

**OPTIMALISASI PENJADWALAN JOB SHOP MENGGUNAKAN
ALGORITMA GENETIKA DAN ALGORITMA
SIMULATED ANNEALING**

Skripsi

Oleh

**SYAHREZA
NPM. 2117031087**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRACT

OPTIMIZATION OF JOB SHOP SCHEDULING USING GENETIC ALGORITHM AND SIMULATED ANNEALING ALGORITHM

By

Syahreza

Job shop scheduling is one of the most complex problems in the manufacturing industry and is classified as an NP-hard problem due to the numerous variables and constraints involved in allocating tasks to machines. This study aims to optimize job shop scheduling in the production process of the Tofu and Oncom Factory "Topo Sejahtera" in Metro, Lampung, by applying two metaheuristic algorithm approaches: Genetic Algorithm (GA) and Simulated Annealing (SA). The Genetic Algorithm operates based on the principles of natural selection through crossover and mutation processes, while Simulated Annealing mimics the physical process of metal cooling to find optimal solutions by exploring the solution space probabilistically. The research involves processing job duration data across seven key production machines. The implementation and analysis results show that both methods can provide efficient scheduling solutions, with comparative results indicating different performance levels in minimizing the makespan. The results of this study indicate that both the Genetic Algorithm and the Simulated Annealing Algorithm can be effectively used to solve complex Job Shop Scheduling Problems (JSSP), achieving the same optimal makespan of 448 minutes. In both the manual and programming approaches, the best result is obtained when the black soybean (Job2) is scheduled first and the anjasmore soybean (Job5) is scheduled last.

Keywords: job shop scheduling, genetic algorithm, simulated annealing, optimization, makespan.

ABSTRAK

OPTIMALISASI PENJADWALAN JOB SHOP MENGGUNAKAN ALGORITMA GENETIKA DAN ALGORITMA SIMULATED ANNEALING

Oleh

Syahreza

Penjadwalan job shop merupakan salah satu masalah kompleks dalam dunia manufaktur yang dikategorikan sebagai masalah NP-hard, karena melibatkan banyak variabel dan kendala dalam pengalokasian pekerjaan ke mesin. Penelitian ini bertujuan untuk mengoptimalkan penjadwalan job shop pada proses produksi di Pabrik Tahu dan Oncom Topo Sejahtera Metro, Lampung, dengan menggunakan dua pendekatan algoritma metaheuristik, yaitu Algoritma Genetika (AG) dan Simulated Annealing (SA). Algoritma Genetika bekerja berdasarkan prinsip seleksi alam dengan operasi crossover dan mutasi, sedangkan Simulated Annealing meniru proses pendinginan logam untuk menemukan solusi optimal dengan menjelajahi ruang solusi secara probabilistik. Penelitian dilakukan dengan pengolahan data waktu proses setiap job pada tujuh mesin utama produksi tahu. Hasil implementasi dan analisis menunjukkan bahwa kedua metode mampu memberikan solusi penjadwalan yang efisien, dengan perbandingan hasil yang memperlihatkan perbedaan performa antara keduanya dalam meminimalkan makespan. Hasil dari penelitian ini menunjukkan bahwa Algoritma Genetika dan Algoritma *Simulated Annealing* dapat digunakan untuk menyelesaikan *Job Shop Scheduling Problem* (JSSP) yang kompleks serta sama optimal yaitu pada angka 448 menit. Pada penelitian ini menggunakan cara manual maupun pemrograman hasil terbaik akan muncul jika kedelai hitam (*Job2*) dikerjakan awal dan kedelai anjasmoro (*Job5*) dikerjakan diakhir.

Kata-kata kunci: *job shop scheduling*, algoritma genetika, *simulated annealing*, optimasi, *makespan*.

**OPTIMALISASI PENJADWALAN JOB SHOP MENGGUNAKAN
ALGORITMA GENETIKA DAN ALGORITMA
SIMULATED ANNEALING**

SYAHREZA

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

Judul Skripsi : **OPTIMALISASI PENJADWALAN *JOB SHOP* MENGGUNAKAN ALGORITMA GENETIKA DAN ALGORITMA *SIMULATED ANNEALING***

Nama Mahasiswa : **Syahreza**

Nomor Pokok Mahasiswa : **2117031087**

Program Studi : **Matematika**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

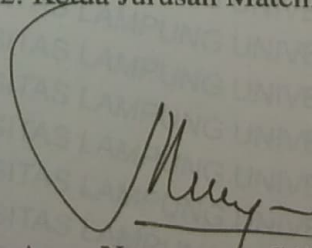


Dr. Notiragayu, S.Si., M.Si.
NIP 197311092000122001



Dr. Muslim Ansori, S.Si., M.Si.
NIP 197202271998021001

2. Ketua Jurusan Matematika

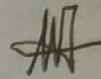


Dr. Aang Nuryaman, S.Si., M.Si.
NIP. 197403162005011001

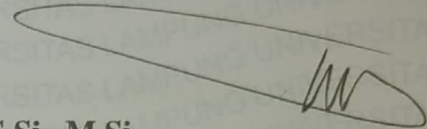
MENGESAHKAN

1. Tim penguji

Ketua : **Dr. Notiragayu, S.Si., M.Si.**

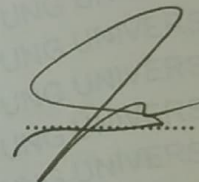


Sekretaris : **Dr. Muslim Ansori, S.Si., M.Si.**



Penguji

Bukan Pembimbing : **Dr. Agus Sutrisno, S.Si., M.Si.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **23 Juli 2025**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : Syahreza
Nomor Pokok Mahasiswa : 2117031087
Jurusan : Matematika
Judul Skripsi : **OPTIMALISASI PENJADWALAN *JOB SHOP* MENGGUNAKAN ALGORITMA GENETIKA DAN ALGORITMA *SIMULATED ANNEALING***

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 23 Juli 2025



Syahreza

RIWAYAT HIDUP

Penulis memiliki nama lengkap Syahreza yang lahir di Metro pada tanggal 12 Februari 2002. Penulis lahir dari pasangan suami istri Bapak Thamrin dan Ibu Ana. Penulis berkebangsaan Indonesia dan beragama Islam. Kini penulis beralamat di Kecamatan Metro Timur, Metro, Lampung.

Penulis menyelesaikan pendidikan di SDN 1 Metro Pusat pada 2014. Kemudian penulis melanjutkan pendidikan di SMPN 4 Metro dan lulus tahun 2017. Penulis melanjutkan pendidikan di SMAN 4 Metro dan mengikuti Ekstrakurikuler yaitu Basket dan Rohis dan lulus pada tahun 2020.

Penulis kemudian melanjutkan studi ke Universitas Lampung jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam. Penulis mengikuti beberapa himpunan dan salah satunya adalah HIMATIKA (Himpunan Mahasiswa Matematika). Selanjutnya mulai tahun 2021 sampai dengan penulisan skripsi ini, penulis masih terdaftar sebagai mahasiswa program S1 Matematika FMIPA Universitas Lampung.

KATA INSPIRASI

”Jangan manja, bergeraklah jika masih bernafas”
(Syahreza)

”Ketika dunia jahat kepadamu, maka berusahalah untuk menghadapinya, karena tidak ada orang yang membantumu jika kau tidak berusaha”
(Roronoa Zoro, One Piece)

”Berhentilah bicara tentang impianmu, mulailah bekerja untuk mencapainya.”
(Eric Thomas)

”Jangan takut untuk gagal. Takutlah untuk tidak mencoba.”
(Roy T. Bennett)

”Jadilah manusia yang lebih baik, bukan untuk menjadi terlihat baik”
(Syahreza)

PERSEMBAHAN

Dengan mengucapkan Alhamdulillah dan syukur kepada Allah SWT atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan rasa syukur dan Bahagia, saya persembahkan rasa terimakasih saya kepada:

Ayah dan Ibuku Tercinta

Terimakasih kepada orang tuaku atas segala pengorbanan, motivasi, doa dan ridho serta dukungannya selama ini. Terimakasih telah memberikan pelajaran berharga kepada anakmu ini tentang makna perjalanan hidup yang sebenarnya sehingga kelak bisa menjadi orang yang bermanfaat bagi banyak orang.

