

---

## Representasi *Chromosome Gray Code* Algoritma Genetika pada *job shop scheduling problem*

A. Amirul Asnan Cirua<sup>1</sup>, Sugiarto Cokrowibowo<sup>2</sup>, Rahmawati<sup>3</sup>, Rahmatia<sup>4</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Universitas Sulawesi Barat

E-mail: <sup>1</sup>amirulasnancirua@unsulbar.ac.id, <sup>2</sup>sugiarto.cokrowibowo@unsulbar.ac.id,

<sup>3</sup>rhmawthy.amma18024@gmail.com, <sup>4</sup>rahmatia002@gmail.com

### **Abstrak**

Penjadwalan mesin untuk setiap job memiliki data mesin dan waktu sehingga penyusunan jadwal untuk mendapatkan makespan terkecil penting untuk dilakukan. Penjadwalan ini bertipe NP-Hard dan berkaitan dengan industri manufaktur yang bertipe make to order. Kombinasi solusi penjadwalan yang muncul membutuhkan metode yang dapat menentukan susunan penjadwalan terbaik. Algoritma genetika dengan representasi kromosom gray code diusulkan pada penelitian ini. Model gray code dikembangkan dengan tujuan mendapatkan rata-rata solusi optimasi yang lebih baik. Gray code dapat mengurangi hamming cliff pada saat proses komputasi dilakukan. Pengujian parameter mutation rate 0.2, 0.5, dan 0.8 pada GA menunjukkan hasil makespan pada mutation rate 0.2 lebih kecil. Pengujian pada dataset Job Shop dengan membandingkan algoritma Genetic Algorithm (GA) tanpa gray code, Penambahan metode gray code terhadap algoritma genetika memberikan rata-rata makespan yang lebih kecil tetapi meningkatkan waktu komputasi.

**Kata Kunci :** Penjadwalan, Job-Shop, Algoritma Genetika, Gray code, Makespan.

### **Abstract**

Machine scheduling for each job has machine and time data so it is important to prepare a schedule to get the smallest makespan. This scheduling is of the NP-Hard type and is related to the manufacturing industry which is of the make to order type. The combination of scheduling solutions that emerge requires a method that can determine the best scheduling arrangement. A genetic algorithm with gray code chromosome representation is proposed in this research. The gray code model was developed with the aim of getting a better average optimization solution. Gray code can reduce hamming cliffs during the computing process. Testing mutation rate parameters 0.2, 0.5, and 0.8 on GA shows that the makespan results at mutation rate 0.2 are smaller. Testing on the Job Shop dataset by comparing the Genetic Algorithm (GA) algorithm without gray code. The addition of the gray code method to the genetic algorithm provides a smaller average makespan but increases computing time..

**Keywords—** Scheduling, Job-Shop, Genetic Algorithm, Gray code, Makespan

## 1. PENDAHULUAN

Penjadwalan tugas atau pekerjaan dengan sumber daya pada saat ini merupakan hal yang menantang dan menarik untuk diteliti, tingkat tantangan masalah yang semakin kompleks

diberikan menyebabkan proses penjadwalan menjadi rumit apabila menggunakan pendekatan konvensional dan sekuensial. Penjadwalan dengan pendekatan tersebut menjadi tidak efektif karena membutuhkan resource yang besar, baik dari segi waktu dan biaya. Seiring pertambahan waktu maka alokasi sumber daya akan semakin besar sehingga untuk masalah penjadwalan pekerjaan atau job termasuk dalam kriteria pengoptimalan dan termasuk dalam permasalahan NP-Hard [1]. Pada perusahaan atau industri di seluruh dunia, dapat dilihat bahwa jenis produksi yang umum adalah jenis job shop sebanyak 75%. Teknologi canggih berkembang dalam dunia industri tetapi tidak merubah permintaan pelanggan sehingga untuk jenis job shop menjadi suatu masalah pengoptimalan dalam proses penjadwalan bagi perusahaan disebabkan oleh keragaman dan perubahan produk yang tinggi [2]. Aspek fleksibilitas dan efisiensi tetapi mengurangi biaya produksi menjadi tantangan utama pada masalah penjadwalan produksi itu sendiri [3]. Penjadwalan dalam industri terbagi menjadi dua jenis, *Flow Shop* dan *Job Shop*. Pada penjadwalan *Flow shop*, semua job akan melewati mesin yang sama dan urutan yang sama pula, sedangkan *Job shop*, setiap job melewati mesin yang berbeda. Ada banyak cara dilakukan untuk menyelesaikan penjadwalan *job shop* seperti *neural networks*, *artificial intelligence*, *neighborhood search techniques*, kemudian algoritma meta-heuristik berbasis kecerdasan kelompok sosial mampu menyelesaikan penjadwalan job shop dengan waktu yang cukup singkat, seperti algoritma *Particle Swarm Optimization*, *Genetic Algorithm*, *Simulated Annealing* [4]. Penelitian sebelumnya membuktikan bahwa tidak ada algoritma yang secara pasti dapat memecahkan permasalahan *job shop* secara konsisten dan bahkan ukuran permasalahan yang kecil [1].

