

Implementasi Algoritma *Particle Swarm Optimization* pada AS/RS Berbasis Mikrokontroler Arduino

Ruminto Subekti¹, Nur Jamiludin Ramadhan², Agnia Hanifah³

ruminto@ae.polman-bandung.ac.id¹, jamil@ae.polman-bandung.ac.id²,

agnia.hanifah@gmail.com³

^{1,2,3} Politeknik Manufaktur Bandung

Informasi Artikel

Diterima : 15 Jul 2024
Direvisi : 13 Agu 2024
Disetujui : 25 Agu 2024

Kata Kunci

Algoritma *Particle Swarm Optimization* (PSO), *Automated Storage and Retrieval System* (AS/RS), Efisiensi Operasional, Gudang

Abstrak

Sistem gudang manual di Indonesia menyebabkan masalah pelacakan dan penyimpanan barang, meningkatkan risiko kerusakan. Penerapan *Particle Swarm Optimization* (PSO) dalam *Automated Storage and Retrieval Systems* (AS/RS) dapat mengoptimalkan jalur perjalanan, mengurangi kesalahan, dan meningkatkan efisiensi penyimpanan. Parameter seperti partikel, iterasi, dan *inertia* diterapkan dalam sistem ini. Studi ini mengintegrasikan PSO dengan AS/RS berbasis Arduino menggunakan metode VDI 2206, dengan rak AS/RS yang dirancang dalam matriks 4x4. Dalam berbagai studi kasus, PSO berhasil mengurangi jarak tempuh antara 1% hingga 32% dan menghemat waktu operasional antara 5% hingga 21%. PSO secara efektif meningkatkan efisiensi operasional sistem AS/RS dengan mengoptimalkan rute perjalanan, sehingga mengurangi waktu yang dibutuhkan untuk operasi penyimpanan dan pengambilan barang. Selain itu, perlu diperhatikan bahwa nilai selisih pengurangan jarak juga dipengaruhi oleh input alamat awal tanpa PSO, yang mungkin sudah mendekati kondisi optimal. Dengan demikian, meskipun PSO secara signifikan meningkatkan efisiensi, besarnya pengurangan dapat bervariasi tergantung pada seberapa dekat input alamat awal tanpa PSO dengan rute optimal.

Keywords

Particle Swarm Optimization (PSO), *Automated Storage and Retrieval System* (AS/RS), Operational Efficiency, Warehouse

Abstract

Manual warehouse systems in Indonesia cause tracking and storage issues, increasing the risk of damage. Implementing *Particle Swarm Optimization* (PSO) in *Automated Storage and Retrieval Systems* (AS/RS) can optimize travel paths, reduce errors, and enhance storage efficiency. Parameters such as particles, iterations, and *inertia* are applied in this system. This study integrates PSO with Arduino-based AS/RS using the VDI 2206 method, with AS/RS racks designed in a 4x4 matrix. In various case studies, PSO successfully reduced travel distances by 1% to 32% and operational time by 5% to 21%. PSO effectively enhances the operational efficiency of AS/RS by optimizing travel routes, thereby reducing the time needed for storage and retrieval operations. Additionally, it is important to note that the reduction in travel distance is influenced by the initial input addresses without PSO, which might already be near optimal. Thus, while PSO significantly improves efficiency, the extent of reduction can vary based on how close the initial input addresses without PSO are to the optimal route.

A. Pendahuluan

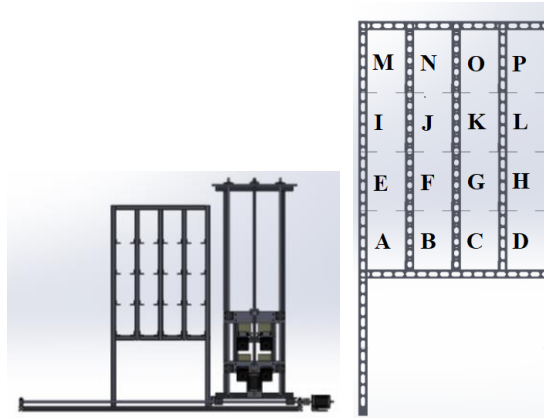
Perkembangan teknologi di perusahaan Indonesia, baik produksi maupun jasa, semakin pesat. Persaingan yang ketat mendorong perusahaan untuk meningkatkan kinerja dan memberikan pelayanan terbaik bagi konsumen. Keberadaan gudang sangat penting sebagai pusat penyimpanan dan distribusi, memastikan ketersediaan barang yang efisien dan tepat waktu [1]. Gudang adalah fasilitas pendukung operasi industri untuk menyimpan bahan baku, produk jadi [2]. Pergudangan harus mengatur barang sesuai jenis, ukuran, berat, dan sifatnya untuk menghindari gangguan, sehingga proses keluar masuk barang dapat efektif dan efisien [3].

