

SKRIPSI

ALGORITMA *BEE COLONY* UNTUK PERMASALAHAN
FLOW SHOP SCHEDULING



Yosua

NPM: 2015730067

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2019

UNDERGRADUATE THESIS

**AN ARTIFICIAL BEE COLONY ALGORITHM FOR FLOW
SHOP SCHEDULING PROBLEM**



Yosua

NPM: 2015730067

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2019**

LEMBAR PENGESAHAN

**ALGORITMA *BEE COLONY* UNTUK PERMASALAHAN
*FLOW SHOP SCHEDULING***

Yosua

NPM: 2015730067

Bandung, 13 November 2019

Menyetujui,

Pembimbing

Dr.rer.nat. Cecilia Esti Nugraheni

Ketua Tim Penguji

Anggota Tim Penguji

Mariskha Tri Adithia, P.D.Eng

Natalia, M.Si.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

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

ALGORITMA *BEE COLONY* UNTUK PERMASALAHAN *FLOW SHOP SCHEDULING*

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 13 November 2019

Meterai Rp. 6000

Yosua
NPM: 2015730067

ABSTRAK

Skripsi ini membahas mengenai penyelesaian permasalahan *Flow Shop Scheduling*. *Flow Shop Scheduling* adalah penjadwalan yang melibatkan n pekerjaan dan m mesin dalam urutan proses yang sama, di mana tiap mesin memproses tepat 1 pekerjaan dalam kurun waktu tertentu. Pada *Flow Shop Scheduling*, saat sebuah mesin melakukan pekerjaan, maka mesin lain tidak dapat melakukan pekerjaan yang sejenis dalam waktu yang sama. Urutan pengerjaan yang berbeda mampu menghasilkan waktu total pengerjaan yang berbeda. Hasil akhir dari permasalahan ini adalah waktu total pengerjaan berdasarkan urutan pengerjaan pekerjaan yang dilakukan.

Ada banyak algoritma yang dapat dipakai untuk menentukan urutan pengerjaan pekerjaan tersebut. Pada skripsi ini algoritma yang digunakan untuk mencari urutan pengerjaan pekerjaan *flow shop* adalah algoritma *bee colony*. Algoritma *bee colony* adalah algoritma yang menerapkan metode metaheuristik dan melakukan optimasi sesuai cara kerja dari koloni lebah. Tujuan dari perangkat lunak yang dibuat adalah menampilkan jadwal (terbaik) beserta waktu akhir dari sebuah jadwal. Perangkat lunak pada penelitian ini memberikan jadwal yang berkualitas baik dari sejumlah jadwal yang dihasilkan *flow shop scheduling*. Pengukuran kualitas dilakukan dengan menggunakan *Taillard's Benchmark*. Untuk mengetahui kinerja dari algoritma ini, akan dilakukan eksperimen yang mengganti nilai parameter yang ada pada perangkat lunak.

Berdasarkan eksperimen yang telah dilakukan dengan mengganti nilai parameter yang ada, ukuran dari populasi lebah, banyaknya iterasi, dan jumlah *limit* dari lebah dapat mempengaruhi kandidat solusi yang didapat. *Limit* merupakan parameter kontrol seekor lebah saat mencari sumber makanan baru. Semakin banyak jumlah lebah, banyak iterasi, dan *limit* yang digunakan, semakin baik waktu akhir urutan pekerjaan yang didapat. Algoritma *bee colony* mampu mencapai batas atas dari kasus *Taillard* pada beberapa kasus di jumlah mesin 5 dan 20 pekerjaan. Semakin banyak jumlah mesin dan pekerjaan pada kasus yang ingin dioptimasi, semakin buruk nilai waktu akhir urutan pekerjaan yang didapat.

Kata-kata kunci: Algoritma Bee Colony, Flow Shop, Penjadwalan

ABSTRACT

This thesis will discuss about solving Flow Shop Scheduling Problem. Flow shop scheduling is a process to determine the order of processing collection of work that will be done in a series of machine tools. Different order of job queue is capable to producing different total processing time.

