

# Hybrid GWO-PSO for Accurate Software Effort Estimation in COCOMO II

Rahmi Rizkiana Putri<sup>1\*</sup>, Desy Candra Novitasari<sup>2</sup>

<sup>1</sup> Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya, Surabaya, 60117, Indonesia

<sup>2</sup> Bisnis Digital, Fakultas Ekonomi dan Bisnis, Universitas Kepanjen, Malang, 65163, Indonesia

## Informasi Artikel

Diterima : 12 Agustus 2025  
Revisi : 3 September 2025  
Publikasi : 30 September 2025

## Kata Kunci:

Akurasi  
COCOMO II  
GWO  
Proyek Perangkat Lunak  
PSO

## ABSTRAK

Perkiraan usaha yang akurat memberikan landasan penting dalam pengelolaan proyek perangkat lunak secara efektif. Jika estimasi usaha kurang tepat, dapat berdampak pada kualitas manajemen proyek menjadi kurang efisien. Adapun dapat berisiko seperti anggaran yang berlebih, hingga gagal memenuhi jadwal yang ditetapkan. Penelitian ini, mencakup metode hybrid Grey Wolf Optimization (GWO)-Particle Swarm Optimization (PSO) untuk mengoptimalkan hasil estimasi usaha, sehingga menghasilkan perkiraan usaha proyek perangkat lunak yang lebih valid dan reliabel. Penelitian ini menggunakan pendekatan hybrid GWO-PSO untuk meningkatkan kemampuan eksplorasi dan eksploitasi dalam proses optimasi, sehingga dapat meningkatkan akurasi estimasi yang lebih tinggi dibandingkan metode konvensional. Implementasi metode yang diusulkan menunjukkan nilai Mean Magnitude Relative Error (MMRE) sebesar 321,16%, yang mana nilai tersebut lebih rendah 1243,23 dibandingkan hasil COCOMO II konvensional. Hasil uji coba membuktikan bahwa keakuratan estimasi usaha meningkat sehingga memberikan kontribusi signifikan dalam peningkatan efektivitas manajemen proyek perangkat lunak. Dengan demikian, penelitian ini menyediakan kerangka kerja estimasi usaha COCOMO II yang lebih handal dan dapat diadopsi oleh praktisi maupun peneliti untuk memperbaiki proses perencanaan, pengendalian, dan evaluasi proyek perangkat lunak.

## ABSTRACT

Accurate business forecasting provides an important foundation for managing software projects effectively. If the business estimate is not accurate, it can have an impact on the quality of project management to become less efficient. It can be risky such as an excess budget, to failing to meet the set schedule. This research includes the hybrid Grey Wolf Optimization (GWO)-Particle Swarm Optimization (PSO) method to optimize the results of business estimation, thereby resulting in more valid and reliable business estimates of software projects. This study employs a hybrid GWO-PSO approach to enhance the exploration and exploitation capabilities in the optimization process, thereby achieving higher estimation accuracy compared to conventional methods. The implementation of the proposed method showed a Mean Magnitude Relative Error (MMRE) value of 321.16%, which is 1243.23% lower than the results of conventional COCOMO II. The results of the trial prove that the accuracy of business estimates has increased, thus making a significant contribution to improving the effectiveness of software project management. Thus, this study provides a more reliable COCOMO II business estimation framework that can be adopted by practitioners and researchers to improve the planning, control, and evaluation process of software projects.

---

This is an open-access article under the [CC BY-SA](#) license



---

**\*Penulis Koresponden**

Email: [rahmi@itats.ac.id](mailto:rahmi@itats.ac.id)

Cara sitasi IEEE:

R. R. Puti & D.C. Novitasari, "Hybrid GWO-PSO for Accurate Software Effort Estimation in COCOMO II," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 3, pp. 1186-1192, September 2025, doi: 10.30811/jaise.v5i3.7603

