

Estimasi Pada *Effort* Perangkat Lunak dengan Pendekatan *Feed Forward Neural Network Backpropagation* (FFNN-BP)

As'ary Ramadhan ¹⁾

¹Fakultas Teknologi Informasi, Universitas Islam Kalimantan Muhammad Arsyad Al Banjari
Banjarmasin
e-mail: as.ary29@gmail.com

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 puluh tahun belakangan ini. Untuk meberikan keakuratan dalam estimasi biaya proyek perangkat lunak masih menjadi tantangan hingga saat ini. Tujuan dilakukannya penelitian ini meningkatkan akurasi estimasi biaya proyek perangkat lunak dengan menerapkan algoritma genetika sebagai proses pelatihan pada *Feed Forward Neural Network Backpropagation* (FFNN-BP) yang mengakomodasi formula dari Post Architecture Model (COCOMO II). 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 dari COCOMO II dan FFNN-BP. Dalam kasus ini MMRE untuk COCOMO II adalah 74.68%, FFNN-BP adalah 39.90% .

Kata kunci: *COCOMO II, Machine Learning, Proyek Manajemen IT, Backpropagation*

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).

ANN merupakan salah satu pendekatan untuk melakukan estimasi biaya proyek perangkat lunak yang banyak digunakan (Jorgensen et al, 2007). Sebagian besar penelitian dalam ranah estimasi biaya proyek perangkat lunak juga menekankan nilai pada model berbasis jaringan syaraf tiruan (Dave & Duta, 2014).

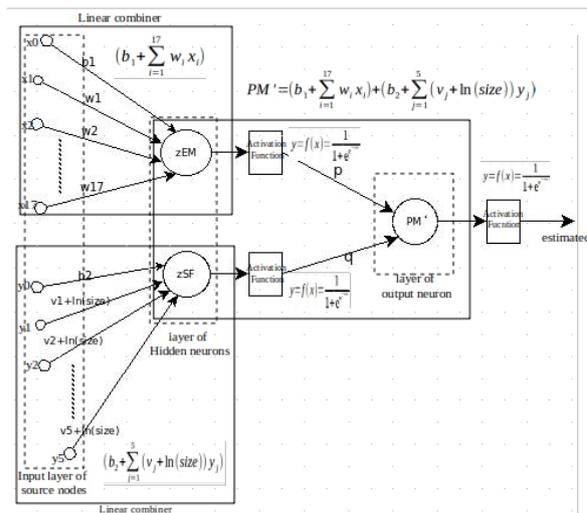
Berdasarkan latar belakang yang telah dipaparkan, permasalahan yang diangkat adalah bagaimana mendapatkan nilai estimasi biaya proyek perangkat lunak yang lebih baik dari segi akurasi dengan menerapkan algoritma genetika pada proses pelatihan *Feed Forward Neural Network Backpropagation* (FFNN-BP) yang telah disesuaikan arsitekturnya untuk

mengakomodasi COCOMO II terhadap dataset COCOMO.

2. METODE PENELITIAN

2.1 Rancangan FFNN-BP

Arsitektur yang dibentuk memiliki struktur yang koneksi jaringannya tidak terhubung sepenuhnya (*partially connected*). Arsitektur dipisahkan menjadi dua domain yaitu zEM dan zSF 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 $X_j = S_{fi}$.



Gambar 1. Rancangan Arsitektur FFNN-BP

2.2 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)

$$MRE_i = \frac{ActualEffort - EstimateEffort_i}{actualEffort_i} \quad (1)$$

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.

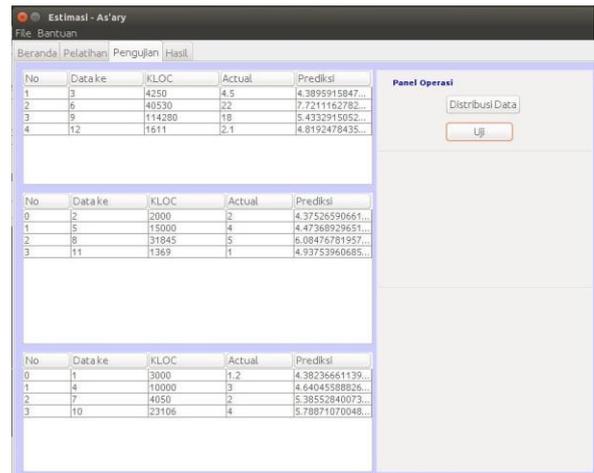
$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE \quad (2)$$

2.3 Data Penelitian

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

2.4 Implementasi Sistem

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

Proses dilakukan dengan mengklik tombol distribusi data maka akan tampil data-data yang telah dikenakan proses k-fold pada saat pelatihan dan melakukan pengujian validasi dengan mengklik tombol uji. hasil pelatihan ini selanjutnya digunakan sebagai dasar untuk proses pengujian serta validasi sebelum masuk ketahap proses 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 inisialisasi proses sistem untuk menyesuaikan parameter yang diperlukan berdasarkan percobaan (trial and error) sampai dengan proses estimasi selesai.