Beberapa permasalahan yang disebabkan oleh sistem pengaturan pada gudang yang tidak tertata dengan baik menyebabkan pencatatan, penyimpanan, dan pengambilan barang masih belum tertata [4]. Penyimpanan manual dengan mesin pengangkat barang berisiko tinggi mengalami kerusakan akibat kecelakaan kerja, seperti alat bantu menabrak rak atau kelalaian manusia [5]. Peningkatan variasi produk akan meningkatkan beban kerja operator. Hal ini dapat menyebabkan kesalahan manusia, yang dapat berakibat pada ketidakakuratan data atau hilangnya barang [6]. Peningkatan efisiensi dalam proses pengambilan pesanan sering kali dicapai dengan mengurangi jarak yang harus ditempuh, yang pada gilirannya mempersingkat waktu perjalanan ke lokasi pengambilan [7].

Gudang manual tidak mendukung kebutuhan saat ini. Gudang modern dengan fasilitas data dan otomatisasi sangat diperlukan. Gudang otomatis ini signifikan meningkatkan kecepatan proses pengiriman barang dan akuisisi data [8]. Kinerja yang lebih tinggi dan sehingga, dibuatlah Penyimpanan Otomatis dan Sistem Pengambilan (AS/RS) semakin populer dan banyak menggantikan gudang manual [9].

Sistem Penyimpanan dan Pengambilan Otomatis (ASRS) adalah sebuah sistem yang dirancang untuk menyimpan dan mengambil suatu barang secara otomatis pada rak penyimpanan [10]. AS/RS adalah contoh aplikasi kontrol gerak dalam manajemen gudang, menggunakan crane atau robot arm untuk otomatis menyimpan dan mengambil barang. Sistem ini harus menjaga stabilitas dan akurasi gerakan, terutama pada kecepatan konstan atau percepatan, untuk memastikan barang ditangani dengan aman dan efisien [11], [12].

Penelitian menunjukkan bahwa penerapan AS/RS berdampak signifikan pada operasi gudang, meningkatkan kecepatan, kapasitas, akurasi, dan penggunaan ruang. Penelitian ini mengumpulkan bukti empiris sebelum dan setelah implementasi AS/RS [13].



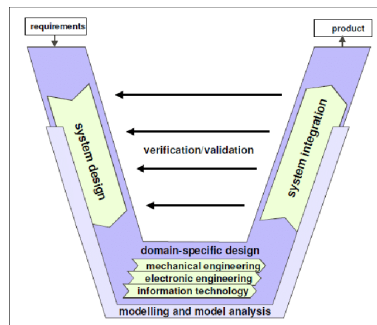
Gambar 1. Rak AS/RS [14]

Penelitian menunjukkan bahwa metode Association Rule Mining yang baru dalam konteks ini memberikan dampak positif dibandingkan solusi gudang tradisional. Metode ini mengidentifikasi pola interaksi item, komunitas item, dan entitas berpengaruh secara lokal dan global [15]. Namun penelitian tersebut masih terdapat kekurangan yakni belum diterapkannya algoritma untuk mempelajari kebijakan pengalokasian lokasi penyimpanan yang memperpendek jarak perjalanan dan membandingkan beberapa kebijakan alokasi tempat penyimpanannya [16].

Berdasarkan penelitian sebelumnya dapat diperbaiki dan digabungkan untuk merancang sistem menggunakan algoritma yang menyelesaikan masalah optimasi dan keputusan pada sistem AS/RS [17]. *Particle Swarm Optimization* (PSO) adalah algoritma cerdas terinspirasi dari perilaku kawanan burung atau ikan untuk memecahkan masalah optimasi nonlinear, nonkonveks, atau kombinatorial di berbagai bidang ilmu dan teknik. [18]. Bertujuan bahwa mesin harus melakukan perjalanan di jalur terpendek dan harus menyelesaikan siklus dalam waktu sesingkat mungkin, dengan mengoptimalkan rute [19]. Maka dari itu diajukan sebuah penelitian dengan judul “Implementasi Algoritma *Particle Swarm Optimization* pada AS/RS Berbasis Mikrokontroler Arduino” untuk meningkatkan efisiensi penyimpanan atau pengambilan barang pada gudang dengan mengoptimalkan urutan perpindahan objek dan jalur lintasannya dengan pendekatan yang lebih efektif.

B. Metode Penelitian

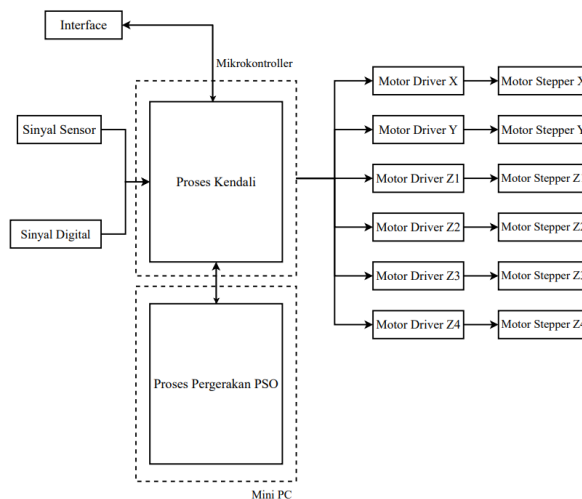
Sistem ini dirancang menggunakan metode VDI 2206, pedoman desain sistem mekatronika. Mekatronika menggabungkan teknologi mekanik, listrik, elektronik, dan komputer untuk meningkatkan produk, proses, dan sistem. Pedoman VDI 2206, yang terdiri dari enam langkah, menyediakan panduan praktis untuk pengembangan produk yang sistematis.



