



UNIVERSITAS INDONESIA

**OPTIMASI PENJADWALAN PRODUKSI KEMASAN KAYU
DENGAN SISTEM *JOB SHOP* MELALUI PENERAPAN
ALGORITMA *DIFFERENTIAL EVOLUTION***

SKRIPSI

**RINI KURNIAPUTRI
0706275012**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2011**



UNIVERSITAS INDONESIA

**OPTIMASI PENJADWALAN PRODUKSI KEMASAN KAYU
DENGAN SISTEM *JOB SHOP* MELALUI PENERAPAN
*ALGORITMA DIFFERENTIAL EVOLUTION***

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**RINI KURNIAPUTRI
0706275012**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INDUSTRI
DEPOK
JUNI 2011**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar

Nama : Rini Kurniaputri
NPM : 0706275012
Tanda Tangan : 
Tanggal : Juni 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Rini Kurniaputri
NPM : 0706275012
Program Studi : Teknik Industri
Judul Skripsi : Optimasi Penjadwalan Produksi Kemasan Kayu dengan Sistem *Job Shop* Melalui Penerapan Algoritma *Differential Evolution*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian dari persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Industri Fakultas Teknik Universitas Indonesia

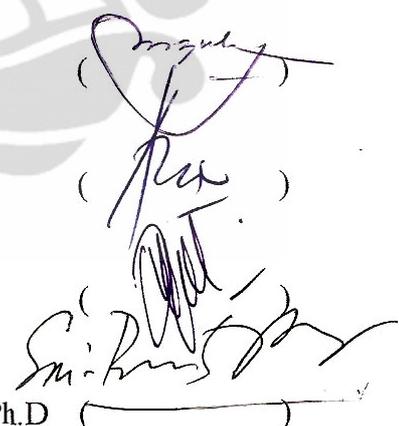
DEWAN PENGUJI

Pembimbing : Ir. Amar Rachman, MEIM

Penguji : Arian Dhini, ST., MT

Penguji : Ir. Isti Surjandari, MT., MA., Ph.D

Penguji : Ir. Sri Bintang Pamungkas, MSc., Ph.D



Handwritten signatures of the examiners and supervisor, including the name 'Sri Bintang Pamungkas'.

Ditetapkan di : Depok

Tanggal : Juni 2011

Universitas Indonesia

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas semua rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Penyusunan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Departemen Teknik Industri Fakultas Teknik Universitas Indonesia. Dapat di sadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sulit bagi penulis untuk dapat menyelesaikan skripsi ini. Untuk itu akan diucapkan terima kasih kepada :

1. Ir. Amar Rachman, MEIM selaku dosen pembimbing yang telah menyediakan waktu, tenaga, pikiran serta dorongan dan bimbingan untuk mengarahkan di dalam penyusunan skripsi ini.
2. Armand Omar Moeis, ST, MSc.; Ir. Akhmad Hidayatno, MBT.; Arian Dhini, ST. MT.; Hj. Erlinda Muslim, Ir., MEE.; Ir. Yadrifil, MSc.; Ir. Sri Bintang Pamungkas, MSc., Ph.D.; Ir. Isti Surjandari, MT., MA., Ph.D yang telah memberikan banyak masukan, perbaikan, motivasi yang luar biasa agar dapat lebih baik lagi dan lebih bersemangat di dalam menyelesaikan skripsi ini.
3. Prof. Dr. Ir. Teuku Yuri Zagloel, MengSc. selaku kepala Departemen Teknik Industri Universitas Indonesia yang telah memberikan bimbingan selama kuliah di Teknik Industri UI.
4. Ir. Rahmat Nurcahyo, MengSc. selaku pembimbing akademis serta seluruh Dosen Teknik Industri UI yang telah mendidik dan membimbing saya selama kuliah di Teknik Industri UI
5. Bapak dan Ibu Boedi Santoso yang telah menyediakan kesempatan untuk melakukan penelitian di pabrik dan membantu memfasilitasi segala hal yang saya butuhkan dalam penyusunan skripsi ini.
6. Kak Lina Astuti dan Pak Kikik Yaranusa atas bantuan yang telah diberikan dalam membuat program.
7. Bapak dan Mama yang selama ini telah memberikan semangat, dukungan, serta doa tanpa henti. Terima kasih sudah menemani saya mengerjakan skripsi sampai larut malam dan dengan sabar mendengarkan segala keluhan dan tidak kalah penting atas dukungan finansial yang tidak sedikit

Universitas Indonesia

untuk transportasi ke pabrik, ke kampus, biaya *print*, dan segala macam keperluan skripsi.

8. Pak Ujang yang selalu setia mengantar keliling pabrik, memberikan penjelasan yang mendalam, serta menyediakan segala kebutuhan yang diperlukan untuk penelitian skripsi.
9. Anisha Puti Lalita, Citra Atma Pertiwi, Ayuning Pramesthi Pintoarsi, Khairiyah, Sekar Melati, Aulya Nuraini, Astriana Gita, Paramitha Mansoer, Gina Adryani, Sarah Noviani Rodjali, Indi Puspita, Annisa Zahara, yang telah menjadi bagian penting yang tak terpisahkan dari kisah hidup selama kuliah 4 tahun di Teknik Industri UI.
10. Remy Hamza Subroto yang senantiasa memberikan hiburan, masukan, serta kesediaannya mendengarkan keluh kesah selama mengerjakan skripsi. Terima kasih sudah meminjamkan buku panduan MATLAB.
11. Seluruh teman-teman Teknik Industri UI angkatan 2007 yang telah mewarnai hari-hari saya selama kuliah di Teknik Industri UI, meninggalkan cerita-cerita tak terlupakan, serta berbagi susah dan senang bersama.
12. Teman-teman UGD SMAN 28 yang sampai saat ini masih menjadi bagian penting dari perjalanan hidup.
13. Seluruh pihak yang telah membantu dari awal sampai akhir yang tidak dapat disebutkan satu per satu.

Akhir kata, diharapkan Allah SWT berkenan membalas segala kebaikan bagi seluruh pihak yang telah banyak membantu di dalam penyusunan skripsi ini. semoga skripsi ini nantinya dapat bermanfaat bagi semuanya. Amin.

Depok, 21 Juni 2011

Penulis

Universitas Indonesia

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama : Rini Kurniaputri
NPM/NIP : 0706275012
Program Studi : Teknik Industri
Fakultas : Teknik
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Non-Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

**Optimasi Penjadwalan Produksi Kemasan Kayu dengan Sistem *Job Shop*
Melalui Penerapan Algoritma *Differential Evolution***

Beserta perangkat yang ada (bila diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan (*database*), merawat dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juni 2011

Yang menyatakan



(Rini Kurniaputri)

Universitas Indonesia

ABSTRAK

Nama : Rini Kurniaputri
Program Studi : Teknik Industri
Judul : Optimasi Penjadwalan Produksi Kemasan Kayu dengan Sistem *Job Shop* Melalui Penerapan Algoritma *Differential Evolution*

Besarnya kebutuhan akan kemasan kayu dalam kegiatan perdagangan menuntut produsen kemasan kayu untuk mengembangkan sistem penjadwalan produksi yang optimal agar dapat memenuhi permintaan pasar. Penelitian ini membahas optimasi penjadwalan produksi kemasan kayu, khususnya pallet, dengan sistem *job shop* melalui penerapan algoritma *Differential Evolution*. Prinsip algoritma DE sesuai dengan analogi evolusi biologi yang terdiri dari proses inisialisasi populasi, proses mutasi, proses pindah silang, dan proses seleksi. Tujuan dari penelitian ini adalah untuk meminimumkan total biaya (dalam satuan waktu) yang timbul sebagai akibat dari keterlambatan dalam proses produksi kemasan kayu, khususnya pallet yang dipergunakan untuk mengemas barang-barang ekspor. Penelitian dilakukan melalui studi kasus dengan mengamati proses produksi pallet pada suatu produsen pallet.

Hasil dari penelitian ini adalah diperolehnya usulan penjadwalan produksi dengan penurunan total biaya keterlambatan sebesar 25,22%, (dari 23590 menit menjadi 17640 menit) serta penurunan pada kriteria lainnya seperti jumlah pesanan yang terlambat, total waktu keterlambatan, dan waktu penyelesaian. Simulasi penambahan kapasitas menunjukkan bahwa penambahan jumlah mesin pada stasiun serut dan potong dapat menghilangkan keterlambatan dan meningkatkan *output* hingga 95%.

Kata kunci :

Job Shop Scheduling, Differential Evolution, optimasi penjadwalan produksi, kemasan kayu

ABSTRACT

Name : Rini Kurniaputri
Study Program : Industrial Engineering
Title : Optimizing *Job Shop* Scheduling System of Wooden Packaging Production through the Application of Differential Evolution Algorithm

The ever increasing need of wooden packaging , whether for international as well as inter-islands transportation of traded goods, has required the producers of wooden packaging to develop an optimized production scheduling system to fulfill the market demand. This research studies the optimization of the *job shop* production scheduling system of wooden packaging, particularly pallet, through the application of the Differential Evolution (DE) algorithm. The principles of the DE algorithm is in line with the biological evolution analogy which consists of the initialization of population, mutation, *crossover*, and selection processes. The DE algorithm has some excellent features namely the concept is simple, easy to apply, quick in producing solutions, and robust. The objektif of this research is to minimize the total cost (in time unit) resulting from the delay in the production process of wooden packaging, particularly pallet. This research is a case-study which is carried-out by observing pallet production process at a wooden packaging manufacturer. The company produces wood packaging, mainly pallet, which are used for the packaging of exported goods.

This research leads to a recommendation on the wooden packaging production schedule with more efficient cost. The total cost resulting from the delay in the production is reduced by 25,22% (from 23590 minutes to 17640 minutes). There are also reductions in the total of tardy job, total of tardiness, and makespan. A simulation of work station capacity upgrade also shows that the addition of the number of machineries in planing and sawing station will eliminate delay and increase output up to 95%.

Keyword :

Job Shop Scheduling, Differential Evolution, production scheduling optimization, wooden packaging

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vi
ABSTRAK	vii
ABSTRACT.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Diagram Keterkaitan Masalah.....	3
1.3. Rumusan Masalah	4
1.4. Tujuan	4
1.5. Ruang Lingkup Penelitian.....	4
1.6. Metodologi Penelitian	5
BAB II DASAR TEORI.....	7
2.1. Perencanaan Produksi	7
2.1.1. Pengertian Penjadwalan Produksi.....	7
2.1.2. Jenis Penjadwalan Produksi.....	7
2.1.3. Istilah dalam Penjadwalan Produksi	8
2.1.5. Karakteristik dan Kendala Proses Produksi	10
2.2. Penjadwalan <i>Job Shop</i>	11
2.3. Penalti <i>Earliness</i> dan <i>Tardiness</i> dalam Penjadwalan Produksi	12
2.4. Metode Penyelesaian Masalah Penjadwalan Produksi	13
2.5. Algoritma Differential Evolution.....	16
2.5.1. Definisi Algoritma <i>Differential Evolution</i>	16
2.5.2. Tahap-tahap Algoritma Diferensial Evolution.....	17
2.5.3. Penerapan Algoritma DE pada Penjadwalan <i>Job Shop</i>	20
2.5.3.1. Elemen-elemen dasar algoritma DE.....	21
2.6. Model Matematis	24
2.7. Design of Experiments (DOE).....	27
2.7.1. Tujuan Design of Experiments	27
2.7.2. Tipe Percobaan.....	27
2.7.3. Prinsip Dasar	29
BAB III PENGUMPULAN DATA	30
3.2. Jenis Produk	31
3.3. Alur Produksi	33
3.4. Data yang Diperlukan	35
3.4.1. Data Aliran Proses Produksi Kayu.....	36
3.4.2. Data Lama Operasi Tiap <i>Work Center</i>	37
3.4.3. Data Pesanan dan Jam Kerja.....	38
BAB IV PENGOLAHAN DATA DAN ANALISA.....	40
4.1. Penyusunan Algoritma.....	40

4.1.1.	Langkah-Langkah Penyusunan Algoritma.....	41
4.2.	Verifikasi dan Validasi Program.....	44
4.2.1.	Verifikasi Program.....	45
4.2.2.	Validasi Program.....	46
4.2.2.1.	Hasil <i>run</i> program.....	46
4.2.2.2.	Hasil perhitungan manual.....	46
4.3.	<i>Input</i> Data.....	58
4.3.1.	<i>Input</i> Parameter.....	58
4.3.2.	<i>Input</i> Data Pesanan.....	59
4.4.	Pengolahan Data dan Hasil.....	59
4.4.1.	Hasil Penjadwalan Produksi Perusahaan.....	60
4.4.2.	Hasil Penjadwalan Dengan Algoritma DE.....	60
4.5.	Analisa Hasil.....	62
4.5.1.	Analisa Skenario Parameter.....	62
4.5.2.	Analisis Hasil.....	67
4.5.3.	Peningkatan Kapasitas yang Diperlukan.....	68
BAB V KESIMPULAN DAN SARAN.....		71
5.1.	Kesimpulan.....	71
5.2.	Saran.....	71
DAFTAR PUSTAKA.....		73

DAFTAR GAMBAR

Gambar 1.1. Diagram Keterkaitan Masalah.....	3
Gambar 1.2. Diagram Alir Metodologi Penelitian.....	6
Gambar 2.1. Contoh Rute Penjadwalan <i>Job Shop</i>	12
Gambar 2.2. Diagram Alir Pengerjaan Algoritma DE.....	17
Gambar 2.3. Diagram Alir Algoritma DE untuk <i>Job Shop</i> <i>Sequencing Problem</i>	24
Gambar 3.1. Aliran Produksi Tiap Tipe Pallet.....	36
Gambar 3.2. Aliran Produksi Tiap Komponen Pallet	36
Gambar 4.1. Keterlambatan Setiap <i>Job (Run 3)</i>	61
Gambar 4.2. Urutan Terbaik Hasil <i>Run</i> Program	61
Gambar 4.3. Grafik Interaksi Antar Parameter.....	63
Gambar 4.4. Pengaruh Parameter Terhadap <i>Output</i>	64

DAFTAR TABEL

Tabel 1.1. Ukuran Pallet yang Banyak Diproduksi oleh Perusahaan.....	2
Tabel 2.1. Algoritma <i>Differential Evolution</i>	18
Tabel 2.2. Kriteria Terminasi	22
Tabel 3.1. Standar Internasional Ukuran Pallet.....	31
Tabel 3.2. Tipe Produk yang Dijadikan Objek Penelitian	32
Tabel 3.3. Waktu Operasi di Setiap <i>Work Center</i>	38
Tabel 3.4. Pesanan Pallet Bulan April 2011.....	39
Tabel 3.5. Waktu Kerja	39
Tabel 4.1. Data <i>Dummy</i>	45
Tabel 4.2. Hasil Verifikasi Program.....	46
Tabel 4.3 Populasi Target.....	47
Tabel 4.4. Permutasi Populasi Target.....	47
Tabel 4.5. Perhitungan Waktu Proses Individu 1 Populasi Target.....	48
Tabel 4.6. Biaya Keterlambatan Setiap <i>Job</i> Individu 1 Populasi Target.....	49
Tabel 4.7. Perhitungan Waktu Proses Individu 2 Populasi Target.....	49
Tabel 4.8. Biaya Keterlambatan Setiap <i>Job</i> Individu 2 Populasi Target.....	49
Tabel 4.9. Perhitungan Waktu Proses Individu 3 Populasi Target.....	49
Tabel 4.10. Biaya Keterlambatan Setiap <i>Job</i> Individu 3 Populasi Target.....	50
Tabel 4.11. Perhitungan Waktu Proses Individu 4 Populasi Target.....	50
Tabel 4.12. Biaya Keterlambatan Setiap <i>Job</i> Individu 4 Populasi Target.....	50
Tabel 4.13. Perhitungan Waktu Proses Individu 5 Populasi Target.....	50
Tabel 4.14. Biaya Keterlambatan Individu 5 Populasi Target.....	51
Tabel 4.15. Vektor Target.....	52
Tabel 4.16. Vektor Acak 1.....	52
Tabel 4.17. Vektor Acak 2.....	52
Tabel 4.18. Populasi Mutan.....	52
Tabel 4.19. Populasi Trial.....	53
Tabel 4.20. Permutasi Populasi Trial.....	54
Tabel 4.21. Perhitungan Waktu Proses Individu 1 Populasi Target.....	54
Tabel 4.22. Biaya Keterlambatan Setiap <i>Job</i> Individu 1 Populasi Target.....	54

Tabel 4.23. Perhitungan Waktu Proses Individu 2 Populasi Target.....	55
Tabel 4.24. Biaya Keterlambatan Setiap <i>Job</i> Individu 2 Populasi Target.....	55
Tabel 4.25. Perhitungan Waktu Proses Individu 3 Populasi Target.....	55
Tabel 4.26. Biaya Keterlambatan Setiap <i>Job</i> Individu 3 Populasi Target.....	55
Tabel 4.27. Perhitungan Waktu Proses Individu 4 Populasi Target.....	56
Tabel 4.28. Biaya Keterlambatan Setiap <i>Job</i> Individu 4 Populasi Target.....	56
Tabel 4.29. Perhitungan Waktu Proses Individu 5 Populasi Target.....	56
Tabel 4.30. Biaya Keterlambatan Setiap <i>Job</i> Individu 5 Populasi Target.....	56
Tabel 4.31. Perbandingan Total Biaya Keterlambatan Populasi Target dan Populasi <i>Trial</i>	57
Tabel 4.32. Populasi Target Baru.....	59
Tabel 4.33. Level Tiap Parameter dalam DOE.....	60
Tabel 4.34. Penjadwalan Produksi Perusahaan.....	60
Tabel 4.35. Hasil <i>Run</i> Program	60
Tabel 4.36. ANOVA Parameter Algoritma DE.....	62
Tabel 4.37. Perbandingan Jadwal Awal dan Usulan Jadwal.....	67
Tabel 4.38. Usulan Jadwal Pengerjaan <i>Job</i> untuk Bulan April 2011	68
Tabel 4.39. Hasil Peningkatan Kapasitas Stasiun Kerja	69

DAFTAR LAMPIRAN

LAMPIRAN 1. Uji Normalitas dan Kecukupan Data Primer

LAMPIRAN 2. Jadwal Produksi Pallet Perusahaan

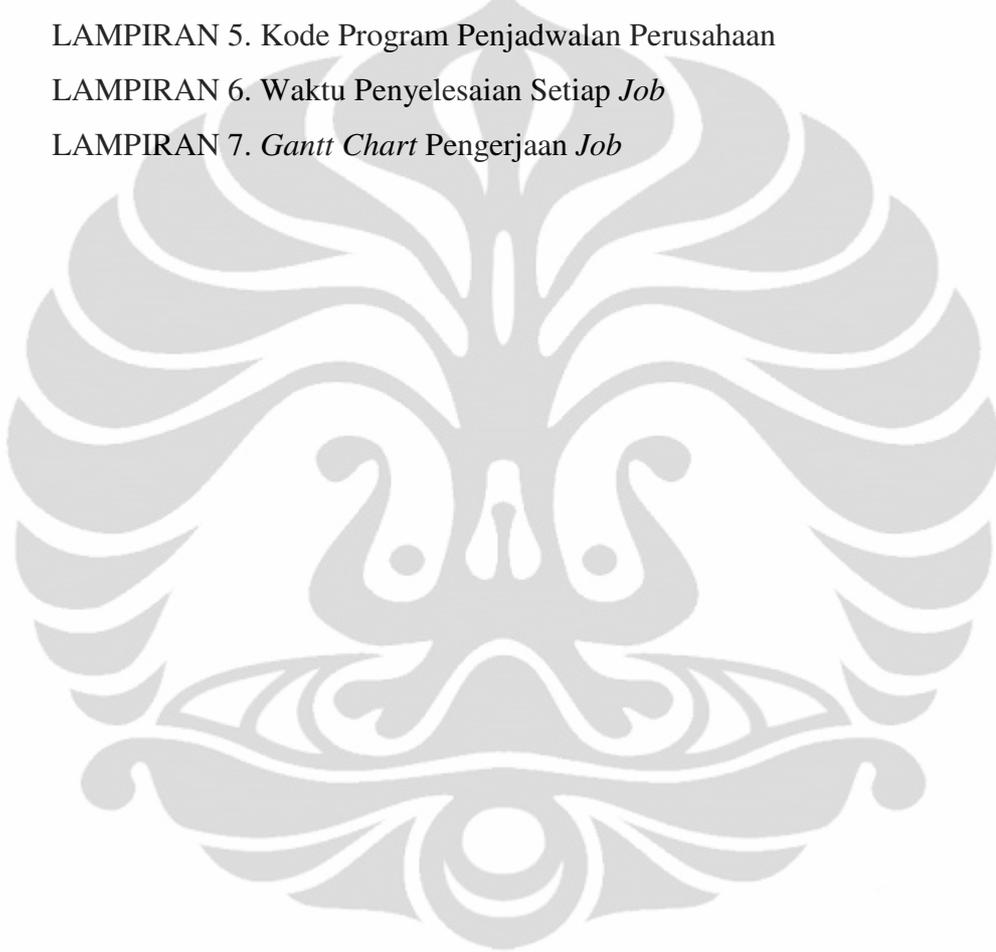
LAMPIRAN 3. Kode Program Solusi Penjadwalan dengan Algoritma DE

LAMPIRAN 4. Hasil *Design of Experiment*

LAMPIRAN 5. Kode Program Penjadwalan Perusahaan

LAMPIRAN 6. Waktu Penyelesaian Setiap *Job*

LAMPIRAN 7. *Gantt Chart* Pengerjaan *Job*



BAB I

PENDAHULUAN

1.1. Latar Belakang

Industri kayu di Indonesia ditujukan untuk memenuhi konsumsi dalam negeri maupun untuk kegiatan ekspor dengan tujuan untuk memperoleh devisa. Salah satu upaya pemerintah dalam mendorong industri kayu di Indonesia adalah dengan menutup kran ekspor kayu bulat agar kebutuhan industri pengolahan kayu dalam negeri dapat terpenuhi. Dengan demikian, industri pengolahan kayu diharapkan dapat memberikan produk yang bernilai tambah (*value added product*) untuk kegiatan ekspor Indonesia (Bank Indonesia, n.d.).

Kayu sebagai material pengepakan, penyangga, pelindung dan pembungkus barang sering digunakan dalam perdagangan internasional, baik ekspor, impor maupun yang dilalulintaskan antar area. Industri pada umumnya menggunakan kemasan yang terbuat dari kayu karena memiliki kelebihan seperti perlindungan mekanis yang baik, karakteristik tumpukan yang baik, dan ratio daya tarik dan berat yang baik (Universitas Sumatera Utara, n.d.). Keadaan tersebut menunjukkan bahwa kehadiran industri kemasan kayu sangatlah penting karena produknya digunakan oleh banyak industri lain dalam aktivitas perdagangannya. Standar kemasan kayu untuk komoditas ekspor pun diatur dalam *International Standard for Phytosanitary Measures* (ISPM#15) yang sudah diterapkan di Indonesia sejak September 2009 (Skim Audit Badan Karantina Pertanian, 2010).

Perusahaan yang menjadi tempat penelitian, sebagai salah satu produsen kemasan kayu di Indonesia, merupakan perusahaan yang telah dinyatakan memenuhi persyaratan Skim Audit Badan Karantina Pertanian untuk melaksanakan perlakuan (*treatment*) dan sertifikasi (*marking*) atas kemasan kayu sesuai dengan ISPM#15 (Skim Audit Badan Karantina Pertanian, 2008). Dengan produk yang sudah terjamin kualitasnya, perusahaan ini dipercaya oleh banyak industri untuk menjadi supplier kemasan kayu mereka. Pelanggan perusahaan beragam, mulai dari industri makanan sampai dengan industri elektronik.

Tantangan yang dihadapi perusahaan ini tidaklah hanya dari segi kualitas kemasan kayu yang diproduksinya, tetapi juga dalam merencanakan kegiatan produksi agar dapat memenuhi kebutuhan pelanggannya karena setiap pelanggan perusahaan memiliki kebutuhan kemasan kayu dengan spesifikasi dan jumlah yang berbeda-beda. Permintaan akan produk dari masing-masing pelanggan pun berbeda-beda untuk setiap periode. Adanya perbedaan ini membuat perusahaan perlu membuat suatu sistem perencanaan produksi agar dapat memenuhi kebutuhan pelanggannya. Dengan mengacu pada perencanaan ini, perusahaan dapat mengetahui berapa serta kapan memproduksi produk agar dapat mencukupi permintaan pasar. Perencanaan produksi yang baik akan memberikan keuntungan bagi pihak perusahaan karena produksi yang optimal dapat meminimalisasi biaya.

Keberadaan industri kemasan kayu lain sebagai kompetitor mendorong perusahaan untuk membuat perencanaan produksi yang seoptimal mungkin agar dapat bersaing tidak hanya dari segi kualitas, tetapi juga keandalan dan ketepatan waktu dalam memenuhi permintaan pasar. Dengan spesifikasi produk yang beragam, penerapan sistem pengendalian material, penjadwalan, serta aliran produksi kemasan kayu harus terkontrol dengan baik (Stevenson, 2008).

Pengembangan kapasitas produksi kemasan kayu mempunyai prospek yang baik untuk dikembangkan karena peluang pasar yang besar mengingat kemasan kayu khususnya pallet yang diproduksi perusahaan memiliki spesifikasi yang berbeda-beda, mulai dari pallet dengan ukuran sesuai kebutuhan pelanggan sampai dengan pallet berukuran standar internasional yang digunakan dalam kegiatan ekspor dan impor. Tabel berikut ini berisi data ukuran pallet yang banyak diproduksi oleh perusahaan.

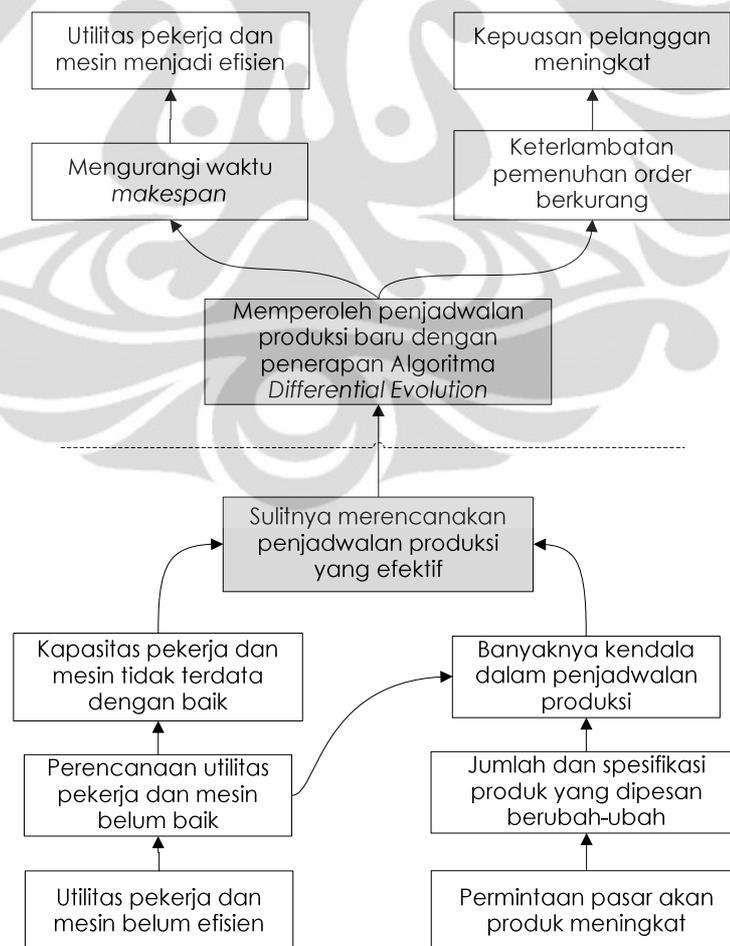
Tabel 1.1. Ukuran Pallet yang Banyak Diproduksi oleh Perusahaan

Jenis Pallet	Tipe
104 x 104 x 13	1
100 x 100 x 10,5	
108 x 108 x 10,5	
114 x 114 x 13,8	2
113,5 x 113,5 x 11	
114 x 114 x 13	
120 x 100 x 14,9	3
120 x 80 x 14,2	
110 x 78 x 10,5	4

Dalam satu bulan, perusahaan bisa memproduksi sampai dengan 10.000 unit kemasan kayu. Banyaknya jenis pallet yang diproduksi menyebabkan sulitnya membuat penjadwalan produksi yang optimal. Oleh karena itu, optimasi penjadwalan produksi kemasan kayu khususnya pallet pada perusahaan akan dilakukan dengan metode *Differential Evolution*. Metode ini sudah banyak dipakai dalam berbagai optimasi penjadwalan proses produksi. Dengan menggunakan metode ini, akan diperoleh suatu usulan penjadwalan produksi yang mendekati optimal. Selain itu dapat diketahui pula kemungkinan peningkatan kapasitas produksi dengan sistem yang ada.

1.2. Diagram Keterkaitan Masalah

Penjabaran latar belakang ditransformasikan dalam diagram keterkaitan masalah berikut ini untuk melihat permasalahan secara sistematis.



Gambar 1.1. Diagram Keterkaitan Masalah

1.3. Rumusan Masalah

Permasalahan yang menjadi fokus penelitian adalah penjadwalan produksi pallet dari suatu perusahaan produsen kemasan kayu yang memiliki konsumen dengan kebutuhan kemasan kayu yang berbeda-beda, baik dari segi jumlah, bentuk, maupun ukuran kemasan kayu. Jumlah permintaan produk dari setiap konsumen juga berfluktuasi dari waktu ke waktu. Selain itu, penelitian juga dilakukan sehubungan dengan adanya rencana peningkatan kapasitas produksi oleh perusahaan untuk memenuhi permintaan pasar.

1.4. Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah memperoleh suatu usulan penjadwalan produksi sistem *job shop* yang optimal melalui penerapan Algoritma *Differential Evolution* untuk meminimalisasi total biaya keterlambatan serta kemungkinan peningkatan kapasitas produksi dengan mengetahui jumlah peralatan yang perlu ditingkatkan.

1.5. Ruang Lingkup Penelitian

Untuk mendapatkan hasil penelitian yang spesifik dan terarah, maka ruang lingkup permasalahan dalam penelitian ini adalah sebagai berikut:

1. Fungsi tujuan yang akan dievaluasi dalam penelitian ini adalah mengoptimalkan penjadwalan produksi.
2. Penjadwalan dianggap bersifat statis dan *non-preemptive*, artinya semua order diterima di awal periode penjadwalan dan penjadwalan dilakukan di awal periode.
3. Data besarnya waktu *setup* mesin tidak diikutsertakan dalam penelitian karena pengaruhnya terhadap keseluruhan waktu produksi sangat kecil.
4. Dalam satu waktu, satu mesin hanya bisa memproses satu *job*.
5. Kondisi mesin produksi dianggap berjalan dengan normal, tidak mengalami kondisi *repair* atau *breakdown* di tengah-tengah waktu proses.
6. Kondisi bahan baku yang dibutuhkan dalam proses produksi dianggap sudah memenuhi syarat dan dalam keadaan *ready stock*.

1.6. Metodologi Penelitian

Berikut langkah-langkah metodologi penelitian dalam penulisan tugas akhir ini, sebagaimana yang tergambar pada diagram alir dari metodologi penelitian (gambar 1.2.):

1. Melakukan identifikasi permasalahan di perusahaan
2. Mengumpulkan dan menyusun studi literatur yang berkaitan

Studi literatur yang dikumpulkan adalah yang berhubungan dengan sistem *job shop* dan metode-metode penjadwalan sistem *job shop* baru yang dapat menghasilkan sistem penjadwalan yang optimal.
3. Melakukan perumusan masalah

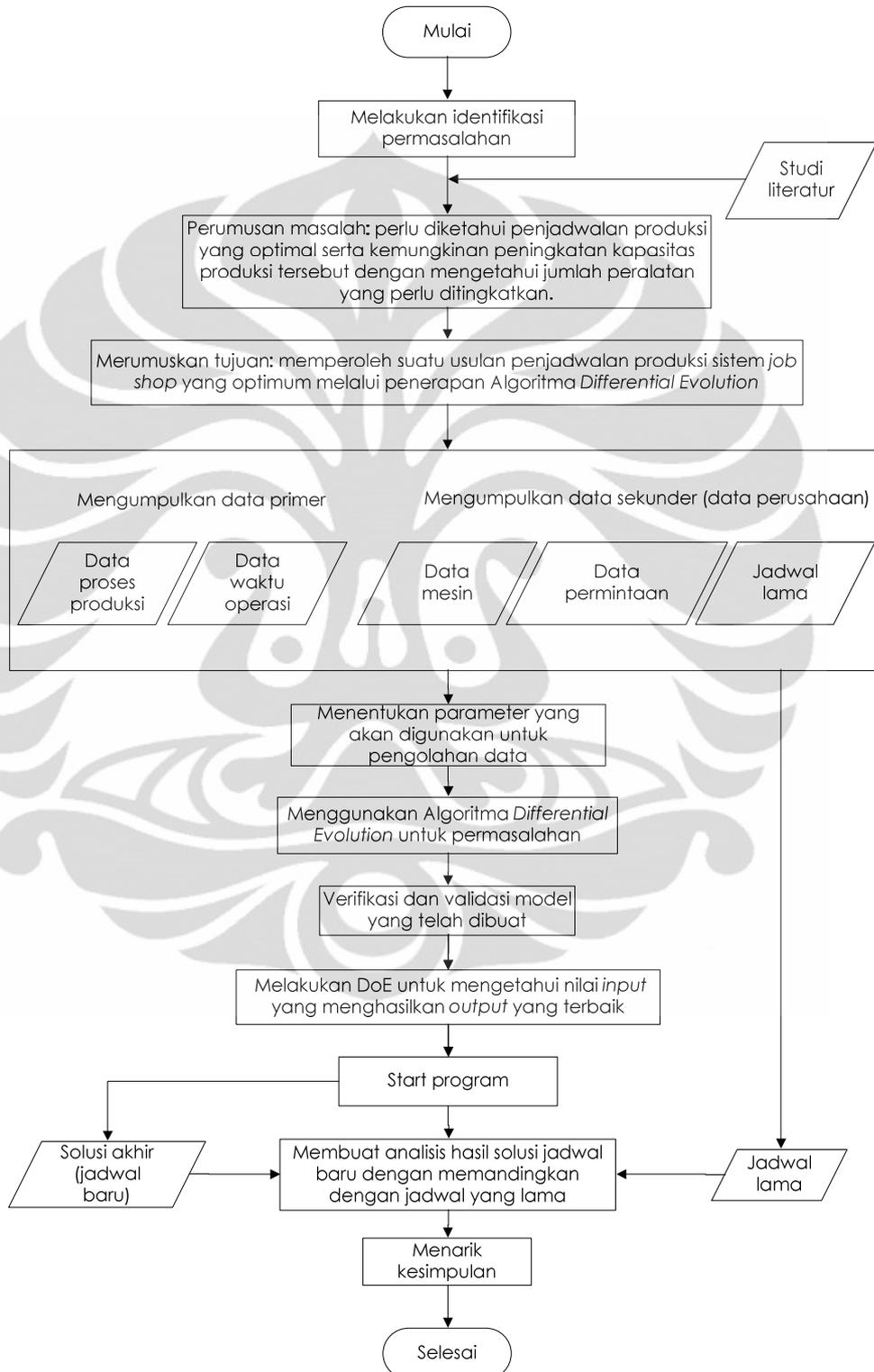
Perumusan permasalahan untuk penelitian ini adalah diperlukannya metode penjadwalan baru dengan menggunakan Algoritma *Differential Evolution*.
4. Menentukan tujuan penelitian

Tujuan dari penelitian ini adalah diperolehnya penjadwalan baru yang lebih optimal.
5. Melakukan identifikasi dan mengumpulkan data

Data yang dibutuhkan berupa data sekunder yang diperoleh dari perusahaan langsung yang terdiri dari data proses produksi, data mesin, data waktu proses, data historis permintaan, dan jadwal lama yang digunakan oleh perusahaan.
6. Membuat Algoritma *Differential Evolution* untuk permasalahan
7. Mengolah data dengan program komputer MATLAB
8. Verifikasi dan validasi model yang telah dibuat
9. Melakukan DOE untuk mengetahui *input* yang dapat menghasilkan *output* yang terbaik
10. Melakukan analisis hasil solusi jadwal yang baru

Membandingkan metode penjadwalan yang lama dengan penjadwalan yang dihasilkan dengan metode Algoritma *Differential Evolution*.
11. Menarik kesimpulan dan mengajukan saran

Mengambil kesimpulan serta memberikan saran mengenai penjadwalan produksi berdasarkan hasil analisis yang telah dilakukan sebelumnya.



Gambar 1.2. Diagram Alir Metodologi Penelitian

BAB II

DASAR TEORI

2.1. Perencanaan Produksi

Proses produksi merupakan hal yang kompleks. Perusahaan memproduksi barang dengan jumlah dan variasi yang berbeda-beda. Pada dasarnya, proses pembuatan produk melibatkan berbagai faktor seperti proses, mesin, peralatan, tenaga kerja, serta material. Agar kegiatan produksi menghasilkan sesuatu yang menguntungkan, perusahaan harus dapat mengelola semua faktor tersebut agar barang diproduksi tepat pada waktu yang dibutuhkan, dengan biaya yang ekonomis, dan memiliki kualitas yang baik. Oleh karena itu, dibutuhkan sistem perencanaan dan pengendalian yang matang atas proses produksi tersebut.

