

SISTEM ESTIMASI BIAYA DAN SUMBER DAYA PROYEK PERANGKAT LUNAK BERBASIS COCOMO II MENGGUNAKAN NEURAL NETWORK

Joko Irawan¹, Yulison Herry Chrisnanto², Asri Maspupah³

^{1,2,3} Informatika, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani Cimahi
E-mail: ¹gonzalesokko@gmail.com, ²y.chrisnanto@gmail.com, ³asri.maspupah89@gmail.com

Abstrak – Estimasi biaya dan sumber daya perangkat lunak merupakan bagian tak terpisahkan dari proyek perangkat lunak. Saat melakukan estimasi sering menghadapi dua permasalahan yaitu estimasi berlebihan dan estimasi yang kurang. Estimasi berlebihan akan menimbulkan penambahan alokasi biaya dan sumber daya, sedangkan estimasi yang kurang akan mengurangi kualitas produk. Untuk mengantisipasi terjadinya kesalahan estimasi maka dikembangkan suatu metode untuk mengestimasi biaya dan sumber daya proyek perangkat lunak berbasis Constructive Cost Model 1997 (COCOMO II) dengan menggunakan Algoritma Backpropagation Neural Network. COCOMO II menghasilkan nilai estimasi dari data latih, sedangkan Algoritma Backpropagation digunakan untuk menghasilkan nilai estimasi data uji. Hasil dari perhitungan Backpropagation dilakukan penggabungan dengan model COCOMO II sehingga menghasilkan estimasi biaya dan sumber daya proyek perangkat lunak. Jumlah data yang digunakan dalam penelitian ini sebanyak 60 data. Model yang dikembangkan ini dievaluasi menggunakan PRED. Berdasarkan hasil dari penelitian diperoleh bahwa penggunaan model COCOMO II, dan algoritma Backpropagation memiliki kedekatan dalam melakukan estimasi biaya, dan sumber daya proyek perangkat lunak dengan nilai PRED sebesar 75%. Kesimpulan dari penelitian ini model COCOMO II, dan algoritma Backpropagation dapat mengurangi tingkat kegagalan dalam proyek perangkat lunak, berdasarkan estimasi biaya, dan sumber daya.

Kata Kunci — Algoritma Backpropagation, COCOMO II, Estimasi, , Proyek perangkat lunak;

1. PENDAHULUAN

Proyek adalah sekumpulan tujuan yang jelas, terjadwal, memiliki batasan anggaran, dan membutuhkan sumber daya yang telah ditentukan (Harold Kerzner, 1998). Pada umumnya, masyarakat beranggapan bahwa keluaran dari proyek berupa bangunan. Namun pada kenyataannya, keluaran dari proyek tidak hanya menghasilkan benda, keluaran dari proyek dapat saja penerapan sistem kepada suatu organisasi yang biasa disebut dengan proyek perangkat lunak.

Saat mengerjakan suatu proyek perangkat lunak, keberhasilan suatu proyek harus dimulai dengan perencanaan dan penyusunan tahap yang benar, karena jika dengan perencanaan yang buruk, maka akan menyebabkan terjadinya kegagalan suatu proyek. Berdasarkan survei yang dilakukan oleh *Project Management Institute* (PMI) pada tahun 2017 terhadap proyek perangkat lunak secara global, sekitar 60% atau kurang proyek diselesaikan tepat waktu dan sesuai anggaran. Sedangkan untuk proyek perangkat lunak di Indonesia terdapat 27% proyek perangkat lunak yang terselesaikan sesuai anggaran, 55% proyek perangkat lunak mengalami masalah, dan 18% proyek perangkat lunak mengalami kegagalan [1]. Dilihat dari fenomena yang ada, tingkat kesuksesan suatu proyek perangkat lunak di Indonesia sangat rendah, salah satu aktivitas dari proyek perangkat lunak yang menyebabkan kegagalan suatu proyek adalah estimasi sumber daya, biaya, serta jadwal proyek [2], hal ini diperjelas dengan survei yang dilakukan oleh PMI pada tahun 2017, 28% kegagalan proyek terjadi akibat tidak cocoknya biaya yang dikeluarkan dengan estimasi biaya yang

direncanakan, dan 26% kegagalan proyek terjadi akibat tidak cocoknya waktu pengerjaan proyek dengan estimasi jadwal proyek yang telah dirancang.

