

## OPTIMASI PENJADWALAN TUGAS CLOUD MENGGUNAKAN ALGORITMA HYBRID GA-PSO: ANALISIS MAKESPAN BERBASIS CLOUDSIM

Puguh Setiyono<sup>1\*</sup>, Bagus Gede Krishna Yudistira,<sup>2</sup>

Jurusan Teknik Informatika, Universitas Pendidikan Ganesha, Jl. Udayana No.11, Banjar Tegal, Singaraja, Kabupaten Buleleng, Bali. 81116

\*Email Correspondence: [puguhsetiyono765@gmail.com](mailto:puguhsetiyono765@gmail.com)

Received: 01-05-2026 | Revised: 20-05-2026 | Accepted: 01-06-2026 | Published: 24-06-2026

### Abstrak

Penjadwalan tugas merupakan komponen penting dalam cloud computing untuk memastikan alokasi sumber daya dan waktu eksekusi yang optimal. Penelitian ini mengusulkan algoritma hybrid yang menggabungkan Genetic Algorithm (GA) dan Particle Swarm Optimization (PSO) untuk meminimalkan makespan dalam penjadwalan tugas cloud. Algoritma ini memanfaatkan kemampuan eksplorasi global dari GA dan kekuatan eksploitasi lokal dari PSO. Simulasi dilakukan menggunakan CloudSim 3.0.3 dengan tiga skenario workload yang berbeda, yaitu ringan, sedang, dan berat. Setiap konfigurasi memiliki variasi jumlah cloudlet, virtual machine, kapasitas CPU, panjang tugas, dan alokasi memori. Hasil penelitian menunjukkan bahwa algoritma hybrid GA-PSO secara konsisten mengungguli GA dan PSO tunggal dalam meminimalkan makespan, terutama pada beban kerja berat. Pengurangan makespan tercatat sebesar 11,0% dibandingkan GA dan 22,3% dibandingkan PSO. Selain itu, pendekatan hybrid menunjukkan kestabilan dan adaptabilitas yang lebih tinggi dalam lingkungan dengan keterbatasan sumber daya. Temuan ini mendukung penerapan praktis algoritma hybrid GA-PSO pada sistem cloud yang dinamis dan membuka peluang untuk pengembangan lanjutan pada optimasi multi-objektif serta implementasi di lingkungan nyata.

**Keywords:** Cloud Computing, Task Scheduling, Genetic Algorithm, Particle Swarm Optimization, CloudSim, Makespan

### Abstract

*Task scheduling is a critical component in cloud computing to ensure optimal resource allocation and execution time. This study proposes a hybrid algorithm that combines Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to minimize makespan in cloud task scheduling. The algorithm leverages the global exploration capability of GA and the local exploitation strength of PSO. The simulation was conducted using CloudSim 3.0.3 with three different workload scenarios: light, medium, and heavy. Each configuration varied in terms of cloudlet count, virtual machines, CPU capacity, task length, and memory allocation. Results indicate that the hybrid GA-PSO algorithm consistently outperformed standalone GA and PSO in minimizing makespan, particularly under heavy workloads. It achieved an 11.0% reduction compared to GA and 22.3% compared to PSO. Moreover, the hybrid approach demonstrated greater stability and adaptability in constrained resource environments. These findings support the practical use of the hybrid GA-PSO algorithm in dynamic cloud systems and highlight its potential for future development in multi-objective optimization and real-world deployments.*

**Kata kunci:** Cloud Computing, Task Scheduling, Genetic Algorithm, Particle Swarm Optimization, CloudSim, Makespan

## PENDAHULUAN

Perkembangan teknologi digital pada era Revolusi Industri 4.0 telah mengubah cara manusia bekerja, belajar, dan berinteraksi. Teknologi seperti *Internet of Things* (IoT), *Artificial Intelligence* (AI), dan *Big Data Analytics* menjadi fondasi utama dalam mendukung transformasi digital di berbagai sektor, mulai dari industri manufaktur hingga layanan kesehatan [1]. Lonjakan eksponensial dalam produksi data yang dihasilkan oleh perangkat IoT, aplikasi *mobile*, dan layanan berbasis *web* telah menciptakan tantangan baru dalam hal penyimpanan, pemrosesan, dan analisis data secara *real time*.