2.1.1. Pengertian Penjadwalan Produksi

Penjadwalan produksi secara umum didefinisikan sebagai penetapan waktu dari penggunaan peralatan, fasilitas, dan aktivitas manusia dalam sebuah organisasi (Adam dan Elbert, 1992). Penjadwalan produksi mencakup tahapan *loading*, *sequencing*, dan *detailed scheduling*. Pada tahap *loading*, setiap *job* ditentukan prosesnya, kemudian beban (*load*) setiap mesin ditentukan melalui pekerjaan yang harus diproses, dan ditentukan urutan pengerjaan *job* yang dikenal dengan sebutan *sequencing*. Dari urutan tersebut diatur waktu mulai dan selesainya pekerjaan melalui penjadwalan secara mendetail.

2.1.2. Jenis Penjadwalan Produksi

Penjadwalan produksi menurut Pinedo dan Chao (1999) dibagi menjadi beberapa kriteria, yaitu:

1. Berdasarkan mesin yang dipergunakan dalam proses:
 - penjadwalan pada mesin tunggal (*single machine shop*)
 - penjadwalan pada mesin jamak
2. Berdasarkan pola kedatangan *job*:
 - Penjadwalan statis

Pekerjaan datang bersamaan dan siap dikerjakan pada mesin yang sedang tidak beroperasi. Kondisi semua stasiun kerja dan perlengkapannya selalu tersedia pada saat itu.

Penjadwalan dinamis

Pekerjaan datang terus-menerus pada waktu yang berbeda-beda. Pendekatan yang sering digunakan pada penjadwalan ini adalah penggunaan aturan *dispatching* yang berbeda untuk setiap stasiun kerja.

3. Berdasarkan lingkungan penjadwalan:

Flow Shop

Tiap *job* atau pesanan memiliki rute pengerjaan (*routing*) yang sama. Aliran bisa bersifat diskrit, kontinu, maupun semikontinu.

Job Shop

Setiap *job* atau pesanan memiliki rute pengerjaan yang berbeda-beda, sesuai permintaan konsumen (*complex routing*). Karena kompleksnya aliran, maka penjadwalan pun sangat kompleks. Aliran bersifat diskrit, dan *part* tidak bersifat multiguna (*part* yang mungkin menjadi WIP pada *job* yang satu tidak bisa digunakan pada *job* yang lain).

Assembly Line

Hampir serupa dengan *flow shop*, akan tetapi proses hanya meliputi bagian perakitan dengan volume yang tinggi dan karakteristik produk yang sedikit. Tidak ditemui *buffer inventory*, kecuali pada bagian awal lini perakitan.

2.1.3. Istilah dalam Penjadwalan Produksi

Berikut istilah-istilah beserta notasinya yang digunakan dalam penjadwalan (Pinedo dan Chao, 1999):

Setiap *Job* $i \{i = 1, 2, \dots, n\}$ yang akan dijadwalkan pada j mesin $\{j = 1, 2, \dots, m\}$. Proses pengerjaan *job* i pada mesin j disebut dengan operasi O_{ij} .

Waktu proses (*processing time*), p_{ij} , yaitu lamanya waktu yang harus dihabiskan *job* i di mesin j untuk memproses operasi O_{ij} .

Waktu tenggat (*due date*), d_i , adalah batas waktu penyelesaian *job i* yang telah ditentukan. Apabila penyelesaian *job* di luar waktu ini, maka akan dikenakan penalti pada *job* tersebut.

Waktu siap (*release date*), r_i , adalah waktu ketika *job i* masuk ke sistem, yaitu waktu paling awal *job i* bisa mulai diproses. Biasanya $r_i = 0$.

Waktu mulai (*start time*), s_{ij} , adalah waktu mulai diprosesnya *job i* di mesin j .

Waktu penyelesaian (*completion time*), C_{ij} , adalah waktu penyelesaian pemrosesan *job i* pada mesin j .

Makespan biasanya dilambangkan dengan C_{max} , yaitu waktu pengerjaan seluruh *job*.

Keterlambatan (*lateness*), $L_i = C_i - d_i$, adalah selisih antara waktu penyelesaian *job i* dengan waktu tenggatnya. *Lateness* baru dapat dihitung setelah *job i* selesai menjalani semua proses, dan dapat bernilai negatif, nol, atau positif.

Keterlambatan positif (*tardiness*), $T_i = \max(L_i, 0)$, adalah besarnya keterlambatan penyelesaian *job i*.

Keterlambatan negatif (*earliness*), $T_i = \min(L_i, 0)$, adalah besarnya keterlambatan penyelesaian *job i*.

2.1.4. Fungsi Tujuan dan Pengukuran Performa Penjadwalan Produksi

Tujuan yang biasa digunakan untuk menilai performa penjadwalan yang dibuat adalah sebagai berikut:

Meminimumkan *flowtime* dan *makespan*

Memaksimumkan utilisasi (minimasi waktu mesin dan pekerja yang menganggur)

Meminimumkan *inventory* dan WIP (*work in process*)

Meminimumkan keterlambatan, baik *earliness* maupun *tardiness*

Meminimumkan jumlah *job* yang terlambat (*number of tardy job*)

Meminimumkan total biaya penalti atas keterlambatan

2.1.5. Karakteristik dan Kendala Proses Produksi

Kendala penjadwalan produksi menurut Pinedo dan Chao (1999), yaitu:

Precedence Constraints

Kendala ini terjadi ketika suatu *job* baru dapat mulai diproses setelah satu atau sekumpulan *job* lainnya telah selesai diproses

Routing Constraints

Kendala jenis ini merupakan kendala yang sangat umum dijumpai dalam kebanyakan proses manufaktur. Kendala *routing* menjelaskan tentang rute mana yang harus ditempuh oleh suatu *job* melalui sebuah sistem. Sebuah *job* biasanya terdiri dari sejumlah operasi yang harus diproses pada mesin-mesin tertentu berdasarkan sebuah urutan. Keharusan inilah yang kemudian menambah kompleksitas permasalahan penjadwalan.

Material-Handling Constraints

Suatu sistem manufaktur modern yang memiliki sistem *conveyor* sebagai *material handling* yang memindahkan *job* dari satu *work station* ke *work station* yang lain. Tingkat otomasi dari sistem *material handling* bergantung pada tingkat otomasi dari *work station*. Sebagai contoh, jika suatu *work station* memiliki tingkat otomasi yang tinggi, maka waktu proses di *work station* tersebut bersifat deterministik dan tidak bervariasi, sehingga sistem *material handling*-nya juga harus memiliki tingkat otomasi yang sesuai. Sebaliknya, jika tingkat otomasi suatu *work station* rendah, misalkan pekerjaan dilakukan secara manual, maka tingkat kecepatan sistem *material handling*-nya dapat disesuaikan dengan tingkat variasi waktu proses di *work station* yang bersangkutan.

Sequence-Dependent Setup Times

Mesin-mesin dalam suatu proses manufaktur atau *assembly*, seringkali harus mengalami pembenahan ulang (*reconfiguration*) dan pembersihan. Kegiatan ini disebut juga dengan *changeover* atau *setup*. Jika lamanya waktu *setup* bergantung pada jenis *job* yang baru saja diselesaikan oleh mesin-mesin tersebut dan juga bergantung pada *job* apa yang akan diproses selanjutnya, maka waktu *setup*-nya dinyatakan dalam *sijk*. Kegiatan *setup* merupakan suatu aktivitas yang merugikan karena

menimbulkan biaya akibat berkurangnya waktu produktif. Oleh karena itu, kegiatan *setup* sedapat mungkin diminimalisasi untuk menghindari kerugian tersebut.

Preemptions

Preemption berarti jika proses produksi sedang berlangsung, maka dapat dihentikan dan digantikan dengan mengerjakan *job* yang baru datang. Keadaan ini biasanya dikarenakan *job* yang berprioritas rendah dapat disela prosesnya oleh *job* yang berprioritas tinggi.

Storage-Space and Waiting-Time Constraints

Seringkali sistem manufaktur yang memproduksi barang dalam jumlah besar memiliki kendala berupa terbatasnya jumlah ruang yang tersedia untuk WIP. Keterbatasan ini mengakibatkan *blocking*, yakni suatu keadaan di mana gudang WIP yang berada di antara dua *work station* yang berfungsi sebagai *buffer* (penyangga) telah terisi penuh sehingga *work station* yang posisinya berada sebelum *buffer* tidak boleh melepas *job* yang telah selesai diproses pada *work station* tersebut. Keadaan ini menimbulkan masalah lain, yakni munculnya antrian *job* yang menunggu untuk diproses pada *workstation* tersebut.

Tooling and Personnel Scheduling Constraints

Dalam lingkungan mesin paralel, karakteristik mesin yang digunakan harus sama. Jika tidak sama, maka akan mengganggu proses produksi. Selain itu, umur mesin juga mempengaruhi kapasitas produksi yang dihasilkan. Lain halnya dengan mesin, kendala pekerja berkaitan dengan penjadwalan jam kerja operator.

2.2. Penjadwalan *Job Shop*

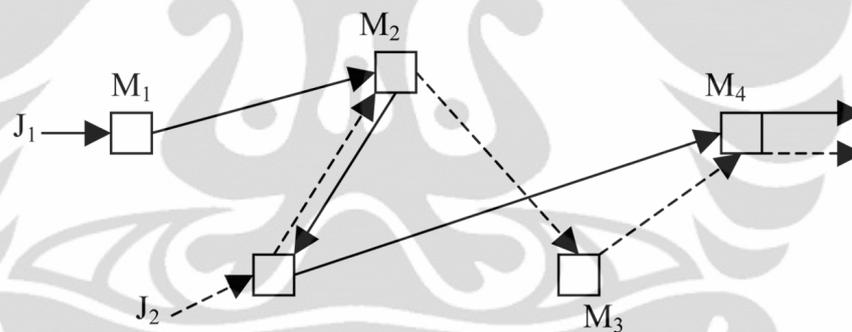
Job shop adalah suatu lingkungan manufaktur dimana *job-job* yang datang memiliki rute pengerjaan atau operasi yang seringkali tidak sama. Bentuk sederhana dari model ini mengasumsikan bahwa setiap *job* hanya melewati satu jenis mesin sebanyak satu kali dalam rutenya pada proses tersebut. Namun ada juga model lainnya dimana setiap *job* diperbolehkan untuk melewati mesin sejenis

lebih dari satu kali pada rutennya. Model ini disebut juga *job shop* dengan *recirculation* (pengulangan).

Karakteristik penjadwalan *job shop* dapat dijabarkan sebagai berikut:

- Ada sejumlah m mesin dan sejumlah n *job*.
- Setiap *job* terdiri dari satu rantai urutan yang dapat berbeda satu sama lain.
- Setiap operasi dalam *job* diproses oleh salah satu mesin yang ada dengan waktu proses yang diasumsikan tetap.
- Setiap proses operasi dapat melewati satu jenis mesin lebih dari satu kali.
- Tidak ada *preemption* (penundaan satu *job* oleh *job* lain).
- Permasalahan penjadwalan untuk model *job shop* merupakan salah satu permasalahan optimasi kombinatorial yang kompleks sehingga disebut *NP-hard* (*NP* merupakan singkatan dari *nondeterministic polynomial*).

Bentuk permasalahan penjadwalan model *job shop* dapat digambarkan dalam bentuk seperti berikut ini:



Gambar 2.1. Contoh Rute Penjadwalan *Job Shop*

2.3. Penalti *Earliness* dan *Tardiness* dalam Penjadwalan Produksi

Seperti telah dibahas sebelumnya dalam istilah penjadwalan produksi, *earliness* dan *tardiness* adalah dua jenis *lateness* (keterlambatan). *Lateness* dapat didefinisikan dengan persamaan:

$$L_j = C_j - d_j \quad (2.1)$$

dengan L_j adalah *lateness* tiap *job* ($j = 1, 2, \dots, n$), C_j adalah total waktu penyelesaian tiap *job*, dan d_j adalah *due date* (batas waktu penyelesaian tiap *job*).

Earliness adalah keterlambatan negatif (nilai seberapa lebih awal waktu penyelesaian suatu *job* dibandingkan *due date* dari *job* tersebut) dan *tardiness* adalah keterlambatan positif (nilai seberapa lebih lambat waktu penyelesaian *job*

dibandingkan due date dari *job* tersebut). Jika E_j dan T_j adalah representasi dari *earliness* dan *tardiness* dari *job* j , maka *earliness* dan *tardiness* dapat didefinisikan sebagai berikut (Baker, 2001, hal. 5-1):

$$E_j = \max\{0, d_j - C_j\} = (d_j - C_j)^+ \quad (2.2)$$

$$T_j = \max\{0, C_j - d_j\} = (C_j - d_j)^+ \quad (2.3)$$

Setiap *job* memiliki sebuah unit penalti *earliness* $\alpha_j > 0$ dan sebuah unit penalti atas *tardiness* $\beta_j > 0$. Dengan asumsi bahwa fungsi penalti adalah linear, maka fungsi objektif dasar masalah *earliness* dan *tardiness* untuk suatu jadwal S dapat ditulis dengan $f(S)$, dimana:

$$f(S) = \sum_{j=1}^n [\alpha_j (d_j - C_j)^+ + \beta_j (C_j - d_j)^+] \quad (2.4)$$

Sesuai dengan definisi sebelumnya,

$$f(S) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j) \quad (2.5)$$

Penelitian untuk minimasi persamaan di atas pernah dilakukan (Nearchou, 2008) untuk mendorong penyelesaian setiap *job* dalam waktu yang sedekat mungkin dengan *due date*.

2.4. Metode Penyelesaian Masalah Penjadwalan Produksi

Masalah penjadwalan produksi dapat diselesaikan dengan menggunakan metode heuristik yang terdiri dari 2 jenis, yaitu:

1. Tipe Heuristik Klasik

Algoritma ini menyusun satu per satu solusi dari masalah penjadwalan. Mulai dari nol, algoritma-algoritma ini memilih mesin-mesin atau *job-job* atau operasi-operasi mana yang harus dijadwalkan terlebih dahulu. Algoritma heuristik klasik yang sering digunakan untuk menyelesaikan penjadwalan *job shop*, yaitu *priority dispatch rule*.

Priority dispatch rule adalah suatu aturan penjadwalan yang mengatur *job* mana pada suatu antrian *job* pada suatu mesin yang harus diproses terlebih dahulu berdasarkan prioritas-prioritas tertentu. Jadi, pada saat suatu mesin telah selesai memroses satu *job*, maka berdasarkan

priority dispatch rule dipilih satu *job* yang memiliki prioritas tertinggi untuk selanjutnya diproses pada mesin tersebut. Berikut ini adalah beberapa aturan yang merupakan *basic priority dispatch rules* (Adam dan Ebert, 1999, hal. 421), yaitu:

First Come First Serve (FCFS)

Menurut aturan ini, urutan penjadwalan dilakukan berdasarkan waktu kedatangan *job* atau pesanan pelanggan. Jadi, *job* yang pertama kali datang, akan dikerjakan terlebih dahulu dan begitu seterusnya untuk *job-job* berikutnya.

Earliest Due Date First (EDD)

Menurut aturan ini, urutan penjadwalan dilakukan berdasarkan pada *due date* setiap *job*. Aturan ini mengabaikan waktu kedatangan dan total waktu proses setiap *job*. Artinya, *job* yang memiliki *due date* yang paling awal di antara *job-job* lainnya dipilih sebagai *job* yang memiliki prioritas paling tinggi untuk diproses pada sebuah mesin. Aturan ini cenderung digunakan untuk meminimumkan maksimum *lateness* pada *job-job* yang ada dalam antrian.

Minimum Slack First (MS)

Menurut aturan ini, *job* diurutkan berdasarkan waktu *slack* yang paling kecil. Pada saat sebuah mesin selesai memroses suatu *job*, maka kemudian dihitung waktu *slack* yang tersisa ($d_i - p_i - t$, 0) dari tiap-tiap *job* yang ada dalam antrian, di mana t adalah waktu sekarang. *Job* yang memiliki waktu *slack* yang paling kecil kemudian dipilih sebagai *job* yang memiliki prioritas paling tinggi untuk diproses selanjutnya. Aturan ini digunakan untuk meminimumkan fungsi tujuan yang berkaitan dengan *due date*, yaitu *lateness* dan *tardiness*.

Shortest Processing Time First (SPT)

Menurut aturan ini, *job* diurutkan berdasarkan pada lamanya waktu proses tiap *job*. Jadi, *job* yang memiliki waktu proses paling singkat akan diproses terlebih dahulu dan kemudian dilanjutkan

oleh *job-job* lainnya sampai pada *job* yang paling lama waktu prosesnya. Aturan ini berguna untuk penyeimbangan beban kerja antar mesin yang disusun secara paralel.

2. Tipe Heuristik Modern (Meta-Heuristik)

Algoritma heuristik modern, atau yang lebih dikenal dengan meta-heuristik, memecahkan masalah penjadwalan produksi dengan melakukan perbaikan mulai dengan satu atau lebih solusi awal. Solusi awal ini dapat dihasilkan secara acak, dapat pula dihasilkan berdasarkan heuristik tertentu. Empat algoritma meta-heuristik yang dapat digunakan dalam memecahkan masalah penjadwalan *job shop* (Takeshi dan Collin, 1998, hal. 1), yaitu:

Simulated Annealing

Ide dasar *Simulated Annealing* terbentuk dari pemrosesan logam. *Annealing* (Memanaskan kemudian mendinginkan) dalam pemrosesan logam ini adalah suatu proses bagaimana membuat bentuk cair berangsur-angsur menjadi bentuk yang lebih padat seiring dengan penurunan temperatur. *Simulated Annealing* biasanya digunakan untuk penyelesaian masalah yang mana perubahan keadaan dari suatu kondisi ke kondisi yang lainnya membutuhkan ruang yang sangat luas.

Tabu Search

Tabu Search merupakan metode optimasi yang menggunakan *short-term memory* untuk menjaga agar proses pencarian tidak terjebak pada nilai *optima local*. Metode ini menggunakan *tabu list* untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Selama proses optimasi, pada setiap iterasi, solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi *tabu list* untuk melihat apakah solusi tersebut sudah ada pada *tabu list*. Apabila sudah ada, maka solusi tersebut tidak akan dievaluasi lagi. Keadaan ini terus berulang sampai tidak ditemukan lagi solusi yang tidak terdapat dalam *tabu list*. Pada metode *tabu search*, solusi baru dipilih jika solusi tersebut, yang merupakan anggota

bagian himpunan solusi tetangga, merupakan solusi dengan fungsi tujuan paling baik jika dibandingkan dengan solusi-solusi lainnya dalam himpunan solusi tetangga tersebut. Tetangga (*neighbour*) dari suatu solusi adalah solusi-solusi lain yang dapat diperoleh dari solusi tersebut dengan cara memodifikasinya berdasarkan aturan-aturan tertentu yang dikenal dengan nama *neighborhood functions*.

Algoritma Genetika

Algoritma Genetika dimodelkan berdasar proses alami, yaitu model seleksi alam oleh Darwin, sedemikian sehingga kualitas individu akan sangat kompatibel dengan lingkungannya (dalam hal ini kendala permasalahan). Algoritma Genetika memberikan suatu alternatif untuk proses penentuan nilai parameter dengan meniru cara reproduksi genetika. Teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang disebut dengan populasi. Setiap individu adalah satu buah solusi unik dan populasi adalah satu himpunan solusi pada setiap tahap iterasi. Algoritma Genetika bekerja untuk mencari struktur individu berkualitas yang terdapat dalam populasi.

2.5. Algoritma Differential Evolution

2.5.1. Definisi Algoritma *Differential Evolution*

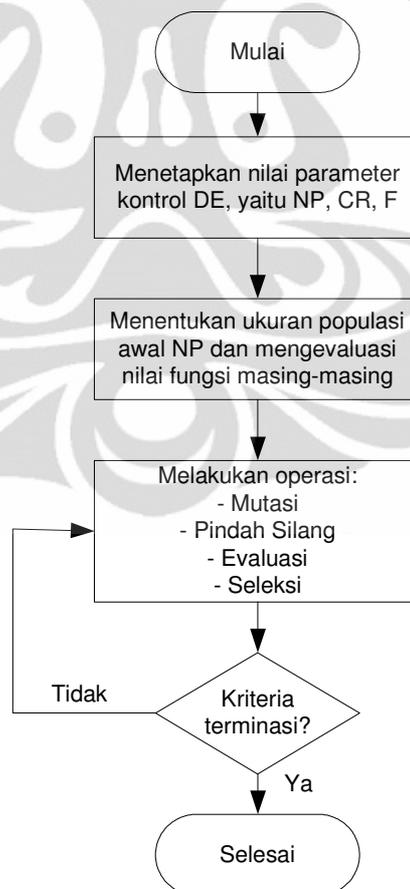
Differential Evolution Algorithm merupakan algoritma optimasi global yang efisien, yang didasarkan pada prinsip evolusi (Price 1999, hal.79-108). Hampir sama seperti algoritma evolusi lainnya, DE menggunakan individu sebagai representasi solusi kandidat. Setiap individu didefinisikan sebagai vektor berdimensi-d ($X \in \mathbb{R}^d$). Individu-individu tersebut merupakan anggota populasi pada generasi ke- g . Populasi dinotasikan sebagai $P(g) = \{X_a, X_b, \dots, X_{NP}\}$. Notasi NP melambangkan ukuran populasi atau jumlah individu dalam populasi pada satu generasi. Nilai NP tidak berubah selama proses pencarian (Lopez, Willigenburg, dan Straten, 2001, hal. 211-216).

Algoritma ini mengeksplorasi populasi solusi potensial untuk menyelidiki ruang pencarian dengan mekanisme operasi mutasi, pindah silang sederhana, dan

penyeleksian. Mutasi merupakan operasi utama yang memberi penekanan pada perbedaan sepasang individu acak anggota populasi (Karaboga dan Okdem, 2004). Pindah silang menampilkan rekombinasi linear antara individu hasil mutasi (*mutant vector*) dengan satu orang tua (*target vector*) untuk menghasilkan satu anak (*Trial vector*). Penyeleksian antara orang tua dengan anak bersifat deterministik (yang terbaik di antara keduanya akan menjadi anggota generasi berikutnya), dengan membandingkan fungsi objektif kedua individu yang bersaing tersebut. Proses ini berulang sampai dicapai titik optimum atau mendekati optimum, berdasarkan kriteria terminasi.

2.5.2. Tahap-tahap Algoritma Diferensial Evolution

Tahap-tahap dalam Algoritma DE meliputi inisialisasi, evaluasi, mutasi, pindah silang, evaluasi, dan penyeleksian. Tahap pengerjaan Algoritma DE dapat dilihat pada gambar 2.3. sedangkan algoritmanya dapat dilihat pada tabel 2.1.



Gambar 2.2. Diagram Alir Pengerjaan Algoritma DE

Tabel 2.1. Algoritma *Differential Evolution*

Inisialisasi
Evaluasi
Ulangi
Mutasi
Pindah silang
Evaluasi
Penyeleksian
Sampai (ditemukan kriteria terminasi)

(Sumber: Karaboga, 2004)

a) Inisialisasi

Tahap ini meliputi penetapan *parameter control* (pengendali parameter) dan penginisialisasian populasi awal. Penentuan *parameter control* berdampak pada performa DE (efektifitas, efisiensi, dan ketangguhan). Tujuan penentuan *parameter control* adalah untuk menentukan solusi yang dapat diterima melalui sejumlah evaluasi fungsi. Tiga *parameter control* dalam DE antara lain:

- 1). *Parameter control* mutasi, F – faktor konstan dan real yang mengendalikan operasi mutasi, berada pada *range* $[0,2]$
- 2). *Parameter control* pindah silang, CR - mengendalikan operasi pindah silang, berada pada *range* $[0,1]$
- 3). Ukuran populasi, NP – jumlah anggota populasi dalam satu generasi. Umumnya $NP = 10 \times d$, di mana d adalah ukuran dimensi vektor/individu, atau jumlah parameter pada individu

DE lebih sensitif terhadap pemilihan F daripada pemilihan CR ¹. CR berperan sebagai *fine tuning element* (elemen penentuan), pada saat operasi pindah silang. Nilai CR yang tinggi, misal $CR = 1$, mempercepat terjadinya konvergensi. Terkadang, untuk beberapa permasalahan, nilai CR perlu diturunkan supaya DE lebih *robust* (tangguh). CR untuk DE yang menggunakan pindah silang binomial (misalnya tipe DE/rand/1/bin) biasanya lebih tinggi daripada untuk DE yang menggunakan pindah silang eksponensial (misalnya tipe DE/rand/1/exp). Umumnya NP tidak berubah

¹ <http://www.aenf.wau.nl/mrs/staff/lopez/research/thesis/chap4.html> (accessed 23 March 2011)

selama pencarian. Namun jika pencarian mengalami kondisi *stuck* , maka NP dapat dinaikkan, atau dengan menaikkan F. F yang berada pada *range* [0.4,1] dinilai efektif. Populasi awal yang diinisialisasikan merupakan populasi solusi awal. Solusi awal yang digunakan bisa diperoleh dari metode heuristik ataupun diperoleh secara acak. Populasi tersebut berisi individu sejumlah NP.

b) Evaluasi

Dari populasi yang ada tersebut, dilakukan evaluasi, untuk menyesuaikan nilai parameter individu terhadap nilai fungsi objektifnya. Pada saat ini dinilai juga individu mana yang layak dijadikan *target vector/target individual*.

c) Mutasi

Individu yang tidak berperan sebagai *target individual* akan mengalami mutasi. Proses mutasi ini melibatkan beberapa individu (umumnya tiga). Proses mutasi diformulasikan dengan rumus:

$$X_{c'} = X_c + F (X_a - X_b)$$

d) Pindah Silang

Dalam rangka mencapai keragaman yang lebih tinggi, individu mutasi $X_{c'}$ dikawinkan dengan X_d (*target individual*) menggunakan operasi pindah silang untuk menghasilkan keturunan atau *Trial individual*. Gen *Trial individual* diwariskan dari $X_{c'}$ dan X_d yang ditentukan melalui operator pindah silang (CF). CF memberi aturan berapa banya rata-rata gen yang bertalian dari individu mutasi dikopi ke turunan.

e) Evaluasi

Trial Individual akan dievaluasi, untuk menyesuaikan nilai parameter individu terhadap nilai fungsi objektifnya.

f) Penyeleksian

Terakhir, dilakukan proses penyeleksian, untuk memilih individu manakah (*target individual* ataukah *Trial individual*) yang akan menjadi anggota

populasi generasi berikutnya. *Trial Individual* dapat menggantikan posisi *target individual* pada generasi berikutnya jika dan hanya jika nilai fungsi objektifnya lebih baik daripada nilai fungsi objektif *target individual*.

g) Terminasi

Proses pencarian akan berhenti jika telah mencapai kriteria terminasi. Kriteria terminasi dapat ditentukan berdasarkan jumlah iterasi maksimum ataupun waktu proses.

DE memiliki beberapa varian (Norman dan Iba, 2005), dinotasikan dalam *DE/x/y/z*, di mana x mendefinisikan vektor/individu yang akan dimutasi, bisa *random* ataupun *best vector*; y mendefinisikan jumlah *difference vector* yang digunakan; dan z mendefinisikan skema pindah silang, yakni *binomial* atau *exponential*. Varian DE berikut ini berturut-turut adalah *DE/rand/1/exp*, *DE/best/1/exp*, *DE/rand/2/exp*, dan *DE/best/2/exp*:

$$y_{G+1}^j = x_{G+1}^j + F(x_{G+1}^k - x_{G+1}^l) \quad (1)$$

$$y_{G+1}^j = x_{G+1}^{best} + F(x_{G+1}^j - x_{G+1}^k) \quad (2)$$

$$y_{G+1}^j = x_{G+1}^j + F(x_{G+1}^k - x_{G+1}^l) + F(x_{G+1}^m - x_{G+1}^n) \quad (3)$$

$$y_{G+1}^j = x_{G+1}^{best} + F(x_{G+1}^j - x_{G+1}^k) + F(x_{G+1}^l - x_{G+1}^m) \quad (4)$$

Algoritma DE dapat diaplikasikan pada fungsi kontinu *nonlinear*. Menurut Srikanta dan Kodali (2005), DE memiliki beberapa keuntungan, antara lain konsepnya sederhana, cocok untuk aplikasi praktis, strukturnya sederhana, penggunaannya mudah, cepat dalam pencarian solusi, dan bersifat *robust* (tangguh, dalam arti memiliki standar deviasi yang kecil).

2.5.3. Penerapan Algoritma DE pada Penjadwalan *Job Shop*

Algoritma DE yang akan diterapkan pada permasalahan ini menggunakan varian *DE/rand/1/bin* yang diperkenalkan oleh Storn dan Price (Tasgetiren et al, 2004). Adapun *pseudo code* algoritma DE yang akan digunakan adalah sebagai berikut:

Inisialisasi parameter
 Inisialisasi *target population*
 Melakukan permutasi operasi
 Menentukan *job repetition*
 Evaluasi
 Do{
 Membentuk *mutant population*
 Membentuk *Trial populaiton*
 Melakukan permutasi operasi
 Menentukan *job repetition*
 Mengevaluasi *Trial population*
 Proses penyeleksian
 }While (Berhenti)

2.5.3.1. Elemen-elemen dasar algoritma DE

Sebelum memulai proses pencarian solusi optimum pada permasalahan penjadwalan *job shop* dengan metode algoritma *Differential Evolution*, diperkenalkan terlebih dahulu elemen-elemen dasar algoritma DE yang akan digunakan. Elemen-elemen dasar tersebut adalah sebagai berikut (Tasgetiren et al, 2004):

Individual target: X_i^t individu ke-i anggota populasi pada generasi ke-t, dan diuraikan sebagai $X_i^t = |X_1^t, X_2^t, X_3^t, \dots, X_n^t|$ di mana X_{ij}^t merupakan nilai dimensi individu ke-i terhadap dimensi ke-j ($j = 1, 2, \dots, n$)

Mutant individual: V_i^t individu ke-i anggota populasi pada generasi ke-t, dan diuraikan sebagai $V_i^t = |V_1^t, V_2^t, V_3^t, \dots, V_n^t|$ di mana V_{ij}^t merupakan nilai dimensi individu ke-i terhadap dimensi ke-j ($j = 1, 2, \dots, n$)

Trial individual: U_i^t individu ke-i anggota populasi pada generasi ke-t, dan diuraikan sebagai $U_i^t = |U_1^t, U_2^t, U_3^t, \dots, U_n^t|$ di mana U_{ij}^t merupakan nilai dimensi individu ke-i terhadap dimensi ke-j ($j = 1, 2, \dots, n$)

Target population: X^t kumpulan individu X_i^t sejumlah NP dalam *target population* pada generasi ke-t

Mutant population: V^t kumpulan individu V_i^t sejumlah NP dalam *target population* pada generasi ke-t

Trial population: U^t kumpulan individu U_i^t sejumlah NP dalam *target population* pada generasi ke-t

Operasi permutasi: π_i^t operasi permutasi *job* terhadap individu X_i^t .

Diuraikan sebagai $\pi_i^t = [\pi_{i1}^t, \pi_{i2}^t, \dots, \pi_{in}^t]$, dimana π_{ij}^t merupakan penugasan operasi ke-j individu ke-i, pada iterasi ke-t.

Konstanta mutasi: $F \in (0,2)$

Konstanta pindah silang: $CR \in (0,1)$

Fungsi objektif: Dalam suatu masalah minimasi, fungsi objektif dilambangkan dengan $f_i(\pi_i^t \leftarrow X_i^t)$. Dalam penelitian ini, fungsi objektif yang digunakan adalah $f_i(\pi_i^t \leftarrow X_i^t) = \min \sum_{j=1}^n (\beta_j T_j)$ dimana β_j adalah penalti atas satu unit tardiness dan T_j adalah total unit tardiness yang terjadi, sedangkan untuk *earliness* tidak dikenakan penalti (penalti dianggap bernilai nol).

Kriteria terminasi: adalah kondisi dimana program akan berhenti berjalan, bisa dalam bentuk jumlah maksimum generasi atau waktu maksimum CPU bekerja.

Tabel 2.2. Kriteria Terminasi

Dimension, j	1	2	3	4	5	6	7	8	9
X_{ij}^t	1.8	-0.99	3.01	0.72	-0.45	-2.25	5.3	4.8	1.9
φ_{ij}^t	6	2	5	4	1	9	3	8	7
π_{ij}^t	2	1	2	2	1	3	1	3	3

(Sumber: Tasgetiren 2004)

Untuk melakukan proses pencarian solusi optimal pada permasalahan penjadwalan *job shop* dengan metode algoritma DE, dilakukan beberapa tahap atau prosedur. Prosedur ditunjukkan dalam diagram alir yang tertera pada gambar 2.5. Prosedur tersebut dijabarkan sebagai berikut (Tasgetiren et al, 2004):

1. Tahap inisialisasi

- Menetapkan $t = 0$, CR, F, dan NP

Dimensi : jumlah *job*

- Membangun individu awal sebanyak NP, yakni $\{X_i^t, i = 1, 2, \dots, NP\}$,
dimana $X_i^0 = [X_{i1}^0, X_{i2}^0, \dots, X_{i, nm}^0]$; $X_{ik}^0 = X_{\min} + (X_{\max} - X_{\min}) \times r$
 $X_{\min} = -1$; $X_{\max} = 1$; $r =$ bilangan random (0-1)
- Melakukan operasi permutasi $\pi_i^0 = [\pi_{i1}^0, \pi_{i2}^0, \dots, \pi_{i, n}^0]$ untuk setiap X_i^0
dengan mengaplikasikan aturan *Smallest Position Value* (SPV).
- Mengevaluasi setiap individu i dalam populasi dengan menggunakan
fungsi objektif $f_i^0(\pi_i^0 \leftarrow X_i^0)$ ($i = 1, 2, \dots, NP$), untuk memilih individu
target.

2. Meng-*update* generasi $t=t+1$

3. Membentuk populasi mutan

Untuk setiap individu target, akan dicari individu mutan

$V_i^{t+1} = [v_{i1}^{t+1}, v_{i2}^{t+1}, \dots, v_{i, nm}^{t+1}]$ yang diperoleh melalui operasi

$$V_i^{t+1} = X_{ai}^t + F(X_{bi}^t - X_{ci}^t), (a_i \neq b_i \neq c_i)$$

4. Membentuk populasi *Trial*

Individu trial $U_i^{t+1} = [u_{i1}^{t+1}, u_{i2}^{t+1}, \dots, u_{i, nm}^{t+1}]$ diperoleh melalui operasi:

$$u_{ik}^{t+1} = \begin{cases} v_{ik}^{t+1}, & \text{if } r_{ik}^{t+1} \leq CR \\ x_{ik}^t, & \text{otherwise} \end{cases}$$

CR adalah konstanta pindah silang pada *range* (0,1), dan r_{ik}^{t+1} adalah bilangan acak *uniform* antara 0 sampai 1.

5. Melakukan operasi permutasi *job*

Operasi permutasi *job* dilakukan dengan menerapkan aturan SPV untuk

melakukan operasi permutasi $\phi_i^t = [\phi_{i1}^t, \phi_{i2}^t, \dots, \phi_{i, nm}^t]$ ($i = 1, 2, \dots, NP$).

6. Mengevaluasi populasi *Trial*

Evaluasi populasi trial menggunakan fungsi objektif $f_i^{t+1}(\pi_i^{t+1} \leftarrow U_i^{t+1})$

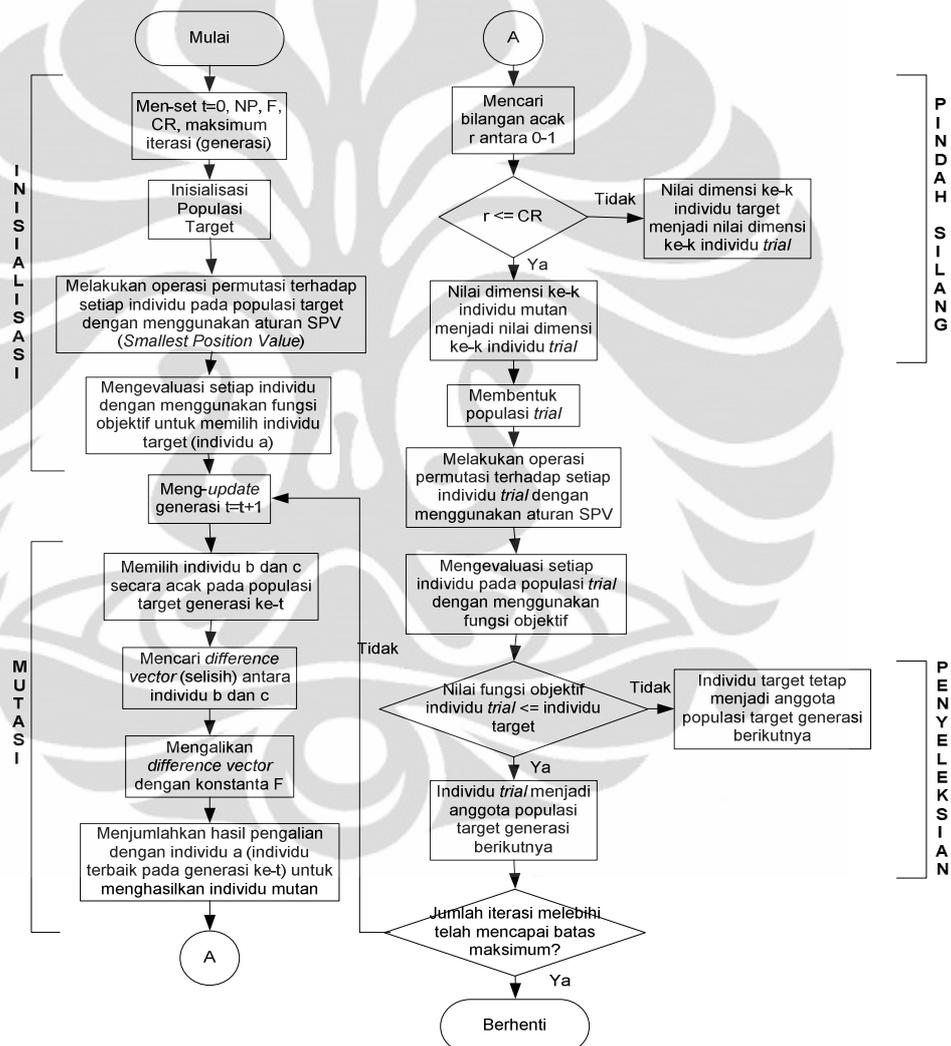
($i = 1, 2, \dots, NP$)

7. Melakukan penyeleksian

Nilai fungsi objektif individu $trial$ U_i^{t+1} akan dibandingkan dengan individu target generasi sebelumnya, X_i^t , untuk menentukan apakah individu $Trial$ tersebut layak menjadi anggota populasi target generasi berikutnya atau tidak.

