

Penerapan Algoritma Genetika pada Persoalan Pemotongan Stok Satu Dimensi

Lisa Harianto^{1✉}, M. Imran², M. D. H. Gamal³

^{1,2,3}Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Riau,
Pekanbaru, Indonesia

email: lisa.harianto6888@grad.unri.ac.id

Received 16 November 2024,

Accepted 21 Maret 2025,

Published 31 Maret 2025

Abstrak

Pemotongan stok dengan ukuran standar menjadi beberapa bagian, sesuai dengan beberapa ukuran panjang yang diinginkan oleh konsumen, menjadi suatu hal yang penting bagi sebuah perusahaan. Seringkali dalam proses pemotongan stok, terdapat sisa pemotongan, yang dikenal dengan *trim loss*, dikarenakan tidak tepatnya dalam proses pemilihan pola pemotongan. Proses menetapkan pola pemotongan stok ini disebut juga dengan *cutting stock problem* (CSP). Dalam konteks perkembangan ilmu komputer dalam menangani pemecahan masalah, alternatif solusi bisa dilakukan dengan penerapan pendekatan metaheuristik, yaitu Algoritma Genetika (AG). Algoritma ini berasal dari proses genetik yang terjadi pada makhluk hidup, di mana generasi dalam sebuah populasi secara alami berkembang secara bertahap, mengikuti prinsip seleksi alam, di mana siapa yang paling kuat dialah yang bertahan. Pada artikel ini, disajikan CSP satu dimensi yang bertujuan untuk mendapatkan pola pemotongan stok yang optimal, yaitu dengan meminimumkan *trim loss* atau ekuivalen dengan meminimumkan jumlah stok yang terpakai. Selanjutnya pendekatan AG diterapkan dengan menggunakan bantuan komputasi matlab R2021a dalam menyelesaikan CSP satu dimensi. Hasil dari penelitian ini didapat solusi dengan penggunaan sejumlah minimum stok ukuran standar yang digunakan untuk memenuhi permintaan konsumen. Setelah 20 kali percobaan, hasilnya menunjukkan bahwa nilai *fitness* terbaik yang dicapai adalah 1.000000 dengan presentase sisa pemotongan sebesar 0%.

Kata Kunci: Persoalan pemotongan stok; nilai *fitness*; algoritma genetika; metaheuristik.

Abstract

Cutting stock into several parts with specific lengths requested by customers is crucial for a company. Often, during the stock-cutting process, waste known as **trim loss** occurs due to inaccurate selection of cutting patterns. This process of determining the stock cutting pattern is referred to as the Cutting Stock Problem (CSP). In the context of advancements in computer science to address problem-solving, alternative solutions can be achieved through a metaheuristic approach, specifically the Genetic Algorithm (GA). This algorithm is inspired by the genetic processes in living organisms, where generations in a population naturally evolve over time following the principle of natural selection – survival of the fittest. This article

presents a one-dimensional CSP aiming to find an optimal cutting pattern by minimizing trim loss, which is equivalent to minimizing the amount of stock used. Subsequently, the GA approach is applied using MATLAB R2021a to solve the one-dimensional CSP. The research results show a solution with the minimum use of standard-size stock to meet customer demands. After 20 trials, the best fitness value achieved was 1.000000, with a trim loss percentage of 0%.

Keywords: Cutting Stock Problem; Fitness Value; Genetic Algorithm; Metaheuristics.

✉ Corresponding author

PENDAHULUAN

Persoalan pemotongan stok umumnya selalu berkaitan pada penggunaan bahan baku. Bahan baku memiliki peranan penting pada kegiatan produksi, sehingga perusahaan perlu mengoptimalkan penggunaannya [1]. Bahan baku yang tersedia tidak memenuhi permintaan konsumen, karena ukuran standar yang besar harus dipotong sesuai dengan permintaan konsumen [2]. Untuk memenuhi permintaan konsumen, digunakan pola pemotongan bahan baku yang dikenal dengan *Cutting Stock Problem* (CSP). Biasanya pada proses pemotongan bahan baku sering terjadi sisa pemotongan stok yang disebut dengan *trim loss*, dikarenakan dari tidak tepatnya dalam proses pemilihan pola pemotongan. Cara yang efektif untuk mengatasi hal ini adalah dengan meminimumkan *trim loss* [3]. CSP dapat diklasifikasikan berdasarkan jumlah dimensi pemotongan yaitu CSP satu dimensi, dua dimensi, dan tiga dimensi [4]. Persoalan pemotongan stok yang melibatkan pemotongan dari ukuran standar berdasarkan panjang, disebut dengan CSP satu dimensi.