Estimasi sumber daya adalah suatu kegiatan pengaturan sumber daya dalam mencapai tujuan dan sasaran dari proyek, sehingga proyek dapat berjalan sesuai dengan tahapan dan target yang dikehendaki [3][4]. Estimasi biaya adalah suatu kegiatan dalam pengolahan biaya yang diatur dalam mencapai tujuan dan sasaran proyek. Sedangkan jadwal proyek adalah perencanaan kegiatan yang akan dilakukan pada suatu proyek. Pada saat melakukan estimasi terdapat dua permasalahan yang sering terjadi yaitu estimasi berlebih dan estimasi yang kurang. Estimasi berlebih akan menimbulkan penambahan alokasi sumber daya dari yang dibutuhkan sehingga akan meningkatkan penanganan manajerial, sedangkan estimasi yang kurang akan mengurangi kualitas dari produk karena tidak sesuai dengan standar. Untuk itu perlu dilakukan langkah yang hati-hati dalam melakukan estimasi suatu proyek perangkat lunak sehingga dapat dicapai keberhasilan proyek yaitu tepat waktu, sesuai biaya dan terpenuhinya standar kualitas produk [5]. Salah satu model yang digunakan untuk melakukan estimasi biaya, dan sumber daya dalam proyek perangkat lunak adalah model *Constructive Cost Model 1997*, biasa disebut COCOMO versi II atau COCOMO II.

COCOMO II adalah suatu model estimasi biaya, dan sumber daya yang sering digunakan dalam proyek perangkat lunak, berdasarkan motivasi dari tiap orang yang terlibat ditempatkan sebagai titik berat [6]. Pada saat melakukan perhitungan menggunakan COCOMO II, terdapat beberapa faktor yang mempengaruhi dalam

melakukan estimasi biaya, dan sumber daya, yaitu: ruang lingkup proyek, sumber daya[4], organisasi, dan proses[7]. Hal ini menunjukkan bahwa untuk melakukan estimasi dibutuhkan atribut yang mendukung dalam menentukan faktor keberhasilan dalam melakukan estimasi. Salah satu metode yang digunakan dalam pemilihan keluaran COCOMO II adalah *Neural Network*.

Neural Network sebenarnya mengadopsi dari kemampuan otak manusia yang mampu memberikan rangsangan[8], melakukan proses[5], dan memberikan keluaran[2], sehingga sangat membantu dalam hal pemilihan keluaran yang terbaik[5]. Salah satu kemampuan *Neural Network* adalah melakukan prediksi, atau estimasi[8]. Salah satu algoritma dalam memprediksi adalah algoritma Backpropagation. Pada penelitian ini Algoritma Backpropagation akan diterapkan untuk melakukan estimasi upaya, estimasi upaya akan dipecah menggunakan model COCOMO II yang akan menghasilkan estimasi biaya dan sumber daya proyek perangkat lunak.

2. METODE PENELITIAN

2.1 COCOMO II

Constructive Cost Model 1997 (COCOMO versi II, atau COCOMO II) merupakan model yang menyempurnakan model sebelumnya yaitu model COCOMO 81 yang dikembangkan oleh Barry Boehm pada tahun 1981. Model COCOMO 81 dapat digunakan untuk menghitung usaha dan waktu dalam proyek perangkat lunak, salah satu kelemahannya adalah ketidaksesuaian pemodelannya dengan lingkungan proyek perangkat lunak pada akhir tahun 1990-an, oleh karena itu pada tahun 1997 dipublikasikan COCOMO II [9]. Kelebihan utama yang terdapat pada COCOMO II sebagai penyempurnaan dari model sebelumnya yaitu, dapat digunakan melakukan estimasi biaya, dan sumber daya berdasarkan perkembangan program. COCOMO II memiliki beberapa tahap, yaitu : mengetahui estimasi panjang baris kode dari perangkat lunak dalam satuan kilo (*Kilo Line of Code* atau KLOC), mengetahui jenis proyek perangkat lunak, mengetahui kemampuan yang dimiliki perusahaan *software house*, dan hitung upaya (*effort*). Adapun bobot nilai yang terdapat pada jenis proyek, dapat dilihat pada Tabel 1. Bobot nilai jenis proyek.

