PENGEMBANGAN ALGORITMA GENETIKA UNTUK OPTIMASI PENJADWALAN PRODUKSI *JOB SHOP* FLEKSIBEL (FJSP)

UJIAN KUALIFIKASI

RIZKY KHAIRUL UMAM

99223137

PROGRAM DOKTOR TEKNOLOGI INFORMASI
UNIVERSITAS GUNADARMA
JUNI 2024

DAFTAR ISI

HALAMAN JUDUL i

BAB I PENDAHULUAN

1.1 Latar Belakang..... 1
1.2 Perumusan Masalah 6
1.3 Batasan Masalah 6
1.4 Tujuan Penelitian 7
1.5 Kontribusi Penelitian 8

BAB II TINJAUAN PUSTAKA

2.1 Penjadwalan..... 9
2.1.1 Tujuan Penjadwalan 10
2.1.2 Terminologi Dalam Penjadwalan 11
2.1.3 Klasifikasi Penjadwalan Produksi 12
2.1.4 Elemen Penjadwalan 14
2.1.5 Penjadwalan *Job Shop* 14
2.1.6 Masalah Penjadwalan 15
2.2 Metode Penyelesaian Masalah Penjadwalan Produksi..... 16
2.2.1 Tipe Heuristik Klasik 16
2.2.2 Tipe Heurtistik Modern 18
2.3 Algoritma Genetika 20
2.3.1 Teknik Pengkodean (Encoding Techniques) 21
2.3.2 Pembangkitan Populasi Awal Kromosom 23
2.3.3 Evaluasi Fitness 24
2.3.4 Seleksi 24
2.3.5 Crossover (Penyilangan) 26
2.3.6 Mutasi 30
2.3.7 Parameter Algoritma Genetika 31

2.4 Penelitian Terdahulu 32

BAB III METODOLOGI

3.1 Tahapan Penelitian 38
3.2 Identifikasi Masalah 38
3.3 Pengumpulan Data 42
3.4 Pengolahan Data 42
3.4.1 Algoritma Genetika Konvensional 42
3.4.2 Pengembangan Algoritma Genetika 44
3.5 Hasil dan Pembahasan 49
3.6 Kesimpulan dan Saran 49
3.7 Rencana Jadwal Kegiatan 50

Daftar Pustaka

BAB I
PENDAHULUAN

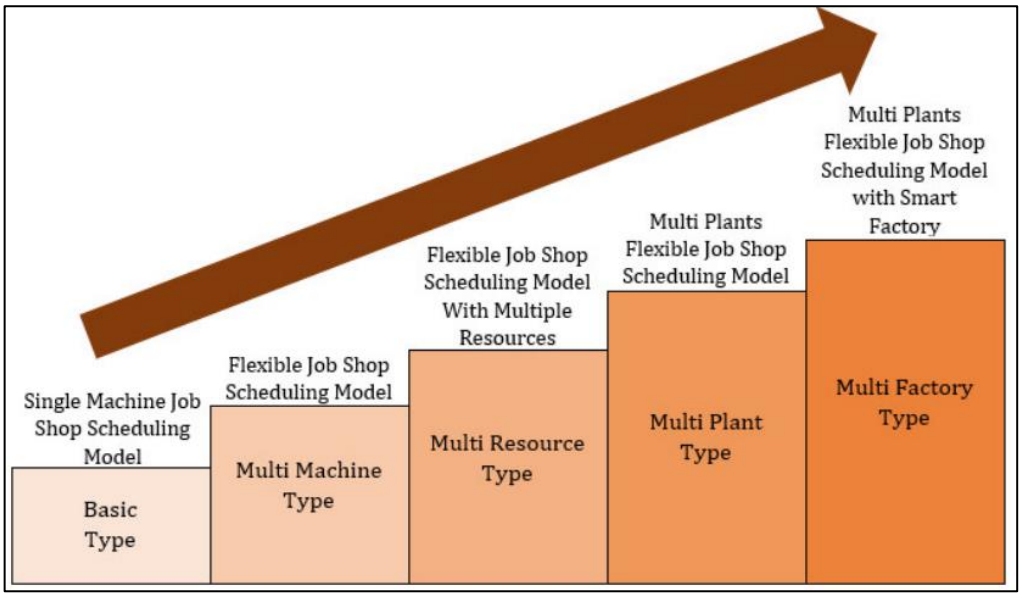
1.1 Latar Belakang

Sektor manufaktur menggunakan teknik manufaktur untuk mengubah sumber daya tertentu menjadi barang yang dapat digunakan masyarakat. Pertumbuhan sektor manufaktur juga merupakan salah satu indikator kunci pembangunan nasional (Li & Xia, 2022). Penjadwalan memainkan peran penting dalam sektor manufaktur pada proses produksi, karena dapat memaksimalkan efisiensi sekaligus meminimalkan berbagai biaya, waktu tunggu dan waktu siklus (Liaqait et al., 2021). Penjadwalan produksi merupakan salah satu proses pengambilan keputusan yang penting dalam lingkungan manufaktur, untuk menjaga bisnis agar tetap kompetitif dan penggunaan sumber daya menjadi efektif (Park et al., 2021). Dengan penjadwalan produksi sumber daya terbatas dialokasikan berdasarkan waktu berbagai operasi untuk memberikan keluaran yang diinginkan pada waktu yang diinginkan (Defersha & Rooyani, 2020).

Penjadwalan produksi juga merupakan salah satu masalah optimasi yang penting dan krusial dalam perencanaan dan manajemen sistem manufaktur modern (Wang & Zhu, 2021). Penjadwalan produksi tidak dilakukan dengan benar, berpotensi menimbulkan masalah dalam proses produksi. Menurut Park et.al., (2021), sebagian besar masalah penjadwalan produksi merupakan masalah optimasi kombinatorial *Non-deterministic Polynomial* (NP-hard), sehingga sulit untuk menemukan jadwal ideal dalam waktu polinomial.

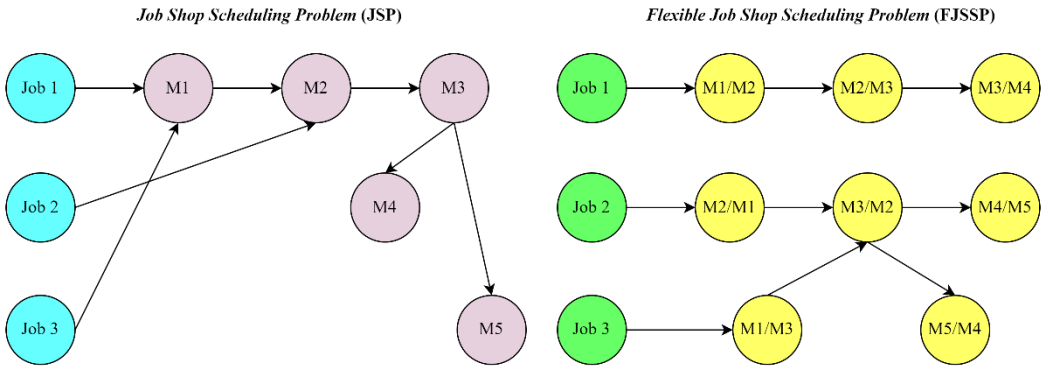
Salah satu masalah penjadwalan yang paling umum digunakan saat ini adalah *Job-shop Scheduling Problem* (JSP) atau masalah penjadwalan *job-shop* dimana serangkaian operasi ditugaskan kepada sekelompok mesin dengan batasan setiap operasi hanya dapat diselesaikan oleh satu mesin dari setiap operasi yang ada (Zhang et al., 2019). (Liaqait et al., 2021) mengklasifikasikan JSP sebagai masalah optimasi NP-hard yang menunjukkan mesin yang berbeda-beda ditugaskan untuk berbagai pekerjaan dengan meminimalkan kriteria yang telah didefinisikan

sebelumnya. Secara umum, model JSP mengharuskan setiap tugas memiliki jalur yang ditetapkan dan diselesaikan pada satu mesin yang telah dikonfigurasi sebelumnya (Huang et al., 2018). Berbagai model JSP telah diusulkan untuk meningkatkan efisiensi operasional fasilitas produksi. Gambar 1.1 menunjukkan klasifikasi model JSP dengan berbagai kompleksitas berdasarkan variasi parameter yang digunakan. Pengembangan model-model JSP pada Gambar 1.1 memperlihatkan adanya kecenderungan untuk mengembangkan pabrik pintar yang sejalan dengan industri 4.0.



Gambar 1. Model dan Klasifikasi Penjadwalan Job Shop
Sumber: (Liaqait et al., 2021)

Perluasan dan generalisasi dari JSP konvensional adalah *Flexible Job-shop Scheduling Problem* (FJSP) atau masalah penjadwalan *job-shop* fleksibel (Huang et al., 2018). Perluasan ini terjadi untuk memenuhi kondisi dimana pemilihan mesin dapat disesuaikan berdasarkan kondisi yang sedang belangsung. Selain itu, banyak mesin dapat memproses satu atau lebih operasi (Liu et al., 2021). Mengartikan bahwa banyak mesin mampu memproses setiap operasi. Hal ini dikenal sebagai FJSP (Zhang et al., 2019). Perbedaan JSP dengan FJSP ditunjukkan pada Gambar 1.2.



Gambar 1.2 Perbedaan JSSP dan FJSSP
(Sumber: Olah Data, 2024)

Berdasarkan Gambar 1.2 JSP memiliki jalur penjadwalan yang tetap dan tidak fleksibel. Setiap pekerjaan harus mengikuti urutan operasi yang sudah ditentukan pada mesin tertentu. Kemudian dapat dilihat bahwa setiap pekerjaan (Job1, Job2, Job3) memiliki urutan mesin (M1, M2, M3, dst) yang tetap tanpa adanya alternatif mesin. Sedangkan FJSP memberikan fleksibilitas dengan memungkinkan setiap operasi dilakukan pada beberapa pilihan mesin. Hal ini memungkinkan penjadwalan yang lebih adaptif dan efisien. Kemudian setiap pekerjaan memiliki beberapa opsi mesin untuk setiap tahap (M1/M2, M2/M3, dst), menunjukkan fleksibilitas dalam pemilihan mesin yang tersedia.

FJSP yang sering digunakan dalam industri manufaktur dan proses diskrit, merupakan topik utama dalam penelitian terkini pada bidang manufaktur cerdas dan otomasi industri. Bisnis dapat meningkatkan efisiensi ekonominya dan mendapatkan komando, kendali, dan pengorganisasian sumber daya yang wajar dengan menemukan solusi terhadap tantangan FJSP (Jiang et al., 2023). FJSP memiliki kegunaan yang signifikan dalam berbagai industri, termasuk tekstil, semikonduktor, perakitan otomotif, dan manufaktur, di mana kumpulan mesin dapat melakukan berbagai tugas (Huang et al., 2018). Mendapatkan jadwal yang bisa diterapkan untuk meminimalkan fungsi tujuan yang diberikan, FJSP dapat dibagi menjadi dua submasalah: yang pertama adalah submasalah perutean, yang menugaskan mesin dari mesin alternatif yang diatur ke suatu operasi, dan yang

kedua adalah penjadwalan sub-masalah, yang terdiri dari operasi pengurutan pada semua mesin yang dipilih dalam satu operasi (Zhang et al., 2019).

Ketika menangani FJSP, ada beberapa fungsi objektif yang perlu dipertimbangkan. Makespan, waktu penyelesaian rata-rata, waktu aliran maksimum, total keterlambatan tertimbang, keterlambatan rata-rata, keterlambatan maksimum, jumlah pekerjaan yang terlambat, dan total beban kerja mesin merupakan beberapa fungsi objektif tersebut (Amjad et al., 2018). Tujuan paling umum dari FJSP adalah untuk mengurangi makespan, atau waktu yang dibutuhkan untuk menyelesaikan pekerjaan secara keseluruhan.

Sejumlah peneliti telah mencoba memecahkan FJSP dalam beberapa tahun terakhir menggunakan algoritma heuristik, algoritma pembelajaran, dan pemrograman bilangan bulat, dengan beberapa keberhasilan (Liang et al., 2022). Menurut Zhang dkk. (2019), teknik heuristik dan meta-heuristik dapat dengan mudah menghasilkan solusi yang mendekati optimal dan menghasilkan jadwal yang cukup dapat diterima dalam waktu komputasi yang baik. Penyelesaian FJSP, berbagai teknik metaheuristik telah digunakan (Liu et al., 2021). Algoritma genetika, optimasi koloni semut, optimasi kawanan partikel, algoritma serigala abu-abu, algoritma pencarian tabu, dan metode optimasi cerdas lainnya saat ini sering digunakan untuk menyelesaikan FJSP (Jiang et al., 2023).

Melihat beberapa dekade terakhir, algoritma genetika telah mendapatkan popularitas sebagai pendekatan metaheuristik untuk menyelesaikan FJSP (Park et al., 2021). Kemampuan pencarian globalnya yang kuat, algoritma genetika merupakan algoritma meta-heuristik yang secara efisien menunjukkan kinerja yang luar biasa untuk masalah penjadwalan (Wang & Zhu, 2021). Algoritma genetika adalah salah satu strategi evolusioner yang paling sukses untuk menyelesaikan FJSP karena meniru evolusi biologis dan memiliki ketahanan yang kuat, pemrosesan yang sederhana, ekstensibilitas yang kuat, dan kemampuan pencarian acak yang cepat (Wu et al., 2018).

Pendekatan algoritma genetika dilakukan oleh penelitian (Wang & Zhu, 2021) untuk menyelesaikan FJSP dengan waktu penyiapan yang bergantung pada urutan dan waktu jeda pekerjaan. Tujuan dari algoritma ini adalah untuk

meminimalkan makespan, dan metode ini menunjukkan kinerja yang baik ketika algoritma genetika melakukan pencarian global yang kuat. Algoritma genetika berbasis inisialisasi ulang diusulkan oleh penelitian (Li & Xia, 2022) untuk meminimalkan makespan, dan menunjukkan keberhasilan yang baik. Penelitian (Wu et al., 2018) mengusulkan algoritma genetika dengan tujuan menurunkan makespan untuk menyelesaikan FJSP ketika sumber daya terbatas. Fungsi tujuan meminimalkan makespan, penelitian (Luo et al., 2018) menggunakan algoritma genetika yang disempurnakan untuk menangani FJSP.

Mengurangi waktu penyelesaian secara keseluruhan, atau makespan, penelitian (Zhang et al., 2019) membahas FJSP dengan waktu perpindahan. Algoritma genetika adaptif berdasarkan kemiripan individu digunakan dalam penelitian (Liang et al., 2022) untuk menangani FJSP dengan tujuan optimasi untuk menurunkan makespan dan konsumsi energi. Menangani kesulitan FJSP dengan tujuan optimasi menurunkan makespan atau waktu penyelesaian secara keseluruhan, penelitian (Liu et al., 2021) menyarankan untuk menggunakan algoritma genetika hibrida.

