April 2026
Volume 7 Nomor 1

p-ISSN 2723-0325 e-ISSN 2723-0333

TENSOR

Pure and Applied Mathematics Journal

PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PATTIMURA

TENSOR

Pure and Applied Mathematics Journal

is an international academic open access journal that gains a foothold in the field of mathematics and its applications which is issued twice a year. The focus is to publish original research and review articles on all aspects of both pure and applied Mathematics. Submitted papers will be reviewed by editorial board members of the Journal and reviewers. All submitted articles should report original, previously unpublished research results, experimental or theoretical, and will be peer-reviewed. Articles submitted to the journal should meet these criteria and must not be under consideration for publication elsewhere. Manuscripts should follow the style of the journal and are subject to both review and editing.

Published by:

Department of Mathematics,
Faculty of Mathematics and Natural Sciences,
Pattimura University.
Ambon
2026

TENSOR
Pure and Applied Mathematics Journal
Volume 7 Number 1 | April 2026

Publisher
Department of Mathematics,
Faculty of Mathematics and Natural Sciences,
Pattimura University, Ambon, Indonesia

Editorial Address
Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Unversitas Pattimura
Jln. Ir. M. Putuhena, Kampus Unpatti, Poka - Ambon 97233, Provinsi Maluku, Indonesia
Contact : +62 82397854220
Email : tensormathematics@gmail.com

Volume 7 Number 1 | April 2026

TENSOR
Pure and Applied Mathematics Journal

TENSOR Pure and Applied Mathematics Journal

April 2026 | Vol. 7 | No. 1 Page 1–8

p-ISSN: 2723-0325 e-ISSN: 2723-0333

Performance Analysis of Grey Wolf Optimizer for Solving Nonlinear Systems with Complex Roots

Merysa Puspita Sari*, Dewi Ika Ainurrofiqoh, Agustina Pradjaningsih, Sailah Ar Rizka, Nadia Kholifia

Jurusan Matematika, Fakultas MIPA, Universitas Jember, Jember 68121, Indonesia

*E-mail: merysa.fmipa@unej.ac.id

Manuscript submitted : May 2026;
Accepted for publication : June 2026.
DOI : https://doi.org/10.30598/tensorvol4iss1pp1-8

Abstract: Nonlinear systems of equations consist of multiple equations that must be solved simultaneously, and analytical solutions are often difficult to obtain, particularly for complex cases. For this reason, numerical and metaheuristic approaches are frequently employed as practical alternatives. This study investigates the performance of the Grey Wolf Optimizer (GWO) in solving nonlinear systems involving both real and complex roots. The problem is reformulated as an optimization task by minimizing a modulus based objective function derived from the given system. The implementation is carried out in MATLAB using several test cases, and a parameter sensitivity analysis is conducted with respect to the number of search agents, search boundaries, and maximum iterations. To evaluate its performance, the results obtained using GWO are compared with those of the Particle Swarm Optimization (PSO) algorithm reported in previous studies. The findings indicate that GWO is able to produce stable solutions with objective function values close to zero across different cases. However, PSO tends to achieve higher accuracy and faster convergence in certain scenarios. Despite this, GWO demonstrates strong exploration capability, which contributes to its robustness and makes it a viable alternative for solving complex nonlinear systems.

2020 Mathematical Subject Classification: 65H10, 65K10, 90C59.
Keywords: Nonlinear Systems of Equations, Complex Roots, Metaheuristic Algorithms, GWO.

1. Introduction

Nonlinear systems of equations is a collection of equations that must be solved simultaneously. In simple cases, solutions can be obtained using analytical methods. However, for more complex problems, numerical approaches are more frequently used because they can provide efficient approximate solutions [1]. Numerical methods operate iteratively until convergence is achieved with a certain error rate [2]. Some commonly used numerical methods include the Half-Interval, Linear Interpolation, Secant, and Newton-Raphson methods [3]. However, despite their effectiveness, numerical methods are not without limitations. Their performance often depends heavily on the choice of initial values, and in highly nonlinear systems, they may converge to local solutions rather than the desired global solution.

To address these issues, metaheuristic approaches have attracted increasing attention. Compared to classical numerical methods, metaheuristic algorithms generally offer better exploration of the search space, which helps reduce the risk of being trapped in local optima [4, 5]. A number of such algorithms have been applied to nonlinear systems, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Firefly Algorithm (FA), and Cuckoo Search (CS). One algorithm that has gained attention in recent years is the Grey Wolf Optimizer (GWO), which is inspired by the social hierarchy and hunting strategy of grey wolves [6, 7]. Previous studies have shown that GWO has competitive performance compared to other metaheuristic algorithms in various optimization problems [8].