8. Menghentikan operasi pencarian

Jika jumlah iterasi sudah mencapai jumlah iterasi yang telah ditentukan, maka algoritma dihentikan, namun jika belum maka kembali ke tahap 2.



Gambar 2.3. Diagram Alir Algoritma DE untuk *Job Shop Sequencing Problem*

2.6. Model Matematis

Untuk model matematis, digunakan pendekatan model Manne yang dimodifikasi oleh Cemal Ozguven et al. (2010). Model pada jurnal ini dipakai karena model ini dibangun dengan tujuan mengurutkan operasi-operasi pada

mesin, sehingga waktu penyelesaian seluruh *job* menjadi minimum. Dari model Cemal Ozguven et al. ini yang dipakai adalah masalah fleksibilitas rute yang bersesuaian dengan permasalahan yang dihadapi dalam penelitian ini dengan tujuan meminimumkan total biaya produksi.

Pada penelitian Cemal Ozguven, model dibandingkan dengan model yang diajukan oleh Fattahi et al. (2007). Untuk masalah dengan ukuran kecil kedua model saling bersinggungan; untuk ukuran sedang model Cemal Ozguven lebih baik dari model Fattahi. Namun, secara keseluruhan model Cemal Ozguven lebih baik dari model Fattahi dari sisi waktu CPU, nilai C_{\max} dan ukuran model permasalahan. Model Cemal Ozguven dibandingkan dengan jurnal terbarunya Fattahi et al. (2009) mempunyai kemiripan. Operasi dari *job* model Fattahi ini bisa *overlap* sepanjang barang pertama sudah selesai kemudian akan diproses otomatis pada mesin berikutnya. Karena belum ada literatur pembandingan terhadap model Fattahi terbaru ini, maka dalam penelitian digunakan model Cemal Ozguven. Gambaran model matematis *job shop* tersebut adalah sebagai berikut:

Fungsi Tujuan : Meminimumkan Total Biaya Produksi

$$f(B) = \sum (b_i + \beta_i(C_i - d_i))$$

Penjelasan notasinya sebagai berikut;

β_i = penalti *job* *i* terhadap keterlambatan positif

d_i = *due date job i*

b_i = biaya produksi *job i*

i = *job* (1,2,3.....n)

j = operasi (1,2,3.....m)

k = mesin (1,2,3.....o)

Keterlambatan negatif sama dengan nol

$$C_{\max} \geq S_{ijk} + P_{ijk}$$

Variabel Keputusan

X_{ijk} = 1, jika mesin k dipilih untuk operasi O_{ij} ; 0, sebaliknya

S_{ijk} = Waktu mulai operasi O_{ij} pada mesin k

C_{ijk} = Waktu selesai operasi O_{ij} pada mesin k

$$Y_{ij,ij+1,k} = \begin{cases} 1 & \text{jika operasi } O_{ij} \text{ mendahului operasi } O_{ij+1} \text{ pada mesin } k \\ 0 & \text{jika operasi } O_{ij+1} \text{ mendahului operasi } O_{ij} \text{ pada mesin } k \end{cases}$$

C_i = Waktu selesai *job i*

M = bilangan besar (∞)

P_{ijk} = Waktu proses operasi O_{ij} pada mesin k

C_{\max} = Waktu selesai keseluruhan pekerjaan (*makespan*)

Dengan kendala-kendala sebagai berikut:

Operasi O_{ij} ditetapkan hanya pada satu mesin

$$\sum_k X_{ijk} = 1$$

Set waktu mulai dan selesai nya pada mesin k sama dengan nol

$$S_{ijk} + C_{ijk} \leq (X_{ijk}) \cdot M$$

Perbedaan antara waktu mulai operasi berikutnya paling tidak sama dengan waktu selesai operasi sebelumnya di mesin k

$C_{ijk} \geq S_{ijk} + P_{ijk}$ atau $S_{ijk} \geq C_{ijk} + P_{ijk}$, dijadikan linier menjadi;

$$MX_{ijk} + C_{ijk} - S_{ijk} \geq P_{ijk}$$

$$M(1-X_{ijk}) + S_{ijk} - C_{ijk} \geq P_{ijk}$$

Operasi O_{ij} dan operasi O_{ij+1} tidak dapat dilakukan pada waktu yang sama pada mesin yang sama

$$C_{ijk} \geq C_{ij+1k} + P_{ijk} \text{ atau}$$

$$C_{ij+1k} \geq C_{ijk} + P_{ij+1k} \quad \text{jadikan linier menjadi;}$$

$$C_{ijk} \geq C_{ij+1,k} + P_{ijk} - M(Y_{ij,ij+1,k})$$

$$C_{ij+1k} \geq C_{ijk} + P_{ij+1k} - (1-Y_{ij,ij+1k}) \cdot M$$

Hubungan antara operasi yang harus didahulukan tidak bertentangan, operasi O_{ij} tidak bisa dimulai sebelum operasi O_{ij-1} selesai

$$\sum_k S_{ijk} \geq \sum_k C_{i,j-1,k}$$

Waktu selesai harus lebih besar atau sama dengan total waktu proses dan waktu pelepasan

$$C_i \geq P_i + r_i$$

Waktu selesai pekerjaan (operasi akhir)

$$C_i \geq \sum_k C_{ijk} \quad \forall i \in J$$

Waktu keseluruhan pekerjaan (*makespan*)

$$C_{\max} \geq C_i \quad \forall i \in J$$

2.7. Design of Experiments (DOE)

Percobaan ialah sebuah pengujian atau rangkaian pengujian yang mana perubahan yang berguna dilakukan pada variabel *input* dari sebuah proses atau sistem sehingga dapat diamati alasan untuk merubah respon dari proses. DOE merupakan suatu alat statistik yang penting dalam dunia industri untuk peningkatan performa/kinerja dari sebuah proses manufaktur. DOE juga dapat menjadi sebuah aplikasi dalam pengembangan proses baru. DOE juga dikaitkan dengan proses memanipulasi faktor-faktor yang terkendali agar tidak hanya menentukan efek faktor tersebut pada *output* yang diharapkan tetapi juga menemukan kombinasi dari faktor-faktor yang ada agar dihasilkannya *output* yang maksimum.

2.7.1. Tujuan Design of Experiments

Tujuan dari dilakukannya percobaan adalah (Montgomery, 1996, hal.2):

1. Menentukan variabel paling berpengaruh pada *output* y .
2. Menentukan nilai optimal variabel x agar dicapai nilai y yang ideal.
3. Menentukan nilai optimal variabel x agar variansi nilai y minimum.
4. Menentukan nilai optimal variabel x agar pengaruh dari faktor yang tidak dapat dikendalikan z_1, z_2, \dots, z_q minimum.

2.7.2. Tipe Percobaan

Ada empat tipe percobaan yaitu *Trial and error*, *one factor at a time*, *full factorial*, *fractional factorial* (Fryman, 2002, hal. 320).

1. *Trial and Error Experiments*

Trial and Error Experiments merupakan percobaan memanipulasi satu faktor tanpa memperhatikan faktor lainnya. Kelemahan dari metode ini adalah kurang akuratnya hasil yang diperoleh, memakan biaya yang tinggi, waktu yang lama, dan tidak efisien.

2. *One Factor at a Time Experiments*

Perbedaan metode ini dengan percobaan *Trial and error* yaitu:

- a. ada rangkaian faktor yang diamati, tetapi hanya satu faktor saja yang diubah pada setiap melakukan percobaan.

- b. hanya satu faktor yang diubah-ubah sementara faktor lainnya dianggap konstan.

Percobaan ini merupakan perkembangan dari metode *Trial and error* yang membentuk pendekatan metode percobaan yang sistematis. Kelemahan dari percobaan ini adalah bahwa hasil yang diharapkan kadang tidak tercapai, memakan waktu yang lama, tidak efisien, dan dapat memberikan kesimpulan yang salah dalam suatu percobaan.

3. *Full Factorial*

Percobaan *full factorial* berbeda dengan dua percobaan sebelumnya di mana setiap kombinasi faktor diuji pada level yang berbeda-beda. Metode ini akan memiliki beberapa keuntungan dibandingkan dua metode sebelumnya, sebab kesimpulan yang didapat akan lebih akurat karena setiap kombinasi faktor diujicobakan. Akan tetapi, kelemahan dari metode ini adalah waktu yang diperlukan serta biaya yang dikeluarkan akan besar dengan menjalankan semua kombinasi faktor. Jumlah percobaan/*treatment* yang harus dicoba akan bertambah besar secara signifikan apabila jumlah faktor bertambah.

Uji Hipotesis dalam *Factorial Design*

Misalkan ada 2 faktor yaitu A dan B, maka uji hipotesis yang terjadi yaitu:

- Melihat apakah ada pengaruh dari faktor A:

H_0 : $\tau_1 = \tau_2 = \dots = \tau_a = 0$ (tidak ada pengaruh yang signifikan dari faktor A)

H_1 : $\tau_i \neq 0$ (ada pengaruh yang signifikan dari faktor A)

- Melihat apakah ada pengaruh dari faktor B:

H_0 : $\beta_1 = \beta_2 = \dots = \beta_b = 0$ (tidak ada pengaruh yang signifikan dari faktor B)

H_1 : $\beta_j \neq 0$ (ada pengaruh yang signifikan dari faktor B)

- Melihat interaksi antara faktor A dan B dilakukan dengan:

H_0 : $(\tau\beta)_{ij} = 0$ (tidak ada interaksi yang signifikan antara faktor A dan B)

H_1 : $(\tau\beta)_{ij} \neq 0$ (ada interaksi yang signifikan dari faktor B)

4. *Fractional Factorial*

Banyaknya jumlah percobaan yang harus dilakukan pada *full factorial*, membuat metode tersebut tidak selalu bisa diterapkan pada semua

eksperimen/percobaan, terlebih dengan adanya keterbatasan waktu dalam melakukan percobaan. Oleh karena itu, ada metode yang disebut *fractional factorial*. Metode ini akan menjalankan percobaan hanya sebagian/seporsi dari setiap kombinasi yang mungkin. Percobaan *fractional factorial* merupakan salah satu metode yang paling banyak digunakan dalam pengembangan produk dan peningkatan proses. Penggunaan utama dari metode ini adalah untuk *screening experiments* (menyeleksi kombinasi percobaan).

2.7.3. Prinsip Dasar

Menurut Montgomery (1996), ada tiga prinsip dasar dalam melakukan perancangan percobaan adalah *replication*, *blocking*, dan *randomization*.

1. *Replication* (Replikasi)

Replikasi memiliki 2 manfaat penting yaitu memudahkan *experimenter* untuk mendapatkan estimasi kesalahan dari percobaan yang dilakukan dan membantu *experimenter* mendapatkan perkiraan pengaruh efek dari faktor yang digunakan dalam percobaan lebih akurat.

2. *Randomization* (Randomisasi)

Randomisasi berarti urutan percobaan yang akan diuji dilakukan secara acak untuk menghindari terjadinya efek luar yang mempengaruhi hasil percobaan sehingga percobaan tidak valid/bias. Apabila kita tidak melakukan randomisasi, maka ada kemungkinan percobaan tersebut bisa dipengaruhi oleh faktor lingkungan, kelelahan operator, dan kelainan material yang digunakan, dan lain-lain.

3. *Blocking*

Blocking adalah suatu teknik yang digunakan untuk meningkatkan keakuratan dari percobaan. Dengan memblok, kita membagi percobaan ke dalam kelompok atau grup. Sistem blok diberlakukan karena ada kemungkinan terjadinya perbedaan nilai akhir yang cukup jauh apabila percobaan tersebut tidak dikelompokkan.

BAB III

PENGUMPULAN DATA

3.1. Profil Perusahaan

Pada awalnya, perusahaan ini merupakan sebuah perseroan komaditer atau yang sering disebut dengan CV (*Comanditaire Vennootschap*). Kegiatan usaha perseroan ini berfokus pada ekspor pallet kayu ke Jepang, Taiwan dan Korea. Pemilik perseroan ini dipercaya oleh pihak pembeli di luar negeri sebagai *Quality Control Man* untuk memeriksa pallet-pallet yang akan di ekspor sekaligus mencari pallet kayu di Indonesia untuk diekspor ke luar negeri. Seiring dengan perkembangan usahanya, perseroan komaditer ini berubah menjadi perseroan terbatas (PT) pada tahun 2003.

Sampai saat ini, perusahaan tempat dilakukannya penelitian merupakan sebuah perusahaan swasta yang bergerak di bidang produksi kemasan kayu (pallet kayu). Pallet kayu yang diproduksi sesuai dengan standar internasional ISPM #15, Semi Permanent Immunization Treatment (SPIT) dan ISO 9001:2008. Aturan ISPM #15 bertujuan untuk mencegah terjadinya perpindahan hama baik berupa serangga, rayap dan jamur yang ada pada kemasan kayu dari satu negara ke negara lainya serta mencegah terjadinya peraturan sepihak antar negara-negara dagang dalam perdagangan. Regulasi ISPM #15 secara khusus diperuntukkan terhadap semua komoditi ekspor yang menggunakan kemasan kayu seperti pallet kayu dan peti kayu atau box kayu, skid kayu serta berbagai produk kayu mentah lainnya.

Perusahaan ini memasarkan dan menjual produknya kepada berbagai macam industri seperti industri kimia, industri keramik, industri kertas, dan sebagainya. Sampai saat ini, perusahaan telah memiliki lebih dari dua puluh pelanggan. Untuk menjaga serta meningkatkan kualitas produknya, perusahaan melakukan kerjasama dengan lembaga peneliti maupun lembaga pendidikan tingkat tinggi dalam melakukan penelitian yang mendukung usaha tersebut.

3.2. Jenis Produk

Pallet adalah bahan kemasan yang sering digunakan untuk menyimpan dan mengangkut barang dari gudang ke distributor. Pallet kayu sering digunakan karena murah, fleksibel dan mudah diperbaiki. Produksi yang cepat dan kemampuan yang handal dalam menahan beban adalah dua faktor tambahan saat memilih pallet kayu. Tidak seperti pallet yang terbuat dari plastik, jika pallet kayu rusak, perbaikan dapat dilakukan untuk mengembalikan mereka ke kondisi semula sehingga dapat digunakan kembali seperti biasa. Banyak model pallet kayu dibuat dari berbagai jenis papan dan balok, sehingga hanya papan atau balok yang rusak saja yang diganti atau dihilangkan

Kayu sebagai material pengepakan, penyangga, pelindung dan pembungkus barang sering digunakan dalam perdagangan internasional, baik ekspor, impor maupun yang dilalulintaskan antar area. Untuk kepentingan kegiatan perpindahan komoditas tersebut, kemasan kayu yang diproduksi perusahaan merupakan pallet kayu yang sudah dinyatakan sesuai dengan standar (sertifikasi) yang berlaku.

Negara tujuan ekspor biasanya memiliki aturan yang berbeda mengenai standar pallet kayu yang memenuhi syarat untuk kegiatan perdagangan. Tabel berikut ini berisi daftar jenis pallet kayu yang utama digunakan dalam perdagangan internasional.

Tabel 3.1. Standar Internasional Ukuran Pallet

Ukuran Pallet	Jenis Pallet
1000mm x 1200mm	Standar Pallet Eropa I (diadaptasi di banyak negara di dunia)
800mm x 1200mm	Standar Pallet Eropa II
1100mm x 1100mm	Standar Pallet Asia (T11)
1165mm x 1165mm	Standar Pallet Australia

Perusahaan memproduksi berbagai jenis pallet kayu baik pallet kayu yang memiliki ukuran dan spesifikasi sesuai dengan standar internasional di atas maupun pallet kayu dengan ukuran dan spesifikasi yang sesuai dengan permintaan pelanggan. Jenis produk pallet kayu yang menjadi objek penelitian ini merupakan jenis pallet kayu yang secara regular diproduksi oleh perusahaan dan memiliki

3.3. Alur Produksi

Karena berbahan dasar kayu, pallet harus melalui serangkaian proses produksi yang mengolah bahan mentah kayu menjadi pallet siap pakai di mana setiap proses tersebut bertujuan mengubah bahan mentah kayu menjadi bahan yang memiliki nilai lebih dan pada akhirnya menjadi barang yang berkualitas. Berikut ini adalah tahap-tahap proses pembuatan pallet kayu:

a. Penyortiran (Pemilihan Bahan Baku)

Supplier kayu mengirim bahan baku kayu dalam bentuk batangan dengan berbagai macam ukuran. Bahan baku kayu yang dikirim oleh supplier harus diseleksi terlebih dahulu sebelum digunakan untuk proses produksi. Batang kayu yang cacat atau tidak layak digunakan akan dikembalikan kepada supplier. Sebelum menentukan jenis kayu yang cocok untuk kemasan, maka faktor-faktor yang harus dipertimbangkan adalah densitas, kemudahan pemakuan, kekuatan, kekakuan, panjang kayu.

b. Penyusunan (*stacking*)

Bahan baku yang lolos tahap penyortiran kemudian disusun (ditumpuk) sebelum dimasukkan ke dalam oven. Setiap lapisan tumpukan kayu dipisahkan oleh sebuah pembatas. Tujuannya adalah agar pada saat proses pengeringan di dalam oven, seluruh permukaan batangan kayu mendapatkan pemanasan yang merata sehingga proses pengeringan pun sempurna.

c. Pemanasan/pengeringan (*oven*)

Proses pemanasan kayu adalah proses penurunan kadar air di dalam kayu sampai mencapai kondisi kadar air yang sesuai untuk diproduksi. Proses pemanasan di dalam oven bertujuan untuk membunuh jamur dan insekta yang terdapat di dalam kayu, menghasilkan kayu dengan kekuatan yang lebih tinggi, menghindari pembengkokan dan pembelahan pada ujung kayu, serta mengurangi berat kayu. Karena bahan baku kayu memiliki ketebalan yang berbeda-beda, maka lama proses pemanasannya pun berbeda. Batangan kayu yang berbentuk balok lebih tebal daripada batangan kayu yang berbentuk papan sehingga proses pemanasan balok kayu lebih lama dari proses pemanasan papan kayu. Untuk mengetahui

Universitas Indonesia

apakah proses pemanasan sudah sempurna, petugas menggunakan alat pengukur kadar air dalam kayu yang dilengkapi dengan jarum. Cara pemakaian alat tersebut adalah dengan menusukkan jarum tersebut paling tidak sepertiga dari ketebalan kayu yang diukur.

d. Pembelahan

Proses pembelahan hanya dilakukan untuk mendapatkan papan kayu dengan lebar ± 6 cm. Lebar awal dari papan kayu ini biasanya berukuran 8 - 12 cm. Untuk mendapatkan papan kayu dengan lebar ± 6 cm, proses penyerutan tidaklah cukup sehingga perlu dilakukan proses pembelahan.

e. Penyerutan

Proses penyerutan bertujuan untuk membentuk batangan kayu sampai memiliki ukuran tebal dan lebar yang diinginkan. Ukuran tebal dan lebar yang diinginkan dapat diatur (*set*) pada mesin penyerut. Batangan kayu yang berbentuk balok diserut dengan menggunakan mesin penyerut dua sisi sedangkan batangan kayu yang berbentuk papan diserut dengan menggunakan mesin penyerut empat sisi atau yang disebut juga dengan *molding machine*. Karena mesin penyerut balok kayu hanya bisa menyerut dua sisi permukaan balok, maka balok kayu terlebih dahulu melalui proses *siku*. Proses *siku* bertujuan membuat ruas balok kayu memiliki bentuk siku yang sempurna (tidak bengkok).

f. Pemotongan

Proses pemotongan kayu bertujuan untuk memotong batangan kayu hingga sesuai dengan ukuran panjang yang diinginkan.

g. Perakitan (*assembly*)

Untuk beberapa jenis pallet dengan spesifikasi yang lebih rumit, proses perakitan dibagi menjadi dua tahap, yaitu tahap pembuatan rangka pallet dan tahap perakitan seluruh komponen pallet. Proses inspeksi (*quality control*) menjadi bagian dari proses ini dan dilakukan di akhir proses perakitan. Pallet kayu yang cacat (*major defect*) dipisahkan untuk diberi penanganan selanjutnya (dibuang atau diperbaiki).

h. *Makeup*

Pada proses *makeup*, hal pertama yang dilakukan pekerja adalah memeriksa komponen pallet kayu yang sedikit terbelah kemudian memberikan besi penyambung berbentuk gerigi di bagian tersebut. Setelah itu, proses dilanjutkan dengan proses pendempulan dan diakhiri dengan proses gerinda. Tujuan dari proses *makeup* adalah memperbaiki cacat minor serta memperhalus permukaan pallet kayu.

i. Imunisasi (*Treatment*)

Perusahaan menerapkan Semi Permanent Immunization Treatment (SPIT) untuk proses imunisasi pallet kayu. Tujuan dari imunisasi ini adalah untuk menghindari pallet kayu dari serangan Organisme Pengganggu Tumbuhan (OPT). Proses imunisasi dilakukan dengan mencelupkan pallet kayu ke dalam larutan khusus yang bersifat semi permanen. Oleh karena itu, SPIT merupakan imunisasi semi permanen di mana masa imun dari pallet kayu terhadap serangan OPT terbatas. Jika masa imun pallet kayu sudah habis, maka pallet kayu perlu mendapatkan proses imunisasi ulang. Setelah pallet dicelupkan ke dalam larutan, pallet kayu dikeringkan dengan cara diangin-anginkan (*air-dried*).

j. Pengecapan (*marking*)

Penting bagi perusahaan untuk memberikan informasi mengenai tanggal imunisasi dari pallet kayu. Tujuannya adalah agar pengguna pallet (konsumen) dapat mengetahui masa imun dari pallet kayu. Pengecapan merupakan pemberian tanda merek pallet kayu dan tanggal imunisasinya.

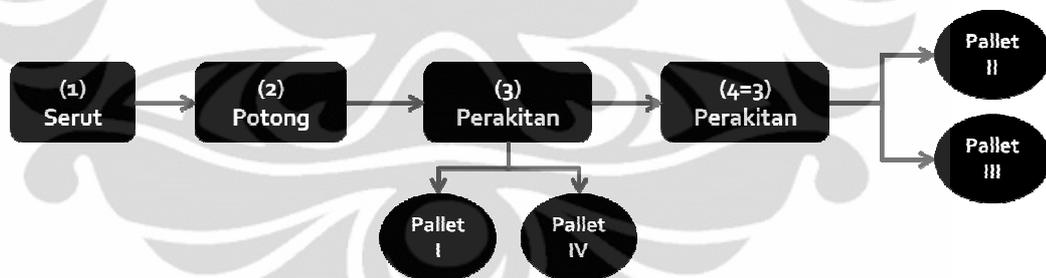
3.4. Data yang Diperlukan

Untuk menyelesaikan permasalahan penjadwalan, diperlukan data-data yang menggambarkan keadaan pelaksanaan proses produksi pallet kayu yang selama ini berlangsung di perusahaan. Dari data-data tersebut, dapat dilakukan pencarian solusi yang optimal serta analisa perbandingan antara pelaksanaan proses produksi yang selama ini berlangsung di perusahaan dengan hasil pencarian solusi yang baru. Data yang dibutuhkan antara lain data aliran proses

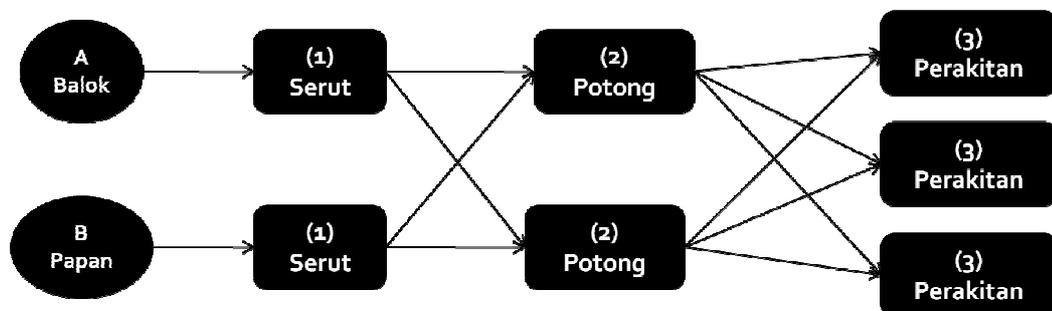
produksi pallet kayu, data lama proses (lama operasi) di tiap *work center*, data jumlah mesin, dan jadwal produksi pallet kayu yang diterapkan perusahaan.

3.4.1. Data Aliran Proses Produksi Kayu

Berdasarkan pengamatan yang telah dilakukan, secara umum proses produksi meliputi penyortiran bahan baku, penyusunan bahan baku (*stacking*), pengeringan bahan baku di dalam oven, penyerutan, penyikuan dan belah (bergantung pada ukuran kayu), pemotongan, perakitan (*assembly*), *make up*, imunisasi (*treatment*), pengecapan (*marking*). Penelitian ini meliputi penjadwalan pada bagian serut, potong, dan perakitan (*assembly*). Ruang lingkup penelitian ini ditentukan berdasarkan pertimbangan bahwa pada proses penyortiran bahan baku, proses pengeringan dalam oven, penyusunan batangan kayu (*stacking*), *makeup*, dan imunisasi tidak terdapat kendala penjadwalan yang berarti. Artinya, kendala yang ada pada proses-proses tersebut kecil pengaruhnya terhadap keberlangsungan seluruh proses produksi. Rute produksi pallet kayu untuk tiap pallet kayu dalam penelitian ini dituangkan dalam gambar 3.1. sedangkan rute proses masing-masing komponen dapat dilihat pada gambar 3.2.



Gambar 3.1. Aliran Produksi Tiap Tipe Pallet



Gambar 3.2. Aliran Produksi Tiap Komponen Pallet

3.4.2. Data Lama Operasi Tiap *Work Center*

Untuk menyelesaikan masalah penjadwalan, perlu diketahui berapa lama waktu yang dibutuhkan sebuah *work center* untuk menyelesaikan operasi produksi. Karena perusahaan belum memiliki data tersebut, pengamatan terhadap lama waktu operasi tiap *work center* dilakukan secara langsung selama proses produksi pallet kayu berlangsung. Pengamatan menggunakan alat bantu *stopwatch* dan dilakukan terhadap sejumlah operasi pada setiap *work center*. Setelah itu, nilai rata-rata lama operasi yang diamati diambil sebagai nilai yang akan digunakan selanjutnya untuk mencari solusi penjadwalan dengan menerapkan Algoritma DE.

Karena data waktu operasi merupakan data primer, perlu dilakukan uji kenormalan dan kecukupan data. Uji kenormalan dilakukan dengan tujuan untuk melihat apakah data yang diambil menunjukkan pola distribusi tertentu. Jika data tersebut menunjukkan pola disitribusi tertentu, maka data dikatakan normal. Uji kenormalan dilakukan dengan metode Anderson-Darling pada Minitab 14. Metode Anderson-Darling digunakan untuk menguji kenormalan data dengan jumlah data yang kecil yaitu n kurang dari sama dengan 25 ($n \leq 25$). Metode Anderson-Darling ini digunakan untuk mengetahui distribusi dari data sampel. Uji ini merupakan modifikasi dari Metode Kolmogorov Smirnov (K-S Test), yaitu K-S Test yang telah diboboti. K-S Test merupakan uji yang bebas distribusi, artinya tidak bergantung pada distribusi data tertentu yang diuji. Sedangkan Metode Anderson-Darling, menggunakan distribusi data tertentu dalam menghitung nilai kritis. Kelebihan Metode Anderson-Darling adalah uji ini lebih sensitif daripada K-S Test, namun mempunyai kelemahan yaitu nilai kritis tersebut harus dihitung dari setiap distribusi data sampel. Metode Anderson-Darling yang merupakan variasi dari Metode Kolmogorov Smirnov, menggunakan *p-value* untuk mengukur apakah sebaran tertentu tersebut menyebar normal atau tidak. *P-value* adalah peluang bahwa sampel yang diuji terletak pada distribusi normal dari suatu populasi (National Institute of Standards and Technology, n.d.). Jika *p-value* yang dihasilkan lebih dari 0.05, maka menerima hipotesa H_0 atau data dikatakan normal. Keseluruhan data primer yang digunakan dalam penelitian telah melalui uji kenormalan dan dikatakan normal. Grafik yang menunjukkan hasil uji normalitas keseluruhan data dapat dilihat pada lampiran 1.

Selain uji normalitas data, perlu dilakukan uji kecukupan data untuk memastikan bahwa jumlah data yang diambil telah memenuhi syarat untuk selanjutnya digunakan dalam pengolahan data. Data waktu operasi di setiap *work center* untuk setiap komponen dituangkan dalam tabel di bawah ini.

Tabel 3.3. Waktu Operasi di Setiap *Work Center*

Jenis Pallet dan Komponen Penyusun		Proses			
		Serut	Potong	Assembly	
		waktu operasi	waktu operasi	Mal	Setel
Pallet I	balok	94,54	36,0165	-	151,71
	papan	104,38	85,5488		
Pallet II	balok	63,94	63,408	61,39	187,13
	papan	72,671	100,253		
Pallet III	balok	38,37	38,0448	46,244	107,969
	papan	97,86	80,202		
Pallet IV	balok	46,49	29,13	-	72,204
	papan	39,14	26,734		

3.4.3. Data Pesanan dan Jam Kerja

Seringkali pelanggan memberikan *order* secara mendadak, misalnya, pallet harus sudah dikirim 2 hari setelah *order* diterima. Hal ini membuat perusahaan harus memprioritaskan jenis pallet mana saja yang harus diproduksi terlebih dahulu. Produksi pallet selama ini dilakukan dengan memprioritaskan pesanan yang memiliki *due date* terdekat. Metode seperti ini dikenal dengan nama *Earliest Due Date*. Data pesanan produk pallet yang diamati dalam penelitian ini untuk periode bulan April 2011 dapat dilihat pada tabel 3.4. sedangkan lampiran 2. merupakan jadwal produksi pallet perusahaan. Jumlah pallet yang diproduksi dituangkan dalam bentuk *job* di mana satu *job* mewakili 50 unit pallet.

BAB IV

PENGOLAHAN DATA

Pengolahan data dilakukan setelah proses pengumpulan data yang diperlukan dalam penelitian selesai agar pengolahan data yang dilakukan dapat menghasilkan keluaran yang sejalan dengan tujuan penelitian. Proses pengolahan data akan dibagi menjadi lima tahap utama, yaitu pembuatan model matematis, pembuatan algoritma pada program, verifikasi dan validasi model yang telah dibuat, *input* data, dan *input* parameter yang digunakan.

4.1. Penyusunan Algoritma

Untuk mencari solusi penjadwalan terbaik dari penelitian ini, data-data yang telah dikumpulkan diolah dengan menggunakan bahasa pemrograman MATLAB. MATLAB secara luas digunakan dalam implementasi algoritma numerik untuk berbagai macam aplikasi. MATLAB bekerja dengan konsep matriks dan memiliki pustaka fungsi matematika dan rekayasa yang super lengkap serta fungsi visualisasi yang bervariasi baik 2D maupun 3D. MATLAB dapat dan biasa digunakan untuk perhitungan teknis seperti elektro, sipil, geodesi, industri, bahkan ekonomi.

Pada MATLAB, setiap data atau variabel akan dianggap sebagai sebuah matriks. Begitupun proses pengolahan datanya, semuanya dilakukan dengan basis matriks. Hal ini sangat membantu para peneliti karena proses pengolahan matriks dalam bahasa pemrograman lainnya membutuhkan fungsi-fungsi yang relatif lebih rumit. Dengan MATLAB, peneliti dapat membuat program algoritma dan mengembangkannya lebih cepat karena peneliti tidak perlu melakukan deklarasi variabel, penentuan tipe data, dan pengalokasian memori terlebih dahulu seperti yang dilakukan jika menggunakan bahasa pemrograman lain. MATLAB memudahkan penggunaannya untuk menjalankan suatu perintah atau beberapa perintah sekaligus tanpa harus mengumpulkan dan menghubungkannya terlebih dahulu sehingga proses iterasi untuk mencari solusi optimal menjadi lebih cepat.²

² <http://www.mathworks.com/products/matlab/description2.html> (accessed 15 May 2011)

4.1.1. Langkah-Langkah Penyusunan Algoritma

Diagram alir penyusunan program algoritma *Differential Evolution* (DE) untuk menyelesaikan penjadwalan *job shop* pada perusahaan dapat dilihat pada gambar 2.4. dan program yang telah dibuat dapat dilihat pada lampiran 3. Prosedur penyusunan program tersebut dapat diuraikan sebagai berikut:

1. Inisialisasi populasi awal

Melakukan penyetelan pada generasi ke-0

Dalam penerapan algoritma DE, diperlukan parameter-parameter kontrol yang nantinya akan digunakan pada tahap mutasi (F) dan pindah silang (CR). Selain dua parameter tersebut, digunakan pula parameter ukuran populasi (NP) serta jumlah iterasi yang akan digunakan untuk keseluruhan proses generasi. Penentuan parameter-parameter ini menggunakan bantuan aplikasi program Minitab 14, yaitu *Design of Experiment* (DOE). Pembahasan lebih dalam mengenai proses penentuan parameter dengan menggunakan DOE ini akan dijelaskan pada bagian selanjutnya. Parameter-parameter berdasarkan hasil DOE tersebut, yaitu: NP = 720; F = 0,5; CR = 0,6; dan jumlah iterasi = 500.

Menentukan populasi awal

Ukuran populasi menggambarkan banyaknya individu dalam satu populasi. Dalam matriks populasi, individu-individu merupakan deretan-deretan kolom di mana satu individu diwakilkan oleh satu kolom. Karena *job* yang ada berjumlah 72, maka setiap individu memiliki 72 gen/dimensi. Gen/dimensi tersebut merupakan deretan baris pada matriks populasi. Populasi awal dibentuk dengan setiap dimensi untuk setiap individu ditentukan secara acak melalui rumus:

$$gen\ populasi\ awal = batas_bawah + (batas_atas - batas_bawah) \times bilangan\ acak \quad (4.1)$$

Nilai-nilai *input* dari rumus tersebut yaitu -1 (batas bawah), 1 (batas atas), dan bilangan acak antara 0 dan 1. Karena populasi terdiri dari 720 individu (nilai NP) dan setiap individu terdiri dari 72 dimensi, maka populasi awal merupakan matriks berukuran 72 x 720.

Menentukan vektor permutasi (urutan)

Pada algoritma DE ini, definisi individu dan vektor adalah sama. Setiap individu awal pada populasi awal terdiri dari gen-gen yang memiliki nilai yang berbeda-beda yang dihasilkan melalui rumus acak. Gen-gen ini kemudian diurutkan dari nilai terkecil sampai dengan nilai yang terbesar. Dari pengurutan tersebut, didapatkan vektor permutasi berdimensi 72 di mana nilai setiap dimensinya merupakan indeks hasil pengurutan. Indeks-indeks dalam vektor permutasi ini menggambarkan urutan-urutan pengerjaan *job*. Sebagai contoh, suatu individu awal memiliki tiga gen yang bernilai -0.11, -0.42, dan 0.43. Jika diurutkan, nilai yang terkecil adalah -0.42, kemudian -0.11, dan terakhir yang terbesar adalah 0.43. Nilai tersebut berturut turut berada pada dimensi ke-2, ke-1, dan ke-3 sehingga vektor permutasi terdiri dari dimensi bernilai 2, 1, dan 3. Dari vektor permutasi tersebut dapat diketahui bahwa urutan pengerjaan *job* dimulai dari *job* 2, dilanjutkan dengan *job* 1, dan diakhiri dengan *job* 3.

Mengevaluasi setiap individu

Evaluasi individu dilakukan dengan melihat fungsi objektif yang dihasilkan dari tiap individu. Fungsi objektif yang dilihat pada penelitian ini berupa total biaya keterlambatan. Setelah mendapatkan urutan pengerjaan *job*, vektor permutasi dihubungkan dengan matriks yang terdiri dari waktu proses, *due date*, serta pinalti keterlambatan. Dengan menghubungkan kedua matriks tersebut, dapat dihitung total waktu penyelesaian *job* dan jumlah keterlambatan. Berdasarkan data pinalti tiap *job* dan jumlah keterlambatan ini, dapat diketahui total biaya keterlambatan yang dihasilkan oleh setiap individu. Evaluasi ini bertujuan untuk mengetahui individu pada generasi (iterasi) awal yang memiliki total biaya keterlambatan terkecil yang selanjutnya akan diturunkan pada generasi selanjutnya.

Memperbaharui generasi (iterasi)

Populasi individu pada iterasi awal akan berevolusi membentuk populasi individu iterasi baru. Individu-individu mengalami evolusi melalui serangkaian proses, yang dimulai dengan proses mutasi, proses pindah silang, dan proses penyeleksian. Jika generasi awal disimbolkan sebagai $t = 0$, maka iterasi baru disimbolkan sebagai $t = t + 1$.

2. Proses mutasi

Setelah proses inisialisasi, DE akan memutasi dan me-rekombinasi populasi awal untuk menghasilkan populasi baru. Mutasi dalam konteks genetika berarti perubahan dengan elemen acak. Oleh karena itu, pada algoritma DE, proses mutasi melibatkan dua buah vektor acak yang selisih antara keduanya akan menghasilkan *difference vector*. *Difference vector* ini kemudian akan dikalikan dengan faktor permutasi (F) dan ditambahkan pada vektor target untuk menghasilkan vektor populasi mutan.

