Print ISSN: 2461-1409
Journal Sensi Online ISSN: 2655-5298

Research Article Open Access (CC–BY-SA)

Optimalisasi Penjadwalan Tenaga Kerja Menggunakan Algoritma Ant Colony Optimization: Studi Kasus PT. Cloud Hosting Indonesia

Gagas Nurfilael*[1], Agung Mulyo Widodo[2], Nizirwan Anwar[3], Arief Ichwani[4]
[1,2,3,4] Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul Jakarta
E-mail: *[1] gagasnrvilael12@student.esaunggul.ac.id , [2]agung.mulyo@esaunggul.ac.id,
[3]nizirwan.anwar@esaunggul.ac.id, [4]arief.ichwani@esaunggul.ac.id

Abstrak

Penjadwalan tenaga kerja yang efektif sangat penting untuk meningkatkan produktivitas menjaga kualitas layanan di PT. Cloud Hosting Indonesia. Penjadwalan tenaga kerja adalah proses pengaturan dan alokasi tenaga kerja untuk berbagai tugas dan tanggung jawab dalam suatu organisasi. Ant Colony Optimization adalah sebuah teknik probabilistik yang digunakan untuk memecahkan masalah komputasi dengan menemukan jalur terbaik melalui graf. Sesuai dengan namanya, algoritma ini terinspirasi oleh perilaku semut, khususnya cara semut menemukan makanan. Namun, metode penjadwalan manual yang ada saat ini menimbulkan berbagai masalah, seperti kinerja yang tidak optimal, ketidakteraturan operasional, waktu istirahat yang tidak memadai, dan jadwal libur yang tidak pasti. Untuk mengatasi masalah ini, Algoritma Ant Colony Optimization diterapkan. Terinspirasi dari perilaku semut dalam mencari makanan, Ant Colony Optimization dapat mengoptimalkan pembagian jadwal shift, mengurangi konflik, dan meningkatkan kinerja karyawan. Hasil dari penelitian ini algoritma Ant Colony Optimization (ACO) mampu menghasilkan jadwal yang lebih optimal dengan efisien tinggi dengan Best Cost 100, dalam waktu 1 menit 6 detik, lebih baik dibandingkan metode lain seperti Particle Swarm Optimization (PSO) dengan Best Cost 7600, dalam waktu 4 detik dan Algoritma Genetika (GA) dengan Best Fitness 8500 dalam waktu 5 detik.

Kata Kunci*—Penjadwalan Tenaga Kerja, Ant Colony Optimization, Efisiensi Operasional*

Abstract

Effective workforce scheduling is crucial for enhancing productivity and maintaining service quality at PT. Cloud Hosting Indonesia. Workforce scheduling is the process of organizing and allocating labor to various tasks and responsibilities within an organization. Ant Colony Optimization (ACO) is a probabilistic technique used to solve computation problems by finding the best path through a graph. Inspired by the behavior of ants, particularly how they find food, Ant Colony Optimization can optimize shift schedules, reduce conflicts, and improve employee performance. However, there are current irregularities, insufficient rest periods, and unpredictable holidays. Ant colony optimization is applied to address these problems. The result of this shows that the Ant Colony Optimization algorithm is capable of producing more optimal schedules with high efficiency, achieving a Best Cost of 100 in 1 minute and 6 seconds. This is better compared to other methods such as Particle Swarm Optimization (PSO), which achieved

a Best Cost of 7600 in 4 seconds, and Genetic Algorithm (GA), which achieved a Best Fitness of 8500 in 5 seconds.

Keywords— *Workforce Scheduling, Ant Colony Optimization, Operational Efficiency*

1. PENDAHULUAN

Penjadwalan tenaga kerja adalah komponen krusial dalam manajemen operasional yang bertujuan memastikan tugas dilaksanakan secara efisien dan optimal. PT. Cloud Hosting Indonesia, yang bergerak dalam layanan hosting dan komputasi awan, menghadapi berbagai tantangan penjadwalan yang berdampak pada efektivitas kerja. PT. Cloud Hosting Indonesia ini memiliki 20 orang karyawan, yang khususnya pada kantor Sukabumi. Karyawan tersebut terdapat kurang optimal dengan jadwal yang ada, menyebabkan ketidakteraturan operasional [1].

Penjadwalan tenaga kerja (*workforce scheduling*) adalah salah satu penerapan strategis ilmu ergonomi dalam sistem organisasi, yang berpotensi meningkatkan kinerja sistem dan kesejahteraan pekerja. Meskipun telah banyak diteliti, sedikit penelitian yang memperhitungkan masalah ergonomi. Faktor ergonomi sering diabaikan dalam pemodelan penjadwalan tenaga kerja karena sulit dimodelkan secara matematis [2].

