

Perancangan Model Sistem Prediksi Nilai Tukar Rupiah terhadap Dollar Amerika Menggunakan Jaringan Syaraf Tiruan QuickPropagation

Arief Budiman Hutauruk¹, Jondri MSI², Rita Rismala ST.,MT³

Fakultas Informatika Universitas Telkom Bandung

¹arrive_13h@yahoo.com, ²jondri@telkomuniversity.ac.id, ³rrs@telkomuniversity.ac.id

Abstrak

Setiap negara mempunyai mata uang masing-masing yang digunakan sebagai alat tukar. Nilai tukar mempunyai peranan penting dalam proses perdagangan, karena dengan mengetahui nilai tukar suatu mata uang terhadap mata uang lain kita dapat membanding harga barang dan jasa yang dihasilkan oleh negara lain. Prediksi nilai tukar Rupiah terhadap Dollar Amerika diperlukan agar pelaku kegiatan ekonomi di Indonesia dapat menentukan strategi bisnis yang tepat.

Jaringan Syaraf Tiruan Quickpropagation merupakan hasil pengembangan dari algoritma Backpropagation standar. Pada algoritma Quickpropagation dilakukan pendekatan dengan asumsi bahwa kurva fungsi error terhadap masing-masing bobot penghubung berbentuk parabola terbuka ke atas, dan gradien dari kurva error untuk suatu bobot tidak terpengaruh oleh bobot-bobot lain, sehingga penggunaan algoritma ini dapat meningkatkan kecepatan training pada Jaringan Syaraf Tiruan.

Kata Kunci : Quickpropagation, Jaringan Syaraf Tiruan, nilai tukar.

Abstract

Each country has individual currency that is used as a medium of exchange. The exchange rate has an important role in the trading process, because by knowing the exchange rate of a currency against another currency we can compare the prices of goods and services produced by other countries. Prediction of the Rupiah against the U.S. dollar is necessary that the performer of economic activities in Indonesia can determine the right business strategy

Quickpropagation Neural Network algorithm is the result of developing a standard Backpropagation. In the algorithm Quickpropagation based on with the assumption that the error function curves for each of the connecting weights parabolic open up, and the gradient of the error curve for a given weight is not affected by the other weights, so the use of this algorithm can improve the speed of training on Neural Network artificial.

Keyword : Quickpropagation, Neural Network, Currency

1. Pendahuluan

Setiap negara memiliki mata uang masing-masing yang digunakan sebagai alat tukar. Nilai tukar mempunyai peranan penting dalam proses perdagangan, karena dengan mengetahui nilai tukar suatu mata uang terhadap mata uang lain kita dapat membandingkan harga barang atau jasa yang dihasilkan oleh suatu negara tertentu. Mata uang selalu menghadapi kemungkinan untuk mengalami penurunan terhadap mata uang lain atau sebaliknya mengalami kenaikan nilai tukar. Hal inilah yang terjadi pada nilai tukar Rupiah terhadap Dollar

Amerika. Dollar Amerika merupakan mata uang utama dalam perdagangan dunia. Ketidaktahuan pelaku kegiatan ekonomi terhadap pergerakan Dollar Amerika tidak jarang menimbulkan kerugian materi yang tidak sedikit. Oleh karena itu penting bagi pelaku ekonomi di Indonesia untuk mengetahui nilai tukar Rupiah terhadap Dollar Amerika agar dapat menentukan strategi bisnis yang tepat[2].

Beberapa penelitian terkait algoritma Quickpropagation lain Analisis Implementasi dan Perbandingan Decision Tree J48 dengan JST Quickprop pada Kasus Teknik Klasifikasi dan

Prediksi Penyakit Demam Tifoid (Thypus) Berdasarkan Gejala Fisik Pasien. Penelitian lainnya adalah Pengenalan Alat Musik dengan Quickpropagation. Dikarenakan masih sedikitnya penggunaan algoritma Jaringan Syaraf Tiruan Quickpropagation, maka penulis mengimplementasikan penggunaan algoritma tersebut pada kasus lain yaitu prediksi nilai tukar Rupiah terhadap Dollar Amerika.

