

Estimasi Biaya Proyek Perangkat Lunak Menggunakan JST dan Algoritma Genetika

Estimation of Project Software Costs Using ANN and Genetic Algorithms

As'ary Ramadhan^{*1}, Agus Sihabuddin², Azhari S. N³

¹Program Studi S2 Ilmu Komputer, FMIPA UGM, Yogyakarta, Indonesia

^{3,4}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *1as.ary29@gmail.com, 2a_sihabudin@ugm.ac.id, 3arisn@ugm.ac.id

Abstrak

Estimasi biaya pengembangan proyek perangkat lunak merupakan salah satu masalah yang kritis dalam rekayasa perangkat lunak. Kegagalan dari proyek perangkat lunak diakibatkan ketidak akuratannya estimasi sumber daya yang dibutuhkan. Beberapa model telah dikembangkan dalam beberapa dekade ini. Untuk memberikan keakuratan dalam estimasi biaya proyek perangkat lunak masih menjadi tantangan tersendiri.

Tujuan dilakukannya penelitian ini meningkatkan akurasi estimasi biaya proyek perangkat lunak dengan menerapkan algoritma genetika sebagai proses pelatihan pada ANN yang mengakomodasi formula dari *Post Architecture Model* (COCOMO II). COCOMO II merupakan model berbasis regresi yang digunakan untuk estimasi biaya proyek perangkat lunak. *Dataset* COCOMO adalah yang biasa digunakan untuk melakukan pelatihan dan pengujian pada jaringan. *Magnitude of Relative Error* (MRE) dan *Mean Magnitude of Relative-Error* (MMRE) digunakan sebagai pengukuran indikasi kinerja. Hasil percobaan menunjukkan bahwa model yang diusulkan memberikan hasil estimasi biaya proyek perangkat lunak menjadi lebih akurat. Dalam kasus ini MMRE untuk COCOMO II adalah 73.01%, FFNN-BP adalah 39.90% dan FFNN-GA adalah 31.48%.

Kata kunci: Jaringan Syaraf Tiruan, Backpropagation, Algoritma Genetika, COCOMO II, Proyek Manajemen IT.

Abstract

The cost estimation of software project development is one critical problem in software engineering. The failure of a software project is caused by inaccurate needed resources. Several models have been improved in the recent decade. Giving the accuracy in software project cost estimation is still becoming a challenge.

The purpose of this research is increasing the accuracy of the software project cost estimation with practicing genetic algorithm as a training process in ANN accomodating the formula from Post Architecture Model (COCOMO II). COCOMO II is a model based on regration used in software project cost estimation. COCOMO dataset is usually used for training and testing in a network. The magnitude of Relative Error (MRE) and Mean Magnitude of Relative-Error (MMRE) are used as the occupation indicator measuring. The result of the experiment reveals that the model investigated shows the result from the software project cost estimation to be more accurate. In the case of The Mean Magnitude of Relative error (MMRE) for the entire validation set was 73.01% for COCOMO II, 39.90 % for The FFNN-BP and 31.48% for The FFNN-GA

Keywords: Artificial Neural Network (ANN), Backpropagation, Genetic Algorithm, COCOMO II, IT Project Management.

1. Pendahuluan

Estimasi biaya pengembangan proyek perangkat lunak merupakan salah satu masalah yang kritis dalam rekayasa perangkat lunak. Biaya pengembangan proyek perangkat lunak berkaitan dengan waktu serta jumlah orang yang dibutuhkan untuk menyelesaikan sebuah proyek perangkat lunak. Kegagalan dari proyek perangkat lunak sering disebabkan karena tidak akuratnya estimasi sumber daya yang dibutuhkan (Charette, 2005).