Meskipun algoritma genetika memiliki kekurangan, algoritma ini merupakan teknik optimasi yang populer untuk menyelesaikan FJSP. Menurut penelitian sebelumnya, terkait algoritma genetika konvensional, variasi populasi menurun seiring iterasi algoritma, dan individu-individu tertentu dapat menjadi hampir sama, yang mengakibatkan stasis evolusioner populasi (Meng et al., 2023). Menurut Huang dan Yang (2019), algoritma genetika konvensional menunjukkan kecepatan konvergensi yang lambat dan berpotensi konvergensi dini.

Kekurangan dari algoritma genetika adalah kapasitasnya yang terbatas untuk pencarian lokal, dalam proses mengatasi masalah, rentan terhadap konvergensi prematur dan pengoptimalan lokal (Liang et al., 2022). Kemudian, kemampuan pencarian lokal yang tidak memadai yang disebabkan oleh tidak adanya teknik pencarian lingkungan sehingga memperlambat pengoptimalan (Wang & Zhu, 2021). Menggabungkan atau melakukan hibridisasi dengan algoritma yang berbeda, dapat mengurangi kekurangan tersebut.

Menurut penelitian sebelumnya tujuan dari sebagian besar fungsi objektif yang digunakan untuk menyelesaikan tantangan FJSP adalah untuk meminimalkan makespan, atau waktu operasi secara keseluruhan. Namun demikian, implementasi sistem produksi mengharuskan penentuan fungsi objektif yang banyak atau multi-objektif. Akibatnya, penelitian tentang masalah penjadwalan *job shop* fleksibel multi-tujuan (MOFJSP) menjadi semakin populer dan penting (Huang et al., 2018). Berdasarkan penelitian sebelumnya sebagian besar penelitian yang ada tidak mempertimbangkan waktu transportasi dan waktu setup dalam menyelesaikan FJSP. Kedua hal tersebut sangat penting karena setiap operasi melakukan perpindahan ke mesin lain untuk operasi berikutnya, sementara mesin memerlukan persiapan untuk memproses operasi berikutnya.

Pengembangan algoritma genetika untuk mengoptimalkan *Multi Objective Flexible Job-shop Scheduling Poblem* (MOFJSP) akan mengatasi tantangan yang disebutkan di atas dengan mengembangkan algoritma genetika untuk mengurangi kekurangan dari metode algoritma genetika konvensional. Kemudian, meminimalkan makespan (Cm), meminimalkan waktu kerja maksimum mesin (Wm), dan meminimalkan total waktu kerja semua mesin (Wt) adalah fungsi objektif yang akan digunakan kemudian mempertimbangkan waktu transportasi dan waktu setup dalam mengoptimasi FJSP. Tujuan dari penelitian ini adalah untuk membantu menyelesaikan masalah optimasi FJSP, khususnya yang berkaitan dengan penjadwalan produksi di industri manufaktur.

1.2 Rumusan Masalah

Mengacu pada topik penelitian yang telah diuraikan diatas, maka untuk mencapai tujuan penelitian ini, berikut terdapat beberapa masalah yang ingin dipecahkan.

a. Bagaimana mengembangkan algoritma genetika konvensional untuk mengatasi kelemahan algoritma tersebut dalam mengoptimasi *Flexible Job-shop Scheduling Problem* (FJSP)?
b. Bagaimana perbandingan hasil dari pengembangan algoritma tersebut terhadap algoritma genetika konvensional dalam mengoptimasi fungsi objektifnya?

1.3 Batasan Masalah

Pembatasan masalah bertujuan untuk membatasi pembahasan pada pokok masalah penelitian. Batasan Masalah dalam penelitian ini terdiri dari:

1. Pekerjaan tidak bergantung satu sama lain, dan semua pekerjaan tersedia di awal.
2. Pekerjaan mempunyai prioritas yang sama untuk mulai diproses.
3. Operasi yang sedang berjalan pada suatu mesin tidak dapat dihentikan sebelum operasi tersebut selesai.
4. Sebuah operasi hanya dapat dilakukan pada satu mesin di antara subset mesin alternatif yang telah ditentukan sebelumnya, namun operasi tersebut harus dilakukan hanya oleh satu mesin dalam satu waktu.
5. Sebuah operasi tidak dapat dilakukan sebelum operasi yang mendahuluinya selesai.
6. Sebuah mesin dapat melakukan paling banyak satu operasi dalam satu waktu.
7. Waktu operasi bervariasi, tergantung pada mesin yang digunakan dan nilainya diketahui.
8. Untuk dua operasi berurutan pada mesin yang berbeda dalam pekerjaan yang sama, terdapat waktu transportasi antara kedua mesin. Jika kedua operasi tersebut dilakukan pada mesin yang sama, maka waktu transportasi adalah 0.
9. Setiap operasi memiliki waktu setup yang berbeda-beda tergantung pada mesin yang digunakan. Jika dua operasi berurutan pada mesin yang sama dari satu pekerjaan, maka waktu setup adalah 0.

1.4 Tujuan Penelitian

Tujuan penelitian merupakan tujuan yang akan dicapai dalam sebuah penelitian. Tujuan penelitian ini adalah sebagai berikut:

a. Pengembangan algoritma genetika konvensional dalam mengoptimasi *Flexible Job-shop Scheduling Problem* (FJSP).
b. Perbandingan hasil pengembangan algoritma genetika dengan algoritma genetika konvensional dalam mengoptimasi fungsi objektifnya.

1.5 Kontribusi Penelitian

Dalam penelitian ini dapat memberikan kontribusi atau novelty keilmuan yaitu memberikan sumbangsih dengan pengembangan algoritma genetika yang dilakukan dapat memberikan hasil optimasi secara akurat dan efektif dalam mengatasi masalah penjadwalan *job shop* fleksibel (FJSP). Mengurangi waktu keseluruhan operasi, meminimalkan waktu kerja maksimum dari sebuah mesin dan meminimalkan total waktu kerja seluruh mesin dalam satuan waktu.

BAB II
TINJAUAN PUSTAKA

2.1 Penjadwalan

Penjadwalan (*scheduling*) adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas, serta pengalokasian sumber daya pada waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada. penjadwalan dapat didefinisikan sebagai proses pengalokasian sumber daya untuk mengerjakan sekumpulan tugas dalam jangka waktu tertentu dengan 2 arti penting sebagai berikut (Baker & Trietsch, 2009):

1. Penjadwalan merupakan suatu fungsi pengambilan keputusan untuk membuat atau menentukan jadwal.
2. Penjadwalan merupakan suatu teori yang berisi sekumpulan prinsip dasar, model, teknik, dan kesimpulan logis dalam proses pengambilan keputusan yang memberikan pengertian dalam fungsi penjadwalan.

Penjadwalan merupakan salah satu kegiatan penting dalam perusahaan. Penjadwalan adalah pengaturan waktu dari suatu kegitan, yang mencakup kegiatan mengalokasikan fasilitas, peralatan maupun tenaga kerja, dan menentukan urutan pelaksanaan bagi suatu kegiatan operasi (Herjanto, 2017).

Berdasarkan beberapa definisi di atas, dapat dikatakan bahwa penjadwalan merupakan proses pengalokasian fasilitas, peralatan maupun tenaga kerja, dan sumber daya produksi yang berupa material, mesin dan operator untuk menentukan dimulainya operasi sehingga proses produksi dapat berjalan dengan lancar. Dalam penjadwalan, harus menjalankan sekumpulan tugas dalam jangka waktu tertentu, dan keputusan yang dibuat didalamnya meliputi 3 hal, yaitu pengurutan pekerjaan, menentukan waktu mulai dan selesai pekerjaan, serta mengurutkan proses suatu pekerjaan.

2.1.1 Tujuan Penjadwalan

Tujuan Penjadwalan adalah untuk mengurangi waktu keterlambatan dari batas waktu yang ditentukan agar perusahaan memenuhi batas waktu yang telah disetujui dengan konsumen. Penjadwalan juga dapat meningkatkan produktifitas mesin dan mengurangi waktu menganggur. Jika produktifitas mesin meningkat maka waktu menganggur berkurang, sehingga secara tidak langsung perusahaan dapat mengurangi biaya produksi. Semakin baik suatu penjadwalan semakin menguntungkan juga bagi perusahaan dan bisa menjadi acuan untuk meningkatkan keuntungan dan strategi bagi perusahaan dalam pemuasan pelanggan. Berikut beberapa tujuan yang ingin dicapai dengan dilaksanakannya penjadwalan adalah sebagai berikut (Nasution, 1999):

1. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggunya, sehingga total waktu proses dapat berkurang dan produktivitas produksi meningkat.
2. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan yang menunggu dalam antrian ketika sumber daya yang ada sedang mengerjakan tugas yang lain.
3. Mengurangi beberapa kelemahan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga akan meminimasi biaya keterlambatan (penalty cost).
4. Membantu pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan, sehingga penambahan biaya yang mahal dapat dihindarkan.

Penjadwalan bertujuan meminimalkan waktu proses, waktu tunggu langganan, dan tingkat persediaan, serta penggunaan yang efisien dari fasilitas, tenaga kerja dan peralatan. Tujuan penjadwalan sebagai berikut (Schroeder, 2020):

1. Untuk mencapai efisiensi yang tinggi.
2. Menekan persediaan serendah mungkin.
3. Meningkatkan pelayanan terhadap pelanggan.

2.1.2 Terminologi Dalam Penjadwalan

Istilah yang sering digunakan pada penjadwalan sistem produksi sangat beragam. Pengertian dari istilah-istilah tersebut akan dijelaskan sebagai berikut (Utama, 2023) :

1. *Ready time* atau waktu siap (ri). Diartikan sebagai waktu sebuah job (pekerjaan) ke-i siap untuk diproses.
2. *Waiting time* atau waktu menunggu (W1), Waktu dimana pekerjaan i menunggu operasi pendahulunya selesai diproses sehingga pekerjaan i bisa mendapatkan gihrannya.
3. *Set up time* atau waktu persiapan, Waktu untuk menyiapkan kebutuhan sebelum job diproses.
4. *Arrival time* atau waktu kedatangan (a1), Waktu datang sebuah job di shop floor.
5. *Delivery date* atau waktu pengiriman, Waktu pengiriman sebuah job dari shop floor ke kegiatan selanjutnya atau waktu pengiriman produk jadi ke konsumen.
6. *Processing time* atau waktu proses (ti), Waktu untuk menyelesaikan suatu pekerjaan. Waktu untuk setup sudah termasuk didalamanya. Bisa juga disebut sebagai waktu yang dibutuhkan untuk memproses job.
7. *Due-date* atau tenggang waktu (di), Batas waktu sebuah pekerjaan (job) harus diselesaikan
8. *Slack time* (SLi), Wakt tersisa yang ditimbulkan oleh nilai waktu proses yang lebih kecil dari due-date.
9. *Flow time* (Fi), Waktu sebuah peerjaan dari awal kedatangan sampai dengan pekerjaan tersebut selesai diproses. Flow time juga bisa diartikan sebagai penjumlahan waktu proses dengan waktu meunggu sebelum pekerjaan tersebut diproses. Atau bisa diartikan juga sebagai waktu siap job untuk dikerjakan sampai dengan job selesai dikerjakan.
10. *Lateness (Li),* Waktu selisih antara due date nya (d,) dengan completion time. Pekerjaan yang selesai sebelum due date yang telah ditentukan akan bernilai lateness positif dan mempunyai keterlambatan negatif. Sedangkan pekerjaan yang selesai melebihi due date yang telah ditentukan akan bemilai lateness negatif dan mempunyai keterlambatan positif.

11. *Completion time* (Ci), Waktu penyelesaian suatu pekerjaan. Waktu penyelesaian tersebut terhitung dari pekerjaan telah tersedia (t = O) sampai dengan pekerjaan tersebut selesai dikerjakan. Waktu tersebut menunjukkan rentang antara waktu pekerjaan mulai diproses hingga pekerjaan tersebut selesai diproses.
12. *Tardiness* (Ti), Tardiness merupakan ukuran keterlambatan. Nilai tardiness akan positif ketika pekerjaan selesai sebelum due date yang telah ditetapkan maka pekeriaan tersebut memiliki keterlambatan yang negatif. Pekerjaan akan memiliki nilai tardiness positif apabila pekerjaan selesai melebihi due date, sehingga akan mempunyai keterlambatan yang positif. Berdasarkan hal tersebut dapat disimpulkan bahwa tardiness merupakan lateness yang bernilai positif.
13. *Makespan* (M), Waktu penyelesaian pekerjaan secara keseluruhan. Waktu penyelesaian tersebut terhitung dari job urutan pertama mulai diproses hingga job urutan terakhir selesai diproses pada mesin terakhir.

2.1.3 Klasifikasi Penjadwalan Produksi

Kriteria model penjadwalan yang diterapkan pada lantai produksi dijelaskan menjadi beberapa aspek yaitu (Utama, 2023):

1. Penjadwalan produksi bisa diklasifikasikan berdasarkan mesin (surnber daya) yang digunakan:

a. Penjadwalan pada single machine (mesin tunggal)
b. Penjadwalan pada parallel machine (mesin jamak)

2. Penjadwalan produksi bisa diklasifikasikan berdasarkan pola aliran pada proses:

a. Penjadwalan flow shop

Pola aliran pada penjadwalan ini berurutan dari satu mesin ke mesin yang lainnya. Pada pola tipe ini setiap job (pekerjaan) mempunyai aliran produksi yang sama atau disebut dengan pure flow shop. Selain pure flow shop terdapat 2 aliran flow shop yang berbeda lagi yaitu, aliran flow shop yang mengerjakan job bervariasi dan aliran flow shop yang memiliki job yang tidak harus diproses pada semua mesin atau biasa disebut dengan general flow shop.

b. Penjadwalan *job shop*

Pola aliran pada penjadwalan ini bisa berbeda-beda untuk setiap job (pekerjaan). Rute proses operasi pada setiap mesin tersusun spesifik. Aliran proses yang tidak searah membuat job yang masuk dan keluar dari sebuah mesin bisa berupa job dalam proses atau job jadi.

c. Penjadwalan proyek

Pola pada penjadwalan ini memiliki aliran yang berbeda-beda dan spesifik untuk setiap job (pekerjaan).

3. Penjadwalan produksi bisa diklasifikasikan berdasarkan pola kedatangan job, yaitu:

a. Penjadwalan statis

Pada penjadwalan ini setiap job datang pada waktu yang sama dan siap untuk diproses dimesin yang ready (menganggur).

b. Penjadwalan dinamis

Pada penjadwalan ini setiap job datang pada waktu yang tidak menentu.