Berdasarkan penelitian-penelitian tersebut, penulis pada tugas akhir ini membangun sistem prediksi nilai tukar Rupiah terhadap Dollar Amerika dengan input berupa kurs beli dan kurs jual Rupiah terhadap Dollar Amerika. Model prediksi yang digunakan adalah Jaringan Syaraf Tiruan Quickpropagation. Algoritma Quickpropagation merupakan hasil pengembangan dari algoritma Backpropagation standar. Pada algoritma Quickpropagation dilakukan pendekatan dengan asumsi bahwa kurva fungsi error terhadap masing-masing bobot penghubung berbentuk parabola terbuka ke atas, dan gradien dari kurva error untuk suatu bobot tidak terpengaruh oleh bobot-bobot lain, sehingga penggunaan algoritma Quickpropagation dapat meningkatkan kecepatan training Jaringan Syaraf Tiruan[2].

2. Dasar Teori

2.1 Jaringan Syaraf Tiruan

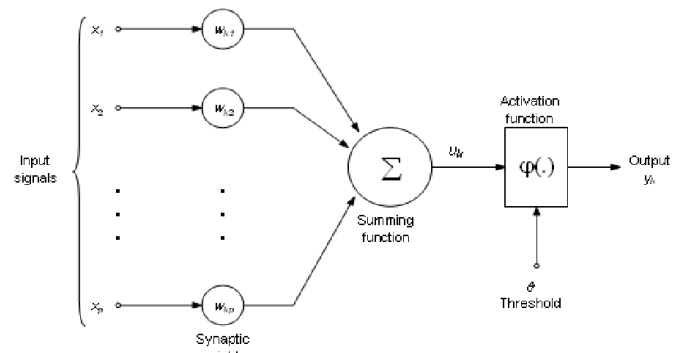
Jaringan syaraf tiruan (JST) adalah sistem pengolah informasi yang meniru sistem syaraf manusia[7]. Sama seperti sistem syaraf manusia yang tersusun oleh neuron, JST juga tersusun oleh perceptron (neuron pada JST) sebagai unit pemrosesan informasi yang merupakan dasar dari operasi JST [7].

2.1.1 Model Neuron

Terdapat tiga elemen dasar dari model neuron, yaitu[7] :

1. Sekumpulan sinapsis yaitu jalur penghubung antar neuron, dimana masing-masing sinapsis memiliki kekuatan hubungan atau bobot.
2. Suatu adder untuk menjumlahkan sinyal-sinyal input yang sudah dikalikan dengan bobot sinapsis yang sesuai.
3. Suatu fungsi aktivasi untuk menentukan output berdasarkan dari hasil penjumlahan sinyal-sinyal input dengan bobot sinapsis yang sesuai.

Contoh model neuron[7] :



Gambar 2.1 model neuron

2.1.2 Fungsi Aktivasi

Fungsi aktivasi digunakan untuk menentukan output dari suatu neuron[9].

Beberapa fungsi aktivasi yang sering digunakan pada JST, antara lain :

1. Fungsi Hard Limit
2. Fungsi Symmetric Hard Limit
3. Fungsi Linear
4. Fungsi Sigmoid Biner
5. Fungsi Sigmoid Bipolar

2.1.3 Arsitektur Jaringan

Beberapa arsitektur jaringan yang sering dipakai dalam JST antara lain[7] :

