

Model Prediksi Berbasis *Neural Network* untuk Pengujian Perangkat Lunak Metode *Black-Box*

Zulkifli

Program Studi Teknik Informatika

STMIK Teknokrat

Bandar Lampung, Indonesia

zulkifli34@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, Pengujian Perangkat Lunak metode black-box, equivalence partitioning*

I. PENDAHULUAN

1.1 Latar Belakang

Pengujian adalah suatu proses pengekseskusan program yang bertujuan untuk menemukan kesalahan[1]. 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[2].

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 inisialisasi dan terminasi[3].

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*.

1.2 Tujuan

Membuat model prediksi untuk mengukur tingkat akurasi berbasis algoritma *neural network* untuk pengujian perangkat lunak metode *black-box*.

1.3 Batasan Masalah

Batasan Masalah pada penelitian ini adalah pada pembuatan model prediksi untuk mengetahui tingkat akurasi pada pengujian perangkat lunak metode *black-box*, dengan cara dataset hasil pengujian perangkat lunak metode *black-box* dilakukan *training* dan *testing* dengan algoritma *neural network*.

II. PENGUJIAN PERANGKAT LUNAK

Pengujian Perangkat lunak merupakan proses eksekusi suatu program atau sistem dengan maksud menemukan atau, melibatkan setiap kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan suatu program atau sistem dan menentukan bahwa itu memenuhi hasil yang dibutuhkan perusahaan[4].

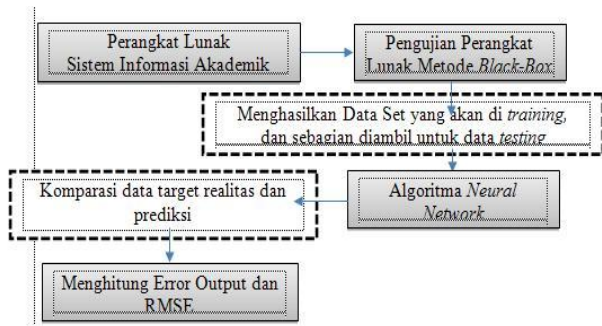
2.1 Algoritma *Neural Network*

Algoritma *Neural Network* sebagai sistem saraf tiruan atau jaringan saraf tiruan adalah sistem selular fisik yang dapat memperoleh, menyimpan dan menggunakan pengetahuan yang didapatkan dari pengalaman[5].

III. 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*

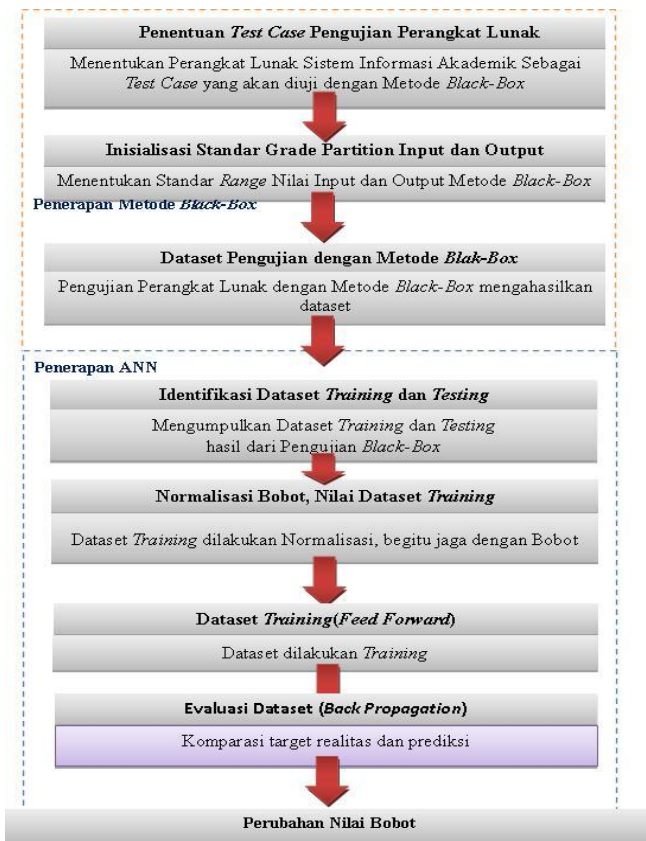
network. Berikut yang menjadi metodologi dalam penelitian ini adalah dapat dilihat pada gambar 3.1., yaitu:



Gambar 3.1. Metodologi penelitian model prediksi tingkat akurasi berbasis algoritma neural network penguujian perangkat lunak metode Blak-Box.

