

Optimasi Koefisien COCOMO II Menggunakan Algoritma Kelelawar untuk Meningkatkan Akurasi Estimasi Biaya dan Waktu Pengembangan Perangkat Lunak

Alifia Puspaningrum¹, Fachrul Pralienka Bani Muhammad², Esti Mulyani³

^{1,2,3}Politeknik Negeri Indramayu

Jl. Lohbener Lama No.08, Lohbener, Indramayu

E-mail : ¹*alifia.puspaningrum@polindra.ac.id*, ²*fachrul.pbm@polindra.ac.id*,
³*estimulyani@polindra.ac.id*

ABSTRAK

Estimasi usaha dan waktu pengembangan perangkat lunak menjadi faktor penting dalam bidang rekayasa perangkat lunak terutama pada proses perbaikan perangkat lunak. Estimasi dapat meningkatkan kinerja dalam mengelola jadwal proyek, alokasi sumber daya manusia, estimasi biaya, dll. Model Biaya Konstruktif atau Constructive Cost Model (COCOMO) II merupakan model estimasi perangkat lunak yang umum digunakan yang dipengaruhi oleh empat koefisien. Koefisien yang optimal akan menghasilkan model yang optimal dalam memperkirakan tenaga dan waktu pengembangan. Jika koefisien terbaik diperoleh, maka model yang optimal dapat diperoleh. Selain itu, terdapat beberapa metode yang telah diusulkan untuk menyelesaikan permasalahan ini. Salah satu metode yang sedang populer digunakan adalah algoritma metaheuristik yang terinspirasi dari fenomena alam. Di samping itu, algoritma kelelawar adalah salah satu algoritma metaheuristik yang memiliki kemampuan yang baik untuk membangun pencarian solusi. Hasil eksperimen menunjukkan bahwa algoritma dapat lebih unggul dibandingkan dengan metode particle swarm optimization dan metode cuckoo search algorithm dengan skor MMRE estimasi biaya sebesar 22.65% dan MMRE estimasi pengembangan waktu sebesar 62.84%.

Kata kunci : *COCOMO II Model; Estimasi Effort; Estimasi Pengembangan Waktu; Algoritma Kelelawar*

ABSTRACT

Software cost and time development estimation are an important factors in software engineering, especially in software evolution. Software estimation can improve performance in managing project schedules, human resource allocation, cost estimation, etc. The Constructive Cost Model (COCOMO) II is a commonly used software estimation model which is influenced by four coefficients. Optimal coefficients will produce optimal models in estimating cost and time development. If the best coefficient is obtained, then the optimal model can be obtained. In addition, several methods have been proposed to solve this problem. One method that is currently being used is a metaheuristic algorithm that is inspired by natural phenomena. In addition, bat algorithm is one of the metaheuristic algorithms that has a good ability to build a solution search. Experimental results show that the algorithm can be superior to the particle swarm optimization method and the cuckoo search algorithm method with an estimated cost MMRE score of 22.65% and an estimated MMRE development time of 62.84%.

Keyword : *COCOMO II Model; Cost Estimation; Time Development Estimation; Bat Algorithm*

