

SKRIPSI

**MULTI OBJECTIVE FLOW SHOP SCHEDULING PROBLEM
DENGAN WHALE OPTIMIZATION ALGORITHM**



Craven Sachio Saputra

NPM: 6181801063

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2022**

UNDERGRADUATE THESIS

**MULTI OBJECTIVE FLOW SHOP SCHEDULING PROBLEM
WITH WHALE OPTIMIZATION ALGORITHM**



Craven Sachio Saputra

NPM: 6181801063

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2022**

LEMBAR PENGESAHAN

**MULTI OBJECTIVE FLOW SHOP SCHEDULING PROBLEM
DENGAN WHALE OPTIMIZATION ALGORITHM**

Craven Sachio Saputra

NPM: 6181801063

Bandung, 27 Juni 2022

Menyetujui,

Pembimbing

**Digitally signed
by Cecilia Esti
Nugraheni**

Dr.rer.nat. Cecilia Esti Nugraheni

**Ketua Tim Penguji
Digitally signed
by Luciana
Abednego**

Luciana Abednego, M.T.

**Anggota Tim Penguji
Digitally signed
by Husnul
Hakim**

Husnul Hakim, M.T.

Mengetahui,

**Ketua Program Studi
Digitally signed
by Mariskha Tri
Adithia**

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

MULTI OBJECTIVE FLOW SHOP SCHEDULING PROBLEM DENGAN WHALE OPTIMIZATION ALGORITHM

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 27 Juni 2022



Craven Sachio Saputra
NPM: 6181801063

ABSTRAK

Penjadwalan adalah proses perencanaan dalam mengalokasikan sumber yang dimiliki untuk menyelesaikan pekerjaan. Salah satu penerapan penjadwalan yang sering ditemui adalah penjadwalan produksi untuk menjadwalkan operasi pekerjaan dalam produksi barang. Penjadwalan produksi yang baik diperlukan untuk meminimalkan waktu produksi dan sumber daya yang digunakan. Untuk mencapai penjadwalan produksi yang baik tersebut, dapat digunakan objektif-objektif untuk mengoptimasi penjadwalan, seperti dengan mengurangi waktu total yang dibutuhkan dalam produksi, yang biasa disebut dengan *makespan*.

Salah satu dari permasalahan penjadwalan produksi adalah penjadwalan *flowshop* yang cocok untuk produksi barang secara massal. Pada penjadwalan *flowshop*, terdapat serangkaian pekerjaan yang terdiri dari beberapa proses pengerjaan, yang akan dikerjakan oleh mesin-mesin yang ada. Semua pekerjaan memiliki urutan pengerjaan operasi pekerjaan yang sama.

Pada skripsi ini, permasalahan penjadwalan *flowshop* ini diselesaikan dengan meminimasi dua buah objektif, yakni *makespan* dengan tambahan objektif *total flow-time*, yakni merupakan total waktu yang dibutuhkan oleh setiap pekerjaan dihitung dari mulainya penjadwalan / pekerjaan pertama dimulai. Penjadwalan yang dibuat menerapkan algoritma metaheuristik yang dikembangkan dari 2016, yakni *Whale Optimization Algorithm*. Untuk menguji efektivitas dan kualitas dari hasil penjadwalan oleh *Whale Optimization Algorithm*, dilakukan pengujian dengan membandingkan hasil dari *Whale Optimization Algorithm* dengan hasil yang didapatkan oleh *Bi-objective Multi-start Simulated-annealing Algorithm* yang juga merupakan algoritma metaheuristik. Selain itu, dilakukan juga pengujian untuk mencari pengaruh dari *parameter* yang digunakan, yakni jumlah paus dan jumlah iterasi maksimum. Semua pengujian menggunakan kasus *flowshop* yang didapatkan dari *Tailard's Benchmark*.

Hasil dari pengujian pengaruh *parameter* yang sudah dilakukan menunjukkan bahwa semakin besar nilai dari *parameter* yang digunakan, hasil dari optimasi penjadwalan semakin baik. Hasil dari pengujian kualitas yang sudah dilakukan menunjukkan bahwa hasil penjadwalan yang dihasilkan oleh *Whale Optimization Algorithm* tidak lebih optimal dibandingkan dengan hasil penjadwalan yang dihasilkan oleh *Bi-objective Multi-start Simulated-annealing Algorithm*.

