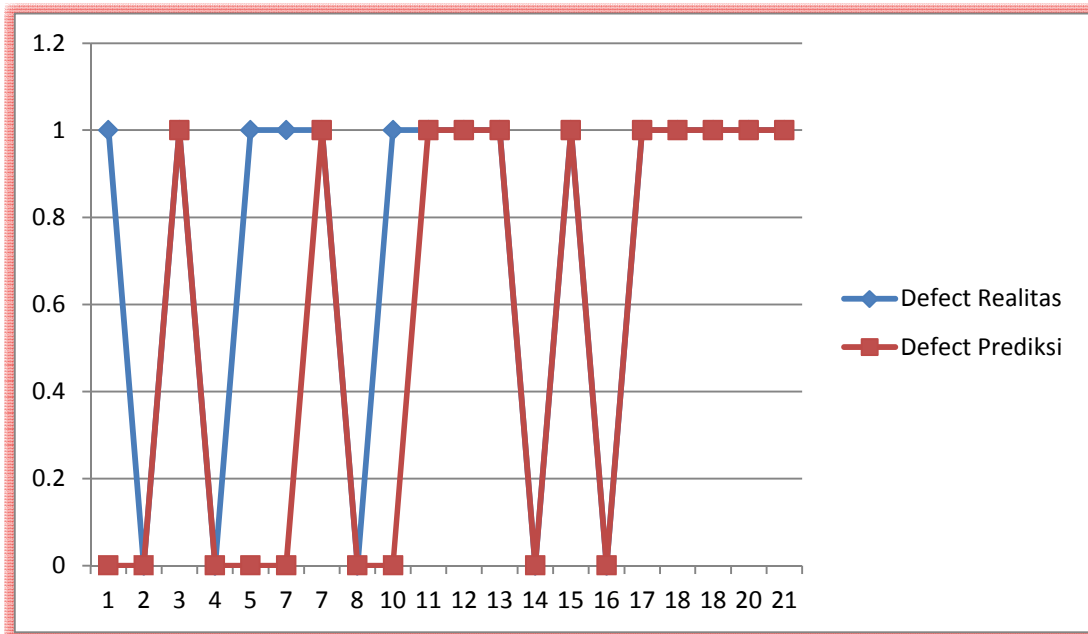
Setelah melewati tahap *training dataset*, yang dilakukan selanjutnya adalah tahapan *testing dataset*. Setelah testing dataset, *algoritma neural network* akan memprediksi tingkat akurasi antara nilai *output* dan *defect dataset* prediksi dengan nilai *output defect dataset* realitas.

Berikut tingkat *error* dari *output* dan *defect* prediksi dengan hasil *output* dan *defect* realitas, dapat dilihat pada tabel 4.1, dengan ketentuan banyaknya *Hidden Layer* 4 dan 5, *Learning Rate* =0,1.

Tabel 4.1 Hasil Prediksi Dataset dengan 4 *Hidden Layer*, pada *Epoch*=900, *Learning rate*=0,1 dan jumlah data testing=20 data(20% dari 100 jumlah data training).

Row	Form	Realitas		Prediksi		Match?	Probability
		Defect	Target	Defect	Output		
0	1	0	InCorrect	0,000812	InCorrect	OK	0,999188
8	2	0	InCorrect	0,000812	InCorrect	OK	0,999188
13	3	1	Correct	0,999903	Correct	OK	0,999903
18	4	0	InCorrect	0,000813	InCorrect	OK	0,999187
22	5	0	InCorrect	0,000813	InCorrect	OK	0,999187
31	7	0	InCorrect	0,000813	InCorrect	OK	0,999187
32	7	1	Correct	0,999908	Correct	OK	0,999908
37	8	0	InCorrect	0,000813	InCorrect	OK	0,999187
48	10	0	InCorrect	0,000812	InCorrect	OK	0,999188
51	11	1	Correct	0,999908	Correct	OK	0,999908
56	12	1	Correct	0,999908	Correct	OK	0,999908
62	13	1	Correct	0,999908	Correct	OK	0,999908
66	14	0	InCorrect	0,000813	InCorrect	OK	0,999187
72	15	1	Correct	0,999908	Correct	OK	0,999908
75	16	0	InCorrect	0,000855	InCorrect	OK	0,999145
80	17	1	Correct	0,999901	Correct	OK	0,999901
85	18	1	Correct	0,999904	Correct	OK	0,999904
89	18	1	Correct	0,999908	Correct	OK	0,999908
96	20	1	Correct	0,999908	Correct	OK	0,999908
100	21	1	Correct	0,999884	Correct	OK	0,999884

Grafik Hasil Prediksi *Dataset 4 Hidden Layer*, pada *Epoch*=900, *Learning rate*=0,1, dan jumlah data testing=20 data(20% dari 100 jumlah data training) dilihat pada gambar dibawah:

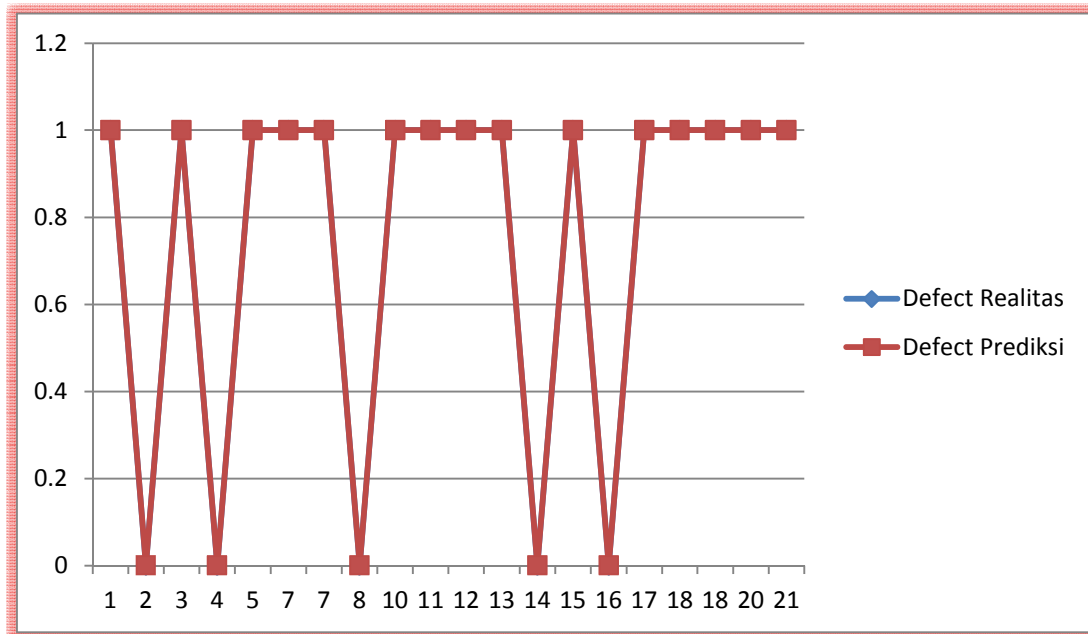


Gambar 4.1 Hasil Prediksi Dataset dengan 4 *Hidden Layer*, pada *Epoch*=900, *Learning rate*=0,1 dan jumlah data testing=20 data(20% dari 100 jumlah data training).

Tabel 4.2 Hasil Prediksi Dataset dengan 4 *Hidden Layer*, pada *Epoch*=1000, *Learning rate*=0,1 dan jumlah data testing=20 data(20% dari 100 jumlah data training).

Row	Form	Realitas		Prediksi		Match?	Probability
		Defect	Target	Defect	Output		
0	1	1	Correct	0,999891	Correct	OK	0,999891
8	2	0	InCorrect	0,000856	InCorrect	OK	0,999144
13	3	1	Correct	0,999885	Correct	OK	0,999885
18	4	0	InCorrect	0,000855	InCorrect	OK	0,999145
22	5	1	Correct	0,999885	Correct	OK	0,999885
31	7	1	Correct	0,999890	Correct	OK	0,99989
32	7	1	Correct	0,999891	Correct	OK	0,999891
37	8	0	InCorrect	0,000855	InCorrect	OK	0,999145
48	10	1	Correct	0,999891	Correct	OK	0,999891
51	11	1	Correct	0,999882	Correct	OK	0,999882
56	12	1	Correct	0,999890	Correct	OK	0,99989
62	13	1	Correct	0,999891	Correct	OK	0,999891
66	14	0	InCorrect	0,000853	InCorrect	OK	0,999147
72	15	1	Correct	0,999891	Correct	OK	0,999891
75	16	0	InCorrect	0,000855	InCorrect	OK	0,999145
80	17	1	Correct	0,999884	Correct	OK	0,999884
85	18	1	Correct	0,999890	Correct	OK	0,99989
89	18	1	Correct	0,999881	Correct	OK	0,999881
96	20	1	Correct	0,999889	Correct	OK	0,999889
100	21	1	Correct	0,999890	Correct	OK	0,99989

Grafik Hasil Prediksi *Dataset* dengan 4 *Hidden Layer*, pada *Epoch*=1000, *Learning rate*=0,1 dan jumlah data testing=20 data(20% dari 100 jumlah data training), dilihat pada gambar dibawah:

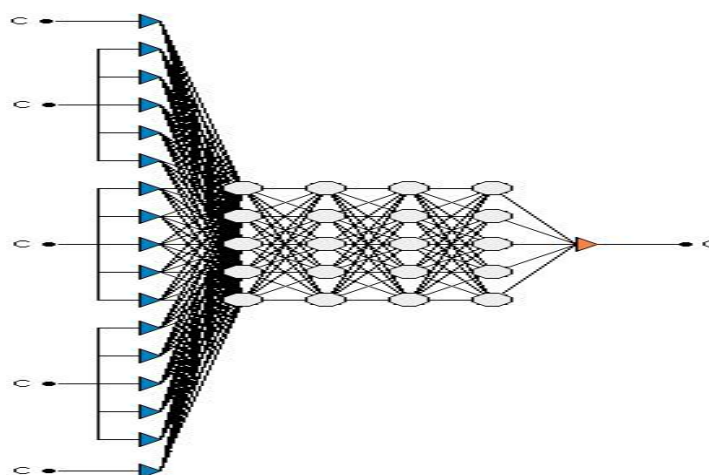


Gambar 4.4 Hasil Prediksi Dataset dengan 4 *Hidden Layer*, pada *Epoch*=1000, *Learning rate*=0,1 dan jumlah data testing=20 data(20% dari 100 jumlah data training).

