

TUGAS AKHIR

**PENJADWALAN *FLOW SHOP* N *JOB* M MESIN DENGAN METODE
FIRST COME FIRST SERVED (FCFS), *EARLIEST DUE DATE* (EDD) DAN
ALGORITMA HEURISTIK POUR**

Diajukan Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Industri



Disusun oleh :
Abdul Latief Irsyad
11660002

**JURUSAN TEKNIK INDUSTRI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2015



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Permohonan persetujuan

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Abdul Latief Irsyad

NIM : 11660002

Judul Skripsi : Penjadwalan *Flow Shop N Job M* Mesin dengan Metode *First Come First Served* (FCFS), *Earliest Due Date* (EDD) dan Algoritma Heuristik Pour

sudah dapat diajukan kembali kepada Program Studi Teknik Industri Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam bidang Teknik Industri

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 2 November 2015

Pembimbing

Dwi Agustina Kurniawati, S.T, M.Eng

NIP.19790806 200604 2 001



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/3597/2015

Skripsi/Tugas Akhir dengan judul : Penjadwalan *Flow Shop N Job M* Mesin dengan Metode *First Come First Served* (FCFS), *Earliest Due Date* (EDD) dan *Algoritma Heuristik Pour*

Yang dipersiapkan dan disusun oleh :

Nama : Abdul Latief Irsyad

NIM : 11660002

Telah dimunaqasyahkan pada : 17 November 2015

Nilai Munaqasyah : A

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Dwi Agustina Kurniawati, S.T, M.Eng
NIP.19790806 200604 2 001

Penguji I

Fauziq Aji, M.T
NIP.19800715 200604 1 002

Penguji II

Syaeful Arief, M.T

Yogyakarta, 20 November 2015

UIN Sunan Kalijaga

Fakultas Sains dan Teknologi

Dekan



Dr. Maizer Said Nahdi, M.Si
NIP. 19550427 198403 2 001

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Abdul Latief Irsyad
NIM : 11660002
Jurusan : Teknik Industri
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya dan sejujurnya bahwa skripsi saya yang berjudul **PENJADWALAN *FLOW SHOP* N *JOB* M MESIN DENGAN METODE *FIRST COME FIRST SERVED* (FCFS), *EARLIEST DUE DATE* (EDD) DAN ALGORITMA HEURISTIK POUR** adalah asli dari penelitian saya sendiri dan bukan plagiasi karya orang lain.

Yogyakarta, 3 November 2015



menyatakan,

Abdul Latief Irsyad

NIM. 11660002

HALAMAN PERSEMBAHAN

Tugas Akhir ini kupersembahkan kepada kedua orang tuaku

Suyitno dan Sudiyah

Kalian adalah penggerak serta semangatku dalam menjalani kehidupan, Anakmu hanya dapat menyampaikan kata “Terima Kasih” atas semua pengorbanan serta kasih sayang yang telah kalian berikan

*“Maaf” atas semua kesalahan dan kekurangan anakmu
Semoga Ibu & Bapak senantiasa dalam kebaikan, kesehatan,
serta rahmat dan hidayah Allah ta’ala.*

*Kepada kakak-kakaku, Esti Mindrati, Siti Hajar Rochmah, Tri
Wibawayanti dan Aghniyatun Nafingah*

Terimakasih atas semua dukungan dan motivasi yang selama ini kalian berikan. .

Serta kepada seluruh Anggota Keluarga, penulis hanya dapat mengucapkan terima kasih atas semua support baik moril maupun materil, semoga senantiasa mendapatkan kebaikan Dunia dan Akhirat.

HALAMAN MOTTO

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap”

(QS. Al Insyirah : 6-8)

“Hai orang-orang yang beriman, apabila dikatakan kepadamu : “Berlapang-lapanglah dalam majelis”, maka lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan : “Berdirilah kamu, maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan.”

(QS. Al Mujadillah : 11)

“Sesungguhnya Allah menyukai hamba yang berkarya dan terampil (ahli/professional). Barang siapa bersusah-payah mencari nafkah untuk keluarganya, maka nilainya sama dengan seorang mujahid di jalan Allah SWT.”-Hadist Nabi

(HR. Ahmad)

Jadilah seperti karang di lautan yang tetap kokoh diterjang ombak, walaupun demikian air laut tetap masuk kedalam pori-porinya.

KATA PENGANTAR

Bismillahirrahmanirrahiim

Segala puji hanya bagi Allah Rabb semesta alam yang telah memberikan kesempatan, kesehatan serta hidayahnya sehingga Tugas Akhir ini dapat diselesaikan. Shalawat dan salam tercurah kepada Sang Uswatun Hasanah, Rasulullah Muhammad SAW. serta kepada *Ahulul Bait* beliau, sahabat, serta kepada mereka yang senantiasa tetap teguh berada dibelakang beliau.

Dalam penyusunan Tugas Akhir ini tentunya melibatkan banyak pihak yang telah memberikan bantuan demi terselesaikannya laporan ini, baik bantuan secara metri, pendampingan, doa dan motivasi. Untuk itu ucapan terimakasih saya haturkan kepada :

1. Bapak dan Ibu tercinta, tak ada kata yang dapat mengungkapkan rasa terima kasih atas pengorbanan serta kasih sayang mereka.
2. Ibu Dwi Agustina Kurniawati, S.T., M.Eng selaku dosen pembimbing yang selalu memberikan arahan dan bimbingan sampai terselesainya Tugas Akhir ini.
3. Segenap Staf Pengajar dan karyawan Program Studi Teknik Industri UIN Sunan Kalijaga yang telah memberikan ilmu yang bermanfaat kepada penulis.
4. Teman-teman Teknik Industri khususnya angkatan 2011, semoga Allah selalu memberikan kemudahan. Jaga selalu kekompakan kita..
5. Teman-teman Ikatan Mahasiswa Muhammadiyah UIN Sunan Kalijaga yang telah memberikan pengalaman organisasi dan kebersamaan.

6. Sahabat serta orang terdekat Teguh Tri Prasetyo, Muhammad Iqbal Masardie dan Nursaadah Syakbani yang selalu memberikan motivasi dan dukungan.
7. Serta teman-teman yang telah memberikan bantuan kepada penulis, penulis ucapkan Jazakumullahu khairan Katsira.

Penulis menyadari banyak pihak lain yang turut mendukung baik moril maupun materil dalam Tugas Akhir ini yang tidak dapat disebutkan satu persatu, penulis mengucapkan terima kasih.

Yogyakarta, 5 November 2015

Penulis,

Abdul Latief Irsyad

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
SURAT PERNYATAAN KEASLIAN	iv
HALAMAN PERSEMBAHAN	v
MOTTO	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
ABSTRAK	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Tujuan Penelitian	7
1.4 Manfaat	7
1.5 Batasan Masalah dan Asumsi	8
1.5.1 Batasan Masalah	8
1.5.2 Asumsi	8
1.6 Sistematika Penulisan	9

BAB II TINJAUAN PUSTAKA	11
2.1 Posisi Penelitian	11
2.2 Definisi dalam Penjadwalan	19
2.3 Kriteria dalam Penjadwalan	22
2.4 Jenis-jenis Penjadwalan	23
2.4.1 Penjadwalan <i>Flow Shop</i>	23
2.4.2 Penjadwalan <i>Job Shop</i>	25
2.5 Metode dalam Penjadwalan Produksi	27
2.5.1 Penjadwalan Produksi dengan Metode FCFS (<i>First Come First Serve</i>)	27
2.5.2 Penjadwalan Produksi dengan Metode EDD (<i>Early Due Date</i>)	27
2.5.3 Penjadwalan Produksi dengan Metode SPT (<i>Shortest Processing Time</i>)	29
2.5.4 Penjadwalan Produksi dengan Metode Algoritma Heuristik Pour ...	31
2.6 Parameter Performansi	36
2.7 Metode Enumerasi	36
BAB III METODOLOGI PENELITIAN	38
BAB IV HASIL DAN PEMBAHASAN	43
4.1 <i>Data Random Number Generator</i>	43
4.2 Hasil Penelitian	43
4.2.1 Hasil Penjadwalan Produksi dengan Metode Enumerasi	43
4.2.2 Hasil Penjadwalan Produksi dengan Metode <i>First Come First Serve</i> (FCFS)	49

4.2.3 Hasil Penjadwalan Produksi dengan Metode <i>Early Due Date</i> (EDD).....	54
4.2.4 Hasil Penjadwalan Produksi dengan Metode Algoritma Heuristik Pour	60
4.2.5 Perbandingan Rata-rata Makespan pada Metode <i>First Come First Serve</i> (FCFS), <i>Early Due Date</i> (EDD) dan Metode Heuristik Pour.....	65
4.2.6 Perhitungan <i>Efficiency Index</i> (EI)	67
4.2.7 Perhitungan <i>Relative Error</i> (RE)	68
4.2.8 Perbandingan <i>Run Time</i> pada Metode <i>First Come First Serve</i> (FCFS), <i>Early Due Date</i> (EDD) dan Metode Algoritma Heuristik Pour	69
4.2.9 Perbandingan Nilai Deviasi Rata-rata Makespan pada Metode <i>First Come First Serve</i> (FCFS), <i>Early Due Date</i> (EDD) dan Metode Algoritma Heuristik Pour	69
4.3 Pembahasan	71
BAB V PENUTUP	74
5.1 Kesimpulan	74
5.2 Saran	75
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1. Perbandingan Penelitian	16
Tabel 2.2. Penjadwalan Metode EDD	28
Tabel 2.3. Perhitungan <i>Makespan</i> Metode EDD	29
Tabel 2.4. Penjadwalan Metode SPT.....	31
Tabel 2.5. Penjadwalan Metode Algoritma Heuristik Pour.....	33
Tabel 2.6. Perhitungan <i>Completion Time</i>	33
Tabel 2.7. Perhitungan <i>Makespan</i> (F_{\max}) Metode Pour.....	34
Tabel 2.8. Perhitungan <i>Makespan</i> Metode Pour.....	35
Tabel 3.1. Kombinasi N <i>Job</i> M Mesin Dengan Proses Random.....	38
Tabel 4.1. <i>Makespan</i> Optimal dari Metode Enumerasi.....	44
Tabel 4.2. <i>Makespan</i> dari Metode FCFS.....	50
Tabel 4.3. <i>Makespan</i> dari Metode EDD.....	55
Tabel 4.4. <i>Makespan</i> Metode Pour	61
Tabel 4.5. Perbandingan <i>Makespan</i> rata-rata pada Metode Enumerasi, FCFS, EDD dan Pour	66
Tabel 4.6. Perbandingan <i>Run Time</i> pada Metode FCFS, EDD dan Pour.....	69
Tabel 4.7. Perbandingan Nilai Deviasi <i>Makespan</i> rata-rata antara Metode FCFS, EDD dan Algoritma Heuristik Pour	70