*Cloud computing* hadir sebagai paradigma komputasi modern yang menawarkan solusi untuk mengatasi tantangan tersebut dengan mengintegrasikan asset komputasi ke dalam suatu wadah yang kolektif guna mengoptimalkan waktu penyelesaian sebuah tugas [12][13]. Melalui penyediaan layanan berbasis jaringan seperti komputasi, penyimpanan, dan aplikasi, layanan *cloud* dapat memungkinkan organisasi untuk mengakses sumber daya komputasi sesuai permintaan (*on-demand*) dengan skala elastis atau fleksibel tanpa perlu investasi infrastruktur besar [2]. Model layanan seperti *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS), dan *Software as a Service* (SaaS) telah banyak diadopsi untuk mendukung skenario bisnis yang membutuhkan fleksibilitas tinggi dan efisiensi biaya [3]. Salah satu keunggulan *cloud* adalah kemampuannya dalam melakukan *provisioning* (menyediakan dan mengelola sumber daya) otomatis dan alokasi sumber daya dinamis, yang menjadi dasar penting dalam manajemen beban kerja [15].

Di dalam arsitektur *cloud*, penjadwalan tugas (*task scheduling*) menjadi proses kritis untuk memastikan bahwa tugas-tugas yang diajukan pengguna (*cloudlets*) dipetakan secara optimal ke sebuah *Virtual Machines* (VM) guna mencapai waktu eksekusi ideal dan pemanfaatan sumber daya yang efisien [10]. Penjadwalan yang buruk dapat menyebabkan meningkatnya *makespan* (yaitu waktu total penyelesaian semua tugas), ketidakseimbangan beban (*load imbalance*), hingga pelanggaran *Service Level Agreement* (SLA) yang dapat menyebabkan menurunkan kepuasan pengguna [4][11]. Algoritma konvensional seperti *First-Come-First-Serve* (FCFS) serta Algoritma *Round Robin* (RR) telah digunakan karena kesederhanaannya. Namun, algoritma tersebut tidak cukup efektif dalam lingkungan *cloud* yang dinamis, heterogen, dan berskala besar [5]. Kompleksitas lingkungan *cloud modern* memerlukan algoritma yang adaptif terhadap perubahan kondisi sistem. Pendekatan berbasis metaheuristik mulai digunakan sebagai alternatif untuk mengatasi keterbatasan algoritma konvensional. *Genetic Algorithm* (GA) menawarkan eksplorasi ruang solusi yang luas melalui mekanisme evolusi populasi, sedangkan *Particle Swarm Optimization* (PSO) memanfaatkan perilaku kolektif partikel untuk mempercepat konvergensi menuju solusi optimal [6]. Kombinasi kedua metode ini dalam bentuk hybrid (GA-PSO) telah menunjukkan potensi yang signifikan dalam meningkatkan efisiensi penjadwalan di lingkungan *cloud* [7]. CloudSim sebagai framework simulasi banyak dimanfaatkan untuk menguji algoritma-algoritma ini dikarenakan kemampuannya memodelkan lingkungan *cloud* secara realistis, termasuk heterogenitas VM dan variasi beban kerja [8][14].

Namun, sebagian besar penelitian sebelumnya masih memiliki keterbatasan. Beberapa studi hanya menitikberatkan pada optimasi *makespan* tanpa mempertimbangkan keseimbangan beban antar-VM secara menyeluruh [9]. Di sisi lain, kajian seperti yang dilakukan oleh Abdulghani dan Singh *et al.* telah menerapkan pendekatan *hybrid* seperti GA-PSO, namun belum mengeksplorasi secara mendalam pengaruh heterogenitas VM serta efektivitasnya dalam kondisi sumber daya terbatas [16][17]. Selain itu, eksperimen-eksperimen sebelumnya umumnya dilakukan dalam konfigurasi sistem berskala besar dengan infrastruktur tinggi, sehingga masih sedikit penelitian yang mengevaluasi performa algoritma dalam lingkungan simulasi dengan keterbatasan RAM dan CPU, serta variasi beban kerja yang ekstrim.