Several related studies have been conducted to solve nonlinear equation systems with complex roots using metaheuristic algorithms. Ariyaratne et al. [9] used a Modified Firefly Algorithm and reported better accuracy compared to the Genetic Algorithm. In another study, Kamsyakawuni et al. [10] compared PSO, FA, and CS, and found that PSO performed best in terms of accuracy and convergence speed. Even so, most existing studies tend to focus on specific algorithms or limited experimental settings, and only a few provide a more comprehensive evaluation across different parameter configurations.

However, studies on the application of the Grey Wolf Optimizer (GWO) algorithm to solve nonlinear equation systems containing complex roots are still limited, particularly in terms of performance analysis and parameter sensitivity. Furthermore, there are not many studies specifically evaluating GWO's ability to produce stable solutions that are close to those of proven superior methods such as PSO.

Based on this, this study aims to analyze the performance of the Grey Wolf Optimizer (GWO) algorithm in solving nonlinear equation systems with real and complex roots. The contributions of this work include: (1) reformulating the nonlinear system as an optimization problem using a modulus based objective function, (2) analyzing the sensitivity of key GWO parameters to assess its robustness and stability, and (3) comparing its performance with PSO in terms of solution accuracy and convergence behavior. The evaluation is carried out using several case studies implemented in MATLAB, providing a more comprehensive understanding of the algorithm’s performance in this context.

2. Research Method

2.1. Nonlinear Systems of Equations

A nonlinear system of equations is defined as a system consisting of n nonlinear equations that must be solved simultaneously [11]. The general form of a nonlinear system can be expressed as follows:

$$f(x) = \begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \tag{1}$$

where $f_1, f_2, \dots, f_n$ are nonlinear functions of the variables $x_1, x_2, \dots, x_n$. The solution of system (1) is a vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ that satisfies all equations simultaneously. To facilitate the application of optimization algorithms, the nonlinear system is transformed into a minimization problem. The objective function is defined as:

$$F(\mathbf{x}) = \sqrt{| f_1(\mathbf{x}) |^2 + | f_2(\mathbf{x}) |^2 + \cdots + | f_n(\mathbf{x}) |^2} \tag{2}$$

The optimization aims to determine a solution vector $\mathbf{x}$ that minimizes the objective function $F(\mathbf{x})$ to a value close to zero, ensuring that the nonlinear system is satisfied. In the case of complex roots, each function value $f_i(\mathbf{x})$ may be a complex number. If $f_i(\mathbf{x}) = a + bi$, then its modulus is defined as [12]:

$$| f_i(\mathbf{x}) | = \sqrt{a^2 + b^2} \tag{3}$$

To handle complex variables, each variable is represented in the form $x_j = a_j + ib_j$, where $a_j, b_j \in \mathbb{R}$. This formulation allows the optimization process to be conducted in a real valued search space while retaining the essential properties of complex solutions.

2.2. Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a metaheuristic optimization algorithm inspired by the hunting behavior of grey wolves (*Canis lupus*), which are known as apex predators in the food chain [13]. Grey wolves live in groups with a well defined social hierarchy consisting of alpha (α), beta (β), delta (δ), and omega (ω). In the context of optimization, each wolf represents a search agent corresponding to a candidate solution $\mathbf{x}$. The quality of each solution is evaluated using the objective function $F(\mathbf{x})$ defined in Equation (2).

The optimization process in GWO is modeled based on three main mechanisms: encircling prey, hunting, and attacking prey [14]. The distance between a search agent and the prey is defined as:

$$\mathbf{D} = | \mathbf{C} \cdot \mathbf{X}_p(t) - \mathbf{X}(t) | \tag{4}$$

The position of the search agent is updated as follows:

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \cdot \mathbf{D} \tag{5}$$

where t denotes the current iteration, $\mathbf{X}_p$ represents the prey position (best solution), and $\mathbf{X}$ is the position of a search agent. The coefficient vectors $\mathbf{A}$ and $\mathbf{C}$ are defined as:

$$\mathbf{A} = 2a \cdot \mathbf{r}_1 - a, \qquad \mathbf{C} = 2 \cdot \mathbf{r}_2 \tag{6}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors in the interval $[0,1]$, and the parameter a decreases linearly from 2 to 0 over the course of iterations to balance exploration and exploitation.

