

Perbandingan Metode Untuk Meningkatkan Akurasi COCOMO II Dalam Proyek Perangkat Lunak

*Rahmi Rizkiana Putri¹, Zuli Maulidati²

¹Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya, Indonesia

²Sistem Informasi, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya, Indonesia

Diterima: 1 Desember, 2025 | Revisi: 30 Januari 2026 | Diterbitkan: 30 Februari 2026

DOI: <https://doi.org/10.33005/scan.v21i1.5625>

ABSTRAK

Estimasi biaya adalah faktor penting yang memengaruhi keberhasilan proyek pengembangan perangkat lunak. Model Constructive Cost Model II (COCOMO II) banyak digunakan karena mampu memberikan estimasi biaya yang relatif akurat. Namun, dalam praktiknya masih sering ditemukan perbedaan antara hasil estimasi dan biaya aktual proyek. Oleh karena itu, diperlukan pendekatan optimasi untuk meningkatkan tingkat akurasi model tersebut. Penelitian ini bertujuan membandingkan kinerja metode COCOMO II yang dikombinasikan dengan beberapa algoritma optimasi, yaitu *Fuzzy Gaussian Grey Wolf Optimization (FG-GWO)*, *Bat Algorithm (BAT)*, dan *Hybrid Grey Wolf Optimization–Particle Swarm Optimization (GWO-PSO)*. Pengujian dilakukan menggunakan dataset Turkish software project untuk mengevaluasi tingkat akurasi estimasi biaya. Metode evaluasi yang digunakan adalah *Mean Magnitude of Relative Error (MMRE)* untuk mengukur kesalahan estimasi. Hasil penelitian menunjukkan bahwa kombinasi metode COCOMO II dengan pendekatan hybrid GWO–PSO menghasilkan nilai MMRE terbaik dengan penurunan hingga 29,33%. Temuan ini menunjukkan bahwa integrasi algoritma optimasi mampu meningkatkan akurasi estimasi biaya pada proyek pengembangan perangkat lunak secara signifikan.

Kata Kunci: Biaya, COCOMO II, GWO, Perkiraan Perangkat Lunak, PSO

Comparison of Methods for Improving the Accuracy of COCOMO II in Software Projects

ABSTRACT

Cost estimation is a crucial factor influencing the success of software development projects, as it affects project planning, resource allocation, and budgeting. The Constructive Cost Model II (COCOMO II) is widely used because it provides relatively reliable cost estimation results. However, in practice, discrepancies between estimated costs and actual project costs are still frequently observed. Therefore, optimization techniques are required to improve the accuracy of the estimation model. This study aims to evaluate the performance of the COCOMO II model integrated with several swarm-based optimization algorithms, namely *Fuzzy Gaussian Grey Wolf Optimization (FG-GWO)*, *Bat Algorithm (BAT)*, and *Hybrid Grey Wolf Optimization–Particle Swarm Optimization (GWO-PSO)*. The experiments were conducted using the Turkish software project dataset to assess the accuracy of cost estimation. The evaluation metric used in this study is the *Mean Magnitude of Relative Error (MMRE)*. The results show that the hybrid COCOMO II–GWO–PSO approach achieves the best performance by reducing the MMRE value by up to 29.33%.

Keywords: Cost, COCOMO II, GWO, software estimation, PSO

How to Cite:

Putri, R. R., & Maulidati, Z. (2026). Perbandingan Metode Untuk Meningkatkan Akurasi COCOMO II Dalam Proyek Perangkat Lunak. *Scan : Jurnal Teknologi Informasi dan Komunikasi*, 21(1), 43-53

*Corresponding Author:

Email : rahmi@itats.ac.id

Alamat : Jl. Arief Rachman Hakim No.100,
Surabaya



This article is published under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

PENDAHULUAN

Tahap inisiasi dalam manajemen proyek pengembangan perangkat lunak merupakan fase yang sangat penting. Tahap ini bertujuan menilai kelayakan proyek dan memastikan bahwa organisasi mampu melaksanakannya [1]. Salah satu aktivitas utama pada fase ini adalah melakukan estimasi usaha, waktu, dan biaya pengembangan. Proses estimasi memiliki peran yang krusial karena membantu manajer proyek menentukan kebutuhan sumber daya serta jadwal pelaksanaan. Tugas ini tidak boleh diremehkan, sebab kesalahan estimasi dapat berujung pada kegagalan proyek. Hasil survei di Eropa [2] menunjukkan bahwa 65% kegagalan proyek disebabkan oleh metode estimasi yang kurang tepat.