Penelitian mengenai CSP pertama kali dilakukan oleh Kantorovich pada tahun 1960, yang kemudian diikuti oleh Gilmore dan Gomory [5] yang mengkaji CSP satu dimensi melalui pendekatan pemrograman linear. Mereka meneliti pola pemotongan stok satu dimensi dengan stok dalam beragam ukuran $L_1, L_2, \dots, L_k$ dan mengaplikasikan fungsi biaya, yang kemudian diselesaikan dengan metode pemrograman dinamik. Winston [6, h. 570] juga membahas CSP satu dimensi menggunakan pendekatan pemrograman linear, dengan menerapkan teknik pembangkit kolom yang kemudian diselesaikan menggunakan metode cabang dan batas. Silalahi et al. [7] juga meneliti CSP stok satu dimensi dengan tujuan meminimalkan jumlah pola berbeda agar biaya pemotongan dapat diminimumkan. Masalah ini diselesaikan menggunakan teknik pembangkit kolom yang dibantu oleh perangkat lunak LINGO.

Seiring dengan perkembangan konsep ilmu komputer untuk memecahkan masalah, solusi yang dapat diterapkan adalah menggunakan pendekatan metaheuristik, seperti Algoritma Genetika. Metode optimisasi ini, yang terinspirasi oleh proses evolusi alam, dikenal sebagai algoritma evolusioner atau Algoritma Genetika. Berdasarkan teori evolusi Charles Darwin, sifat-sifat unggul akan bertahan dan membentuk sifat baru, sementara sifat-sifat yang kurang diinginkan dievaluasi kembali untuk menghasilkan individu baru [8]. Algoritma

Genetika bekerja berdasarkan prinsip seleksi alam dan beroperasi pada seluruh populasi solusi, tidak hanya berfokus pada satu solusi saja [9].

Penelitian mengenai Algoritma Genetika yaitu pada tulisan Andika Indra Kusuma et al. [10] membahas mengenai persoalan pemotongan stok satu dimensi menggunakan algoritma Genetika pada industri mebel. Santoso dan Haerudin [11] menyelesaikan persoalan pemotongan stok satu dimensi dengan Algoritma Genetika pada kontruksi industri batang baja.

Dengan kemajuan sektor industri, tantangan dalam masalah pemotongan stok pun menjadi semakin kompleks. Operator dasar dalam algoritma Genetika terus berkembang dan beragam. Penulis tertarik untuk membahas masalah pemotongan stok satu dimensi yang bertujuan dapat meminimalkan *trim loss* dengan pendekatan metaheuristik, yaitu algoritma Genetika, menggunakan perangkat lunak MATLAB R2021a. Pada penelitian ini akan digunakan operator genetik yang berbeda dari yang digunakan dalam penelitian [10] dan [11] yang mencakup seleksi *roulette wheel*, *uniform crossover*, mutasi flip, yang tidak dibahas pada penelitian sebelumnya.

METODOLOGI

1. Mendefinisikan Formula Permasalahan

Persoalan pemotongan stok satu dimensi dalam penelitian ini bertujuan untuk meminimalkan *trim loss*. Peneliti mengilustrasikan permasalahan ini, misalkan sebuah perusahaan akan memotong papan dengan panjang 425 cm. Yang mana perusahaan memiliki papan panjang 425 cm yang harus dipotong dan diasumsikan bahwa papan yang tersedia jumlahnya tidak terbatas. Contoh pesanan konsumen dapat dilihat pada Tabel 1.

Tabel 1. Contoh pesanan

Pesanan (<i>i</i>)	Ukuran Panjang (cm)	Jumlah yang dipesan (papan)
1	75	25
2	125	20
3	225	15

Persoalan pemotongan stok dalam meminimalkan *trim loss* dapat ditulis sebagai,

$$\min z = \sum_{j=1}^n x_j ,$$

$$\text{kendala } \sum_{j=1}^n a_{ij}x_j \geq b_i, \quad i = 1, 2, \dots, m,$$

$$x_j \geq 0 \text{ dan bilangan bulat, } j = 1, 2, \dots, n,$$

dengan

x_j := ukuran papan standar yang dipotong sesuai dengan pola ke- j .

a_{ij} := jumlah permintaan untuk pesanan ke- i yang dihasilkan dari pola pemotongan ke- j .

b_i := jumlah permintaan untuk pesanan ke- i .