1. PENDAHULUAN

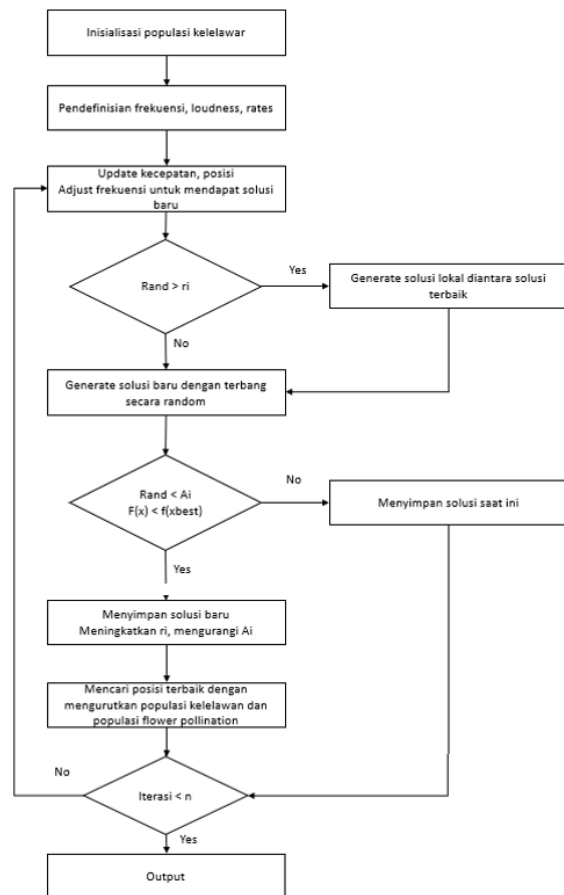
Seiring berkembangnya kebutuhan manusia, perangkat lunak juga diharapkan bisa berevolusi sesuai kebutuhan. Namun, evolusi perangkat tersebut harus bisa diandalkan dan memiliki performa yang baik. Dalam rangka mengatasi permasalahan tersebut diperlukan estimasi biaya yang mampu memperkirakan segala kebutuhan yang terdapat pada sebuah aplikasi. Estimasi biaya perangkat lunak merupakan hal yang sangat penting dalam pengembangan aplikasi (Rijwani, 2016). Saat ini, teknik estimasi yang paling umum digunakan adalah *Constructive Cost Model* (COCOMO). COCOMO merupakan teknik estimasi biaya aplikasi berbasis pada metode algoritma. Estimasi yang baik dapat meningkatkan performa perusahaan dalam penjadwalan proyek, alokasi SDM, estimasi biaya, dll (Ghatasheh, 2015). Keuntungan itu dapat mengurangi kemungkinan kegagalan dan terlambatnya proyek (Sarno, 2015).

COCOMO II terdiri dari koefisien-koefisien yang sangat mempengaruhi keakuratan estimasi. Dalam mengestimasi empat parameter COCOMO II terdapat banyak metode untuk mengestimasi nilai yang optimal, salah satunya adalah algoritma-algoritma metaheuristik sebagai salah satu akar keilmuan dari machine learning. Saat ini, kemampuan pemodelan usaha dan komponennya dalam pengembangan perangkat lunak membuat metode pembelajaran mesin lebih menarik perhatian (Zare, 2016). Tujuan dari metode pembelajaran mesin dalam memperkirakan upaya dan waktu perkembangan adalah menemukan solusi yang optimal. Beberapa metode pembelajaran mesin antara lain Ant Bee Colony Algorithm (Chalotra, 2015), Differential Evolution (Aljahdali, 2010), Firefly Algorithm (Ghatasheh, 2015), Genetic Algorithm (Sachan, 2016), Harmony Search (Jafari, 2016), Cuckoo Search (Parwita, 2016), Neural Network (Sarno, 2015), Fuzzy Logic Model (Madheswaran, 2014), dll.

Hingga tahun 2020, terdapat banyak algoritma metaheuristik yang telah dikembangkan, salah satunya adalah bat algorithm. Algoritma bat dapat memberikan solusi yang optimal. Sehingga proses penemuan ruang solusi diharapkan dapat menjadi lebih luas. Oleh karena itu, penelitian ini mengusulkan algoritma Bat dalam mengoptimisasi Estimasi Biaya dan Waktu Pengembangan Perangkat Lunak untuk Mendukung Perencanaan Proyek Perangkat Lunak.

2. METODE PENELITIAN

Seperti yang ditunjukkan pada Gambar 1, tahapan algoritma yang diimplementasikan adalah sebagai berikut:



Gambar 1. Bagan Alur Tahapan Algoritma Kelelawar

3. LANDASAN TEORI

3.1 Algoritma Kelelawar

Algoritma bat adalah pencarian metaheuristic yang diteliti oleh Xin-She Yang pada tahun 2010. Algoritma ini meniru perilaku kelelawar dengan menggunakan ekolokasi dalam pencarian mangsa, dapat membedakan objek apakah itu mangsa, halangan atau benda lain.

