



**PERBANDINGAN ALGORITMA *HARMONY SEARCH*
DAN ALGORITMA *ARTIFICIAL BEE COLONY*
PADA PENJADWALAN *JOBSHOP***

SKRIPSI

Oleh
Trian Ghofarul Maroghi
NIM 101810101047

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015**



**PERBANDINGAN ALGORITMA *HARMONY SEARCH*
DAN ALGORITMA *ARTIFICIAL BEE COLONY*
PADA PENJADWALAN *JOBSHOP***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Trian Ghofarul Maroghi
NIM 101810101047

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Umalah dan Ayahanda Endi Sunarto yang memberikan kasih sayang, segala dukungan dan doa yang tiada henti;
2. Kakakku Risma Fitria Fadma, Ira Sagitia Ningtyas beserta keluarga atas bantuan serta doanya;
3. Tutus Julantika yang memberi kasih sayang, motivasi, dan inspirasinya;
4. Almarhum sahabat terbaik Slamet Syaifuddin Zuhri yang selalu memberi dukungan dan doanya;
5. Guru-guru TK Dharma Wanita Sragen, SDN 2 Benculuk, MTsN Srono, SMAN 1 Giri Banyuwangi, dan dosen-dosen Universitas Negeri Jember yang telah memberikan ilmu dan membimbing dengan penuh kesabaran;
6. Almamater Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

” Barangsiapa bertawakkal pada Allah, maka Allah akan memberikan kecukupan padanya, sesungguhnya Allah lah yang akan melaksanakan urusan (yang dikehendaki)-Nya.”

(terjemahan QS. Ath – Thalaq: 3)*)

You can't always get what you want, but, if you try, sometimes you just might find you get what you need .”

(Raditya Dika)**)

*) Departemen Agama Republik Indonesia. 2004. *Al-Qur'an dan Terjemahannya*. Bandung: CV Penerbit Diponegoro.

***) Dika, R. 2011. *Manusia Setengah Salmon* . Jakarta: Gagas Media.

PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama : Trian Ghofarul Maroghi

NIM : 101810101047

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul **“Perbandingan Algoritma *Harmony Search* dan Algoritma *Artificial Bee Colony* pada Penjadwalan *Jobshop*”** adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juli 2015

Yang menyatakan,

Trian Ghofarul Maroghi
NIM. 101810101047

SKRIPSI

**PERBANDINGAN ALGORITMA *HARMONY SEARCH*
DAN ALGORITMA *ARTIFICIAL BEE COLONY*
PADA PENJADWALAN *JOBSHOP***

Oleh

Trian Ghofarul Maroghi
NIM 101810101047

Pembimbing

Dosen Pembimbing Utama : Ahmad Kamsyakawuni, S.Si., M.Kom.

Dosen Pembimbing Anggota : Kusbudiono, S.Si., M.Si.

PENGESAHAN

Skripsi berjudul “**Perbandingan Algoritma *Harmony Search* dan Algoritma *Artificial Bee Colony* pada Penjadwalan *Jobshop***” telah diuji dan disahkan oleh Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember pada:

hari :

tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji

Dosen Pembimbing Utama,

Dosen Pembimbing Anggota,

Ahmad Kamsyakawuni, S.Si., M.Kom
NIP. 197211291998021001

Kusbudiono, S.Si., M.Si
NIP. 197704302005011001

Anggota Tim Penguji

Dosen Penguji I,

Dosen Penguji II,

Drs. Rusli Hidayat, S.Si, M.Sc
NIP. 196610121993031001

Kosala Dwidja Purnomo, S.Si., M.Si
NIP. 196908281998021001

Mengesahkan,
Dekan,

Prof. Drs. Kusno, DEA., Ph.D
NIP. 196101081986021001

RINGKASAN

Perbandingan Algoritma *Harmony Search* dan Algoritma *Artificial Bee Colony* pada Penjadwalan *Jobshop*. Trian Ghofarul Maroghi , 101810101047; 2015: 59 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Banyak sekali perindustrian yang berkompetisi untuk mendapat perhatian pasar. Semakin banyak persaingan maka semakin menuntut industri untuk mendapatkan penjadwalan kerja yang optimal agar menghasilkan keuntungan yang maksimal. Dalam hal ini dibutuhkan suatu penjadwalan kerja yang baik agar dapat mengatur kinerja industri yang akan memberikan hasil efektif dan efisien. Penjadwalan ini biasanya berupa minimasi *makespan* yaitu total waktu yang dibutuhkan dalam menyelesaikan suatu pekerjaan. Penjadwalan yang dibutuhkan salah satunya adalah penjadwalan *jobshop* yaitu penjadwalan yang setiap *job*nya diproses serta aliran mesinnya bersifat acak atau tidak harus urut. Jadi untuk setiap komponennya bisa dikerjakan bersamaan atau mendahulukan salah satu komponen.

Data dalam penelitian ini adalah data sekunder berupa pengamatan mesin pembuatan 4 jenis *springbed*, yaitu *Platinum*, *Golden*, *Silver*, dan *Bigline*. Data tersebut berupa data nomer mesin dan waktu proses pembuatan tiap komponen dari masing-masing mesin. Jumlah mesin yang digunakan untuk membuat *springbed* ada 9 jenis mesin yang direpresentasikan dengan nomer urut dan angka dalam menit untuk waktu proses mesin. *Springbed* tersebut diproses oleh 9 jenis mesin yang pengelompokannya berdasarkan mesin yang dibutuhkan untuk membuat komponen tiap *springbed*. Jumlah jenis mesin pembuat masing-masing komponen akan berbeda-beda. Komponen keseluruhan yang dibuat hingga proses perakitan berjumlah 4

komponen inti yaitu matras, sandaran, divan dan kaki *springbed* dilanjutkan dengan 1 perlakuan terakhir yaitu perakitan komponen keseluruhan. Sehingga 9 jenis mesin pembuat *springbed* ini akan direduksi menjadi 5 mesin notasi baru sesuai dengan komponen-komponennya.

Pada penelitian ini menggunakan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* yang diterapkan pada industri pabrik *springbed* PT. Cahaya Kawi Ultra Polyintraco untuk menyelesaikan permasalahan penjadwalan *jobshop*. Perbandingan dari algoritma ini bertujuan untuk mencari *makespan* dan tingkat kekonvergenan dari masing-masing.

Penelitian dilakukan melalui beberapa tahap, yaitu studi literature kemudian pengambilan data. Setelah data didapatkan lalu membuat program untuk mempermudah pengerjaannya. Selanjutnya simulasi program dan kedua algoritma tersebut di analisis berdasarkan *makespan* dan tingkat konvergensinya. Langkah terakhir diambil kesimpulan dari penelitian tersebut adalah diambil kesimpulan dari perbandingan ini.

Berdasarkan penelitian yang dilakukan maka didapatkan *makespan* terbaik sebesar 247 menit pada algoritma *Harmony Search* sementara pada algoritma *Artificial Bee Colony* sebesar 262 menit. Sehingga dalam hal *makespan* menyatakan algoritma *Harmony Search* lebih baik daripada algoritma *Artificial Bee Colony*. Namun dari tingkat kekonvergenan algoritma *Artificial Bee Colony* cenderung lebih cepat konvergen daripada algoritma *Harmony Search*.

PRAKATA

Puji syukur Alhamdulillah atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Perbandingan Algoritma *Harmony Search* dan Algoritma *Artificial Bee Colony* pada Penjadwalan *Jobshop*”. Skripsi ini disusun sebagai salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Pada kesempatan ini penulis menyampaikan terima kasih kepada semua pihak yang telah membantu terselesaikannya skripsi ini diantaranya:

1. Prof. Drs. Kusno, DEA, Ph.D selaku Dekan FMIPA Universitas Jember;
2. Kosala Dwidja Purnomo, S.Si., M.Si selaku Ketua Jurusan Matematika;
3. Ahmad Kamsyakawuni, S.Si., M.Kom selaku Dosen Pembimbing Utama dan Dosen Pembimbing Akademik dan Kusbudiono, S.Si, M.Si selaku Dosen Pembimbing Anggota yang telah membimbing penulisan skripsi ini;
4. Drs. Rusli Hidayat, S.Si., M.Sc dan Kosala Dwidja Purnomo, S.Si., M.Si selaku Dosen Penguji yang banyak memberi kritik dan saran dalam penulisan skripsi ini;
5. Ibunda Umalah dan Ayahanda Endi Sunarto beserta keluarga besar yang selalu memberi doa dan dukungan;
6. Orang-orang terdekat yang selalu menjadi semangat dalam menyelesaikan kuliah; Tutus Julantika, Slamet Syaifuddin Zuhri, Octavian Edwin, Laily Fauziah, dan Fitri Kumalasari;
7. Teman-teman KKN 133206 dan teman-teman kos Mastrip 5 untuk dukungannya;
8. Teman-teman Mathgic 2010 yang telah menjadi keluarga selama kuliah serta memberikan dukungan dan semangat dalam menyelesaikan tugas akhir ini;
9. Keluarga besar HMJ Himatika “Geokompstat” yang banyak memberikan pelajaran serta pengalaman yang tidak saya dapatkan di bangku kuliah;

10. Orang-orang yang ikut andil membantu terselesaikannya tugas akhir ini; Sri Astutik, Vina F .Martin, Jauharin Insiyah, Arista Rosita, Nurhalimatus Sa'dia, Rima Lutviyana, Putri Rukmana, Maghfira, Mbak Eka Farista, dan Mas Arif Riyanto.
11. Semua pihak yang tidak dapat saya sebutkan satu per satu.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi, sehingga kritik dan saran dari semua pihak sangat dibutuhkan demi kesempurnaan skripsi ini. Semoga skripsi ini berguna bagi penulis maupun pihak lain yang memanfaatkan. Amin.

Jember, Juli 2015

Penulis

DAFTAR ISI

	Halaman
HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	iv
HALAMAN PERNYATAAN	v
HALAMAN PEMBIMBING	vi
HALAMAN PENGESAHAN	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Penjadwalan	5
2.2 Penjadwalan <i>Jobshop</i>	7
2.2.1 <i>Jobshop Loading</i>	8
2.2.1 <i>Jobshop Sequencing</i>	9
2.3 Diagram Gantt (<i>Gantt Chart</i>)	10
2.4 Pengertian Algoritma	11

2.5 Algoritma <i>Harmony Search</i>	12
2.5.1 Parameter Algoritma <i>Harmony Search</i>	12
2.5.2 Mekanisme Algoritma <i>Harmony Search</i>	13
2.6 Algoritma <i>Bee Colony</i>	16
BAB 3. METODE PENELITIAN	20
3.1 Data Penelitian	20
3.2 Langkah Penyelesaian	21
BAB 4. HASIL DAN PEMBAHASAN	23
4.1 Hasil	23
4.1.1 Penjadwalan <i>Jobshop</i> dengan Program MATLAB.....	23
4.1.2 Perhitungan Manual.....	28
4.1.3 <i>Makespan</i> dan Kekonvergenan Algoritma.....	36
4.2 Pembahasan	37
BAB 5. PENUTUP	54
5.1 Kesimpulan	54
5.2 Saran	54
DAFTAR PUSTAKA	55

DAFTAR TABEL

	Halaman
3.1 Data Penelitian.....	20
4.1 Waktu Proses Produksi.....	29
4.2 Jadwal Solusi Awal.....	30
4.3 Perhitungan <i>Makespan</i> VS ke-1	31
4.5 Perhitungan <i>Makespan Artificial Bee Colony</i>	34
4.6 Hasil <i>Makespan</i> dan Pengujian Konvergensi	36
4.7 Hasil Percobaan Pertama.....	38
4.8 Hasil Percobaan Kedua	39
4.9 Hasil Percobaan Ketiga	41
4.10 Hasil Percobaan Keempat.....	42
4.11 Hasil Percobaan Kelima	43
4.12 Hasil Percobaan Keenam.....	45
4.13 Hasil Percobaan Ketujuh	46
4.14 Hasil Percobaan Kedelapan	47
4.15 Hasil Percobaan Kesembilan	49
4.16 Hasil Percobaan Kesepuluh	50
4.17 Sepuluh Percobaan dengan Algoritma <i>Harmony Search</i>	51
4.18 Sepuluh Percobaan dengan Algoritma <i>Artificial Bee Colony</i>	52

DAFTAR GAMBAR

	Halaman
2.1 Diagram <i>Gantt</i> (<i>Gantt Chart</i>)	11
3.1 Skema Langkah-langkah Penelitian.....	22
4.1 Tampilan Awal Program Penjadwalan <i>Jobshop</i>	24
4.2 Tampilan Tabel Data Manual	25
4.3 Tampilan Tabel Data dalam Dokumen	26
4.4 Tampilan Tabel Input Parameter Kedua Algoritma.....	26
4.5 Tampilan Output Parameter Kedua Algoritma.....	27
4.6 Tampilan <i>Gantt Chart</i> Algoritma <i>Harmony Search</i>	28
4.7 Tampilan <i>Gantt Chart</i> Algoritma <i>Artificial Bee Colony</i>	28
4.8 <i>Gantt Chart</i> VS-1.....	31

DAFTAR LAMPIRAN

	Halaman
A. Data Waktu Proses Mesin	56
B. Data Reduksi 9 Mesin ke 5 Mesin Notasi Baru	57
C. Data Waktu Proses Mesin pada 4 Jenis <i>Springbed</i>	58

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Dewasa ini banyak sekali perindustrian yang saling berkompetisi untuk mendapatkan perhatian pasar. Semakin banyaknya permintaan dari konsumen maka akan menuntut sebuah industri untuk dapat meningkatkan produksinya. Beberapa cara telah dilakukan oleh industri untuk menghasilkan suatu produk yang unggul, namun belum diproduksi dalam waktu yang singkat serta biaya yang rendah. Untuk itu industri tersebut harus memahami konsep dasar pengelolaan yang baik dan benar. Pengelolaan tersebut adalah berupa pengelolaan terhadap penjadwalan yang efektif dan efisien untuk mengoptimalkan waktu maupun biaya produksi.