There are many algorithms that can be used to determine the order of the job queue. One algorithm that can be used in flow shop scheduling is bee colony algorithm. Bee colony algorithm is an algorithm that applies the metaheuristic method and optimizes according to the workings of the bee colonies. This algorithm uses fitness values to determine the optimal solution. The greater the fitness value of a solution, the greater the possibility of choosing the solution. The purpose of the software created is to display the (best) schedule along with the end time of a schedule. The software in this study provides a good quality schedule from a number of schedules produced by the flow shop scheduling. Quality measurement can be done using Taillard's Benchmark. To find out the performance of this algorithm, experiments will be carried out that replace the parameter values that exist in the software.

Based on experiments that have been carried out by changing the value of existing parameters, the size of the bee population, the number of iterations, and the limit number of bees can affect the candidate's solution. The more number of bees, the many iterations, and limits used, the better the makespan obtained. Bee colony algorithm is able to reach the upper limit of the Taillard case in some cases in the number of 5 machines and 20 jobs, but the more machines and jobs in the case you want to optimize, the worse the value of makespan obtained.

Keywords: Bee Colony Algorithm, Flow Shop, Scheduling

*Dipersembahkan untuk Tuhan Yang Maha Esa, kedua orang tua,
dan Ibu Cecila Esti Nugraheni selaku dosen pembimbing.*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat, rahmat, dan perlindungan-Nya, penulis dapat menyelesaikan penulisan skripsi ini yang berjudul "Algoritma Artificial Bee Colony untuk Permasalahan Flow Shop Scheduling". Penulis ingin berterima kasih kepada:

1. Orang tua dan kerabat yang telah memberikan dukungan kepada penulis baik secara moral dan financial selama penulis mengerjakan skripsi.
2. Ibu Dr. rer. nat. Cecilia Esti Nugraheni selaku dosen pembimbing yang telah membimbing penulis dan memberikan dukungan serta bantuan kepada penulis selama proses pengerjaan skripsi.
3. Dosen penguji yang telah menguji dan memberikan masukan untuk kelancaran penulisan skripsi kepada penulis.
4. Teman-teman Teknik Informatika Unpar terutama Adrian Stefanus, Glenn Reysan, Matthew Alvredo, Stephen Senjaya, Vincent Joel Sinatra, Hengky Surya, dan Yonathan Kristian Purnama yang telah membantu dan memberikan saran kepada penulis dalam penulisan dokumen skripsi maupun dalam penulisan perangkat lunak.

Penulis berharap bahwa penulisan skripsi ini dapat membantu bagi para pembaca yang sedang meneliti atau mempelajari topik penjadwalan flow shop. Akhir kata, penulis bersedia menerima kritik maupun saran yang dapat membangun masing-masing pihak penulis maupun pembaca agar dapat berkarya dengan lebih baik lagi. Selain itu, penulis meminta maaf jika terdapat kekurangan dalam karya tulis ini. Semoga karya tulis ini dapat bermanfaat bagi semua pihak. Terima kasih.

Bandung, November 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Flow Shop Scheduling	5
2.2 Metaheuristik	6
2.2.1 Konsep Umum Metaheuristik	6
2.2.2 Klasifikasi Metaheuristik	6
2.3 Algoritma Bee Colony	7
2.3.1 Analogi	7
2.3.2 Cara Kerja	7
2.3.3 Nilai Fitness	9
2.3.4 Tahap Employed Bee	10
2.3.5 Insert dan Swap	10
2.3.6 Menentukan Nilai Probabilitas	10
2.3.7 Tahap Onlooker Bee	11
2.3.8 Tahap Scout Bee	11
2.4 <i>Taillard's Benchmark</i>	11
3 ANALISIS	15
3.1 Analisis Masalah Penjadwalan	15
3.2 Analisis Metode Algoritma <i>Bee Colony</i> untuk Penjadwalan	16
3.2.1 Flowchart	16
3.2.2 Representasi Urutan Pengerjaan Sebagai Sumber Makanan	17
3.2.3 Pencarian Solusi Tetangga	17
3.2.4 Mendapatkan Peluang Solusi	19
3.2.5 Kondisi Berhenti	19
3.3 Analisis Perangkat Lunak	19
3.3.1 Deskripsi Perangkat Lunak	19
3.3.2 Input Output	20

