OPTIMASI PENJADWALAN NO-WAIT FLOWSHOP DENGAN SEQUENCE-DEPENDENT SETUP TIME MENGGUNAKAN ALGORITMA EVOLUSI

TUGAS AKHIR

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

Nama : Sofi Hendra Mukti Wibowo
No. Mahasiswa 21522033

PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2025

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.

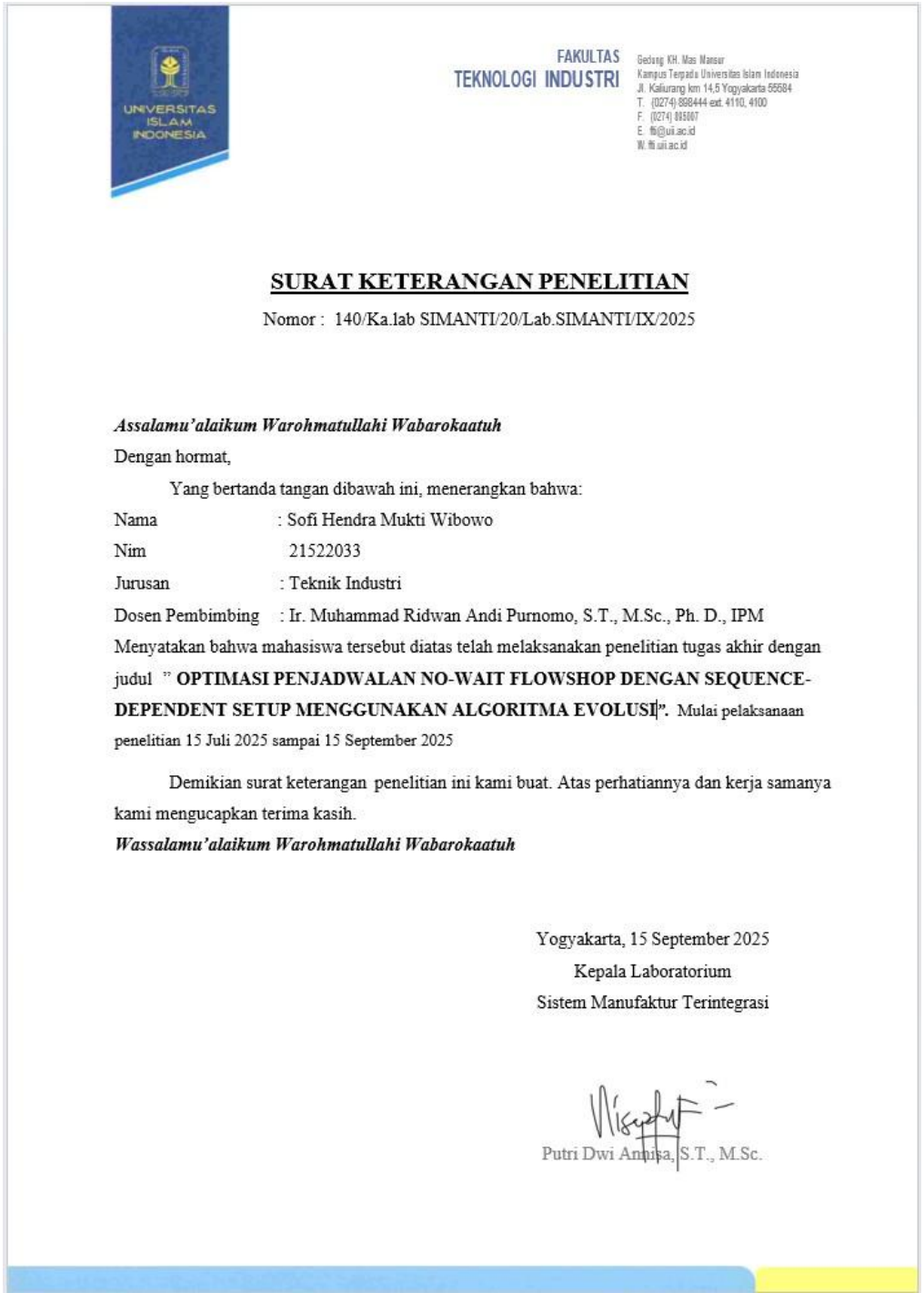
Yogyakarta, 23 – 04 - 2025



(Sofi Hendra Mukti Wibowo)

21522033

SURAT BUKTI PENELITIAN



UNIVERSITAS ISLAM INDONESIA

FAKULTAS TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4110, 4100
F. (0274) 895007
E. ft@uii.ac.id
W. ft.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor : 140/Ka.lab SIMANTI/20/Lab.SIMANTI/IX/2025

Assalamu'alaikum Warohmatullahi Wabarokaatuh

Dengan hormat,

Yang bertanda tangan dibawah ini, menerangkan bahwa:

Nama : Sofi Hendra Mukti Wibowo

Nim 21522033

Jurusan : Teknik Industri

Dosen Pembimbing : Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph. D., IPM

Menyatakan bahwa mahasiswa tersebut diatas telah melaksanakan penelitian tugas akhir dengan judul " **OPTIMASI PENJADWALAN NO-WAIT FLOWSHOP DENGAN SEQUENCE-DEPENDENT SETUP MENGGUNAKAN ALGORITMA EVOLUSI".** Mulai pelaksanaan penelitian 15 Juli 2025 sampai 15 September 2025

Demikian surat keterangan penelitian ini kami buat. Atas perhatiannya dan kerja samanya kami mengucapkan terima kasih.

Wassalamu'alaikum Warohmatullahi Wabarokaatuh

Yogyakarta, 15 September 2025

Kepala Laboratorium

Sistem Manufaktur Terintegrasi

Putri Dwi Amisa, S.T., M.Sc.

LEMBAR PENGESAHAN PEMBIMBING

OPTIMASI PENJADWALAN NO-WAIT FLOWSHOP DENGAN SEQUENCE-DEPENDENT SETUP TIME MENGGUNAKAN ALGORITMA EVOLUSI

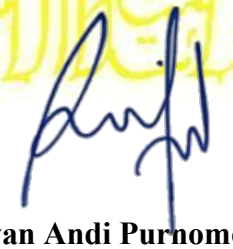
TUGAS AKHIR

Disusun Oleh :

Nama : Sofi Hendra Mukti Wibowo
No. Mahasiswa : 21522033

Yogyakarta, 16 September 2025

Dosen Pembimbing



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

LEMBAR PENGESAHAN DOSEN PENGUJI

OPTIMASI PENJADWALAN NO-WAIT FLOWSHOP DENGAN SEQUANCE-DEPENDET SETUP TIME MENGGUNAKAN ALGORITMA EVOLUSI

TUGAS AKHIR

Disusun Oleh :
Nama : Sofi Hendra Mukti Wibowo
No. Mahasiswa : 21522033

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

Yogyakarta, Tanggal - Bulan – Tahun

Tim Penguji

Ir. Muhammmad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM	
Ketua	
Ir. Vembri Noor Helia, S.T., M.T., IPM	
Anggota I	
Putri Dwi Annisa S.T., M.Sc	
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

Bismillahirrahmanirrahim, dengan penuh rasa syukur ke hadirat *Allah Subhanahu Wa Ta'ala*, laporan tugas akhir ini saya persembahkan untuk kedua orang tua tercinta yang senantiasa mendoakan, membimbing, dan memberikan kasih sayang tanpa batas, serta untuk kakak saya tersayang yang selalu memberikan dukungan, semangat, dan inspirasi dalam setiap langkah perjalanan hidup saya. Semoga karya sederhana ini dapat menjadi wujud kecil dari rasa terima kasih atas segala pengorbanan, doa, dan kasih sayang yang telah diberikan. Saya berharap laporan ini juga menjadi penyemangat bagi saya untuk terus berusaha, belajar, dan berbakti kepada keluarga tercinta.

MOTTO

"Dan katakanlah: Bekerjalah kamu!"
(QS. At-Taubah: 105)

"Sesungguhnya bersama kesulitan ada kemudahan."
(QS. Al-Insyirah: 6)

"Allah menyukai orang-orang yang bertawakal."
(QS. Ali Imran: 159)

KATA PENGANTAR

Bagian ***Bismillahirrohmanirrohim***

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil'alamin Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga dan para sahabatnya, yang telah menjadi teladan dalam menuntut ilmu dan menjalankan kehidupan yang diridhai oleh Allah SWT.

Dalam proses penyusunan Tugas Akhir ini, penulis telah menerima banyak dukungan, arahan, dan semangat dari berbagai pihak. Penulis menyampaikan penghargaan yang setinggi-tingginya kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU., ASEAN Eng., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM., selaku Ketua Program Studi Teknik Industri dan juga selaku Dosen Pembimbing Tugas Akhir yang telah dengan sabar memberikan bimbingan, arahan, serta masukan yang sangat berharga kepada penulis.
3. Seluruh Dosen Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia, yang telah membekali penulis dengan ilmu pengetahuan dan pengalaman selama masa perkuliahan.
4. Orang tua dan keluarga tercinta yang selalu menjadi sumber semangat, doa, dan kasih sayang yang tak pernah henti selama perjalanan pendidikan ini.
5. Rekan-rekan mahasiswa Teknik Industri atas kebersamaan, dukungan, dan kerja sama yang luar biasa selama masa studi.

Penulis menyadari bahwa dalam laporan Tugas Akhir ini masih terdapat kekurangan dan keterbatasan. Semoga Tugas Akhir ini dapat memberikan manfaat dan kontribusi bagi pengembangan ilmu pengetahuan, khususnya di bidang Teknik Industri.

Akhir kata, penulis berharap segala kebaikan dan bantuan yang telah diberikan mendapat balasan dari Allah SWT. Aamiin ya Rabbal 'Alamin.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 23 April 2025



Sofi Hendra Mukti Wibowo

ABSTRAK

Penelitian ini memiliki permasalahan penjadwalan produksi pada sistem *no-wait flowsho*p dengan *sequence-dependent setup time* di Perusahaan Elektronik X, yang menghadapi tantangan dalam menyusun urutan pekerjaan yang efisien untuk meminimalkan makespan atau waktu total produksi. Untuk mengatasi kompleksitas masalah ini, digunakan Algoritma Genetika (AG) yang merupakan salah satu metode optimasi dari *Evolusioner Algoritma* (EA). Melalui mekanisme seleksi, *crossover*, dan mutasi, Algoritma Evolusi dapat menemukan solusi terbaik dari ruang solusi yang sangat besar. Temuan dari penelitian ini menunjukkan bahwa pendekatan tersebut efektif dalam menentukan urutan produksi yang optimal, di mana total waktu produksi (*makespan*) berhasil berkurang dari 171 menit menjadi 151 menit setelah dilakukan proses optimasi. Penurunan sebesar 20 menit tersebut membuktikan bahwa metode ini secara signifikan mampu meningkatkan efisiensi sistem produksi. Selain itu, penelitian ini juga menegaskan pentingnya mempertimbangkan faktor *sequence-dependent setup time* untuk meningkatkan kualitas solusi penjadwalan. Hasil yang dicapai sejalan dengan tujuan penelitian untuk mencapai penjadwalan yang lebih efisien dan meningkatkan daya saing perusahaan.

Kata Kunci: Penjadwalan *No-Wait Flowshop*, *Sequence-Dependent Setup*, Algoritma Genetika, Optimasi Minimasi *Makespan*.

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 5
1.3 Tujuan Penelitian 6
1.4 Manfaat Penelitian 6
1.4.1 Bagi Peneliti 6
1.4.2 Bagi Perguruan Tinggi 6
1.4.3 Bagi Perusahaan 6
1.5 Batasan Penelitian 7
1.6 Sistematika Penelitian 7
BAB II TINJAUAN PUSTAKA **12**
2.1 Kajian Literatur 12
2.2 *Research Gap* 15
2.3 Landasan Teori 17
2.3.1 Flowshop scheduling 17
2.3.2 No-Wait Flowshop Scheduling 17
2.3.3 Sequence-Dependent Setup Time 19
2.3.4 Processing Time Dependency 20
2.3.5 Makespan, Erliness, dan Lateness 21
2.3.6 Algoritma Genetika (Genetic Algorithm) 21
2.3.7 Evolutionary Algorithm 23
BAB III METODE PENELITIAN **24**
3.1 Objek dan Subjek Penelitian 24
3.2 Jenis Data Penelitian 24
3.3 Alur Penelitian 24
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA **34**
4.1 Pengumpulan Data 34
4.2 Pengolahan Data 36
4.2.1 Penentuan solusi terbaik 40
BAB V PEMBAHASAN **42**
5.1 Analisis Perbandingan Variabel Keputusan 42
5.2 Reaktif *Scheduling* 44
BAB VI KESIMPULAN DAN SARAN **45**
6.1 Kesimpulan 45
6.2 Saran 45

DAFTAR PUSTAKA 46
LAMPIRAN 49

DAFTAR TABEL

Tabel 1.1	Data Penjadwalan Produksi	5
Tabel 2.1	Posisi Penelitian yang dibandingan pada Penelitian Sebelumnya	15
Tabel 3.1	Validasi Model Penelitian	28
Tabel 3.2	Langkah-langkah perhitungan model *Evolutionary Algorithm*	31
Tabel 4.1	Data sekunder *Ready Time*	34
Tabel 4.2	Proses waktu penyiapan yang bergantung pada urutan pekerjaan	35
Tabel 4.3	Populasi	36
Tabel 4.4	Seleksi	37
Tabel 4.5	Kromosom Induk	38
Tabel 4.6	Kromosom Anak	38
Tabel 4.7	Kromosom Anak Induk 1 & 2	38
Tabel 4.8	Kromosom sebelum dan sesudah mutasi	40
Tabel 4.9	Sebelum dioptimasi	40
Tabel 4.10	Sesudah dioptimasi	41
Tabel 5.1	Sebelum dioptimasi	42
Tabel 5.2	Sesudah dioptimasi	43

DAFTAR GAMBAR

Gambar 3.1 Alur penelitian 25
Gambar 5.1 Sebelum dioptimasi 42
Gambar 5.2 Sesudah dioptimasi 43

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perusahaan Elektronik X merupakan produsen modul kontrol elektronik yang beroperasi pada sektor manufaktur peralatan rumah tangga seperti televisi, *microwave*, pendingin udara (Ac), kulkas, mesin cuci, mesin pencuci piring, *oven*, *rice cooker*, lampu, mesin kopi, *vacum cleaner*, *robotic vacum cleaner*, *water heater*, dispenser, hair dryer, catokan rambut, panci listrik, kompor listrik, *speaker* dan *microphone*. Dalam proses produksinya, setiap produk harus melewati lima tahap mesin secara berurutan, yaitu pemotongan PCB, aplikasi pasta solder, penempatan komponen, penyolderan *reflow*, dan pengujian akhir. Seluruh proses ini dilakukan pada lantai produksi dengan menerapkan sistem *no-wait flowshop*, yaitu kondisi di mana produk tidak diperbolehkan mengalami jeda antar mesin. Selain itu, waktu penyiapan antar produk berbeda-beda tergantung urutan pengerjaannya, yang disebut dengan *sequence-dependent setup time*.

Secara umum, penjadwalan merupakan proses mengalokasikan pekerjaan ke sumber daya terbatas seperti mesin, tenaga kerja, atau waktu, dengan tujuan meningkatkan efisiensi, meminimalkan keterlambatan, serta mengoptimalkan penggunaan sumber daya (Pinedo, 2012). Dalam dunia manufaktur, penjadwalan yang baik sangat penting untuk menghindari terjadinya penumpukan pekerjaan, *idle time* pada mesin, dan keterlambatan dalam pemenuhan pesanan.