Suatu penjadwalan yang baik mampu menghasilkan produk yang akan menghemat waktu serta biaya sehingga industri akan dapat memenuhi pesanan konsumen baik dalam jumlah kolektif maupun individu. Dengan demikian kepuasan konsumen akan meningkatkan penjualan suatu produk dan semakin lama akan berimbas terhadap peningkatan ekonomi industri tersebut.

Optimasi adalah salah satu disiplin ilmu dalam matematika yang fokus untuk mendapatkan nilai maksimum atau minimum secara sistematis dari suatu fungsi, peluang maupun pencarian nilai lainnya dalam berbagai kasus. Optimasi sangat berguna dalam berbagai bidang dalam rangka melakukan usaha yang efektif dan efisien untuk mendapatkan hasil yang diinginkan. Optimasi penjadwalan produksi yang banyak dianalisa adalah penjadwalan *jobshop* dan *flowshop*, keduanya banyak diselesaikan dengan pendekatan metaheuristik.

Penjadwalan *jobshop* adalah penjadwalan yang memiliki kendala urutan pemrosesan pekerjaan (*job*), dan setiap *job* harus melalui setiap mesin tepat satu kali. Tujuannya adalah untuk meminimalkan waktu yang dibutuhkan dalam menyelesaikan

seluruh proses operasi yang dijadwalkan, mulai pemrosesan operasi pertama hingga terakhir setelah diproses.

Pada tahun 2001 Geem mengenalkan Algoritma *Harmony Search*. Algoritma ini terinspirasi oleh perbaikan harmoni musik yang dilakukan suatu kelompok musik. Suyanto, (2010) mengemukakan bahwa teknik ini menggunakan proses pencarian seperti pada improvisasi musik yang berusaha mendapatkan keadaan terbaik dengan cara mengevaluasi fungsi obyektif. Sedangkan algoritma *Artificial Bee Colony* terinspirasi dari kebiasaan hidup lebah. Lebah merupakan serangga yang terpadu. Kemampuan bertahan hidup seluruh koloni lebah tergantung dari tiap individu lebah. Lebah menggunakan tugas-tugas yang sistematis yang diantaranya bertujuan menjaga eksistensi koloninya. Lebah melakukan bermacam-macam pekerjaan seperti foraging, reproduksi, membangun sarang dan lain-lain. Dari semua kebiasaan ini foraging merupakan aktivitas penting untuk menjaga pasokan makanan pada sarang lebah. Kebiasaan lebah diadaptasi menjadi algoritma untuk memecahkan permasalahan yang kompleks.

Pada penelitian sebelumnya tentang penjadwalan *jobshop*, Hapsari (2014) mengkaji penggunaan algoritma *Harmony Search* dan algoritma Genetika pada penjadwalan produksi Dandang pada industri rumah tangga UD.Alif Jember. Pada kajian tersebut dihasilkan *makespan* algoritma *Harmony Search* lebih baik dibandingkan algoritma Genetika. Pada penelitian lain, Indriana (2011) mengkaji penggunaan algoritma *Harmony Search*, algoritma *Differential Evolution*, dan algoritma *Swam Optimization* dalam menyelesaikan penjadwalan pada perusahaan penerbangan. Hasilnya dari segi waktu komputasi algoritma *Harmony Search* memerlukan waktu yang lebih cepat dibanding metode lain. Sugioko (2012) mengkaji penggunaan Modifikasi Algoritma *Bee Colony* dengan *Tabu List* pada penjadwalan *jobshop* dengan kriteria biaya keterlambatan. Pada analisis tersebut penggunaan *Tabu List* akan menghasilkan waktu komputasi dan *makespan* tidak memberi pengaruh yang signifikan sedangkan pada *Artificial Bee Colony* memberikan pengaruh yang signifikan terhadap fungsi tujuan (biaya keterlambatan) dan waktu komputasinya.

Berdasarkan keterangan diatas, penulis memilih untuk meneliti dan membandingkan algoritma *Bee Colony* dengan algoritma *Harmony Search* pada penjadwalan industri pabrik springbed PT. Cahaya Kawi Ultra Polyintraco untuk mendapatkan *makespan* dan tingkat kekonvergenan dari kedua algoritma serta dengan asumsi tidak terjadi kendala-kendala yang dapat menghambat proses produksi. Diharapkan dengan menggunakan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* dapat menyelesaikan permasalahan penjadwalan tersebut.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas pada skripsi ini adalah :

- a. Bagaimana penerapan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* untuk menyelesaikan permasalahan penjadwalan *jobshop*?
- b. Bagaimana perbandingan dari algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* berdasarkan efisiensi dilihat dari nilai *makespan* dan kekonvergenan dari masing-masing algoritma yang diterapkan?

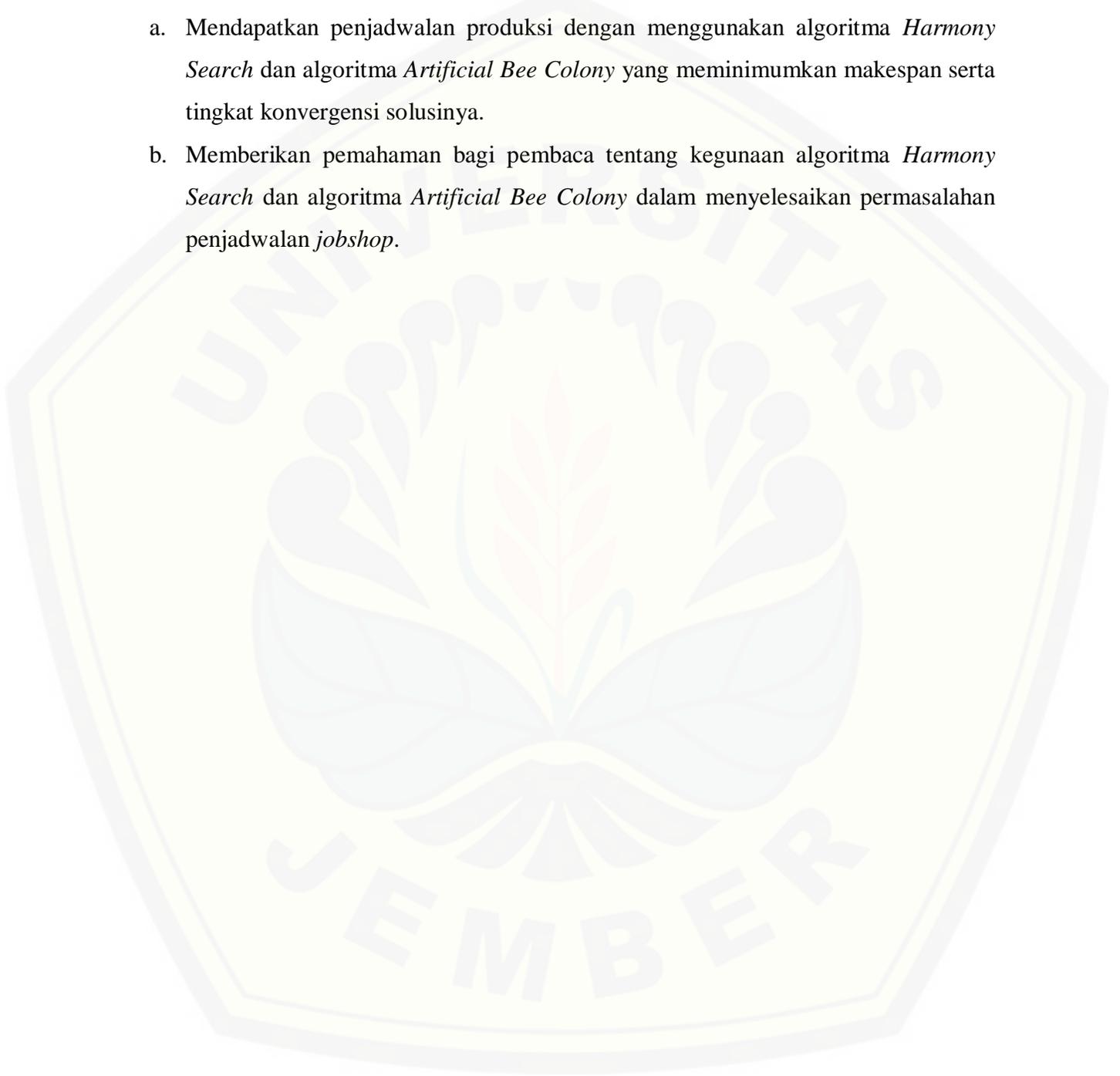
1.3 Tujuan

Tujuan yang ingin dicapai penulis dalam skripsi ini adalah mendapatkan penjadwalan *jobshop* secara optimal serta dengan efisiensi waktu dari nilai *makespan* produksi yang minimum serta tingkat konvergensi solusi menggunakan masing-masing analisis dari algoritma *Harmony Search* maupun algoritma *Artificial Bee Colony*.

1.4 Manfaat

Manfaat yang dapat diperoleh dari skripsi ini adalah :

- a. Mendapatkan penjadwalan produksi dengan menggunakan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* yang meminimumkan makespan serta tingkat konvergensi solusinya.
- b. Memberikan pemahaman bagi pembaca tentang kegunaan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* dalam menyelesaikan permasalahan penjadwalan *jobshop*.



BAB 2. TINJAUAN PUSTAKA

2.1 Penjadwalan

Penjadwalan adalah pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin. Pengertian penjadwalan secara umum adalah : “*Scheduling is the allocation of resources to perform collection of risk*”, yang artinya penjadwalan adalah pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan. Permasalahan muncul apabila pada tahap operasi tertentu beberapa atau seluruh pekerjaan itu membutuhkan stasiun kerja yang sama. Dengan dilakukannya pengurutan pekerjaan ini unit-unit produksi (*resources*) dapat dimanfaatkan secara optimal. Pemanfaatan ini antara lain dilakukan dengan jalan meningkatkan unit-unit produksi melalui usaha-usaha mereduksi waktu menganggur (*idle time*) dari unit-unit yang bersangkutan. Pemanfaatan lainnya dapat juga dilakukan dengan cara meminimumkan *in-process inventory* melalui reduksi terhadap waktu rata-rata pekerjaan yang menunggu (antri) dalam baris antrian pada unit-unit produksi.

Pengertian penjadwalan diatas tidak terbatas hanya untuk penjadwalan mesin saja sebagai factor utama dalam penentuan penjadwalan tetapi meliputi unit-unit produksi (*resources*) yang berkaitan langsung pada proses produksi. Setiap aktivitas yang diminta pada jenis sumber daya sebuah waktu proses. Sumber daya juga memiliki bagian-bagian yang disebut mesin, sel, transportasi, penundaan dan sebagainya. Hal ini menunjukkan bahwa penjadwalan tidak hanya terbatas pada mesin saja tetapi setiap elemen kerja yang membutuhkan waktu (Ginting, 2009).

Penjadwalan merupakan alat ukur yang baik bagi perencanaan agregat. Pesanan-pesanan aktual pada tahap ini akan ditugaskan pertama kalinya pada

sumberdaya tertentu (fasilitas, pekerja, peralatan), kemudian dilakukan pada tiap-tiap pusat pemrosesan sehingga akan mencapai optimalitas utilitas kapasitas yang ada.

Penjadwalan mempunyai beberapa elemen penting yang harus diperhatikan. Elemen-elemen tersebut adalah sebagai berikut.

a. *Job*

Job dapat didefinisikan sebagai sebuah pekerjaan yang harus diselesaikan untuk mendapatkan suatu produk atau hasil. *Job* umumnya mewakili satu operasi yang harus dikerjakan.

b. *Operasi*

Operasi adalah himpunan bagian dari *job*. Untuk menyelesaikan suatu *job*, operasi-operasi dalam *job* diurutkan dalam suatu urutan pengerjaan tertentu. Suatu operasi baru dapat dikerjakan apabila operasi atau proses yang mendahuluinya sudah dikerjakan terlebih dahulu. Setiap operasi mempunyai waktu proses, yaitu waktu pengerjaan yang diperlukan untuk melakukan operasi tersebut. Waktu proses operasi untuk suatu *job* biasanya telah diketahui sebelumnya dan mempunyai nilai tertentu.

c. Mesin

Mesin adalah sumber daya yang diperlukan untuk mengerjakan proses penyelesaian suatu *job*. Setiap mesin hanya dapat memproses suatu *job* pada saat tertentu.

Permasalahan penjadwalan merupakan pengalokasian *job* ke mesin, pada kondisi mesin yang mempunyai kapasitas dan jumlah terbatas. Secara umum masalah penjadwalan dapat dijelaskan sebagai n *job* ($J_1, J_2, J_3, \dots, J_n$) yang diproses di m mesin ($M_1, M_2, M_3, \dots, M_m$).

