

## ESTIMASI BIAYA PERANGKAT LUNAK MENGGUNAKAN IMPROVED BACKPROPAGATION

Rocky Yefrenes Dillak<sup>1)</sup>, Martini Ganantowe Bintiri<sup>2)</sup>, Azhari SN<sup>3)</sup>

<sup>1)</sup>Jurusan Teknik Informatika AMIKOM Yogyakarta  
Jl. Ring Road Utara Condong Catur Sleman Yogyakarta

<sup>2)</sup>Jurusan Teknik Sipil Universitas Sintuwu Maroso  
Jl. P.Timor No.1 Poso Sulawesi Tengah 94615

<sup>3)</sup>Jurusan Ilmu Komputer Fakultas MIPA Universitas Gadjah Mada  
Bulaksumur, Yogyakarta 55281, Telp (0274) 588688, Fax (0274) 565223

e-mail : [rocky\\_dillak@yahoo.com](mailto:rocky_dillak@yahoo.com), [gana75mart@yahoo.com](mailto:gana75mart@yahoo.com), [arism.softcom@yahoo.com](mailto:arism.softcom@yahoo.com)

### Abstrak

Estimasi biaya perangkat lunak merupakan bagian tak terpisahkan dari pengembangan perangkat lunak. Mengabaikan atau salah memprediksi biaya pengembangan perangkat lunak dapat menyebabkan rendahnya kualitas perangkat lunak yang dihasilkan. Untuk mengantisipasi terjadinya kesalahan prediksi biaya perangkat lunak maka dikembangkanlah suatu metode untuk menghitung atau memprediksi biaya perangkat lunak menggunakan algoritma jaringan syaraf tiruan *improved backpropagation*. Penelitian ini menggunakan data dari proyek NASA yang dikerjakan antara tahun 1980 dan 1990. Model yang dikembangkan ini dievaluasi menggunakan RMSSE. Penelitian ini menghasilkan RMSSE sebesar 3.72 % yang berarti bahwa model yang dikembangkan ini cukup handal untuk digunakan sebagai metode dalam estimasi biaya perangkat lunak.

**Kata Kunci :** estimasi biaya perangkat lunak, jaringan syaraf tiruan, *improved backpropagation*, *cocomo*, RMSSE.

### 1. PENDAHULUAN

Estimasi biaya perangkat lunak merupakan bagian tak terpisahkan dari pengembangan perangkat lunak. Estimasi biaya perangkat lunak sangat dibutuhkan dalam membuat usulan proposal, negosiasi kontrak, penjadwalan, kontrol dan monitoring. Untuk itu akurasi biaya perangkat lunak sangat dibutuhkan karena (Leung & Fan, 2001) :

- Dapat membantu untuk mengklasifikasi dan menentukan prioritas pengembangan proyek sehubungan dengan perencanaan bisnis secara keseluruhan.
- Dapat digunakan untuk menentukan *resources* yang sesuai dengan proyek yang dikerjakan dan seberapa baik *resources* ini akan digunakan.
- Dapat digunakan untuk menilai pengaruh terhadap perubahan dan dukungan dalam perencanaan ulang.
- Dapat memudahkan untuk mengatur dan mengontrol proyek ketika *resources* benar-benar sesuai dengan yang dibutuhkan.
- Dapat memenuhi harapan *customer* agar biaya pengembangan yang aktual dapat sejalan dengan biaya pengembangan yang diestimasi.

Penelitian estimasi biaya perangkat lunak telah banyak dilakukan baik menggunakan pendekatan konvensional (statistik) maupun pendekatan dengan *machine learning* (misal jaringan syaraf tiruan). Jaringan Syaraf Tiruan memiliki kelebihan dari metode lain yaitu mampu belajar dari pengalaman (data) historis (Bintiri, et al., 2012) sehingga dalam melakukan prediksi/estimasi dengan baik. Meskipun penggunaan jaringan syaraf tiruan (JST) dalam melakukan estimasi biaya memberikan kinerja yang cukup handal (Kaur, et al., 2010), namun JST memiliki satu masalah utama dalam proses pelatihan yakni *slow convergence* dan sering terjebaknya fase pelatihan JST pada lokal minima. Untuk mengatasi masalah tersebut maka dikembangkan proses perubahan bobot menggunakan metode momentum (Fausett, 1994) dimana penambahan momentum dapat mengatasi masalah *slow convergence*. Berdasarkan permasalahan tersebut maka pada penelitian ini, peneliti menggunakan JST *Improved Backpropagation* untuk melakukan estimasi biaya perangkat lunak.

Tujuan dari penelitian ini adalah mengkaji kemampuan dari JST *Improved Backpropagation* dalam melakukan estimasi biaya perangkat lunak sehingga JST *Improved Backpropagation* dapat dipakai sebagai model atau metode alternative dalam melakukan estimasi biaya perangkat lunak dengan hasil yang akurat dan dapat dipercaya.