Berbagai model estimasi usaha dan waktu pengembangan perangkat lunak telah dikembangkan untuk mengatasi permasalahan tersebut, salah satunya adalah *Constructive Cost Model* (COCOMO) II yang banyak digunakan oleh Boehm [3]. COCOMO II merupakan model estimasi berbasis regresi yang membantu manajer proyek memperkirakan usaha dan biaya pengembangan. Model ini merupakan pengembangan dari COCOMO 81 [4]. Untuk menghasilkan estimasi, COCOMO II memanfaatkan *Kilo Line of Code* (KLOC) dan *Cost Drivers*. Secara keseluruhan terdapat 22 *Cost Drivers*, terdiri atas 17 *Effort Multiplier* (EM) dan 5 *Scale Factor* (SF), [5] yang nilainya ditetapkan oleh manajer proyek. Berbeda dari metode yang hanya bergantung pada jumlah baris kode, COCOMO II juga mempertimbangkan faktor eksternal yang memengaruhi proyek.

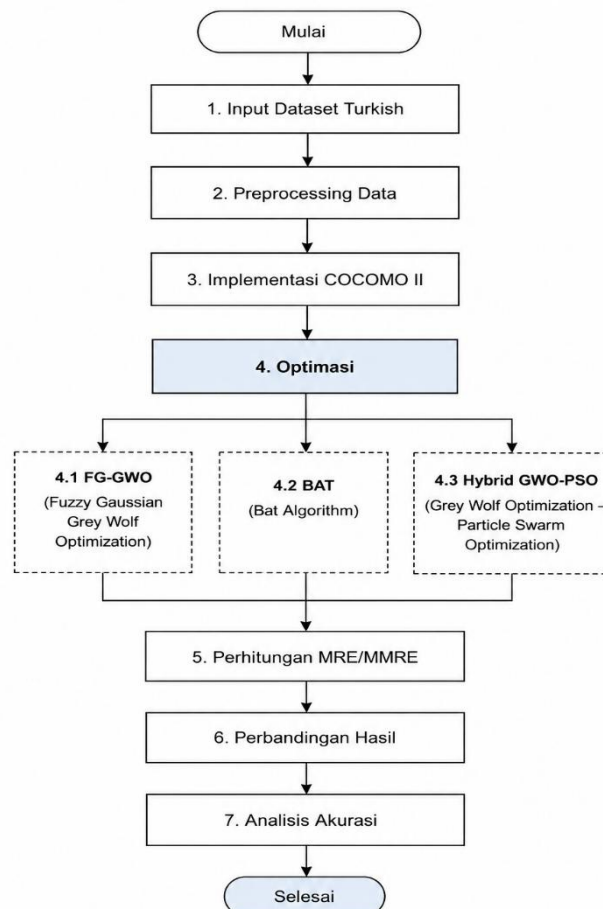
Banyak penelitian sebelumnya menerapkan algoritma berbasis swarm sebagai metode optimasi, seperti menggunakan *Genetic Algorithms* (GA) untuk meningkatkan kinerja berbagai model [6], termasuk M5P, *Multi-Layer Perceptron* (MLP), dan *Support Vector Machine* (SVM), dengan mengoptimalkan fitur terpilih dan parameter menggunakan teknik pemodelan pembelajaran. Secara khusus, mereka membandingkan dua kernel berbeda, yaitu linear dan radial basis function, pada model SVM untuk estimasi biaya perangkat lunak [7]. Memanfaatkan *Particle Swarm Optimization* (PSO) untuk meningkatkan akurasi metode *Analogy-based Estimation* (ABE) dalam mengidentifikasi proyek yang serupa. Mereka menggunakan algoritma PSO untuk pembobotan fitur guna mengatasi masalah dalam mengidentifikasi proyek serupa secara tepat pada metode ABE [8]. Algoritma *Bee Colony Optimization* (BCO) digunakan untuk mengoptimalkan parameter model COCOMO I [9]. Pendekatan ini, yang membagi tahap non-heuristik ke dalam berbagai kategori termasuk pemrograman Integer dan Dinamis serta teori graf, menghasilkan keluaran estimasi yang lebih baik dibandingkan COCOMO I dan COCOMO II [10].