Algoritma *Ant Colony Optimization* (ACO) adalah algoritma sebuah pengembangan paradigma yang digunakan untuk menyelesaikan masalah optimasi [3]. Algoritma ini menggunakan metode probabilistik dan prinsip yang meniru perilaku kelompok koloni semut dalam mencari makanan. Melalui penggunaan feromon untuk menandai jalur yang dilalui, semut secara kolektif dapat menentukan rute terpendek menuju sumber makanan. Algoritma *Ant Colony Optimzation* mengadopsi prinsip ini untuk menemukan solusi optimal dalam berbagai masalah optimasi [4]

Salah satu masalah utama adalah kurangnya sistem untuk mengoptimalkan penjadwalan kerja, mengakibatkan ketidakcocokan antara permintaan dan ketersediaan tenaga kerja. Metode manual yang digunakan saat ini tidak fleksibel dan sulit disesuaikan dengan kebutuhan operasional yang dinamis [5]. Hal ini diperburuk oleh waktu istirahat yang tidak memadai dan jadwal libur yang tidak teratur, yang menambah beban pada karyawan dan menurunkan produktivitas serta kesejahteraan karyawan. Tekanan berlebihan yang mengalami karyawan berdampak negatif pada kualitas layanan dan efisiensi operasional Perusahaan [6].

Untuk mengatasi masalah ini, penerapan algoritma *Ant Colony Optimization* menawarkan solusi efektif. *Ant Colony Optimization*, sebuah metode metauristik yang terinspirasi dari perilaku semut dalam mencari makanan dengan meninggalkan jejak feromon, telah terbukti efektif dalam berbagai masalah optimasi seperti penjadwalan, rute kendaraan, dan alokasi sumber daya [7]. Menggunakan *Ant Colony Optimization* dalam penjadwalan tenaga kerja dapat membantu mengoptimalkan pembagian jadwal shift, mengurangi konflik, dan meningkatkan kinerja karyawan [8]. Penelitian ini bertujuan untuk mengembangkan model penjadwalan tenaga kerja yang optimal menggunakan algoritma *Ant Colony Optimization* dan mengevaluasi efektivitas penerapannya di PT. Cloud Hosting Indonesia.

2. METODE PENELITIAN

2.1 Metode Pengembangan Sistem

Metode penelitian yang digunakan dengan pendekatan penelitian berencana (*Planning Approaches*) [9], pengumpulan data dengan sistem internal dari PT. Cloud Hosting Indonesia. Data yang dikumpulkan berupa data penjadwalan *shift* kerja [10]. Metode analisis yakni dengan analisis fishbone dilakukan untuk mengidentifikasi berbagai penyebab masalah (cause) yang

terjadi dalam suatu situasi (Effect) [11]. Adapun alternatif Solusi pada implementasi algoritma yang digunakan dengan Ant Colony Optimization [12]. Adapun untuk penyelesaian masalah pada peneltian diguinakan flowchart perubahan tingkat pheromone dilanjutkan dengan persamaan penguapan feromon yang telah disebutkan sebelumnya [13] dan untuk Bahasa pemograman yang digunakan yakni dengan Python dimana yakni salah satu bahasa pemrograman yang sering digunakan oleh pengembang dalam pembuatan program [14]. Proses selanjutnya adalah menggunakan *tool software google collabs*. Untuk mengkonfigurasi pengaturan langkah demi langkah untuk menangani semua optimalisasi algoritma. *tool goolge collabs* adalah *tool* data *science* dan *mechine learning* berbasis *python* populer di kalangan praktisi dan akademis dan banyak digunakan dalam penelitian [15].

3. HASIL DAN PEMBAHASAN

3.1 Tempat Penelitian

Analisis dan perancangan ini berda pada PT Cloud Hosting Indonesia yang beralamat PT. Cloud Hosting Indonesia yang berlokasi di Jl. Raya Bojonggenteng No.2, Bojonggenteng, Kab. Sukabumi, Jawa Barat, 43353. [16].

3.2 Analisis Sistem Berjalan

PT. Cloud Hosting Indonesia menghadapi berbagai kendala dalam penjadwalan tenaga kerja yang menyebabkan kinerja kurang efektif. Tidak semu karyawan dapat berfungsi optimal dengan jadwal yang ada, mengakibatkan ketidakteraturan operasional. Waktu istirahat yang tidak memadai dan libur yang tidak menentu menambah beban karyawan, mengurangi produktivitas dan kesejahteraan mereka. Kurangnya sistem untuk mengoptimalkan penjadwalan kerja menyebabkan ketidakcocokan antara permintaan dan ketersediaan tenaga kerja. Metode manual dalam penjadwalan yang kaku dan sulit disesuaikan dengan kebutuhan, ditambah dengan tekanan yang berlebihan, berdampak negatif pada kualitas layanan dan efisiensi operasional perusahaan.