3.1 Perancangan Penelitian

Dalam penelitian ini akan dilakukan beberapa tahapan, dimana tahapan ini dapat dilihat pada gambar 3.2.

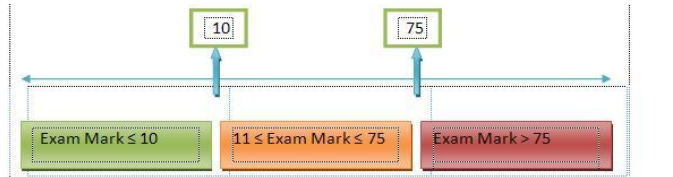


Gambar 3.2. Tahapan-tahapan penguujian perangkat lunak metode Black-Box berbasis Algoritma Neural Network.

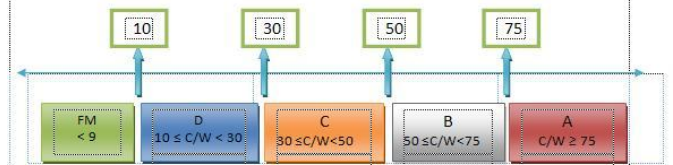
Tahapan pertama diawali dengan Penentuan Test Case Perangkat Lunak, Inisialisasi Standar Grade Partition Input Dan Output, Dokumentasi Penguujian Dengan Metode Blak-Box, Identifikasi Dataset Training dan Testing, Normalisasi Bobot, Nilai Dataset Training, Dataset Training(Feed

Forward) Evaluasi Dataset(Back Propagation), Perubahan Nilai Bobot.

3.1.1 Inisialisasi Standar Grade Partition Input dan Output
 Penentuan *value dataset* yang akan di implementasikan berdasarkan *partition error*, menurut Kelvin tahun 1998, untuk analisa *error partion 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 Penguujian dengan Metode Blak-Box

Berikutnya akan dilakukan dokumentasi penguujian 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:

TABEL III.1 PENGUJIAN FORM KE-1 PENGUJIAN 1

Form Ke-1 Penguujian 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 penguujian perangkat lunak metode *black-box*, didapat *dataset* seperti yang terlihat pada tabel 3.1, dan *dataset* ini akan menjadi data yang akan diprediksi akurasi dengan *algoritma neural network*.

Dari penguujian perangkat lunak metode *Black-Box*, didapat *dataset* seperti yang terlihat pada tabel 3.2, dan *dataset* ini akan menjadi data yang akan diprediksi akurasi dengan *Algoritma neural network*:

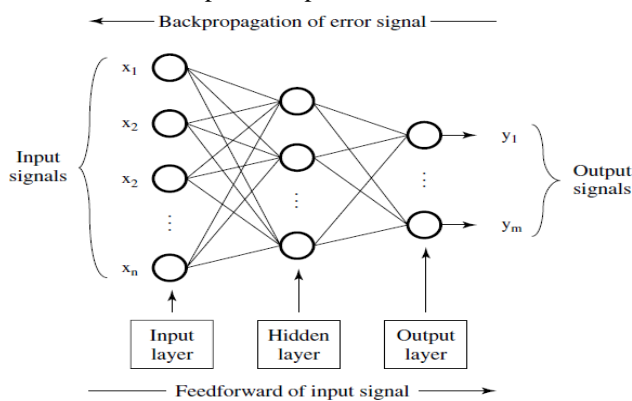
TABEL III.2. DATASET HASIL PENGUJIAN PERANGKAT LUNAK METODE BLACKBOX

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
5	1	5	1	0	0	0	InCorrect
6	2	1	1	0	0	0	InCorrect
7	2	2	1	0	0	0	InCorrect
8	2	3	1	0	0	0	InCorrect
9	2	4	1	0	0	0	InCorrect
...							
199	40	4	1	0	0	0	InCorrect
200	40	5	1	0	0	0	InCorrect

Jumlah data 200 set.