Namun demikian, belum banyak penelitian yang secara langsung membandingkan perbedaan karakteristik dan nilai estimasi yang dihasilkan masing-masing metode dalam lingkungan pengujian yang sama. Meskipun berbagai penelitian sebelumnya telah menerapkan algoritma metaheuristik untuk meningkatkan akurasi estimasi perangkat lunak, sebagian besar penelitian hanya berfokus pada satu metode optimasi tertentu dan belum melakukan analisis komparatif secara langsung dalam lingkungan pengujian yang sama. Selain itu, penelitian sebelumnya umumnya hanya mengevaluasi kemampuan optimasi tanpa mengkaji efektivitas pendekatan hybrid dalam menurunkan nilai error estimasi secara signifikan. Oleh karena itu, penelitian ini mengusulkan perbandingan beberapa pendekatan optimasi berbasis *swarm*

intelligence, termasuk metode hybrid GWO-PSO, untuk mengevaluasi efektivitasnya dalam meningkatkan akurasi model COCOMO II pada dataset Turkish Project. Penelitian ini bertujuan menganalisis perilaku setiap metode optimasi tersebut pada kondisi uji yang seragam. Makalah ini menyajikan perbandingan tiga algoritma berbasis swarm seperti *Bat Algorithm*, *Fuzzy Gaussian Grey Wolf Optimization*, dan Hybrid GWO-PSO yang diimplementasikan dan diuji menggunakan dataset Turkish Project. Hasil yang diperoleh dibandingkan berdasarkan nilai *Magnitude Relative Error* (MRE) untuk setiap proyek serta *Mean Magnitude Relative Error* (MMRE) untuk masing-masing dataset.

METODE PENELITIAN

Bagian ini menjelaskan metodologi yang diterapkan dalam proses implementasi dan evaluasi algoritma. Pertama, dipaparkan secara ringkas deskripsi himpunan data yang digunakan dalam penelitian, yaitu Turkish Project. Kedua, dijelaskan perangkat lunak dan pustaka pemrograman yang dimanfaatkan pada tahap implementasi. Ketiga, diuraikan pendekatan yang digunakan untuk menilai hasil yang diperoleh dari proses implementasi. Kajian-kajian sebelumnya yang membahas *Software Development Effort Estimation* (SDEE) secara umum dapat dikelompokkan ke dalam tiga pendekatan utama, yaitu pendekatan non-algoritmik, algoritmik, dan berbasis pembelajaran mesin. Flowchart yang dilakukan dalam penelitian ini, dapat dilihat pada Gambar 1.



Gambar 1. Alur penelitian

Pendekatan non-algoritmik atau non-parametrik mencakup metode Expert Judgment [11], Analogy-Based Estimation, Price-to-Win, Top-Down, Bottom-Up, serta Wideband Delphi. Pendekatan ini melakukan estimasi usaha pengembangan perangkat lunak dengan memanfaatkan pengalaman praktisi maupun data historis dari proyek-proyek terdahulu. Pendekatan algoritmik melibatkan sejumlah model seperti *Source Lines of Code* (SLOC), *Function Point*, *Object Point*, *Constructive Cost Model* (COCOMO), dan *Use Case Points* (UCP) [12]. Model-model tersebut menggunakan formulasi matematis tertentu untuk menghitung estimasi biaya dan usaha pada proyek perangkat lunak [13].

Dataset

Pada penelitian ini digunakan satu himpunan data, yaitu Turkish project, untuk menganalisis kinerja setiap algoritma pada konteks data yang berbeda. Kedua himpunan data tersebut memiliki karakteristik yang serupa, meliputi atribut project ID, *Effort Multiplier* (EM), *Scale Factor* (SF), *Lines of Code* (LoC), serta *Actual Effort* yang dinyatakan dalam *person-months* [14]. Perbedaan utama hanya terletak pada jumlah observasi.