4. Penjadwalan produksi bisa diklasifikasikan berdasarkan sifat informasi yang diterima, yaitu:

a. Penjadwalan deterministic

Penjadwalan ini memiliki informasi yang pasti. Informasi ini bisa terkait dengan job (pekerjaan) dan sumber daya (mesin) seperti waktu proses dan waktu datang sebuah job.

b. Penjadwalan stokastik

Penjadwalan ini memiliki informasi yang tidak pasti, tapi mempunyai kecenderungan tertentu atau tergolong pada distribusi probabilitas tertentu. Informasi ini bisa terkait dengan job (pekerjaan), yaitu waktu kedatangan, due date (batas waktu penyelesaian), perbedaan prioritas di antara job yang dijadwalkan, jumlah operasi, serta waktu proses setiap job. Ada juga informasi terkait karakteristik mesin, seperti fleksibilitas dan efisiensi penggunaan mesin, jumlah mesin yang berbeda untuk job yang berbeda, kapasitas yang tersedia.

2.1.4 Elemen Penjadwalan

Elemen penting pada proses penjadwalan adalah sebagai berikut (Utama, 2023)**:**

1. Job

Definisi job pada penjadwalan adalah sebuah pekerjaan yang diproses untuk menghasilkan suatu produk. Ada beberapa proses operasi yang harus dilakukan (minimal 1 operasi) untuk menjadikan job sebagai sebuah produk. Bagian manajemen perencanaan mengirimkan informasi terkait job yang dipesan (order) pelanggan kepada lantai produksi untuk diproses. Setiap job yang dikirim ke lantai produksi memiliki informasi untuk bagian engineering, seperti tenggang waktu penyelesaian job, waktu job tersebut bisa mulai dikerjakan

2. Operasi

Definisi operasi pada penjadwalan adalah sebuah himpunan proses dari sebuah job. Sebuah operasi diurutkan menggunakan teknik tertentu untuk menyelesaikan sebuah job. Sebuah operasi bisa dikerjakan ketika operasi pendahulunya selesai dikerjakan lebih dulu. Urutan sebuah pekerjaan dan jenis mesin untuk setiap operasi diinformasikan dalam sebuah matriks routing. Masing-masing operasi selalu memiliki waktu proses.

3. Mesin

Mesin merupakan sebuah resource (sumber daya) yang digunakan untuk menyelesaikan sebuah job. Sebuah mesin hanya bisa melakukan satu tugas pada periode waktu tertentu.

2.1.5 Penjadwalan *Job Shop*

Job shop scheduling secara umum lebih dikenal sebagai shop floor control, yang merupakan kegiatan penyusunan input (memasang yang diperlukan) menjadi output (produk). Penjadwalan job shop adalah pengurutan pekerjaan untuk lintas produk yang tidak beraturan (tata letak pabrik berdasarkan proses). Penjadwalan pada proses produksi tipe job shop lebih sulit dibandingkan penjadwalan flow shop. Hal ini disebabkan oleh 3 alasan, yaitu (Suradi, 2023):

1 Job shop menangani variasi produk yang sangat banyak, dengan pola aliran yang berbeda-beda melalui work center.
2 Peralatan pada job shop digunakan secara bersama-sama oleh bermacam-macam order dalam prosesnya, sedangkan peralatan pada flow shop digunakan khususnya hanya untuk satu jenis produk.
3 Job-job yang berbeda mungkin ditentukan Oleh prioritas yang bet-beda pula, hal ini mengakibatkan order tertentu yang dipilih harus diproses seketika pada Saat order tertentu yang dipilih harus diproses seketika pada Saat order tersebut ditugaskan pada suatu work center.

Sedangkan pada flow shop tidak terjadi masalah seperti diatas karena keseragaman output yang diproduksi untuk persediaan. Prioritas order pada flow shop dipengaruhi terutama pada pengirimannya dibandingkan tanggal pemerosesan.

2.1.6 Masalah Penjadwalan

Masalah penjadwalan adalah masalah pengalokasian pekerjaan ke mesin, pada kondisi mesin mempunyai kapasitas dan jumlah terbatas. Secara umum masalah penjadwalan dapat dijelaskan sebagai n job (J1, J2,…,Jn) yang harus diproses di m mesin (M1, M2,…,Mn). Waktu yang diperlukan untuk memproses pekerjaan J1 pada mesin M adalah P setiap job harus diproses tanpa dihentikan selama waktu proses p mesin hanya dapat menangani satu job pada saat yang sama, dan secara terus menerus tersedia sejak waktu nol (time zero). Pemecahan masalah yang diinginkan adalah mendapatkan jadwal yang optimal, yaitu menyelesaikan semua pekerjaan dengan mendapatkan jadwal yang optimal yaitu menyelesaikan semua pekerjaan dengan adanya keterbatasan kapasitas dan ketersediaan mesin dengan memenuhi fungsi tujuannya. Pada penelitian ini Jumlah kapasitas mesin pada pabrik masih tidak bisa difungsikan dengan baik, mengakibatkan waktu mengganggur yang cukup besar.

Priority Rules untuk menyelesaikan masalah job shop banyak cara yang dapat ditempuh diantaranya dengan metode matematis, heuristic dan simulasi. Salah satunya adanya priority rules. Biasaanya priority rules ini dipakai baik untuk

menentukkan prioritas. Penjadwalan produksi adalah suatu kegiatan yang sangat penting pada perusahaan manufaktur. Model terdahulu, yakni flow shop scheduling juga dapat diaplikasikan pada perusahaan atau kegiatan jasa. Tetapi seperti telah disebutkan, urutan proses di mesin tiap job (pekerjaan) haruslah sama. Sebagai contoh, kegiatan foto kopi, periklanan, restoran, dan lain-lain. Jika urutan proses di mesin berbeda, maka model penjadwalan yang tepat adalah job shop scheduling problem. Pada masalah penjadwalan job shop, ada sejumlah job dan sejumlah mesin dengan setiap job dapat memiliki urutan proses yang berbeda dengan job yang lainnya. Hal inilah yang membuat masalah ini lebih rumit dibandingkan dengan masalah penjadwalan flow shop. Sebagaimana masalah penjadwalan flow shop, masalah job shop dapat ditujukan untuk meminimumkan makespan, meminimumkan keterlambatan, meminimumkan total biaya (termasuk biaya penalti), dan lain sebagainya. Dengan keleluasaan ini, banyak masalah pen'adwalan dapat dimodelkan sebagai job shop scheduling problem. Model ini terutama dapat diaplikasikan untuk perusahaan manufaktur yang menganut sistem make to order, seperti pencetakan mold, bengkel bubut, dsb. Selain ittl, sebagaimana disebutkan di atas, model ini juga dapat digunakan untuk menyelesaikan masalah penjadwalan pada bidang Jasa (Suradi, 2023).

2.2 Metode Penyelesaian Masalah Penjadwalan Produksi

Masalah penjadwalan produksi dapat diselesaikan dengan menggunakan metode heuristik yang terdiri dari 2 jenis, yaitu tipe heuristik klasik dan tipe heuristik modern (Metaheuristik) (Hasanudin, 2011).

2.2.1 Tipe Heuristik Klasik

Heuristik (heuristics) atau aproksimasi suatu teknik yang didesain untuk memecahkan masalah dengan sedikit mengabaikan apakah solusinya bisa dibuktikan benar, tetapi biasanya menghasilkan solusi yang bagus, dalam arti optimal atau mendekati optimal. Heuristik dimaksudkan untuk mendapatkan hasil yang secara komputasi lebih cepat dengan konsekuensi mengurangi kepresisian atau akurasi. Jadi kecepatan perhitungan biasanya lebih baik (dibanding optimasi

eksak) dengan sidikit mengorbankan akursi. Walaupun pada kenyataanya solusinya bisa juga mempunyai akurasi yang tinggi. Pendekatan heuristik biasanya sangat spesifik untuk problem tertentu. Sehingga, diperlukan algoritma yang lain untuk problem yang berbeda. Tentu saja ini kurang menguntungkan (Santosa, 2011).

Algoritma ini menyusun satu per satu solusi dari masalah penjadwalan. Mulai dari nol, algoritma-algoritma ini memilih meisn-mesin atau job-job atau operasi-operasi mana yang harus dijadwalan terlebih dahulu. Algoritma heuristik klasik yang sering digunakan untuk menyelesaian penjadwalan yaitu priority dispatch rule (Hasanudin, 2011).

Priority dispatch rule adalah suatu aturan penjadwalan yang mengatur job mana pada suatu antrian job pada suatu meisn yang harus diproses terlebih dahulu berdasarkan prioritas-prioritas tertentu. Jadi, pada suatu meisn telah selesai memroses satu job, aka berdasarkan priority dispatch rule dipilih satu job yang memiliki prioritas tertinggi untuk selanjutnya diproses pada mesin tersebut (Hasanudin, 2011).

Beberapa cara penentuan prioritas yang dapat digunakan sebagai simulasi untuk menetapkan pedoman dispatching prioritas yang terbaik. Beberapa aturan prioritas yang umum adalah FCFS, SPT dan EDD (Eddy, 2008 dikutip oleh Widodo, 2014):

1. FCFS (First Come First Serve), pekerjaan yang datang lebih awal pada suatu pusat kerja akan dikerjakan lebih dahulu. Aturan ini banyak digunakan pada bank, supermarket, kantor pos dan sebagainya.
2. SPT (Shortest Processing Time), pekerjaan yang paling cepat selesainya mendapat prioritas pertama untuk dikerjakan lebih dahulu. Cara ini seringkali diterapkan bagi perusahaan perakitan atau jasa.
3. EDD (Earliest Due Date), pekerjaan yang harus selesai paling awal dikerjakan lebih dahulu. Cara ini seringkali digunakan pada perusahaan yang bergerak di bidang konveksi dan tekstil.

Disamping ketiga aturan prioritas tersebut, dikenal juga beberapa cara, antara lain critical ratio dan least slack. Dalam critical ratio (CR), pekerjaan yang rasio antara due date terhadap lama waktu kerja palng kecil mendapat proritas lebih

dulu. Sementara dalam least slack (LS) pokerjaan yang memiki slack time terkecil mendapat prioritas untuk dikenakan lebih dulu. Slack time menunjukkan perbedaan antara waktu tersisa hingga tanggal jatuh tempo dengan waktu proses yang tersisa (Herjanto, 2007).

2.2.2 Tipe Heuristik Modern (Metaheuristik)

Metaheuristik adalah pendekatan komputasi untuk mencari solusi optimal atau mendekati optimal dari suatu problem optimasi dengan cara mencoba seraca iteratif untuk memperbaiki kandidat solusi dengan memperhatikan batasan kualitas solusi yang diinginkan. Metaheuristik tidak menjamin solusi optimal akan ditemukan . istilah lain untuk metaheuristik adalah derivative-free, direct search, black-box atau heuristik optimizer (Santosa, 2011).

Metaheuristik, mencari solusi dengan memadukan interaksi antara prosedur pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik local optima dan melakukan pencarian di ruang solusi untuk menemukan solusi global. Meteheuristik biasanya berupa prosedur umum yang bisa diterapkan untuk berbagai problem. Tentu saja diperlukan berbagai modifikasi agar metode metaheuristik sesuai dapat menyelesaikan masalah khusus yang dihadapi (Santosa, 2011).

Algoritma heuristik modern atau yang lebih dikenal dengan meta-heuristik memecahkan masalah penjadwalan produksi dengan melakukan perbaikan mulai dengan satu atau lebih solusi awal. Solusi awal ini dapat dihasilkan secara acak dapat pula dihasilkan berdasarkan heuristik tertentu. Empat algoritma meta-heuristik yang dapat digunakan dalam memecahkan masalah penjadwalan yaitu (Hasanudin, 2011):

1. Simulated Annealing

   lde dasar Simulated Annealing terbentuk dari pemrosesan logam. Annealing (memanaskan kemudian mendinginkan) dalam pemrosesan logam ini adalah suatu proses bagaimana membuat bentuk cair berangsur- angsur menjadi bentuk yang lebih padat seiring dengan penurunan temperatur. Simulated annealing

biasanya digunakan untuk penyelesaian masalah yang mana perubahan keadaan dari suatu kondisi ke kondisi yang lainnya membutuhkan ruang yang sangat luas.

2. Tabu Search

   Tabu search merupakan metode optimasi yang menggunakan short-term memory untuk menjaga agar proses pencarian tidak terjebak pada nilai optima local. Metode ini menggunakan tabu list untuk menyimpan sekumpulan solusi yang baru saja dievalusi. Selama proses optimasi, pada setiap iterasi, solusi yang akan dievalusi akan dicocokkan terlebih dahulu dengan isi tabu list untuk melihat apakah solusi tersebut sudah ada pada tabu list. Apabila sudah ada, maka solusi tersebut tidak akan dievalusi lagi. Keadaan ini terus berulang sampai tidak ditemukan lagi solusi yang tidak terdapat dalam tabu list. Pada metode tabu search, solusi baru dipilih jika solusi tersebut yang merupakan anggota bagian himpunan solusi tetangga merupakan solusi dengan fungsi tujuan paling baik jika dibandingkan dengan solusi-solusi lainnya dalam himpunan solusi tetangga tersebut. Tetangga (neighbour) dari suatu solusi adalah solusi-solusi lain yang dapat diperoleh dari solusi tersebut dengan cara memodifikasinya berdasarkan aturan-aturan tertentu yang dikenal dengan nama neighborhood functions.

3. Algoritma Genetika

   Algoritma Genetika dimodelkan berdasar proses alami, yaitu model seleksi alam oleh Darwin, sedermikia hingga kualitas individu akan sangat kompatibel dengan lingkungannya (dalam hal ini kendala masalah). Algoritma genetika memberikan suatu alternatif untuk proses penentuan nilai parameter dengan meniru cara reproduksi genetika. Teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang disebut dengan populasi. Setiap individu adalah satu buah solusi unik dan populasi adalah satu himpunan solusi pada setiap tahapan iterasi. Algoritma genetika bekerja untuk mencari struktur individu berkualitas tinggi yang terdapat dalam populasi.

4. Algoritma Differential Evolition

   Algoritma Differential Evolition merupakan metode metaheuristik akhir. Metode ini terbilang cukup baru, merupakan versi pengembangan dari algoritma Genetika. Prinsipnya adalah berdasarkan analogi evolusi biologi, yang terdiri

dari proses penginilisasian populasi, proses mutasi, proses penyilangan, dan proses penyeleksian. Keunggulan algoritma ini adalah berstruktur sederhana, mudah dalam pengimplementasian, cepat dalam menca ai solusi, dan bersifat tangguh (memiliki standar deviasi yang kecil).