3.1.3 Penerapan Algoritma Neural Network

Penelitian ini akan membuat model prediksi tingkat akurasi berbasis Algoritma *neural network* metode *Black-Box*, sehingga secara tidak langsung peneliti dapat menentukan apakah perangkat lunak yang diuji dapat membuat sebuah kesimpulan apakah penerapan Algoritma *neural network* untuk prediksi tingkat akurasi pengujian perangkat lunak metode *Black-Box* dapat diterapkan.



Gambar 3.3 Struktur Multi-Layer dan sinyal *propagation* dalam network

Inisialisasi

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 tidak pernah mencapai 0 ataupun 1[6]. Rumusnya :

TABEL III.3. 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
2	1	0,12	0,1	0,1	0,1	InCorrect
2	2	0,12	0,1	0,1	0,1	InCorrect
2	3	0,12	0,1	0,1	0,1	InCorrect
...						
40	5	0,12	0,1	0,1	0,1	InCorrect
200	5	0,12	0,1	0,1	0,1	InCorrect

TABEL 3.4. DATASET PREDIKSI BERBASIS ALGORITMA NEURAL NETWORK PENGUJIAN PERANGKAT LUNAK METODE BLACK-BOX, NILAI RMSE=0,00142438 PADA ITERASI 481

Form	Realitas		Prediksi		Error	Probability
	Defect	Target	Defect	Output		
1	1	Correct	0,991385	Correct	0,008615	0,991385
1	1	Correct	0,985423	Correct	0,014577	0,985423
1	1	Correct	0,951609	Correct	0,048391	0,951609
1	1	Correct	0,812001	Correct	0,187999	0,812001
1	0	InCorrect	0,011255	InCorrect	-0,01126	0,988744
2	0	InCorrect	0,006872	InCorrect	-0,00687	0,993128
2	0	InCorrect	0,014721	InCorrect	-0,01472	0,985279
2	0	InCorrect	0,013219	InCorrect	-0,01322	0,986781
2	0	InCorrect	0,012621	InCorrect	-0,01262	0,987379
2	1	Correct	0,994963	Correct	0,005037	0,994963
3	0	InCorrect	0,007019	InCorrect	-0,00702	0,992981
3	1	Correct	0,996158	Correct	0,003842	0,996158
3	1	Correct	0,984434	Correct	0,015566	0,984434
3	1	Correct	0,9837	Correct	0,0163	0,9837
3	1	Correct	0,995069	Correct	0,004931	0,995069
...						
4	0	InCorrect	0,011989	InCorrect	-0,01199	0,988011
7	1	Correct	0,99241	Correct	0,00759	0,99241
7	1	Correct	0,99647	Correct	0,00353	0,99647
7	1	Correct	0,996065	Correct	0,003935	0,996065
7	1	Correct	0,995877	Correct	0,004123	0,995877
7	1	Correct	0,995469	Correct	0,004531	0,99547

IV. HASIL DAN PEMBAHASAN

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.

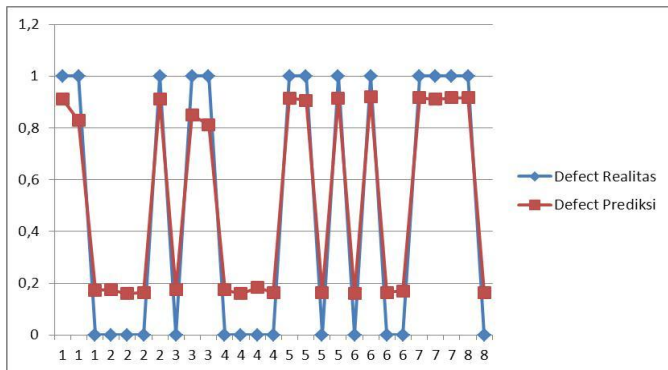
4.1 Hasil Prediksi

Berikut tingkat error dari *output* dan *defect* prediksi dengan hasil *output* dan *defect* realitas, dapat dilihat pada tabel 4.1,

dengan ketentuan banyaknya *neuron* pada *Hidden Layer* 4 *neuron*, *Learning Rate* =0,5, dan lamanya *EPOCH* = 500.