Tabel 1
Dataset Turkish

ID	EM	SF	LOC	Actual Effort
1	0.35076609	19.9	3000	1.2
	5	2		
2	0.45384183	18.8	2000	2
	5	3		
3	0.64730812	18.6	4250	4.5
	6	8		
4	1.12134958	10.3	10000	3
	7	1		
5	1.08410286	19.2	15000	4
	1	8		
6	0.23786807	8.41	40530	22
7	0.19650911	7.42	4050	2
	2			
8	1.08372096	19.7	31845	5
	4	3		
9	0.37338309	27.2	11428	18
	2	3	0	
1	0.64996474	20.8	23106	4
0	7	2		
1	0.22501369	15.3	1369	1
1	9	6		
1	0.41091405	19.1	1611	2.1
2	8	1		

Dataset Turkish project memuat 12 proyek yang berasal dari 5 perusahaan perangkat lunak seperti yang terlihat pada Tabel 1 menyajikan data Turkish Project.

Constructive Cost Model II

Model COCOMO yang diperkenalkan oleh Boehm pada tahun 1981 merupakan model berbasis regresi yang dirancang untuk membantu pengguna dalam menganalisis implikasi biaya dan jadwal dari berbagai keputusan pengembangan perangkat lunak [15]. Model ini memungkinkan pengambilan keputusan terkait investasi, penetapan anggaran dan jadwal proyek, negosiasi dengan klien, permintaan perubahan, pertukaran antara biaya, waktu, kinerja, dan fungsionalitas, serta pengelolaan risiko dan peningkatan proses, sebagaimana dijelaskan dalam [13]. Secara ringkas, COCOMO berfungsi sebagai suatu rumus untuk memperkirakan biaya pengembangan perangkat lunak melalui estimasi usaha (*effort*) dan waktu pengembangan.

Selanjutnya, pada tahun 2000, diperkenalkan COCOMO II sebagai pengembangan dari COCOMO 81 untuk mengatasi berbagai keterbatasan yang terdapat pada model sebelumnya. Perbedaan utama antara COCOMO II dan COCOMO 81 terletak pada struktur dan komponen perhitungannya. COCOMO II memiliki tiga model perhitungan, yaitu *Application Composition Model*, *Early Design Model*, dan *Post-Architecture Model*, serta menggunakan 17 *Effort Multiplier* dan 5 *Scale Factor*. Sementara itu, COCOMO 81 hanya memiliki tiga model sistem, yakni *Organic*, *Semi-detached*, dan *Embedded*, dengan 15 Cost Driver pada model Intermediate COCOMO. Berikut ini ditampilkan persamaan COCOMO II yang digunakan untuk estimasi usaha pengembangan perangkat lunak [16].

$$PM = A (KLOC)^E \prod_{1}^{17} EM$$

Dimana,

$$E = B + 0,01 \sum_{1}^{5} SF$$

PM (*Person-Month*) merepresentasikan besarnya usaha yang dibutuhkan dalam pengembangan perangkat lunak. Parameter a merupakan faktor kalibrasi, sedangkan KLOC menyatakan ukuran perangkat lunak yang diukur berdasarkan jumlah baris kode. Parameter b berfungsi sebagai faktor skala. Selanjutnya, EM adalah hasil akumulasi pengali usaha yang memengaruhi besarnya usaha dan waktu pengembangan, sementara SF digunakan untuk melakukan penskalaan terhadap nilai KLOC. Penentuan nilai EM dan SF didasarkan pada penilaian para ahli (*expert judgement*) sebagaimana dijelaskan dalam [13]. Sementara itu, untuk menghitung waktu pengembangan perangkat lunak, digunakan persamaan (3) yang ditunjukkan sebagai berikut:

$$TDEV=c(PM^F)$$

Dimana,

$$F = D + 0,2(E - B)$$

TDEV menunjukkan waktu pengembangan perangkat lunak, PM merepresentasikan besarnya usaha pengembangan, dan F merupakan faktor skala. Setiap konstanta dalam model memiliki nilai tertentu, yaitu $A = 2,94$, $B = 0,91$, $C = 3,67$, dan $D = 0,28$.