3. Proses pindah silang (*crossover*)

Untuk melengkapi strategi pencarian *differential mutation*, DE menggunakan proses pindah silang dengan tujuan meningkatkan diversitas parameter populasi. *Crossover* membangun vektor *trial* dari nilai parameter yang telah dikopi dari dua vektor yang berbeda (Price dan Storn, 2005, hal. 39). Vektor/individu *trial* merupakan hasil rekombinasi (*crossover*) antara individu target dan individu mutan. Oleh karena itu, sebagian gen pada individu *trial* berasal dari individu pada populasi target dan sebagian lainnya berasal dari individu pada populasi mutan. Pengambilan gen dilakukan berdasarkan perbandingan antara bilangan acak yang dihasilkan untuk setiap gen yang bersangkutan dengan operator pindah silang (CR).

4. Proses seleksi

Proses seleksi bertujuan untuk memilih individu yang layak mengikuti proses iterasi selanjutnya. Pemilihan individu ini dilakukan dengan membandingkan fungsi objektif yang dihasilkan tiap individu pada populasi target dengan tiap individu pada populasi *Trial*. Sebagai contoh,

individu 1 pada populasi target menghasilkan total biaya keterlambatan lebih kecil dibandingkan dengan individu 1 pada populasi *trial*, maka yang digunakan dalam proses iterasi selanjutnya adalah individu 1 dari populasi target. Dengan adanya proses seleksi ini, populasi generasi berikutnya akan menjadi lebih baik.

5. Proses terminasi

Setelah populasi baru terbentuk, proses mutasi, pindah silang, dan seleksi akan terjadi berulang kali. Proses ini akan terus berlangsung sampai kriteria terminasi yang telah ditentukan. Pada penelitian ini, kriteria terminasi yang digunakan adalah jumlah iterasi. Program komputer akan secara otomatis menghentikan proses perhitungan jika iterasi yang dilakukan telah mencapai jumlah yang telah ditentukan. Penentuan jumlah iterasi dipengaruhi oleh lamanya waktu perhitungan. Jumlah iterasi yang sangat besar memiliki kemungkinan untuk mencapai hasil yang optimal, tetapi waktu perhitungan yang dibutuhkan akan sangat lama.

4.2. Verifikasi dan Validasi Program

Sebelum program digunakan untuk mengolah keseluruhan data *job*, perlu dilakukan verifikasi dan validasi terhadap program. Tujuan dari verifikasi program adalah untuk memastikan bahwa program berjalan sesuai dengan konsep yang telah ditentukan. Dalam penelitian ini, konsep model dari program yang dibuat adalah adanya perubahan pada total biaya keterlambatan. Total biaya keterlambatan ini merupakan hasil perkalian antara total keterlambatan dengan biaya keterlambatan per satuan waktu. Total keterlambatan yang akan diperhitungkan hanyalah total keterlambatan yang bernilai positif, artinya waktu penyelesaian *job* lebih besar dari batas waktu (*due date*) *job* yang bersangkutan. Jika total keterlambatan bernilai negatif, di mana waktu penyelesaian *job* lebih kecil dari *due date*, maka total biaya keterlambatan tidak diperhitungkan.

Selanjutnya, validasi program dilakukan dengan tujuan membandingkan *output* yang dihasilkan oleh program dengan perhitungan manual. Validasi program dilakukan dengan menggunakan data *dummy* sebanyak 5 buah *job* seperti

yang dapat dilihat pada tabel 4.1. Jika *output* yang dihasilkan oleh keduanya sama, maka program telah tervalidasi.

Walaupun data yang digunakan untuk proses verifikasi dan validasi program merupakan data *dummy*, rute proses pada data *dummy* tersebut sama dengan rute proses produksi pallet yang telah dijelaskan pada bagian pengumpulan data. Rute proses tersebut dapat dilihat pada gambar 3.2.

Tabel 4.1. Data *Dummy*

Data dummy untuk verifikasi dan validasi								
No	Waktu Proses						Due Date	Pinalti
	A1	A2	B1	B2	C1	C2		
1	150	160	60	120	200	0	700	90
2	60	150	60	120	50	150	600	60
3	140	130	70	110	190	0	650	70
4	70	60	50	40	130	0	480	70
5	160	180	60	130	80	170	800	80

Proses A1 dan B1 masing-masing mewakili proses serut dan proses potong untuk balok sedangkan proses A2 dan B2 masing-masing mewakili proses serut dan proses potong untuk papan. Proses selanjutnya, yaitu C1 dan C2, mewakili proses perakitan di mana untuk beberapa tipe pallet, proses perakitan dibagi menjadi dua bagian, yaitu proses pembuatan rangka dan proses perakitan akhir. Untuk tipe pallet yang proses perakitannya tidak terbagi menjadi dua tahap, waktu proses pada rute proses C2 bernilai nol. Nilai parameter Algoritma DE yang digunakan untuk proses verifikasi dan validasi program adalah 0.6 untuk faktor permutasi (F), 0.5 untuk operator pindah silang (CR), dan 5 untuk ukuran populasi (NP). Untuk memudahkan perhitungan, verifikasi dan validasi program hanya menggunakan satu kali iterasi.

4.2.1. Verifikasi Program

Sesuai dengan tujuan penelitian, konsep dari program ini adalah pencarian solusi penjadwalan yang mendekati optimal yang dapat mengurangi waktu *makespan* dan total biaya keterlambatan. Untuk itu, dilakukan perhitungan waktu *makespan* dan total biaya keterlambatan sebelum dan sesudah Algoritma DE

diterapkan untuk pencarian solusi penjadwalan terbaik pada program yang telah dibuat. Tabel di bawah ini menunjukkan perbandingan waktu *makespan* dan total biaya keterlambatan awal dan akhir setelah program dijalankan sebanyak 10 kali iterasi.

Tabel 4.2. Hasil Verifikasi Program

	Awal	Akhir
Total Biaya Keterlambatan	30300	27800
Total Keterlambatan	470	320
Jumlah Keterlambatan	2	2
<i>Makespan</i>	1070	920

Dari tabel di atas dapat dilihat bahwa terjadi perubahan *output* baik pada total biaya keterlambatan, total keterlambatan, jumlah keterlambatan, maupun *makespan*. Dengan demikian, program telah terverifikasi.

4.2.2. Validasi Program

4.2.2.1. Hasil *run* program

Hasil *run* program menunjukkan urutan pengerjaan terbaik yaitu 3-4-1-5-2 dengan total biaya keterlambatan yaitu 23000.

4.2.2.2. Hasil perhitungan manual

Tahap-tahap perhitungan manual pada proses validasi program adalah sebagai berikut:

1. Melakukan pengurutan (permutasi) pada setiap individu pada populasi target

Populasi target untuk proses verifikasi dan validasi merupakan matriks berukuran 5 x 5 di mana kolom pada matriks populasi mewakili individu dalam populasi sedangkan baris menyatakan *job*. Matriks ini berisi bilangan acak yang dibuat oleh program dengan kisaran nilai antara -1 sampai 1. Permutasi bertujuan untuk menemukan urutan pengerjaan *job* untuk setiap individu. Permutasi dilakukan dengan mengurutkan bilangan acak pada suatu kolom mulai dari nilai yang terkecil hingga nilai yang terbesar. Sebagai contoh, pada kolom pertama pada tabel 4.3., bilangan

acak dengan nilai terkecil terletak pada baris ke-5 , yaitu -0,0360; kemudian diikuti dengan bilangan acak pada baris ke-1 dengan nilai -0,4869; baris ke-4 dengan nilai -0,7271; baris ke-2 dengan nilai 0,7423; dan terakhir yang terbesar terletak pada baris ke-3 dengan nilai 0,9908. Dengan demikian, hasil permutasi individu 1 menghasilkan urutan pengerjaan *job* 5-1-4-2-3. Langkah yang serupa dilakukan pada setiap kolom untuk mendapatkan urutan pengerjaan *job* untuk setiap individu. Hasil pengurutan tersebut berupa indeks urutan pengerjaan yang dapat dilihat pada tabel 4.4.

Tabel 4.3 Populasi Target

Individu				
1	2	3	4	5
0,4869	-0,8336	-0,0157	-0,0340	-0,8838
0,7423	-0,9739	-0,1712	0,6579	-0,3118
0,9908	-0,0103	-0,5258	-0,8771	0,0961
0,7271	0,4572	0,8341	-0,1810	-0,6794
-0,0360	0,7985	-0,3232	0,1187	-0,0652

Tabel 4.4. Permutasi Populasi Target

Individu				
1	2	3	4	5
5	2	3	3	1
1	1	5	4	4
4	3	2	1	2
2	4	1	5	5
3	5	4	2	3

2. Menghitung fungsi objektif tiap individu

Fungsi objektif yang dicari adalah total biaya keterlambatan tiap individu.

Langkah perhitungan total biaya keterlambatan adalah sebagai berikut:

- Mengeset waktu mula-mula adalah sama dengan nol.
- Untuk setiap individu, sesuai urutan *job* masing-masing, dicari waktu penyelesaian untuk setiap *job*. Waktu penyelesaian *job* dihitung dengan menambahkan waktu proses *job* yang bersangkutan dengan waktu penyelesaian proses sebelumnya

(*predecessor*). Waktu *predecessor* yang digunakan merupakan nilai yang terbesar yang dipilih dari waktu proses *job* yang bersangkutan pada stasiun kerja sebelumnya atau waktu proses *job* sebelumnya pada stasiun kerja yang sama. Komponen yang melalui mesin A1 untuk proses selanjutnya hanya bisa diproses di mesin B1. Begitupula komponen yang diproses oleh mesin A2 hanya bisa melalui mesin B2 untuk proses selanjutnya. Pada proses ke-3, produk akhir dapat diproses pada mesin manapun dengan waktu *predecessor* terkecil.

- Kemudian nilai keterlambatan dari setiap *job* akan didapatkan dengan mencari selisih waktu penyelesaian setiap *job* dengan *due date* dari *job* yang bersangkutan dengan rumus berikut:

$$\text{Keterlambatan} = \text{Waktu penyelesaian job} - \text{due date} \quad (4.2)$$

Apabila nilai keterlambatan bernilai negatif (*earliness*), nilai tersebut akan berubah menjadi nol.

- Selanjutnya, keterlambatan setiap *job* dikalikan dengan penalti (bobot) masing-masing untuk memperoleh biaya keterlambatan setiap *job*. Kemudian biaya keterlambatan keseluruhan *job* pada individu yang bersangkutan dijumlahkan untuk mendapatkan fungsi objektif, yaitu total biaya keterlambatan.

Tabel-tabel berikut ini menunjukkan proses perhitungan total biaya keterlambatan untuk setiap individu dalam populasi.

Tabel 4.5. Perhitungan Waktu Proses Individu 1 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
5	160	180	220	310	390	560	0
1	310	340	370	460	0	0	660
4	380	400	430	500	630	0	0
2	440	550	500	670	870	720	0
3	580	680	650	790	0	0	980

Tabel 4.6. Biaya Keterlambatan Setiap *Job* Individu 1 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	660	870	980	630	560
Due Date	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-40	270	330	150	-240
Biaya Keterlambatan	0	16200	23100	10500	0
Total Biaya Keterlambatan	49800				

Tabel 4.7. Perhitungan Waktu Proses Individu 2 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
2	60	150	120	270	320	470	0
1	210	310	270	430	0	0	630
3	350	440	420	550	740	0	0
4	420	500	470	590	0	720	0
5	580	680	640	810	1060	0	890

Tabel 4.8. Biaya Keterlambatan Setiap *Job* Individu 2 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	630	470	740	720	1060
Due Date	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-70	-130	90	240	260
Biaya Keterlambatan	0	0	6300	16800	20800
Total Biaya Keterlambatan	43900				

Tabel 4.9. Perhitungan Waktu Proses Individu 3 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
3	140	130	210	240	430	0	0
5	300	310	360	440	0	520	690
2	360	460	420	580	630	780	0
1	510	620	570	740	940	0	0
4	580	680	630	780	0	0	910

Tabel 4.10. Biaya Keterlambatan Setiap *Job* Individu 3 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	940	780	430	910	690
<i>Due Date</i>	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	240	180	-220	430	-110
Biaya Keterlambatan	21600	10800	0	30100	0
Total Biaya Keterlambatan	62500				

Tabel 4.11. Perhitungan Waktu Proses Individu 4 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
3	70	60	140	170	360	0	0
4	140	120	190	210	0	340	0
1	290	280	350	400	0	0	600
5	450	460	510	590	670	840	0
2	510	610	570	730	930	0	780

Tabel 4.12. Biaya Keterlambatan Setiap *Job* Individu 4 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	600	930	360	340	840
<i>Due Date</i>	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-100	330	-290	-140	40
Biaya Keterlambatan	0	19800	0	0	3200
Total Biaya Keterlambatan	23000				

Tabel 4.13. Perhitungan Waktu Proses Individu 5 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
1	150	160	210	280	480	0	0
4	220	220	270	320	0	450	0
2	280	370	340	490	690	0	540
5	440	550	500	680	0	760	930
3	580	680	650	790	980	0	0

Tabel 4.14. Biaya Keterlambatan Individu 5 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	480	690	980	450	930
Due Date	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-220	90	330	-30	130
Biaya Keterlambatan	0	5400	23100	0	10400
Total Biaya Keterlambatan	38900				

- Perhitungan di atas menunjukkan bahwa individu yang menghasilkan fungsi objektif terbaik adalah individu 4 dengan total biaya keterlambatan terkecil, yaitu sebesar 23000 dan urutan pengerjaan *job* 3-4-1-5-2.
- Setelah mendapatkan individu terbaik pada populasi target, selanjutnya gen-gen individu tersebut dijadikan vektor target untuk proses mutasi. Proses mutasi dilakukan untuk menciptakan populasi mutan dengan bantuan dua vektor acak. Selisih (*difference vector*) dari nilai setiap gen pada dua vektor acak ini akan dikalikan dengan faktor permutasi (F) dan kemudian ditambahkan dengan nilai gen yang bersangkutan pada vektor target. Secara singkat, langkah tersebut dapat dituangkan dalam rumus berikut:

$$\text{Populasi Mutan} = ((\text{Vektor Acak 1} - \text{Vektor Acak 2}) \times F) + \text{Vektor Target} \quad (4.3)$$

Individu-individu pada vektor acak berasal dari individu populasi target yang dipilih secara acak oleh program. Akan tetapi, individu pada suatu vektor acak tidak boleh menduduki posisi yang sama pada vektor acak yang lain maupun pada vektor target. Sebagai contoh, individu pertama vektor target, vektor acak 1, dan vektor acak 2 secara berturut-turut berasal dari individu 2, 4, dan 3 pada populasi target. Ketiga kolom tersebut berbeda, tidak boleh ada yang sama. Tabel-tabel di bawah ini menunjukkan vektor target, vektor acak 1, vektor acak 2, dan populasi mutan yang dihasilkan.

Tabel 4.15. Vektor Target

Individu				
1	2	3	4	5
-0,0340	-0,0340	-0,0340	-0,0340	-0,0340
0,6579	0,6579	0,6579	0,6579	0,6579
-0,8771	-0,8771	-0,8771	-0,8771	-0,8771
-0,1810	-0,1810	-0,1810	-0,1810	-0,1810
0,1187	0,1187	0,1187	0,1187	0,1187

Tabel 4.16. Vektor Acak 1

Individu				
1	2	3	4	5
-0,0157	-0,0157	-0,8336	0,4869	-0,8838
-0,1712	-0,1712	-0,9739	0,7423	-0,3118
-0,5258	-0,5258	-0,0103	0,9908	0,0961
0,8341	0,8341	0,4572	0,7271	-0,6794
-0,3232	-0,3232	0,7985	-0,0360	-0,0652

Tabel 4.17. Vektor Acak 2

Individu				
1	2	3	4	5
-0,8336	-0,8838	0,4869	-0,0157	-0,8336
-0,9739	-0,3118	0,7423	-0,1712	-0,9739
-0,0103	0,0961	0,9908	-0,5258	-0,0103
0,4572	-0,6794	0,7271	0,8341	0,4572
0,7985	-0,0652	-0,0360	-0,3232	0,7985

Tabel 4.18. Populasi Mutan

Individu				
1	2	3	4	5
0,4568	0,4869	-0,8263	0,2676	-0,0641
1,1395	0,7423	-0,3718	1,2060	1,0552
-1,1864	-1,2502	-1,4778	0,0329	-0,8133
0,0452	0,7271	-0,3429	-0,2451	-0,8629
-0,5543	-0,0360	0,6195	0,2910	-0,3995

Dari tabel-tabel tersebut dapat dilihat bahwa nilai gen (baris) pertama pada individu (kolom) pertama populasi mutan diperoleh dari hasil perhitungan berikut:

$$\begin{aligned} \text{Populasi Mutan} &= ((\text{Vektor Acak 1} - \text{Vektor Acak 2}) \times F) + \text{Vektor Target} \\ &= ((-0,0157 - (-0,8336)) \times 0.6) + (-0,0340) \\ &= 0,4568 \end{aligned}$$

5. Langkah selanjutnya adalah pindah silang (*crossover*). Proses pindah silang bertujuan untuk membuat populasi trial di mana gen-gen pada populasi trial berasal dari populasi target maupun populasi mutan yang diseleksi dengan menggunakan operator pindah silang (CR). Caranya adalah dengan membuat bilangan acak untuk setiap gen kemudian membandingkan nilai bilangan acak tersebut dengan operator pindah silang. Jika nilai bilangan acak kurang dari atau sama dengan CR, maka gen populasi trial diambil dari gen populasi mutan yang bersangkutan. Sebaliknya, jika nilai bilangan acak lebih besar dari nilai CR maka gen yang digunakan pada populasi trial adalah gen populasi target. Untuk lebih jelasnya, tabel 4.19. menunjukkan populasi trial yang dihasilkan. Angka-angka yang bercetak tebal merupakan gen yang berasal dari populasi mutan sedangkan yang lainnya berasal dari populasi target.

Tabel 4.19. Populasi Trial

Individu				
1	2	3	4	5
0,4568	-0,8336	-0,8263	-0,0340	-0,8838
1,1395	-0,9739	-0,1712	1,2060	1,0552
0,9908	-0,0103	-0,5258	-0,8771	0,0961
0,0452	0,7271	0,8341	-0,1810	-0,8629
-0,5543	0,7985	-0,3232	0,1187	-0,0652

6. Selanjutnya, menghitung fungsi objektif yang dihasilkan masing-masing individu pada populasi trial. Langkah-langkah yang dilakukan untuk mencari total nilai keterlambatan sama dengan langkah yang dilakukan pada saat mencari total nilai keterlambatan populasi target, yaitu dimulai dengan pengurutan *job* dan dilanjutkan dengan perhitungan waktu

penyelesaian masing-masing *job* pada tiap individu untuk mendapatkan biaya keterlambatan. Tabel-tabel berikut ini menunjukkan urutan *job* serta perhitungan total biaya keterlambatan populasi trial.

Tabel 4.20. Permutasi Populasi Trial

Individu				
1	2	3	4	5
5	2	1	3	1
4	1	3	4	4
1	3	5	1	5
3	4	2	5	3
2	5	4	2	2

Tabel 4.21. Perhitungan Waktu Proses Individu 1 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
5	160	180	220	310	390	560	0
4	230	240	280	350	0	0	480
1	380	400	440	520	720	0	0
3	520	530	590	640	0	830	0
2	580	680	650	800	1000	0	850

Tabel 4.22. Biaya Keterlambatan Setiap *Job* Individu 1 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	720	1000	830	480	560
Due Date	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	20	400	180	0	-240
Biaya Keterlambatan	1800	24000	12600	0	0
Total Biaya Keterlambatan	38400				

Tabel 4.23. Perhitungan Waktu Proses Individu 2 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
2	60	150	120	270	320	470	0
1	210	310	270	430	0	0	630
3	350	440	420	550	740	0	0
4	420	500	470	590	0	720	0
5	580	680	640	810	1060	0	890

Tabel 4.24. Biaya Keterlambatan Setiap *Job* Individu 2 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	630	470	740	720	1060
<i>Due Date</i>	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-70	-130	90	240	260
Biaya Keterlambatan	0	0	6300	16800	20800
Total Biaya Keterlambatan	43900				

Tabel 4.25. Perhitungan Waktu Proses Individu 3 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
1	150	160	210	280	480	0	0
3	290	290	360	400	0	590	0
5	450	470	510	600	0	850	680
2	510	620	570	740	790	0	940
4	580	680	630	780	0	980	0

Tabel 4.26. Biaya Keterlambatan Setiap *Job* Individu 3 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	480	940	590	980	850
<i>Due Date</i>	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-220	340	-60	500	50
Biaya Keterlambatan	0	20400	0	35000	4000
Total Biaya Keterlambatan	59400				

Tabel 4.27. Perhitungan Waktu Proses Individu 4 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
3	70	60	140	170	360	0	0
4	140	120	190	210	0	340	0
1	290	280	350	400	0	0	600
5	450	460	510	590	670	840	0
2	510	610	570	730	930	0	780

Tabel 4.28. Biaya Keterlambatan Setiap *Job* Individu 4 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	600	930	360	340	840
<i>Due Date</i>	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-100	330	-290	-140	40
Biaya Keterlambatan	0	19800	0	0	3200
Total Biaya Keterlambatan	23000				

Tabel 4.29. Perhitungan Waktu Proses Individu 5 Populasi Target

No	Waktu Proses						
	A1	A2	B1	B2	C1	C2	C3
1	70	60	130	180	380	0	0
4	140	120	190	220	0	350	0
5	300	300	360	430	680	0	510
3	440	430	510	540	0	730	0
2	500	580	570	700	900	0	750

Tabel 4.30. Biaya Keterlambatan Setiap *Job* Individu 5 Populasi Target

Job	1	2	3	4	5
Waktu Penyelesaian	380	900	730	350	680
<i>Due Date</i>	700	600	650	480	800
Penalti	90	60	70	70	80
Keterlambatan	-320	300	80	-130	-120
Biaya Keterlambatan	0	18000	5600	0	0
Total Biaya Keterlambatan	23600				

7. Langkah selanjutnya adalah membandingkan total biaya keterlambatan yang dihasilkan masing-masing individu pada populasi target dan populasi

trial. Gen dari individu yang menghasilkan fungsi objektif terbaik akan dijadikan gen populasi target untuk proses iterasi berikutnya. Sebagai contoh, pada tabel tabel 4.31., dapat dilihat bahwa individu 1 populasi trial memiliki total biaya keterlambatan lebih kecil daripada individu 1 pada populasi target. Oleh karena itu, gen individu 1 populasi target baru diambil dari gen individu 1 populasi trial.

Tabel 4.31. Perbandingan Total Biaya Keterlambatan Populasi Target dan Populasi *Trial*

Total Biaya Keterlambatan Populasi Target (menit)	Total Biaya Keterlambatan Populasi Trial (menit)	Asal Populasi Target Iterasi Selanjutnya
49800	38400	populasi trial
43900	43900	populasi trial
62500	59400	populasi trial
23000	23000	populasi trial
38900	23600	populasi trial

Dari perbandingan tersebut diperoleh populasi target baru dengan gen semua individu pada populasi target baru diambil dari populasi trial. Populasi target baru yang terbentuk dapat dilihat pada tabel tabel 4.32.

Tabel 4.32. Populasi Target Baru

Individu				
1	2	3	4	5
0,4568	-0,8336	-0,8263	-0,0340	-0,8838
1,1395	-0,9739	-0,1712	1,2060	1,0552
0,9908	-0,0103	-0,5258	-0,8771	0,0961
0,0452	0,7271	0,8341	-0,1810	-0,8629
-0,5543	0,7985	-0,3232	0,1187	-0,0652

- Karena proses iterasi untuk validasi program hanya dilakukan satu kali, maka perhitungan berhenti sampai didapatkan populasi target baru. Berdasarkan perbandingan total biaya keterlambatan pun dapat diketahui bahwa solusi terbaik yang dihasilkan pada proses validasi ini adalah urutan pengerjaan *job* 3-4-1-5-2 dengan total biaya keterlambatan sebesar 23000. Hasil perhitungan manual ini sesuai dengan keluaran program. Dengan demikian, program telah tervalidasi.

4.3. *Input Data*

4.3.1. *Input Parameter*

Dalam pemilihan parameter, digunakan *Design of Experiment* (DOE). Tujuan dilakukannya DOE adalah untuk mendapatkan kombinasi parameter yang menghasilkan *output* yang terbaik. Algoritma DE menggunakan empat parameter, yakni faktor permutasi (F), faktor pindah silang (CR), ukuran populasi (NP), dan jumlah iterasi. Karena faktor yang digunakan lebih dari dua faktor, maka jenis DOE yang digunakan adalah *full factorial design*. Masing-masing faktor memiliki tiga level, yaitu rendah, sedang, dan tinggi. Dengan demikian, jumlah kombinasi parameter yang diperoleh adalah sebanyak $3^4 = 81$ kombinasi parameter. Untuk setiap kombinasi parameter, dilakukan uji coba sebanyak lima kali sehingga total percobaan pencarian solusi penjadwalan pada program adalah sebanyak 405 kali percobaan.

Pemilihan nilai untuk setiap parameter dilakukan berdasarkan studi literatur. Faktor permutasi (F) umumnya memberikan hasil yang efektif jika nilainya berada pada kisaran 0,4-0,95 (Rönkkönen, Kukkonen, dan Price, 2005, hal. 506-513). Untuk pilihan awal, nilai F yang disarankan adalah sebesar 0,5 atau 0,6 (Storn dan Price, 1997, hal. 341-359). Sedangkan untuk operator pindah silang (CR), nilai yang umum digunakan berkisar antara 0-1. Pilihan awal yang baik untuk nilai CR adalah sebesar 0,1 sedangkan pada umumnya nilai CR yang digunakan relatif besar, yaitu 0,9 sampai 1 (Storn dan Price, 1997). Pada literatur yang sama disebutkan pula bahwa ukuran populasi yang efektif adalah sebesar 5-10 kali dimensi sedangkan Gämperle, Müller, dan Koumoutsakos (2002) merekomendasikan ukuran populasi sebesar 3-8 kali ukuran dimensi (hal. 293-298). Selain itu, jumlah iterasi yang semakin besar memungkinkan pencarian solusi yang lebih optimal. Untuk parameter ukuran populasi dan iterasi, nilai yang digunakan dalam DOE tidak besar. Nilai yang digunakan hanya untuk menunjukkan level masing-masing parameter. Hal ini bertujuan mempersingkat waktu komputasi percobaan. Tabel 4.33. menunjukkan nilai setiap level dari keempat faktor yang digunakan dalam DOE.

Tabel 4.33. Level Tiap Parameter dalam DOE

Level	F	CR	NP	Iterasi
Rendah	0,5	0,4	10	10
Sedang	0,7	0,6	30	30
Tinggi	0,9	0,8	50	50

DOE dilakukan dengan menggunakan program Minitab 14 dengan level signifikansi sebesar 0,05%. Hasil dari 405 percobaan yang dilakukan dapat dilihat pada lampiran 4. Dari percobaan tersebut, diperoleh kombinasi parameter yang menghasilkan *output* total biaya keterlambatan terkecil sebesar 14560, yaitu kombinasi dengan F sebesar 0,5, CR sebesar 0,6, NP sebesar 50, dan jumlah iterasi sebanyak 50 kali iterasi.

4.3.2. *Input Data Pesanan*

Data yang dimasukkan ke dalam program berupa data pesanan pallet yang terdiri dari 72 *job*. Setiap rute produksi yang dilalui oleh setiap *job* diwakili oleh kolom pada matriks data. Kolom 1 menyatakan waktu proses serut balok, kolom 2 untuk proses serut papan, kolom 3 untuk proses potong balok, kolom 4 untuk proses potong papan, dan kolom 5 dan 6 untuk proses perakitan. Selain waktu proses, dimasukkan pula data *due date* serta bobot pinalti untuk masing-masing *job* yang berturut-turut dinyatakan oleh kolom 7 dan 8. Dengan demikian, data yang dimasukkan merupakan data waktu proses keseluruhan *job*, *due date*, serta bobot pinalti yang terangkum dalam matriks berukuran 72 x 8.

4.4. *Pengolahan Data dan Hasil*

Program untuk pengolahan data penelitian ini dijalankan dengan spesifikasi komputer, yaitu Intel Core Duo Processor T5500 (1.66GHz, 667 MHz FSB, 2 MB L2 cache). *Script M File* program untuk perhitungan jadwal perusahaan dapat dilihat pada lampiran 5. dan *Script M File* program untuk mendapatkan solusi (usulan) jadwal dapat dilihat pada lampiran 3.

4.4.1. Hasil Penjadwalan Produksi Perusahaan

Hasil penjadwalan perusahaan dapat dilihat pada tabel 4.34. Penjadwalan perusahaan menghasilkan keterlambatan pada *job* 3, 4, dan 5 (angka yang bercetak tebal). Besarnya keterlambatan untuk masing-masing *job* tersebut adalah 8 untuk *job* 3 182 untuk *job* 4, dan 95 untuk *job* 5.

Tabel 4.34. Penjadwalan Produksi Perusahaan

Penjadwalan Perusahaan												
Urutan Pengerjaan	1	2	3	4	5	6	7	8	9	10	11	12
	13	14	15	16	17	18	19	20	21	22	23	24
	25	26	27	28	29	30	31	32	33	34	35	36
	37	38	39	40	41	42	43	44	45	46	47	48
	49	50	51	52	53	54	55	56	57	58	59	60
	61	62	63	64	65	66	67	68	69	70	71	72
Jumlah Pesanan Yang Terlambat	3											
Total Waktu Keterlambatan	285											
Total Biaya Keterlambatan	23590											
Makespan	12735											

4.4.2. Hasil Penjadwalan Dengan Algoritma DE

Pencarian solusi dengan program MATLAB dilakukan dengan menggunakan parameter yang telah ditentukan berdasarkan hasil DOE dan pertimbangan waktu komputasi untuk mendapatkan hasil penjadwalan yang terbaik. Tabel berikut ini menunjukkan hasil *run* tersebut.

Tabel 4.35. Hasil *Run* Program

Usulan Penjadwalan												
Urutan Pengerjaan	1	4	2	5	3	7	6	8	9	10	11	12
	33	13	16	15	31	30	14	18	19	17	37	22
	68	51	20	24	60	23	27	63	46	29	67	28
	21	65	26	35	25	38	34	36	40	32	49	57
	41	58	39	50	64	42	43	56	69	44	48	72
	47	59	45	53	52	70	66	71	61	54	55	62
Jumlah Pesanan Yang Terlambat	2											
Total Waktu Keterlambatan	252											
Total Biaya Keterlambatan	17640											
Makespan	12544											

Selain urutan pengerjaan *job*, ditampilkan pula *job* yang mengalami keterlambatan melalui gambar di bawah ini. Dari gambar tersebut dapat dilihat bahwa *job* yang mengalami keterlambatan adalah *job* 3 dan 5 dengan total keterlambatan secara berturut-turut masing-masing sebesar 4 dan 248 menit.

```
Keterlambatan terbaik setiap job =
Columns 1 through 13
    0    0    4    0  248    0    0    0    0    0    0    0    0
Columns 14 through 26
    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 27 through 39
    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 40 through 52
    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 53 through 65
    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 66 through 72
    0    0    0    0    0    0    0
```

Gambar 4.1. Keterlambatan Setiap *Job* (*Run* 3)

```
Urutan terbaik=
Columns 1 through 13
    1    4    2    5    3    7    6    8    9    10   11   12   33
Columns 14 through 26
    13   16   15   31   30   14   18   19   17   37   22   68   51
Columns 27 through 39
    20   24   60   23   27   63   46   29   67   28   21   65   26
Columns 40 through 52
    35   25   38   34   36   40   32   49   57   41   58   39   50
Columns 53 through 65
    64   42   43   56   69   44   48   72   47   59   45   53   52
Columns 66 through 72
    70   66   71   61   54   55   62
```

Gambar 4.2. Urutan Terbaik Hasil *Run* Program

4.5. Analisa Hasil

4.5.1. Analisa Skenario Parameter

Seperti yang sudah disebutkan sebelumnya, penelitian ini menggunakan *Design of Experiment* untuk mencari kombinasi parameter yang menghasilkan *output* total biaya keterlambatan yang terkecil. Selanjutnya, hasil 405 percobaan terhadap 81 kombinasi parameter yang telah dilakukan dapat digunakan untuk melihat pengaruh dari setiap parameter serta interaksi antar parameter terhadap *output* yang dihasilkan. Untuk itu, dilakukan *Analysis of Variance* (ANOVA) dengan menggunakan program Minitab 14. Pengaruh tersebut dinilai berdasarkan nilai signifikansi (*p-value*) dari masing-masing faktor. Tingkat kepercayaan yang digunakan adalah 95% sehingga $\alpha = 5\%$ (0,05). Jika nilai *p-value* pada suatu faktor $\leq 0,05$; maka faktor yang bersangkutan berpengaruh signifikan terhadap *output* yang diamati. Sebaliknya, jika *p-value* $> 0,05$; maka faktor tersebut tidak memiliki pengaruh terhadap *output* yang diamati. Tabel berikut ini menunjukkan pengaruh parameter terhadap total biaya keterlambatan.

Tabel 4.36. ANOVA Parameter Algoritma DE

Analysis of Variance for Output, using Adjusted SS for Tests

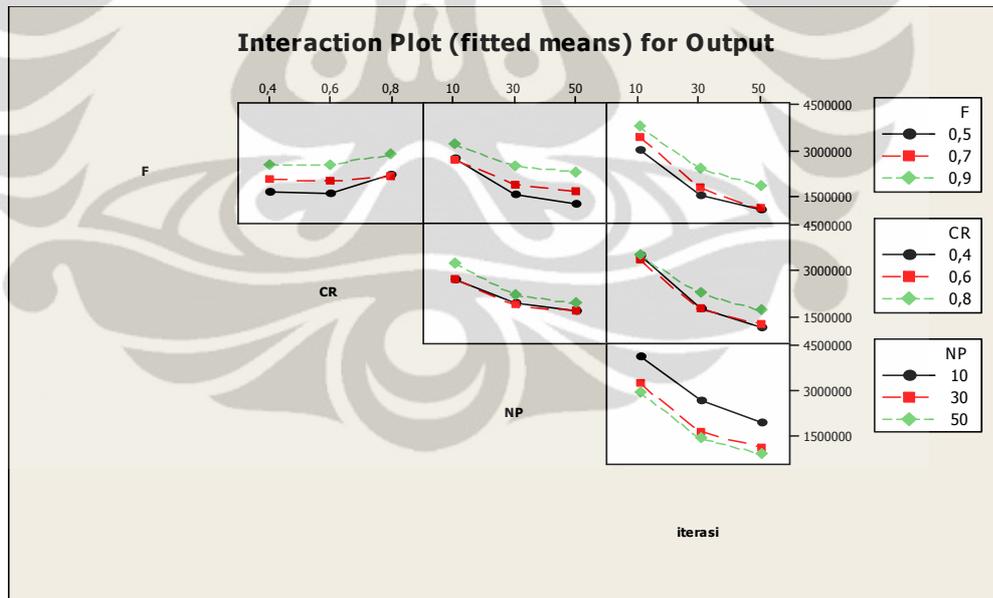
Source	DF	Seq SS	Adj SS	Adj MS	F	P
F	2	4,86673E+13	4,86673E+13	2,43336E+13	193,20	0,000
CR	2	1,16100E+13	1,16100E+13	5,80498E+12	46,09	0,000
NP	2	1,01713E+14	1,01713E+14	5,08564E+13	403,79	0,000
iterasi	2	3,26977E+14	3,26977E+14	1,63489E+14	1298,07	0,000
F*CR	4	3,99436E+12	3,99436E+12	9,98590E+11	7,93	0,000
F*NP	4	4,99141E+12	4,99141E+12	1,24785E+12	9,91	0,000
F*iterasi	4	2,67043E+12	2,67043E+12	6,67607E+11	5,30	0,000
CR*NP	4	1,41373E+12	1,41373E+12	3,53432E+11	2,81	0,026
CR*iterasi	4	5,22017E+12	5,22017E+12	1,30504E+12	10,36	0,000
NP*iterasi	4	7,04479E+11	7,04479E+11	1,76120E+11	1,40	0,234
F*CR*NP	8	7,29181E+11	7,29181E+11	91147603704	0,72	0,671
F*CR*iterasi	8	6,42773E+12	6,42773E+12	8,03466E+11	6,38	0,000
F*NP*iterasi	8	1,38769E+12	1,38769E+12	1,73462E+11	1,38	0,205
CR*NP*iterasi	8	1,68504E+12	1,68504E+12	2,10630E+11	1,67	0,104
F*CR*NP*iterasi	16	1,77395E+12	1,77395E+12	1,10872E+11	0,88	0,593
Error	324	4,08070E+13	4,08070E+13	1,25948E+11		
Total	404	5,60772E+14				

S = 354891 R-Sq = 92,72% R-Sq(adj) = 90,93%

Dari tabel tersebut dapat dilihat bahwa keempat parameter yang digunakan dalam Algoritma DE memberikan pengaruh yang signifikan terhadap *output* total biaya keterlambatan yang dihasilkan. Hal ini ditunjukkan oleh *p-value* masing-

masing parameter yang nilainya di bawah 0,05. Selain itu, pengaruh parameter juga dapat dinilai dengan melihat nilai F_{Anova} . Nilai F_{Anova} menggambarkan pengaruh suatu faktor terhadap *output* dibandingkan dengan pengaruh faktor lainnya. Semakin besar nilai F_{Anova} , maka pengaruh faktor tersebut terhadap *output* yang dihasilkan lebih besar dibandingkan dengan faktor lainnya. Dengan demikian, berdasarkan nilai F_{Anova} tersebut dapat diketahui bahwa parameter yang memiliki pengaruh yang paling besar adalah jumlah iterasi.