Adapun tujuan dari aktivitas penjadwalan adalah sebagai berikut (Ginting 2009):

- a. meningkatkan penggunaan sumber daya atau mengurangi waktu tungguanya sehingga total waktu proses dapat berkurang dan produktivitas dapat meningkat;

- b. mengurangi persediaan barang setengah jadi atau mengurangi sejumlah *job* yang menunggu dalam antrian ketika sumber daya yang ada masih mengerjakan tugas yang lain;
- c. mengurangi beberapa keterlambatan pada *job* yang mempunyai batas waktu penyelesaian sehingga akan meminimasi *penalty cost* (biaya keterlambatan);
- d. membantu dalam pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan sehingga penambahan biaya yang mahal dapat dihindarkan.

Proses penjadwalan akan menyelesaikan masalah penentuan mesin yang akan digunakan (pengalokasian mesin) untuk menyelesaikan suatu proses produksi dan penentuan waktu suatu mesin dikerjakan (pengurutan). Hal itu dilakukan pada seluruh *job* sampai seluruh *job* dijadwalkan dan memenuhi setiap batasan masalah yang dirumuskan. Terdapat dua target yang ingin dicapai dalam pengalokasian mesin yaitu jumlah output yang dihasilkan, serta batas waktu penyelesaian yang telah ditetapkan. Kedua target itu dinyatakan melalui kriteria penjadwalan, misalnya *minimasi makespan* yaitu jumlah waktu yang dibutuhkan untuk menyelesaikan seluruh proses pada semua operasi yang dijadwalkan, mulai dari saat pemrosesan operasi pertama sampai operasi terakhir selesai dilaksanakan.

Ada beberapa jenis model penjadwalan *flowshop* dan penjadwalan *jobshop*. Penjadwalan *flowshop* mempunyai pola aliran kerja dari urutan tertentu yang sama untuk setiap *job* sedangkan penjadwalan *jobshop* mempunyai pola aliran kerja yang berbeda untuk setiap *job*.

2.2 Penjadwalan *Jobshop*

Penjadwalan pada proses produksi tipe *jobshop* lebih sulit dibandingkan penjadwalan *flowshop*. Hal ini disebabkan oleh 3 alasan, yaitu:

- a. *jobshop* menangani variasi produk yang sangat banyak, dengan pola aliran yang berbeda-beda melalui pusat-pusat kerja.

- b. peralatan pada *jobshop* digunakan bersama-sama oleh bermacam-macam order pada prosesnya, sedangkan peralatan pada *flowshop* digunakan khusus untuk satu jenis produk.
- c. *job-job* yang berbeda-beda mungkin ditentukan oleh prioritas berbeda pula. Hal ini mengakibatkan produk tertentu yang dipilih harus diproses seketika pada saat order tersebut ditugaskan pada suatu pusat kerja. Sedangkan pada *flow shop* tidak terjadi permasalahan seperti tersebut karena keseragaman output yang diproduksi untuk persediaan. Prioritas order pada *flow shop* dipengaruhi terutama pada pengirimannya dibandingkan tanggal pemrosesan.

Faktor-faktor tersebut menghasilkan sangat banyak kemungkinan kombinasi dari pembebanan (*loading*) dan urutan-urutan (*sequencing*). Perhitungan dari identifikasi dan evaluasi jadwal-jadwal yang mungkin menjadi sulit sehingga banyak perhatian diarahkan pada riset penjadwalan *jobshop*. Selain itu, persiapan suatu penjadwalan *jobshop*, penyesuaian dan pembaharuannya membutuhkan investasi yang besar untuk fasilitas komputer.

Pada bagian ini akan dibahas masalah penjadwalan *jobshop* dengan memperhatikan permasalahan pada *job loading* dan *job sequencing*.

Adapun masalah *jobshop* tersebut sebagai berikut :

- a. *jobshop loading* artinya memutuskan pusat-pusat kerja yang mana suatu *job* harus ditugaskan. Menggunakan *Gantt chart* dan metode penugasan.
- b. *job sequencing* artinya kita harus menentukan bagaimana urutan proses dari bermacam-macam *job* harus ditugaskan pada mesin-mesin tertentu atau pusat kerja tertentu.

2.2.1 *Jobshop Loading*

Ketika order-order pada suatu *job shop*, kegiatan pertama dari penjadwalan adalah menugaskan *order-order* tersebut tersebut kepada bermacam-macam pusat-pusat kerja untuk diproses. Permasalahan *loading* menjadi lebih sederhana ketika suatu *job* tidak dapat dipisah. Meskipun hal ini sering terjadi, biasanya suatu industri

sering dalam prakteknya melakukan pemisahan *job* dan menugaskan bagian-bagian terpisah dari *job* tersebut kepada pusat-pusat yang berbeda untuk tujuan meningkatkan *utilitas* sumber daya. Untuk permasalahan yang sederhana, kita mengasumsikan tidak ada pemisahan *job*, maka *jobshop loading* dapat dibuat dengan mudah dengan menggunakan *Gantt Chart* dan *Metode Penugasan*.

Loading dengan *Gantt Chart* merupakan cara yang paling sederhana dan paling banyak digunakan untuk bermacam-macam aktivitas penjadwalan. Meskipun sederhana dan tervisualisasikan, *Gantt Chart* sangat lemah dalam mengevaluasi rencana-rencana alternative untuk loading. Pengguna harus menggunakan cara *trial error* dan improvisasi jadwal. Bila jumlah *job* meningkat, proses ini menjadi cukup sulit dan tidak layak.

Loading dengan metode penugasan merupakan cara pembebanan pekerja-pekerja untuk *job-job* yang tersedia dengan tujuan meminimasi total waktu kerja atau total biaya kerja. Metode Hungarian adalah metode yang biasa dipakai untuk permasalahan ini. Dalam situasi yang lebih kompleks, permasalahan loading dapat diformulasikan sebagai suatu bentuk problem transportasi.

2.2.2 *Jobshop Sequencing*

Sekali beberapa *job* telah ditugaskan (loading) pada pusat kerja tertentu, maka langkah berikutnya adalah menentukan urutan-urutan memprosesnya. Pemrosesan order merupakan hal yang penting karena mempengaruhi lamanya suatu *job* akan diproses dalam sistem tertentu. Lamanya *job* dalam proses ini akan mempengaruhi batas waktu janji pengiriman kepada konsumen. Yang tidak kalah pentingnya adalah pengaruh urutan-urutan pemrosesan *job* terhadap utilitas sumberdaya-sumberdaya organisasi, Khususnya pada kondisi suplai yang kritis.

Penjadwalan *Jobshop* meliputi aturan-aturan prioritas *sequencing*. Aturan-aturan prioritas *sequencing* diaplikasikan untuk seluruh *job* yang sedang menunggu dalam antrian. Bila pusat kerja real lowong untuk satu *job* baru, maka *job* dengan prioritas terdahulu akan diproses. Pemilihan prioritas *sequencing* tersebut

mempertimbangkan efisiensi penggunaan fasilitas dengan kriteria antara lain biaya setup, biaya persediaan WIP, waktu menganggur stasiun kerja, presentase waktu menganggur, rata-rata jumlah *job* yang menunggu, dan sebagainya.

Beberapa aturan *sequencing* yang umum antara lain adalah sebagai berikut:

a. First-Come-First-Served (FCFS)

Job yang datang diproses sesuai dengan *job* mana yang datang terlebih dahulu.

b. Earlist Due Date (EDD)

Prioritas diberikan kepada *job-job* yang mempunyai tanggal batas waktu penyerahan (*due date*) paling awal.

c. Shortest Processing Time (SPT)

Job dengan waktu proses terpendek akan diproses terlebih dahulu, demikian berlanjut untuk *job* yang waktu prosesnya terpendek kedua. Aturan SPT ini tidak memperdulikan *due date* maupun kedatangan order baru.

Beberapa kasus yang akan dibahas pada bagian penjadwalan *jobshop* ini adalah *jobshop* dengan pola kedatangan statis. Beberapa buku mendefinisikan *jobshop* dengan pola kedatangan *statis* sebagai suatu penjadwalan *jobshop* dengan urutan proses sama, atau disebut juga *Flowshop Scheduling*. Penjadwalan ini akan melibatkan permasalahan *job loading* dan *job sequencing* untuk kasus tanpa dugaan ataupun dengan *due date* sebagai berikut :

a. Penjadwalan “*n*” job dengan “satu prosesor”

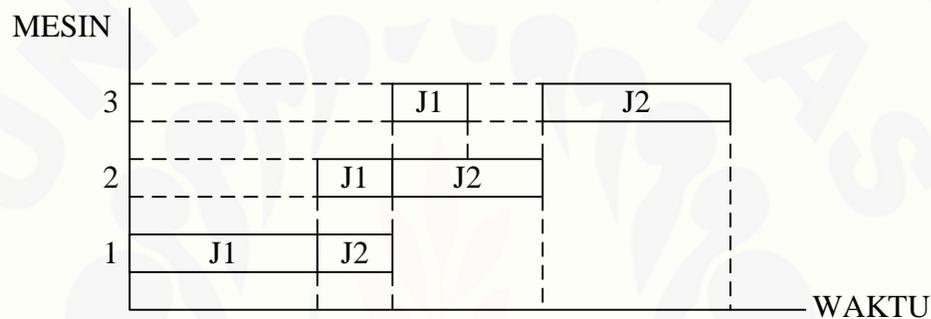
b. Penjadwalan “*n*” job pada “*m*” prosesor, baik untuk penjadwalan paralel maupun penjadwalan seri.

Adapun kriteria utama yang digunakan untuk mengevaluasi penjadwalan *jobshop* ini adalah *Makespan*, *Flow Time* dan *Tardiness*.

2.3 Diagram Gantt (*Gantt Chart*)

Sebagai alat bantu untuk menyelesaikan masalah penjadwalan ini maka digunakan *Gantt Chart*. Masalah penjadwalan adalah masalah murni pengalokasian dan dengan bantuan model matematis akan dapat ditentukan solusi optimal. Model-

model penjadwalan akan memberikan rumusan masalah yang sistematis dengan solusi yang diharapkan. *Gantt Chart* dikenalkan oleh Henry Laurence Gantt (1916). *Gantt Chart* merupakan grafik hubungan antar alokasi sumber daya dengan waktu. Gambar merupakan grafik hubungan antar alokasi sumber daya dengan waktu. Pada sumbu vertikal digambarkan jenis sumber daya yang digunakan sedangkan pada sumbu horisontal digambarkan satuan waktu. Adapun contoh *Gantt Chart* dapat dilihat pada contoh gambar 2.1 berikut .



Gambar 2.1 Diagram *Gantt Chart*

2.4 Pengertian Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar (Shofwatul'uyun, 2009).

Sebuah algoritma tidak saja harus benar, tetapi juga harus efisien. Algoritma yang bagus adalah algoritma yang efisien. Keefisienan algoritma diukur dari berapa jumlah waktu dan ruang (*space*) memori yang dibutuhkan untuk menjalankan. Algoritma yang efisien ialah algoritma yang meminimumkan kebutuhan waktu dan ruang. Kebutuhan waktu dan ruang suatu algoritma bergantung pada ukuran masukan, yang secara khas adalah jumlah data yang diproses. Ukuran masukan itu disimbolakan dengan n . Waktu atau ruang yang dibutuhkan oleh algoritma

dinyatakan sebagai fungsi dari n . Bila n meningkat, maka sumberdaya waktu atau ruang yang dibutuhkan juga meningkat. Seberapa besar peningkatan sumberdaya itu menentukan apakah algoritmanya efisien atau tidak (Munir, 2005).

2.5 Algoritma *Harmony Search*

Algoritma *Harmony Search* pertama kali dikembangkan oleh Geem pada tahun 2001. Ide dasar algoritma *Harmony Search* adalah melalui proses perbaikan harmony musik yang dilakukan kelompok paduan musik. Algoritma *Harmony Search* meniru fenomena proses improvisasi musik. Ada tiga kemungkinan pilihan yang akan dimainkan untuk melakukan perbaikan pada harmoni musik, yaitu memainkan harmoni musik yang terkenal berdasarkan ingatan kelompok paduan musik tersebut, memainkan harmoni musik yang terkenal namun ada sedikit penyesuaian, atau membuat harmoni musik yang baru (Suyanto, 2010).

Setiap musisi dianalogikan sebagai variable keputusan, rentang *pitch* dari instrument musik dianalogikan sebagai rentang nilai variable keputusan, harmoni music dianalogikan sebagai vektor solusi dan penonton dianalogikan sebagai fungsi objektif. Sama seperti harmoni musik yang mengalami improvisasi dari waktu ke waktu, vector solusi juga mengalami improvisasi dari iterasi ke iterasi (Hanggraeni, 2014).

Usaha untuk menemukan harmony yang sempurna dianalogikan dengan proses penemuan solusi optimal dalam optimasi. Penemuan solusi optimal tersebut harus dicari berdasarkan fungsi tujuan (minimasi atau maksimasi) dan konstrain (batasan) yang ada.