3.3.3	Pemodelan Kelas	20
4	PERANCANGAN PERANGKAT LUNAK	23
4.1	Perancangan Antarmuka Pengguna	23
4.1.1	Antarmuka Halaman Awal	23
4.1.2	Antarmuka Masukan	24
4.2	Perancangan <i>Input</i> dan <i>Output</i>	25
4.3	Diagram Kelas Rinci	27
4.3.1	Kelas <i>SimpleTextParser</i>	27
4.3.2	Kelas <i>Job</i>	28
4.3.3	Kelas <i>Input</i>	29
4.3.4	Kelas <i>Output</i>	30
4.3.5	Kelas <i>SimpleScheduler</i>	30
4.3.6	Kelas <i>Bee</i>	31
4.3.7	Kelas <i>MakespanCounter</i>	32
4.3.8	Kelas <i>RandomPositionGenerator</i>	33
4.3.9	Kelas <i>OnlookerChooser</i>	34
4.3.10	Kelas <i>SelfAdaptiveSearchStrategy</i>	34
4.3.11	Kelas <i>ABCScheduler</i>	36
4.3.12	Kelas <i>ArgumentCheck</i>	37
4.3.13	Kelas Program	37
5	IMPLEMENTASI DAN PENGUJIAN	39
5.1	Implementasi Antarmuka	39
5.2	Hasil Implementasi	42
5.3	Pengujian	42
5.3.1	Lingkungan Pengujian	42
5.3.2	Data Pengujian	43
5.3.3	Hasil Pengujian Fungsional	43
5.3.4	Pengujian Eksperimental	44
5.3.5	Analisis Eksperimen	54
6	KESIMPULAN DAN SARAN	57
6.1	Kesimpulan	57
6.2	Saran	57
	DAFTAR REFERENSI	59
	A HASIL EKSPERIMEN DENGAN PARAMETER KARABOGA	61

DAFTAR GAMBAR

2.1	Jadwal yang dibuat dengan penjadwalan <i>flow shop</i> .	6
2.2	<i>Flow chart</i> algoritma <i>bee colony</i> .	8
2.3	Contoh penggunaan operator <i>swap</i> dan <i>insert</i> .	10
2.4	Pilihan kasus <i>Flow Shop Scheduling</i> pada <i>Taillard's benchmark</i> .	11
2.5	Contoh kasus <i>Flow Shop Scheduling</i> pada <i>Taillard's benchmark</i> .	12
2.6	Contoh hasil <i>Flow Shop Scheduling</i> pada <i>Taillard's benchmark</i> .	13
3.1	Jadwal yang dibuat dengan jumlah kursi 1 lusin terlebih dahulu.	16
3.2	Jadwal yang dibuat dengan jumlah kursi 2 lusin terlebih dahulu.	16
3.4	Pemetaan antara sumber makanan dengan operasi penjadwalan.	17
3.3	Flowchart algoritma ABC.	18
3.5	Hasil Pencarian Solusi Tetangga.	19
3.6	Contoh penulisan input.	20
3.7	Diagram kelas pada perangkat lunak pada tahap analisis.	21
4.1	Antarmuka Halaman Awal.	24
4.2	Antarmuka Masukan.	24
4.3	Contoh Masukan yang didapat pada <i>Taillard's benchmark page</i> .	26
4.4	Contoh Masukan yang dikonversi menjadi file teks.	26
4.5	Diagram Kelas Rinci.	27
4.6	Diagram Kelas <i>SimpleTextParser</i> .	27
4.7	Diagram Kelas <i>Job</i> .	28
4.8	Diagram Kelas <i>Input</i> .	29
4.9	Diagram Kelas <i>Output</i> .	30
4.10	Diagram Kelas <i>SimpleScheduler</i> .	30
4.11	Diagram Kelas <i>Bee</i> .	31
4.12	Diagram Kelas <i>MakespanCounter</i> .	32
4.13	Diagram Kelas <i>RandomPositionGenerator</i> .	33
4.14	Diagram Kelas <i>OnlookerChooser</i> .	34
4.15	Diagram Kelas <i>SelfAdaptiveSearchStrategy</i> .	34
4.16	Diagram Kelas <i>ABCScheduler</i> .	36
4.17	Diagram Kelas <i>ArgumentCheck</i> .	37
4.18	Diagram Kelas Program.	37
5.1	Antarmuka Halaman Awal.	39
5.2	Antarmuka Halaman Masukan.	40
5.3	Antarmuka halaman saat mengklik <i>button</i> cari.	40
5.4	Antarmuka halaman saat selesai memilih <i>file</i> teks yang akan digunakan.	41
5.5	Antarmuka halaman saat selesai mengklik <i>button</i> lanjut.	42
5.6	Contoh <i>input</i> masukan pada perangkat lunak.	43
5.7	Hasil pengujian fungsional pada perangkat lunak.	44
5.8	<i>Gantt chart</i> untuk urutan pekerjaan hasil perangkat lunak.	44
5.9	Chart perbandingan lebah.	54