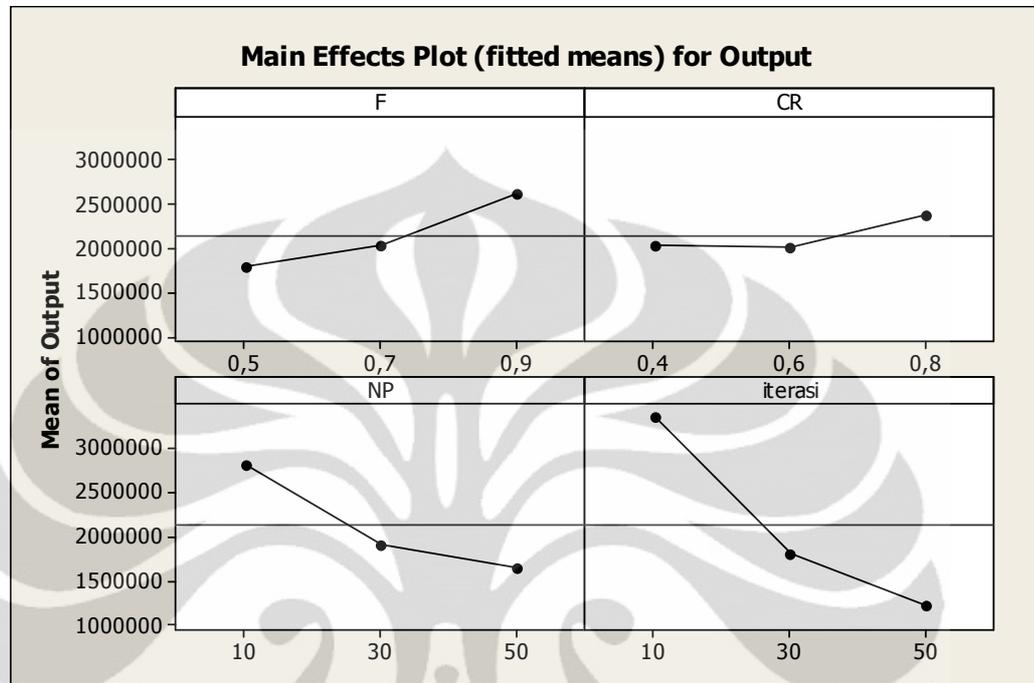
Selain itu, melalui tabel 4.3. dapat pula pengaruh interaksi antar faktor terhadap *output*. Berdasarkan *p-value*, interaksi parameter yang berpengaruh signifikan terhadap total biaya keterlambatan adalah interaksi F dengan jumlah iterasi, F dengan CR, F dengan ukuran populasi (NP), CR dengan jumlah iterasi, serta CR dengan ukuran populasi (NP). Artinya, jika terjadi perubahan pada nilai kedua parameter dalam interaksi tersebut, akan terjadi perubahan yang signifikan terhadap *output* yang dihasilkan. Interaksi antar 2 faktor dijelaskan melalui grafik berikut ini secara lebih detail.



Gambar 4.3. Grafik Interaksi Antar Parameter

Garis yang miring pada grafik menunjukkan bahwa perubahan pada salah satu parameter memberikan pengaruh yang signifikan terhadap total biaya keterlambatan. Sebagai contoh, nilai F baik dari level rendah, sedang, maupun tinggi, jika dikombinasikan dengan nilai jumlah iterasi dari semua level (10; 30; 50) menghasilkan perubahan yang signifikan terhadap total biaya keterlambatan.

Gambar 4.4. menunjukkan pengaruh tiap faktor terhadap *output* secara lebih detail tanpa melibatkan interaksi di antaranya.



Gambar 4.4. Pengaruh Parameter Terhadap *Output*

Dari grafik tersebut dapat dilihat bahwa untuk faktor permutasi (F), nilai yang semakin kecil akan memberikan *output* yang semakin baik, yaitu total biaya keterlambatan yang semakin kecil. Untuk operator pindah silang (CR), nilai yang menghasilkan total biaya keterlambatan terbaik adalah yang bernilai sedang, yaitu 0,6. Sedangkan parameter lainnya seperti ukuran populasi (NP) dan jumlah iterasi memiliki hubungan yang berbanding terbalik terhadap total biaya keterlambatan yang dihasilkan. Total biaya keterlambatan yang semakin kecil justru dihasilkan dengan nilai NP dan jumlah iterasi yang semakin besar. Dengan demikian, diperoleh kombinasi parameter terbaik sebagai berikut:

Faktor permutasi yang kecil

Pada teorinya, nilai F sebaiknya berada di atas level tertentu untuk mencegah terjadinya konvergensi prematur. Akan tetapi, jika nilai F terlalu besar, pencarian solusi akan menjadi lamban. Nilai F yang besar (lebih besar dari 1) membuat pergerakan yang cukup besar dari vektor

target ke populasi mutan. Jika nilai CR yang digunakan tinggi, maka sebagian besar gen populasi target akan berasal dari populasi mutan. Akibatnya, solusi urutan yang dihasilkan populasi trial tidak akan berbeda signifikan dari urutan yang dihasilkan populasi target. Untuk populasi dengan ukuran yang sangat besar, diperlukan nilai F yang lebih kecil dengan alasan semakin banyaknya ukuran populasi, semakin banyak pula individu yang mengisi ruang pencarian. Oleh sebab itu, diperlukan amplitudo pergerakan yang lebih sedikit agar pencarian solusi menjadi lebih efektif. Karena ukuran populasi yang digunakan relatif besar, maka nilai F yang digunakan sesuai dengan hasil DOE, yaitu yang bernilai kecil (0,5).

Operator pindah silang yang sedang

Hasil DOE menunjukkan bahwa nilai CR yang memberikan hasil terbaik adalah yang bernilai sedang. Hal ini sesuai dengan teori, di mana nilai CR yang terlalu kecil memperbesar kemungkinan sebagian besar gen populasi trial berasal dari populasi target. Hal ini dapat membuat pencarian solusi menjadi lamban karena urutan pada populasi trial yang dihasilkan tidak jauh berbeda dari urutan yang dihasilkan populasi target. Nilai CR yang relatif besar akan mempercepat terjadinya konvergensi atau pencapaian solusi optimal. Akan tetapi, ada kondisi di mana nilai CR yang terlalu tinggi (misalnya $CR = 1$) tidak baik karena membuat DE menjadi tidak tangguh. Sebagai contoh, jika nilai F yang digunakan terlalu besar, di mana vektor target mengalami pergerakan cukup besar pada populasi mutan. Jika nilai CR yang digunakan tinggi, akan muncul lebih banyak bilangan acak yang nilainya di bawah CR sehingga sebagian besar gen populasi trial berasal dari populasi mutan. Akibatnya urutan yang dihasilkan populasi trial tidak jauh berbeda dari urutan populasi target. Keadaan ini juga membuat pencarian solusi menjadi lambat. Oleh karena itu, sebaiknya nilai CR diturunkan agar DE lebih tangguh. Sesuai dengan hasil DOE, nilai CR yang digunakan sebesar 0,6.

Ukuran populasi yang besar

Pada DOE, ukuran populasi yang besar terbukti menghasilkan total biaya keterlambatan yang semakin kecil. Ukuran populasi yang semakin besar memungkinkan munculnya urutan pengerjaan *job* yang lebih bervariasi pula. Hal ini mendorong pencarian solusi yang mendekati optimal karena banyaknya pilihan solusi yang tersedia. Pada DOE yang telah dilakukan, angka pada ukuran populasi yang digunakan sangat kecil dengan tujuan mempersingkat waktu komputasi percobaan. Pada saat menjalankan program pencarian solusi penjadwalan yang sebenarnya, perlu dilakukan penyesuaian ukuran populasi agar pencarian solusi lebih optimal. Oleh karena itu, digunakan ukuran populasi sebesar sepuluh kali dimensi (10D) dengan pertimbangan bahwa ukuran populasi tersebut cukup efektif dalam memberikan solusi penjadwalan yang mendekati optimal.

Jumlah iterasi yang besar

Hasil percobaan pada DOE menunjukkan bahwa semakin besar jumlah iterasi, semakin besar pula kemungkinan memperoleh solusi penjadwalan yang mendekati optimal. Hal ini disebabkan semakin banyaknya kesempatan bagi program untuk melakukan permutasi dan penyilangan individu dengan solusi terbaik yang dihasilkan oleh iterasi sebelumnya agar dapat menghasilkan individu-individu dengan solusi yang lebih baik lagi untuk iterasi selanjutnya. Proses ini akan terus berlanjut sampai kriteria terminasi yang ditentukan. Akan tetapi, sama halnya dengan ukuran populasi, jumlah iterasi akan sangat mempengaruhi waktu komputasi. Jumlah iterasi yang terlalu banyak dapat membuat waktu komputasi menjadi sangat lama. Oleh karena itu, untuk pencarian solusi penjadwalan sebenarnya pada program, dilakukan penyesuaian jumlah iterasi terhadap waktu komputasi, yaitu sebanyak 500 kali iterasi dengan waktu komputasi sebesar 3121,7848 detik.

4.5.2. Analisis Hasil

Perbandingan solusi penjadwalan *Earliest Due Date* perusahaan dengan usulan penjadwalan berdasarkan Algoritma DE dapat dilihat pada tabel 4.37. Sesuai dengan fungsi tujuan model program yang telah dibuat, solusi penjadwalan yang diperoleh melalui penerapan Algoritma DE dapat mengurangi total biaya keterlambatan.

Tabel 4.37. Perbandingan Jadwal Awal dan Usulan Jadwal

	Penjadwalan Perusahaan												Usulan Penjadwalan											
	1	2	3	4	5	6	7	8	9	10	11	12	1	4	2	5	3	7	6	8	9	10	11	12
Urutan Pengerjaan	13	14	15	16	17	18	19	20	21	22	23	24	33	13	16	15	31	30	14	18	19	17	37	22
	25	26	27	28	29	30	31	32	33	34	35	36	68	51	20	24	60	23	27	63	46	29	67	28
	37	38	39	40	41	42	43	44	45	46	47	48	21	65	26	35	25	38	34	36	40	32	49	57
	49	50	51	52	53	54	55	56	57	58	59	60	41	58	39	50	64	42	43	56	69	44	48	72
	61	62	63	64	65	66	67	68	69	70	71	72	47	59	45	53	52	70	66	71	61	54	55	62
Jumlah Pesanan Yang Terlambat	3												2											
Total Waktu Keterlambatan	285												252											
Total Biaya Keterlambatan	23590												17640											
Makespan	12735												12544											

Penjadwalan perusahaan menghasilkan total biaya keterlambatan sebesar 23590 menit sedangkan usulan penjadwalan hasil penerapan Algoritma DE dapat menghasilkan total biaya keterlambatan yang lebih kecil, yaitu sebesar 17640 menit. Usulan penjadwalan baru tidak hanya menentukan prioritas pengerjaan *job* berdasarkan *due date*, tetapi juga turut memperhitungkan bobot pinalti dari masing-masing *job* yang dikerjakan. Lain halnya dengan metode *Earliest Due Date* di mana urutan pengerjaan *job* hanya ditentukan berdasarkan *due date* dari *job* yang bersangkutan. Melalui penerapan Algoritma DE, total biaya keterlambatan berkurang sebesar 25,22% berdasarkan perhitungan berikut:

$$\text{hasil awal} = 23590$$

$$\text{hasil akhir} = 17640$$

$$\begin{aligned} \text{perubahan hasil} &= \text{hasil awal} - \text{hasil akhir} & (4.4) \\ &= 23590 - 17640 = 5950 \end{aligned}$$

$$\begin{aligned} \text{persentase perubahan} &= (\text{perubahan hasil} : \text{hasil awal}) \times 100\% & (4.5) \\ &= (5950 : 23590) \times 100\% = 25,22\% \end{aligned}$$

yang ada. Untuk itu, dilakukan percobaan peningkatan kapasitas produksi dengan menambah jumlah mesin atau stasiun kerja. Percobaan dilakukan sebanyak tujuh kali dengan penambahan mesin pada bagian serut saja, bagian potong saja, bagian perakitan saja, bagian serut dan potong, bagian serut dan perakitan, bagian potong dan perakitan, dan kombinasi ketiga stasiun kerja. Hasilnya dituangkan dalam tabel 4.39.

Tabel 4.39. Hasil Peningkatan Kapasitas Stasiun Kerja

Mesin	Makespan	Jumlah Keterlambatan	Total Keterlambatan	Total Biaya Keterlambatan
Serut	10077	1	48	3360
Potong	12477	2	203	14730
Assembly	12364	3	349	26370
Serut dan Potong	6409	0	0	0
Serut dan Assembly	10077	1	48	3360
Potong dan Assembly	12455	2	155	10890
Serut, Potong, dan Assembly	6409	0	0	0

Untuk mesin serut, ditambahkan sejumlah satu mesin untuk masing-masing komponen. Jadi total mesin serut yang ditambahkan ada dua. Mesin potong ditambah menjadi empat mesin sedangkan untuk bagian perakitan ditambah satu stasiun kerja perakitan. Hasil di atas menunjukkan bahwa penambahan mesin pada salah satu stasiun kerja saja tidak akan memberikan pengaruh yang signifikan terhadap waktu penyelesaian. Penambahan mesin sebaiknya dilakukan pada mesin penyerutan dan pemotongan karena hasil menunjukkan perubahan waktu penyelesaian yang cukup besar. Selain itu, penambahan kapasitas pada bagian penyerutan dan pemotongan dapat menghilangkan keterlambatan. Penambahan kapasitas pada stasiun perakitan tidak memberikan dampak yang signifikan terhadap waktu penyelesaian sekalipun disertai dengan penambahan kapasitas stasiun lain. Oleh karena itu, penambahan cukup dilakukan pada stasiun penyerutan dan pemotongan. Dengan waktu penyelesaian yang berkurang sebanyak 48,90% $((12544-6409)/12544) \times 100\%$, perusahaan dapat memproduksi hampir dua kali lebih banyak dari kapasitas semula jika melakukan penambahan mesin pada bagian serut dan potong. Peningkatan kapasitas tersebut dapat dihitung sebagai berikut:

Output awal = 1

Output akhir setelah penambahan kapasitas = $\frac{1}{1-0,489} = 1,95$

Peningkatan output = 95%

Selain untuk mengurangi keterlambatan, penambahan mesin atau stasiun kerja pun dapat dilakukan sehubungan dengan adanya rencana perusahaan untuk meningkatkan kapasitas produksinya.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil analisis dari permasalahan penjadwalan *job-shop* pada perusahaan menggunakan algoritma Differential Evolution dengan bantuan bahasa pemrograman MATLAB 7.8.0 (R2009a) diperoleh kesimpulan sebagai berikut:

1. Jadwal perusahaan memiliki total biaya keterlambatan seluruh *job* sebesar 23590 menit.
2. Usulan jadwal hasil program dengan metode Algoritma *Differential Evolution* memiliki total biaya keterlambatan seluruh *job* sebesar 17640 menit.
3. Total biaya keterlambatan untuk usulan jadwal memiliki penurunan dibanding jadwal perusahaan, yaitu sebesar 25,22%.
4. Selain fungsi tujuan (total biaya keterlambatan), fungsi yang lain juga mengalami penurunan, yaitu:
 - Jumlah *job* yang terlambat mengalami penurunan sebesar 33%.
 - Total keterlambatan seluruh *job* mengalami penurunan sebesar 11,58%.
 - Makespan (total waktu penyelesaian seluruh *job*) mengalami penurunan sebesar 1,50%.
5. Waktu *run* program rata-rata untuk memperoleh usulan jadwal yaitu adalah 3121,7848 detik.
6. Penambahan mesin serut sebanyak satu mesin untuk masing-masing jenis penyerut (total menjadi 4 mesin serut) serta mesin potong sebanyak dua buah dapat mengurangi waktu penyelesaian hingga 48,90% serta menghilangkan keterlambatan. Selain untuk mengurangi keterlambatan, penambahan mesin juga dapat membantu perusahaan meningkatkan kapasitas produksi hingga sebesar 95%.

5.2. Saran

Pada penelitian ini, masih terdapat kekurangan serta keterbatasan. Oleh karena itu, beberapa saran dapat dipertimbangkan sebagai bahan penyempurnaan

untuk penelitian selanjutnya. Pertama, selain mencari solusi penjadwalan optimal melalui penerapan Algoritma DE, selanjutnya dapat dilakukan perbandingan hasil penerapan Algoritma DE tersebut dengan hasil penerapan metode lainnya seperti *Tabu Search*, Algoritma Genetika, atau *Simulated Annealing*. Melalui perbandingan tersebut dapat dilihat metode yang dapat menghasilkan solusi penjadwalan yang lebih baik.

Kedua, periode pemesanan yang menjadi objek penelitian ini adalah periode pemesanan bulan April 2011. Penjadwalan perusahaan untuk bulan April 2011 tidak begitu banyak menghasilkan keterlambatan sehingga peningkatan yang diperoleh melalui usulan penjadwalan berdasarkan penerapan Algoritma DE pun tidak terlalu signifikan. Pada penelitian selanjutnya, penambahan periode untuk objek penelitian dapat dilakukan untuk memperluas ruang lingkup penelitian. Penambahan periode yang diiringi penambahan jumlah *job* memungkinkan munculnya keterlambatan yang lebih banyak. Dengan demikian, dapat dilihat apakah Algoritma DE dapat menghasilkan usulan penjadwalan yang memberikan peningkatan yang signifikan.

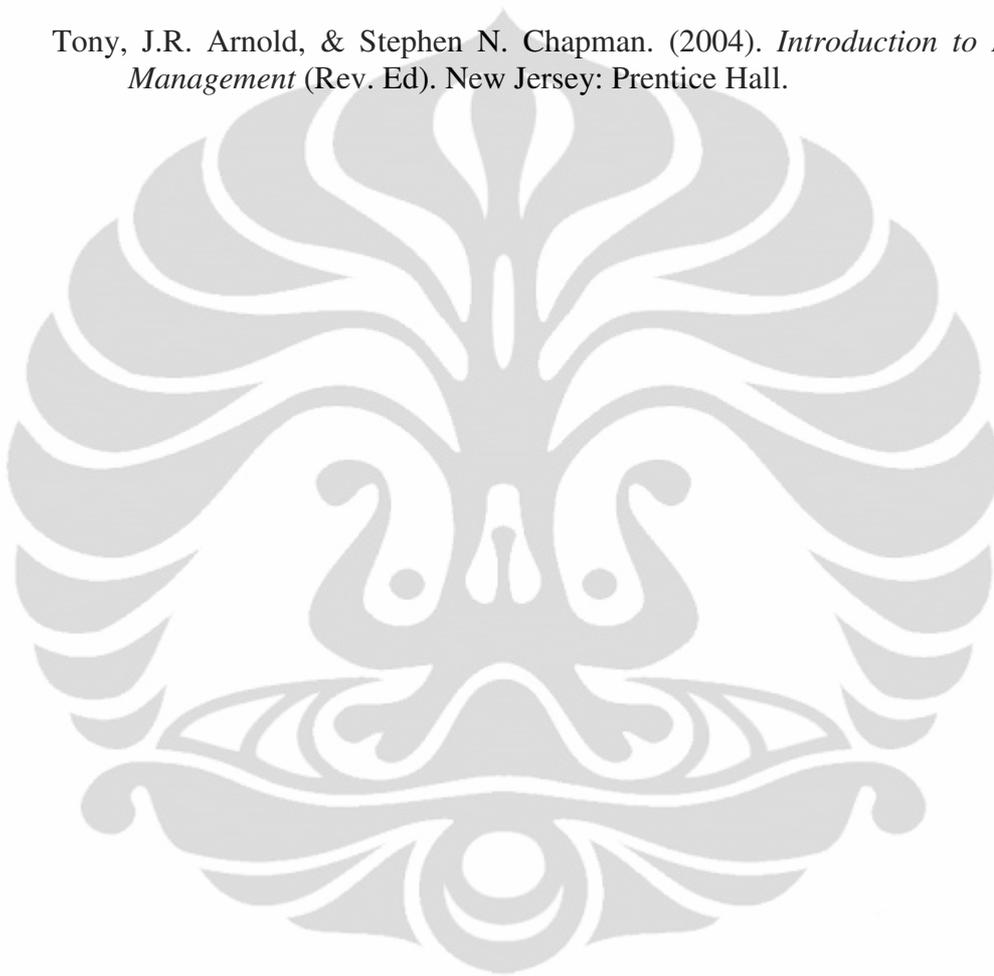
Terakhir, pembelajaran lebih dalam mengenai kode pemrograman sebaiknya dilakukan agar penelitian selanjutnya dapat menampilkan suatu *interface* yang *user friendly* yang memudahkan perusahaan dalam melakukan *input* data pemesanan untuk mencari solusi penjadwalan di periode lainnya.

DAFTAR PUSTAKA

- Adam & Elbert. (1992). *Production and Operations Management*. New Jersey: Prentice Hall
- Baker, Kenneth R. (2001). *Elements of Sequencing and Scheduling*. Hanover: Author.
- Bank Indonesia. (n.d.). *Prospek Industri Perabot dan Barang Jadi Kayu*. 18 Januari 2011.
http://www.bi.go.id/sipuk/id/lm/kayu_olahan/pendahuluan.asp
- Chao, Xiuli & Pinedo, Michael. (1999). *Operations scheduling with applications in manufacturing and services*. Boston: Irwin McGraw-Hill
- Developing Algorithms and Applications*. (n.d.). 15 Mei 2011.
<http://www.mathworks.com/products/MATLAB/description2.html>
- Everett & Robert. (1999). *Production and Operation Managemen*. New Jersey: Prentice Hall.
- Fattahi, P. Et al. (2007). Mathematical modeling and heuristic approaches to flexible *job shop* scheduling problems. *Jurnal Intell. Manuf.*, 18, 331-342.
- Fattahi, Parviz, et al. (2009). Flexible *job shop* scheduling with overlapping in operations. *Applied Mathematical Modeling*, 33, 3076-3087.
- Friman, Mark A. (2002). *Quality and Process Improvement*. USA.
- J. Rönkkönen, S. Kukkonen, & K. V. Price. (2005). Real-parameter Optimization with Differential Evolution. *IEEE Congress on Evolutionary Computation*. p.^pp. 506-513.
- Karaboga, Dervis & Okdem, Selcuk. (2004). A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm. *Turk J. Elec Engin*, vol. 12, no. 1, 53-60.
- Kuliah V: Kemasan Kayu*. (n.d.). 18 Januari 2011.
http://ocw.usu.ac.id/course/.../thp_407_slide_kuliah_4_-_kemasan_kayu.pdf
- Lopez Cruz, Willigenburg, L.G. Van, I.L., & Straten, G. Van. (2001). Parameter Control Strategy in Differential Evolution Algorithm for Optimal Control. *Proceedings of the IASTED International Conference Artificial Intelligence and Soft Computing*, Cancun, Mexico, 211-216.
- Montgomery Douglas C. (1996). *Design and Analysis of Experiments*. New York: John Wiley & Sons.

- Nearchou, Andreas C. (2008). A Differential Evolution for Common Due Date Early/Tardy Job Scheduling Problem. *Computers and Operations Research*, vol. 35, 1329-1343.
- NIST (n.d.). *Engineering Statistics Handbook : Anderson-Darling Test*. 10 Juni 2011.
<http://www.itl.nist.gov/div898/handbook/eda/section3/eda35e.htm>
- Noman, Nasimul & Iba, Hitoshi. (2005). Enhancing Differential Evolution Performance with Local Search for High Dimensional Function Optimization. *GECCO'05*, Washington, DC.
- Ozguven, Cemal. Et al. (2010). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modeling*, 34, 1539-1548.
- Price, K., & Storn, R. (1997). Differential Evolution. *Dr. Dobb's Journal*, 18–24.
- Price, K.V. (1999). An Introduction to Differential Evolution. In: *Corne D, Dorigo M, Glover F (eds.), New ideas in optimization*. McGraw-Hill, London, 79–108.
- Price, K.V., & Storn, R.M. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *Journal of Global Optimization* 11, 4, 341-359.
- Price, K.V., Storn, R.M., & Lampinen, J.A. (2005). *Differential Evolution: A Practical Approach to Global Optimization-Natural Computing Series*. Berlin: Springer-Verlag.
- Routroy, Srikanta & Kodali, Rambabu (2005), Differential Evolution Algorithm for Supply Chain Inventory Planning, *Journal of Manufacturing Technology Management*, vol. 16, no. 1, 7-17.
- R. Gämperle, S. D. Müller and P. Koumoutsakos. (2002). "A parameter study for differential evolution," *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. WSEAS Press, Interlaken, Switzerland, pp. 293-298.
- Skim Audit Badan Karantina Pertanian. (14 Mei 2008). *Daftar Perusahaan Kemasan Kayu*. 18 Januari 2011.
<http://www.docstoc.com/docs/20580879/DAFTAR-PERUSAHAAN-KEMASAN-KAYU-SKIM-AUDIT-BADAN-KARANTINA>
- Skim Audit Badan Karantina Pertanian. (27 Agustus 2010). *ISPM 15 Kemasan Kayu*. 18 Januari 2011.
<http://karantina.deptan.go.id/sab/index.php?link=008&id=7>
- Stevenson, William J. (2008). *Operations Management*. New York: McGraw-Hill Irwin

- Takeshi, & Collin. (1998). Solving the C_{sum} Permutation *Flow Shop* Scheduling Problems by Genetic Local Search. *IEEE International; Conference on Evolutionary Computation*, 1.
- Tasgetiren, M. Fatih, et al. (2004). Particle Swarm Optimization and Differential Evolution Algorithm for *Job Shop* Scheduling Problem. *Proceedings of the 4th International Symposium on Intelligent Manufacturing System (IMS2004)*, Sakarya, Turkey, 1-18.
- Tony, J.R. Arnold, & Stephen N. Chapman. (2004). *Introduction to Materials Management* (Rev. Ed). New Jersey: Prentice Hall.





LAMPIRAN 1. Uji Normalitas dan Kecukupan Data Primer

Balok 5 x 7 x 108

No	Siku	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	35,12	1233,414	35,321	0,201	0,040	52,961	17,681
2	48,93	2394,145	35,321	13,609	185,207	52,961	17,681
3	40,20	1616,040	35,321	4,879	23,806	52,961	17,681
4	32,46	1053,652	35,321	2,861	8,185	52,961	17,681
5	33,28	1107,558	35,321	2,041	4,165	52,961	17,681
6	31,78	1009,968	35,321	3,541	12,538	52,961	17,681
7	28,36	804,290	35,321	6,961	48,454	52,961	17,681
8	34,97	1222,901	35,321	0,351	0,123	52,961	17,681
9	29,29	857,904	35,321	6,031	36,372	52,961	17,681
10	33,85	1145,823	35,321	1,471	2,164	52,961	17,681
11	40,29	1623,284	35,321	4,969	24,692	52,961	17,681

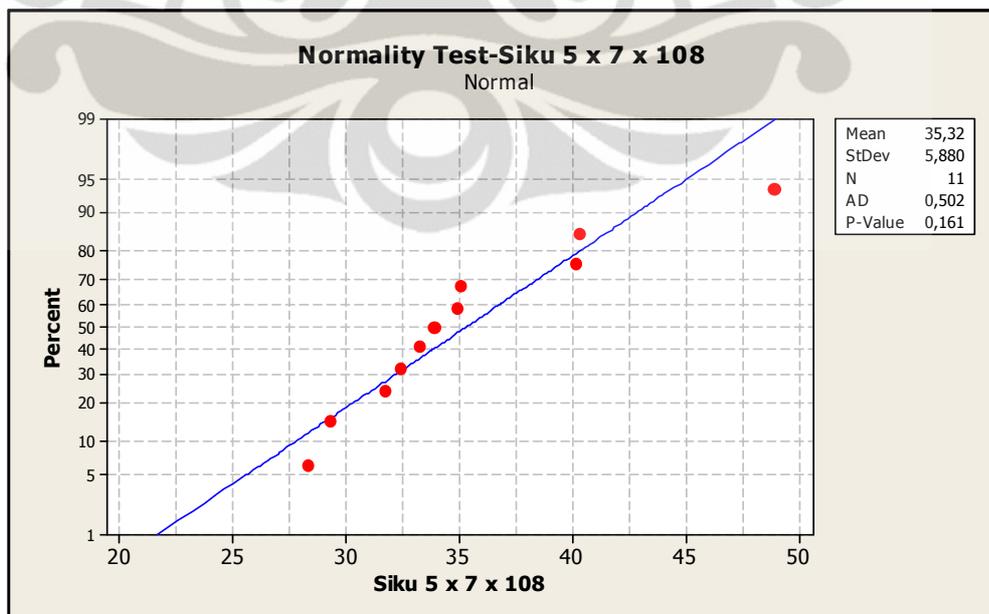
sum **388,53** **14.068,98** **46,915** **345,746**
 sum² 150955,561
 Average 35,321
 35,33
 n 11

UJI SERAGAM

Rata2= 35,321
 σ = 5,88001778
 BKA = 52,9609624
 BKB = 17,6808558 17,680856

UJI KECUKUPAN DATA

N' = 11
 Karena data yang tersedia ada 11, maka data dapat dikatakan cukup



(lanjutan)

Balok 5 x 7 x 108

No	Serut	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	39,73	1578,473	34,635	5,095	25,964	41,694	27,575
2	33,31	1109,556	34,635	1,325	1,754	41,694	27,575
3	34,32	1177,862	34,635	0,315	0,099	41,694	27,575
4	35,77	1279,493	34,635	1,135	1,289	41,694	27,575
5	32,26	1040,708	34,635	2,375	5,638	41,694	27,575
6	32,80	1075,840	34,635	1,835	3,366	41,694	27,575
7	32,48	1054,950	34,635	2,155	4,642	41,694	27,575
8	33,49	1121,580	34,635	1,145	1,310	41,694	27,575
9	34,09	1162,128	34,635	0,545	0,297	41,694	27,575
10	37,95	1440,203	34,635	3,315	10,992	41,694	27,575
11	34,78	1209,648	34,635	0,145	0,021	41,694	27,575

sum **380,98** **13.250,44** **19,384** **55,372**
sum² 145145,760
Average 34,635
34,64
n 11

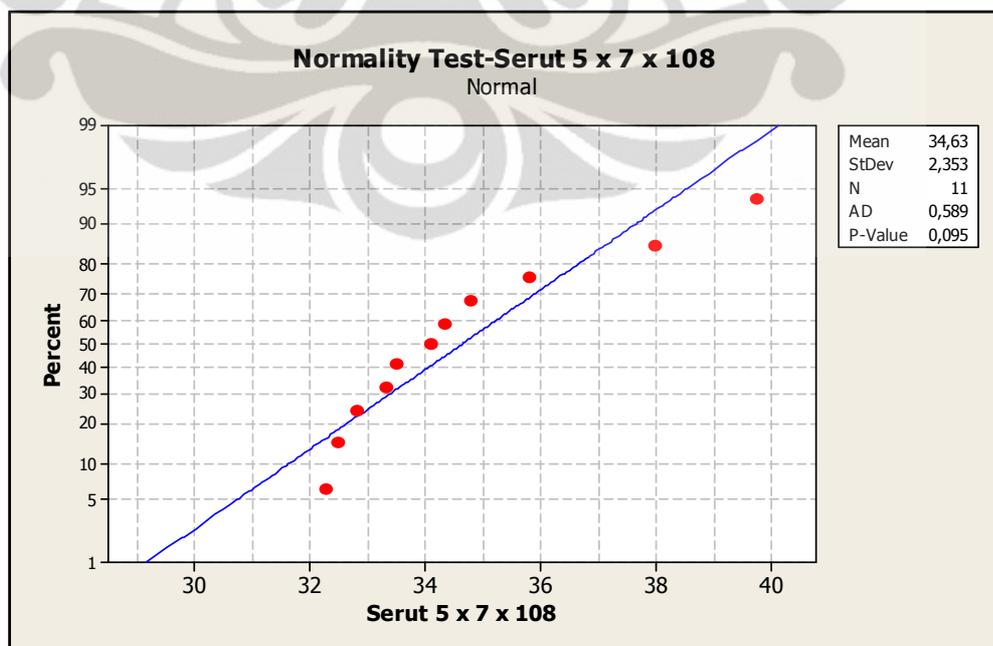
UJI SERAGAM

Rata2= 34,635
 σ = 2,3531314
BKA = 41,69394
BKB = 27,575151 27,575151

UJI KECUKUPAN DATA

N' = 2

Karena data yang tersedia ada 11, maka data dapat dikatakan cukup



(lanjutan)

Balok 5 x 7 x 108

No	Potong	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	22,98	528,080	24,011	1,031	1,063	28,707	19,315
2	23,38	546,624	24,011	0,631	0,398	28,707	19,315
3	27,55	759,003	24,011	3,539	12,525	28,707	19,315
4	23,20	538,240	24,011	0,811	0,658	28,707	19,315
5	22,14	490,180	24,011	1,871	3,500	28,707	19,315
6	22,75	517,563	24,011	1,261	1,590	28,707	19,315
7	22,97	527,621	24,011	1,041	1,083	28,707	19,315
8	25,62	656,384	24,011	1,609	2,589	28,707	19,315
9	24,75	612,563	24,011	0,739	0,546	28,707	19,315
10	24,75	612,563	24,011	0,739	0,546	28,707	19,315
11	24,03	577,441	24,011	0,019	0,000	28,707	19,315

sum **264,12** **6.366,26** **13,291** **24,499**
sum² 69759,374
Average 24,011
 24,02
n 11

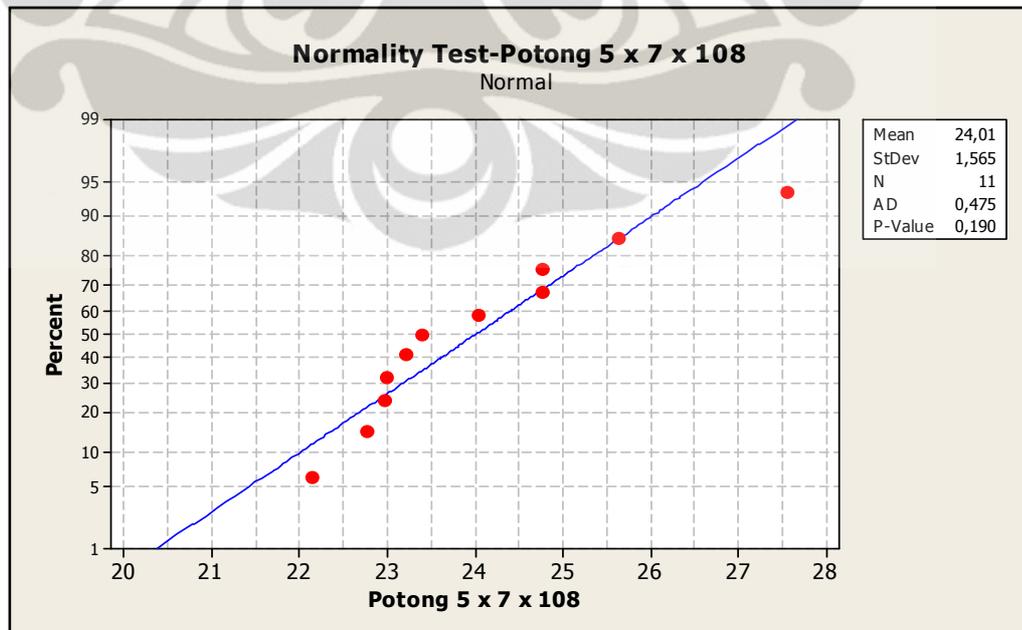
UJI SERAGAM

Rata2= 24,011
 σ = 1,5652249
BKA = 28,706584
BKB = 19,315234 19,315234

UJI KECUKUPAN DATA

N' = 2

Karena data yang tersedia ada 11, maka data dapat dikatakan cukup



(lanjutan)

Balok Kubus

No	Siku	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	44,73	2000,773	39,450	5,280	27,878	50,415	28,485
2	41,63	1733,057	39,450	2,180	4,752	50,415	28,485
3	38,13	1453,897	39,450	1,320	1,742	50,415	28,485
4	36,71	1347,624	39,450	2,740	7,508	50,415	28,485
5	36,05	1299,603	39,450	3,400	11,560	50,415	28,485

sum **197,25** **7.834,95** **14,92** **53,44**

sum² 38907,563

Average 39,450

39,45

n 5

UJI SERAGAM

Rata2= 39,450

σ = 3,6551607

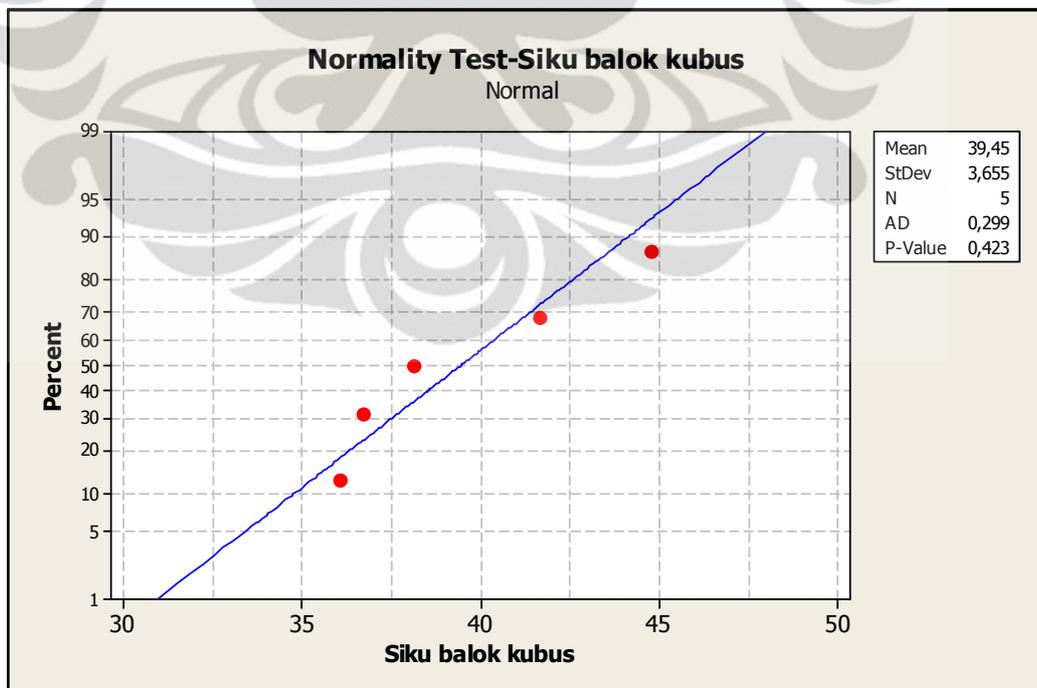
BKA = 50,415482

BKB = 28,484518 28,48452

UJI KECUKUPAN DATA

N' = 3

Karena data yang tersedia ada 5, maka data dapat dikatakan cukup



(lanjutan)