3.3 Analisis Kebutuhan Sistem

Pengumpulan data berdasarkan sistem internal dari PT. Cloud Hosting Indonesia. Data yang dikumpulkan berupa data penjadwalan *shift* kerja. Analisis *fishbone* dilakukan untuk mengidentifikasi berbagai penyebab masalah (cause) yang terjadi dalam suatu situasi (Effect). Masalah utama yang dianalisis adalah ketidakefisien penjadwalan tenaga kerja. Implementasi algoritma dilakukan dengan inisialisasi (menentukan parameter-parameter awal yang diperlukan untuk algoritma *Ant Colony Optimization*, seperti jumlah semut, Tingkat evaporasi feromon, dan intensitas feromon dan fungsi heuristik. Tahap inisialisasi memerlukan nilai visibilitas atau invers jarak antar node), Penentuan Nilai visibilitas dinyatakan dengan η, kontruksi solusi yakni setiap semut membangun solusi penjadwalan tenaga kerja dengan memilih elemen-elemen penjadwalan berdasarkan probabilitas yang dipengaruhi oleh jejak feromon dan fungsi heuristik. Pada tahap ini, semut akan memilih node tujuan secara acak. Setelah semua semut menyelesaikan konstruksi solusi, jejak feromon diperbarui berdasarkan kualitas solusi yang ditemukan. Solusi terbaik akan meninggalkan jejak feromon yang lebih kuat. Proses kontruksi solusi dan pembaruan feromon diulang hingga mencapai kondisi penghentian, seperti jumlah iterasi maksimum atau konvergensi solusi. Setelah semua semut melewati semua node dan menemukan rute terpendek, kemudian perubahan tingkat *pheromone* dilanjutkan dengan persamaan penguapan feromon yang telah disebutkan sebelumnya.

3.4 Data Hasil Penelitian

Data hasil penelitian ini merupakan data internal sejumlah 3 bulan, dengan total karyawan 20 orang.

Hasil Penerapan Penjadwalan Tenaga Kerja Menggunakan Algoritma *Ant Colony Optimization*

a. Hasil Iterasi Pertama

Gambar di atas menunjukkan hasil dari iterasi algoritma *Ant Colony Optimization* (ACO) untuk penjadwalan tenaga kerja. Setiap baris mengindikasikan iterasi ke-x dari total 100 iterasi, serta *cost* terbaik yang ditemukan pada iterasi tersebut. Setiap iterasi mencatat nilai *best cost*, yang merupakan *cost* atau hasil terbaik yang dicapai oleh algoritma pada iterasi tersebut. Lihat gambar 1 dibawah ini.

Gambar 1. Hasil Iterasi Pertama

Ada penurunan signifikan pada nilai *Best Cost* selama iterasi awal, dari 700 ke 300 dalam 4 iterasi. Setelah iterasi ke-4, *Best Cost* tetap stabil pada nilai 300, menunjukkan bahwa algoritma mungkin telah mencapai nilai optimal atau minimum lokal pada iterasi ke-5 dan seterusnya. Algoritma menunjukkan peningkatan signifikan dalam *Best Cost* selama 4 iterasi pertama. Setelah iterasi ke-4 *Best Cost* mencapai stabilitas pada nilai 300, menunjukkan bahwa algoritma telah menemukan solusi optimal atau lokal.

b. Hasil Iterasi Kedua

Pada gambar menunjukkan hasil iterasi dari algoritma *Ant Colony Optimization* (ACO) untuk penjadwalan tenaga kerja, dengan masing-masing baris menunjukkan iterasi ke-x dari total 100 iterasi dan *cost* terbaik yang ditemukan pada iterasi tersebut. Ada penurunan signifikan pada nilai *Best Cost* pada iterasi awal, dari 500 menjadi 400 dalam 2 iterasi. *Best Cost* kemudian stabil pada nilai 400 selama 61 iterasi (dari iterasi 3 hingga 63). Ada penurunan bertahap pada iterasi 64, 65, dan 70, masing-masing menjadi 300, 200, dan akhirnya 100. Setelah iterasi ke-70, *Best Cost* stabil pada nilai 100 hingga iterasi ke-100. Algoritma menunjukkan peningkatan signifikan dalam *Best Cost* selama iterasi awal, dengan penurunan bertahap pada beberapa iterasi berikutnya. Lihat gambar 2 di halaman selanjutnya.

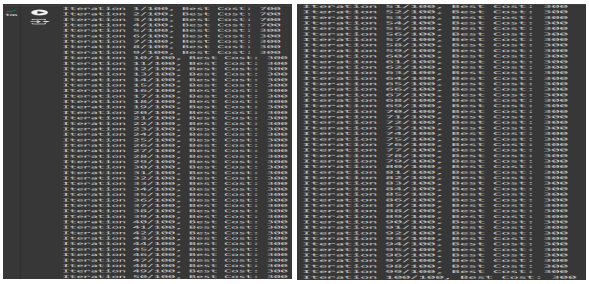
Gambar 2. Hasil Iterasi Kedua

c. Hasil Iterasi Ketiga

Pada Gambar menunjukkan hasil iterasi dari Algoritma *Ant Colony Optimization* (ACO) untuk penjadwalan tenaga kerja. Masing – masing baris menggambarkan iterasi ke-x dari total 100 iterasi dan *cost* terbaik yang ditemukan pada iterasi tersebut. Algoritma tidak menunjukkan perubahan dalam *Best Cost* selama 50 iterasi pertama, stabil di angka 400. Pada iterasi ke-51, terdapat peningkatan signifikan dengan *Best Cost* turun menjadi 300. Setelah iterasi ke-51, *Best Cost* tetap stabil di angka 300 hingga iterasi ke-100, menunjukkan algoritma telah menemukan solusi optimal atau minimum lokal. Lihat gambar 3 di bawah ini.