Dosen Pembimbing dan Pembahas

Terimakasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga.

Sahabat-sahabatku

Terimakasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasinya, serta doa-doanya dan senantiasa memberikan dukungan dalam hal apapun.

Almamater Tercinta

Universitas Lampung

SANWACANA

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "Optimalisasi Penjadwalan *Job Shop* Menggunakan Algoritma Genetika dan Algoritma *Simulated Annealing*" dengan baik dan lancar serta tepat pada waktu yang telah ditentukan. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW.

Dalam proses penyusunan skripsi ini, banyak pihak yang telah membantu memberikan bimbingan, dukungan, arahan, motivasi serta saran sehingga skripsi ini dapat terselesaikan. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Notiragayu, S.Si., M.Si. selaku Pembimbing 1 dan selaku dosen pembimbing akademik yang telah banyak meluangkan waktunya untuk memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Dr. Muslim Ansori, S.Si., M.Si. selaku Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Dr. Agus Sutrisno, S.Si., M.Si. selaku Penguji yang telah bersedia memberikan kritik dan saran serta evaluasi kepada penulis sehingga dapat menjadi lebih baik lagi.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Seluruh dosen, staff dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

6. Orang tua tercinta yang telah memberikan dorongan secara moral maupun spiritual kepada penulis.
7. Zainal, Nanda, Jumi, Eri, Ridho, Jonathan, Alfin, Thomi, Rena, Miranda, dan David yang selalu memberi bantuan , dukungan, serta motivasi dalam menyelesaikan skripsi ini.
8. Arlinda Febriyanti selaku *support system* penulis yang selalu memberikan dorongan dan motivasi untuk menyelesaikan skripsi ini.
9. Seluruh pihak terkait lainnya yang telah banyak membantu dalam penyelesaian skripsi.

Semoga skripsi ini dapat bermanfaat bagi kita semua. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menjadikan skripsi ini lebih baik lagi.

Bandar Lampung, 23 Juli 2025

Syahreza

DAFTAR ISI

DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Tujuan Penelitian	2
1.3 Manfaat Penelitian	3
II TINJAUAN PUSTAKA	4
2.1 Optimalisasi	4
2.2 Penjadwalan	5
2.3 Model Matematika Masalah Penjadwalan	8
2.4 Penjadwalan <i>Job Shop</i>	9
2.5 Algoritma Genetika (AG)	11
2.5.1 Inisialisasi Populasi Awal	13
2.5.2 Representasi Individu dan Kromosom	13
2.5.3 Evaluasi Fitness	14
2.5.4 Seleksi	14
2.5.5 <i>Crossover</i>	14
2.5.6 Mutasi	15
2.5.7 Evaluasi Fitness dan Penempatan Kembali	16
2.5.8 Iterasi	16
2.6 Simulated Annealing	17
2.6.1 Teori Dasar	18
2.6.2 Pengaplikasian <i>Simulated Annealing</i>	18
III METODE PENELITIAN	19
3.1 Waktu dan Tempat Penelitian	19
3.2 Data Penelitian	19
3.3 Tahapan Penelitian	20
IV HASIL DAN PEMBAHASAN	21

4.1	Data Penelitian JSSP	21
4.2	Model Matematika JSSP	22
4.2.1	Variabel Keputusan	22
4.2.2	Fungsi Objektif	22
4.2.3	Kendala-Kendala	23
4.3	Penyelesaian JSSP Menggunakan Algoritma Genetika	26
4.3.1	Perhitungan Manual hingga Generasi Ke-5	26
4.3.2	Penyelesaian JSSP Menggunakan Algoritma Genetika Dalam Bahasa Pemrograman Python	32
4.4	Penyelesaian JSSP Menggunakan Algoritma Simulated Annealing .	38
4.4.1	Perhitungan Manual	38
4.5	Penyelesaian JSSP Menggunakan Algoritma Simulated Annealing dengan Bahasa Pemrograman Python	46
V	KESIMPULAN DAN SARAN	50
5.1	Kesimpulan	50
5.2	Saran	51
	DAFTAR PUSTAKA	52

DAFTAR TABEL

2.1	Tabel Penjadwalan <i>Job Shop</i>	10
2.2	Kromosom A dan Hasil Mutasi	16
4.1	Durasi Waktu <i>Job</i> pada setiap mesin untuk 50kg Kedelai (menit) . .	22
4.2	Durasi Waktu <i>Job</i> 1 pada setiap mesin untuk 50kg Kedelai (menit) .	24
4.3	Waktu Penyelesaian Operasi untuk <i>Job</i> 1 sampai <i>Job</i> 6 dalam Menit	24
4.4	Populasi Awal	27
4.5	Waktu Penyelesaian Operasi untuk urutan <i>Job</i> (2,5,3,4,1,6) dalam Menit	27
4.6	Evaluasi Fitness Generasi 0	27
4.7	Evaluasi Fitness Generasi 1	28
4.8	Evaluasi Fitness Generasi 2	29
4.9	Evaluasi Fitness Generasi 3	29
4.10	Evaluasi Fitness Generasi 4	30
4.11	Evaluasi Fitness Generasi 5	31
4.12	Waktu Penyelesaian Operasi untuk <i>Job</i> (2,6,1,4,3,5) dalam Menit . .	31
4.13	Hasil 100 Generasi Menggunakan Python	36
4.14	Hasil 100 iterasi Menggunakan Python	37
4.15	Waktu Penyelesaian Operasi untuk <i>Job</i> (2,3,4,1,6,5) dalam Menit . .	38
4.16	Parameter Simulated Annealing	39
4.17	Populasi Awal Simulated Annealing	40
4.18	Penyelesaian Operasi Iterasi 1	41
4.19	Penyelesaian Operasi Iterasi 2	42
4.20	Penyelesaian Operasi Iterasi 3	43
4.21	Penyelesaian Operasi Iterasi 4	44
4.22	Penyelesaian Operasi Iterasi 5	45
4.23	Tabel Perhitungan Hasil Iterasi	46
4.24	Hasil 22 iterasi Menggunakan Python	49

DAFTAR GAMBAR

2.1	Ilustrasi Kromosom, Individu, Populasi	13
2.2	Siklus Algoritma Genetika	16
4.1	Urutan <i>Job</i> acak menggunakan <i>syntax</i> python	26
4.2	Impor <i>script</i>	32
4.3	Input Data	32
4.4	Paramter AG	32
4.5	Inisialisasi Populasi Awal	33
4.6	Evaluasi Fitness	33
4.7	Proses Seleksi	33
4.8	Proses <i>Crossover</i>	34
4.9	Proses Mutasi	34
4.10	Pengujian Data	34
4.11	Membuat Diagram Gantt	35
4.12	Menjalankan Program	35
4.13	Diagram Gantt Urutan Pekerjaan Terbaik Dari 100 Generasi	37
4.14	Populasi Awal Simulated Annealing	39
4.15	Impor <i>script</i> SA	46
4.16	Input Data SA	47
4.17	Evaluasi Fitness Simulated Annealing	47
4.18	Proses Mutasi SA	47
4.19	Proses Mutasi SA	48
4.20	Algoritma SA	48

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penjadwalan merupakan salah satu aspek krusial dalam manajemen produksi yang bertujuan untuk mengoptimalkan penggunaan sumber daya dan meningkatkan efisiensi operasional. Dalam konteks *job shop*, di mana berbagai produk diproses secara bersamaan dengan jalur produksi yang berbeda, penjadwalan yang tepat menjadi tantangan tersendiri (Pinedo, 2016). Penjadwalan *job shop* dikenal sebagai masalah NP-hard, yang berarti tidak ada algoritma yang dapat menyelesaikannya dalam waktu polinomial untuk semua kasus. Kompleksitas ini disebabkan oleh banyaknya variabel yang harus dipertimbangkan, seperti urutan pekerjaan, waktu proses, dan keterbatasan mesin (Baker & Scudder, 1990).