Berdasarkan Tabel 1, kemungkinan semua pola yang dapat dibentuk dengan tujuan untuk meminimalkan *trim loss* sebagai berikut:

Tabel 2. Kemungkinan Semua Pola

Pola (j)	Jumlah Ukuran 75 cm	Jumlah Ukuran 125 cm	Jumlah Ukuran 225 cm	Sisa (cm)
1	0	3	0	50
2	1	1	1	0
3	2	0	1	50
4	2	2	0	25
5	4	1	0	0
6	5	0	0	50

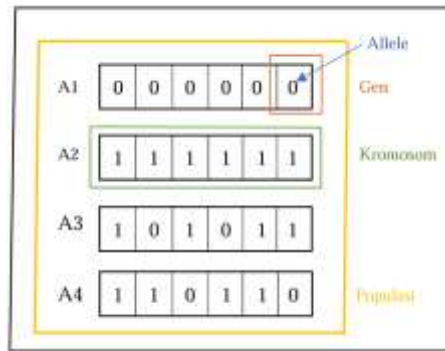
2. Algoritma Genetika

Algoritma Genetika adalah metode metaheuristik yang digunakan untuk menemukan solusi optimal dalam masalah optimisasi. Dijelaskan dalam [12] bahwa algoritma ini didasarkan pada prinsip genetika yang terjadi pada makhluk hidup, yaitu evolusi generasi dalam populasi alami mengikuti prinsip seleksi alam, di mana yang terkuat bertahan. Algoritma Genetika yaitu algoritma yang mengimplementasikan pemahaman mengenai evolusi alam dalam memecahkan sebuah masalah dengan menggunakan unsur acak namun terstruktur untuk menciptakan solusi terbaik [13].

Struktur dasar dalam algoritma Genetika meliputi [14] [15]:

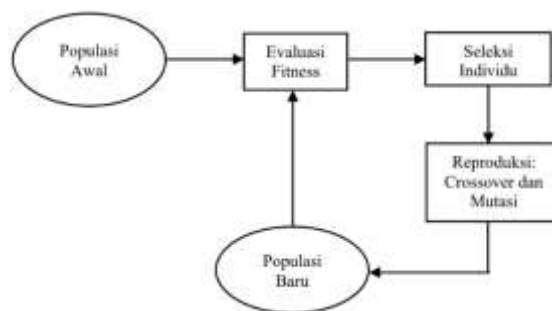
- (i) Individu, yaitu nilai atau kondisi yang mewakili solusi dari masalah yang dihadapi, sering disebut juga sebagai kromosom .
- (ii) Kromosom, yaitu kumpulan gen yang membentuk suatu nilai solusi.
- (iii) Genotipe (Gen) adalah elemen yang menyusun kromosom atau sekumpulan gen, dapat berupa nilai biner, integer, bilangan real, float, atau struktur lainnya.
- (iv) Allele, yaitu nilai yang dimiliki oleh suatu gen.
- (v) Populasi adalah kumpulan individu yang akan diproses bersama dalam satu siklus evolusi atau iterasi.
- (vi) Generasi, yaitu satu siklus dalam proses evolusi atau iterasi pada Algoritma Genetika.

Istilah-istilah di atas dapat diilustrasikan pada Gambar 1.



Gambar 1. Ilustrasi istilah-istilah Algoritma Genetika

Proses penyelesaian dalam Algoritma Genetika diperkenalkan pertama kali oleh David Goldberg [16]. Proses ini digambarkan dalam siklus yang dapat dilihat pada Gambar 2.



Gambar 2. Siklus Algoritma Genetika oleh David Goldberg

Algoritma ini dimulai dengan populasi awal yang terdiri dari kromosom atau rangkaian gen, yang dikenal sebagai individu. Setiap individu dievaluasi berdasarkan nilai *fitness* atau fungsi tujuan yang mencerminkan solusi dari masalah yang ingin dipecahkan. Individu dalam populasi ini kemudian diseleksi dan diurutkan. Selanjutnya, proses *crossover* atau kawin silang dilakukan dengan menukar gen antara dua individu. Untuk menciptakan variasi, nilai pada beberapa individu dimodifikasi melalui mutasi. Proses ini berlanjut hingga solusi terbaik ditemukan dengan mengoptimalkan nilai *fitness* setiap individu, dan siklus ini terus diulang hingga mencapai hasil yang diinginkan [17].