---

## 1. PENDAHULUAN

Estimasi perangkat lunak memiliki berbagai masalah dalam pengembangan proyek perangkat lunak, ada satu masalah yang cukup besar dan mempengaruhi perusahaan perangkat lunak itu sendiri saat melakukan manajemen yang baik dalam mengembangkan perangkat lunak. Perkiraan perhitungan usaha dan biaya dalam kegiatan pengembangan proyek perangkat lunak adalah hal yang sangat penting. Estimasi usaha dan biaya adalah proses evaluasi dan estimasi yang dilakukan dalam bentuk jumlah sumber daya yang dibutuhkan untuk mengakhiri kegiatan proyek. Hal ini berguna untuk memberikan layanan agar persyaratan fungsional yang diberikan kepada pelanggan dapat terpenuhi [1]. Selain itu, memperkirakan kualitas perangkat lunak harus dimungkinkan dengan seberapa baik kerangka kerja dapat memenuhi kebutuhan pemangku kepentingan yang berbeda [2]. Ada begitu banyak alasan ketika proyek perangkat lunak tidak dapat diselesaikan tepat waktu ketika seorang manajer proyek perangkat lunak dihadapkan pada produksi yang tidak pasti dan tidak stabil yang sulit dikendalikan. Selain itu, kebutuhan pelanggan dan perkembangan teknologi dan alat yang berubah pesat di bidang ini. Ada juga masalah komunikasi yang dapat menjadi faktor keterlambatan dan kegagalan proyek perangkat lunak. Semua ini membuat prediksi upaya pengembangan menjadi sulit dalam proyek perangkat lunak [3]. Di dalam proyek, ada beberapa fakta tentang aktivitas proyek perangkat lunak. Seperti yang dinyatakan oleh Boehm bahwa, 20% dari waktu proyek dan 8% dari total sumber daya digunakan untuk mengembangkan persyaratan pada *baseline*. Ini menunjukkan bahwa sebagian besar proyek berhasil menggunakan 7% hingga 15% dari total sumber daya proyek perangkat lunak. Atribut proyek yang dimaksud adalah deskripsi karakteristik interkoneksi dengan proyek rekayasa perangkat lunak. Ada 7 atribut yang ditemukan pada metode sebelumnya, yaitu ukuran proyek, kompleksitas proyek, volatilitas persyaratan, kategori proyek, tingkat keamanan, kendala waktu, dan kendala biaya. Setiap atribut memiliki jumlah entri yang telah ditentukan sebelumnya. Metode populer untuk memperkirakan biaya perangkat lunak yang digunakan oleh industri perangkat lunak adalah *Constructive Cost Model (COCOMO II)* yang telah terbukti lebih akurat dengan beberapa peningkatan pada beberapa *cost driver*.

Saat ini, penggunaan algoritma meta-heuristik digunakan untuk keperluan estimasi dan memperoleh hasil yang akurat dalam peningkatan hasil perkiraan. Algoritma meta-heuristik adalah metode yang tepat untuk masalah pengoptimalan. Algoritma genetik biasanya diterapkan sebagai teknik pengoptimalan standar. Perbandingan antara upaya aktual dan perhitungan model standar melalui perhitungan genetik mengungkapkan bahwa kombinasi model standar dan algoritma genetik mengarah pada perkiraan yang dioptimalkan [4]. Beberapa penelitian menggunakan algoritma genetik untuk pemilihan fitur dan untuk mengoptimalkan parameter pembelajaran mesin; yang bertujuan untuk meningkatkan perkiraan cara yang tepat dengan pembelajaran mesin. Optimalisasi teknik perkiraan berbasis analogi menerapkan algoritma pengoptimalan kawanan partikel. Beberapa dari metode ini diperiksa untuk persamaan dan perbedaan dengan regresi bertahap, jaringan saraf buatan, regresi berganda, klasifikasi, dan pohon regresi dan model berbasis analogi ternyata memberikan hasil yang lebih baik daripada metode ini [3]. Penulis mengimplementasikan kombinasi kecerdasan komputasi sebagai dasar untuk pemodelan COCOMO dan algoritma pengoptimalan kawanan partikel untuk perkiraan upaya. Dari hasil pengujian, hasil penggabungan algoritma clustering dengan algoritma PSO lebih akurat untuk mengevaluasi COCOMO dibandingkan dengan model COCOMO primer. Selain itu, model ini juga memperoleh hasil yang lebih baik dibandingkan dengan jaringan saraf buatan, dan

analisis relasional fuzzy abu-abu [4]. Mengoptimalkan model COCOMO 1 menggunakan metode *Bee Colony Optimization* dan membandingkan hasil estimasi COCOMO 1 dengan metode BCO [7].