DAFTAR GAMBAR

Gambar 2.1. Alur Penjadwalan <i>Flow shop</i>	24
Gambar 2.2. Alur Penjadwalan <i>Job shop</i>	26
Gambar 2.3. Pasangan <i>Job</i> yang ditukar.....	28
Gambar 2.4. Aturan SPT.....	30
Gambar 3.1. <i>A Steady-State Analysis of the Model</i>	39
Gambar 3.2. Diagram Alir Penelitian.....	40
Gambar 4.1. <i>Makespan</i> Optimal Metode Enumerasi.....	49
Gambar 4.2. <i>Makespan</i> Metode FCFS.....	54
Gambar 4.3. <i>Makespan</i> Metode EDD.....	60
Gambar 4.4. <i>Makespan</i> Metode Pour.....	65
Gambar 4.5. Perbandingan <i>Makespan</i> rata-rata antara Metode Enumerasi, FCFS, EDD dan Algoritma Heuristik Pour.....	66
Gambar 4.6. Grafik Perbandingan <i>Makespan</i> pada Metode Enumerasi, FCFS, EDD dan Algoritma Heuristik Pour.....	67

DAFTAR LAMPIRAN

- Lampiran 1. Data waktu proses 300 *run*
- Lampiran 2. Data *due date* 300 *run*
- Lampiran 3. *Flow Chart* Metode Enumerasi
- Lampiran 4. *Coding/Program* Matlab Metode Enumerasi
- Lampiran 5. *Flow Chart* Metode FCFS
- Lampiran 6. *Coding/Program* Matlab Metode FCFS
- Lampiran 7. *Flow Chart* Metode EDD
- Lampiran 8. *Coding/Program* Matlab Metode EDD
- Lampiran 9. *Flow Chart* Metode Algoritma Heuristik Pour
- Lampiran 9.1. *Flow Chart* Program Iterasi 1 (A)
- Lampiran 9.2. *Flow Chart* Program Iterasi 1 (B pada *makespan* 1)
- Lampiran 9.3. *Flow Chart* Program Iterasi 1 (B pada *makespan* 2)
- Lampiran 9.4. *Flow Chart* Program Iterasi 1 (C pada *makespan* 1)
- Lampiran 9.5. *Flow Chart* Program Iterasi 1 (C pada *makespan* 2)
- Lampiran 9.6. *Flow Chart* Program Iterasi 1 (C pada *makespan* 3)
- Lampiran 9.7. *Flow Chart* Program Iterasi 1 (D pada *makespan* 1)
- Lampiran 9.8. *Flow Chart* Program Iterasi 1 (D pada *makespan* 2)
- Lampiran 9.9. *Flow Chart* Program Iterasi 1 (D pada *makespan* 3)
- Lampiran 9.10. *Flow Chart* Program Iterasi 1 (D pada *makespan* 4)
- Lampiran 9.11. *Flow Chart* Program Iterasi 2 (A1)
- Lampiran 9.12. *Flow Chart* Program Iterasi 2 (A2)

Lampiran 9.13. *Flow Chart* Program Iterasi 2 (A3)

Lampiran 9.14. *Flow Chart* Program Iterasi 2 (B1)

Lampiran 9.15. *Flow Chart* Program Iterasi 2 (B2)

Lampiran 9.16. *Flow Chart* Program Iterasi 2 (B3)

Lampiran 9.17. *Flow Chart* Program Iterasi 2 (C1)

Lampiran 9.18. *Flow Chart* Program Iterasi 2 (C2)

Lampiran 9.19. *Flow Chart* Program Iterasi 2 (C3)

Lampiran 9.20. *Flow Chart* Program Iterasi 2 (D1)

Lampiran 9.21. *Flow Chart* Program Iterasi 2 (D2)

Lampiran 9.22. *Flow Chart* Program Iterasi 2 (D3)

Lampiran 10. *Coding/Program* Matlab Metode Pour

Lampiran 10.1. *Function* Program Iterasi 1 (A)

Lampiran 10.2. *Function* Program Iterasi 1 (B)

Lampiran 10.3. *Function* Program Iterasi 1 (C)

Lampiran 10.4. *Function* Program Iterasi 1 (D)

Lampiran 10.5. *Function* Program Iterasi 2 (A1)

Lampiran 10.6. *Function* Program Iterasi 2 (A2)

Lampiran 10.7. *Function* Program Iterasi 2 (A3)

Lampiran 10.8. *Function* Program Iterasi 2 (B1)

Lampiran 10.9. *Function* Program Iterasi 2 (B2)

Lampiran 10.10. *Function* Program Iterasi 2 (B3)

Lampiran 10.11. *Function* Program Iterasi 2 (C1)

Lampiran 10.12. *Function* Program Iterasi 2 (C2)

Lampiran 10.13. *Function* Program Iterasi 2 (C3)

Lampiran 10.14. *Function* Program Iterasi 2 (D1)

Lampiran 10.15. *Function* Program Iterasi 2 (D2)

Lampiran 10.16. *Function* Program Iterasi 2 (D3)

Lampiran 10.17. *Function* Mengurutkan 2 *Job*

Lampiran 10.18. *Function* Mengurutkan 3 *Job*

Lampiran 10.19. *Function* Mengurutkan 4 *Job*

Lampiran 10.20. Mengambil Data Terkecil untuk 2 *Job*

Lampiran 10.21. Mengambil Data Terkecil untuk 3 *Job*

Lampiran 10.22. Mengambil Data Terkecil untuk 4 *Job*

Lampiran 11. Rekapitulasi Hasil Perhitungan Metode Enumerasi

PENJADWALAN *FLOW SHOP* N *JOB* M MESIN DENGAN METODE *FIRST COME FIRST SERVED* (FCFS), *EARLIEST DUE DATE* (EDD) DAN ALGORITMA HEURISTIK POUR

Abdul Latief Irsyad

11660002

Program Studi Teknik Industri Fakultas Sains dan Teknologi
Universitas Islam Negeri Sunan Kalijaga Yogyakarta

Abstrak

Penelitian ini membahas tentang penjadwalan *flow shop* n *job* m mesin dengan pendekatan metode *First Come First Served* (FCFS), *Earliest Due Date* (EDD) dan Algoritma Heuristik Pour. Tujuan penelitian ini adalah melakukan perbandingan performansi antara ketiga metode tersebut untuk meminimalkan *makespan*. Dalam penelitian ini data waktu proses dan *due date* yang digunakan merupakan data random yang dibangkitkan dengan *software* Matlab 7.1. Kombinasi jumlah *job* dan mesin yaitu 4 *job* dan 2 mesin.

Penelitian ini dilakukan dalam dua tahap. Tahap pertama yaitu melakukan perhitungan menggunakan metode enumerasi. Tahap kedua yaitu melakukan perhitungan dengan metode FCFS, EDD dan Pour. Parameter performansi yang digunakan yaitu *makespan* rata-rata, *Efficiency Index* (EI), *Relative Error* (RE), *run time*, dan nilai deviasi *makespan* rata-rata. Hasil dari penelitian ini menunjukkan bahwa metode terbaik untuk penjadwalan *flow shop* n *job* m mesin dalam meminimalkan *makespan* adalah Algoritma Heuristik Pour karena memiliki nilai *makespan* rata-rata terkecil yaitu 30,70 *time unit*, nilai $EI < 1$, nilai RE lebih dari 5% dan memiliki nilai deviasi yang terkecil yaitu 0,35 *time unit*. Meskipun metode Algoritma Heuristik Pour memiliki nilai *run time* yang paling lama akan tetapi nilai *run time* yang diperoleh masih dapat ditoleransi yaitu sebesar 2.741351 detik.

Kata kunci : *Penjadwalan, flow shop, enumerasi, FCFS, EDD, Pour, makespan rata-rata, efficiency index, relative error, run time*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Proses produksi merupakan serangkaian proses untuk menghasilkan barang atau jasa. Pada proses produksi inilah terdapat berbagai macam kegiatan seperti proses perencanaan produksi, penentuan kualitas, penjadwalan kegiatan produksi, sampai pengolahan limbah yang dihasilkan selama proses produksi berlangsung. Semua proses produksi tersebut sangat penting, salah satunya adalah penjadwalan kegiatan produksi.

Menurut Ginting (2009), penjadwalan adalah pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin. Penjadwalan merupakan alat ukur yang baik bagi perencanaan agregat. Pesanan-pesanan aktual pada tahap ini akan ditugaskan pertamakalinya pada sumber daya tertentu (fasilitas, pekerja dan peralatan), kemudian dilakukan pengurutan kerja pada tiap-tiap pemrosesan sehingga dicapai optimalitas utilitas kapasitas yang ada. Pada penjadwalan ini, permintaan akan produk-produk tertentu (jenis dan jumlah) dari *Master Production Schedule* (MPS) akan ditugaskan pada pusat-pusat pemrosesan tertentu untuk periode harian (Nasution dan Prasetyawan, 2008).

Jika terdapat n *job* dan m mesin dengan setiap *job* akan melewati semua mesin yang ada maka kondisi ini termasuk dalam kategori penjadwalan *flow shop*. Menurut Baker (2009), penjadwalan *flow shop* adalah salah satu jenis penjadwalan produksi dimana setiap n *job* akan melalui setiap m mesin dengan urutan yang seragam.

Masalah penjadwalan yang sering muncul yaitu terjadi kesulitan dalam menentukan *job* mana yang harus dikerjakan terlebih dahulu agar mendapatkan hasil yang optimal. Metode atau cara untuk menentukan *job* mana yang harus dikerjakan terlebih dahulu dalam penjadwalan *flow shop* adalah dengan aturan prioritas. Menurut Heizer dan Render (2010) aturan prioritas (*priority rule*) memberikan panduan untuk mengurutkan pekerjaan yang harus dilakukan. Aturan ini terutama diterapkan untuk fasilitas yang terfokus pada proses. Aturan prioritas mencoba untuk meminimalkan waktu penyelesaian jumlah pekerjaan dalam sistem, dan keterlambatan pekerjaan seraya memaksimalkan penggunaan fasilitas.