Post-Architecture model (COCOMO II) adalah model berbasis regresi untuk estimasi biaya proyek perangkat lunak yang telah dikembangkan oleh Boehm pada tahun 1981 (Boehm, dkk., 1995) Model COCOMO II lebih baik dari COCOMO (Attarzadeh dan Ow, 2010). Beberapa peneliti melakukan pendekatan dengan menerapkan *Artificial neural network* (ANN) atau jaringan syaraf tiruan berbasiskan COCOMO II, seperti yang dilakukan Kaushik, Attarzadeh, dan Tadayon (Kaushik, dkk., 2012; Attarzadeh dan Ow, 2010; Tadayon, 2005). Pendekatan ini dilakukan karena ANN lebih fleksibel dalam mengintegrasikan *expert knowledge* ke dalam model sehingga dapat didesain sedemikian rupa untuk mengakomodasi COCOMO II (Kaushik, dkk., 2012). ANN dapat diimplementasikan dengan baik pada permasalahan terkait hubungan antar variabel atau dengan pendekatan konvensional dimana keterkaitan antar variabel sulit untuk digambarkan secara memadai (Dave, dkk., 2014).

Algoritma genetika adalah alternatif metode untuk *training neural network* (Chen dkk., 2014). Algoritma genetika ini lebih unggul dalam efisiensi terhadap ruang pencarian yang besar (*search space*), adanya rekombinasi antar individu sehingga didapatkan hasil yang lebih baik pada iterasi berikutnya, dan tidak sensitif terhadap masalah lokal optima.

Algoritma genetika yang diterapkan pada penelitian ini membantu proses pelatihan pada ANN yang telah dikostumisasi arsitekturnya sehingga dapat mengakomodasi model COCOMO II. Alasan pemilihan kombinasi algoritma genetika dengan jaringan syaraf tiruan *backpropagation* adalah untuk mempercepat pencapaian titik konvergensi pada proses pelatihan dan memperoleh nilai estimasi yang lebih mendekati nilai aktual.

Berdasarkan latar belakang yang telah dipaparkan, permasalahan yang diangkat adalah bagaimana mendapatkan nilai estimasi biaya proyek perangkat lunak yang lebih baik dari segi akurasi dengan menerapkan algoritma genetika pada proses pelatihan ANN yang telah disesuaikan arsitekturnya untuk mengakomodasi COCOMO II terhadap dataset COCOMO.

2. Metode Penelitian

2.1 Algoritma Genetika

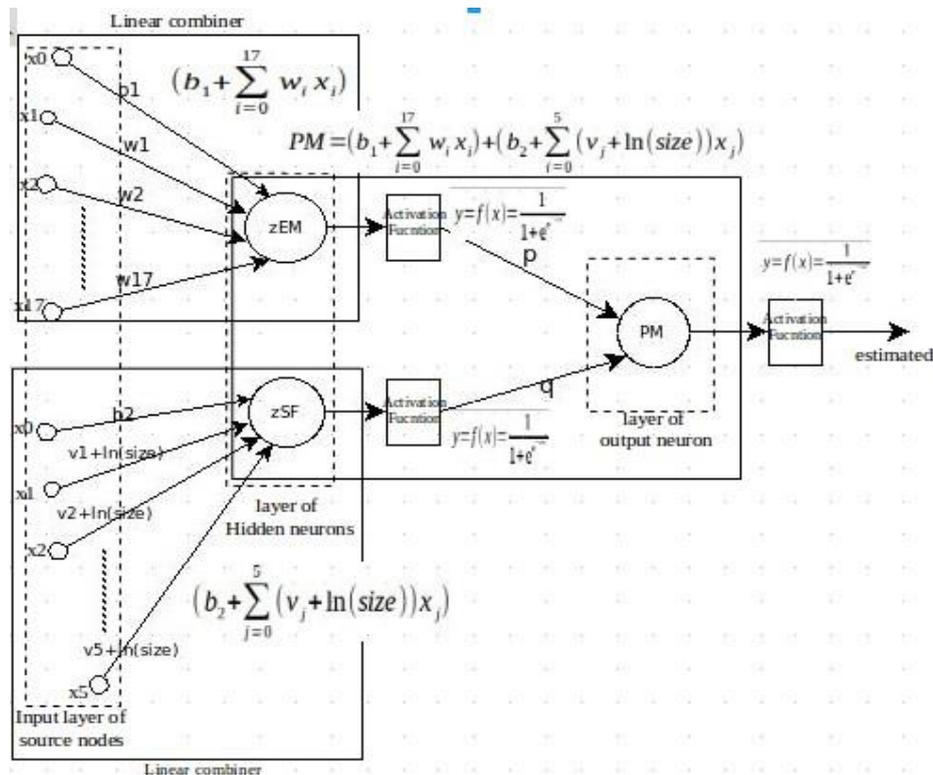
Algoritma genetika adalah algoritma pencarian *stochastic* berdasarkan pada mekanisme dari seleksi alami dan genetik alami (Gen dan Cheng, 1997). Algoritma genetika mengadopsi prinsip teori evolusi Darwin yang menyatakan bahwa kelangsungan hidup suatu makhluk dapat dipertahankan melalui proses reproduksi, *crossover*, dan mutasi (Hermawanto, 2007). Dalam tiap generasi kromosom dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan menggunakan ukuran yang disebut dengan nilai *fitness*. Adanya nilai *fitness* ini dapat menjadi gambaran kualitas individu untuk menjadi dasar optimasi algoritma genetika.