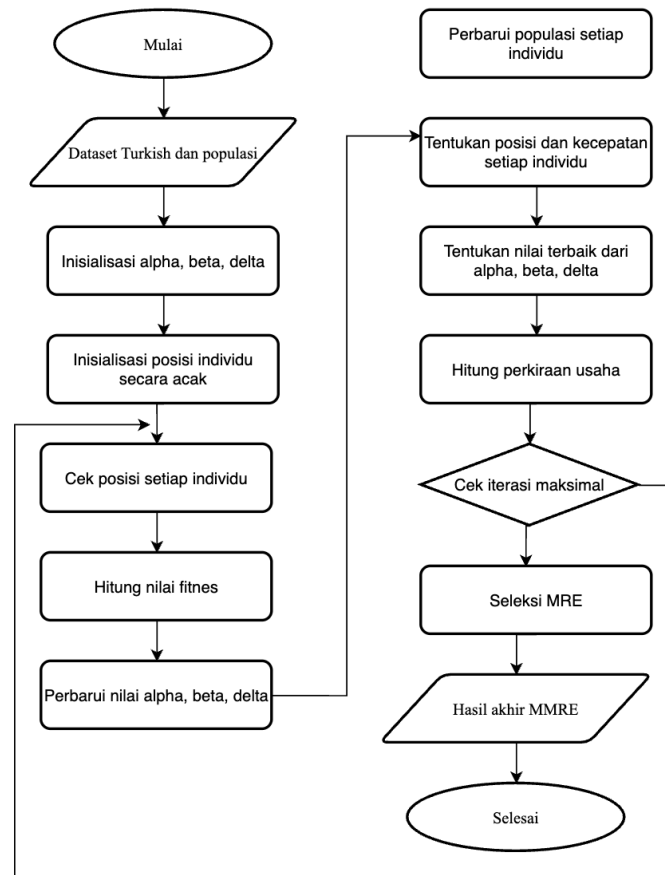
Metode non-heuristik dikelompokkan ke dalam berbagai jenis seperti pemrograman bilangan bulat, pemrograman dinamis, teori grafik, dll. Pemrograman bilangan bulat adalah pemrograman optimasi matematis di mana semua variabel terbatas pada bilangan bulat sedangkan variannya sebagai pemrograman linier berhubungan dengan fungsi objektif yang kendalanya linier. Pemrograman dinamis adalah pemrograman yang menggunakan strategi pengoptimalan berdasarkan pemecahan masalah besar menjadi sub-masalah yang lebih kecil. Komposisi menggunakan pemodelan teori grafik untuk mendapatkan masalah jalur terpendek. Dalam metode pemrograman dinamis ini, setiap kandidat layanan akan mewakili node. Mereka memindahkan parameter QoS dari node ke tepi yang sesuai untuk membangun grafik langsung. Algoritma meta-heuristik juga dikenal sebagai algoritma genetik dan algoritma koloni semut yang bertujuan untuk memecahkan masalah sesuai dengan optimasi. Metode ini dapat sangat membantu dalam mengoptimalkan algoritma pemodelan dan mengoptimalkan faktor estimasi biaya [8]. Optimasi koloni semut juga merupakan metode optimasi yang biasanya digunakan untuk analisis cluster tetapi memiliki waktu komputasi yang terbatas [9]. Selain itu, algoritma meta-heuristik untuk pencarian harmoni telah diterapkan pada berbagai pemecahan masalah. Namun, COCOMO II tidak cukup akurat dalam menghitung estimasi biaya dan upaya jika dibandingkan dengan metode optimasi yang ada. Oleh karena itu dengan keunggulan metode optimasi dan kelengkapan atribut COCOMO II, penelitian ini mengusulkan kombinasi antara COCOMO II dan hybrid *Grey Wolf Optimization-Particle Swarm Optimization* (GWO-PSO) pada tahap proses untuk meningkatkan akurasi biaya dan upaya dalam suatu proyek perangkat lunak.