TABEL IV.1. HASIL PREDIKSI DATASET DENGAN *HIDDEN LAYER* 4 *NEURON*, *EPOCH*=500, DAN JUMLAH DATA 27, DENGAN *RMSE*= 0,020594

Form	Realitas		Prediksi		Error	Match?	Probability
	Defect	Target	Defect	Output			
1	1	Correct	0,912736	Correct	0,087264	OK	0,912736
1	1	Correct	0,827954	Correct	0,172046	OK	0,827954
1	0	InCorrect	0,173407	InCorrect	-0,17341	OK	0,826593
2	0	InCorrect	0,173981	InCorrect	-0,17398	OK	0,826019
2	0	InCorrect	0,160153	InCorrect	-0,16015	OK	0,839847
2	0	InCorrect	0,164163	InCorrect	-0,16416	OK	0,835837
2	1	Correct	0,91139	Correct	0,08861	OK	0,91139
3	0	InCorrect	0,17393	InCorrect	-0,17393	OK	0,82607
3	1	Correct	0,849083	Correct	0,150917	OK	0,849083
3	1	Correct	0,812788	InCorrect	0,187212	OK	0,812788
4	0	InCorrect	0,174118	InCorrect	-0,17412	OK	0,825882
4	0	InCorrect	0,159457	InCorrect	-0,15946	OK	0,840543
4	0	InCorrect	0,183345	InCorrect	-0,18335	OK	0,816655
4	0	InCorrect	0,163388	InCorrect	-0,16339	OK	0,836612
5	1	Correct	0,915393	Correct	0,084607	OK	0,915393
.
5	1	Correct	0,914469	Correct	0,085531	OK	0,914469
6	0	InCorrect	0,159688	InCorrect	-0,15969	OK	0,840312
6	1	Correct	0,920657	Correct	0,079343	OK	0,920657
6	0	InCorrect	0,163517	InCorrect	-0,16352	OK	0,836483
6	0	InCorrect	0,170497	InCorrect	-0,1705	OK	0,829503
7	1	Correct	0,91589	Correct	0,08411	OK	0,91589
7	1	Correct	0,910784	Correct	0,089216	OK	0,910784
7	1	Correct	0,915842	Correct	0,084158	OK	0,915842
8	1	Correct	0,915931	Correct	0,084069	OK	0,915931
8	0	InCorrect	0,164555	InCorrect	-0,16456	OK	0,835445



Gambar 4.1. Hasil Prediksi Dataset dengan *Hidden Layer* 4 *Neuron*, *Epoch*=500, dan jumlah data 27, dengan *RMSE*= 0,020594

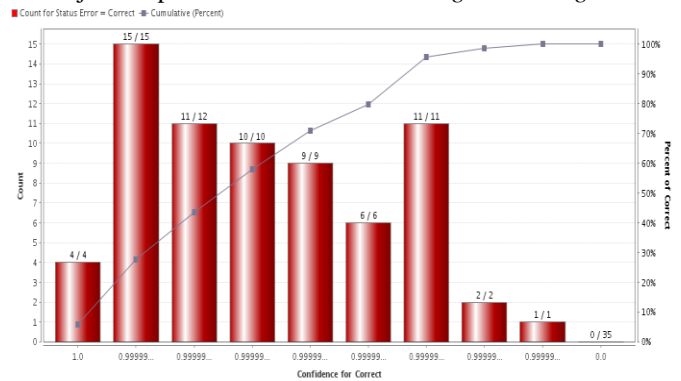
4.2 Hasil Prediksi Tingkat Akurasi berbasis Algoritma Neural Network Pengujian Perangkat Lunak dengan Metode Black-Box

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

Jumlah Hidden Layer	Epochs	Error Target	Confusion Matrix (Accuracy)	AUC
2	500	0.01	99.00%	0.976 +/- 0.071
3	500	0.01	85.00%	0.856 +/- 0.071
2	1000	0.01	98.00%	0.966 +/- 0.071
3	1000	0.01	86.00%	0.836 +/- 0.071

4.2.1 Grafik Performance ROC dalam bentuk Lift Chart dengan target class "Correct"

Pada gambar 4.2 merupakan Grafik ROC dalam bentuk *Lift Chart* dengan target class "*Correct*", grafik yang menunjukkan performance dataset *training* dan *testing*.