2.2 Jaringan Syaraf Tiruan

Jaringan syaraf merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk menjalankan simulasi proses pembelajaran pada otak (Kusumadewi dan Hartati, 2010). Jaringan syaraf terdiri dari komponen dan pembobotan yang terhubung, setiap lapisan pada jaringan syaraf terdiri dari sekumpulan komponen yang menerima input data yang disebut *Neuron*. *Neuron* adalah unit pemrosesan informasi (*information-processing unit*) yang merupakan fundamental terhadap operasi dari sebuah jaringan syaraf (Haykin, S. 2009). Masing masing neuron pada jaringan syaraf memiliki fungsi aktivasi untuk menentukan sinyal keluaran dari *neuron* tersebut untuk digunakan sebagai keluaran akhir atau masukan pada *neuron* lainnya.

2.3 Rancangan FFNN-BP

Arsitektur yang dibentuk memiliki struktur yang koneksi jaringannya tidak terhubung sepenuhnya (*partially connected*).Arsitektur dipisahkan menjadi dua domain yaitu z^{EM} dan z^{SF} sebagai lapisan tersembunyi yang digunakan untuk menghitung 17 nilai-nilai *Cost Driver* dan 5 nilai dari *Scaling Factor* yang dapat dilihat pada Gambar 1. Dimana $PM = \ln(PM)$, $X_i = \ln(EM_i)$ dan $X_j = SF_i$



Gambar 1. Rancangan Arsitektur FFNN-BP

2.4 Rancangan FFNN-GA

Bobot dan bias dalam jaringan FFNN digambarkan oleh algoritma genetika ke dalam suatu individu. Cara melakukan konversi bias dan bobot pada jaringan syaraf tiruan ke dalam bentuk individu pada algoritma genetik adalah melakukan seleksi, *crossover*, dan mutasi terhadap bobot *input*, bias, dan bobot *output* yang ada pada jaringan syaraf sehingga menjadi suatu individu yang digunakan pada iterasi pelatihan selanjutnya seperti.