Penelitian ini juga menggunakan metode GWO yang berfungsi khusus untuk meningkatkan akurasi usaha dan biaya dalam COCOMO II dengan beberapa perubahan nilai kuantitatif pada atribut COCOMO II atau cost driver. Hal ini dikarenakan COCOMO II memberikan nilai akurasi yang tidak cukup akurat jika dibandingkan dengan nilai proyek yang sebenarnya. Selain itu, GWO adalah algoritma metaheuristik baru yang dapat memberikan hasil kompetitif untuk eksplorasi pencarian yang lebih luas dan dapat mencegah terjebak dalam optimum lokal. PSO sebagai algoritma pengoptimalan yang digunakan untuk mengoptimalkan parameter dalam model COCOMO II, sehingga menghasilkan upaya dan biaya pengembangan yang lebih akurat dan tepat. Kontribusi COCOMO II hybrid GWO-PSO digunakan untuk mengontrol eksplorasi (mencari mangsa) dan eksploitasi (menyerang mangsa) menggunakan dataset Turkish yang memiliki atribut sesuai persyaratan COCOMO II karena banyak penelitian terkait estimasi proyek perangkat lunak juga menggunakan dataset tersebut.

## 2. METODE

Algoritma GWO meniru kepemimpinan sosial dan perilaku berburu serigala abu-abu di alam dan memiliki sistem yang ketat serigala abu-abu biasanya hidup berkelompok, di mana dibagi menjadi empat bagian. Serigala pemimpin disebut alfa, tingkat kedua adalah beta, yang ketiga adalah delta, dan bagian bawah piramida disebut omega. Melalui tinjauan terperinci, kriteria yang mempengaruhi biaya proyek dibahas, metode yang dikembangkan sebelumnya untuk upaya, praktik baik, dan praktik buruk secara rinci [17]. COCOMO II telah membahas alat mana yang paling populer untuk mengusahakan biaya dan menggunakan baris kode untuk mengevaluasi ukuran program. COCOMO II diimplementasikan dengan beberapa proyek untuk sebuah solusi. Model ini dikoreksi dalam model algoritma serigala abu-abu dan perbandingan terperinci dibuat antara perkiraan upaya dan upaya aktual yang dilakukan, ditemukan algoritma ini cocok dengan data estimasi upaya dan biaya dan memberikan hasil nyata yang lebih baik [9]. Dalam penelitian ini, kami mengusulkan kumpulan data Turkish. Data yang diperoleh dari Turkish terdiri dari 12 proyek perangkat lunak, Termasuk 25 atribut, 17 pendorong biaya, 5 faktor skala, SLOC, dan upaya aktual yang memiliki atribut yang sama dengan atribut COCOMO II. Perkiraan upaya, waktu pengembangan, model yang diusulkan memberikan MRE dan keseimbangan kesalahan relatif. Metode hybrid GWO-PSO diterapkan ke COCOMO II untuk meningkatkan akurasi biaya di COCOMO II. Penjelasan metodologi kerja yang diusulkan secara rinci dengan menggunakan diagram alur. Penelitian yang diusulkan dilakukan dengan menerapkan COCOMO II kemudian dioptimalkan oleh hybrid GWO-PSO sebagai output.

Tahapan optimasi yang dilakukan sesuai Gambar 1 dengan tahapan berikut.



Gambar 1. Flowchart COCOMO II hybrid GWO-PSO

Tahapan yang pertama, yaitu memasukkan kumpulan data Turkish, menginisialisasi populasi awal jumlah agen (serigala abu-abu),  $a$ ,  $A$ ,  $C$ , posisi, nilai alfa, nilai beta, nilai delta, iterasi maksimum, menghitung nilai fitness menggunakan persamaan (13), menemukan solusi terbaik dari 3 agen selama proses iterasi,  $x-\alpha$ ,  $x-\beta$ ,  $x-\delta$ , menghitung  $a$ ,  $A$ ,  $C$ .

$$A = 2l \cdot r_1 - 1 \quad (13)$$

$$A = 2a \cdot r_1 - a \quad (14)$$

$$C = 2 \cdot r_2 \quad (15)$$

Kemudian perbarui posisi menggunakan persamaan (6), perbarui populasi, menentukan jarak dan percepatan pergerakan setiap individu pada populasi baru menggunakan PSO, hitung nilai fitness (penentuan nilai terbaik), perbarui solusi terbaik dari 3 agen, hitung estimasi biaya, periksa iterasi maksimum, pilih MRE yang digunakan atau tidak untuk proyek. Hasil akhir berupa MMRE. Perolehan solusi dari masalah estimasi biaya perangkat lunak pada metode hybrid GWO-PSO memiliki tujuan untuk mendapatkan tingkat akurasi estimasi biaya proyek perangkat lunak sehingga dapat mendekati biaya proyek yang sebenarnya.

