ISSN 2085-4552

Kombinasi *Backpropagation* dan Hopfield Modifikasi untuk Persamaan *Polynomial*

Rina Pramitasari[1], Imam Rofiki[2]
[1] Teknik Komputer, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
rina.pramitasari@amikom.ac.id
[2] Tadris Matematika, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia
imam.rofiki@uin-malang.ac.id

Diterima 23 Mei 2020
Disetujui 23 Juni 2020

***Abstract*—Salah satu sistem kriptografi yang bertahan di era komputer quantum adalah kriptografi kunci publik multivariat. Tingkat keamanan sistem kriptografi kunci publik multivariat adalah sulitnya menyelesaikan sistem persamaan polinomial multivariable. Tujuan artikel adalah mengkriptanalisis dengan mencari akar menyelesaikan persamaan polinomial. Penelitian ini menggunakan metode kombinasi *Backpropagarian* dan Hopfield Modifikasi. Hasil penelitian menunjukkan bahwa metode kombinasi *Backpropagarian* dan Hopfield Modifikasi lebih baik dari pada metode Hopfield Modifikasi saja. Karena menjamin nilai awal yang diberikan pada Hopfield Modifikasi selalu dekat dengan nilai solusi. Metode kombinasi *Backpropagarian* dan Hopfield Modifikasi memberikan solusi yang akurat.**

***Index Terms*—Backpropagation, Hopfield Network Modifikasi, Persamaan Polinomial**

I. PENDAHULUAN

Sistem kriptografi kunci publik yang paling banyak digunakan adalah RSA. Algoritme RSA berasal dari nama akhir penemunya pada tahun 1977, yaitu Rivest-Shamir-Adleman. Tingkat keamanan RSA terletak pada faktorisasi bilangan bulat. Kunci publik yang aman pada RSA setidaknya 1024 bits atau 2048 bits. Hal ini mengakibatkan proses enkripsi tidak efisien. Ancaman baru juga muncul yang bisa melemahkan RSA yaitu komputer quantum yang sedang berkembang. Ada beberapa kandidat sistem kriptografi yang mampu bertahan dengan komputer quantum yaitu *Lattice-Based Cryptography*, *Code-Based Cryptography*, *Multivariate-Based Cryptography* [4], [7], [8], [11].

Pengujian sistem kriptografi salah satunya adalah dengan menggunakan kriptanalisis. Sistem kriptografi kunci publik atau sistem kriptografi asimetris, menggunakan dua jenis kunci, yaitu kunci publik dan kunci rahasia. Kunci publik digunakan untuk mengenkripsi suatu pesan. Kunci publik bersifat umum, sehingga dapat dilihat oleh semua orang termasuk pihak penyerang. Kunci rahasia digunakan untuk mendekripsi suatu pesan. Kunci rahasia bersifat dirahasiakan dan hanya orang tertentu yang boleh mengetahuinya [7].

Sistem kriptografi kunci publik multivariat didasarkan pada sistem persamaan *multivariable* nonlinear pada bidang hingga. Tingkat keamanan didasarkan pada sulitnya penyelesaian sebuah sistem persamaan polynomial multivariable kuadratik yang disebut MQ-problem (*multivariate quadratic problem*) [4], [7], [8], [11].

Menyelesaikan sistem persamaan nonliner bentuk yang sederhana dapat diselesaikan dengan metode analitik. Sedangkan faktanya sering menemukan bentuk yang rumit, tidak semua persamaan dapat dipecahkan dengan metode analitik, maka solusinya dapat dicari dengan metode numerik. Sedangkan metode numerik harus memenuhi syarat dekat dengan solusi untuk bisa konvergen. Maka untuk menyelesaikan sistem persamaan tersebut digunakan jaringan saraf tiruan di mana sebuah prosesor yang terdistribusi parallel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan [9].