Untuk inisialisasi harus diidentifikasi aturan tentang posisi (x_i) dan kecepatan (v_i) dalam ruang pencarian dimensi (d) diperbaharui. Kecepatan v_i^t dan solusi baru x_i^t pada suatu waktu (t) digambarkan oleh persamaan berikut:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_o) f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

Keterangan:

f_i = Frekuensi yang dicari

f_{min} = Frekuensi minimum

f_{max} = Frekuensi maksimum

β = vektor acak

V_i = kecepatan yang dicari

X_i = posisi yang dicari

X_o = solusi posisi ideal

Algoritma bat juga menggunakan pencarian lokal dimana solusi dipilih diantara solusi terbaik dan terbaru.

$$X_{new} = X_{old} + \epsilon At \quad (4)$$

dimana:

$\epsilon \in [-1, 1]$: random number

$At = Ait >$ elemen kenyaringan suara terhadap waktu

Selain itu, dalam setiap proses iterasi, kenyaringan A_i dan laju r_i pulsa emisi harus selalu terupdate sehingga solusi baru ditingkatkan. Karena kenyaringan biasanya berkurang saat kelelawar telah menemukan mangsanya, sementara laju emisi denyut nadi meningkat, kita dapat memilih kenyaringan sebagai nilai kenyamanan apa pun. Persamaan kenyaringan A_i dan laju r_i dinyatakan dalam persamaan berikut:

$$A_{i,t+1} = \alpha A_{i,t}, r_{i,t+1} = r_{i,0} [1 - \exp(-\gamma t)] \quad (5)$$

dimana:

α dan γ = konstanta

r = emisi getaran

3.2 Constructive Cost Model (COCOMO)

Cost Constructive Model (COCOMO) adalah teknik estimasi perangkat lunak menggunakan source of line code (SLOC) yang digunakan untuk mengestimasi biaya, tenaga yang diperlukan, dan penjadwalan dalam pengembangan perangkat lunak. COCOMO merupakan model estimasi perangkat lunak yang menggunakan algorithmic model dimana model ini dikembangkan dengan cara mengumpulkan banyak data dari perangkat lunak yang sudah pernah dikerjakan. Data ini dianalisa untuk menemukan formula yang tepat sesuai dengan hasil observasi.

COCOMO II menggunakan spiral model sebagai siklus (life-cycle) dalam pengembangan perangkat lunak. Dengan model ini dimungkinkan untuk menghitung biaya evolusi perangkat lunak. Artinya apabila perangkat lunak sudah selesai dikerjakan dan pada perkembangannya dikehendaki adanya evolusi atau perubahan, maka estimasi biaya evolusi perangkat lunak dapat dilakukan. Hal ini dikarenakan dalam pengembangan perangkat lunak sangat penting untuk dapat mengakomodir evolusi dengan cepat.

Model ini menggunakan beberapa cost drivers yaitu jumlah baris kode (KSLOC), 17 Effort Multiplier, dan 5 Scale Factors. Persamaan berikut menjelaskan bagaimana cara mendapatkan usaha dalam person month dan lama pengerjaan dalam bulan.

$$PM = A * (KSLOC)^{B+C} * \prod_{i=1}^{17} EM_i \quad (6)$$

$$C = 0.01 * \sum_{i=1}^5 SF_i \quad (7)$$

$$TDEV = C * (PM)^F \quad (8)$$

$$F = D + 0.2 * 0.01 * \sum_{i=1}^5 SF_i = D + 0.2 * (E - B) \quad (9)$$

Dimana:

- PM adalah usaha dalam person-month.
- TDEV adalah waktu yang diperlukan untuk mengembangkan sebuah perangkat lunak.
- EM adalah perkalian dari setiap variabel effort mutiplier.

- d. SF adalah penjumlahan dari setiap variabel scale factors.
- e. A adalah konstanta produktifitas. $A = 2.94$, $B = 0.91$, $D = 0.28$

3.4 Akurasi

Terdapat banyak macam-macam metode evaluasi yang dapat digunakan untuk mengukur kehandalan dari metode yang diusulkan, beberapa diantaranya yang digunakan pada penelitian ini adalah Magnitude of Relative Error (MRE) dan Mean Magnitude of Relative Error (MMRE).

4. HASIL DAN PEMBAHASAN

Pada paper ini diterapkan pada skenario uji coba utama, yaitu pencarian koefisien effort dan time development terbaik untuk metode algoritma kelelawar.

Untuk mendapatkan hasil yang baik, makalah ini menganalisis beberapa macam skenario. Skenario pertama adalah mencari iterasi terbaik dengan menilai beberapa iterasi yaitu 500, 1000, 1500, 2000, dan 2500 yang diilustrasikan pada Gambar 2.