Dalam sistem *no-wait flowshop*, setiap pekerjaan harus diproses secara terus-menerus dari satu mesin ke mesin berikutnya tanpa mengalami waktu tunggu *(waiting time)* di antara mesin. Artinya, waktu penyelesaian pada satu mesin harus disesuaikan sedemikian rupa agar pekerjaan dapat langsung diproses di mesin berikutnya tanpa antre. Hal ini menjadikan penjadwalan menjadi lebih kompleks dibandingkan dengan *flowshop* konvensional, karena setiap penjadwalan harus mempertimbangkan waktu proses dan urutan secara presisi untuk menghindari konflik jadwal antar mesin (Mucha, 2013).

Terdapat masalah utama yang dihadapi Perusahaan Elektronik X dalam proses produksinya, yaitu belum adanya sistem penjadwalan yang efisien. Produk harus melewati lima mesin secara berurutan tanpa boleh berhenti di tengah *(no-wait)*, tapi pada kenyataannya sering terjadi antrean atau penumpukan di salah satu mesin karena urutan

pengerjaan yang kurang tepat. Menurut pengakuan seorang pengawas produksi, pihak manajemen mengakui bahwa mereka sering keteteran dalam menyelesaikan semua order yang masuk. Kondisi ini menyebabkan beberapa pesanan tidak selesai tepat waktu. Hal ini menegaskan pentingnya penjadwalan yang efisien untuk meminimasi *makespan* sehingga seluruh pesanan dapat terselesaikan sesuai target waktu. Selain itu, setiap kali berpindah dari satu jenis produk ke produk lain, mesin membutuhkan waktu penyesuaian *(setup)* yang berbeda-beda tergantung urutan produk sebelumnya (Ruiz, Maroto, & Alcaraz, 2005). Hal ini menyebabkan waktu mesin menganggur lebih lama dan produksi menjadi lebih lambat. Kondisi di lantai produksi juga memperlihatkan banyak *work in process* (WIP) yang berserakan di antara work station. Perusahaan belum menyediakan ruang penyimpanan sementara, sehingga produk yang masih berupa komponen elektronik kecil sering kali berisiko hilang atau rusak. Permasalahan ini semakin menegaskan perlunya sistem penjadwalan yang lebih baik agar aliran produksi dapat berjalan lancar tanpa penumpukan barang di setiap stasiun kerja. Dari dua permasalahan tersebut dapat disimpulkan bahwa penelitian ini berfokus pada dua hal utama: pertama, meminimasi *makespan* untuk mengurangi jumlah pekerjaan yang tidak selesai tepat waktu, dan kedua, menerapkan konsep *no-wait flowshop* agar aliran barang lebih lancar sehingga tidak ada lagi produk yang berserakan di antara *workstation*.

Masalah semakin rumit karena jenis produk yang diproduksi cukup banyak dan permintaan dari pelanggan bisa berbeda-beda. Meminimalkan *makespan*, atau waktu total yang dibutuhkan untuk menyelesaikan serangkaian tugas dari awal hingga akhir, adalah tujuan krusial dalam manajemen produksi dan proyek karena secara langsung meningkatkan produktivitas (Pinedo, 2012). Ketika *makespan* diperpendek, perusahaan dapat menghasilkan lebih banyak barang atau jasa dalam periode waktu yang sama. Hal ini terjadi karena optimalisasi jadwal mengurangi waktu tunggu *(idle time)* untuk mesin dan pekerja, memastikan sumber daya yang ada seperti peralatan, tenaga kerja, dan bahan baku digunakan secara lebih efisien (Baker, 2009). Peningkatan efisiensi ini tidak hanya mempercepat waktu penyelesaian pesanan, tetapi juga mengurangi biaya operasional, seperti upah lembur dan konsumsi energi, yang pada akhirnya meningkatkan profitabilitas (Jain, 2014). Oleh karena itu, makespan *minimization* merupakan strategi inti yang memberikan keunggulan kompetitif, memungkinkan perusahaan untuk merespons permintaan pasar dengan lebih cepat sambil mengoptimalkan penggunaan semua sumber daya yang tersedia. Untuk mengatasi masalah ini, dibutuhkan metode yang

bisa membantu menentukan urutan pengerjaan yang paling efisien. Salah satu metode yang cocok digunakan adalah algoritma genetika, karena metode ini bisa mencari solusi terbaik dari banyak kemungkinan secara cepat dan efektif, terutama untuk masalah yang kompleks seperti ini.

Penelitian ini yang memiliki masalah penjadwalan produksi dapat diselesaikan dengan berbagai metode, terutama dalam sistem produksi yang kompleks seperti *no-wait flowshop* dengan waktu penyiapan *(setup time)* yang bergantung pada urutan produk. Pada studi kasus produksi modul kontrol elektronik ini, waktu proses setiap produk berbeda di tiap mesin, dan pergantian produk pada mesin membutuhkan waktu setup yang tidak konstan, melainkan tergantung pada jenis produk sebelumnya. Kondisi *no-wait* yang mengharuskan produk berjalan tanpa jeda antar mesin menambah tingkat kesulitan dalam menyusun jadwal produksi yang efisien (Verdi, 2014). Oleh karena itu, penulis memilih menggunakan algoritma genetika sebagai metode optimasi, karena algoritma ini mampu menangani masalah dengan ruang solusi yang sangat besar dan kompleks seperti ini.

Algoritma Genetika (AG) merupakan salah satu metode dari algoritma evolusi yang bersifat adaptif dan digunakan untuk mencari solusi terbaik dalam permasalahan optimasi. Dalam penerapannya, AG memiliki beberapa parameter penting, seperti peluang terjadinya mutasi, proses *crossover* (pertukaran bagian solusi), serta jumlah individu dalam setiap kelompok populasi (Subanar, 2014). Algoritma Genetika cocok digunakan untuk menyelesaikan masalah penjadwalan pada sistem *no-wait flowshop* karena mampu mencari urutan produksi terbaik di antara banyak kemungkinan yang kompleks. Dengan mekanisme seleksi, *crossover*, dan mutasi, Algoritma Genetika dapat menemukan kombinasi urutan kerja yang meminimalkan waktu total produksi *(makespan)* dan mematuhi aturan tanpa jeda serta waktu setup yang bergantung urutan.

Selain Algoritma Genetika (GA), berbagai metode metaheuristik lain seperti Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), dan Simulated Annealing (SA) juga telah banyak digunakan untuk menyelesaikan permasalahan penjadwalan flowshop. Masing-masing memiliki karakteristik dan keunggulan tersendiri. PSO, misalnya, dikenal memiliki konvergensi yang cepat dalam menemukan solusi, namun terkadang mudah terjebak pada solusi lokal jika ruang pencarian terlalu kompleks (Zhang, 2019). Sementara itu, ACO meniru perilaku koloni semut dalam mencari jalur optimal dan unggul dalam eksplorasi solusi secara global, namun membutuhkan waktu

komputasi yang relatif lebih lama (Dorigo, 2004). Simulated Annealing (SA) juga efektif untuk masalah penjadwalan berukuran sedang dengan mekanisme penerimaan solusi buruk secara terkontrol untuk menghindari jebakan lokal, meskipun performanya sangat bergantung pada parameter suhu pendinginan.

Dalam konteks sistem no-wait flowshop dengan sequence-dependent setup time, Algoritma Genetika dipilih karena mampu menyeimbangkan eksplorasi dan eksploitasi solusi, fleksibel dalam pengaturan parameter, dan terbukti efektif pada sistem kompleks dengan banyak variabel keputusan. Keunggulan ini menjadikan GA lebih adaptif dan efisien dibanding PSO atau ACO dalam menemukan urutan job yang optimal untuk meminimalkan makespan pada kondisi produksi nyata.

Evolutionary Algorithm (EA) merupakan metode optimasi yang meniru prinsip seleksi alam melalui proses iteratif berupa seleksi, reproduksi, dan mutasi. Menurut Taha (2017), EA efektif digunakan untuk menemukan solusi mendekati optimal pada permasalahan yang rumit, *non-linear*, dan berskala besar. Tidak seperti metode eksak yang sering membutuhkan waktu komputasi lama, EA lebih fleksibel dan efisien dalam menjelajahi ruang solusi yang luas sehingga cocok diterapkan pada persoalan alokasi beban kerja operator dengan banyak kemungkinan kombinasi. EA juga terbukti mampu meningkatkan kualitas solusi pada model MIP karena mempercepat pencarian dan mengurangi risiko terjebak pada solusi lokal.

Sementara itu, Excel Solver merupakan salah satu alat bantu optimasi yang populer di kalangan peneliti maupun praktisi bisnis berkat kemudahan penggunaannya. Solver berfungsi mencari nilai terbaik dari suatu fungsi tujuan *(objective function)* dengan mengatur variabel keputusan serta batasan yang ditentukan. Dalam bidang penjadwalan *no-wait flowshop*, Solver dapat dimanfaatkan untuk menekan *waiting time*, dan mengoptimasi *processing time*. Keunggulan utama Solver terletak pada sifatnya yang sederhana, mudah diakses, serta tidak memerlukan perangkat lunak canggih, sehingga sangat relevan bagi prusahaan elektronik. Melalui antarmuka Excel, pengguna dapat langsung memodelkan persoalan matematis dan menjalankan optimasi menggunakan metode seperti Simplex LP, GRG *Nonlinear*, maupun *Evolutionary Algorithm*.

Dengan demikian, Excel Solver berperan sebagai penghubung antara teori optimasi matematis seperti MIP dan EA dengan penerapannya di dunia nyata, terutama bagi perusahaan manufaktur yang membutuhkan solusi cepat, murah, dan efektif (Rahman & Wulandari, 2022). Integrasi antara MIP, EA, dan penjadwalan *no-wait*

flowshop yang bergantung pada *setup time* untuk membangun *waiting time* yang lebih efektif dan efisien. Waktu *setup* yang bergantung pada urutan dan tidak ada *space*, MIP difokuskan untuk menekan urutan perkerjaan, sedangkan EA mendukung optimasi *processing time* yang bergantung pada *setup* pekerjaan dan urutan *job*.

Tabel 1. 1 Data Penjadwalan Produksi

Mesin	Waktu *Job* Pertama	Waktu selesai *Job* Terakhir	Total Waktu (menit)
M1	0	138	138
M2	27	146	146
M3	32	153	153
M4	39	161	161
M5	45	171	171 (makespan)

Berdasarkan data aktual pada tabel penjadwalan produksi, terlihat bahwa proses pada mesin pertama (M1) sudah selesai pada menit ke-138, namun mesin kelima (M5) baru menyelesaikan seluruh pekerjaan pada menit ke-171. Hal ini menunjukkan adanya ketidakseimbangan beban kerja antar mesin, di mana M1 hingga M4 menyelesaikan pekerjaannya lebih cepat, tetapi produk harus menunggu giliran di mesin berikutnya sehingga menimbulkan *idle time* dan penumpukan *work in process* (WIP). Kondisi ini memperlihatkan bahwa aliran produksi tidak benar-benar lancar meskipun sistem yang diterapkan adalah *no-wait flowshop*. Nilai *makespan* sebesar 171 menit dapat dikategorikan panjang karena sebagian besar waktu terbuang akibat adanya *hidden waiting time* di antara perpindahan *job* antar mesin. Sebagai contoh, pada mesin kedua (M2) pekerjaan terakhir selesai pada menit ke-146, sementara mesin kelima (M5) baru dapat menyelesaikan pekerjaan terakhir pada menit ke-171, sehingga terdapat selisih 25 menit yang menandakan terjadinya antrean. Ketidakseimbangan ini mengakibatkan pemanfaatan mesin tidak optimal, aliran barang tersendat, dan risiko kerusakan produk meningkat akibat penumpukan komponen di lantai produksi. Sehingga memperkuat bahwa penjadwalan aktual belum efisien sehingga diperlukan metode optimasi untuk memperpendek *makespan* dan mengurangi waktu tunggu.

1.2 Rumusan Masalah

Berikut merupakan rumusan masalah dalam penelitian ini:

Bagaimana susunan urutan *job* penjadwalan produksi yang dapat meminimasi makespan sekaligus menghilangkan tumpukan barang antar stasiun untuk meminimalkan waktu penyelesaian produksi?

1.3 Tujuan Penelitian

Berikut merupakan tujuan dari penelitian ini:

Penelitian ini bertujuan untuk menyusun urutan *job* pada penjadwalan produksi dengan menerapkan konsep *no-wait flowshop* sehingga diperoleh jadwal yang mampu meminimasi *makespan* serta menghilangkan tumpukan barang antar stasiun kerja, sehingga waktu penyelesaian produksi dapat lebih efisien dan aliran proses produksi berjalan lancar.

1.4 Manfaat Penelitian

Berikut adalah manfaat penelitian bagi peneliti, perguruan tinggi, dan perusahaan dalam penelitian tugas akhir ini:

1.4.1 Bagi Peneliti

1. Menambah wawasan dan keterampilan praktis dalam bidang penjadwalan produksi, khususnya pada sistem *no-wait flowshop* dengan waktu penyiapan yang bergantung pada urutan.
2. Mengasah kemampuan dalam menganalisis dan memecahkan masalah produksi kompleks dengan menggunakan pendekatan algoritma genetika.
3. Memberikan pengalaman langsung dalam menerapkan teori teknik industri untuk menyelesaikan permasalahan nyata di sektor manufaktur elektronik.

1.4.2 Bagi Perguruan Tinggi

1. Menambah kontribusi keilmuan dan referensi akademik di bidang optimasi sistem produksi dan algoritma evolusioner.
2. Menjadi contoh studi kasus yang dapat digunakan dalam pembelajaran dan penelitian lanjutan, terutama di bidang penjadwalan produksi kompleks.
3. Meningkatkan kolaborasi dengan dunia industri sebagai mitra dalam pengembangan solusi teknik industri berbasis riset.

1.4.3 Bagi Perusahaan

1. Memberikan solusi penjadwalan yang lebih efisien untuk proses produksi modul elektronik, sehingga mengurangi antrean dan waktu tunggu antar mesin.

2. Membantu perusahaan meminimalkan makespan dan memaksimalkan pemanfaatan mesin dengan urutan produksi yang optimal.
3. Menjadi dasar untuk pengembangan sistem penjadwalan produksi yang lebih baik di masa mendatang guna meningkatkan daya saing perusahaan.

1.5 Batasan Penelitian

Berikut merupakan batasan dalam penelitian ini:

1. Penelitian ini dilakukan pada satu studi kasus, yaitu proses produksi di perusahaan elektronik yang memproduksi 20 jenis modul kontrol elektronik.
2. Sistem produksi yang dianalisis menggunakan pendekatan *no-wait flowshop*, di mana produk harus diproses secara berurutan tanpa adanya jeda antar mesin.
3. Waktu penyiapan *(setup time)* dan waktu proses *(processing time)* diasumsikan bergantung pada urutan pengerjaan produk *(sequence-dependent)*.
4. Penjadwalan dilakukan hanya pada lima mesin utama, yaitu pemotongan PCB, aplikasi pasta solder, penempatan komponen, penyolderan *reflow*, dan pengujian akhir.
5. Penelitian bertujuan untuk meminimalkan waktu penyelesaian total *(makespan)* tanpa mempertimbangkan gangguan eksternal seperti kerusakan mesin, keterlambatan material, atau perubahan permintaan pelanggan.

1.6 Sistematika Penelitian

Sistematika penelitian laporan pada umumnya terdiri dari beberapa bab pokok yang berfungsi untuk menjelaskan tahapan serta hasil penelitian secara runtut. Uraian singkat mengenai masing-masing bagian disajikan sebagai berikut.

BAB I PENDAHULUAN

Bab I Pendahuluan berfungsi sebagai bagian awal laporan penelitian yang memberikan gambaran menyeluruh mengenai dasar dan arah penelitian. Pada bab ini dijelaskan latar belakang penelitian yang menguraikan alasan serta fenomena yang mendasari pentingnya penelitian dilakukan, kemudian dilanjutkan dengan rumusan masalah yang menegaskan inti persoalan agar kajian lebih terarah. Selanjutnya disampaikan tujuan penelitian yang menggambarkan capaian yang ingin diraih, baik dalam memecahkan masalah teoretis maupun memberikan solusi praktis, serta manfaat penelitian yang menunjukkan

kontribusi akademis maupun praktis bagi pihak terkait. Selain itu, bab ini juga mencakup ruang lingkup penelitian yang berfungsi sebagai batasan agar penelitian tetap fokus dan realistis, serta sistematika penulisan tesis yang memaparkan susunan bab dan subbab sehingga pembaca dapat memahami alur penyajian laporan penelitian secara utuh.