Berikut ini adalah penjelasan mengenai peningkatan akurasi COCOMO II menggunakan metode hybrid GWO-PSO berdasarkan Gambar 1. Proses pertama, inisialisasi kumpulan data Turkish yang diperlukan, total populasi, jumlah maksimum iterasi, nilai parameter (Alpha, Beta, dan Delta), posisi serigala secara acak, dan tingkat sosial. Inisialisasi posisi serigala berdasarkan nilai atribut setiap proyek. Kedua, periksa posisi serigala berdasarkan usaha (E). Inisialisasi posisi serigala dilakukan secara acak pada setiap proyek. Dan juga hitung nilai fitness dengan MRE.

$$MRE = \frac{\text{Predicted value} - \text{Actual Value}}{\text{Actual Value}} \quad (16)$$

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE \quad (17)$$

MRE digunakan untuk menghitung nilai kesalahan setiap proyek dan akan dipilih berdasarkan nilai kesalahan terkecil. Perhitungan nilai fitness berdasarkan proses melingkari musuh dan berburu. Dalam proses mengelilingi musuh, nilai Alpha, Beta, dan Delta yang diperbarui akan diperiksa apakah masih dalam jangkauan atau tidak. Proses berburu digunakan untuk memperbarui posisi serigala hingga iterasi terakhir. Ketiga, perbarui nilai (Alpha, Beta, dan Delta) sesuai dengan hasil proses pengecekan posisi serigala. Pembaruan akan dilakukan sesuai dengan jumlah iterasi. Hasilnya adalah Alpha membawa hasil terbaik pertama, Beta membawa hasil terbaik kedua, dan Delta membawa hasil terbaik ketiga. Nilai yang diperbarui juga akan berdampak pada populasi baru. Keempat, menghitung estimasi biaya dan upaya setiap proyek atau serigala yang terdapat dalam Turkish menggunakan rumus (2) dan (3) yang menghasilkan MRE. Kelima, periksa iterasi maksimum. Jika setiap proyek telah menyelesaikan iterasi untuk memeriksa nilai terbaik pada (Alpha, Beta, dan Delta), iterasi akan berhenti. Tetapi jika belum mencapai batas maksimum, ia akan kembali memproses pemeriksaan posisi serigala.