5.10	Chart perbandingan iterasi	55
5.11	Chart perbandingan limit	55
A.1	Chart Eksperimen jumlah lebah pada 5 mesin 20 pekerjaan	62
A.2	Chart Eksperimen jumlah iterasi pada 20 mesin 20 pekerjaan	62
A.3	Chart Eksperimen jumlah lebah pada 10 mesin 20 pekerjaan	64
A.4	Chart Eksperimen jumlah iterasi pada 20 mesin 20 pekerjaan	64
A.5	Chart Eksperimen jumlah lebah pada 20 mesin 20 pekerjaan	65

DAFTAR TABEL

2.1	Tabel data <i>processing time</i>	5
3.1	Tabel data <i>processing time</i> pada perakitan kayu.	15
4.1	Tabel untuk Objek Antarmuka Halaman Awal	24
4.2	Tabel untuk Objek Antarmuka Masukan	25
4.3	Ilustrasi contoh tabel yang akan dihasilkan untuk soal yang dipilih.	27
5.1	Tabel data <i>processing times</i> pengujian fungsional	43
5.2	Tabel parameter	45
5.3	Tabel eksperimen 5 lebah pada 20 mesin 20 pekerjaan	45
5.4	Tabel eksperimen 50 lebah pada 20 mesin 20 pekerjaan	46
5.5	Tabel eksperimen 5 lebah pada 20 mesin 50 pekerjaan	46
5.6	Tabel eksperimen 50 lebah pada 20 mesin 50 pekerjaan	47
5.7	Tabel eksperimen 5 lebah pada 20 mesin 100 pekerjaan	47
5.8	Tabel eksperimen 50 lebah pada 20 mesin 100 pekerjaan	48
5.9	Tabel eksperimen 250 iterasi pada 20 mesin 20 pekerjaan	48
5.10	Tabel eksperimen 2500 iterasi pada 20 mesin 20 pekerjaan	49
5.11	Tabel eksperimen 250 iterasi pada 20 mesin 50 pekerjaan	49
5.12	Tabel eksperimen 2500 iterasi pada 20 mesin 50 pekerjaan	50
5.13	Tabel eksperimen 250 iterasi pada 20 mesin 100 pekerjaan	50
5.14	Tabel eksperimen 2500 iterasi pada 20 mesin 100 pekerjaan	51
5.15	Tabel eksperimen 10 limit pada 20 mesin 20 pekerjaan	51
5.16	Tabel eksperimen 100 limit pada 20 mesin 20 pekerjaan	52
5.17	Tabel eksperimen 10 limit pada 20 mesin 50 pekerjaan	52
5.18	Tabel eksperimen 100 limit pada 20 mesin 50 pekerjaan	53
5.19	Tabel eksperimen 10 limit pada 20 mesin 100 pekerjaan	53
5.20	Tabel eksperimen 100 limit pada 20 mesin 100 pekerjaan	54
5.21	Tabel eksperimen rata-rata lebah	54
5.22	Tabel eksperimen rata-rata iterasi	54
5.23	Tabel eksperimen rata-rata limit	55
A.1	Tabel eksperimen <i>Taillard's benchmark</i> 5 mesin 20 pekerjaan	61
A.2	Tabel eksperimen 50 lebah <i>Taillard's benchmark</i> 5 mesin 20 pekerjaan	61
A.3	Tabel eksperimen <i>Taillard's benchmark</i> 10 mesin 20 pekerjaan	63
A.4	Tabel eksperimen 50 lebah <i>Taillard's benchmark</i> 10 mesin 20 pekerjaan	63
A.5	Tabel eksperimen <i>Taillard's benchmark</i> 5 mesin 50 pekerjaan	65
A.6	Tabel eksperimen 5000 iterasi <i>Taillard's benchmark</i> 20 mesin 20 pekerjaan	65
A.7	Tabel eksperimen <i>Taillard's benchmark</i> 10 mesin 50 pekerjaan	66
A.8	Tabel eksperimen <i>Taillard's benchmark</i> 20 mesin 50 pekerjaan	66
A.9	Tabel eksperimen <i>Taillard's benchmark</i> 5 mesin 100 pekerjaan	66
A.10	Tabel eksperimen <i>Taillard's benchmark</i> 10 mesin 100 pekerjaan	67
A.11	Tabel eksperimen <i>Taillard's benchmark</i> 20 mesin 100 pekerjaan	67