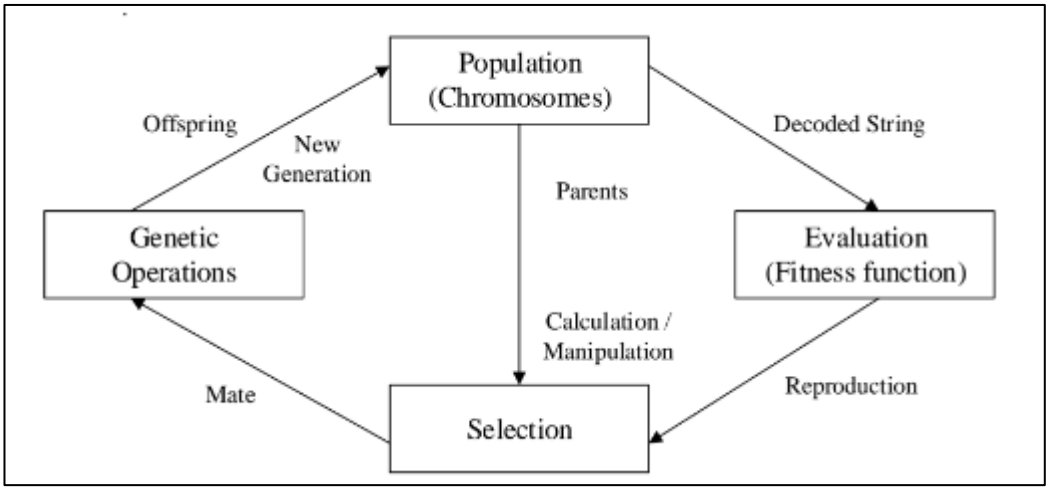
2.3 Algoritma Genetika

Algoritma genetika merupakan teknik optimisasi dan pencarian yang berdasarkan pada prinsip ilmu genetika dan scleksi alam dimana algoritma ini terinspirasi dari proscs biologi evoulsi Darwin. Dalam algoritma genetika, seleksi, mutasi, dan crossover mempakan hal yang utama. Algoritma genetika berbeda dengan teknik pencarian konvensional karena algoritma ini dimulai dengan sekumpulan inisialisasi dari solusi acak yang disebut populasi. Populasi terdiri dari kumpulan kromosom yang disebut sebagai gen. Nilai gen dapat berupa bilangan numerik, biner, simbol atau karakter, tergantung pada masalah yang akan diselesaikan (Silitonga, 2021).

Kromosom-kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan generasi. Pada tiap generasi, tingkat keberhasilan nilai solusi dari kromosom-kromosom tersebut dievaluasi terhadap masalah yang ingin diselesaikan menggunakan ukuran yang disebut dengan fitness (Silitonga, 2021).

Secara umum siklus proses dari algoritma genetika diperlihatkan pada Gambar 2.1. Dimulai dari sejumlah 'n' kromosom pada populasi acak genetika, dilakukan evaluasi fitness dari masing-masing kromosom pada populasi. Kemudian dilakukan proses seleksi dengan memilih dua kromosom parent dari populasi berdasarkan fimessnya masing-masing, semakin haik nilaifimess, semakin besar kesempatan untuk dipilih. Dengan probabilitas crossover, kedua parent disilangkan untuk menghasilkan keturunan. Apabila tidak terjadi crossover, keturunan yang dihasilkan merupakan hasil penggandaan parent. Setelah crossover, maka proses selanjutnya adalah mutasi dengan probabilitas mutasi yang telah ditentukan. Selanjutnya, offspring haru ditempatkan dalam populasi yang baru. Offspring yang baru, digunakan untuk pengulangan proses berikutnya. Apabila kondisi akhir sesuai

dengan yang diharapkan, maka proses akan dihentikan dan akan dikembalikan kepada solusi terbaik dalam populasi tersebut (Silitonga, 2021).



Gambar 2.1. Siklus Proses Algoritma Genetika
(Sumber: Silitonga, 2021).

2.3.1 Teknik Pengkodean (Encoding Techniques)

Agar gen dapat diproses melalui algoritma genetika, maka gen tersebut harus dikodekan terlebih dahulu kedalam bentuk kromosom. Masing - masing kromosom berisi sejumlah gen yang mengkodekan informasi dan kemudian disimpan didalam individu atau kromosom. Gen dapat direpresentasikan ke dalam bentuk bit, bilangan real, string, daftar aturan, gabungan dari beberapa kode, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. Beberapa jenis teknik pengkodean menurut diantaranya (Silitonga, 2021):

a. Pengkodean biner (Binary Encoding)

Setiap kromosom mengkodekan string biner (bit). Setiap bit dalam string dapat mewakili beberapa karakteristik dari solusi. Setiap string bit merupakan solusi namun belum tentu menjadi solusi terbaik. Keseluruhan string dapat mewakili sebuah angka. Berikut merupakan Gambar 2.2 Pengkodean Biner

Kromosom 1	1	1	0	1	0	0	0	1	1	0	1	0
Kromosom 2	0	1	1	1	1	1	1	1	1	1	0	0

Gambar 2.2 Pengkodean Biner
(Sumber: Silitonga, 2021).

b. Pengkodean Oktal (Octal Encoding)

Pengkodean ini menggunakan string yang terdiri dari bilangan octal (0-7). Berikut merupakan Gambar 2.3 Pengkodean Oktal.

Kromosom 1	0	3	4	6	7	2	1	6
Kromosom 2	1	5	7	2	3	3	1	4

Gambar 2.3 Pengkodean Oktal
(Sumber: Silitonga, 2021).

c. Pengkodean Heksadesimal (Hexadesimal Encoding)

Pengkodean ini menggunakan string yang terdiri dari bilangan hekasadesimal (0-9, A-F). Berikut merupakan Gambar 2.4 Pengkodean Heksadesimal.

Kromosom 1	9	C	E	7
Kromosom 2	1	5	7	2

Gambar 2.4 Pengkodean Heksadesimal
(Sumber: Silitonga, 2021).

d. Pengkodean Permutasi (Permutation Encoding)

Setiap kromosom merupakan serangkaian angka yang mewakili bilangan secara berurutan. Terkadang koreksi harus dilakukan setelah operasi genetika selesai. Dalam pengkodean permutai, setiap kromosom adalah string integer atau nilai sebenarnya yang mewakili bilangan secara berurutan. Berikut merupakan Gambar 2.5 Pengkodean Permutasi.

Kromosom A	1	5	3	2	6	4	7	9	8
Kromosom B	8	5	6	7	2	3	1	4	9

Gambar 2.5 Pengkodean Permutasi
(Sumber: Silitonga, 2021).

e. Pengkodean nilai (Value Encoding)

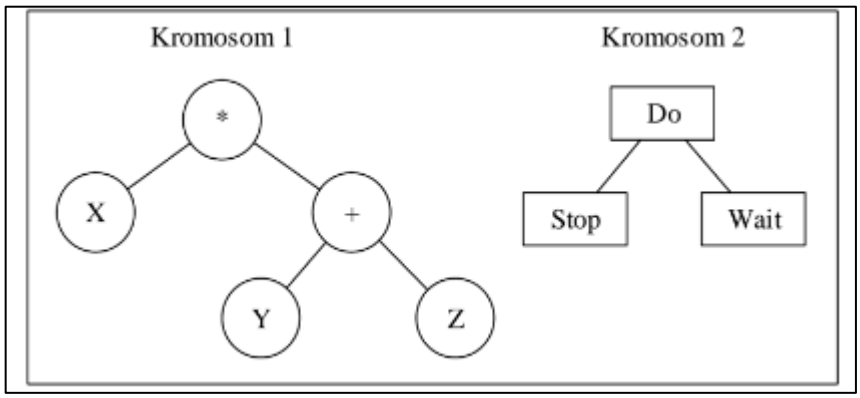
Dalam pengkodean nilai, setiap kromosom adalah serangkaian beberapa nilai. Nilai dapat berupa sesuatu yang berhubungan dengan masalah, bentuk angka, bilangan real atau karakter ke beberapa objek rumit. Pengkodean ini memberikan hasil yang terbaik untuk beberapa masalah khusus. Berikut merupakan Gambar 2.6 Pengkodean Nilai.

Kromosom A	1.2324	5.3243	0.4556	2.3293	2.4545
Kromosom B	ABDJEIFJDHDIERJFDLDFLFEGT				
Kromosom C	(back)	(back)	(right)	(forward)	(left)

Gambar 2.6 Pengkodean Nilai
(Sumber: Silitonga, 2021)**.**

f. Pengkodean Tree (Tree Encoding)

Pengkodean ini sebagian besar digunakan untuk mengembangkan ekspresi program pada pemrograman genetika. Setiap kromosom merupakan tree dari beberapa objek seperti fungsi dan perintah bahasa pemrograman. Berikut merupakan Gambar 2.7 Pengkodean Tree.



Gambar 2.7 Pengkodean Tree
(Sumber: Silitonga, 2021)**.**

2.3.2 Pembangkitan Populasi Awal dan Kromosom

Sebuah populasi merupakan kumpulan solusi dari masalah sehah setiap kromosom yang terdapat pada populasi merepresentasikan sebuah solusi. Pembangkitkan populasi awal merupakan proses membangkitkan sejumlah

individu atau kromosom sccara acak atau melalui prosedur tertentu. Ukuran populasi awal memberikan dampak yang besar terhadap kinerja algoritma. Ukuran populasi begantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Ukuran populasi memiliki efek langsung terhadap ukuran memori, kecepatan konvergensi dan kecepatan pencarian algoritma. Apabila ukuran populasi teralu besar, algoritma genetika cenderung membutuhkan waklu lebih lama untuk menemukan sebuah solusi, sebaliknya, apabila ukuran populasi terlalu kecil, algoritma genetika akan sulit menemukan solusi yang suboptimal (Silitonga, 2021).

2.3.3 Evaluasi Fitness

Fungsi fitness didesain untuk mempresentasikan kekuatan adaptasi individu terhadap lingkungan dan pada umumnya algoritma genetika mencari solusi optimal dengan memaksimalkan atau meminimalisasikan fungsi fitness. Jadi, suatu individu dievaluasi berdasarkan suatu fungsi fitness sebagai ukuran performansinya. Dalam evolusi alam, individu yang bernilai fitness tinggi akan bertahan hidup sedangkan individu yang bernilai fitness rendah akan mati (Silitonga, 2021).

2.3.4 Seleksi

Seleksi merupakan proses individu yang akan dipilih untuk dilakukan penyilangan dengan tujuan agar mendapatkan individu berkualitas pada generasi berikutnya. Schelum dilakukan proses sclcksi, terlcbih dahulu dilakukan pencarian nilai fitness dari masing - masing individu. Nilai fitness berfungsi untuk mendefinisikan kemampuan atau kualitas individu tersebut, yang nantinya akan digunakan pada tahap - tahap seleksi berikutnya. Beberapa jenis metode seleksi yaitu (Silitonga, 2021):

a. Seleksi roda roulette (Roulere wheel selection)

   Prinsip seleksi roulette adalah pencarian linier melalui roda roulette dengan beberapa slot di dalam roda yang perbandingan ukurannya sesuai dengan nilai fitness masing - masing individu. Berikut adalah langkah - langkah seleksi roulette wheel:

1. Hitung jumlah nilai fitness dari setiap individu dalam populasi.
2. Hitung nilai fitness dari masing - masing individu dan probabilitas seleksi mereka dengan membagi fitness kromosom individu melalui penjumlahan nilai fitness dari keseluruhan populasi.
3. Bagi roulette wheel menjadi sektor - sektor menurut probabilitas yang telah dihitung dalam tahap kedua.
4. Putar roda sebanyak 'N' kali. Ketika roulette berhenti, sektor yang terpilih merupakan individu yang terpilih.

Berikut probabilitas seleksi roulette wheel pada Persamaan 2.1

$$ps(a_i) = \frac{f(a_i)}{\sum_{i=1}^{n} f(a_i)} \quad (2.1)$$

Keterangan:

J = 1,2, ….., n

n = ukuran populasi

F(ai) = nilai fitness dari individu ai

b. Seleksi Peringkat Linier (Linear Ranking Selection)

Seleksi peringkat dalam algoritma genetika diperkenalkan Oleh Baker untuk mengeliminasi kekurangan dari pemilihan proporsinal. Pada metode seleksi peringkat linier, individu dipilih berdasarkan nilai fitness dan kemudian peringkat tersehut ditentukan terhadap individu. Individu terbaik mendapat peringkat ‘N’ dan yang terburuk mcndapat pcringkat ‘1’ probabilitas seleksi kcmudian diciptakan secara linier terhadap individu berdasarkan peringkat mereka.

$$P_i = \frac{1}{N}(n^- + (n^+ - n^-)\frac{i-1}{N-1}) \quad (2.2)$$

Keterangan:

$i \in \{1, \ldots, N\}$

P_i = probabilitas seleksi individu

$\frac{n^-}{N}$ = probabilitas seleksi individu terburuk

$\frac{n^+}{N}$ = probabilitas seleksi individu terbaik

c. Seleksi Peringkat Eksponensial (Exponential Ranking Selection)

Baik seleksi exponential ranking maupun sceeksi linear ranking adalah sama. Perbedaan keduanya terletak pada perhitungan probabilitas seleksi. Pada algoritma exponential ranking individu yang terbaik diberikan peringkat 'N' dan yang terburuk diberikan ' 1’ Teknik ini berbeda dari linear ranking selection dimana probabilitas seleksi exponential ranking dipengaruh secara eksponensial.

$$P_i = \frac{c^{N-i}}{\sum_{j=1}^{N} c^{N-j}} \tag{2.3}$$

Keterangan:

$0 < c < 1$

$\sum_{j=1}^{N} c^{N-j}$ menormalisasikan probabilitas untuk memastikan $\sum_{i=1}^{N} P_i = 1$

d. Seleksi Turnamen (Tournament Selection)

Seleksi turnamen merupakan teknik seleksi algoritma genetika yang paling populer karena efisiensi dan kemudahan implementasinya. Pada seleksi turnamen, sejumlah 'N' individu dipilih secara acak dari keseluruhan populasi. Individu — individu ini Saling berkompetisi antara satu dengan yang lainnya. Individu dengan nilai fitness teninggi akan menang dan terpilih untuk proses algoritma genetika berikutnya. Adapun kelebihan dari seleksi turnamen yaitu waktu kompleksitas yang lehih rendah O(n), mudah diimplementasikan, rendah kemungkinan terjadinya dominasi individu tertentu. Kelebihan - kelebihan tersebut menjadikan seleksi tumamen lebih etisien daripada teknik lainnya.

