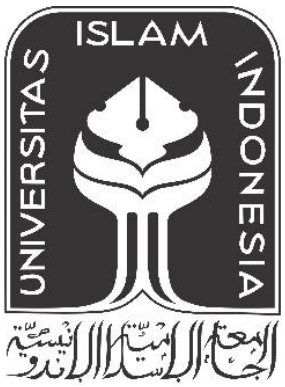
OPTIMASI MULTI-OBYEKTIF NSGA-II PADA PENJADWALAN PRODUKSI UNTUK MINIMALISASI *MAKESPAN* DAN BIAYA KETERLAMBATAN

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Program Studi Teknik Industri - Fakultas Teknologi Industri
Universitas Islam Indonesia



Nama : Ayu Oktamariska Putri
No. Mahasiswa : 20522312

PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2024

PERNYATAAN KEASLIAN

ii

PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 25 – 10 - 2024

(Ayu Oktamariska Putri)
NIM. 20522312

SURAT BUKTI PENELITIAN

INKA

SURAT KETERANGAN

No. 508/SKKP/221/INKA/2023

Yang bertanda tangan di bawah menerangkan bahwa :

Nama	: AYU OKTAMARISKA PUTRI
NIM	: 20522312
Institusi	: UNIVERSITAS ISLAM INDONESIA
Jurusan	: TEKNIK INDUSTRI

Telah selesai melaksanakan KERJA PRAKTEK di PT Industri Kereta Api (Persero) Madiun, mulai tanggal 01 Oktober 2023 s/d 30 November 2023.

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Madiun, 30 November 2023
PT INDUSTRI KERETA API (Persero)
SM Pengelolaan Organisasi & SDM

Sigit Sugiarto

PT INDUSTRI KERETA API (Persero)
Kantor Pusat : Jl. Yos Sudarso No. 71 Madiun, Telp. (62-351) 452271 - 74, Facs. (62-351) 452275, Website : www.inka.co.id, email : sekretariat@inka.co.id
kantor Perwakilan : Jl. Menara Taspen Lt. 3 Jl. Jend Sudirman Kav. 2 Jakarta, Telp.(62-21) 2514424, Facs. (62-21) 2514423 email : inkajkt@inka.co.id

LEMBAR PENGESAHAN PEMBIMBING

OPTIMASI MULTI-OBYEKTIF NSGA-II PADA PENJADWALAN PRODUKSI UNTUK MINIMALISASI *MAKESPAN* DAN BIAYA KETERLAMBATAN

TUGAS AKHIR

Disusun Oleh :

Nama : Ayu Oktamariska Putri

No. Mahasiswa : 20522312

Yogyakarta, 25 Oktober 2024

Dosen Pembimbing

(Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM)

LEMBAR PENGESAHAN DOSEN PENGUJI

OPTIMASI MULTI-OBYEKTIF NSGA-II PADA PENJADWALAN PRODUKSI UNTUK MINIMALISASI *MAKESPAN* DAN BIAYA KETERLAMBATAN
TUGAS AKHIR

Disusun Oleh :
Nama : Ayu Oktamariska Putri
No. Mahasiswa : 20 522 312

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 25 - 10 – 2024

Tim Penguji

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM Ketua	
Ir. Vembri Noor Helia, S.T., M.T., IPM Anggota I	
Wahyudhi Sutrisno, S.T., M.M., M.T Anggota II	

Mengetahui,
Ketua Program Studi Teknik Industri Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM
NIK. 015220101

HALAMAN PERSEMBAHAN

Karya sederhana ini saya persembahkan kepada Ayah dan Ibu, sumber inspirasi dan kekuatan tak terhingga. Terima kasih atas doa, dukungan, dan cinta yang tak pernah lekang.

MOTTO

"Urip iku urup"

(Filosofi Jawa)

"Dan bersabarlah kamu, sesungguhnya janji Allah adalah benar."

(Q.S Ar-Rum: 60)

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, Tuhan Yang Maha Esa, atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan judul "Optimasi Multi-Obyektif NSGA-II pada Penjadwalan Produksi untuk Minimalisasi *Makespan* dan Biaya Keterlambatan". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Industri di Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulis menyadari bahwa penyelesaian skripsi ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo., M.T., IPU, ASEAN. Eng, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia atas kesempatan dan fasilitas yang diberikan kepada penulis selama menempuh pendidikan di fakultas ini.
2. Bapak Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., selaku Ketua Program Studi Teknik Industri, Universitas Islam Indonesia, atas arahan dan dukungannya selama penulis menempuh studi.
3. Bapak Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D. selaku Dosen Pembimbing, yang dengan sabar dan penuh dedikasi telah memberikan bimbingan, arahan, koreksi, serta motivasi kepada penulis dalam menyelesaikan skripsi ini.
4. Bapak Eko Purwanto selaku Direktur Utama PT INKA (Persero), yang telah memberikan izin dan kesempatan kepada penulis untuk melakukan penelitian di perusahaan tersebut.
5. Ibu Ana selaku pembimbing lapangan di PT INKA (Persero), yang telah memberikan bimbingan dan arahan selama penulis melakukan penelitian.
6. Kedua orang tua tercinta, Iskandar Maliki dan Sri Margowati, yang senantiasa memberikan doa, dukungan, kasih sayang, dan pengorbanan yang tak terhingga kepada penulis.
7. Teman – teman saya, terima kasih atas semangat, dukungan, dan bantuan yang telah diberikan kepada penulis.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan teknologi, khususnya di bidang optimasi penjadwalan produksi.

Yogyakarta, 25 Oktober 2024

Ayu Oktamariska Putri

ABSTRAK

Penjadwalan produksi yang efisien merupakan faktor krusial dalam mencapai target produksi dan meminimalkan biaya. Penelitian ini dilatarbelakangi oleh kompleksitas penjadwalan produksi yang menyebabkan penambahan durasi penyelesaian atau *makespan* sehingga menimbulkan adanya biaya untuk setiap keterlambatan yang terjadi. Oleh karena itu, sebuah penelitian diusulkan untuk menyusun penjadwalan yang melibatkan berbagai tujuan, yaitu meminimalkan *makespan* dan biaya keterlambatan. Tujuan penelitian ini adalah mengembangkan sistem penjadwalan produksi yang optimal dengan mempertimbangkan kedua tujuan tersebut secara simultan. Metode yang digunakan adalah algoritma genetika *Non-dominated Sorting Genetic Algorithm* II (NSGA-II) yang mampu mencari solusi optimal *Pareto* untuk permasalahan multi-obyektif. Implementasi NSGA-II pada penjadwalan produksi telah menghasilkan sejumlah solusi *Pareto* yang merepresentasikan kombinasi optimal antara *makespan* dan biaya keterlambatan. Hasil penelitian menunjukkan bahwa NSGA-II efektif dalam menemukan solusi penjadwalan produksi yang lebih baik dibandingkan dengan metode yang ada sebelumnya. Parameter yang membuktikan adalah adanya pengurangan dirasi penyelesaian dan biaya keterlambatan. Solusi-solusi *Pareto* dihasilkan menggunakan algoritma NSGA-II dalam waktu yang singkat dan memberikan alternatif penjadwalan yang dapat dipilih oleh pengambil keputusan dengan tetap menyesuaikan prioritas dan kebutuhan produksi yang ada di perusahaan. Penelitian ini memberikan kontribusi dalam meningkatkan efisiensi penjadwalan produksi dan memberikan rekomendasi bagi perusahaan dalam mengambil keputusan penjadwalan yang lebih baik dengan tujuan-tujuan lain yang dapat disesuaikan.

Kata Kunci : *Penjadwalan Produksi, NSGA-II, Optimasi Multi-Obyektif, Makespan, Biaya Keterlambatan*

.

DAFTAR ISI

PERNYATAAN KEASLIAN ..... **ii**
SURAT BUKTI PENELITIAN ..... **iii**
LEMBAR PENGESAHAN PEMBIMBING ..... **iv**
LEMBAR PENGESAHAN DOSEN PENGUJI ..... **v**
HALAMAN PERSEMBAHAN ..... **vi**
MOTTO ..... **vii**
KATA PENGANTAR ..... **viii**
ABSTRAK ..... **ix**
DAFTAR ISI ..... **x**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Tujuan Penelitian 5
1.4 Manfaat Penelitian 5
1.5 Batasan Penelitian 6
BAB II TINJAUAN PUSTAKA **7**
2.1 Kajian Literatur 7
2.2 Landasan Teori 15
2.2.1 Penjadwalan Produksi 16
2.2.2 Definisi dalam Penjadwalan Produksi 17
2.2.3 Penjadwalan Flowshop 20
2.2.4 Optimasi Multi-Obyektif 22
2.2.5 Non-Dominated Sorting Genetic Algorithm II (NSGA-II) 25
2.2.6 Pustaka Distributed Evolutionary Algorithm in Python (DEAP) 34
BAB III METODE PENELITIAN **38**
3.1 Obyek Penelitian 38
3.2 Jenis Data Penelitian 39
3.2.1 Data Primer 39
3.2.2 Data Sekunder 39
3.3 Alur Penelitian 39
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA **44**
4.1 Pengumpulan Data 44
4.1.1 Proses Produksi 44
4.1.2 Data Pekerjaan 46
4.1.3 Biaya Keterlambatan 47
4.1.4 Penjadwalan Yang Berlaku 47
4.2 Pengolahan Data 49
4.2.1 Identifikasi Keterlambatan 49
4.2.2 Pengaturan Parameter Algoritma NSGA-II 51
4.2.3 Penjadwalan dengan Algoritma NSGA-II 55
BAB V PEMBAHASAN **69**
5.1 Analisis Performa Algoritma NSGA-II 69
5.1.1 Persebaran Pareto Front 69

5.1.2 Analisis *Hypervolume* 70
5.2 Analisis Penjadwalan oleh NSGA-II 72
BAB VI PENUTUP 78
6.1 Kesimpulan 78
6.2 Saran 78
DAFTAR PUSTAKA 80
LAMPIRAN A-1

DAFTAR TABEL

Tabel 2.1 Kajian Literatur 15
Tabel 2.2 Metode *Flowshop Scheduling* 21
Tabel 2.3 Pendekatan pada Metode Posteriori 24
Tabel 4.1 Stasiun Kerja 46
Tabel 4.2 Biaya Keterlambatan Mesin 47
Tabel 4.3 Penjadwalan Berlaku dan Realisasi Penjadwalan 48
Tabel 4.4 Identifikasi Keterlambatan 49
Tabel 4.5 Translasi Keluaran *Script* 63
Tabel 4.6 Nilai *Fitness* Seluruh Generasi NSGA-II 63
Tabel 4.7 Nilai *Hypervolume* NSGA-II 67
Tabel 5.1 Persebaran *Pareto* Generasi 80 69
Tabel 5.2 Persebaran *Pareto* Generasi 81 70
Tabel 5.3 Rangkuman *Timeline* Penjadwalan Usulan 74
Tabel 5.4 Identifikasi Keterlambatan Penjadwalan Usulan 75

DAFTAR GAMBAR

Gambar 1.1 Perbandingan Tren Pengguna dan Performansi Produsen Kereta Api 2
Gambar 2.1 Klasifikasi Metode Heuristik 22
Gambar 2.2 Kurva Pareto 24
Gambar 2.3 Visual *Non-dominated Sorting* 26
Gambar 2.4 Visual *Crowding Distance* 27
Gambar 2.5 Alur *Fast Non-dominated Sorting* 30
Gambar 2.6 Iterasi Secara Matematis 33
Gambar 2.7 Prosedur NSGA-II 33
Gambar 3.1 Diagram Alur Penelitian 40
Gambar 4.1 *Script* Inisialisasi Populasi NSGA-II 55
Gambar 4.2 *Script* Evalualsi Populasi NSGA-II 58
Gambar 4.3 *Script Fast Non-Dominated Sorting* NSGA-II 59
Gambar 4.4 Script *Crowding Distance Assignment* NSGA-II 60
Gambar 4.5 *Script Crowded Tournamnet Selection* NSGA-II 61
Gambar 4.6 *Script Crossover* NSGA-II 61
Gambar 4.7 *Script* Mutasi NSGA-II 62
Gambar 4.8 *Script* Kalkulasi *Hypervolume* NSGA-II 62
Gambar 5.1 Perbandingan Persebaran Nilai Pareto Antar Generasi 70
Gambar 5.2 Tren Nilai *Hypervolume* Seluruh Generasi 71
Gambar 5.3 Statistik Nilai Obyektif *Makespan* Seluruh Generasi 71
Gambar 5.4 Statistik Nilai Obyektif Biaya Keterlambatan Seluruh Generasi 72
Gambar 5.5 *Gantt Chart* Penjadwalan Usulan 73
Gambar 5.6 *Gantt Chart* Penjadwalan Yang Berlaku 73

BAB I

PENDAHULUAN

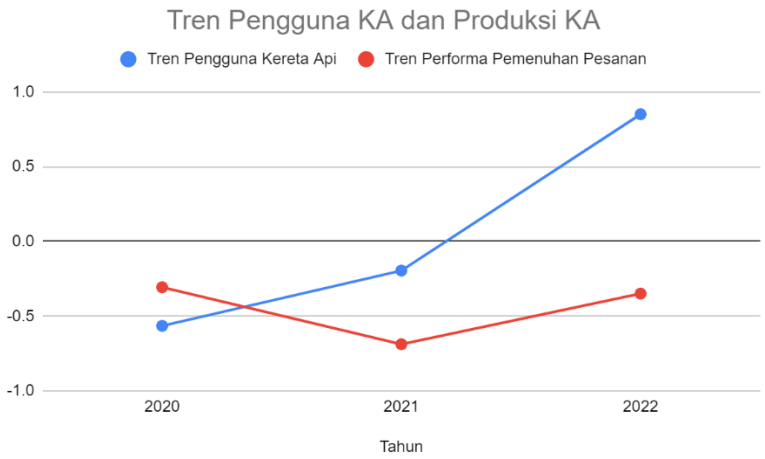
1.1 Latar Belakang

Perkembangan industri telah mencapai revolusi keempatnya yang eranya disebut industri 4.0. Revolusi tersebut terjadi seiring perkembangan dan penemuan teknologi baru. Penemuan teknologi baru memerlukan banyak persyaratan, salah satunya adalah kemahiran transportasi. Hal tersebut berkaitan dengan industri sebagai pemakai teknologi yang memiliki kondisi alamiahnya yaitu tersebar secara geografis guna menyesuaikan target pasarnya (Szostak, 1991). Transportasi adalah pemindahan manusia atau barang dari satu tempat ke tempat lainnya dalam waktu tertentu dengan menggunakan sebuah kendaraan yang digerakkan oleh manusia, hewan, maupun mesin (Kemendikbud-Ristek, 2016). Salah satu jenis transportasi adalah transportasi umum, yang merupakan layanan angkutan penumpang oleh sistem perjalanan kelompok yang tersedia untuk digunakan oleh masyarakat umum pada lingkup lokal maupun regional (Carter *et al*., 2004).

Penggunaan transportasi umum di Indonesia memberikan dampak yang signifikan terhadap pertumbuhan perekonomian di Indonesia dengan koefisien determinasi sebesar 99% (Kartiasih, 2019). Di Indonesia sendiri, terdapat berbagai macam transportasi umum yang tersedia, salah satunya adalah kereta api. Kereta api adalah transportasi yang terdiri atas rangkaian gerbong (kereta) yang ditarik oleh lokomotif kemudian dijalankan dengan tenaga uap atau listrik dan berjalan di atas rel yang digunakan sebagai transportasi umum (Kemendikbud-Ristek, 2016). Kereta api memberikan kontribusi nyata terhadap pertumbuhan perekonomian di Indonesia berkaitan dengan peningkatan produktivitas penumpang akibat waktu tempuh untuk perpindahan ke satu tempat menjadi lebih sedikit (Andiyan & Rachmat, 2021). Seiring dengan kemudahan yang ditawarkan, jumlah

penumpang kereta api juga meningkat, setidaknya untuk 2 (dua) tahun terakhir. Total jumlah penumpang pada tahun 2021 adalah 149,763 penumpang, sedangkan pada tahun 2022 sebanyak 277,115 penumpang (PT. KAI, 2023) tahu meningkat 85%. Peningkatan jumlah ini merupakan sinyal bagi penyedia layanan kereta api di Indonesia untuk memastikan kebutuhan infrastrukturnya terpenuhi. Salah satu produsen dalam negeri untuk infrastruktur kereta api yang menyuplai layanan kereta api di Indonesia adalah PT. Industri Kereta Api (PT. INKA).

PT. INKA (Persero) merupakan salah satu Badan Usaha Milik Negara (BUMN) yang berfokus pada manufaktur sarana kereta api terintegrasi, seperti lokomotif, kereta penumpang, kereta penggerak, gerbong barang, kereta khusus, dan produk pengembangan. Berdasarkan laporan tahunan untuk kinerja penjualan di PT. INKA, pertumbuhan rata-rata penjualan selama 5 (lima) tahun terakhir bernilai negatif atau mengalami penurunan (PT. INKA, 2022). Hal tersebut dapat dibandingkan dengan jumlah pengguna kereta api yang meningkat 85% pada tahun 2022, sedangkan performa pemenuhan pesanan menurun sebesar 34% pada tahun yang sama.



Gambar 1.1 Perbandingan Tren Pengguna dan Performansi Produsen Kereta Api

Banyak faktor yang menyebabkan kelesuan tersebut baik dari internal perusahaan maupun eksternal perusahaan. Salah satu faktor internal yang menyebabkan kelesuan tersebut adalah kurangnya pengendalian terhadap berjalannya proses produksi yang disebabkan oleh ketidaktahuan dalam penentuan *lead time* pengadaan pada jadwal

produksi dan kesalahan estimasi ketersediaan *tool* atau fasilitas produksi dan penumpukan material / *part* (Penumpukan *Working Sequence*) (PT. INKA, 2022). Kebijakan yang diambil oleh perusahaan untuk menangani kasus tersebut adalah dengan melakukan *review* bersama fungsi terkait mengenai *update lead time* kedatangan material berdasarkan kondisi terakhir dan melakukan *review* kebutuhan *tools* dan *consumable part* serta *monitoring* harian terhadap *load* di *station* kerja dan mengatur *load* permintaan material produksi (PT. INKA, 2022).

Berdasarkan permasalahan tersebut, sebuah penelitian disusun untuk mengatasi tantangan yang dihadapi oleh PT. INKA dalam penjadwalan produksi dan pengendalian biaya. Beberapa penelitian telah mengembangkan model yang berkaitan dengan penjadwalan mesin produksi *flow shop* untuk *multi-objective*. Masalah penjadwalan *flowshop* yang terdiri dari banyak mesin dan pekerjaan mengakibatkan kemungkinan kombinasi semakin besar seiring pertambahan jumlah masukannya. Oleh karena itu, masalah optimasinya dikategorikan sebagai masalah optimasi kombinasi yang termasuk dalam kelas NP-*Hard* dan memerlukan metode penemuan solusi optimal terdekat untuk memecahkannya seperti pendekatan metaheuristik (Yagmahan & Yenisey, 2008). Pendekatan metaheuristik tidak menjamin penemuan solusi teroptimal namun solusi optimal pada waktu yang paling wajar, salah satu contohnya adalah algoritma genetika. Penelitian penjadwalan dengan 2 (dua) objektif yaitu minimalisasi total keterlambatan dan waktu *setup* yang dilakukan oleh Yu *et al* (2020) pada aliran *flow shop* telah berhasil menemukan penjadwalan teroptimal menggunakan NSGA-II. Begitu pula, penelitian yang dilakukan oleh Anjana *et al* (2020) pada aliran produksi *flow shop* dengan obyektif minimalisasi *makespan* dan rerata keterlambatan menggunakan NSGA-II yang dikombinasikan dengan VNS juga menunjukkan hasil positif untuk kedua obyektifnya. Selain itu, penelitian oleh Kusuma (2021) dan Valledor *et al* (2022) pada aliran *flow shop* pula telah menunjukkan hasil yang baik untuk optimalisasi penjadwalan pada obyektif minimalisasi *makespan*, biaya produksi, dan biaya energi. Secara umum, penelitian-penelitian tersebut telah menunjukkan hasil positif untuk optimalisasi produksi. Namun, diantara penelitian yang telah ada belum ada yang secara spesifik

mengombinasikan obyektif *makespan* dan biaya *penalty* akibat keterlambatan pada industri manufaktur kereta api.

Dalam kasus optimasi *multi-objective*, algoritma *Non-Dominated Sorting Genetic Algorithm for Multi-objective Optimization*: NSGA-II yang merupakan kelompok algoritma *metaheuristic* yang telah diuji keandalannya dibandingkan dengan optimasi *multi-objective* lainnya. Meskipun algoritma genetika (*Genetic Algorithms*/GAs) dapat digunakan untuk memecahkan masalah tersebut, namun perlu diwaspadai adanya konvergensi dini ke daerah optimum lokal dan penurunan kecepatan pencarian saat mendekati titik solusi (Mahmudy & Rahman, 2011). Oleh karena itu, NSGA-II sebagai metode pengembangan dari *Genetic Algorithm* (GA) dan NSGA melalui penerapan prinsip *elitism* dan operator *crowding distance* agar menghasilkan solusi pareto optimal yang lebih baik (Deb *et al*., 2002). Penerapan prinsip *elitism* memastikan bahwa solusi terbaik dari satu generasi diwariskan ke generasi berikutnya sehingga solusi yang baik tetap dipertahankan dan mencegah konvergensi dini ke optimum lokal. Kemudian dengan operator *crowding distance* yang mengukur kerapatan solusi di sekitar suatu solusi target untuk memprioritaskan solusi di daerah yang kurang padat, NSGA-II mencegah konvergensi dini dan memastikan variasi solusi yang lebih baik. Berdasarkan pertimbangan tersebut, penelitian ini mengusulkan untuk memanfaatkan keandalan metode NSGA-II untuk mengatasi masalah keterlambatan pada proses manufaktur kereta api di PT. INKA dengan obyektifnya minimalisasi *makespan* dan biaya keterlambatan. Penelitian ini diharapkan dapat memberikan solusi praktis dan efektif untuk tantangan yang dihadapi oleh PT. INKA dan industri manufaktur kereta api pada umumnya.

1.2 Rumusan Masalah

Permasalahan keterlambatan penyelesaian pesanan memerlukan analisis lebih lanjut untuk mengukur keparahan fenomena dan tingkat keberhasilan pasca penelitian yang diusulkan , untuk menemukan jawaban tersebut maka penelitian ini akan menjawab setiap rumusan masalah berikut:

1. Bagaimana kondisi penjadwalan produksi saat ini di PT. INKA dan bagaimana dampaknya terhadap biaya keterlambatan dan *makespan*?

2. Bagaimana implementasi metode terpilih dapat meningkatkan efisiensi dan produktivitas di PT. INKA?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disusun, maka jawaban yang diharapkan muncul akan dapat memenuhi tujuan penelitian berikut:

1. Menganalisis kondisi penjadwalan produksi saat ini di PT. INKA dan dampaknya terhadap biaya keterlambatan dan *makespan*.
2. Membuat rekomendasi implementasi metode terpilih ini untuk meningkatkan efisiensi dan produktivitas di PT. INKA

1.4 Manfaat Penelitian

Berdasarkan rumusan masalah dan tujuan penelitian yang telah diuraikan, berikut adalah beberapa manfaat yang diharapkan dari penelitian ini:

1. Manfaat Bagi Perusahaan (PT. INKA)
    a. Penelitian ini dapat membantu PT. INKA dalam mengoptimalkan penjadwalan produksi dan mengurangi biaya penalti, sehingga dapat meningkatkan efisiensi dan produktivitas.
    b. Hasil penelitian ini dapat digunakan sebagai referensi dalam pengambilan keputusan terkait penjadwalan produksi dan pengendalian biaya.
    c. Penelitian ini dapat memberikan solusi praktis untuk tantangan yang dihadapi oleh PT. INKA dalam penjadwalan produksi dan pengendalian biaya.
2. Manfaat Bagi Dunia Akademik
    a. Penelitian ini dapat memberikan kontribusi pada literatur terkait penjadwalan produksi menggunakan algoritma genetika dan simulasi komputer.
    b. Hasil penelitian ini dapat digunakan sebagai bahan ajar atau referensi untuk mahasiswa atau peneliti lain yang tertarik pada topik serupa.
    c. Penelitian ini dapat mendorong penelitian lebih lanjut terkait optimasi penjadwalan produksi dan pengendalian biaya di industri manufaktur lainnya.