Jariangan *Backpropagation* diambil dari cara kerja yaitu bahwa gradien *error* unit-unit tersembunyi diturunkan dari penyiaran kembali *error* yang diasosiasikan dengan unit *output*. Hal ini karena nilai target untuk unit-unit tersembunyi tidak diberikan [2].

Jaringan Hopfield pertama kali diperkenalkan John Hopfield pada tahun 1982. Jaringan Hopfield adalah pelatihantak terbimbing (*unsupervised learning*). Jaringan Hopfield merupakan jaringan syaraf tiruan yang terhubung penuh (*fully connected*), yaitu bahwa setiap neuron terhubung dengan neuron lainnya [2]. Jaringan Hopfield Modifikasi mampu untuk menyelesaikan sistem persamaan nonlinear di dalam satu variabel. Hasil penelitian menunjukkan bahwa jaringan Hopfield modifikasi akan selalu konvergen terhadap sembarang nilai awal, dimana sebaliknya metode numerik yang harus memenuhi syarat “dekat” dengan solusi [1], [6].

II. LANDASAN TEORI

A. Persamaan Polynomial

Persoalan mencari solusi persamaan nonlinear adalah: Tentukan nilai x yang memenuhi persamaan.

$$f(x) = 0$$

yaitu nilai $x = s$ sedemikian sehingga $f(s)$ sama dengan nol [9]. Persamaan nonlinear yang melibatkan fungsi transeden, diantaranya sinus, cosinus, eksponensial, logaritma. Selain itu, persamaan nonlinear yang tidak melibatkan fungsi transenden, yaitu persamaan polynomial. Bentuk umum persamaan polynomial satu variabel x adalah

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$$

Contohnya:

a. Satu variabel yaitu *x*,

$$4x^5 + 9x^3 + 2x - 5 = 0$$

b. Dua variabel yaitu *x* dan *y*,

$$3x^2 + 9xy - y = 0$$

Pada peneltian ini, penulis menggunakan satu variabel.

B. Backpropagation

Backpropagation adalah metode yang menurukan gradien untuk meminimkan penjumlahan *error* kuadrat *output* jaringan. Nama lain dari *Backpropagation* adalah aturan delta yang digeneralisasi.

Dalam *Backpropagation*, setiap unit yang berbeda di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Hal serupa berlaku pula pada lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output [3], [5], [6].

Langkah *Backpropagation* adalah sebagai berikut:

- Inisialisasi bobot dan bias
  1. **While** kondisi berhenti tidak terpenuhi **do** langkah ke-2 sampai langkah ke-9.
  2. Untuk setiap pasang pola training, lakukan langkah ke-3 sampai langkah ke-8.
- Umpan Maju (Feedforward)
  3. Setiap unit *input*, mengirimkan sinyal *input* ke semua unit pada lapisan tersembunyi.
  4. Pada setiap unit di lapisan tersembunyi, sinyal *input output* lapisan tersembunyi dihitung dengan dengan fungis aktivasi terhadap penjumlahan sinyal *input* berbobot.
  5. Pada setiap unit di lapisan *output* unit, dihitung sinyal *output*-nya dengan menerapkan fungsi aktivasi terhadapt penjumlahan sinyal input berbobot.
- Umpan Mundur (*Backpropagation Error*)
  6. Pada setiap unit *output* menerima pola target lalu informasi kesalahan lapisan *output* dihitung. Dikirim ke lapisan di bawahnya dan digunakan untuk menghitung besar koreksi bobot dan bias antara lapisan tersembunyi dengan lapisan *output*.
  7. Pada setiap unit di lapisan tersembunyi dilakukan perhitungan informasi kesalahan lapisan tersembunyi. Kemudian digunakan untuk menghitung besar koreksi bobot dan bias antara lapisan input dan lapisan tersembunyi.
- Update Bobot dan Bias (*Adjustment*)
  8. Pada setiap unit output, dilakukan peng-*update*-an bias dan bobot.
  9. Tes kondisi berhenti. Digunakan kriteria MSE (*Mean Square Error*).