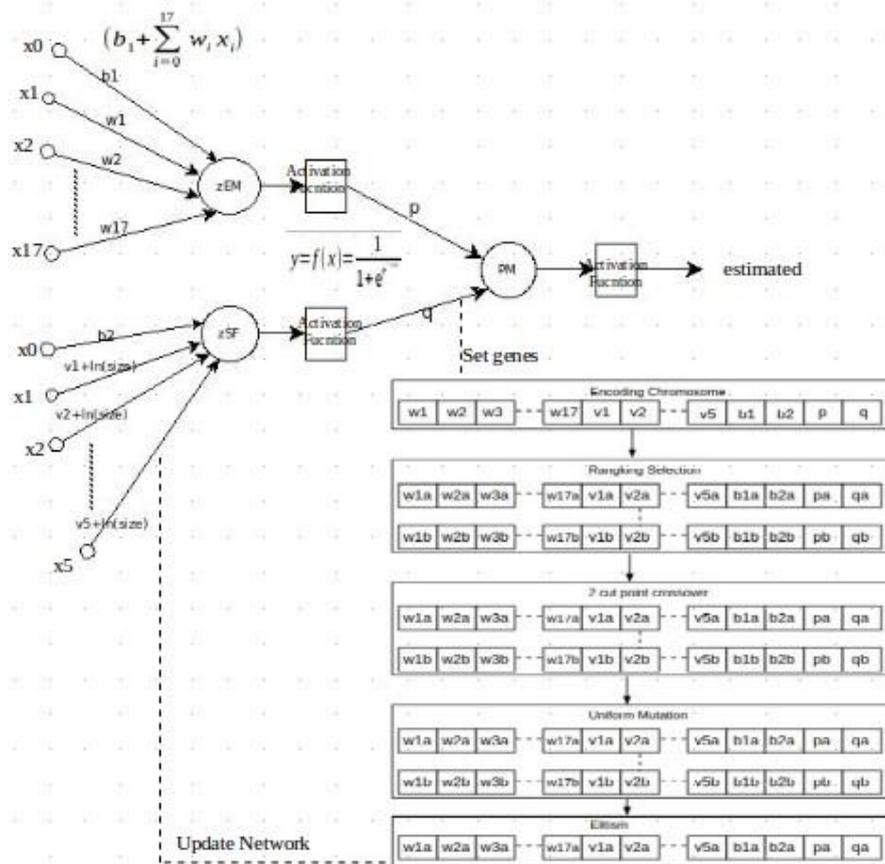
2.5 Metode Evaluasi

Metode evaluasi diperlukan untuk menentukan seberapa akurat saat penggunaan suatu model atau teknik. Penelitian ini menggunakan nilai *Magnitude of Relative Error* (MRE) maupun *Mean Magnitude of Relative Error* (MMRE). MRE merupakan pengukuran yang menunjukkan perbedaan antara nilai estimasi yang disarankan dan nilai aktual estimasi. Semakin kecil nilai MRE, maka semakin kecil pula jarak antara nilai aktual dengan nilai prediksi. MRE dari setiap estimasi didefinisikan pada Persamaan (1)

$$MRE_i = \frac{|ActualEffort_i - EstimatedEffort_i|}{ActualEffort_i} \times 100\% \tag{1}$$

MMRE adalah rata-rata dari nilai absolut *mean relative error* (MRE) terhadap sekumpulan data lengkap yang didefinisikan sebagai Persamaan (2). variable N menunjukkan jumlah data sedangkan MRE merupakan nilai MRE dari setiap data.