2.5.1 Parameter Algoritma *Harmony Search*

Parameter yang akan digunakan pada algoritma *Harmony Search*, sebagai berikut (Hapsari, 2014).

a. *Harmony Memory Size (HMS)*

Jumlah baris dari matrik *harmony memory* yang akan digunakan pada saat inisialisasi *harmony memory*. HMS merupakan sampel kemungkinan solusi dari jumlah solusi yang ada.

b. *Harmony Memory Considering Rate (HMCR)*

HMCR bernilai $0 \leq \text{HMCR} < 1$, namun nilai yang digunakan umumnya berkisar antara 0,7 – 0,95 karena jika nilai HMCR terlalu kecil maka hanya sedikit harmoni baik yang terpilih dan juga dapat menyebabkan proses konvergensi terlalu lambat. Jika nilai HMCR terlalu besar maka akan menyebabkan vektor solusi pada *harmony memory* banyak terpakai dan tidak sempat mengeksplorasi yang lain, dimana pada akhirnya sulit mencapai solusi optimal.

c. *Pitch Adjusting Rate (PAR)*

PAR bernilai $0 \leq \text{PAR} \leq 1$, namun nilai yang digunakan umumnya berkisar antara 0,1 – 0,5 karena jika PAR bernilai kecil maka keterbatasan eksplorasi pada ruang pencarian yang besar dapat menyebabkan proses konvergensi lambat. Jika PAR bernilai besar maka dapat menyebabkan solusi-solusi yang ada terlalu menyebar dari potensi solusi optimal.

d. *Bandwidth*

Bandwidth merupakan nilai perubahan maksimal dalam penyesuaian nada. Rentang dari *Bandwidth* berkisar antara 0,001 – 0,01.

e. Kriteria Pemberhentian

Kriteria Pemberhentian adalah *Number of Improvisation (NI)* yang mempresentasikan jumlah iterasi.

2.5.2 Mekanisme Algoritma *Harmony Search*

Mekanisme *Harmony Search* adalah sebagai berikut (Geem, 2001).

a. Formulasi masalah dan parameter

Masalah optimasi yang akan diselesaikan menggunakan algoritma *Harmony Search* yaitu :

(Minimasi atau maksimasi) $f(x_i)$,

Dimana $x_i \in X_i ; i = 1, 2, \dots, N$, (2.1)

dengan:

$f(x_i)$ = fungsi tujuan

x_i = variable keputusan ke- i

X_i = himpunan variable keputusan ke- i

N = jumlah variable keputusan

Parameter yang digunakan yaitu HMS, HMC, PAR, *bandwidth* dan Kriteria Pemberhentian (jumlah iterasi).

b. Inisialisasi *Harmony Memory*

Harmony Memory berisi vector-vector solusi yang dibangkitkan secara *random* sebanyak HMS, sehingga terbentuk matrik *harmony memory* seperti dibawah ini:

$$HM = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1N-1} & X_{1N} \\ X_{21} & X_{22} & \dots & X_{2N-1} & X_{2N} \\ \vdots & \dots & \dots & \dots & \dots \\ X_{(HMS-1)1} & X_{(HMS-1)2} & \dots & X_{(HMS-1)N-1} & X_{(HMS-1)N} \\ X_{(HMS)1} & X_{(HMS)2} & \dots & X_{(HMS)(N-1)} & X_{(HMS)N} \end{bmatrix}$$

Dari masing-masing vektor solusi akan dievaluasi nilai fungsinya,

$$f(x) = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(X_{HMS-1}) \\ f(X_{HMS}) \end{bmatrix}$$

dengan:

$f(x)$ = fungsi tujuan

HM = matriks *harmony memory*

N = jumlah variabel keputusan

HMS = jumlah baris matrik *harmony memory*

$X_{HMS} N$ = variabel keputusan ke- N pada sebanyak HMS

c. Membuat vektor solusi yang baru

Untuk membuat vektor solusi yang baru $x' = (x'_1, x'_2, \dots, x'_n)$, didasarkan pada aturan tertentu yang dilakukan dengan membandingkan nilai yang dibangkitkan secara *random* dengan HMCR dan PAR. Pembangkit vektor solusi yang baru secara *random* didasarkan pada aturan berikut .

1) Penggunaan *harmony memory*

Pembangkitan vektor solusi yang baru diambil dari HM, jika nilai pertama dari bilangan *random* yang dibangkitkan lebih kecil dari HMCR dan nilai kedua dari bilangan *random* yang dibangkitkan lebih besar dari PAR.

Formulasinya sebagai berikut :

$$d_1 = \text{int}[1 + (\text{HMS} - 1)\text{rand}] \quad (2.2)$$

$$d_2 = \text{HM}(d_1, i)$$

$$x'_i = d_2$$

dengan:

d_1 = nilai yang menyatakan pemilihan lokasi variabel keputusan pada *harmony memory* secara *random*;

int = integer ;

d_2 = nilai yang menyatakan pemilihan lokasi variabel keputusan yang diambil dari *harmony memory*.

2) Penyesuaian nada

Pembangkitan vector solusi yang baru dilakukan melalui proses penyesuaian nada, jika nilai pertama dari bilangan *random* yang dibangkitkan lebih kecil dari HMCR dan nilai kedua bilangan *random* yang dibangkitkan lebih kecil dari PAR. Rumus penyesuaian nada yaitu:

$$x'_i = d_2 + bw \cdot (\text{rand} [-1,1]) \quad (2.3)$$

dengan,

x'_i = nada baru setelah dilakukan penyesuaian nada

bw = *bandwidth*.

3) Proses pembangkitan secara *random*

Pembangkitan vektor solusi yang baru secara *random* dengan nilai interval yang memungkinkan ($x_i \in X_i$). Hal ini terjadi apabila nilai pertama dari bilangan *random* dibangkitkan lebih besar dari HMCR, tidak perlu membangkitkan nilai kedua karena nilai yang dibangkitkan pertama sudah lebih besar dari HMCR.

d. Memperbarui *Harmony Memory*

Jika vektor solusi yang baru lebih baik daripada vector solusi di HM, maka vektor solusi baru dimasukkan ke dalam HM dan vector solusi yang diganti dikeluarkan.

e. Mengecek kriteria pemberhentian

Kriteria pemberhentian dari algoritma *Harmony Search* yaitu jumlah iterasi yang telah ditentukan pada tahap inisialisasi parameter. Jika criteria pemberhentian telah tercapai maka proses pengerjaan dihentikan tetapi jika belum tercapai maka kembali pada mekanisme tiga.

2.6 Algoritma *Artificial Bee Colony*

Algoritma *Artificial Bee Colony* (ABC) merupakan algoritma yang meniru tingkah laku kawanan lebah dalam mencari dan mengeksplorasi sumber-sumber makanan secara efisien. Ketika beberapa ekor lebah menemukan sumber-sumber makanan, mereka akan mengundang lebah lainnya melalui tarian yang dilakukan di lokasi tertentu. Informasi mengenai sumber makanan tersebut terdiri dari tiga hal yaitu arah, jarak dari sarang dan kualitas (jumlah nektar). Semakin bagus kualitas sumber makanan, semakin lama durasi tarian yang dilakukan sehingga semakin banyak lebah yang mengikuti ajakannya. Ketika sumber makanan sudah habis, maka lebah akan memberitahukan informasi ini kepada lebah lainnya. Dengan demikian sumber-sumber makanan bisa dihabiskan dalam waktu secepat mungkin dan dengan menggerakkan lebah pencari makanan dengan jumlah seminimum mungkin (Suyanto, 2010).

Algoritma ABC merupakan salah satu algoritma optimasi yang berdasarkan pada kecerdasan kelompok dari perilaku lebah. Model ini dikenalkan oleh Dervis Karaboga pada tahun 2005, berdasarkan perilaku lebah dalam mencari sumber makanan serta cara lebah berkomunikasi untuk mendapatkan informasi kepada lebah yang berada di sarang tentang letak sumber makanan. Pada model ini, lebah dibagi 3 tipe yaitu Lebah pekerja (*Employed bee*), Lebah penjaga dan penyimpan makanan (*Onlooker bee*) dan Lebah Pengintai (*Scout bee*).

Setiap tipe lebah memiliki peran atau tugas yang berbeda dalam pencarian makanan. Lebah pekerja bertugas untuk mencari sumber makanan dan menginformasikan tentang letak sumber makanan kepada lebah penjaga. Lebah penjaga bertugas menyimpan makanan dan menjaga sarang saat lebah pekerja mencari sumber makanan dan menentukan jalur untuk mendapatkan sumber makanan sesuai dengan informasi yang disarankan oleh lebah penjaga. Lebah pengintai bertugas untuk mengikuti jalur yang disarankan oleh lebah penjaga untuk menemukan sumber makanan (Chong, 2006).

Untuk langkah kerjanya sebagai berikut :

- a. Beberapa lebah pekerja meninggalkan sarangnya untuk mencari sumber makanan. Kemudian para lebah pekerja akan menyimpan informasi yang didapatnya dan menghitung nilai *profitability* dari sumber makanan yang dikunjunginya,
- b. Lebah pekerja kembali ke sarang dan menginformasikan letak sumber makanan kepada lebah penjaga dengan menggunakan tarian *Waggle Dance*, dimana dalam tarian tersebut terdapat 3 informasi yaitu arah, jarak, dan kualitas nektar,
- c. Lebah penjaga yang mendapatkan informasi dari lebah pekerja akan memilih sumber makanan mana yang memiliki nilai *probability* tertinggi untuk dijadikan tujuan sebagai titik awal untuk pencarian secara *random* sumber makanan berikutnya oleh lebah pengintai.

Adapun langkah-langkah penjadwalan *jobshop* menggunakan algoritma ABC adalah sebagai berikut:

a. Inisialisasi solusi awal

Sebelum perhitungan generasi dimulai, dilakukan input parameter-parameter, yaitu ukuran jumlah populasi lebah, jumlah lebah pengintai, dan panjang *list* solusi yang akan digunakan, serta criteria berhenti yaitu jumlah iterasi yang dipakai. Inisialisasi awal menggunakan solusi yang diperoleh secara acak. Data kemudian diperluas dengan persamaan sebagai berikut.

$$x_{ij} = x_{j \min} + \text{rand}(0,1) \cdot (x_{j \max} - x_{j \min}) \quad (2.4)$$

dengan:

x_{ij} = inisialisasi kemungkinan solusi ke- i dengan parameter ke- j

$x_{j \min}$ = nilai kemungkinan solusi terkecil berdasarkan parameter j

$x_{j \max}$ = nilai kemungkinan solusi terbesar berdasarkan parameter j

$\text{rand}(0,1)$ = nilai random antara 0 sampai 1

i = 1 sampai jumlah kemungkinan solusi (sumber makanan)

j = 1 sampai jumlah parameter yang digunakan

b. Tahap penentuan solusi alternatif

Solusi awal akan dijadikan acuan sejumlah n lebah untuk dilakukan pencarian sumber nektar, sehingga didapatkan sejumlah n solusi alternatif, menggunakan persamaan sebagai berikut.

$$v_{ij} = x_{ij} + \phi_{ij} \cdot (x_{kj} - x_{ij}) \quad (2.5)$$

dengan:

v_{ij} = nilai perluasan kemungkinan solusi ke- i dengan parameter j

x_{ij} = nilai kemungkinan solusi ke- i dengan parameter j

ϕ_{ij} = bilangan real *random* antara $[-1,1]$

K = 1 sampai jumlah parameter yang digunakan

c. Tahap evaluasi populasi awal

Solusi alternatif yang didapat pada tahap b, lalu dilakukan pemilihan solusi sejumlah panjang *list* solusi yang telah ditentukan. Dengan kriteria yang digunakan memilih hasil yang terbaik. Lalu dilakukan pemilihan secara acak untuk

solusi-solusi yang terbaik untuk dijadikan acuan pencarian sumber nektar oleh n lebah. Solusi yang terbaik akan dimasukkan pada *list* solusi. Solusi terbaik langkah-langkahnya akan ditabukan untuk mencegah terjebaknya pada optimum lokal dan persamaan sebagai berikut.

$$fitness(x_i) = \begin{cases} \frac{1}{(1+f(x_i))}, & f(x_i) \geq 0 \\ 1 + f(x_i), & f(x_i) < 0 \end{cases} \quad (2.6)$$

dengan $f(x_i) = v_{ij}$

d. Tahap evaluasi populasi alternatif

Solusi yang didapat oleh n lebah akan dibandingkan kembali dengan solusi yang ada dalam *list*, jika solusi baru memiliki nilai yang lebih baik akan menggantikan solusi lama sebagai berikut:

$$p_i = \frac{fitness}{\sum_{i=1}^{SN} fitness \ i} \quad (2.7)$$

p_i = nilai *probability*

fitness = nilai *fitness* solusi ke- i

$\sum_{i=1}^{SN} fitness \ i$ = jumlah dari nilai *fitness* ke- i sampai SN

e. Tahap evaluasi populasi akhir

Pada tahap ini pengulangan proses dilakukan hanya pada tahap b dan c hingga kriteria berhenti.

f. Kriteria Berhenti

Kriteria ini yang digunakan adalah jumlah iterasi, proses pembentukan iterasi baru akan terus berulang sampai jumlah iterasi yang telah ditentukan tercapai.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini adalah data sekunder milik Junida (2009) yang kemudian digunakan kembali oleh Shandiputra (2011) dan Tri Gunarso (2014) untuk penjadwalan pabrik springbed PT. Cahaya Kawi Ultra Polyintraco. Pabrik ini memproduksi 4 tipe springbed yaitu platinum, silver, golden dan bigline. Perbedaan 4 jenis springbed didasarkan pada kain quilting, rakitan per-bulat dan busa yang digunakan. Data ini berupa urutan mesin pada masing-masing *job*. Setiap *job* memiliki 5 urutan operasi sehingga total urutan pekerjaan sebanyak 20 dengan 4 *job* dan 5 mesin. Untuk lebih jelasnya data tersebut disajikan dalam Tabel 3.1

Tabel 3.1 Data Penelitian

Job	Operasi 1	Operasi 2	Operasi 3	Operasi 4	Operasi 5
1	(3;66)	(1;91)	(4;8)	(2;48)	(5;20)
2	(2;48)	(3;66)	(1;91)	(4;8)	(5;18)
3	(4;8)	(1;86)	(3;63)	(2;45)	(5;14)
4	(4;8)	(3;61)	(1;85)	(2;44)	(5;12)

Keterangan :

1 = Platinum

2 = Golden

3 = Silver

4 = Bigline

Pada Tabel 3.1 diatas, angka (3;66) pada baris dan kolom kedua menyatakan *job* 1 urutan operasi 1 masuk ke mesin 3. Waktu yang dibutuhkan untuk operasi ini adalah 66 menit. Begitu juga untuk keterangan pada operasi dan *job* yang lainnya.