Tabel 1. Bobot nilai jenis proyek

Jenis proyek	Panjang kode	a	b	c	d
Kecil	50 KLOC	3.2	1.05	2.5	0.38
Sedang	51-300 KLOC	3	1.12	2.5	0.35
Besar	301- 5000 KLOC	2.8	1.2	2.5	0.32

Terdapat 4 faktor utama secara subjektif yang berpengaruh terhadap proyek perangkat lunak[4], yaitu: produk, perangkat keras (*hardware*), personal, dan atribut proyek. Setiap faktor memiliki jumlah atribut yang berbeda, total dari atribut berjumlah 17 atribut, dengan memiliki tingkatan dari sangat rendah hingga sangat tinggi pada setiap atribut[10]. Untuk lebih jelas dapat dilihat pada Tabel 2. Faktor penyesuaian usaha

Tabel 2. Faktor penyesuaian usaha

	SR	R	N	T	CT	ST
Produk						
Keandalan Perangkat Lunak	0.75	0.88	1	1.15	1.39	
Ukuran Database		0.93	1	1.09	1.19	
Kompleksitas Produk	0.75	0.88	1	1.15	1.3	1.66
Produk dapat dikembangkan kembali		0.91	1	1.14	1.29	1.49
Dokumentasi	0.98	0.95	1	1.06	1.13	
Perangkat Keras						
Kendala Waktu			1	1.11	1.31	1.67
Batasan Penyimpanan			1	1.06	1.21	1.57
Platform Volatility		0.87	1	1.15	1.3	
Sumber Daya Manusia						
Kemampuan Analisis	1.5	1.22	1	0.83	0.67	
Kemampuan Programming	1.37	1.16	1	0.87	0.74	
Kesinambungan SDM	1.24	1.1	1	0.92	0.84	
Pengalaman Membangun Aplikasi	1.22	1.1	1	0.89	0.81	
Pengalaman Terhadap Perangkat Keras	1.25	1.12	1	0.88	0.81	
Pengalaman Terhadap Bahasa Pemrograman	1.22	1.1	1	0.91	0.84	
Atribut Proyek						
Penggunaan alat Perangkat lunak	1.24	1.12	1	0.86	0.72	
Pengembangan Melalui Site	1.25	1.1	1	0.92	0.84	0.78
Jadwal	1.29	1.1	1	1	1	

Upaya merupakan hasil akhir dari perhitungan COCOMO II, upaya dapat diakumulasi dengan menggunakan persamaan (1).

$$Upaya = \prod EAF \times a \times KLOC^b \dots\dots\dots(1)$$

Keterangan:

EAF = Bobot nilai yang didapat dari Tabel 2,

KLOC = Panjang kode pemrograman,

a & b = koefisien yang didapat dari Tabel 1, berdasarkan jenis proyek.

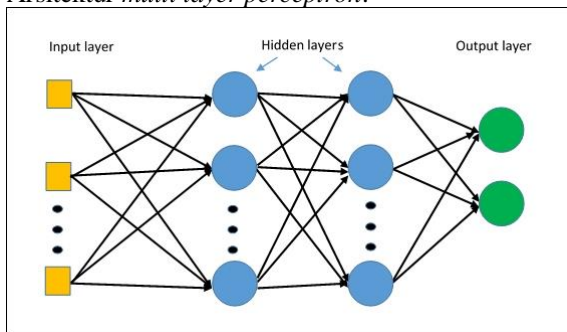
2.2 Algoritma Backpropagation

Algoritma backpropagation pertama kali dirumuskan oleh Paul Werbos pada tahun 1974 dan dipopulerkan oleh Rumelhart bersama McClelland untuk dipakai pada Neural Network. Algoritma backpropagation menggunakan metode pembelajaran terawasi (*supervised learning*), dalam artian mempunyai

target yang akan dicari. Arsitektur jaringan yang digunakan oleh algoritma *backpropagation* yaitu *multi layer perceptron* (MLP). MLP terdiri dari 3 lapisan (*layer*)[11], [12], yaitu:

1. lapisan masukan (*input layer*), terdiri dari neuron yang menerima masukan dari lingkungan luar. Neuron tersebut tidak mengalami perubahan, tetapi hanya mengirimkan informasi ke lapisan tersembunyi.
2. lapisan tersembunyi (*hidden layer*), terdiri dari neuron yang menerima masukan dari lapisan masukan, melakukan perubahan terhadap informasi yang diterima, dan mengirimkan hasil informasi ke lapisan berikutnya.
3. lapisan keluaran (*output layer*), terdiri dari neuron yang menerima hasil dari pengolahan lapisan tersembunyi, dan mengirimkan hasil terhadap pengguna.