Gambar 3. Hasil Iterasi Ketiga

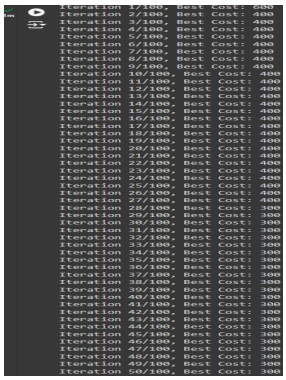
d. Hasil Iterasi Keempat

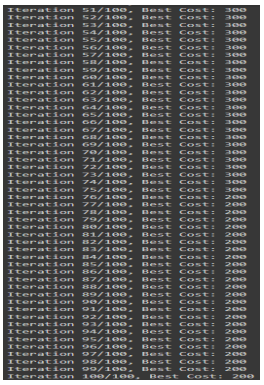


Gambar 4 Hasil Iterasi Keempat

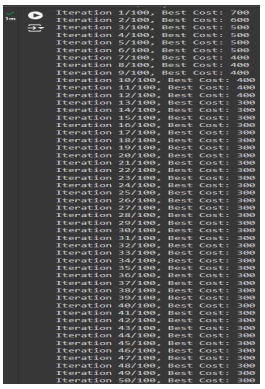
Algoritma menunjukkan peningkatan signifikan dalam *Best Cost* selama 4 iterasi pertama, dengan penurunan dari 700 menjadi 300. Setelah iterasi ke-4, *Best Cost* stabil pada nilai 300, menunjukkan bahwa algoritma telah menemukan solusi optimal atau minimum lokal dan mempertahankannya hingga iterasi ke-100. Hal ini ditampilkan pada gambar 4 sebelumnya.

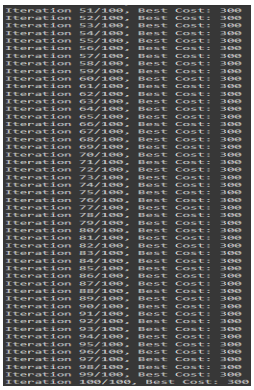
e. Hasil Iterasi Kelima dan Keenam





Gambar 5. Hasil Iterasi Kelima



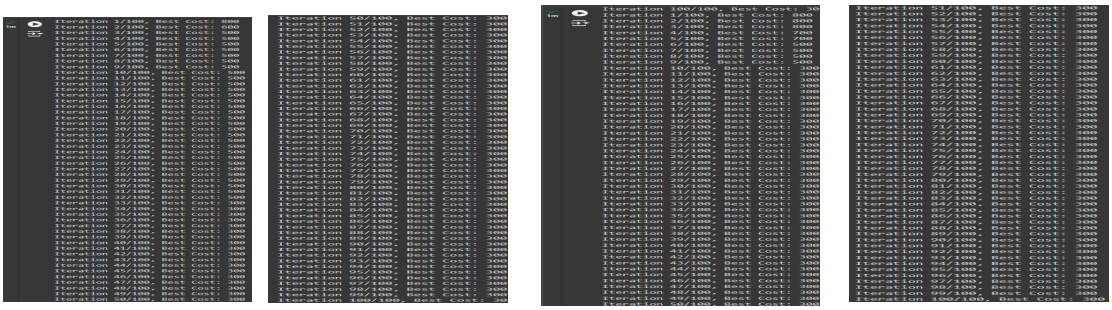


Gambar 6. Hasil Iterasi Keenam

Pada hasil iterasi kelima Algoritma menunjukkan peningkatan signifikan dalam *Best Cost* selama iterasi awal, dengan penurunan dari 500 menjadi 400 dalam 2 iterasi pertama. *Best Cost* tetap stabil pada nilai 400 selama 48 iterasi berikutnya (dari iterasi 3 hingga 50). Pada iterasi ke-51, *Best Cost* kembali turun signifikan menjadi 300 dan stabil pada nilai tersebut hingga iterasi ke-83. Pada iterasi ke-84 Best menurun lagi menjadi 200 dan tetap stabil iterasi ke-100. Lihat gambar 5 di atas.
Dan pada ieterasi keenam Algoritma menunjukkan beberapa penurunan signifikan dalam *Best Cost* selama 16 iterasi pertama, dari 700 menjadi 300. Setelah iterasi ke-16, *Best Cost* stabil pada nilai 300 hingga iterasi ke-100, menunjukkan bahwa algoritma telah menemukan solusi optimal atau minimum lokal dan mempertahankannya hingga akhir iterasi. Lihat gambar 6 di atas.