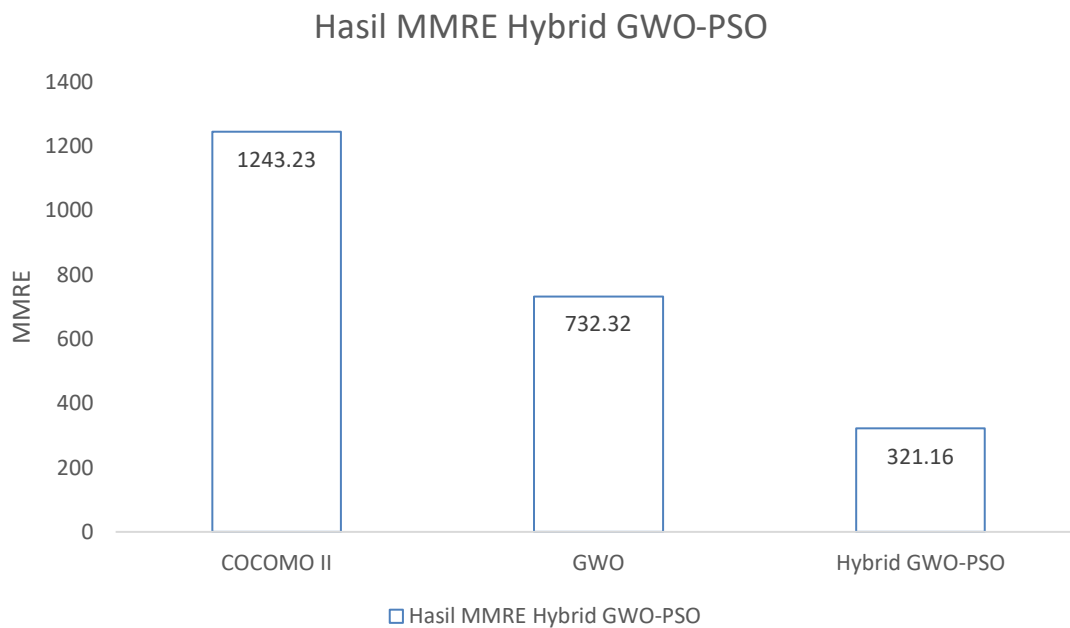
Keenam, pemilihan nilai kesalahan untuk setiap proyek (MRE) akan didasarkan pada nilai kesalahan terkecil. Sementara itu, nilai kesalahan yang besar tidak akan digunakan. Karena akan mempengaruhi total nilai kesalahan akhir (MMRE) dari dataset Turkish. MMRE digunakan untuk mengukur penilaian kinerja cara memperkirakan perangkat lunak. Sedangkan MRE dihitung untuk setiap proyek perangkat lunak. MMRE digunakan untuk menghitung rata-rata proyek N angka yang tersedia dalam kumpulan data [18]. Ketujuh, hasil akhir MMRE. Nilai akhir yang diperoleh adalah nilai kesalahan total (MMRE) dari seluruh proyek, dalam hal ini, Turkish. MMRE adalah hasil evaluasi untuk melihat seberapa besar atau kecil nilai total di seluruh proyek. Apa yang diinginkan dari hasil MMRE adalah perkiraan nilai kesalahan total bisa lebih kecil, sehingga bisa lebih dekat dengan nilai proyek yang sebenarnya. Dalam penelitian ini, pengujian metode hybrid GWO-PSO pada COCOMO II menggunakan dataset Turkish. Nilai MMRE yang lebih kecil yang diperoleh berarti hasil estimasi semakin baik dan mendekati biaya dan upaya proyek yang sebenarnya. MRE dan MMRE mengukur persentase kesalahan lengkap antara biaya yang diprediksi dan aktual untuk setiap proyek pengembangan perangkat lunak referensi seperti yang ditunjukkan pada persamaan (16) dan (17) [19].

### 3. HASIL DAN PEMBAHASAN

Kumpulan data ini menggunakan Turkish dan terdiri dari 12 proyek yang dilakukan antara tahun 1971 dan 1987 (seperti yang diunduh dari PROMISE Repository) yang dikembangkan oleh Barry Boehm, yang terdiri dari 24 atribut yang 15 adalah Effort Multiplier. Melalui tinjauan terperinci, kriteria yang mempengaruhi biaya proyek dibahas, metode yang dikembangkan sebelumnya untuk upaya, praktik baik, dan praktik buruk secara rinci. COCOMO II telah membahas alat mana yang paling populer untuk mengusahakan biaya program dan menggunakan baris kode untuk mengevaluasi ukuran program. Pemodelan COCOMO II yang khas diterapkan dengan beberapa proyek untuk solusi dari permasalahan estimasi proyek. Model ini dikoreksi dalam model algoritma hybrid GWO-PSO dan perbandingan terperinci dibuat antara perkiraan upaya dan upaya sebenarnya yang dilakukan. Setelah menerapkan metode yang diusulkan, langkah selanjutnya adalah melakukan uji coba. Uji coba dilakukan untuk melihat kinerja yang dihasilkan oleh metode yang diusulkan dapat dilihat pada Tabel 1. Data input dalam uji coba ini adalah Turkish yang terdiri dari 12 proyek. Masing-masing terdiri dari 25 atribut, 17 pendorong biaya, 5 faktor skala, SLOC, upaya aktual. Ada 3 skenario pengujian yang dilakukan dalam penelitian ini, yaitu pertama membandingkan hasil akurasi estimasi COCOMO II menggunakan dataset Turkish. Kedua, membandingkan hasil akurasi estimasi biaya COCOMO II GWO menggunakan kumpulan data Turkish [10]. Ketiga, membandingkan hasil akurasi estimasi biaya COCOMO II hybrid GWO-PSO menggunakan kumpulan data Turkish.