BAB II TINJAUAN PUSTAKA

Bab II Tinjauan Pustaka berisi pembahasan mengenai kajian literatur, *research gap*, dan landasan teori yang menjadi dasar dari penelitian. Kajian literatur mencakup penelitian-penelitian terdahulu, baik berupa jurnal, buku, maupun karya ilmiah lain yang relevan dengan topik penelitian, sehingga dapat menunjukkan sejauh mana isu yang diteliti telah dibahas sebelumnya serta menegaskan posisi penelitian ini dalam pengembangan ilmu pengetahuan. Selanjutnya, *research gap* dijelaskan untuk menampilkan perbedaan, keterbatasan, maupun kekosongan dari penelitian terdahulu yang kemudian menjadi dasar pembeda sekaligus keunikan dari penelitian ini, baik dari sisi masalah, metode, maupun hasil yang diharapkan. Sementara itu, landasan teori menyajikan konsep, prinsip, definisi, serta teori yang relevan untuk memperkuat pemahaman, mendukung analisis, dan memberikan pijakan akademis yang jelas dalam menjelaskan fenomena yang diteliti. Dengan demikian, Bab II tidak hanya merangkum teori dan penelitian sebelumnya, tetapi juga menyusun kerangka konseptual yang utuh guna memperjelas konteks, arah, dan kontribusi penelitian yang dilakukan.

BAB II METODE PENELITIAN

Bab III Metodologi Penelitian berisi uraian mengenai subjek dan objek penelitian, data yang digunakan, serta alur penelitian yang ditempuh secara sistematis. Subjek penelitian menjelaskan pihak atau kelompok yang terlibat secara langsung, sedangkan objek penelitian menggambarkan aspek atau fenomena yang menjadi fokus kajian. Data penelitian yang digunakan dapat berupa data primer, yang diperoleh melalui observasi, wawancara, eksperimen, atau kuesioner, maupun data sekunder yang bersumber dari laporan, arsip, maupun literatur pendukung. Selanjutnya, alur penelitian dijabarkan untuk menggambarkan tahapan penelitian mulai dari identifikasi masalah, perumusan tujuan, pengumpulan data, analisis, hingga penarikan kesimpulan dan rekomendasi. Dengan adanya uraian ini, pembaca dapat memahami bagaimana penelitian dirancang dan

dilaksanakan secara terukur sehingga hasil yang diperoleh memiliki dasar metodologis yang jelas dan dapat dipertanggung jawabkan secara ilmiah.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab IV Pengumpulan dan Pengolahan Data berisi penjelasan mengenai proses pengumpulan data serta langkah-langkah pengolahan data sesuai dengan metode penelitian yang telah ditetapkan. Pada tahap pengumpulan data dijelaskan secara rinci sumber data yang digunakan, baik data primer yang diperoleh melalui observasi, wawancara, eksperimen, maupun kuesioner, maupun data sekunder yang berasal dari laporan terdahulu, dokumen, atau literatur pendukung, sehingga keaslian dan keabsahan data dapat dipertanggungjawabkan. Setelah data diperoleh, tahap selanjutnya adalah pengolahan data, yaitu proses mengolah data mentah menjadi informasi yang lebih terstruktur dan bermakna dengan menggunakan metode, teknik analisis, algoritma, atau pemodelan yang relevan dengan tujuan penelitian. Hasil dari pengolahan data kemudian dipaparkan secara sistematis dalam bentuk tabel, grafik, maupun gambar agar mudah dipahami serta dapat mendukung proses analisis dan pembahasan. Dengan demikian, bab ini memiliki peran penting dalam menunjukkan bagaimana data dikumpulkan, diproses, serta disajikan untuk menjawab rumusan masalah dan mencapai tujuan penelitian.

BAB V PEMBAHASAN

Bab V Pembahasan merupakan bagian penting dalam laporan penelitian yang berisi analisis mendalam terhadap hasil penelitian yang telah diperoleh, di mana data yang telah dikumpulkan dan diolah tidak hanya dipaparkan dalam bentuk angka, tabel, atau grafik, tetapi juga dianalisis secara kritis untuk menemukan makna yang lebih luas. Pada tahap ini, hasil penelitian dikaitkan dengan teori-teori yang relevan, konsep dasar pada landasan teori, serta penelitian terdahulu untuk menunjukkan posisi penelitian dalam konteks ilmu pengetahuan maupun praktik lapangan. Pembahasan tidak hanya mendeskripsikan data, tetapi juga memberikan interpretasi mengenai alasan di balik temuan, menjelaskan faktor-faktor yang memengaruhi hasil, serta menegaskan kontribusi penelitian terhadap penguatan teori maupun pemecahan masalah praktis. Selain itu, bagian ini juga memaparkan implikasi penelitian, baik akademis maupun praktis, sehingga hasil yang diperoleh dapat memberikan manfaat yang nyata. Dengan demikian, Bab V berperan penting dalam memastikan bahwa hasil penelitian mampu menjawab rumusan masalah, mencapai tujuan yang telah ditetapkan, serta memberikan pemahaman yang komprehensif bagi pembaca.

BAB VI KESIMPULAN DAN SARAN

Bab VI Kesimpulan dan Saran merupakan bagian akhir dari laporan penelitian yang menyajikan ringkasan menyeluruh atas hasil penelitian sebagai jawaban terhadap rumusan masalah yang telah dirumuskan sebelumnya. Kesimpulan disusun secara singkat, padat, dan tegas berdasarkan hasil analisis serta pembahasan, sehingga memberikan pemahaman akhir mengenai temuan utama penelitian sekaligus menunjukkan kontribusinya terhadap pengembangan ilmu pengetahuan maupun praktik di lapangan. Selain itu, bagian ini juga memuat saran yang berfungsi sebagai rekomendasi atas temuan penelitian, baik dalam bentuk saran akademis berupa arahan untuk penelitian selanjutnya agar dapat memperluas atau menyempurnakan kajian dengan pendekatan yang berbeda, maupun dalam bentuk saran praktis yang ditujukan kepada perusahaan, instansi, masyarakat, maupun akademisi agar hasil penelitian dapat dimanfaatkan sebagai dasar pengambilan keputusan, perbaikan sistem, atau implementasi kebijakan. Dengan demikian, Bab VI tidak hanya menutup keseluruhan penelitian, tetapi juga menegaskan manfaat, kontribusi, serta arah pengembangan penelitian di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Literatur

Kajian literatur dalam penelitian ini bertujuan untuk memberikan landasan teoretis yang kuat serta menggambarkan perkembangan penelitian terkait topik yang dibahas. Dalam kajian ini, akan diuraikan konsep, teori, dan hasil penelitian sebelumnya yang relevan untuk membangun kerangka pemikiran penelitian serta mengidentifikasi kesenjangan atau permasalahan yang belum terjawab. Dengan demikian, kajian literatur ini diharapkan dapat mengarahkan penelitian ini menuju pencapaian tujuan yang lebih jelas, dengan pendekatan yang didasarkan pada referensi yang valid dan relevan.

Penelitian oleh (Urip Sarwo Sambodo, 2003) bertujuan mengembangkan solusi penjadwalan untuk *no-wait flowshop* dengan meminimasi total waktu aktual produksi. Menggunakan algoritma genetika sebagai metode pencarian solusi, penelitian ini menunjukkan bahwa pendekatan *evolusioner* mampu menghasilkan urutan job yang lebih efisien, mempercepat penyelesaian produksi dibandingkan metode konvensional. Relevansinya terletak pada penggunaan *metaheuristik* untuk mengatasi kompleksitas *flowshop* tanpa waktu tunggu.

Penelitian oleh (Lely Herlina, 2021) berfokus pada optimasi jadwal pengiriman pallet dan dunnage di lingkungan produksi. Dengan mengaplikasikan model optimasi berbasis heuristik dan mempertimbangkan *makespan* sebagai kriteria utama, hasil penelitian menunjukkan pengurangan waktu total produksi secara signifikan. Pendekatan ini memperkuat pentingnya minimasi *makespan* dalam konteks *flowshop*.

Penelitian oleh (Ginting, 2019) menerapkan metode LPT untuk menjadwalkan produksi dalam sistem *flexible flowshop*. Dengan mengutamakan urutan pekerjaan berdasarkan waktu proses terpanjang, penelitian ini membuktikan bahwa LPT efektif dalam mengurangi jumlah keterlambatan dan meningkatkan utilisasi mesin.

Penelitian oleh (Herry Christian Palit, 2003) mengeksplorasi dampak waktu *setup* yang bergantung pada urutan terhadap penjadwalan produksi. Penelitian ini menggunakan metode relaksasi Lagrangian untuk mengatasi permasalahan optimasi yang kompleks dan menunjukkan bahwa mempertimbangkan waktu *setup* secara spesifik dapat memperbaiki *makespan* dan total *processing time*.

Penelitian oleh (Dian Setiya Widodo, 2018) berfokus pada pengembangan algoritma Hodgson untuk mengurangi jumlah pekerjaan terlambat *(tardy jobs)*. Dengan modifikasi algoritma klasik, penelitian ini membuktikan bahwa strategi penyesuaian urutan dapat mengurangi *lateness* dalam jadwal produksi.

Penelitian oleh (Alex Saleh, 2009) mengembangkan model penjadwalan untuk flowshop empat tahap. Dengan tujuan meminimalkan *lateness* maksimum dan jumlah keterlambatan, penelitian ini menggabungkan analisis *multi-kriteria* dalam optimasi produksi, menunjukkan bahwa penjadwalan *multistage* memerlukan perhatian khusus pada urutan dan waktu antar *stage*.

Penelitian oleh (Sari, 2007) melakukan studi komparatif antara beberapa metode penjadwalan. Hasilnya menunjukkan bahwa pemilihan metode penjadwalan berpengaruh signifikan terhadap ketepatan penyelesaian *job*, menegaskan pentingnya pemilihan algoritma yang sesuai dengan karakteristik sistem produksi.

Penelitian oleh (Zaini, 2023) bertujuan untuk meminimalkan makespan dan keterlambatan total dalam *no-wait flowshop* menggunakan algoritma genetika. Penelitian ini memperlihatkan bahwa algoritma *evolusioner* efektif dalam menangani permasalahan *scheduling* kompleks tanpa waktu tunggu antar proses.

Penelitian oleh (Ali Verdia, 2022) mengusulkan algoritma optimasi yang mempertimbangkan total *tardiness* dan batas total *completion time*. Penelitian ini menawarkan pendekatan *hybrid* yang mengintegrasikan strategi pencarian lokal untuk meningkatkan kinerja jadwal produksi.

Penelitian oleh (Jun Pei, 2018) memperkenalkan model penjadwalan yang mempertimbangkan pengalaman operator dan ketergantungan waktu proses terhadap sumber daya. Studi ini membuktikan bahwa *learning effect* dan *resource allocation* secara signifikan mempengaruhi hasil akhir penjadwalan.

Penelitian oleh (Xinna Geng, 2018) mengembangkan model matematis baru yang mengintegrasikan alokasi sumber daya konveks dan efek pembelajaran untuk mengoptimalkan urutan pekerjaan dalam *flowshop*. Hasil penelitian memperlihatkan bahwa faktor-faktor ini dapat memperbaiki total *completion time* secara signifikan.

Penelitian oleh (Garside, 2019) mengembangkan algoritma *hybrid* untuk mengoptimasi jadwal *flowshop* dengan tiga *stage* dan waktu *setup* yang diperhitungkan. Penelitian ini menggabungkan metode *heuristik* dan *preventive maintenance*, memperlihatkan peningkatan efisiensi produksi dan pengurangan *downtime.*

Penelitian oleh (Nainggolan, 2018) mengkaji integrasi antara jadwal produksi dan *preventive maintenance*. Penelitian ini membuktikan bahwa integrasi keduanya dapat meminimalkan *makespan* dan meningkatkan keandalan sistem produksi secara keseluruhan.

Penelitian oleh (Hebrard, 2011) mengkaji berbagai varian dari permasalahan *job shop scheduling*, termasuk masalah penjadwalan dengan waktu *setup* tergantung urutan. Hebrard memperkenalkan model-model matematis dan strategi penyelesaian berbasis *constraint* programming dan teknik optimasi kombinatorial untuk menghadapi kompleksitas jadwal. Penelitian ini relevan karena memberikan dasar metode optimasi canggih yang dapat diadaptasi untuk kasus *flowshop*, khususnya dalam menangani pengaruh waktu *setup* yang tidak tetap.

Penelitian oleh (Yuniaristanto, 2003) berfokus pada penjadwalan dalam *no-wait flowshop* dengan mempertimbangkan pembatasan waktu penyelesaian seragam *(common due-date)*. Dalam studi ini, dikembangkan model penjadwalan dengan tujuan meminimasi deviasi terhadap *common due-date*. Penelitian ini memperlihatkan pentingnya pengelolaan urutan kerja dalam *no-wait flowshop* untuk mengurangi keterlambatan kolektif, sebuah isu yang sangat relevan terhadap tujuan minimasi *lateness* dalam penelitian ini.

Berdasarkan kajian literatur, penelitian sebelumnya telah banyak membahas penjadwalan produksi pada *flowshop* dan variannya dengan beragam pendekatan, seperti algoritma genetika untuk no-wait flowshop, metode LPT pada *flexible*, maupun integrasi penjadwalan dengan logistik dan *preventive maintenance*. Beberapa penelitian juga menyoroti pengaruh *sequence-dependent setup time*, namun belum secara spesifik diterapkan pada sistem *no-wait flowshop*. Perbedaan utama penelitian ini adalah mengkaji penjadwalan ***no-wait flowshop*** **dengan** ***sequence-dependent setup time*** menggunakan **algoritma evolusi**, yang menghadirkan kompleksitas tambahan sekaligus relevan dengan kondisi aktual produksi modul kontrol elektronik. Algoritma evolusi dipilih karena memiliki kemampuan eksplorasi solusi yang luas, dapat menghindari jebakan solusi lokal, dan lebih efisien dibandingkan metode eksak maupun heuristik sederhana dalam menangani masalah penjadwalan yang kompleks. Dengan demikian, penelitian ini diharapkan mampu memberikan solusi penjadwalan yang lebih efisien, memperpendek *makespan*, serta meningkatkan kelancaran aliran produksi dibandingkan penelitian terdahulu.

2.2 *Research Gap*

Berdasarkan kajian literatur, maka dapat ditentukan posisi penelitian dibandingkan penelitian sebelumnya seperti yang tertera pada **Error! R eference source not found.** sebagai berikut:

Tabel 2. 1 Posisi Penelitian yang dibandingan pada Penelitian Sebelumnya

No	Peneliti	Sequence dependent set-up and procesing time	Minimasi earliness and lateness	No-wait flowshop	Algortima Evolusi
1	(Urip Sarwo Sambodo, 2003)		√	√	√
2	(Lely Herlina, 2021)	√	√		
3	(Ginting, 2019)		√		
4	(Herry Christian Palit, 2003)	√	√		
5	(Dian Setiya Widodo, 2018)		√	√	
6	(Alex Saleh, 2009)	√	√		
7	(PUSPITA, 2007)		√		
8	(Zaini, 2023)			√	√
9	(Ali Verdia, 2022)		√	√	√
10	(Jun Pei, 2018)	√	√		√
11	(Xinna Geng, 2018)		√	√	
12	(Garside, 2019)	√		√	
13	(Nainggolan, 2018)	√	√		
14	(Hebrard, 2011)		√	√	

No	Peneliti	Sequence dependent set-up and procesing time	Minimasi earliness and lateness	No-wait flowshop	Algortima Evolusi
15	(Yuniaristanto, 2003)	√	√	√	
16	Penelitian Saat Ini	√		√	√

2.3 Landasan Teori

2.3.1 *Flowshop scheduling*

Flowshop scheduling merupakan salah satu bentuk sistem penjadwalan yang diterapkan pada proses produksi dengan alur kerja linier, di mana setiap pekerjaan *(job)* harus melewati sejumlah mesin dalam urutan yang sama. Setiap *job* akan diproses di setiap mesin secara berurutan, dengan waktu proses yang bisa berbeda-beda pada tiap mesin. Tujuan utama dari *flowshop scheduling* adalah untuk mengoptimalkan kriteria kinerja tertentu, seperti meminimalkan *makespan* (waktu penyelesaian total), total *flow time*, ataupun total *tardiness* (Rubén Ruiz, 2003).