Penjadwalan *job shop* adalah salah satu masalah penting dalam dunia manufaktur dan industri, di mana sebuah set pekerjaan (*job*) harus diproses pada berbagai mesin, dengan batasan waktu dan urutan tertentu. Tujuan dari penjadwalan *job shop* adalah untuk meminimalkan total waktu penyelesaian (*makespan*), mengoptimalkan penggunaan sumber daya, dan memenuhi berbagai constraint yang ada, seperti prioritas pekerjaan, waktu proses, dan kapasitas mesin.

Dalam penyelesaian masalah penjadwalan *job shop* terdapat beberapa metode yang digunakan antara lain adalah metode Algoritma Genetika, Algoritma Johnson, Algoritma NEH, Tabu Search, dan Simulated Annealing.

Dua metode yang sering digunakan dalam menyelesaikan masalah penjadwalan *job shop* adalah algoritma genetika dan algoritma *Simulated Annealing*. Algoritma genetika adalah metode optimasi yang terinspirasi oleh proses evolusi di alam, menggunakan prinsip seleksi, crossover, dan mutasi untuk mengeksplorasi ruang solusi yang besar (Yang et al., 2020). Penelitian menunjukkan bahwa algoritma genetika dapat menghasilkan solusi yang baik untuk berbagai masalah penjadwalan,

meskipun sering kali memerlukan waktu komputasi yang lebih lama dan pengaturan parameter yang cermat (Zhang dkk., 2019).

Di sisi lain, Simulated Annealing (SA) adalah algoritma pencarian berbasis probabilitas yang terinspirasi oleh proses pendinginan logam dalam fisika, di mana sistem secara bertahap mengurangi suhu untuk mencapai keadaan stabil. Dalam SA, solusi secara acak diubah, dan perubahan yang meningkatkan kualitas solusi diterima, sementara perubahan yang menurunkan kualitas solusi diterima dengan probabilitas yang semakin kecil seiring berjalannya waktu.

Optimalisasi penjadwalan tidak hanya berdampak pada efisiensi produksi, tetapi juga pada pengurangan biaya dan peningkatan kepuasan pelanggan. Dengan meningkatkan produktivitas melalui penjadwalan yang lebih baik, perusahaan dapat bersaing secara lebih efektif di pasar global (Kubiak & Gajewski, 2018). Dalam era digitalisasi, perkembangan teknologi informasi dan sistem produksi cerdas membuka peluang baru untuk inovasi dalam penjadwalan. Penggunaan algoritma canggih seperti AG dapat diintegrasikan dengan sistem manajemen produksi untuk mencapai hasil yang lebih baik (Zhou et al., 2018).

Pada penelitian ini, Penulis ingin membandingkan metode Algoritma Genetika dengan metode Simulated Annealing pada data produksi Tahu Topo Sejahtera Metro Lampung. Diharapkan dapat memberikan wawasan mengenai efektivitas Algoritma Genetika dan metode *Simulated Annealing* dalam penjadwalan *job shop*, serta memberikan rekomendasi praktis bagi industri dalam memilih metode yang paling sesuai untuk meningkatkan efisiensi sistem produksi.

1.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Menentukan *makespan* minimum pada penjadwalan *job shop* menggunakan Algoritma Genetika dan Algoritma *Simulated Annealing*.
2. Membandingkan hasil yang diperoleh menggunakan Algoritma Genetika dan Algoritma *Simulated Annealing* untuk mengetahui mana yang memberikan hasil terbaik.

1.3 Manfaat Penelitian

Adapun manfaat yang didapat dari penelitian ini adalah :

1. Mengoptimalkan waktu dalam penjadwalan *job shop* sehingga meningkatkan efisiensi produksi.
2. Membantu Meningkatkan efisiensi oprasional dan produktivitas dalam industri manufaktur.
3. Manufaktur dapat memperoleh strategi penjadwalan yang lebih efisien guna meningkatkan produktivitas dan mengoptimalkan penggunaan sumber daya yang tersedia.

BAB II

TINJAUAN PUSTAKA

2.1 Optimalisasi

Optimalisasi merupakan proses mencari nilai terbaik dari suatu fungsi atau sistem di bawah kendala tertentu. Dalam berbagai disiplin ilmu, konsep ini memiliki aplikasi yang luas, mulai dari matematika, ekonomi, teknik, hingga ilmu komputer. Dalam tinjauan pustaka ini, akan dibahas beberapa aspek penting dari optimalisasi, termasuk definisi, metode, dan aplikasinya.

Definisi 2.1.1 Optimalisasi didefinisikan sebagai proses menemukan solusi terbaik dari suatu masalah yang melibatkan variabel-variabel tertentu. Menurut Winston (2004), optimalisasi adalah "proses memilih nilai terbaik dari fungsi objektif yang dipengaruhi oleh sejumlah kendala". Fungsi objektif adalah fungsi yang ingin dimaksimalkan atau diminimalkan, sedangkan kendala adalah batasan yang harus dipenuhi oleh solusi. Optimalisasi dapat dibedakan menjadi beberapa jenis, antara lain :

1. Optimalisasi Linier: Melibatkan fungsi objektif dan kendala yang bersifat linier. Metode Simplex adalah salah satu teknik yang umum digunakan dalam optimalisasi linier (Dantzig, 1963).
2. Optimalisasi Non-linier: Terjadi ketika fungsi objektif atau kendala tidak linier. Metode *Lagrange* dan metode *gradient descent* adalah beberapa teknik yang digunakan dalam konteks ini (Boyd Vandenberghe, 2004).
3. Optimalisasi Diskrit: Berfokus pada masalah di mana variabel keputusan hanya dapat mengambil nilai tertentu, seperti dalam pemrograman integer (Papadimitriou Steiglitz, 1998).
4. Optimalisasi Stokastik: Menghadapi ketidakpastian dalam data atau model. Pendekatan ini sering digunakan dalam ekonomi dan keuangan (Birge Louveaux, 2011).

Berbagai metode telah dikembangkan untuk menyelesaikan masalah optimisasi. Beberapa metode yang umum digunakan antara lain:

1. Metode *Gradient*: Digunakan untuk menemukan titik maksimum atau minimum dari fungsi dengan menghitung gradiennya (Nocedal Wright, 2006).
2. Metode Simplex: Khusus untuk masalah optimisasi linier, metode ini mencari solusi optimal dengan berpindah dari satu titik sudut ke titik sudut lainnya dalam ruang solusi (Dantzig, 1963).
3. Pemrograman Dinamis: Metode ini digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi sub-masalah yang lebih kecil dan lebih sederhana (Bellman, 1957).
4. Algoritma Genetika: Merupakan metode heuristik yang terinspirasi oleh proses evolusi alam. Algoritma ini digunakan untuk menyelesaikan masalah yang kompleks dan tidak dapat diselesaikan dengan metode tradisional (Holland, 1975).
5. Algoritma *Simulated Annealing*: Metode yang digunakan untuk menemukan solusi optimal dengan mengeksplorasi ruang solusi secara acak dan menghindari solusi yang sudah pernah dieksplorasi sebelumnya (Kirkpatrick et al., 1983; Glover, 1989).