Tabel 1. Hasil Uji Coba COCOMO II Hybrid GWO-PSO

Dataset	Metode	MMRE (%)
Turkish	COCOMO II	1243,23
Turkish	GWO	732,32
Turkish	Hybrid GWO-PSO	321,16



Gambar 2. Perbandingan Hasil Uji Coba COCOMO II Hybrid GWO-PSO

Evaluasi uji coba dilakukan dengan membandingkan keakuratan hasil dari akurasi COCOMO II. Pencarian dilakukan hingga memenuhi syarat. Batas pencarian juga dapat dibuat berdasarkan jumlah iterasi yang akan dilakukan menggunakan GWO sedangkan kecepatan individu dalam pencarian jarak, menggunakan metode PSO. Setiap kemungkinan akan dicari untuk mendapatkan nilai terbaik dari parameter perhitungan. Penentuan solusi awal adalah langkah pertama, kemudian dilakukan pencarian kemungkinan solusi baru, kemudian memeriksa apakah solusi baru memenuhi kriteria atau tidak. Jika memenuhi kriteria, kondisi akan berhenti dan nilai individu baru yang terbaik menjadi nilai solusi akhir yang dipilih. Hasil yang dianalisis dapat dilihat pada Tabel I dan Gambar 2 adalah nilai akurasi MMRE terkecil diperoleh dari metode hybrid GWO-PSO yaitu sebesar 321,16%. Dapat dibuktikan hasil uji coba dari metode yang diusulkan memberikan hasil yang lebih akurat daripada metode sebelumnya. Hal ini dikarenakan algoritma hybrid GWO-PSO memiliki 3 parameter, yaitu populasi, iterasi maksimum, dan percepatan pencarian. Berdasarkan hasil pengujian, diketahui bahwa dengan pengaruh parameter terhadap data yang digunakan, parameter populasi dan iterasi maksimum memiliki dampak yang berpengaruh terhadap hasil MMRE yang diperoleh. Dengan nilai parameter populasi yang lebih besar dan nilai parameter iterasi maksimum serta percepatan dalam pencarian, hasil yang diperoleh juga semakin mendekati nilai optimal.

#### 4. KESIMPULAN

Penelitian ini bertujuan untuk meningkatkan akurasi COCOMO II dalam memperkirakan usaha proyek perangkat lunak dengan mengusulkan metode COCOMO II hybrid GWO-PSO. Metode yang diusulkan telah dilakukan perbandingan dengan metode lainnya dengan menggunakan dataset Turkish. Hasil uji coba menunjukkan nilai MMRE pada masing-masing metode yaitu 1243,23% menggunakan model COCOMO II, 732,32% menggunakan metode GWO, dan 321,16% menggunakan metode COCOMO II hybrid GWO-PSO yang merupakan metode yang diusulkan. Sehingga, dapat disimpulkan bahwa penelitian ini menghasilkan estimasi usaha yang lebih akurat dibandingkan dengan COCOMO II dan metode estimasi lainnya.

#### UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada rekan-rekan sejawat dan yayasan pendidikan teknik Surabaya-Institut Teknologi Adhi Tama Surabaya yang telah memberikan saran terhadap penulisan pada penelitian ini.

#### REFERENSI

- [1] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão, and E. Insfran, "An update on effort estimation in agile software development: A systematic literature review," *IEEE Access*, vol. 8, pp. 166768–166800, 2020, doi: 10.1109/ACCESS.2020.3021664.
- [2] Li,B.,Li,Z.,&Yang,Y. "Automatically classifying non-functional requirements using a deep neural network (NFRNet).Information