f. Hasil Iterasi Ketujuh dan Kedelapan

Hasil ieterasi ketujuh bahwa Algoritma menunjukkan penurunan signifikan dalam *Best Cost* selama iterasi awal, dari 800 menjadi 500 dalam dua iterasi pertama. Setelah iterasi ke-2, *Best Cost* tetap stabil pada nilai 500 hingga iterasi ke-36. Pada iterasi ke-37, ada penurunan signifikan lagi dengan *Best Cost* turun menjadi 300. Dari iterasi ke-41 hingga 100, *Best Cost* tetap stabil pada nilai 300, menunjukkan bahwa algoritma telah menemukan solusi optimal atau minimum lokal dan mempertahankannya hingga akhir iterasi. Lihat gambar 7 di halaman selanjutnya.



Gambar 7. Hasil Iterasi Ketujuh

```
Total melakukan optimasi: 2
Total waktu optimasi : 66.629962682724 detik
```

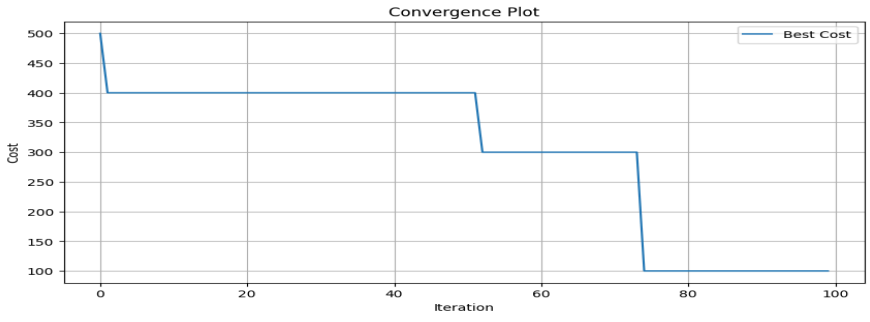
Gambar 8. Hasil Iterasi Kedelapan

Dan hasil ieterasi kedelapan Algoritma menunjukkan beberapa penurunan signifikan dalam *Best Cost* selama beberapa iterasi pertama, dengan penurunan terbesar dari 800 menjadi 300. Setelah iterasi ke-37, *Best Cost* stabil pada nilai 300 hingga iterasi ke-100, menunjukkan bahwa algoritma telah menemukan solusi optimal dan mempertahankannya. Proses ini telah dilakukan sebanyak dua kali, dengan total waktu eksekusi sekitar 1 menit 63 detik. Lihat gambar 8 di atas.

Evaluasi Kinerja Algoritma

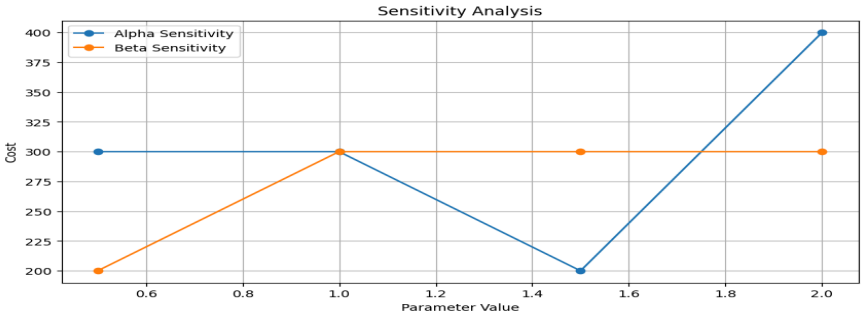
a. Analisis Konvergensi

Plot konvergensi menunjukkan nilai *Best Cost* menurun secara bertahap melalui beberapa penurunan tajam dan periode stabilitas. Nilai *Best Cost* awalnya menurun dari sekitar 500 menjadi sekitar 400, kemudian stabil untuk beberapa iterasi. Penurunan lebih lanjut terjadi di sekitar iterasi ke-50 dan ke-80, dengan *Best Cost* akhirnya mencapai nilai sekitar 100 dan stabil hingga iterasi akhir. Ini menunjukkan bahwa algoritma secara bertahap menemukan solusi yang lebih baik, dengan perbaikan signifikan pada beberapa titik iterasi, hingga mencapai nilai minimum yang stabil. Plot ini membantu dalam memvisualisasikan bagaimana algoritma berprogresi dan konvergen ke solusi optimal selama iterasi. Lihat gambar 9 di bawah ini.