Menurut (Jan Riezebos, 1995), *flowshop scheduling* banyak diterapkan pada sistem manufaktur yang memiliki urutan operasi tetap, seperti pada industri otomotif, farmasi, percetakan, dan makanan-minuman. Pada sistem ini, setiap pekerjaan tidak hanya diproses secara berurutan tetapi juga dapat memiliki *multiple operations* pada setiap tahap dengan *time lags* tertentu yang harus diperhitungkan dalam penjadwalan. *Flowshop scheduling* memiliki beberapa karakteristik utama sebagai berikut:

a. Urutan Proses Tetap: Setiap *job* harus diproses melalui mesin-mesin dalam urutan yang sama.
b. Mesin Tidak Paralel: Setiap mesin hanya memproses satu *job* dalam satu waktu.
c. *Non-Preemptive*: Proses yang sedang berlangsung tidak boleh dihentikan sebelum selesai.
d. *Setup Time*: Pada beberapa kasus, terdapat waktu *setup* yang bergantung pada urutan pekerjaan.

2.3.2 No-Wait Flowshop Scheduling

No-wait flowshop scheduling merupakan salah satu varian dari *flowshop scheduling* di mana tidak diperbolehkan adanya waktu tunggu *(waiting time)* antara penyelesaian suatu operasi pada mesin dan dimulainya operasi berikutnya di mesin selanjutnya. Dengan kata lain, setiap *job* harus diproses secara berkesinambungan dari satu mesin ke mesin berikutnya tanpa jeda. Model ini sering dijumpai pada industri yang proses produksinya sensitif terhadap waktu, seperti industri kimia, makanan-minuman, logam cair, dan plastik, di mana waktu tunggu dapat menyebabkan penurunan kualitas produk atau kerusakan barang (Jarosław Pempera, 2007).

Menurut (Ali Verdi, 1999), permasalahan *no-wait flowshop* muncul ketika setiap pekerjaan *(job)* tidak hanya harus dijadwalkan untuk diproses pada setiap mesin secara berurutan, tetapi juga harus dipindahkan dari satu mesin ke mesin berikutnya tanpa mengalami penundaan. Hal ini menambah tingkat kompleksitas dibandingkan *flowshop* biasa karena jadwal harus disusun sedemikian rupa agar tidak terjadi tumpang tindih atau kekosongan antara operasi berturut-turut.

Secara formal, *no-wait flowshop* dapat dijelaskan sebagai berikut:

a. Terdapat *n job* dan *m mesin*.
b. Setiap *job j* memiliki waktu pemrosesan P_{ij} dimesin *i*.
c. Tidak boleh ada jeda antara selesainya proses pada mesin *i* dan dimulainya pada mesin *i* +1.

Tujuan umum dari *no-wait flowshop scheduling* adalah meminimalkan kriteria seperti makespan (C_{max}) total *flow time*, atau total *tardiness*, meskipun fokus yang paling umum adalah meminimalkan makespan (Urip Sarwo Sambodo, 2003).

$$C_{max} = \frac{max}{j} \ (C_j)$$

Dengan *constraint*:

$$S_{j,i} + 1 = S_{j,i} + P_{j,i}$$

Dimana $S_{j,i}$ adalah start time *job j* dimesin *i*.

Meminimalkan *makespan*, yaitu total waktu yang diperlukan untuk menyelesaikan seluruh rangkaian tugas dari awal hingga akhir, merupakan salah satu tujuan utama dalam penjadwalan produksi dan manajemen proyek. Menurut (Pinedo, 2012)., *makespan* sering dijadikan indikator kinerja karena berhubungan langsung dengan produktivitas. Ketika *makespan* diperpendek, perusahaan dapat meningkatkan *output* dalam periode tertentu tanpa harus menambah sumber daya baru. Dengan kata lain, semakin kecil nilai *makespan*, semakin efisien sistem produksi dalam memanfaatkan waktu yang tersedia.

Dari sudut pandang operasional, pengurangan *makespan* berdampak besar pada pemanfaatan sumber daya. (Baker, 2009). menjelaskan bahwa optimalisasi jadwal mampu mengurangi waktu tunggu *(idle time)* pada mesin maupun tenaga kerja, sehingga

tidak terjadi pemborosan akibat mesin menganggur atau pekerja menunggu material. Kondisi ini membuat penggunaan peralatan, tenaga kerja, dan bahan baku menjadi lebih maksimal. Selain itu, minimisasi *makespan* juga berpengaruh pada penurunan biaya operasional, seperti biaya tenaga kerja, konsumsi energi, serta biaya penyimpanan material.

Lebih jauh lagi, meminimalkan *makespan* memberikan nilai strategis bagi perusahaan dalam meningkatkan daya saing. Dengan jadwal yang efisien, perusahaan dapat memenuhi pesanan lebih cepat, meningkatkan kepuasan pelanggan, dan memperkuat posisi di pasar yang kompetitif. Hal ini juga mendukung penerapan strategi produksi modern, seperti *just-in-time* (JIT), yang menekankan efisiensi dan pengurangan pemborosan. Tidak heran, fokus pada minimisasi makespan kemudian mendorong lahirnya berbagai metode optimasi penjadwalan, mulai dari algoritma *heuristik*, *metaheuristik*, hingga model matematis yang dirancang untuk menghadapi kondisi kompleks, misalnya pada sistem *no-wait flowshop* dengan *sequence-dependent setup time* (SDST).

2.3.3 *Sequence-Dependent Setup Time*

Sequence-dependent setup time adalah waktu persiapan mesin yang tergantung pada urutan pekerjaan sebelumnya dan yang akan datang. Jika urutan pekerjaan berubah, maka waktu *setup* juga berubah. Ini menambah kompleksitas dalam penjadwalan, karena penentuan urutan job sangat mempengaruhi efisiensi waktu produksi. Hal ini sangat umum dalam industri manufaktur seperti tekstil, makanan, atau otomotif, di mana setiap produk memerlukan penyesuaian alat, pembersihan, atau pergantian komponen sebelum proses dapat dilanjutkan. SDST memperkenalkan tantangan tambahan karena urutan pengerjaan akan memengaruhi total waktu produksi dan membuat penjadwalan menjadi masalah kombinatorial yang lebih kompleks (Palit, Soetanto, & Novianti, 2004).

Masalah penjadwalan dengan SDST umumnya diformulasikan dengan fungsi obyektif untuk meminimalkan *makespan* atau total waktu penyelesaian. Salah satu bentuk matematis dari waktu setup dalam kasus SDST adalah:

$$CS_{ijk} = setup\ time\ dari\ job\ i\ ke\ job\ j\ pada\ mesin\ k$$

Dengan mempertimbangkan variabel ini dalam model penjadwalan, maka waktu penyelesaian tiap *job* pada mesin akan menjadi:

$$C_{(jk)} = C_{(j-1)} + s_{(j-1)jk} + P_{jk}$$

Dimana $C_{(jk\ )}$ adalah waktu penyelesaian job ke-j di mesin ke-k, P_{jk} adalah waktu proses, dan $s_{(j-1\ )jk}$ adalah waktu setup yang bergantung pada urutan *job* sebelumnya.

2.3.4 *Processing Time Dependency*

Processing time dependency adalah fenomena di mana waktu pemrosesan suatu *job* dapat bergantung pada faktor lain seperti pengalaman operator *(learning effect)* atau ketersediaan sumber daya. (Jun Pei, 2018) menunjukkan bahwa memperhitungkan *learning effect* dan resource *allocation* dalam penjadwalan dapat meningkatkan akurasi perencanaan waktu produksi.

Selain efek pembelajaran, variabilitas waktu pemrosesan juga dapat terjadi akibat faktor alokasi sumber daya *(resource allocation)*. Dalam pendekatan ini, waktu pemrosesan suatu pekerjaan dapat dikendalikan atau dikompresi melalui penambahan sumber daya seperti mesin tambahan, operator, atau energi produksi. Salah satu rumus yang merepresentasikan hal ini adalah

$$P_j = \frac{p_j^0}{1 + r_j}\ dimana\ r_j\ adalah$$

jumlah sumber daya tambahan yang dialokasikan untuk pekerjaan ke-j. Penelitian oleh (Zong-Jun Wei, 2023) mengembangkan model kombinasi antara efek pembelajaran berbasis waktu dan alokasi sumber daya konveks dalam penjadwalan mesin tunggal. Studi tersebut menemukan bahwa pengintegrasian dua aspek ini dalam model penjadwalan mampu mengoptimalkan efisiensi penyelesaian *job* dan membuat sistem lebih *fleksibel* dalam menyesuaikan perubahan kapasitas produksi dan keterampilan tenaga kerja.

Implikasi dari pemodelan *processing time dependency* ini sangat penting dalam dunia industri, karena pendekatan ini mencerminkan kondisi nyata yang lebih kompleks. Dalam lingkungan produksi nyata, operator tidak selalu bekerja dengan efisiensi yang konstan, dan sumber daya sering dialokasikan secara dinamis berdasarkan urgensi dan prioritas *job*. Dengan mempertimbangkan *learning effect* dan *resource allocation* dalam pemodelan penjadwalan, perusahaan dapat mengembangkan strategi produksi yang lebih responsif, efisien, dan realistis. Selain itu, pendekatan ini membuka peluang penerapan algoritma optimasi seperti algoritma genetika dan *ant colony optimization* dalam menyelesaikan persoalan penjadwalan yang lebih kompleks dan (Jun Pei, 2018).

2.3.5 *Makespan, Erliness, dan Lateness*

Dalam sistem penjadwalan produksi, evaluasi kinerja jadwal sering kali didasarkan pada tiga ukuran utama: *makespan, earliness, dan lateness*. Ketiga metrik ini memainkan peran penting dalam mengukur efisiensi dan efektivitas jadwal produksi.

Makespan merupakan total waktu yang dibutuhkan untuk menyelesaikan seluruh pekerjaan dalam suatu sistem produksi, dari awal hingga pekerjaan terakhir selesai. Tujuan utama dalam banyak masalah penjadwalan adalah meminimalkan makespan untuk meningkatkan efisiensi operasional dan mengurangi waktu menganggur mesin. Sebagai contoh, penelitian oleh (Mousavi, 2018) mengembangkan algoritma genetika multi-objektif untuk meminimalkan *makespan* dan total *tardiness* dalam lingkungan *hybrid flow shop* dengan *setup time* yang bergantung pada urutan pekerjaan.

Earliness mengacu pada penyelesaian pekerjaan sebelum *due date* yang ditetapkan. Meskipun menyelesaikan pekerjaan lebih awal dapat tampak menguntungkan, hal ini dapat menyebabkan biaya penyimpanan tambahan dan ketidakefisienan dalam sistem *just-in-time*. Sebaliknya, *lateness* mengukur keterlambatan penyelesaian pekerjaan melewati *due date*, yang dapat mengakibatkan penalti dan ketidakpuasan pelanggan. Dalam konteks ini, penelitian oleh (Karande, 2023) mengembangkan model penjadwalan *job-shop* dengan *due date* umum untuk meminimalkan biaya dan *makespan*, mempertimbangkan biaya *earliness* dan *tardiness* yang setara.

Selain itu, dalam sistem *no-wait flow shop*, di mana pekerjaan harus diproses secara berurutan tanpa jeda antar mesin, minimisasi *earliness* dan *tardiness* menjadi lebih kompleks. (Chen-Yang Cheng, 2019) mengusulkan model pemrograman *linear integer* untuk masalah penjadwalan *no-wait flow shop* dengan *setup time* yang bergantung pada urutan, bertujuan untuk meminimalkan *makespan* dan total *tardiness*.

2.3.6 Algoritma Genetika (Genetic Algorithm)

Algoritma genetika adalah metode optimasi berbasis prinsip evolusi dan seleksi alam. Dalam konteks penjadwalan, algoritma genetika digunakan untuk menemukan solusi terbaik dengan merepresentasikan urutan *job* sebagai kromosom dan mengaplikasikan operasi seleksi, *crossover*, dan mutasi (Zaini, 2023). Pendekatan ini sangat efektif untuk

menyelesaikan masalah kombinatorial seperti *no-wait flowshop* dengan *sequence-dependent setup time.*

Algoritma genetika mempunyai beberapa komponen utama yaitu:

a. Teknik Pengkodean

   Teknik pengkodean adalah salah satu metode yang digunakan dalam algoritma genetika untuk merepresentasikan gen dalam sebuah kromosom, di mana gen merupakan komponen dari kromosom tersebut. Umumnya, satu gen akan merepresentasikan satu variabel (Kusumadewi, 2005).

b. Metode Seleksi Dengan Roda *Roulette*

   Metode seleksi ini bertujuan untuk memberikan peluang reproduksi lebih besar kepada anggota populasi yang memiliki tingkat fitness atau kekuatan tertinggi. Langkah awal dalam proses seleksi ini adalah menghitung nilai *fitness*. Setiap individu dalam kelompok seleksi akan memperoleh peluang reproduksi yang bergantung pada nilai objektif dari seluruh individu di dalam kelompok tersebut. Nilai *fitness* ini kemudian akan digunakan pada tahap-tahap selanjutnya. Metode seleksi *roulette wheel* merupakan metode paling sederhana, juga dikenal sebagai stochastic sampling *with replacement*. Dalam metode ini, individu-individu dipetakan ke dalam segmen-segmen pada sebuah garis secara berurutan, di mana setiap segmen memiliki panjang yang proporsional dengan nilai *fitness* individunya. Proses ini diulang sampai diperoleh jumlah individu yang diinginkan (Kusumadewi, 2005).

$$fitness = Makespan\ setiap\ n\ job\ pada\ m\ mesin$$

$$F_{ij} = S_{ij} \qquad i = 1.2 \ldots\ldots, n \qquad j = 1,2, \ldots m$$

c. Crossover

   Proses *crossover* (penyilangan), yang juga dikenal sebagai rekombinasi, dilakukan terhadap dua kromosom untuk menghasilkan kromosom baru *(offspring)* yang diharapkan memiliki kualitas lebih baik dibandingkan induknya. Kromosom baru ini akan mewarisi sebagian sifat dari kromosom induk. Dalam algoritma genetika, terdapat berbagai metode rekombinasi yang dapat digunakan, seperti *Partially Mapped Crossover* (PMX), *Order Crossover, Cycle Crossover*, dan lainnya.

d. Mutasi

Setelah proses rekombinasi, mutasi dilakukan dengan memilih kromosom secara acak untuk diubah, kemudian menentukan lokasi mutasi pada kromosom tersebut. Jumlah kromosom yang mengalami mutasi ditentukan berdasarkan probabilitas mutasi yang telah ditetapkan (Zaini, 2023).