Optimalisasi adalah bidang yang penting dan luas dengan banyak aplikasi praktis. Dengan pemahaman yang mendalam tentang metode dan teknik yang tersedia, individu dan organisasi dapat membuat keputusan yang lebih baik dan lebih efisien. Penelitian lebih lanjut dalam bidang ini dapat menghasilkan metode baru dan inovatif yang dapat diterapkan dalam berbagai konteks.

2.2 Penjadwalan

Penjadwalan merujuk pada proses pengaturan dan pengorganisasian sumber daya untuk melaksanakan serangkaian tugas atau pekerjaan dalam periode waktu tertentu. Dalam berbagai bidang seperti *manufaktur*, transportasi, dan manajemen proyek, penjadwalan memiliki peran penting dalam meningkatkan *efisiensi* dan *produktivitas*. Tinjauan pustaka ini akan membahas pemahaman dasar tentang penjadwalan, berbagai jenis penjadwalan, metode yang digunakan, serta tantangan dan aplikasi dalam berbagai konteks.

Definisi 2.2.1 Penjadwalan dapat didefinisikan sebagai proses menentukan urutan dan waktu pelaksanaan tugas atau pekerjaan untuk mencapai tujuan tertentu, seperti meminimalkan waktu penyelesaian, mengurangi biaya, atau meningkatkan penggunaan sumber daya (Pinedo, 2016). Penjadwalan melibatkan pengalokasian sumber daya, seperti mesin, tenaga kerja, dan bahan, untuk menyelesaikan pekerjaan dalam batas waktu yang ditentukan.

Penjadwalan dapat dibedakan menjadi beberapa jenis, antara lain:

1. Penjadwalan Proyek: Mengatur urutan dan waktu pelaksanaan tugas dalam proyek untuk mencapai tujuan proyek secara efisien. Metode seperti PERT (Program Evaluation and Review Technique) dan CPM (Critical Path Method) sering digunakan dalam konteks ini (Kerzner, 2017).
2. Penjadwalan Produksi: Mengatur urutan dan waktu produksi barang di pabrik. Ini termasuk penjadwalan mesin, penjadwalan tenaga kerja, dan penjadwalan bahan baku (Graham, 2000).
3. Penjadwalan Transportasi: Mengatur rute dan waktu pengiriman barang atau penumpang. Ini melibatkan pengoptimalan rute untuk mengurangi biaya dan waktu perjalanan (Toth Vigo, 2001).
4. Penjadwalan Komputasi: Mengatur tugas dalam sistem komputer untuk memaksimalkan penggunaan CPU dan meminimalkan waktu tunggu. Ini termasuk penjadwalan proses dan penjadwalan tugas dalam sistem operasi (Silberschatz et al., 2018).

Berbagai metode dikembangkan untuk menyelesaikan masalah penjadwalan. Beberapa metode yang umum digunakan antara lain :

1. Metode *Heuristik*: Pendekatan berbasis aturan yang digunakan untuk menemukan solusi yang baik dalam waktu yang lebih singkat. Contoh metode heuristik termasuk Algoritma Genetik, Algoritma NEH, Algoritma Simulated Annealing dan Algoritma Greedy (Holland, 1975; Nawaz et al., 1983).
2. Pemrograman Linier: Metode matematis yang digunakan untuk memodelkan dan menyelesaikan masalah penjadwalan dengan kendala linier. Metode Simplex adalah salah satu teknik yang umum digunakan dalam pemrograman linier (Dantzig, 1963).

3. Pemrograman Dinamis: Metode yang digunakan untuk menyelesaikan masalah penjadwalan yang dapat dipecah menjadi sub-masalah yang lebih kecil dan lebih sederhana (Bellman, 1957)
4. *Simulated Annealing* dan *Tabu Search*: Metode yang digunakan untuk menemukan solusi optimal dengan mengeksplorasi ruang solusi secara acak dan menghindari solusi yang sudah pernah dieksplorasi sebelumnya (Kirkpatrick et al., 1983; Glover, 1989).

Penjadwalan sering kali menghadapi berbagai tantangan, antara lain :

1. Ketidakpastian: Ketidakpastian dalam waktu penyelesaian tugas, ketersediaan sumber daya, dan permintaan dapat mempengaruhi efektivitas penjadwalan (Baker Trietsch, 2009).
2. Kendala Sumber Daya: Keterbatasan dalam jumlah mesin, tenaga kerja, dan bahan dapat membatasi fleksibilitas dalam penjadwalan (Pinedo, 2016).
3. Kompleksitas Masalah: Masalah penjadwalan sering kali NP-Hard, yang berarti tidak ada algoritma yang dapat menyelesaikannya dalam waktu polinomial untuk semua kasus (Garey Johnson, 1979).

Penjadwalan memiliki aplikasi yang luas di berbagai bidang, antara lain :

1. Industri Manufaktur: Dalam pengaturan produksi, penjadwalan digunakan untuk mengoptimalkan penggunaan mesin dan tenaga kerja, serta meminimalkan waktu tunggu dan biaya produksi (Graham, 2000).
2. Manajemen Proyek: Dalam manajemen proyek, penjadwalan digunakan untuk merencanakan dan mengelola urutan tugas, memastikan bahwa proyek selesai tepat waktu dan sesuai anggaran (Kerzner, 2017).
3. Transportasi dan Logistik: Penjadwalan rute pengiriman dan pengaturan jadwal transportasi sangat penting untuk efisiensi operasional dan pengurangan biaya (Toth Vigo, 2001).
4. Sistem Komputer: Dalam sistem operasi, penjadwalan proses digunakan untuk mengelola eksekusi tugas dan memaksimalkan penggunaan sumber daya komputer (Silberschatz et al., 2018).

Penjadwalan merupakan aspek penting dalam manajemen sumber daya yang berpengaruh terhadap efisiensi dan produktivitas di berbagai sektor. Dengan mempelajari berbagai jenis penjadwalan, metode yang diterapkan, serta tantangan yang dihadapi, peneliti dan praktisi dapat merumuskan strategi yang lebih efektif untuk mengatasi masalah penjadwalan yang rumit. Penelitian lebih lanjut dibutuhkan untuk mengembangkan metode baru yang dapat mengatasi 8 tantangan yang ada dan meningkatkan kinerja penjadwalan dalam berbagai konteks.

2.3 Model Matematika Masalah Penjadwalan

Variabel Keputusan

- x_{ijk} : Variabel *biner*, bernilai 1 jika operasi— $_{ij}$ (operasi ke- j pada *job* ke- i) dijadwalkan pada mesin k , bernilai 0 jika tidak.
- C_{ij} : Waktu penyelesaian operasi— $_{ij}$.
- P_{ij} : Waktu pemrosesan operasi— $_{ij}$ pada Mesin ke- j .
- S_{ij} : Waktu mulai operasi— $_{ij}$.

Fungsi Objektif

Fungsi objektif untuk meminimalkan *makespan* (total waktu penyelesaian):

$$\min C_{max} \quad (2.3.1)$$

Dengan

$$C_{max} = \max(C_{ij}), \quad \forall (i,j) \quad (2.3.2)$$

Kendala-Kendala

1. Urutan Operasi dalam *Job* :

Operasi pada *job* harus dikerjakan secara berurutan:

$$S_{i,j+1} \geq C_{ij}, \quad \forall i = 1, 2, \dots, n, \quad \forall j = 1, 2, \dots, n - 1 \quad (2.3.3)$$

(Waktu mulai operasi ke- $j + 1$ dari *job* i harus lebih besar atau sama dengan waktu selesai operasi ke- j dari *job* yang sama).