3.2 Langkah Penyelesaian

Langkah-langkah yang akan dilakukan dalam menyelesaikan permasalahan penjadwalan *jobshop* adalah sebagai berikut (lihat gambar 3.1):

a. Studi Literatur

Studi literatur yang dilakukan pada penelitian ini adalah mempelajari mengenai penjadwalan *jobshop*, *gantt chart*, algoritma *Harmony Search*, dan algoritma *Artificial Bee Colony* yang didapat dari berbagai sumber baik berupa buku, artikel, maupun jurnal yang diperoleh melalui internet;

b. Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil data yang ada pada skripsi milik Tri Gunarso (2014);

c. Pembuatan Program

Pembuatan program menggunakan *software* matematika *MATLAB*, pada langkah ini dibuat skrip dan desain program sesuai dengan algoritma yang digunakan;

d. Simulasi Program

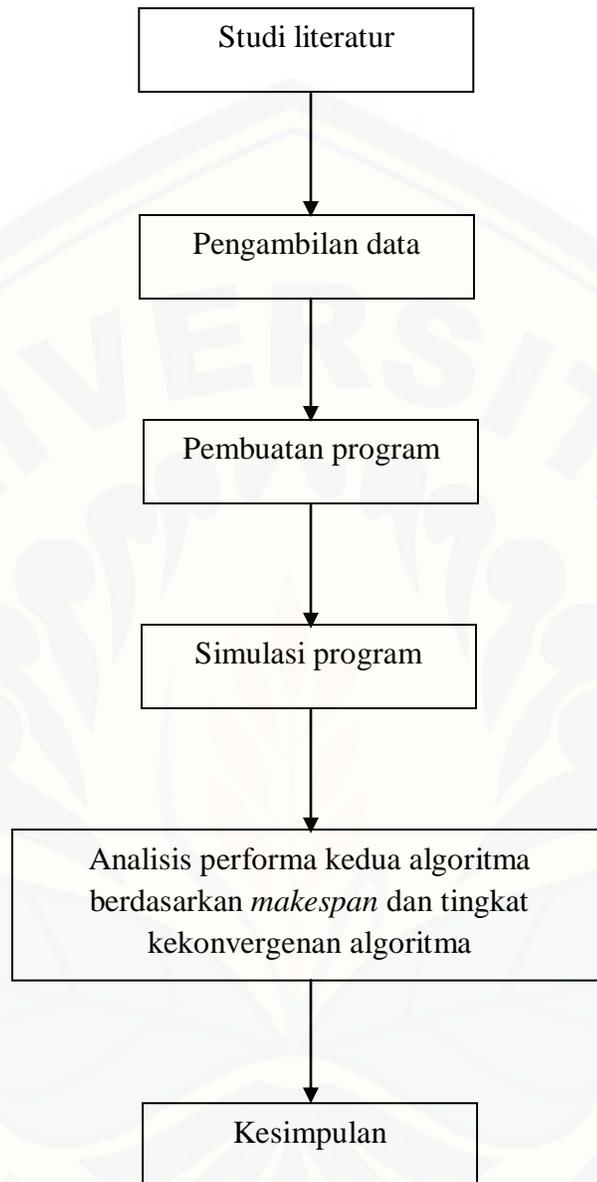
Simulasi program dilakukan dengan menggunakan beberapa data yang telah dipilih untuk tes performa suatu algoritma untuk penjadwalan *jobshop*. Data-data ini disimulasikan dengan menggunakan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony*;

e. Analisis Performa

Analisis performa dilakukan untuk mengetahui nilai *makespan* yang paling minimum yaitu waktu yang lebih efektif serta tingkat kekonvergenan dari algoritma *Harmony Search* dan algoritma *Artificial Bee Colony*;

f. Kesimpulan

Membuat kesimpulan berdasarkan analisis yang telah dilakukan.



Gambar 3.1 Skema Langkah-langkah Penelitian

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil dari penerapan perbandingan Algoritma *Harmony Search* dan Algoritma *Artificial Bee Colony* pada industri pabrik springbed PT. Cahaya Kawi Ultra Polyintraco dengan menggunakan program matlab. Selain melalui program yang akan menghasilkan makespan minimum, kedua algoritma tersebut juga akan dibandingkan berdasarkan kekonvergenan masing-masing.

4.1 Hasil

Data dalam penelitian ini adalah data sekunder berupa pengamatan mesin pembuatan 4 jenis *springbed*, yaitu *Platinum*, *Golden*, *Silver*, dan *Bigline*. Data tersebut berupa data nomer mesin dan waktu proses pembuatan tiap komponen dari masing-masing mesin. Jumlah mesin yang digunakan untuk membuat *springbed* ada 9 jenis mesin yang direpresentasikan dengan nomer urut dan angka dalam menit untuk waktu proses mesin.

Springbed tersebut diproses oleh 9 jenis mesin yang pengelompokannya berdasarkan mesin yang dibutuhkan untuk membuat komponen tiap *springbed*. Jumlah jenis mesin pembuat masing-masing komponen akan berbeda-beda. Komponen keseluruhan yang dibuat hingga proses perakitan berjumlah 4 komponen inti yaitu matras, sandaran, divan dan kaki *springbed* dilanjutkan dengan 1 perlakuan terakhir yaitu perakitan komponen keseluruhan. Sehingga 9 jenis mesin pembuat *springbed* ini akan direduksi menjadi 5 mesin notasi baru sesuai dengan komponen-komponennya (lihat lampiran B) sehingga didapat waktu proses mesin pada 4 jenis *springbed* tersebut (lihat lampiran A).

4.1.1 Penjadwalan *Jobshop* dengan Program MATLAB

Pada skripsi ini dihasilkan sebuah program penjadwalan *jobshob* dengan menggunakan *software* MATLAB. Tujuan dari pembuatan program ini adalah untuk mendapatkan penjadwalan yang baik dengan membandingkan performa dari algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* yang nantinya akan

didapatkan *makespan* paling minimum serta tingkat konvergensi dari algoritma-algoritma tersebut. Tampilan awal dari program yang telah dibuat disajikan pada Gambar 4.1.



Gambar 4.1 Tampilan Awal Program Penjadwalan *Jobshop*

Berdasarkan Gambar 4.1 diatas merupakan tampilan awal GUI program algoritma *Harmony Search* serta algoritma *Artificial Bee Colony*, dimana tampilan awal terdiri dari tombol-tombol dan kolom yang memiliki pengertian sebagai berikut:

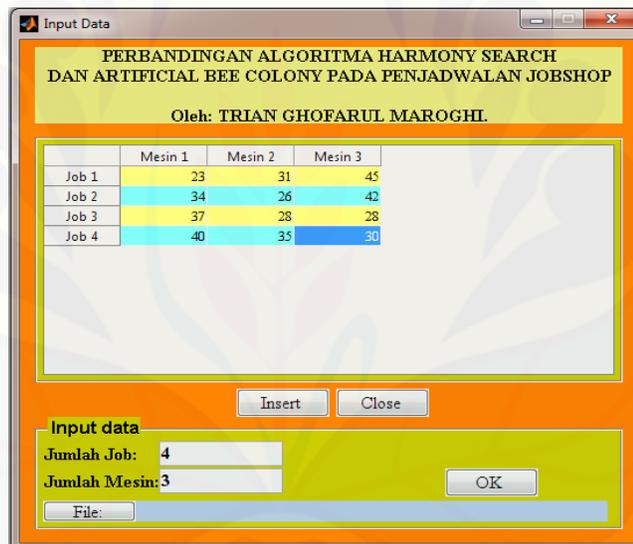
1. Input data, dimana akan diinputkan data sejumlah *job* x mesin yang akan diproses.
2. Jumlah *job*, adalah banyaknya pekerjaan yang dikerjakan dalam suatu produksi.
3. Jumlah mesin, adalah banyaknya mesin yang digunakan dalam suatu produksi.
4. Tombol file, adalah berisi data yang sebelumnya sudah disimpan dalam program untuk diproses.
5. Tombol Ok, digunakan untuk memunculkan data yang telah dipilih atau diinputkan.
6. Tombol Insert, digunakan untuk memproses data menggunakan algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* setelah memilih atau menginputkan data.

7. Tombol Close, digunakan untuk menutup tampilan program.

Adapun langkah-langkah yang dilakukan untuk menjalankan program akan dijelaskan sebagai berikut:

a. Input data

Untuk menginput data ada dua cara yang dilakukan tergantung data yang kita inginkan. Cara yang pertama yaitu menggunakan data manual. Isikan angka pada kotak jumlah *job* dan jumlah mesin sesuai keinginan lalu klik tombol “OK”. Setelah itu akan muncul tabel pada jendela input data sebanyak *job* x mesin . Kemudian kita isikan angka sesuai keinginan dan selanjutnya klik tombol “Insert” untuk diproses.



The screenshot shows a window titled "Input Data" with the following content:

PERBANDINGAN ALGORITMA HARMONY SEARCH
DAN ARTIFICIAL BEE COLONY PADA PENJADWALAN JOBSHOP
Oleh: TRIAN GHOFARUL MAROGHI.

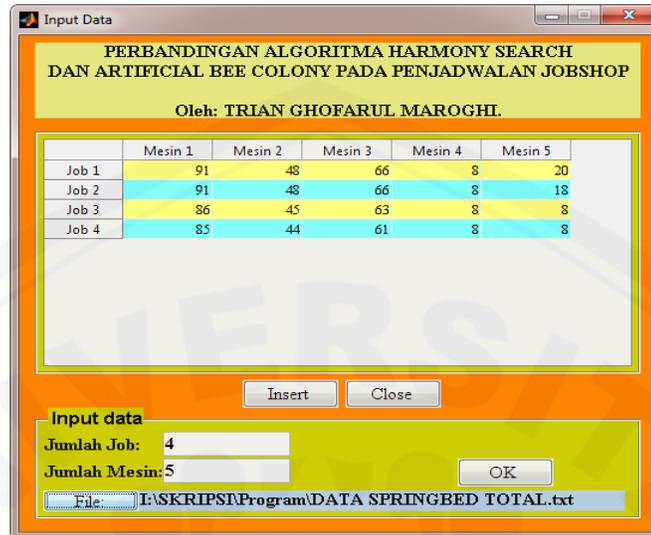
	Mesin 1	Mesin 2	Mesin 3
Job 1	23	31	45
Job 2	34	26	42
Job 3	37	28	28
Job 4	40	35	30

Below the table are buttons for "Insert" and "Close".

Input data
Jumlah Job: 4
Jumlah Mesin: 3
File: OK

Gambar 4.2 Tampilan Tabel Data Manual

Cara kedua adalah dengan memunculkan data yang sudah disimpan terlebih dahulu. Klik tombol “File” kemudian pilih data yang diinginkan. Selanjutnya klik tombol “Open”. Maka data yang dipilih akan dimunculkan pada jendela input data dalam bentuk tabel. Kemudian klik tombol “Insert” untuk diproses.



Gambar 4.3 Tampilan Tabel Data dalam Dokumen

b. Proses Data

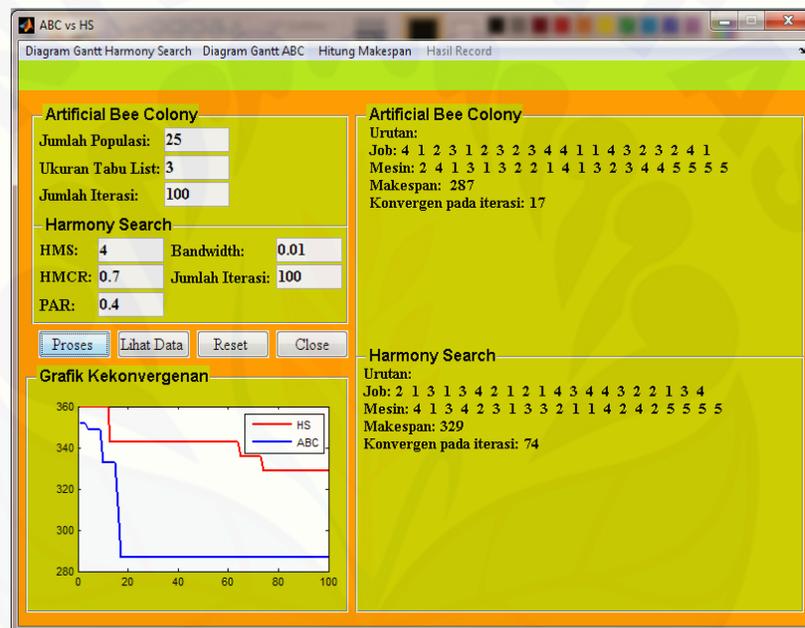
Pada langkah ini akan muncul jendela hasil insert data untuk diproses. Terdapat sejumlah kotak parameter yang akan diinputkan untuk algoritma *Harmony Search* dan algoritma *Artificial Bee Colony*. Inputkan nilai-nilai parameter jumlah populasi, ukuran tabu list, dan jumlah iterasi pada algoritma *Artificial Bee Colony* serta parameter HMS, HMCR, PAR, Bandwidth, dan juga jumlah iterasi pada algoritma *Harmony Search*. Klik tombol “Proses” untuk memproses data yang telah diinputkan nilai parameternya.