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i \tag{1}$$



Gambar 4. Rancangan Arsitektur FFNN-GA

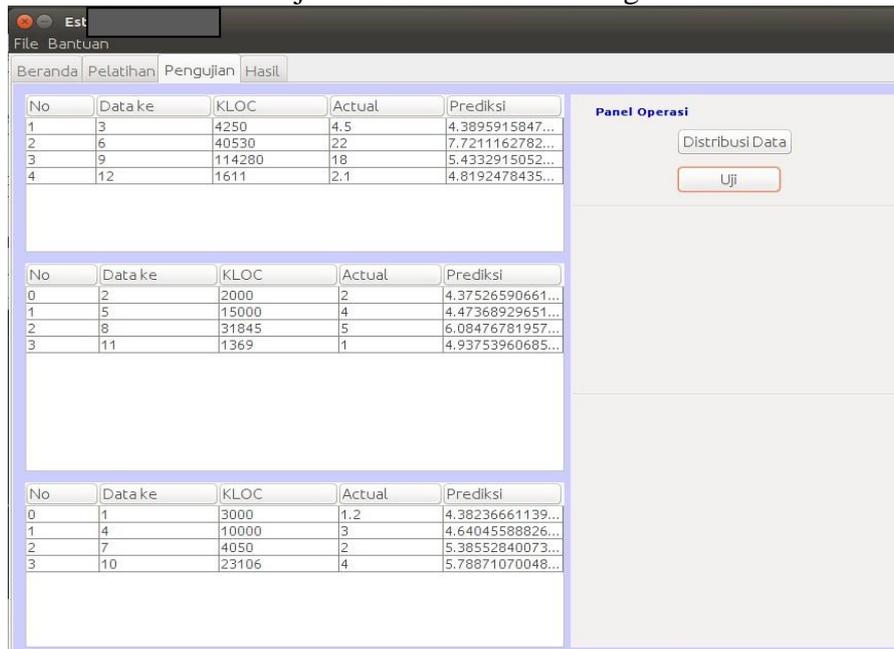
2.6 Data Penelitian

Data input berasal dari dataset COCOMO_SDR yang didapat dari <http://openscience.us/repo/effort/cocomo/cocomosdr.html>. Perlakuan sistem terhadap dataset melibatkan proses *cross validation* yang merupakan proses pembagian data latihan

ke dalam beberapa potongan yang nantinya digunakan sebagai data uji dari keseluruhan dataset.

2.7 Implementasi Sistem

Hasil analisis konseptual menggunakan diagram-diagram perancangan sistem pada algoritma genetika untuk melatih jaringan syaraf digunakan sebagai dasar implementasi. Implementasi sistem dibuat menggunakan perangkat lunak pemrograman JAVA (Netbeans 7.0) dan Mysql database untuk melakukan pengujian dan menyimpan data dan hasil analisis. Gambar 2 menunjukkan sistem telah dibangun.



Gambar 2. Implementasi Sistem Pengujian

3. Hasil dan Pembahasan

3.1 Pengujian Sistem

Pengujian sistem adalah proses pelatihan secara berulang-ulang untuk mendapatkan bobot-bobot yang terbaik berdasarkan evaluasi dengan nilai MRE terkecil atau mendekati angka 0 untuk menyelesaikan masalah estimasi biaya proyek perangkat lunak. Pengujian sistem terdiri pembagian dataset menjadi 3 *fold* $k=3$ dengan rasio 70:30 untuk pelatihan dan pengujian validasi data. Kemudian inialisasi proses sistem untuk menyesuaikan parameter yang diperlukan berdasarkan percobaan (*trial and error*) sampai dengan proses estimasi selesai. data .

3.2 Inialisasi Parameter

Inialisasi parameter dilakukan pada dua sistem yang memiliki algoritme pelatihan yang berbeda yaitu sistem dengan penerapan ANN berdasarkan pendekatan COCOMO II yang dilatih dengan algoritma *backpropagation* maupun algoritma genetika. Hasil perolehan terhadap percobaan pengujian parameter dapat dilihat pada Tabel 1. Hasil ini merupakan parameter terbaik yang diambil dari 50 kali percobaan yang telah dilakukan.