1.5 Batasan Penelitian

Dalam melakukan penelitian, ada beberapa batasan yang perlu diperhatikan untuk memastikan fokus dan kualitas penelitian. Berikut adalah batasan-batasan dalam penelitian ini:

1. Penelitian ini akan berfokus pada penjadwalan produksi di PT. INKA dan tidak akan membahas aspek lain dari operasi perusahaan.
2. Penelitian ini akan berfokus pada optimasi penjadwalan produksi pada proses *finishing* untuk pekerjaan gerbong penumpang untuk sebuah produk kereta eksekutif
3. Penelitian ini akan berfokus pada pengurangan biaya keterlambatan dan *makespan*, dan tidak akan membahas optimasi faktor lain seperti kualitas produk atau kepuasan pelanggan.
4. Penelitian ini akan berfokus pada aspek biaya tenaga kerja harian sebagai representasi biaya keterlambatan dan tidak mempertimbangkan dampak finansial lain yang diakibatkan keterlambatan penyelesaian pesanan.
5. Penelitian ini akan menggunakan data historis dari PT. INKA dan tidak akan mencoba untuk memprediksi kondisi masa depan perusahaan.
6. Penelitian ini akan dilakukan dalam konteks industri manufaktur kereta api dan hasilnya mungkin tidak berlaku untuk industri lain.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Literatur

Penelitian ini berlandaskan pada kajian literatur yang mendalam, merangkum berbagai sumber terpercaya yang relevan dengan metode dan permasalahan yang dikaji. Kajian literatur ini berperan penting dalam memetakan lanskap penelitian terdahulu, membantu menentukan arah penelitian yang akan dilakukan, dan mengidentifikasi celah pengetahuan yang perlu diisi. Berikut merupakan beberapa rangkuman dari penelitian terdahulu yang relevan dengan topik dan metode penelitian:

Pada penelitian oleh Mayer *et al* (2020) dengan judul "*Environmental and Economic Multi-Objective Optimization Of A Household Level Hybrid Renewable Energy System by Genetic Algorithm"* mengangkat kasus berkaitan penggunaan energi terbarukan pada skala rumahan yang semakin marak dan menilai perlu diidentifikasi sistemnya agar mencapai titik ekonomi dan lingkungan yang optimal. Penelitian menggunakan metode NSGA-II yang menghasilkan penggunaan NSGA-II untuk setiap tujuan yakni minimalisasi biaya dan dampak lingkungan dan berhasil menemukan titik *trade-off* teroptimal untuk kedua obyektif tersebut. Penelitian tersebut berhasil menunjukkan keandalan NSGA-II dalam menyelesaikan kasus berkaitan 2 (dua) obyektif biaya dan dampak lingkungan.

Pada penelitian oleh Abdi & Zarandi (2019) dengan judul "*A Meta Heuristic-Based Task Scheduling and Mapping Method to Optimize Main Design Challenges Of Heterogeneous Multiprocessor Embedded Systems"* mengangkat kasus berkaitan cara menjadwalkan tugas pada MPSoC heterogen untuk mengoptimalkan ketahanan masa pakai (*lifetime reliability*), kinerja (*performance*), dan konsumsi daya (*power consumption*) secara bersamaan. Penelitian menggunakan metode NSGA-II yang menujukkan metode penjadwalan yang diusulkan mampu meningkatkan kinerja sebesar 24%, ketahanan masa pakai sebesar 30%, dan konsumsi daya sebesar 3.6% (rata-rata)

dibandingkan dengan dua penelitian terkait lainnya. Penelitian tersebut berhasil menunjukkan keandalan NSGA-II terbukti efektif dalam mengoptimalkan ketiga parameter penting pada MPSoC heterogen: ketahanan masa pakai, kinerja, dan konsumsi daya.

Pada penelitian oleh Pirozmand *et al* (2021) dengan judul "*Multi-Objective Hybrid Genetic Algorithm for Task Scheduling Problem in Cloud Computing"* mengangkat kasus berkaitan tantangan dalam sistem *cloud computing* akibat berbagai macam klien dan berbagai layanan yang tersedia dalam sistem ini, dapat dikatakan bahwa masalah penjadwalan untuk minimalisasi biaya penyedia dan, tentu saja, konsumsi energi adalah tantangan penting dari sistem ini.. Penelitian menggunakan metode algoritma genetika dan heuristik penjadwalan hemat energi yang menghasilkan penggunaan algoritma GAECS mencapai kinerja superior dalam hal *makespan* dan konsumsi energi. Penelitian tersebut berhasil menunjukkan bahwa algoritma GAECS menawarkan pendekatan yang menjanjikan untuk penjadwalan dan optimasi energi dalam sistem *cloud computing*.

Pada penelitian oleh Chen *et al* (2020) dengan judul "*Multi-Objective Genetic Algorithm for Energy-Efficient Hybrid Flow Shop Scheduling with Lot Streaming"* mengangkat kasus berkaitan perencanaan produksi di lingkungan pabrikasi hibrida (urutan tetap dan fleksibel) dengan konsumsi energi yang efisien. Tujuannya adalah untuk meminimalkan waktu penyelesaian produksi (*makespan*) dan konsumsi listrik secara bersamaan. Ada pertentangan antara kedua tujuan ini, dan semakin optimal satu tujuan, semakin buruk tujuan lainnya. Ditambah lagi, model matematika yang digunakan untuk masalah ini kompleks secara komputasi. Penelitian menggunakan metode NSGA-II dan MOEES yang menghasilkan pembagian pekerjaan menjadi sub-pekerjaan dengan batas atas yang berbeda berdampak pada efisiensi produksi. Hasil penelitian menunjukkan bahwa penggunaan teknik "*lot streaming*" (pembagian pekerjaan) dapat meningkatkan efisiensi produksi. Penelitian tersebut berhasil menunjukkan NSGA-II yang dikembangkan peneliti lebih unggul dari dua algoritma evolusi lainnya dalam hal kualitas solusi. Preferensi pengguna dapat sepenuhnya mengontrol area pencarian dalam ruang

solusi melalui vektor preferensi yang ditetapkan. Pengguna dapat memilih solusi perencanaan produksi yang sesuai dengan preferensi mereka.

Pada penelitian oleh Zan *et al* (2020) dengan judul "*A Pareto-Based Genetic Algorithm for Multi-Objective Scheduling Of Automated Manufacturing Systems"* mengangkat kasus berkaitan penjadwalan pada Sistem Manufaktur Otomatis (AMS) yang rentan *deadlock* (macet) menjadi rumit karena dua tujuan yang perlu dicapai yaitu urutan kerja yang optimal dan pencegahan *deadlock*. Selain itu, kapasitas sumber daya di AMS terbatas. Penelitian menggunakan metode NSGA-II yang menghasilkan perbandingan metode diusulkan terhadap MNSGAII (algoritma lain) menunjukkan bahwa PGA yang diusulkan efektif dan efisien. Penelitian tersebut berhasil menciptakan PGA yang efektif untuk penjadwalan produksi pada AMS yang rentan *deadlock* dengan kapasitas sumber daya terbatas.

Pada penelitian oleh Alonso Campos *et al* (2020) dengan judul "*Real-Time Energy Optimization Of Irrigation Scheduling by Parallel Multi-Objective Genetic Algorithms"* mengangkat kasus berkaitan biaya energi untuk irigasi tetes dan *sprinkler* meningkat karena tingginya harga energi dan kebutuhan tekanan air yang besar. Para peneliti telah menunjukkan bahwa pengelolaan jaringan irigasi yang tepat dapat mengurangi biaya pengoperasian pompa. Tantangan utamanya adalah mendapatkan parameter operasi yang optimal secara mendekati waktu nyata karena rumitnya masalah optimasi yang membutuhkan banyak perhitungan.. Penelitian menggunakan metode NSGA-II yang menghasilkan pendekatan baru untuk optimasi energi jaringan irigasi dengan mengatur ulang jadwal irigasi berdasarkan volume air yang dibutuhkan setiap hidran. Pendekatan ini dapat mengurangi biaya energi 6-7% tanpa mengurangi tekanan air. Pendekatan multi-objgektif dan paralelisasi algoritma dapat mengurangi jumlah perhitungan dan waktu perhitungan secara signifikan. Penelitian tersebut berhasil mengembangkan algoritma yang dapat memberikan solusi optimal dalam beberapa menit. Algoritma ini dapat diterapkan untuk mengoptimalkan kebutuhan air harian. Dengan menggunakan teknologi *web service*, algoritma ini dapat dengan mudah diintegrasikan ke dalam platform manajemen jaringan irigasi.

Pada penelitian oleh Xu *et al* (2021) dengan judul "*A Multi-Objective Scheduling Method for Distributed And Flexible Job Shop Based on Hybrid Genetic Algorithm and Tabu Search Considering Operation Outsourcing and Carbon Emission"* mengangkat kasus berkaitan banyaknya jenis operasi yang dibutuhkan untuk mengerjakan produk besar dan rumit sering kali membuat perusahaan manufaktur kesulitan menyelesaikan semua operasi tersebut sendiri. Oleh karena itu, selain ketiga sub-masalah tersebut, penelitian juga mempertimbangkan karakteristik dimana beberapa pekerjaan membutuhkan *outsourcing*, sehingga model yang dibuat menjadi lebih sesuai dengan keadaan nyata perusahaan manufaktur dibandingkan dengan penelitian sebelumnya. Selain itu, mengingat masalah lingkungan yang semakin serius dan meningkatnya biaya pengelolaan lingkungan, model matematika yang dibuat dalam penelitian ini mencakup empat tujuan optimasi: *makespan* (waktu penyelesaian keseluruhan), biaya, kualitas, dan emisi karbon.. Penelitian menggunakan metode GA Hybrid dan *Tabu Search* yang menunjukkan bahwa algoritma ini memiliki kinerja keseluruhan yang lebih baik daripada GA dan TS sederhana dalam memecahkan masalah yang diteliti dalam penelitian ini. Penelitian tersebut berhasil menunjukkan secara keseluruhan H-GA-TS merupakan algoritma yang efektif untuk mengatasi DFJSP untuk produk besar dan rumit. Dengan perkembangan teknologi dan manufaktur, algoritma ini dapat diterapkan pada situasi lain.

Pada penelitian oleh Yu *et al* (2020) dengan judul "*Multi-Objective Scheduling in Hybrid Flow Shop: Evolutionary Algorithms Using Multi-Decoding Framework"* mengangkat kasus berkaitan penjadwalan di lingkungan produksi *hybrid flow shop* (HFS) dengan mesin yang tidak terkait, kelayakan mesin, dan waktu setup tergantung urutan (SDST) menjadi rumit karena melibatkan dua tujuan optimasi: (1) meminimalkan total keterlambatan dan (2) meminimalkan total waktu setup. Penelitian menggunakan metode NSGA-II yang berhasil mengembangkan beberapa algoritma *decoding* yang efektif untuk memetakan urutan pekerjaan ke jadwal produksi. Selain itu, kerangka kerja *multi-decoding* (MDF) yang diusulkan bermanfaat untuk membantu EA menyesuaikan skema *decoding* yang digunakan dan menghasilkan solusi yang sesuai dengan preferensi pengguna (pendekatan "apriori" dan memperluas ruang pencarian dan meningkatkan kualitas solusi (pendekatan "aposteriori"). Penelitian tersebut berhasil menunjukkan

pendekatan *hyper-heuristic* yang efektif untuk menyelesaikan masalah penjadwalan di lingkungan produksi HFS dengan SDST. Pendekatan ini menawarkan dua keuntungan: (1) solusi yang optimal berdasarkan preferensi pengguna dan (2) kumpulan solusi yang beragam untuk pengguna pilih nanti.

Pada penelitian oleh Anjana *et al* (2020) dengan judul "*Metaheuristics for Solving A Multi-Objective Flow Shop Scheduling Problem with Sequence-Dependent Setup Times"* mengangkat kasus berkaitan penjadwalan produksi di lini produksi urut (*flow shop*) dengan waktu setup tergantung urutan (SDST) menjadi rumit karena melibatkan dua tujuan: (1) meminimalkan waktu penyelesaian keseluruhan (*makespan*) dan (2) meminimalkan keterlambatan rata-rata. Umumnya, masalah di dunia nyata melibatkan optimasi lebih dari satu aspek kinerja. Penelitian menggunakan metode NSGA-II, NSGA-II *Hybrid*, DPSO, dan DPSO *Hybrid* yang menghasilkan bahwa gabungan dari NSGA II dengan teknik pencarian lingkungan variabel (VNS) menghasilkan kinerja terbaik dibandingkan dengan tiga algoritma lainnya untuk semua ukuran masalah. Secara umum, penggabungan metaheuristik dengan VNS meningkatkan kinerja. Penelitian tersebut berhasil menunjukkan keunggulan penggabungan NSGA II dengan VNS.

Pada penelitian oleh Kusuma (2021) dengan judul "*Multi-objective Batch Scheduling in Collaborative Multi-product Flow Shop System by using Non-dominated Sorting Genetic Algorithm"* mengangkat kasus untuk mengembangkan model penjadwalan *batch* yang baru untuk sistem *collaborative multi-product flow shop*. Tujuannya adalah untuk meminimalkan *makespan* (waktu penyelesaian keseluruhan) dan total biaya produksi. Penelitian menggunakan metode NSGA-II yang menghasilkan *makespan* dan total biaya produksi yang lebih rendah dibandingkan dengan model non-kolaboratif yang ada, dimana *makespan* model yang diusulkan 10% sampai 17% lebih rendah daripada model non-kolaboratif dan total biaya produksi model yang diusulkan 0.3% sampai 3.5% lebih rendah daripada model non-kolaboratif. Penelitian tersebut berhasil menunjukkan bahwa model penjadwalan *batch* kolaboratif yang diusulkan efektif dalam mencapai tujuan penelitian untuk meminimalkan *makespan* dan total biaya produksi. Model NSGA II kolaboratif lebih unggul daripada model non-kolaboratif (NSGA II dan *adjacent pairwise interchange).*

Pada penelitian oleh Grosch *et al* (2021) dengan judul "*Multi-Objective Hybrid Genetic Algorithm for Energy Adaptive Production Scheduling in Job Shops"* mengangkat kasus berkaitan kebutuhan penjadwalan produksi yang optimal untuk memanfaatkan energi terbarukan secara efisien dan mengurangi biaya. Kenaikan penggunaan energi terbarukan membuat pasokan energi menjadi tidak stabil, sehingga program manajemen permintaan energi (*demand response*) menjadi penting. Penjadwalan produksi yang optimal harus mempertimbangkan tujuan operasional dan harga energi. Penelitian menggunakan metode NSGA-II yang menyeimbangkan biaya energi dan *makespan* sesuai kebutuhan mereka. Hasil penelitian bergantung pada model sistem produksi dan harga energi. Namun, pada model pabrik percontohan ETA-Fabrik, penghematan energi sekitar 6% dan pengurangan emisi gas rumah kaca dapat dicapai dengan peningkatan *makespan* yang masih dapat diterima.. Penelitian tersebut berhasil mengembangkan algoritma penjadwalan produksi adaptif energi yang dapat menghemat energi terbarukan dan mengurangi emisi gas rumah kaca di industri logam.

Pada penelitian oleh Valledor *et al* (2022) dengan judul "*Solving Rescheduling Problems in Dynamic Permutation Flow Shop Environments with Multiple Objectives Using the Hybrid Dynamic Non-Dominated Sorting Genetic II Algorithm"* mengangkat kasus berkaitan penjadwalan ulang produksi di lini produksi urut (*flow shop*) dinamis dengan permutasi menjadi rumit karena tujuan produksi bisa berbeda dan diukur dengan satuan yang berbeda (misalnya waktu penyelesaian, keterlambatan tertimbang, dan stabilitas) dan gangguan yang tidak terduga seperti kerusakan mesin atau kedatangan pekerjaan prioritas baru bisa terjadi dan membutuhkan penjadwalan ulang yang cepat. Penelitian menggunakan metode NSGA-II *Hybrid* dan *Dynamic* (HDNSGA-II) yang menghasilkan kinerja baik dari algoritma tersebut terutama untuk masalah yang lebih besar, dibandingkan dengan algoritma lain yang pernah ada (RIPG) dan berkontribusi untuk membuat *benchmark* untuk memodelkan masalah penjadwalan ulang dinamis dengan gangguan, membuat kerangka kerja penjadwalan ulang periodik berdasarkan strategi prediksi-reaktif, mengembangkan dan membandingkan dua algoritma (HDNSGA-II dan RIPG) untuk masalah penjadwalan ulang dinamis, serta algoritma HDNSGA-II menggunakan pengetahuan dari titik penjadwalan ulang sebelumnya untuk

inisialisasi populasi. Penelitian tersebut berhasil menunjukkan algoritma HDNSGA-II lebih baik dari RIPG dalam hal performa, terutama untuk masalah yang lebih besar. RIPG lebih mudah diatur karena parameter apa pun menghasilkan hasil yang mirip. Evaluasi fungsi tujuan pada RIPG lebih banyak daripada HDNSGA-II.

Pada penelitian oleh Afira & Wijayanto (2022) dengan judul "*Optimization of Waste Transportation Routes using Multi-objective Non-dominated Sorting Genetic Algorithm II (MNSGA-II) in the Eastern and Southern Regions of Bandung City, Indonesia"* mengangkat kasus berkaitan pengelolaan sampah perkotaan yang berkualitas dan efektif menjadi prioritas dengan rute pengangkutan sampah merupakan salah satu faktor penentu utama biaya pengelolaan sampah. Penelitian menggunakan metode NSGA-II yang menghasilkan Percobaan yang dilakukan menghasilkan solusi terbaik berupa 14 rute dengan total jarak 152,63 km. Optimasi rute yang diusulkan berpotensi bermanfaat untuk mendukung peningkatan sistem layanan pengelolaan sampah berkelanjutan di Kota Bandung. Penelitian tersebut berhasil menunjukkan bahwa penggunaan MNSGA-II yang diusulkan mampu menemukan solusi rute optimal untuk memodelkan kegiatan pengangkutan sampah yang dikelola oleh lembaga pemerintah daerah, yaitu PD Kebersihan. Solusi terbaik yang dicapai menghasilkan 14 rute dengan total jarak 152,63 km. Hasil penelitian ini berpotensi bermanfaat dalam memberikan informasi pendukung untuk meningkatkan kualitas pelayanan sistem pengumpulan sampah di wilayah Timur dan Selatan Kota Bandung serta untuk diterapkan di wilayah metropolitan lain di Indonesia.

Pada penelitian oleh Tampubolon (2021) dengan judul "Implementasi *Hybrid Algorithm* Untuk Optimalisasi Konsumsi Energi Pada *Job Shop Scheduling"* mengangkat kasus berkaitan isu pencemaran udara menjadi salah satu hal yang sedang banyak dibicarakan akhir-akhir ini. Salah satu cara untuk mengurangi pencemaran udara adalah dengan melakukan efisiensi terhadap konsumsi energi. Penelitian ini akan berfokus pada penyelesaian *job shop* yang tidak hanya meminimalkan waktu penyelesaian pekerjaan akan tetapi juga konsumsi energi.. Penelitian menggunakan metode *Hybrid* Algoritma Genetika bersama *Simulated Annealing* yang menghasilkan nilai minimal *makespan* maupun konsumsi energi yang dihasilkan, untuk bobot yang sama untuk *makespan* dan

konsumsi energi $w1$=0,5 dan $w2$=0,5. Untuk *makespan* hasil yang diperoleh cukup mendekati dengan *best known solution.* Penelitian tersebut berhasil menunjukkan bahwa algoritma Genetika dan *Simulated Annealing* mempunyai performa yang cukup baik dalam menyelesaikan penjadwalan *job shop,* hal ini dikarenakan adanya penambahan pencarian lokal pada solusi yang sudah ditemukan oleh Algoritma Genetika. Meski demikian Algoritma ini sedikit mempunyai sedikit penurunan performa pada kasus yang memiliki jumlah pekerjaan dan mesin yang besar..

Pada penelitian oleh Bahy & Musdholifah (2022) dengan judul "*Fast Non-dominated Sorting in Multi Objective Genetic Algorithm for Bin Packing Problem"* mengangkat kasus berkaitan permasalahan *bin packing* klasik adalah masalah memasukkan barang dengan volume dan dimensi berbeda ke dalam wadah sehingga volume barang yang dimasukkan maksimal. Namun, dalam praktiknya, sering kali terdapat beberapa tujuan lain yang perlu dipertimbangkan selain volume, seperti pusat massa barang di dalam wadah dan berat barang di dalam wadah. Permasalahan *bin packing* multi-objektif bertujuan untuk mencari solusi yang optimal untuk semua tujuan tersebut secara bersamaan. Penelitian menggunakan metode NSGA-II yang menghasilkan algoritma yang diusulkan berhasil menemukan beberapa solusi optimal dengan mempertimbangkan tiga objektif yang berbeda: volume, pusat massa, dan berat dengan parameter ukuran populasi 100, jumlah populasi 500, dan probabilitas mutasi 10%. Penelitian tersebut berhasil menunjukkan bahwa algoritma genetika multi-objektif yang diusulkan dapat menyelesaikan permasalahan *bin packing* multi-objektif dengan efektif.

Setelah mengidentifikasi sejumlah 15 (lima belas) jurnal berkaitan optimasi multi-obyektif menggunakan algoritma evolusi pada berbagai studi kasus dengan, maka dapat disusun *research* gap dari penelitian terdahulu terhadap penelitian yang akan dibangun. Berikut ini merupakan tabel yang menunjukkan spesifikasi penelitian terdahulu berdasarkan parameter yang akan digunakan dalam penelitian saat ini sebagai berikut:

1. Penjadwalan
2. Multi-Obyektif
3. Algoritma Genetika dan NSGA-II

4. *Makespan*
5. Biaya Keterlambatan

Tabel 2.1 Kajian Literatur

No	Penulis	Tahun	Objek	1	2	3	4	5
1.	Martin János Mayer dkk	2020	Sistem energi terbarukan		√	√		
2.	Abdi, Zarandi	2019	Sistem *chip multiprocessor*	√	√	√		
3.	Poria Pirozmand	2020	*Cloud computing*	√	√	√		√
4.	Tzu-Li Chen dkk	2020	Mesin produksi	√	√	√	√	
5.	Xin Zan dkk	2019	Mesin produksi	√	√	√	√	
6.	J.C. Alonso Campos dkk	2020	Sistem irigasi		√	√		√
7.	Wenxiang Xu dkk	2021	Mesin produksi	√	√	√	√	
8.	Chunlong Yu dkk	2020	Mesin produksi	√	√	√		√
9.	V. Anjana, R dkk	2020	Mesin produksi	√	√	√	√	
10.	Purba Daru Kusuma	2021	Mesin produksi	√	√	√	√	
11.	Benedikt Grosch dkk	2021	Mesin produksi	√	√	√		
12.	Pablo Valledor Alberto dkk	2022	Mesin produksi	√	√	√	√	
13.	Natasya Afira dkk	2022	Pengelolaan sampah		√	√		√
14.	Ferdinan Rinaldo Tampubolon	2021	Mesin produksi	√	√	√	√	
15.	Muhammad Bintang Bahy dkk	2022	Sistem *bin-packing*		√	√		
16.	Penelitian yang diusulkan	2024	Mesin produksi	√	√	√	√	√

Berdasarkan rangkuman kajian literatur dengan parameter yang akan digunakan dalam penelitian ini, penelitian terdahulu belum ada yang secara spesifik melakukan optimasi pada lantai produksi dengan objek mesin produksi untuk mengatasi *makespan* dan biaya keterlambatan melalui penjadwalan dengan optimasi multi-obyektif NSGA-II. Oleh karena itu, penelitian ini akan mengangkat objek penelitian mesin produksi untuk dijadwalkan sehingga mencapai angka *makespan* dan biaya keterlambatan optimal.