Balok Kubus

No	Serut	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	24,07	579,365	22,904	1,166	1,360	26,551	19,257
2	21,67	469,589	22,904	1,234	1,523	26,551	19,257
3	21,69	470,456	22,904	1,214	1,474	26,551	19,257
4	22,94	526,244	22,904	0,036	0,001	26,551	19,257
5	24,15	583,223	22,904	1,246	1,553	26,551	19,257

sum **114,52** **2.628,88** **4,90** **5,91**

sum² 13114,830

Average 22,904

22,91

n 5

UJI SERAGAM

Rata2= 22,904

σ = 1,21551635

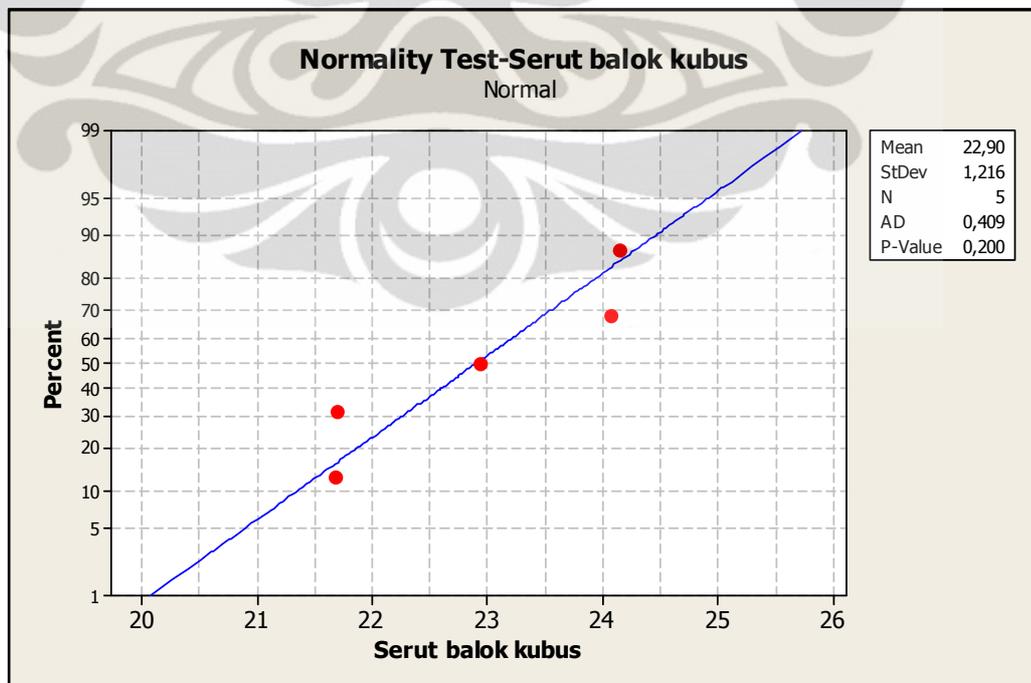
BKA = 26,5505491

BKB = 19,2574509 19,25745

UJI KECUKUPAN DATA

N' = 1

Karena data yang tersedia ada 5, maka data dapat dikatakan cukup



(lanjutan)

Balok Kubus

No	Potong	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	48,75	2376,563	42,272	6,478	41,964	54,218	30,326
2	42,50	1806,250	42,272	0,228	0,052	54,218	30,326
3	41,87	1753,097	42,272	0,402	0,162	54,218	30,326
4	38,29	1466,124	42,272	3,982	15,856	54,218	30,326
5	39,95	1596,003	42,272	2,322	5,392	54,218	30,326

sum **211,36** **8.998,04** **13,41** **63,43**

sum² 44673,050

Average 42,272

 42,28

n 5

UJI SERAGAM

Rata2= 42,272

σ = 3,9820246

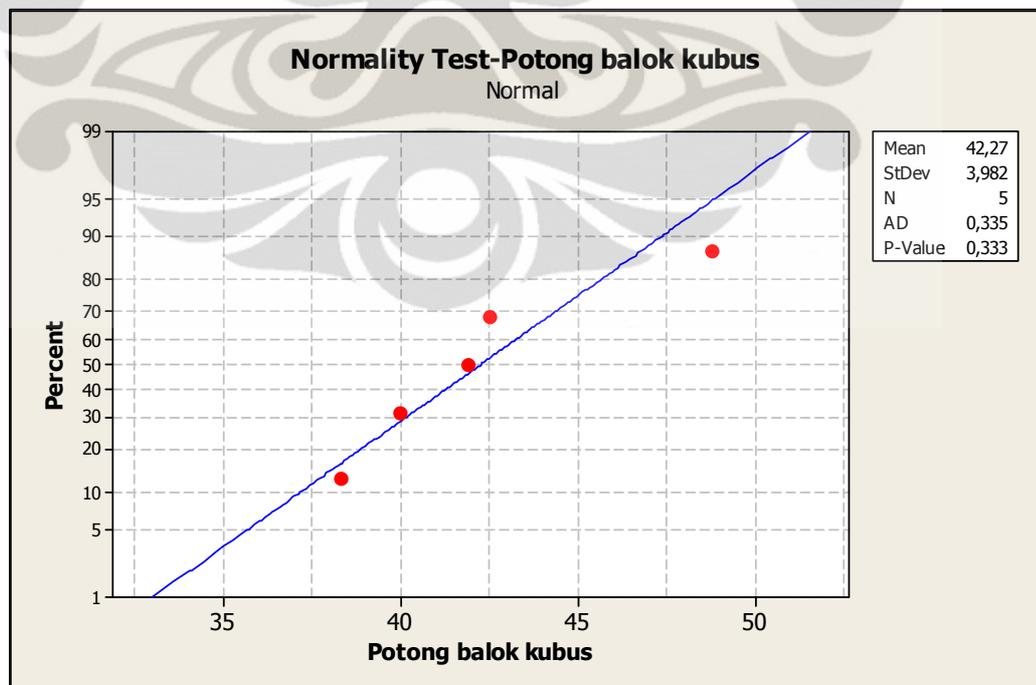
BKA = 54,218074

BKB = 30,325926 30,32593

UJI KECUKUPAN DATA

N' = 3

Karena data yang tersedia ada 5, maka data dapat dikatakan cukup



(lanjutan)

Papan

No	Serut	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	64,67	4182,209	65,238	0,568	0,322	71,181	59,294
2	65,41	4278,468	65,238	0,172	0,030	71,181	59,294
3	63,70	4057,690	65,238	1,538	2,364	71,181	59,294
4	63,44	4024,634	65,238	1,798	3,231	71,181	59,294
5	66,05	4362,603	65,238	0,812	0,660	71,181	59,294
6	65,48	4287,630	65,238	0,242	0,059	71,181	59,294
7	63,74	4062,788	65,238	1,498	2,243	71,181	59,294
8	69,47	4826,081	65,238	4,232	17,914	71,181	59,294
9	67,35	4536,023	65,238	2,112	4,463	71,181	59,294
10	65,35	4270,623	65,238	0,112	0,013	71,181	59,294
11	61,08	3730,766	65,238	4,158	17,285	71,181	59,294
12	67,11	4503,752	65,238	1,872	3,506	71,181	59,294

sum **782,85** **51.123,27** **12,49** **43,18**
sum² 612854,123
Average 65,238
 65,24
n 12

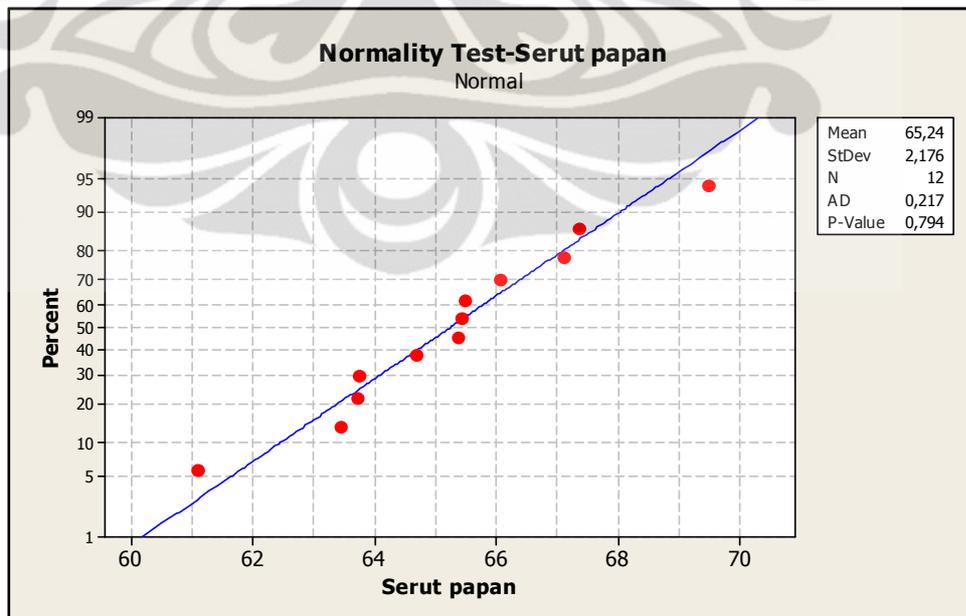
UJI SERAGAM

Rata2= 65,238
 σ = 1,981285883
BKA = 71,18135765
BKB = 59,29364235 59,29364

UJI KECUKUPAN DATA

N' = 1

Karena data yang tersedia ada 12, maka data dapat dikatakan cukup



(lanjutan)

Papan							
No	Potong	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	27,51	756,800	26,734	0,776	0,602	30,411	23,058
2	26,62	708,624	26,734	0,114	0,013	30,411	23,058
3	30,23	913,853	26,734	3,496	12,221	30,411	23,058
4	24,17	584,189	26,734	2,564	6,575	30,411	23,058
5	22,07	487,085	26,734	4,664	21,754	30,411	23,058
6	24,16	583,706	26,734	2,574	6,626	30,411	23,058
7	26,13	682,777	26,734	0,604	0,365	30,411	23,058
8	30,31	918,696	26,734	3,576	12,787	30,411	23,058
9	27,96	781,762	26,734	1,226	1,503	30,411	23,058
10	27,22	740,928	26,734	0,486	0,236	30,411	23,058
11	28,09	789,048	26,734	1,356	1,838	30,411	23,058
12	26,34	693,796	26,734	0,394	0,155	30,411	23,058

sum **320,81** **8.641,26** **7,04** **16,52**

sum² 102919,056

Average 26,734

26,74

n 12

UJI SERAGAM

Rata2= 26,734

σ = 1,22544748

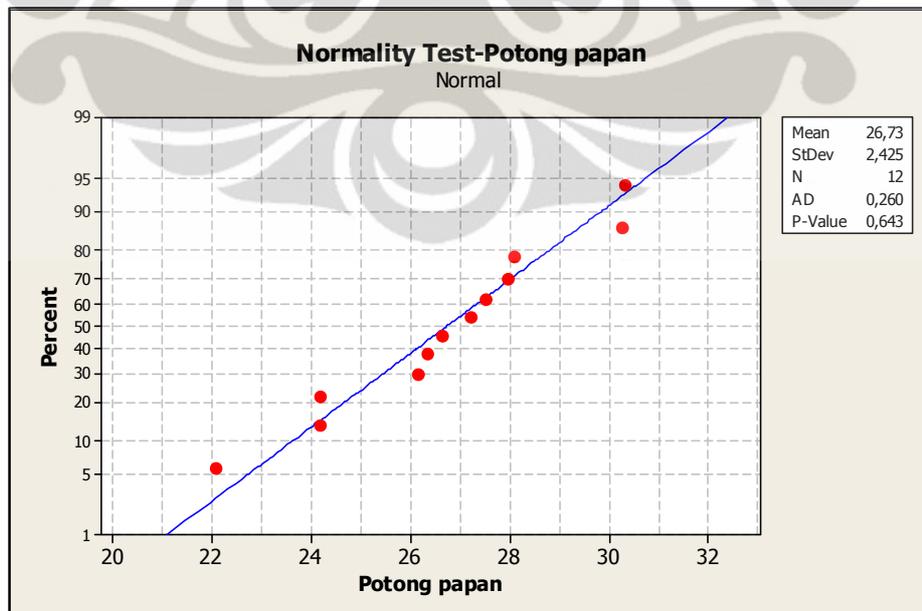
BKA = 30,4105091

BKB = 23,0578242 23,05782

UJI KECUKUPAN DATA

N' = 4

Karena data yang tersedia ada 12, maka data dapat dikatakan cukup



(lanjutan)

Papan

No	Belah	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	4,56	20,794	4,081	0,479	0,229	6,485	1,677
2	4,26	18,148	4,081	0,179	0,032	6,485	1,677
3	4,56	20,794	4,081	0,479	0,229	6,485	1,677
4	4,47	19,981	4,081	0,389	0,151	6,485	1,677
5	5,49	30,140	4,081	1,409	1,985	6,485	1,677
6	3,73	13,913	4,081	0,351	0,123	6,485	1,677
7	6,06	36,724	4,081	1,979	3,916	6,485	1,677
8	3,90	15,210	4,081	0,181	0,033	6,485	1,677
9	3,09	9,548	4,081	0,991	0,982	6,485	1,677
10	4,51	20,340	4,081	0,429	0,184	6,485	1,677
11	3,24	10,498	4,081	0,841	0,707	6,485	1,677
12	4,76	22,658	4,081	0,679	0,461	6,485	1,677
13	3,45	11,903	4,081	0,631	0,398	6,485	1,677
14	2,93	8,585	4,081	1,151	1,325	6,485	1,677
15	3,63	13,177	4,081	0,451	0,203	6,485	1,677
16	3,36	11,290	4,081	0,721	0,520	6,485	1,677
17	4,35	18,923	4,081	0,269	0,072	6,485	1,677
18	3,51	12,320	4,081	0,571	0,326	6,485	1,677
19	4,23	17,893	4,081	0,149	0,022	6,485	1,677
20	3,53	12,461	4,081	0,551	0,304	6,485	1,677

sum **81,62** **345,30** **12,88** **12,20**
sum² 6661,824
Average 4,081
 4,09
n 20

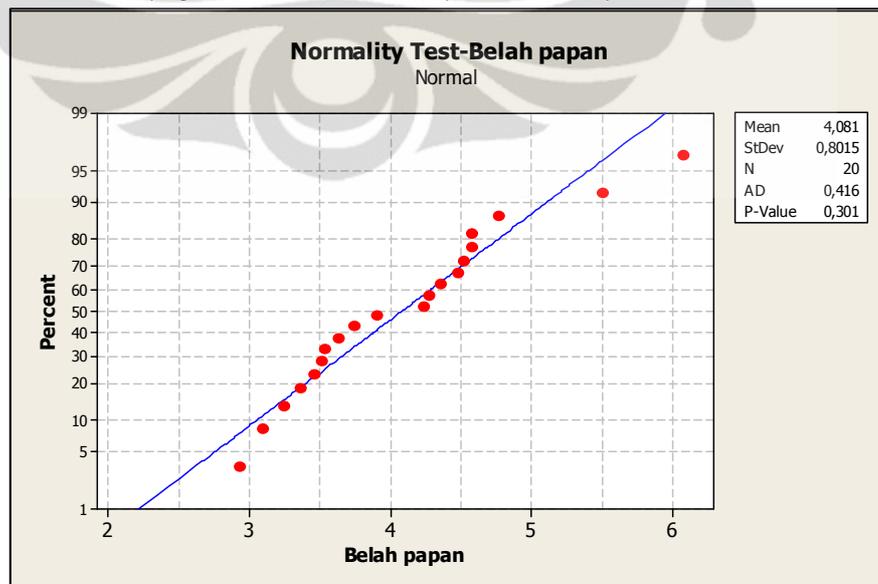
UJI SERAGAM

Rata2= 4,081
 σ = 0,80147167
BKA = 6,48541502
BKB = 1,67658498 1,676585

UJI KECUKUPAN DATA

N' = 15

Karena data yang tersedia ada 20, maka data dapat dikatakan cukup



(lanjutan)

Balok 3,5 x 9 x 78

No	Siku	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	13,88	192,654	11,896	1,984	3,936	15,970	7,822
2	11,14	124,100	11,896	0,756	0,572	15,970	7,822
3	12,68	160,782	11,896	0,784	0,615	15,970	7,822
4	11,23	126,113	11,896	0,666	0,444	15,970	7,822
5	10,55	111,303	11,896	1,346	1,812	15,970	7,822

sum **59,48** **714,95** **5,54** **7,38**

sum² 3537,870

Average 11,896

11,9

n 5

UJI SERAGAM

Rata2= 11,896

σ = 1,35809793

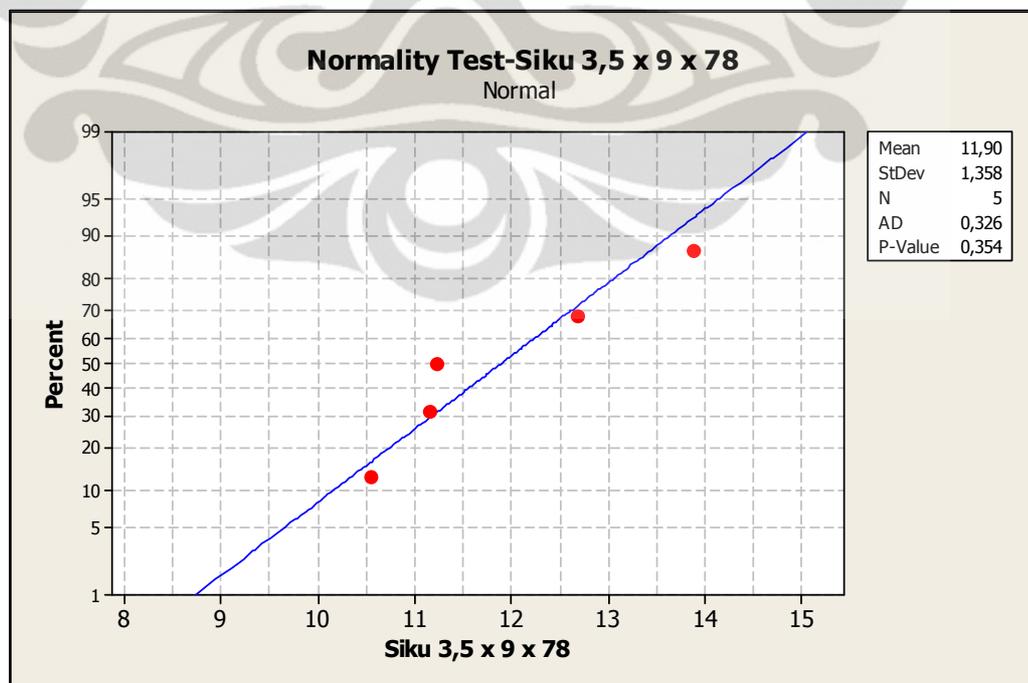
BKA = 15,9702938

BKB = 7,8217062 7,821706

UJI KECUKUPAN DATA

N' = 5

Karena data yang tersedia ada 5, maka data dapat dikatakan cukup



(lanjutan)

Balok 3,5 x 9 x 78

No	Serut	xi ²	x bar	xi-xbar	xi - x bar) ²	BKA	BKB
1	18,31	335,256	19,094	0,784	0,615	21,682	16,506
2	19,97	398,801	19,094	0,876	0,767	21,682	16,506
3	20,08	403,206	19,094	0,986	0,972	21,682	16,506
4	18,41	338,928	19,094	0,684	0,468	21,682	16,506
5	18,70	349,690	19,094	0,394	0,155	21,682	16,506

sum 95,47 1.825,88 3,72 2,98

sum² 9114,521

Average 19,094

19,1

n 5

UJI SERAGAM

Rata2= 19,094

σ = 0,862745617

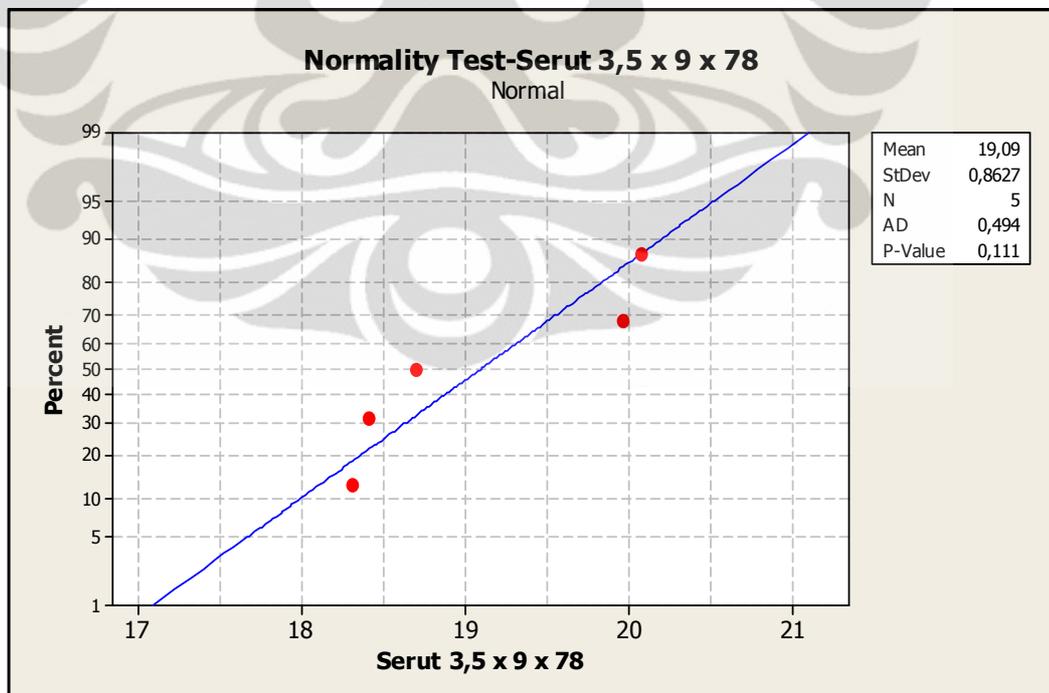
BKA = 21,68223685

BKB = 16,50576315 16,50576

UJI KECUKUPAN DATA

N' = 1

Karena data yang tersedia ada 5, maka data dapat dikatakan cukup



(lanjutan)

Balok 3,5 x 9 x 78

No	Potong	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	19,70	388,090	19,420	0,280	0,078	20,623	18,217
2	19,94	397,604	19,420	0,520	0,270	20,623	18,217
3	18,94	358,724	19,420	0,480	0,230	20,623	18,217
4	19,18	367,872	19,420	0,240	0,058	20,623	18,217
5	19,34	374,036	19,420	0,080	0,006	20,623	18,217

sum 97,10 1.886,33 1,60 0,64

sum² 9428,410

Average 19,420

19,42

n 5

UJI SERAGAM

Rata2= 19,420

σ = 0,400999

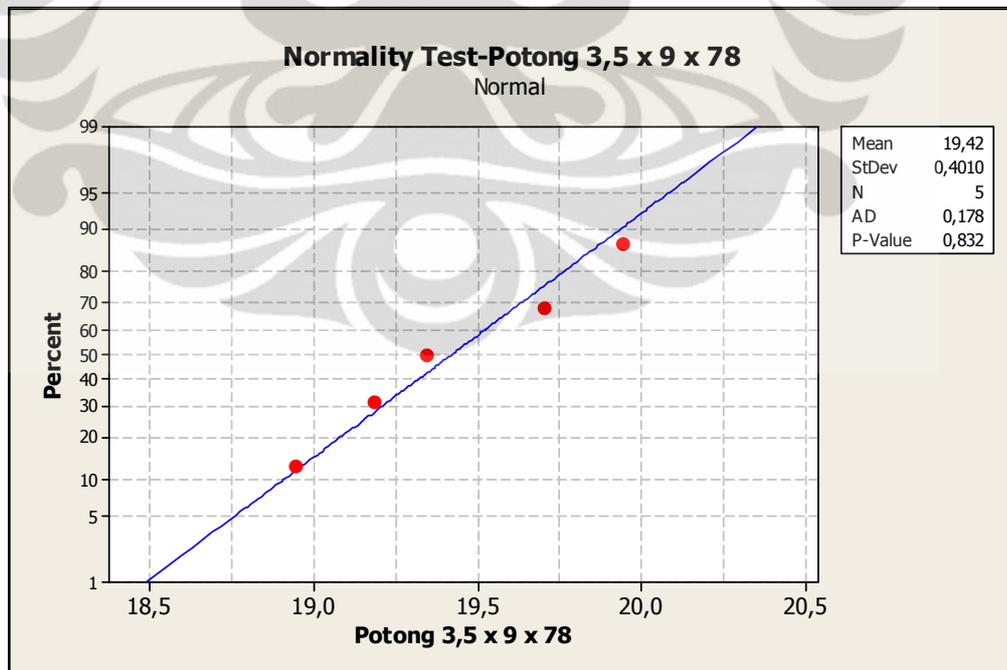
BKA = 20,623

BKB = 18,217 18,217

UJI KECUKUPAN DATA

N' = 1

Karena data yang tersedia ada 5, maka data dapat dikatakan cukup



(lanjutan)

No	Assy IV	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	68,53	4696,361	72,204	3,674	13,498	92,933	51,475
2	77,40	5990,760	72,204	5,196	26,998	92,933	51,475
3	64,71	4187,384	72,204	7,494	56,160	92,933	51,475
4	73,66	5425,796	72,204	1,456	2,120	92,933	51,475
5	70,41	4957,568	72,204	1,794	3,218	92,933	51,475
6	68,70	4719,690	72,204	3,504	12,278	92,933	51,475
7	65,13	4241,917	72,204	7,074	50,041	92,933	51,475
8	67,36	4537,370	72,204	4,844	23,464	92,933	51,475
9	81,37	6621,077	72,204	9,166	84,016	92,933	51,475
10	84,77	7185,953	72,204	12,566	157,904	92,933	51,475

sum **722,04** **52.563,88** **56,77** **429,70**
sum² 521341,762
Average 72,204
72,21
n 10

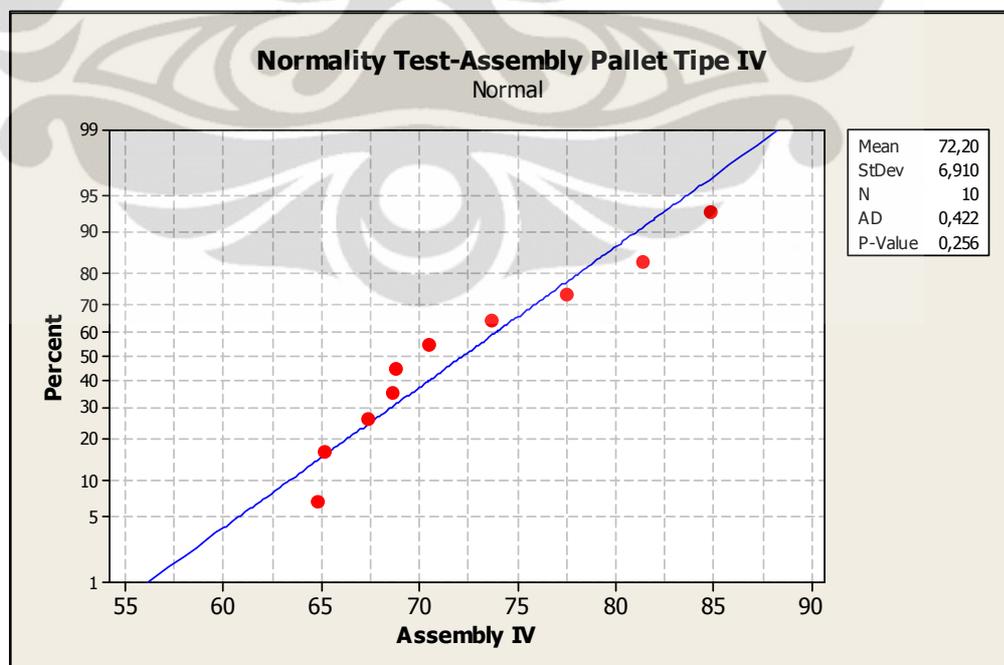
UJI SERAGAM

Rata2= 72,204
 σ = 6,909726156
BKA = 92,93317847
BKB = 51,47482153 51,4748215

UJI KECUKUPAN DATA

N' = 4

Karena data yang tersedia ada 10, maka data dapat dikatakan cukup



(lanjutan)

No	Assy III-1	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	47,22	2229,728	46,244	0,976	0,953	52,256	40,232
2	43,48	1890,510	46,244	2,764	7,640	52,256	40,232
3	47,76	2281,018	46,244	1,516	2,298	52,256	40,232
4	46,79	2189,304	46,244	0,546	0,298	52,256	40,232
5	45,67	2085,749	46,244	0,574	0,329	52,256	40,232
6	44,78	2005,248	46,244	1,464	2,143	52,256	40,232
7	43,01	1849,860	46,244	3,234	10,459	52,256	40,232
8	49,33	2433,449	46,244	3,086	9,523	52,256	40,232
9	46,62	2173,424	46,244	0,376	0,141	52,256	40,232
10	47,78	2282,928	46,244	1,536	2,359	52,256	40,232

sum **462,44** **21.421,22** **16,072** **36,144**

sum² 213850,754

Average 46,244

46,25

n 10

UJI SERAGAM

Rata2= 46,244

σ = 2,004002661

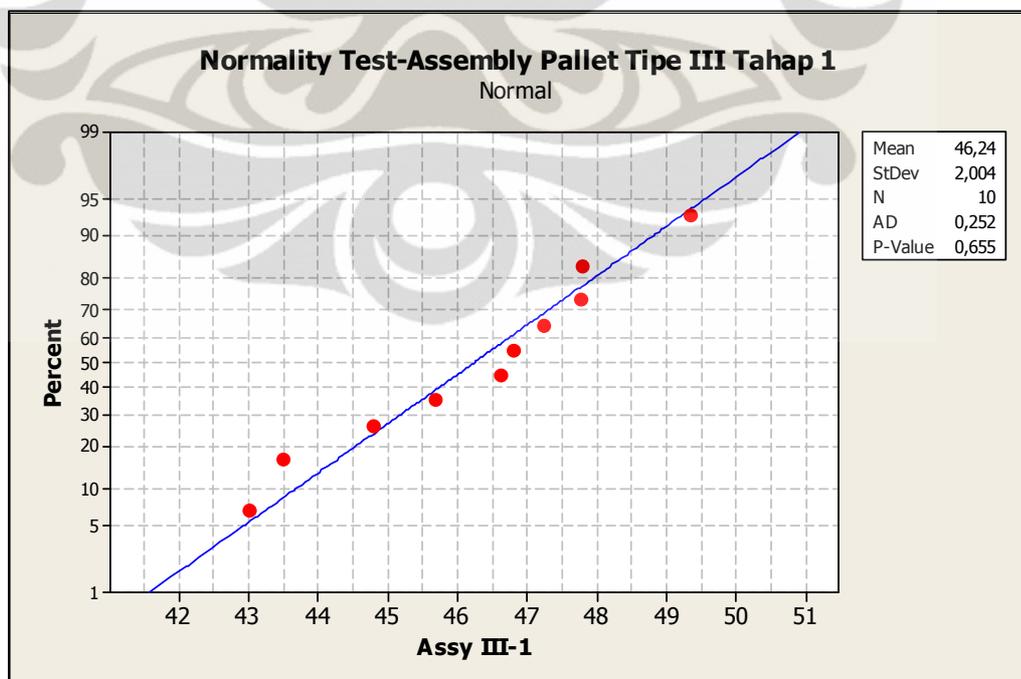
BKA = 52,25600798

BKB = 40,23199202 40,23199

UJI KECUKUPAN DATA

N' = 1

Karena data yang tersedia ada 10, maka data dapat dikatakan cukup



(lanjutan)

No	Assy III-2	ξ^2	\bar{x}	$\xi - \bar{x}$	$(\xi - \bar{x})^2$	BKA	BKB
1	111,85	12510,423	107,969	3,881	15,062	133,238	82,700
2	95,25	9072,563	107,969	12,719	161,773	133,238	82,700
3	116,26	13516,388	107,969	8,291	68,741	133,238	82,700
4	109,39	11966,172	107,969	1,421	2,019	133,238	82,700
5	121,74	14820,628	107,969	13,771	189,640	133,238	82,700
6	106,74	11393,428	107,969	1,229	1,510	133,238	82,700
7	102,52	10510,350	107,969	5,449	29,692	133,238	82,700
8	112,28	12606,798	107,969	4,311	18,585	133,238	82,700
9	108,00	11664,000	107,969	0,031	0,001	133,238	82,700
10	95,66	9150,836	107,969	12,309	151,511	133,238	82,700

sum 1.079,69 117.211,58 63,412 638,535
sum² 1165730,496
Average 107,969
107,97
n 10

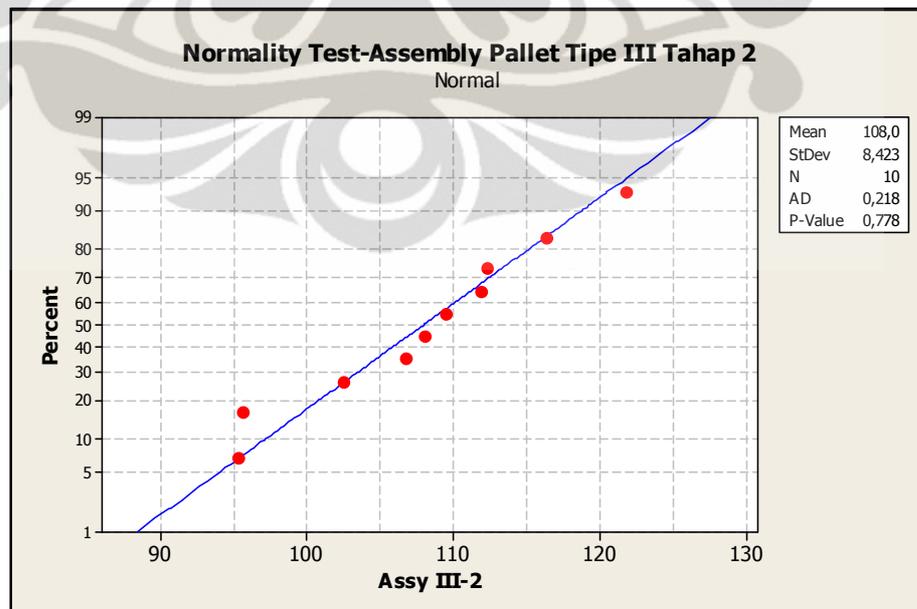
UJI SERAGAM

Rata2= 107,969
 σ = 8,423081318
BKA = 133,238244
BKB = 82,69975605 82,699756

UJI KECUKUPAN DATA

N' = 3

Karena data yang tersedia ada 10, maka data dapat dikatakan cukup



(lanjutan)

No	Assy II-1	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	59,83	3579,629	61,381	1,551	2,406	69,840	52,922
2	60,67	3680,849	61,381	0,711	0,506	69,840	52,922
3	60,40	3648,160	61,381	0,981	0,962	69,840	52,922
4	60,67	3680,849	61,381	0,711	0,506	69,840	52,922
5	67,08	4499,726	61,381	5,699	32,479	69,840	52,922
6	62,15	3862,623	61,381	0,769	0,591	69,840	52,922
7	61,31	3758,916	61,381	0,071	0,005	69,840	52,922
8	65,19	4249,736	61,381	3,809	14,508	69,840	52,922
9	58,10	3375,610	61,381	3,281	10,765	69,840	52,922
10	58,41	3411,728	61,381	2,971	8,827	69,840	52,922

sum **613,81** **37.747,83** **20,554** **71,554**

sum² 376762,716

Average 61,381

 61,39

n 10

UJI SERAGAM

Rata2= 61,381

σ = 2,81965896

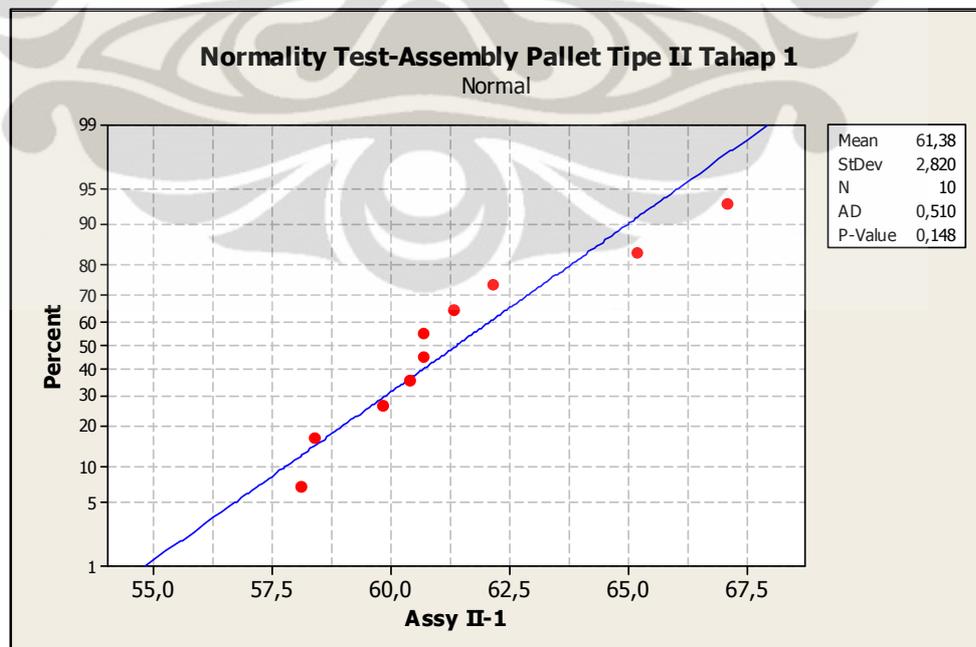
BKA = 69,8399769

BKB = 52,9220231 52,92202

UJI KECUKUPAN DATA

N' = 1

Karena data yang tersedia ada 10, maka data dapat dikatakan cukup



(lanjutan)