C. Hopfield Modifikasi

Hopfield mengembangkan rancangan jaringan biner sehingga neuron dapat memperhitungkan nilai real [3]. Pengembangan dari jaringan Hopfield Kontinu adalah menyerupai kerja jaringan diskrit, tetapi jaringan ini mempunyai kemampuan lebih karena arsitekturnya lebih komplek dan tidak hanya nilai biner 0 dan 1 [1].

Hopfield Modifikasi didasarkan pada prinsip Hopfield kontinu, karena nilai *input* dan *output* yang diinginkan adalah real antara 0 sampai 1, tidak hanya 0 atau 1. Jumlah neuron pada jaringan sama dengan jumlah variabel persamaan. Hubungan antara neuron pada Hopfield Modifikasi untuk menyelesaikan persamaan *polynomial* bergantung pada hubungan antara variabel persamaan dengan koefisien, yang diturunkan sebagai bobot pada jaringan. Hubungan antara variabel pada persamaan *polynomial* adalah hubungan nonlinear, sehingga jaringan Hopfield harus dimodifikasi [1].

Langkah-langkah Hopfield Modifikasi adalah sebagai berikut:

1. Mengubah persamaan *polynomial* menjadi bentuk fungsi energi.

$$g_i(.) = (f_i(x_1, \cdots, x_j, \cdots, x_n) + (-P_i))$$

$$E = -\frac{1}{2}\sum_{i=1}^{m}(g_i(.))^2$$

2. Menurunkan persamaan fungsi energi.

$$\frac{du_j}{dt} = \frac{\partial E}{\partial x_j}$$

3. Kemudian menyesuaikan dengan persamaan Hopfield.

$$\sum_i W_{ji}\, f_{ji}(x_1, x_2, \cdots, x_n) + I_j, \quad (i,j) \in n$$

4. Menginisialisasi nilai awal $x(1)$ dan $u(1)$.
5. Menghitung dengan metode Euler untuk memperbaiki nilai $u(t)$.

$$u(t+1) = u_j(t) + \Delta t\left(\frac{\partial E}{\partial x_j}\right)$$

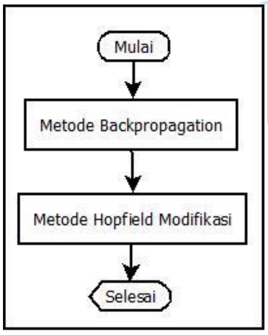
6. Memperbarui nilai $x(t)$.

$$x(t) = \frac{1}{1 + e^{-u(t)}}$$

7. Hingga syarat terpenuhi.

III. Metode Penelitian

Gambar 1 menunjukan *flowchart* penelitian. Langkah pertama, dengan menggunakan metode *Backpropagation* didapat nilai *output* yang optimal. Langkah kedua, menggunakan nilai *output* untuk memberilkan nilai awal pada metode Hopfield Modifikasi untuk menyelesaikan persamaan *polynomial* [10]. Hasil penelitian akan dibandingkan dengan metode Hopfield Modifikasi dan metode konvensional yaitu metode numerik Newton Raphson.



Gambar 1. *Flowchart* penelitian

Misalkan persamaan *polynomial* berbentuk:

$$Ax^2 + Bx + C = 0$$

Proses *Backpropagation* menggunakan *layer input* 3 neuron, *layer hidden* 10 neuron, dan *layer output* 1 neuron. Fungsi aktivasi yang digunakan adalah Sigmoid Biner dan *learning rate* (α) = 0,5. *Epoch* = 10000. Toleransi *error* yang digunakan adalah 0,000000000000001.