2. Satu Mesin Tidak Bisa Memproses Dua *Job* Bersamaan:

Setiap mesin hanya dapat menjalankan satu operasi dalam satu waktu:

$$S_{ij} \geq C_{lj} \quad \text{atau} \quad S_{lj} \geq C_{ij} \quad , \quad \forall (i, j) \neq (l, j), \quad (2.3.4)$$

jika operasi dijalankan pada mesin yang sama.

3. Waktu Penyelesaian Operasi :

$$C_{ij} = S_{ij} + p_{ij} \quad , \quad \forall (i, j) \quad (2.3.5)$$

di mana p_{ij} adalah durasi operasi- ij .

4. Alokasi Mesin :

Operasi hanya dapat dijalankan pada mesin yang sesuai:

$$\sum_k x_{ijk} = 1 \quad , \quad \forall i, j \quad (2.3.6)$$

2.4 Penjadwalan *Job Shop*

Penjadwalan *job shop* adalah metode pengaturan proses produksi yang digunakan untuk mengelola berbagai jenis pekerjaan yang berbeda dalam satu fasilitas produksi. Sistem ini ditandai dengan fleksibilitas tinggi, di mana setiap pekerjaan dapat memiliki urutan dan waktu proses yang berbeda (Pinedo, 2016). Konsep utama dalam penjadwalan *job shop* adalah untuk meminimalkan waktu penyelesaian total (*makespan*), keterlambatan (*tardiness*), mengurangi waktu tunggu, dan memaksimalkan penggunaan sumber daya.

Kriteria yang umum digunakan dalam penjadwalan *job shop* meliputi *makespan* (total waktu penyelesaian), *tardiness* (keterlambatan), dan total waktu tunggu (Cheng et al., 2005). Penjadwalan yang efektif tidak hanya berfokus pada pengurangan waktu penyelesaian, tetapi juga pada pengoptimalan penggunaan sumber daya dan peningkatan kepuasan pelanggan.

Makespan adalah waktu total yang diperlukan untuk menyelesaikan semua pekerjaan dalam sistem penjadwalan. Ini diukur dari waktu mulai pekerjaan pertama hingga waktu selesai pekerjaan terakhir. *Makespan* yang lebih pendek menunjukkan

efisiensi yang lebih baik dalam penjadwalan, karena berarti semua pekerjaan dapat diselesaikan lebih cepat. Dalam konteks industri, mengurangi *makespan* dapat meningkatkan produktivitas dan mengurangi biaya operasional. Sedangkan *Tardiness* mengacu pada seberapa lama suatu pekerjaan terlambat dalam penyelesaiannya dibandingkan dengan waktu yang dijadwalkan. Jika pekerjaan selesai setelah tenggat waktu yang ditentukan, maka pekerjaan tersebut dianggap terlambat. Keterlambatan dapat berdampak negatif pada kepuasan pelanggan dan dapat menyebabkan penalti atau biaya tambahan. Oleh karena itu, mengurangi tingkat keterlambatan adalah salah satu tujuan penting dalam penjadwalan. Begitu juga dengan Total waktu tunggu mengacu pada jumlah waktu yang dihabiskan oleh pekerjaan dalam antrean sebelum mulai diproses. Waktu tunggu yang tinggi dapat menunjukkan adanya kemacetan dalam sistem penjadwalan dan dapat mengindikasikan bahwa sumber daya tidak digunakan secara efisien. Mengurangi total waktu tunggu dapat meningkatkan *throughput* (jumlah pekerjaan yang diselesaikan dalam periode tertentu) dan mengurangi waktu siklus keseluruhan.

Tantangan utama dalam penjadwalan *job shop* termasuk ketidakpastian dalam waktu proses, perubahan permintaan, dan keterbatasan sumber daya (Baker dan Kanet, 2005). Penjadwalan yang adaptif dan responsif menjadi semakin penting dalam menghadapi dinamika pasar yang cepat berubah.

Sebagai contoh penerapan penjadwalan *job shop*, pertimbangkan sebuah pabrik yang memproduksi tiga jenis produk (A, B, dan C) dengan dua mesin (M1 dan M2). Setiap produk memiliki waktu proses yang berbeda pada masing-masing mesin: Dalam penjadwalan ini, manajer produksi harus menentukan urutan pekerjaan

Tabel 2.1 Tabel Penjadwalan *Job Shop*

Produk	M1	M2
A	3	2
B	2	1
C	4	3

untuk meminimalkan *makespan*. Misalnya, jika urutan pekerjaan adalah $A \rightarrow B \rightarrow C$, maka total waktu penyelesaian dapat dihitung berdasarkan waktu proses pada masing-masing mesin.

2.5 Algoritma Genetika (AG)

Algoritma Genetika (AG) dikembangkan pada tahun 1975 di Universitas Michigan oleh Jhon Holland. AG diartikan sebagai algoritma yang menggunakan proses seleksi alam atau bisa juga disebut proses evolusi sesuai dengan yang diutarakan oleh Charles Darwin. Individu yang ada akan melakukan perubahan gen secara terus-menerus hingga sesuai dengan lingkungannya (Mubarok dan Chotijah, 2021). Menurut (Lukas, dkk., 2005), proses AG terdapat beberapa langkah, yaitu pengkodean (*encoding*), seleksi (*selection*), persilangan (*crossover*), dan mutasi (*mutation*).

Salah satu pendekatan yang cukup efektif untuk menyelesaikan permasalahan *NP-Hard* adalah menggunakan algoritma heuristik. Meskipun algoritma ini tidak dapat menjamin solusi yang paling optimal, namun ia mampu menangani permasalahan yang kompleks dan berskala besar secara efisien, dengan mencari solusi yang cukup baik dan memenuhi berbagai kriteria dalam waktu komputasi yang relatif singkat. Salah satu jenis algoritma heuristik yang cukup dikenal adalah *neighborhood search methods*, yang mencakup tiga teknik utama, yaitu: *Genetic Algorithm*, *Simulated Annealing*, dan *Tabu Search*.

Algoritma Genetika (*Genetic Algorithm (GA)*) adalah metode optimasi berbasis *heuristic* yang terinspirasi oleh prinsip evolusi biologis, seperti seleksi alam, *crossover*, dan mutasi. Algoritma ini digunakan untuk menyelesaikan permasalahan optimasi kompleks, termasuk penjadwalan **Job Shop Scheduling Problem (JSSP)**. **Job Shop Scheduling Problem (JSSP)** adalah masalah optimasi yang bertujuan untuk menjadwalkan serangkaian pekerjaan (*job*) pada sejumlah mesin agar memenuhi fungsi objektif tertentu, seperti:

- *Makespan* : Total waktu penyelesaian semua pekerjaan.
- *Flow Time* : Jumlah total waktu penyelesaian semua pekerjaan.
- *Tardiness* : Penundaan pekerjaan yang selesai setelah tenggat waktu.

Karena **JSSP** termasuk dalam kelas masalah **NP-hard**, metode optimasi seperti algoritma genetika cocok digunakan untuk mencari solusi yang mendekati optimal dalam waktu komputasi yang terbatas.

Penggunaan Algoritma Genetika

1. Representasi Solusi

Solusi dalam GA direpresentasikan sebagai kromosom. Dalam konteks JSSP, kromosom dapat berupa urutan operasi untuk setiap mesin.

Contoh kromosom:

$$[O_{11}, O_{12}, O_{21}, O_{22}, \dots] \quad (2.5.7)$$

2. Evaluasi Fitness

Setiap kromosom dievaluasi berdasarkan fungsi objektif (C_{max}).

3. Operator Evolusi

- **Seleksi:** Memilih individu terbaik berdasarkan nilai *fitness*.
- **Crossover:** Menggabungkan dua solusi untuk menghasilkan solusi baru.
- **Mutasi:** Memodifikasi solusi untuk menjaga keberagaman populasi.