2.2 Landasan Teori

Landasan teori berisikan tentang istilah, teori atau formula yang terkait dengan topik penelitian. Landasan teori disusun dengan bersumber pada jurnal bereputasi dan/atau buku.

2.2.1 Penjadwalan Produksi

Penjadwalan produksi merupakan proses mengalokasikan sumber daya dan mesin untuk pekerjaan dalam sebuah sistem dengan mempertimbangkan batas dan waktu yang tersedia (Baker & Trietsch, 1974). Penjadwalan memiliki peran yang strategis dalam proses produksi, karena melibatkan pembagian sumber daya untuk menyelesaikan sejumlah tugas dengan efisien dan sesuai spesifikasi, sehingga perlu untuk menemukan alur teroptimalnya melalui penjadwalan. Penjadwalan sebagai salah satu masalah optimasi memiliki beberapa cara untuk menemukan solusinya, yaitu meminimalkan waktu menganggur mesin, mengurangi ketersediaan barang setengah jadi dengan mengurangi *work-in-progress*, dan mengurangi jumlah keterlambatan rata-rata. Berikut ini merupakan klasifikasi model penjadwalan berdasarkan 4 (empat) jenis keadaannya, yaitu:

1. Mesin yang digunakan, dalam hal ini adalah jumlah mesin yang digunakan untuk serangkaian proses produksi.
   a. Penjadwalan pada Mesin Tunggal (*Single Machine)*

      Produksi dilakukan hanya menggunakan 1 (satu) mesin sehingga perlu dijadwalkan *job* yang perlu dikerjakan terlebih dahulu.
   b. Penjadwalan pada Mesin Ganda

      Produksi dilakukan menggunakan 2 (dua) mesin untuk setiap pekerjaannya,
   c. Penjadwalan pada Banyak Mesin (*Multi Machine*)

      Produksi dilakukan menggunakan 2 (dua) atau lebih mesin paralel dan bisa dijadwalkan secara *single stage* maupun *multi stage*.
2. Pola aliran proses, dalam hal ini perlu diperhatikan keseragaman aliran proses setiap jenis produk yang dibuat.
   a. *Flow Shop*

      Pola aliran produksi seragam / mirip untuk seluruh pekerjaan dan konstan dimana stasiun kerja disusun sesuai urutan operasi produksi produk tersebut. Terdapat 2 (dua) jenis *flow shop* berdasarkan ada tidaknya variasi dalam produk, dimana apabila tidak ada variasi dalam aliran produksi maka disebut *pure flow shop*, dan sebaliknya maka disebut *general flow shop.*
   b. *Job Shop*

Pola aliran produksi berbeda untuk setiap produk sehingga mesin yang sama dapat berada di kondisi mana pun, baik mengerjakan operasi awal untuk pekerjaan maupun operasi terakhir.

3. Sistem informasi, dalam hal ini informasi yang dimaksudkan merupakan informasi krusial baik berkaitan permesinan maupun material.
   a. Informasi Deterministik

      Kepastian informasi krusial dalam produksi dijamin, dengan contoh informasinya adalah waktu proses operasi dan waktu tenggat (*due date)*.
   b. Informasi Stokastik

      Kepastian informasi krusial dalam produksi belum tentu ada, dengan contoh informasinya adalah waktu kedatangan bahan baku dan waktu antar kedatangan pekerjaan.
4. Pola kedatangan proses, dalam hal ini perlu diperhatikan kepastian datangnya *job* pada lantai produksi.
   a. Pola Kedatangan Statis

      Pekerjaan sampai pada lantai produksi dengan rentang waktu yang konstan terhadap kesiapan fasilitas produksi.
   b. Pola Kedatangan Dinamis

      Pekerjaan sampai pada lantai produksi dengan rentang waktu yang acak sehingga sulit diprediksi dan mengakibatkan belum siapnya fasilitas produksi.

Berdasarkan klasifikasi untuk model penjadwalannya, studi kasus pada penelitian ini dapat dirumuskan sebagai model penjadwalan untuk *multi-machine* pada aliran produksi *flowshop* dengan kehadiran informasi deterministik dan pola kedatangan proses statis.

2.2.2 Definisi dalam Penjadwalan Produksi

Terdapat beberapa istilah yang digunakan dalam penjadwalan produksi, diantaranya ada yang merupakan elemen penjadwalan yaitu, pekerjaan (*job),* operasi, dan mesin. Selain itu terdapat juga beberapa istilah umum dalam penjadwalan produksi. Berikut ini merupakan definisi yang digunakan dalam penjadwalan produksi.

1. Pekerjaan (*Job*), merupakan tugas yang harus diselesaikan untuk menghasilkan produk dan terdiri dari beberapa operasi. Karakteristik dari pekerjaan adalah:
   a. Terdiri dari operasi yang ditetapkan
   b. Setiap mesin hanya dapat melakukan operasi tertentu
   c. Waktu proses dan *due date* diketahui
   d. Penentuan waktu *setup* independen dan waktu transportasi antar mesin dapat diabaikan
   e. Setiap mesin hanya melakukan 1 (satu) operasi pada 1 (satu) waktu
2. Operasi, merupakan proses yang diurutkan untuk menyelesaikan pekerjaan dan bersifat dependen satu sama lain. Karakteristik dari operasi adalah:
   a. Operasi dikerjakan hingga selesai pada suatu mesin
   b. 1 (satu) operasi dikerjakan 1 (satu) mesin
   c. Alur operasi harus urut
   d. Operasi dipetakan urutannya dalam matriks *routing*
3. Mesin, merupakan salah satu fasilitas produksi yang bertugas menyelesaikan operasi. Karakteristik dari mesin adalah:
   a. 1 (satu) mesin mengerjakan 1 (satu) operasi pada waktu tertentu
   b. Mesin dapat secara *berkelanjutan stand by* pada setiap tugas yang dibebankan.
   c. Tidak masuk pada penjadwalan apabila rusak atau *maintenance.*
   d. Data waktu dan distribusi pada operasinya diketahui
4. *Processing Time* (pj), merupakan jumlah waktu yang dibutuhkan sebuah operasi dari sebuah pekerjaan.
5. *Release Date* (rj), merupakan waktu dimana sebuah pekerjaan siap untuk masuk ke operasi.
6. *Due Date* (dj), merupakan waktu dimana sebuah operasi dari sebuah pekerjaan dijadwalkan selesai.
7. *Completion Time* (Cj), merupakan waktu dimana sebuah operasi dari sebuah pekerjaan selesai.
8. *Waiting Time,* merupakan jumlah waktu menunggu sebelum pekerjaan benar-benar dimulai.

9. *Setup time*,
10. *Flowtime* (Fj), merupakan total waktu yang dihabiskan oleh suatu pekerjaan dalam sistem termasuk waktu tunggu dan waktu pelayanan.
11. *Lateness* (Lj), merupakan jumlah kelebihan *completion time* (Cj) dibandingkan *due date* (dj) sehingga, Lj = Cj - dj
12. *Tardiness* (Tj), merupakan nilai yang mewakili besar *lateness* (Lj), apabila Lj > 0, maka Tj = Lj dan sebaliknya maka Tj = 0
13. *Earliness* (Ej), merupakan nilai yang mewakili besar *lateness* (Lj), apabila Lj > 0, maka Ej = 0 dan sebaliknya maka Ej = Lj
14. *Makespan*, merupakan total waktu penyelesaian yang diperlukan dari penjadwalan guna menyelesaikan keseluruhan pada sebuah pekerjaan.

Definisi - definisi non elemen tersebut merupakan variabel ukur yang umumnya digunakan untuk menentukan kinerja suatu penjadwalan untuk mencapai tujuan penjadwalan tersebut. Baker & Trietsch (1974) menuliskan beberapa tujuan umum dari sebuah penjadwalan, diantaranya:

1. Minimalisasi *makespan*
2. Minimalisasi total waktu tertimbang
3. Minimalisasi keterlambatan tertimbang
4. Minimalisasi keterlambatan maksimum
5. Minimalisasi jumlah *job* yang terlambat
6. Minimalisasi total *flow time*

Penjadwalan akan dilakukan berdasarkan informasi yang diketahui berkaitan elemen produksi, dimana terdapat n pekerjaan pada 6 mesin dan operasi. Pada penelitian ini proses penjadwalan yang dilakukan akan terfokus pada bagaimana cara meminimalkan *makespan* untuk mengendalikan keterlambatan yang ada dan minimalisasi keterlambatan tertimbang yang diukur menggunakan biaya sehingga mampu mengatasi dampak finansial yang terjadi saat ini menggunakan variabel ukur keterlambatan .

2.2.3 Penjadwalan *Flowshop*

Flowshop sebagai salah satu aliran produksi dengan ciri khas kemiripan operasi antar pekerjaan yang menjadikan fasilitas produksi disusun sesuai urutan operasi. Aliran tersebut menjadikan pekerjaan diproses dengan mengalirkan produk secara sekuensial sehingga *flowshop* dapat disebut sebagai *product flow*. Kondisi tersebut menjadikan model *flow shop* memiliki karakteristik sebagai berikut:

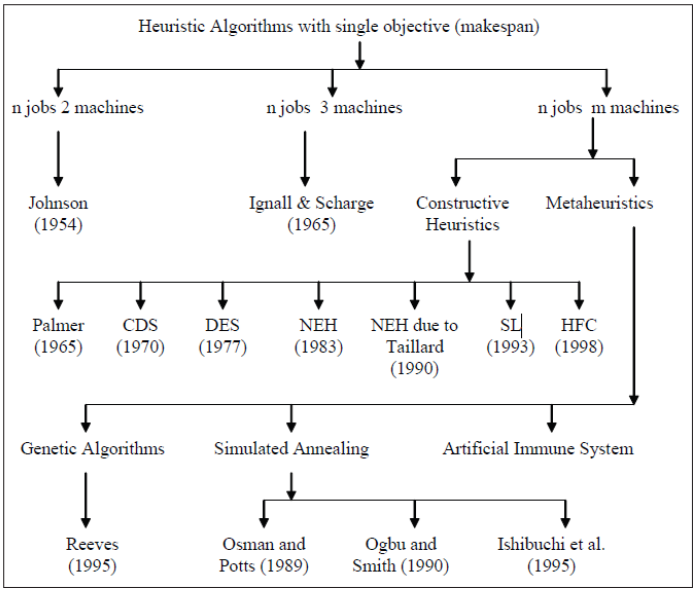
1. Setiap mesin (1, 2, 3,...., m) disusun untuk mengerjakan setiap operasi berbeda ((1, j), (2, j), ... (m, j)) dalam menyelesaikan sebuah pekerjaan (j)
2. *Setup time* dimiliki masing-masing mesin operasi secara independen satu sama lain dan ditambahkan dalam *processing time*.
3. Deskripsi pekerjaan jelas diawal
4. Seluruh mesin selalu siap digunakan
5. Tidak ada interupsi setelah pekerjaan dimulai

Berdasarkan karakteristik tersebut, metode dan pendekatan untuk optimasi penjadwalannya juga beragam utamanya untuk tujuan minimalisasi *makespan*, seperti "*Branch-and-Bound", integer programming*, dan metode heuristik. Masing-masing metode tersebut memiliki kelebihan dan kekurangannya, berikut ini merupakan rangkuman ketiga metode tersebut pada Tabel 2.2.

Tabel 2.2 Metode *Flowshop Scheduling*

Metode	***Branch-and-Bound Solutions***	***Integer Programming Solutions***	***Heuristics Solutions***
Cara Kerja	1. Membagi masalah menjadi sub-masalah yang lebih kecil 2. Menentukan batas atas dan bawah untuk solusi optimal 3. Melakukan pencarian cabang dan batas untuk menemukan solusi optimal	1. Formulasi masalah sebagai program integer 2. Memecahkan program integer menggunakan *solver*	1. Mengikuti aturan untuk menghasilkan urutan pekerjaan 2. Melakukan evaluasi solusi dan iterasi hingga solusi yang memuaskan ditemukan
Kelebihan	1. Menemukan solusi optimal 2. Dapat menangani masalah kompleks	1. Menemukan solusi optimal 2. Dapat menangani masalah dengan banyak variabel dan batasan	1. Cepat dan mudah diterapkan 2. Dapat menangani masalah besar
Kekurangan	1. Komputasi yang lama 2. Sulit untuk masalah besar	1. Komputasi yang lama 2. Sulit untuk masalah besar	1. Tidak selalu menemukan solusi optimal 2. Kualitas solusi tergantung pada heuristik yang digunakan

Studi kasus yang dirumuskan mengharapkan penyelesaian secara cepat dan optimal, oleh karena itu penggunaan metode *heuristics* dinilai sesuai dengan tujuan. Solusi *heuristics* yang secara spesifik ditujukan untuk menangani penjadwalan *flowshop* untuk minimalisasi *makespan* dapat diklasifikasikan seperti tampak pada Gambar 2.1 berikut.



Gambar 2.1 Klasifikasi Metode Heuristik

Sumber : (Laha, 2007)

Gambar 2.1 tersebut menampilkan klasifikasi awal berdasarkan kriteria jumlah pekerjaan dan mesin, dimana untuk studi kasus yang diteliti maka metode yang sesuai adalah *Constructive Heuristics* dan *Metaheuristics*. *Constructive Heuristics* akan menghasilkan penjadwalan yang langsung optimal dan tidak dapat diubah untuk peningkatan, sedangkan *Metaheuristics* memulai penemuan solusi dengan menyusun urutan penjadwalan untuk kemudian diubah-ubah urutannya menggunakan iterasi hingga memenuhi fungsi tujuan teroptimalnya (Laha, 2007). Tujuan penelitian untuk menemukan solusi teroptimal untuk 2 (dua) obyektif dinilai lebih sesuai jika menggunakan metode *Metaheuristics*.

2.2.4 Optimasi Multi-Obyektif

Optimasi multi-obyektif atau *Multi-objective Optimization* (MO) merupakan sebuah proses simultan guna mencapai titik optimal 2 (dua) atau lebih tujuan (Deb *et al*., 2002). Kompleksitas MO ditentukan salah satunya oleh jenis fungsi obyektif, apakah

bertentangan atau tidak. Permasalahan MO dapat ditulis secara matematis dengan persamaan sebagai berikut.

$$Min/\max y = f(x) = (f_1(x), f_2(x), \ldots\ldots, f_k(x), \qquad k \geq 2 \tag{1}$$

$$Subject\ to\ g_i(x) \leq 0, i = 1,2,3, \ldots\ldots, m \tag{2}$$

$$h_j(x) = 0, i = 1,2,3, \ldots\ldots, p \tag{3}$$

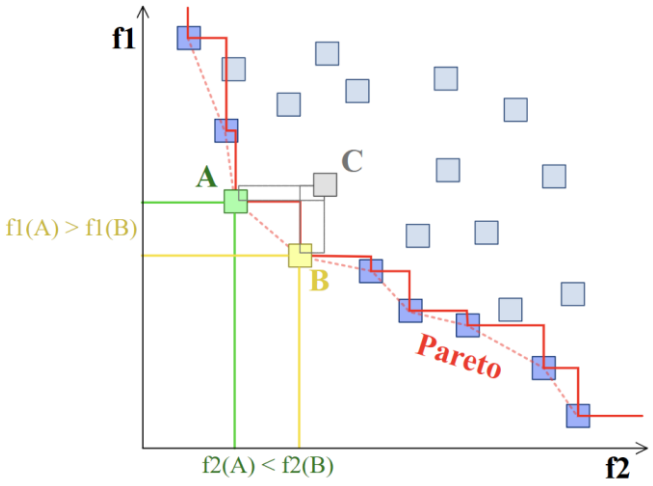
Keterangan,

- x (x_1, x_2, x_3,.....,x_n) = n-dimensi vektor keputusan
- y = k-dimensi vektor obyektif dalam R^k
- f = fungsi pemetaan
- g_i = batasan ketidaksamaan ke-i
- h_j = batasan kesamaan ke-j

Menurut Deb *et al* (2002), keluaran dari MO merupakan sekumpulan solusi optimal yang dikenal sebagai *pareto-optimal solution.* Solusi *Pareto*-optimal, juga dikenal sebagai solusi *non-dominated* atau solusi efisien, adalah solusi dalam MO yang tidak dapat ditingkatkan dalam satu tujuan tanpa memperburuk tujuan lainnya. Sebuah solusi x dianggap *Pareto*-optimal jika tidak ada solusi y lain yang memenuhi dua kondisi berikut:

- f1(x) ≤ f1(y) untuk semua tujuan i.
- f1(x) < f1(y) untuk setidaknya satu tujuan i.

f(x) merupakan vektor nilai objektif untuk solusi x, sehingga makna dari persamaan tersebut adalah kedua solusi x dan y, tidak mendominasi satu sama lain. Oleh karena itu, kedua solusi tersebut dapat dianggap *Pareto*-optimal. Set solusi *pareto* divisualisasikan dalam bentuk kurva guna memperlihatkan *trade-off* terbaik diantara berbagai tujuan disebut *Pareto Front* atau kurva *pareto* yang diperlihatkan pada Gambar 2.2.



Gambar 2.2 Kurva *Pareto*

Metode dalam MO untuk menemukan semua solusi *Pareto* optimal atau sub kumpulan representatif dari solusi *Pareto* optimal disebut metode posteriori (Hwang, 1979). Terdapat sejumlah 3 (tiga) pendekatan utama dari metode tersebut, yaitu pemrograman matematika, algoritma evolusi, dan *deep learning*. Berikut ini merupakan rangkuman dari ketiga pendekatan tersebut pada Tabel 2.3.

Tabel 2.3 Pendekatan pada Metode Posteriori

(Sumber : Hwang (1979))

Pendekatan	**Cara Kerja**	**Kelebihan**	**Kekurangan**	**Contoh Metode**
Pemrograman Matematika	Algoritma diulang, setiap iterasi menghasilkan satu solusi *Pareto* optimal.	- Solusi *Pareto* optimal terjamin. - Perkiraan akurat *Pareto front*.	- Kompleksitas komputasi tinggi. - Sulit dimodifikasi.	NBI, NBIm, NC, SPO, DSD
Algoritma Evolusi	Algoritma evolusi mencari solusi *Pareto* optimal secara iteratif.	- Mudah dimodifikasi - Kumpulan solusi yang beragam.	- Konvergensi lambat. - Jaminan *Pareto* optimal tidak mutlak.	NSGA-II, NSGA-III, SPEA-2, Diferensial Evolusi multi-objektif

Pendekatan	Cara Kerja	Kelebihan	Kekurangan	Contoh Metode
Deep Learning	Model *deep learning* dilatih untuk mempelajari *Pareto* front.	- Konvergensi cepat. - Solusi *Pareto* optimal yang beragam.	- Membutuhkan data pelatihan yang cukup. - Sulit diinterpretasikan.	*Pareto Front Learning*

Penelitian memilih menggunakan jenis metode *metaheurisctis* yang termasuk dalam pendekatan algoritma evolusi untuk dapat menghadapi tantangan manufaktur produk inovatif yang memiliki masalah yang dinamis sehingga solusi yang diharapkan adaptif terhadap berbagai obyektif.

2.2.5 *Non-Dominated Sorting Genetic Algorithm* II (NSGA-II)

Salah satu jenis metode dalam pendekatan algoritma evolusi yang cukup populer adalah NSGA-II atau *Non-dominated Sorting Genetic Algorithm* kedua. NSGA-II adalah peningkatan dari algoritma genetika non-dominasi (NSGA) yang sebelumnya banyak dikritik karena keterbatasannya. Beberapa keterbatasan NSGA termasuk tidak adanya *elitism*, kebutuhan untuk menentukan parameter *sharing* untuk menjaga keberagaman, dan kompleksitas komputasi yang tinggi. NSGA-II mengatasi kekurangan tersebut dengan beberapa keunggulan:

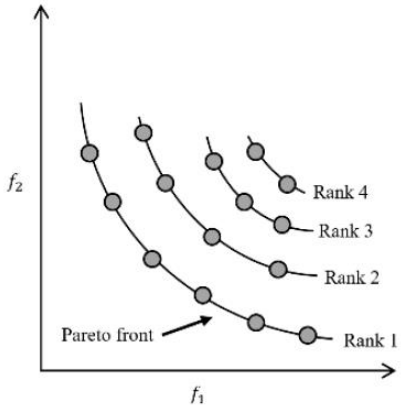
- *Elitism* : NSGA-II memiliki sifat *elitism*, artinya solusi terbaik dari generasi sebelumnya selalu diikutsertakan ke generasi berikutnya.
- Tidak perlu parameter *sharing* : NSGA-II tidak memerlukan parameter *sharing* untuk menjaga keberagaman populasi.
- *Crowding distance* : NSGA-II menggunakan operator *crowding distance* untuk menjaga keberagaman populasi. Operator ini mempertimbangkan kepadatan solusi di sekitar suatu solusi tertentu. Solusi dengan *crowding distance* yang lebih besar dianggap berada di area yang kurang padat, sehingga lebih mungkin untuk dipilih pada generasi berikutnya.

- Cepat dan efisien : NSGA-II memiliki kompleksitas komputasi maksimal $O(MN^2)$, dimana M adalah jumlah fungsi tujuan dan N adalah ukuran populasi. Ini membuatnya lebih cepat dibandingkan algoritma sebelumnya.

Keunggulan-keunggulan tersebut dapat diperoleh berkat 4 (empat) prinsip utama penerapan NSGA-II (Deb *et al*., 2002), yaitu :

1. *Fast Non-Dominated Sorting*

   Populasi diurutkan berdasarkan dominasi *Pareto* dimana solusi yang tidak didominasi oleh solusi lain diberi peringkat lebih baik. Proses ini berlanjut hingga semua solusi diberi peringkat dan dikelompokkan ke dalam *front* yang berbeda. Jenis yang digunakan pada NSGA-II adalah pengembangan dari teknik *sorting* yang lama yang diberi nama *Fast Non-dominated Sorting.* Algoritma ini lebih efisien secara komputasi dibandingkan *Non-dominated Sorting* biasa, dengan kompleksitas waktu O(N log N) dibandingkan O(N^2) untuk *Non-dominated Sorting* biasa. Berikut merupakan contoh kurva pada proses *non-dominated sorting*.



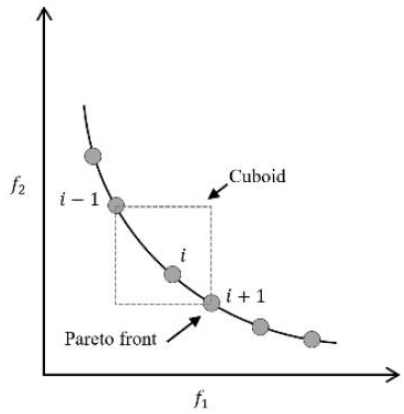
Gambar 2.3 Visual *Non-dominated Sorting*
(Sumber : (Verma *et al*., 2021)

2. Operator Pemelihara Elit (*Elite Preserving Operator*)

   Solusi terbaik dari setiap generasi secara langsung dipindahkan ke generasi berikutnya. Ini memastikan bahwa informasi berharga dari generasi sebelumnya tidak hilang.

3. *Crowding Distance*

   Crowding distance dihitung untuk setiap solusi untuk memperkirakan kepadatan populasinya. Solusi dengan *crowding distance* yang lebih besar dianggap lebih beragam dan lebih mungkin dipilih pada generasi berikutnya. Gambaran dari kepadatan jarak dari setiap solusi juga dapat dilihat dari kurva *pareto*, berikut ini merupakan contoh visualnya,



Gambar 2.4 Visual *Crowding Distance*
(Sumber : (Verma *et al*., 2021)

4. Operator Seleksi (*Selection Operator*)

   Populasi untuk generasi berikutnya dipilih menggunakan operator seleksi turnamen yang penuh sesak (*crowded tournament selection*). Operator ini mempertimbangkan peringkat dan *crowding distance* dari setiap solusi untuk memilih solusi yang lebih baik dan lebih beragam. Aturan dalam seleksinya adalah sebagai berikut:

   i. Jika terdapat dua set solusi memiliki peringkat yang berbeda, maka set solusi dengan peringkat yang lebih baik akan dipilih untuk generasi berikutnya.
   ii. Jika kedua anggota populasi memiliki peringkat yang sama, maka set solusi dengan *crowding distance* yang lebih tinggi akan dipilih untuk generasi berikutnya.