1. Single-Layer Network
Sekumpulan unit/neuron pada input layer dihubungkan dengan sekumpulan unit pada output layer. Setiap unit input diproyeksikan dengan semua unit output.
2. Multi-Layer Network
Jaringan dengan satu atau lebih hidden layer. Setiap unit input diproyeksikan ke semua hidden unit pada hidden layer dan setiap hidden unit diproyeksikan ke setiap unit output. Jika hidden layer lebih dari satu, maka setiap unit input diproyeksikan ke semua hidden unit pada hidden layer pertama, kemudian setiap hidden unit pada layer pertama diproyeksikan ke semua hidden unit pada hidden layer berikutnya dan seterusnya sampai hidden layer terakhir dan kemudian dilanjutkan dengan proyeksi dari hidden unit pada hidden layer terakhir ke semua unit output.
3. Recurrent Network
Jaringan yang mempunyai minimal satu feedback loop. Recurrent network bisa terdiri dari satu atau lebih layer. Unit output pada jaringan ini memberikan kembali outputnya sebagai sinyal input pada semua unit input.

2.1.4 Proses Pelatihan

Proses pelatihan pada JST dapat didefinisikan sebagai suatu proses dimana parameter-parameter bebas JST diadaptasikan melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada [7].

Jenis pelatihan pada JST yaitu[7] :

1. Supervised Learning (Pelatihan dengan pengawasan)

Dalam poses pelatihan ini, JST diberikan suatu rangsangan dari lingkungan dimana pengetahuan tentang lingkungan tersebut sudah diketahui (biasa direpresentasikan dengan input-output).

Dengan rangsangan tersebut JST akan memberikan respon. Respon tersebut

merepresentasikan aksi yang dilakukan oleh JST. Parameter-parameter JST berubah-ubah berdasarkan rangsangan dan sinyal kesalahan (sinyal kesalahan adalah perbedaan antara

output JST dan output yang diinginkan). Sinyal kesalahan digunakan sebagai umpan balik ke

jaringan. Proses perubahan ini dilakukan terus-menerus sampai JST mampu memetakan sekumpulan input-output dengan akurasi yang tinggi.

2. Unsupervised Learning (Pelatihan tanpa pengawasan)

Dalam proses pelatihan ini, terdapat suatu layer pada jaringan dimana neuron-neuronnya bersaing meraih “kesempatan” untuk memberikan respon kepada jaringan tentang ciri khas dari data masukan. Berdasarkan dari ciri-ciri yang diberikan oleh neuron tersebutlah, jaringan menghasilkan output.

2.2 Algoritma Quickpropagation

Quickpropagation merupakan modifikasi dari algoritma Backpropagation standar yang diperkenalkan oleh Fahlman pada 1988. Pada algoritma Quickpropagation dilakukan pendekatan dengan asumsi bahwa kurva fungsi error terhadap masing-masing bobot penghubung berbentuk parabola yang terbuka ke atas dan gradient dari kurva error tidak terpengaruh oleh bobot-bobot yang lain. Dengan demikian perhitungan perubahan bobot hanya menggunakan informasi local pada masing-masing bobot sehingga menyebabkan proses belajar pada Jaringan Syaraf Tiruan dapat dilakukan dengan cepat[2].

Tahapan algoritma Quickpropagation dengan fungsi aktivasi sigmoid bipolar[5]:

1. Mendefinisikan matrix masukan (P) dan matrix target (T).
2. Menginisialisasi parameter-parameter Jaringan Syaraf Tiruan seperti jumlah elemen input, jumlah neuron pada hiddenlayer, jumlah neuron pada outputlayer, bobot W_1, W_2 , Mean Square Error (MSE), Learning Rate (lr), Max Growth Factor (μ) dan jumlah epoch.
3. Melakukan pelatihan Jaringan Syaraf Tiruan:

- a. Perhitungan Maju

Menghitung nilai keluaran dari hiddenlayer (A_1) dengan persamaan:

$$A_1 = () = \frac{(\quad)}{(\quad)}$$

Hasil keluaran dari A_1 digunakan untuk mendapatkan keluaran dari outputlayer dengan persamaan:

$$A_2 = () = \frac{(\quad)}{(\quad)}$$

Selanjutnya keluaran dari Jaringan Syaraf Tiruan dibandingkan dengan target dan dihitung selisihnya sehingga menghasilkan nilai error (E) dengan persamaan:

$$E = T - A_2$$

Hitung derivative error untuk tiap bobot antara hiddenlayer dengan outputlayer dan inputlayer dengan hiddenlayer dengan persamaan:

$$\begin{aligned} \delta_j &= () \\ \delta_j &= \sum_k () () \\ &= (-) \left(\frac{(\quad)}{(\quad)} \right) \\ \delta_j &= \sum_k () () \\ &= \left(\frac{(\quad)}{(\quad)} \right) \end{aligned}$$

n= jumlah data latih.