2.3.5 Crossover (Penyilangan)

Crossover merupakan process pengambilan solusi kedua parent dan menghasilkan keturunan. Fungsi crossover adalah menghasilkan kromosom offspring dari kombinasi materi gen dua kromosom parent. Beberapa jenis crossover yaitu (Silitonga, 2021):

a. One point crossover

Crossover ini menggunakan fragmentasi single point dari parent dan kemudian menggabungkan parem tersebut pada titik yang telah ditentukan untuk menghasilkan offspring. pertama, parent yang akan disilangkan dipilih terlebih

dahulu, kemudian titik crossover Pi (I = O sampai n - 1) ditentukan secara acak. Offspring dihasilkan dengan menggabungkan kedua parent pada titik crossover yang telah ditentukan sebelumnya. Berikut merupakan Gambar 2.8 One Point Crossover.

Parent 1	1	0	1	0	|	1	0	0	1	0
Parent 2	1	0	1	1	|	1	0	1	1	0
Offspring 1	1	0	1	0	|	1	0	1	1	0
Offspring 2	1	0	1	1	|	1	0	0	1	0

Gambar 2.8 One Point Crossover
(Sumber: Silitonga, 2021)**.**

b. K-Point crossover

Pada prinsipnya crossover ini sama seperti one point crossover yaitu menggunakan titik crossover acak untuk menggabungkan kedua parent. Untuk memberikan kombinasi yang baik dari parents, maka dipilih lebih dari satu titik crossover untuk menghasilkan offspring. Parent yang akan disilangkan dipilih terlenih dahulu, kemudian titik crossover P1i to Pk-1i (I = 0 sampai n – 1) dipilih secara acak, selanjutnya kedua parent digabungkan pada titik – titik crossover untuk menghasilkan offspring. Berikut merupakan Gambar 2.9 K-Point Crossover.

Parent 1	1	0	|	1	0	|	1	0	0	|	1	0
Parent 2	1	1	|	0	0	|	1	0	1	|	1	0
Offspring 1	1	0	|	0	0	|	1	0	0	|	1	0
Offspring 2	1	1	|	1	0	|	1	0	1	|	1	0

Gambar 2.9 K-Point Crossover
(Sumber: Silitonga, 2021)**.**

c. Shuffle Crossover

Shuffle crossover menggunakan teknik yang sama dengan one point crossover. Crossover ini mengacak kedua parents dengan cara yang sama, kemudian menerapkan teknik one point crossover dengan secara acak memilih titik crossover dan menggabungkan kedua parenls untuk menghasilkan dua

offspring. Setelah melakukan one point crossover, gen pada offspring tidak diacak dengan cara yang sama seperti yang telah diacak.

Tentukan titik acak Berikut merupakan Gambar 2.10 Tentukan Titik Acak.

Parent 1	1	1	1	0	1	0	0	1	0
Parent 2	1	0	0	0	1	0	1	1	0

Gambar 2.10 Tentukan Titik Acak
(Sumber: Silitonga, 2021).

Acak gen seperti mengacak titik. Berikut merupakan Gambar 2.11 Acak Gen.

Offspring 1	0	1	0	1	1	0	1	1	0
Offspring 2	0	0	1	1	1	0	0	1	0

Gambar 2.11 Acak Gen
(Sumber: Silitonga, 2021).

Tentukan titik one point crossover. Berikut merupakan Gambar 2.12 Penentuan titik one point crossover.

Parent 1	0	1	0	1	|	1	0	1	1	0
Parent 2	0	0	1	1	|	1	0	0	1	0

Gambar 2.12 Penentuan Titik One Point Crossover
(Sumber: Silitonga, 2021).

Terapkan one point crossover pada gen parents. Berikut merupakan Gambar 2.13 Penerapan One Point Crossover pada Gen Parents.

Parent 1	0	1	0	1	|	1	0	0	1	0
Parent 2	0	0	1	1	|	1	0	1	1	0

Gambar 2.13 Penerapan One Point Crossover pada Gen Parents
(Sumber: Silitonga, 2021).

Tentukan titik non-acak sama seperti titik acak. Berikut merupakan Gambar 2.14 Penentuan Titik Non-Acak.

Offspring 1	0	1	0	1	1	0	0	1	0
Offspring 2	0	0	1	1	1	0	1	1	0

Gambar 2.14 Penentuan Titik Non-Acak
(Sumber: Silitonga, 2021).

Kembalikan gen pada offspring. Berikut merupakan Gambar 2.15 Pengembalian Gen pada Offspring.

Offspring 1	1	1	0	0	1	0	0	1	0
Offspring 2	1	0	1	0	1	0	1	1	0

Gambar 2.15 Pengembalian Gen pada Offspring
(Sumber: Silitonga, 2021).

d. Uniform Crossover

Uniform crossover tidak membagi kromosom untuk rekombinasi. Tiap gen yang terdapat pada offspring dihasilkan dengan menyalin gen tersebut dari parent yang terpilih berdasarkan bit yang sesuai ke dalam mask persilangan binary yang panjangnya sama dengan panjang parent kromosom. Mask persilangan dibangkitkan secara bervariusi untuk setiap pasang parent kromosom. Jumlah titik crossover pada awalnya tidak ditentukan. Jadi, offspring merupakan gabungan gen dari kedua parent. Berikut merupakan Gambar 2.16 Uniform Crossover.

Parent 1	1	2	3	4	5	6	7	8	9
Parent 2	3	4	5	6	7	8	9	10	11
Mask	0	1	0	1	0	0	1	1	0
Inverse Mask	1	0	1	0	1	1	0	0	1
Offspring 1	1	4	3	6	5	6	9	10	9
Offspring 2	3	2	5	4	7	8	7	8	11

Gambar 2.16 Uniform Crossover
(Sumber: Silitonga, 2021).

e. Average Crossover

Average crossover merupakan teknik crossover berbasis nilai. Cmssover ini hanya menghusilkan sebuah offspring dari dua parents yang disilangkan. Offspring yang dihasilkan merupakan nilai rata - rata dari kedua parents. Berikut merupakan Gambar 2.17 Average Crossover.

Parent 1	5	3	3	2	3	9	7	6	5
Parent 2	5	4	7	6	5	2	6	1	3
Offspring 1	5	3	5	4	4	5	6	3	4

Gambar 2.17 Average Crossover
(Sumber: Silitonga, 2021)**.**

f. Discrete Crossover

Crossover ini menggunakan bilangan ril acak untuk menghasilkan satu offspring dari kedua parents. Bilangan ril acak tersebut akan menentukan dari gen parents mana yang akan diturunkan ke offspring. Cara kerja crossover ini sama dengan uniform crossover yaitu memilih gen dari kedua parents secara seragam, tetapi hanya menghasilkan satu offspring. Berikut merupakan Gambar 2.18 Discrete Crossover.

Parent 1	1	1	1	0	1	0	0	1	0
Parent 2	1	0	0	0	1	0	1	1	0
Offspring 1	1	1	0	0	1	0	1	1	0

Gambar 2.18 Discrete Crossover
(Sumber: Silitonga, 2021)**.**

2.3.6 Mutasi

Mutasi bertujuan untuk menambahkan keberagaman ciri genetika populasi yaitu dengan mengubah susunan gen dalam suatu kromosom. Proses mutasi juga berperan untuk menggantikan gen yang hilang dari populiLsi sehingga memungkinkan munculnya kembali gen yang tidak muncul pada inisialis&si popul&si. Dengan adanya mutasi, premature convergence dapat dihindari, yaitu kondisi dimana kromosom mengalami convergence disaat nilai fimess belum mencapai optimal. Terdapat beberapa enis mutasi, diantaranya (Silitonga, 2021):

a. Metode Pembalikan (Inversion Mutation)

Dilakukan dengan mengambil suatu substring yang terletak diantara dua titik yang dipilih secara acak pada kromosom. Kemudian dilakukan proses pembalikan gen pada substring tersebut. Berikut merupakan Gambar 2.19 Inversion Mutation.

Parent	2	9	1	0	7	4	5	8	3
Offspring	2	9	4	7	0	1	5	8	3

Gambar 2.19 Inversion Mutation
(Sumber: Silitonga, 2021).

b. Metode Penyisipan (Insertion Mutation)

Pada metode ini, salah satu gen yang ada pada kromosom dipilih, selanjutnya gen tersebut disisipkan pada posisi yang dipilih secara acak. Berikut merupakan Gambar 2.20 Insertion Mutation.

Parent	2	9	1	0	7	4	5	8	3
Offspring	2	9	4	1	0	7	5	8	3

Gambar 2.20 Insertion Mutation
(Sumber: Silitonga, 2021).

c. Metode Pemindahan (Displacement Mutation)

Metode ini dilakukan dengan memilih dua titik pada kromosom. Gen yang terdapat diantara kedua titik tersebut disisipkan pada suatu posisi yang dipilih secara acak. Berikut merupakan Gambar 2.21 Displacement Mutation.

Parent	2	9	1	0	7	4	5	8	3
Offspring	2	7	4	9	1	0	5	8	3

Gambar 2.21 Displacement Mutation
(Sumber: Silitonga, 2021).

d. Metode Penukaran (Swap Mutation)

Dilakukan dengan memilih dua buah gen yang ada pada kromosom dan kemudian menukarkan posisi kedua gen tersebut. Berikut merupakan Gambar 2.22 Swap Mutation.

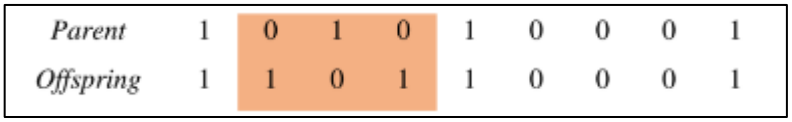
Parent	2	9	1	0	7	4	5	8	3
Offspring	2	9	1	8	1	0	5	0	3

Gambar 2.22 Swap Mutation
(Sumber: Silitonga, 2021).

e. Metode Penggantian (Flip Mutation)

Metode ini digunakan pada persoalan dengan representasi biner, yaitu mengganti nilai gen yang terpilih. Apabila gen terpilih tersebut bernilai 1,

maka nilai gen diganti dengan 0, demikian sebaliknya. Berikut merupakan Gambar 2.23 Flip Mutation.

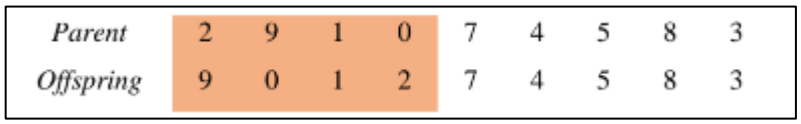


Parent	1	0	1	0	1	0	0	0	1
Offspring	1	1	0	1	1	0	0	0	1

Gambar 2.23 Flip Mutation
(Sumber: Silitonga, 2021).

f. Metode Camput Aduk (Scramble Mutation)

Proses mutasi dilakukan dengan memilih posisi beberapa gen secara acak, selanutnya urutan gen – gen tersebut ditukar secara acak juga. Berikut merupakan Gambar 2.24 Scramble Mutation.



Parent	2	9	1	0	7	4	5	8	3
Offspring	9	0	1	2	7	4	5	8	3

Gambar 2.24 Scramble Mutation
(Sumber: Silitonga, 2021).

2.3.7 Parameter Algoritma Genetika

Ukuran populasi, probabilitas crossover (Pc), probabilitas mutasi (Pm) merupakan parameter dalam algoritma genetika yang nilainya harus diatur. Pemilihan ukuran populasi yang digunakan bergantung pada masalah yang akan diselesaikan. Ukuran populasi yang lebih besar diperlukan untuk masalah yang lebih kompleks guna mencegah terjadinya konvergensi prematur, dalam hal ini dihasilkannya optimum local (Silitonga, 2021).

Probabilitas crossover (Pc) adalah parameter dasar yang ditentukan untuk mengendalikan frekuensi crossover dan untuk menggambarkan seberapa sering Crossover akan dilakukan. Di dalam populasi, sebanyak Pc*N individu yang mengalami pindah silang pada tiap generasi. Nilai probabilitas crossover berkisar antara O sampai dengan 1. Nilai probabilitas crossovew mempengaruhi struklur individu baru yang akan terbentuk dimana struktur individu akan semakin cepat dihasilkan, apabila nilai probabilitas yang diberikan semakin besar. Nilai prohablitas yang terlalu besar akan semakin cepat menghilangkan individu yang merupakan kandidat solusi terhaik. Sebaliknya angka eksplorasi akan menurun dan

stagnasi akan terjadi apabila nilai probabilitas yang diberikan terlalu rendah (Silitonga, 2021).

Probabilitas mutasi (Pm) merupakan parameter mutasi yang didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang akan mengalami perubahan secara acak setelah proses crossover. Adapun banyaknya gen yang akan mengalami mutasi yaitu Pm dimana N adalah jumlah individu dalam populasi dan L adalah panjang bit dalam kromosom. Offspring segera dihasilkan setelah crossover tanpa ada perubahan apapun apabila tidak terjadi mutasi, sebaliknya satu atau lebih bagian dari kromosom akan berubah apabila mutasi dilakukan. Jika probabilitas mutasi sebesar 100%. seluruh kromosom akan berubah, jika probabilitas mutasi sebesar, tidak akan ada kromosom yang berubah (Silitonga, 2021).