Dalam melakukan penerapan NSGA-II terdapat 9 (sembilan) langkah yang dapat dilakukan (Deb *et al*., 2002) yaitu:

1. Inisialisasi Populasi (Pt)

   Buat populasi awal (Pt) dengan N individu, di mana N adalah ukuran populasi yang telah ditentukan sebelumnya dalam *pop_size*. Setiap individu dalam populasi direpresentasikan oleh kromosom. Representasi kromosom biasanya berupa *real-coding*, di mana setiap gen mewakili nilai real untuk variabel dalam masalah optimasi multi-tujuan dalam kasus ini menunjukkan urutan pekerjaan dalam penjadwalan. Inisiasi populasi awal dapat dilakukan dengan dua cara: secara heuristik atau secara *random*. Pada penelitian ini dilakukan inisiasi populasi awal secara *random*, dengan ukuran populasi sebanyak 150 individu yang mana kromosom pada individu tersebut menggambarkan *job sequence* setiap mesin. Contoh pada penjadwalannya yang ada bentuk kromosomnya akan menjadi [([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8])].

2. Evaluasi Populasi

   Pada tahap evaluasi individu dalam populasi digunakan nilai *fitness* yang menunjukkan perhitungan nilai yang harus dimaksimalkan, diminimalkan, atau disesuaikan dengan nilai yang terdekat Penghitungan nilai fungsi tujuan (*fitness*) dilakukan terhadap untuk setiap individu dalam populasi Pt. Setiap fungsi tujuan mewakili kriteria optimasi yang ingin dicapai. Bergantung pada masalahnya, fungsi *fitness* dapat berupa nilai tunggal atau vektor nilai untuk masalah multi-tujuan. Pada studi kasus ini digunakan 2 (dua) buah obyektif yaitu minimalisasi *makespan* dan biaya keterlambatan, sehingga nilai *fitness*-nya merupakan jumlah *makespan* dan biaya keterlambatan setiap individu.

3. Fast Non-dominated Sorting

Prosedur *fast-non-dominated-sort* digunakan untuk mengidentifikasi dan mengurutkan solusi ke dalam beberapa tingkatan *front non-dominated* yang berbeda. Algoritma ini memiliki kompleksitas komputasi yang lebih efisien dibandingkan dengan algoritma *non-dominated sorting* biasa. Langkah-langkah yang diperlukan antara lain:

i. Inisialisasi dengan menghitung np (jumlah dominasi) dan Sp (himpunan solusi yang didominasi) untuk setiap solusi p dan inisialisasi himpunan kosong Q.
ii. Identifikasi solusi di *front* pertama dengan menemukan semua solusi p dengan np bernilai nol untuk kemudian tambahkan solusi ini ke *front non-dominated* pertama.
iii. Perbarui solusi yang didominasi dimana untuk setiap solusi p di *front non-dominated* pertama lakukan penghitungan nq (jumlah dominasi) untuk setiap solusi q di Sp lalu Kurangi nq dengan satu dan jika nq ada yang bernilai nol, pindahkan solusi q ke himpunan Q.
iv. Identifikasi solusi di *front* berikutnya dengan mengulangi langkah 2 dan 3 dengan himpunan Q sebagai *front non-dominated* berikutnya, sampai semua *front* berhasil diidentifikasi.

Gambaran dari proses tersebut dapat ditulis secara matematis dengan cara berikut:

```
fast-non-dominated-sort(P)
for each p ∈ P
    Sp = ∅
    np = 0
    for each q ∈ P
        if (p ≺ q) then          jika p mendominasi q
            Sp = Sp ∪ {q}        tambahkan q ke himpunan solusi yg didominasi p
        else if (q ≺ p) then
            np = np + 1          increment jumlah dominasi p
    if np = 0 then               p termasuk front pertama
        prank = 1
        F1 = F1 ∪ {p}
i = 1                            inisialisasi jumlah front
while Fi ≠ ∅
    Q = ∅                        untuk menyimpan anggota front berikutnya
    for each p ∈ Fi
        for each q ∈ Sp
            nq = nq – 1
            if nq = 0 then       q termasuk front berikutnya
                qrank = i + 1
                Q = Q ∪ {q}
    i = i + 1
    Fi = Q
```

Gambar 2.5 Alur *Fast Non-dominated Sorting*

(Sumber : (Deb *et al*., 2002))

Keluaran dari tahap ini adalah populasi yang diurutkan berdasarkan tingkat non-dominasi. Populasi dibagi menjadi beberapa *front*, di mana setiap front berisi individu dengan tingkat non-dominasi yang sama. Individu pada *front* yang lebih tinggi dianggap lebih baik daripada individu pada front yang lebih rendah.

4. *Crowding Distance Assignment*

Crowding distance mengukur kepadatan area di sekitar individu dalam ruang objektif (hasil dari fungsi *fitness*). Individu dengan *crowding distance* yang lebih tinggi berada di area yang lebih luas dan beragam, yang membantu menjaga keberagaman populasi. Penghitungan *crowding distance* dilakukan terhadap setiap individu dalam setiap *front*. Angka yang menyatakan *crowding distance* setiap solusi diperoleh dari persamaan berikut:

$$cd(i) = \sum_{i=1}^{k} \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{max} - f_j^{min}} \quad (4)$$

Keterangan:

- cd(i): *Crowding distance* untuk solusi i.
- k: Jumlah fungsi tujuan.
- f_j^{i+1}: Nilai fungsi tujuan j untuk solusi i+1.
- f_j^{i-1}: Nilai fungsi tujuan j untuk solusi i-1.
- f_j^{max}: Nilai maksimum fungsi tujuan j dalam populasi.
- f_j^{min}: Nilai minimum fungsi tujuan j dalam populasi.

5. Seleksi Turnamen Ramai (*Crowded Tournament Selection*):

   Penggunaan seleksi turnamen ramai untuk memilih dua orang tua dari populasi menggunakan turnamen biner dengan mempertimbangkan nilai fungsi tujuan dan *crowding distance*. Individu dengan nilai fungsi tujuan yang lebih baik dan *crowding distance* yang lebih tinggi memiliki peluang yang lebih tinggi untuk dipilih. Proses seleksi ini membantu memastikan bahwa individu yang lebih baik secara keseluruhan dan individu yang berada di area yang beragam dalam ruang objektif memiliki peluang yang lebih tinggi untuk bereproduksi. Keluaran dari tahap ini adalah dua orang tua yang dipilih dari populasi untuk proses *crossover.*

6. *Crossover* dan Mutasi

   Proses *crossover* dan mutasi pada individu yang dipilih digunakan untuk menghasilkan populasi baru (Qt). *Crossover* menggabungkan informasi genetik dari dua individu untuk menghasilkan keturunan baru. Operator *crossover* yang umum digunakan dalam NSGA-II adalah *Simulated Binary Crossover* (SBX) atau *Differential Evolution* (DE).

Mutasi secara acak mengubah beberapa gen dalam individu untuk memperkenalkan variasi baru ke dalam populasi. Operator mutasi yang umum digunakan adalah *gaussian mutation*.

7. Gabungkan Populasi (Rt)
   Penggabungan populasi Pt (populasi awal) dan Qt (populasi baru hasil *crossover* dan mutasi) untuk membentuk populasi baru yang lebih besar (Rt). Rt akan berisi individu-individu dari generasi sebelumnya (Pt) dan variasi barunya (Qt).

8. Seleksi Individu untuk Generasi Berikutnya (Pt+1):
   Pemilihan N individu dari *front-front* yang baru untuk membentuk populasi generasi berikutnya (Pt+1). Strategi pemilihan ini memastikan bahwa individu terbaik dan individu yang berada di area yang beragam di ruang objektif diprioritaskan untuk generasi berikutnya.). Pemilihan ini dilakukan dengan ketentuan berikut:
   - Jika ukuran *front* teratas lebih besar atau sama dengan N, maka pilih N individu dari area yang paling tidak padat (*crowding distance* tertinggi) pada front teratas tersebut untuk mengisi Pt+1.
   - Jika ukuran *front* teratas kurang dari N, maka semua individu pada *front* teratas tersebut langsung dimasukkan ke Pt+1. Sisa individu yang dibutuhkan Pt+1 diambil dari *front* berikutnya dengan mengikuti aturan yang sama (memilih dari area yang paling tidak padat). Proses ini berlanjut hingga Pt+1 terpenuhi.

9. Iterasi dengan mengulangi langkah 2 sampai 9 sampai kriteria terminasi. Urutan proses iterasi adalah pertama, populasi *Rt* dibentuk dari penggabungan populasi *Pt* dan *Qt*. Populasi *Rt* berjumlah 2N. Populasi *Rt* kemudian diurutkan menurut nilai *non-domination* sehingga dihasilkan solusi pada *front F*1 merupakan solusi terbaik. Jika ukuran *F*1 lebih kecil dari N, maka seluruh solusi pada *F*1 dimasukkan pada populasi baru *Pt*+1. Sisa jumlah populasi *Pt*+1 diisikan *F*2, *F*3 dan seterusnya. Untuk memenuhi jumlah populasi baru tepat sebanyak N solusi, maka solusi pada *front*

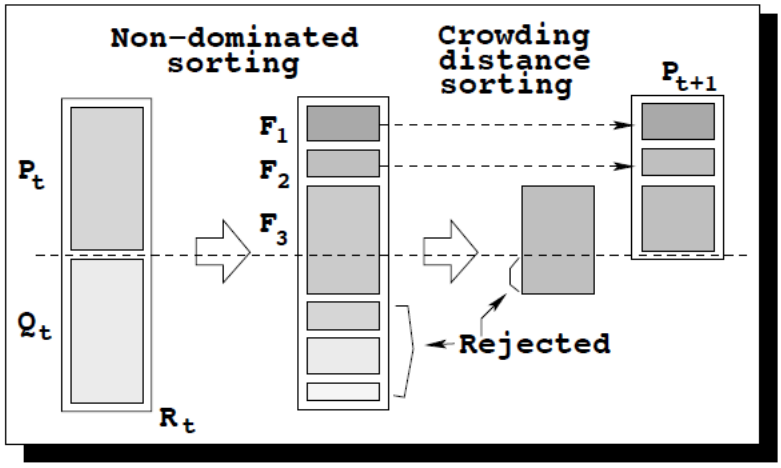
terakhir diurutkan menggunakan operator $\prec n$. Algoritma pengulangan utama dari NSGA-II, dirincikan oleh Deb *et al* (2002) sebagai berikut :

$R_t = P_t \cup Q_t$	menggabungkan parent dan offspring
F = fast-non-dominated-sort(R_t)	$F = (F_1, F_2, \ldots)$ semua *front non-dominated* R_t
$P_{t+1} = \varnothing$ and $i = 1$	
until $\lvert P_{t+1}\rvert + \lvert F_i\rvert \le N$	sampai populasi *parent* terisi
crowding-distance-assignment(F_i)	hitung crowding-distance di F_i
$P_{t+1} = P_{t+1} \cup F_i$	masukkan *front non-dominated* ke-*i* ke populasi *parent*
$i = i + 1$	periksa *front* berikutnya untuk disertakan
Sort($F_i, \prec_n$)	urutkan mengecil menggunakan $\prec_n$
$P_{t+1} = P_{t+1} \cup F_i\,[1:(N - \lvert P_{t+1}\rvert)]$	pilih elemen pertama F_i sebanyak $(N-\lvert P_{t+1}\rvert)$
Q_{t+1} = make-new-pop(P_{t+1})	gunakan seleksi, *crossover* dan mutasi untuk menghasilkan populasi baru Q_{t+1}
$t = t + 1$	increment jumlah generasi

Gambar 2.6 Iterasi Secara Matematis

(Sumber : Deb *et al* (2002))

Prosedur yang dilakukan dalam NSGA-II dapat divisualisasikan sebagai berikut,



Gambar 2.7 Prosedur NSGA-II

(Sumber : Deb *et al* (2002))

Penggunaan algoritma NSGA-II pada studi kasus penelitian dilakukan dengan urutan pekerjaan sebagai isian kromosomnya dengan besar nilai *fitness* yang bergantung pada data waktu proses, *due date*, dan biaya keterlambatan setiap operasi. Oleh karena itu, keluaran yang dihasilkan dari algoritma ini adalah urutan pekerjaan yang memiliki

lama hari *makespan* dan rupiah keterlambatan paling optimal. Pengoperasian algoritma dijalankan menggunakan sebuah *library* DEAP oleh *python* untuk memperoleh efisiensi waktu kalkulasi dan kemudahan modifikasi.

2.2.6 Pustaka Distributed Evolutionary Algorithm in Python (DEAP)

DEAP (*Distributed Evolutionary Algorithm in Python*) adalah pustaka *open-source* yang dirancang untuk memfasilitasi pengembangan dan eksperimen dengan algoritma evolusi / *evolutionary algorithm* (EA) dalam bahasa pemrograman *Python* (De Rainville *et al*., 2012). DEAP menawarkan beberapa keunggulan utama:

1. Fleksibilitas dan Kemudahan Penggunaan:
    a. DEAP menyediakan antarmuka yang intuitif dan mudah digunakan, memungkinkan pengguna untuk fokus pada desain dan implementasi EA mereka tanpa terjebak dalam detail teknis yang rumit.
    b. Kerangka kerja ini mendukung berbagai jenis EA, termasuk Pemrograman Genetik (GP), Algoritma Genetik (GA), dan Algoritma Evolusi Multi-Objektif (MOEA).
    c. DEAP menyediakan banyak komponen dasar yang diperlukan untuk membangun EA, seperti struktur data untuk kromosom, operator genetik (*crossover*, mutasi), fungsi evaluasi *fitness*, dan mekanisme seleksi.
2. Transparansi dan Kontrol:
    a. DEAP mempromosikan desain EA yang eksplisit, memungkinkan pengguna untuk melacak dan mengontrol setiap langkah proses evolusi.
    b. Pengguna memiliki akses langsung ke semua komponen kerangka kerja, sehingga mereka dapat dengan mudah memodifikasinya atau menambahkan fitur baru sesuai kebutuhan.
3. Dukungan Paralelisme:
    a. DEAP menyediakan alat untuk mendistribusikan bagian komputasi intensif dari EA ke beberapa prosesor atau mesin.
    b. Model distribusi yang digunakan tidak memiliki hierarki, memungkinkan setiap pekerja untuk membuat tugas baru dan berbagi beban kerja total aplikasi.

4. Kemudahan Prototipe Cepat:
    a. DEAP dirancang untuk memudahkan pembuatan prototipe EA baru.
    b. Pengguna dapat dengan cepat membangun dan menguji berbagai desain EA tanpa harus menulis banyak kode *boilerplate*.
5. Dukungan Komunitas:
    a. DEAP memiliki komunitas pengguna dan pengembang yang aktif, yang menyediakan dukungan dan sumber daya bagi pengguna kerangka kerja.

Versi rilis terbaru DEAP adalah 1.2.2, dan mendukung *Python* 2 dan 3. DEAP dikelola oleh Laboratorium Komputer Visi dan Sistem (CVSL) di Laval *University*, Kanada, dan tersedia di https://github.com/DEAP/deap. Kelebihan yang dimiliki oleh pustaka DEAP meningkatkan kepercayaan terhadap kekuatan pustaka tersebut. Kelebihan-kelebihan tersebut didukung fitur yang dimiliki, berikut ini fitur dalam DEAP yang akan dimanfaatkan untuk implementasi algoritma NSGA-II (De Rainville *et al*., 2012):

1. Operator Pembangkit Individu:

    Digunakan untuk membangun struktur individu yang mewakili solusi dalam masalah yang dihadapi. Operator ini bervariasi tergantung pada representasi solusi yang digunakan. Contohnya:

    - tools.initIterate: Membangun individu dengan memanggil fungsi yang diberikan berulang kali untuk mengisi elemen-elemennya.
    - tools.initUniform: Membangun individu dengan nilai elemen acak yang diambil dari rentang yang ditentukan.
    - tools.initList: Membangun individu dengan daftar nilai yang ditentukan sebelumnya.
    - creator.Individual: Struktur data yang mendefinisikan format individu dalam populasi.
2. Operator Evaluasi Fitness:

    Menghitung nilai *fitness* (kebugaran) untuk setiap individu dalam populasi. Nilai *fitness* mencerminkan seberapa baik individu tersebut menyelesaikan masalah. Operator ini harus didefinisikan oleh pengguna berdasarkan masalah yang dihadapi.

3. Operator Genetika:

   Digunakan untuk memodifikasi individu dalam populasi selama proses evolusi. Operator ini terbagi menjadi dua kategori utama:

   a. *Crossover*: Menggabungkan dua individu untuk menghasilkan individu keturunan baru. Contohnya:
      - tools.cxOnePoint: Melakukan *crossover* satu titik, memotong individu di satu titik acak dan menukar bagian yang tersisa.
      - tools.cxTwoPoint: Melakukan *crossover* dua titik, memotong individu di dua titik acak dan menukar bagian di antara titik tersebut.
      - tools.cxUniform: Melakukan *crossover* seragam, mencampur elemen acak dari kedua individu.

   b. Mutasi: Memperkenalkan variasi acak pada individu. Contohnya:
      - tools.mutFlipBit: Membalik nilai bit acak dalam individu.
      - tools.mutShuffleIndexes: Mengacak sebagian indeks elemen dalam individu.
      - tools.mutGaussian: Menambahkan *noise gaussian* acak ke nilai elemen dalam individu.

4. Operator Seleksi:

   Digunakan untuk memilih individu dari populasi untuk reproduksi. Operator ini menentukan individu mana yang akan memiliki keturunan pada generasi berikutnya. Contohnya:
   - tools.selTournament: Memilih individu secara acak dari turnamen kecil dan memilih individu dengan nilai *fitness* terbaik.
   - tools.selRoulette: Memilih individu dengan probabilitas yang sebanding dengan nilai *fitness* mereka.
   - tools.selNSGA2: Memilih individu berdasarkan nilai non-dominasi dan *crowding distance* dalam algoritma NSGA-2.

5. Algoritma Genetika:

Mendefinisikan kerangka kerja keseluruhan untuk menjalankan proses evolusi. Algoritma ini menggabungkan operator-operator yang disebutkan di atas untuk menghasilkan populasi solusi yang lebih baik dari waktu ke waktu. Contohnya:

- algorithms.eaSimple: Algoritma EA sederhana yang menjalankan *crossover*, mutasi, dan seleksi secara berulang.
- algorithms.esSimple: Algoritma ES sederhana yang menjalankan mutasi dan seleksi secara berulang.
- algorithms.eaMuLambda: Algoritma EA dengan strategi μ+λ, di mana μ individu dipilih untuk reproduksi dan λ individu keturunan dihasilkan.

6. Operator Statistik:

   Digunakan untuk mengumpulkan informasi tentang populasi selama proses evolusi. Informasi ini dapat digunakan untuk menganalisis kinerja algoritma dan memvisualisasikan hasil. Contohnya:

   - tools.Statistics: Menyimpan dan memperbarui statistik populasi seperti nilai rata-rata, standar deviasi, minimum, dan maksimum dari nilai *fitness*.
   - tools.PlotX: Membuat plot nilai *fitness* terhadap generasi.
   - tools.Diversity: Menghitung metrik keragaman populasi.

Penelitian akan menyusun sebuah *script* menggunakan operator DEAP yang relevan untuk membangun individu, mengevaluasi *fitness*, melakukan *crossover*, mutasi, seleksi, dan menjalankan algoritma NSGA-2 secara keseluruhan. Kombinasi operator ini memungkinkan pencarian solusi optimal untuk masalah penjadwalan *flowshop* dengan kriteria *makespan* dan keterlambatan terkecil.

BAB III

METODE PENELITIAN

3.1 Obyek Penelitian

Obyek penelitian merupakan situasi sosial yang ingin diketahui terkait apa yang terjadi (Sugiyono, 2013). Pada penelitian "Optimasi Multi-Obyektif pada Penjadwalan Produksi untuk Minimalisasi *Makespan* dan Biaya Keterlambatan", obyek penelitian atau permasalahan yang dibahas adalah aliran produksi produk di PT. INKA. Berikut ini merupakan profil dari perusahaan PT. INKA:

Nama Perusahaan	:	PT. Industri Kereta Api /PT. INKA
Alamat	:	Jl. Yos Sudarso No. 71 Madiun 63122, Jawa Timur
No. Telepon	:	(0351) 452271-74
Bidang Usaha	:	- Manufaktur Sarana Kereta Api - Konstruksi pengadaan rekayasa - *Service* dan *Retail* - Jasa rehabilitasi atau *retrofit* sarana kereta api - Jasa e*ngineering* dan desain kereta api
Bentuk Perusahaan	:	Perseroan Terbatas
Tahun Pendirian	:	1981

Aliran produksi akan diatur penjadwalannya untuk mencapai rute yang optimal ditandai dengan *makespan* dan biaya keterlambatan yang lebih kecil. Penjadwalan produksi dilakukan menggunakan data dari aliran produksi produk di PT. INKA yaitu data dari lantai produksi berupa jumlah stasiun kerja, jumlah mesin, jumlah pekerjaan, aliran produksi, bobot keterlambatan, dan penjadwalan yang berlaku. Data tersebut dikumpulkan dan diolah menggunakan algoritma NSGA-II sehingga keluarannya berupa jadwal produksi yang dinilai optimal dan efisien terhadap *due date* pekerjaan.

3.2 Jenis Data Penelitian

Jenis data yang digunakan dalam penelitian "Optimasi Multi-Obyektif NSGA-II pada Penjadwalan Produksi untuk Minimalisasi *Makespan* dan Biaya Keterlambatan" ada dua macam, yaitu:

3.2.1 Data Primer

Data primer merupakan data yang bersumber langsung dari narasumber dan diberikan kepada pengumpul data, dalam hal ini *expert* merupakan narasumber, yaitu berupa jumlah stasiun kerja, jumlah mesin, jumlah pekerjaan, aliran produksi, biaya keterlambatan, dan penjadwalan yang berlaku. Pada penelitian ini kriteria yang diterapkan untuk narasumber atau *expert* ada 5 (lima) yaitu (Faisal, 1990):

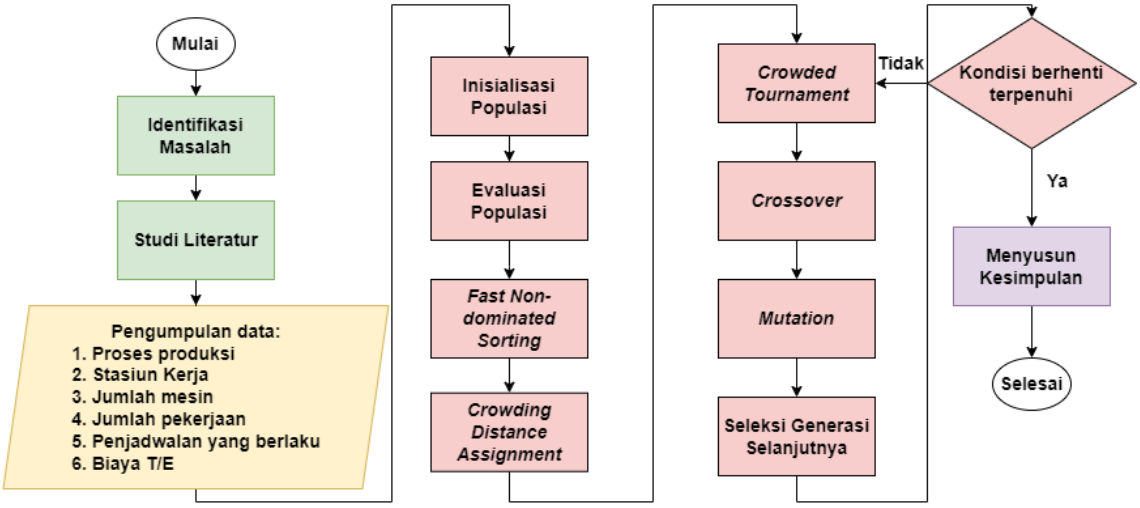
a. Mereka yang menguasai atau memahami proses manufaktur di PT. INKA.
b. Mereka yang tergolong masih berkecimpung atau terlibat dalam proses manufaktur di PT. INKA
c. Mereka yang memiliki waktu yang memadai untuk dimintai informasi
d. Mereka yang cenderung menyampaikan informasi bukan hasil kemasannya sendiri
e. Mereka yang tergolong "asing" dengan peneliti

3.2.2 Data Sekunder

Data sekunder merupakan data yang bersumber tidak langsung dari narasumber dan diberikan kepada pengumpul data, namun dalam hal ini berasal dari orang lain atau dokumen atau dapat dikatakan data pendukung penelitian yang berasal dari jurnal atau buku. Seperti informasi pada kajian literatur dan pendukung data pada latar belakang.