Gambar 4.2. Grafik ROC dalam bentuk *Lift Chart* dengan Target Class "*Correct*"

4.3.3 Grafik Performance ROC dalam bentuk Lift Chart dengan target class "InCorrect"

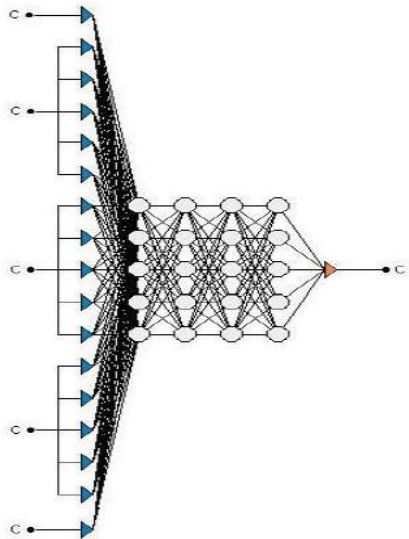
Pada gambar 4.6 merupakan Grafik ROC dalam bentuk *Lift Chart* dengan target class "*InCorrect*", grafik yang menunjukkan performance dataset *training* dan *testing*.



Gambar 4.3. Grafik ROC dalam bentuk *Lift Chart* dengan Target Class "*InCorrect*"

4.3.3 Arsitektur Design Neural Network dengan 4 hidden layer, 5 input layer, dan 1 output layer.

Pada gambar 4.4 merupakan Arsitektur *Design Neural Network* dengan 4 hidden layer, 5 input layer, dan 1 output layer, dan ini merupakan arsitektur *neural network* yang paling akurat dalam prediksi pengujian perangkat lunak.



Gambar 4.4 Arsitektur *Design Neural Network* dengan 4 hidden layer, 5 input layer, dan 1 output layer.

V. KESIMPULAN

Dari hasil penelitian yang dilakukan mulai dari tahap awal hingga proses pengujian, prediksi tingkat akurasi berbasis algoritma neural network pengujian perangkat lunak metode Black-Box dapat disimpulkan sebagai berikut:

1. Algoritma neural network bisa diterapkan untuk mengetahui tingkat akurasi pengujian perangkat lunak metode Black-Box dan akurasi sangat akurat, karena prediksi menunjukkan nilai rata-rata diatas 80%.
2. Equivalence Grade Partition menurut Kelvin tahun 1998, dapat dijadikan acuan analisa error untuk partion input dan output perangkat lunak.

ACKNOWLEDGMENT

Ucapan terima kasih disampaikan kepada STMIK Teknokrat atas dukungan data dan pendanaan dalam program dukungan penelitian dosen muda.

REFERENSI

- [1] Berard, C. (1994). Issues in the Testing of Object-Oriented Software..
- [2] Fourmier, Cs. (2009). Essential Software Testing A Use-Case Approach.
- [3] B. B. Agarwad, C. (2010). Software Engineering & Testing. Boston.
- [4] Perry, W. E. (1990). A standard for testing application software. 1990.
- [5] J.M, Z. (1992). Introduction to artificial neural systems.
- [6] S, H. (1999:p20). Neural networks – A comprehensive Foundation.
- [7] Albert Endres, Cs. (2003). Hanbook software and System Engineering,Empirical Observations, Laws and Theories.
- [8] Beizer, B. (1990). Software Testing Techniques.
- [9] Hetzel, W. C. (1988). The Complete Guide to Software Testing, 2nd ed.
- [10] Mark Last, Cs. (2002). Effective Black-Box Testing with Genetic Algorithms. ACM
- [11] Myers, G. J. (1979). The art of software testing. New York: New York
- [12] Oscar Pastor, Cs. (2007). Model-Driven Architecture in Practice,A Software Production Environment Based on Conceptual Modeling.
- [13] Patrick J, C. (2000). Black-Box Test Reduction Using Input-Output Analysis. ACM .