k=jumlah neuron pada outputlayer.

j=jumlah neuron pada hiddenlayer.

- b. Perhitungan Mundur

Setelah didapatkan derivative error terhadap tiap bobot antara hiddenlayer dengan outputlayer dan hiddenlayer dengan inputlayer untuk semua data latih, maka selanjutnya dilakukan perubahan bobot pada epoch pertama dengan persamaan:

$$W(t) = W(t-1) + \Delta W(t)$$

Sedangkan untuk selanjutnya:

□ Jika $\Delta W(t-1) > 0$, maka:

□ Jika $W(t-1) > (\mu/(1 + \mu)) W(t-2)$ maka:

$$\Delta W(t) = a + \mu * \Delta W(t-1)$$

□ Jika $W(t-1) \leq (\mu/(1 + \mu)) W(t-2)$ maka:

$$W(t) = W(t-1) + \Delta W(t)$$

$$\Delta W(t) = \frac{W(t-1) - W(t-2)}{W(t-1) - W(t-2)} * (W(t-1) - W(t-2))$$

Keterangan:

a bernilai $1 - \mu$ jika nilai $W(t-1) > 0$.

a bernilai 0 jika nilai $W(t-1) \leq 0$.

□ Jika $\Delta W(t-1) < 0$ maka:

□ Jika $W(t-1) < (\mu/(1 + \mu)) W(t-2)$ maka:

$$\Delta W(t) = a + \mu * (\Delta W(t-1)) \tag{2.11}$$

□ Jika $W(t-1) \geq (\mu/(1 + \mu)) W(t-2)$ maka:

$$W(t) = W(t-1) + \frac{W(t-1) - W(t-2)}{W(t-1) - W(t-2)} * (W(t-1) - W(t-2))$$

Keterangan:

a bernilai $1 - \mu$ jika nilai $W(t-1) < 0$.

a bernilai 0 jika nilai $W(t-1) \geq 0$.

Setelah nilai-nilai perbaikan untuk setiap bobot didapat, maka selanjutnya adalah perbaikan nilai bobot dengan persamaan:

$$W_1 = W_1 + \Delta W_1$$

$$W_2 = W_2 + \Delta W_2$$

MSE dapat dihitung dengan persamaan:

$$= \sum$$

c. Langkah tersebut adalah untuk satu kali siklus pelatihan (1 x epoch). Pelatihan dilakukan berulang-ulang sampai epoch tertentu atau sampai MSE yang diinginkan.

Hasil akhir dari pelatihan adalah nilai W_1 dan W_2 .

3. Hasil Pengujian

Data yang digunakan untuk pengujian adalah data nilai tukar Rupiah terhadap Dollar Amerika

yang sudah dilakukan tahap preprocessing. Input Neuron yang digunakan adalah 5 dan hidden neuron yang digunakan adalah 5, sedangkan

output neuron adalah 1. Berikut ini adalah beberapa skenario yang akan diuji:

1. Skenario Pengujian 1

Kombinasi data training-data testing: 80%-20%.

Learning Rate: 0.1, 0.2, 0.3

Max Growth Factor: 1.75, 2, 2.25.

Tabel 4.1 Data Hasil Pengujian Skenario 1.