Artificial Neural Network dengan arsitektur MLP dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal. Arsitektur MLP dapat dilihat pada Gambar 1 Arsitektur *multi layer perceptron*.



Gambar 1 Arsitektur *multi layer perceptron*

Setidaknya terdapat 3 tahapan dalam melakukan pelatihan algoritma *backpropagation*, yaitu: inialisasi bobot awal, *feed forward*, dan *backpropagation*.

Inialisasi bobot awal dilakukan dengan cara menggunakan nilai acak dengan rentang antara 0-1, tahapan inialisasi bobot awal juga menentukan jumlah masa (*epoch*) yaitu 10000, dan pembelajaran (*learning rate* atau α) sebesar 0.1.

Tahap selanjutnya yaitu *feed forward*, tahap ini melakukan perhitungan berdasarkan neuron input, bobot, serta fungsi aktivasi. Adapun tahapan yang dilakukan pada proses *feed forward*, yaitu :

1. Tiap neuron masukan ($x_i, i = 1,2,3 \dots n$), tiap nilai neuron meneruskan nilai tersebut ke lapisan tersembunyi.
2. Tiap neuron lapisan tersembunyi ($z_j, j = 1,2,3 \dots m$), mendapatkan nilai dari tiap neuron masukan dengan tujuan kepada neuron tersembunyi yang telah diberi bobot sebelumnya. Proses ini dapat dikalkulasi dengan persamaan (2).

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots(2)$$

Keterangan:

- z_{in_j} = Nilai neuron lapisan tersembunyi,
- v_{0j} = Nilai bobot bias,
- x_i = Nilai neuron layer masukan yang ke-i,

v_{ij} = Nilai bobot neuron masukan ke-i yang menuju neuron j.

Setelah mendapatkan nilai dari lapisan masukan, nilai neuron yang telah diakumulasi dengan menggunakan persamaan (2) di masukan pada fungsi aktivasi sehingga memiliki bobot nilai pada setiap neuron. Fungsi aktivasi yang digunakan yaitu sigmoid biner, dan dapat dikalkulasi dengan persamaan (3).

$$z_j = f(z_{in_j}) = \frac{1}{1+e^{-z_{in_j}}} \dots\dots\dots(3)$$

Keterangan:

- z_j = Nilai output dari layer tersembunyi setelah melakukan fungsi aktivasi.
- z_{in_j} = Nilai output dari layer tersembunyi sebelum melakukan fungsi aktivasi.

3. Tiap neuron lapisan keluaran ($y_k, k = 1,2,3 \dots p$), mendapatkan nilai dari setiap neuron tersembunyi dengan tujuan kepada neuron keluaran yang telah diberi bobot sebelumnya. Proses ini dapat dikalkulasi dengan persamaan (4).

$$y_{in_k} = w_{0k} + \sum_{j=1}^m z_j w_{jk} \dots\dots\dots(4)$$

Keterangan:

- y_{in_k} = Nilai neuron lapisan keluaran,
- w_{0k} = Nilai bobot bias,
- z_j = Nilai neuron lapisan tersembunyi yang ke-j,
- w_{jk} = Nilai bobot neuron lapisan tersembunyi ke-j yang menuju neuron k.

Setelah mendapatkan nilai dari lapisan tersembunyi, nilai neuron yang telah diakumulasi dengan menggunakan persamaan (4) di masukan pada fungsi aktivasi sehingga memiliki bobot nilai pada setiap neuron. Fungsi aktivasi yang digunakan yaitu sigmoid biner, dan dapat dikalkulasi dengan persamaan (5).

$$y_k = f(y_{in_k}) = \frac{1}{1+e^{-y_{in_k}}} \dots\dots\dots(5)$$

Tahap selanjutnya yaitu perhitungan *backpropagation*, yaitu tahapan yang berfungsi untuk melakukan perhitungan balik dari neuron keluaran agar memiliki nilai bobot yang sesuai dalam jaringan neural network[13], dengan perhitungan balik ini nilai kesalahan atau *error* bisa dikurangi dengan cukup baik. Adapun tahapan yang dilakukan pada proses *Backpropagation*, yaitu :