Gambar 9 Hasil Analisis Konvergensi

b. Analisis Sensivitas



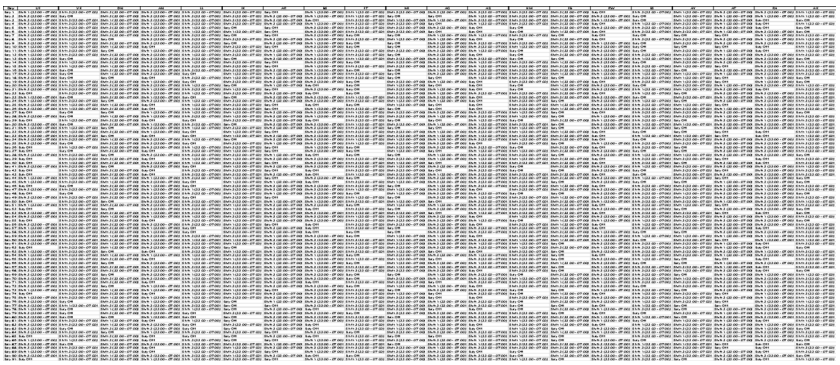
Gambar 10 Hasil Analisis Sensivitas

Alpha Sensitivity menunjukkan bahwa perubahan nilai *alpha* memiliki dampak signifikan terhadap *cost*, dengan penurunan tajam pada nilai 1.4 dan peningkatan drastis pada nilai 2.0. *Beta sensitivity* menunjukkan bahwa perubahan nilai *beta* memiliki dampak yang lebih stabil terhadap *cost*, dengan peningkatan awal dari 200 ke 300 pada nilai 1.0 dan stabil pada nilai-nilai berikutnya. Dengan memantau *optimality gap* pada setiap iterasi, dapat memahami seberapa cepat dan seberapa baik algoritma mendekati solusi optimal. Lihat gambar 10 di atas.

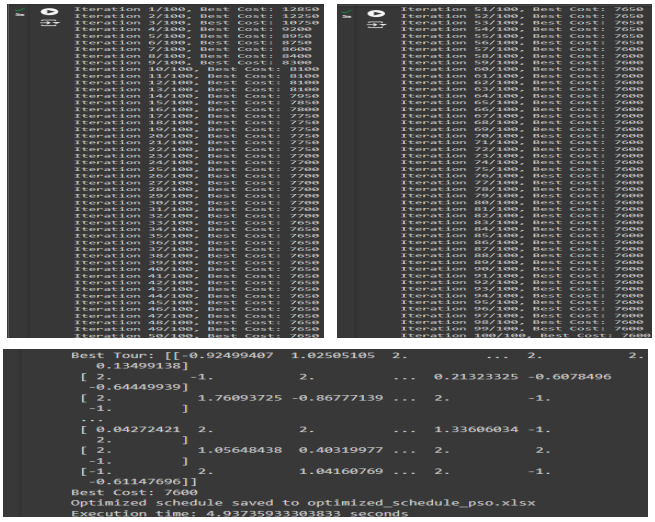
Perbandingan Hasil Penjadwalan Menggunakan Algoritma PSO dan Genetika

A. Hasil Penjadwalan Tenaga Kerja Setelah di Optimasi menggunakan Algoritma PSO

Pada gambar di bawah ini menunjukkan hasil dari sebuah proses iteratif yang berjalan sebanyak 100 kali. Jumlah Iterasi: Gambar menampilkan total iterasi, dimulai dari iterasi 1/100 hingga iterasi 100/100. *Best Cost*: Setiap iterasi memiliki nilai *Best Cost* yang menunjukkan *cost* terbaik yang dicapai oleh algoritma pada iterasi tersebut. Lihat gambar 11 dan gambar 12 di bawah ini.



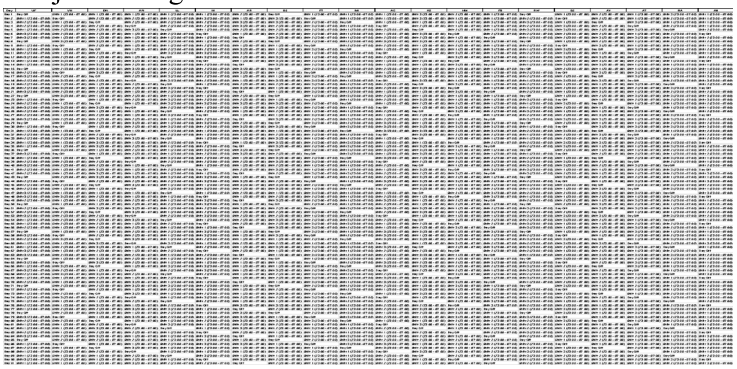
Gambar 11. Hasil Optimasi dari Algoritma PSO