2.4 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu acuan yang digunakan dalam melakukan penelitian pengembangan algoritma genetika untuk optimasi penjadwalan produksi job shop fleksibel (FJSP) sehingga acuan yang digunakan mengenai penelitian yang berkaitan dengan topik penelitian tersebut. Berikut merupakan Tabel 2.1 menyajikan penelitian terdahulu

Tabel 2.1 Penelitian Terdahulu

No	Penulis	Single Objektif	Multi Objektif	Fungsi Objektif	Waktu Transportasi	Waktu Setup	Algoritma	Hasil
1	(Wang & Zhu, 2021)	✓		Meminimalkan Makespan	-	✓	Hybrid Algoritma Genetika dan Tabu Search	81
2	(Huang & Yang, 2019)		✓	Meminimalkan Makespan, Beban kerja mesin, dan Total beban kerja mesin	✓	-	Hybrid Algoritma Genetika dan Simulated Annealing	15.45,14 ,78
3	(Zhang, Zhang, et al., 2019)	✓		Meminimalkan Makespan	-	-	Hybrid Algoritma Genetika dan Variable Neigborhood Search	40

4	(Defersha & Rooyani, 2020)	✓		Meminimalkan Makespan	-	✓	Algoritma Genetika dua tahap	4609
5	(Zhang et al., 2020)		✓	Meminimalkan Makespan, Total waktu setup dan Total waktu transportasi	✓	✓	Algoritma Genetika	65,75,63
6	(Zhang et al., 2019)	✓		Meminimalkan Makespan	✓	-	Algoritma Genetika	-
7	(Liang et al., 2022)		✓	Meminimalkan Makespan dan Energy	-	-	Algoritma Genetika	51
8	(Liu et al., 2021)	✓		Meminimalkan Makespan	-	-	Hybrid Algoritma Genetika dan Variable Neigborhood Search	40
9	(Li & Xia, 2022)	✓		Meminimalkan Makespan	-	-	Hybrid Algoritma Genetika	40
10	(Jamrus et al., 2018)	✓		Meminimalkan Makespan	-	-	Hybrid Algoritma Genetika dan PSO	-
11	(Wu et al., 2018)	✓		Meminimalkan Makespan	-	-	Hybrid Algoritma Genetika dan VNS	-
12	(Meng et al., 2023)	✓		Meminimalkan Makespan	-	✓	Algoritma Genetika	70
13	(Park et al., 2021)	✓		Meminimalkan Makespan	-	-	Algoritma Genetika	47
14	(Huang et al., 2018)		✓	Meminimalkan Makespan, Beban kerja mesin dan Total beban kerja mesin.	-	-	Hybrid Algoritma Genetika dan PSO	42,42,162
15	(Jiang et al., 2023)	✓		Meminimalkan Makespan	-	-	Algoritma Genetika	40
16	(Luo et al., 2018)	✓		Meminimalkan Makespan	-	-	Algoritma Genetika	20

Penjadwalan produksi adalah masalah optimasi terpenting dalam perencanaan dan manajemen sistem manufaktur modern (Wang & Zhu, 2021). Menurut Park dkk. (2021), Dalam sistem industri, penjadwalan produksi merupakan hal yang krusial, dan sebagian besar merupakan masalah optimasi kombinatorial NP-hard. Salah satu masalah penjadwalan yang paling umum digunakan saat ini adalah masalah penjadwalan *job shop* (JSP). Menugaskan serangkaian operasi ke sekelompok mesin sehingga setiap tindakan hanya dapat diselesaikan pada satu mesin adalah tugas yang ada (Zhang et al., 2019). Secara umum, model *job shop* mengharuskan setiap tugas memiliki jalur yang ditetapkan dan diselesaikan pada satu mesin yang telah dikonfigurasi sebelumnya (Huang et al., 2018). Namun, pemilihan mesin dapat disesuaikan dalam produksi sebenarnya. Selain itu, banyak mesin dapat memproses satu atau lebih operasi (Liu et al., 2021). Ini menyiratkan bahwa banyak mesin mampu memproses setiap operasi. Akibatnya, masalah ini dikenal sebagai masalah penjadwalan *job shop* fleksibel (FJSP) (Zhang et al., 2019).

Ketika menangani masalah penjadwalan *job shop* yang fleksibel, ada beberapa fungsi objektif yang perlu dipertimbangkan. Makespan, waktu penyelesaian rata-rata, waktu aliran maksimum, total keterlambatan tertimbang, keterlambatan rata-rata, keterlambatan maksimum, jumlah pekerjaan yang terlambat, dan total beban kerja mesin adalah beberapa fungsi objektif tersebut (Amjad et al., 2018). Tujuan paling umum dari masalah penjadwalan *job shop* fleksibel adalah untuk mengurangi makespan, atau waktu yang dibutuhkan untuk menyelesaikan pekerjaan secara keseluruhan. Berdasarkan tabel 2.2 penelitian terdahulu tujuan dari sebagian besar fungsi objektif yang digunakan untuk menyelesaikan tantangan penjadwalan *job shop* fleksibel yaitu single objektif dimana untuk meminimalkan makespan, atau waktu operasi secara keseluruhan. Namun demikian, implementasi sistem produksi mengharuskan penentuan fungsi objektif yang banyak atau multi-objektif. Akibatnya, penelitian tentang masalah penjadwalan *job shop* fleksibel multi-tujuan (MOFJSP) menjadi semakin populer dan penting (Huang et al., 2018).

Berdasarkan tabel 2.2 penelitian sebelumnya sebagian besar penelitian yang ada tidak mempertimbangkan waktu transportasi dan waktu setup dalam menyelesaikan masalah FJSP. Kedua hal tersebut sangat penting karena setiap operasi melakukan perpindahan ke mesin lain untuk operasi berikutnya, sementara mesin memerlukan persiapan untuk memproses operasi berikutnya. Oleh karena itu, penelitian ini akan mempertimbangkan waktu transportasi dan waktu setup dalam mengatasi masalah penjadwalan FJSP tersbut.

Berdasarkan Tabel 2.2 penelitian terdahulu dimana algoritma genetika telah mendapatkan popularitas sebagai pendekatan metaheuristik untuk menyelesaikan FJSP. Karena kemampuan pencarian globalnya yang kuat, algoritma genetika adalah algoritma meta-heuristik yang secara efisien menunjukkan kinerja yang luar biasa untuk masalah penjadwalan (Wang & Zhu, 2021). Seleksi alam menjadi inspirasi bagi algoritma genetika (Meng et al., 2023) Algoritma genetika adalah salah satu strategi evolusioner yang paling sukses untuk menyelesaikan masalah penjadwalan *job shop* fleksibel (FJSP) karena meniru evolusi biologis dan memiliki ketahanan yang kuat, pemrosesan yang sederhana, ekstensibilitas yang kuat, dan kemampuan pencarian acak yang cepat (Wu et al., 2018).

Meskipun algoritma genetika memiliki kekurangan, algoritma ini merupakan teknik optimasi yang populer untuk menyelesaikan masalah penjadwalan *job shop* yang fleksibel (FJSP), menurut penelitian sebelumnya. Terkait algoritma genetika konvensional, variasi populasi menurun seiring iterasi algoritma, dan individu-individu tertentu dapat menjadi hampir sama, yang mengakibatkan stasis evolusioner populasi (Meng et al., 2023). Menurut Huang dan Yang (2019), algoritma genetika konvensional menunjukkan kecepatan konvergensi yang lambat dan potensi konvergensi dini.

Kekurangan dari algoritma genetika adalah kapasitasnya yang terbatas untuk pencarian lokal, dalam proses mengatasi masalah, mereka rentan terhadap konvergensi prematur dan pengoptimalan lokal (Liang et al., 2022). Kemudian, kemampuan pencarian lokal yang tidak memadai yang disebabkan oleh tidak adanya teknik pencarian lingkungan memperlambat pengoptimalan (Wang & Zhu,

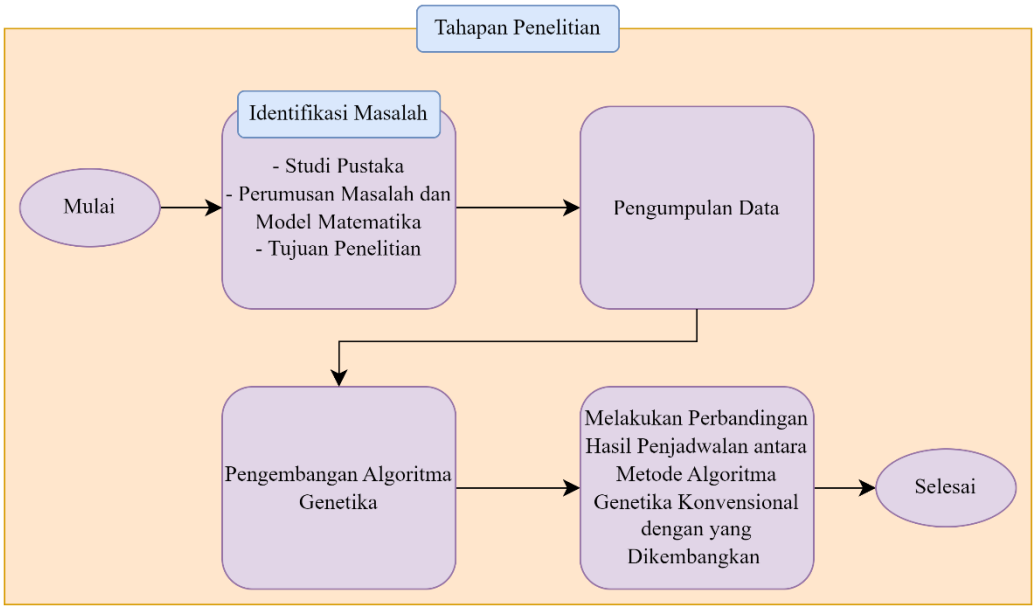
2021). Dengan menggabungkan atau melakukan hibridisasi dengan algoritma yang berbeda, dapat mengurangi kekurangan tersebut.

Penelitian berjudul "Pengembangan Algoritma Genetika untuk Mengoptimalkan Penjadwalan Produksi *Job Shop* Fleksibel (FJSP)" akan mengatasi tantangan yang disebutkan di atas dengan membuat algoritma genetika yang dikembangan dan dimodifikasi untuk mengurangi kekurangan dari metode algoritma genetika konvensional. Kemudian, meminimalkan makespan (Cm), meminimalkan waktu kerja maksimum mesin (Wm), dan meminimalkan total waktu kerja semua mesin (Wt) adalah fungsi tujuan yang akan digunakan kemudian mempertimbangkan waktu transportasi dan waktu setup dalam mengoptimasi masalah FJSP dalam penelitian ini. Tujuan dari penelitian ini adalah untuk membantu menyelesaikan masalah optimasi penjadwalan *job shop* yang fleksibel, khususnya yang berkaitan dengan penjadwalan produksi di industri manufaktur.

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini digambarkan menggunakan diagram alir atau flow chart, ini akan menggambarkan alur proses atau tahapan-tahapan penelitian mulai dari awal sampai selesai. Secara lengkap tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.2 Identifikasi Masalah

Tahap identifikasi masalah adalah cara dari peneliti untuk dapat menduga, memperkirakan dan menguraikan apa yang sedang menjadi masalah pada perusahaan. Identifikasi masalah dalam penelitian ini ada beberapa tahapan, antara lain :

3.2.1 Studi pustaka

Tahap ini bertujuan untuk menunjang pencapaian tujuan pemecahan masalah dengan menggunakan pendekatan teori yang sesuai. Studi pustaka berisi teori-teori yang dijadikan sebagai acuan untuk menjawab masalah yang ada. Tinjauan pustaka didapat dari buku, jurnal, ebook, penelitian terdahulu, dan jenis

sumber lainnya yang dapat dipertanggungjawabkan agar dapat menjawab setiap masalah

3.2.2 Perumusan Masalah dan Model Matematika

Masalah penjadwalan *job shop* fleksibel yaitu terdapat sekumpulan pekerjaan $\{J_1, J_2, \ldots, J_n\}$ independent yang harus dilakukan pada sekumpulan mesin $\{M_1, M_2, \ldots, M_m\}$. Setiap pekerjaan terdiri dari satu atau lebih jumlah operasi dan semua pekerjaan mungkin memiliki rute pemrosesan yang berbeda di lantai produksi. Diasumsikan bahwa suatu operasi dapat dilakukan pada lebih dari satu mesin dan waktu operasi bergantung pada mesin. Batasan dari asumsi tersebut menguraikan struktur utama masalah dalam penelitian ini:

1. Pekerjaan tidak bergantung satu sama lain, dan semua pekerjaan tersedia di awal.
2. Pekerjaan mempunyai prioritas yang sama untuk mulai diproses.
3. Operasi yang sedang berjalan pada suatu mesin tidak dapat dihentikan sebelum operasi tersebut selesai.
4. Sebuah operasi hanya dapat dilakukan pada satu mesin di antara subset mesin alternatif yang telah ditentukan sebelumnya, namun operasi tersebut harus dilakukan hanya oleh satu mesin dalam satu waktu.
5. Sebuah operasi tidak dapat dilakukan sebelum operasi yang mendahuluinya selesai.
6. Sebuah mesin dapat melakukan paling banyak satu operasi dalam satu waktu.
7. Waktu operasi bervariasi, tergantung pada mesin yang digunakan dan nilainya diketahui.
8. Untuk dua operasi berurutan pada mesin yang berbeda dalam pekerjaan yang sama, terdapat waktu transportasi antara kedua mesin. Jika kedua operasi tersebut dilakukan pada mesin yang sama, maka waktu transportasi adalah 0.
9. Setiap operasi memiliki waktu setup yang berbeda-beda tergantung pada mesin yang digunakan. Jika dua operasi berurutan pada mesin yang sama dari satu pekerjaan, maka waktu setup adalah 0.

Variabel Keputusan yaitu variabel yang menguraikan secara lengkap keputusan-keputusan yang akan dibuat dan akan mempengaruhi nilai tujuan yang hendak dicapai. Variabel keputusan yang digunakan ditunjukkan pada Rumus (3.1) dan Rumus (3.2).

$$X_{ijh} = \begin{cases} =1, & \text{jika operasi } O_{jh} \text{ ditugaskan ke mesin } M_i \\ =0, & \text{Jika tidak} \end{cases} \quad (3.1)$$

$$Y_{ijhkl} = \begin{cases} =1, & \text{jika operasi } O_{jh} \text{ mendahului } O_{kl} \text{ pada mesin } M_i \\ =0, & \text{Jika tidak} \end{cases} \quad (3.2)$$

Keterangan:

X_{ijh} = Varibel keputusan yang menunjukkan apakah operasi O_{jh} ditugaskan kedalam mesin M_i atau tidak

Y_{ijhkl} = Variabel keputusan yang menunjukkan dua proses operasi O_{jh} dan O_{kl} pada mesin M_i

O_{jh} = Operasi ke-*h* yang berhubungan dengan pekerjaan *Jj* (*j* = 1,…, n, dan *h* =1,…, *hj*)

O_{kl} = Operasi ke-*l* yang berhubungan dengan pekerjaan *kj* (*j* = 1,…, n, dan *h* =1,…, *hj*)

Berdasarkan Rumus (3.1) yaitu jika nilai X_{ijh} sama dengan satu maka operasi O_{jh} memang benar ditugaskan ke mesin M_i, sedangkan jika nilai X_{ijh} sama dengan 0 maka operasi O_{jh} tidak ditugaskan ke mesin Mi. Berdasarkan Rumus (3.2) yaitu jika nilai Y_{ijhkl} sama dengan satu maka operasi O_{jh} mendahului operasi O_{kl} pada mesin M_i, sedangkan jika nilai Y_{ijhkl} sama dengan 0 maka operasi O_{jh} tidak mendahului operasi O_{kl} pada mesin M_i

Fungsi tujuan pada penelitian ini memiliki tiga fungsi tujuan yang dipertimbangkan secara bersamaan. Fungsi tujuan yang dimaksud yaitu Meminimumkan makespan atau waktu keseluruhan (Cm), Meminimalkan beban kerja maksimum dari sebuah mesin (Wm) dan Meminimalkan total waktu kerja seluruh mesin (Wt). Fungsi tujuan ini ditunjukkan dalam persamaan berikut:

$$C_{max} = \min\left(\max_{1 \le j \le n} (C_j)\right) \quad (3.3)$$

$$W_m = \min \left(\max_{1 \le i \le m} \sum_{j=1}^{n} \sum_{h=1}^{h_j} P_{ijh} X_{ijh} \right) \quad (3.4)$$

$$W_t = \min \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{h_j} P_{ijh} X_{ijh} \right) \quad (3.5)$$

Keterangan:

Cmax = Waktu penyelesaian total, yaitu waktu paling akhir dari semua pekerjaan.