1. Tiap neuron lapisan keluaran ($y_k, k = 1,2,3 \dots p$), hitung selisih kesalahan dari nilai target y_k dengan nilai y_k , dan dikalikan dengan turunan fungsi aktivasi. Nilai kesalahan dapat dikalkulasi dengan persamaan (6).

$$\delta_k = (target\ y_k - y_k) f'(y_{in_k}) \dots\dots\dots(6)$$

Keterangan:

- δ_k = Nilai kesalahan pada neuron keluaran ke-k,
- $target\ y_k$ = Nilai target yang akan dicapai,
- y_k = Nilai hasil kalkulasi *Feed Forward* pada neuron keluaran ke-k,
- $f'(y_{in_k}) = f(y_{in_k})(1 - f(y_{in_k}))$

Setelah mendapatkan selisih kesalahan selanjutnya proses perbaikan bobot pada setiap neuron lapisan tersembunyi, dengan persamaan (7).

$$w_{jk}(baru) = w_{jk}(lama) + (\alpha\ x\ z_j\ x\ \delta_k) \dots\dots\dots(7)$$

Keterangan:

w_{jk} = Nilai bobot dari neuron j ke neuron k,

α = Nilai pembelajaran mesin,

z_j = Nilai neuron lapisan tersembunyi yang ke-j,

δ_k = Nilai kesalahan pada neuron keluaran ke-k.

2. Tiap neuron lapisan tersembunyi ($z_j, j = 1,2,3 \dots m$) menjumlahkan kesalahan dari setiap neuron keluaran. Penjumlahan kesalahan tersebut dapat dilakukan dengan menggunakan persamaan (8).

$$\delta_{in_j} = \sum_{k=1}^p \delta_k w_{jk} \dots \dots \dots (8)$$

Keterangan:

δ_{in_j} = Menjumlahkan kesalahan pada setiap neuron keluaran,

δ_k = Nilai kesalahan pada neuron keluaran ke-k,

w_{jk} = Nilai bobot dari neuron j ke neuron k.

Setelah mendapatkan nilai total kesalahan pada setiap neuron tersembunyi, proses selanjutnya yaitu menghitung selisih kesalahan yang terdapat pada neuron tersembunyi dengan menggunakan turunan dari fungsi aktivasi, dapat dihitung dengan persamaan (9).

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \dots \dots \dots (9)$$

Keterangan:

δ_j = Nilai kesalahan pada neuron tersembunyi ke-j,

δ_{in_j} = Nilai target yang akan dicapai,

$$f'(z_{in_j}) = f(z_{in_j})(1 - f(z_{in_j}))$$

Setelah mendapatkan selisih kesalahan selanjutnya proses perbaikan bobot pada setiap neuron lapisan masukan, dengan persamaan (10).

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + (\alpha \times x_i \times \delta_j) \dots \dots \dots (10)$$

Keterangan:

v_{ij} = Nilai bobot dari neuron i ke neuron j,

α = Nilai pembelajaran mesin,

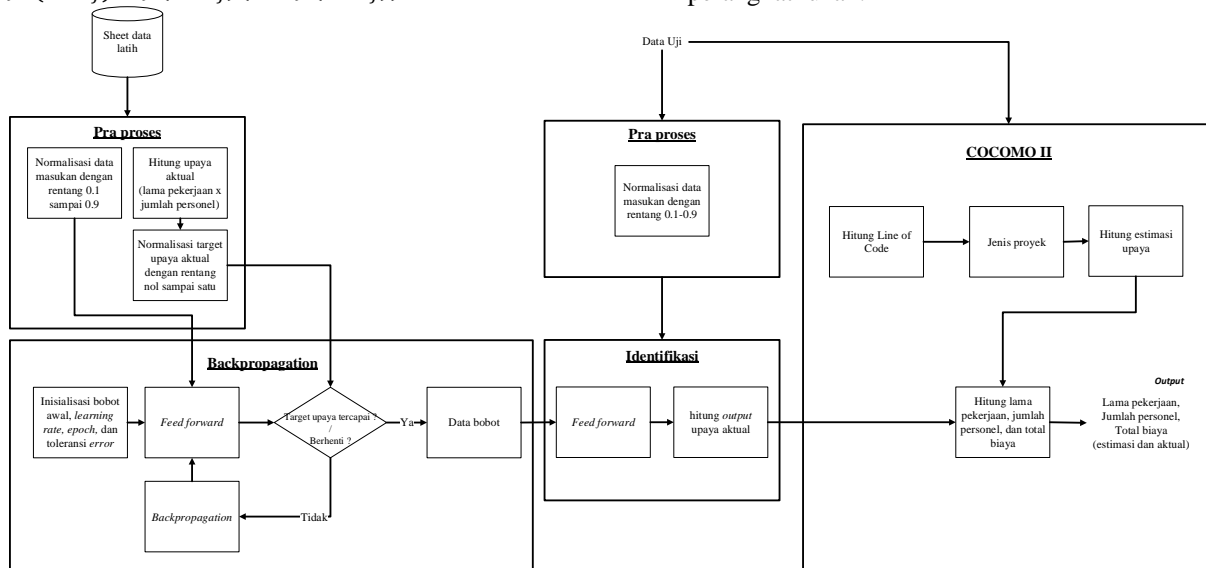
x_i = Nilai neuron lapisan masukan yang ke-i,