Menurut Bedworth (1987) dalam Rasjidin dan Hidayat (2006) metode-metode yang digunakan untuk menentukan *job* mana yang dikerjakan terlebih dahulu, yaitu : *Short Processing Time* (SPT), *Weighted Short Processing Time* (WSPT), *First Come First Served* (FCFS), *Earliest Due Date* (EDD), *Slack Time* (ST), *Longest Processing Time* (LPT), *Random Selection* (RS), *Highest Expected Profitability* (HEP), *Preferred Customer Order* (PCO), dan Algoritma Hodgson.

Aturan *First Come First Served* (FCFS) yaitu pekerjaan yang datang pertama kali pada stasiun kerja akan diproses terlebih dahulu. Aturan FCFS merupakan aturan prioritas yang paling banyak digunakan oleh perusahaan. Novriansyah (2013), melakukan penelitian di PT Karya Makmur Armada, perusahaan tersebut menggunakan metode FCFS pada penjadwalan produksinya. Sulaksmi et al. (2014) melakukan penelitian di Konveksi *One Way* Malang, perusahaan tersebut juga menggunakan metode FCFS dalam proses penjadwalan produksinya. Kuncoro (2013) melakukan penelitian di PT. Kertas Leces Probolinggo, perusahaan tersebut juga menggunakan metode FCFS pada penjadwalan produksinya.

Sedangkan aturan prioritas *Earliest Due Date* (EDD) merupakan aturan prioritas yang menggunakan data *due date* sebagai dasar dalam penjadwalannya. Menurut aturan ini *due date* yang terkecil akan dikerjakan terlebih dahulu. Dalam kehidupan sehari-hari aturan EDD ini juga banyak diterapkan pada perusahaan.

Dalam perkembangan selanjutnya banyak teknik penjadwalan dengan menggunakan metode heuristik dan metaheuristik. Beberapa metode heuristik yang telah digunakan untuk menyelesaikan permasalahan penjadwalan *flow shop* adalah Algoritma Pour, Algoritma Campbell, Dudek dan Smith (CDS), Algoritma Palmer, Algoritma Nawaz, Enscore dan Ham (NEH), Algoritma Gupta, Algoritma Dannenbring. Adapun metode metaheuristik yang telah digunakan adalah *Tabu Search*, *Genetic Algorithm*, *Simulated Annealing* dll.

Hamman (2015) dalam penelitian tugas akhirnya menyimpulkan bahwa algoritma heuristik CDS memiliki performansi yang lebih baik dari algoritma Palmer, Dannenbring dan Ignall-Scharge dalam meminimalkan *makespan* pada sebuah lintasan produksi *flow shop* n job m mesin di CV. Bonjor Jaya.

Penelitian Soetanto dan Palit (2004) menunjukkan bahwa algoritma heuristik Pour memberikan *performansi* yang cukup baik dalam menyelesaikan permasalahan penjadwalan *flow shop* untuk meminimalkan *makespan* dibandingkan dengan metode optimasi *Mixed Integer Programming* (MIP).

Penelitian yang dilakukan oleh Kuncoro (2013) menunjukkan bahwa algoritma heuristik Pour memiliki performansi yang sama dengan algoritma Nawaz, Ensore dan Ham (NEH) dengan *makespan* minimum dan kompleksitas waktu yang sama.

Banyak peneliti yang telah melakukan *computational study* atau studi komputasi untuk menentukan performansi masing-masing metode penjadwalan dan untuk membandingkan performansi dari beberapa metode penjadwalan tersebut. Dari *computational study* yang telah dilakukan akan didapatkan metode penjadwalan terbaik sesuai dengan parameter yang telah ditentukan diawal penelitian, misalnya *makespan* terkecil dan *lateness* terkecil.

Arisha et al. (2002) melakukan penelitian dengan program simulasi untuk menemukan metode penjadwalan yang optimum/mendekati optimum

dengan tujuan untuk meminimalkan *makespan*. Dalam penelitian tersebut dilakukan perbandingan performansi dari lima aturan prioritas yaitu FCFS, SPT, LPT, SRPT dan LRPT. Data yang digunakan dalam penelitian tersebut merupakan data random yang dibangkitkan menggunakan *software* pada komputer.

Bancila dan Buzatu (2014) melakukan penelitian untuk mengetahui performansi metode *Hybrid Algorithm* dan *Standard Genetic Algorithm* untuk meminimalkan *makespan* pada penjadwalan *flow shop*. Mereka melakukan studi komputasi dimana data yang digunakan dalam penelitian mereka adalah data random yang dibangkitkan oleh komputer.

Hejazi et al. (2009) melakukan studi komputasi pada penjadwalan *flow shop* dua mesin dengan *fuzzy processing time* untuk meminimalkan *makespan* menggunakan algoritma heuristik. Penelitian Hejazi et al. (2009) juga menggunakan data random. Kombinasi *job* yang digunakan adalah 3, 4, 5, 6, dan 7 *job*.

Modrak dan Pandian (2010) juga melakukan studi komputasi untuk membandingkan performansi Algoritma CDS, *Slope Index*, *Genetic Algorithm* dan *Proposed Approach*. Untuk mengetahui performansi keempat metode tersebut, Modrak dan Pandian (2010) juga menggunakan data random dengan beberapa kombinasi jumlah *job* dan mesin.

Pour (2001) mengusulkan metode heuristik baru yang diberi nama Algoritma Pour untuk menyelesaikan permasalahan penjadwalan *flow shop* n *job* m mesin. Dalam penelitian tersebut, metode heuristik Pour

dibandingkan dengan metode CDS, NEH dan Palmer. Pour (2001) juga melakukan studi komputasi dimana data yang digunakan untuk mengetahui performansi keempat metode tersebut menggunakan data random yang dibangkitkan dengan komputer.

Berdasarkan hal-hal di atas, maka dalam tugas akhir ini akan dilakukan *computational study* (studi komputasi) untuk membandingkan performansi dari metode FCFS, EDD dan Pour dalam menentukan penjadwalan *flow shop* n *job* m mesin. Penelitian ini bertujuan untuk meminimalkan *makespan*, dimana data yang digunakan dalam studi komputasi ini adalah data random yang dibangkitkan oleh komputer.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas rumusan masalah dalam penelitian ini adalah :

- a. Bagaimana perbandingan performansi metode FCFS, EDD dan algoritma heuristik Pour dalam penjadwalan *flow shop* n *job* m mesin untuk meminimalkan *makespan* ?
- b. Apakah metode terbaik untuk penjadwalan *flow shop* n *job* m mesin dalam meminimalkan *makespan* ?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

- a) Melakukan studi komputasi dengan menggunakan data random yang dibangkitkan dengan komputer pada metode FCFS, EDD dan algoritma heuristik Pour dalam menyelesaikan permasalahan pada penjadwalan *flow shop* untuk meminimalkan *makespan*.
- b) Membandingkan performansi penjadwalan *flow shop* n *job* m mesin pada metode FCFS, EDD dan algoritma heuristik Pour.
- c) Mengetahui performansi penjadwalan terbaik untuk penjadwalan n *job* m mesin diantara FCFS, EDD dan algoritma Pour dalam meminimalkan *makespan*.

1.4. Manfaat Penelitian

Manfaat penelitian yang didapat adalah sebagai berikut :

- a. Mengetahui metode terbaik untuk penjadwalan *flow shop* n *job* m mesin dalam meminimalkan *makespan*.
- b. Memberikan usulan metode penjadwalan terbaik untuk penjadwalan produksi perusahaan.

1.5. Batasan Masalah dan Asumsi

Batasan masalah dan asumsi yang digunakan dalam penelitian tugas akhir ini adalah :

1.5.1 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

- a. Data waktu proses *job* pada mesin dan *due date* setiap *job* dibangun secara random.
- b. Pada penelitian ini penjadwalan *flow shop* n *job* m mesin dibatasi hanya untuk permasalahan 4 *job* dan 2 mesin (data skala kecil).
- c. Studi komputasi dalam penelitian ini dilakukan dengan membuat program pada *software* Matlab versi 7.1.

1.5.2 Asumsi

Asumsi-asumsi yang digunakan dalam penelitian ini yaitu :

- a. Waktu *set-up* mesin diabaikan.
- b. Tidak terdapat *preemption*.
- c. Semua *job* memiliki *ready time* yang sama.
- d. Waktu transportasi diabaikan.
- e. Mesin selalu dalam kondisi baik/dapat beroperasi normal.

1.6. Sistematika Penulisan

Sistematika penulisan dalam tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab I berisi uraian latar belakang penelitian, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan asumsi, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab II berisi beberapa penelitian terdahulu untuk melihat perbandingan tujuan, metode dan hasil analisa. Bab ini juga mencakup segala hal yang dapat dijadikan dasar bagi tema penelitian. Seperti beberapa definisi dalam penjadwalan, kriteria dalam penjadwalan, jenis-jenis penjadwalan, penjadwalan *flow shop*, penjadwalan *job shop*, penjadwalan produksi dengan metode FCFS, EDD, SPT dan algoritma Pour, parameter performansi, dan metode enumerasi.

BAB III METODOLOGI PENELITIAN

Bab III berisi penentuan jumlah *job* dan mesin yang digunakan dalam penelitian, penentuan interval data waktu proses dan *due date*, penentuan *number of run* dalam penelitian dan kerangka alir penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab IV berisi tentang hasil perhitungan *makespan* dengan 4 metode yaitu metode enumerasi, metode FCFS, metode EDD dan metode Algoritma Heuristik Pour yang ditampilkan pada tabel dan grafik, hasil perhitungan parameter performansi *Efficiency Index* (EI) dan *Relative Error* (RE) antara FCFS dengan Pour dan EDD dengan Pour, *run time* dan nilai deviasi.