A.12	Tabel eksperimen <i>Taillard's benchmark</i> 10 mesin 200 pekerjaan	67
A.13	Tabel eksperimen <i>Taillard's benchmark</i> 20 mesin 200 pekerjaan	68
A.14	Tabel eksperimen <i>Taillard's benchmark</i> 20 mesin 500 pekerjaan	68

BAB 1

PENDAHULUAN

Bab ini berisi tentang pendahuluan dalam penelitian, yang berisi latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan. Latar belakang masalah akan berisi alasan kenapa topik ini dipilih, rumusan masalah akan berisi masalah-masalah yang akan diteliti, tujuan penelitian adalah untuk menjawab rumusan masalah, pembatasan masalah dilakukan agar penelitian menjadi lebih fokus ke hal yang akan dituju, metodologi akan berisi landasan teori yang akan membantu dalam penelitian, sistematika pembahasan akan berisi detail dari bab-bab yang akan ditulis.

1.1 Latar Belakang

Scheduling atau penjadwalan merupakan penyusunan beberapa aktivitas, di mana aktivitas tersebut memiliki beberapa operasi beserta masing-masing waktunya yang diatur sedemikian rupa hingga menjadi sebuah jadwal. Salah satu tujuan utama dari penjadwalan adalah untuk meminimalkan waktu total penyelesaian keseluruhan aktivitas tersebut. Masalah penjadwalan banyak ditemukan di dunia nyata contohnya, penjadwalan kuliah, penjadwalan kerja dan lain-lain. Masalah penjadwalan juga merupakan masalah yang penting dalam dunia industri. Dalam dunia industri terdapat istilah yang dikenal dengan *shop scheduling*.

Shop scheduling adalah masalah penjadwalan di mana di dalamnya terdapat beberapa pekerjaan yang tiap pekerjaannya memiliki beberapa proses atau operasi yang dikerjakan oleh suatu mesin. Salah satu jenis dari *shop scheduling* adalah *flow shop scheduling*, yaitu penjadwalan yang melibatkan n buah pekerjaan dan m buah mesin, di mana urutan mesin yang mengerjakan pekerjaan tidak dapat berubah. Salah satu permasalahan yang dapat dimodelkan pada permasalahan *flow shop scheduling* adalah perakitan kursi. Dalam permasalahan ini, perakitan kursi harus melewati urutan proses yang tidak dapat berubah. Hasil dari permasalahan ini berupa waktu akhir keseluruhan kursi tersebut dibuat.

Permasalahan *flow shop scheduling* dapat diselesaikan dengan metode FIFO, selain itu terdapat juga beberapa algoritma yang menggunakan metode metaheuristik untuk menyelesaikan permasalahan *flow shop scheduling*, seperti algoritma *ant colony*, algoritma *cat swarm*, algoritma *bee colony*, dan lain-lain. Dari berbagai macam algoritma yang dapat digunakan untuk melakukan penjadwalan *flow shop*, dipilih sebuah algoritma yaitu algoritma *bee colony*.

Algoritma *bee colony* merupakan salah satu algoritma pencarian yang terinspirasi dari perilaku lebah madu dalam proses pencarian sumber makanan. Algoritma ini sudah banyak digunakan pada beberapa kasus penelitian. Hasil dari algoritma ini adalah lokasi-lokasi yang memiliki jumlah dan hasil makanan yang baik untuk koloni lebah.