Gambar 2. Iterasi Algoritma Kelelawar

Dalam percobaan ini, empat koefisien dicari nilai optimalnya. MMRE menjadi nilai objektif atau nilai fitness dari algoritma optimasi. Untuk menghitung estimasi usaha, koefisien A dan B menjadi parameter penyeimbang. Sedangkan, untuk time development, menggunakan koefisien C dan D. Selain itu, gambar 4 menunjukkan nilai MMRE kinerja estimasi menurut empat jenis skenario. Percobaan menunjukkan bahwa nilai

terbaik dicapai pada 2000 iterasi. Jadi, itu berarti nilai prediksi dari metode yang diusulkan lebih dekat dengan upaya yang sebenarnya.

Tabel 1. Perbandingan MMRE berdasarkan iterasi algoritma kelelawar

Iteration	a	b	MMRE Effort
A=0.5, r0=0.9, alpha=0.95, gamma=0.6, freq = [0 10]			
2500	3.51	0.48	88.00
	1.12	0.21	97.57
	3.74	1.59	1124.82
2000	0.79	0.04	98.81
	2.89	1.25	105.56
	2.49	0.43	91.88
1500	3.22	-0.09	96.25
	3.86	0.83	64.39
	0.84	-0.33	99.35
1000	1.86	-0.40	98.72
	2.89	0.76	77.43
	3.02	0.41	91.05
A=1, r0=0.5, alpha=0.95, gamma=0.99, freq = [0, 2]			
2500	0.96	1.68	412.39
	0.95	0.85	87.71
	0.48	-0.21	99.55
2000	1.56	-0.98	99.47
	2.45	1.65	987.51
	3.06	1.22	93.55
1500	0.89	0.39	97.01
	1.68	-0.72	99.25
	1.87	0.85	79.30
1000	0.55	-0.94	99.81
	2.06	0.17	95.91
	2.16	1.04	62.09

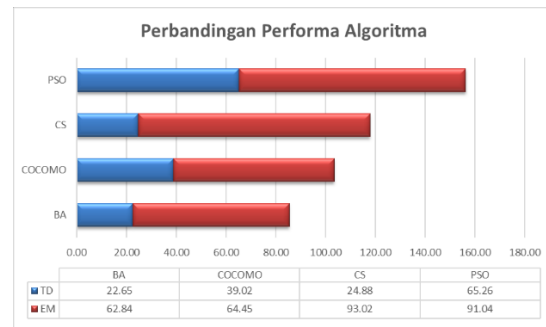
A=1, r0=0.5, alpha=0.95, gamma=0.99, freq = [0, 2]			
2500	2.52	1.07	57.20
	0.56	0.63	96.31
	3.62	0.41	89.70
2000	1.8	0.73	85.76
	3.79	0.14	93.80
	2.27	0.81	78.52
1500	3.46	0.88	62.84
	3.91	0.96	54.22
	1.54	-0.47	99.04
1000	3.62	0.41	89.70
	1.8	-0.05	97.74
	2.52	1.07	57.20

Tabel 2. Perbandingan MMRE Estimasi TDev berdasarkan iterasi algoritma Kelelawar

Iteration	c	d	MMRE TDev
A=0.5, r0=0.9, alpha=0.95, gamma=0.6, freq = [0 10]			
2500	3.82	0.50	21.60
	3.72	1.05	55.81
	1.49	-0.40	98.97
2000	0.95	0.48	95.19
	2.22	0.67	503.05
	2.48	0.58	48.70
1500	3.95	0.61	64.87
	2.31	-0.16	93.19
	3.26	0.57	90.69
1000	2.23	0.66	91.18
	2.34	0.56	15.03
	0.58	0.66	84.08
A=1, r0=0.5, alpha=0.95, gamma=0.99, freq = [0, 2]			
2500	1.43	0.35	38.34
	2.17	0.51	49.39
	2.42	1.14	96.70
2000	2.18	1.78	97.88
	0.53	0.57	115.06
	0.91	0.29	77.83
1500	3.21	-0.04	83.54
	0.50	0.52	98.69
	1.19	0.73	33.17
1000	2.45	-0.28	61.58
	3.43	1.74	169.24
	3.48	1.18	8620.70
A=1, r0=0.5, alpha=0.95, gamma=0.99, freq = [0, 2]			
2500	0.64	0.71	51.58
	0.4	1.31	77.33
	3.45	0.53	29.39
2000	0.65	0.87	46.29
	0.99	0.81	76.86
	0	1.6	100
1500	2.35	0.52	22.65
	2.87	0.46	33.67
	0.02	2.07	99.94
1000	3.45	0.53	29.39
	3.99	0.59	72.22
	0.64	0.71	51.58

Dengan menggunakan nilai iterasi 2000, metode yang diusulkan dibandingkan dengan

nilai tetap COCOMO dan beberapa algoritma seperti algoritma pencarian cuckoo dan optimasi swarm partikel.