Grey Wolf Optimization

Seluruh *cost driver* dalam COCOMO II diintegrasikan ke dalam algoritma GWO yang telah disempurnakan setelah melalui beberapa tahap modifikasi. Proses ini dilakukan melalui sejumlah langkah [18]. Pertama, dilakukan inisialisasi seluruh individu (X_i). Kedua, dilakukan inisialisasi parameter a , A , dan C untuk “mengurung” mangsa, yang menekankan perhitungan setiap individu terhadap nilai fitness MRE sebanyak n kali. Kemudian dilakukan pemilihan individu menggunakan metode tournament selection dengan mempertimbangkan nilai MRE terkecil. Ketiga, dihitung nilai fitness MRE dari individu acak yang diperoleh melalui tournament selection. Dengan menggunakan nilai A dan B dari individu terpilih, analisis selanjutnya menghasilkan x_a , x_b , dan x_d pada setiap iterasi, di mana X_α , X_β , dan X_δ masing-masing merepresentasikan agen pencarian terbaik pertama, kedua, dan ketiga [17], [18]. Berdasarkan [19] hal-hal yang menjadi catatan penting dalam *Grey Wolf Optimization* yaitu:

- Tingkatan sosial dalam GWO digunakan untuk menyimpan solusi terbaik dalam rangkaian iterasi.
- Mekanisme mengitari musuh menetapkan solusi dalam area berbentuk lingkaran dimana dapat dikembangkan ke dimensi yang lebih besar.
- Nilai random A dan C membantu kandidat solusi dalam lingkup yang lebih besar dengan nilai jari-jari yang *random*.
- Metode berburu memperbolehkan kandidat solusi untuk menempatkan posisi mangsa yang paling memungkinkan.
- Eksplorasi dan eksploitasi ditanggung oleh nilai adaptif dari a dan A .
- Nilai adaptif dari parameter a dan A memperbolehkan GWO untuk melakukan transisi secara halus saat antara eksplorasi dan eksploitasi terjadi.
- Dengan turunnya nilai A , sebagian dari iterasi bekerja dalam eksplorasi ($|A|= 1$) dan sebagian lainnya bekerja dalam eksploitasi ($|A|< 1$).
- GWO hanya memiliki 2 parameter utama yang harus diatur sesuai keadaan yaitu nilai a dan C .

Particle Swarm Optimization

Particle Swarm Optimization (PSO) merupakan algoritma optimasi evolusioner bersifat stokastik yang berlandaskan konsep kecerdasan kawanan, pertama kali diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 [18]. Algoritma ini memiliki beberapa keunggulan, antara lain jumlah parameter yang relatif sedikit, kecepatan konvergensi yang tinggi, serta tingkat akurasi yang baik. Oleh karena itu, PSO banyak diterapkan dalam berbagai bidang, seperti perencanaan lintasan, optimasi parameter, dan sistem tenaga listrik. Seiring berkembangnya algoritma PSO standar berbagai penelitian telah dilakukan untuk mengkaji dan menyempurnakan metode tersebut. Penelitian yang dilakukan yaitu sifat konvergensi PSO dengan menerapkan faktor kontraksi, sehingga

proses pencarian solusi optimal menjadi lebih stabil. Adapun penelitian yang mengusulkan penggunaan bobot inersia dan faktor pembelajaran adaptif guna meningkatkan kemampuan PSO dalam menyelesaikan permasalahan optimasi yang kompleks. Beberapa peneliti juga mengombinasikan teori chaos dengan PSO yang telah dimodifikasi untuk meningkatkan kemampuan pencarian global serta efisiensi optimasi. Selain itu, prinsip mekanika kuantum turut diintegrasikan ke dalam PSO untuk memodifikasi mekanisme pembaruan partikel, sehingga distribusi partikel dalam ruang pencarian menjadi lebih seimbang [20].