BAB V KESIMPULAN DAN SARAN

Bab V terdiri dari hasil kesimpulan dari pembahasan terhadap metode FCFS, EDD dan Pour berdasarkan hasil *makespan* rata-rata dan parameter performansi *Efficiency Index* (EI), *Relative Error* (RE), *run time* dan nilai deviasi. Bab V juga berisi saran bagi penelitian selanjutnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan dapat disimpulkan bahwa :

	Parameter performansi	FCFS	EDD	Pour
1.	<i>Makespan</i> rata-rata	33,83 <i>time unit</i>	33,29 <i>time unit</i>	30,70 <i>time unit</i>
2.	<i>Run Time</i>	0.077761 detik	0.151779 detik	2.741351 detik
3.	Deviasi <i>makespan</i> rata-rata	3,48 <i>time unit</i>	2,94 <i>time unit</i>	0,35 <i>time unit</i>

1. Nilai *Efficiency Index* (EI) antara metode FCFS dengan metode Pour adalah 0,9075. Nilai *Efficiency Index* (EI) antara metode EDD dengan metode Pour adalah 0,9222. Karena nilai $EI < 1$ maka metode algoritma heuristik Pour memiliki performansi yang lebih baik dari pada metode FCFS dan EDD.
2. Nilai *Relative Error* (RE) antara metode FCFS dan metode Pour adalah 10,19 %. Nilai *Relative Error* (RE) antara metode EDD dan metode Pour adalah 8,44 %. Karena hasil perhitungan parameter RE tersebut didapatkan nilai cukup besar atau lebih dari 5% maka menunjukkan bahwa penggunaan metode FCFS dan EDD akan memiliki perbedaan nilai *makespan* yang cukup besar dibandingkan dengan metode algoritma heuristik Pour.

3. Dalam penelitian ini metode terbaik untuk penjadwalan *flow shop* n job m mesin dalam meminimalkan *makespan* adalah Algoritma Heuristik Pour. Metode algoritma heuristik Pour lebih baik karena memiliki nilai *makespan* rata-rata terkecil yaitu 30,70 *time unit*, nilai $EI < 1$, nilai RE lebih dari 5%, dan nilai deviasi yang terkecil yaitu 0,35 *time unit*.
4. Meskipun metode Algoritma Heuristik Pour memiliki nilai *run time* yang paling lama akan tetapi nilai *run time* yang diperoleh masih dapat ditoleransi yaitu sebesar 2.741351 detik.

5.2 Saran

1. Untuk penelitian selanjutnya perlu dilakukan penelitian dalam skala sedang dan skala besar terkait dengan perbandingan performansi metode Pour, FCFS dan EDD.
2. Untuk penelitian selanjutnya perlu dilakukan investigasi dan eksperimen metode heuristik dan metaheuristik yang lain untuk mendapatkan metode penjadwalan *flow shop* yang lebih baik. Metode heuristik dan metode metaheuristik tersebut misalnya metode CDS, *Tabu Search*, *Ant Colony*, Palmer dll .

DAFTAR PUSTAKA

- Arisha, A., Young, P., El Baradie, M., 2002. *Flow Shop Schedulling Problem : a Computational Study*. Sixth International Conference on Production Engineering and Design for Development (PEDD6), Cairo Egypt, pp 543-557.
- Baker, K.R. dan Trietsch, D., 2009. *Principles of Sequencing and Scheduling*. Canada : John Willey and Son
- Bancila, D. dan Buzatu, C., 2014. *Minimizing Makespan In A Flow Shop Schedulling Problem*. Transilvania University of Brasov, Vol. 15, No. 3. pp.43.
- Bedworth, D.D. 1987, *Integrated Prooiluction Control Systems, Management, Analysis, Design 2/E*, John Wiley & Sons, New York.
- Ginting, R., 2009. *Perancangan Produk*. Medan : Graha Ilmu.
- Hamman, M.K., 2015. *Penjadwalan Produksi Flow Shop untuk Meminimalkan Makespan dengan Metode Campbell, Dudek and Smith (CDS), Metode Palmer, Metode Dannenbring dan Metode Ignall-Scharge (Studi Kasus di CV. Bonjor Jaya, Klaten)*. S-1. Teknik Industri, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta.
- Heizer, J. dan Render, B., 2010. *Manajemen Operasi*. Jakarta : Salemba Empat.
- Hejazi, S.R., Emami, S., Arkan, A., 2009. *A Heuristic Algorithm for Minimizing the Expected in Two-Machine Flow Shop with Fuzzy Processing Time*.

- Department of Industrial & Systems Engineering, Isfahan University of Technology, Vol. 3, No. 2, pp. 114-122.
- Kuncoro, C., 2013. *Penjadwalan Produksi Kertas Menggunakan Algoritma Pour dan Algoritma NEH di PT. Kertas Leces Probolinggo*. S-1. Matematika, Universitas Jember.
- Kusuma, H., 2009. *Perancangan dan Pengendalian Produksi*. Yogyakarta : Penerbit ANDI.
- Modrak, V. dan Pandian, R.S., 2010. *Flow Shop Schedulling Algorithm to Minimize Completion Time For n Job m Machine Problem*. Technical Gazette, Vol. 17, pp. 273-278.
- Nasution, A. H. dan Prasetyawan, Y., 2008. *Perancangan dan Pengendalian Produksi*. Yogyakarta : Graha Ilmu.
- Novriansyah, R., 2013. *Peningkatan Perencanaan Penjadwalan Produksi dengan Metode Campbell Dudek and Smith*. S-1. Teknik Industri, Universitas Bina Darma Palembang.
- Pour, H. D., 2001. *A New Heuristic for n-Job m-Machine Flowshop Problem*. Production Planning Control, Vol. 12, No. 7, 648-653.
- Rasjidin, R. dan Hidayat, I., 2006. *Penjadwalan Produksi Mesin Injection Moulding pada PT. Duta Flow Plastic Machinery*. Jurnal Inovasi, Vol. 5, No. 2, pp. 52-57.
- Riska. 2013. *Penerapan Kombinatorial pada Tanda Nomor Kendaraan Bermotor (TNKB) Kota Bandung*. S-1. Teknik Informatika, Institut Teknologi Bandung.

- Soetanto, T.V. dan Palit, H.C., 2004. *Studi Perbandingan Performance Algoritma Heuristik Pour terhadap Mixed Integer Programming dalam Menyelesaikan Penjadwalan Flowshop*. Jurnal Teknik Industri Vol. 6, No. 1, pp. 79-85.
- Stevenson, W. J., 2007. *Operation Management*. New York : McGraw-Hill.
- Sulaksmi, A., Garside, A.K., dan Hadziqah, F. 2014. *Penjadwalan Produksi dengan Algoritma Heuristik Pour (Studi Kasus: Konveksi One Way–Malang)*. Jurnal Teknik Industri, Vol. 15, No. 1, pp. 35–44.

LAMPIRAN

Lampian 1

Data waktu proses 300 run

Number of run					
1-50	51-100	101-150	151-200	201-250	251-300
data1 =	data =	data =	data =	data =	data =
2 3	4 2	6 3	10 3	7 3	8 6
5 3	10 2	2 8	7 9	4 2	7 7
9 3	8 4	8 2	9 5	8 4	10 2
2 7	2 6	6 10	2 10	7 3	9 6
data =	data =	data =	data =	data =	data =
4 6	6 2	3 5	8 3	2 8	6 9
3 10	8 3	3 4	8 3	9 6	9 7
2 6	7 5	7 2	5 5	3 4	9 2
8 5	6 4	2 2	3 9	2 5	6 3
data =	data =	data =	data =	data =	data =
9 9	8 6	5 8	6 6	2 7	6 10
6 2	4 10	9 9	9 5	10 3	2 4
3 8	7 9	9 8	6 10	9 7	4 4
8 5	10 10	2 10	6 2	4 4	4 5
data =	data =	data =	data =	data =	data =
9 4	3 3	7 5	4 10	7 9	9 4
6 3	9 8	8 2	2 6	9 3	3 5
8 3	10 8	5 8	8 5	7 9	6 9
5 8	3 7	9 10	7 3	10 3	6 8
data =	data =	data =	data =	data =	data =
4 5	8 5	3 9	7 5	7 5	9 2
6 9	7 3	5 5	8 8	8 6	6 5

3 9 8 7	9 2 4 2	2 4 6 5	5 9 2 10	8 7 5 3	9 5 3 6
data = 8 5 8 5 2 3 6 8	data = 9 4 5 9 6 7 10 10	data = 8 6 7 7 10 2 9 6	data = 9 3 5 10 7 7 8 7	data = 2 2 9 3 6 5 3 5	data = 10 6 5 3 4 6 7 6
data = 8 3 8 6 6 8 6 10	data = 6 3 10 6 5 5 6 7	data = 6 9 9 7 9 2 6 3	data = 4 7 6 5 10 7 5 8	data = 2 4 10 3 3 10 5 4	data = 9 7 9 7 9 2 7 3
data = 4 9 4 10 9 4 4 4	data = 8 7 10 10 3 2 3 9	data = 6 10 2 4 4 4 4 5	data = 5 5 5 5 7 7 9 7	data = 8 3 5 6 5 7 6 2	data = 5 5 7 2 4 7 9 7
data = 2 9 2 3 7 3 3 10	data = 2 4 6 10 3 3 8 6	data = 9 4 3 5 6 9 6 8	data = 8 3 6 8 8 10 6 9	data = 9 2 9 9 8 3 6 6	data = 3 3 2 2 10 7 4 6
data = 5 5 5 7 4 3 5 2	data = 10 6 10 3 5 3 5 6	data = 9 2 6 5 9 5 3 6	data = 8 5 6 3 3 7 7 9	data = 7 6 7 8 7 6 8 7	data = 2 5 7 4 5 10 2 7

data =	data =	data =	data =	data =	data =
6 3	8 9	10 6	8 5	5 7	9 10
9 9	5 7	5 3	10 6	6 10	7 2
10 4	5 4	4 6	10 4	6 5	9 6
4 7	4 10	7 6	2 7	7 4	5 6
data =	data =	data =	data =	data =	data =
10 3	7 3	9 7	2 8	10 3	4 9
7 9	4 2	9 7	7 8	9 4	5 3
9 5	8 4	9 2	8 5	4 9	2 3
2 10	7 3	7 3	10 7	7 8	6 2
data =	data =	data =	data =	data =	data =
8 3	2 8	5 5	9 7	2 4	9 10
8 3	9 6	7 2	2 4	7 2	3 3
5 5	3 4	4 7	10 9	6 6	9 2
3 9	2 5	9 7	10 6	10 10	6 5
data =	data =	data =	data =	data =	data =
6 6	2 7	3 3	8 6	10 2	5 10
9 5	10 3	2 2	5 4	7 2	5 8
6 10	9 7	10 7	10 9	7 6	5 10
6 2	4 4	4 6	2 5	8 2	5 3
data =	data =	data =	data =	data =	data =
4 10	7 9	2 5	8 5	6 2	8 7
2 6	9 3	7 4	2 7	3 10	5 5
8 5	7 9	5 10	5 7	9 8	6 9
7 3	10 3	2 7	6 3	7 10	2 9

data = 7 5 8 8 5 9 2 10	data = 2 2 9 3 6 5 3 5	data = 9 10 7 2 9 6 5 6	data = 10 8 8 3 9 4 9 7	data = 8 7 5 8 4 9 6 6	data = 5 6 4 5 5 8 10 5
data = 9 3 5 10 7 7 8 7	data = 10 4 6 7 9 5 7 10	data = 4 9 5 3 2 3 6 2	data = 6 5 2 6 2 10 4 7	data = 9 7 6 8 3 10 7 8	data = 10 6 3 2 5 8 10 2
data = 4 7 6 5 10 7 5 8	data = 2 4 10 3 3 10 5 4	data = 9 10 3 3 9 2 6 5	data = 4 9 6 10 7 7 3 2	data = 8 7 9 9 10 6 6 6	data = 7 8 10 10 5 3 8 7
data = 5 5 5 5 7 7 9 7	data = 8 3 5 6 5 7 6 2	data = 5 10 5 8 5 10 5 3	data = 9 5 7 5 8 3 2 9	data = 3 3 3 6 3 9 2 4	data = 10 5 4 6 6 8 7 3
data = 8 3 6 8 8 10 6 9	data = 9 2 9 9 8 3 6 6	data = 8 7 5 5 6 9 2 9	data = 9 9 6 8 10 5 3 7	data = 4 6 10 7 2 4 7 4	data = 2 6 5 3 8 6 10 10
data =	data =	data =	data =	data =	data =