2.3.7 *Evolutionary Algorithm*

Excel Solver merupakan salah satu perangkat bantu optimasi yang banyak digunakan oleh peneliti maupun praktisi bisnis karena kemudahan dan kesederhanaannya. Solver berfungsi untuk mencari nilai optimal dari suatu fungsi tujuan *(objective function)* dengan cara mengatur variabel keputusan serta batasan yang ada. Dalam konteks penjadwalan *no-wait flowshop*, Solver dapat dimanfaatkan untuk meminimalkan *waiting time* dan mengoptimalkan *processing time*. Keunggulan utama Solver adalah sifatnya yang praktis, mudah diakses, dan tidak memerlukan perangkat lunak khusus, sehingga sangat sesuai digunakan oleh perusahaan elektronik maupun manufaktur. Dengan antarmuka Excel yang sederhana, pengguna dapat langsung membangun model matematis dan menjalankan optimasi menggunakan metode seperti *Simplex* LP, GRG *Nonlinear*, maupun *Evolutionary Algorithm*.

Dengan demikian, Excel Solver dapat berperan sebagai jembatan antara teori optimasi matematis seperti *Mixed Integer Programming* (MIP) dan *Evolutionary Algorithm* (EA) dengan implementasi nyata di dunia industri, khususnya bagi perusahaan manufaktur yang membutuhkan solusi cepat. (Rahman & Wulandari, 2022).. *Integrasi* antara MIP, EA, dan penjadwalan *no-wait flowshop* dengan *sequence-dependent setup time* dapat menghasilkan sistem penjadwalan yang lebih efektif. Pada kasus ini, MIP berfokus untuk menentukan urutan pekerjaan yang optimal guna menekan *waiting time*, sedangkan EA membantu mengoptimalkan *processing time* yang dipengaruhi oleh setup time untuk menentukan urutan pekerjaan yang optimal. Dengan sinergi ini, perusahaan dapat mencapai efisiensi produksi sekaligus meminimalkan hambatan dalam proses manufaktur.

BAB III

METODE PENELITIAN

3.1 Objek dan Subjek Penelitian

Objek penelitian ini adalah sistem penjadwalan produksi pada perusahaan manufaktur yang menerapkan sistem *no-wait flowshop*, yaitu sistem di mana setiap pekerjaan harus berpindah antar mesin secara langsung tanpa adanya waktu tunggu. Subjek dalam penelitian ini adalah urutan penjadwalan *job* yang akan diproses pada beberapa mesin produksi secara berurutan, dengan mempertimbangkan waktu *setup* dan waktu proses yang bergantung pada urutan pekerjaan. Penelitian ini dilakukan pada sebuah industri manufaktur pada peerusahaan elektronik, yang memiliki alur produksi *linier* dan waktu proses yang sensitif terhadap urutan pengerjaan.

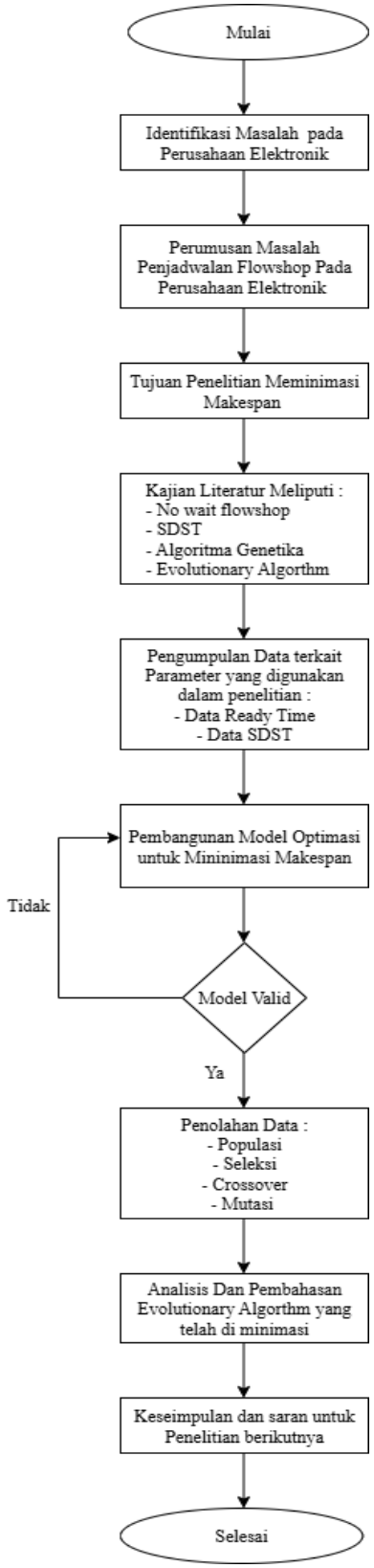
3.2 Jenis Data Penelitian

Jenis data yang digunakan dalam penelitian ini meliputi data sekunder. Data sekunder adalah data yang diperoleh lewat pihak lain, tidak langsung diperoleh oleh peneliti dari subjek penelitiannya (Hervin Rizky Pratama, 2024). Data sekunder diperoleh melalui studi pustaka, seperti buku, jurnal, dan artikel ilmiah yang membahas tentang *flowshop*, penjadwalan tanpa tunggu, dan waktu setup bergantung urutan *(sequence-dependent setup time)*. Dalam penelitian ini data-data yang didapat kan dari proses produksi di lantai pabrik elektronik sebagai berikut :

1. Data *Ready Time.*
2. Data *Sequence Dependent Setup Time.*

3.3 Alur Penelitian

Agar dapat mencapai tujuan yang diharapkan, maka adapun tahapan
yang akan dilakukan pada penelitian ini yaitu Gambar 3.1 adalah alur dalam penelitian ini sebagai berikut:



Gambar 3.1 Alur penelitian

1. Mulai

   Tahap awal merupakan titik permulaan penelitian, di mana peneliti mempersiapkan diri dengan menetapkan tujuan utama serta ruang lingkup penelitian yang akan dilaksanakan. Pada *fase* ini, diperlukan pemahaman awal mengenai permasalahan yang akan dipecahkan.

2. Identifikasi Masalah

   Peneliti mengidentifikasi secara detail proses produksi pada perusahaan elektronik yang menjadi objek penelitian, sehingga peneliti memperoleh gambaran utuh mengenai sistem yang berjalan. Proses identifikasi mencakup pemetaan aliran kerja dari awal hingga akhir produksi, pendataan jenis dan variasi produk yang dihasilkan, serta jumlah dan spesifikasi mesin yang digunakan pada setiap tahap proses. Selain itu, tahap ini juga mengkaji karakteristik sistem produksi, seperti urutan operasi yang wajib dilalui setiap produk, penerapan sistem *no-wait flowshop* yang tidak memperbolehkan adanya jeda antar proses, serta adanya *sequence-dependent setup time* yang membuat waktu penyiapan mesin berbeda tergantung urutan produk sebelumnya. Informasi ini menjadi dasar penting dalam perumusan masalah, penentuan variabel penelitian, dan pemilihan metode optimasi yang tepat.

3. Rumusan Masalah

   Berdasarkan pada tahap studi pendahuluan, peneliti mengidentifikasi permasalahan inti dalam sistem produksi, yaitu ketiadaan metode penjadwalan yang optimal pada lingkungan *no-wait flowshop* dengan *sequence-dependent setup time* (SDST). Kondisi tersebut memicu terjadinya antrean, waktu tunggu yang tidak diinginkan, serta rendahnya tingkat pemanfaatan mesin, yang pada akhirnya memperpanjang total waktu penyelesaian produksi *(makespan)*. Oleh sebab itu, fokus perumusan masalah diarahkan pada penyusunan urutan produksi yang mampu mengurangi hambatan tersebut dengan tetap mempertimbangkan karakteristik khusus sistem.

4. Tujuan

   Tujuan penelitian kemudian ditetapkan untuk menghasilkan urutan pengerjaan produk yang optimal pada seluruh mesin, sehingga *makespan* dapat diminimalkan

sekaligus menjaga kelancaran aliran proses sesuai prinsip *no-wait* dan perbedaan waktu penyiapan mesin berdasarkan urutan pengerjaan.

5. Kajian Literatur

   Pada penelitian ini terdiri dari kajian deduktif dan kajian induktif. Kajian deduktif berisi tentang kajian teoritis pada penelitian, pada penelitian ini kajian deduktif membahas mengenai *flowshop scheduling*, *no-wait flowshop*, SDST, serta metode penyelesaian menggunakan algoritma genetika. Sedangkan untuk kajian induktif yaitu membahas mengenai penelitian-penelitian terdahulu yang memiliki topik yang relevan dengan penelitian yang diangkat yang kemudian dibuat *research* gap untuk membedakan antara penelitian terdahulu dengan penelitian yang dilakukan.

6. Pengumpulan Data

   Data yang diperlukan adalah Data *Ready Time* (waktu kesiapan pekerjaan pada mesin pertama) dan Data *Sequence Dependent Setup Time* (waktu penyiapan yang bergantung pada urutan pekerjaan). Data ini berupa data sekunder dari penelitian sebelumnya untuk digunakan dalam model optimasi.

7. Pengembangan Model Optimasi

   Pada tahap ini, peneliti merumuskan model matematis yang menjadi inti dari penelitian. Proses diawali dengan mendeskripsikan karakteristik sistem yang menjadi objek kajian, kemudian dilanjutkan dengan pengembangan model matematis yang disusun berdasarkan variabel-variabel relevan. Model yang terbentuk selanjutnya dilakukan verifikasi untuk memastikan kesesuaiannya, baik dari sisi representasi kondisi nyata maupun konsistensi satuan dimensi. Apabila model dinyatakan valid, maka penelitian dapat berlanjut pada tahap pengolahan data dengan pendekatan analisis numerik. Sebaliknya, apabila ditemukan ketidaksesuaian, maka model harus diperbaiki atau dikembangkan kembali hingga memenuhi kriteria validasi.

8. Validasi Model

   Pada tahap ini, peneliti melakukan validasi terhadap model matematis yang telah dirumuskan. Validasi dilakukan untuk memastikan bahwa model benar-benar merepresentasikan kondisi sistem penjadwalan *no-wait flowshop* dengan *sequence-*

dependent setup time serta memiliki kesesuaian satuan dan logika. Langkah awal yang dilakukan adalah mendeskripsikan karakteristik sistem produksi yang menjadi objek penelitian, kemudian mengembangkan model matematis dengan memperhatikan variabel dan kendala yang relevan, seperti waktu proses, waktu penyiapan antar pekerjaan, serta batasan *no-wait*.

Model yang telah diformulasikan kemudian diverifikasi melalui pemeriksaan satuan dan konsistensi matematis. Proses verifikasi ini bertujuan memastikan bahwa fungsi tujuan dan kendala yang digunakan telah tepat dan dapat dipertanggungjawabkan. Jika model menunjukkan kesalahan, maka perlu dilakukan perbaikan hingga model dinyatakan valid. Setelah model tervalidasi, proses dapat dilanjutkan pada tahap pengolahan data menggunakan metode algoritma genetika untuk mencari urutan pekerjaan dengan *makespan* minimum.

Tabel 3.1 berikut menyajikan validasi model penelitian dengan memeriksa kesesuaian satuan pada fungsi tujuan dan kendala utama:

Tabel 3.1 Validasi Model Penelitian

Persamaan	Pemeriksaan Satuan	Status
Fungsi tujuan : min C_{max}	dinyatakan C_{max} dalam satuan waktu (menit/jam), konsisten dengan hasil penjumlahan waktu proses + *setup time*	Valid
Kendala urutan pekerjaan (setiap *job* dijadwalkan tepat sekali)	Variabel biner (0/1), tanpa dimensi fisik	Valid
Kendala *no-wait* (tidak ada waktu tunggu antar mesin)	Selisih waktu mulai & selesai antar mesin tetap dalam satuan waktu	Valid
Kendala *sequence-dependent setup time*	dinyatakan S_{ij} dalam satuan waktu dan ditambahkan pada waktu proses pekerjaan berikutnya	Valid

Berdasarkan tabel di atas, dapat disimpulkan bahwa model matematis yang digunakan dalam penelitian ini telah sesuai dari sisi dimensi, konsistensi logika, dan

kesesuaian dengan sistem nyata yang diteliti. Oleh karena itu, model dinyatakan valid dan dapat digunakan untuk tahap pengolahan data lebih lanjut.

9. Pengolahan Data

   Pada tahap ini peneliti menggunakan model matematis dan algoritma genetika. Model matematis adalah sebuah model yang akan membantu hasil nilai yang optimal karena pada penelitian ini berfokus untuk minimasi makespan untuk menemukan urutan *job* yang optimal. Model matematis dalam optimasi ini juga merupakan suatu masalah nyata ke dalam fungsi objektif, variabel keputusan dan kendala yang harus dipenuhi. Data yang telah diperoleh diolah menggunakan algoritma genetika untuk menentukan urutan pengerjaan terbaik yang meminimalkan *makespan*. Proses ini melibatkan populasi, seleksi, rekombinasi (*crossover*), dan mutasi.

   Berikut ini adalah model matematis untuk masalah penjadwalan *flowsop* dengan tujuan minimasi *makespan*:

Asumsi

- Semua pekerjaan *(job)* tersedia pada waktu t=0
- Setiap pekerjaan harus diproses melalui urutan mesin yang sama dan tetap (*flowshop*).
- Setiap mesin hanya dapat memproses satu pekerjaan pada satu waktu.
- Waktu pemrosesan pekerjaan j pada mesin i sudah diketahui dan tetap.
- Tidak ada preemption (pemotongan) pekerjaan yang sedang diproses.

Notasi

- n: jumlah total pekerjaan, dinotasikan dengan j = 1, 2,..., n.
- m: jumlah total mesin, dinotasikan dengan i = 1, 2, ..., m.
- Pij: waktu pemrosesan pekerjaan j pada mesin i.
- Cij: waktu penyelesaian pekerjaan j pada mesin i.
- $Cmax$: *makespan*, yaitu waktu penyelesaian pekerjaan terakhir pada mesin terakhir.

Model Matematis

Tujuan dari model ini adalah meminimalkan *makespan* (C_{max}).

Fungsi Tujuan:

Minimalkan C_{max}

1) Waktu penyelesaian pekerjaan pertama pada setiap mesin:
   Waktu penyelesaian pekerjaan pertama pada mesin pertama (C_{i1}) adalah waktu pemrosesannya, P_{i1} . Waktu penyelesaian pekerjaan *j* pada mesin *i* harus mempertimbangkan waktu penyelesaian sebelumnya.

$$C_{ij} = C_{ij} \text{-} 1 + P_{ij} \text{ untuk } j = 2,..., n$$

$$C_{i1} = C_{i1} ,1 + P_{i1} \text{ untuk } i = 2,..,m$$

2) Hubungan antara waktu penyelesaian:
   Waktu penyelesaian pekerjaan *j* pada mesin *i* harus lebih besar dari atau sama dengan waktu penyelesaian pekerjaan *j* pada mesin sebelumnya dan juga harus lebih besar dari atau sama dengan waktu penyelesaian pekerjaan sebelumnya pada mesin yang sama, ditambah waktu pemrosesan saat ini.
   $C_{ij} = C_{ij}$ -1 + P_{ij} untuk i = 1,.., m dan j = 2,...,n
   $C_{ij} = C_{i-1,J} + P_{ij}$ untuk i = 2, .., m dan j = 1, ..., n
3) Definisi *Makespan*
   Makespan adalah waktu penyelesaian pekerjaan terakhir (pekerjaan *n*) pada mesin terakhir (mesin *m*).

$$C_{max} = C_{nm}$$

Selanjutnya menghitung menggunakan algoritma genetika yang melibatkan populasi, seleksi, rekombinasi (*crossover*), dan mutasi.