4. Iterasi

Algoritma dijalankan hingga *konvergensi* atau mencapai batas iterasi yang ditentukan.

Beberapa generasi baru yang terbentuk dan memiliki ketahanan hidup yang kuat akan bermunculan didalam populasi tersebut, yang kemudian dari proses seleksi, kawin silang, dan mutasi yang berkelanjutan menghasilkan generasi dengan kemampuan bertahan hidup yang bagus. Algoritma ini berbeda dari algoritma tradisional pada beberapa hal (Goldberg, 1989) yaitu:

- AG berkeja dalam bentuk kode sekumpulan parameter, bukan parameter itu sendiri.
- AG mencari solusi dari sekumpulan populasi, bukan dari satu titik solusi.
- AG langsung menggunakan informasi berupa fungsi tujuan dan bukan turunannya.
- AG menggunakan aturan probabilitas, bukan aturan deterministik. Penerapan metode Algoritma Genetika melibatkan delapan langkah utama (Chairul Saleh, 2002) :

1. Inisialisasi populasi
2. Representasi individu dan kromosom

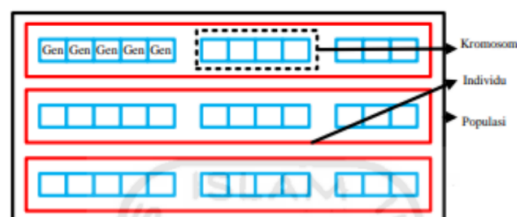
3. Evaluasi fitness
4. Seleksi
5. Crossover
6. Mutasi
7. Penempatan
8. Iterasi

2.5.1 Inisialisasi Populasi Awal

Ada dua cara untuk menentukan inisialisasi populasi awal, yaitu secara random dan secara heuristic (Gen dan Cheng, 2000). Dalam penelitian ini inisialisasi populasi dibangkitkan secara random.

2.5.2 Representasi Individu dan Kromosom

Representasi memberikan pengaruh yang kuat terhadap performansi Algoritma Genetika. Representasi individu bergantung pada masalah yang akan diselesaikan. Individu menyatakan salah satu solusi yang mungkin. Setiap individu merupakan kumpulan kromosom sedangkan kromosom merupakan kumpulan dari gen. Sedangkan populasi merupakan kumpulan dari beberapa individu. Pada gambar 2.1 diilustrasikan perbedaan istilah-istilah tersebut. Individu merupakan kombinasi



Gambar 2.1 Ilustrasi Kromosom, Individu, Populasi

permutasi dari penjadwalan pekerjaan yang dilakukan oleh mesin, dari mesin-1 hingga mesin-n. Setiap individu terdiri dari n kromosom. Kromosom menunjukkan urutan *job* yang dibebankan pada mesin tersebut. Sebagai contoh, misalkan ada 2 mesin dan jumlah *job* yang melalui mesin satu yaitu *job* 1, 2, 3 dan *job* yang melalui mesin 2 yaitu *job* 1, 2, 3 maka representasinya yaitu (1-2-3)-(1-2-3).

2.5.3 Evaluasi Fitness

Fungsi *fitness* digunakan untuk mengevaluasi individu dalam populasi. Fitness adalah sebuah perhitungan nilai yang harus dimaksimalkan, diminimalkan atau disesuaikan dengan nilai terdekat paling mungkin yang diberikan. Pada penjadwalan, waktu yang diperlukan untuk menyelesaikan semua pekerjaan yang ada di shop (*makespan*) harus diminimalkan. Pada penelitian ini, *fitness (obyektif)* yang dicari yaitu *makespan*. Obyektif *makespan* adalah total waktu yang dibutuhkan untuk menyelesaikan semua *job* yang ada atau dapat juga dikatakan sebagai waktu saat selesainya *job* terakhir. Obyektif *makespan* ini dapat dinyatakan dengan persamaan matematis :

$$Makespan = \max[C_1, C_2, \dots, C_n] \quad (2.5.8)$$

Individu yang mempunyai nilai fungsi *fitness* terkecil adalah individu terbaik yang akan mendekati optimal. Tujuan penjadwalan pada penelitian adalah minimasi fungsi *fitness* F supaya didapatkan nilai *makespan* yang minimal. Dengan nilai *makespan* yang kecil maka diharapkan akan mengurangi keterlambatan dalam penyelesaian pembuatan produk.

2.5.4 Seleksi

Prosedur seleksi pada intinya mengarahkan pencarian pada Algoritma Genetik ke daerah yang lebih menjanjikan pada ruang pencarian. Terdapat tiga hal yang perlu diperhatikan dalam fase seleksi, meliputi ruang sampling, mekanisme sampling dan peluang sampling (Gen and Cheng, 2000). Ada dua metode seleksi, yaitu proporsional dan ranking. Dalam penelitian ini akan digunakan metode ranking yaitu dengan cara mengurutkan hasil *evaluasi fitness* setiap individu lalu akan dipilih individu dengan hasil terbaik.

2.5.5 Crossover

Dalam algoritma genetika, "*crossover*" adalah proses menggabungkan dua individu untuk membuat individu baru karena hal tersebut *crossover* umumnya diistilahkan dengan X . Dua individu baru tersebut dibentuk oleh dua individu induk dalam populasi melalui proses seleksi sebelumnya. Dalam proses persilangan, perlu ditentukan probabilitas persilangan (*Crossover Probability*) yang menunjukkan

kemungkinan *crossover* terjadi antara 2 individu. Jika tidak terjadi *crossover*, maka keturunannya akan sama persis dengan individu orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. *Crossover* dilakukan dengan harapan bahwa individu yang baru akan lebih baik.

Crossover memiliki beberapa metode yang umum di gunakan yaitu:

1. *One-Point Crossover* (Persilangan Satu Titik) : Dalam metode ini, memilih satu titik pada kromosom induk, lalu menukar bagian kromosom di sebelah kanan dari titik tersebut antara kedua induk.
2. *Two-Point Crossover* (Persilangan Dua Titik) : Dalam metode ini, memilih dua titik pada kromosom induk, lalu menukar bagian di antara kedua titik tersebut antara kedua induk.
3. *Order Crossover* (OX) : Metode *Order Crossover* (OX) adalah salah satu metode yang cocok untuk masalah penjadwalan, di mana harus memastikan bahwa setiap pekerjaan hanya muncul sekali dalam kromosom anak. Metode ini menjaga urutan gen dari satu induk, sambil menyisipkan elemen yang hilang dari induk lain.
4. *Partially Mapped Crossover* (PMX) : Metode PMX adalah metode *crossover* lain yang cocok untuk masalah penjadwalan. Dalam metode ini, menukar segmen antara dua titik potong antara dua kromosom induk, kemudian memperbaiki duplikasi yang mungkin terjadi.

2.5.6 Mutasi

Mutasi adalah sebuah proses dari perubahan gen dalam kondisi random untuk membentuk gen baru dalam kelompok gen. Dalam Algoritma Genetika, mutasi memberikan aturan untuk menempatkan kembali gen yang hilang dari populasi selama proses seleksi sehingga dapat memberikan gen yang tidak ada pada populasi sebelumnya. Mutasi digunakan untuk mencegah generasi premature dalam operasi *crossover*. Pada proses mutasi, diperlukan adanya probabilitas mutasi. Probabilitas Mutasi (*Mutation Probability*) menunjukkan kemungkinan mutasi terjadi pada gen-gen yang menyusun sebuah individu. Jika tidak terjadi mutasi maka keturunan yang dihasilkan setelah *crossover* tidak berubah. Jika terjadi mutasi bagian individu akan berubah.

Tabel 2.2 menunjukkan proses mutasi pada gen ketiga dan ketujuh.