Kata-kata kunci: penjadwalan, *flowshop*, *whale optimization algorithm*

ABSTRACT

Scheduling is a planning process in allocationing sources owned to accomplish job. One of the application of scheduling often found is production scheduling to arrange operations of a job in goods production. A good production scheduling is necessary to minimize production time and resources used. To achieve a good production scheduling, objectives for optimized scheduling can be used, for instance, with minimizing total of times needed on production, commonly called makespan.

One of the production scheduling problem is flowshop scheduling that suits for mass production of products. In flowshop scheduling, there is a series of jobs consist of several processing operations, that will be done by existing machines. All of the jobs have the same sequence of processing operations.

In this thesis, the flowshop scheduling problem is solved by minimizing two objectives, which is makespan with additional total flow-time, which is a total time needed from each works counted from the start of the schedule / first work. This scheduling applies a metaheuristic algorithm that have been developed since 2016, Whale Optimization Algorithm. To test the effectiveness and quality of the scheduling results by Whale Optimization Algorithm, testings are conducted by comparing results gathered by Whale Optimization Algorithm with results gathered by Bi-objective Multi-start Simulated-annealing Algorithm which also is a metaheuristic algorithm. Moreover, more testings are conducted to search for effects from the parameters used, which is the amount of whales used and the amount of maximum iterations used. All of the testings cases used flowshop cases obtained from Tailard's Benchmark.

Results obtained from the testings to search for effects from the parameters used, shows that the greater the value of the parameter used, the better the scheduling optimization will be. Results from the testings to test the effectiveness and quality of the scheduling results that are done, shows that the schedules obtained by the Whale Optimization Algorithm is not more optimal than the schedules obtained by the Bi-objective Multi-start Simulated-annealing Algorithm.

Keywords: scheduling, flowshop, whale optimization algorithm

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus atas berkat dan penyertaan yang diberikan kepada penulis sehingga skripsi yang berjudul “Multi Objective Flow Shop Scheduling Problem dengan Whale Optimization Algorithm” dapat diselesaikan dengan baik. Penulis menyadari bahwa dalam proses penyusunan skripsi ini, telah banyak pihak yang memberikan dukungan. Oleh karena itu, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan Yesus Kristus, Allah Bapa di Sorga, dan Roh Kudus yang selalu memberikan hikmat dan kekuatan kepada penulis
2. Keluarga, terutama orang tua penulis yang selalu memberikan doa serta dukungan selama proses perkuliahan dan pengerjaan skripsi
3. Ibu Dr.rer.nat. Cecilia Esti Nugraheniselaku dosen pembimbing yang telah membimbing penulis dengan sabar dalam proses pengerjaan skripsi
4. Ibu Luciana Abednego, M.T.selaku ketua tim penguji dan Bapak Husnul Hakim, M.T.selaku anggota tim penguji yang telah membantu dalam proses sidang serta memberikan saran-saran untuk skripsi
5. Dosen Teknik Informatika Universitas Katolik Parahyangan yang telah memberikan ilmu selama masa perkuliahan
6. Winda Aurelia Muljadi, Calvin Rynaldi Kosasih, Johan Adi Wiguna, Warren Mazmur, Jiang Han, Vicky Ricardo Savero, Richard Lionggiwijaya, Alfonsius Geraldi, Alvin Satrio Wibowo, serta teman-teman penulis lainnya yang telah memberikan doa serta dukungan selama proses perkuliahan dan pengerjaan skripsi

Penulis juga menyadari bahwa skripsi ini masih jauh dari kata sempurna. Maka dari itu, segala bentuk kritik dan saran akan sangat diharapkan dan dihargai. Akhir kata, penulis berharap skripsi ini dapat bermanfaat bagi banyak pihak.