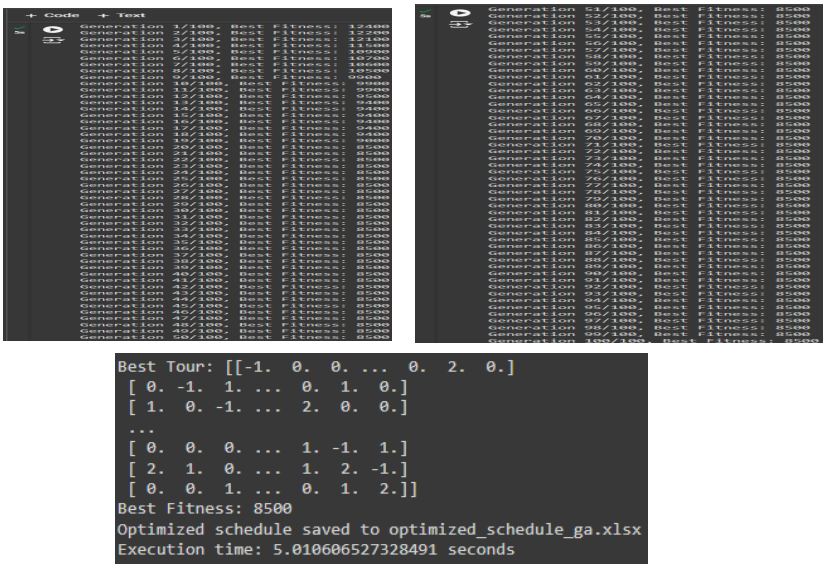


Gambar 12. Hasil Iterasi dari Algoritma PSO

B. Algoritma Genetika

Hasil Penjadwalan Tenaga Kerja Setelah di Optimasi Menggunakan Algoritma Genetika, dimana penurunan *Best Fitness* signifikan terjadi pada generasi awal, dari 12400 menjadi 8500 dalam 21 generasi. Setelah generasi ke-21, *Best Fitness* stabil pada angka 8500, menunjukkan bahwa algoritma telah mencapai nilai optimal atau minimum lokal dan tidak ada peningkatan lebih lanjut. Lihat gambar 13 di bawah ini.


Gambar 13. Hasil Penjadwalan yang Setelah di Optimasi Algoritma Genetika



Gambar 14. Hasil dari Algoritma Genetika

Algoritma menunjukkan peningkatan signifikan dalam *best fitness* selama 21 generasi pertama. Setelah generasi ke-21, *Best Cost* stabil pada nilai 8500, menunjukkan bahwa algoritma telah menemukan solusi optimal atau minimum lokal. Proses ini memerlukan waktu eksekusi sekitar 5,01 detik. Lihat gambar 14 di atas.

C. Hasil Perbandingan dari Tiga Algoritma

Tabel 1. Performa dari Masing-Masing Algoritma

Algoritma	Waktu Mesin	Best Cost	Best Fitness
Ant Colony Optimization	1 menit 6 detik	100	-
Particle Swarm Optimization	4 detik	7600	-
Genetika	5 detik	-	8500

Pada tabel 1 di atas menunjukkan perbandingan kinerja tiga algoritma optimasi, yaitu *Ant Colony Optimization, Particle Swarm Optimization*, dan Algoritma Genetika. Berikut penjelasannya:

- *Ant Colony Optimization* = Algoritma ini menghasilkan solusi dengan *Best Cost* sebesar 100, namun memerlukan waktu eksekusi yang paling lama yaitu 1 menit 6 detik.
- *Particle Swarm Optimization* = Meskipun memiliki *Best Cost* sebesar 7600, algoritma ini hanya memerlukan waktu 4 detik, menjadikannya yang tercepat di antara ketiga algoritma.
- Algoritma Genetika = Memiliki *Best Fitness* sebesar 8500 dan waktu eksekusi, menunjukkan keseimbangan antara kualitas solusi dan waktu eksekusi

Perbandingan ini mengilustrasikan efektivitas dalam hal waktu dan kualitas solusi yang dihasilkan oleh masing-masing algoritma untuk masalah optimasi tertentu. Berikut ini adalah penulis dapat menyimpulkan bahwa berdasarkan hasil dari perbandingan dan optimasi penjadwalan di atas, *Particle Swarm Optimization* adalah yang paling efisien dalam waktu eksekusi dan untuk optimasi

penjadwalan masih banyak yang menumpuk dari *shift* 3 ke *shift* 1 dan semakin tidak teratur pada jadwal libur, sementara Algoritma Genetika memberikan hasil terbaik dalam hal *fitness* dan optimasi penjadwalan sudah teratur namun masih banyak jadwal *shift* 3 ke *shift* 1 dari beberapa karyawan. *Ant Colony Optimization*, meskipun memberikan *Best Cost* dan hasil penjadwalan terbaik, membutuhkan waktu yang jauh lebih lama.

4. KESIMPULAN

Penelitian ini membuktikan bahwa penerapan Algoritma *Ant Colony Optimization* (ACO) dalam penjadwalan tenaga kerja di PT. Cloud Hosting Indonesia berhasil meningkatkan efisien dan produktivitas karyawan. ACO terbukti efektif dalam mengoptimalkan distribusi jadwal konflik, serta memastikan adanya waktu istirahat dan jadwal libur yang memadai. Hasil dari penelitian ini algoritma *Ant Colony Optimization* (ACO) mampu menghasilkan jadwal yang lebih optimal dengan efisien tinggi dengan B*est Cost* 100, dalam waktu 1 menit 6 detik, lebih baik dibandingkan metode lain seperti *Particle Swarm Optimization* (PSO) dengan *Best Cost* 7600, dalam waktu 4 detik dan Algoritma Genetika (GA) dengan *Best Fitness* 8500, dalam waktu 5 detik.