Gambar 4.4 Tampilan Input Parameter Kedua Algoritma

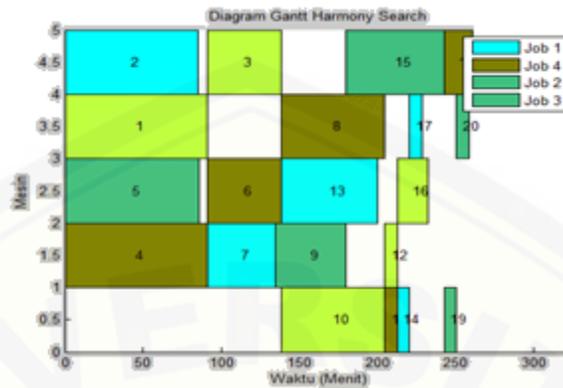
c. Output data

Pada tahap output setelah diproses maka akan didapat nilai makespan dan konvergensi dari kedua algoritma pada jendela sebelah kanan serta grafik kekonvergenan pada jendela sebelah kiri bawah. Terdapat fungsi tombol selain tombol “proses”, diantaranya tombol “Lihat Data” untuk menampilkan data awal, tombol “Reset” untuk kembali ke menu utama, serta tombol “Close” untuk menutup program.

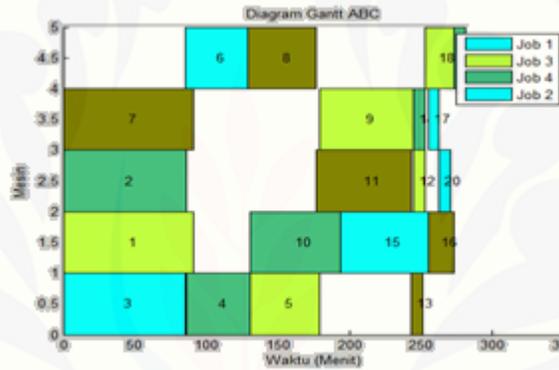


Gambar 4.5 Tampilan Output kedua algoritma

Pada tampilan output juga bisa diperlihatkan Diagram *Gantt* dan “Hasil Record” dari algoritma *Harmony Search* maupun algoritma *Artificial Bee Colony* dengan mengklik menu sebelah atas. Sedangkan menu “Hitung Makespan” untuk menghitung *makespan data* yang digunakan dengan urutan penjadwalan acak. Adapun *Gantt Chart* yang dihasilkan kedua algoritma dapat dilihat pada Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Tampilan *Gantt Chart* HS



Gambar 4.7 Tampilan *Gantt Chart* ABC

4.1.2 Perhitungan Manual

Untuk dapat mengetahui langkah-langkah kerja dari algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* dalam menyelesaikan permasalahan *jobshop* diperlukan perhitungan manual. Data yang digunakan berupa data waktu proses produksi 4 *job* dan 3 mesin yang dapat dilihat pada Tabel 4.2. Tujuan penghitungan manual adalah untuk memudahkan dalam memahami langkah-langkah kerja kedua algoritma tersebut.

Tabel 4.1 Waktu Proses Produksi (menit)

	M_1	M_2	M_3
J_1	23	31	45
J_2	34	26	42
J_3	37	28	28
J_4	40	35	30

a. Perhitungan manual dengan Algoritma *Harmony Search*

Berikut ini adalah penjadwalan manual *jobshop* menggunakan Algoritma *Harmony Search* dalam satu iterasi. Langkah-langkah pengerjaannya adalah sebagai berikut:

1) Inisialisasi Parameter

Parameter yang digunakan yaitu :

Jumlah Job = 4

Jumlah Mesin = 3

HMS = 4

HMCR = 0,7

PAR = 0,4

Bandwidth = 0,01

2) Inisialisasi *Harmony Memory*

Membangkitkan HM secara random dengan jumlah baris sama dengan nilai HMS dan jumlah kolom sama dengan jumlah *job* x mesin ($n \times m$) seperti ini :

HM=

0,60	0,65	0,82	0,92	0,43	0,39	0,15	0,81	0,64	0,78	0,26	0,55
0,22	0,82	0,71	0,65	0,30	0,90	0,48	0,83	0,77	0,87	0,52	0,44
0,86	0,39	0,58	0,19	0,71	0,47	0,51	0,83	0,18	0,28	0,11	0,69
0,73	0,64	0,47	0,38	0,15	0,85	0,72	0,82	0,50	0,22	0,25	0,98

Urutan HM secara mendatar dalam matriks adalah dimulai dari (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3) (4,1) (4,2) (4,3)

(1,1) hingga (4,3) adalah urutan *job* dengan mesin ($n \times m$). Maka nilai vektor solusi awalnya adalah sebagai berikut:

VS1	=
[0,60 0,65 0,82 0,92 0,43 0,39 0,15 0,81 0,64 0,78 0,26 0,55]	
VS2	=
[0,22 0,82 0,71 0,65 0,30 0,90 0,48 0,83 0,77 0,87 0,52 0,44]	
VS3	=
[0,86 0,39 0,58 0,19 0,71 0,47 0,51 0,83 0,18 0,28 0,11 0,69]	
VS4	=
[0,73 0,64 0,47 0,38 0,15 0,85 0,72 0,82 0,50 0,22 0,25 0,98]	

Kemudian diurutkan untuk mendapat jadwal solusi awal sebagai berikut:

Tabel 4.2 Jadwal Solusi Awal

VS	Jadwal	Makespan
1	(2,1),(1,3),(3,2),(4,1),(1,2),(3,3), (1,1),(4,3),(2,2),(2,3),(4,2),(3,1)	151
2	(2,3),(3,1),(4,2),(1,2),(3,3),(1,3), (2,1),(4,2),(3,1),(4,3),(2,2),(1,1)	141
3	(1,1),(3,2),(2,2),(4,3),(1,3),(3,1), (2,3),(1,2),(4,1),(2,1),(3,3),(4,2)	146
4	(4,3),(2,3),(3,2),(1,1),(3,1),(1,2), (3,3),(1,3),(2,1),(4,2),(4,1),(2,2)	147

Perhitungan *makespan* VS ke-1 dapat dilihat pada Tabel 4.3 berikut ini kemudian *makespan* VS ke-2, VS ke-3 dan VS ke-4 dihitung dengan cara yang sama.

Tabel 4.3 Perhitungan *makespan* VS ke-1

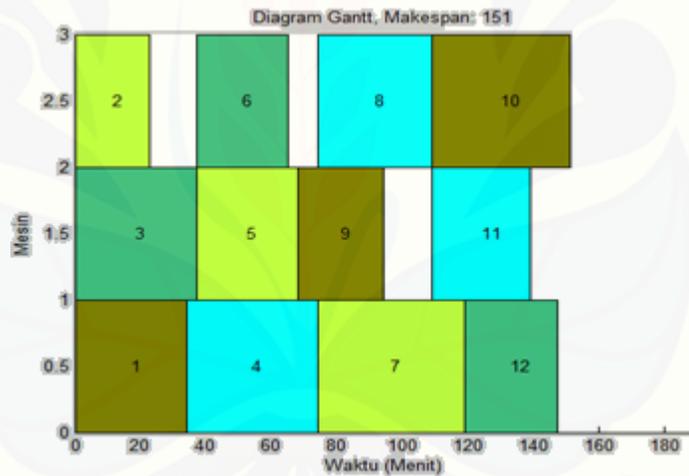
	<i>M1</i>		<i>M2</i>		<i>M3</i>	
	S	E	S	E	S	E
J1	0	23	37	68	74	119
J2	0	34	68	94	109	151
J3	0	37	37	65	119	147
J4	34	74	74	109	109	139

Keterangan :

S = Start

E = End

Gantt Chart dari jadwal VS ke-1 dengan perhitungan Start dan End adalah sebagai berikut :



Gambar 4.8 Gantt Chart VS-1

Dari tabel diketahui solusi awal, solusi terbaik (*makespan* best) terdapat pada VS ke-2 yaitu sebesar 141 menit, sedangkan solusi terburuk (*makespan* worst) terdapat pada VS ke-1 sebesar 151 menit.

3) Membuat vektor solusi baru

Pembangkitan Vektor Solusi Baru (VSB) menggunakan aturan yaitu membandingkan nilai yang dibangkitkan random dengan HMCR dan PAR.

a. Membangkitkan vektor solusi variabel pertama (x'_1)

$R1 = 0,39$ dibandingkan dengan $HMCR = 0,7$

Maka $R1$ lebih kecil daripada $HMCR$ sehingga variabel keputusan baru akan dipilih secara random dengan menggunakan persamaan 2.2

Hasil keputusan baru adalah

$R2 = 0,94$ dibandingkan dengan $PAR = 0,4$

Maka $R2$ lebih besar daripada PAR sehingga variabel keputusan baru tetap

Maka variabel keputusan baru pertama yang terbentuk adalah $x'_1 = 0,38$

b. Membangkitkan vektor solusi variabel kedua (x'_2)

$R1 = 0,39$ dibandingkan dengan $HMCR = 0,7$

Maka $R1$ lebih kecil daripada $HMCR$ sehingga variabel keputusan baru dipilih secara random dengan persamaan 2.3

Hasil variabel keputusan baru =

$R2 = 0,14$ dibandingkan dengan $PAR = 0,4$

Maka $R2$ lebih kecil daripada PAR sehingga variabel keputusan baru $x'_2 = 0,80$

c. Membangkitkan vektor solusi variabel ketiga (x'_3)

$R1 = 0,97$ dibandingkan $HMCR = 0,7$

$R1$ lebih besar daripada $HMCR$ sehingga variabel keputusan baru dipilih random dan tetap menghasilkan $x'_3 = 0,97$

Proses pembangkitan dilakukan sampai (x'_{12}), kemudian dihasilkan VSB =

[0,38 0,80 0,97 0,96 0,95 0,94 0,93 0,92 0,91 0,90 0,89 0,88]

Kemudian diurutkan sehingga jadwalnya adalah :

(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3),(4,1),(4,2),(4,3),(1,2),(1,1)

Dengan makespan vektor solusi baru sebesar 222 menit

4) Memperbaharui Harmony Memory

Karena *makespan* lebih besar daripada *makespan worst* maka tidak perlu menggantikan Harmony Memory. Apabila *makespan* lebih kecil dari *makespan worst* maka akan dibentuk lagi Harmony Memory. Kemudian VSB akan menggantikan baris didalam HM sehingga terbentuk jadwal baru dengan *makespan worst* yang telah digantikan. Kesimpulannya *makespan best* dan *makespan worst* tetap.

5) Mengecek kriteria pemberhentian

Kriterianya adalah jumlah iterasi yang telah ditentukan pada tahap inisialisasi parameter. Jika Kriteria telah tercapai maka proses pengerjaan dihentikan. Dalam hal ini dikarenakan perhitungan manual dihitung 1 iterasi maka *makespan* optimalnya adalah dari Vektor Solusi 2 yaitu 141 menit.

b. Perhitungan manual dengan Algoritma *Artificial Bee Colony*

Berikut ini adalah penjadwalan manual *jobshop* menggunakan Algoritma *Artificial Bee Colony* dalam satu iterasi. Langkah-langkah pengerjaannya adalah sebagai berikut:

1) Menentukan Parameter

Seperti halnya Algoritma lain, kita terlebih dahulu menentuka parameter yang akan digunakan dalam Algoritma *Artificial Bee Colony*. Parameter yang digunakan adalah sebagai berikut:

Jumlah lebah pengintai (solusi) = 3

Panjang *list solusi* = 3

Jumlah populasi lebah (iterasi) = 100

2) Inisialisasi Solusi Awal

Setelah parameter ditentukan, selanjutnya membuat jadwal secara *random* kemudian mencari nilai *makespan*. Dengan konsep penjadwalan *jobshop* yang pengerjaannya tidak harus urut serta mengacu pada urutan jadwal pada

Algoritma *Harmony Search* yang maka didapatkan perhitungan seperti yang tertera pada Tabel 4.5

Tabel 4.5 Perhitungan Makespan ABC

	M1		M2		M3	
	S	E	S	E	S	E
J1	0	23	37	68	74	119
J2	0	34	68	94	109	151
J3	0	37	37	65	119	147
J4	34	74	74	109	109	139

Keterangan :

S = Start

E = End

Berdasarkan Tabel 4. Diketahui nilai *makespan* sebesar 151 menit. Kemudian dilakukan metode *Swap Operation (SO)* yaitu melakukan pertukaran kedua *job* yang dibangkitkan secara random. Dalam hal ini akan dilakukan *Swap Operation (SO)* sebanyak 3 kali sesuai dengan parameter dari jumlah solusi yang telah ditentukan. Setelah dilakukan metode *Swap Operation (SO)* maka akan didapatkan hasil sebagai berikut:

- makespan* 141 menit
- makespan* 146 menit
- makespan* 147 menit

Sehingga diperoleh nilai $x_{j \max} = 147$ dan $x_{j \min} = 141$. Kemudian inisialisasi solusi awal menggunakan persamaan (2.) sehingga didapatkan hasil sebagai berikut :

$$x_{ij} = x_{j \min} + \text{rand}(0,1) \cdot (x_{j \max} - x_{j \min})$$

$$x_{11} = 141 + (0,82) (147-141) = 145,92$$

$$x_{12} = 141 + (0,77) (147-141) = 145,62$$

$$x_{13} = 141 + (0,54) (147-141) = 144,24$$

3) Menentukan Solusi Alternatif

Pada tahap ini penginisialisasian yang didapat sebelumnya akan diperluas dengan menggunakan persamaan (2.5).

$$v_{ij} = x_{ij} + \phi_{ij} \cdot (x_{kj} - x_{ij})$$

$$v_{11} = 145,92 + (0,7) (145,62 - 145,92) = 145,710$$

$$v_{12} = 145,62 + (-0,5) (144,24 - 145,62) = 146,310$$

$$v_{13} = 144,24 + (-0,3) (145,92 - 144,24) = 143,736$$

Selanjutnya akan dilakukan *greedy selection* yaitu membandingkan nilai x_{ij} dan nilai v_{ij} . Jika nilai v_{ij} lebih kecil dari nilai x_{ij} maka nilai keduanya dianggap sama sehingga nilai percobaan akan tetap bernilai 0. Apabila nilai v_{ij} tidak lebih kecil dari nilai x_{ij} maka nilai x_{ij} yang disimpan dengan nilai percobaan ke- i akan ditambah 1. Hasil *greedy selection* sebagai berikut:

$$f(x_i) = 145,710 \text{ dengan nilai percobaan } 0$$

$$145,62 \text{ dengan nilai percobaan } 1$$

$$143,736 \text{ dengan nilai percobaan } 0$$

4) Evaluasi Populasi Awal

Pada langkah ini akan dihitung nilai *fitness* pada setiap populasi dengan menggunakan persamaan (2.6).

$$fitness(x_i) = \frac{1}{(1+145,71)} = 0,006862947$$

$$fitness(x_i) = \frac{1}{(1+145,62)} = 0,006820352$$

$$fitness(x_i) = \frac{1}{(1+143,736)} = 0,006909131$$

5) Evaluasi Populasi Alternatif

Pada langkah ini akan dihitung nilai *probability* untuk kemungkinan semua solusi dengan menggunakan persamaan (2.7).

$$\sum_{i=1}^{SN} fitness_i = 0,006862947 + 0,006820352 + 0,006909131 = 0,02059243$$

$$p_1 = \frac{0,006862947}{0,02059243} = 0,33327524$$

$$p_2 = \frac{0,006820352}{0,02059243} = 0,33120676$$

$$p_3 = \frac{0,006909131}{0,02059243} = 0,335518$$

6) Evaluasi Populasi Akhir

Dari nilai *probability* selanjutnya akan dibangkitkan nilai *random* antara 0 sampai 1. Jika nilai p_i lebih besar daripada bilangan *random* yang ditentukan, maka kemungkinan solusi akan diperluas kembali menggunakan persamaan (2.5). Misalkan dibangkitkan nilai *random* 0,3315 maka kemungkinan solusi p_1 dan p_3 diperluas.

4.1.3 Makespan dan Kekonvergenan Algoritma

Untuk mengetahui *makespan* dan kekonvergenan dari masing-masing algoritma dalam 10 kali percobaan menggunakan MATLAB dapat dilihat pada Tabel 4.6.

Tabel 4.6. Hasil *Makespan* dan Pengujian Konvergensi

Pengujian ke-	Makespan		Iterasi Konvergen	
	Algoritma	Algoritma	Algoritma	Algoritma
	<i>HS</i>	<i>ABC</i>	<i>HS</i>	<i>ABC</i>
1	247	267	52	59
2	267	267	259	157
3	264	284	269	191
4	267	264	304	141
5	267	264	382	320
6	247	267	942	429
7	247	264	479	334
8	247	264	343	759
9	249	267	611	382
10	267	262	227	144

Dari Tabel 4.6 Diatas dapat dilihat bahwa *makespan* dari sepuluh kali percobaan menunjukkan waktu paling minimum sebanyak 247 menit pada algoritma *Harmony Search* dan 262 menit pada algoritma *Artificial Bee Colony*. Dari percobaan ke-2 didapatkan *makespan* minimum yang sama besar yaitu 267 menit namun algoritma *Artificial Bee Colony* lebih cepat konvergen pada iterasi ke-157 dibandingkan algoritma *Harmony Search* konvergen pada iterasi ke-259.

4.2 Pembahasan

Dalam hal ini akan dibahas penerapan penjadwalan *jobshop* menggunakan Algoritma *Harmony Search* dan Algoritma *Artificial Bee Colony* pada industri pabrik springbed PT. Cahaya Kawi Ultra Polyintraco yang akan menghasilkan *makespan* paling optimal serta tingkat konvergensi kedua algoritma. Penerapan perhitungan akan dilakukan pada data waktu proses pembuatan 4 tipe springbed kedalam 9 mesin yang telah direduksi sesuai fungsinya menjadi 5 mesin. Data tersebut diproses dengan menggunakan program MATLAB dengan sepuluh kali percobaan. Masing-masing percobaan dilakukan 10 kali *running* dengan nilai-nilai parameter yang berbeda. Tujuannya adalah untuk mencari hasil yang paling optimal. Hasil percobaannya adalah sebagai berikut.

a. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,7
PAR	= 0,1
Bandwidth	= 0,001
Jumlah iterasi	= 500

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi) = 5

Panjang *list solusi* = 3

Jumlah populasi lebah (iterasi) = 500

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.7

Tabel 4.7 Hasil Percobaan Pertama

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	302	475	287	12
2	287	463	292	111
3	267	277	298	18
4	304	51	284	46
5	302	1	267	60
6	297	390	297	252
7	310	238	267	59
8	259	353	297	129
9	313	148	315	301
10	247	52	301	477

Berdasarkan Tabel 4.7 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 247 menit dan konvergen pada iterasi ke-52. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-59.

b. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,7
PAR	= 0,2
Bandwidth	= 0,002
Jumlah iterasi	= 500

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi)	= 5
Panjang <i>list solusi</i>	= 4
Jumlah populasi lebah (iterasi)	= 500

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.8

Tabel 4.8 Hasil Percobaan Kedua

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	299	251	267	157
2	323	138	293	183
3	290	116	302	99
4	330	368	307	19
5	302	327	307	50
6	292	451	300	286
7	299	60	305	430
8	305	96	291	358
9	284	68	284	32
10	267	259	303	175

Berdasarkan Tabel 4.8 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-259. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal juga dengan nilai *makespan* 267 menit namun konvergen pada iterasi ke-157.

c. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,7
PAR	= 0,3
Bandwidth	= 0,003
Jumlah iterasi	= 500

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi)	= 5
Panjang <i>list solusi</i>	= 5
Jumlah populasi lebah (iterasi)	= 500

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.9

Tabel 4.9 Hasil Percobaan Ketiga

Running ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	279	297	293	140
2	264	269	286	233
3	288	283	307	128
4	292	300	307	40
5	303	373	310	382
6	267	494	299	138
7	303	296	292	77
8	316	1	284	191
9	284	381	290	143
10	308	241	300	182

Berdasarkan Tabel 4.9 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 264 menit dan konvergen pada iterasi ke-269. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 284 menit dan konvergen pada iterasi ke-191.

d. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job = 4
 Jumlah Mesin = 5
 HMS = 4
 HMCR = 0,74
 PAR = 0,4
 Bandwidth = 0,005
 Jumlah iterasi = 500

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi) = 5

Panjang *list solusi* = 6

Jumlah populasi lebah (iterasi) = 500

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.10

Tabel 4.10 Hasil Percobaan Keempat

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	332	1	305	54
2	304	498	310	449
3	267	384	305	331
4	277	421	267	93
5	282	164	299	311
6	291	409	264	141
7	282	473	297	58
8	292	151	308	466
9	267	304	284	17
10	306	354	298	381

Berdasarkan Tabel 4.10 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-304. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* yang sama yaitu 264 menit dan namun konvergen pada iterasi ke-141.

e. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,78
PAR	= 0,4
Bandwidth	= 0,006
Jumlah iterasi	= 500

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi)	= 5
Panjang <i>list solusi</i>	= 7
Jumlah populasi lebah (iterasi)	= 500

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.11

Tabel 4.11 Hasil Percobaan Kelima

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	279	441	297	333
2	267	382	279	114
3	267	408	300	179
4	279	414	310	450
5	284	321	294	108
6	290	181	267	380
7	279	384	300	406
8	282	206	311	283
9	284	310	301	80
10	295	129	264	320

Berdasarkan Tabel 4.11 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-382. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 264 menit dan konvergen pada iterasi ke-320.

f. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job = 4
Jumlah Mesin = 5
HMS = 4
HMCR = 0,82
PAR = 0,4
Bandwidth = 0,007
Jumlah iterasi = 1000

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi) = 10
Panjang *list solusi* = 8
Jumlah populasi lebah (iterasi) = 1000

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.12

Tabel 4.12 Hasil Percobaan Keenam

Running ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	264	660	279	540
2	267	520	267	429
3	267	797	281	174
4	247	942	290	868
5	254	483	300	506
6	269	159	292	614
7	259	258	267	540
8	284	715	283	746
9	269	575	279	756
10	267	813	284	482

Berdasarkan Tabel 4.12 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 247 menit dan konvergen pada iterasi ke-942. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-429.

g. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,86
PAR	= 0,4
Bandwidth	= 0,009
Jumlah iterasi	= 1000

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi) = 15

Panjang *list solusi* = 8

Jumlah populasi lebah (iterasi) = 1000

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.13

Tabel 4.13 Hasil Percobaan Ketujuh

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	267	450	267	846
2	279	213	267	219
3	249	398	291	77
4	284	83	264	464
5	264	404	267	352
6	269	817	267	535
7	259	450	264	334
8	289	323	267	924
9	291	236	288	114
10	247	479	267	316

Berdasarkan Tabel 4.13 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 247 menit dan konvergen pada iterasi ke-479. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 264 menit dan konvergen pada iterasi ke-334.

h. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,9
PAR	= 0,44
Bandwidth	= 0,01
Jumlah iterasi	= 1000

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi)	= 20
Panjang <i>list solusi</i>	= 8
Jumlah populasi lebah (iterasi)	= 1000

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.14

Tabel 4.14 Hasil Percobaan Kedelapan

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	249	76	282	570
2	259	534	267	933
3	256	269	267	875
4	254	531	267	151
5	279	836	279	692
6	264	773	264	759
7	247	343	267	438
8	267	938	282	459
9	267	346	267	221
10	247	550	267	27

Berdasarkan Tabel 4.4 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 247 menit dan konvergen pada iterasi ke-343. Sementara algoritma *Artificial Bee Colony* juga mencapai hasil yang optimal dengan nilai *makespan* yang sama yaitu 264 menit namun lebih cepat konvergen pada iterasi ke-759.

i. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job = 4
Jumlah Mesin = 5
HMS = 4
HMCR = 0,94
PAR = 0,48
Bandwidth = 0,01
Jumlah iterasi = 1000

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi) = 25
Panjang *list solusi* = 8
Jumlah populasi lebah (iterasi) = 1000

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.15

Tabel 4.15. Hasil Percobaan Kesembilan

Running ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	256	510	267	773
2	282	587	267	434
3	259	915	267	927
4	284	206	287	37
5	284	406	279	734
6	287	31	279	824
7	249	611	267	461
8	278	1	267	382
9	283	44	279	284
10	264	963	267	469

Berdasarkan Tabel 4.15 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 249 menit dan konvergen pada iterasi ke-611. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-382.

j. Ditentukan parameter percobaan untuk Algoritma *Harmony Search* yaitu :

Jumlah Job	= 4
Jumlah Mesin	= 5
HMS	= 4
HMCR	= 0,99
PAR	= 0,5
Bandwidth	= 0,01
Jumlah iterasi	= 1000

Ditentukan parameter percobaan untuk Algoritma *Artificial Bee Colony* yaitu :

Jumlah lebah pengintai (solusi) = 30

Panjang *list solusi* = 8

Jumlah populasi lebah (iterasi) = 1000

Dari parameter yang telah ditentukan akan diproses melalui 10 kali *running* yang hasilnya dapat dilihat pada Tabel 4.16

Tabel 4.16 Hasil Percobaan Kesepuluh

<i>Running</i> ke-	Algoritma <i>HS</i>		Algoritma <i>ABC</i>	
	<i>Makespan</i>	Konvergen iterasi ke-	<i>Makespan</i>	Konvergen iterasi ke-
1	279	158	267	106
2	277	1	262	144
3	282	779	264	286
4	291	121	264	266
5	267	523	264	507
6	267	227	279	679
7	267	326	267	278
8	318	214	264	109
9	298	240	262	910
10	299	236	264	825

Berdasarkan Tabel 4.16 Dapat dilihat bahwa algoritma *Harmony Search* mencapai hasil yang optimal dengan nilai *makespan* 267 menit dan konvergen pada iterasi ke-227. Sementara algoritma *Artificial Bee Colony* mencapai hasil yang optimal dengan nilai *makespan* 262 menit dan konvergen pada iterasi ke-144.