Gambar 2. Metode VDI 2206

Tahap awal proses desain dimulai dengan menentukan dan merancang kebutuhan sistem, dituangkan dalam formulir kebutuhan sebagai alat ukur evaluasi produk akhir. Sistem dirancang menjadi beberapa subsistem yang masing-masing memiliki tugas berdasarkan prinsip operasional dan kinerja yang diharapkan. Dalam desain domain-spesifik, detail penafsiran dan perhitungan memastikan fungsi sistem bekerja baik setelah subsistem dibangun, menggunakan metode *waterfall* untuk perancangan perangkat lunak, meliputi analisis kebutuhan pengguna, desain sistem dengan Data Flow Diagram (DFD) dan Entity Relationship Diagram (ERD), pembuatan kode, pengujian, dan implementasi. Hasil desain tiap domain diintegrasikan menjadi sistem menyeluruh dengan modul-modul terpisah yang kemudian dihubungkan. Proses desain terus ditinjau untuk memastikan solusi sesuai karakteristik dan kebutuhan. Tahap pemodelan dan analisis karakteristik dilakukan dengan perangkat lunak dan simulasi. Produk akhir dihasilkan setelah prosedur perancangan, menandai tahap untuk mengubah desain menjadi sistem nyata atau meningkatkan kematangan produk.

1. Gambaran Umum Sistem

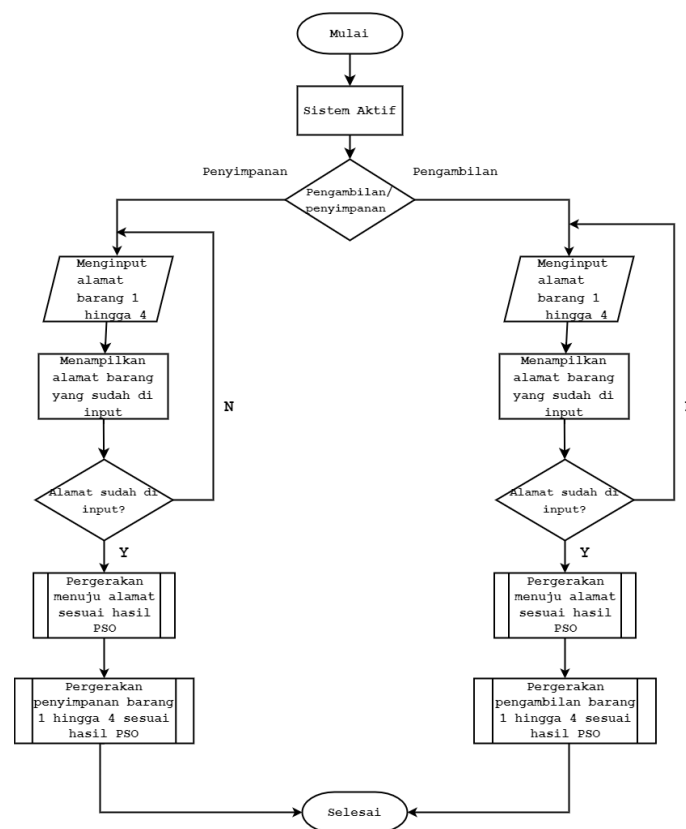


Gambar 3. Gambaran Umum Sistem

Pada gambar 3 menunjukkan gambaran sistem penyimpanan dan pengambilan barang secara otomatis. Sistem ini tentu nya memiliki proses pada umum nya, yaitu *input*, *process*, dan *output*. Selain itu, pada sistem ini menggunakan mikrokontroler Arduino. Tahap pertama dimulai dengan pengiriman sinyal sensor serta sinyal digital, sehingga pada proses kendali,

arduino akan mengolah data yang telah terbaca dari sinyal sensor, sinyal digital, dan HMI. Selanjut nya Sistem pergerakan ini yang nantinya akan menentukan jalur mana yang paling efisien dari home position menuju alamat rak yang sudah di input melalui interface HMI pada tahap awal. Penentuan jalur ini di proses dengan menggunakan algoritma PSO yang akan diolah dengan *mini pc*. Untuk pengaturan alamat barang menggunakan HMI yang dimana pada HMI ini juga dapat mengontrol, memilih tujuan barang yang akan disimpan atau diambil pada rak yang tersedia. Lalu setelah mendapatkan alamat yang sesuai dari apa yang telah di olah dengan menggunakan algoritma PSO, arduino akan secara otomatis mengaktifkan motor stepper untuk digerakan sehingga barang akan di simpan atau di ambil sesuai yang user inginkan melalui *interface*.