Bandung, Juni 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Metodologi	4
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 Penjadwalan Produksi	5
2.2 Flowshop Scheduling	6
2.2.1 Definisi Umum [1]	6
2.2.2 Multi-Objective Flowshop Scheduling	8
2.3 <i>Whale Optimization Algorithm</i> [2]	10
2.3.1 <i>Encircling Prey</i> [2]	12
2.3.2 <i>Bubble-net Attacking Method</i> [2]	13
2.3.3 <i>Search for Prey</i> [2]	14
2.4 Tailard's Benchmark	15
3 ANALISIS MASALAH DAN PERANGKAT LUNAK	17
3.1 Analisis Kasus	17
3.2 Pemodelan Ikan Paus	19
3.3 Pencarian Solusi Menggunakan Paus	19
3.4 Pembentukan Populasi Awal	20
3.5 Menghitung Nilai <i>Multicriterion</i> Setiap Paus	21
3.6 Mengganti Posisi Setiap Paus Sesuai dengan Persamaan yang Terpilih	21
3.7 Analisis Perangkat Lunak	22
3.7.1 Analisis Input dan Output	22
3.7.2 Analisis Diagram Kelas	22
3.7.3 Analisis Sequence Diagram	23
4 PERANCANGAN PERANGKAT LUNAK	25
4.1 Perancangan Input dan <i>Output</i>	25
4.2 Perancangan Tampilan Antarmuka	25
4.3 Diagram Kelas Rinci	26
4.4 <i>Pseudocode</i>	30

5	IMPLEMENTASI DAN PENGUJIAN	35
5.1	Lingkungan Pengujian	35
5.2	Implementasi Perangkat Lunak	35
5.3	Implementasi Antarmuka Perangkat Lunak	35
5.4	Pengujian	37
5.4.1	Pengujian Eksperimental	37
5.5	Analisis dan Kesimpulan Pengujian Eksperimental	48
6	KESIMPULAN DAN SARAN	49
6.1	Kesimpulan Penelitian	49
6.2	Saran Penelitian Lanjutan	50
	DAFTAR REFERENSI	51
	A KODE PROGRAM	53
	B HASIL EKSPERIMEN	61
	C DAFTAR ISTILAH	85

DAFTAR GAMBAR

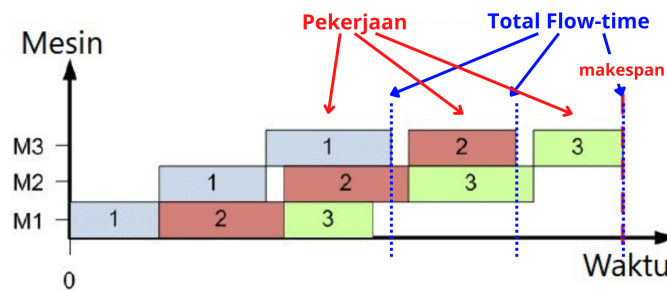
1.1	Contoh dari <i>flowshop scheduling</i>	1
2.1	Contoh dari <i>jobshop scheduling</i>	6
2.2	Contoh dari <i>flowshop scheduling</i>	6
2.3	Struktur preseden dari sebuah pekerjaan pada <i>flowshop</i> . [1]	7
2.4	Ilustrasi <i>flowshop</i>	8
2.5	Perilaku <i>bubble-net feeding</i> dari paus bungkuk. [2]	11
2.6	Vektor posisi pada 2D dan 3D beserta posisi yang mungkin. [2]	13
2.7	Kedua mekanisme pencarian <i>bubble-net</i> yang diimplementasikan pada WOA: (a) <i>Shrinking encircling mechanism</i> (b) <i>Spiral Updating Position</i> . [2]	14
2.8	Mekanisme eksplorasi yang diimplementasikan pada WOA. [2]	15
2.9	Contoh dari data kasus <i>Tailard's Benchmark</i>	15
3.1	Contoh dari pemodelan paus.	19
3.2	Diagram alir dari WOA.	20
3.3	Proses perbaharuan posisi baru untuk <i>whale</i>	21
3.4	Diagram Kelas.	22
3.5	Diagram <i>sequence</i> untuk memilih <i>file</i> kasus <i>flowshop</i>	24
4.1	Contoh dari input kasus <i>flowshop</i>	25
4.2	<i>Mockup</i> dari perangkat lunak.	26
4.3	Diagram kelas rinci.	27
5.1	Tampilan awal perangkat lunak.	36
5.2	Tampilan saat memilih <i>file</i> untuk perangkat lunak.	36
5.3	Tampilan saat sudah mengisi semua <i>parameter</i> untuk perangkat lunak.	37
5.4	Tampilan perangkat lunak saat menampilkan optimasi penjadwalan.	37
5.5	Diagram batang perbandingan setiap iterasi pada kasus 20 pekerjaan 5 mesin.	39
5.6	Diagram batang perbandingan setiap iterasi pada kasus 20 pekerjaan 10 mesin.	39
5.7	Diagram batang perbandingan setiap iterasi pada kasus 20 pekerjaan 20 mesin.	40
5.8	Diagram batang perbandingan setiap iterasi pada kasus 50 pekerjaan 5 mesin.	41
5.9	Diagram batang perbandingan setiap iterasi pada kasus 50 pekerjaan 10 mesin.	41
5.10	Diagram batang perbandingan setiap iterasi pada kasus 50 pekerjaan 20 mesin.	42
5.11	Diagram batang perbandingan setiap paus pada kasus 20 pekerjaan 5 mesin.	43
5.12	Diagram batang perbandingan setiap paus pada kasus 20 pekerjaan 10 mesin.	44
5.13	Diagram batang perbandingan setiap paus pada kasus 20 pekerjaan 20 mesin.	44
5.14	Diagram batang perbandingan setiap paus pada kasus 50 pekerjaan 5 mesin.	45
5.15	Diagram batang perbandingan setiap paus pada kasus 50 pekerjaan 10 mesin.	46
5.16	Diagram batang perbandingan setiap paus pada kasus 50 pekerjaan 20 mesin.	46