Epoch	L R	Max GF	MAPE Tr	Akura siTr	MAP ETs	Akura siTs
500	0.		0.339	99.66	0.204	99.79
	1	1.75	03	097	8	52
750	0.		0.524	99.47	0.230	99.76
	1	1.75	28	572	1	99
1000	0.		0.645	99.35	0.168	99.83
	1	1.75	03	497	2	18
500	0.				2.530	97.46
	2	1.75	1.16	98.84	5	95
750	0.		0.578	99.42	0.279	99.72
	2	1.75	69	131	8	02
1000	0.		1.762	98.23	1.720	98.27
	2	1.75	1	79	2	98
500	0.		0.217	99.78	0.162	99.83
	3	1.75	56	244	9	71
750	0.		0.553	99.44	0.279	99.72
	3	1.75	16	684	8	02
1000	0.		0.654	99.34	0.179	99.82
	3	1.75	41	559	6	04
500	0.		0.367	99.63	0.279	99.72
	1	2	35	265	8	02
750	0.		0.555	99.44	0.279	99.72
	1	2	78	422	8	02

1000	0.1	2	0.71064	99.28936	0.2466	99.7534
500	0.2	2	0.29513	99.70487	0.1409	99.8591
750	0.2	2	0.55138	99.44862	0.2798	99.7202
1000	0.2	2	0.69279	99.30721	0.2346	99.7654
500	0.3	2	0.33636	99.66364	0.1962	99.8038
750	0.3	2	0.055108	99.94489	0.2789	99.7211
1000	0.3	2	0.62296	99.37704	0.1548	99.8452
500	0.1	2.25	0.36734	99.63266	0.2798	99.7202
750	0.1	2.25	1.3211	98.6789	1.7202	98.2798
1000	0.1	2.25	0.7354	99.2646	0.2798	99.7202
500	0.2	2.25	0.36733	99.63267	0.2798	99.7202
750	0.2	2.25	0.55136	99.44864	0.2798	99.7202
1000	0.2	2.25	0.68286	99.31714	0.2074	99.7926
500	0.3	2.25	0.36733	99.63267	0.2798	99.7202
750	0.3	2.25	0.53486	99.46514	0.2519	99.749
1000	0.3	2.25	1.7621	98.2379	1.7202	98.2798

2. Skenario Pengujian 2

Kombinasi data training-data testing 70%-30%.

Learning Rate: 0.1, 0.2, 0.3.

Max Growth Factor: 1.75, 2, 2.25.

Tabel 4.2 Data Hasil Pengujian Skenario 2.

Epo ch	L R	Max GF	MAP ETr	Akura siTr	MAP ETs	Akuras iTs
500	0.1	1.75	0.97362	99.02638	1.6045	98.3955
750	0.1	1.75	0.67866	99.32134	0.3955	99.6045
1000	0.1	1.75	0.87751	99.12249	0.3657	99.6343

500	0.2	1.75	0.97362	99.02638	1.6045	98.3955
750	0.2	1.75	0.67867	99.32133	0.3955	99.6045
1000	0.2	1.75	1.9492	98.0508	1.6045	98.3955
500	0.3	1.75	0.97205	99.02795	1.6045	98.3955
750	0.3	1.75	1.4614	98.5386	1.6045	98.3955
1000	0.3	1.75	1.9492	98.0508	1.6045	98.3955
500	0.1	2	0.4173	99.583	0.3191	99.6809
750	0.1	2	0.67867	99.32133	0.3955	99.6045
1000	0.1	2	1.9492	98.0508	1.6045	98.3955
500	0.2	2	0.97362	99.02638	1.6045	98.3955
750	0.2	2	0.67868	99.32132	0.3955	99.6045
1000	0.2	2	0.90528	99.09472	0.3955	99.6045
500	0.3	2	0.45214	99.54786	0.3955	99.6045
750	0.3	2	1.4614	98.5386	1.6045	98.3955
1000	0.3	2	0.51139	99.48861	1.3603	98.6397
500	0.1	2.25	0.44988	99.55012	0.696	99.304
750	0.1	2.25	1.4682	98.532	0.6978	99.3022
1000	0.1	2.25	0.90501	99.09499	0.6978	99.3022
500	0.2	2.25	0.45215	99.54785	0.6978	99.3022
750	0.2	2.25	0.67868	99.32132	0.6978	99.3022
1000	0.2	2.25	1.9492	98.0508	1.6045	98.3955
500	0.3	2.25	0.45215	99.54785	0.6978	99.3022
750	0.3	2.25	1.4614	98.5386	1.6045	98.3955
1000	0.3	2.25	1.9062	98.0938	1.6045	98.3955