- and Software Technology.,2022, doi: 10.1016/j.infsof.2022.107185
- [3] Mahmood, Y., Kama, N., Azmi, A., Khan, A. S., & Ali, M. (2021). *Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation*. *Software: Practice and Experience*, 52(1), 39–65. DOI: 10.1002/spe.3009.
- [4] Hameed, S., Elsheikh, Y., & Azzeh, M. (2023). *An optimized case-based software project effort estimation using Genetic Algorithm (CBR-GA)*. *Information and Software Technology*. DOI: 10.1016/j.infsof.2022.107088.
- [5] Setiadi, A., Hidayat, W. F., Sinnun, A., Setiawan, A., Faisal, M., & Alamsyah, D. P. (2021). *Analyze the datasets of software effort estimation with Particle Swarm Optimization*. In 2021 International Seminar on Intelligent Technology and Its Applications (ISITIA) (pp. 197–201). IEEE. DOI: 10.1109/ISITIA52817.2021.9502208.
- [6] Kruti Lavingia, Raj Patel, Vivek Patel, Ami Lavingia. (2024). *Software Effort Estimation using Machine Learning Algorithms. Scalable Computing: Practice and Experience*, 25(2), 1276-1285. DOI: 10.12694/spe.v25i2.2213
- [7] Sharif, H. O., Ghareb, M. I., & Mohamedyusf, H. M. (2024). *A Hybrid Artificial Bee Colony and Artificial Fish Swarm Algorithms for Software Cost Estimation*. *UHD Journal of Science and Technology*, 8(1), 129–141. DOI: 10.21928/uhdjst.v8n1y2024.pp129-141.
- [8] Mustyala, S., Bisi, M., & Azzeh, M. (2024). *Ensembling Harmony Search Algorithm with case-based reasoning for software development effort estimation*. *Cluster Computing* (Springer). DOI: 10.1007/s10586-024-04858-w.
- [9] Hashemi, S. E. (2022). *A New Particle Swarm Optimization Algorithm for Optimizing Big-Data Clustering (Multistart Pattern Reduction-Enhanced PSO)*. *SN Computer Science*, 3, Article 311. Springer. DOI: 10.1007/s42979-022-01208-8
- [10] R. R. Putri, D. O. Siahaan, and S. Sarwosri, "Peningkatan Akurasi Estimasi Usaha dan Biaya COCOMO II Berdasarkan Gaussian dan BCO," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 6, no. 3, 2017, doi: 10.22146/jnteti.v6i3.333.
- [11] Ranković, N., Ranković, D., Ivanović, M., & Lazić, L. (2021). Improved Effort and Cost Estimation Model Using Artificial Neural Networks and Taguchi Method with Different Activation Functions. *Entropy*, 23(7), 854. DOI: 10.3390/e23070854.
- [12] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays," *IEEE Access*, vol. 9, pp. 26926–26936, 2021, doi: 10.1109/ACCESS.2021.3057807.
- [13] A. A. Musa, S. H. Imam, A. Choudhary, and A. P. Agrawal, "Parameter estimation of software reliability growth models: A comparison between grey Wolf optimizer and improved grey Wolf optimizer," *Proc. Conflu. 2021 11th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 611–617, 2021, doi: 10.1109/Confluence51648.2021.9377194.
- [14] F. Yan, X. Xu, and J. Xu, "Grey Wolf Optimizer with a Novel Weighted Distance for Global Optimization," *IEEE Access*, vol. 8, pp. 120173–120197, 2020, doi: 10.1109/ACCESS.2020.3005182.
- [15] Kitonyi, P. M., & Segera, D. R. (2021). *Hybrid Gradient Descent Grey Wolf Optimizer for Optimal Feature Selection*. *BioMed Research International*, 2021, Article ID 2555622. DOI: 10.1155/2021/2555622.
- [16] Xu, L., & Chen, X. (2021). *An improved particle swarm optimization algorithm with adaptive weighted delay velocity*. *International Journal of Swarm Intelligence Research*, 12(3), 45–66. DOI: 10.1080/21642583.2021.1891153.
- [17] Rai, R., Das, A., & Dhal, K. G. (2022). *Nature-inspired optimization algorithms and their significance in multi-thresholding image segmentation: an inclusive review*. *Evolving Systems*, 13, 889–945. DOI: 10.1007/s12530-022-09425-5.
- [18] M. S. Khan, F. Jabeen, S. Ghouzali, Z. Rehman, S. Naz, and W. Abdul, "Metaheuristic Algorithms in Optimizing Deep Neural Network Model for Software Effort Estimation," *IEEE Access*, vol. 9, pp. 60309–60327, 2021, doi: 10.1109/ACCESS.2021.3072380.
- [19] Bajusová, D., Šilhavý, P., & Šilhavý, R. (2024). *Enhancing Software Effort Estimation With Self-Organizing Migration Algorithm: A Comparative Analysis of COCOMO Models*. *IEEE Access*. DOI: 10.1109/ACCESS.2024.3399060.
- [20] A. A. Fadhil, R. G. H. Alsarraj, and A. M. Altaie, "Software Cost Estimation Based on Dolphin Algorithm," *IEEE Access*, vol. 8, pp. 75279–75287, 2020, doi: 10.1109/ACCESS.2020.2988867.