Proses analisis dilakukan dengan menggunakan bantuan komputasi matlab R2021a. Adapun langkah analisis yang akan dilakukan pada penelitian ini sebagai berikut:

1. Teknik Pengkodean

Tahap ini menjelaskan cara mengkodekan gen dalam sebuah kromosom, di mana gen merupakan bagian dari kromosom itu sendiri [14]. Pada penelitian ini, pengkodean gen akan dilakukan dengan menggunakan nilai biner, yaitu angka 0 dan 1.

2. Inisialisasi

Inisialisasi, atau populasi awal, adalah tahap pertama dalam Algoritma Genetika, di mana populasi diinisialisasi untuk menghasilkan sekelompok individu yang direpresentasikan oleh kromosom. Proses ini menghasilkan sejumlah individu secara acak, yang juga dikenal sebagai pembangkitan populasi awal [18]. Dalam representasi biner, populasi acak ini dapat dihasilkan menggunakan rumus

$$IPOP = \text{round}\{\text{random}(N_{ipop}, N_{bits})\},$$

IPOP adalah gen yang akan berisi pembulatan dari bilangan acak yang akan dibangkitkan sebanyak N_{ipop} (jumlah populasi) $\times$ N_{bits} (jumlah gen dalam setiap kromosom).

3. Nilai *Fitness*

Nilai *fitness* digunakan untuk menunjukkan kualitas individu, dengan menunjukkan mana yang lebih unggul [11]. Fungsi *fitness* ini harus mencerminkan tujuan dari solusi yang ingin dicapai, yaitu menentukan pola pemotongan bahan baku guna mencapai hasil optimal sambil meminimalkan *trim loss*. Dalam penelitian ini, rumus nilai *fitness* yang akan digunakan adalah

$$fitness = 1 - cost,$$

dengan rumus

$$cost = \frac{1}{m} \sum_{j=1}^m \frac{w_j}{L_j},$$

dengan w_j merupakan sisa pola ke- j , L_j adalah panjang stok pola ke- j , m adalah jumlah pola, $j = 1, 2, \dots, m$.

4. Seleksi Individu

Seleksi individu merupakan salah satu operator genetik yang penting. Seleksi adalah proses acak untuk memilih kromosom sebagai orang tua dari populasi, berdasarkan nilai *fitness* setiap kromosom [19]. Dalam proses ini, beberapa orang tua dipilih untuk melakukan *crossover*, yang menghasilkan keturunan [20]. Metode seleksi yang digunakan adalah seleksi *roulette wheel*. Berikut cara kerja metode seleksi ini [21] [22]:

(i) Menghitung total *fitness* (f)

$$f = \sum f_i,$$

dengan f_i yaitu *fitness* ke- i , dan $i = 1, 2, \dots, m$

(ii) Menghitung probabilitas *fitness* (p_i) untuk masing-masing individu

$$p_i = \frac{f_i}{f}$$

(iii) Menghitung probabilitas kumulatif (q_i) untuk masing-masing individu

$$q_i = \sum_i p_i$$

(iv) Membangkitkan bilangan acak *random* r pada rentang $[0,1]$. Apabila $r_i < q_i$ maka kromosom ke- i dijadikan sebagai induk, selai itu pilih kromosom ke- i sebagai imduk dengan syarat

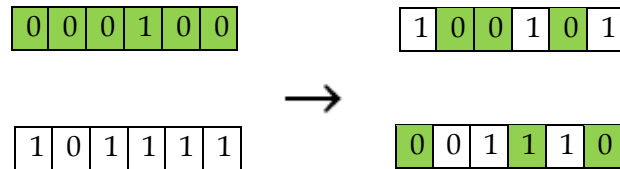
$$q_{(i-1)} \leq r_i \leq q_i, \quad i = 1, 2, \dots, m$$

dengan $q_{(i-1)}$ merupakan probabilitas kumulatif sebelum ke- i , r_i yaitu acak (*random*) ke- i , dan q_i adalah probabilitas kumulatif.