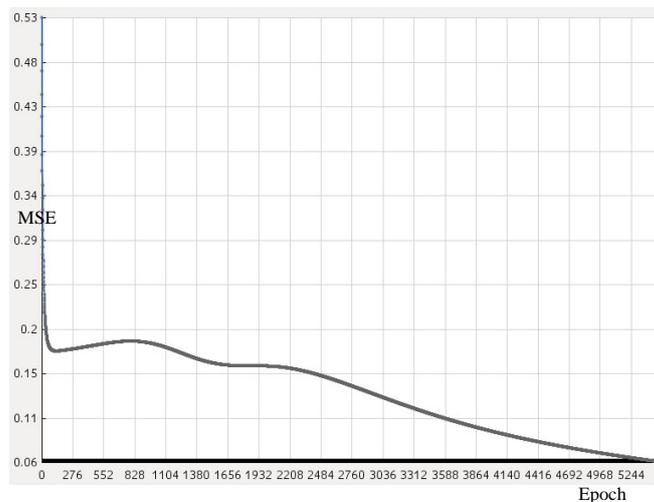
Tabel 1 Inisialisasi Parameter

No	ANN-BP				ANN-GA					
	∂	Max epoch	Error Toleransi	MSE	Pop	pcross (%)	pmut (%)	Max Epoch	Error Toleransi	MSE
1.	0.1	5500	1.0*10E-06	0.064	3000	26	10	70	1.0*10E-06	0.047

Dimana ∂ disimbolkan sebagai nilai dari laju pembelajaran (*learning rate*), Max *epoch* merupakan jumlah *epoch* yang digunakan selama iterasi berlangsung, toleransi *error* merupakan kondisi berhentinya iterasi program, waktu adalah lamanya pelatihan data, pop merupakan nilai ukuran populasi yang dibangkitkan, pmut menunjukkan bahwa 10% dari populasi akan dipilih untuk dilakukan proses perkawinan. pcross 26 % menunjukkan bahwa probabilitas dalam satu generasi ada beberapa generasi yang mengalami *crossover*.

3.3 Pengujian FFNN-BP

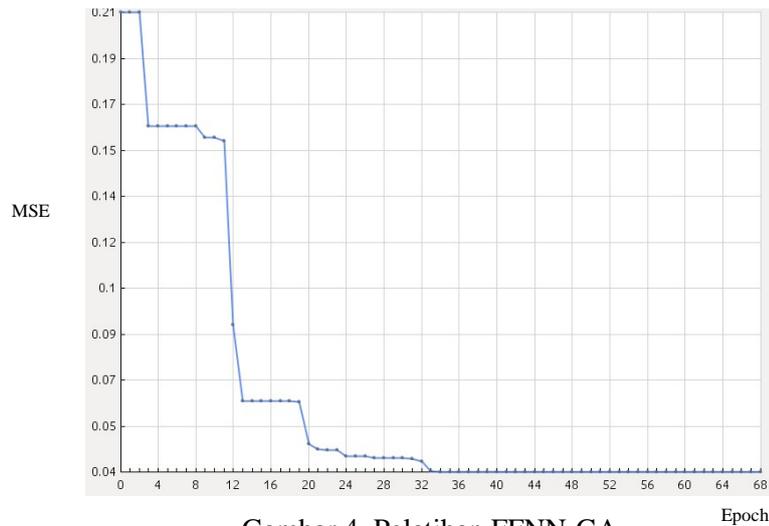
Algoritma *backpropagation* bekerja dengan cara mengupdate masing masing bobot pada sebuah jaringan secara *backward pass* sampai nilai *actual output* mendekati nilai *target output*. Dengan artian meminimalkan atau menurunkan nilai *error* untuk masing masing neuron *output* dan jaringan secara keseluruhan. Gambar 3 menunjukkan tampilan secara grafis penurunan nilai MSE pelatihan *Backpropogation* terhadap dataset *cocomo_sdr* menggunakan nilai parameter pada Tabel 1.



Gambar 3. Pelatihan FFNN-BP

3.4 Pengujian FFNN-GA

Berdasarkan uji coba dapat diketahui bahwa kemampuan individu dalam memproduksi dibatasi oleh probabilitas–probabilitas populasi, *epoch*, *mating* dan *mutase* yang sangat mempengaruhi dalam meminimalkan nilai *fitness* (MSE).



Gambar 4. Pelatihan FFNN-GA

Pelatihan algoritma terhadap dataset *cocomo_sdr* dengan inialisasi paramater pada Tabel 1 ditunjukkan pada Gambar 4. Dari gambar tersebut terlihat bahwa pelatihan dengan algoritma genetika tidak sensitif terhadap masalah lokal optima dan cepat konvergen.