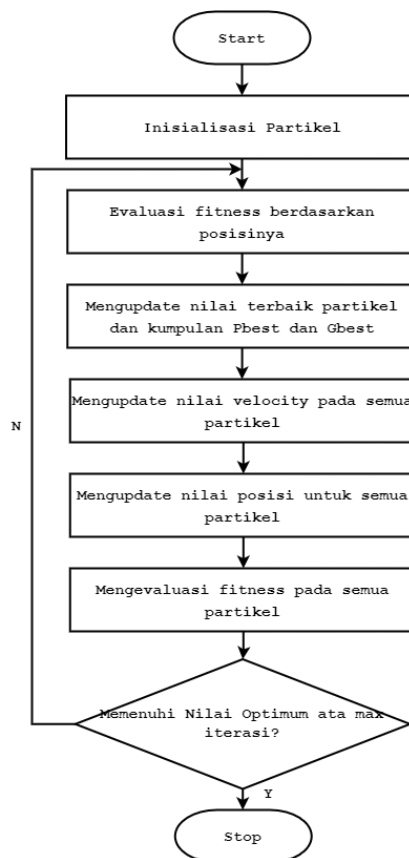
2. Diagram Alir AS/RS



Gambar 4. Diagram Alir AS/RS

Flowchart ini menggambarkan proses pengambilan dan penyimpanan barang menggunakan algoritma PSO. Proses dimulai dengan aktivasi sistem, setelah itu sistem bercabang menjadi dua alur yaitu pengambilan dan penyimpanan barang. Pada setiap cabang, pengguna menginput alat atau barang dari 1 hingga 4, lalu sistem menampilkan barang yang sudah diinput. Kemudian, sistem memeriksa apakah alat atau barang sudah diinput sepenuhnya. Jika belum, alur kembali ke langkah penginputan, namun jika sudah, sistem bergerak sesuai hasil optimasi PSO untuk mengambil atau menyimpan barang. Akhirnya, proses berakhir setelah semua barang diproses.

3. Subproses Pergerakan Menuju Alamat dengan Algoritma PSO



Gambar 5. Subproses Pergerakan Menuju Alamat dengan Algoritma PSO

Flowchart diatas menggambarkan langkah-langkah dalam algoritma PSO. Pertama, proses dimulai dengan inisialisasi variabel. Kemudian, dilanjutkan dengan inisialisasi partikel yang mewakili solusi dalam ruang pencarian. Setelah itu, setiap partikel dievaluasi untuk menentukan nilai fitness berdasarkan posisinya. Selanjutnya, kecepatan dan posisi partikel diperbarui, mengacu pada posisi terbaik dari partikel itu sendiri dan keseluruhan partikel. Kemudian, evaluasi fitness dilakukan kembali setelah posisi diperbarui untuk mengukur perbaikan solusi. Proses ini berulang, mengecek apakah nilai fitness optimal telah tercapai atau iterasi maksimum telah dipenuhi. Jika salah satu kondisi terpenuhi, algoritma bergerak menuju setiap sumbu untuk eksplorasi menyeluruh dan akhirnya berakhir setelah seluruh iterasi selesai, menghasilkan solusi optimal dari proses optimasi.

C. Hasil dan Pembahasan

1. Pengujian efesiensi algortima *Particle Swarm Optimization*

Pada pengujian ini dilakukan beberapa pengujian seperti perbedaan parameter partikel, iterasi, dan inertia.

Tabel 1. Pengujian Perbedaan Parameter Partikel

Pengujian	Alamat	Parameter			Jarak Terkecil (mm)
		Partikel	Iterasi	Inertia	
Pengujian 1	H - A - N - I	3	3	0.1	1333
Pengujian 2	H - A - N - I	5	3	0.1	1183
Pengujian 3	H - A - N - I	7	3	0.1	1169

Pada pengujian tabel 1 menunjukkan bahwa penambahan jumlah partikel dalam algoritma PSO memiliki pengaruh signifikan dalam menemukan jalur paling optimal. Pada pengujian pertama dengan 3 partikel dan 3 iterasi, jarak terkecil yang diperoleh adalah 1333. Pada pengujian kedua dengan 5 partikel dan 3 iterasi, jarak terkecilnya menurun menjadi 1183, menunjukkan peningkatan efisiensi. Pada pengujian ketiga dengan 7 partikel dan 3 iterasi, jarak terkecil yang diperoleh adalah 1169, jarak terkecil dari semua pengujian. Hasil ini menunjukkan bahwa dengan meningkatnya jumlah partikel, jarak terkecil yang ditemukan semakin berkurang, menandakan peningkatan efisiensi dalam pencarian solusi optimal. Kesimpulannya, lebih banyak partikel dalam algoritma PSO dapat membantu menemukan jalur yang lebih optimal, dan iterasi yang cukup memastikan konvergensi ke solusi terbaik.