Penelitian ini bertujuan untuk menjembatani kesenjangan tersebut dengan merancang dan mengimplementasikan algoritma *hybrid* yaitu *hybrid Genetic Algorithm – Particle Swarm Optimization* (GA-PSO) yang disertai dengan mekanisme mutasi adaptif pada pBest dan gBest, guna meningkatkan kemampuan eksplorasi dan eksploitasi solusi. Evaluasi dilakukan dalam tiga skenario workload berbeda (ringan, sedang, dan berat) menggunakan lingkungan simulasi berbasis CloudSim dengan keterbatasan sumber daya (RAM 3 GB, prosesor dual-core), yang merepresentasikan kondisi realistik dari pengguna *cloud* skala kecil hingga menengah.

Dengan pendekatan tersebut, diharapkan kontribusi penelitian ini dapat memperluas cakupan studi sebelumnya dan memberikan alternatif solusi penjadwalan yang lebih efisien pada arsitektur *cloud* modern yang bersifat heterogen dan dinamis.

## METODE

Penelitian ini menggunakan metode kuantitatif dengan pendekatan eksperimen berbasis simulasi untuk mengevaluasi kinerja algoritma *hybrid Genetic Algorithm–Particle Swarm Optimization* (GA-PSO) dalam penjadwalan tugas pada lingkungan *cloud computing*. Data yang dianalisis berupa nilai numerik, yaitu *makespan* dan waktu eksekusi, yang diperoleh melalui implementasi algoritma pada simulator *CloudSim*. Pengujian dilakukan dengan membandingkan algoritma *hybrid* GA-PSO terhadap algoritma GA dan PSO murni pada tiga skenario *workload* yang berbeda guna mengukur kemampuan algoritma dalam meminimalkan *makespan* dan menjaga distribusi beban kerja secara optimal (Calheiros et al., 2011; Kennedy & Eberhart, 1995; Holland, 1992).

Tahapan penelitian disusun secara sistematis dan *reproducible*, dimulai dari analisis kebutuhan untuk mengidentifikasi permasalahan penjadwalan tugas serta merancang solusi berbasis algoritma *hybrid* GA-PSO. Lingkungan simulasi dikembangkan menggunakan JDK 8, Eclipse IDE, dan *CloudSim* 3.0.3 dengan konfigurasi *data center*, *virtual machine* (VM), dan *cloudlet* yang terintegrasi dengan mekanisme penjadwalan GA-PSO. Algoritma ini menggabungkan kemampuan eksplorasi dari GA dan eksploitasi dari PSO untuk meningkatkan efektivitas pencarian solusi penjadwalan pada lingkungan *cloud* (Calheiros et al., 2011; Holland, 1992; Kennedy & Eberhart, 1995).

Evaluasi dilakukan melalui tiga skenario *workload* yaitu ringan, sedang, dan berat yang dibedakan berdasarkan jumlah *cloudlet*, VM, kapasitas CPU, panjang tugas, dan memori. Setelah implementasi pada *CloudSim*, data berupa nilai *makespan* dan waktu eksekusi dikumpulkan serta dianalisis dengan melakukan tiga kali replikasi pada setiap skenario untuk menjamin konsistensi hasil. Validasi penelitian mencakup validasi internal melalui pengulangan simulasi dan validasi eksternal dengan membandingkan hasil algoritma GA-PSO terhadap metode *First Come First Served* (FCFS) dan hasil penelitian terdahulu guna memastikan keandalan serta akurasi kinerja algoritma yang diusulkan.

## HASIL DAN PEMBAHASAN

Hasil penelitian menunjukkan bahwa algoritma *hybrid* GA-PSO secara konsisten menghasilkan nilai *makespan* yang lebih rendah dibandingkan dengan algoritma GA dan PSO tunggal pada semua skenario pengujian. Penurunan *makespan* ini menunjukkan bahwa strategi kombinasi antara eksplorasi dari GA dan eksploitasi dari PSO dapat meningkatkan efisiensi penjadwalan tugas *cloud*.