3.5 Analisis Hasil Pengujian

Proses pelatihan dengan menggunakan inialisasi parameter yang ditunjukkan pada Tabel 1. Hasil perbandingan uji coba nilai estimasi usaha proyek perangkat lunak menggunakan penerapan FFNN berdasarkan pendekatan COCOMO II yang dilatih dengan algoritme Backpropagation dan penerapan FFNN berdasarkan pendekatan COCOMO II yang dilatih dengan algoritma genetika pada dataset *cocomo_sdr* ditunjukkan pada Tabel 2.

Tabel 2 Perbandingan Nilai Cost pada COCOMO_SDR

No Project	Nilai Aktual	COCOMO II	FFNN-BP	FFNN-GA	MSE-BP	MSE-GA
1	1.2	5.87	1.34	2.20	0.019939	1.007268
2	2	4.38	1.30	2.20	0.485380	0.041464
3	4.5	5.59	1.30	2.20	10.235789	5.273326
4	3	2.47	5.48	2.20	0.027108	0.633858
5	4	1.43	3.56	2.35	0.189332	2.719082
6	22	1.67	19.62	21.97	5.657558	0.000446
7	2	1.50	4.37	2.21	5.625108	0.048136
8	5	7.59	5.20	5.69	0.041012	0.476252
9	18	8.31	17.51	16.93	0.234984	1.143718
10	4	3.79	3.30	3.59	0.487155	0.165525
11	1	1.26	1.49	2.20	0.247169	1.448742
12	2.1	1.52	3.46	2.20	1.854207	0.012086

Tidak semua *project id* mendapatkan nilai akurasi yang mendekati nilai aktual karena hasil pelatihan dapat memiliki bobot dan bias akhir yang hanya cocok untuk nilai aktual rendah atau nilai aktual tinggi saja.

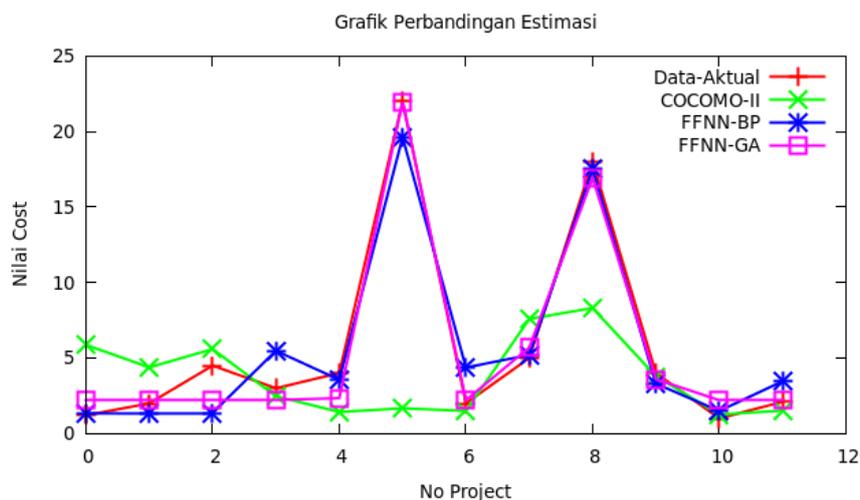
Pada Tabel 2 dapat dilihat bahwa perbandingan *output* pada No Project 2, No Project 6, serta No Project 12 berdasarkan hasil pengujian FFNN yang dilatih dengan algoritma genetika memiliki akurasi yang lebih baik daripada FFNN dilatih dengan *Backpropagation* dengan nilai MSE masing masing yaitu No Project 2 = 0.041464, No

Project 6 = 0.000446, serta No Project 12 = 0.012086. Untuk nilai MRE pada keseluruhan *Project* dapat dilihat Pada Tabel 3.