No	Assy II-2	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	196,13	38466,977	187,124	9,006	81,108	250,493	123,755
2	174,86	30576,020	187,124	12,264	150,406	250,493	123,755
3	175,65	30852,923	187,124	11,474	131,653	250,493	123,755
4	190,56	36313,114	187,124	3,436	11,806	250,493	123,755
5	201,32	40529,742	187,124	14,196	201,526	250,493	123,755
6	224,55	50422,703	187,124	37,426	1400,705	250,493	123,755
7	173,98	30269,040	187,124	13,144	172,765	250,493	123,755
8	166,72	27795,558	187,124	20,404	416,323	250,493	123,755
9	210,44	44284,994	187,124	23,316	543,636	250,493	123,755
10	157,03	24658,421	187,124	30,094	905,649	250,493	123,755

sum **1.871,24** **354.169,49** **174,760** **4015,577**

sum² 3501539,138

Average 187,124

187,13

n 10

UJI SERAGAM

Rata2= 187,124

σ = 21,12286029

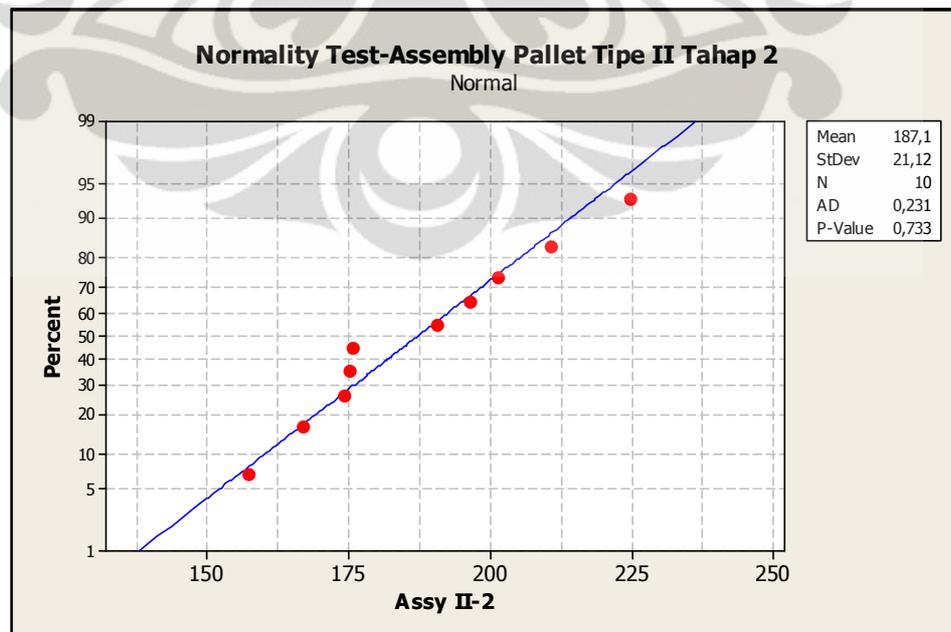
BKA = 250,4925809

BKB = 123,7554191 123,75542

UJI KECUKUPAN DATA

N' = 5

Karena data yang tersedia ada 10, maka data dapat dikatakan cukup



(lanjutan)

No	Assy I	xi ²	x bar	xi-xbar	(xi - x bar) ²	BKA	BKB
1	150,76	22728,578	151,711	0,951	0,905	164,137	139,286
2	153,34	23513,156	151,711	1,629	2,653	164,137	139,286
3	149,67	22401,109	151,711	2,041	4,167	164,137	139,286
4	146,95	21594,303	151,711	4,761	22,670	164,137	139,286
5	141,33	19974,169	151,711	10,381	107,772	164,137	139,286
6	151,73	23021,993	151,711	0,019	0,000	164,137	139,286
7	150,49	22647,240	151,711	1,221	1,492	164,137	139,286
8	155,44	24161,594	151,711	3,729	13,903	164,137	139,286
9	159,03	25290,541	151,711	7,319	53,563	164,137	139,286
10	152,77	23338,673	151,711	1,059	1,121	164,137	139,286
11	153,66	23611,396	151,711	1,949	3,797	164,137	139,286
12	155,25	24102,563	151,711	3,539	12,522	164,137	139,286
13	151,22	22867,488	151,711	0,491	0,241	164,137	139,286
14	149,26	22278,548	151,711	2,451	6,009	164,137	139,286
15	154,77	23953,753	151,711	3,059	9,355	164,137	139,286

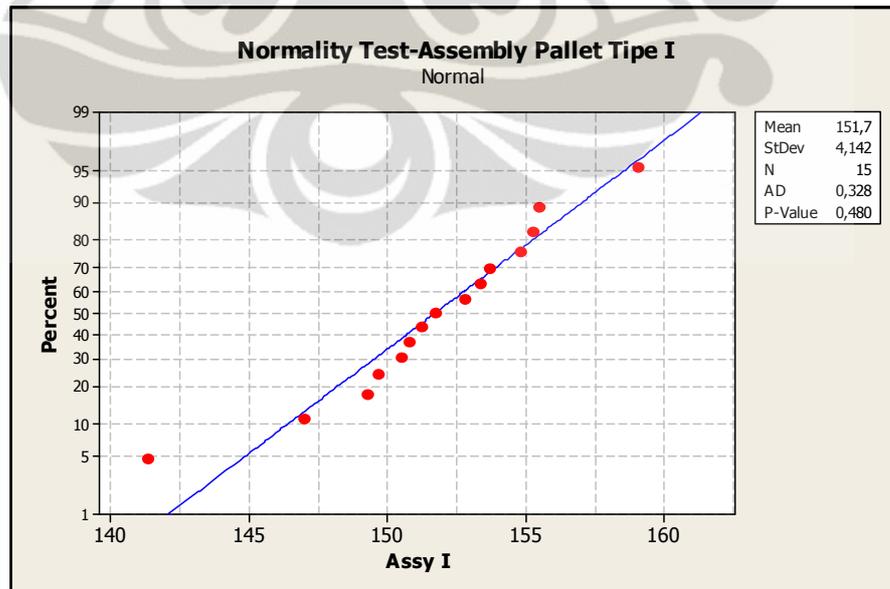
sum **2.275,67** **345.485,10** **44,60** **240,17**
sum² 5178673,949
Average 151,711
151,72
n 15

UJI SERAGAM

Rata2= 151,711
 σ = 4,141867879
BKA = 164,136937
BKB = 139,2857297 139,28573

UJI KECUKUPAN DATA

N' = 1
Karena data yang tersedia ada 20, maka data dapat dikatakan cukup



LAMPIRAN 3. Kode Program Solusi Penjadwalan dengan Algoritma DE

```
clc;
clear;
tic;

jumlah_pesanan = 72;
jumlah_rute = 6;
jumlah_dimensi = jumlah_pesanan;
ukuran_populasi = 720;
jumlah_iterasi = 500;
F = 0.5; %operator mutasi
CR = 0.6; %operator pindah silang
batas_bawah = -1;
batas_atas = 1;

%---Data waktu proses, due date, dan penalti-----
%Kolom 1-6 adalah waktu proses tiap rute
%kolom 7 adalah due date tiap pesanan (job)
%kolom 8 adalah penalti keterlambatan tiap job
data2 = [158    174 61  143 253 0   900 90
64  164 66  134 78  180 900 70
158 174 61  143 253 0   900 70
158 174 61  143 253 0   900 90
78  66 49  45 121 0   900 70
158 174 61  143 253 0   1800   90
158 174 61  143 253 0   1800   90
64  164 66  134 78  180 1800  70
107 217 106 168 103 312 2700  80
158 174 61  143 253 0   2700  90
158 174 61  143 253 0   2700  90
78  66 49  45 121 0   2700  70
158 174 61  143 253 0   3600  90
158 174 61  143 253 0   3600  90
78  66 49  45 121 0   3600  70
64  164 66  134 78  180 3600  70
158 174 61  143 253 0   4500  90
158 174 61  143 253 0   4500  70
158 174 61  143 253 0   5400  90
158 174 61  143 253 0   5400  90
158 174 61  143 253 0   6300  90
158 174 61  143 253 0   6300  90
78  66 49  45 121 0   6300  70
107 217 106 168 103 312 7200  80
107 217 106 168 103 312 7200  80
64  164 66  134 78  180 7200  70
64  164 66  134 78  180 7200  70
158 174 61  143 253 0   7200  90
158 174 61  143 253 0   7200  90
64  164 66  134 78  180 7200  70
64  164 66  134 78  180 8100  70
158 174 61  143 253 0   8100  70
158 174 61  143 253 0   8100  90
64  164 66  134 78  180 8100  70
64  164 66  134 78  180 9000  70
158 174 61  143 253 0   9000  70
```

(lanjutan)

```
biaya_keterlambatan=zeros(jumlah_pesanan,1);
due_date2=zeros(jumlah_pesanan,1);
penalti2=zeros(jumlah_pesanan,1);
for a = 1 : jumlah_pesanan
    for b = 1 : jumlah_rute
        if a == 1 && b == 1
            w(a,b)=wp(urutan_awal(a,c),b);
            w(a,b+1)=wp(urutan_awal(a,c),b+1);
            w(a,b+2)=wp(urutan_awal(a,c),b+2)+w(a,b);
            w(a,b+3)=wp(urutan_awal(a,c),b+3)+w(a,b+1);
        end
        if a > 1 && b == 1
            w(a,b)=wp(urutan_awal(a,c),b)+w(a-1,b);
            w(a,b+1)=wp(urutan_awal(a,c),b+1)+w(a-1,b+1);
        end
        if a > 1 && b == 3
            w(a,b)=wp(urutan_awal(a,c),b)+max(w(a-1,b),w(a,b-
2));
            w(a,b+1)=wp(urutan_awal(a,c),b+1)+max(w(a-
1,b+1),w(a,b-1));
        end
        if a == 1 && b == 5
            if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
                if wp(urutan_awal(a,c),b+1)>0
                    w(a,b) = w(a,b-2)+wp(urutan_awal(a,c),b);
                    w(a,b+1) = w(a,b)+wp(urutan_awal(a,c),b+1);
                    w(a,b+2) = 0;
                elseif wp(urutan_awal(a,c),b+1)== 0
                    w(a,b) = w(a,b-2)+wp(urutan_awal(a,c),b);
                    w(a,b+1) = 0;
                    w(a,b+2) = 0;
                end
            elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
                if wp(urutan_awal(a,c),b+1)>0
                    w(a,b) = w(a,b-1)+wp(urutan_awal(a,c),b);
                    w(a,b+1) = w(a,b)+wp(urutan_awal(a,c),b+1);
                    w(a,b+2) = 0;
                elseif wp(urutan_awal(a,c),b+1)== 0
                    w(a,b) = w(a,b-1)+wp(urutan_awal(a,c),b);
                    w(a,b+1) = 0;
                    w(a,b+2) = 0;
                end
            end
        end
    end
end
if a == 2 && b == 5
    if w(a-1,b+1) == 0
        if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_awal(a,c),b+1)>0
                w(a,b+1)= w(a,b-2)+wp(urutan_awal(a,c),b);
            end
        end
    end
    w(a,b+2)=w(a,b+1)+wp(urutan_awal(a,c),b+1);
    w(a,b) = 0;
    elseif wp(urutan_awal(a,c),b+1)== 0
        w(a,b+1)= w(a,b-2)+wp(urutan_awal(a,c),b);
        w(a,b+2)=0;
        w(a,b) = 0;
```

(lanjutan)

```
end
elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
if wp(urutan_awal(a,c),b+1)>0
w(a,b+1)= w(a,b-1)+wp(urutan_awal(a,c),b);

w(a,b+2)=w(a,b+1)+wp(urutan_awal(a,c),b+1);
w(a,b) = 0;
elseif wp(urutan_awal(a,c),b+1)== 0
w(a,b+1)= w(a,b-1)+wp(urutan_awal(a,c),b);
w(a,b+2)=0;
w(a,b) = 0;
end
end
elseif w(a-1,b+1) > 0
if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
w(a,b+2)=w(a,b-2)+wp(urutan_awal(a,c),b);
if w(a-1,b)>w(a,b+2)
if wp(urutan_awal(a,c),b+1)>0
w(a,b)=
w(a-1,b)+wp(urutan_awal(a,c),b);
w(a,b+1) = 0;
elseif wp(urutan_awal(a,c),b+1)==0;
w(a,b)=0;
w(a,b+1)=0;
end
elseif w(a-1,b)<w(a,b+2)
if wp(urutan_awal(a,c),b+1)>0
w(a,b)=
w(a,b+2)+wp(urutan_awal(a,c),b);
w(a,b+1) = 0;
elseif wp(urutan_awal(a,c),b+1)==0;
w(a,b)=0;
w(a,b+1)=0;
end
end
elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
w(a,b+2)=w(a,b-1)+wp(urutan_awal(a,c),b);
if w(a-1,b)>w(a,b+2)
if wp(urutan_awal(a,c),b+1)>0
w(a,b)=
w(a-1,b)+wp(urutan_awal(a,c),b);
w(a,b+1) = 0;
elseif wp(urutan_awal(a,c),b+1)==0
w(a,b)=0;
w(a,b+1)=0;
end
elseif w(a-1,b)<w(a,b+2)
if wp(urutan_awal(a,c),b+1)>0
w(a,b)=
w(a,b+2)+wp(urutan_awal(a,c),b);
w(a,b+1) = 0;
elseif wp(urutan_awal(a,c),b+1)==0
w(a,b)=0;
w(a,b+1)=0;
end
end
end
```

(lanjutan)

```
elseif waktu_totalassy2>waktu_totalassy1    &&
waktu_totalassy2>waktu_totalassy3
    waktu_total = waktu_totalassy2;
elseif waktu_totalassy3>waktu_totalassy1    &&
waktu_totalassy3>waktu_totalassy2
    waktu_total = waktu_totalassy3;
end
nilai_makespan_awal(c)=max(w(72,:));
end
[total_biaya_keterlambatan_min_awal, index_vektor_target] =
min(total_biaya_keterlambatan_awal);
total_keterlambatan=total_keterlambatan_awal(:,index_vektor_target
);
makespan_awal = waktu_total;
urutan=urutan_awal(:,index_vektor_target);
keterlambatan_sekarang=keterlambatan_awal(:,index_vektor_target);
jumlah_keterlambatan_awal=numel(find(keterlambatan_sekarang));

%---Hasil Awal Sebelum Iterasi-----
disp('Keterlambatan setiap job =');
disp(keterlambatan_sekarang');
disp('Urutan =');
disp(urutan');
disp('Nilai total biaya keterlambatan Awal Minimum =');
disp(total_biaya_keterlambatan_min_awal);
disp('Nilai total keterlambatan Awal =');
disp(total_keterlambatan);
disp('Jumlah keterlambatan awal =');
disp(jumlah_keterlambatan_awal);
disp('Nilai makespan Awal =');
disp(makespan_awal);

%---Memulai iterasi-----
proses = 0;
jumlah_maksimum_iterasi = 0;
while (proses == 0)
    jumlah_maksimum_iterasi = jumlah_maksimum_iterasi + 1;
    if jumlah_maksimum_iterasi == jumlah_iterasi
        proses =1;
    end
end

%---Mencari vektor target-----
[total_biaya_keterlambatan_min_awal, index_vektor_target] =
min(total_biaya_keterlambatan_awal);
vektor_target = populasi_target(:,index_vektor_target);
for a = 2 : ukuran_populasi
    vektor_target(:, a) = vektor_target(:, a-1);
end

%---Mencari Populasi Mutan-----
%---Mencari 2 vektor acak---
for a = 1 : ukuran_populasi
    index_vektor_acak1 = 1;
    index_vektor_acak2 = 1;
```

(lanjutan)

```
while (index_vektor_acak1 == index_vektor_acak2) |
(index_vektor_acak1 == index_vektor_target) | (index_vektor_acak2
== index_vektor_target)
    index_vektor_acak1 = randint(1,1,ukuran_populasi) + 1;
    index_vektor_acak2 = randint(1,1,ukuran_populasi) + 1;
end
vektor_acak1(:,a) = populasi_target(:,index_vektor_acak1);
vektor_acak2(:,a) = populasi_target(:,index_vektor_acak2);
end

%---Membentuk populasi mutan---
populasi_mutan = (vektor_acak1 - vektor_acak2) * F +
vektor_target;

%---Membentuk populasi trial-----
-
for a = 1 : jumlah_pesanan
    for b = 1 : ukuran_populasi
        if (rand() <= CR)
            populasi_trial(a,b) = populasi_mutan(a,b);
        else
            populasi_trial(a,b) = populasi_target(a,b);
        end
    end
end

%---Mencari urutan pengerjaan populasi trial-----
--
for a = 1 : ukuran_populasi;
    [hasil, urutan_trial(:,a)] = sort(populasi_trial(:,a));
end

%---Menghitung total biaya keterlambatan populasi trial-----
-----
keterlambatan_anak=[];
total_keterlambatan_anak = [];
total_biaya_keterlambatan_anak = [];
nilai_makespan_anak = [];
for c = 1 : ukuran_populasi
    %disp('ukuran pop c');disp(c);
    w = zeros(jumlah_pesanan, 7);
    keterlambatan=zeros(jumlah_pesanan,1);
    biaya_keterlambatan=zeros(jumlah_pesanan,1);
    due_date2=zeros(jumlah_pesanan,1);
    penalti2=zeros(jumlah_pesanan,1);
    for a = 1 : jumlah_pesanan
        %disp(' jumlah pesanan a');disp(a);
        for b = 1 : jumlah_rute
            if a == 1 && b == 1
                w(a,b)=wp(urutan_trial(a,c),b);
                w(a,b+1)=wp(urutan_trial(a,c),b+1);
                w(a,b+2)=wp(urutan_trial(a,c),b+2)+w(a,b);
                w(a,b+3)=wp(urutan_trial(a,c),b+3)+w(a,b+1);
            end
            if a > 1 && b == 1
                w(a,b)=wp(urutan_trial(a,c),b)+w(a-1,b);
            end
        end
    end
end
```

(lanjutan)

```
w(a,b+1)=wp(urutan_trial(a,c),b+1)+w(a-1,b+1);
end
if a > 1 && b == 3
    w(a,b)=wp(urutan_trial(a,c),b)+max(w(a-1,b),w(a,b-
2));
    w(a,b+1)=wp(urutan_trial(a,c),b+1)+max(w(a-
1,b+1),w(a,b-1));
end
if a == 1 && b == 5
    if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
        if wp(urutan_trial(a,c),b+1)>0
            w(a,b) = w(a,b-2)+wp(urutan_trial(a,c),b);
            w(a,b+1) = w(a,b)+wp(urutan_trial(a,c),b+1);
            w(a,b+2) = 0;
        elseif wp(urutan_trial(a,c),b+1)== 0
            w(a,b) = w(a,b-2)+wp(urutan_trial(a,c),b);
            w(a,b+1) = 0;
            w(a,b+2) = 0;
        end
    elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
        if wp(urutan_trial(a,c),b+1)>0
            w(a,b) = w(a,b-1)+wp(urutan_trial(a,c),b);
            w(a,b+1) = w(a,b)+wp(urutan_trial(a,c),b+1);
            w(a,b+2) = 0;
        elseif wp(urutan_trial(a,c),b+1)== 0
            w(a,b) = w(a,b-1)+wp(urutan_trial(a,c),b);
            w(a,b+1) = 0;
            w(a,b+2) = 0;
        end
    end
end
end
if a == 2 && b == 5
    if w(a-1,b+1) == 0
        if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        end
    end
end
if a == 3 && b == 5
    if w(a-1,b+1) == 0
        if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        end
    end
end
if a == 4 && b == 5
    if w(a-1,b+1) == 0
        if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        end
    end
end
if a == 5 && b == 5
    if w(a-1,b+1) == 0
        if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-2)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
            if wp(urutan_trial(a,c),b+1)>0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=w(a,b+1)+wp(urutan_trial(a,c),b+1);
                w(a,b) = 0;
            elseif wp(urutan_trial(a,c),b+1)== 0
                w(a,b+1)=w(a,b-1)+wp(urutan_trial(a,c),b);
                w(a,b+2)=0;
                w(a,b) = 0;
            end
        end
    end
end
end
```

(lanjutan)

```
end

%---Memilih antara populasi target awal atau trial-----
----
    for a = 1 : ukuran_populasi
        if total_biaya_keterlambatan_anak(a) <
total_biaya_keterlambatan_awal(a)
            populasi_target(:, a) = populasi_trial(:, a);
            urutan_awal(:, a) = urutan_trial(:, a);
            keterlambatan_awal(:, a) = keterlambatan_anak(:, a);
            total_keterlambatan_awal(a) =
total_keterlambatan_anak(a);
            total_biaya_keterlambatan_awal(a) =
total_biaya_keterlambatan_anak(a);
            nilai_makespan_awal(a) = nilai_makespan_anak(a);
        end
    end
end
[total_biaya_keterlambatan_min, index_vektor_target_akhir] =
min(total_biaya_keterlambatan_awal);
total_keterlambatan = total_keterlambatan_awal(:, index_vektor_target_akhir);
makespan_akhir = waktu_total;
urutan_akhir = urutan_awal(:, index_vektor_target_akhir);
keterlambatan_akhir = keterlambatan_awal(:, index_vektor_target_akhir);
jumlah_keterlambatan = numel(find(keterlambatan_akhir));

%---Solusi Akhir-----
disp('Keterlambatan terbaik setiap job =');
disp(keterlambatan_akhir);
disp('Urutan terbaik =');
disp(urutan_akhir);
disp('Total biaya keterlambatan terbaik =');
disp(total_biaya_keterlambatan_min);
disp('Total keterlambatan =');
disp(total_keterlambatan);
disp('Jumlah keterlambatan =');
disp(jumlah_keterlambatan);
disp('Makespan =');
disp(makespan_akhir);
disp(strcat('Waktu Komputasi :', ' ', num2str(toc), ' detik'));

save var3 data2 w wp urutan_akhir urutan_trial;
```

LAMPIRAN 4. Hasil *Design of Experiment*

F	CR	NP	Iterasi	Run 1	Run 2	Run 3	Run 4	Run 5
0,5	0,4	10	10	3848150	4266990	3816420	3661570	4369740
0,5	0,4	10	30	2508630	2863410	1612090	2244380	2675170
0,5	0,4	10	50	668800	1174800	881430	1291940	1204470
0,5	0,4	30	10	3693680	2868540	2938210	2305430	2766740
0,5	0,4	30	30	875750	593630	1110330	583720	909120
0,5	0,4	30	50	228710	181480	301050	115900	199070
0,5	0,4	50	10	2815440	2001550	2350700	2583450	2851780
0,5	0,4	50	30	478920	466700	736170	579410	493970
0,5	0,4	50	50	57720	153170	212630	110620	41900
0,5	0,6	10	10	4204340	3783000	2959430	3779430	3456400
0,5	0,6	10	30	1552190	2309740	1500780	1720160	2114510
0,5	0,6	10	50	822010	1765190	1440800	1650830	1938310
0,5	0,6	30	10	2176660	2918310	2585460	2958450	2436940
0,5	0,6	30	30	573250	1098070	1072780	773980	723830
0,5	0,6	30	50	624080	637610	160030	227980	648730
0,5	0,6	50	10	2050490	1455150	2568020	2101040	2761840
0,5	0,6	50	30	358110	447320	556610	397210	478840
0,5	0,6	50	50	232130	215040	14560	215500	104480
0,5	0,8	10	10	3820200	3460870	4195210	3588200	3644250
0,5	0,8	10	30	3294510	3412450	3044690	3455670	2815900
0,5	0,8	10	50	2902220	3087620	3174870	2021690	2669620
0,5	0,8	30	10	2773660	2904150	3128540	2180540	2443790
0,5	0,8	30	30	1185850	1663560	2081660	1711200	1404750
0,5	0,8	30	50	1437230	1224070	1278220	1217860	1197310
0,5	0,8	50	10	1968960	2548910	2257070	2045510	2677040
0,5	0,8	50	30	975670	1015880	1336890	1435100	1003840
0,5	0,8	50	50	601210	923800	1262330	1159390	975380
0,7	0,4	10	10	3130810	4080080	4361360	4246660	3863510
0,7	0,4	10	30	1777350	3618120	2442180	2082010	2392430
0,7	0,4	10	50	1125510	1733130	1278510	1380960	1579180
0,7	0,4	30	10	3338970	3096460	3344860	3160510	3356910
0,7	0,4	30	30	1517760	1343650	1585600	1229810	1211130
0,7	0,4	30	50	829880	1116070	1225480	589240	856190
0,7	0,4	50	10	3106640	2790630	2506820	2979180	3319050
0,7	0,4	50	30	1082810	1109360	1472910	1301250	1002310
0,7	0,4	50	50	481710	329340	395030	683400	678370
0,7	0,6	10	10	4446290	4379580	5029800	4298940	3990650
0,7	0,6	10	30	2224540	1656850	2624670	1764200	1251950
0,7	0,6	10	50	1295470	1354080	965070	1331580	1194460

(lanjutan)

F	CR	NP	Iterasi	Run 1	Run 2	Run 3	Run 4	Run 5
0,7	0,6	30	10	3272160	2565660	2565660	3478930	3786630
0,7	0,6	30	30	1556780	1362290	1543460	1254270	1121740
0,7	0,6	30	50	1069490	709050	817920	922370	636240
0,7	0,6	50	10	2292450	2446730	3042200	3245170	3068690
0,7	0,6	50	30	1395780	989190	1431820	1192900	1176610
0,7	0,6	50	50	987120	643560	680120	664560	410190
0,7	0,8	10	10	4045970	4469120	4202790	4297970	3727240
0,7	0,8	10	30	2856170	2073350	2526650	2035070	2578690
0,7	0,8	10	50	1755670	1364970	2236450	1572520	2586050
0,7	0,8	30	10	2820720	2957410	3229200	2955950	2756850
0,7	0,8	30	30	2356890	1474650	1516130	1681720	1561620
0,7	0,8	30	50	717900	824160	631750	755530	1120180
0,7	0,8	50	10	2965120	2423840	3108420	3179880	2855740
0,7	0,8	50	30	1594240	1126810	1162970	2009150	1054010
0,7	0,8	50	50	602070	621730	641490	745250	659990
0,9	0,4	10	10	4802220	4715480	3291120	3990910	3973410
0,9	0,4	10	30	2052160	2887760	2921020	2112780	2659620
0,9	0,4	10	50	1442700	2467070	2369800	1700310	2045720
0,9	0,4	30	10	4012920	3245350	4404990	3836700	3669310
0,9	0,4	30	30	1559470	2253380	1856690	2278570	1880040
0,9	0,4	30	50	1170760	1651210	1168290	2023370	1584830
0,9	0,4	50	10	4098970	3266680	3121520	3293190	3283230
0,9	0,4	50	30	2067520	1441570	2050290	1282970	1369090
0,9	0,4	50	50	1526790	1191850	1187080	1268210	1037380
0,9	0,6	10	10	4166680	3326020	4107110	3964880	4673050
0,9	0,6	10	30	2540050	2625510	3309430	2540790	3632740
0,9	0,6	10	50	2306190	2213950	2777830	2012040	1883910
0,9	0,6	30	10	3058260	2914130	3015220	3432710	3384140
0,9	0,6	30	30	1794860	2234900	1714310	2194910	2144320
0,9	0,6	30	50	1518100	1616180	1536190	1655380	1483960
0,9	0,6	50	10	3023290	2792880	3345590	3507630	2944460
0,9	0,6	50	30	1804480	2318830	2913860	1793750	1830730
0,9	0,6	50	50	1780810	1558430	1398470	917080	1031730
0,9	0,8	10	10	5014440	4187470	3728750	4846010	4453300
0,9	0,8	10	30	4270750	2933050	2745020	3781620	3494530
0,9	0,8	10	50	3138370	2643390	2177740	1938330	2743920
0,9	0,8	30	10	4020130	4050020	4187980	3634910	3770670
0,9	0,8	30	30	1925130	2494280	2270910	2553220	1845410
0,9	0,8	30	50	1858260	1599650	2198000	1484870	1616620
0,9	0,8	50	10	2890780	2888760	3976740	3872520	3683010
0,9	0,8	50	30	2314770	2419490	2274450	2340280	1289820
0,9	0,8	50	50	1805580	2096770	1323290	1260800	1104600

LAMPIRAN 5. Kode Program Penjadwalan Perusahaan

```
clc;
clear;
tic;

jumlah_pesanan = 72;
jumlah_rute = 6;

%---Data waktu proses, due date, dan penalti-----
job_kolom = [158    174 61  143 253 0   900 90
64  164 66  134 78  180 900 70
158 174 61  143 253 0   900 70
158 174 61  143 253 0   900 90
78  66  49  45 121 0   900 70
158 174 61  143 253 0   1800   90
158 174 61  143 253 0   1800   90
64  164 66  134 78  180 1800   70
107 217 106 168 103 312 2700   80
158 174 61  143 253 0   2700   90
158 174 61  143 253 0   2700   90
78  66  49  45 121 0   2700   70
158 174 61  143 253 0   3600   90
158 174 61  143 253 0   3600   90
78  66  49  45 121 0   3600   70
64  164 66  134 78  180 3600   70
158 174 61  143 253 0   4500   90
158 174 61  143 253 0   4500   70
158 174 61  143 253 0   5400   90
158 174 61  143 253 0   5400   90
158 174 61  143 253 0   6300   90
158 174 61  143 253 0   6300   90
78  66  49  45 121 0   6300   70
107 217 106 168 103 312 7200   80
107 217 106 168 103 312 7200   80
64  164 66  134 78  180 7200   70
64  164 66  134 78  180 7200   70
158 174 61  143 253 0   7200   90
158 174 61  143 253 0   7200   90
64  164 66  134 78  180 7200   70
64  164 66  134 78  180 8100   70
158 174 61  143 253 0   8100   70
158 174 61  143 253 0   8100   90
64  164 66  134 78  180 8100   70
64  164 66  134 78  180 9000   70
158 174 61  143 253 0   9000   70
107 217 106 168 103 312 9900   80
107 217 106 168 103 312 9900   80
158 174 61  143 253 0   9900   90
158 174 61  143 253 0   9900   90
64  164 66  134 78  180 9900   70
107 217 106 168 103 312 10800  80
158 174 61  143 253 0   10800  70
107 217 106 168 103 312 10800  70
64  164 66  134 78  180 11700  70
64  164 66  134 78  180 11700  70
107 217 106 168 103 312 11700  80
107 217 106 168 103 312 11700  80
```

(lanjutan)

```
w(a,b+2) = 0;
end
elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
if wp(a,b+1)>0
w(a,b) = w(a,b-1)+wp(a,b);
w(a,b+1) = w(a,b)+wp(a,b+1);
w(a,b+2) = 0;
elseif wp(a,b+1)== 0
w(a,b) = w(a,b-1)+wp(a,b);
w(a,b+1) = 0;
w(a,b+2) = 0;
end
end
end
if a == 2 && b == 5
if w(a-1,b+1) == 0
if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
if wp(a,b+1)>0
w(a,b+1)= w(a,b-2)+wp(a,b);
w(a,b+2)=w(a,b+1)+wp(a,b+1);
w(a,b) = 0;
elseif wp(a,b+1)== 0
w(a,b+1)= w(a,b-2)+wp(a,b);
w(a,b+2)=0;
w(a,b) = 0;
end
elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
if wp(a,b+1)>0
w(a,b+1)= w(a,b-1)+wp(a,b);
w(a,b+2)=w(a,b+1)+wp(a,b+1);
w(a,b) = 0;
elseif wp(a,b+1)== 0
w(a,b+1)= w(a,b-1)+wp(a,b);
w(a,b+2)=0;
w(a,b) = 0;
end
end
elseif w(a-1,b+1) > 0
if w(a,b-2)>w(a,b-1) || w(a,b-2)==w(a,b-1)
w(a,b+2)=w(a,b-2)+wp(a,b);
if w(a-1,b)>w(a,b+2)
if wp(a,b+1)>0
w(a,b) = w(a-1,b)+wp(a,b);
w(a,b+1) = 0;
elseif wp(a,b+1)==0;
w(a,b)=0;
w(a,b+1)=0;
end
elseif w(a-1,b)<w(a,b+2)
if wp(a,b+1)>0
w(a,b) = w(a,b+2)+wp(a,b);
w(a,b+1) = 0;
elseif wp(a,b+1)==0;
w(a,b)=0;
w(a,b+1)=0;
end
end
end
```

(lanjutan)

```
end
elseif w(a,b-2)<w(a,b-1) || w(a,b-2)==w(a,b-1)
w(a,b+2)=w(a,b-1)+wp(a,b);
if w(a-1,b)>w(a,b+2)
if wp(a,b+1)>0
w(a,b)= w(a-1,b)+wp(a,b);
w(a,b+1) = 0;
elseif wp(a,b+1)==0
w(a,b)=0;
w(a,b+1)=0;
end
elseif w(a-1,b)<w(a,b+2)
if wp(a,b+1)>0
w(a,b)= w(a,b+2)+wp(a,b);
w(a,b+1) = 0;
elseif wp(a,b+1)==0
w(a,b)=0;
w(a,b+1)=0;
end
end
end
end
end
if a > 2 && b == 5
%----- begin of modification -----
wtemp=[0 max(w(1:a,b));1 max(w(1:a,b+1));2
max(w(1:a,b+2))];
% assamble mana yg paling min proses sebelumnya?
wsort =sortrows(wtemp,2);
w(a,b+wsort(1,1)) = max(w(a,b-1),wsort(1,2))+wp(a,b);
if wp(a,b+1)>0
w(a,b+wsort(2,1)) =
max(w(a,b+wsort(1,1)),wsort(2,2))+wp(a,b+1);
end
%----- end of modification -----
end
end
end

waktu_totalassy1 = max(w(:,5));
waktu_totalassy2 = max(w(:,6));
waktu_totalassy3 = max(w(:,7));
if waktu_totalassy1>waktu_totalassy2 &&
waktu_totalassy1>waktu_totalassy3
waktu_total = waktu_totalassy1;
elseif waktu_totalassy2>waktu_totalassy1 &&
waktu_totalassy2>waktu_totalassy3
waktu_total = waktu_totalassy2;
elseif waktu_totalassy3>waktu_totalassy1 &&
waktu_totalassy3>waktu_totalassy2
waktu_total = waktu_totalassy3;
end
keterlambatan=zeros(jumlah_pesanan,1);
```