## Hasil Implementasi

Implementasi algoritma *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), dan *Hybrid GA-PSO* dilakukan pada simulator CloudSim versi 3.0.3 menggunakan IDE Eclipse dengan JDK 8. Lingkungan simulasi dikonfigurasi berdasarkan tiga skenario *workload*, yaitu ringan, sedang, dan berat. Setiap skenario memiliki variasi pada jumlah *cloudlet*, jumlah *Virtual Machine*, panjang tugas (*Million Instructions*), kapasitas CPU (MIPS), serta Kapasitas Memori (RAM). Hasil implementasi menunjukkan bahwa simulator berhasil menjalankan ketiga algoritma pada semua skenario *workload* tanpa terjadi *error*, meskipun dijalankan pada konfigurasi perangkat keras terbatas (RAM 3 GB, CPU dual-core). Setiap skenario pengujian diulang sebanyak tiga kali untuk memastikan konsistensi hasil, dan nilai rata-rata *makespan* dari setiap pengujian digunakan sebagai dasar untuk analisis performa algoritma.

## Hasil Pengujian Skenario

Hasil pengujian menunjukkan adanya perbedaan signifikan pada performa ketiga algoritma. Algoritma *Hybrid GA-PSO* secara konsisten menghasilkan nilai *makespan* yang lebih rendah pada pengujian skenario berat dibandingkan dengan GA dan PSO murni pada skenario *workload*.

### Skenario Ringan

Pada skenario ringan, pengujian dilakukan dengan konfigurasi 20 *cloudlet* dan 3 *virtual machine*. Tujuan pengujian ini adalah untuk mengevaluasi performa algoritma GA, PSO, dan *Hybrid GA-PSO* dalam kondisi *workload* yang relatif kecil. Setiap algoritma diuji sebanyak tiga kali iterasi, dan hasil yang diperoleh berupa nilai *makespan* (waktu penyelesaian total). Hasil pengujian skenario ringan ditampilkan pada Tabel 1.

**Tabel 1. Hasil Pengujian Skenario Ringan**

No	Algoritma	Iterasi 1	Iterasi 2	Iterasi 3	Rata-rata Makespan
1	Genetic Algorithm (GA)	6.22	6.51	7.19	6.64
2	Particle Swarm Optimization (PSO)	7.58	8.79	9.99	8.79
3	Hybrid GA-PSO	6.86	6.78	7.25	6.96

Sumber : Diolah oleh peneliti

### Skenario Sedang

Pada skenario sedang, yaitu pengujian dilakukan dengan 50 *cloudlet* dan 5 *Virtual Machine*. Pada Tabel 2, Algoritma *Hybrid GA-PSO* memberikan hasil terbaik dalam mengoptimalkan *makespan* dibandingkan kedua algoritma lainnya pada skenario ini.

**Tabel 2. Hasil Pengujian Skenario Sedang**

No	Algoritma	Iterasi 1	Iterasi 2	Iterasi 3	Rata-rata Makespan
1	Genetic Algorithm (GA)	10.82	11.32	11.46	11,02
2	Particle Swarm Optimization (PSO)	9.56	10.72	13.42	11.23
3	Hybrid GA-PSO	10.56	10.99	12.31	11,28

Sumber : Diolah oleh peneliti

### Skenario Berat

Pada skenario berat, pengujian dilakukan dengan 100 *cloudlet* dan 10 *Virtual Machine*. Algoritma *Hybrid GA-PSO* pada skenario berat, memberikan kinerja terbaik dengan pengurangan *makespan* signifikan, menunjukkan efisiensi pada beban kerja yang tinggi. Hasil dari pengujian skenario berat ditampilkan pada Tabel 3.

Tabel 3. Hasil Pengujian Skenario Berat

No	Algoritma	Iterasi 1	Iterasi 2	Iterasi 3	Rata-rata Makespan
1	Genetic Algorithm (GA)	15.91	16.43	17.08	16.47
2	Particle Swarm Optimization (PSO)	18.66	18.98	19.01	18.88
3	Hybrid GA-PSO	14.25	14.63	15.11	14.66

Sumber : Diolah oleh peneliti

### Analisis Perbandingan Algoritma

Berdasarkan hasil pengujian pada ketiga skenario *workload* (ringan, sedang, berat). Kinerja algoritma *Hybrid GA-PSO*, *Genetic Algorithm (GA)*, dan *Particle Swarm Optimization (PSO)* dianalisis secara komparatif melalui parameter *makespan*.