Dari sepuluh percobaan didapatkan urutan *makespan*, iterasi kekonvergenan dan urutan jadwal dari algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* seperti yang ditunjukkan dalam Tabel 4.17 dan 4.18.

Tabel 4.17 Sepuluh Percobaan dengan Algoritma *Harmony Search*

Percobaan ke-	Urutan jadwal	Makespan (menit)	Iterasi Konvergen
1	(7-1-19-13-17-9-3-18-11-6-2-16-8-14-12-4-15-20-5-10)	247	52
2	(7-3-11-19-1-9-17-13-2-18-6-8-16-14-4-12-10-5-20-15)	267	259
3	(9-11-2-12-18-16-8-7-4-1-19-6-13-3-14-17-15-5-20-10)	264	269
4	(17-4-13-6-11-3-19-7-8-14-16-2-9-18-1-12-10-15-20-5)	267	304
5	(7-18-14-1-4-17-11-8-9-3-16-12-2-19-6-13-10-20-15-5)	267	382
6	(16-3-9-12-14-17-8-1-7-11-4-18-13-19-6-2-5-15-20-10)	247	942
7	(6-2-18-14-1-19-7-17-13-11-9-12-3-16-8-4-10-20-5-15)	247	479
8	(6-12-18-4-14-7-3-16-1-9-17-19-13-11-8-2-10-15-5-20)	247	343
9	(7-16-13-4-11-17-9-6-14-3-2-12-18-19-1-8-5-20-10-15)	249	611
10	(8-1-19-12-13-9-2-16-4-11-18-3-14-7-6-17-15-20-10-5)	267	227

Tabel 4.18 Sepuluh Percobaan dengan Algoritma *Artificial Bee Colony*

Percobaan ke-	Urutan jadwal	Makespan (menit)	Iterasi Konvergen
1	(13-17-4-6-2-11-18-9-1-8-3-19-16-12-14-7-5-10-15-20)	267	59
2	(18-14-1-7-11-8-4-17-16-9-2-13-19-6-3-12-20-5-15-10)	267	157
3	(18-6-4-9-12-3-16-17-8-1-2-14-11-19-13-7-15-20-5-10)	284	191
4	(16-8-2-14-3-19-7-11-18-1-12-17-13-9-4-6-15-20-5-10)	264	141
5	(6-17-3-1-14-12-19-8-7-4-2-16-13-9-18-11-10-15-5-20)	264	320
6	(19-11-3-7-8-2-16-14-4-6-9-13-17-12-18-1-20-5-15-10)	267	429
7	(9-11-3-17-4-12-16-8-6-13-2-19-1-18-14-7-15-20-10-5)	264	334
8	(16-12-3-17-9-1-2-14-8-13-19-18-6-11-7-4-5-20-15-10)	264	759
9	(16-7-3-14-6-17-4-13-12-8-19-18-9-1-11-2-20-5-10-15)	267	382
10	(3-6-14-17-12-1-9-18-4-7-16-13-11-8-19-2-5-15-20-10)	262	144

Keterangan :

1 = <i>job</i> 1 masuk pada mesin ke-1	11 = <i>job</i> 3 masuk pada mesin ke-1
2 = <i>job</i> 1 masuk pada mesin ke-2	12 = <i>job</i> 3 masuk pada mesin ke-2
3 = <i>job</i> 1 masuk pada mesin ke-3	13 = <i>job</i> 3 masuk pada mesin ke-3
4 = <i>job</i> 1 masuk pada mesin ke-4	14 = <i>job</i> 3 masuk pada mesin ke-4
5 = <i>job</i> 1 masuk pada mesin ke-5	15 = <i>job</i> 3 masuk pada mesin ke-5
6 = <i>job</i> 2 masuk pada mesin ke-1	16 = <i>job</i> 4 masuk pada mesin ke-1
7 = <i>job</i> 2 masuk pada mesin ke-2	17 = <i>job</i> 4 masuk pada mesin ke-2
8 = <i>job</i> 2 masuk pada mesin ke-3	18 = <i>job</i> 4 masuk pada mesin ke-3
9 = <i>job</i> 2 masuk pada mesin ke-4	19 = <i>job</i> 4 masuk pada mesin ke-4
10 = <i>job</i> 2 masuk pada mesin ke-5	20 = <i>job</i> 4 masuk pada mesin ke-5

Bersadarkan hasil percobaan pada algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* terdapat perbedaan hasil *makespan* maupun tingkat kekonvergenannya dilihat dari urutan penjadwalannya. Hal ini diakibatkan oleh penentuan parameter yang bervariasi serta bilangan *random* yang ditentukan untuk mendapatkan penjadwalan yang paling optimal dari 10 percobaan tersebut.

Dari percobaan diatas didapat jadwal yang paling optimal dari kedua algoritma. Pada algoritma *Harmony Search* didapatkan *makespan* paling optimal yaitu pada percobaan pertama sebesar 247 menit sementara algoritma *Artificial Bee Colony* didapatkan pada percobaan kesepuluh adalah sebesar 262 menit. Untuk penjadwalan yang didapat pada algoritma *Harmony Search* adalah (7-1-19-13-17-9-3-18-11-6-2-16-8-14-12-4-15-20-5-10) dan pada algoritma *Artificial Bee Colony* adalah (3-6-14-17-12-1-9-18-4-7-16-13-11-8-19-2-5-15-20-10). Sehingga kedua algoritma tersebut memiliki efektifitas yang hampir sama apabila diterapkan ke pada penjadwalan di industri pabrik *springbed* PT.Cahaya Kawi Ultra Polyintraco dikarenakan pada beberapa percobaan terdapat beberapa perbedaan *makespan* yang tidak terlalu signifikan. Namun dari hasil perhitungan menggunakan 10 percobaan dengan parameter yang berbeda, algoritma *Harmony Search* cenderung lebih optimal

dibandingkan algoritma *Artificial Bee Colony* ditinjau dari *makespan*. Hal ini dibuktikan dengan 6 dari 10 percobaan algoritma *Harmony Search* yang lebih baik dibandingkan algoritma *Artificial Bee Colony*. Namun dilihat dari tingkat kekonvergenan, algoritma *Artificial Bee Colony* lebih cepat konvergen daripada algoritma *Harmony Search*. Hal ini dibuktikan dengan 8 dari 10 percobaan algoritma *Artificial Bee Colony* yang lebih cepat konvergen daripada algoritma *Harmony Search*. Sementara pada saat percobaan kedua dihasilkan *makespan* yang sama besar yaitu 267 menit dimana algoritma *Artificial Bee Colony* konvergen pada iterasi ke-157 sedangkan algoritma *Harmony Search* konvergen pada iterasi ke-259.

BAB 5. PENUTUP

5.1 Kesimpulan

Dari penelitian yang dilakukan dapat diambil kesimpulan sebagai berikut:

- a. Penjadwalan *jobshop* ini diterapkan pada industri pabrik *springbed* PT.Cahaya Kawi Ultra Polyintraco dengan memproduksi 4 tipe *springbed* yaitu platinum, silver, golden dan bigline. Dalam 10 kali percobaan yang masing-masing melalui 10 kali *running* program didapatkan penjadwalan algoritma *Harmony Search* yaitu (7-1-19-13-17-9-3-18-11-6-2-16-8-14-12-4-15-20-5-10) dengan *makespan* terbaik sebesar 247 menit. Sedangkan pada penjadwalan algoritma *Artificial Bee Colony* didapatkan penjadwalan (3-6-14-17-12-1-9-18-4-7-16-13-11-8-19-2-5-15-20-10) dengan *makespan* terbaik sebesar 262 menit.
- b. Ditinjau dari *makespan*, algoritma *Harmony Search* lebih optimal dibandingkan algoritma *Artificial Bee Colony* dimana 6 dari 10 percobaan algoritma *Harmony Search* menghasilkan *makespan* yang lebih baik daripada algoritma *Artificial Bee Colony*. Sedangkan ditinjau dari tingkat kekonvergenan, algoritma *Artificial Bee Colony* lebih baik daripada algoritma *Harmony Search* dimana 8 dari 10 percobaan algoritma *Artificial Bee Colony* lebih cepat konvergen daripada algoritma *Harmony Search*.

5.2 Saran

Algoritma *Harmony Search* dan algoritma *Artificial Bee Colony* merupakan algoritma yang dapat digunakan untuk menyelesaikan masalah optimasi sehingga masih terbuka untuk peneliti lain apabila ingin mengembangkan algoritma tersebut. Masih banyak algoritma lain yang juga bisa digunakan sebagai pembanding kedua algoritma tersebut serta banyak kasus lain tentang optimasi yang dapat diselesaikan dengan kedua algoritma tersebut.

DAFTAR PUSTAKA

- Chong, C.S., Low, M.Y.H., Sivakumar, A.I, Gay, K.L. 2006. *A Bee Colony Optimization Algorithm to Job Shop Scheduling*. Singapore: Nayang Technological University
- Geem, Z.W., Kim, J., Loganathan, 2001. *A New Heuristic Optimization Algorithm: Harmony Search*. Simulation. 76: 60–68.
- Ginting, R. 2009. *Penjadwalan Mesin*. Yogyakarta: Graha Ilmu.
- Gunarso, T 2014. “Aplikasi algoritma *Codec* untuk penyelesaian masalah *jobshop scheduling* pada produksi *springbed*”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Hapsari, V.D. 2014. “Penggunaan algoritma *Harmony Search* dan algoritma Genetika pada penjadwalan produksi Dandang pada industri rumah tangga UD.Alif Jember”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Indriana. 2011. “Penggunaan algoritma *Harmony Search*, algoritma *Differential Evolution*, dan algoritma *Swam Optimization* dalam menyelesaikan penjadwalan pada perusahaan penerbangan”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Mattfeld, Dirk C. & Vaessens, Rob J. M. 2004. *OR-Library : FSP Test Instances*. [serial online] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/flow_shop1.txt [7 November 2014]
- Munir, R. 2005. *Matematika Diskrit*. Revisi 5. Bandung: Informatika.
- Sugioko, A. 2012. “Penggunaan Modifikasi Algoritma *Bee Colony* dengan *Tabu List* pada penjadwalan *jobshop* dengan kriteria biaya keterlambatan”. Tidak Diterbitkan. Skripsi. Jember: Universitas Jember.
- Shofwatul’uyun. 2009. *Kompleksitas Algoritma*. Teknik Informatika UIN Kalijaga. [serial online] <http://www.scribd.com/doc/38673595/null>. [9 April 2014].
- Suyanto. 2010. *Algoritma Optimasi*. Yogyakarta: Graha Ilmu.

LAMPIRAN

LAMPIRAN A. Data Waktu Proses Mesin

Nomor	Nama	Nomor Mesin				
Job	Job	M1	M2	M3	M4	M5
1	Platinum	91	48	66	8	20
2	Golden	91	48	66	8	18
3	Silver	86	45	63	8	8
4	Bigline	85	44	61	8	8

LAMPIRAN B. Data Reduksi 9 Mesin ke 5 Mesin Notasi Baru

No	Komponen	Notasi Lama	Tugas Mesin	Notasi Baru	Tugas Mesin
1	Matras	m1	Perakitan Pegas	M1	Pembuatan Matras Springbed
		m2	Perakitan Kawat		
		m7	Pembuatan Rangka Kayu		
		m5	Pembuatan Hardpad		
		m3	Pemotongan,Pemasangan Busa		
		m8	Finishing Komponen rangka		
		m6	Penjahitan Kain		
2	Sandaran	m7	Pembuatan Rangka Kayu	M2	Pembuatan Sandaran Springbed
		m5	Pembuatan Hardpad		
		m3	Pemotongan,Pemasangan Busa		
		m8	Finishing Komponen rangka		
		m6	Penjahitan Kain		
3	Divan	m7	Pembuatan Rangka Kayu	M3	Pembuatan Divan Springbed
		m5	Pembuatan Hardpad		
		m3	Pemotongan,Pemasangan Busa		
		m5	Pembuatan Hardpad		
		m6	Penjahitan Kain		
4	Kaki	m4	Pembuatan Kaki Springbed	M4	Pembuatan Kaki Springbed
5	Perakitan	m9	Perakitan Springbed	M5	Perakitan Springbed

LAMPIRAN C. Data Waktu Proses Mesin pada 4 Jenis *Springbed*

No	Komponen Springbed	Notasi Lama	Waktu (job)				Notasi Baru	Waktu (job)			
			1	2	3	4		1	2	3	4
			1	Matras	m1	14		14	13	13	M1
		m2	15	15	15	14					
		m7	18	18	16	16					
		m5	3	3	2	2					
		m3	6	6	5	5					
		m8	15	15	15	15					
		m6	20	20	20	20					
2	Sandaran	m7	10	10	10	10	M2	48	48	45	44
		m5	3	3	2	2					
		m3	8	8	8	7					
		m8	12	12	11	11					
		m6	15	15	14	14					
3	Divan	m7	16	16	15	15	M3	66	66	63	61
		m5	10	10	9	9					
		m3	8	8	8	7					
		m5	15	15	14	14					
		m6	17	17	17	16					
4	Kaki	m4	8	8	8	8	M4	8	8	8	8
5	Perakitan	m9	20	18	14	12	M5	20	18	14	12