LAMPIRAN 6. Waktu Penyelesaian Setiap Job

Job	Serut 1	Serut 2	Potong 1	Potong 2	Assambly 1	Assambly 2	Assambly 3							
1	datang:	0	datang:	0	datang:	158	datang:	174	datang:	317	datang:	0	datang:	0
	mulai:	0	mulai:	0	mulai:	158	mulai:	174	mulai:	317	mulai:	0	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	253	proses:	0	proses:	0
	akhir:	158	akhir:	174	akhir:	219	akhir:	317	akhir:	570	akhir:	0	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	-
4	datang:	0	datang:	0	datang:	316	datang:	348	datang:	0	datang:	491	datang:	744
	mulai:	158	mulai:	174	mulai:	316	mulai:	348	mulai:	0	mulai:	491	mulai:	744
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	253	proses:	253
	akhir:	316	akhir:	348	akhir:	377	akhir:	491	akhir:	0	akhir:	744	akhir:	997
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2
2	datang:	0	datang:	0	datang:	380	datang:	512	datang:	646	datang:	724	datang:	0
	mulai:	316	mulai:	348	mulai:	380	mulai:	512	mulai:	646	mulai:	744	mulai:	0
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	78	proses:	180	proses:	0
	akhir:	380	akhir:	512	akhir:	446	akhir:	646	akhir:	724	akhir:	924	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	rakit 2	Komp:	-
5	datang:	0	datang:	0	datang:	458	datang:	578	datang:	623	datang:	0	datang:	0
	mulai:	380	mulai:	512	mulai:	458	mulai:	578	mulai:	724	mulai:	0	mulai:	0
	proses:	78	proses:	66	proses:	49	proses:	45	proses:	121	proses:	0	proses:	0
	akhir:	458	akhir:	578	akhir:	507	akhir:	623	akhir:	845	akhir:	0	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	-
3	datang:	0	datang:	0	datang:	616	datang:	752	datang:	895	datang:	0	datang:	0
	mulai:	458	mulai:	578	mulai:	616	mulai:	752	mulai:	895	mulai:	0	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	253	proses:	0	proses:	0
	akhir:	616	akhir:	752	akhir:	677	akhir:	895	akhir:	1148	akhir:	0	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	-
7	datang:	0	datang:	0	datang:	774	datang:	926	datang:	0	datang:	1069	datang:	0
	mulai:	616	mulai:	752	mulai:	774	mulai:	926	mulai:	0	mulai:	1069	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	253	proses:	0
	akhir:	774	akhir:	926	akhir:	835	akhir:	1069	akhir:	0	akhir:	1322	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	-
6	datang:	0	datang:	0	datang:	838	datang:	1090	datang:	1302	datang:	0	datang:	1224
	mulai:	774	mulai:	926	mulai:	838	mulai:	1090	mulai:	1302	mulai:	0	mulai:	1224
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	180	proses:	0	proses:	78
	akhir:	838	akhir:	1090	akhir:	904	akhir:	1224	akhir:	1482	akhir:	0	akhir:	1302
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	-	Komp:	rakit 1
8	datang:	0	datang:	0	datang:	996	datang:	1264	datang:	0	datang:	0	datang:	1407
	mulai:	838	mulai:	1090	mulai:	996	mulai:	1264	mulai:	0	mulai:	0	mulai:	1407
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	0	proses:	253
	akhir:	996	akhir:	1264	akhir:	1057	akhir:	1407	akhir:	0	akhir:	0	akhir:	1660
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	-	Komp:	rakit 1
9	datang:	0	datang:	0	datang:	1103	datang:	1481	datang:	1752	datang:	1649	datang:	0
	mulai:	996	mulai:	1264	mulai:	1103	mulai:	1481	mulai:	1752	mulai:	1649	mulai:	0
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	312	proses:	103	proses:	0
	akhir:	1103	akhir:	1481	akhir:	1209	akhir:	1649	akhir:	2064	akhir:	1752	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	rakit 1	Komp:	-
10	datang:	0	datang:	0	datang:	1181	datang:	1547	datang:	0	datang:	0	datang:	1592
	mulai:	1103	mulai:	1481	mulai:	1181	mulai:	1547	mulai:	0	mulai:	0	mulai:	1660
	proses:	78	proses:	66	proses:	49	proses:	45	proses:	0	proses:	0	proses:	121
	akhir:	1181	akhir:	1547	akhir:	1230	akhir:	1592	akhir:	0	akhir:	0	akhir:	1781
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	-	Komp:	rakit 1
11	datang:	0	datang:	0	datang:	1259	datang:	1613	datang:	0	datang:	1658	datang:	0
	mulai:	1181	mulai:	1547	mulai:	1259	mulai:	1613	mulai:	0	mulai:	1752	mulai:	0
	proses:	78	proses:	66	proses:	49	proses:	45	proses:	0	proses:	121	proses:	0
	akhir:	1259	akhir:	1613	akhir:	1308	akhir:	1658	akhir:	0	akhir:	1873	akhir:	0
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	-
12	datang:	0	datang:	0	datang:	1417	datang:	1787	datang:	0	datang:	0	datang:	1930
	mulai:	1259	mulai:	1613	mulai:	1417	mulai:	1787	mulai:	0	mulai:	0	mulai:	1930
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	0	proses:	253
	akhir:	1417	akhir:	1787	akhir:	1478	akhir:	1930	akhir:	0	akhir:	0	akhir:	2183
	Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	-	Komp:	rakit 1

(lanjutan)

33	datang: 0 mulai: 1417 proses: 64 akhir: 1481 Komp: papan	datang: 0 mulai: 1787 proses: 164 akhir: 1951 Komp: balok	datang: 1481 mulai: 1481 proses: 66 akhir: 1547 Komp: papan	datang: 1951 mulai: 1951 proses: 134 akhir: 2085 Komp: balok	datang: 2163 mulai: 2163 proses: 180 akhir: 2343 Komp: rakit 2	datang: 2085 mulai: 2085 proses: 78 akhir: 2163 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
13	datang: 0 mulai: 1481 proses: 64 akhir: 1545 Komp: papan	datang: 0 mulai: 1951 proses: 164 akhir: 2115 Komp: balok	datang: 1545 mulai: 1545 proses: 66 akhir: 1611 Komp: papan	datang: 2115 mulai: 2115 proses: 134 akhir: 2249 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 2249 mulai: 2249 proses: 253 akhir: 2502 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
16	datang: 0 mulai: 1545 proses: 158 akhir: 1703 Komp: papan	datang: 0 mulai: 2115 proses: 174 akhir: 2289 Komp: balok	datang: 1703 mulai: 1703 proses: 61 akhir: 1764 Komp: papan	datang: 2289 mulai: 2289 proses: 143 akhir: 2432 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 2432 mulai: 2432 proses: 253 akhir: 2685 Komp: rakit 1
15	datang: 0 mulai: 1703 proses: 158 akhir: 1861 Komp: papan	datang: 0 mulai: 2289 proses: 174 akhir: 2463 Komp: balok	datang: 1861 mulai: 1861 proses: 61 akhir: 1922 Komp: papan	datang: 2463 mulai: 2463 proses: 143 akhir: 2606 Komp: balok	datang: 2606 mulai: 2606 proses: 253 akhir: 2859 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
31	datang: 0 mulai: 1861 proses: 64 akhir: 1925 Komp: papan	datang: 0 mulai: 2463 proses: 164 akhir: 2627 Komp: balok	datang: 1925 mulai: 1925 proses: 66 akhir: 1991 Komp: papan	datang: 2627 mulai: 2627 proses: 134 akhir: 2761 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 2761 mulai: 2761 proses: 78 akhir: 2839 Komp: rakit 1	datang: 2839 mulai: 2839 proses: 180 akhir: 3019 Komp: rakit 2
30	datang: 0 mulai: 1925 proses: 158 akhir: 2083 Komp: papan	datang: 0 mulai: 2627 proses: 174 akhir: 2801 Komp: balok	datang: 2083 mulai: 2083 proses: 61 akhir: 2144 Komp: papan	datang: 2801 mulai: 2801 proses: 143 akhir: 2944 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 2944 mulai: 2944 proses: 253 akhir: 3197 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
14	datang: 0 mulai: 2083 proses: 64 akhir: 2147 Komp: papan	datang: 0 mulai: 2801 proses: 164 akhir: 2965 Komp: balok	datang: 2147 mulai: 2147 proses: 66 akhir: 2213 Komp: papan	datang: 2965 mulai: 2965 proses: 134 akhir: 3099 Komp: balok	datang: 3099 mulai: 3099 proses: 78 akhir: 3177 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 3177 mulai: 3177 proses: 180 akhir: 3357 Komp: rakit 2
18	datang: 0 mulai: 2147 proses: 158 akhir: 2305 Komp: papan	datang: 0 mulai: 2965 proses: 174 akhir: 3139 Komp: balok	datang: 2305 mulai: 2305 proses: 61 akhir: 2366 Komp: papan	datang: 3139 mulai: 3139 proses: 143 akhir: 3282 Komp: balok	datang: 3282 mulai: 3282 proses: 253 akhir: 3535 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
19	datang: 0 mulai: 2305 proses: 107 akhir: 2412 Komp: papan	datang: 0 mulai: 3139 proses: 217 akhir: 3356 Komp: balok	datang: 2412 mulai: 2412 proses: 106 akhir: 2518 Komp: papan	datang: 3356 mulai: 3356 proses: 168 akhir: 3524 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 3524 mulai: 3524 proses: 103 akhir: 3627 Komp: rakit 1	datang: 3627 mulai: 3627 proses: 312 akhir: 3939 Komp: rakit 2
17	datang: 0 mulai: 2412 proses: 158 akhir: 2570 Komp: papan	datang: 0 mulai: 3356 proses: 174 akhir: 3530 Komp: balok	datang: 2570 mulai: 2570 proses: 61 akhir: 2631 Komp: papan	datang: 3530 mulai: 3530 proses: 143 akhir: 3673 Komp: balok	datang: 3673 mulai: 3673 proses: 253 akhir: 3926 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
37	datang: 0 mulai: 2570 proses: 164 akhir: 2734 Komp: papan	datang: 0 mulai: 3530 proses: 164 akhir: 3694 Komp: balok	datang: 2734 mulai: 2734 proses: 66 akhir: 2800 Komp: papan	datang: 3694 mulai: 3694 proses: 134 akhir: 3828 Komp: balok	datang: 3906 mulai: 3926 proses: 180 akhir: 4106 Komp: rakit 2	datang: 3828 mulai: 3828 proses: 78 akhir: 3906 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
22	datang: 0 mulai: 2734 proses: 78 akhir: 2812 Komp: papan	datang: 0 mulai: 3694 proses: 66 akhir: 3760 Komp: balok	datang: 2812 mulai: 2812 proses: 49 akhir: 2861 Komp: papan	datang: 3760 mulai: 3760 proses: 45 akhir: 3805 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 3805 mulai: 3906 proses: 121 akhir: 4027 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
68	datang: 0 mulai: 2812 proses: 107 akhir: 2919 Komp: papan	datang: 0 mulai: 3760 proses: 217 akhir: 3977 Komp: balok	datang: 2919 mulai: 2919 proses: 106 akhir: 3025 Komp: papan	datang: 3977 mulai: 3977 proses: 168 akhir: 4145 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 4248 mulai: 4248 proses: 312 akhir: 4560 Komp: rakit 2	datang: 4145 mulai: 4145 proses: 103 akhir: 4248 Komp: rakit 1

(lanjutan)

51	datang: 0 mulai: 2919 proses: 158 akhir: 3077 Komp: papan	datang: 0 mulai: 3977 proses: 174 akhir: 4151 Komp: balok	datang: 3077 mulai: 3077 proses: 61 akhir: 3138 Komp: papan	datang: 4151 mulai: 4151 proses: 143 akhir: 4294 Komp: balok	datang: 4294 mulai: 4294 proses: 253 akhir: 4547 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
20	datang: 0 mulai: 3077 proses: 158 akhir: 3235 Komp: papan	datang: 0 mulai: 4151 proses: 174 akhir: 4325 Komp: balok	datang: 3235 mulai: 3235 proses: 61 akhir: 3296 Komp: papan	datang: 4325 mulai: 4325 proses: 143 akhir: 4468 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 4468 mulai: 4468 proses: 253 akhir: 4721 Komp: rakit 1
24	datang: 0 mulai: 3235 proses: 158 akhir: 3393 Komp: papan	datang: 0 mulai: 4325 proses: 174 akhir: 4499 Komp: balok	datang: 3393 mulai: 3393 proses: 61 akhir: 3454 Komp: papan	datang: 4499 mulai: 4499 proses: 143 akhir: 4642 Komp: balok	datang: 4642 mulai: 4642 proses: 253 akhir: 4895 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
60	datang: 0 mulai: 3393 proses: 64 akhir: 3457 Komp: papan	datang: 0 mulai: 4499 proses: 164 akhir: 4663 Komp: balok	datang: 3457 mulai: 3457 proses: 66 akhir: 3523 Komp: papan	datang: 4663 mulai: 4663 proses: 134 akhir: 4797 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 4797 mulai: 4797 proses: 78 akhir: 4875 Komp: rakit 1	datang: 4875 mulai: 4875 proses: 180 akhir: 5055 Komp: rakit 2
23	datang: 0 mulai: 3457 proses: 158 akhir: 3615 Komp: papan	datang: 0 mulai: 4663 proses: 174 akhir: 4837 Komp: balok	datang: 3615 mulai: 3615 proses: 61 akhir: 3676 Komp: papan	datang: 4837 mulai: 4837 proses: 143 akhir: 4980 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 4980 mulai: 4980 proses: 253 akhir: 5233 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
27	datang: 0 mulai: 3615 proses: 158 akhir: 3773 Komp: papan	datang: 0 mulai: 4837 proses: 174 akhir: 5011 Komp: balok	datang: 3773 mulai: 3773 proses: 61 akhir: 3834 Komp: papan	datang: 5011 mulai: 5011 proses: 143 akhir: 5154 Komp: balok	datang: 5154 mulai: 5154 proses: 253 akhir: 5407 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
63	datang: 0 mulai: 3773 proses: 158 akhir: 3931 Komp: papan	datang: 0 mulai: 5011 proses: 174 akhir: 5185 Komp: balok	datang: 3931 mulai: 3931 proses: 61 akhir: 3992 Komp: papan	datang: 5185 mulai: 5185 proses: 143 akhir: 5328 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 5328 mulai: 5328 proses: 253 akhir: 5581 Komp: rakit 1
46	datang: 0 mulai: 3931 proses: 64 akhir: 3995 Komp: papan	datang: 0 mulai: 5185 proses: 164 akhir: 5349 Komp: balok	datang: 3995 mulai: 3995 proses: 66 akhir: 4061 Komp: papan	datang: 5349 mulai: 5349 proses: 134 akhir: 5483 Komp: balok	datang: 5561 mulai: 5561 proses: 180 akhir: 5741 Komp: rakit 2	datang: 5483 mulai: 5483 proses: 78 akhir: 5561 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
29	datang: 0 mulai: 3995 proses: 78 akhir: 4073 Komp: papan	datang: 0 mulai: 5349 proses: 66 akhir: 5415 Komp: balok	datang: 4073 mulai: 4073 proses: 49 akhir: 4122 Komp: papan	datang: 5415 mulai: 5415 proses: 45 akhir: 5460 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 5460 mulai: 5561 proses: 121 akhir: 5682 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
67	datang: 0 mulai: 4073 proses: 158 akhir: 4231 Komp: papan	datang: 0 mulai: 5415 proses: 174 akhir: 5589 Komp: balok	datang: 4231 mulai: 4231 proses: 61 akhir: 4292 Komp: papan	datang: 5589 mulai: 5589 proses: 143 akhir: 5732 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 5732 mulai: 5732 proses: 253 akhir: 5985 Komp: rakit 1
28	datang: 0 mulai: 4231 proses: 158 akhir: 4389 Komp: papan	datang: 0 mulai: 5589 proses: 174 akhir: 5763 Komp: balok	datang: 4389 mulai: 4389 proses: 61 akhir: 4450 Komp: papan	datang: 5763 mulai: 5763 proses: 143 akhir: 5906 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 5906 mulai: 5906 proses: 253 akhir: 6159 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
21	datang: 0 mulai: 4389 proses: 78 akhir: 4467 Komp: papan	datang: 0 mulai: 5763 proses: 66 akhir: 5829 Komp: balok	datang: 4467 mulai: 4467 proses: 49 akhir: 4516 Komp: papan	datang: 5829 mulai: 5829 proses: 45 akhir: 5874 Komp: balok	datang: 5874 mulai: 5874 proses: 121 akhir: 5995 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
65	datang: 0 mulai: 4467 proses: 107 akhir: 4574 Komp: papan	datang: 0 mulai: 5829 proses: 217 akhir: 6046 Komp: balok	datang: 4574 mulai: 4574 proses: 106 akhir: 4680 Komp: papan	datang: 6046 mulai: 6046 proses: 168 akhir: 6214 Komp: balok	datang: 6317 mulai: 6317 proses: 312 akhir: 6629 Komp: rakit 2	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 6214 mulai: 6214 proses: 103 akhir: 6317 Komp: rakit 1

(lanjutan)

26	datang:	0	datang:	0	datang:	4638	datang:	6210	datang:	0	datang:	6344	datang:	6422
	mulai:	4574	mulai:	6046	mulai:	4638	mulai:	6210	mulai:	0	mulai:	6344	mulai:	6422
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	0	proses:	78	proses:	180
	akhir:	4638	akhir:	6210	akhir:	4704	akhir:	6344	akhir:	0	akhir:	6422	akhir:	6602
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2	
35	datang:	0	datang:	0	datang:	4702	datang:	6374	datang:	0	datang:	6508	datang:	6586
	mulai:	4638	mulai:	6210	mulai:	4702	mulai:	6374	mulai:	0	mulai:	6508	mulai:	6602
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	0	proses:	78	proses:	180
	akhir:	4702	akhir:	6374	akhir:	4768	akhir:	6508	akhir:	0	akhir:	6586	akhir:	6782
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2	
25	datang:	0	datang:	0	datang:	4766	datang:	6538	datang:	6750	datang:	6672	datang:	0
	mulai:	4702	mulai:	6374	mulai:	4766	mulai:	6538	mulai:	6750	mulai:	6672	mulai:	0
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	180	proses:	78	proses:	0
	akhir:	4766	akhir:	6538	akhir:	4832	akhir:	6672	akhir:	6930	akhir:	6750	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	rakit 1	Komp:	-	
38	datang:	0	datang:	0	datang:	4830	datang:	6702	datang:	0	datang:	6836	datang:	6914
	mulai:	4766	mulai:	6538	mulai:	4830	mulai:	6702	mulai:	0	mulai:	6836	mulai:	6914
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	0	proses:	78	proses:	180
	akhir:	4830	akhir:	6702	akhir:	4896	akhir:	6836	akhir:	0	akhir:	6914	akhir:	7094
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2	
34	datang:	0	datang:	0	datang:	4937	datang:	6919	datang:	7190	datang:	7087	datang:	0
	mulai:	4830	mulai:	6702	mulai:	4937	mulai:	6919	mulai:	7190	mulai:	7087	mulai:	0
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	312	proses:	103	proses:	0
	akhir:	4937	akhir:	6919	akhir:	5043	akhir:	7087	akhir:	7502	akhir:	7190	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	rakit 1	Komp:	-	
36	datang:	0	datang:	0	datang:	5095	datang:	7093	datang:	0	datang:	0	datang:	7236
	mulai:	4937	mulai:	6919	mulai:	5095	mulai:	7093	mulai:	0	mulai:	0	mulai:	7236
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	0	proses:	253
	akhir:	5095	akhir:	7093	akhir:	5156	akhir:	7236	akhir:	0	akhir:	0	akhir:	7489
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	-	Komp:	rakit 1	
40	datang:	0	datang:	0	datang:	5159	datang:	7257	datang:	0	datang:	7391	datang:	7469
	mulai:	5095	mulai:	7093	mulai:	5159	mulai:	7257	mulai:	0	mulai:	7391	mulai:	7489
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	0	proses:	78	proses:	180
	akhir:	5159	akhir:	7257	akhir:	5225	akhir:	7391	akhir:	0	akhir:	7469	akhir:	7669
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2	
32	datang:	0	datang:	0	datang:	5317	datang:	7431	datang:	0	datang:	7574	datang:	0
	mulai:	5159	mulai:	7257	mulai:	5317	mulai:	7431	mulai:	0	mulai:	7574	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	253	proses:	0
	akhir:	5317	akhir:	7431	akhir:	5378	akhir:	7574	akhir:	0	akhir:	7827	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	-	
49	datang:	0	datang:	0	datang:	5475	datang:	7605	datang:	7748	datang:	0	datang:	0
	mulai:	5317	mulai:	7431	mulai:	5475	mulai:	7605	mulai:	7748	mulai:	0	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	253	proses:	0	proses:	0
	akhir:	5475	akhir:	7605	akhir:	5536	akhir:	7748	akhir:	8001	akhir:	0	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	-	
57	datang:	0	datang:	0	datang:	5582	datang:	7822	datang:	0	datang:	8093	datang:	7990
	mulai:	5475	mulai:	7605	mulai:	5582	mulai:	7822	mulai:	0	mulai:	8093	mulai:	7990
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	0	proses:	312	proses:	103
	akhir:	5582	akhir:	7822	akhir:	5688	akhir:	7990	akhir:	0	akhir:	8405	akhir:	8093
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 2	Komp:	rakit 1	
41	datang:	0	datang:	0	datang:	5740	datang:	7996	datang:	8139	datang:	0	datang:	0
	mulai:	5582	mulai:	7822	mulai:	5740	mulai:	7996	mulai:	8139	mulai:	0	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	253	proses:	0	proses:	0
	akhir:	5740	akhir:	7996	akhir:	5801	akhir:	8139	akhir:	8392	akhir:	0	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	-	
58	datang:	0	datang:	0	datang:	5818	datang:	8062	datang:	0	datang:	0	datang:	8107
	mulai:	5740	mulai:	7996	mulai:	5818	mulai:	8062	mulai:	0	mulai:	0	mulai:	8107
	proses:	78	proses:	66	proses:	49	proses:	45	proses:	0	proses:	0	proses:	121
	akhir:	5818	akhir:	8062	akhir:	5867	akhir:	8107	akhir:	0	akhir:	0	akhir:	8228
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	-	Komp:	rakit 1	
39	datang:	0	datang:	0	datang:	5925	datang:	8279	datang:	8550	datang:	0	datang:	8447
	mulai:	5818	mulai:	8062	mulai:	5925	mulai:	8279	mulai:	8550	mulai:	0	mulai:	8447
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	312	proses:	0	proses:	103
	akhir:	5925	akhir:	8279	akhir:	6031	akhir:	8447	akhir:	8862	akhir:	0	akhir:	8550
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	-	Komp:	rakit 1	

(lanjutan)

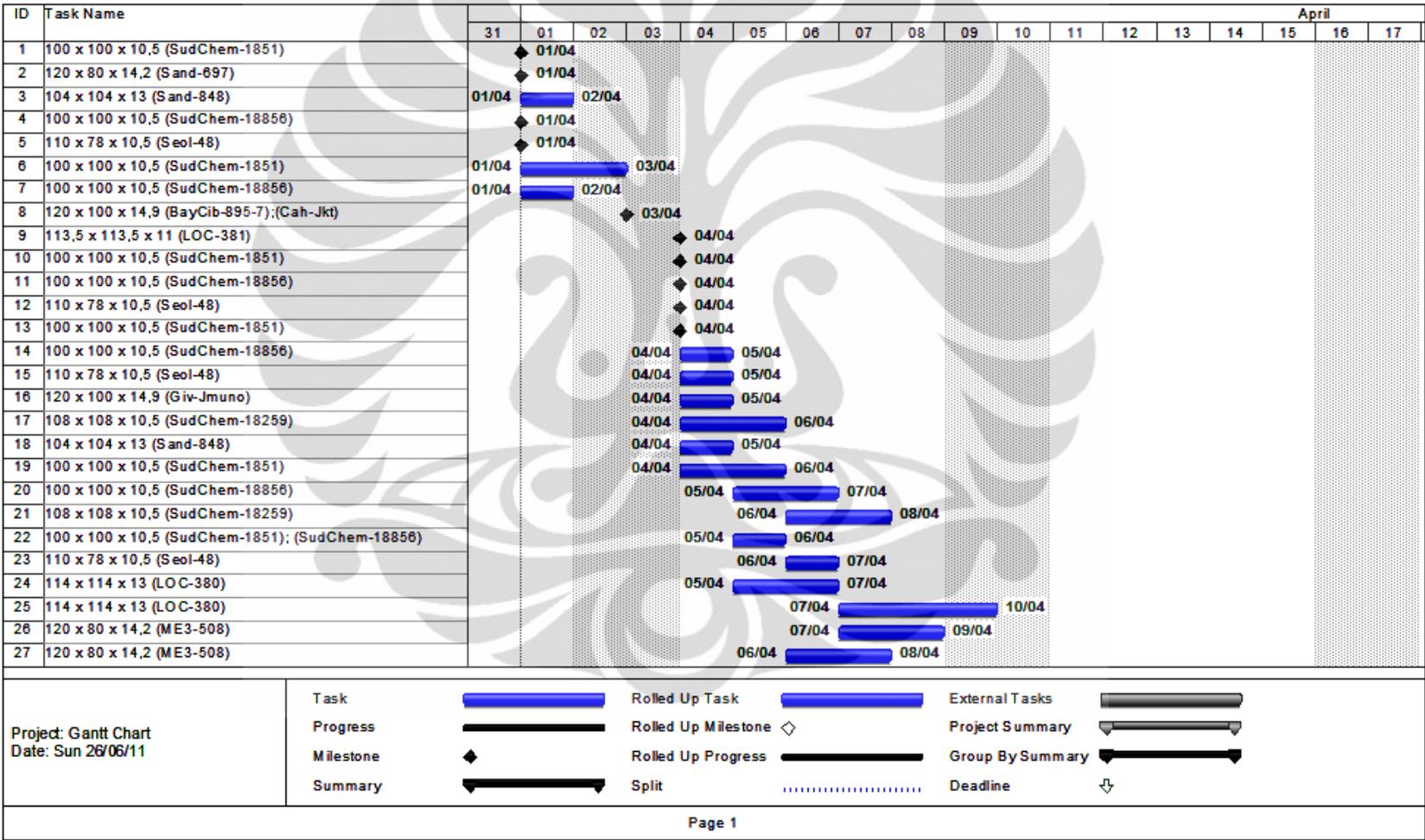
50	datang:	0	datang:	0	datang:	6032	datang:	8496	datang:	0	datang:	8664	datang:	8767
	mulai:	5925	mulai:	8279	mulai:	6032	mulai:	8496	mulai:	0	mulai:	8664	mulai:	8767
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	0	proses:	103	proses:	312
	akhir:	6032	akhir:	8496	akhir:	6138	akhir:	8664	akhir:	0	akhir:	8767	akhir:	9079
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2	
64	datang:	0	datang:	0	datang:	6139	datang:	8713	datang:	8984	datang:	8881	datang:	0
	mulai:	6032	mulai:	8496	mulai:	6139	mulai:	8713	mulai:	8984	mulai:	8881	mulai:	0
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	312	proses:	103	proses:	0
	akhir:	6139	akhir:	8713	akhir:	6245	akhir:	8881	akhir:	9296	akhir:	8984	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	rakit 1	Komp:	-	
42	datang:	0	datang:	0	datang:	6246	datang:	8930	datang:	0	datang:	9098	datang:	9201
	mulai:	6139	mulai:	8713	mulai:	6246	mulai:	8930	mulai:	0	mulai:	9098	mulai:	9201
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	0	proses:	103	proses:	312
	akhir:	6246	akhir:	8930	akhir:	6352	akhir:	9098	akhir:	0	akhir:	9201	akhir:	9513
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	rakit 2	
43	datang:	0	datang:	0	datang:	6404	datang:	9104	datang:	0	datang:	9247	datang:	0
	mulai:	6246	mulai:	8930	mulai:	6404	mulai:	9104	mulai:	0	mulai:	9247	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	0	proses:	253	proses:	0
	akhir:	6404	akhir:	9104	akhir:	6465	akhir:	9247	akhir:	0	akhir:	9500	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 1	Komp:	-	
56	datang:	0	datang:	0	datang:	6511	datang:	9321	datang:	9489	datang:	9592	datang:	0
	mulai:	6404	mulai:	9104	mulai:	6511	mulai:	9321	mulai:	9489	mulai:	9592	mulai:	0
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	103	proses:	312	proses:	0
	akhir:	6511	akhir:	9321	akhir:	6617	akhir:	9489	akhir:	9592	akhir:	9904	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	rakit 2	Komp:	-	
69	datang:	0	datang:	0	datang:	6575	datang:	9485	datang:	9697	datang:	0	datang:	9619
	mulai:	6511	mulai:	9321	mulai:	6575	mulai:	9485	mulai:	9697	mulai:	0	mulai:	9619
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	180	proses:	0	proses:	78
	akhir:	6575	akhir:	9485	akhir:	6641	akhir:	9619	akhir:	9877	akhir:	0	akhir:	9697
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	-	Komp:	rakit 1	
44	datang:	0	datang:	0	datang:	6639	datang:	9649	datang:	9861	datang:	0	datang:	9783
	mulai:	6575	mulai:	9485	mulai:	6639	mulai:	9649	mulai:	9877	mulai:	0	mulai:	9783
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	180	proses:	0	proses:	78
	akhir:	6639	akhir:	9649	akhir:	6705	akhir:	9783	akhir:	10057	akhir:	0	akhir:	9861
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	-	Komp:	rakit 1	
48	datang:	0	datang:	0	datang:	6746	datang:	9866	datang:	0	datang:	10137	datang:	10034
	mulai:	6639	mulai:	9649	mulai:	6746	mulai:	9866	mulai:	0	mulai:	10137	mulai:	10034
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	0	proses:	312	proses:	103
	akhir:	6746	akhir:	9866	akhir:	6852	akhir:	10034	akhir:	0	akhir:	10449	akhir:	10137
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 2	Komp:	rakit 1	
72	datang:	0	datang:	0	datang:	6904	datang:	10040	datang:	10183	datang:	0	datang:	0
	mulai:	6746	mulai:	9866	mulai:	6904	mulai:	10040	mulai:	10183	mulai:	0	mulai:	0
	proses:	158	proses:	174	proses:	61	proses:	143	proses:	253	proses:	0	proses:	0
	akhir:	6904	akhir:	10040	akhir:	6965	akhir:	10183	akhir:	10436	akhir:	0	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	-	
47	datang:	0	datang:	0	datang:	6968	datang:	10204	datang:	10416	datang:	0	datang:	10338
	mulai:	6904	mulai:	10040	mulai:	6968	mulai:	10204	mulai:	10436	mulai:	0	mulai:	10338
	proses:	64	proses:	164	proses:	66	proses:	134	proses:	180	proses:	0	proses:	78
	akhir:	6968	akhir:	10204	akhir:	7034	akhir:	10338	akhir:	10616	akhir:	0	akhir:	10416
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 2	Komp:	-	Komp:	rakit 1	
59	datang:	0	datang:	0	datang:	7075	datang:	10421	datang:	0	datang:	10692	datang:	10589
	mulai:	6968	mulai:	10204	mulai:	7075	mulai:	10421	mulai:	0	mulai:	10692	mulai:	10589
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	0	proses:	312	proses:	103
	akhir:	7075	akhir:	10421	akhir:	7181	akhir:	10589	akhir:	0	akhir:	11004	akhir:	10692
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	-	Komp:	rakit 2	Komp:	rakit 1	
45	datang:	0	datang:	0	datang:	7182	datang:	10638	datang:	10806	datang:	0	datang:	10909
	mulai:	7075	mulai:	10421	mulai:	7182	mulai:	10638	mulai:	10806	mulai:	0	mulai:	10909
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	103	proses:	0	proses:	312
	akhir:	7182	akhir:	10638	akhir:	7288	akhir:	10806	akhir:	10909	akhir:	0	akhir:	11221
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	-	Komp:	rakit 2	
53	datang:	0	datang:	0	datang:	7289	datang:	10855	datang:	11023	datang:	11126	datang:	0
	mulai:	7182	mulai:	10638	mulai:	7289	mulai:	10855	mulai:	11023	mulai:	11126	mulai:	0
	proses:	107	proses:	217	proses:	106	proses:	168	proses:	103	proses:	312	proses:	0
	akhir:	7289	akhir:	10855	akhir:	7395	akhir:	11023	akhir:	11126	akhir:	11438	akhir:	0
Komp:	papan	Komp:	balok	Komp:	papan	Komp:	balok	Komp:	rakit 1	Komp:	rakit 2	Komp:	-	

(lanjutan)

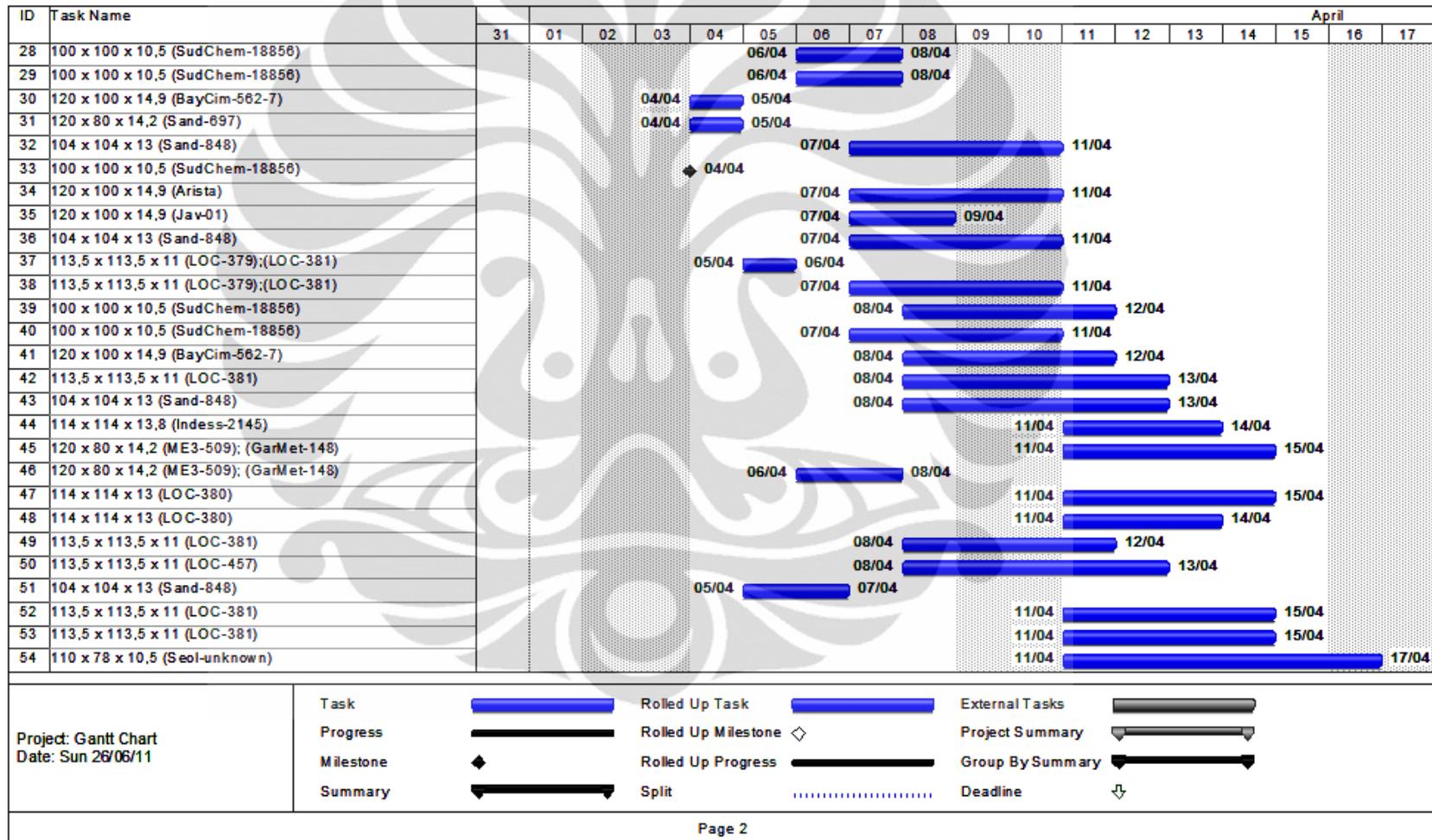
52	datang: 0 mulai: 7289 proses: 64 akhir: 7353 Komp: papan	datang: 0 mulai: 10855 proses: 164 akhir: 11019 Komp: balok	datang: 7353 mulai: 7353 proses: 66 akhir: 7419 Komp: papan	datang: 11019 mulai: 11019 proses: 134 akhir: 11153 Komp: balok	datang: 11153 mulai: 11153 proses: 78 akhir: 11231 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 11231 mulai: 11231 proses: 180 akhir: 11411 Komp: rakit 2
70	datang: 0 mulai: 7353 proses: 64 akhir: 7417 Komp: papan	datang: 0 mulai: 11019 proses: 164 akhir: 11183 Komp: balok	datang: 7417 mulai: 7417 proses: 66 akhir: 7483 Komp: papan	datang: 11183 mulai: 11183 proses: 134 akhir: 11317 Komp: balok	datang: 11317 mulai: 11317 proses: 78 akhir: 11395 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 11395 mulai: 11411 proses: 180 akhir: 11591 Komp: rakit 2
66	datang: 0 mulai: 7417 proses: 158 akhir: 7575 Komp: papan	datang: 0 mulai: 11183 proses: 174 akhir: 11357 Komp: balok	datang: 7575 mulai: 7575 proses: 61 akhir: 7636 Komp: papan	datang: 11357 mulai: 11357 proses: 143 akhir: 11500 Komp: balok	datang: 11500 mulai: 11500 proses: 253 akhir: 11753 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
71	datang: 0 mulai: 7575 proses: 158 akhir: 7733 Komp: papan	datang: 0 mulai: 11357 proses: 174 akhir: 11531 Komp: balok	datang: 7733 mulai: 7733 proses: 61 akhir: 7794 Komp: papan	datang: 11531 mulai: 11531 proses: 143 akhir: 11674 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 11674 mulai: 11674 proses: 253 akhir: 11927 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
61	datang: 0 mulai: 7733 proses: 107 akhir: 7840 Komp: papan	datang: 0 mulai: 11531 proses: 217 akhir: 11748 Komp: balok	datang: 7840 mulai: 7840 proses: 106 akhir: 7946 Komp: papan	datang: 11748 mulai: 11748 proses: 168 akhir: 11916 Komp: balok	datang: 12019 mulai: 12019 proses: 312 akhir: 12331 Komp: rakit 2	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 11916 mulai: 11916 proses: 103 akhir: 12019 Komp: rakit 1
54	datang: 0 mulai: 7840 proses: 158 akhir: 7998 Komp: papan	datang: 0 mulai: 11748 proses: 174 akhir: 11922 Komp: balok	datang: 7998 mulai: 7998 proses: 61 akhir: 8059 Komp: papan	datang: 11922 mulai: 11922 proses: 143 akhir: 12065 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 12065 mulai: 12065 proses: 253 akhir: 12318 Komp: rakit 1	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -
55	datang: 0 mulai: 7998 proses: 78 akhir: 8076 Komp: papan	datang: 0 mulai: 11922 proses: 66 akhir: 11988 Komp: balok	datang: 8076 mulai: 8076 proses: 49 akhir: 8125 Komp: papan	datang: 11988 mulai: 11988 proses: 45 akhir: 12033 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 12033 mulai: 12033 proses: 121 akhir: 12154 Komp: rakit 1
62	datang: 0 mulai: 8076 proses: 64 akhir: 8140 Komp: papan	datang: 0 mulai: 11988 proses: 164 akhir: 12152 Komp: balok	datang: 8140 mulai: 8140 proses: 66 akhir: 8206 Komp: papan	datang: 12152 mulai: 12152 proses: 134 akhir: 12286 Komp: balok	datang: 0 mulai: 0 proses: 0 akhir: 0 Komp: -	datang: 12364 mulai: 12364 proses: 180 akhir: 12544 Komp: rakit 2	datang: 12286 mulai: 12286 proses: 78 akhir: 12364 Komp: rakit 1



LAMPIRAN 7. Gantt Chart Pengerjaan Job



(lanjutan)



(lanjutan)