Gambar 1. Grafik Perbandingan Makespan



Pada Gambar 1, Analisis komparatif terhadap kinerja algoritma *Hybrid GA-PSO*, *Genetic Algorithm (GA)*, dan *Particle Swarm Optimization (PSO)* mengungkap pola kinerja yang signifikan. Pada skenario ringan (20 *cloudlet*, 3 *Virtual Machine*), GA mencapai *makespan* terendah (6.64), mengungguli *Hybrid GA-PSO* (6.96) dan PSO (8.79). Hal ini menunjukkan keefektifan mekanisme eksplorasi GA dalam ruang solusi terbatas, sementara PSO menunjukkan keterbatasan adaptasi akibat kebutuhan iterasi tinggi untuk konvergensi *swarm*.

Dalam skenario sedang (50 *cloudlet*, 5 *Virtual Machine*), GA mempertahankan kinerja superior dengan *makespan* 11.02, diikuti PSO (11.23) dan *Hybrid GA-PSO* (11.28). PSO mengalami peningkatan kinerja berkat efektivitas *social learning* pada kompleksitas moderat, sedangkan *Hybrid GA-PSO* mulai menunjukkan stabilitas melalui sinergi mekanisme eksplorasi GA dan eksploitasi berbasis *velocity* (kecepatan menuju posisi optimal) PSO. Perbedaan ini tergolong kecil dan berada dalam rentang toleransi yang wajar, yakni kurang dari 2.5%. Algoritma GA mampu mengeksplorasi ruang solusi secara efisien tanpa *overhead* tambahan, sementara algoritma *hybrid GA-PSO* membutuhkan waktu dan iterasi yang lebih banyak untuk menyeimbangkan proses eksplorasi dan eksploitasi secara optimal.

Selain itu, adanya variasi kecil dalam hasil juga dapat dipengaruhi oleh faktor randomisasi awal pada proses inialisasi populasi, terutama karena algoritma dijalankan dalam jumlah iterasi terbatas. Meskipun algoritma *hybrid GA-PSO* dirancang untuk menghindari jebakan lokal dan mempercepat konvergensi, pada skala menengah kompleksitasnya justru belum memberi dampak signifikan.

Pola berbalik signifikan teramati pada skenario berat (100 *cloudlet*, 10 VM). *Hybrid GA-PSO* mendominasi dengan *makespan* 14.66 dan mengurangi 11.0% dari GA (16.47) serta 22.3% dari PSO (18.88).

Keunggulan pada algoritma *hybrid* ini disebabkan kemampuan ganda yaitu menghindari *local optimum* melalui operator mutasi GA, dan mempercepat konvergensi dengan memanfaatkan mekanisme *tracking* PSO. Sinergi ini menjadi kritis dalam lingkungan dinamis berskala besar.

Tren kinerja ketiga algoritma semakin jelas ketika beban kerja ditingkatkan. *Genetic Algorithm* (GA) menunjukkan peningkatan waktu penyelesaian (*makespan*) yang sangat signifikan dari 6.64 menjadi 16.47 akibat proses komputasi yang berat. *Particle Swarm Optimization* (PSO) memiliki kinerja tidak stabil dengan hasil buruk pada beban ringan (8.79) maupun berat (18.88), terutama karena sensitivitas terhadap kondisi awal serta kecenderungan terjebak di solusi lokal. Sebaliknya, *Hybrid GA-PSO* mencatat pola stabil dengan *makespan* yang meningkat secara bertahap (6.96 → 11.28 → 14.66). Temuan ini sejalan dengan penelitian yang dilakukan oleh *Sigh et al* dan Abdulghani yang juga membuktikan konsistensi *Hybrid GA-PSO* di lingkungan *cloud* heterogen, termasuk pada simulasi dengan sumber daya terbatas [16][17].

Untuk mengevaluasi signifikansi perbedaan kinerja antar algoritma, dilakukan uji statistik ANOVA satu arah terhadap nilai *makespan* yang dihasilkan oleh masing-masing algoritma (GA, PSO, dan *Hybrid GA-PSO*).