Tabel 2. Pengujian Perbedaan Parameter Iterasi

Pengujian	Alamat	Parameter			Jarak Terkecil (mm)
		Partikel	Iterasi	Inertia	
Pengujian 1	H - A - N - I	3	3	0.1	1509
Pengujian 2	H - A - N - I	3	5	0.1	1358
Pengujian 3	H - A - N - I	3	7	0.1	1183

Berdasarkan data pengujian tabel 2 menghasilkan bahwa penambahan jumlah iterasi memiliki pengaruh signifikan dalam menemukan jalur paling optimal. Pada pengujian pertama dengan 3 iterasi, jarak terkecil yang diperoleh adalah 1509. Pada pengujian kedua dengan 5 iterasi, jarak terkecil berkurang menjadi 1358. Pada pengujian ketiga dengan 7 iterasi, jarak terkecil semakin berkurang menjadi 1183, yang merupakan jarak terkecil dari semua pengujian. Hasil ini menunjukkan bahwa dengan meningkatnya jumlah iterasi, jarak terkecil yang ditemukan semakin berkurang, yang menunjukkan peningkatan efisiensi dalam pencarian solusi optimal. Hal ini menandakan bahwa lebih banyak iterasi memungkinkan partikel untuk mengeksplorasi ruang solusi lebih baik dan menemukan rute yang lebih optimal. Kesimpulannya, jumlah iterasi yang lebih banyak dalam algoritma PSO dapat membantu menemukan jalur yang lebih optimal.

Tabel 3. Pengujian Perbedaan Parameter Inertia

Pengujian	Alamat	Parameter			Jarak Terkecil (mm)
		Partikel	Iterasi	Inertia	
Pengujian 1	H - A - N - I	3	3	0.1	1183
Pengujian 2	H - A - N - I	3	3	0.5	1333
Pengujian 3	H - A - N - I	3	3	0.7	1534

Berdasarkan data hasil pengujian tabel 3 bahwa perubahan nilai inertia mempengaruhi stabilitas dan efisiensi dalam menemukan jalur paling optimal. Pada pengujian pertama dengan inertia sebesar 0.1, jarak terkecil yang diperoleh adalah 1183, menunjukkan hasil yang lebih optimal namun dengan variasi antar partikel yang lebih besar. Pada pengujian kedua dengan inertia sebesar 0.5, jarak terkecil yang diperoleh adalah 1333, dengan hasil yang lebih stabil dibandingkan dengan inertia 0.1. Pada pengujian ketiga dengan inertia sebesar 0.7, jarak terkecil yang diperoleh adalah 1534, menunjukkan hasil yang paling stabil namun kurang optimal dibandingkan nilai inertia yang lebih rendah. Hasil ini menunjukkan bahwa dengan meningkatnya nilai inertia, stabilitas pencarian meningkat, tetapi efisiensi dalam menemukan solusi optimal menurun. Nilai inertia yang lebih tinggi cenderung menjaga partikel dalam pencarian yang lebih stabil dan eksplorasi yang lebih terkontrol, tetapi mungkin kurang efektif dalam menemukan solusi optimal dibandingkan nilai inertia yang lebih rendah. Kesimpulannya, nilai inertia yang lebih tinggi dalam algoritma PSO dapat membantu menjaga stabilitas pencarian, meskipun dapat mengorbankan efisiensi dalam menemukan jalur yang paling optimal.

2. Hasil perbandingan waktu pergerakan dengan algoritma PSO dan tanpa PSO

Dibawah ini adalah data hasil pengujian pergerakan adanya tambahan algoritma *Particle Swarm Optimization* dengan menggunakan parameter berikut.

Tabel 4. Penggunaan Parameter

No	Parameter	Nilai
1	Partikel	30
2	Iterasi	20
3	Inertia	0.7

Pada tabel 5, studi kasus 1, jarak tempuh tanpa PSO adalah 1873 mm, sedangkan dengan PSO adalah 1270 mm, sehingga terdapat pengurangan jarak sebesar 603 mm atau sekitar 32.20%. Untuk waktu, tanpa PSO diperlukan 02:33.59 (2.55998 menit), sementara dengan PSO hanya diperlukan 02:14.20 (2.23367 menit), menghemat waktu sebesar 0.317317 menit atau sekitar 12.40%.

Tabel 5. Pengujian Jarak dan Waktu Alamat 1

No	Studi kasus	Perbandingan	Jarak (mm)	Waktu
1	Tanpa PSO	Z - M - D - E - H - Z	1873	02.33.59
	PSO	Z - E - M - H - D - Z	1270	02.14.20
	Selisih		603	00.19.39

Pada tabel 6, studi kasus 2, jarak tempuh tanpa PSO adalah 1259 mm, sedangkan dengan PSO adalah 983 mm, menghasilkan pengurangan jarak sebesar 276 mm atau sekitar 21.92%. Waktu yang diperlukan tanpa PSO adalah 02:52.32 (2.87200 menit), sedangkan dengan PSO adalah 02:30.21 (2.50035 menit), menghemat waktu sebesar 0.36665 menit atau sekitar 12.77%.