In the hunting phase, the three best agents denoted as $\mathbf{X}_\alpha$, $\mathbf{X}_\beta$, and $\mathbf{X}_\delta$ guide the search process. The distances to these leading agents are computed as:

$$\begin{aligned} \mathbf{D}_\alpha &= | \mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X} | \\ \mathbf{D}_\beta &= | \mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X} | \\ \mathbf{D}_\delta &= | \mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X} | \end{aligned} \tag{7}$$

The updated positions are then calculated as:

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{X}_\alpha - \mathbf{A}_1 \cdot \mathbf{D}_\alpha \\ \mathbf{X}_2 &= \mathbf{X}_\beta - \mathbf{A}_2 \cdot \mathbf{D}_\beta \\ \mathbf{X}_3 &= \mathbf{X}_\delta - \mathbf{A}_3 \cdot \mathbf{D}_\delta \\ \mathbf{X}(t+1) &= \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3} \end{aligned} \tag{8}$$

In this study, each search agent is evaluated using the objective function $F(\mathbf{x})$, and the best solution is determined by the minimum objective value. To provide a clearer description of the GWO procedure in solving nonlinear systems of equations, the main steps of the algorithm are summarized in Algorithm 1 [15].

Algorithm 1. Grey Wolf Optimizer (GWO)

```
Initialize population Xi (i = 1, 2, ..., N) within the search bounds
Initialize parameter a = 2
Evaluate fitness of each agent using F(x)
Identify Xα, Xβ, and Xδ as the best three agents
While (t < Max_iter)
    For each search agent Xi
        Generate random vectors r1 and r2
        Compute A and C:
            A = 2a * r1 - a
            C = 2 * r2
        Compute distances:
```

```
            Dα = |C1 * Xα − Xi|
            Dβ = |C2 * Xβ − Xi|
            Dδ = |C3 * Xδ − Xi|
        Compute candidate positions:
            X1 = Xα − A1 * Dα
            X2 = Xβ − A2 * Dβ
            X3 = Xδ − A3 * Dδ
        Update position:
            Xi = (X1 + X2 + X3) / 3
    End for
    Decrease parameter a linearly
    Evaluate fitness of all agents
    Update Xα, Xβ, and Xδ
    t = t + 1
End while
Return Xα as the best solution
```

3. Results

In this study, the Grey Wolf Optimizer (GWO) algorithm is applied to solve four case studies of nonlinear systems of equations. The implementation is carried out using MATLAB. The optimization process follows the problem formulation described in Section 2.1, where the nonlinear system is transformed into a minimization problem based on a modulus based objective function. Each candidate solution is evaluated by substituting the variables into the corresponding system of equations, including cases involving complex roots represented in the form $z = x + iy$. The objective function value reflects how close a candidate solution is to satisfying the system, with values approaching zero indicating better solutions.

3.1. Parameter Setting of GWO

The Grey Wolf Optimizer (GWO) algorithm is employed to determine solutions of nonlinear systems involving complex roots. Prior to the main simulation, a series of parameter tuning experiments is conducted to evaluate the influence of key parameters on the performance of the algorithm. This step aims to obtain a set of parameters that yields optimal, stable, and consistent results. The main parameters considered in this study include the number of search agents (population size), lower bound (lb), upper bound (ub), and maximum number of iterations ($maxiter$).

The number of search agents is varied across 10, 30, 50, 70, and 100 to evaluate its impact on solution quality, convergence speed, and algorithm stability. The initial solutions are randomly generated within the bounds $lb = -1$ and $ub = 1$, with the maximum number of iterations set to $maxiter = 1000$. The corresponding results are presented in Figure 1.