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini, dunia sudah memasuki era revolusi industri 4.0, termasuk negara Indonesia. Industri 4.0 mencakup sistem *cyber-physical*, internet untuk segala, komputasi awan, dan komputasi kognitif. Hal-hal tersebut yang memungkinkan terbentuknya sebuah pabrik cerdas yang mengoptimalkan komputasi serta otomasi dalam industri. Banyak industri di Indonesia yang sudah banyak memanfaatkan sistem robotik dan infrastruktur internet untuk segala dalam pemrosesan produksi. Dengan memanfaatkan mesin-mesin yang ada dalam proses produksi, penting bagi industri untuk melakukan penjadwalan pekerjaan-pekerjaan tersebut sehingga didapatkan hasil pekerjaan yang maksimal dalam suatu batas waktu tertentu. Permasalahan ini dikenal dengan *Optimal Job Scheduling* yang pada umumnya memiliki tujuan untuk mencari solusi penjadwalan terbaik dari semua kemungkinan solusi penjadwalan yang terdiri dari sebuah daftar pekerjaan (*jobs*) yang ingin diselesaikan dan daftar mesin untuk menyelesaikannya. Pada *optimal job scheduling* dapat memiliki karakteristik penjadwalan yang berbeda dan dibagi menjadi dua, yakni *jobshop scheduling* dan *flowshop scheduling*. Perbedaan karakteristik tersebut terdapat pada arah pengerjaan pekerjaan. Arah dapat diartikan sebagai urutan penggunaan mesin dalam menyelesaikan pekerjaan, dimana *jobshop* diperbolehkan untuk memiliki lebih dari satu arah, sedangkan pada *flowshop* hanya terdapat satu arah saja. *Flowshop scheduling* dibahas lebih lanjut sebagai masalah utama dari skripsi ini.



Gambar 1.1: Contoh dari *flowshop scheduling*.

Pada masalah *Flowshop Scheduling*, diberikan sebanyak n buah pekerjaan yang harus diselesaikan, dimana setiap pekerjaan tersebut memiliki serangkaian tahapan operasi yang dikerjakan oleh mesin yang tersedia, dengan jumlah dari tahapan operasi dan mesin adalah sama yakni m buah [1]. Jumlah pekerjaan, jumlah mesin, dan detail waktu pengerjaan diberikan dari awal dan bersifat statis. Dapat dilihat pada contoh Gambar 1.1, terdapat sebuah kasus *flowshop* dimana terdapat

tiga buah pekerjaan dan tiga buah mesin, dengan semua pekerjaan memiliki urutan arah pengerjaan yang sama, dimulai dari mesin ke-1 hingga mesin ke-3. Penjadwalan ini biasanya dilakukan dengan tujuan memperkecil sebuah objektif. Salah satu objektif yang umum untuk digunakan adalah sebuah variabel bernama *makespan*, yang merupakan total waktu yang dibutuhkan oleh jadwal tersebut (yakni, saat semua operasi pekerjaan sudah diselesaikan). Hasil meminimalkan objektif *makespan* ini dapat menghasilkan pemanfaatan sumber daya yang efisien.