5. *Crossover*

Crossover atau persilangan adalah proses menggabungkan dua induk untuk membentuk kromosom baru. Persilangan ini dilakukan dengan menukar gen-gen yang sesuai antarinduk sehingga menghasilkan individu baru yang menggabungkan sifat terbaik dari keduanya [17]. Tidak semua kromosom akan mengalami *crossover*; hanya kromosom tertentu yang dipilih berdasarkan probabilitas yang telah ditentukan. Jika *crossover* tidak terjadi pada suatu kromosom, maka sifat dari induk akan diwariskan langsung kepada keturunannya. Proses *crossover* yang akan

digunakan adalah *uniform crossover*, seperti yang diilustrasikan pada Gambar 3. Setiap alel dipertukarkan antara sepasang kromosom yang dipilih dengan probabilitas tertentu (P_c), yang dikenal sebagai probabilitas *swapping*. Biasanya nilai probabilitas *swapping* yaitu 0,5 [12]. *Crossover* dapat terjadi apabila bilangan acak yang dihasilkan lebih kecil dari probabilitas *crossover* (P_c) [12] [23].



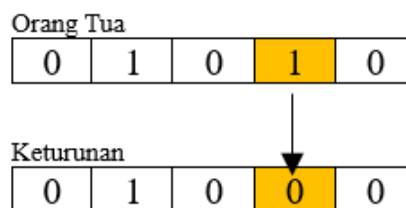
Gambar 3. Ilustrasi *uniform crossover*

6. Mutasi

Mutasi adalah proses menciptakan individu baru dengan memodifikasi gen pada individu tersebut [18]. Mutasi biner dilakukan dengan mengganti satu atau beberapa nilai gen dalam kromosom. Proses ini disebut mutasi flip, di mana nilai gen dibalik: jika gen awalnya bernilai 1, maka diubah menjadi 0, dan sebaliknya [24]. Probabilitas mutasi (P_m) biasanya berada dalam kisaran 0,001 hingga 0,1 [12]. Langkah-langkah untuk melakukan mutasi flip adalah sebagai berikut [18]:

- (i) Menghitung jumlah gen dalam populasi, dengan mengalikan ukuran populasi dengan jumlah total gennya.
- (ii) Memilih gen yang akan dimutasi secara acak berdasarkan nilai probabilitas mutasi (P_m) yang telah ditetapkan, lalu bandingkan setiap bilangan acak tersebut. Bilangan acak dihasilkan dalam rentang [0,1] sebanyak jumlah gen yang ada.
- (iii) Menentukan kromosom dari gen yang sudah dipilih untuk dilakukan mutasi.
- (iv) Mengganti nilai gen pada kromosom yang akan dimutasi, dari 0 menjadi 1 atau dari 1 menjadi 0.

Mutasi flip ditunjukkan pada Gambar 4.



Gambar 4. Ilustrasi mutasi flip

7. Penggantian Populasi

Populasi keturunan yang terbentuk melalui proses seleksi, crossover, dan mutasi akan menggantikan populasi awal induk atau disebut juga dengan populasi baru. Oleh karena itu, solusi tersebut harus dimasukkan ke dalam populasi induk untuk generasi berikutnya [12] [25] [26]. Selanjutnya setelah diperoleh hasil populasi baru tersebut, proses akan melakukan pengulangan kembali dari Langkah 3 yaitu evaluasi nilai *fitness* sampai langkah terakhir. Pengulangan akan berhenti, apabila sudah mencapai kriteria berhenti.

8. Syarat Berhenti

Pada algoritma genetika ini, kondisi penghentian yang digunakan adalah apakah jumlah maksimum generasi telah tercapai, yaitu sebanyak 200 generasi dalam 20 kali percobaan.

HASIL DAN PEMBAHASAN

Lihat kembali Tabel 1, dimisalkan P mewakili ukuran 75 cm, Q mewakili ukuran 125 cm, dan R mewakili ukuran 225 cm. Terdapat 6 pola pemotongan yang tercantum pada Tabel 2. Selanjutnya, semua pola pemotongan yang memungkinkan akan ditampilkan dalam bentuk gambar pada Tabel 3.