PSO merupakan algoritma optimasi yang terinspirasi dari perilaku kawanan burung di alam. Dalam PSO, solusi potensial direpresentasikan sebagai partikel yang bergerak di dalam ruang pencarian untuk memperoleh solusi terbaik. Pergerakan partikel dinyatakan sebagai kecepatan, yang menyebabkan perubahan posisi partikel dari waktu ke waktu. Kecepatan partikel diperbarui secara stokastik berdasarkan posisi terbaik yang pernah dicapai oleh partikel itu sendiri dan posisi terbaik di lingkungan sekitarnya. Proses pembaruan ini dilakukan menggunakan persamaan tertentu agar partikel dapat mendekati solusi optimal. Setiap partikel ke- i didefinisikan oleh vektor posisi x_i dan vektor kecepatan v_i [18]

$$v_i^{k+1} = v_i^k + c_1 r_1 (Pbest_i^k + x_i^k) + c_2 r_2 (gbest - x_i^k)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

Implementasi

Dalam penelitian ini, pola lama yang sering digunakan oleh COCOMO II dalam berbagai proyek perangkat lunak dipertimbangkan dan diimplementasikan. Pendekatan metode optimalisasi ini dikenal sebagai serangkaian aturan GWO yang digunakan untuk menyempurnakan COCOMO II. Perbandingan evaluasi juga disajikan antara output yang diprediksi dan yang sebenarnya kemudian dilakukan analisis terhadap hasil pengujian.

1. Pengujian COCOMO II-Fuzzy Gaussian GWO

Uji coba ini dilakukan dengan menggunakan nilai upaya pengganda fuzzy Gaussian yang dioptimalkan melalui metode GWO. Dari konteks ini, optimalisasi dilakukan dengan menghitung nilai kebugaran menggunakan MRE COCOMO II. Setelah menentukan nilai terbaik untuk masing-masing individu, perkiraan biaya juga dihitung melalui persamaan PM dan MRE COCOMO II. Dalam hal ini, pemilihan nilai terkecil dari setiap individu yang diprioritaskan MRE COCOMO II. Oleh karena itu, COCOMO II MMRE dihitung untuk mendapatkan nilai kesalahan terkecil, yang merupakan koefisien optimal.

2. Pengujian COCOMO II-BA

Percobaan kedua dengan kumpulan data Proyek Turki memiliki sifat acak dari algoritma metaheuristik dalam menemukan solusi yang optimal. Dalam percobaan ini, Algoritma Kelelawar mencapai solusi optimal pada iterasi ke-420.

3. Pengujian Optimalisasi COCOMO II GWO hybrid PSO

Percobaan ketiga yaitu melakukan inialisasi terhadap partikel secara acak, menghitung effort estimation COCOMO II, menghitung fitness COCOMO II, melakukan perhitungan

fitness menggunakan MMRE, melakukan update kecepatan PSO dan posisi, melakukan local search sederhana pada gBest hingga mencapai iterasi maksimum atau konvergen.

HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk melakukan analisis komparatif terhadap tiga algoritma optimasi, yaitu COCOMO II-Fuzzy Gaussian GWO, COCOMO II-Bat Algorithm, dan Optimalisasi COCOMO II GWO hybrid PSO, untuk menurunkan tingkat kesalahan estimasi serta meningkatkan akurasi prediksi usaha pengembangan perangkat lunak. Hasil eksperimen menunjukkan bahwa seluruh algoritma yang diuji memberikan kinerja estimasi yang lebih baik dibandingkan model COCOMO II asli tanpa proses optimasi. Nilai MMRE awal pada model COCOMO II menunjukkan error estimasi yang relatif tinggi yaitu sebesar 733,14% sebelum proses optimasi dilakukan. Setelah penerapan metode optimasi, nilai error berhasil diturunkan secara signifikan.

1. Pengujian COCOMO II-Fuzzy Gaussian GWO

Uji coba pertama menerapkan COCOMO II-Fuzzy Gaussian GWO yang melakukan mekanisme pemilihan individu melalui tournament selection untuk memperoleh koefisien A dan B terkecil pada tahap berikutnya. Hasil pengujian menggunakan Turkish yang berpengaruh terhadap hasil MMRE menjadi lebih optimal, seperti yang dapat dilihat pada Tabel 2. Berdasarkan sejumlah studi terdahulu yang melakukan pengujian menggunakan berbagai metode antara lain tabu search, ANFIS, hybrid K-Means PSO, serta HACO-BA telah dilakukan analisis perbandingan dengan beragam Teknik dan model, seperti Doty, Bailey, Halstead, ACO, BA, DNN, dan COCOMO II. Tingginya nilai MMRE sebesar 733,14% menunjukkan bahwa model COCOMO II standar masih menghasilkan overestimation pada beberapa proyek di dataset Turkish. Hal ini dipengaruhi oleh sensitivitas parameter Cost Driver, Scale Factor, dan variasi karakteristik proyek yang menyebabkan selisih estimasi effort terhadap actual effort menjadi sangat besar pada beberapa data uji.