Manfaat dari penelitian ini memberikan informasi kinerja dari JST *Improved Backpropagation* untuk estimasi biaya perangkat lunak dan menjadi bahan pertimbangan bagi para manajer dalam memilih metode / model estimasi yang akan digunakan dalam melakukan estimasi proyek yang akan dikerjakan. Hasil dan saran dari penelitian ini dapat menjadi bahan perbandingan dan masukan bagi penelitian lebih lanjut.

## 2. TINJAUAN PUSTAKA

Penelitian tentang estimasi biaya telah banyak dilakukan. Seperti yang dilakukan oleh Sameet, et al. (2009) yang memprediksi *software development effort* dengan metode Link Artificial Neural Network menggunakan *standart NASA 60 project dataset*.

Idri, Kooshgoftaar, dan Abran (2002) melakukan penelitian tentang interpretasi model *based on backpropagation three multilayer Perceptron network* menggunakan Cocomo 81 data set.

Kaur, et al. (2010) melakukan penelitian untuk mengetahui metode estimasi yang tepat untuk NASA *software project* dengan membandingkan antara Artificial Neural Network Based Model (ANN) dan Halstead, Walston-Felix, Bailey-Basili dan Doty model dengan hasil bahwa ANN lebih efektif dalam estimasi *effort*.

Kaur dan Verma (2012) melakukan penelitian untuk menghitung MMRE dan RMSSE dari *software effort* menggunakan JST *backpropagation* dibandingkan dengan model estimasi tradisional dengan hasil MMRE = 12,67 dan RMSSE = 18,59 lebih kecil dibandingkan dengan estimasi model tradisional. Dengan demikian JST *backpropagation* mempunyai kinerja yang lebih baik dibandingkan dengan model estimasi tradisional.

Dalam penelitian ini estimasi biaya perangkat lunak dilakukan menggunakan JST *improved backpropagation*. JST *improved backpropagation* adalah JST *backpropagation* yang dimodifikasi pada momentumnya. Pada momentum konvensional perubahan bobot didasarkan atas perubahan bobot satu iterasi sebelumnya, sedangkan pada *improved backpropagation* perubahan bobot didasarkan atas perubahan bobot saat ini, perubahan bobot satu iterasi sebelumnya dan perubahan bobot dua iterasi sebelumnya. Perubahan bobot dihitung dengan persamaan (Choudhary dan Rishi, 2011), sebagai berikut :

Perubahan bobot konvensional :

$$W_{ik}(t+1) = W_{ik}(t) + \eta \Delta W_{ik}(t) + \alpha \Delta W_{ik}(t-1) \quad (1)$$

Perubahan bobot dimodifikasi :

$$W_{ik}(t+1) = W_{ik}(t) + \eta \Delta W_{ik}(t) + \alpha \Delta W_{ik}(t-1) + \beta \Delta W_{ik}(t-2) \quad (2)$$

Dimana

$W_{ik}(t)$  adalah bobot matriks pada iterasi t.

$W_{ik}(t+1)$  adalah bobot matriks pada iterasi berikutnya.

$W_{ik}(t-1)$  adalah bobot matriks pada 1 iterasi sebelumnya.

$W_{ik}(t-2)$  adalah bobot matriks pada 2 iterasi sebelumnya.

$\Delta W_{ik}(t)$  adalah perubahan bobot matriks saat ini.

$\alpha$  adalah variabel momentum standar untuk mempercepat proses pembelajaran.

$\beta$  adalah variabel momentum modifikasi untuk mempercepat proses pembelajaran.

## 3. METODE PENELITIAN

Penelitian ini terdiri atas beberapa langkah proses sebagai berikut :

1. Normalisasi dataset
2. Pelatihan JST
3. Pengujian JST.

### 3.1. Normalisasi dataset

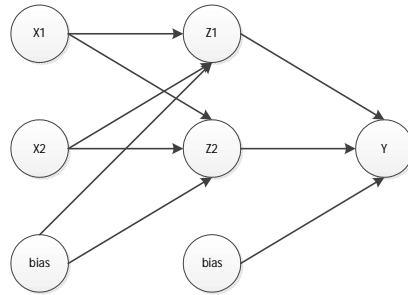
Dataset NASA memiliki nilai yang sangat tinggi sehingga perlu dilakukan proses normalisasi data agar memudahkan pada proses pelatihan untuk pengenalan pola menggunakan JST *Improved Backpropagation*. Proses normalisasi data menggunakan persamaan :

$$X = \left( \frac{a}{m} \right) \quad (3)$$

Dimana a adalah nilai hasil pemeriksaan yaitu nilai yang akan dinormalisasi dan m adalah nilai tertinggi dari hasil pemeriksaan, dan X adalah hasil normalisasi.