Proses Hopfield Modifikasi adalah:

1. Mengubah persamaan *polynomial* menjadi bentuk fungsi energi.

$$Ax^2 + Bx + C = 0$$

$$E = -\frac{1}{2}\sum_{i=1}^{m}\left(g_i(.)\right)^2$$

$$E = -\frac{1}{2}(Ax^2 + Bx + C)^2$$

2. Menurunkan persamaan fungsi energi.

$$\frac{du_j}{dt} = \frac{\partial E}{\partial x_j}$$

$$= -(2A^2x^3 + (AB + 2AB)x^2 + (B^2 + 2AC)x + BC)$$

3. Kemudian menyesuaikan dengan persamaan Hopfield.

$$\frac{du}{dt} = W_1 x^3 + W_2 x^2 + W_3 x + I_{bias}$$

$$W_1 = 2A^2, W_2 = AB + 2AB,$$

$$W_3 = B^2 + 2AC, I_{bias} = BC$$

4. Menginisialisasi nilai awal $x(1)$ adalah hasil dari *Backpropagation* dan $u(1)$ adalah -0.5, 1, atau 0.01.
5. Menghitung dengan metode Euler untuk memperbaiki nilai $u(t)$.

$$u(t+1) = u(t) + \Delta t(W_1 x_t^3 + W_2 x_t^2 + W_3 x_t + I_{bias})$$

6. Memperbarui nilai $x(t)$

$$x(t) = \frac{1}{1 + e^{-u(t)}}$$

7. Hingga syarat terpenuhi.

IV. Pembahasan

Dalam pembahasan menggunakan pemrograman *Python*. Kemudian menyiapkan data pelatihan dan data pengujian dengan merujuk penelitian [1], [4], [7], [12].

Data pelatihan terdiri dari 2 bentuk persamaan yaitu Pertama berbentuk:

$$Ax^2 + Bx + C = 0$$

Jumlah dari data pelatihan terdiri dari 10 dan 20 data. Data pengujian tersebut terdiri dari 3 persamaan yaitu:

$x^2 - 0.0324 = 0$ dengan *Output* 0.18

$x^2 - 0.3364 = 0$ dengan *Output* 0.58

$x^2 - 0.9604 = 0$ dengan *Output* 0.98

Jaringan *Backpropagation* menggunakan *layer input* 3 neuron, *layer hidden* 10 neuron, dan *layer output* 1 neuron. Fungsi aktivasi yang digunakan adalah Sigmoid Biner dan *learning rate* (α) = 0,5. *Epoch* = 10000. Toleransi *error* yang digunakan adalah 0,000000000000001.

Kedua berbentuk:

$$Ax^4 + Bx^3 + Cx^2 + Dx + E = 0$$

Jumlah dari data pelatihan terdiri dari 10 dan 20 data. Data pengujian tersebut terdiri dari 3 persamaan yaitu:

$$x^4 + x^3 - 0.0324x - 0.00104976 = 0$$

dengan *Output* 0.18

$$x^4 + x^3 - 0.3364x - 0.11316496 = 0$$

dengan *Output* 0.58

$$x^4 + x^3 - 0.9604x - 0.92236816 = 0$$

dengan *Output* 0.98

Jaringan *Backpropagation* menggunakan *layer input* 5 neuron, *layer hidden* 10 neuron, dan *layer output* 1 neuron, Fungsi aktivasi yang digunakan adalah Sigmoid Biner dan *learning rate* (α) = 0,5. *Epoch* = 10000. Toleransi *error* yang digunakan adalah 0,000000000000001.

Kemudian melakukan proses perhitungan jaringan Hopfield Modifikasi berdasarkan berbedaan nilai $u(t)$. Nilai $u(t)$ yang dipilih adalah -0.5, 1, dan 0.01.