4.1 Hasil Prediksi Tingkat Akurasi berbasis *Algoritma Neural Network* Pengujian Perangkat Lunak dengan Metode *Black-Box*

Pengujian Perangkat lunak Metode *Black-Box* berbasis algoritma *neural network* didapat akurasi sebesar 85% dari 100%, dan ini sudah sangat akurat.

Jumlah <i>HiddenLayer</i>	<i>Epochs</i>	<i>Learning rate</i>	Confusion Matrix (Accuracy)	AUC
4	900	0,1	85.00%	0.850 +/- 0.071
4	1000	0,1	99.00%	0.976 +/- 0.071
5	1000	0,1	80.00%	0.800 +/- 0.071



Gambar 4.5 Arsitektur Design Nerual Network dengan 4 hidden layer, 5 input layer, dan 1 output layer.

5.1 KESIMPULAN

Dari hasil penelitian yang disimpulkan sebagai berikut:

Terbentuknya model prediksi tingkat akurasi berbasis *algoritma neural network* untuk pengujian perangkat lunak metode *black-box*, dan bisa diterapkan untuk mengetahui prediksi tingkat akurasi pengujian perangkat

lunak metode *black-box* dengan akurasi sangat akurat, karena prediksi menunjukkan nilai rata-rata diatas 80% yaitu: 85%(4 hidden layer, epoch=900, learning rate=0,1) , 99%(4 hidden layer, epoch=1000, learning rate=0,1), dan 80%(5 hidden layer, epoch=1000, learning rate=0,1), dan model desain training neural network yang paling akurat adalah dengan 4 hidden layer, epoch=1000, learning rate=0,1) dengan tingkat akurasi 99%.

5.2. Saran

Adapun saran-saran yang dapat penulis berikan adalah sebagai berikut:

1. Model prediksi tingkat akurasi pengujian perangkat lunak metode *black-box* selain bisa diprediksi dengan algoritma *neural network*, kedepan ada yang melakukan penelitian pendekatan algoritma yang lainnya.
2. Ada model prediksi *class level metrics* dataset untuk mengetahui kesalahan perangkat lunak dengan algoritma *neural network*.
3. Komparasi model prediksi *class level metrics* dataset untuk mengetahui kesalahan perangkat lunak dengan algoritma *neural network* dan algoritma *genetic*.

DAFTAR PUSTAKA

- Albert Endres, Cs. (2003). Handbook software and System Engineering, Empirical Observations, Laws and Theories.
- B. B. Agarwad, C. (2010). Software Engineering & Testing. Boston.
- Beizer, B. (1990). Software Testing Techniques.
- Berard, C. (1994). Issues in the Testing of Object-Oriented Software,.
- Fournier, Cs. (2009). Essential Software Testing A Use-Case Approach.
- Hendrowati. (2003). Perancangan pengujian perangkat lunak Berorientasi obyek: berbasis status (state-based Testing). Jurnal Universitas Paramadina , Vol. 2 No. 2.
- Hetzel, W. C. (1988). The Complete Guide to Software Testing, 2nd ed.
- J.M, Z. (1992). Introduction to artificial neural systems.
- Mark Last, Cs. (2002). Effective Black-Box Testing with Genetic Algorithms. ACM
- Myers, G. J. (1979). The art of software testing. New York: New York :
- Myers, G. (1979). The Art of Software Testing.
- Oscar Pastor, Cs. (2007). Model-Driven Architecture in Practice, A Software Production Environment Based on Conceptual Modeling.
- Patrick J, C. (2000). Black-Box Test Reduction Using Input-Output Analysis. ACM .
- Perry, W. E. (1990). A standard for testing application software. 1990.
- S, H. (1999:p20). Neural networks – A comprehensive Foundation.
- Siang. (2005). Jaringan Syaraf Tiruan dan Pemogramannya.
- Sri Kusumadewi, C. (2010). Neuro-Fuzzy. Yogyakarta.
- Sthamer, C. (1995). The Automatic Generation of Software Test Data Using Genetic Algorithms.
- William E. Howden, C. (1987). Functional program Testing and Analysis.
- Wellesley, Mass.: Wellesley, Mass. : QED Information Sciences, 1988. ISBN: 0894352423. Physical description: ix, 280 p. : ill ; 24 cm.
- Wiley, c1979. ISBN: 0471043281 Physical description: xi, 177 p. : ill. ; 24 cm.