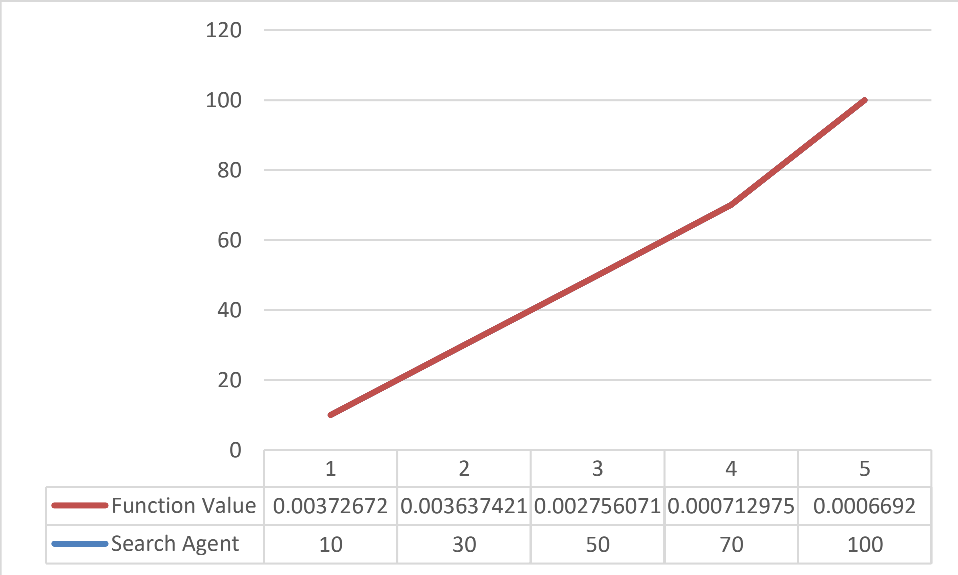

Fig. 1. Effect of the Number of Search Agents on Algorithm Performance

The experimental results indicate that increasing the number of search agents generally improves the quality of the obtained solutions. This improvement is attributed to the broader exploration capability of the algorithm, which increases the likelihood of approaching the global optimum. However, a larger population size also leads to higher computational cost. Among the tested values, a population size of 100 provides the best performance and is therefore selected for subsequent experiments.

Next, the search space boundaries are evaluated by testing several pairs of lower and upper bounds, namely $[-10,10]$, $[-30,30]$, $[-50,50]$, $[-70,70]$, and $[-100,100]$, while maintaining $search\ agent = 100$ and $maxiter = 1000$. The results are shown in Figure 2.

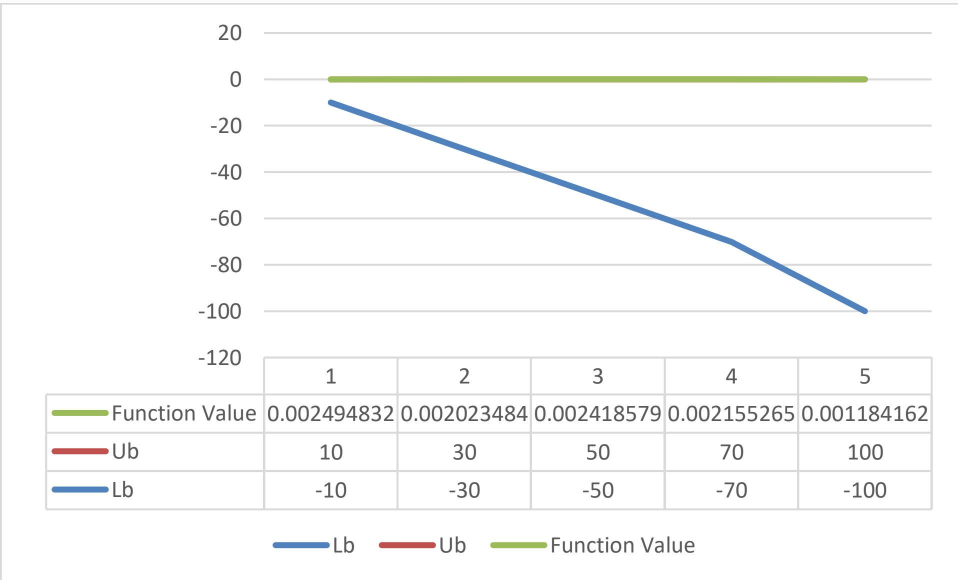

Fig. 2. Effect of Upper and Lower Bounds on Algorithm Performance

The findings show that the boundary range $[-100,100]$ produces the most optimal solution. This suggests that a wider search space increases the probability of locating better solutions. Therefore, this range is selected for further analysis.

Subsequently, the effect of the maximum number of iterations is examined. In this experiment, $search\ agent = 100$, $lb = -100$, and $ub = 100$, while the number of iterations is varied as 50, 100, 300, 500, 1000, 1500, and 2000. The results are illustrated in Figure 3.