3. Skenario Pengujian 3

Kombinasi data training-data testing 60%-40%.

Learning Rate: 0.1, 0.2, 0.3.

Max Growth Factor: 1.75, 2, 2.25.

Tabel 4.3 Data Hasil Pengujian Skenario 3.

Epo ch	L R	Max GF	MAP ETr	Akura siTr	MAP ETs	Akuras iTs
500	0.1	1.75	1.111	98.889	1.5398	98.4602
750	0.1	1.75	0.83578	99.16422	1.5398	98.4602
1000	0.1	1.75	2.2242	97.7758	1.5398	98.4602
500	0.2	1.75	1.1232	98.8768	1.5398	98.4602
750	0.2	1.75	0.86588	99.13412	0.7301	99.2699
1000	0.2	1.75	1.1228	98.8772	1.1771	98.8229
500	0.3	1.75	1.2205	98.7795	1.6737	98.3263
750	0.3	1.75	1.6676	98.3324	1.5398	98.4602
1000	0.3	1.75	1.1148	98.8852	1.5398	98.4602
500	0.1	2	1.3262	98.6738	1.654	98.346
750	0.1	2	1.6693	98.3307	1.5398	98.4602
1000	0.1	2	2.2243	97.7757	1.5398	98.4602
500	0.2	2	0.51072	99.48928	0.7116	99.2884
750	0.2	2	1.2864	98.7136	1.5398	98.4602
1000	0.2	2	2.2326	97.7674	1.5398	98.4602
500	0.3	2	1.1119	98.889	1.5398	98.4602
750	0.3	2	0.83915	99.16085	0.7301	99.2699
1000	0.3	2	1.1216	98.8784	1.5398	98.4602
500	0.1	2.25	0.49698	99.50302	0.7025	99.2975
750	0.1	2.25	0.82904	99.17096	0.7301	99.2699
1000	0.1	2.25	1.1058	98.8942	0.7301	99.2699

500	0.2	2.25	0.56731	99.43269	0.7301	99.2699
750	0.2	2.25	1.6764	98.324	1.5398	98.4602
1000	0.2	2.25	1.1401	98.8599	0.7301	99.2699
500	0.3	2.25	0.46233	99.53767	0.6829	99.3171
750	0.3	2.25	0.84236	99.15764	0.7301	99.2699
1000	0.3	2.25	1.1424	98.8576	1.5398	98.4602

4. Skenario Pengujian 4

Setelah dilakukan terhadap pengujian terhadap tiga scenario pengujian di atas maka didapatkan struktur JST dengan akurasi testing terbaik yaitu JST dengan input neuron lima, hidden neuron 5, learning rate 0.2, epoch 500, dengan data training 80% dan data testing 20%. Parameter-parameter ini kemudian akan digunakan untuk melakukan pengujian pada JST Backpropagation sebagai perbandingan performansinya.