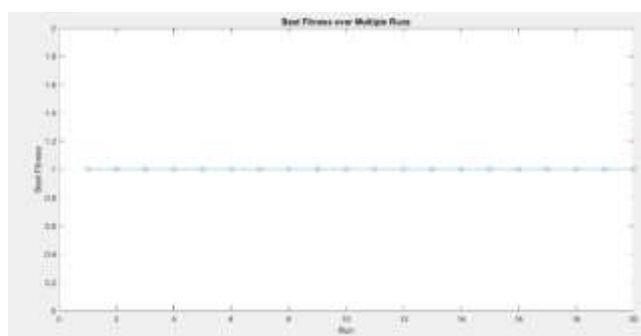
Tabel 3. Gambar Kemungkinan Semua Pola

Tabel 6. Gambar Kerangka Pola							
Pola (<i>j</i>)	Gambar						
1	<table><tr><td>Q</td><td>Q</td><td>Q</td><td></td></tr></table>	Q	Q	Q			
Q	Q	Q					
2	<table><tr><td>P</td><td>Q</td><td>R</td><td></td></tr></table>	P	Q	R			
P	Q	R					
3	<table><tr><td>P</td><td>P</td><td>R</td><td></td></tr></table>	P	P	R			
P	P	R					
4	<table><tr><td>P</td><td>P</td><td>Q</td><td>Q</td><td></td></tr></table>	P	P	Q	Q		
P	P	Q	Q				
5	<table><tr><td>P</td><td>P</td><td>P</td><td>P</td><td>Q</td><td></td></tr></table>	P	P	P	P	Q	
P	P	P	P	Q			
6	<table><tr><td>P</td><td>P</td><td>P</td><td>P</td><td>P</td><td></td></tr></table>	P	P	P	P	P	
P	P	P	P	P			

Menggunakan bantuan *software* matlab R2021a, dengan parameter yang digunakan yaitu

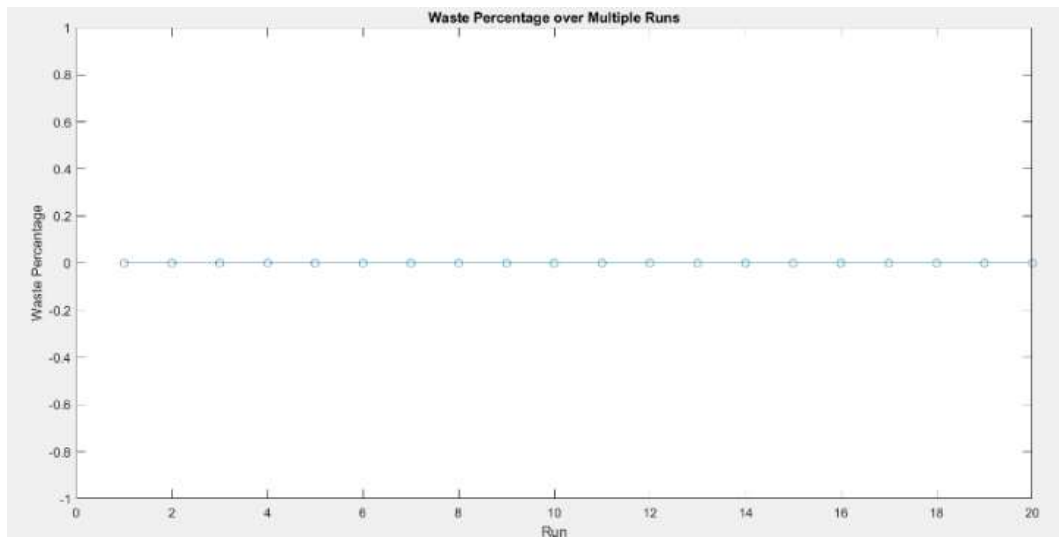
Panjang standar = 425 dalam cm
 Panjang permintaan = [75, 125, 225] dalam cm
 Banyaknya permintaan = [25, 20, 15]
 Ukuran populasi = 60
 Jumlah generasi = 200
 Probabilitas *crossover* = 0.70
 Probabilitas mutasi = 0.01

menggunakan Tabel 3, program akan dijalankan sebanyak 20 kali percobaan, setiap percobaan akan memberikan hasil yang berbeda dikarenakan angka acak yang selalu berbeda. Hasil dari setiap percobaan dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Grafik *fitness* terbaik pada setiap percobaan

Grafik pada Gambar 8 menunjukkan hasil penerapan algoritma Genetika untuk menyelesaikan masalah pemotongan stok satu dimensi. Grafik ini menggambarkan nilai *fitness* terbaik yang diperoleh pada setiap iterasi algoritma. Nilai *fitness* merupakan indikator seberapa baik solusi yang ditemukan oleh algoritma tersebut. Pada grafik, sumbu horizontal (x) mewakili jumlah percobaan (dari 1 hingga 20), sedangkan sumbu vertikal (y) menunjukkan nilai *fitness* yang terbaik pada setiap percobaan. Dari grafik tersebut, dapat dilihat bahwa nilai *fitness* terbaik tetap stabil di sekitar angka 1 pada semua percobaan.