3.3 Alur Penelitian

Alur penelitian "Optimasi Multi-Obyektif pada Penjadwalan Produksi untuk Minimalisasi *Makespan* dan Biaya Keterlambatan" dijelaskan pada Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Detail dari setiap tahapan dalam alur penelitian pada Gambar 3.1 dijelaskan sebagai berikut.

1. Mulai

Tahapan mulai merupakan tahapan awal penelitian dimana pada tahap ini dilakukan penyiapan alat dan bahan, serta metode yang digunakan.

2. Identifikasi Masalah

Berdasarkan pokok masalah yang akan diteliti kemudian menentukan subjek dan objek penelitian.

3. Studi Literatur

Terdapat dua kajian literatur pada penelitian kali ini, yaitu landasan teori dan penelitian terdahulu. Landasan teori yaitu berupa teori keilmuan dari penjadwalan produksi, definisi dalam penjadwalan produksi, penjadwalan *flowshop*, optimasi multi-obyektif, dan *Non-dominated Sorting Genetic Algorithm* II (NSGA-II). Penelitian terdahulu merupakan hasil penelitian sebelumnya sesuai dengan metode yang digunakan sebagai bahan acuan dasar dan pendukung.

4. Pengumpulan Data

Tahapan pengumpulan data dilakukan melalui wawancara dengan pengawas maupun operator dan melalui observasi serta pengukuran langsung.

5. Inisialisasi Populasi

Tahapan ini merupakan tahap menyusun kromosom awalan yang merupakan susunan penjadwalan yang telah ada, dimana mewakili setiap pekerjaan (j) dikerjakan pada mesin (m) dengan susunan operasi (o) bagaimana. Pekerjaan (j) merupakan produk yang dikerjakan, mesin (m) merupakan stasiun kerja, sedangkan operasi (o) merupakan penempatan sebuah pekerjaan (n) pada sebuah stasiun kerja (m) pada waktu tertentu. Pada penelitian ini dilakukan inisiasi populasi awal secara *random* , dengan ukuran populasi sebanyak 100 individu yang mana kromosom pada individu tersebut menggambarkan *job sequence*. Pada penelitian ini terdapat 9 (Sembilan) pekerjaan yang akan dijadwalkan, maka kromosom akan berisi angka 0 hingga 8 yang mewakili pekerjaan tersebut. Contoh pada penjadwalannya yang ada bentuk kromosomnya akan menjadi [([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8]), ([0, 1, 2, 3, 4, 5, 6, 7, 8])].

6. Evaluasi Populasi

Evaluasi populasi merupakan tahapan untuk menentukan nilai *fitness* pada seluruh populasi awal. Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai *fitness* ini yang dijadikan acuan dalam mencapai nilai optimal dalam NSGA-II. Pada penelitian ini nilai *fitness* merupakan total biaya keterlambatan dalam sebuah penjadwalan. Berikut ini merupakan hitungan yang digunakan untuk mengevaluasi individu optimal.

$$C_{max} = \max_{j,m} C_{j,m} \tag{5}$$

$$TLC = \sum_{j=1}^{n} \sum_{m=1}^{m} \left(L_{j,m} \cdot c_{j,m}\right) \tag{6}$$

$$Fitness = (Makespan, \text{Biaya Keterlambatan})$$

Keterangan:

- j = Jumlah pekerjaan
- m = Jumlah mesin
- $C_{j,m}$ = Waktu penyelesaian pekerjaan ke-i pada mesin ke-j

- $L_{j,m}$ = Waktu keterlambatan pekerjaan ke-i pada mesin ke-j
- $c_{j,m}$ = Biaya keterlambatan pekerjaan ke-i pada mesin ke-j

7. *Fast Non-dominated Sorting*

Tahapan ini dilakukan terhadap seluruh individu dengan mengukur dominasi sebuah solusi terhadap solusi lainnya untuk kemudian dikelompokkan dalam *front* sesuai ukuran dominasinya untuk menentukan angka *front*-nya.

8. *Crowding Distance Assignment*

Tahapan ini dilakukan terhadap seluruh individu dengan mengukur kepadatan area di sekitar individu dalam ruang objektif (hasil dari fungsi *fitness*) untuk menentukan skor *crowding distance.*

9. *Crowded Tournament Selection*

Tahapan ini ditujukan untuk memilih N individu dari populasi dengan mempertimbangkan peringkat dominasi *Pareto* / *front* dan *crowding distance* dengan ketentuan sebagai berikut.

- Individu dengan peringkat dominasi *Pareto* yang lebih baik selalu dipilih di atas individu dengan peringkat yang lebih rendah.
- Jika individu memiliki peringkat yang sama, individu dengan *crowding distance* yang lebih tinggi memiliki peluang yang lebih tinggi untuk dipilih.

10. *Crossover*

Tahapan *crossover* merupakan tahapan operator dari NSGA-II yang melibatkan dua induk untuk membentuk kromosom baru. Proses ini dilakukan dengan menukar sebagian informasi pada kromosom induk pertama dengan informasi dari kromosom induk kedua. Penelitian ini akan menggunakan jenis *crossover* seragam, mencampur elemen acak dari kedua individu namun tetap mempertahankan segmen yang baik dari orang tua. Besar probabilitas *crossover* yang diinginkan harus cukup besar untuk memperoleh keragaman populasi namun tidak boleh sepenuhnya mengacak karena harus ada segmen yang dipertahankan. Oleh karena itu, probabilitas yang digunakan untuk *crossover* adalah 0.7 di antara rentang 0 hingga 1.

11. Mutasi

Tahapan mutasi merupakan tahapan operator yang dapat mengubah gen-gen dalam kromosom. Dalam parameter NSGA-II, terdapat beberapa batasan. Salah satunya adalah ukuran populasi. Dalam algoritma genetika, ukuran populasi setiap generasi adalah tetap. Mutasi yang ingin dilakukan adalah mengacak hanya sebagian indeks elemen dalam individu untuk mempertahankan segmen yang memiliki nilai baik. Oleh karena itu, dipilih nilai probabilitas mutasi yang tidak terlalu tinggi untuk mengimbangi nilai *crossover* yang sudah tinggi. Sehingga, nilai probabilitas mutasi yang dipilih adalah 0.3.

10. Seleksi Individu untuk Generasi Berikutnya (Pt+1):

Menggabungkan populasi orang tua dan anak. Urutkan populasi gabungan berdasarkan tingkat non-dominasi dan *crowding distance.* Pilih N individu terbaik dari populasi gabungan untuk membentuk populasi baru.

12. Menyusun Kesimpulan

Memberikan kesimpulan dan saran yang menjawab tujuan penelitian yang telah ditentukan sebelumnya.

13. Selesai

Tahapan selesai merupakan tahapan terakhir penelitian yang ditandai dengan adanya solusi penjadwalan dan tuntasnya laporan penelitian

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

Pengumpulan data penelitian yang akan diolah dengan metode terpilih dilakukan secara langsung pada lokasi studi kasus untuk menjawab rumusan masalah yang ditentukan. Berikut ini merupakan data yang digunakan dalam penelitian:

4.1.1 Proses Produksi

Proses produksi merupakan langkah-langkah dari sebuah kegiatan yang ada dalam perusahaan untuk mengubah sebuah *input* menjadi *output* yang mempunyai nilai tambah. Berikut merupakan alur proses produksi PT. INKA (Persero) mulai dari kontrak dengan konsumen hingga proses manufaktur produk secara umum dimana alur proses produksi manufaktur dari PT. INKA (Persero) terbagi menjadi dua, yaitu:

1. Proses Fabrikasi

   Proses fabrikasi merupakan proses awal dari pembuatan kereta api mulai dari kedatangan *raw material* hingga proses *assembly* menjadi beberapa bagian kereta, seperti *underframe, roof, end wall, sidewall,* dan *carbody*. Adapun proses-proses yang ada dalam proses fabrikasi adalah sebagai berikut:

   a. Proses *Steel Work*, proses dilakukan menggunakan peralatan dan mesin, seperti mesin *grinding, cutting, bending, rolling, punching, lathe, milling, drilling,* dan *heat treat*. Produk yang dihasilkan pada proses ini, yaitu plat- plat metal dengan berbagai dimensi dan berat menjadi komponen *detail part manufacturing* (DPM)
   b. Proses Minor *Assembly*, proses dilakukan dengan cara menggabungkan bagian-bagian kecil atau minor *part* yang sudah diselesaikan pada proses *steel work. Minor part* tersebut akan diproses menggunakan mesin *welding* dan menjadi produk, yaitu *bolster underframe, ceiling, accessories, end underframe,* dan partisi.

c. Proses Sub *Assembly*, proses dilakukan dengan cara menggabungkan komponen-komponen yang sudah digabungkan pada proses minor *assembly*. Komponen-komponen tersebut akan diproses menggunakan mesin *welding* dan menjadi produk, yaitu *underframe, roof, end wall*, dan *side wall*.
d. Proses *Carbody Assembly*, proses dilakukan dengan cara menggabungkan komponen-komponen yang sudah digabungkan pada proses sub *assembly* menjadi satu badan atau struktur kereta yang lengkap.

2. Proses *Finishing*

Pada proses *finishing* merupakan proses akhir dari pembuatan kereta api. Adapun proses-proses yang ada dalam proses *finishing* adalah sebagai berikut:

a. Proses *Blasting*, proses dilakukan dengan cara *sand blasting carbody, black coat*, dan *solvent cleaning* untuk membersihkan permukaan *carbody* dengan cara menyemprotkan *abasive* material sebelum masuk ke proses painting.
b. Proses *Painting*, proses dilakukan dengan cara mengecat dan memberi warna pada *carbody* mulai dari primer *painting* hingga *finishing painting body*.
c. Proses *Equipment Fitting*, terdapat tiga proses yang dilakukan, yaitu:
   - *Equipmnet Fitting Electrical* yang dilakukan dengan cara instalasi kelistrikan, seperti *fitting head lamp, fitting hog lamp, fitting wiring, fitting cable duct, fitting panel*, dan instalasi lainnya yang berhubungan dengan kelistrikan.
   - *Equipment Fitting Mechanical* yang dilakukan dengan cara instalasi permesinan, seperti sistem pengereman, alat perangkai, dan instalasi lainnya.
   - *Equipment Fitting Interior* yang dilakukan dengan cara pemasangan interior pada kereta api, seperti kursi penumpang, meja, jendela, lantai, air *conditioner*, alat pemadam api ringan (APAR), dan interior lainnya.
d. Proses *Bogie Mounting*, proses dilakukan dengan cara menggabungkan *carbody* atau badan kereta dengan *bogie* atau kerangka roda menjadi sebuah kereta api yang siap dioperasikan.
e. Proses Testing, terdapat dua proses yang dilakukan, yaitu:
   - Testing *Static* yang dilakukan dengan cara pengujian statis menggunakan simulasi hujan badai atau menyemprotkan air ke badan kereta api dengan

tekanan tinggi. Hal ini dilakukan untuk menguji ketahanan kereta api terhadap kondisi cuaca apabila terjadi hujan badai.

- Testing *Dynamic* dilakukan dengan cara pengujian dinamis atau uji kelayakan jalan. Hal ini dilakukan untuk mengetahui kondisi kereta api dan kemampuan kerja sarana perkeretaapian saat kondisi bergerak atau beroperasi.

f. Proses *Delivery*, proses pengiriman produk kepada pihak konsumen setelah kereta api berhasil melewati seluruh proses produksi.

Pada studi kasus ini, data yang digunakan adalah tahapan *finishing* tepatnya pada proses *fitting* dan *bogie mounting*. Proses *fitting* dan *bogie mounting* terbagi menjadi sejumlah 6 (enam) stasiun kerja atau pada perusahaan disebut Takt. Takt pada proses tersebut digunakan pada studi kasus ini sebagai mesin yang melakukan setiap operasi dan disimbolkan sebagai Mn dimana n mewakili urutan mesin. Berikut ini merupakan rincian untuk setiap Takt pada kedua proses tersebut:

Tabel 4.1 Stasiun Kerja

Proses	**Stasiun Kerja**	**No.**	**Nama Aktivitas**
Fitting	Takt 3 (M1)	1.	*Fitting Lavatory*
		2.	*Lavatory Module*
		3.	*Roof Equipment*
	Takt 4 (M2)	1.	Perpipaan *Brake System*
		2.	*Equipment Beneath Underframe*
		3.	*Routing* Perkabelan
	Takt 5 (M3)	1.	*Fitting* Perpintuan
		2.	*Fitting* Panel Interior
		3.	*Fitting Passenger Seat*
	Takt 6 (M4)	1.	*Modular Partition*
		2.	*Piping Of Lavatory*
	Takt 7 (M5)	1.	*Accessories* & Komponen Interior
		2.	Koneksi Elektrik
Bogie Mounting	Takt 8 (M6)	1.	*Bogie Mounting*

4.1.2 Data Pekerjaan

Data pekerjaan pada penelitian ini merupakan gerbong yang diproduksi untuk menyelesaikan sebuah *trainset*. Sebuah *trainset* untuk kereta Eksekutif terdiri dari 9 (sembilan) gerbong penumpang (K1), 1 (satu) gerbong makan, dan 1 (satu) kereta penggerak. Pada studi kasus ini, data pekerjaan yang digunakan adalah 9 (sembilan)

gerbong penumpang sehingga pekerjaan yang dilakukan adalah K1-1, K1-2, K1-3, K1-4, K1-5, K1-6, K1-7, K1-8, dan K1-9.

4.1.3 Biaya Keterlambatan

Keterlambatan produksi sering kali menyebabkan domino peristiwa krusial pada perusahaan, baik secara langsung ke sistem produksi maupun secara tidak langsung ke tata kelola organisasi. Pada penelitian ini, peristiwa krusial yang dipilih adalah penambahan biaya tenaga kerja harian untuk operator produksinya. Setiap mesin memiliki jumlah kebutuhan pekerja yang berbeda sehingga besar biaya tenaga kerja hariannya juga berbeda dan mewakili biaya keterlambatan setiap mesin. Berikut ini merupakan rangkuman biaya tenaga kerja setiap mesinnya.

Tabel 4.2 Biaya Keterlambatan Mesin

Mesin	Jumlah Pekerja	Biaya Tenaga Kerja Harian	Biaya Keterlambatan Harian
M1	8	Rp100,000.00	Rp800,000.00
M2	4	Rp100,000.00	Rp400,000.00
M3	6	Rp100,000.00	Rp600,000.00
M4	4	Rp100,000.00	Rp400,000.00
M5	3	Rp100,000.00	Rp300,000.00
M6	4	Rp100,000.00	Rp400,000.00

Tabel 4.2 menunjukkan skema tenaga kerja pada setiap mesin, dimana pekerja terbanyak berada pada mesin 1 (M1) yaitu sejumlah 8 (delapan) orang. Setiap pekerja diberi upah harian yang sama yaitu sejumlah Rp100,000.00 per hari kerja. Oleh karena itu, hasil perkalian jumlah pekerja pada suatu mesin / stasiun kerja dengan besar upah harian menunjukkan besar biaya tenaga kerja per hari apabila terdapat kemungkinan penambahan hari karena penyelesaian pekerjaan tidak tepat waktu.

4.1.4 Penjadwalan Yang Berlaku

Pekerjaan pada studi kasus ini mewakili jenis gerbong pada sebuah *trainset* kereta eksekutif, dimana pada sebuah *trainset* tersebut jumlah gerbong penumpang adalah sebanyak 9 (sembilan) gerbong. Setiap gerbong tersebut dibuat melalui serangkaian operasi yang dijadwalkan masing-masing waktunya. Berikut ini merupakan penjadwalan yang saat ini berlaku beserta realisasi dari penjadwalan tersebut.

Tabel 4.3 Penjadwalan Berlaku dan Realisasi Penjadwalan

Job	Mesin	Jadwal			Realisasi		
		Mulai	**Selesai**	**Durasi**	**Mulai**	**Selesai**	**Durasi**
K1	M1	03/08/23	04/08/23	2 hari	17/08/23	20/08/23	3 hari
K3	M1	05/08/23	06/08/23	2 hari	20/08/23	23/08/23	3 hari
K4	M1	07/08/23	08/08/23	2 hari	23/08/23	26/08/23	3 hari
K5	M1	09/08/23	10/08/23	2 hari	26/08/23	29/08/23	3 hari
K6	M1	11/08/23	12/08/23	2 hari	29/08/23	01/09/23	3 hari
K7	M1	13/08/23	14/08/23	2 hari	01/09/23	04/09/23	3 hari
K8	M1	15/08/23	16/08/23	2 hari	04/09/23	07/09/23	3 hari
K9	M1	17/08/23	18/08/23	2 hari	07/09/23	10/09/23	3 hari
K10	M1	19/08/23	20/08/23	2 hari	10/09/23	14/09/23	4 hari
K1	M2	05/08/23	06/08/23	2 hari	18/08/23	20/08/23	2 hari
K3	M2	07/08/23	08/08/23	2 hari	20/08/23	22/08/23	2 hari
K4	M2	09/08/23	10/08/23	2 hari	22/08/23	24/08/23	2 hari
K5	M2	11/08/23	12/08/23	2 hari	24/08/23	26/08/23	2 hari
K6	M2	13/08/23	14/08/23	2 hari	26/08/23	28/08/23	2 hari
K7	M2	15/08/23	16/08/23	2 hari	28/08/23	30/08/23	2 hari
K8	M2	17/08/23	18/08/23	2 hari	30/08/23	03/09/23	4 hari
K9	M2	19/08/23	20/08/23	2 hari	03/09/23	07/09/23	4 hari
K10	M2	21/08/23	22/08/23	2 hari	07/09/23	11/09/23	4 hari
K1	M3	07/08/23	08/08/23	2 hari	23/08/23	27/08/23	4 hari
K3	M3	09/08/23	10/08/23	2 hari	27/08/23	31/08/23	4 hari
K4	M3	11/08/23	12/08/23	2 hari	31/08/23	04/09/23	4 hari
K5	M3	13/08/23	14/08/23	2 hari	04/09/23	08/09/23	4 hari
K6	M3	15/08/23	16/08/23	2 hari	08/09/23	12/09/23	4 hari
K7	M3	17/08/23	18/08/23	2 hari	12/09/23	16/09/23	4 hari
K8	M3	19/08/23	20/08/23	2 hari	16/09/23	22/09/23	6 hari
K9	M3	21/08/23	22/08/23	2 hari	22/09/23	28/09/23	6 hari
K10	M3	23/08/23	24/08/23	2 hari	28/09/23	04/10/23	6 hari
K1	M4	09/08/23	10/08/23	2 hari	03/09/23	06/09/23	3 hari
K3	M4	11/08/23	12/08/23	2 hari	06/09/23	09/09/23	3 hari
K4	M4	13/08/23	14/08/23	2 hari	09/09/23	12/09/23	3 hari
K5	M4	15/08/23	16/08/23	2 hari	12/09/23	15/09/23	3 hari
K6	M4	17/08/23	18/08/23	2 hari	15/09/23	18/09/23	3 hari
K7	M4	19/08/23	20/08/23	2 hari	18/09/23	21/09/23	3 hari
K8	M4	21/08/23	22/08/23	2 hari	21/09/23	25/09/23	4 hari
K9	M4	23/08/23	24/08/23	2 hari	25/09/23	29/09/23	4 hari
K10	M4	25/08/23	26/08/23	2 hari	29/09/23	03/10/23	4 hari
K1	M5	11/08/23	12/08/23	2 hari	30/08/23	02/09/23	3 hari
K3	M5	13/08/23	14/08/23	2 hari	02/09/23	05/09/23	3 hari
K4	M5	15/08/23	16/08/23	2 hari	05/09/23	08/09/23	3 hari
K5	M5	17/08/23	18/08/23	2 hari	08/09/23	11/09/23	3 hari
K6	M5	19/08/23	20/08/23	2 hari	11/09/23	14/09/23	3 hari
K7	M5	21/08/23	22/08/23	2 hari	14/09/23	17/09/23	3 hari
K8	M5	23/08/23	24/08/23	2 hari	17/09/23	22/09/23	5 hari
K9	M5	25/08/23	26/08/23	2 hari	22/09/23	27/09/23	5 hari
K10	M5	27/08/23	28/08/23	2 hari	27/09/23	03/10/23	6 hari
K1	M6	13/08/23	14/08/23	2 hari	04/09/23	06/09/23	2 hari

Job	Mesin	Jadwal			Realisasi		
		Mulai	Selesai	Durasi	Mulai	Selesai	Durasi
K3	M6	15/08/23	16/08/23	2 hari	06/09/23	08/09/23	2 hari
K4	M6	17/08/23	18/08/23	2 hari	08/09/23	10/09/23	2 hari
K5	M6	19/08/23	20/08/23	2 hari	10/09/23	12/09/23	2 hari
K6	M6	21/08/23	22/08/23	2 hari	12/09/23	14/09/23	2 hari
K7	M6	23/08/23	24/08/23	2 hari	14/09/23	16/09/23	2 hari
K8	M6	25/08/23	26/08/23	2 hari	16/09/23	18/09/23	2 hari
K9	M6	27/08/23	28/08/23	2 hari	18/09/23	20/09/23	2 hari
K10	M6	29/08/23	30/08/23	2 hari	20/09/23	22/09/23	2 hari

Tabel 4.3 menunjukkan tanggal mulai dan selesai dari setiap operasi, dimana terdapat 2 (dua) jenis tanggal mulai dan selesai yaitu tanggal mulai dan selesai berdasarkan penjadwalan yang berlaku dan berdasarkan realisasi operasi dilaksanakan. Penjadwalan yang berlaku memiliki tanggal mulai dari 3 Agustus 2023 dan berakhir tanggal 30 Agustus 2023, sedangkan realisasi memiliki tanggal mulai 17 Agustus 2023 dan berakhir tanggal 4 Oktober 2023. Setiap rentang waktu atau durasi, baik penjadwalan maupun realisasi memiliki fungsi dalam pengolahan data. Durasi operasi pada penjadwalan digunakan sebagai data *due date*, sedangkan durasi operasi pada realisasi digunakan sebagai data waktu proses.

4.2 Pengolahan Data

4.2.1 Identifikasi Keterlambatan

Identifikasi keterlambatan merupakan aktivitas membandingkan tanggal penyelesaian operasi pada jadwal terhadap penyelesaian operasi pada realisasi. Selisih dari kedua tanggal tersebut mewakili jumlah hari keterlambatan setiap aktivitas. Guna mengukur nilai keterlambatan total yang akan digunakan sebagai parameter ukur, maka jumlah hari keterlambatan tersebut akan dikalikan dengan biaya keterlambatan per hari untuk setiap operasi. Nilai total dari biaya keterlambatan mewakili besar kerugian yang dialami perusahaan atas penjadwalan yang sedang berlaku. Berikut ini merupakan rangkuman keterlambatan setiap operasi.