1) Populasi adalah tahap awal dalam algoritma genetika di mana Anda membuat sekumpulan solusi awal secara acak. Solusi ini disebut individu atau kromosom, dan kumpulan solusi ini disebut populasi.
2) Seleksi merupakan proses pemilihan orang tua untuk reproduksi (biasanya didasarkan pada nilai *fitness*). Seleksi bertujuan untuk memberikan kesem-patan reproduksi yang paling besar bagi anggota populasi yang paling baik.
3) Rekombinasi (*crossover*) adalah sebuah proses yang membentuk kromosom baru dari dua kromosom induk dengan menggabungkan bagian informasi dari masing-masing kromosom. *Crossover* menghasilkan kromosom baru yang disebut

kromosom anak *(offspring). Crossover* bertujuan untuk menambah keanekaragaman *string* dalam satu populasi dengan penyilangan antar *string* yang diperoleh dari reproduksi sebelumnya

4) Mutasi merupakan proses mengubah secara acak nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi adalah operator algoritma genetika yang bertujuan untuk membentuk individu-individu yang baik atau memiliki kualitas diatas rata-rata. Selain itu mutasi dipergunakan untuk mengembalikan kerusakan materi genetik akibat proses *crossover.*

Data yang telah divalidasi selanjutnya diolah menggunakan metode analisis yang sesuai, baik melalui pendekatan statistik maupun algoritma optimasi. Tujuan pengolahan ini adalah menghasilkan *output* yang dapat diinterpretasikan untuk mendukung analisis penjadwalan *no-wait flowshop* dengan *sequence-dependent setup time* (SDST). Pada tahap ini, disusun model penjadwalan, yaitu model yang berfokus pada urutan pekerjaan menggunakan *Evolutionary Algorithm* (EA). Model Algoritma Genetika pertama digunakan sebagai pembanding, sedangkan model kedua *Evolutionary Algorithm* diarahkan untuk meminimalkan *makespan* dengan mempertimbangkan *setup time* yang bergantung pada urutan serta sifat *no-wait* antar mesin. Tabel 3.2 menunjukkan langkah-langkah perhitungan model *Evolutionary Algorithm.*

Tabel 3.2 Langkah-langkah perhitungan model *Evolutionary Algorithm*

1	Menentukan urutan pekerjaan (*job sequence*) yang akan diproses. Urutan ini direpresentasikan dengan variabel JobIndex (angka unik 1 sampai n) yang dipetakan menjadi nama *job* menggunakan fungsi Excel: $$Job = INDEX(Data_{ProcTime!A_n}; JobIndex)$$
2	Menghitung waktu *setup* antar *job* menentukan *setup time* antar pekerjaan berdasarkan matriks *setup*. Untuk *job* pertama, nilai *setup* = 0. Untuk *job* selanjutnya, *setup* diambil dari tabel *Setup_Matrix* sesuai kombinasi *job* sebelumnya dan *job* saat ini. $Setup_{ij}$ $\begin{cases} 0{,}1 & J = 1 \\ INDEX(Setup_{Matrix};\ Job_{j-1},\ Job_j) & \end{cases}$
3	Menghitung total *processing time* tiap *job* total waktu proses dihitung dengan menjumlahkan semua waktu proses pada setiap mesin.

	$$Proc_i = \sum_{m=1}^{M} PT_{im}$$
4	Menghitung waktu mulai pada M1 untuk *job* pertama, waktu mulai = 0. Untuk *job* berikutnya, waktu mulai dihitung dari waktu mulai *job* sebelumnya + total *processing time job* sebelumnya + *setup time* antar *job*. $$Start_j = Start_{j-1} + Proc_{j-1} + Setup_{j-1,j}$$
5	Menghitung waktu penyelesaian tiap *job* Waktu penyelesaian *job* diperoleh dari waktu mulai pada mesin pertama ditambah total *processing time job* tersebut. $$C_j = Start_j + Proc_j$$
6	Menentukan *makespan* yang didefinisikan sebagai waktu penyelesaian maksimum dari semua *job* dalam urutan. $$Makespan = \boldsymbol{max}(Cj), j = 1,2, \dots, n$$
7	Melakukan Optimasi dengan Solver Evolutionary pada Ms. Excel • Fungsi tujuan (*Objective*) : *Minimize Makespan* • Variabel pengubah : JobIndex (urutan *job*) • *Constraint* : nilai JobIndex harus *alldifferent* (tidak ada job yang diulang).

10. Analisis Hasil dan Pembahasan

    Pada tahap ini, dilakukan analisis dan pembahasan terhadap hasil perhitungan penjadwalan yang diperoleh dari penerapan *solver Evolutionary* pada model *no-wait flowshop* dengan *sequence-dependent setup time*. Analisis difokuskan pada perbandingan kinerja berbagai urutan pekerjaan yang dihasilkan, dengan variabel keputusan berupa urutan *job (job sequence)* dan kriteria performansi berupa nilai *makespan*. Pembahasan dilakukan untuk menjelaskan sejauh mana algoritma mampu meminimalkan *makespan* dibandingkan urutan awal. Selain itu, ditinjau pula efisiensi yang diperoleh dari solusi optimal atau mendekati optimal, baik dalam hal pengurangan waktu penyelesaian total maupun peningkatan kelancaran aliran produksi.

11. Kesimpulan & Saran

    Peneliti merumuskan kesimpulan berdasarkan hasil analisis dan pembahasan. Kesimpulan ini mencakup jawaban terhadap pertanyaan penelitian, pencapaian

tujuan, dan rekomendasi untuk penelitian lebih lanjut atau penerapan praktis. Saran biasanya diberikan untuk mengatasi keterbatasan penelitian atau mengusulkan arah penelitian lanjutan.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

Pengumpulan data dilakukan bertujuan agar dapat memenuhi semua kebutuhan pada penelitian yang akan dilakukan. Dalam penelitian ini, data yang dibutuhkan merupakan data sekunder dari Perusahaan Elektronik. Tabel 4.1 dan Tabel 4.2 dibawah ini merupakan data sekunder yang digunakan dalam penelitian ini.

Tabel 4.1 Data sekunder *Ready Time*

	M1	**M2**	**M3**	**M4**	**M5**	**Release**
P1	3	5	7	6	4	24
P2	4	6	5	7	3	20
P3	3	5	6	6	5	28
P4	5	6	8	7	6	24
P5	4	4	5	5	4	22
P6	3	5	7	6	4	24
P7	5	6	6	7	5	34
P8	4	5	6	6	4	22
P9	4	6	7	5	3	23
P10	3	4	5	5	4	34
P11	4	6	8	6	4	35
P12	3	5	6	6	3	26
P13	2	4	5	4	3	20
P14	2	3	4	3	2	23
P15	3	5	6	5	3	24
P16	2	4	5	5	3	22
P17	4	6	7	6	4	26
P18	2	3	4	3	2	22
P19	5	7	9	8	6	30
P20	3	5	6	5	4	14

Tabel 4.2 Merupakan proses waktu penyiapan yang bergantung pada urutan pekerjaan sebagai berikut:

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**	**P18**	**P19**	**P20**
P1	0	4	1	3	3	4	1	1	3	2	3	3	3	3	4	1	4	4	4	3
P2	2	0	2	4	4	2	2	2	4	4	1	1	4	2	2	1	4	1	1	3
P3	3	3	0	4	4	4	4	3	2	2	3	2	3	4	3	4	4	1	3	1
P4	3	3	1	0	3	2	4	1	4	2	2	2	1	2	1	2	4	4	3	4
P5	3	4	1	4	0	3	2	1	4	2	4	4	2	2	2	2	2	4	2	1
P6	3	2	2	4	2	0	2	4	2	3	4	3	4	2	3	4	1	2	4	1
P7	4	1	2	3	1	4	0	1	4	4	4	1	1	1	3	1	1	1	3	1
P8	4	1	4	4	4	3	3	0	1	4	3	3	1	3	1	2	3	2	1	4
P9	3	1	4	4	2	1	4	3	0	2	4	1	3	4	4	2	3	3	1	3
P10	1	3	2	3	1	1	2	3	3	0	3	3	1	3	3	2	2	4	1	3
P11	3	4	3	1	4	1	4	4	2	1	0	3	1	3	3	1	4	1	4	3
P12	3	3	2	4	2	2	1	2	1	1	2	0	4	4	4	4	2	2	3	4
P13	2	3	4	1	3	2	1	1	1	3	2	1	0	1	1	3	3	2	4	3
P14	1	2	1	1	3	2	4	4	3	3	2	4	4	0	1	4	1	2	1	2
P15	4	4	2	3	2	3	1	1	1	4	1	3	1	2	0	4	2	3	1	4
P16	1	1	3	2	2	1	4	2	4	2	4	3	4	3	3	0	3	1	3	2
P17	4	1	4	2	2	2	3	3	4	4	1	1	4	3	2	4	0	3	4	4
P18	3	4	3	2	3	3	3	4	4	3	4	1	1	4	4	3	2	0	1	3
P19	4	1	1	2	3	3	2	2	3	3	4	4	2	4	1	4	4	1	0	1
P20	4	2	4	1	1	4	3	3	1	1	3	3	4	4	3	4	4	2	4	0

4.2 Pengolahan Data

1. Pembentukan Kromosom (Populasi)

Populasi adalah bagaimana membuat gen dari kromosom, gen merupakan bagian dari kromosom. Satu gen akan mewakili satu variabel. Agar dapat diproses melalui algoritma genetika, maka alternatif solusi tersebut harus dibuat terlebih dahulu kedalam bentuk kromosom. Masing-masing kromosom berisi sejumlah gen yang menginformasi untuk disimpan didalam individu atau kromosom.

Dalam penelitian ini, kromosom dibentuk dengan menggunakan pendekatan kombinatorial, di mana urutan pekerjaan (*job sequence*) direpresentasikan dengan notasi *job* ke - *n*. Notasi tersebut berfungsi sebagai gen yang kemudian tersusun menjadi satu kesatuan kromosom. Adapun proses pembentukan kromosom dalam penelitian ini dijelaskan pada Tabel 4.3 dibawah ini sebagai berikut:

Tabel 4.3 Populasi

KROMOSOM 1	KROMOSOM 2
1	20
2	13
3	12
4	9
5	6
6	17
7	2
8	7
9	16
10	10
11	1
12	4
13	8
14	19
15	5
16	3
17	11

18	15
19	14
20	18

Sehingga dapat diartikan dalam 1 kromosom terdapat 20 gen yang mewakili urutan tiap *job* pada proses produksi.

2. Menentukan Nilai *Fitness* (Seleksi)

Seleksi merupakan proses pemilihan orang tua untuk reproduksi (biasanya didasarkan pada nilai *fitness*). Seleksi bertujuan untuk memberikan kesem-patan reproduksi yang paling besar bagi anggota populasi yang paling baik.

Setelah populasi awal dibentuk, langkah berikutnya adalah menyeleksi individu (kromosom) yang paling unggul untuk proses reproduksi. Tujuannya agar kromosom dengan tingkat *fitness* lebih baik dan memiliki nilai *makespan* lebih rendah memperoleh peluang lebih besar untuk dipilih dan menghasilkan keturunan baru. Kemudian menentukan nilai *fitness* untuk setiap kromosom. Karena dalam penelitian ini ingin meminimalkan *makespan*, maka kromosom dengan *makespan* yang lebih kecil harus memiliki *fitness* yang lebih tinggi. Berikut rumus menghitung nilai *fitness*:

$$Nilai\ Fitness = \frac{1}{Total\ Processing\ Time}$$

Probabilitas kumulatif adalah total probabilitas dari suatu kromosom ditambah semua probabilitas kromosom sebelumnya. Ini akan menciptakan "rentang" pada roda *roulette* yang digunakan untuk seleksi. Berikut Tabel 4.4 adalah hasil seleksi roda *roulette* dibawah ini:

Tabel 4.4 Seleksi

KROMOSOM	FITNESS	PROBABILITAS	PROBABILITAS KMULATIF
KROMOSOM 1	0.005848	0.468944099	0.468944099
KROMOSOM 2	0.006623	0.531055901	1

Karena Kromosom 2 memiliki probabilitas tertinggi (1), ia memiliki peluang paling besar untuk terpilih.

3. *Crossover* Parsial

Crossover adalah sebuah proses yang membentuk kromosom baru dari dua kromosom induk dengan menggabungkan bagian informasi dari masing-masing kromosom. *Crossover* parsial atau ***Partially Mapped Crossover*** **(PMX)** adalah salah satu teknik *crossover* yang umum digunakan dalam Algoritma Genetika, khususnya untuk masalah optimasi yang melibatkan urutan. Teknik ini bertujuan untuk menghasilkan keturunan yang mewarisi sebagian gen dari kedua induknya sambil mempertahankan *validitas* urutan gen.

menggunakan kromosom 1 dan kromosom 2 sebagai induk. Proses ini akan menghasilkan dua kromosom anak yang valid, yaitu permutasi dari 1 hingga 20. Memilih secara acak dua titik potong yaitu pada indeks ke-5 dan ke-10 yang dijelaskan pada Tabel 4.5 dibawah ini:

Tabel 4.5 Kromosom Induk

Induk 1	[1 2 3 4 5	| 6 7 8 9 10 |	11 12 13 14 15 16 17 18 19 20]
Induk 2	[20 13 12 9 6	| 17 2 7 16 10 |	1 4 8 19 5 3 11 15 14 18]

Berdasarkan Tabel 4.5 diatas segmen tengah dari setiap induk ke kromosom anak masing-masing. Bagian ini juga akan membentuk pemetaan untuk menghindari duplikasi yang dijelaskan pada Tabel 4.6 sebagai berikut:

Tabel 4.6 Kromosom Anak

Anak 1	[? ? ? ? ?	| 17 2 7 16 10 |	? ? ? ? ? ? ? ? ? ?]
Anak 2	[20 13 12 9 6	| 6 7 8 9 10 |	? ? ? ? ? ? ? ? ? ?]

Berdasarkan Tabel 4.6 langkah terakhir isi gen yang kosong pada setiap anak dengan gen dari induk yang sesuai, sambil menggunakan pemetaan untuk memastikan semua gen unik dapat dilihat pada Tabel 4.7 sebagai berikut:

Tabel 4.7 Kromosom Anak Induk 1 & 2

Anak 1 (Anak Induk 1 & 2)	[1 8 3 4 5	| 17 2 7 16 10 |	11 12 13 14 15 9 6 18 19 20]
Anak 2 (Anak Induk 1 & 2)	[20 13 12 16 17	| 6 7 8 9 10 |	1 4 2 19 5 3 11 15 14 18]

Proses pembentukan Anak 1 (Anak induk 1 & 2) dilakukan dengan menyalin segmen tengah dari Parent 2 pada posisi ke-6 hingga ke-10, kemudian mengisi posisi lain menggunakan gen dari Parent 1. Pada saat pengisian, apabila gen yang akan ditempatkan sudah terdapat dalam segmen tengah, maka digunakan aturan pemetaan *(mapping)* sampai diperoleh gen pengganti yang belum terpakai. Misalnya, pada posisi ke-2 gen 2 dari Parent 1 sudah muncul di segmen tengah, sehingga dipetakan ke 7. Karena 7 juga sudah ada, maka dilakukan pemetaan ulang ke 8, yang kemudian ditempatkan pada posisi tersebut. Mekanisme yang sama terjadi pada posisi ke-16, di mana gen 16 dipetakan ke 9 karena sudah terpakai, dan pada posisi ke-17 gen 17 dipetakan ke 6. Dengan prosedur ini, semua posisi dapat terisi tanpa duplikasi sehingga menghasilkan kromosom anak yang valid.

Sementara itu, pembentukan Anak 2 (Anak induk 1 & 2) dimulai dengan menyalin segmen tengah dari Parent 1 pada posisi ke-6 hingga ke-10. Posisi lain diisi dengan gen dari Parent 2 pada indeks yang sama. Jika gen yang akan ditempatkan sudah ada di segmen tengah, maka digunakan pemetaan hingga ditemukan gen pengganti yang belum ada. Contohnya, pada posisi ke-4 gen 9 dari Parent 2 sudah terdapat pada segmen tengah, sehingga dipetakan ke 16 yang kemudian ditempatkan karena belum terpakai. Hal serupa terjadi pada posisi ke-5, di mana gen 6 dipetakan ke 17, dan pada posisi ke-13 gen 8 dipetakan ke 7, lalu ke 2, yang akhirnya ditempatkan karena masih kosong. Dengan langkah-langkah ini, Child 2 terbentuk sebagai kromosom yang tetap memenuhi aturan tanpa adanya gen yang hilang maupun ganda.