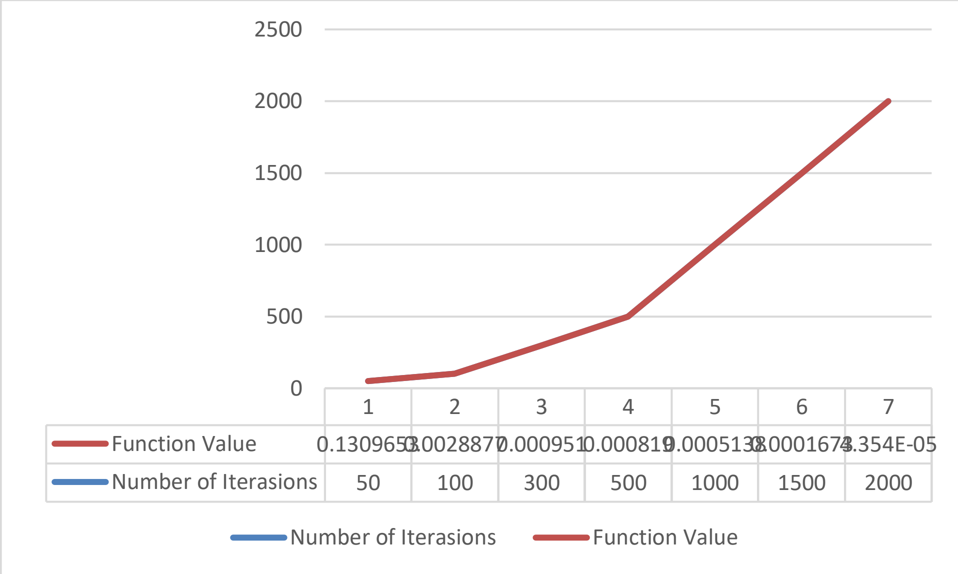

Fig. 3. Effect of Maximum Iterations on Algorithm Performance

The results indicate that increasing the number of iterations improves the convergence process. At early iterations, the objective function value decreases significantly, reflecting the exploration phase. As the number of iterations increases, the rate of improvement slows down, indicating a transition to the exploitation phase. Among the tested values, $maxiter = 2000$ yields the best performance and is selected for the final simulation.

3.2. Final Simulation

The parameter configuration used in the final simulation is determined based on the results of the parameter tuning phase. The selected parameters are $search\ agent = 100$, $lb = -100$, $ub = 100$, and $maxiter = 2000$. These parameters are then used to perform the main simulation using the GWO algorithm. The results obtained from GWO are compared with those of the Particle Swarm Optimization (PSO) algorithm reported by Kamsyakawuni [10], as presented in Table 1. This comparison aims to evaluate the relative performance of GWO in terms of solution accuracy and convergence characteristics.

Table 1. Performance Comparison of GWO and PSO Algorithms

No	Nonlinear Systems of Equations	Solution		Function Value		Convergence Iteration	
		GWO	PSO	GWO	PSO	GWO	PSO
1	$e^x + xy - y - 0{,}5 = 0$ $\sin(xy) + x + y - 1 = 0$	$x = 0{,}139578 - 0{,}000572i$ $y = 0{,}755205 + 0{,}000004i$	$x = 0{,}139581 - 0{,}000000i$ $y = 0{,}755203 + 0{,}000000i$	$2{,}091071399510 \times 10^{-3}$	$4{,}643383591126 \times 10^{-17}$	992	874
2	$x^2 - 10x - y^2 + 8 = 0$ $xy^2 + x - 10y - 8 = 0$	$x = 0{,}821223 - 0{,}001225i$ $y = -0{,}679895 - 0{,}000383i$	$x = 0{,}821210 + 0{,}000000i$ $y = -0{,}679916 - 0{,}000000i$	$1{,}426847960281 \times 10^{-2}$	$7{,}222149558420 \times 10^{-17}$	1000	892
3	$x + \cos(xy) - z^2 - 1{,}1 = 0$ $x^2 - 10y - e^{xy} + 0{,}8 = 0$ $xz + y^2 - z - 0{,}3 = 0$	$x = 0{,}267614 - 0{,}002028i$	$x = 0{,}267600 + 0{,}000044i$	$2{,}124307260116 \times 10^{-3}$	$2{,}441561519138 \times 10^{-04}$	1000	992

		$y = -0{,}012505 + 0{,}000192i$ $z = -0{,}409395 - 0{,}001034i$	$y = -0{,}012505 - 0{,}000002i$ $z = -0{,}409383 - 0{,}000053i$				
4	$2x^2 + y - z^2 - 10 = 0$ $3x^2 + 6y - z^2 - 25 = 0$ $x^2 - 5y + 6z^2 - 4 = 0$	$x = 2{,}180503 + 0{,}038347i$ $y = 2{,}053711 - 0{,}104384i$ $z = 1{,}260151 - 0{,}045574i$	$x = 2{,}182854 - 0{,}000712i$ $y = 2{,}047349 + 0{,}001906i$ $z = -1{,}256443 - 0{,}000839i$	$3{,}565727520882 \times 10^{-1}$	$6{,}617172638488 \times 10^{-03}$	999	1000