Tabel 2.2 Kromosom A dan Hasil Mutasi

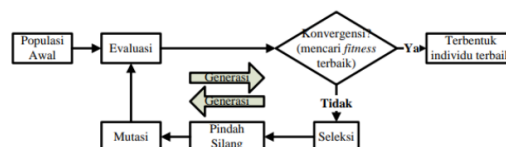
A	Job7	Job6	Job4	Job5	Job3	Job2	Job1
Hasil	Job7	Job6	Job1	Job5	Job3	Job2	Job4

2.5.7 Evaluasi Fitness dan Penempatan Kembali

Setelah mutasi, perlu mengevaluasi fitness individu baru ini dengan cara yang sama seperti sebelumnya. Hitung waktu penyelesaian total untuk individu yang telah dimutasi. Berdasarkan pada pendekatan pembentukan generasi, semua populasi ditempatkan kembali dalam masing-masing generasi (Goldberg, 1989). Dalam penelitian ini menggunakan generator Algoritma Genetika. Dalam generator ini, jumlah dari n populasi terbaik dalam generasi terakhir akan digunakan untuk menempatkan kembali populasi yang terburuk dari generasi sebelumnya. Metode ini membuat sejumlah individu terbaik tiap generasi akan otomatis diturunkan ke generasi berikutnya (Koza, 2001).

2.5.8 Iterasi

Proses pemilihan, *crossover*, mutasi, *evaluasi fitness*, dan penempatan diulang selama beberapa generasi. Setiap generasi diharapkan menghasilkan individu dengan waktu penyelesaian yang lebih baik. Proses ini berlanjut hingga mencapai solusi yang memuaskan atau hingga jumlah generasi tertentu tercapai. Jumlah generasi (jumlah iterasi digunakan sebagai batas akhir proses seleksi, persilangan dan mutasi. Jumlah generasi besar berarti semakin banyak iterasi yang dilakukan, dan semakin besar domain solusi yang akan dieksplorasi tetapi Algoritma Genetik akan berjalan lambat. Jika diilustrasikan, maka proses Algoritma Genetika yang dilakukan dapat terlihat seperti di bawah ini :



Gambar 2.2 Siklus Algoritma Genetika

2.6 Simulated Annealing

Simulated Annealing (SA) adalah salah satu metode metaheuristik yang digunakan untuk memecahkan masalah optimasi. Metode ini terinspirasi oleh proses fisika *annealing*, yaitu proses pendinginan logam atau kaca yang dilakukan secara bertahap untuk mencapai keadaan energi minimal. Dalam konteks optimasi, SA digunakan untuk menemukan solusi terbaik (global optimum) dari masalah yang memiliki banyak solusi lokal.

SA pertama kali diperkenalkan oleh Kirkpatrick, Gelatt, dan Vecchi pada tahun 1983 dalam artikel mereka yang berjudul "*Optimization by Simulated Annealing*" (Kirkpatrick et al., 1983). Mereka mengadaptasi teori fisika untuk aplikasi *optimasi kombinatorial*, di mana pencarian dilakukan dalam ruang solusi yang besar dan tidak terstruktur.

Penggunaan Algoritma *Simulated Annealing*

1. Inisialisasi

- Tentukan solusi awal S_0 dan suhu awal T_0 .
- Tentukan nilai α (*cooling rate*), suhu minimum T_0 , dan jumlah iterasi maksimum.

2. Evauasi Solusi Awal

Hitung nilai fungsi objektif $C_{maxlama}$ dari solusi awal.

3. Iterasi

- Tukar solusi secara acak untuk menghasilkan solusi baru S_{baru} .
- Hitung nilai fungsi objektif $C_{maxbaru}$ dari solusi baru.
- Tentukan perubahan energi $\Delta C_{max} = C_{maxbaru} - C_{maxlama}$.
- Jika $\Delta C_{max} \leq 0$, terima solusi baru.
- Jika $\Delta C_{max} > 0$, terima solusi baru dengan probabilitas $P = \exp\left(-\frac{\Delta C_{max}}{T}\right)$.

4. Pendinginan

Pada setiap iterasi, suhu diperbarui dengan menggunakan rumus $T_{k+1} = \alpha \cdot T_k$.

5. Penyelesaian

Setelah jumlah iterasi maksimum tercapai atau suhu mencapai T_{min} , algoritma berhenti dan solusi terbaik ditemukan.

2.6.1 Teori Dasar

Simulated Annealing mengandalkan konsep fisika dalam proses pendinginan sistem. Secara matematis, SA diimplementasikan dengan langkah-langkah berikut.

- Suhu awal (T_0) dimulai pada nilai tinggi.
- Suhu berkurang secara bertahap sesuai dengan fungsi pendinginan, sering kali dengan *cooling rate* α .
- Pada suhu tinggi, sistem eksplorasi lebih bebas dan bisa menerima solusi yang lebih buruk.
- Seiring suhu menurun, algoritma semakin fokus pada pencarian solusi terbaik dengan mengurangi probabilitas menerima solusi buruk.

2.6.2 Pengaplikasian *Simulated Annealing*

Simulated Annealing digunakan dalam berbagai masalah optimasi seperti:

- **Penjadwalan (*Job Shop Scheduling*):** Penjadwalan pekerjaan dengan berbagai mesin dan waktu proses yang berbeda.
- **Optimasi Fungsi Multidimensi:** Dalam masalah yang melibatkan fungsi kompleks yang sulit dihitung secara langsung.
- **Pencarian Parameter dalam Machine Learning:** Menyusun hyperparameter model.
- **Pengolahan Citra dan Pemrograman Komputasi:** Misalnya dalam pemecahan masalah pemrograman linear.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada Semester Genap Tahun Ajaran 2024/2025 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang beralamatkan di Jalan Prof. Dr. Ir. Soemantri Brojonegoro, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Lampung.

3.2 Data Penelitian

Penelitian ini menggunakan metode pengumpulan data dengan melakukan wawancara bersama narasumber serta didukung oleh studi literatur yang diperoleh. Wawancara yang dilakukan Bersama narasumber bertujuan untuk mendapatkan data terkait PABRIK PRODUKSI TAHU dan ONCOM TOPO SEJAHTERA METRO, LAMPUNG.

Tahapan dalam penelitian ini adalah identifikasi masalah, perumusan masalah, penyelesaian masalah, dan penarikan kesimpulan. Identifikasi masalah dilakukan untuk mendapat gambaran mengenai permasalahan yang akan diselesaikan. Data pada penelitian ini diperoleh dengan survey langsung ke lapangan. Penyelesaian masalah dilakukan dengan menggunakan Algoritma Genetika dan Algoritma Simulated Annealing.

3.3 Tahapan Penelitian

Tahapan penelitian ini meliputi penentuan populasi dan sampel, pengumpulan data, penerapan metode yang tepat, analisis data, serta penarikan kesimpulan. Penjelasan ini diharapkan dapat memberikan gambaran yang jelas tentang cara penulis melakukan penelitian. Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Menentukan dengan jelas masalah penjadwalan *Job Shop* yang akan diteliti.
2. Memilih metode yang tepat, seperti Algoritma Genetika dan Algoritma *Simulated Annealing*, untuk menyelesaikan masalah.
3. Mengumpulkan data penjadwalan produksi barang dari studi kasus Pabrik Tahu Topo Sejahtera Metro, Lampung.
4. Menerapkan Algoritma Genetika dan Algoritma *Simulated Annealing* untuk memecahkan masalah penjadwalan *Job Shop*.
5. Menentukan solusi yang dihasilkan oleh masing-masing algoritma.
6. Menganalisis dan membandingkan solusi dari Algoritma Genetika dan Algoritma *Simulated Annealing*.
7. Menyimpulkan hasil analisis dan perbandingan, serta memberikan rekomendasi untuk solusi terbaik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai optimalisasi penjadwalan Job Shop menggunakan Algoritma Genetika dan Algoritma Simulated Annealing dapat diketahui beberapa kesimpulan yaitu, Algoritma Genetika (AG) dan Algoritma Simulated Annealing dapat digunakan untuk menyelesaikan masalah penjadwalan Job Shop Scheduling Problem (JSSP) yang kompleks dan menghasilkan solusi yang dapat meningkatkan efisiensi waktu produksi.