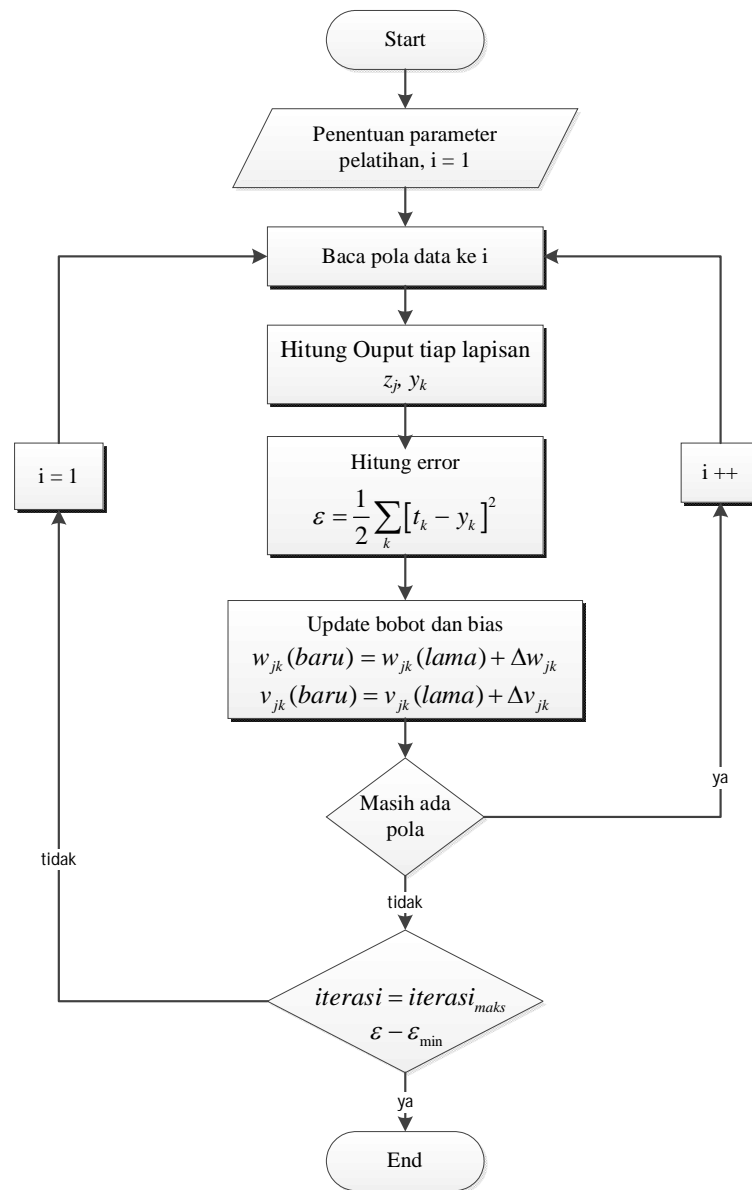
### 3.2. Pelatihan jaringan syaraf tiruan

Proses pelatihan jaringan pada dasarnya merupakan proses penyesuaian bobot-bobot untuk masing – masing simpul antara lapisan input, lapisan tersembunyi, dan lapisan output. Penyesuaian bobot dilakukan secara terus-menerus sampai dicapai *error* yang paling minimum. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *tangent hyperbolic*. Arsitektur JST yang digunakan dalam penelitian ini seperti pada Gambar 1.



**Gambar 1.** Arsitektur JST

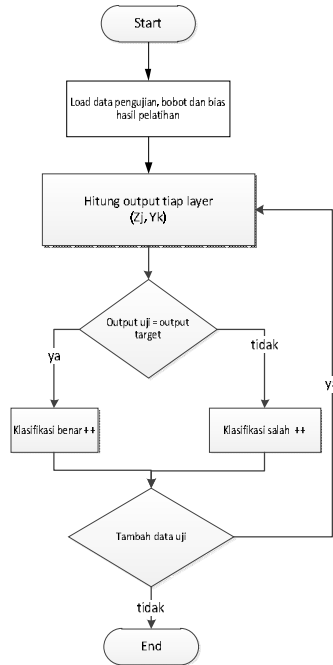
Flowchart algoritma pelatihan JST *backpropagation* seperti Gambar 4.7



**Gambar 4.7** Pelatihan JST

### 3.3. Pengujian jaringan syaraf tiruan

Setelah JST dilatih dengan sekumpulan pola maka tahap selanjutnya adalah melakukan pengujian terhadap kinerja dari JST sekaligus untuk mengukur kinerja dari sistem yang dibangun. Algoritma pengujian jaringan syaraf tiruan mengikuti *flowchart* seperti Gambar 4.8.