δ_j = Nilai kesalahan pada neuron tersembunyi ke-j.

Lakukan proses *feed forward*, dan *backpropagation* hingga nilai target tercapai, atau *epoch* sudah selesai.

2.3. Perancangan Sistem

Perancangan sistem estimasi biaya dan sumber daya proyek perangkat lunak diawali dengan cara pengambilan data berupa survei, wawancara, dan melakukan kuesioner dari *software house* yang ada di kawasan Cimahi dan Kabupaten Bandung Barat. Setelah pengambilan data dilakukan, data yang didapat diproses menggunakan model COCOMO II dan tahapan normalisasi agar dapat digunakan pada metode algoritma Backpropagation. Adapun perancangan sistem estimasi biaya dan sumber daya proyek perangkat lunak dapat dilihat pada Gambar 2. Perancangan estimasi biaya dan sumber daya proyek perangkat lunak.



Gambar 2. Perancangan sistem estimasi biaya dan sumber daya proyek perangkat lunak

3. HASIL DAN PEMBAHASAN

Dari analisis awal, didapat bahwa biaya yang dikeluarkan oleh *software house* untuk membangun sistem informasi sangat beragam. Kecenderungan yang diperoleh adalah semakin tinggi panjang kode, dan faktor penyesuaian usaha, maka semakin besar biaya yang dikeluarkan untuk membangun sistem informasi yang akan dilakukan oleh *software house*.

Dengan menggunakan faktor-faktor yang dapat berpengaruh dalam perhitungan biaya yang terdapat pada Tabell, dan Tabel 2, dilakukan perhitungan estimasi biaya dan sumber daya proyek perangkat lunak. Hasil dari perhitungan estimasi akan dibandingkan

dengan biaya aktual yang dilakukan terhadap proyek tersebut.

Evaluasi terdiri membandingkan estimasi upaya dengan upaya aktual. Ada banyak kriteria evaluasi untuk estimasi upaya proyek perangkat lunak, salah satu yang digunakan pada penelitian ini yaitu menggunakan evaluasi Relative Error(RE), dan Magnitude Relative Error (MRE). Perhitungan RE dapat dilakukan dengan menggunakan persamaan (11), dan MRE dapat dilakukan dengan menggunakan persamaan (12).

$$RE = \frac{Upaya_{estimasi} - Upaya_{aktual}}{Upaya_{aktual}} \dots \dots \dots (11)$$

$$MRE = \frac{|Upaya_{estimasi} - Upaya_{aktual}|}{Upaya_{aktual}} \dots \dots \dots (12)$$

Hasil dari perhitungan RE, dan MRE dapat dilihat pada Tabel 3. Hasil perhitungan evaluasi RE, dan MRE