Tabel 4.4 Identifikasi Keterlambatan

Pekerjaan	Mesin	Keterlambatan (hari)	Total Biaya Keterlambatan
K1	M1	16	Rp12,800,000.00
K3	M1	17	Rp13,600,000.00
K4	M1	18	Rp14,400,000.00
K5	M1	19	Rp15,200,000.00
K6	M1	20	Rp16,000,000.00

Pekerjaan	Mesin	Keterlambatan (hari)	Total Biaya Keterlambatan
K7	M1	21	Rp16,800,000.00
K8	M1	22	Rp17,600,000.00
K9	M1	23	Rp18,400,000.00
K10	M1	25	Rp20,000,000.00
K1	M2	14	Rp5,600,000.00
K3	M2	14	Rp5,600,000.00
K4	M2	14	Rp5,600,000.00
K5	M2	14	Rp5,600,000.00
K6	M2	14	Rp5,600,000.00
K7	M2	14	Rp5,600,000.00
K8	M2	16	Rp6,400,000.00
K9	M2	18	Rp7,200,000.00
K10	M2	20	Rp8,000,000.00
K1	M3	19	Rp11,400,000.00
K3	M3	21	Rp12,600,000.00
K4	M3	23	Rp13,800,000.00
K5	M3	25	Rp15,000,000.00
K6	M3	27	Rp16,200,000.00
K7	M3	29	Rp17,400,000.00
K8	M3	33	Rp19,800,000.00
K9	M3	37	Rp22,200,000.00
K10	M3	41	Rp24,600,000.00
K1	M4	27	Rp10,800,000.00
K3	M4	28	Rp11,200,000.00
K4	M4	29	Rp11,600,000.00
K5	M4	30	Rp12,000,000.00
K6	M4	31	Rp12,400,000.00
K7	M4	32	Rp12,800,000.00
K8	M4	34	Rp13,600,000.00
K9	M4	36	Rp14,400,000.00
K10	M4	38	Rp15,200,000.00
K1	M5	21	Rp6,300,000.00
K3	M5	22	Rp6,600,000.00
K4	M5	23	Rp6,900,000.00
K5	M5	24	Rp7,200,000.00
K6	M5	25	Rp7,500,000.00
K7	M5	26	Rp7,800,000.00
K8	M5	29	Rp8,700,000.00
K9	M5	32	Rp9,600,000.00
K10	M5	36	Rp10,800,000.00
K1	M6	23	Rp9,200,000.00
K3	M6	23	Rp9,200,000.00
K4	M6	23	Rp9,200,000.00
K5	M6	23	Rp9,200,000.00
K6	M6	23	Rp9,200,000.00
K7	M6	23	Rp9,200,000.00
K8	M6	23	Rp9,200,000.00
K9	M6	23	Rp9,200,000.00

Pekerjaan	Mesin	Keterlambatan (hari)	Total Biaya Keterlambatan
K10	M6	23	Rp9,200,000.00
Total			Rp621,200,000.00

Keterlambatan telah diidentifikasi untuk setiap operasi dengan angka *makespan* yang diharapkan adalah 27 hari, namun keterlambatan yang ditunjukkan oleh kenaikan angka *makespan* menjadi 62 hari. Hal tersebut menunjukkan bahwa proyek terlambat selama 35 hari dengan total biaya keterlambatannya adalah sebesar Rp621,200,000.00. Perubahan parameter angka *makespan* dan biaya keterlambatan akan menjadi variabel dependen yang terus diamati pasca penjadwalan dengan metode yang diusulkan.

4.2.2 Pengaturan Parameter Algoritma NSGA-II

Penggunaan *python* sebagai *tools* untuk mengaplikasikan algoritma NSGA-II memerlukan beberapa modul dan operator yang berkaitan. Berikut ini merupakan modul dan operator yang digunakan dalam penelitian ini:

A. Pustaka yang digunakan:

1. Pandas (pd)

   Pandas adalah pustaka Python yang digunakan untuk manipulasi dan analisis data yang mudah dan cepat. Pada penelitian ini, pandas digunakan untuk membaca dan menyimpan dataset dari *file* CSV serta mengelola data dalam bentuk *DataFrame*. Berikut ini merupakan penggunaan Pandas pada penelitian:

   - pd.read_csv, berfungsi untuk membaca data dari dokumen dengan format CSV agar terbaca sebagai obyek *DataFrame* Pandas. Dokumen yang dibaca adalah dokumen csv berisi matriks waktu pemrosesan operasi (‘processing_times.csv), due date operasi (‘due_dates.csv), dan biaya keterlambatan operasi (‘tardiness_costs.csv)
   - pd.DataFrame, berfungsi untuk mennyusun list mauapun array menjadi sebuah DataFrame yang rapi. Pada penelitian ini list yang disusun merupakan rekap data berkaitan urutan generasi, pareto, dan nilai *fitness* setiap generasi disimpan sebagai 'all_generations_fitness.csv'.

2. NumPy (np)

NumPy merupakan pustaka untuk komputasi ilmiah yang menyediakan dukungan untuk *array* besar multidimensi dan berbagai fungsi matematika tingkat tinggi. Pada penelitian ini, NumPy digunakan untuk operasi numerik seperti membuat permutasi acak, menginisialisasi *array*, dan menghitung nilai *hypervolume*. Berikut ini merupakan penggunaan NumPy pada penelitian:

- np.random.permutation(), berfungsi untuk menghasilkan urutan acak dari elemen array. Urutan tersebut yang kemudian menjadi populasi yang akan diolah selanjutnya sebagai calon solusi.
- np.zeros(), berfungsi untuk membuat *array* baru yang diisi dengan nilai nol dimana ia akan menginisialisasi array *schedule* yang nantinya akan menyimpan jadwal pengerjaan pekerjaan pada tiap mesin.
- np.array, berfungsi untuk mengonversi list dari nilai *fitness* menjadi sebuah NumPy array untuk dapat dilanjutkan dalam komputasi yang menggunakan np.
- np.max(), berfungsi untuk mencari nilai maksimum dalam *array* dari nilai *fitness* setiap individu.
- np.min(), berfungsi untuk mencari nilai minimum dalam *array* dari nilai *fitness* setiap individu.
- np.mean(), berfungsi untuk mencari nilai rata-rata dalam *array* dari nilai *fitness* setiap individu.
- np.std(), berfungsi untuk mencari nilai standar deviasi dalam *array* dari nilai *fitness* setiap individu.

3. DEAP (deap.base, deap.creator, deap.tools, deap.algorithms)

   Pustaka DEAP menyediakan berbagai jenis modul yang dapat digunakan untuk mendukung berjalannya NSGA-II. Berikut ini merupakan modul yang digunakan pada penelitian:

   - deap.base, berperan dalam menyediakan kelas-kelas untuk individu, populasi, *sequence* dan operator genetik. Pada penelitian ini deap.base digunakan beberapa fungsi berikut:
     - *Individual*, untuk merepresentasikan solusi yang merupakan urutan pekerjaan yang ditugaskan pada setiap mesin.
     - *Population*, untuk merepresentasikan kumpulan individu untuk proses evolusi
     - *Fitness*, untuk merepresentasikan evaluasi *fitness* sebuah individu dengan format *makespan, cost_of_tardiness*.

    - Operator, untuk merepresentasikan operator genetik seperti *crossover* dan mutasi guna yang memanipulasi individu
- deap.creator, berperan dalam memfasilitasi pembuatan kelas-kelas seperti individu, *fitness*, dan operator.
- deap.tools, berperan dalam menyediakan fungsi yang umum untuk mengoperasikan algoritma genetika. Berikut ini merupakan *tools* yang digunakan dalam penelitian ini:
    - register(), untuk mendaftarkan sebuah fungsi sehingga dapat terakses oleh algoritma yang sedang berjalan.
    - initRepeat(), untuk membuat urutan berulang dari sebuah elemen pada setiap kelas
    - select(), untuk memilih k individu dari sebuah populasi berdasarkan strategi *select* tertentu seperti seleksi turnamen maupun *elitism*.
    - mate(), untuk mengaplikasikan operasi *crossover* pada orang tua sehingga menghasilkan keturunan
    - mutate(), untuk mengaplikasi mutasi pada individu sehingga terjadi perubahan tertentu
    - statistics(), untuk melacak kondisi statistik dari sebuah populasi selama proses evolusi.
- deap.algorithms, berperan dalam menyediakan penerapan algoritma evolusi seperti algoritma genetika, strategi evolusi, dan pemrograman genetik. Pada penelitian ini, algoritma diterapkan menggunakan fungsi algorithms,eaMuPlusLambda yang bekerja dengan mengambil populasi dan mengembangkannya dengan menggunakan fungsi varOr (*crossover* dan mutasi) dan memastikan seleksi untuk generasi berikutnya bisa berasal dari gabungan orang tua dan keturunan. Parameter utamanya adalah Mu (ukuran populasi), lambda (jumlah keturunan), cxpb (probabilitas *crossover*), dan mutpb (probabilitas mutasi).

4. Pymoo (pymoo.indicators,hv.HV), merupakan pustaka Python untuk optimasi multi-objektif (MOO). Pustaka ini menyediakan berbagai algoritma, indikator, dan alat untuk menyelesaikan masalah MOO. Modul pymoo.indicators.hv berisi kelas HV, yang mengimplementasikan indikator *hypervolume*. *Hypervolume* adalah indikator kinerja untuk mengukur kualitas set yang digunakan pada optimasi multi objektif

dengan mengukur volume ruang yang didominasi oleh set titik dalam ruang objektif, dibatasi oleh titik referensi tertentu (Guerreiro *et al*., 2021).

B. Parameter yang digunakan:

1. Mu (mu=148), menunjukkan jumlah individu yang dipilih dari populasi saat ini untuk menjadi orang tua generasi berikutnya atau dikenal sebagai ukuran populasi induk. Jumlah mu yang besar meningkatkan variasi genetik, yang membantu eksplorasi ruang solusi secara lebih efektif. Hal ini dapat mengurangi risiko terjebak dalam optimum lokal dan meningkatkan kemampuan algoritma untuk menemukan solusi yang lebih baik.
2. Lambda (lambda=150), menentukan jumlah individu baru yang akan dihasilkan dari orang tua yang dipilih melalui *crossover* dan mutasi dan dikenal sebagai ukuran populasi keturunan. Nilai lambda yang ditentukan lebih besar dari mu agar mencapai kondisi tekanan seleksi lebih tinggi dimana hanya individu terbaik yang akan bertahan dan diteruskan ke generasi berikutnya sehingga mempercepat konvergensi.
3. Probabilitas *Crossover* (cxpb=0.7), menentukan seberapa sering *crossover* atau persilangan akan dilakukan pada pasangan individu dari populasi. Nilai cxpb yang tinggi (0.7) meningkatkan peluang pertukaran informasi genetik antara dua orang tua, yang dapat menghasilkan individu keturunan yang memiliki kombinasi karakteristik yang lebih baik dan mempercepat eksplorasi solusi baru
4. Probabilitas Mutasi (mutpb=0.3), menentukan seberapa sering mutasi akan dilakukan pada individu dalam populasi. Nilai mutpb yang moderat (0.3) membantu dalam menjaga keragaman genetik dalam populasi, yang penting untuk menghindari prematur konvergensi dan memastikan bahwa algoritma dapat mengeksplorasi berbagai area ruang solusi.
5. Jumlah Generasi (ngen=1), menunjukkan jumlah generasi yang akan dievolusikan oleh algoritma. Jumlah generasi menentukan jumlah iterasi bagi algoritma untuk menyempurnakan solusi. Ini meningkatkan peluang menemukan solusi yang mendekati optimal.
6. *Hall of Fame* (hof=1), menentukan jumlah individu terbaik yang akan disimpan dalam "*Hall of Fame*" selama evolusi. *Hall of Fame* adalah koleksi individu terbaik yang pernah ditemukan oleh algoritma. Menyimpan hof=1 memastikan bahwa individu terbaik selalu dilacak dan dapat digunakan untuk evaluasi atau analisis lebih lanjut. Ini

membantu memastikan bahwa solusi optimal yang pernah ditemukan tidak hilang selama proses evolusi

7. *Verbose* (verbose=true), mengontrol apakah detail proses evolusi akan dicetak ke konsol selama eksekusi. Mengaktifkan verbose=True memberikan visibilitas ke dalam proses evolusi, memungkinkan pemantauan kemajuan, diagnosis masalah, dan penyesuaian parameter yang mungkin diperlukan selama eksperimen
8. *Reference Point* (ref_point=[1e6, 1e6]), digunakan untuk menghitung *hypervolume*, yang merupakan metrik kinerja yang mengevaluasi kualitas set *Pareto*. *Reference point* harus lebih buruk dari semua solusi dalam set *Pareto*. Memilih nilai besar untuk ref_point (seperti [1e6, 1e6]) memastikan bahwa semua solusi yang ditemukan akan berada dalam ruang yang dihitung oleh *hypervolume*, memungkinkan evaluasi yang tepat dari kemajuan evolusi multi-objektif.

4.2.3 Penjadwalan dengan Algoritma NSGA-II

Penjadwalan menggunakan algoritma NSGA-II menghasilkan keluaran berupa urutan pekerjaan yang perlu dilakukan untuk memastikan nilai minimal pada *makespan* dan biaya keterlambatan. Berikut ini merupakan alur yang dilakukan pada *script python* yang disusun beserta rangkuman keluaran akhirnya,

1. Inisialisasi Populasi

Tahapan ini membuat populasi awal yang terdiri dari berbagai susunan penjadwalan dan diimplementasikan dalam *script python* berikut:

```
# Attribute generator
toolbox.register("attr_job", np.random.permutation, num_jobs)

# Structure initializers
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_job, num_machines)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

pop = toolbox.population(n=300)
```

Gambar 4.1 *Script* Inisialisasi Populasi NSGA-II

Keterangan:

a. Masukan
   - num_jobs: Jumlah pekerjaan yang akan dijadwalkan (9 pekerjaan)
   - num_machines: Jumlah mesin yang tersedia (6 mesin)

b. Proses

- Sebuah 'toolbox' dari library DEAP digunakan untuk mengelola operator-operator genetik.
- toolbox.register("attr_job", np.random.permutation, num_jobs) : Mendaftarkan fungsi untuk menghasilkan urutan acak dari pekerjaan (attr_job).
- toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_job, num_machines) : Membuat individu dengan mengulang attr_job sebanyak num_machines kali. Setiap individu mewakili sebuah solusi potensial, yaitu urutan pekerjaan pada setiap mesin.
- toolbox.register("population", tools.initRepeat, list, toolbox.individual): Membuat populasi awal dengan menciptakan beberapa individu.

c. Keluaran

- pop: Populasi awal yang terdiri dari individu-individu yang masing-masing berisi urutan pekerjaan pada setiap mesin.

2. Evaluasi Populasi

Tahapan ini menentukan nilai *fitness* untuk seluruh populasi awal dengan menerapkan persamaan matematis dari obyektif berikut ini:

$$C_{max} = \max_{j,m} C_{j,m} \quad (7)$$

$$TLC = \sum_{j=1}^{n} \sum_{m=1}^{m} \left(L_{j,m} \cdot c_{j,m}\right) \quad (8)$$

$$L_{j,m} = max(0, C_{j,m} - pC_{j,m}) \quad (9)$$

$$pC_{j,m} = ps_{j,m} + d_{j,m} \quad (10)$$

$$ps_{j,m} = \begin{cases} 0, \text{jika m} = 1 \text{ dan j} = 1 \\ s_{j,1}, \text{jika m} = 1 \text{ dan j} \neq 1 \\ C_{j,m-1}, \text{jika m} \neq 1 \end{cases} \quad (11)$$

Dimana,

- Cmax = *Makespan* yang merupakan fungsi maksimum dari waktu penyelesaian semua pekerjaan pada semua mesin
- TLC = Total *Lateness Cost* yang merupakan total biaya keterlambatan individual dari perkalian antara jumlah waktu keterlambatan (Ljm) terhadap biaya per waktu keterlambatan (cjm)
- Ljm = Waktu keterlambatan yang merupakan selisih antara waktu penyelesaian nyata terhadap waktu penyelesaian yang direncanakan
- pCjm = Waktu penyelesaian yang direncanakan merupakan penjumlahan antara waktu mulai yang direncanakan dan *due date*
- psjm = Waktu mulai yang direncanakan merupakan waktu suatu pekerjaan dimulai berdasarkan kondisi tertentu, dimana:
    - Apabila ia merupakan pekerjaan pertama yang dimulai pada mesin pertama maka waktu mulai yang direncanakan adalah 0.
    - Apabila ia bukan merupakan pekerjaan pertama yang dimulai pada mesin pertama maka waktu mulai yang direncanakan sama dengan waktu mulai yang ditentukan.
    - Apabila ia bukan merupakan pekerjaan pada mesin pertama maka waktu mulai yang direncanakan adalah waktu penyelesaian pekerjaan tersebut pada mesin sebelumnya.

Persamaan matematis tersebut kemudian diimplementasikan dalam *script python* berikut:

```
# Evaluation function
def evaluate(individual):
    schedule = np.zeros((num_jobs, num_machines))
    completion_times = np.zeros((num_jobs, num_machines))

    for machine_idx in range(num_machines):
        for job_idx in range(num_jobs):
            job = individual[machine_idx][job_idx]
            if job_idx == 0:
                start_time = 0
            else:
                previous_job = individual[machine_idx][job_idx - 1]
                start_time = completion_times[previous_job, machine_idx]

            if machine_idx > 0:
                prev_machine_end_time = completion_times[job, machine_idx - 1]
                start_time = max(start_time, prev_machine_end_time)

            end_time = start_time + processing_times.iloc[job, machine_idx]
            schedule[job, machine_idx] = start_time
            completion_times[job, machine_idx] = end_time

    # Total lateness cost
    total_lateness_cost = 0
    for job in range(num_jobs):
        for machine_idx in range(num_machines):
            if job_idx == 0:  # First job on any machine
                planned_start_time = 0
            elif machine_idx == 0:
                planned_start_time = schedule[job, machine_idx]
            else:
                planned_start_time = completion_times[job, machine_idx - 1]
            planned_completion_time = planned_start_time + due_dates.iloc[job, machine_idx]
            lateness = max(0, completion_times[job, machine_idx] - planned_completion_time)
            total_lateness_cost += lateness* lateness_costs.iloc[job, machine_idx]

    # Makespan is the maximum completion time
    makespan = np.max(completion_times)

    return makespan, total_lateness_cost
```

Gambar 4.2 *Script* Evaluasi Populasi NSGA-II

Keterangan:

a. Masukan
   - Individual : Satu individu dari populasi yang akan dievaluasi.
   - processing_times : *DataFrame* berisi waktu pemrosesan setiap pekerjaan pada setiap mesin.
   - due_dates : *DataFrame* berisi *due date* setiap pekerjaan pada setiap mesin.
   - lateness_costs : *DataFrame* berisi biaya keterlambatan per unit waktu untuk setiap pekerjaan pada setiap mesin.

b. Proses

- Inisialisasi schedule dan completion_times untuk menyimpan waktu mulai dan selesai setiap pekerjaan pada setiap mesin.
- Iterasi pada setiap mesin dan setiap pekerjaan pada mesin tersebut.
- Menghitung start_time dan end_time untuk setiap pekerjaan berdasarkan waktu penyelesaian pekerjaan sebelumnya pada mesin yang sama dan waktu penyelesaian pekerjaan tersebut pada mesin sebelumnya (jika ada).
- Mengisi schedule dan completion_times.
- Menghitung total_lateness_cost dengan menjumlahkan biaya keterlambatan setiap pekerjaan pada setiap mesin.
    - Keterlambatan (*lateness*) dihitung sebagai selisih antara completion_times dan planned_completion_time (due date + waktu mulai terjadwal), dengan nilai minimum 0.
    - total_lateness_cost dihitung dengan mengalikan *lateness* dengan lateness_costs untuk setiap pekerjaan dan mesin, lalu menjumlahkan semua biaya tersebut.
- Menghitung *makespan* sebagai waktu penyelesaian maksimum dari semua pekerjaan.

c. Keluaran

- makespan, total_lateness_cost: Nilai *fitness* individu, yang akan digunakan dalam proses seleksi..

3. Fast Non-Dominated Sorting

Tahapan ini mengelompokkan individu berdasarkan tingkat dominasi *Pareto* dan diimplementasikan pada *script python* berikut:

```
# Sort the population into Pareto fronts
pareto_fronts = tools.sortNondominated(pop, len(pop), first_front_only=False)
```

Gambar 4.3 *Script Fast Non-dominated Sorting* NSGA-II

Keterangan:

a. Masukan

- pop: Populasi setelah evaluasi *fitness*

b. Proses

- tools.sortNondominated(pop, len(pop), first_front_only=False) : Melakukan pengurutan individu-individu dalam populasi berdasarkan konsep dominasi *Pareto*.
- Individu-individu yang tidak didominasi oleh individu lain ditempatkan pada front *Pareto* pertama.
- Individu-individu yang hanya didominasi oleh individu-individu di front pertama ditempatkan pada front kedua, dan seterusnya.

c. Keluaran

- pareto_fronts : *List* dari *front-front Pareto*, dengan setiap front berisi individu-individu yang tidak saling mendominasi

4. Crowding Distance Assignment

Tahapan ini mengukur kepadatan area di sekitar individu dalam ruang objektif untuk menentukan *crowding distance* dan diimplementasikan pada bagian *script python* berikut:

```
toolbox.register("select", tools.selNSGA2)
```

Gambar 4.4 Script *Crowding Distance Assignment* NSGA-II

Keterangan:

a. Masukan

- pareto_fronts: *List* dari *front-front Pareto* hasil dari *fast non-dominated sorting*

b. Proses

- tools.selNSGA2: Secara otomatis menghitung *crowding distance* untuk setiap individu dalam setiap *front Pareto*.
- *Crowding distance* mengukur seberapa padat daerah sekitar suatu individu dalam ruang objektif.
- Individu yang berada di daerah yang lebih jarang memiliki *crowding distance* yang lebih besar.

c. Keluaran

- Tidak ada keluaran eksplisit, tetapi nilai *crowding distance* disimpan dalam atribut individu

5. Crowded Tournament Selection

Tahapan ini memilih individu berdasarkan peringkat dominasi *Pareto* dan *crowding distance* dan diimplementasikan pada bagian *script python* berikut:

```
pop, logbook = algorithms.eaMuPlusLambda(pop, toolbox, mu=148, lambda_=150, cxpb=0.7, mutpb=0.3,
                                         ngen=1, stats=stats, halloffame=hof, verbose=True)
```

Gambar 4.5 *Script Crowded Tournamnet Selection* NSGA-II

Keterangan:

a. Masukan
   - Pop : Populasi setelah evaluasi *fitness* dan *crowding distance assignment*

b. Proses
   - algorithms.eaMuPlusLambda : Menggunakan tools.selNSGA2 untuk memilih individu-individu yang akan menjadi orang tua pada generasi berikutnya.
   - Seleksi didasarkan pada peringkat *front Pareto* dan *crowding distance*.
   - Individu pada front yang lebih rendah (lebih baik) akan dipilih terlebih dahulu
   - Jika dua individu berada pada *front* yang sama, individu dengan *crowding distance* yang lebih besar (daerah lebih jarang) akan dipilih.

c. Keluaran
   - Sekumpulan individu terpilih yang akan digunakan untuk menghasilkan *offspring* melalui *crossover* dan mutasi

6. Crossover

Tahapan ini menggabungkan dua induk untuk menghasilkan kromosom baru dan diimplementasikan pada bagian *script python* berikut:

```
toolbox.register("mate", tools.cxUniform, indpb=0.7)
```

Gambar 4.6 *Script Crossover* NSGA-II

Keterangan:

a. Masukan
   - Individu-individu terpilih dari proses seleksi.

b. Proses
   - tools.cxUniform, indpb=0.7 : Melakukan *crossover* antara dua individu terpilih.
   - indpb=0.7 menunjukkan probabilitas setiap gen untuk mengalami *crossover*
   - *Crossover* menukar sebagian gen antara dua individu untuk menghasilkan *offspring* baru yang mewarisi karakteristik dari kedua orang tuanya.

c. Keluaran
   - *Offspring* baru hasil *crossover*

7. Mutasi

Tahapan ini mengubah gen-gen dalam kromosom untuk menjaga keragaman dan diimplementasikan pada bagian *script python* berikut:

```
toolbox.register("mutate", tools.mutShuffleIndexes, indpb=0.3)
```

Gambar 4.7 *Script* Mutasi NSGA-II

Keterangan:

a. Masukan
   - *Offspring* baru hasil *crossover*

b. Proses
   - tools.mutShuffleIndexes, indpb=0.3 : Melakukan mutasi pada *offspring*.
   - indpb=0.3 menunjukkan probabilitas setiap gen untuk mengalami mutasi.
   - Mutasi mengubah sebagian gen pada *offspring* secara acak untuk menambah variasi genetik dalam populasi dan mencegah konvergensi prematur.

c. Keluaran
   - *Offspring* yang telah mengalami mutasi

8. Uji *hypervolume*

Tahapan ini mengevaluasi performa populasi pada setiap generasi guna memberikan informasi tentang kualitas dan distribusi solusi pada front *Pareto*. Cara yang digunakan adalah menghitung ukuran wilayah yang didominasi oleh set titik dan dibatasi oleh titik referensi sehingga diketahui kedekatan set titik dengan *pareto front*, keragaman, dan penyebaran titik-titik tersebut. Berikut adalah implementasi *hypervolume* pada *script python*:

```
from pymoo.indicators.hv import HV

def calculate_hypervolume(pop, ref_point):
    fitnesses = np.array([ind.fitness.values for ind in pop])
    hv = HV(ref_point=ref_point)
    return hv(fitnesses)
```

Gambar 4.8 *Script* Kalkulasi *Hypervolume* NSGA-II

Keterangan:

a. Masukan
   - Pop : Populasi pada setiap generasi.
   - ref_point : Titik referensi untuk perhitungan *hypervolume*. Biasanya, titik ini dipilih sedemikian rupa sehingga mendominasi semua titik dalam populasi (1e6)

b. Proses

- Fitnesses : Mengambil semua nilai *fitness* dari setiap individu dalam populasi pop.
- hv = HV(ref_point=ref_point): Membuat objek *hypervolume* dari library pymoo dengan titik referensi yang ditentukan.
- hv(fitnesses) : Menghitung *hypervolume* dari populasi berdasarkan nilai *fitness* dan titik referensi.

c. Keluaran

- hypervol: Nilai *hypervolume* dari populasi saat ini, yang menunjukkan seberapa baik populasi mencakup ruang objektif.