8 5 6 3 3 7 7 9	2 4 4 9 9 6 9 7	5 6 4 5 5 8 10 5	10 3 7 6 4 5 5 6	5 5 4 7 8 3 10 3	5 3 4 8 8 4 4 6
data = 8 5 10 6 10 4 2 7	data = 7 6 7 8 7 6 8 7	data = 10 6 3 2 5 8 10 2	data = 7 3 2 7 4 4 7 7	data = 4 4 7 6 6 6 10 9	data = 9 10 9 5 8 3 8 8
data = 2 8 7 8 8 5 10 7	data = 10 3 9 4 4 9 7 8	data = 7 8 10 10 5 3 8 7	data = 6 5 6 5 6 4 10 5	data = 3 8 10 10 10 9 2 10	data = 10 2 3 7 2 10 6 6
data = 9 7 2 4 10 9 10 6	data = 2 4 7 2 6 6 10 10	data = 10 10 4 8 7 6 6 2	data = 4 4 5 3 6 5 8 6	data = 3 7 8 7 10 2 5 10	data = 2 9 8 5 8 4 7 5
data = 8 6 5 4 10 9 2 5	data = 10 2 7 2 7 6 8 2	data = 9 10 6 8 7 3 9 5	data = 2 4 7 8 8 9 4 10	data = 10 7 8 4 5 9 8 3	data = 8 8 3 10 9 5 6 8
data = 8 5 2 7	data = 6 2 3 10	data = 10 2 10 5	data = 6 9 10 3	data = 4 10 2 9	data = 7 10 8 7

5 7 6 3	9 8 7 10	5 9 10 2	6 5 9 6	5 4 4 2	2 9 7 2
data = 10 8 8 3 9 4 9 7	data = 8 7 5 8 4 9 6 6	data = 3 4 3 3 3 2 7 8	data = 8 2 7 5 6 5 6 4	data = 2 7 7 4 8 3 9 7	data = 8 2 7 5 6 5 6 4
data = 6 5 2 6 2 10 4 7	data = 9 7 6 8 3 10 7 8	data = 6 9 10 6 6 3 5 8	data = 9 2 8 7 10 9 7 10	data = 10 4 9 6 10 3 4 3	data = 5 4 3 10 2 10 2 6
data = 4 9 6 10 7 7 3 2	data = 8 7 9 9 10 6 6 6	data = 9 9 2 6 8 8 5 5	data = 3 8 8 4 6 6 10 3	data = 3 8 7 4 2 4 4 8	data = 10 10 8 3 4 8 9 4
data = 9 5 7 5 8 3 2 9	data = 3 3 3 6 3 9 2 4	data = 4 4 3 6 3 3 8 8	data = 10 10 5 4 2 4 8 10	data = 10 3 3 10 2 3 3 7	data = 7 7 6 4 10 6 4 6
data = 9 9 6 8 10 5 3 7	data = 4 6 10 7 2 4 7 4	data = 5 6 9 10 9 9 7 7	data = 3 3 10 6 8 2 9 9	data = 4 10 3 4 8 8 6 10	data = 8 5 4 5 8 10 9 9

data =	data =	data =	data =	data =	data =
10 3 7 6 4 5 5 6	5 5 4 7 8 3 10 3	9 5 7 6 5 8 4 4	7 5 4 2 5 3 9 2	7 8 9 3 8 10 5 3	8 4 4 10 9 5 10 2
data =	data =	data =	data =	data =	data =
7 3 2 7 4 4 7 7	4 4 7 6 6 6 10 9	9 6 7 6 5 8 8 7	7 8 3 5 6 9 3 2	8 4 7 10 3 10 2 4	4 3 8 10 6 5 2 3
data =	data =	data =	data =	data =	data =
6 5 6 5 6 4 10 5	3 8 10 10 10 9 2 10	9 3 10 10 6 4 9 4	7 2 10 2 4 4 9 10	10 5 8 3 2 3 7 2	2 2 7 10 8 8 2 7
data =	data =	data =	data =	data =	data =
2 4 7 8 8 9 4 10	3 7 8 7 10 2 5 10	9 10 8 3 3 4 2 7	2 6 4 4 10 8 3 10	2 5 9 8 2 10 9 3	8 7 10 10 6 2 8 10
data =	data =	data =	data =	data =	data =
10 9 7 8 6 8 9 9	10 7 8 4 5 9 8 3	4 7 6 5 2 6 10 5	8 2 7 4 3 9 2 10	3 2 4 9 2 10 5 2	3 2 10 6 7 9 6 4

data = 4 2 3 5 10 8 3 10	data = 4 10 2 9 5 4 4 2	data = 5 6 4 7 7 3 8 5	data = 2 4 9 9 7 4 7 8	data = 7 4 3 5 4 3 3 4	data = 8 8 4 7 8 10 8 3
data = 7 2 3 5 4 7 4 2	data = 2 7 7 4 8 3 9 7	data = 9 9 8 2 6 7 7 2	data = 7 7 3 3 7 7 7 3	data = 10 2 5 3 6 9 2 9	data = 3 3 9 7 2 4 6 2
data = 8 4 10 3 8 4 9 8	data = 10 4 9 6 10 3 4 3	data = 5 8 4 10 9 10 2 9	data = 6 9 7 2 8 4 8 9	data = 8 9 6 10 4 5 7 5	data = 6 10 9 3 7 9 2 7
data = 3 8 8 4 6 6 10 3	data = 3 8 7 4 2 4 4 8	data = 5 4 6 10 3 8 7 5	data = 4 2 10 2 8 4 2 6	data = 3 6 10 6 8 4 9 2	data = 9 6 3 8 7 3 3 6
data = 10 10 5 4 2 4 8 10	data = 10 3 3 10 2 3 3 7	data = 8 8 4 3 5 9 10 7	data = 6 2 8 3 7 5 6 4	data = 4 2 3 5 10 8 3 10	data = 2 4 7 2 7 5 4 8
data =	data =	data =	data =	data =	data =

3 3 10 6 8 2 9 9	4 10 3 4 8 8 6 10	3 5 3 7 7 6 7 2	8 6 4 10 7 9 10 10	7 2 3 5 4 7 4 2	7 8 5 9 3 8 6 10
data = 7 5 4 2 5 3 9 2	data = 7 8 9 3 8 10 5 3	data = 2 8 4 2 2 2 5 7	data = 3 3 9 8 10 8 3 7	data = 8 4 10 3 8 4 9 8	data = 9 7 6 10 10 2 2 4
data = 7 8 3 5 6 9 3 2	data = 8 4 7 10 3 10 2 4	data = 7 7 2 2 2 5 3 7	data = 8 5 7 3 9 2 4 2	data = 10 9 7 8 6 8 9 9	data = 8 6 7 10 2 8 9 7
data = 7 2 10 2 4 4 9 10	data = 10 5 8 3 2 3 7 2	data = 8 5 8 5 2 3 6 8	data = 9 4 5 9 6 7 10 10	data = 10 4 6 7 9 5 7 10	data = 10 4 3 6 3 5 8 2
data = 2 6 4 4 10 8 3 10	data = 2 5 9 8 2 10 9 3	data = 8 3 8 6 6 8 6 10	data = 6 3 10 6 5 5 6 7	data = 6 3 2 8 8 2 6 10	data = 5 5 4 4 10 2 10 4
data = 8 2 7 4	data = 3 2 4 9	data = 4 9 4 10	data = 8 7 10 10	data = 3 5 3 4	data = 10 8 4 3

3 9 2 10	2 10 5 2	9 4 4 4	3 2 3 9	7 2 2 2	7 5 3 9
data = 2 4 9 9 7 4 7 8	data = 7 4 3 5 4 3 3 4	data = 2 9 2 3 7 3 3 10	data = 2 4 6 10 3 3 8 6	data = 5 8 9 9 9 8 2 10	data = 9 7 9 3 4 3 10 9
data = 7 7 3 3 7 7 7 3	data = 10 2 5 3 6 9 2 9	data = 5 5 5 7 4 3 5 2	data = 10 6 10 3 5 3 5 6	data = 7 5 8 2 5 8 9 10	data = 10 6 7 7 5 4 4 6
data = 6 9 7 2 8 4 8 9	data = 3 6 10 6 8 4 9 2	data = 6 3 9 9 10 4 4 7	data = 8 9 5 7 5 4 4 10	data = 3 9 5 5 2 4 6 5	data = 6 3 2 8 8 3 2 4

Lampiran 2

Data due date 300 run

Number of run					
1-50	51-100	101-150	151-200	201-250	251-300
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
20	15	13	19	12	12
13	14	14	20	17	15
17	14	16	14	20	11
15	14	18	15	14	15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
19	14	14	15	16	19
18	16	12	12	17	14
15	12	15	12	11	13
11	11	15	16	20	14
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
19	15	11	18	19	18
15	19	17	14	17	17
17	20	18	14	15	20
18	13	13	13	18	19
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
20	12	13	19	17	15
18	19	18	17	13	17
12	13	18	13	19	11
15	17	20	20	12	16
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
20	20	15	17	13	15
20	17	20	13	11	19