Algoritma meta-heuristik yang terinspirasi dari spesies biologis menjadi sangat populer untuk beberapa tahun terakhir. Kecerdasan kolektif beberapa makhluk hidup seperti semut, lebah, tawon, ikan telah dikembangkan menjadi beberapa algoritma *Swarm Intelligence* (SI) [5]. Beberapa penelitian melakukan perbandingan kinerja dari algoritma dalam pemecahan masalah optimasi, algoritma genetika menjadi metode yang paling populer, pada penelitian dilakukan [6] pada kasus optimalisasi aliran daya dan *economic dispatch* dengan menggunakan algoritma genetika *gray code* dan menunjukkan hasil yang lebih baik dari pada algoritma genetika biner.

Algoritma genetika merupakan algoritma pencarian dan model ilmiah evolusi alami yang diangkat dari mekanisme seleksi alam oleh Darwin. Ide dasar dari algoritma genetika adalah untuk mensimulasikan evolusi populasi individu yang mewakili masalah pencarian tertentu, mempromosikan kelangsungan hidup dan reproduksi yang paling cocok. Algoritma genetika merupakan algoritma pencarian tujuan umum serta sebagai algoritma optimasi dan model ilmiah evolusi yang sering digunakan diberbagai bidang sains dan teknik [7].

Berdasarkan pada studi literatur, model *gray code* diterapkan pada representasi kromosom pada individu dari algoritma genetika, model ini mengurangi kesalahan pembacaan bit pada proses komputasi terutama pada fase evolusi pada algoritma genetika. *gray code* memiliki struktur bit yang hanya memiliki tepat satu perbedaan bit pada dua nilai yang berdekatan tersebut.

## 2. METODE

### 2.1 Job Shop Scheduling

Proses penjadwalan terdapat tiga elemen yaitu produksi, sumber daya, dan waktu, kemudian dituntut untuk menentukan mana, kapan, dan bagaimana sumber daya digunakan untuk menyelesaikan pekerjaan. Permasalahan job shop memiliki struktur yang kompleks dalam hal jumlah kriteria. Tidak terbatas dalam jumlah proses dan ada banyak jenis penjadwalan alternatif. Pada penjadwalan job shop, setiap pesanan atau pekerjaan yang diproses pada mesin yang berbeda memiliki urutan prosesnya masing-masing. Penyusunan jadwal berdasarkan urutan proses tersebut harus mempertimbangkan kendala lainnya untuk mendapatkan susunan jadwal terbaik dengan waktu yang minimal [2]. Untuk menghasilkan jadwal yang mencakup seluruh job yang

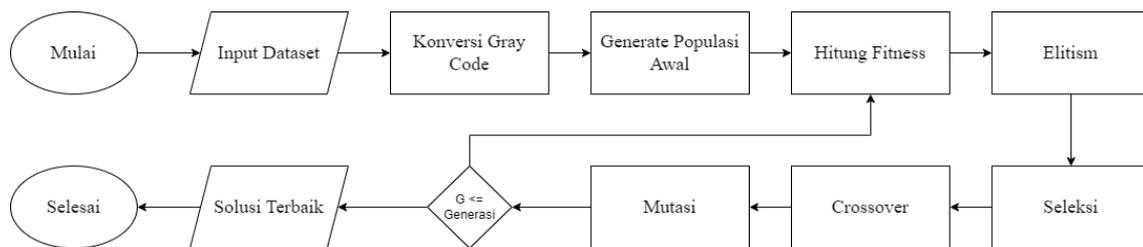
akan dikerjakan perlu memperhatikan karakteristik dari job shop itu sendiri. Masalah penjadwalan job shop dapat didefinisikan sebagai berikut [8]:

- Terdapat n Job,  $J = \{ Job1, Job2, Job3, \dots, Jobn \}$
- Terdapat m Machine,  $M = \{ M1, M2, M3, \dots, Mm \}$
- Setiap pekerjaan terdapat beberapa operasi,  $O = \{ O1, O2, O3, \dots, Oi \}$
- Setiap operasi harus diproses ketika operasi sebelumnya telah selesai
- Total waktu disebut makespan dan berada dalam satu satuan waktu
- Setiap job bersifat independen

Defenisi dataset job shop termasuk dalam data diskrit yaitu data yang dapat dihitung dan nilainya adalah bilangan bulat. Informasi pada data diskrit berisi sejumlah nilai yang tidak dapat dipecah..

## 2.2 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian yang berdasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi. Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi. Berikut adalah flowchart algoritma genetika.



Gambar 1. Algoritma Genetika

### 2. 2.1 Populasi

Inisialisasi populasi merupakan tahap pembentukan populasi yang dibentuk dari sekumpulan individu secara acak. Individu bisa dikatakan sama dengan kromosom yang merupakan kumpulan dari gen. Tahapan ini merupakan tahapan awal yang paling penting dalam algoritma genetika agar menghasilkan solusi yang optimal.

### 2. 2.2 Fungsi Fitness

kromosom akan dievaluasi menggunakan fungsi *fitness*. fungsi ini sebagai ukuran dari performansi atau kualitas solusi yang diberikan. Proses evaluasi fungsi *fitness* ini akan terus berjalan sampai terpenuhinya kriteria. Pada penelitian ini, fungsi fitness didapatkan dari hasil durasi. Dimana menggunakan rumus :

$$fitness = \frac{1}{durasi\_jadwal} \quad (1)$$

### 2. 2.3 Seleksi

Seleksi merupakan proses pemilihan individu/kromosom yang terbaik dari generasi lama untuk dijadikan *parent* yang akan saling di-crossover (kawin silang) untuk membentuk individu baru pada generasi selanjutnya. Berdasarkan teori evolusi Darwin, kromosom yang terbaik seharusnya dapat bertahan hidup dan membentuk keturunan baru (offspring).

### 2. 2.4 Crossover

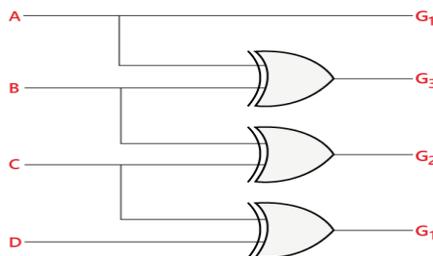
*Crossover* atau perkawinan silang merupakan fase dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Perkawinan silang dilakukan dengan melakukan pertukaran gen dari dua induk secara acak.

### 2. 2.5 Mutasi

Mutasi merupakan proses mengganti atau mengubah nilai dari beberapa gen dalam suatu kromosom. Mutasi ini berperan untuk memberikan model solusi baru berdasarkan proses yang dialami pada evolusi.