Cj = Waktu penyelesaian dari pekerjaan j.

Wm = Beban kerja maksimum pada mesin.

P_{ijh} = Waktu pemrosesan operasi O_{jh} oleh mesin M_i

X_{ijh} = Varibel keputusan yang menunjukkan apakah operasi O_{jh} ditugaskan kedalam mesin M_i atau tidak.

Wt = Total beban kerja pada semua mesin.

m = Jumlah total mesin di lantai pabrik

n = Jumlah total pekerjaan

h_j = Jumlah operasi yang berhubungan dengan pekerjaan *Jj* (*j* = 1,…, n)

Alasan mengapa fungsi objektif ini dipilih adalah karena fungsi-fungsi ini tidak hanya merupakan perwakilan yang baik dari berbagai kriteria evaluasi yang dapat dipertimbangkan, tetapi juga saling bertentangan satu sama lain. Waktu proses yang singkat menunjukkan keluaran yang tinggi dari lantai produksi, yang dapat diperoleh dengan meningkatkan beban kerja pada mesin. Akibatnya, makespan yang minimum biasanya mengimplikasikan utilisasi mesin yang tinggi, terutama pada mesin-mesin yang sangat efisien, yang akan menyebabkan utilisasi mesin yang tidak seimbang. Dari Persamaan (3.4), meminimalkan beban kerja mesin sama dengan menyeimbangkan beban kerja di antara mesin-mesin. Selain itu, dalam keadaan mempertahankan makespan tidak berubah, karena mesin biasanya memiliki efisiensi yang tinggi, pengurangan beban kerja dari mesin akan mengakibatkan peningkatan total beban kerja mesin.

3.2.3 Tujuan Penelitian

Tujuan penelitian ini adalah melakukan pengembangan model algoritma genetika konvensional untuk optimasi masalah penjadwalan job shop fleksibel (FJSP). Meminimumkan makespan atau waktu keseluruhan (Cm), Meminimalkan beban kerja maksimum dari sebuah mesin (Wm) dan Meminimalkan total waktu kerja seluruh mesin (Wt).

3.3 Pengumpulan Data

Pengumpulan data yaitu mengumpulkan data yang diperlukan untuk kepentingan pengolahan data dan berhubungan dengan masalah yang menjadi objek penelitian. Data yang diperlukan dan dikumpulkan pada penelitian menggunakan data dari (Kacem et al., 2002). Data terdiri dari 4 skema yang merupakan masalah penjadwalan job shop fleksibel (FJSP) dengan masing – masing menunjukkan 4 job 5 mesin dengan 12 operasi, 10 job 7 mesin dengan 29 operasi, 10 job 10 mesin dengan 30 operasi dan 15 job 10 mesin dengan 56 operasi.

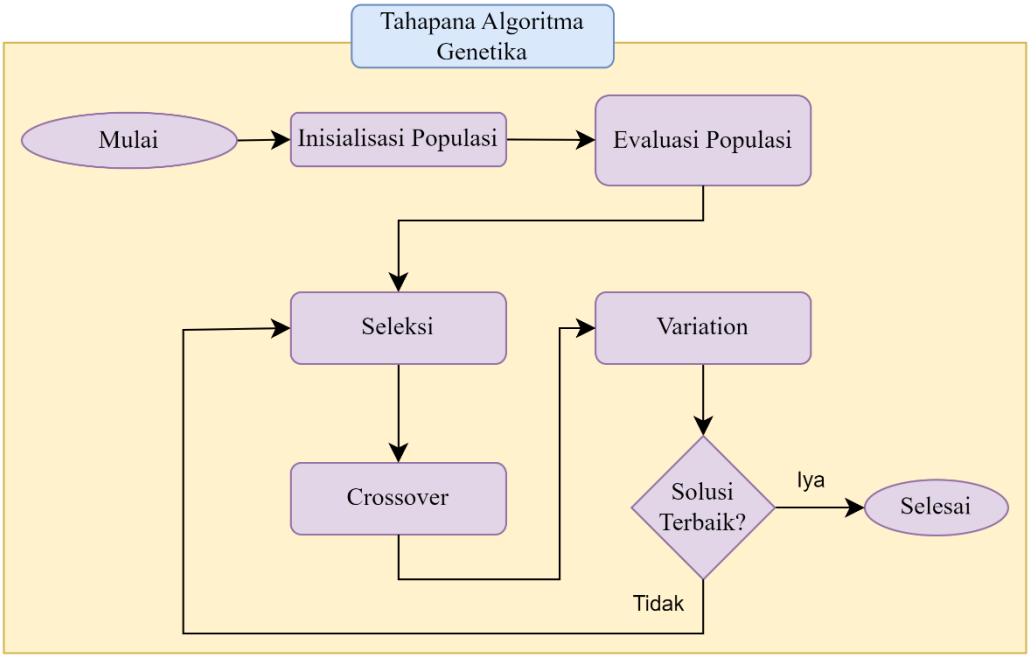
3.4 Pengolahan Data

Pengolahan data merupakan proses pengolahan data input menjadi output yang dilakukan sebagai dasar pemecahan masalah objek penelitian. Pengolahan data yang dilakukan pada penelitian ini menggunakan metode algoritma genetika konvensional dan pengembangan algoritma genetika yang telah ditetapkan untuk menyelesaikan masalah agar tujuan penelitian tercapai.

3.4.1 Algoritma Genetika Konvensional

Algoritma genetika berasal dari studi simulasi komputer yang dilakukan pada sistem biologis. Algoritma genetika merupakan metode pencarian dan pengoptimalan global stokastik yang dikembangkan untuk meniru mekanisme evolusi biologis di alam, mengacu pada teori evolusi Darwin dan doktrin genetika Mendel. Intinya adalah metode pencarian global yang efisien, paralel yang secara otomatis memperoleh dan mengumpulkan pengetahuan tentang ruang selama proses pencarian dan secara adaptif mengontrol proses pencarian untuk menemukan

solusi terbaik (Jiang et al., 2023). Berikut merupakan diagram alir algoritma genetika konvensional yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian
(Sumber: Jiang et al., 2023)

Langkah 1: Tetapkan jumlah maksimum iterasi (Max_Itertions) ke 0, tetapkan jumlah maksimum generasi evolusi ke Max, dan bangkitkan secara acak individu-individu Pop_size sebagai populasi awal P.

Langkah 2: Hitung fitness setiap individu dalam populasi P.

Langkah 3: Terapkan operator seleksi pada populasi. Tujuan dari seleksi adalah untuk meneruskan individu yang telah dioptimasi secara langsung ke generasi berikutnya atau menghasilkan individu baru dengan crossover berpasangan dan kemudian meneruskannya ke generasi berikutnya. Operasi seleksi didasarkan pada evaluasi kebugaran individu-individu dalam populasi.

Langkah 4: Terapkan operator crossover pada populasi. Operator crossover memainkan peran sentral dalam algoritma genetika.

Langkah 5: Terapkan operator variasi pada populasi. Artinya, ini adalah perubahan nilai gen pada fokus tertentu dari string individu dalam populasi. Populasi P dipilih dan disilangkan dan operator mutasi diterapkan untuk mendapatkan populasi P generasi berikutnya (Max_Itertions + 1).

Langkah 6: Tentukan apakah syarat-syarat untuk akhir algoritma terpenuhi; jika tidak terpenuhi, lanjutkan ke langkah 2-5, dan jika terpenuhi, akhiri algoritma dan keluarkan hasilnya.

3.4.2 Pengembangan Algoritma Genetika

1. Pengenalan

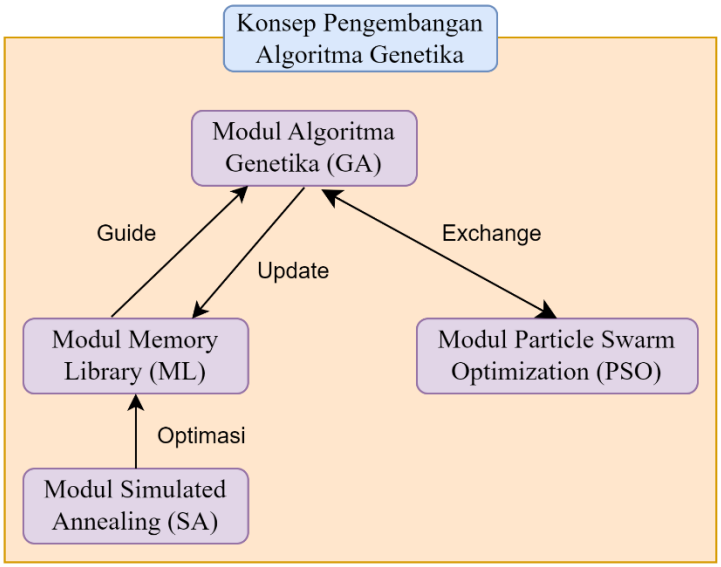
GA merupakan salah satu algoritma yang banyak digunakan untuk mengatasi MOFJSP. GA sangat penting untuk mencari ruang solusi yang besar untuk menentukan solusi yang mendekati optimal dan memiliki manfaat yang signifikan dibandingkan teknik optimasi yang lebih umum seperti pemrograman linier, heuristik, depth-first, dan sebagainya. Oleh karena itu, metode ini banyak digunakan untuk MOFJSP. Namun, ada dua batasan untuk GA asli untuk menyelesaikan MOFJSP, yang pertama adalah karena ukuran populasi GA yang terbatas, kompetisi yang berlebihan di antara individu dapat menyebabkan konvergensi dini GA, yang kedua yaitu pencariannya yang dilakukan secara acak, di mana GA tidak dapat memanfaatkan sepenuhnya pengetahuan yang diperoleh selama evolusi untuk mengarahkan pencarian ke wilayah dengan kinerja yang lebih baik, menyebabkan kecepatan konvergensi GA yang lambat. Karena masalah konvergensi dini dan kecepatan konvergensi GA yang lambat, maka GA tidak dapat langsung diterapkan untuk menyelesaikan masalah MOFJSP yang berskala besar. Bagaimana mengatasi konvergensi dini dan mempercepat konvergensi GA sambil mempertahankan kemampuan pencarian globalnya merupakan sebuah tantangan. Banyak literatur terbaru mengadaptasi GA atau dan memasukkan algoritma lain untuk mengatasi konvergensi dini dan mempercepat konvergensi.

PSO berbeda satu sama lain dalam hal mekanisme optimasi dan skema berbagi informasi. Di GA, informasi dibagikan antar kromosom. Melalui eksploitasi informasi yang dibagikan antar kromosom dan operasi seleksi, persilangan, dan mutasi pada kromosom. GA menghasilkan solusi generasi baru. Artinya, GA dapat mencari solusi dengan baik di seluruh ruang pencarian. Namun, GA tidak memiliki memori dan kemampuan belajar yang memadai bagi individu elit selama proses evolusi, sehingga mengakibatkan proses konvergensi yang relatif

lambat. Selain itu, kinerja GA mudah dipengaruhi oleh parameter dan berada dalam optimum lokal. Berbeda dengan GA, PSO memiliki kemampuan memori dan pembelajaran, dan skema berbagi informasinya bersifat satu arah. Dalam PSO, sebuah partikel memperoleh informasi tentang partikel terbaik dalam kawanannya dan posisi terbaiknya dalam sejarah partikel tersebut, serta memperbarui kecepatan dan posisinya hanya berdasarkan informasi ini. Hasilnya, PSO menyatu ke solusi yang memuaskan dengan kecepatan tinggi, namun juga mengurangi pencarian ketepatan. Oleh karena itu, PSO cocok untuk pencarian kasar dalam ruang solusi skala besar. Secara keseluruhan, dalam menyelesaikan MOFJSP, GA memiliki kemampuan pencarian global yang kuat tetapi konvergensinya lambat sedangkan PSO memiliki konvergensi yang cepat tetapi konvergensi akurasinya rendah.

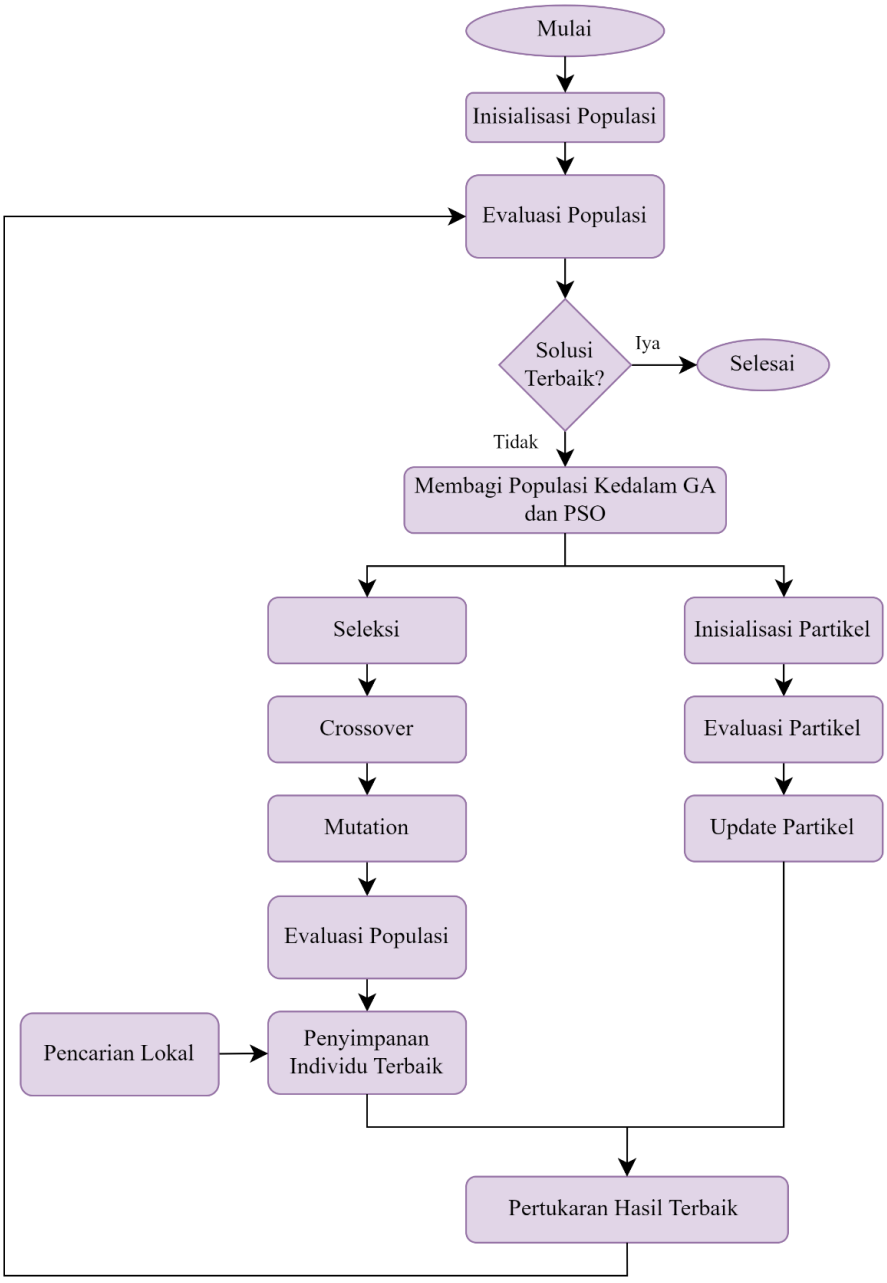
Sebagai perluasan dari algoritma pencarian lokal, dengan menggunakan kriteria penerimaan probabilitas Metropolis, SA memberikan proses pencarian probabilitas tertentu untuk menerima solusi yang lebih buruk. Hal ini memungkinkan algoritma untuk keluar dari optimal lokal. Namun, SA hanya mendapatkan satu solusi pada setiap iterasi. Oleh karena itu, sulit untuk menyelesaikan masalah berskala besar seperti MOFJSP secara efisien hanya dengan menggunakan SA.