Pada skripsi ini, akan dibuat perangkat lunak yang mengimplementasikan algoritma *bee colony* untuk mencari urutan penjadwalan terbaik yang dimodelkan dalam *flow shop scheduling*. Hasil dari perangkat lunak berupa waktu total pengerjaan sebuah urutan pekerjaan beserta urutan pekerjaan yang dikerjakan. Untuk mengukur kualitas jadwal yang didapat dari algoritma *bee colony* digunakan sebuah *benchmark*. *Benchmark* merupakan teknik pengetesan dengan menggunakan suatu nilai standar. *Benchmark* yang digunakan dapat berbeda-beda, salah satu contoh *benchmark* yang akan

digunakan yaitu *Taillard's Benchmark*. *Taillard's Benchmark* berisi kumpulan data waktu operasi dari beberapa pekerjaan, di mana waktu total dari jadwal terbaik dan terburuk yang didapat dari data tersebut dijadikan sebagai batas atas dan batas bawah untuk dilakukan pengukuran kualitas jadwal yang didapat dari algoritma *bee colony*.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara membangun perangkat lunak untuk menyelesaikan permasalahan *flow shop scheduling* dengan menggunakan algoritma *bee colony*?
2. Bagaimana kinerja yang dihasilkan perangkat lunak?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Membangun perangkat lunak yang berfungsi untuk menyelesaikan permasalahan *flow shop scheduling* dengan menggunakan algoritma *bee colony*.
2. Mengukur kinerja yang dihasilkan perangkat lunak.

1.4 Batasan Masalah

Batasan dan asumsi untuk penelitian ini adalah:

Benchmark yang digunakan yaitu *benchmark* milik Taillard.

1.5 Metodologi

Metodologi penyelesaian masalah dalam penelitian ini adalah:

1. Studi Literatur
Mencari referensi dari sumber-sumber tertentu untuk memperdalam pemahaman mengenai cara kerja proses penjadwalan *flow shop* dan cara kerja algoritma *bee colony* agar dapat mengaplikasikan algoritma tersebut pada proses penjadwalan *flow shop*.
2. Analisa Kasus
Menentukan cara pengaplikasian algoritma *bee colony* untuk optimasi penjadwalan *flow shop*. Menentukan data masukan dan data keluaran yang dibutuhkan oleh perangkat lunak. Menentukan fungsi-fungsi apa saja yang dibutuhkan perangkat lunak.
3. Pengembangan perangkat lunak
Membentuk struktur kelas dari perangkat lunak. Mendesain *interface* yang sesuai untuk perangkat lunak. Membangun perangkat lunak untuk optimasi penjadwalan *flow shop* dengan algoritma *bee colony*. Melakukan pengujian fungsional pada perangkat lunak.
4. Eksperimen
Melakukan proses optimasi dengan menggunakan perangkat lunak pada beberapa sampel kasus *flow shop*. Mencatat dan mengolah data hasil proses optimasi untuk mengukur performa perangkat lunak. Mengukur tingkat keoptimalan dari proses optimasi yang dilakukan oleh perangkat lunak.

5. Pengambilan kesimpulan
Mengambil kesimpulan yang bisa didapatkan dari hasil eksperimen. Melakukan dokumentasi dari skripsi ini.
6. Dokumentasi
Penyusunan Laporan hasil seluruh kegiatan ke dalam dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematika penulisan karya tulis ini adalah sebagai berikut:

1. Bab 1: Pendahuluan untuk mendefinisikan masalah yang akan dibahas, alasan pemilihan topik dan usulan solusi.
2. Bab 2: Dasar teori yang digunakan dalam penelitian ini. Pembahasan permasalahan *flow shop*. Pembahasan cara kerja dari algoritma *bee colony*. Pembahasan cara pengaplikasian algoritma *bee colony* untuk optimasi proses penjadwalan *flow shop*
3. Bab 3: Analisis masalah yang akan dilakukan pada penelitian ini. Pembahasan cara penerapan algoritma *bee colony* dan rancangan awal dari perangkat lunak yang telah dibuat.
4. Bab 4: Perancangan perangkat lunak. Detil informasi mengenai perangkat lunak yang telah dibuat. Struktur kelas dan desain antarmuka grafis dari perangkat lunak yang telah dibuat.
5. Bab 5: Implementasi dan pengujian. Hasil implementasi algoritma *bee colony* pada perangkat lunak. Penjelasan cara penggunaan perangkat lunak. Hasil pengujian dan eksperimen kasus *flow shop* pada perangkat lunak.
6. Bab 6: Kesimpulan dan saran. Hal-hal yang dapat disimpulkan dari penelitian ini. Saran pengembangan yang dapat dilakukan pada penelitian selanjutnya.