### 2. 3 Gray Code

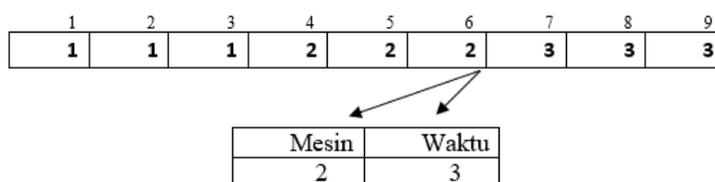
*Gray code* atau kode abu-abu dikenal sebagai *reflected binary code*, nilai biner yang berdekatan hanya tepat memiliki satu digit yang berbeda. *Gray code* memiliki hamming cliff yang rendah karena hanya memiliki satu bit yang berbeda untuk nilai yang berdekatan [9]. Penelitian yang dilakukan oleh [10] Pada algoritma genetika, sering terjadi konvergensi prematur, *gray code* dan *real code* diimplementasikan pada proses mutasi dan persilangan. *Gray code* menghasilkan kemampuan pencarian solusi yang lebih besar serta mempertahankan solusi terbaik. Konversi nilai yang akan diproses dilakukan perubahan kedalam bentuk *gray code* dengan menggunakan gerbang logika XOR atau exclusive OR yaitu operator logika yang dimana akan menghasilkan nilai logika '1' jika hanya input yang diberikan berbeda, sedangkan apabila input bernilai sama maka akan menghasilkan output nilai berlogika '0', berikut operator konversi biner ke *gray code*. Berikut adalah sirkuit logika untuk konversi biner ke *gray code*.



Gambar 2. Gerbang Logika konversi biner ke *gray code*

## 3. HASIL DAN PEMBAHASAN

Penjadwalan dilakukan untuk mencari makespan. Makespan didapatkan setelah melakukan ilustrasi pada diagram gantt. Berikut adalah contoh dataset untuk penjadwalan mesin. Pemodelan dilakukan dalam bentuk array 1 dimensi. Array tersebut memuat job job tiap proses. Dapat dilihat pada gambar berikut.



Gambar 3. Model Representasi Individu

Pada solusi di atas terlihat urutan job teratur, index dimulai dari 1 sehingga terlihat ada 9 index proses pada satu permasalahan. Index 1 = job 1 proses 1, index 2 = job 1 proses 2, index 3 = job 1 proses 3, index 4 = job 2 proses 1, index 5 = job 2 proses 2, index 6 = job 2 proses 3, index 7 = job 3 proses 1, index 8 = job 3 proses 2, index 9 = job 3 proses 3. Urutan index menjadi data

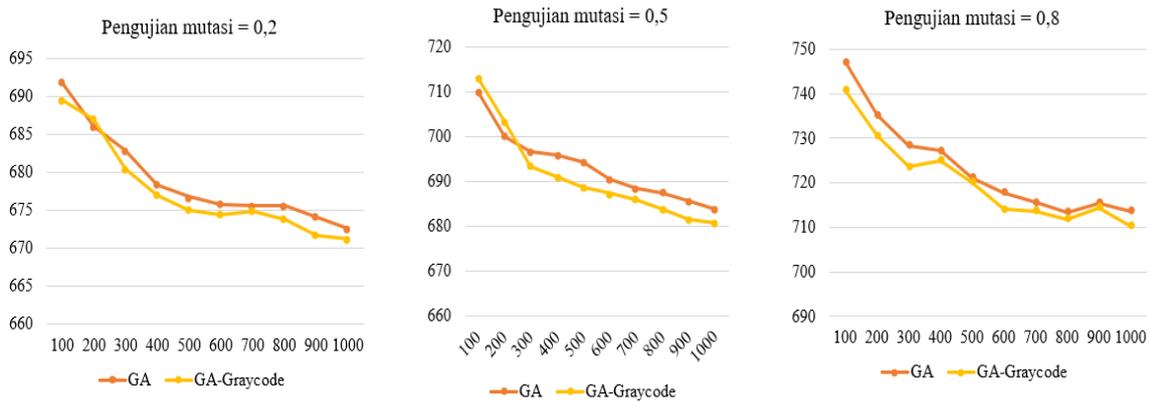
proses dan setiap job tersebut memiliki data mesin dan waktu. Sedangkan model solusi yang

1	2	3	4	5	6	7	8	9
0011	0010	0011	0010	0001	0001	0001	0011	0010

diusulkan yaitu menggunakan gray code sebagai berikut.

Gambar 4. Model Representasi Individu gray code

pengujian dilakukan dengan membandingkan algoritma genetika dan algoritma genetika gray code, serta menguji parameter mutasi 0.2, 0.5, dan 0.8. dapat dilihat pada grafik berikut.