Dalam penelitian ini, sebuah algoritma hibrida dengan mengintegrasikan GA, PSO dan SA diusulkan untuk mengatasi MOFJSP dengan mempertimbangkan waktu transportasi dan waktu setup. Algoritma ini terdiri dari modul GA, modul memory library (ML), modul PSO dan modul SA dapat dilihat pada Gambar 3.3



Gambar 3.3 Konsep Pengembangan Algoritma Genetika

Berdasarkan Gambar 3.3 pada modul GA, operator genetika dimodifikasi untuk meningkatkan kinerja GA untuk MOFJSP. Pada modul PSO tujuan nya yaitu untuk meningkatkan kecepatan konvergensi GA dengan menggunakan kemampuan pencarian cepat PSO, dan di sisi lain, untuk menghindari GA jatuh ke dalam optimum lokal sampai batas tertentu. Sementara itu, kemampuan pencarian global yang kuat dari GA digunakan untuk meningkatkan akurasi pencarian PSO. Melalui pertukaran informasi timbal balik, modul GA dan modul PSO meningkatkan kualitas populasinya. Selain itu, karena GA tidak memiliki kemampuan untuk belajar dari individu elit, dalam modul ML, sebuah ML elitisme eksternal dirancang untuk menyimpan individu elit yang dihasilkan selama proses evolusi, dan dapat digunakan sebagai pustaka pengetahuan untuk mengarahkan pencarian GA ke wilayah yang memiliki kinerja yang lebih baik. Selain itu, untuk mengatasi konvergensi dini GA, SA dimasukkan ke dalam untuk melakukan pencarian lokal untuk modul ML. Singkatnya tujuan dari pengembangan ini untuk mencegah konversi dini dan kecepatan konvergensi yang lambat pada algoritma genetika. Diagram alir pengembangan algoritma genetika konvensional ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram Alir Pengembangan Algoritma Genetika

1. Inisialisasi Populasi

   Menginisialisasi populasi awal, dimulai dengan menghasilkan populasi awal secara acak. Setiap individu dalam populasi ini adalah solusi potensial untuk masalah MOFJSP, di mana setiap individu diwakili oleh kromosom yang mengkodekan urutan operasi, alokasi mesin, waktu transportasi, dan waktu setup.

2. Evaluasi Populasi

   Menghitung fitness function**,** setiap individu dievaluasi menggunakan fitness function yang mempertimbangkan fungsi objektif yang digunakan. Nilai fitness ini menentukan seberapa baik setiap solusi menyelesaikan masalah yang diberikan.

3. Kriteria Penghentian, jika evaluasi populasi telah mendapatkan solusi terbaik maka hasil akan ditampilkan, apabila evaluasi populasi tidak mencapai solusi terbaik maka perhitungan akan berlanjut.

4. Modul GA

   Seleksi Individu Terbaik, Setelah populasi dievaluasi, individu-individu terbaik dipilih menggunakan metode seleksi seperti Roulette Wheel Selection, Tournament Selection, atau lainnya.

   Crossover (Penyilangan), Pasangan individu dipilih dari populasi yang tersaring dan mengalami crossover untuk menghasilkan individu baru. Crossover menggabungkan dua kromosom parents untuk menghasilkan keturunan yang mungkin memiliki karakteristik lebih baik dari kedua parents.

   Mutasi, setelah crossover, beberapa individu akan mengalami mutasi, di mana bagian dari kromosom diubah secara acak. Mutasi ini bertujuan untuk memperkenalkan variasi dalam populasi dan membantu menghindari konvergensi prematur.

   Evaluasi Populasi, Individu baru hasil dari crossover dan mutasi dievaluasi kembali untuk menentukan nilai fitness mereka

5. Modul ML

   Individu terbaik dari generasi saat ini disimpan dalam Memory Library (ML). ML bertindak sebagai repositori elitisme eksternal, yang menyimpan solusi berkualitas tinggi untuk digunakan kembali.

6. Modul SA

   Mencegah GA dari konvergensi prematur ke solusi lokal yang tidak optimal, Simulated Annealing (SA) diterapkan pada individu-individu yang tersimpan dalam ML. SA melakukan pencarian lokal dengan probabilitas menerima solusi yang lebih buruk di awal dan secara bertahap mengurangi probabilitas tersebut

seiring waktu. Solusi yang ditingkatkan melalui SA disimpan kembali dalam ML, memperkaya pustaka solusi elit.

7. Modul PSO

   Inisialisasi Partikel, beberapa individu dari populasi digunakan untuk menginisialisasi partikel dalam PSO. Setiap partikel mewakili solusi potensial yang akan ditingkatkan melalui pencarian PSO.

   Evaluasi Partikel**,** Setiap posisi baru dari partikel dievaluasi, dan jika posisi tersebut lebih baik, maka posisi tersebut akan diupdate sebagai personal best atau global best.

   Update Posisi dan Kecepatan**,** Partikel-partikel ini bergerak dalam ruang solusi, di mana kecepatan dan arah mereka diperbarui berdasarkan personal best (posisi terbaik yang pernah dicapai oleh partikel itu sendiri) dan global best (posisi terbaik yang pernah dicapai oleh semua partikel).

8. Pertukaran Informasi

   Informasi dari hasil pencarian PSO ditransfer kembali ke GA untuk memperbaiki individu dalam populasi GA. Sebaliknya, solusi dari GA dapat digunakan untuk memperbaiki posisi partikel dalam PSO. Hal ini membantu kedua modul meningkatkan kualitas populasi secara keseluruhan.

3.5 Hasil dan Pembahasan

Analisa merupakan uraian dari hasil pengolahan data. Pada langkah ini dilakukan analisa lebih lanjut terhadap hasil penjadwalan yang dilakukan menggunakan pengembangan algoritma genetika. Kemudian hasil penjadwalan tersebut digambarkan dalam diagram gant chart.

3.6 Kesimpulan dan Saran

Kesimpulan merupakan hasil yang disimpulkan dari hasil data yang diperoleh pada kegiatan penelitian berupa pernyataan singkat yang beracuan pada analisa hasil penelitian untuk menjawab tujuan penelitian. Saran merupakan pendapat yang berkaitan dengan solusi masalah yang menjadi objek penelitian untuk penerapan dan pengembangan penelitian berikutnya.

3.7 Rencana Jadwal Kegiatan

Rencana jadwal kegiatan dibuat bertujuan untuk menentukan rencana waktu suatu kegiatan yang menunjang penelitian ini dilakukan. Berikut ini merupakan Tabel rencana jadwal kegiatan pada penelitian ini.

Kegiatan	**Tahun Pertama (Semester 1 dan 2)**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Studi Literatur												
Pembuatan Proposal												
Kegiatan	**Tahun Kedua (Semester 3 dan 4)**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Pengumpulan Data dan Pengembangan Model												
Pengujian dan Evaluasi Model												
Kegiatan	**Tahun Ketiga (Semester 5 dan 6)**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Penulisan Disertasi												
Penulisan Jurnal Pertama												
Publish Jurnal Pertama												
Penulisan Jurnal Kedua												
Publish Jurnal Kedua												

Daftar Pustaka

Amjad, M. K., Butt, S. I., Kousar, R., Ahmad, R., Agha, M. H., Faping, Z., Anjum, N., & Asgher, U. (2018). Recent Research Trends in Genetic Algorithm Based Flexible Job Shop Scheduling Problems. In *Mathematical Problems in Engineering* (Vol. 2018). Hindawi Limited. https://doi.org/10.1155/2018/9270802

Baker, K. R., & Trietsch, D. (2009). *Principles of Sequencing and Scheduling*. Wiley. https://doi.org/10.1002/9780470451793

Defersha, F. M., & Rooyani, D. (2020). An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time. *Computers and Industrial Engineering*, *147*. https://doi.org/10.1016/j.cie.2020.106605

Herjanto, E. (2017). *Manajemen Operasi* (3nd ed.). Grasindo.

Huang, X., Guan, Z., & Yang, L. (2018). An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem. *Advances in Mechanical Engineering*, *10*(9). https://doi.org/10.1177/1687814018801442

Huang, X., & Yang, L. (2019). A hybrid genetic algorithm for multi-objective flexible job shop scheduling problem considering transportation time. *International Journal of Intelligent Computing and Cybernetics*, *12*(2), 154–174. https://doi.org/10.1108/IJICC-10-2018-0136

Jamrus, T., Chien, C. F., Gen, M., & Sethanan, K. (2018). Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Time for Semiconductor Manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, *31*(1), 32–41. https://doi.org/10.1109/TSM.2017.2758380

Jiang, M., Yu, H., & Chen, J. (2023). Improved Self-Learning Genetic Algorithm for Solving Flexible Job Shop Scheduling. *Mathematics*, *11*(22). https://doi.org/10.3390/math11224700

Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. In *Mathematics and Computers in Simulation* (Vol. 60).

Liang, X., Liu, Y., Gu, X., Huang, M., & Guo, F. (2022). Adaptive Genetic Algorithm Based on Individual Similarity to Solve Multi-Objective Flexible Job-Shop Scheduling Problem. *IEEE Access*, *10*, 45748–45758. https://doi.org/10.1109/ACCESS.2022.3170032

Liaqait, R. A., Hamid, S., Warsi, S. S., & Khalid, A. (2021). A critical analysis of job shop scheduling in context of Industry 4.0. In *Sustainability (Switzerland)* (Vol. 13, Issue 14). MDPI AG. https://doi.org/10.3390/su13147684

Li, B., & Xia, X. (2022). A Self-Adjusting Search Domain Method-Based Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/4212556

Liu, Z., Wang, J., Zhang, C., Chu, H., Ding, G., & Zhang, L. (2021). A hybrid genetic-particle swarm algorithm based on multilevel neighbourhood structure for flexible job shop scheduling problem. *Computers and Operations Research*, *135*. https://doi.org/10.1016/j.cor.2021.105431

Luo, G., Song, J., Zhang, Z., & Li, J. (2018). Solving Flexible Job Shop Scheduling Problem based on Improved Genetic Algorithm. *IOP Conference Series: Materials Science and Engineering*, *394*(3). https://doi.org/10.1088/1757-899X/394/3/032135

Meng, L., Cheng, W., Zhang, B., Zou, W., Fang, W., & Duan, P. (2023). An Improved Genetic Algorithm for Solving the Multi-AGV Flexible Job Shop Scheduling Problem. *Sensors*, *23*(8). https://doi.org/10.3390/s23083815

Nasution, A. H. (1999). *Perencanaan dan Pengendalian Produksi*. Gema Widya.

Park, J. S., Ng, H. Y., Chua, T. J., Ng, Y. T., & Kim, J. W. (2021). Unified genetic algorithm approach for solving flexible job-shop scheduling problem. *Applied Sciences (Switzerland)*, *11*(14). https://doi.org/10.3390/app11146454

Schroeder, R. G. (2020). *Operations Management: Contemporary Concepts and Cases* (12nd ed.). McGraw-Hill International Edition.

Silitonga, A. I. (2021). *Uniform Crossover Pada Algoritma Genetika Untuk Mereduksi Noise Pada*. Agnes Irene Silitonga.

Suradi. (2023). *Sistem Produksi*. Tohar Media.

Tay, J. C., & Wibowo, D. (2004). An Effective Chromosome Representation for Evolving Flexible Job Shop Schedules. In K. Deb (Ed.), *Genetic and Evolutionary Computation – GECCO 2004* (pp. 210–221). Springer Berlin Heidelberg.

Utama, D. M. (2023). *Penjadwalan Teori dan Aplikasi*. UMMPress.

Wang, Y., & Zhu, Q. (2021). A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem with Sequence-Dependent Setup Times and Job Lag Times. *IEEE Access*, *9*, 104864–104873. https://doi.org/10.1109/ACCESS.2021.3096007

Wu, R., Li, Y., Guo, S., & Xu, W. (2018). Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. *Advances in Mechanical Engineering*, *10*(10). https://doi.org/10.1177/1687814018804096

Zhang, G., Hu, Y., Sun, J., & Zhang, W. (2020). An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. *Swarm and Evolutionary Computation*, *54*. https://doi.org/10.1016/j.swevo.2020.100664

Zhang, G., Sun, J., Liu, X., Wang, G., & Yang, Y. (2019). Solving flexible job shop scheduling problems with transportation time based on improved genetic algorithm. *Mathematical Biosciences and Engineering*, *16*(3), 1334–1347. https://doi.org/10.3934/mbe.2019065

Zhang, G., Zhang, L., Song, X., Wang, Y., & Zhou, C. (2019). A variable neighborhood search based genetic algorithm for flexible job shop scheduling problem. *Cluster Computing*, *22*, 11561–11572. https://doi.org/10.1007/s10586-017-1420-4