Hasil dari penelitian ini menunjukkan bahwa Algoritma Genetika dan Algoritma Simulated Annealing mampu menghasilkan solusi dengan *makespan* yang sama optimal yaitu pada angka 448 menit setelah beberapa generasi/iterasi. Algoritma Simulated Annealing, sebagai metode heuristik yang memberikan hasil dengan perhitungan manual yang lebih sederhana dibandingkan Algoritma Genetika dengan hasil yang sama optimal yaitu 448 menit. Berdasarkan hasil simulasi menggunakan pemrograman python, SA menghasilkan lebih sedikit iterasi dengan iterasi 22 sedangkan untuk AG membutuhkan 100 generasi untuk menentukan hasil optimal.

Perbandingan antara kedua metode menunjukkan bahwa AG lebih unggul dalam penggunaan *syntax* python dalam menemukan solusi optimal, sementara Algoritma Simulated Annealing lebih banyak *syntax* dalam menemukan solusi yang optimal. Pada penelitian menggunakan cara manual maupun pemrograman hasil terbaik akan muncul jika kedelai hitam (*Job2*) dikerjakan awal dan kedelai anjasmoro (*Job5*) dikerjakan diakhir.

5.2 Saran

Beberapa saran dapat diberikan dalam pengembangan lebih lanjut sebagai berikut:

1. **Eksperimen dengan Parameter Berbeda:** Dalam implementasi Algoritma Genetika, perlu dilakukan pengujian lebih lanjut terhadap parameter seperti jumlah populasi, tingkat mutasi, dan probabilitas crossover untuk melihat dampaknya terhadap hasil akhir. Dalam parameter Algoritma Simulated Annealing, perlu dilakukan pengujian lebih lanjut terhadap parameter seperti T_{start} , T_{end} , dan α .
2. **pengujian dengan Dataset Lebih Besar:** Untuk menguji kehandalan metode yang digunakan, penelitian lebih lanjut dapat dilakukan dengan dataset yang lebih variatif.
3. **Implementasi pada Industri Lain:** Penelitian ini dapat diperluas dengan mengaplikasikan metode yang sama pada industri lain yang memiliki permasalahan penjadwalan produksi serupa, seperti industri otomotif atau manufaktur lainnya.

DAFTAR PUSTAKA

- Baker, K. R., dan Kanet, J. J. (2005). *Introduction to Scheduling. In Production and Operations Management*. (pp. 123-145). New York: Wiley.
- Baker, K. R., Scudder, G. D. (1990). *Sequencing with minimization of maximum lateness*. Operations Research.
- Baker, K. R., dan Trietsch, D. (2009). *Principles of Scheduling*. Wiley.
- Bierwirth, C., D.C. Mattfield, (1999). “*Production Scheduling and Rescheduling with Genetic Algorithm*”, *Evolutionary Computation*,. 7 (1), 1999, 1-17.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
- Birge, J. R., dan Louveaux, F. (2011). *Introduction to Stochastic Programming Springer*.
- Boyd, S., dan Vandenberghe, L. (2004). *Group Theory and Physics Cambridge University Press*.
- Chairul S, Azmi H, Md. Yusoff J, (2002), Optimasi Ukuran Lot Menggunakan Pengulangan Ganda Algoritma Genetika. *Proceeding Seminar Nasional Teknik Industri dan Manajemen Produksi TIMP*, 207- 212, Surabaya.
- Cheng, T. C. E., Gupta, J. N. D., dan Kuo, R. J. (2005). *Scheduling in a Job Shop Environment: A Review. European Journal of Operational Research*, 164(3), 689-709.
- Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton University Press.
- Gen, M. dan Cheng, R. (2000), *Genetic Algorithms and Engineering Design*, John Wiley and Sons.
- Glover, F. (1989). *Tabu Search—Part I. ORSA Journal on Computing*, 1(3), 190-206.

- Graham, R. L. (2000). *An Efficient Algorithm for Scheduling Independent Tasks*. Operations Research, 28(1), 156-165.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Ilhamsyah, M., Mangkudjaja, R., Hertiana, S. N. (2016). Simulasi dan Uji Kinerja Algoritma johnson untuk Penentuan Rute Terbaik pada Jaringan Software Defined Network. Jurnal Sistem Komputer, 6(2).
- Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley.
- Kirkpatrick, S., Gelatt, C. D., dan Vecchi, M. P. (1983). *Optimization by Simulated Annealing*. Science, 220(4598), 671-680.
- Koza, J., (2001). Genetic algorithms. *In Genetic Programming: On the Programming of Computers by Means of Natural Selection*. 1-20, MIT Press.
- Kubiak, W., Gajewski, R. (2018). *The impact of scheduling on production efficiency*. Journal of Manufacturing Systems.
- Lee, S., Kim, H., dan Park, J. (2018). "Comparative Analysis of Genetic Algorithm and NEH Heuristic for Job Shop Scheduling". Computers and Industrial Engineering, 115, 1-10.
- Lukas, S., Anwar, T., Yuliani, W. (2005). Penerapan Algoritma Genetika untuk Traveling Salesman Problem dengan Menggunakan Metode Order Crossover dan Insertion Mutation. Seminar Nasional Aplikasi Teknologi Informasi 2005.ISBN: 979-756-061-6.
- Mubarok, A.Y., Chotijah, U. (2021). Penerapan Algoritma Genetika untuk Mencari Optimasi Kombinasi Jalur Terpendek dalam Kasus Travelling Salesman Problem. Jurnal Teknologi Terpadu. 7(2): 77-82.
- Nocedal, J., dan Wright, S. J. (2006). *Numerical Optimization*. Springer.
- Papadimitriou, C. H., dan Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall.
- Pinedo, M. (2016). *Scheduling: Theory, Algorithms, and Systems*. Springer.

- Sari, D., Rahardjo, B. (2022). *Optimizing job shop scheduling using hybrid approaches*. International Journal of Production Research.
- Silberschatz, A., Galvin, P. B., dan Gagne, G. (2018). *Operating System Concepts*. Wiley.
- Suprpto, Widiyaningsih, E. N., dan Munawaroh, R. (2022). Edukasi Tentang Pemanfaatan Amilum Singkong (Tapioka) untuk Kesehatan dan Pembuatan Makanan Jajanan.
- Toth, P., dan Vigo, D. (2001). *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*.
- Wang, Y., Xu, Z., dan Wang, H. (2014). *A Survey of Job Shop Scheduling Problems*. Journal of Scheduling, 17(3), 297-308.
- Wargiono. (1993). Bertanam Ubi-ubian. Jakarta: Penebar Swadaya.
- Winston, W. L. (2004). *Operations Research: Applications and Algorithms*. (4th ed.). Duxbury Press.
- Yang, S., Zhang, Y., dan Liu, H. (2020). "A Novel Genetic Algorithm for Job Shop Scheduling Problem". Soft Computing, <http://cs.felk.cvut.cz/xobitko/ga/intro.html>
- Zhang, Y., Wang, J., dan Chen, L. (2019). "Performance Analysis of Genetic Algorithms for Job Shop Scheduling". Journal of Intelligent Manufacturing, 30(5), 1987-2000.
- Zhou, H., et al. (2018). *Smart manufacturing and industry 4.0: The digital transformation of the manufacturing industry*. Journal of Manufacturing Technology Management.