15 19	19 11	15 19	17 17	14 14	19 15
duedate = 11 14 19 11	duedate = 12 19 15 19	duedate = 19 12 14 16	duedate = 12 11 13 12	duedate = 20 19 13 11	duedate = 18 17 11 12
duedate = 12 13 12 17	duedate = 18 17 14 12	duedate = 18 16 15 15	duedate = 11 19 12 11	duedate = 11 16 18 19	duedate = 15 11 13 14
duedate = 13 12 11 18	duedate = 12 12 15 19	duedate = 11 15 14 14	duedate = 18 16 13 14	duedate = 17 13 12 17	duedate = 19 14 13 15
duedate = 15 20 15 15	duedate = 15 19 15 15	duedate = 19 18 20 16	duedate = 11 19 19 13	duedate = 20 19 20 14	duedate = 19 12 16 15
duedate = 19 16 13 17	duedate = 15 15 20 11	duedate = 11 16 19 20	duedate = 16 12 16 14	duedate = 13 16 12 13	duedate = 14 14 19 18

duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
19	13	13	17	13	19
11	11	18	19	17	15
17	17	16	16	11	18
14	17	20	20	13	12
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
19	20	18	18	17	11
16	16	13	12	14	14
18	15	15	19	14	14
15	12	12	12	18	15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
14	17	20	17	20	20
12	18	14	17	12	14
12	14	11	18	11	13
17	11	18	14	12	16
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
14	15	19	15	12	16
16	18	13	15	20	12
12	18	13	17	12	15
17	20	20	12	17	15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
14	19	12	11	13	19
19	14	20	18	13	19
19	17	18	16	18	19
16	18	19	13	16	17

duedate = 15 19 19 17	duedate = 12 20 16 17	duedate = 13 15 11 19	duedate = 12 12 15 15	duedate = 20 14 18 20	duedate = 16 17 12 12
duedate = 19 17 14 13	duedate = 13 16 20 14	duedate = 16 14 14 19	duedate = 11 20 12 14	duedate = 16 19 18 14	duedate = 15 16 13 18
duedate = 14 16 18 14	duedate = 17 14 17 17	duedate = 14 11 12 11	duedate = 14 12 19 14	duedate = 18 12 20 12	duedate = 14 11 16 16
duedate = 19 16 14 18	duedate = 14 15 17 19	duedate = 16 12 16 12	duedate = 18 15 14 16	duedate = 18 17 12 11	duedate = 13 11 20 13
duedate = 16 15 17 17	duedate = 14 15 16 16	duedate = 18 14 19 11	duedate = 12 16 17 11	duedate = 13 20 20 13	duedate = 13 11 17 16
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =

18	18	16	19	20	11
20	16	20	19	17	17
16	18	13	17	11	15
19	15	19	15	16	11
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
12	12	12	11	14	14
20	18	11	19	13	13
13	20	13	12	12	20
13	19	20	15	11	17
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
19	18	11	11	11	19
18	15	13	14	18	17
12	12	20	19	11	18
11	17	13	19	19	15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
19	14	15	14	14	20
12	12	13	19	18	11
13	16	17	15	20	16
17	19	20	16	12	16
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
13	17	18	17	12	13
15	20	17	17	14	14
11	20	12	17	11	11
20	11	11	17	14	16
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
16	14	11	16	11	19
15	16	13	18	18	12

16 14	13 16	19 20	16 17	20 12	12 11
duedate = 15 13 16 18	duedate = 11 16 18 20	duedate = 11 19 17 16	duedate = 20 19 14 16	duedate = 17 12 14 12	duedate = 19 12 19 16
duedate = 16 17 13 14	duedate = 18 18 15 17	duedate = 13 19 13 18	duedate = 12 13 19 17	duedate = 13 14 12 13	duedate = 20 12 11 14
duedate = 18 17 15 16	duedate = 19 11 20 20	duedate = 17 13 17 17	duedate = 11 16 15 20	duedate = 20 14 16 11	duedate = 15 15 15 14
duedate = 18 11 17 11	duedate = 17 13 19 16	duedate = 17 12 17 12	duedate = 13 11 15 20	duedate = 11 12 19 19	duedate = 20 17 20 12
duedate = 15 14 19 11	duedate = 18 15 20 11	duedate = 16 17 17 17	duedate = 20 16 17 18	duedate = 12 20 18 19	duedate = 20 13 17 15

duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
18	16	19	12	15	19
20	13	11	11	15	18
20	19	14	16	13	15
18	14	18	11	11	11
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
15	17	14	15	13	19
15	11	20	12	12	15
13	14	17	18	20	17
17	15	11	17	12	18
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
14	15	11	11	11	20
20	16	11	19	14	18
18	17	13	18	17	12
15	12	16	20	19	15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
18	19	15	18	16	20
13	18	18	14	12	20
15	18	16	14	14	15
20	19	16	16	13	19
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
17	17	11	16	11	11
13	13	12	18	14	14
19	13	14	18	17	19
17	17	13	15	11	11

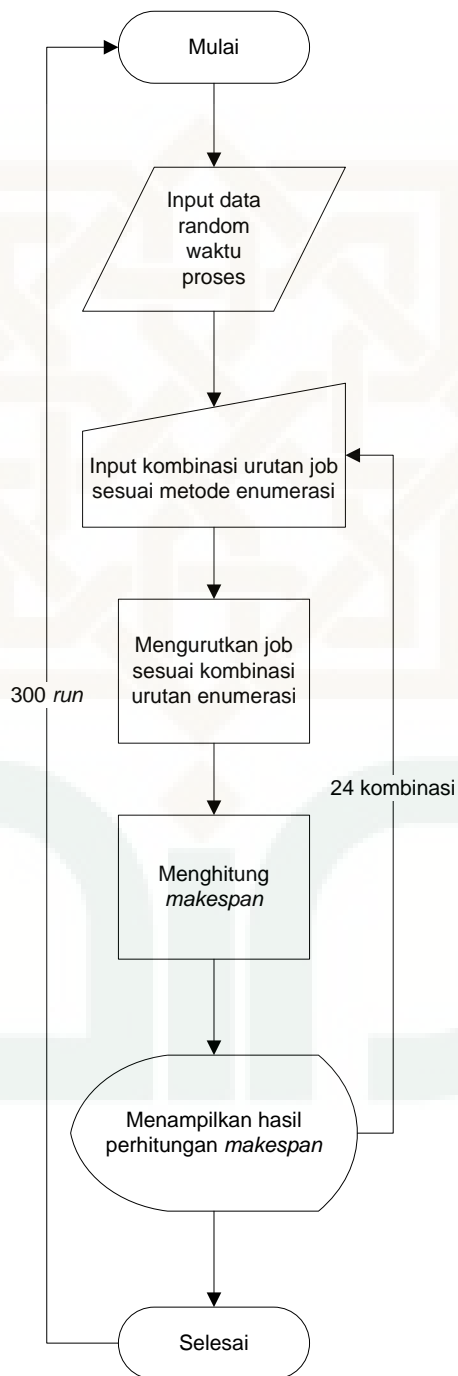
duedate = 12 13 17 17	duedate = 16 11 11 13	duedate = 18 14 17 20	duedate = 19 15 13 16	duedate = 18 20 18 18	duedate = 12 13 12 17
duedate = 14 16 15 11	duedate = 15 15 20 16	duedate = 16 20 19 20	duedate = 17 17 20 18	duedate = 13 12 13 18	duedate = 13 12 11 18
duedate = 11 14 11 14	duedate = 14 15 16 12	duedate = 12 19 20 12	duedate = 18 19 20 16	duedate = 20 17 16 19	duedate = 15 20 15 15
duedate = 17 11 11 17	duedate = 19 20 16 11	duedate = 12 18 18 17	duedate = 17 18 15 16	duedate = 19 17 17 19	duedate = 19 16 13 17
duedate = 17 11 11 12	duedate = 19 17 18 11	duedate = 18 17 19 13	duedate = 12 12 13 12	duedate = 20 15 19 17	duedate = 20 13 17 15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =

16	15	15	12	14	19
11	14	13	15	17	18
14	12	11	18	14	15
17	19	11	13	20	11
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
18	19	19	13	15	19
17	15	14	20	11	15
11	20	15	11	17	17
15	12	20	16	16	18
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
15	19	13	16	13	20
14	18	19	17	18	18
12	15	17	14	11	12
17	17	19	13	20	15
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
17	20	15	15	13	20
18	17	20	13	12	20
15	13	14	17	17	15
16	14	16	20	11	19
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
12	12	12	14	14	11
15	15	16	16	13	14
18	15	15	12	11	19
19	15	17	12	11	11
duedate =	duedate =	duedate =	duedate =	duedate =	duedate =
13	17	17	13	15	12
13	11	20	16	19	13

19 13	14 17	12 12	15 20	19 11	12 17
duedate = 19 20 13 13	duedate = 12 17 13 16	duedate = 16 20 11 19	duedate = 13 15 16 18	duedate = 18 19 18 20	duedate = 13 12 11 18
duedate = 11 11 17 12	duedate = 16 15 16 20	duedate = 11 16 12 18	duedate = 12 20 20 11	duedate = 17 18 14 19	duedate = 15 20 15 15
duedate = 19 12 12 20	duedate = 14 15 14 15	duedate = 13 20 12 16	duedate = 18 20 19 20	duedate = 14 11 18 20	duedate = 19 16 13 17