Flowshop Scheduling memiliki beberapa karakteristik, seperti urutan pengerjaan suatu tahapan pekerjaan tidak dapat ditukar (tahapan operasi kedua pekerjaan baru dapat diselesaikan setelah pengerjaan tahapan operasi pertama selesai), lalu untuk setiap mesin hanya dapat mengerjakan satu saja tahapan operasi pengerjaan. Seiring dengan berjalannya waktu, objektif yang terdapat pada permasalahan *Flowshop Scheduling* berkembang sehingga tidak hanya *makespan* saja, seperti contoh, ditemukannya objektif *tardiness* yang merupakan waktu keterlambatan dari suatu mesin untuk mengerjakan operasi pekerjaan tertentu, *idle time* yang merupakan waktu dengan kondisi dimana mesin tidak melakukan proses operasi apapun, dan yang lainnya, sehingga pada permasalahan *flowshop scheduling* dapat memiliki lebih dari satu objektif yang dimanfaatkan untuk melakukan penjadwalan paling optimal, yang dapat disebut sebagai *Multi Objective Flow-shop Scheduling Problem* atau yang biasa disingkat dengan MOFSSP. Pada MOFSSP yang ingin diselesaikan pada skripsi ini, terdapat penambahan dari *Flowshop Scheduling* yang biasa, sesuai dengan namanya yang mengandung *multi-objective*, terdapat satu variabel tambahan yang dimanfaatkan untuk diperkecil, yakni *total-flow-time*, yang merupakan total waktu yang dibutuhkan oleh setiap pekerjaan dihitung dari mulainya penjadwalan / pekerjaan pertama dimulai. Jika suatu pekerjaan menghabiskan waktu yang lebih sedikit, maka inventaris yang dibutuhkan untuk mencatat *work-in-progress*, istilah dalam *supply-chain management* yang menjelaskan barang sebagian jadi yang menunggu penyelesaian, menjadi berkurang. Alasan dipilihnya kedua variabel tersebut dikarenakan kedua variabel tersebut umum untuk digunakan.

Untuk setiap permasalahan dari *Flowshop Scheduling*, termasuk dengan berbagai variasinya, dikatakan jika jumlah dari banyaknya operasi dan mesin yang ada, yakni $m \geq 3$, maka permasalahan ini menjadi permasalahan NP-complete (*non-deterministic polynomial-time hardness*), yakni kelas kompleksitas yang digunakan untuk mengklasifikasikan masalah keputusan tersulit pada masalah-masalah NP karena belum adanya algoritma yang cukup efisien untuk menyelesaikannya. Hal ini didukung dengan alasan tidak ada algoritma yang efisien untuk penyelesaian masalah ini [3]. Masalah *Flowshop Scheduling* dengan jumlah $m = 2$, sudah memiliki cara optimal untuk mendapatkan hasil penjadwalan optimum, dengan menggunakan algoritma yang diusulkan oleh S. M. Johnson [4]. Algoritma tersebut mengusulkan aturan kerja dengan *lemma*, teorema, dengan solusi yang terbukti dapat menghasilkan *makespan* minimum, membutuhkan waktu paling proporsional dengan $n \cdot \log n$, dengan n merupakan jumlah pekerjaan yang harus diselesaikan [4]. Kumar *et al.* [5] sudah pernah melakukan penelitian sebelumnya terhadap penyelesaian *Multi Objective Flow-shop Scheduling Problem* dengan menggunakan algoritma *modified gravitational emulation local search*, algoritma *gravitational emulation local search* (GELS) yang diberikan modifikasi (disebut juga sebagai MGELS). Algoritma GELS merupakan algoritma yang berdasarkan konsep pengacakan dan dua parameter yakni kecepatan dan posisi, dan menggunakan angka-angka acak yang berasal dari algoritma *local search* untuk mencegah solusi *local optimum* (yang diharapkan hasilnya adalah *global optimum*) sehingga hasil menjadi lebih akurat. Algoritma tersebut dipilih dikarenakan dapat mencapai 0 *relative error percentage* (REP), sedangkan untuk algoritma yang lain masih belum dapat memberikan 0 REP. Penelitian dilakukan dengan menggunakan algoritma-algoritma lainnya sebagai pembanding dan ukuran keberhasilan algoritma *gravitational emulation local search*, antara lain yakni algoritma CR, HAMC1, HAMC2, HAMC3, CR(MC), MOACSA, GA, dan DT. Hasil dari penelitian tersebut juga menunjukkan bahwa algoritma MGELS dapat memberikan hasil yang lebih baik daripada algoritma-algoritma yang lainnya. Penelitian lainnya juga dilakukan oleh Arroyo *et al.* dengan menggunakan algoritma yang berbeda, yakni algoritma *multi-objective genetic local search* yang menggunakan strategi *elitism* dan konsep dominasi Pareto [6]. Pada intinya, penelitian tersebut