Setelah dilakukan serangkaian tahapan tersebut melalui implementasi pada *script python*, berikut ini merupakan keluaran akhir dari proses penjadwalan yang dianggap sebagai individu yang memiliki nilai *fitness* paling optimal untuk kedua obyektif.

Tabel 4.5 Translasi Keluaran *Script*

Jenis	**Keluaran *Script***	**Translasi ke Studi Kasus**
Urutan	[3, 2, 6, 7, 8, 0, 4, 5, 1]	K5, K4, K8, K9, K10, K1, K6, K7, K3
Besar *Fitness*	55, 682	*Makespan* = 55 hari Biaya Keterlambatan = Rp68,200,000.00

Pada Tabel 4.5, terdapat 4 (empat) jenis urutan pekerjaan yang menghasilkan nilai *fitness* yang dianggap terbaik yaitu *makespan* 55 hari dengan total biaya keterlambatan Rp68,200,000.00. Besar *fitness* tersebut telah dianggap sebagai nilai paling optimal dari 150 generasi yang dihasilkan dengan jumlah populasi per generasi adalah 148 individu. Berikut ini merupakan data nilai *fitness* dari generasi ke generasi.

Tabel 4.6 Nilai *Fitness* Seluruh Generasi NSGA-II

	Makespan			**Biaya Keterlambatan**		
Gen	***Min***	***Max***	***Average***	***Min***	***Max***	***Average***
0	104	142	129	2367	4074	3614
1	104	137	124	2367	3871	3452
2	99	134	121	2367	3753	3319
3	99	129	119	2367	3513	3230
4	99	126	117	2367	3533	3159
5	98	126	114	2367	3482	3076
6	98	121	112	2367	3341	3013
7	98	119	111	2367	3286	2967
8	98	117	110	2367	3201	2909
9	98	117	109	2367	3137	2871

	Makespan			**Biaya Keterlambatan**		
Gen	***Min***	***Max***	***Average***	***Min***	***Max***	***Average***
10	95	117	108	2292	3048	2806
11	95	117	107	2292	3047	2778
12	94	114	105	2120	2991	2713
13	94	112	104	2120	2989	2670
14	84	111	102	1965	2989	2602
15	84	109	101	1910	2837	2553
16	84	109	100	1910	2837	2509
17	84	107	99	1910	2837	2478
18	81	106	98	1775	2742	2439
19	81	105	98	1775	2777	2370
20	81	104	97	1775	2541	2319
21	79	104	96	1605	2541	2269
22	79	103	95	1605	2451	2230
23	79	106	93	1605	2451	2179
24	79	106	93	1605	2451	2133
25	74	106	91	1423	2329	2075
26	74	103	90	1423	2329	2020
27	74	103	89	1423	2329	1977
28	74	96	88	1423	2229	1915
29	74	96	88	1423	2087	1864
30	74	96	87	1423	2083	1848
31	74	96	86	1423	2091	1818
32	74	96	85	1423	2091	1767
33	73	96	83	1423	2083	1714
34	70	91	81	1309	1896	1627
35	67	87	78	1024	1792	1551
36	67	86	76	1024	1734	1490
37	66	83	75	1024	1597	1456
38	65	79	73	1024	1550	1399
39	61	79	72	1024	1550	1364
40	61	79	71	1024	1548	1315
41	61	79	68	1024	1484	1266
42	60	75	68	1024	1460	1193
43	60	70	66	1024	1314	1157
44	60	70	66	1024	1314	1076
45	60	70	67	1024	1126	1030
46	60	70	70	1024	1126	1026
47	60	70	70	1024	1126	1026
48	60	70	70	1024	1126	1026
49	60	70	70	1024	1126	1026
50	60	70	70	1024	1126	1026
51	60	70	70	1024	1126	1026
52	60	70	70	1024	1126	1026
53	60	70	70	1024	1126	1026
54	60	70	70	1024	1126	1026
55	60	70	70	1024	1126	1026

	Makespan			**Biaya Keterlambatan**		
Gen	***Min***	***Max***	***Average***	***Min***	***Max***	***Average***
56	60	70	70	1024	1126	1026
57	60	70	70	1024	1126	1026
58	60	70	70	1024	1126	1026
59	60	70	70	1024	1126	1026
60	60	70	70	1024	1126	1026
61	60	70	70	1024	1126	1026
62	60	70	70	1024	1126	1026
63	60	70	70	1024	1126	1026
64	60	70	70	1024	1126	1026
65	60	70	70	1024	1126	1026
66	60	70	70	1024	1126	1026
67	60	70	70	1024	1126	1026
68	60	70	70	1024	1126	1026
69	60	70	70	1024	1126	1026
70	60	70	70	1024	1126	1026
71	60	70	70	1024	1126	1026
72	60	70	70	1024	1126	1026
73	60	70	70	1024	1126	1026
74	60	70	70	1024	1126	1026
75	60	70	70	1024	1126	1026
76	60	70	70	1024	1126	1026
77	60	70	70	1006	1126	1025
78	60	70	69	1006	1126	1026
79	55	70	69	682	1126	1025
80	55	70	67	682	1126	1019
81	55	70	65	682	1126	1014
82	55	70	61	682	1126	1003
83	55	64	60	682	1126	973
84	55	64	58	682	1126	900
85	55	64	56	682	1126	724
86	55	55	55	682	682	682
87	55	55	55	682	682	682
88	55	55	55	682	682	682
89	55	55	55	682	682	682
90	55	55	55	682	682	682
91	55	55	55	682	682	682
92	55	55	55	682	682	682
93	55	55	55	682	682	682
94	55	55	55	682	682	682
95	55	55	55	682	682	682
96	55	55	55	682	682	682
97	55	55	55	682	682	682
98	55	55	55	682	682	682
99	55	55	55	682	682	682
100	55	55	55	682	682	682
101	55	55	55	682	682	682

	Makespan			**Biaya Keterlambatan**		
Gen	***Min***	***Max***	***Average***	***Min***	***Max***	***Average***
102	55	55	55	682	682	682
103	55	55	55	682	682	682
104	55	55	55	682	682	682
105	55	55	55	682	682	682
106	55	55	55	682	682	682
107	55	55	55	682	682	682
108	55	55	55	682	682	682
109	55	55	55	682	682	682
110	55	55	55	682	682	682
111	55	55	55	682	682	682
112	55	55	55	682	682	682
113	55	55	55	682	682	682
114	55	55	55	682	682	682
115	55	55	55	682	682	682
116	55	55	55	682	682	682
117	55	55	55	682	682	682
118	55	55	55	682	682	682
119	55	55	55	682	682	682
120	55	55	55	682	682	682
121	55	55	55	682	682	682
122	55	55	55	682	682	682
123	55	55	55	682	682	682
124	55	55	55	682	682	682
125	55	55	55	682	682	682
126	55	55	55	682	682	682
127	55	55	55	682	682	682
128	55	55	55	682	682	682
129	55	55	55	682	682	682
130	55	55	55	682	682	682
131	55	55	55	682	682	682
132	55	55	55	682	682	682
133	55	55	55	682	682	682
134	55	55	55	682	682	682
135	55	55	55	682	682	682
136	55	55	55	682	682	682
137	55	55	55	682	682	682
138	55	55	55	682	682	682
139	55	55	55	682	682	682
140	55	55	55	682	682	682
141	55	55	55	682	682	682
142	55	55	55	682	682	682
143	55	55	55	682	682	682
144	55	55	55	682	682	682
145	55	55	55	682	682	682
146	55	55	55	682	682	682
147	55	55	55	682	682	682

	Makespan			Biaya Keterlambatan		
Gen	***Min***	***Max***	***Average***	***Min***	***Max***	***Average***
148	55	55	55	682	682	682
149	55	55	55	682	682	682

Selain menghasilkan urutan penjadwalan, *script* juga dirancang untuk menghasilkan angka uji *hypervolume* dengan besar *reference point*-nya adalah 1e6. Berikut ini merupakan keluaran dari uji *hypervolume* pada setiap generasinya.

Tabel 4.7 Nilai *Hypervolume* NSGA-II

Generasi	***Hypervolume***	**Generasi**	***Hypervolume***
0	997529246168	75	998916061212
1	997529246168	76	998916061212
2	997534234253	77	998934060144
3	997534234253	78	998934060144
4	997534234253	79	999263037510
5	997535231511	80	999263037510
6	997535231511	81	999263037510
7	997535231511	82	999263037510
8	997535231776	83	999263037510
9	997535231776	84	999263037510
10	997613217740	85	999263037510
11	997613217740	86	999263037510
12	997786199280	87	999263037510
13	997786199280	88	999263037510
14	997951165060	89	999263037510
15	998006160165	90	999263037510
16	998006160165	91	999263037510
17	998006160165	92	999263037510
18	998144143775	93	999263037510
19	998144143775	94	999263037510
20	998144143775	95	999263037510
21	998316126795	96	999263037510
22	998316126795	97	999263037510
23	998316126795	98	999263037510
24	998316126795	99	999263037510
25	998503105302	100	999263037510
26	998503105302	101	999263037510
27	998503105302	102	999263037510
28	998503105302	103	999263037510
29	998503105302	104	999263037510
30	998503105302	105	999263037510
31	998503105302	106	999263037510
32	998503105302	107	999263037510
33	998504103763	108	999263037510
34	998621091543	109	999263037510

Generasi	*Hypervolume*	Generasi	*Hypervolume*
35	998909067921	110	999263037510
36	998909068025	111	999263037510
37	998910066565	112	999263037510
38	998911065397	113	999263037510
39	998915062338	114	999263037510
40	998915062338	115	999263037510
41	998915062338	116	999263037510
42	998916061212	117	999263037510
43	998916061212	118	999263037510
44	998916061212	119	999263037510
45	998916061212	120	999263037510
46	998916061212	121	999263037510
47	998916061212	122	999263037510
48	998916061212	123	999263037510
49	998916061212	124	999263037510
50	998916061212	125	999263037510
51	998916061212	126	999263037510
52	998916061212	127	999263037510
53	998916061212	128	999263037510
54	998916061212	129	999263037510
55	998916061212	130	999263037510
56	998916061212	131	999263037510
57	998916061212	132	999263037510
58	998916061212	133	999263037510
59	998916061212	134	999263037510
60	998916061212	135	999263037510
61	998916061212	136	999263037510
62	998916061212	137	999263037510
63	998916061212	138	999263037510
64	998916061212	139	999263037510
65	998916061212	140	999263037510
66	998916061212	141	999263037510
67	998916061212	142	999263037510
68	998916061212	143	999263037510
69	998916061212	144	999263037510
70	998916061212	145	999263037510
71	998916061212	146	999263037510
72	998916061212	147	999263037510
73	998916061212	148	999263037510
74	998916061212	149	999263037510

Nilai *hypervolume* setiap generasi tersebut mengalami perubahan pada beberapa generasi. Perubahan tersebut mengindikasikan performa dari algoritma dalam membangun ruang pencarian, dimana semakin besar nilai *hypervolume* menunjukkan semakin besar ruang pencarian yang dijajaki oleh algoritma tersebut. Oleh karena itu, semakin besar nilai *hypervolume* meningkatkan keragaman individu calon solusi optimal.

BAB V

PEMBAHASAN

5.1 Analisis Performa Algoritma NSGA-II

Penerapan algoritma NSGA-II menggunakan *script* python pada penelitian menghasilkan sejumlah keluaran yang dapat digunakan untuk mengukur keefektifan algoritma pada studi kasus. Berikut ini merupakan beberapa keluaran dari algoritma NSGA-II tersebut.

5.1.1 Persebaran *Pareto Front*

Populasi diurutkan berdasarkan dominasi *pareto* dimana solusi yang tidak didominasi oleh solusi lain diberi peringkat lebih baik. Berikut ini merupakan grafik untuk salah satu generasi yang mulai memunculkan nilai *fitness* yang saat ini merupakan nilai *fitness* paling optimal.

Tabel 5.1 Persebaran *Pareto* Generasi 80

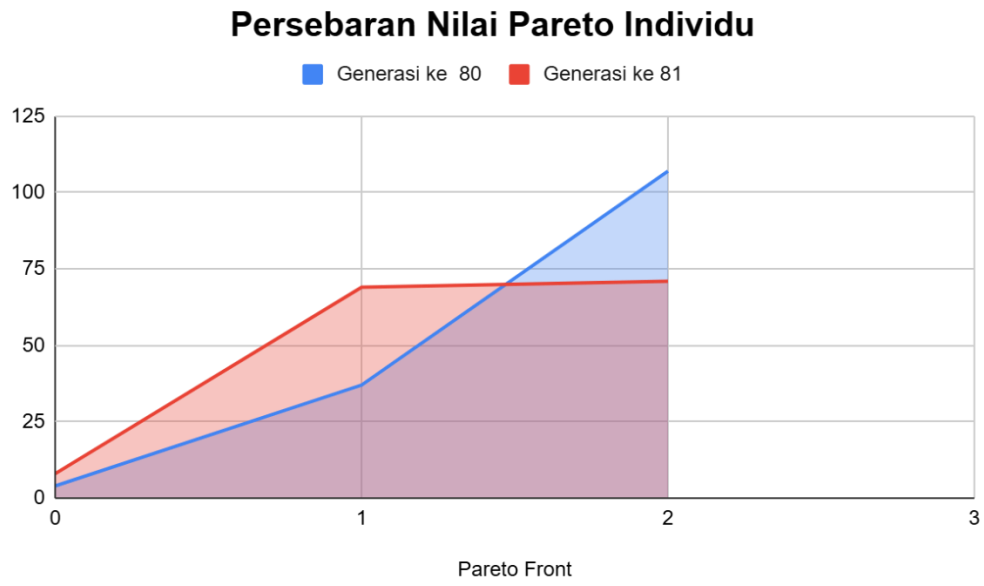
Pareto Front	**Jumlah**	**Biaya Keterlambatan**	***Makespan***
0	4	682	55
1	37	1126	60
2	107	1024	70

Berdasarkan Tabel 5.1, individu dengan nilai *fitness* terbaik (55, 682) tidak didominasi oleh individu solusi lain berdasarkan fungsi obyektif Minimalisasi. Oleh karena itu, individu tersebut diberikan peringkat lebih baik dibandingkan individu lain pada generasi tersebut. Selain itu, individu tersebut merupakan satu-satunya individu yang berada pada daerah solusi tersebut, yang menunjukkan nilai *crowding distance* terhadap solusi lainnya semakin besar. Oleh karena itu, individu tersebut merupakan yang paling memungkinkan untuk dikawinkan untuk mendapat solusi keturunan berikutnya. Hal tersebut tampak dari

individu pada generasi berikutnya yang memiliki kecenderungan semakin dekat dengan nilai *fitness* individu dengan *front* terkecil pada generasi sebelumnya. Berikut ini merupakan persebaran *pareto front* pada generasi 81.

Tabel 5.2 Persebaran *Pareto* Generasi 81

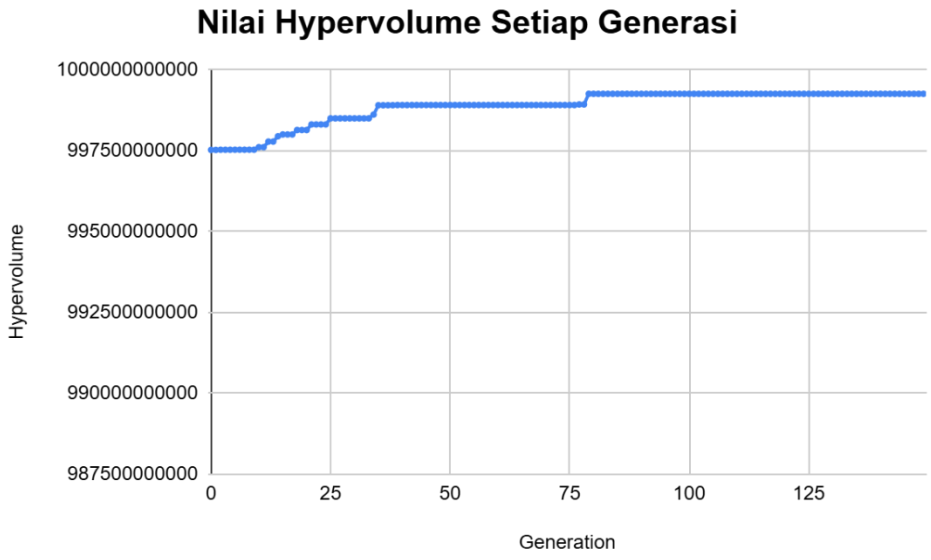
Pareto Front	**Jumlah**	**Biaya Keterlambatan**	***Makespan***
0	8	682	55
1	69	1126	60
2	71	1024	70



Gambar 5.1 Perbandingan Persebaran Nilai *Pareto* Antar Generasi

5.1.2 Analisis *Hypervolume*

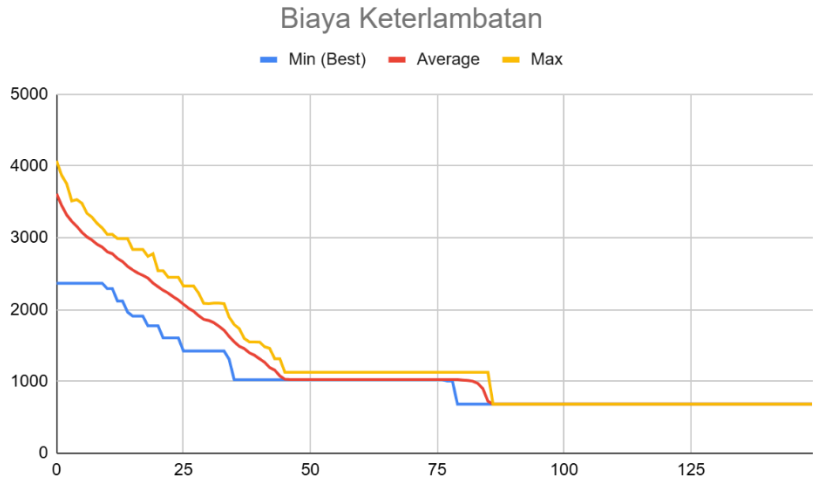
Hypervolume adalah indikator kinerja untuk MOO yang mengukur volume wilayah yang didominasi di ruang objektif yang ditempati oleh sekumpulan solusi (Guerreiro *et al*., 2021). Semakin besar nilai *hypervolume* untuk setiap generasi menunjukkan semakin besar ruang pencarian yang diambil oleh algoritma sehingga menjaga keragaman solusi seiring pengerucutan nilai *fitness* yang semakin dekat ke titik optimal. Berikut ini merupakan grafik untuk melihat pola tren nilai *hypervolume* pada setiap generasi.



Gambar 5.2 Tren Nilai *Hypervolume* Seluruh Generasi

Peningkatan nilai *hypervolume* mengindikasikan performa dari algoritma dalam membangun ruang pencarian, dimana semakin besar nilai *hypervolume* menunjukkan semakin besar ruang pencarian yang dijajaki oleh algoritma tersebut. Oleh karena itu, semakin besar nilai *hypervolume* meningkatkan keragaman individu calon solusi optimal. Selain itu, potensi konvergensi dini dapat diperhatikan juga pada persebaran rentang nilai setiap generasi untuk setiap obyektif. Apabila nilai antara obyektif individu terbaik (nilai minimal) terhadap rata-rata nilai obyektif dalam generasi tersebut tidak sama maka dapat dipastikan ruang yang dijelajahi tetap besar dan terhindar dari konvergensi dini. Berikut merupakan visualisasi nilai obyektif seluruh generasi.

Gambar 5.3 Statistik Nilai Obyektif *Makespan* Seluruh Generasi

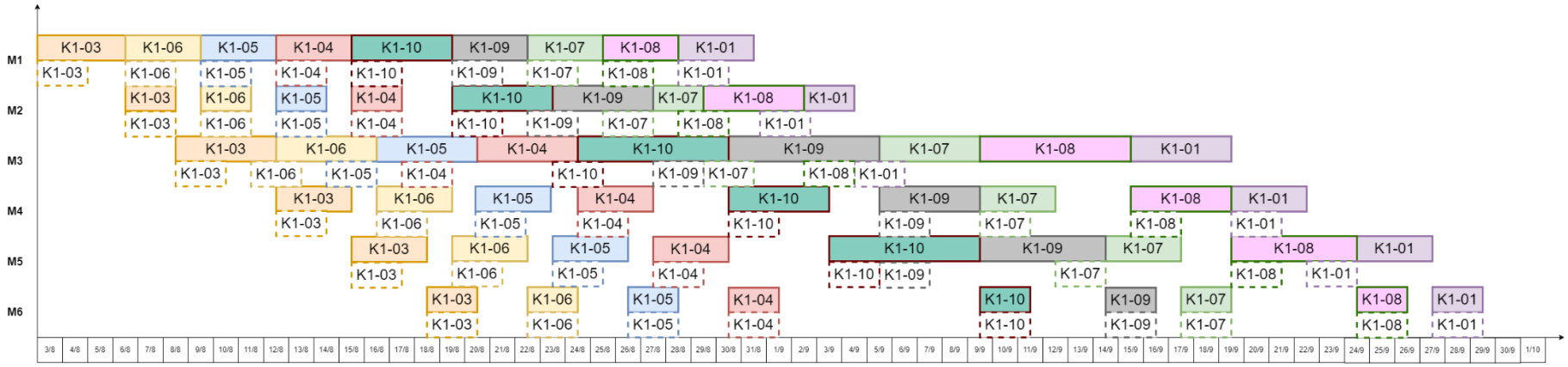


Gambar 5.4 Statistik Nilai Obyektif Biaya Keterlambatan Seluruh Generasi

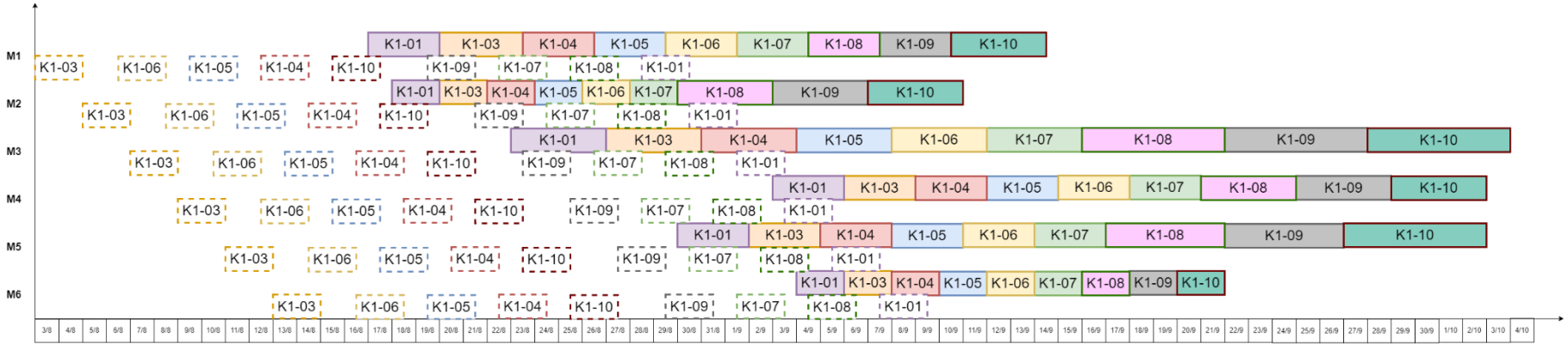
Pada Gambar 5.3 dan 5.4 ditampilkan bahwa pada generasi ke 87 dari 150 generasi terlihat bahwa solusi yang dimunculkan mulai memiliki kesamaan nilai untuk minimal, rata-rata, dan maksimalnya yang menunjukkan solusi tidak menghasilkan individu baru yang memiliki nilai obyektif berbeda. Fenomena tersebut disebut konvergensi, dimana algoritma telah menemukan solusi paling optimalnya. Namun, apakah konvergensi yang dialami terhitung prematur atau bukan dapat ditentukan dari generasi mulainya konvergensi. Pada penelitian ini, generasi tercapainya konvergensi adalah generasi ke-87 dari 150 atau 42% dari generasinya memiliki nilai optimal yang sama. Jumlah generasi yang bukan termasuk dalam konvergensi dini terhitung lebih dari setengah generasi atau 58% menunjukkan bahwa konvergensi dini tidak terjadi.