Tabel 3. Hasil perhitungan evaluasi RE, dan MRE

Data	Upaya Aktual	Upaya Estimasi	RE	MRE
1	2040,00	1977,08	-0,03	0,03
2	1600,00	1324,52	-0,17	0,17
3	243,00	249,07	0,02	0,02
4	240,00	278,42	0,16	0,16
5	33,00	41,00	0,24	0,24
6	43,00	42,00	-0,02	0,02
7	8,00	54,00	5,75	5,75
8	1075,00	922,00	-0,14	0,14
9	423,00	352,00	-0,17	0,17
10	321,00	310,00	-0,03	0,03
11	218,00	447,00	1,05	1,05
12	201,00	189,00	-0,06	0,06
13	79,00	76,00	-0,04	0,04
14	60,00	71,00	0,18	0,18
15	61,00	55,00	-0,10	0,10
16	40,00	40,00	0,00	0,00
17	9,00	8,32	-0,08	0,08
18	33,00	35,00	0,06	0,06
19	43,00	53,00	0,23	0,23
20	8,00	7,00	-0,13	0,13
21	1075,00	798,00	-0,26	0,26
22	423,00	823,00	0,95	0,95
23	539,00	425,16	-0,21	0,21
24	453,00	231,00	-0,49	0,49
25	453,00	612,00	0,35	0,35
26	387,00	512,24	0,32	0,32
27	88,00	56,00	-0,36	0,36
28	98,00	201,00	1,05	1,05
29	1063,00	1132,00	0,06	0,06
30	702,00	614,48	-0,12	0,12
31	605,00	321,00	-0,47	0,47
32	230,00	791,00	2,44	2,44
33	82,00	92,00	0,12	0,12
34	55,00	54,00	-0,02	0,02
35	47,00	44,00	-0,06	0,06
36	12,00	16,71	0,39	0,39
37	8,00	6,71	-0,16	0,16
38	8,00	3,00	-0,63	0,63
39	69,00	68,00	-0,01	0,01
40	45,00	44,00	-0,02	0,02
41	83,00	79,11	-0,05	0,05
42	87,00	89,00	0,02	0,02
43	106,00	122,00	0,15	0,15
44	126,00	132,00	0,05	0,05
45	36,00	30,00	-0,17	0,17
46	1272,00	1752,80	0,38	0,38
47	156,00	139,00	-0,11	0,11
48	170,00	190,00	0,12	0,12
49	122,00	105,00	-0,14	0,14
50	41,00	32,00	-0,22	0,22
51	14,00	18,71	0,34	0,34
52	20,00	20,46	0,02	0,02
53	18,00	20,46	0,14	0,14
54	958,00	777,03	-0,19	0,19
55	237,00	221,00	-0,07	0,07
56	130,00	154,00	0,18	0,18
57	70,00	53,00	-0,24	0,24
58	57,00	54,00	-0,05	0,05
59	50,00	60,00	0,20	0,20
60	38,00	47,00	0,24	0,24

Percentage of the Prediction (PRED) adalah pengujian akurasi yang didapat setelah melakukan proses perhitungan MRE, pengujian akurasi menggunakan PRED dapat dilakukan dengan menggunakan persamaan (13).

$$PRED(x) = \frac{A}{N} x 100\% \dots \dots \dots (13)$$

Keterangan:

- x = Bobot nilai yang dicari,
- A = Jumlah data yang memiliki nilai kurang atau sama dengan x,
- N = Jumlah data latih.

Pada penelitian ini nilai x yang digunakan yaitu 0.25, sehingga nilai akurasi yang diperoleh adalah

$$PRED(0,25) = \frac{15}{60} x 100\% = 75\%$$

Berdasarkan hasil perhitungan MRE, dan PRED, dapat disimpulkan bahwa pengujian akurasi sistem yang dibangun memiliki nilai persentase sebesar 75% berdasarkan 60 data latih, dengan 45 data latih yang memiliki nilai usaha estimasi mendekati nilai usaha aktual, atau sebaliknya, 15 data latih yang memiliki nilai usaha estimasi atau nilai usaha aktual yang memiliki perbandingan dua kali lipat atau lebih.

4. SIMPULAN

Sistem estimasi biaya, dan sumber daya proyek perangkat lunak merupakan sistem yang dapat mengurangi tingkat kegagalan proyek perangkat lunak berdasarkan atribut pendukung dalam melakukan estimasi biaya, dan sumber daya perangkat lunak. Berdasarkan hasil pengujian perangkat lunak, dan hasil pengujian sistem, maka dapat disimpulkan bahwa sistem ini dapat membantu dalam menangani proses estimasi biaya, dan sumber daya proyek perangkat lunak. Penelitian ini dapat memudahkan *Chief Information Office*(CIO) untuk melakukan estimasi biaya, dan sumber daya proyek perangkat lunak berdasarkan faktor-faktor yang mempengaruhi atribut. Faktor-faktor yang digunakan, antara lain : faktor produk, perangkat keras (*hardware*), sumber daya manusia, dan proyek. Hasil dari penelitian berupa perangkat lunak estimasi biaya, dan sumber daya proyek perangkat lunak telah berhasil diimplementasikan dengan akurasi sistem sebesar 75%.