Tabel 4.4.1 Data Hasil Pengujian Skenario 4 Backproppagation

Epo ch	LR	MAPET r	Akurasi Tr	MAPET s	Akurasi Ts
500	0.2	0.96305	99.03695	1.7202	98.2798

Tabel 4.4.2 Data Hasil Pengujian Skenario 4 Quickpropagation

Epo ch	L R	Max GF	MAP ETr	Akura siTr	MAP ETs	Akuras iTs
500	0.2	2	0.29513	99.70487	0.1409	99.8591

4. Analisis

Hasil pengujian terhadap berbagai skenario adalah sebagai berikut: Setelah melakukan percobaan terhadap empat skenario di atas, dapat disimpulkan bahwa akurasi testing terbaik didapat pada scenario 1 dengan input neuron lima, hidden neuron 5,

learning rate 0.2, epoch 500, dan max growth factor 2 yang menghasilkan akurasi 99,8591%. Setelah melakukan pengujian pada scenario 4 yaitu dengan membandingkan performansi antara JST Quickpropagation dengan JST Backpropagation dengan parameter-parameter yang menghasilkan akurasi terbaik, JST Quickpropagation menghasilkan akurasi testing yang lebih baik daripada JST Backpropagation, begitu pula dengan akurasi pada saat training.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari Tugas Akhir ini yaitu:

1. Pada Pengujian kali ini, yang memiliki akurasi terbaik merupakan kombinasi antara 80% data training, 20% data testing, Learning Rate 0.2, Max Growth Factor 2, dan epoch 500 menghasilkan akurasi sebesar 99.8591%.
2. Setelah melakukan pengujian terhadap beberapa scenario di atas dapat disimpulkan bahwa JST QuickPropagation dapat memberikan akurasi yang lebih baik daripada JST Backpropagation baik pada saat training maupun testing.

5.2. Saran

Saran yang dapat diberikan pada Tugas Akhir ini sebagai pengembangan system yaitu:

1. Dapat dilakukan pengujian lebih lanjut untuk data nilai tukar terhadap mata uang lainnya.
2. Menggunakan Evolutionary Computation untuk mendapatkan arsitektur Jaringan Syaraf Tiruan yang optimal.

6. Daftar Pustaka

- [1] Hadi, Umar. 2012. Analisis Proses Data Preprocessing pada Jaringan Syaraf Tiruan Quickpropagation untuk Peramalan Time Series. Teknik Informatika. IT Telkom
- [2] Ilie, Constantin. Ilie, Margareta. Topalu, Ana-Maria. Melnic, Lucia. 2012. The Simulation of US Consumer Credit Fluctuation Using Artificial Neural Networks. OVIDIUS University from Constanta, Faculty of Mechanical, Industrial and Maritime Engineering, Mamaia Ave, Romania
- [3] Novanto, Prabowo Gandhi. 2010. Sistem Prediksi Kurs Mata Uang Menggunakan Jaringan Syaraf Tiruan dengan Metode Quickpropagation. Fakultas Teknologi Industri Universitas Pembangunan Nasional "Veteran" Jawa Timur
- [4] Rismala, Rita. 2011. Analisis Perbandingan Evolution Strategies dan differential Evolution pada Prediksi Data Time Series Studi Kasus Kurs Jual Emas. Bandung. Institut Teknologi Telkom
- [5] Siregar, Ferry Agustian. 2013. Pengenalan Manusia Menggunakan Iris Mata dengan Metode Principal Component Analysis dan Algoritma Quickpropagation. Teknik Informatika. IT Telkom
- [6] Suyanto, ST, MSc. 2011. Artificial Intelligence Searching-Reasoning-Planning- Learning. Bandung. INFORMATIKA
- [7] http://ann.thwien.de/index.php/Multilayer_perception. Diakses pada tanggal 26 Maret 2014
- [8] <http://www.bioinfo.de/isb/2007/07/0010/main.html>. Diakses pada tanggal 26 Maret 2014
- [9] <http://www.fmi.uni-sofia.bg/fmi/statist/education/textbook/ENG/glosq.html>. Diakses pada tanggal 26 Maret 2014
- [10] http://www.yaldex.com/game-development/1592730043_ch20lev1sec5.html. Diakses pada tanggal 26 Maret 2014
- [11] http://www.yaldex.com/game-development/1592730043_ch20lev1sec5.html. Diakses pada tanggal 26 Maret 2014
- [12] http://id.wikipedia.org/wiki/Nilai_tukar. Diakses pada tanggal 26 Maret 2014