4. Mutasi Insersi (*Insertion Mutation*)

Mutasi merupakan proses mengubah secara acak nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi insersi adalah operator mutasi yang digunakan untuk memperkenalkan keragaman genetik dalam populasi Algoritma Genetika. Operator ini bekerja dengan cara menyisipkan gen dari satu posisi ke posisi lain dalam satu kromosom.

Untuk menghitung mutasi memakai kromosom anak dari hasil *crossover*, selanjutnya memilih gen secara acak yaitu 10 pada posisi ke – 10 dan pilih posisi baru secara acak untuk disisipkan antara 12 dan 16 berarti posisis ke – 4, langkah terakhir pindahkan gen 10 dari posisinya semula dan sisipkan pada posisi barunya. Seperti Tabel 4.8 dibawah ini:

Tabel 4.8 Kromosom sebelum dan sesudah mutasi

SESUDAH MUTASI	20 13 12 16 17 6 7 8 9 10 1 4 2 19 5 3 11 15 14 18
SEBELUM MUTASI	20 13 12 1 16 17 6 7 8 9 10 4 2 19 5 3 11 15 14 18

4.2.1 Penentuan solusi terbaik

Hasil penerapan *Evolutionary Algorithm* dalam optimasi penjadwalan produksi menunjukkan adanya penurunan nilai *makespan*. Setelah proses optimasi dilakukan, diperoleh perbedaan *makespan* antara kondisi aktual dan kondisi setelah optimasi. Perbandingan nilai *makespan* sebelum dan sesudah dioptimasi ditampilkan pada Tabel 4.9 dan Tabel 4.10 sebagai berikut :

Tabel 4.9 Sebelum dioptimasi

		M 1			M 2			M 3			M 4			M 5		
Job	Setup	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish
1	0	24	3	27	27	5	32	32	7	39	39	6	45	45	4	49
2	4	27	4	35	35	6	41	41	5	46	46	7	53	53	3	56
3	2	35	3	40	41	5	46	46	6	52	53	6	59	59	5	64
4	4	40	5	49	49	6	55	55	8	63	63	7	70	70	6	76
5	3	49	4	56	56	4	60	63	5	68	70	5	75	76	4	80
6	3	56	3	62	62	5	67	68	7	75	75	6	81	81	4	85
7	2	62	5	69	69	6	75	75	6	81	81	7	88	88	5	93
8	1	69	4	74	75	5	80	81	6	87	88	6	94	94	4	98
9	1	74	4	79	80	6	86	87	7	94	94	5	99	99	3	102
10	2	79	3	84	86	4	90	94	5	99	99	5	104	104	4	108
11	3	84	4	91	91	6	97	99	8	107	107	6	113	113	4	117
12	3	91	3	97	97	5	102	107	6	113	113	6	119	119	3	122
13	4	97	2	103	103	4	107	113	5	118	119	4	123	123	3	126
14	1	103	2	106	107	3	110	118	4	122	123	3	126	126	2	128
15	1	106	3	110	110	5	115	122	6	128	128	5	133	133	3	136
16	4	110	2	116	116	4	120	128	5	133	133	5	138	138	3	141
17	3	116	4	123	123	6	129	133	7	140	140	6	146	146	4	150
18	3	123	2	128	129	3	132	140	4	144	146	3	149	150	2	152
19	1	128	5	134	134	7	141	144	9	153	153	8	161	161	6	167
20	1	134	3	138	141	5	146	153	6	159	161	5	166	167	4	171

Tabel 4.10 Sesudah dioptimasi

		M 1			M 2			M 3			M 4			M 5		
Job	Setup	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish
20	0	14	3	17	17	5	22	22	6	28	28	5	33	33	4	37
13	4	20	2	26	26	4	30	30	5	35	35	4	39	39	3	42
12	1	26	3	30	30	5	35	35	6	41	41	6	47	47	3	50
9	1	30	4	35	35	6	41	41	7	48	48	5	53	53	3	56
6	1	35	3	39	41	5	46	48	7	55	55	6	61	61	4	65
17	1	39	4	44	46	6	52	55	7	62	62	6	68	68	4	72
2	1	44	4	49	52	6	58	62	5	67	68	7	75	75	3	78
7	2	49	5	56	58	6	64	67	6	73	75	7	82	82	5	87
16	1	56	2	59	64	4	68	73	5	78	82	5	87	87	3	90
10	2	59	3	64	68	4	72	78	5	83	87	5	92	92	4	96
1	1	64	3	68	72	5	77	83	7	90	92	6	98	98	4	102
4	3	68	5	76	77	6	83	90	8	98	98	7	105	105	6	111
8	1	76	4	81	83	5	88	98	6	104	105	6	111	111	4	115
19	1	81	5	87	88	7	95	104	9	113	113	8	121	121	6	127
5	3	87	4	94	95	4	99	113	5	118	121	5	126	127	4	131
3	1	94	3	98	99	5	104	118	6	124	126	6	132	132	5	137
11	3	98	4	105	105	6	111	124	8	132	132	6	138	138	4	142
15	3	105	3	111	111	5	116	132	6	138	138	5	143	143	3	146
14	2	111	2	115	116	3	119	138	4	142	143	3	146	146	2	148
18	2	115	2	119	119	3	122	142	4	146	146	3	149	149	2	151

Berdasarkan hasil analisis, diperoleh usulan perbaikan *sequencing* dengan mengubah urutan proses menjadi 20 – 13 – 12 – 9 – 6 – 17 – 2 – 7 – 16 – 10 – 1 – 4 – 8 – 19 – 5 – 3 – 11 – 15 – 14 – 18. Urutan ini menghasilkan makespan sebesar 151 menit lebih kecil dari sebelumnya 171 menit yang berselisih 20 menit.

BAB V

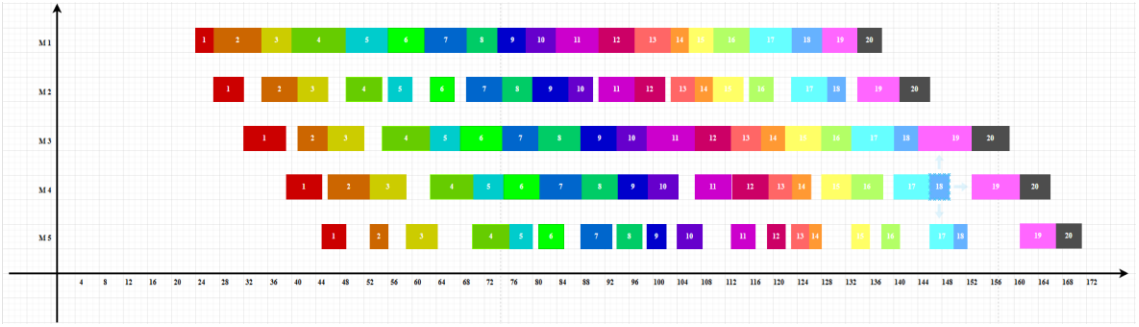
PEMBAHASAN

5.1 Analisis Perbandingan Variabel Keputusan

Berdasarkan hasil perhitungan pada bab sebelumnya, pada bagian ini dilakukan analisis perbedaan antara dua model yang diperoleh. Kromosom pertama adalah kromosom urutan awal, yang merepresentasikan kondisi sebelum dilakukan optimasi. Sedangkan kromosom kedua adalah kromosom urutan akhir, yang menggambarkan urutan yang sudah dilakukan optimasi untuk kedua kromosom tersebut.

Tabel 5.1 Sebelum dioptimasi

		M 1			M 2			M 3			M 4			M 5		
Job	Setup	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish
1	0	24	3	27	27	5	32	32	7	39	39	6	45	45	4	49
2	4	27	4	35	35	6	41	41	5	46	46	7	53	53	3	56
3	2	35	3	40	41	5	46	46	6	52	53	6	59	59	5	64
4	4	40	5	49	49	6	55	55	8	63	63	7	70	70	6	76
5	3	49	4	56	56	4	60	63	5	68	70	5	75	76	4	80
6	3	56	3	62	62	5	67	68	7	75	75	6	81	81	4	85
7	2	62	5	69	69	6	75	75	6	81	81	7	88	88	5	93
8	1	69	4	74	75	5	80	81	6	87	88	6	94	94	4	98
9	1	74	4	79	80	6	86	87	7	94	94	5	99	99	3	102
10	2	79	3	84	86	4	90	94	5	99	99	5	104	104	4	108
11	3	84	4	91	91	6	97	99	8	107	107	6	113	113	4	117
12	3	91	3	97	97	5	102	107	6	113	113	6	119	119	3	122
13	4	97	2	103	103	4	107	113	5	118	119	4	123	123	3	126
14	1	103	2	106	107	3	110	118	4	122	123	3	126	126	2	128
15	1	106	3	110	110	5	115	122	6	128	128	5	133	133	3	136
16	4	110	2	116	116	4	120	128	5	133	133	5	138	138	3	141
17	3	116	4	123	123	6	129	133	7	140	140	6	146	146	4	150
18	3	123	2	128	129	3	132	140	4	144	146	3	149	150	2	152
19	1	128	5	134	134	7	141	144	9	153	153	8	161	161	6	167
20	1	134	3	138	141	5	146	153	6	159	161	5	166	167	4	171



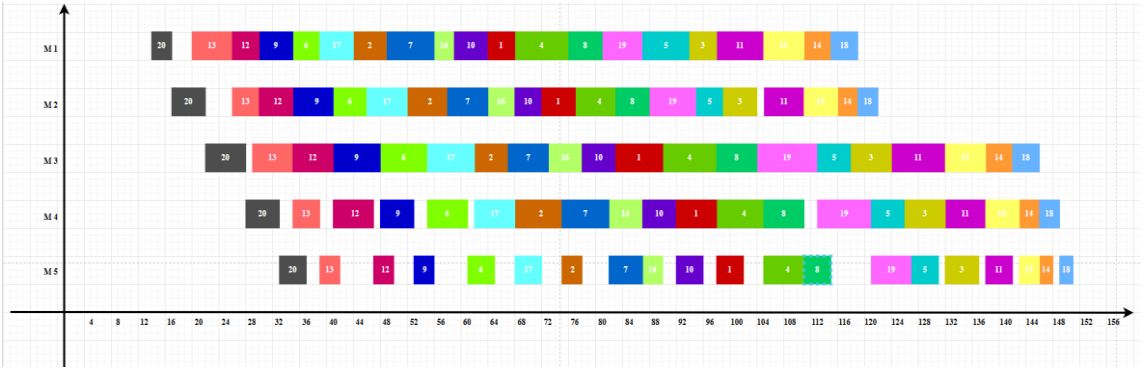
Gambar 5.1 Sebelum dioptimasi

Pada kondisi sebelum optimasi, urutan pekerjaan masih mengikuti jadwal aktual perusahaan tanpa mempertimbangkan secara menyeluruh pengaruh *setup time* yang bergantung pada urutan maupun batasan *no-wait flowshop*. Hal ini mengakibatkan adanya gap atau jeda yang cukup besar antara waktu selesai pada suatu mesin dengan waktu

mulai di mesin berikutnya. Dampaknya, *waiting time* yang terjadi relatif tinggi, *utilisasi* mesin menjadi tidak seimbang karena beberapa mesin harus menunggu lama sebelum dapat memproses pekerjaan berikutnya, serta nilai *makespan* atau total waktu penyelesaian seluruh pekerjaan menjadi lebih panjang. Selain itu, setup antar pekerjaan masih cukup tinggi karena urutan *job* belum diatur untuk meminimalkan efek *sequence-dependent setup time (SDST)*.

Tabel 5.2 Sesudah dioptimasi

		M 1			M 2			M 3			M 4			M 5		
Job	Setup	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish
20	0	14	3	17	17	5	22	22	6	28	28	5	33	33	4	37
13	4	20	2	26	26	4	30	30	5	35	35	4	39	39	3	42
12	1	26	3	30	30	5	35	35	6	41	41	6	47	47	3	50
9	1	30	4	35	35	6	41	41	7	48	48	5	53	53	3	56
6	1	35	3	39	41	5	46	48	7	55	55	6	61	61	4	65
17	1	39	4	44	46	6	52	55	7	62	62	6	68	68	4	72
2	1	44	4	49	52	6	58	62	5	67	68	7	75	75	3	78
7	2	49	5	56	58	6	64	67	6	73	75	7	82	82	5	87
16	1	56	2	59	64	4	68	73	5	78	82	5	87	87	3	90
10	2	59	3	64	68	4	72	78	5	83	87	5	92	92	4	96
1	1	64	3	68	72	5	77	83	7	90	92	6	98	98	4	102
4	3	68	5	76	77	6	83	90	8	98	98	7	105	105	6	111
8	1	76	4	81	83	5	88	98	6	104	105	6	111	111	4	115
19	1	81	5	87	88	7	95	104	9	113	113	8	121	121	6	127
5	3	87	4	94	95	4	99	113	5	118	121	5	126	127	4	131
3	1	94	3	98	99	5	104	118	6	124	126	6	132	132	5	137
11	3	98	4	105	105	6	111	124	8	132	132	6	138	138	4	142
15	3	105	3	111	111	5	116	132	6	138	138	5	143	143	3	146
14	2	111	2	115	116	3	119	138	4	142	143	3	146	146	2	148
18	2	115	2	119	119	3	122	142	4	146	146	3	149	149	2	151



Gambar 5.2 Sesudah dioptimasi

Sementara itu, pada kondisi sesudah optimasi, urutan *job* diatur ulang sehingga aliran proses antar mesin menjadi lebih efisien. Perubahan ini menghasilkan penurunan makespan yang signifikan karena total waktu penyelesaian seluruh pekerjaan menjadi lebih singkat dibanding kondisi awal. Selain itu, *waiting time* juga berkurang dari kondisi awal, yang menunjukkan bahwa jadwal baru mampu mengurangi jeda antar mesin. *Utilisasi* mesin pun menjadi lebih seimbang, dengan *idle time* yang lebih kecil, sehingga beban kerja dapat tersebar lebih merata. Di sisi lain, pemilihan urutan pekerjaan yang tepat membuat beberapa *setup time* dapat diminimalkan, sehingga alur produksi menjadi

lebih lancar. Dengan demikian, optimasi terbukti memberikan peningkatan kinerja sistem produksi baik dari segi waktu penyelesaian, pemanfaatan mesin, maupun efisiensi proses secara keseluruhan.

5.2 Reaktif *Scheduling*

Dalam praktik di lantai produksi Perusahaan Elektronik X, selain permasalahan penjadwalan utama yang dibahas pada penelitian ini, terdapat pula kondisi nyata di mana order baru dari customer dapat masuk setelah jadwal produksi terbentuk. Order ini sering kali datang dari *customer prioritas* yang menuntut pelayanan cepat. Hal ini menimbulkan dilema dalam penjadwalan, yaitu apakah order baru tersebut ditempatkan di akhir setelah seluruh job yang sudah terjadwal selesai dikerjakan, ataukah disisipkan di tengah jadwal produksi yang sedang berjalan. Apabila order ditempatkan di akhir jadwal, maka terdapat risiko ketidakpuasan dari *customer prioritas* karena harus menunggu terlalu lama. Sebaliknya, jika order baru disisipkan di tengah jadwal, maka perlu ditentukan di posisi mana order tersebut dapat dimasukkan agar tidak mengganggu stabilitas jadwal yang sudah terbentuk. Kondisi inilah yang dikenal dengan konsep reaktif *scheduling*, yaitu kemampuan sistem penjadwalan untuk menyesuaikan diri terhadap perubahan atau gangguan tak terduga, termasuk kedatangan order baru yang mendesak.