Based on the comparison results, the PSO algorithm demonstrated a higher level of accuracy and faster convergence than GWO. This could be due to PSO's speed based update mechanism, which allows for faster exploitation. On the other hand, the GWO algorithm demonstrated stable performance in finding solutions close to the root of the system. The position update mechanism involving the three best agents (α, β, δ) provides a balance between exploration and exploitation, allowing GWO to still produce competitive solutions, although it does not always excel in convergence speed.

4. Conclusion

Based on the research results, the Grey Wolf Optimizer (GWO) algorithm can be used to solve nonlinear systems of equations containing both real and complex roots through an optimization approach. Test results show that GWO is capable of producing stable solutions with function values close to zero in various case studies. Parameter analysis indicates that increasing the number of search agents and iterations has a positive impact on solution quality, although this does increase computational time. Based on comparisons with the PSO algorithm, PSO has advantages in terms of accuracy and convergence speed. However, GWO still demonstrates competitive performance with more stable exploration characteristics, making it potentially useful as an alternative method for solving complex nonlinear systems of equations.

References

[1] Munir, R. (2006). *Metode numerik*. Jakarta, Indonesia: Erlangga.

[2] Munir, R. (2021). *Metode numerik: Revisi kelima*. Bandung, Indonesia: Informatika.

[3] Nanang, H. (2025). *Buku ajar metode numerik*. Widina.

[4] Arief, M. H., Saputra, M. A. W., Afandi, K., Novtahaning, D., & Ranggianto, N. A. (2025). Optimalisasi klasifikasi performa akademik mahasiswa dengan pendekatan metaheuristik berbasis educational data mining. *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer, 9*(3), 943–951. DOI: https://doi.org/10.33395/remik.v9i3.15168.

[5] Madi, M., Marković, D., & Radovanović, M. (2013). Comparison of meta-heuristic algorithms for solving machining optimization problem. *Mechanical Engineering, 11*(1), 29–44.

[6] Robandi, I. (2021). *Artificial intelligence: Mengupas rekayasa kecerdasan tiruan*. Andi.

[7] Pangestu, M. H. R. (2021). *Rekonfigurasi jaringan distribusi untuk meminimalisasi rugi-rugi daya dengan menggunakan metode Grey Wolf Optimizer (GWO)*.

[8] Sulistiawati, M., Aqillah, A. R., Isnaini, A. N., Novitasari, E. K., & Amarakamini, L. (2025). Analisis komparatif kinerja algoritma klasik, metaheuristik, dan hibrida untuk optimisasi portofolio saham berbasis clustering. *Jurnal Matematika, Komputasi, dan Statistika, 5*(3), 1133–1146.

[9] Ariyaratne, M. K. A., Fernando, T. G. I., & Weerakoon, S. (2015). A modified firefly algorithm to solve univariate nonlinear equations with complex roots. In *International Conference on Advances in ICT for Emerging Regions* (pp. 160–167). DOI: https://doi.org/10.1109/ICTER.2015.7377683.

[10] Kamsyakawuni, A., Sari, M. P., Riski, A., & Santoso, K. A. (2020). Metaheuristic algorithm approach to solve non-linear equations system with complex roots. *Journal of Physics: Conference Series*. DOI: http://dx.doi.org/10.1088/1742-6596/1494/1/012001.

[11] Sasongko, S. B. (2010). *Metode numerik dengan Scilab*. Andi.

[12] Enny, Y. A. (2021). *Variabel kompleks*.

[13] Ghoni, U. (2022). *Swarm intelligence: Perbaikan algoritma Grey Wolf Optimizer berdasarkan paper "Inspired Grey Wolf Optimizer for solving large-scale function optimization problems" dilengkapi perhitungan manual Grey Wolf Optimizer dan Inspired Grey Wolf Optimizer*. GAMADEV.

[14] Muro, C., Escobedo, R., Spector, L., & Coppinger, R. (2011). Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behavioural Processes, 88*(3), 192–197. DOI: https://doi.org/10.1016/j.beproc.2011.09.006.

[15] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software, 69*, 46–61. DOI: https://doi.org/10.1016/j.advengsoft.2013.12.007.