**Tabel 4. Uji Anova Makespan**

Cases	Sum of Squares	df	Mean Square	F	p	$\eta^2$
Algoritma	6.696	2	3.348	0.150	0.864	0.048
Residuals	133.726	6	22.288			

Note: Type III Sum of Squares

Sumber : Diolah oleh peneliti

Hasil pengujian menunjukkan bahwa nilai F sebesar 0.150 dengan signifikansi p sebesar 0.864. Karena nilai p jauh lebih besar dari batas signifikansi 0.05, maka dapat disimpulkan bahwa tidak terdapat perbedaan yang signifikan secara statistik antara ketiga algoritma dalam menghasilkan rata-rata *makespan* pada skenario tersebut. Hal ini diperkuat dengan nilai eta-squared ( $\eta^2$ ) sebesar 0.048, yang menunjukkan bahwa hanya sekitar 4.8% variasi *makespan* dapat dijelaskan oleh jenis algoritma yang digunakan, sementara sisanya yaitu disebabkan oleh faktor lain atau variasi acak. Ketiadaan perbedaan signifikan ini menunjukkan bahwa ketiga algoritma memiliki kinerja yang relatif seimbang dalam menangani beban kerja dengan kompleksitas sedang. Kondisi ini dimungkinkan karena pada skenario moderat, ruang solusi belum cukup kompleks menunjukkan keunggulan eksplorasi dan eksploitasi yang lebih kuat dari algoritma *hybrid GA-PSO*. Selain itu, variasi acak pada populasi awal serta sensitivitas parameter *metaheuristik* juga turut memengaruhi hasil. Meskipun secara numerik GA memiliki rata-rata *makespan* terendah pada skenario ini, perbedaannya tidak cukup signifikan untuk disimpulkan sebagai keunggulan secara statistik. Di sisi lain, pada skenario beban berat, selisih kinerja algoritma menjadi lebih mencolok secara numerik, yang menandakan bahwa efektivitas algoritma *hybrid GA-PSO* mulai terlihat lebih dominan pada kompleksitas yang lebih tinggi.

## KESIMPULAN

Berdasarkan hasil eksperimen dan analisis terhadap algoritma *Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO), serta *hybrid GA-PSO* pada tiga skenario dengan beberapa beban tugas atau *workload* (ringan, sedang, berat), diperoleh kesimpulan sebagai berikut:

- Algoritma *Hybrid GA-PSO* menunjukkan performa paling konsisten dan unggul pada beban kerja berat. Pada skenario berat (100 *cloudlet*, 10 VM), algoritma *hybrid GA-PSO* berhasil menurunkan nilai

*makespan* secara signifikan sebesar 11,0% jika dibandingkan algoritma GA dan 22,3% dibandingkan algoritma PSO. Efektivitas algoritma *hybrid* GA-PSO sangat terlihat ketika kompleksitas tinggi, berkat sinergi eksplorasi dari algoritma GA dan eksploitasi dari algoritma PSO. Hal ini menjadikan algoritma *hybrid* GA-PSO cocok diimplementasikan di lingkungan *cloud* dinamis berskala besar atau dengan beban komputasi tinggi.

- b. Kinerja algoritma *Hybrid* GA-PSO tetap stabil pada skenario sedang, meskipun algoritma GA sedikit lebih baik secara numerik. Pada beban sedang, ruang solusi belum cukup kompleks untuk mengekspresikan keunggulan eksplorasi dari PSO atau sinergi *hybrid* GA-PSO. Namun, kestabilan hasil tetap menunjukkan bahwa pendekatan *hybrid* dapat beradaptasi meski tidak selalu unggul.
- c. Algoritma GA unggul pada skenario ringan karena *overhead hybrid* belum memberikan dampak positif. Pada beban ringan (20 *cloudlet*, 3 VM), GA mencatat *makespan* terendah (6.64), mengungguli *hybrid* GA-PSO (6.96) dan PSO (8.79). Dalam skenario dengan ruang solusi terbatas, eksplorasi GA cukup efektif dan efisien tanpa memerlukan kombinasi algoritma. *Hybrid* GA-PSO masih memerlukan *overhead* iterasi dan perhitungan *velocity* yang tidak sebanding manfaatnya pada *workload* ringan.
- d. *Hybrid* GA-PSO memiliki tren kinerja paling stabil terhadap peningkatan *workload*. Nilai *makespan* algoritma *hybrid* meningkat secara bertahap (6.96 → 11.28 → 14.66) seiring meningkatnya *workload*. Sebaliknya, GA dan PSO menunjukkan fluktuasi dan ketidakstabilan kinerja, terutama PSO yang sangat sensitif terhadap kondisi awal. Stabilitas ini menjadi keunggulan penting dalam sistem *cloud* yang heterogen dan rentan terhadap variasi beban yang mendadak.
- e. Untuk penelitian selanjutnya, disarankan mengarah pada pengembangan *multi-objektif* (seperti *energy consumption*, *load balancing*), serta pengujian pada lingkungan *cloud* fisik atau simulasi berbasis *container* (misalnya Docker atau Kubernetes).