Tabel 2.
COCOMO II-Fuzzy Gaussian GWO

Metode	Datase t	MMRE (%)
COCOMO II		733,14
COCOMO II-Fuzzy Gaussian GWO	Turkish	51,08

2. Pengujian COCOMO II-BAT

Uji coba kedua menggunakan dataset Turkish menunjukkan bahwa seluruh algoritma menghasilkan estimasi lebih baik dibandingkan model COCOMO II asli. Berdasarkan pengujian tersebut, *Bat Algorithm* memiliki nilai MMRE lebih kecil dibandingkan COCOMO II. Perbedaan tersebut disebabkan oleh sifat acak pada algoritma metaheuristik dalam mencari solusi optimal, seperti yang dapat dilihat pada Tabel 3.

Tabel 3
COCOMO II-BAT

Metode	Datase t	MMRE (%)
--------	-------------	-------------

COCOMO II		733,14
COCOMO II-BAT	Turkish	34,19

3. Pengujian COCOMO II GWO hybrid PSO

Uji coba ketiga menggunakan COCOMO II GWO hybrid PSO menunjukkan bahwa seluruh algoritma menghasilkan estimasi lebih baik dibandingkan model COCOMO II asli. Berdasarkan pengujian tersebut, GWO hybrid PSO memiliki nilai MMRE lebih kecil dibandingkan metode COCOMO II, COCOMO II-Fuzzy Gaussian, dan COCOMO II-BAT. Penurunan nilai kesalahan estimasi yang signifikan diperoleh setelah penerapan tahapan optimasi secara bertahap. Metode awal menunjukkan nilai error yang relatif tinggi, namun setelah proses penyesuaian parameter dan optimasi lanjutan, nilai error akhir berhasil ditekan dengan rata-rata penurunan mencapai 96% dibandingkan kondisi awal, seperti yang dapat dilihat pada Tabel 4.

Tabel 4
COCOMO II GWO hybrid PSO

Metode	Datase t	MMRE (%)
COCOMO II		733,14
COCOMO II GWO hybrid PSO	Turkish	29,33

SIMPULAN

Penelitian ini bertujuan untuk meningkatkan akurasi COCOMO II dalam memperkirakan biaya proyek perangkat lunak dengan mengusulkan metode COCOMO II GWO hybrid PSO. Metode yang diusulkan telah dilakukan perbandingan dengan metode lainnya dengan menggunakan dataset Turkish. Hasil percobaan menunjukkan nilai MMRE pada dataset Turkish mencapai nilai error lebih rendah jika dibandingkan dengan COCOMO II karena terdapat perubahan mekanisme atau hybrid pada metode PSO. Pendekatan metode hybrid memungkinkan menemukan solusi optimal secara efisien dan cepat.

DAFTAR PUSTAKA

- [1] A. Rahman, "IT project management frameworks: Evaluating best practices and methodologies for successful IT project management," *Academic Journal on Science, Technology, Engineering & Mathematics Education*, vol. 1, no. 1, pp. 57–76, Oct. 2024, doi: 10.69593/ajaimldsms.v1i01.128.
- [2] S. S. Ali, J. Ren, and J. Wu, "Framework to improve software effort estimation accuracy using novel ensemble rule," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 9, Nov. 2024, Art. no. 102189, doi: 10.1016/j.jksuci.2024.102189.
- [3] B. W. Boehm and R. Madachy, *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. [Online]. Available: <http://dl.acm.org/citation.cfm?id=557000>