Tabel 6. Pengujian Jarak dan Waktu Alamat 2

No	Studi kasus	Perbandingan	Jarak (mm)	Waktu
1	Tanpa PSO	Z - H - F - E - C - Z	1259	02.52.32
	PSO	Z - C - F - E - H - Z	954	02.30.21
	Selisih		305	00.22.11

Pada tabel 7, studi kasus 3, jarak tempuh tanpa PSO adalah 1077 mm, sedangkan dengan PSO adalah 901 mm, sehingga terdapat pengurangan jarak sebesar 176 mm atau sekitar 16.34%. Waktu yang diperlukan tanpa PSO adalah 02:30.21 (2.50535 menit), sedangkan dengan PSO adalah 02:19.27 (2.31712 menit), menghemat waktu sebesar 0.183233 menit atau sekitar 7.31%.

Tabel 7. Pengujian Jarak dan Waktu Alamat 3

No	Studi kasus	Perbandingan	Jarak (mm)	Waktu
1	Tanpa PSO	Z - A - F - E - N - Z	1077	02.30.21
	PSO	Z - E - N - F - A - Z	901	02.19.27
	Selisih		176	00.10.99

Pada tabel 8, studi kasus 4, jarak tempuh tanpa PSO adalah 911 mm, sedangkan dengan PSO adalah 901 mm, menghasilkan pengurangan jarak sebesar 10 mm atau sekitar 1.10%. Waktu yang diperlukan tanpa PSO adalah 02:17.56 (2.29267 menit), sedangkan dengan PSO adalah 02:10.19 (2.17538 menit), menghemat waktu sebesar 0.117283 menit atau sekitar 5.12%.

Tabel 8. Pengujian Jarak dan Waktu Alamat 4

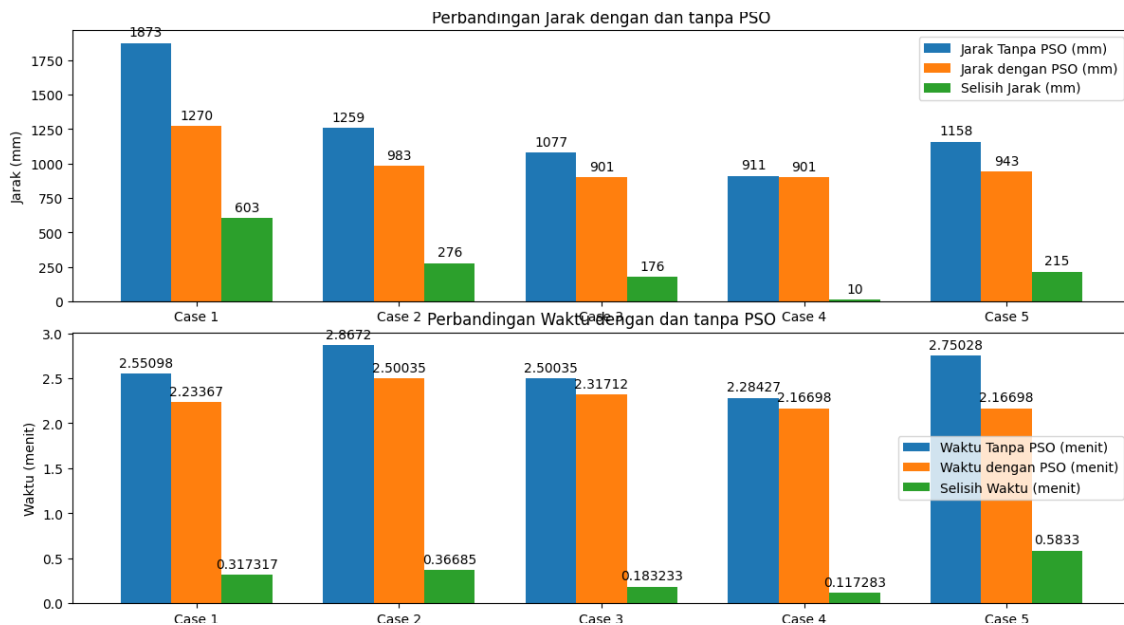
No	Studi kasus	Perbandingan	Jarak (mm)	Waktu
1	Tanpa PSO	Z - A - H - G - F - Z	911	02.17.56
	PSO	Z - H - G - F - A - Z	901	02.10.19
	Selisih		10	00.07.03

Pada tabel 9, studi kasus 5, jarak tempuh tanpa PSO adalah 1158 mm, sedangkan dengan PSO adalah 943 mm, sehingga terdapat pengurangan jarak sebesar 215 mm atau sekitar 18.56%. Waktu yang diperlukan tanpa PSO adalah 02:45.17 (2.75028 menit), sedangkan dengan PSO adalah 02:10.19 (2.16698 menit), menghemat waktu sebesar 0.5833 menit atau sekitar 21.20%.