Tabel 1. Hasil pengujian dengan persamaan $x^2 - 0.0324 = 0$

Dengan 10 ***Training*** Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.1800000045	840	0.0
1	0.1800000045	845	0.0
0.01	0.1800000046	842	0.0
Dengan 20 *Training* Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.1800000045	840	0.0
1	0.1800000045	845	0.0
0.01	0.1800000046	842	0.0

Tabel 2. Hasil pengujian dengan persamaan $x^2 - 0.3364 = 0$

Dengan 10 ***Training*** Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.5799999999	58	0.0
1	0.5800000001	54	0.0
0.01	0.5799999999	54	0.0
Dengan 20 *Training* Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.5799999999	58	0.0
1	0.5800000001	54	0.0
0.01	0.5799999999	54	0.0

Tabel 3. Hasil pengujian dengan persamaan $x^2 - 0.9604 = 0$

Dengan 10 ***Training*** Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.979999999	218	0.0
1	0.9799999991	217	0.0
0.01	0.9799999989	216	0.0
Dengan 20 *Training* Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.979999999	218	0.0
1	0.979999999	216	0.0
0.01	0.9799999988	215	0.0

Tabel 4. Hasil pengujian dengan persamaan $x^4 + x^3 - 0.0324x - 0.00104976 = 0$

Dengan 10 ***Training*** Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.1839713339	8865	0.0
1	0.1839713265	8875	0.0
0.01	0.1839713339	8870	0.0
Dengan 20 *Training* Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.1839713339	8865	0.0
1	0.1839713265	8875	0.0
0.01	0.1839713339	8870	0.0

Tabel 5. Hasil pengujian dengan persamaan $x^4 + x^3 - 0.3364x - 0.11316496 = 0$

Dengan 10 ***Training*** Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.5799851742	41	0.0
1	0.5799851742	33	0.0
0.01	0.5799851742	34	0.0
Dengan 20 *Training* Data			
U1	**X akhir**	**Iterasi**	***Error***
-0.5	0.5799851742	41	0.0
1	0.5799851742	33	0.0
0.01	0.5799851741	32	0.0

Tabel 6. Hasil pengujian dengan persamaan $x^4 + x^3 - 0.9604x - 0.92236816 = 0$

Dengan 10 *Training* Data			
U1	X akhir	Iterasi	*Error*
-0.5	0.980001311	21	0.0
1	0.980001311	19	0.0
0.01	0.980001311	19	0.0
Dengan 20 *Training* Data			
U1	X akhir	Iterasi	*Error*
-0.5	0.9800013111	20	0.0
1	0.9800013111	20	0.0
0.01	0.980001311	21	0.0

Pada Tabel 1 sampai 6 menunjukkan hasil akar persamaan polinomial dari jumlah data pelatihan 10 dan 20 dan jumlah data pengujian 0.18, 0.58, 0.98.

Tabel 7. Hasil pengujian Hopefield Modifikasi dengan persamaan $x^2 - 0.0324 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	*Error*
-2.05	-0.5	0.1800000046	852	0.0
0.11	-0.5	0.1800000045	840	0.0
1.65	-0.5	-	-	-
-2.05	1	0.1800000045	854	0.0
0.11	1	0.1800000045	845	0.0
1.65	1	-	-	-
-2.05	0.01	0.1800000042	857	0.0
0.11	0.01	0.1800000046	842	0.0
1.65	0.01	-	-	-

Tabel 8. Hasil pengujian Hopefield Modifikasi dengan persamaan $x^2 - 0.3364 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	*Error*
-2.05	-0.5	0.5800000002	64	0.0
0.11	-0.5	0.5799999999	57	0.0
1.65	-0.5	0.5799999999	6514	0.0
-2.05	1	0.5800000002	65	0.0
0.11	1	0.5800000001	54	0.0
1.65	1	0.5799999999	1507	0.0
-2.05	0.01	0.5800000001	66	0.0
0.11	0.01	0.5799999999	53	0.0
1.65	0.01	0.5799999998	3938	0.0

Tabel 9. Hasil pengujian Hopefield Modifikasi dengan persamaan $x^2 - 0.9604 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	*Error*
-2.05	-0.5	0.9800000012	326	0.0
0.11	-0.5	0.9799999989	217	0.0
1.65	-0.5	0.9799999989	510	0.0
-2.05	1	0.9800000013	344	0.0
0.11	1	0.979999999	216	0.0
1.65	1	0.9799999989	285	0.0
-2.05	0.01	0.980000001	335	0.0
0.11	0.01	0.9799999988	215	0.0
1.65	0.01	0.9799999991	2397	0.0