Lampiran 3

Flow Chart Metode Enumerasi



Lampiran 4

Coding/Program Matlab Metode Enumerasi

```
tic

load datarandom1;           %input data random
Data_diambil=data;
Simpanan_E=[1;2;3;4];
s=[4;3;2;1];               %kombinasi urutan job pada metode enumerasi

if s(1,1)==Simpanan_E(1);   %mengurutkan job sesuai
    x3=Data_diambil(1,:);   kombinasi urutan enumerasi
    Data_diambil(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
elseif s(1,1)==Simpanan_E(2);
    x3=Data_diambil(2,:);
    Data_diambil(2,:)=[];
    s(1,:)=[];
    Simpanan_E(2,:)=[];
elseif s(1,1)==Simpanan_E(3);
    x3=Data_diambil(3,:);
    Data_diambil(3,:)=[];
    s(1,:)=[];
    Simpanan_E(3,:)=[];
elseif s(1,1)==Simpanan_E(4);
    x3=Data_diambil(4,:);
    Data_diambil(4,:)=[];
    s(1,:)=[];
    Simpanan_E(4,:)=[];
end

if s(1,1)==Simpanan_E(1);
    x4=Data_diambil(1,:);
    Data_diambil(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
elseif s(1,1)==Simpanan_E(2);
    x4=Data_diambil(2,:);
    Data_diambil(2,:)=[];
    s(1,:)=[];
    Simpanan_E(2,:)=[];
elseif s(1,1)==Simpanan_E(3);
    x4=Data_diambil(3,:);
    Data_diambil(3,:)=[];
    s(1,:)=[];
    Simpanan_E(3,:)=[];
end

if s(1,1)==Simpanan_E(1);
    x5=Data_diambil(1,:);
    Data_diambil(1,:)=[];
    s(1,:)=[];
```

```

        Simpanan_E(1,:)=[];
elseif s(1,1)==Simpanan_E(2);
    x5=Data_diambil(2,:);
    Data_diambil(2,:)=[];
    s(1,:)=[];
    Simpanan_E(2,:)=[];
end

if s(1,1)==Simpanan_E(1);
    x6=Data_diambil(1,:);
    Data_diambil(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
end

data=[x3; x4; x5; x6];

[J M]=size(data); Data=data; %menghitung makespan
for i1=1:M;
    S(1,1)=0;
    if i1==1;
        S(1,1)=0;
        for it=2:J;
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J;
            if i2==1;
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end

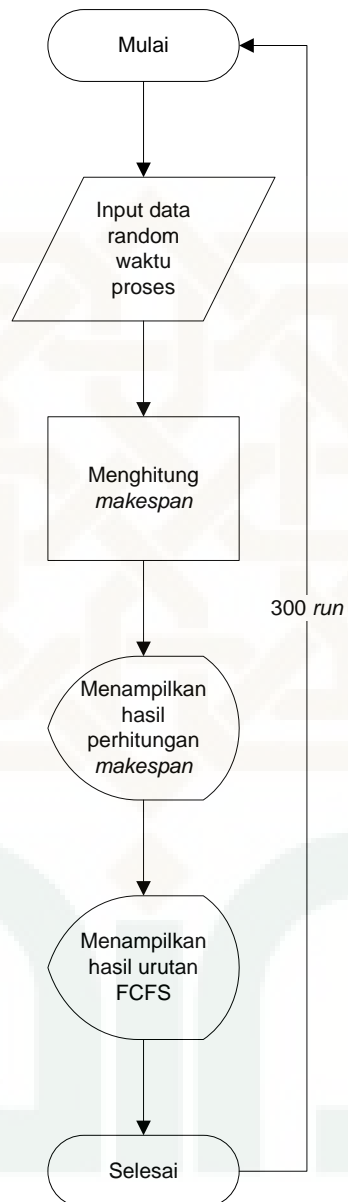
makespan1=E(J,M); %hasil makespan

toc

```


Lampiran 5

Flow Chart Metode FCFS



Lampiran 6

Coding/Program Matlab Metode FCFS

```
tic

load datarandom1 %input data random
data=data

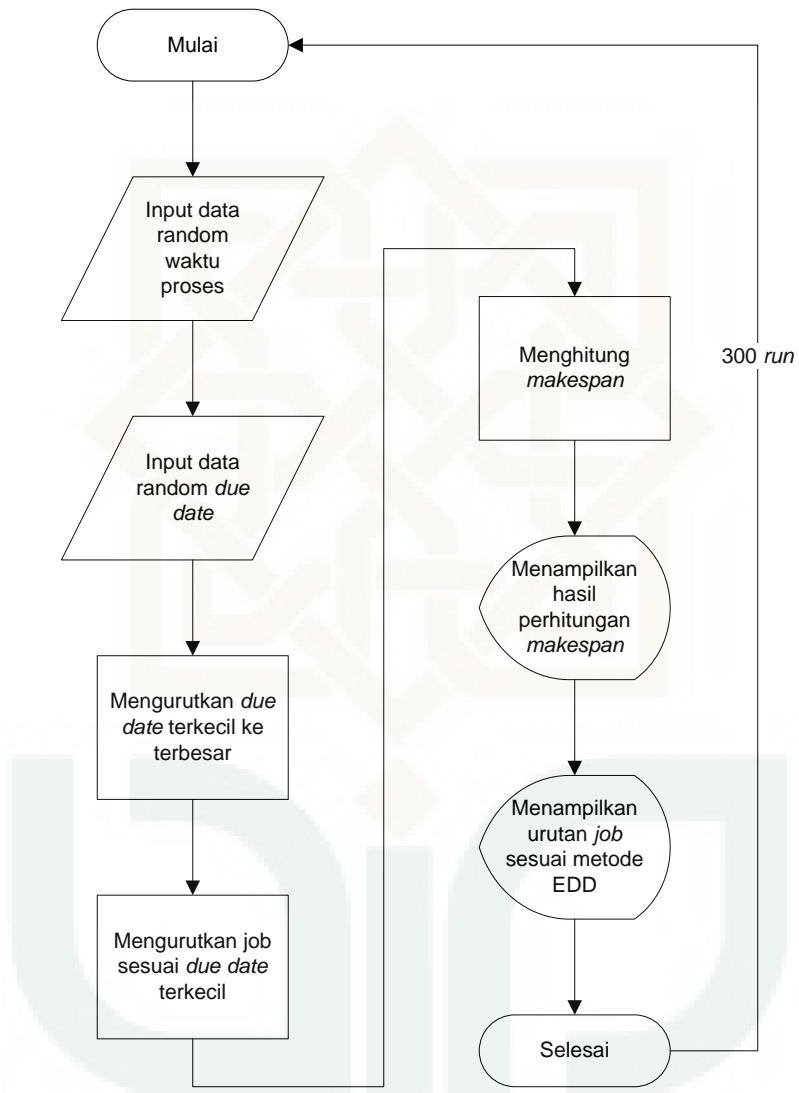
[J M]=size(data); Data=data; %menghitung makespan
for i1=1:M;
    S(1,1)=0;
    if i1==1;
        S(1,1)=0;
        for it=2:J;
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J;
            if i2==1;
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
end

makespan=E(J,M) %menampilkan hasil perhitungan makespan
urutan_FCFS=[1;2;3;4] %menampilkan hasil urutan FCFS

toc
```

Lampiran 7

Flow Chart Metode EDD



Lampiran 8

Coding/Program Matlab Metode EDD

```
tic

load datarandom50           %input data random waktu proses
load datarandom_duedate50   %input data random due date
s=sort(duedate);            %mengurutkan due date terkecil ke
                             terbesar

Hasil_urutan=[1;2;3;4];

if s(1,1)==duedate(1);      %mengurutkan job sesuai urutan due
    x3=data(1,:);           date terkecil
    c=Hasil_urutan(1,:);
    data(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    duedate(1,:)=[];
elseif s(1,1)==duedate(2);
    x3=data(2,:);
    c=Hasil_urutan(2,:);
    data(2,:)=[];
    Hasil_urutan(2,:)=[];
    s(1,:)=[];
    duedate(2,:)=[];
elseif s(1,1)==duedate(3);
    x3=data(3,:);
    c=Hasil_urutan(3,:);
    data(3,:)=[];
    Hasil_urutan(3,:)=[];
    s(1,:)=[];
    duedate(3,:)=[];
elseif s(1,1)==duedate(4);
    x3=data(4,:);
    c=Hasil_urutan(4,:);
    data(4,:)=[];
    Hasil_urutan(4,:)=[];
    s(1,:)=[];
    duedate(4,:)=[];
end

if s(1,1)==duedate(1);
    x4=data(1,:);
    d=Hasil_urutan(1,:);
    data(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    duedate(1,:)=[];
elseif s(1,1)==duedate(2);
    x4=data(2,:);
    d=Hasil_urutan(2,:);
    data(2,:)=[];
    Hasil_urutan(2,:)=[];
    s(1,:)=[];
```

```

        duedate(2,:)=[];
elseif s(1,1)==duedate(3);
    x4=data(3,:);
    d=Hasil_urutan(3,:);
    data(3,:)=[];
    Hasil_urutan(3,:)=[];
    s(1,:)=[];
    duedate(3,:)=[];
end

if s(1,1)==duedate(1);
    x5=data(1,:);
    e=Hasil_urutan(1,:);
    data(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    duedate(1,:)=[];
elseif s(1,1)==duedate(2);
    x5=data(2,:);
    e=Hasil_urutan(2,:);
    data(2,:)=[];
    Hasil_urutan(2,:)=[];
    s(1,:)=[];
    duedate(2,:)=[];
end

if s(1,1)==duedate(1);
    x6=data(1,:);
    f=Hasil_urutan(1,:);
    data(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    duedate(1,:)=[];
end

p=[x3; x4; x5; x6];
q=[c; d; e; f];

data=p;
[J M]=size(data); Data=data; %menghitung makespan
for i1=1:M;
    S(1,1)=0;
    if i1==1;
        S(1,1)=0;
        for it=2:J;
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J;
            if i2==1;
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            end
        end
    end
end