Gambar 6. Grafik persentase sisa

Grafik pada Gambar 6 menampilkan persentase sisa yang dihasilkan dari berbagai iterasi algoritma Genetika yang diterapkan untuk menyelesaikan masalah pemotongan stok satu dimensi. Persentase sisa menunjukkan bagian papan yang tidak terpakai setelah proses pemotongan. Grafik ini memiliki sumbu horizontal (x) yang merepresentasikan jumlah percobaan (dari 1 hingga 20) dan sumbu vertikal (y) yang menunjukkan nilai persentase sisa pada setiap percobaan. Grafik tersebut menunjukkan bahwa algoritma Genetika yang diterapkan menunjukkan performa yang sangat baik dalam mencapai efisiensi dan konsistensi. Algoritma ini tidak hanya berhasil meminimalkan sisa pemotongan secara efektif (hampir tanpa sisa) tetapi juga menunjukkan kestabilan hasil di setiap percobaan. Dengan persentase sisa yang tetap di sekitar 0 pada semua percobaan, tampak bahwa solusi optimal telah tercapai, menandakan bahwa algoritma Genetika ini telah berfungsi secara optimal dalam mencapai tujuan penelitian untuk mengurangi sisa pemotongan.

Dari hasil percobaan tersebut, diperoleh kombinasi pola pemotongan yang ditampilkan pada Gambar 7.



Gambar 7. Kombinasi pola pemotongan

Pada Gambar 7, menunjukkan kombinasi pola pemotongan dengan ukuran 75 cm berwarna biru, ukuran 125 cm berwarna merah maroon, dan ukuran 225 cm berwarna coklat

SIMPULAN

Dapat disimpulkan bahwa pendekatan metaheuristik menggunakan algoritma Genetika mampu memberikan solusi yang sangat baik untuk masalah pemotongan stok satu dimensi. Setelah melakukan 20 kali percobaan, hasil yang diperoleh menunjukkan nilai *fitness* terbaik mencapai 1, yang merupakan tingkat *fitness* optimal, dengan persentase sisa pemotongan sebesar 0%. Ini menandakan bahwa tidak ada sisa papan yang terbuang selama proses pemotongan. Algoritma Genetika terbukti menjadi alat yang efektif untuk mengatasi masalah pemotongan stok. Namun, penting untuk dicatat bahwa hasil yang terlihat pada setiap grafik tidak selalu konsisten di setiap percobaan. Variasi ini disebabkan oleh elemen acak yang melekat dalam algoritma Genetika.

DAFTAR PUSTAKA

- [1] P. B. J. Bangun, S. Octarina, L. Hanum, R. Sawitri, and E. S. Cahyono, "Column generation model in capacitated multi-periods cutting stock problem with pattern set-up cost," *Sci. Technol. Indones.*, vol. 6, no. 1, pp. 8–13, Jan. 2021.
- [2] S. Octarina, D. Septimianti, and E. Yuliza, "Implementation of arc flow model incapacitated multi-period cutting stock problem with the pattern set up cost to minimize the trim loss," *J. Phys. Conf. Ser.*, pp. 1–6, July 23–24, 2021.
- [3] S. Octarina, D. G. Juita, N. Eliyati, and P. B. J. Bangun, "Set covering model in solving multiple cutting stock problem," *Sci. Technol. Indones.*, vol. 5, no. 4, pp. 121–130, Oct. 2020.
- [4] J. Fang, Y. Rao, Q. Luo, and J. Xu, "Solving one-dimensional cutting stock problems with the deep reinforcement learning," *Mathematics*, vol. 11, no. 4, pp. 1–16, Feb. 2023.
- [5] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Inst. Oper. Res. Manag. Sci.*, vol. 9, no. 6, pp. 849–859, Jan. 1961.
- [6] W. L. Winston, *Operations Research Applications and Algorithms*, Fourth Edition. USA: Thomson Learning, 2004.
- [7] B. P. Silalahi, F. Hanum, F. Setyawan and P. T. Supriyo, "One dimensional cutting stock problem that minimizes the number of different patterns," *Barekeng: Journal of Mathematics and Its Application.*, vol. 16, no. 3, pp. 805–814, Sept. 2022.
- [8] J. Bagherinejad and M. Shoeib, "Dynamic capacitated maximal covering location problem by considering dynamic capacity," *Int. J. Ind. Eng. Comput.*, vol. 9, no. 2, pp. 249–264, March 2018.
- [9] M. Herda, "Parallel genetic algorithm for capacitated p-median problem," *Procedia Eng.*, pp. 313–317, May 31–June 2, 2017.
- [10] A. I. Kusuma, A. W. Widodo and M.A. Fauzi, "optimasi sisa bahan baku pada industri mebel menggunakan algoritma genetika," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 3, pp. 1119–1125, March 2018.
- [11] B. Santoso and H. Haerudin, "Genetic algorithm approach for cutting stock problems