Namun demikian, penelitian ini tidak mengimplementasikan konsep reaktif *scheduling* secara langsung karena ruang lingkup penelitian hanya terbatas pada penjadwalan *no-wait flowshop* dengan *setup time* bergantung pada urutan pekerjaan *(sequence-dependent setup time*). Oleh sebab itu, isu mengenai reaktif *scheduling* hanya diangkat sebagai catatan tambahan untuk menunjukkan adanya tantangan nyata di perusahaan yang relevan untuk penelitian lanjutan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penggunaan Algoritma Genetika berhasil menyusun urutan produksi yang lebih efisien pada sistem *no-wait flowshop* dengan *sequence-dependent setup time*. Hasil optimasi dergan mengubah urutan urutan proses menjadi 20 – 13 – 12 – 9 – 6 – 17 – 2 – 7 – 16 – 10 – 1 – 4 – 8 – 19 – 5 – 3 – 11 – 15 – 14 – 18. Urutan ini menghasilkan *makespan* sebesar 151 menit lebih kecil dari sebelumnya 171 menit yang berselisih 20 menit yang menunjukkan adanya penurunan *makespan* dan *waiting time* dibandingkan kondisi aktual, sehingga proses produksi modul kontrol elektronik dapat berjalan lebih cepat, lancar, dan seimbang antar mesin tanpa hambatan yang berarti.

6.2 Saran

Berdasarkan hasil penelitian dan kesimpulan yang telah diperoleh, peneliti memberikan saran untuk penelitian selanjutnya dengan mempertimbangkan beberapa aspek berikut:

1. Penelitian lanjutan dapat mengombinasikan Algoritma Evolusi dengan metode optimasi lain, seperti *Simulated Annealing* atau *Ant Colony Optimization*, untuk memperoleh hasil yang lebih optimal.
2. Penelitian berikutnya disarankan mempertimbangkan variabel tambahan seperti biaya produksi, keterlambatan (*tardiness*), dan utilisasi mesin agar hasil lebih komprehensif.
3. Penelitian selanjutnya disarankan untuk memperhatikan aspek reaktif *scheduling*, khususnya terkait penanganan order baru dari *customer prioritas* yang datang setelah jadwal produksi terbentuk. Hal ini penting karena keputusan mengenai penempatan order baru dan apakah dikerjakan di akhir atau disisipkan di tengah jadwal sehingga dapat memengaruhi kepuasan *customer* dan efisiensi sistem produksi secara keseluruhan.
4. Hasil penelitian ini sebaiknya dikembangkan menjadi sistem pendukung keputusan *(Decision Support System)* berbasis perangkat lunak sehingga dapat langsung diimplementasikan oleh perusahaan manufaktur.

DAFTAR PUSTAKA

Alex Saleh, E. Z. (2009). MODEL PENJADWALAN PADA FLOWSHOP-4-STAGE DENGAN KRITERIA MINIMISASI LATENESS MAKSIMUM DAN JUMLAH TARDY JOBS . *Jurnal Teknik Industri*, Vol. 11, No. 2, Desember 2009, pp. 148-162 .

Ali Allahverdi, J. G. (1999). A review of scheduling research involving setup considerations. *Omega – International Journal of Management Science*, 219–239.

Ali Allahverdia, H. A. (2022). An algorithm for a no-wait flowshop scheduling problem for minimizing total tardiness with a constraint on total completion time. *International Journal of Industrial Engineering Computations* , 13 (2022) 43–50 .

Allahverdi, A. &. (2014). Total completion time with makespan constraint in no-wait flowshops with setup times. *European Journal of Operational Research*, 724–734.

Baker, K. R. (2009). Principles of Sequencing and Scheduling. *Wiley*.

Bianco, L. D. (1999). Flow shop no-wait scheduling with sequence dependent setup times and release dates. *INFOR: Information Systems and Operational Research*, 37(1), 3–19.

Chen-Yang Cheng, K.-C. Y.-C. (2019). Minimizing makespan in mixed no-wait flowshops with sequence-dependent setup times. *Computers & Industrial Engineering*, 338–347.

Dian Setiya Widodo. (2018). Improve Algoritma Hodgson Untuk Meminimasi Jumlah Job Terlambat Pada Penjadwalan Flow shop . *Jurnal Teknik Industri*, , Vol. 19, No. 1, Februari 2018, pp. 73-81, ISSN 2527-4112 online.

Garside, A. K. (2019). Pengembangan Algoritma Hybrid Flowshop Three Stage Dengan Mempertimbangkan Waktu Setup. *JURNAL ILMIAH TEKNIK INDUSTRI* , e-ISSN 2460-4038 .

Ginting, K. F. (2019). Penjadwalan Produksi untuk Penjadwalan Flexible Flowshop dengan Menggunakan Metode Longest Process Time (LPT). *TALENTA Conference Series*, Volume 2 Issue 3 – 2019.

Hebrard, D. G. (2011). Models and Strategies for Variants of the Job Shop Scheduling Problem.

Herry Christian Palit. (2003).

Hervin Rizky Pratama, I. N. (2024). ANALISIS KETERAMPILAN KOLABORASI SISWA SEKOLAH DASAR MENGGUNAKAN MEDIA DIORAMA PADA PEMBELAJARAN MATERI EKOSISTEM. *Jurnal Ilmiah Pendidikan Dasar*, ISSN Cetak : 2477-2143 ISSN Online : 2548-6950.

Jain, A. K. (2014). An improved genetic algorithm for job shop scheduling problem with makespan minimization. *Journal of Industrial Engineering International*, 10(1), 1-10.

Jan Riezebos, G. J. (1995). Flow shop scheduling with multiple operations and time lags. *Journal of Intelligent Manufacturing*, 105–115.

Jarosław Pempera, J. &. (2007). Some local search algorithms for no-wait flow-shop problem with makespan criterion. *Computers & Operations Research*, 2197–2212.

Jun Pei, X. L. (2018). Single-machine scheduling with learning effect and resource-dependent processting time in the serial basth production. *Applied Mathematical Modelling* , 58 (2018) 245–253 .

Karande, P. B. (2023). Hybrid genetic algorithm to minimize scheduling cost with unequal and job dependent earliness tardiness cost. *International Journal of Production Management and Engineering*, 19–30.

Kusumadewi, . (2005). Artificial Intelligence (Teknik dan Aplikasinya). *Graha Ilmu, Yogyakarta*.

Lely Herlina, M. A. (2021). Model optimasi penjadwalan pallet dan dunnage untuk meminimasi makespan . *Journal Industrial Servicess* , Vol. 7, No. 1, Oktober 2021.

Mousavi, S. M. (2018). An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times. *Operational Research*, 123–158.

Mucha, M. &. (2013). No-Wait Flowshop Scheduling is as Hard as Asymmetric Traveling Salesman Problem. *ArXiv*.

Nainggolan, S. B. (2018). Integrasi Penjadwalan Produksi dan Preventive Maintenance untuk Meminimasi Makespan dengan Menggunakan Metode Heijunka dan Batch – Backward Scheduling (Studi Kasus PT. BMC) . *Jurnal Teknik Industri* , Vol. 8 No. 3 ISSN 2622-5131 (Online) .

Neufeld, J. S. (2015). Minimising makespan in flowshop group scheduling with sequence-dependent family set-up times using inserted idle times. *International Journal of Production Research*, 7(3), 291–307.

Palit, H. C., Soetanto, T. V., & Novianti, H. (2004). Penjadwalan produksi flexible flowshops dengan sequence-dependent setup times menggunakan metode relaksasi Lagrangian (Studi Kasus pada PT. Cahaya Angkasa Abadi). *Jurnal Teknik Industri*, 111–119.

Pinedo, M. L. (2012). Scheduling: Theory, Algorithms, and Systems. *Spinger*.

Rubén Ruiz, C. M. (2003). Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, 34–54.

Ruiz, R., Maroto, C., & Alcaraz, J. (2005). Solving the flow-shop scheduling problem with sequence-dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, 34–54.

Sari, P. &. (2007). ANALISIS PERBANDINGAN PADA PENJADWALAN JOB UNTUK KETEPATAN WAKTU PROSES. *Jurnal Teknik Industri*, Vol. 8, No. 1, Februari 2007: 12–22.

Subanar, Z. I. (2014). Optimasi Biaya Distribusi Rantai Pasok Tiga Tingkat dengan Menggunakan Algoritma Genetika Adaptif dan Terdistribusi. *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, 1–11 .

Urip Sarwo Sambodo, Y. d. (2003). Penjadwalan Pekerjaan Pada No-Wait Flowshop dengan Kriteria Meminimasi Total Waktu Aktual Menggunakan Algoritma Genetika. Vol, 2, No1:24-30.

Wang, S. W. (2019). Two-stage no-wait hybrid flow-shop scheduling with sequence-dependent setup times. *nternational Journal of Systems Science: Operations & Logistics*, 7(3), 291–307.

Xinna Geng, J. W.-J. (2018). Flow Shop Scheduling Problem with Convex Resource Allocation and Learning Effect. *Journal of Computer and Communications*, 2018, 6, 239-246 .

Yuniaristanto. (2003). Penjadwalan Pekerjaan Pada No-Wait Flowshop dengan Pembatas Common Due-Date. Vol, 2, No1:1-5.

Zaini, M. A. (2023). OPTIMASI NO-WAIT FLOWSHOP SCHEDULING MENGGUNAKAN METODE ALGORITMA GENETIKA.

Zong-Jun Wei, L.-Y. W.-B. (2023). Single-Machine Maintenance Activity Scheduling with Convex Resource Constraints and Learning Effects. *Mathematics*, 3536.

LAMPIRAN

- Populasi

KROMOSOM 1	KROMOSOM 2
1	20
2	13
3	12
4	9
5	6
6	17
7	2
8	7
9	16
10	10
11	1
12	4
13	8
14	19
15	5
16	3
17	11
18	15
19	14
20	18

- Seleksi

KROMOSOM	FITNESS	PROBABILITAS	PROBABILITAS KMULATIF
KROMOSOM 1	0.00585	0.468944099	0.468944099
KROMOSOM 2	0.00662	0.531055901	1

- Crossover

INDUK 1	INDUK 2
1	20
2	13
3	12
4	9
5	6
6	17
7	2
8	7
9	16
10	10
11	1
12	4
13	8
14	19
15	5
16	3
17	11
18	15
19	14
20	18

ANAK 1	ANAK 2
?	20
?	13
?	12
?	9
?	6
6	17
7	2
8	7
9	16
10	10
?	?
?	?
?	?
?	?
?	?
?	?
?	?
?	?
?	?
?	?

ANAK 1 INDUK 1&2	ANAK 2 INDUK 1&2
1	20
8	13
3	12
4	16
5	17
17	6
2	7
7	8
16	9
10	10
11	1
12	4
13	2
14	19
15	5
9	3
6	11
18	15
19	14
20	18

- Mutasi

SEBELUM MUTASI	SESUDAH MUTASI
20	20
13	13
12	12
16	1
17	16
6	17
7	6
8	7
9	8
10	9
1	10
4	4
2	2
19	19
5	5
3	3
11	11
15	15
14	14
18	18

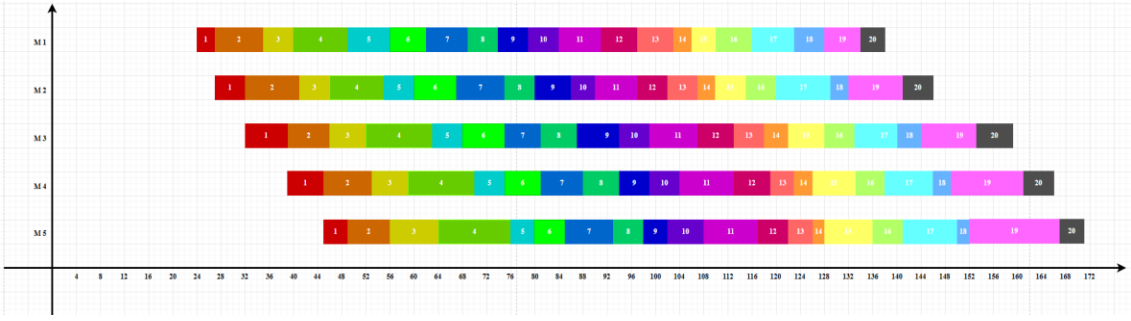
- SDST Sebelum

		M 1			M 2			M 3			M 4			M 5		
Job	Setup	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish
1	0	24	3	27	27	5	32	32	7	39	39	6	45	45	4	49
2	4	27	4	35	35	6	41	41	5	46	46	7	53	53	3	56
3	2	35	3	40	41	5	46	46	6	52	53	6	59	59	5	64
4	4	40	5	49	49	6	55	55	8	63	63	7	70	70	6	76
5	3	49	4	56	56	4	60	63	5	68	70	5	75	76	4	80
6	3	56	3	62	62	5	67	68	7	75	75	6	81	81	4	85
7	2	62	5	69	69	6	75	75	6	81	81	7	88	88	5	93
8	1	69	4	74	75	5	80	81	6	87	88	6	94	94	4	98
9	1	74	4	79	80	6	86	87	7	94	94	5	99	99	3	102
10	2	79	3	84	86	4	90	94	5	99	99	5	104	104	4	108
11	3	84	4	91	91	6	97	99	8	107	107	6	113	113	4	117
12	3	91	3	97	97	5	102	107	6	113	113	6	119	119	3	122
13	4	97	2	103	103	4	107	113	5	118	119	4	123	123	3	126
14	1	103	2	106	107	3	110	118	4	122	123	3	126	126	2	128
15	1	106	3	110	110	5	115	122	6	128	128	5	133	133	3	136
16	4	110	2	116	116	4	120	128	5	133	133	5	138	138	3	141
17	3	116	4	123	123	6	129	133	7	140	140	6	146	146	4	150
18	3	123	2	128	129	3	132	140	4	144	146	3	149	150	2	152
19	1	128	5	134	134	7	141	144	9	153	153	8	161	161	6	167
20	1	134	3	138	141	5	146	153	6	159	161	5	166	167	4	171

- SDST Sesudah

		M 1			M 2			M 3			M 4			M 5		
Job	Setup	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish	Start	Duration	Finish
20	0	14	3	17	17	5	22	22	6	28	28	5	33	33	4	37
13	4	20	2	26	26	4	30	30	5	35	35	4	39	39	3	42
12	1	26	3	30	30	5	35	35	6	41	41	6	47	47	3	50
9	1	30	4	35	35	6	41	41	7	48	48	5	53	53	3	56
6	1	35	3	39	41	5	46	48	7	55	55	6	61	61	4	65
17	1	39	4	44	46	6	52	55	7	62	62	6	68	68	4	72
2	1	44	4	49	52	6	58	62	5	67	68	7	75	75	3	78
7	2	49	5	56	58	6	64	67	6	73	75	7	82	82	5	87
16	1	56	2	59	64	4	68	73	5	78	82	5	87	87	3	90
10	2	59	3	64	68	4	72	78	5	83	87	5	92	92	4	96
1	1	64	3	68	72	5	77	83	7	90	92	6	98	98	4	102
4	3	68	5	76	77	6	83	90	8	98	98	7	105	105	6	111
8	1	76	4	81	83	5	88	98	6	104	105	6	111	111	4	115
19	1	81	5	87	88	7	95	104	9	113	113	8	121	121	6	127
5	3	87	4	94	95	4	99	113	5	118	121	5	126	127	4	131
3	1	94	3	98	99	5	104	118	6	124	126	6	132	132	5	137
11	3	98	4	105	105	6	111	124	8	132	132	6	138	138	4	142
15	3	105	3	111	111	5	116	132	6	138	138	5	143	143	3	146
14	2	111	2	115	116	3	119	138	4	142	143	3	146	146	2	148
18	2	115	2	119	119	3	122	142	4	146	146	3	149	149	2	151

- GanttChart Sebelum



- GanttChart Sesudah