Tabel 9. Pengujian Jarak dan Waktu Alamat 5

No	Studi kasus	Perbandingan	Jarak (mm)	Waktu
1	Tanpa PSO	Z - B - E - D - F - Z	1158	02.45.17
	PSO	Z - B - D - F - E - Z	943	02.10.19
	Selisih		215	00.21.47

Berdasarkan data data di atas dapat di akumulasikan dari segi jarak dengan adanya pso serta tanpa adanya pso, dan akumulasikan pergerakan waktu dengan adanya pso serta tanpa adanya PSO.



Gambar 7. Efisiensi Jarak dan Waktu Dengan PSO dan Tanpa PSO

Berdasarkan grafik perbandingan jarak dan waktu dengan dan tanpa PSO, terlihat bahwa penggunaan algoritma PSO secara konsisten mengurangi jarak tempuh dan waktu operasional di setiap studi kasus. Jarak tempuh dengan PSO selalu lebih pendek dibandingkan tanpa PSO, dengan persentase pengurangan terbesar pada studi kasus pertama dan terkecil pada studi kasus keempat. Waktu operasional juga menunjukkan penghematan dengan PSO di setiap studi kasus, di mana persentase pengurangan waktu terbesar tercatat pada studi kasus kelima dan terkecil pada studi kasus keempat. Secara keseluruhan, algoritma PSO terbukti meningkatkan efisiensi operasional sistem AS/RS dengan mengoptimalkan rute perjalanan, mengurangi jarak tempuh.

D. Simpulan

Berdasarkan hasil pengujian, algoritma PSO terbukti meningkatkan efisiensi pencarian jalur optimal dalam berbagai parameter dan studi kasus. Pertama, peningkatan jumlah partikel dalam PSO secara signifikan mengurangi jarak yang ditempuh, menunjukkan bahwa lebih banyak partikel dapat mengeksplorasi ruang solusi lebih baik. Kedua, peningkatan jumlah iterasi juga mengurangi jarak secara signifikan, menandakan bahwa lebih banyak iterasi memungkinkan partikel untuk menemukan rute yang lebih optimal. Ketiga, nilai inertia yang lebih rendah memberikan hasil yang lebih optimal, meskipun nilai inertia yang lebih tinggi meningkatkan stabilitas pencarian.

Pada studi kasus 1, penggunaan PSO mengurangi jarak tempuh sekitar 32% dan menghemat waktu operasional sekitar 12%. Studi kasus 2 menunjukkan pengurangan jarak sekitar 22% dan penghematan waktu sekitar 13% dengan PSO. Dalam studi kasus 3, PSO mengurangi jarak tempuh sekitar 16% dan waktu operasional sekitar 7%. Pada studi kasus 4, pengurangan jarak dengan PSO sekitar 1% dan penghematan waktu sekitar 5%. Studi kasus 5 menunjukkan pengurangan jarak sekitar 19% dan penghematan waktu sekitar 21% dengan penggunaan PSO.

Secara umum, penggunaan algoritma PSO pada sistem AS/RS sangat efisien dan berguna dalam mengurangi jarak tempuh dan waktu tempuh. Persentase pengurangan jarak berkisar antara 1% hingga 32%, sedangkan persentase pengurangan waktu berkisar antara 5% hingga 21%. Hasil ini menunjukkan bahwa PSO secara efektif meningkatkan efisiensi operasional sistem AS/RS dengan mengoptimalkan rute perjalanan dan mengurangi waktu yang dibutuhkan untuk operasi penyimpanan dan pengambilan barang. Perlu diperhatikan bahwa nilai selisih dari pengurangan jarak juga dipengaruhi oleh input alamat awal tanpa PSO, yang mungkin sudah mendekati kondisi optimal. Oleh karena itu, meskipun PSO meningkatkan efisiensi, besarnya pengurangan dapat bervariasi tergantung pada seberapa dekat input alamat awal tanpa PSO dengan rute optimal.