Gambar 4.8 Pengujian JST

## 4. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan dataset dari NASA yang diambil sebanyak 18 proyek sebagaimana ditunjukkan pada Tabel 1. Dataset tersebut dibagi 2 yaitu 13 proyek pertama digunakan sebagai data pelatihan dan sisanya 5 proyek terakhir digunakan sebagai data pengujian pada JST *Improved Backpropagation*.

Tabel 1. NASA Data (Bailey dan Basili, 1981) of Effort Estimation

Project No.	KDLOC	Methodology	Actual Effort
1	90.2	30	115.8
2	46.2	20	96
3	46.5	19	79
4	54.5	20	90.8
5	31.1	35	39.6
6	67.5	29	98.4
7	12.8	26	18.9
8.	10.5	34	10.3
9.	21.5	31	28.5
10.	3.1	26	7
11	4.2	19	9
12	7.8	31	7.3
13	2.1	28	5
14	5	29	8.4
15	78.6	35	98.7
16	9.7	27	15.6
17	12.5	27	23.9
18	100.8	34	138.3

Proses pengujian metode dilakukan menggunakan lima (5) buah lalu diuji menggunakan *backpropagation* konvensional dan *backpropagation improved* (Tabel 2)

**Tabel 2.** Aktual *effort* menggunakan model estimasi yang berbeda

Actual Effort	BP Konvensional	Improved BP
8.4	7.9455	8.33
98.7	98.9744	98.71
15.6	14.6092	16.97
23.9	19.3829	22.81
138.3	99.5649	130.15

Proses evaluasi kinerja dari metode *improved backpropagation* dilakukan berdasarkan informasi pada Tabel 2 dengan menggunakan rumus evaluasi RMSSE (Bintiri, et al. 2012)

**Tabel 3.** Perhitungan *error* menggunakan model estimasi yang berbeda

Performance Criteria	Model Estimasi	
	BP Konvensional	Improved BP
RMSSE	17.4475	3.72

Pada tahap evaluasi system perhitungan *effort* dan *errors* menggunakan model *JST Backpropagation* konvensional dan *JST Improved Backpropagation* ditunjukkan pada Tabel 2. dan ngahTabel 3. Sebagaimana ditunjukkan pada Tabel 2. Hasil perhitungan *effort* menggunakan model *JST Improved Backpropagation* lebih mendekati nilai *effort* sebenarnya hal diperkuat dengan hasil evaluasi sistem dari *JST Improved Backpropagation* yang menunjukkan nilai *error* yang kecil dengan RMSSE = 3.72 lebih kecil dibandingkan dengan RMSSE BP konvensional (17.4475). Berdasarkan informasi tersebut di atas (Tabel 3) maka terlihat bahwa kontribusi BP Improved terhadap menurunnya RMSSE prediksi adalah sebesar 13.71

## 5. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa *JST Improved Backpropagation* dapat digunakan sebagai metode untuk estimasi biaya perangkat lunak dengan hasil yang akurat dan dapat dipercaya. Penambahan momentum mempercepat proses konvergensi dan menunjukkan kinerja dari *JST* semakin baik. Hal ini ditunjang dengan hasil pengujian yakni menurunnya *error* prediksi (RMSSE) dari 17.4475 menjadi 3.72.

## DAFTAR PUSTAKA

- Bailey, J., W., dan Basili, V., R., 1981. A meta model for software development resource expenditure, *Proceedings of the International Conference on Software Engineering*, pp. 107-115.
- Bintiri, M., G., SN. Azhari., Dillak, R. Y., 2012. Perbandingan Model Algoritmik Dan Non Algoritmik Untuk Estimasi Biaya Perangkat Lunak, *SNATI 2012*.
- Fausett, L., 1994, *Fundamental of Neural Networks: Architectures Algorithms and Applications*, Prentice Hall
- Choudhary, A., Rishi,R., 2011. Improving The Character Recognition Efficiency of Feed Forward BP Neural Network, *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol. 3(1), 85-95.
- Idri, A., Khoshgoftaar, T., M., Abran, A., 2002. Can Neural Network be easily Interpreted in Software Cost Estimation ?, *Word Congress on Computational Intelligence*, Honolulu, Hawaii.
- Kaur, J., Singh, S., Kahlon, K., S., Bassi, P., 2010. Neural Network-A Novel Techniquefor Software Effort Estimation, *International Journal od Computer Theory and Engineering*, Vol 2(1), 1793-8201.
- Leung, H., & Fan, Z. (2001). *Software Cost Estimation*, Departemen of Computing, The Hongkong Polytechnic University, 1- 14.
- Sameet, B., Swathi, G., K., Gupta, K., V., RaviTeja, Ch., Sumana, S., 2009. A Novel Neural Network Approach for Software Cost Estimation Using Functional Link Artificial Neural Network (FLANN), *IJCSNS International Journal od computer Science and Network Security*, Vol.9(6), 126-130.