Tabel 10. Hasil pengujian Hopefield Modifikasi dengan persamaan $x^4 + x^3 - 0.0324x - 0.00104976 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	*Error*
-2.05	-0.5	-	-	-
0.11	-0.5	0.1839713339	8865	0.0
1.65	-0.5	-	-	-
-2.05	1	-	-	-
0.11	1	0.1839713265	8875	0.0
1.65	1	-	-	-
-2.05	0.01	-	-	-
0.11	0.01	0.1839713339	8870	0.0
1.65	0.01	-	-	-

Tabel 11. Hasil pengujian Hopefield Modifikasi dengan persamaan $x^4 + x^3 - 0.3364x - 0.11316496 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	*Error*
-2.05	-0.5	-	-	-
0.11	-0.5	0.5799851742	41	0.0
1.65	-0.5	-	-	-
-2.05	1	-	-	-
0.11	1	0.5799851741	31	0.0
1.65	1	-	-	-
-2.05	0.01	-	-	-
0.11	0.01	0.5799851742	34	0.0
1.65	0.01	-	-	-

Tabel 12. Hasil pengujian Hopefield Modifikasi dengan persamaan $x^4 + x^3 - 0.9604x - 0.92236816 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	*Error*
-2.05	-0.5	-	-	-
0.11	-0.5	-	-	-
1.65	-0.5	-	-	-
-2.05	1	-	-	-
0.11	1	0.9800013111	22	0.0
1.65	1	-	-	-
-2.05	0.01	-	-	-
0.11	0.01	-	-	-
1.65	0.01	-	-	-

Pada Tabel 7 sampai 12 menunjukkan hasil pengujian dengan metode Hopfield Modifikasi. Hasil akar persamaan polinomial dari data pengujian 0.18, 0.58, 0.98.

Tabel 13. Hasil pengujian Newton Raphson dengan persamaan $x^2 - 0.0324 = 0$

Metode Newton Raphson			
X1	X akhir	Iterasi	*Error*
-2.05	-	-	-
0.11	0.18	5	0.0
1.65	0.18	8	0.0

Tabel 14. Hasil pengujian Newton Raphson dengan persamaan $x^2 - 0.3364 = 0$

Metode Newton Raphson			
X1	X akhir	Iterasi	*Error*
-2.05	-	-	-
0.11	0.58	7	0.0
1.65	0.58	6	0.0

Tabel 15. Hasil pengujian Newton Raphson dengan persamaan $x^2 - 0.9604 = 0$

Metode Newton Raphson			
X1	X akhir	Iterasi	*Error*
-2.05	-	-	-
0.11	0.98	8	5.66440318e-17
1.65	0.98	5	5.66440318e-17

Tabel 16. Hasil pengujian Newton Raphson dengan persamaan $x^4 + x^3 - 0.0324x - 0.00104976 = 0$

Metode Newton Raphson			
X1	X akhir	Iterasi	*Error*
-2.05	-	-	-
0.11	0.18	8	-6.62617204e-15
1.65	0.18	11	-1.99055818e-15

Tabel 17. Newton Raphson Hasil pengujian dengan persamaan $x^4 + x^3 - 0.3364x - 0.11316496 = 0$

Metode Newton Raphson			
X1	X akhir	Iterasi	*Error*
-2.05	-	-	-
0.11	-	-	-
1.65	0.58	8	2.86484918e-17

Tabel 18. Newton Raphson Hasil pengujian dengan persamaan $x^4 + x^3 - 0.9604x - 0.92236816 = 0$

Metode Newton Raphson			
X1	X akhir	Iterasi	*Error*
-2.05	-	-	-
0.11	-	-	-
1.65	0.98	6	-1.34736562e-15

Pada Tabel 13 sampai 18 menunjukkan hasil pengujian dengan metode Newton Raphson. Hasil akar persamaan polinomial dari data pengujian 0.18, 0.58, 0.98.