berhasil untuk membuktikan bahwa algoritma MGOLS mengungguli algoritma lainnya dikarenakan beberapa alasan yang diberikan, yakni: *local search* yang dilakukan dengan *multi-objective* dilakukan secara paralel, penggunaan konsep dominasi Pareto untuk membandingkan dengan tetangga, dan pemilihan elemen dari populasi. Untuk keterangan lebih lanjut dapat dilihat langsung pada sumber yang tertera.

Berhubungan dengan pembuktian dari riset bahwa masalah MOFSSP merupakan masalah *NP-complete* yang tidak dapat diselesaikan oleh algoritma dengan metode pencarian solusi yang biasa, maka pada skripsi ini digunakan salah satu algoritma optimasi *metaheuristic* yang baru beberapa tahun ini dikembangkan, yakni *Whale Optimization Algorithm* atau yang biasa disingkat dengan WOA, sebuah algoritma yang meniru perilaku paus bungkuk saat sedang mencari mangsa untuk dimakan [2]. Menurut Sörensen dan Glover [7], *metaheuristic* adalah sebuah kerangka kerja algoritmik tingkat tinggi dan tidak bergantung pada suatu masalah, yang berisikan strategi untuk mengembangkan algoritma optimasi dengan heuristik. Algoritma ini diterapkan pada perangkat lunak yang dibangun dengan tujuan untuk melakukan penjadwalan optimum dengan meminimalkan nilai dari *makespan* dan *total-flow-time* yang ada pada permasalahan MOFSSP ini.

Pada skripsi ini, dibuat sebuah perangkat lunak berbasis desktop yang mengimplementasikan *Whale Optimization Algorithm* dalam menyelesaikan *Multi-objective Flowshop Scheduling Problem*. Alasan dari dipilihnya perangkat lunak berbasis desktop dikarenakan fokus dari skripsi adalah pada implementasi *whale optimization algorithm* untuk *multi-objective flowshop scheduling problem* dan belum ada rencana untuk pengembangan perangkat lunak selain kebutuhan skripsi. Perangkat lunak ini menggunakan bahasa *Java* dan memiliki tempat-tempat untuk memasukan input ke dalam perangkat lunak seperti *parameter* yang dibutuhkan dalam penyelesaian MOFSSP dengan WOA ini. Hasil dari penjadwalan yang dihasilkan oleh perangkat lunak dapat dilihat pada sebuah *text area*. Untuk melakukan pengujian, perangkat lunak menggunakan kasus dari *benchmark Tailard*. Untuk pengujian pengaruh *parameter*, dilakukan eksperimen menggunakan nilai *parameter* yang berbeda-beda untuk jumlah paus dan jumlah iterasi maksimum. Untuk pengujian kualitas, dilakukan perbandingan dengan hasil yang didapatkan oleh algoritma *Bi-objective Multi-start Simulated-annealing Algorithm*, sebuah algoritma metaheuristik yang menggabungkan pemanfaatan *simulated annealing* dalam mencapai konvergensi pencarian secara efektif dengan *multi-start hill climbing* untuk mencapai diversifikasi solusi sehingga dapat keluar dari *local optimum* secara optimal. Perbandingan ini dilakukan dengan melihat kedua nilai objektif yang digunakan, yakni *makespan* dan *total flow-time*. Hasil dari perbandingan ini dicatat dan digunakan untuk menganalisis performa WOA dalam menyelesaikan MOFSSP.