E. Referensi

- [1] A. Haslindah and R. Mansyur, "Pengaruh Implementasi Warehouse Management System Terhadap Inventory Control Finish Good Berbasis Barcode Pt. Dharana Inti Boga," *ILTEK*, vol. 12, 2017.
- [2] H. Wijaya, "Analisa Area Gudang Dengan Metode Kaizen Di Pt. Indah Prakasa Sentosa Tbk.Cab Cilegon," *Journal of Industrial Engineering*, vol. 4, no. 3, 2023.
- [3] F. Fadhilah, R. Firdiansyah Suryawan, L. Suryaningsih, and L. Lestari, "TEORI GUDANG DIGUNAKAN DALAM PROSES PERGUDANGAN (TINJAUAN EMPAT ASPEK)," *JTLA*, vol. 1, no. 2, pp. 153–156, Jan. 2022, doi: 10.52909/jtla.v1i2.63.
- [4] L. Noerfajr and H. Suliantoro, "Usulan Perancangan Tata Letak Gudang Dengan Menerapkansistem Management Warehousedi Pt. Sandang Asia Maju Abadi," *Industrial Engineering Online Journal*, vol. 5, no. 4, Nov. 2016.
- [5] G. G. Maulana, H. Rudiansyah, and S. N. Tazkia, "Automation Forklift System untuk Penyimpanan Produk pada Gudang Berbasis Labview," *JuTISI*, vol. 6, no. 1, Apr. 2020, doi: 10.28932/jutisi.v6i1.2334.
- [6] P. Paryanto, A. G. Reinhard, and S. A. Widyanto, "Pengembangan Prototype Smart Inventory System berbasis teknologi RFID untuk Industri Garmen," *ROTASI*, vol. 24, no. 3, 2022.
- [7] Mukhlisin and L. H. Kusumah, "Perbaikan Produktivitas Picking Order Dengan Metode Routing Heuristic Di Gudang Pusat Suku Cadang Otomotif," *Operations Excellence*, vol. 10, no. 2, pp. 164–174, 2020.
- [8] Wahyu Wiguna, "Analisis Dan Perbaikan Error Collapse Stacker Crane Automatic Warehouse Dengan Menggunakan Metode PDCA DI PT. INDAH KIAMAT PULP & PAPER SERANG,TBK," *INT P ECON DEV RES*, vol. 13, no. 1, pp. 112–129, Feb. 2020, doi: 10.46306/jbbe.v13i1.35.

- [9] F. Zammori, M. Neroni, and D. Mezzogori, "Cycle time calculation of shuttle-lift-crane automated storage and retrieval system," *IISE Transactions*, pp. 1–31, Jan. 2021, doi: 10.1080/24725854.2020.1861391.
- [10] M. Alhilal, M. S. Hajah, and A. A. Kurdianto, "Rancang Bangun Robot ASR (Automated Storage And Retrieval) 3 Axis Untuk Peminjaman Instrumen Di laboratorium Electric Drive," *Jurnal Techno Bahari*, vol. 9, no. 1, 2022.
- [11] D. Li, L. Wang, S. Geng, and B. Jiang, "Path Planning of AS/RS Based on Cost Matrix and Improved Greedy Algorithm," *Symmetry*, vol. 13, no. 8, p. 1483, Aug. 2021, doi: 10.3390/sym13081483.
- [12] F. Umam, S. Wahyuni, and H. Budiarto, "Kestabilan Kecepatan Mobile Robot pada Lintasan Mendatar, Tanjakan Serta Turunan," *Rekayasa*, vol. 12, no. 2, pp. 168–173, Oct. 2019, doi: 10.21107/rekayasa.v12i2.6396.
- [13] A. Nordeide and S. Rortveit, "The Impact of Automated Storage and Retrieval Systems on Warehouse Operations," *Molde University Collage*, p. 69, May 2021.
- [14] H. L., "Rancang Bangun Kendali Path Planning pada AS/RS dengan Rak Modular Menggunakan Metode Ant Colony Optimization," Politeknik Manufaktur Bandung, 2019.
- [15] S. Antomarioni, L. Lucantoni, F. E. Ciarapica, and M. Bevilacqua, "Data-driven decision support system for managing item allocation in an ASRS: A framework development and a case study," *Expert Systems with Applications*, vol. 185, p. 115622, Dec. 2021, doi: 10.1016/j.eswa.2021.115622.
- [16] S. I. W. Jacobus and J. S. Sumarauw, "ANALISIS SISTEM MANAJEMEN PERGUDANGAN PADA CV. PASIFIC INDAH MANADO," *Jurnal EMBA*, vol. 6 No.4, pp. 2278–2287, 2018.
- [17] H. Yilmaz and A. Tuncer, "Genetic Algorithm Based Storage and Retrieval System Optimization Considering Operational Constraints in a Multidimensional Warehouse," *International Journal of Applied Mathematics Electronics and Computers*, vol. 8, no. 4, pp. 148–153, Dec. 2020, doi: 10.18100/ijamec.802125.
- [18] J. C. Bansal, "Particle Swarm Optimization," in *Evolutionary and Swarm Intelligence Algorithms*, vol. 779, J. C. Bansal, P. K. Singh, and N. R. Pal, Eds., in *Studies in Computational Intelligence*, vol. 779, Cham: Springer International Publishing, 2019, pp. 11–23. doi: 10.1007/978-3-319-91341-4_2.
- [19] P. I. Adamu, J. T. Jegede, H. I. Okagbue, and P. E. Oguntunde, "Shortest Path Planning Algorithm – A Particle Swarm Optimization (PSO) Approach," *World Congress on Engineering*, vol. 1, 2018.