5. SARAN

a. Semakin banyak sampel dataset pada penelitian selanjutnya diharapkan dapat meningkatkan nilai akurasi.

b. Pada peneliti selanjutnya penulis mengharapkan agar membuat sistem aplikasi penjadwalan tenaga kerja di PT. Cloud Hosting Indonesia.

c. Penelitian berikutnya dapat mempertimbangkan lebih banyak variabel dan kondisi operasional untuk meningkatkan akurasi dan efisiensi algoritma

DAFTAR PUSTAKA

[1] N. S. W. G. R. B. I. &. A. G. Luh, "IMPLEMENTASI ALGORITMA GENETIKA BERBASIS WEB PADA SISTEM PENJADWALAN MENGAJAR DI SMK DWIJENDRA DENPASAR.," *JuTIK,* no. https://jurnal.undhirabali.ac.id/index.php/jutik/article/view/646/pdf, pp. 1-9, 2019.

[2] D. Setiawan, "Model Penjadwalan Tenaga Kerja Mempertimbangkan Batasan Lingkungan dan Konsumsi Energi," *Jurnal UII,* vol. 25 (2), no. https://journal.uii.ac.id/jurnal-teknoin/article/view/12206, pp. 1-10, 2019.

[3] M. B. M. &. S. T. Dorigo, The Ant Colony Optimization Metaheuristic: Algorithm, Applications, and Advances, Iridia, Belgium, 2016.

[4] S. Nurharyanto & Perdana, "Menentukan Rute Distribusi Di PT Sinar Harapan Plastik Dengan Metode Algoritma Ant Colony Optimization," *IKRAITH-Teknologi,* 2021.

[5] D. &. H. A. Oktarina, " PERANCANGAN SISTEM PENJADWALAN SEMINAR PROPOSAL DAN SIDANG SKRIPSI DENGAN METODE ALGORITMA GENETIKA," *JOISIE Journal Of Information System And Informatics Engineering, ,* vol. 3(1), p. 32–40, 2019.

[6] H. Gandhi, "Penyelesaian Assignment Problem Dengan Algoritma Metaheuristik Ant Colony Optimization (ACO)," *JISS-Jurnal Industrial Servicess,* 2019.

[7] C. X. Blum, "Swarm Intelligence in Optimization," *Science Direct Journal,* 2018.

[8] N. I. P. A. &. F. N. Sari, "ANALISIS PENJADWALAN TENAGA KERJA DENGAN MENGGUNAKAN METODE TIBREWALA, PHILIPPE DAN BROWNE PADA PT. JAPFA COMFEED INDONESIA TBK. UNIT MAKASSAR," *Journal of Industrial Engineering Management,* vol. 4(2), no. https://doi.org/10.33536/jiem.v4i2.455, p. 58–62, 2019.

[9] A. S. H. Riana, "Simulasi Penentuan Lintasan Terpendek Pada Complete Graph Dengan Menggunakan Ant Colony Optimization Algorithm," *KARISMATIKA, Vol. 6(2),* 2020.

[10] A. Arisky, "Laporan Kerja Praktek PT Cloud Hosting Indonesia Aplikasi To Do List Menggunakan Flutter Berbasis Mobile," Politeknik Negeri Bengkalis Riau, Riau, 2022.

[11] I. &. S. B. Aisyiah, "Optimalisasi Pelayanan Instalasi Gawat Darurat Menggunakan Analisis Fishbone (Studi Kasus Pada RSIA X Padang) The Optimalization Of Accident And Emergency Services Using Fishbone Analysis (A Case Study At RSIA X Padang)," *Jurnal Apresiasi Ekonomi, 9(1),* pp. 30-37, 2021.

[12] V. Y. H. &. S. R. Rizqiyanti, "Pencarian Jalur Terpendek Menggunakan Metode Algoritma “Ant Colony Optimization” Pada GUI Matlab (Studi Kasus: PT Distriversa Buana Mas cabang Purwokerto)," *Jurnal Universitas Diponegoro. 8(2),* pp. 272-284, 2019.

[13] M. A. J. H. J. &. K. R. N. P. Pinedo, "Scheduling Theory, Algorithms, and Systems 4th Edition.," 2020.

[14] I. G. Harjumawan Wiratmaja KS, "Program Menghitung Banyak Bata pada Ruangan Menggunakan Bahasa Python," *TIERS Information Technology Journa, Vol, 2, No. 1,* p. 12–22, 2021.

[15] S. S. Sukhdeve Dr. Shitalkumar R. and Sukhdeve, "Google Colaboratory. In Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services (pp. 11–34).," 2023. [Online]. Available: https://doi.org/10.1007/978-1-4842-9688-.

[16] Arthur, "Hosting Indonesia Performa Terbaik," Jakarta, 2024.