- [4] A. A. D. P., L. Azizah, and Y. U. L., "Analisis ekonomi perangkat lunak dalam pengembangan sistem informasi Pondok Pesantren Islamic Center Elkisi (SIP-E)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 1, pp. 12–22, May 2022.
- [5] V. Jaglan, "Apply fuzzy optimization in proficient managing COCOMO model cost drivers," *International Journal of Recent Aspects*, vol. 3, no. 1, pp. 20–26, 2016.
- [6] S. Asghari, A. Dizaj, and S. Gharehchopogh, "A new approach to software cost estimation by improving genetic algorithm with bat algorithm," *Journal of Computer & Robotics*, vol. 11, no. 2, pp. 17–30, 2018.
- [7] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," *Information and Software Technology*, vol. 52, no. 11, pp. 1155–1166, 2010, doi: 10.1016/j.infsof.2010.05.009.
- [8] V. Khatibi Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "A PSO-based model to increase the accuracy of software development effort estimation," *Software Quality Journal*, vol. 21, no. 3, pp. 501–526, 2013, doi: 10.1007/s11219-012-9183-x.
- [9] R. Forsati, "A novel approach for feature selection based on the bee colony optimization," *International Journal of Computer Applications*, vol. 43, no. 8, pp. 30–34, 2012.
- [10] S. Chalotra, S. K. Sehra, Y. S. Brar, and N. Kaur, "Tuning of COCOMO model parameters by using bee colony optimization," *Indian Journal of Science and Technology*, vol. 8, no. 14, 2015, doi: 10.17485/ijst/2015/v8i14/70010.
- [11] Z. B. Mansor, Z. M. Kasirun, N. H. H. Arshad, and S. Yahya, "E-cost estimation using expert judgment and COCOMO II," in *Proc. International Symposium on Information Technology (ITSim)*, Kuala Lumpur, Malaysia, 2010, vol. 3, pp. 1262–1267, doi: 10.1109/ITSIM.2010.5561466.
- [12] M. Azzeh, A. B. Nassif, and I. Attili, "Predicting software effort from use case points: A systematic review," Dec. 2020.
- [13] H. L. T. K. Nhung, V. Van Hai, R. Silhavy, Z. Prokopova, and P. Silhavy, "Parametric software effort estimation based on optimizing correction factors and multiple linear regression," *IEEE Access*, vol. 10, pp. 2963–2986, 2022, doi: 10.1109/ACCESS.2021.3139183.
- [14] R. R. Putri, D. O. Siahaan, and S. Sarwosri, "Peningkatan akurasi estimasi usaha dan biaya COCOMO II berdasarkan Gaussian dan BCO," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 6, no. 3, 2017, doi: 10.22146/jnteti.v6i3.333.
- [15] R. Sarno and J. Sidabutar, "Improving the accuracy of COCOMO's effort estimation based on neural networks and fuzzy logic model," in *Proc. International Conference on Information, Communication Technology and Systems (ICTS)*, Surabaya, Indonesia, 2015, pp. 197–202, doi: 10.1109/ICTS.2015.7379898.
- [16] I. C. Suherman, "Implementation of random forest regression for COCOMO II effort estimation," in *Proc. International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2020, pp. 476–481, doi: 10.1109/iSemantic50169.2020.9234269.

- [17] A. Seyyedabbasi and F. Kiani, "I-GWO and Ex-GWO: Improved algorithms of the Grey Wolf Optimizer to solve global optimization problems," *Engineering with Computers*, vol. 37, pp. 509–532, Jan. 2021, doi: 10.1007/s00366-019-00837-7.
- [18] M. A. Elaziz, D. Oliva, and S. Xiong, "Hybrid gray wolf and particle swarm optimization for feature selection," *International Journal of Innovative Computing, Information and Control*, vol. 16, no. 3, pp. 831–844, 2020, doi: 10.24507/ijicic.16.03.831.
- [19] S. Nosratabadi, K. Szell, B. Beszedes, F. Imre, S. Ardabili, and A. Mosavi, "Comparative analysis of ANN-ICA and ANN-GWO for crop yield prediction," in *Proc. RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, doi: 10.1109/RIVF48685.2020.9140786.
- [20] X. Cao, C. Gu, D. Luo, and H. Fu, "Function optimization of two improved particle swarm algorithms," *International Core Journal of Engineering*, vol. 11, no. 12, 2025, doi: 10.6919/ICJE.202512_11(12).0020.