Gambar 3. Perbandingan Performa Algoritma

Percobaan ini dilakukan untuk mengkonfirmasi kemampuan algoritma penyerbukan bunga dibandingkan algoritma metaheuristik lainnya dalam menemukan nilai estimasi usaha yang optimum. Gambar 3 menunjukkan bahwa metode yang diusulkan lebih unggul dengan mencapai nilai MMRE yang lebih rendah daripada algoritma lainnya.

5. KESIMPULAN

Setelah diimplementasikan, hasil penelitian menunjukkan bahwa metode yang diusulkan lebih unggul dengan mencapai nilai MMRE yang lebih rendah daripada algoritma lainnya. Hasil eksperimen menunjukkan bahwa algoritma dapat lebih unggul dibandingkan dengan metode particle swarm optimization dan metode cuckoo search algorithm dengan skor MMRE estimasi biaya sebesar 22.65% dan MMRE estimasi pengembangan waktu sebesar 62.84%.

DAFTAR PUSTAKA

Aljahdali S, Sheta AF. Software effort estimation by tuning COOCMO model parameters using differential evolution. In: ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010. ; 2010:1-6. doi:10.1109/AICCSA.2010.5586985.

Chalotra S, Sehra SK, Brar YS, Kaur N. Tuning of COCOMO model parameters by using Bee Colony Optimization. Indian J Sci

- Technol. 2015;8(14).
doi:10.17485/ijst/2015/v8i14/70010.
- Ghatasheh N, Faris H, Aljarah I, Al-sayyed RMH. Optimizing Software Effort Estimation Models Using Firefly Algorithm. *J Softw Eng Appl.* 2015;8(March):133-142.
- Jafari SMS, Ziaaddini F. Optimization of Software Cost Estimation using Harmony Search Algorithm. In: 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC). ; 2016:131-135. doi:10.1007/978-3-662-47926-1.
- Madheswaran M, Sivakumar D. Enhancement of Prediction Accuracy in Cocomo Model for Software Project Using Neural Network. In: 5th International Conference on Communication and Network Technology. ; 2014.
- Parwita IMM, Sarno R, Puspaningrum A. Optimization of COCOMO II Coefficients using Cuckoo Optimization Algorithm to Improve The Accuracy of Effort Estimation. In: International Conference on Information, Communication Technology and System (ICTS), 2016 Int. ; 2017.
- Rijwani P, Jain S. Enhanced Software Effort Estimation Using Multi Layered Feed Forward Artificial Neural Network Technique. *Procedia Comput Sci.* 2016; 89:307-312. doi:10.1016/j.procs.2016.06.073.
- Sachan RK, Nigam A, Singh A, et al. Optimizing Basic COCOMO Model Using Simplified Genetic Algorithm. *Procedia Comput Sci.* 2016;89:492-498. doi:10.1016/j.procs.2016.06.107.
- Sarno R, Sidabutar J, Sarwosri. Improving the Accuracy of COCOMO's Effort Estimation Based on Neural Networks and Fuzzy Logic Model. In: International Conference on Information, Communication Technology and System. ; 2015:197-202. doi:10.1109/ICTS.2015.7379898.
- Sarno R, Sidabutar J, Sarwosri. Comparison of different Neural Network architectures for software cost estimation. In: 2015 International Conference on Computer, Control, Informatics and Its Applications (IC3INA). ; 2015:68-73. doi:10.1109/IC3INA.2015.7377748.
- Zare F, Khademi Zare H, Fallahnezhad MS. Software effort estimation based on the optimal Bayesian belief network. *Appl Soft Comput.* 2016;49:968-980. doi:10.1016/j.asoc.2016.08.004.