1.2 Rumusan Masalah

Berikut dipaparkan rumusan masalah yang teridentifikasi pada pembuatan skripsi ini:

1. Bagaimana cara memodelkan *Whale Optimization Algorithm* untuk penyelesaian masalah *Multi Objective Flow-shop Scheduling*?
2. Bagaimana cara mengimplementasikan *Whale Optimization Algorithm* untuk menyelesaikan masalah *Multi Objective Flow-shop Scheduling*?
3. Bagaimana kinerja dari perangkat lunak dengan pemanfaatan *Whale Optimization Algorithm* untuk penyelesaian masalah *Multi Objective Flow-shop Scheduling*?

1.3 Tujuan

Berikut dipaparkan secara lebih terperinci dan terstruktur mengenai tujuan pembuatan skripsi ini:

1. Mempelajari *Multi Objective Flow-shop Scheduling* dan *Whale Optimization Algorithm*, sehingga dapat memodelkan MOFSSP dalam algoritma WOA.
2. Membangun perangkat lunak untuk menyelesaikan masalah *Multi Objective Flow-shop Scheduling* dengan menggunakan *Whale Optimization Algorithm*.

3. Menguji keefektifan dan kualitas *Whale Optimization Algorithm* dengan perbandingan dengan *benchmark* dalam penyelesaian masalah *Multi Objective Flow-shop Scheduling*.

1.4 Batasan Masalah

Berikut dipaparkan batasan-batasan masalah dalam pembuatan perangkat lunak dan skripsi ini:

1. Waktu proses dari setiap pekerjaan telah diketahui dan tidak dapat berubah.
2. Jumlah operasi dari pekerjaan tidak dapat berubah.
3. Objektif yang digunakan adalah *makespan* dan *total flow-time*.

1.5 Metodologi

Berikut dipaparkan metodologi yang digunakan dalam pembuatan perangkat lunak dan skripsi ini:

1. Melakukan studi literatur mengenai *Multi-objective Flowshop Scheduling Problem*.
2. Melakukan studi literatur mengenai penyelesaian masalah *Multi-objective Flowshop Scheduling Problem* dengan algoritma *Bi-objective Multi-start Simulated-annealing Algorithm*.
3. Melakukan studi literatur mengenai *Whale Optimization Algorithm*.
4. Melakukan perancangan model untuk MOFSSP dengan menggunakan *Whale Optimization Algorithm*.
5. Menganalisis kebutuhan perangkat lunak.
6. Melakukan perancangan kebutuhan perangkat lunak.
7. Membangun perangkat lunak.
8. Melakukan pengujian dan eksperimen terhadap perangkat lunak.
9. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematika dari skripsi ini terdiri dari enam bab yang berisi:

1. Bab 1 Pendahuluan

Bab ini berisi latar belakang tentang penjadwalan, *multi-objective flowshop scheduling problem*, dengan *whale optimization algorithm*. Bab ini juga berisi rumusan masalah, tujuan yang ingin dicapai, batasan masalah, metodologi penelitian, serta sistematika penulisan.

2. Bab 2 Landasan Teori

Bab ini berisi dasar teori mengenai penjadwalan, *flowshop*, *multi-objective flowshop*, *whale optimization algorithm*, dan *tailard's benchmark*.

3. Bab 3 Analisis Masalah dan Perangkat Lunak

Bab ini berisi analisis kasus, penerapan dan pemodelan masalah *multi-objective flowshop scheduling* pada *whale optimization algorithm*, dan analisis kebutuhan perangkat lunak.

4. Bab 4 Perancangan Perangkat Lunak

Bab ini berisi perancangan input dan *output*, perancangan tampilan antarmuka, serta diagram kelas dengan lebih detail.

5. Bab 5 Implementasi dan Pengujian

Bab ini berisi lingkungan pengujian, implementasi perangkat lunak, implementasi antarmuka perangkat lunak, pengujian yang dilakukan dengan perangkat lunak, serta analisis dan kesimpulan pengujian eksperimental.

6. Bab 6 Kesimpulan dan Saran

Bab ini berisi kesimpulan dari proses yang sudah dilakukan dan hasil yang sudah didapatkan, dan saran untuk penelitian lanjutan.