Gambar 5. Grafik Pengujian Parameter Mutasi

Pengujian dengan parameter mutasi 0,2, makespan pada populasi 100 hingga 1000 hasil yang diberikan setiap model algoritma cukup baik, algoritma GA-GrayCode konsisten memberikan hasil yang lebih baik dibandingkan dengan algoritma genetika terlihat pada makespan dari populasi 100 hingga 1000. Semakin kecil rate maka peluang semakin besar untuk memodifikasi individu solusi. Rata-rata makespan yang didapatkan memiliki perbedaan yang tidak terlalu signifikan dari tabel sebelumnya yaitu pada rate 0,2. Sedangkan rata-rata makespan yang didapatkan tersebut cukup besar dibandingkan dengan tabel rate 0,2 dan 0,5. Penyebab dari perbedaan yang cukup besar tersebut karena pada rate 0,8 peluang individu untuk masuk ke dalam fase mutasi semakin kecil.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan terhadap algoritma Genetika dengan kombinasi Gray Code dalam menyelesaikan masalah penjadwalan mesin menunjukkan bahwa Representasi Kromosom dalam bentuk Gray Code memberikan hasil optimasi makespan (durasi penjadwalan) lebih kecil dibandingkan algoritma genetika tanpa gray code. Pengaruh dari kombinasi gray code mengurangi masalah pada kesalahan pembacaan bit saat proses komputasi. namun meningkatkan waktu komputasi karena program terlebih dahulu melakukan konversi ke dalam model gray code. Pengaruh parameter mutasi juga menunjukkan bahwa semakin besar nilai mutasi maka kemampuan pencarian solusi menjadi semakin kecil, disebabkan oleh peluang setiap individu memasuki proses mutasi semakin kecil.

#### REFERENSI

- [1] Anshulika and L. A. Bewoor, "A genetic algorithm approach for solving a job shop scheduling problem," 2017, doi: 10.1109/ICCCI.2017.8117720.

- [2] Y. Atay and H. Kodaz, "Optimization of job shop scheduling problems using modified clonal selection algorithm," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 22, no. 6, pp. 1528–1539, 2014, doi: 10.3906/elk-1212-26.
- [3] X. Zhang, D. Zhang, and X. Hu, "A Novel Optimization Algorithm for Job-Shop Production Scheduling," 2021 IEEE 6th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2021, pp. 183–189, 2021, doi: 10.1109/ICCCBDA51879.2021.9442504.
- [4] Y. Zhang, J. Jin, and S. Yin, "Hybrid frog-leap algorithm for job shop scheduling," in *Proceedings of the 14th IEEE Conference on Industrial Electronics and Applications, ICIEA 2019*, 2019, no. 5133403, pp. 429–432, doi: 10.1109/ICIEA.2019.8834138.]
- [5] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete Spider Monkey Optimization for Travelling Salesman Problem," *Appl. Soft Comput. J.*, vol. 86, Jan. 2020, doi: 10.1016/j.asoc.2019.105887.
- [6] Y. dongzhi and Y. W. Ye lvshan, "Artificial Bee Colony Algorithm with Genetic Algorithm for Job Shop Scheduling Problem," in *2017 International Symposium on Intelligent Signal Processing and Communication Systems*, 2017, pp. 433–438.
- [7] Aswandi, S. Cokrowibowo, and A. Irianti, "Model Penentuan Rute Terpendek Penjemputan Sampah Menggunakan Metode MTSP dan Algoritma Genetika," *J. Appl. Comput. Sci. Technol.*, vol. 2, no. 1, pp. 43–48, 2021, doi: 10.52158/jacost.v2i1.168.
- [8] S. N. Mhasawade and L. A. Bewoor, "A survey of hybrid metaheuristics to minimize makespan of job shop scheduling problem," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, 2018, pp. 1957–1960, doi: 10.1109/ICECDS.2017.8389792.
- [9] K. Nagata and F. Nemenzo, "Some Properties of Binary Gray Code," *Proc. - 2015 Int. Conf. Comput. Appl. Technol. CCATS 2015*, pp. 72–75, 2016, doi: 10.1109/CCATS.2015.27.
- [10] Z. J. Yao, Q. H. Meng, G. W. Li, and H. Y. Peng, "Multi-objective chaotic optimization algorithm by combining gray and real codes," *5th Int. Conf. Nat. Comput. ICNC 2009*, vol. 4, pp. 608–612, 2009, doi: 10.1109/ICNC.2009.423.