## DAFTAR PUSTAKA

- [1] N. Rana *et al.*, “A Systematic Literature Review on Contemporary and Future Trends in Virtual Machine Scheduling Techniques in Cloud and Multi-Access Computing,” *Research Square*, preprint, Apr. 2024. doi: 10.3389/fcomp.2024.1288552
- [2] P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, NIST SP 800-145, Sep. 2011, doi: 10.6028/NIST.SP.800-145
- [3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, Jun. 2009, doi: 10.1016/j.future.2008.12.001.
- [4] A. Beloglazov and R. Buyya, “Energy efficient allocation of virtual machines in cloud data centers,” *Proc. IEEE/ACM CCGrid*, 2010, pp.577–578, doi: 10.1109/CCGRID.2010.45.
- [5] Y. C. Lee and A. Y. Zomaya, “Energy efficient utilization of resources in cloud computing systems,” *J. Supercomput.*, vol. 60, pp. 268–280, 2012, doi: 10.1007/s11227-010-0421-3.
- [6] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. IEEE ICNN*, vol. 4, pp. 1942–1948, 1995, doi: 10.1109/ICNN.1995.488968.
- [7] A.M. Manasrah and H. B. Ali, “Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing,” *Wireless Commun. Mobile Comput.*, vol. 2018, art. no. 1934784, 2018, doi: 10.1155/2018/1934784.
- [8] R. N. Calheiros *et al.*, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Softw. Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011, doi: 10.1002/spe.995.

- [9] Z. Wu and J. Xiong, "A Novel Task-Scheduling Algorithm of Cloud Computing Based on Particle Swarm Optimization," *International Journal of Gaming and Computer-Mediated Simulations*, vol. 13, no. 2, pp. 1–15, 2021, doi: 10.4018/IJGCMS.2021040101.
- [10] S. K. Patel and A. Singh, "Task Scheduling in Cloud Computing Using Hybrid Meta-heuristic: A Review," *arXiv preprint arXiv:2201.09242*, 2022, doi: [https://doi.org/10.1007/978-981-15-7533-4\\_35](https://doi.org/10.1007/978-981-15-7533-4_35)
- [11] Y. Home Prasanna Raju and N. Devarakonda, "Makespan Efficient Task Scheduling in Cloud Computing," in *Emerging Technologies in Data Mining and Information Security*, Springer, 2019, pp. 283–298. doi:[10.1007/978-981-13-1951-8\\_26](https://doi.org/10.1007/978-981-13-1951-8_26)
- [12] H. Fathi, Musawarman, M. Nugraha, I. Salsabilla, A. Fauzia, "Perancangan Cloud Storage pada Laboratorium Komputer Teknologi Rekayasa Perangkat Lunak dengan Menggunakan FreeNAS," *JITET*, vol. 12, no. 3, pp. 1806–1817, Aug. 2024, doi: 10.23960/jitet.v12i3.4446.
- [13] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010, doi: 10.1007/s13174-010-0007-6.
- [14] S. Potluri *et al.*, "Enhanced Task Scheduling Using Optimized Particle Swarm Optimization Algorithm in Cloud Computing Environment," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 11, no. 3, 2023, doi:10.4108/eetsis.4042.
- [15] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012, doi: 10.1016/j.future.2011.04.017.
- [16] P. Singh, M. Dutta, and N. Aggarwal, "A Review of Task Scheduling Based on Meta-Heuristics Approach in Cloud Computing," *Knowledge and Information Systems*, vol. 52, pp. 1–35, 2017, doi: 10.1007/s10115-017-1044-2.
- [17] A. M. Abdulghani, "Hybrid Task Scheduling Algorithm for Makespan Optimisation in Cloud Computing: A Performance Evaluation," *Journal of Artificial Intelligence*, vol. 6, no. 1, pp. 241–259, 2024. doi: 10.32604/jai.2024.056259.
- [18] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.