- in construction industries,” *Int. J. Integr. Sci.*, vol. 2, no. 7, pp. 957–970, July 2023.
- [12] E. K. Burke and G. Kendall, *Introduction Tutorials in Optimization and Decision Support Techniques*. New York: Springer, 2005.
- [13] S. Kartolo and C. Natalia, “Website design optimization of cutting stock problem on 1 and 2 dimentions by using genetic algorithm,” *Jurnal IPTEK Media Komunikasi Teknologi*, vol. 24, no. 2, pp. 85–94, Dec. 2020.
- [14] H. A. Taha, *Operations Research an Introduction , Tenth Edition*. Harlow: Pearson Education Hall, Newark, 2017.
- [15] A. Kusnadi and D. S. Santoso, “Implementasi algoritma genetika pada penempatan tugas asisten laboratorium berbasis web (studi kasus: LAB ICT universitas multimedia nusantara),” *ULTIMACS*, vol. 7, no. 2, pp. 139–147, 2015.
- [16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Canada: Addison-Wesley Publishing Company, INC, 1989.
- [17] A. Mehdary, A. Chehri, A. Jakimi, and R. Saadane, “Hyperparameter optimization with genetic algorithms and xgboost: a step forward in smart grid fraud detection,” *Sensors*, vol. 24, no. 4, pp. 1–24, Feb. 2024.
- [18] T. F. D. Aditama and Azhari. SN, “Determining community structure and modularity in social network using genetic algorithm,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 3, pp. 219–230, July 2020.
- [19] G. Valerian, T. Sutrisno and D. E. Herwindianti, “Image clustering using genetic algorithm with tournament selection and uniform crossover,” *IOP Conferences Series: Materials Science and Engineering*, 852 (2020), 012043.
- [20] Z. Michalewicz, *Genetic algorithms + Data Structures = Evolution Programs*, Third Edition. New York: Springer, 1995.
- [21] M. Devilawanti, S. H. Sitorus and Ilhamsyah, “Sistem seleksi calon pejabat tinggi pratama dengan menggunakan algoritma genetika,” *Coding Jurnal Komputer dan Aplikasi*, vol. 07, no. 03, pp. 75–84, Dec. 2019.
- [22] A. Tohari and Y. P. Astuti, “Penerapan algoritma genetika dalam menentukan rute terpendek pt. pos cabang lamongan,” *MATH UNESA Jurnal Ilmiah Matematika*, vol. 11, no. 03, pp. 458–467, 2023.
- [23] N. W. S. Saraswati, I. D. M. D. Dwiastuti and I. G. A. A. D. Indradewi, “Modified genetic algorithm for employee work shifts schedulling optimization”, *Journal of Physics: Conferences Series*, 1810 (2021),012014.
- [24] A. Syarif, Aristoteles, A. Dwiastuti and R. Malinda, “Performance evaluation of various genetic algorithm approaches for knapsack problem”, *ARPN Journal of Engineering and Applied Scinces*, 11 (2016), 4713 - 4719.
- [25] M. Affenzeller, S. Winkler, S. Wagner and A. Beham, *Genetic Algorithms and Genetic Programming Modern Concepts and Practical Applications*, CRC Press, New York, 2009.

- [26] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms, Second Edition*, A John Wiley & Sons, INC., Publication, Canada, 2004.