5. SARAN

Pada penelitian ini tentu terdapat kekurangan yang dilatarbelakangi oleh keterbatasan kemampuan yang dimiliki, maka dijabarkan beberapa saran yang dapat dilanjutkan pada penelitian berikutnya. Saran tersebut dapat dijelaskan sebagai berikut:

1. Pemberian estimasi biaya, dan sumber daya proyek perangkat lunak dapat direduksi apabila menemukan proyek perangkat lunak yang serupa.
2. Atribut yang digunakan dapat ditambah agar akurasi sistem dalam estimasi biaya, dan sumber daya proyek perangkat lunak dapat meningkat

DAFTAR PUSTAKA

[1] R. D. Apriyanto and H. P. Putro, "Tingkat kegagalan dan keberhasilan proyek sistem

- informasi di indonesia,” *Seminar Nasional Teknologi Informasi dan Komunikasi*, vol. 4, no. 4, pp. 23–24, 2018.
- [2] S. Ravishankar and P. Latha, “Software Cost Estimation using Fuzzy Logic,” *International Conference on Recent Trends in Computational Methods, Communication and Controls (ICON3C 2012)*, pp. 38–42, 2012.
- [3] F. J. Heemstra, “Software cost estimation,” *Information and Software Technology*, vol. 34, no. 10, pp. 627–639, 1992.
- [4] R. S. Pressman, *Software Engineering*, SEVENTH ED. New York: Raghathan Srinivasan, 2010.
- [5] A. Kaushik, a K. Soni, and R. Soni, “A Simple Neural Network Approach to Software Cost Estimation,” *Global Journal of Computer Science and Technology Neural & Artificial Intelligence*, vol. 13, no. 1, pp. 23–30, 2013.
- [6] D. E. Damayanti and A. R. Perdanakusuma, “Analisis Estimasi Biaya Pembuatan Perangkat Lunak Menggunakan Metode COCOMO II di Inagata Technosmith (Studi Kasus : Sistem Informasi Monitoring dan Evaluasi Penerimaan Beasiswa Santri Berprestasi UIN Malang),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 10, pp. 1220–1229, 2017.
- [7] S. L. Lam, R. Cheung, S. Wong, and E. S. K. Chan, “A SURVEY STUDY OF CRITICAL SUCCESS FACTORS IN INFORMATION,” *International Conference on Internet Studies*, vol. 13, no. 9, pp. 1–12, 2013.
- [8] A. Kaushik, A. Chauhan, D. Mittal, and S. Gupta, “COCOMO Estimates Using Neural Networks,” *International Journal of Intelligent Systems and Applications*, vol. 4, no. 9, pp. 22–28, 2012.
- [9] T. A. Wijaya, “Rancangan Estimasi Biaya dengan Teknik COCOMO II dan Neuro Fuzzy Studi Kasus : Sistem Informasi Rumah Sakit.”
- [10] Y. Wahyudi, “Perkiraan Biaya Perangkat Lunak dengan Jaringan Saraf Tiruan Propagasi Balik,” *Seminar Ilmiah Ilmu Komputer*, vol. 7, no. 4, pp. 43–65, 2017.
- [11] I. K. Hadihardaja and S. Sutikno, “Pemodelan Curah Hujan-Limpasan Menggunakan Artificial Neural Network (ANN) dengan Metode Backpropagation,” vol. 12, 2005.
- [12] E. S. Noviando, E. Ervianto, and I. Yasri, “Studi Penerapan ANN (Artificial Neural Network) Untuk Menghilangkan Harmonisa Pada Gedung Pusat Komputer,” *Jom FTEKNIK*, vol. 3, no. 2, pp. 1–6, 2016.
- [13] G. Kumar and P. K. Bhatia, “Automation of Software Cost Estimation using Neural Network Technique,” *International Journal of Computer Applications*, vol. 98, no. 20, pp. 11–17, 2014.