5.2 Analisis Penjadwalan oleh NSGA-II

Analisis skenario penjadwalan oleh NSGA-II dilakukan dengan cara memvisualisasi skenario sebagai *gantt chart*. Interpretasi yang diinginkan adalah kesesuaian urutan penjadwalan terhadap total *makespan* dan biaya keterlambatan yang ditemukan dan dianggap sebagai solusi optimal. Berikut ini merupakan rangkuman *plotting* waktu penjadwalan yang terpilih dan yang sedang berlaku menggunakan visualisasi *gantt chart*.



Gambar 5.5 *Gantt Chart* Penjadwalan Usulan



Gambar 5.6 *Gantt Chart* Penjadwalan Yang Berlaku

Tabel 5.3 Rangkuman *Timeline* Penjadwalan Usulan

Job	**Mesin**	**Mulai**	**Selesai**	***Job***	**Mesin**	**Mulai**	**Selesai**	***Job***	**Mesin**	**Mulai**	**Selesai**
K1	M1	19/08/23	22/08/23	K1	M3	03/09/23	07/09/23	K1	M5	13/09/23	16/09/23
K3	M1	28/08/23	31/08/23	K3	M3	15/09/23	19/09/23	K3	M5	22/09/23	25/09/23
K4	M1	06/08/23	09/08/23	K4	M3	12/08/23	16/08/23	K4	M5	19/08/23	22/08/23
K5	M1	03/08/23	06/08/23	K5	M3	08/08/23	12/08/23	K5	M5	15/08/23	18/08/23
K6	M1	22/08/23	25/08/23	K6	M3	07/09/23	11/09/23	K6	M5	16/09/23	19/09/23
K7	M1	25/08/23	28/08/23	K7	M3	11/09/23	15/09/23	K7	M5	19/09/23	22/09/23
K8	M1	09/08/23	12/08/23	K8	M3	16/08/23	22/08/23	K8	M5	26/08/23	31/08/23
K9	M1	12/08/23	15/08/23	K9	M3	22/08/23	28/08/23	K9	M5	01/09/23	06/09/23
K10	M1	15/08/23	19/08/23	K10	M3	28/08/23	03/09/23	K10	M5	07/09/23	13/09/23
K1	M2	24/08/23	26/08/23	K1	M4	07/09/23	10/09/23	K1	M6	16/09/23	18/09/23
K3	M2	31/08/23	02/09/23	K3	M4	19/09/23	22/09/23	K3	M6	25/09/23	27/09/23
K4	M2	09/08/23	11/08/23	K4	M4	16/08/23	19/08/23	K4	M6	22/08/23	24/08/23
K5	M2	06/08/23	08/08/23	K5	M4	12/08/23	15/08/23	K5	M6	18/08/23	20/08/23
K6	M2	26/08/23	28/08/23	K6	M4	11/09/23	14/09/23	K6	M6	19/09/23	21/09/23
K7	M2	28/08/23	30/08/23	K7	M4	15/09/23	18/09/23	K7	M6	22/09/23	24/09/23
K8	M2	12/08/23	16/08/23	K8	M4	22/08/23	26/08/23	K8	M6	31/08/23	02/09/23
K9	M2	16/08/23	20/08/23	K9	M4	28/08/23	01/09/23	K9	M6	06/09/23	08/09/23
K10	M2	20/08/23	24/08/23	K10	M4	03/09/23	07/09/23	K10	M6	13/09/23	15/09/23

Gambar 5.5 merupakan sebuah *gantt chart* untuk urutan *job* yang dikerjakan berdasarkan optimasi oleh NSGA-II untuk memperoleh *makespan* dan biaya keterlambatan optimal. *Job* dijadwalkan untuk mulai pertama kali tanggal 3 Agustus 2023, selanjutnya *job* lainnya dijadwalkan menyesuaikan *sequencing flowshop*. Dimana balok penuh mewakili rentang realisasi sedangkan balok bergaris putus mewakili rentang rencana berdasarkan *duse date* yang ditetapkan. Kemudian, Gambar 5.6 merupakan *gantt chart* untuk penjadwalan yang sedang berlaku. Berdasarkan tampilan tersebut terlihat bahwa antara kedua *gantt chart* tersebut, penjadwalan yang berlaku memiliki jarak lebih besar antara rencana dan realisasi dibandingkan dengan penjadwalan yang diusulkan. Rangkuman tanggal mulai dan tanggal selesai terdapat pada Tabel 5.3. Pada tabel tersebut, tanggal mulai paling cepat adalah 3 Agustus 2023 dan tanggal selesai paling lambat adalah 27 September 2023. Sehingga, besar *makespan* berdasarkan *timeline* yang disusun adalah 55 hari. Kemudian dilakukan identifikasi keterlambatan menyesuaikan *due date* setiap operasi. Cara yang digunakan sama dengan identifikasi keterlambatan pada penjadwalan yang sedang berlaku yaitu dengan membandingkan tanggal penyelesaian operasi pada jadwal usulan terhadap penyelesaian operasi pada realisasi. Selisih dari kedua tanggal tersebut mewakili jumlah hari keterlambatan setiap aktivitas. Jumlah hari keterlambatan tersebut akan dikalikan dengan biaya keterlambatan per hari untuk setiap operasi yang mewakili besar kerugian yang akan dialami perusahaan atas penjadwalan yang diusulkan. Berikut ini merupakan rangkuman identifikasi keterlambatan berdasarkan *timeline* yang disusun.

Tabel 5.4 Identifikasi Keterlambatan Penjadwalan Usulan

Pekerjaan	Mesin	Keterlambatan (hari)	Total Biaya Keterlambatan
K1	M1	1	Rp800,000.00
K3	M1	1	Rp800,000.00
K4	M1	1	Rp800,000.00
K5	M1	1	Rp800,000.00
K6	M1	1	Rp800,000.00
K7	M1	1	Rp800,000.00
K8	M1	1	Rp800,000.00
K9	M1	1	Rp800,000.00
K10	M1	2	Rp1,600,000.00
K1	M2	2	Rp800,000.00
K3	M2	0	Rp0.00
K4	M2	0	Rp0.00
K5	M2	0	Rp0.00
K6	M2	1	Rp400,000.00
K7	M2	0	Rp0.00

Pekerjaan	Mesin	Keterlambatan (hari)	Total Biaya Keterlambatan
K8	M2	2	Rp800,000.00
K9	M2	3	Rp1,200,000.00
K10	M2	3	Rp1,200,000.00
K1	M3	10	Rp6,000,000.00
K3	M3	15	Rp9,000,000.00
K4	M3	3	Rp1,800,000.00
K5	M3	2	Rp1,200,000.00
K6	M3	12	Rp7,200,000.00
K7	M3	14	Rp8,400,000.00
K8	M3	4	Rp2,400,000.00
K9	M3	6	Rp3,600,000.00
K10	M3	8	Rp4,800,000.00
K1	M4	1	Rp400,000.00
K3	M4	1	Rp400,000.00
K4	M4	1	Rp400,000.00
K5	M4	1	Rp400,000.00
K6	M4	1	Rp400,000.00
K7	M4	1	Rp400,000.00
K8	M4	2	Rp800,000.00
K9	M4	2	Rp800,000.00
K10	M4	2	Rp800,000.00
K1	M5	4	Rp1,200,000.00
K3	M5	1	Rp300,000.00
K4	M5	1	Rp300,000.00
K5	M5	1	Rp300,000.00
K6	M5	3	Rp900,000.00
K7	M5	2	Rp600,000.00
K8	M5	3	Rp900,000.00
K9	M5	3	Rp900,000.00
K10	M5	4	Rp1,200,000.00
K1	M6	0	Rp0.00
K3	M6	0	Rp0.00
K4	M6	0	Rp0.00
K5	M6	0	Rp0.00
K6	M6	0	Rp0.00
K7	M6	0	Rp0.00
K8	M6	0	Rp0.00
K9	M6	0	Rp0.00
K10	M6	0	Rp0.00
Total			Rp68,200,000.00

Keterlambatan telah diidentifikasi untuk setiap operasi dengan angka *makespan* yang diharapkan adalah 27 hari, namun keterlambatan ditunjukkan oleh kenaikan angka *makespan* menjadi 55 hari. Hal tersebut menunjukkan bahwa proyek akan terlambat selama 28 hari dengan total biaya keterlambatannya adalah sebesar Rp68,200,000.00. Besar *makespan* dan biaya keterlambatan pada penjadwalan yang sedang berlaku adalah

62 hari dan Rp621,200,000.00. Hal tersebut mengindikasikan adanya penurunan untuk *makespan* dan biaya keterlambatan sebesar 11.29% dan 89% untuk penjadwalan yang diusulkan.

Dalam penelitian ini, biaya keterlambatan direpresentasikan melalui biaya tenaga kerja harian. Meskipun menyederhanakan perhitungan, pendekatan ini memiliki keterbatasan karena tidak memperhitungkan komponen biaya lain yang mungkin timbul akibat keterlambatan, seperti denda, biaya *overhead,* dan potensi kehilangan pendapatan. Oleh karena itu, hasil analisis ini perlu diinterpretasikan dengan memperhatikan keterbatasan tersebut. Selain itu, penelitian telah berhasil menemukan solusi optimal untuk penjadwalan proyek manufaktur kereta dengan menggunakan berbagai skenario dalam komputasinya. Skenario yang diusulkan adalah modifikasi pada besar populasi, mu, dan lambda untuk menjaga keseimbangan antara eksplorasi dan eksploitasi. Pada angka mu dan lambda yang dipilih, mereka memiliki selisih cukup kecil dimana hal ini baik untuk menjaga solusi terbaik namun juga berisiko mengalami konvergensi prematur akibat ruang pencarian semakin kecil. Namun telah dibuktikan dengan uji *hypervolume* dimana besar ruang eksplorasi meningkat yang menandakan ruang eksplorasi tetap membesar dan tidak terjebak konvergensi prematur. Nilai populasi, mu, dan lambda yang dipilih cukup besar sehingga mempengaruhi durasi komputasi dan beban kerja perangkat komputer. Oleh karena itu, perlu diujikan kembali parameter tersebut pada skenario besar data berbeda dengan obyektif optimasi yang sama untuk menemukan formulasi angka populasi, mu, dan lambda yang efisien terhadap waktu dan perangkat yang digunakan.

BAB VI

PENUTUP

6.1 Kesimpulan

Penelitian telah dilakukan berdasarkan rumusan masalah yang disusun, serta telah ditemukan kesimpulan untuk menjawab masalah-masalah tersebut. Berikut ini merupakan kesimpulan yang diperoleh:

1. Penjadwalan yang saat ini berlaku di PT INKA mengakibatkan terjadinya keterlambatan sebanyak 35 hari dengan *makespan*-nya adalah 62 hari yang menyebabkan muncul biaya keterlambatan sebesar Rp621,200,000.00.
2. Penelitian telah mengusulkan penggunaan algoritma NSGA-II untuk menyusun penjadwalan produksi dengan obyektif meminimalkan nilai *makespan* dan biaya keterlambatan. Algoritma NSGA-II berhasil menyusun penjadwalan produksi dengan ketentuan aliran *flowshop* pada proses produksinya dengan total *makespan* adalah 55 hari dan besar biaya keterlambatan adalah Rp68,200,000.00. Penjadwalan yang disusun oleh algoritma NSGA-II mengakibatkan penurunan *makespan* yang berimbas pada kenaikan produktivitas dari penjadwalan yang berlaku sebesar 11.29% dan penurunan biaya keterlambatan yang meningkatkan efisiensi produksi dari penjadwalan yang berlaku sebesar 89%.

6.2 Saran

Penelitian yang dilakukan telah berhasil menjawab seluruh rumusan masalah yang disusun, namun masih terdapat beberapa potensi pengembangan dari penelitian ini. Berikut ini merupakan beberapa saran untuk penelitian mendatang:

1. Penelitian selanjutnya dapat menggunakan studi kasus lain yang memiliki ukuran data berbeda dengan obyektif optimasi yang sama untuk menemukan formulasi angka parameter yang paling sesuai untuk studi kasus serupa
2. Penelitian selanjutnya dapat menambah obyektif yang digunakan dalam penjadwalan menyesuaikan dampak keterlambatan bukan hanya dari sisi finansial, contoh dampak pemborosan energi.
3. Penelitian selanjutnya disarankan untuk mempertimbangkan komponen biaya keterlambatan lainnya selain biaya tenaga kerja harian, seperti biaya penalti, biaya

sewa alat, dan biaya *overhead*, untuk mendapatkan gambaran yang lebih komprehensif tentang dampak keterlambatan.

DAFTAR PUSTAKA

Abdi, A., & Zarandi, H. R. (2019). A meta heuristic-based task scheduling and mapping method to optimize main design challenges of heterogeneous multiprocessor embedded systems. *Microelectronics Journal*, *87*, 1–11. https://doi.org/10.1016/j.mejo.2019.03.006

Afira, N., & Wijayanto, A. W. (2022). Optimization of Waste Transportation Routes using Multi-objective Non-dominated Sorting Genetic Algorithm II (MNSGA-II) in the Eastern and Southern Regions of Bandung City, Indonesia. *Proceedings of The International Conference on Data Science and Official Statistics*, *2021*(1), 20–30. https://doi.org/10.34123/icdsos.v2021i1.27

Alonso Campos, J. C., Jiménez-Bello, M. A., & Martínez Alzamora, F. (2020). Real-time energy optimization of irrigation scheduling by parallel multi-objective genetic algorithms. *Agricultural Water Management*, *227*(September 2019). https://doi.org/10.1016/j.agwat.2019.105857

Andiyan, A., & Rachmat, A. (2021). Analisis Manfaat Pembangunan Infrastruktur Keretaapi Di Pulau Jawa. *Jurnal Pendidikan dan Teknologi Indonesia*, *1*(3), 121–129. https://doi.org/10.52436/1.jpti.22

Anjana, V., Sridharan, R., & Ram Kumar, P. N. (2020). Metaheuristics for solving a multi-objective flow shop scheduling problem with sequence-dependent setup times. *Journal of Scheduling*, *23*(1), 49–69. https://doi.org/10.1007/s10951-019-00610-0

Bahy, M. B., & Musdholifah, A. (2022). Fast Non-dominated Sorting in Multi Objective Genetic Algorithm for Bin Packing Problem. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *16*(1), 55. https://doi.org/10.22146/ijccs.70677

Baker, K. R., & Trietsch, D. (1974). *Introduction To Sequencing and Scheduling*. John Wiley and Sons. http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203

Carter, M. P., Depallo, M. P., Ford, N. P., Gilliam, F. M., Green, K. R., Inglish, J. M., Lochte, W. D., Mcneil, G. W., Nelson, J. A., Shaw, J. W., Turney, D. L., Waters, K. D., & Watson, L. S. (2004). *Public Transportation Fact Book*. American Public Transportation Association.

Chen, T. L., Cheng, C. Y., & Chou, Y. H. (2020). Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming. *Annals of Operations Research*, *290*(1–2), 813–836. https://doi.org/10.1007/s10479-018-2969-x

De Rainville, F. M., Fortin, F. A., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: A Python framework for Evolutionary Algorithms. *GECCO'12 - Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Companion*, 85–92. https://doi.org/10.1145/2330784.2330799

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Faisal, S. (1990). *Penelitian Kualitatif : Dasar-dasar dan Aplikasinya*. Yayasan Asih Asah Asuh (YA3).

Grosch, B., Kohne, T., & Weigold, M. (2021). Multi-objective hybrid genetic algorithm for energy adaptive production scheduling in job shops. *Procedia CIRP*, *98*(March), 294–299. https://doi.org/10.1016/j.procir.2021.01.106

Guerreiro, A. P., Fonseca, C. M., & Paquete, L. (2021). The Hypervolume Indicator: Computational Problems and Algorithms. *ACM Computing Surveys*, *54*(6). https://doi.org/10.1145/3453474

Hwang, C. . (1979). Multiple objective decision making, methods and applications : a state-of-the-art survey. *Lecture Notes in Economics and Mathematical Systems*, 310–351. https://doi.org/10.1007/978-3-642-45511-7

Kartiasih, F. (2019). Dampak Infrastruktur Transportasi Terhadap Pertumbuhan Ekonomi di Indonesia Menggunakan Regresi Data Panel. *Jurnal Ilmiah Ekonomi dan Bisnis*, *16*(1), 67–77.

Kemendikbud-Ristek. (2016a). *KBBI Daring*. https://kbbi.kemdikbud.go.id/entri/transportasi

Kemendikbud-Ristek. (2016b). *KBBI Daring*. https://kbbi.kemdikbud.go.id/entri/kereta api

Kusuma, P. D. (2021). Multi-objective Batch Scheduling in Collaborative Multi-product Flow Shop System by using Non-dominated Sorting Genetic Algorithm. *International Journal of Advanced Computer Science and Applications*, *12*(9), 349–357. https://doi.org/10.14569/IJACSA.2021.0120939

Laha, D. (2007). Heuristics and metaheuristics for solving scheduling problems. *Handbook of Computational Intelligence in Manufacturing and Production Management, January 2007*, 1–18. https://doi.org/10.4018/978-1-59904-582-5.ch001

Mahmudy, W. F., & Rahman, M. A. (2011). Optimasi Fungsi Multi-Obyektif Berkendala Menggunakan Algoritma Genetika Adaptif Dengan Pengkodean Real. *Jurnal Ilmiah KURSOR*, *6*(1), 19–26.

Mayer, M. J., Szilágyi, A., & Gróf, G. (2020). Environmental and economic multi-objective optimization of a household level hybrid renewable energy system by genetic algorithm. *Applied Energy*, *269*(January). https://doi.org/10.1016/j.apenergy.2020.115058

Pirozmand, P., Hosseinabadi, A. A. R., Farrokhzad, M., Sadeghilalimi, M., Mirkamali, S., & Slowik, A. (2021). Multi-objective hybrid genetic algorithm for task scheduling problem in cloud computing. *Neural Computing and Applications*, *33*(19), 13075–13088. https://doi.org/10.1007/s00521-021-06002-w

PT. INKA. (2022). *Laporan Tahunan 2021*.

PT. KAI. (2023). *Jumlah Penumpang Kereta Api*. BPS-Statistic Indonesia. https://www.bps.go.id/

Sugiyono, D. (2013). *Metode Penelitian Kuantitatif, Kualitatif, dan Tindakan*.

Szostak, R. (1991). *The Role of Transportation in the Industrial Revolution*. McGill-Queen's University Press.

Tampubolon, F. R. (2021). Implementasi Hybrid Algorithm Untuk Optimalisasi Konsumsi Energi Pada Job Shop Scheduling. *Jurnal Elektro dan Mesin Terapan*, *7*(2), 58–65. https://doi.org/10.35143/elementer.v7i2.5140

Valledor, P., Gomez, A., Puente, J., & Fernandez, I. (2022). Solving Rescheduling Problems in Dynamic Permutation Flow Shop Environments with Multiple Objectives Using the Hybrid Dynamic Non-Dominated Sorting Genetic II Algorithm. *Mathematics*, *10*(14). https://doi.org/10.3390/math10142395

Verma, S., Pant, M., & Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*, *9*, 57757–57791. https://doi.org/10.1109/ACCESS.2021.3070634

Xu, W., Hu, Y., Luo, W., Wang, L., & Wu, R. (2021). A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission. *Computers and Industrial Engineering*, *157*(April), 107318. https://doi.org/10.1016/j.cie.2021.107318

Yagmahan, B., & Yenisey, M. M. (2008). Ant colony optimization for multi-objective flow shop scheduling problem. *Computers and Industrial Engineering*, *54*(3), 411–420. https://doi.org/10.1016/j.cie.2007.08.003

Yu, C., Andreotti, P., & Semeraro, Q. (2020). Multi-objective scheduling in hybrid flow shop: Evolutionary algorithms using multi-decoding framework. *Computers and Industrial Engineering*, *147*, 106570. https://doi.org/10.1016/j.cie.2020.106570

Zan, X., Wu, Z., Guo, C., & Yu, Z. (2020). A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems. *Advances in Mechanical Engineering*, *12*(1), 1–15. https://doi.org/10.1177/1687814019885294

LAMPIRAN

A- Skrip Algoritma NSGA-II

(Sumber : https://github.com/mangomash1/NSGA-II-Scheduling.git)

```
Run Cell | Run Below | Debug Cell
# %%
# Import Library
import pandas as pd
import numpy as np
from deap import base, creator, tools, algorithms
from pymoo.indicators.hv import HV

Run Cell | Run Above | Debug Cell
# %%
# Load datasets (example datasets, replace with your actual data loading)
processing_times = pd.read_csv('processing_times.csv', index_col=0)
due_dates = pd.read_csv('due_dates.csv', index_col=0)
lateness_costs = pd.read_csv('tardiness_costs.csv', index_col=0)

# Number of jobs and machines
num_jobs = processing_times.shape[0]
num_machines = processing_times.shape[1]

Run Cell | Run Above | Debug Cell
# %%
# Create fitness and individual classes
creator.create("FitnessMin", base.Fitness, weights=(-1.0, -1.0))  # Two objectives: makespan and total lateness cost
creator.create("Individual", list, fitness=creator.FitnessMin)

Run Cell | Run Above | Debug Cell
# %%
# Initialize toolbox
toolbox = base.Toolbox()

# Attribute generator
toolbox.register("attr_job", np.random.permutation, num_jobs)

# Structure initializers
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_job, num_machines)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

Run Cell | Run Above | Debug Cell
# %%
# Evaluation function
def evaluate(individual):
    schedule = np.zeros((num_jobs, num_machines))
    completion_times = np.zeros((num_jobs, num_machines))

    for machine_idx in range(num_machines):
        for job_idx in range(num_jobs):
            job = individual[machine_idx][job_idx]
            if job_idx == 0:
                start_time = 0
            else:
                previous_job = individual[machine_idx][job_idx - 1]
                start_time = completion_times[previous_job, machine_idx]

            if machine_idx > 0:
                prev_machine_end_time = completion_times[job, machine_idx - 1]
                start_time = max(start_time, prev_machine_end_time)

```