Tabel 3 Perbandingan MRE COCOMO_SDR

No Project	MRE(%) COCOMO	MRE(%) FFNN-BP	MRE(%) FFNN-GA
1	389,17	11,67	83,33
2	119	35,00	10,00
3	24,22	71,11	51,11
4	17,67	82,67	26,67
5	64,25	11,00	41,25
6	92,41	10,82	0,14
7	25	119	10,50
8	51,80	4,00	13,80
9	53,83	2,72	5,94
10	5,25	17,50	10,25
11	26	49	120
12	27,62	64,76	4,76

Sedangkan Gambar 5 menunjukkan tampilan grafis hasil perbandingan nilai *effort* pada Tabel 3. Nilai MMRE pada keseluruhan data yang ada pada Tabel 3 untuk COCOMO II adalah 73.01 %, FFNN-BP adalah 39.90% dan FFNN-GA adalah 31.48%.



Gambar 5 Grafik Perbandingan MRE COCOMO_SDR

4. Kesimpulan

Parameter pada ANN dan GA sangat berpengaruh pada hasil estimasi berdasarkan evaluasi menggunakan MRE dan MMRE. Dari 50 kali percobaan maka didapatkan parameter pelatihan terbaik untuk ANN 0,1 untuk learning rate, 0,000001 toleransi error, maximum epoch 5500, dan untuk GA probabilitas crossover (26%), probabilitas mutasi (10%), maximum epoch 70 dan populasi (size) 3000. Algoritma genetika sebagai proses FFNN yang telah disesuaikan arsitekturnya untuk memetakan model COCOMO II dapat memberikan hasil akurasi yang baik berdasarkan evaluasi nilai error antara data aktual dan data target menggunakan parameter pelatihan terbaik dengan perolehan nilai MMRE untuk COCOMO II adalah 73.01 %, FFNN-BP adalah 39.90% dan FFNN-GA adalah 31.48%.

Daftar Pustaka

- Attarzadeh, I, and Ow, S, H., 2010, A novel soft computing model to increase the accuracy of software development cost estimation, *Computer and Automation Engineering (ICCAE), The 2nd International Conference on.*, 3, 603-607
- Boehm, B, Clark, B, Horowitz, B, Madachy, E, Shelby, R, Westland, R, and Chris., 1995, *Cost Models for Future Software Life Cycle Processes: COCOMO 2.0*, *Annals of Software Engineering*, 1, 57-94
- Charette, 2005, We waste billions of dollars each year on entirely preventable mistakes, *IEEE Spectrum*, 42-49.
- Chen, Wen, C, T, Lin YuWu, and Sheng, C, 2014, A unified evolutionary training scheme for single and ensemble of feedforward neural network, *Neurocomputing*, 143, 347-361.
- Dave, Dutta, V, S, and Kamlesh, 2014, Neural network based models for software effort estimation: A review, *Artificial Intelligence Review*, 42, 295-307.
- Gen, Mitsuo, and Cheng, Runwei, 1997, *Genetic Algorithms and Engineering Design*, WILEY, Japan.
- Haykin, S. 2009, *Neural Networks and Learning Machines*, 3rd., Pearson Prentice Hall, Hamilton, Ontario, Canada.
- Hermawanto, Denny, 2007, *Algoritma Genetika dan Contoh Aplikasinya*, 1-10.
- Kaushik, A Soni, a K, and Soni, R, 2012, An adaptive learning approach to software cost estimation, *Computing and Communication Systems (NCCCS), National Conference on*, 1-6.
- Kusumadewi, S, and Hartati, S., 2010, *Neuro-Fuzzy - Integrasi Sistem Fuzzy & Jaringan Syaraf*, Edisi 2, Graha Ilmu, Yogyakarta.
- Tadayon, N., 2005, Neural network approach for software cost estimation, *Proceedings of the International Conference on Information Technology: Coding and Computing*.