3.2 Inisialisasi Parameter

Inisialisasi parameter dilakukan pada dua sistem yang memiliki algoritme pelatihan yang berbeda yaitu sistem dengan penerapan FFNN-BP berdasarkan pendekatan COCOMO II yang dilatih dengan algoritma backpropagation. 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

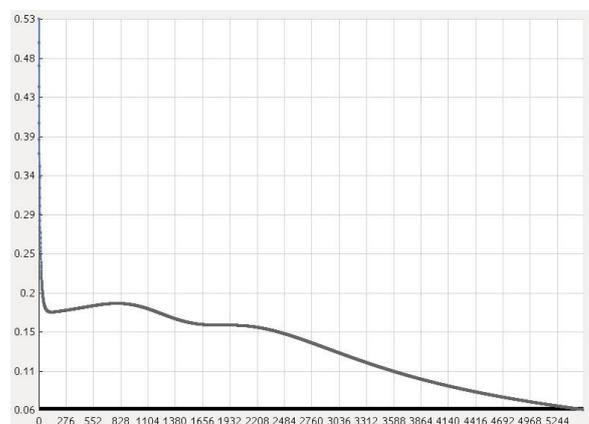
No	ANN-BP			
	∂	Max epoch	Error Toleransi	MSE
1.	0.1	5500	1.0*10E-06	0.064

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.

3.3 Pengujian FFNN-BP

Algoritma backpropagation bekerja dengan cara mengupdate masing masing bobot pada sebuah jaringan secara backward pass sampai nilai actual ouput 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. Proses pelatihan BP sangat mudah terjebak pada lokal minimum daru fungsi error, hal ini menyebabkan proses pelatihan mungkin dapat berhenti sebelum waktunya tanpa mencapai target error yang diinginkan.

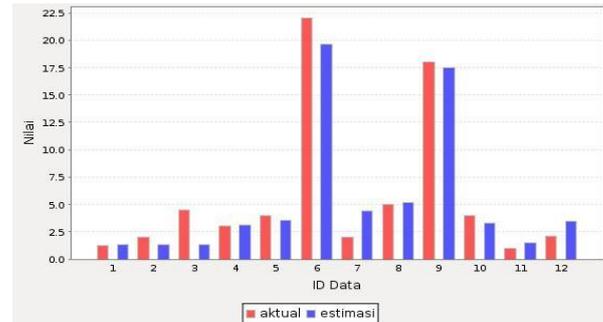


Gambar 3. Pelatihan FFNN-BP

pada Gambar 4. dapat dilihat bahwa beberapa proyek seperti Id 1, Id 4 serta Id 8, berdasarkan hasil pengujian FFNN yang dilatih dengan backpropagation memiliki nilai yang sangat mendekati nilai aktual dengan nilai MSE untuk Id 1 = 0.019939, Id 4 = 0.027108, dan Id 8 = 0.041012. Ada beberapa proyek dengan nilai MSE yang besar seperti Id 2, 3, 5, 7, 9, 11 dan 12.

No	Id	Aktual	Prediksi	MRE(%)	MSE(%)
1	3	4.5	1.30065791970531...	71.09649067321526	10.2357897467443...
2	6	22	19.62143770435241	10.8116467983981...	5.657558594276336
3	9	18	17.5152484137603...	2.69306436799791...	10.23498410036183...
4	12	2.1	3.4616929690883	64.84252233753809	11.85420774206450...
5	2	2	1.303307398725394	34.8346300637303	10.48538058067077...
6	5	4	3.56487678968362...	10.8780802579094...	10.18933220815603...
7	8	5	5.202515316363489	4.050306327269784	10.04101245336180...
8	11	1	1.49715109840425...	49.71510984042562	10.24715921464455...
9	1	1.2	1.34120601985829...	11.7671683215244...	10.01993914004422...
10	4	3	3.164647427176566	5.488247572552204	10.02710877527586...
11	7	2	4.371731067413843	118.586553370692...	5.625108256136007
12	10	4	3.30203460802790...	17.4491347993024...	10.48715568839076

Gambar 4. Pengujian FFNN-BP



Gambar 5. Perbandingan Nilai Effort

3.4 Analisis Hasil Pengujian

Proses pelatihan dengan menggunakan inisialisasi parameter yang ditunjukkan pada Tabel 1. Setelah dilakukan 50 kali percobaan terhadap metode maka didapatkan nilai estimasi biaya proyek perangkat lunak terbaik. Hasil perbandingan uji coba nilai estimasi usaha proyek perangkat lunak menggunakan penerapan FFNN berdasarkan pendekatan COCOMO II yang dilatih dengan algoritme Backpropagation pada dataset cocomo_sdr ditunjukkan pada Tabel 2. Berdasarkan hasil pengujian 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 saja atau nilai aktual tinggi saja/

Tabel 2. Perbandingan nilai Effort pada COCOMO_SDR

No Project	Nilai Aktual	COCOMO II	FFNN-BP
1	1.2	5.87	1.34
2	2	4.38	1.30
3	4.5	5.59	1.30
4	3	2.47	5.48
5	4	1.43	3.56
6	22	1.67	19.62
7	2	1.50	4.37
8	5	7.59	5.20
9	18	8.31	17.51
10	4	3.79	3.30
11	1	1.26	1.49
12	2.1	1.52	3.46

secara grafik hasil perbandingan dapat dilihat pada Gambar 5.

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.

4. KESIMPULAN

Parameter pada FFNN-BP sangat berpengaruh pada hasil estimasi berdasarkan evaluasi menggunakan MRE dan MMRE. Dari 50 kali percobaan maka didapatkan parameter pelatihan terbaik untuk FFNN-BP 0,1 untuk learning rate, 0,000001 toleransi error, maximum epoch 5500.

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% .

5. REFERENSI

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.
- 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.