V. SIMPULAN

Berdasarkan hasil penelitian yang dilakukan metode kombinasi *Backpropagation* dan Hopfield Modifikasi telah berhasil dilakukan. Tingkat akurasi tersebut dipengaruhi oleh beberapa faktor yaitu pemilihan data pelatihan, nilai awal yang diberikan pada Hopfield Modifikasi terjamin dekat dengan nilai akar data pengujian. Konvergensi metode *Backpropagation* dapat terjebak di dalam minimum lokal. Sehingga dikombinasi dengan metode Hopfield Modifikasi dapat keluar dari konvergensi yang terjebak di dalam minimum lokal.

Kemudian dibandingkan dengan metode Hopfield Modifikasi saja. Hasil pengujian didapat jika diberikan nilai awal jauh dari nilai akar dan kompleksnya persamaan polynomial, tidak mendapatkan hasil yang akurat. Kemudian dibandingkan dengan metode Newton Raphson. Hasil pengujian didapat nilai akar akan konvergensi jika dekat dengan nilai solusinya.

Karena nilai awal yang diberikan terlalu jauh, menyebabkan terjadinya iterasi divergen.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Amikom Yogyakarta yang telah memberikan dana penelitian.

DAFTAR PUSTAKA

[1] D. Mishra, P. K. Kalra, *Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations*, Engineering Letters, 14:1, EL_14_1_23 (Advance online publication: 12 February 2007).

[2] D. Puspitaningrum, *Pengantar Jaringan Syaraf Tiruan.* Yogyakarta: Andi, 2006.

[3] D. Ratnawati, *Penerapan JST dengan Metode Backpropagation Studi Kasus : Prakiraan Cuaca dan Kualitas Udara Wilayah DKI Jakarta*, *Tesis*, Program Pascasarjana, Universitas Gadjah Mada, Yogyakarta. 2008.

[4] J. Ding, J. E. Gower, dan D. S. Schmidt, 2006, Multivariate Public Key Cryptosystem, Springer, USA. [13]

[5] L. Fausett, 1994, *Fundamentals of Neural Networks (Architectures, Algorithms, and Applications)*, Prentice Hall, Englewood Cliffs, New Jersey.

[6] M. M. Gupta, L. Jin, N. Homma, 2003, *Static and Dynamic Neural Networks, From Fundamentals to Advanced Theory*, John Wiley & Sons, Inc., Hoboken, New Jersey.

[7] M. Z. Riyanto, "Sistem Kriptografi Kunci Publik Multivariat", Seminar Nasional Matematika dan Pendidikan Matematika, 27 November 2010 di Jurusan Pendidikan Matematika FMIPA UNY.

[8] O. Billet, and J. Ding, "Overview of Cryptanalysis Techniques in Multivariate Public Key Cryptography", DOI 10.1007/978-3-540-93806-4_15, Springer-Verlag Berlin Heidelberg, 2009.

[9] R. Munir, *Metode Numerik.* Bandung: Informatika, 2008.

[10] U. M. Kaczmar, and T. Switek, "Combined Unsupervised-Supervised Classification Method", J.D. Vel´asquez et al. (Eds.): KES 2009, Part II, LNAI 5712, pp. 861–868, 2009.

[11] X. Wang, B. Yang, "*An Improved Signature Model of Multivariate Polynomial Public Key cryptosystem Against Key Recovery Attack", DOI: 10.3934/mbe.2019388, http://www.aimspress.com/journal/MBE, 2019.*

[12] Y. Shin, J. Ghosh, 1991, *Approximation of Multivariate Functions Using Ridge Polynomial Networks*, Department of Electrical and Computer Engineering, The University of Texas at Austin, 1991.