```

```

        E(i2,i1)=S(i2,i1)+Data(i2,i1);
    end
end
end
end

makespan=E(J,M)           %menampilkan hasil perhitungan
                           makespan
urutan_EDD=q'             %menampilkan hasil urutan EDD

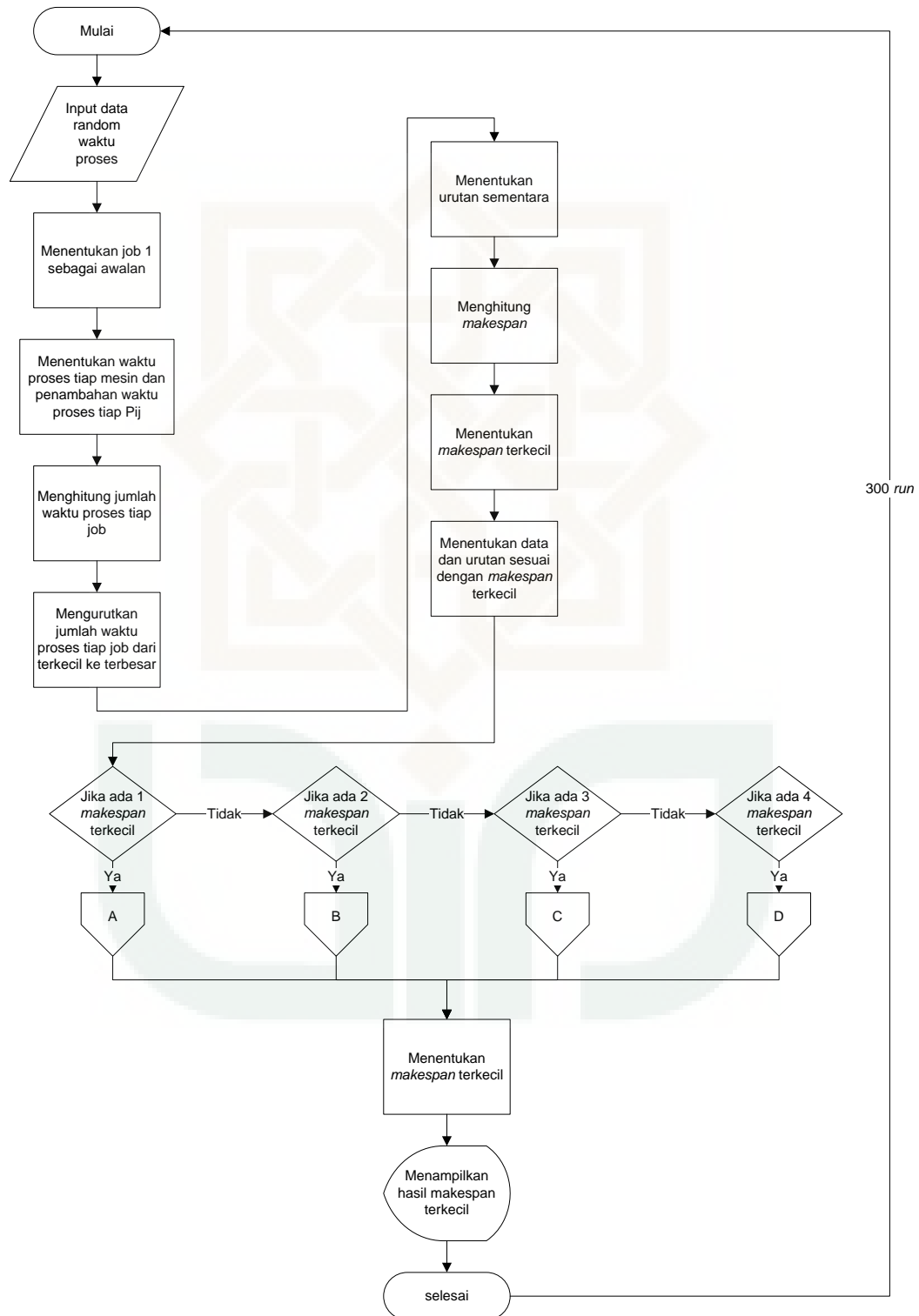
toc

```



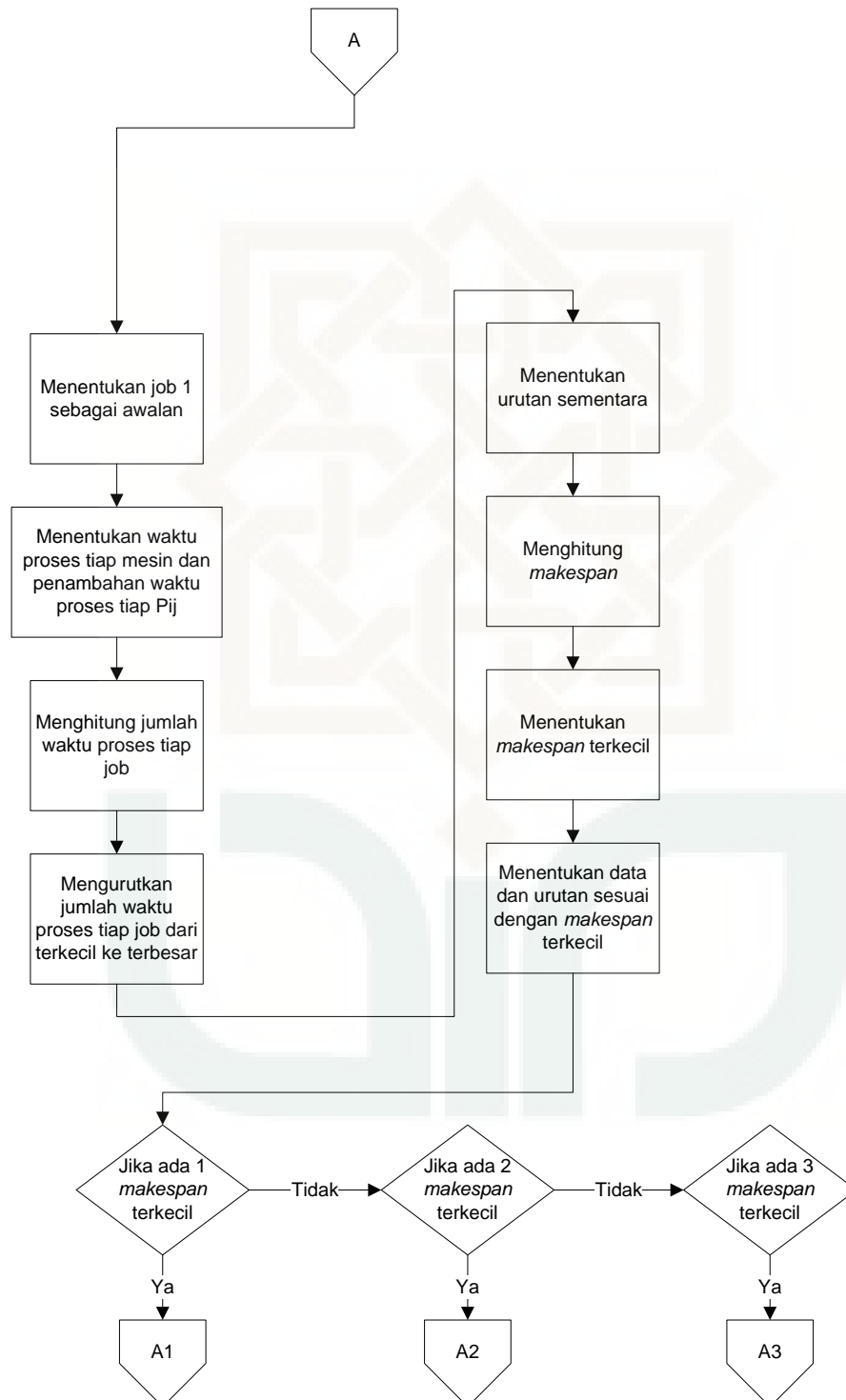
Lampiran 9

Flow Chart Metode Algoritma Heuristik Pour



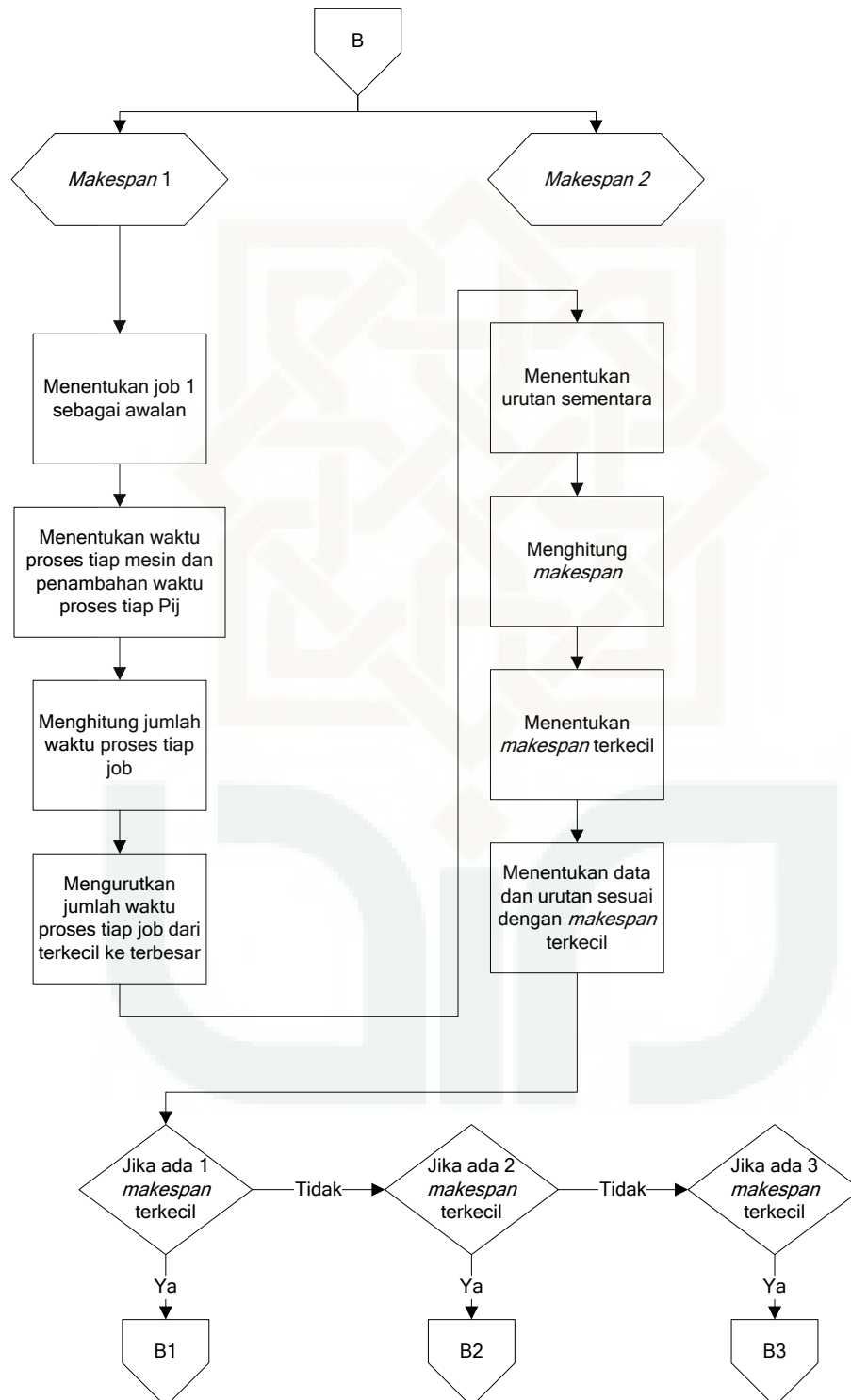
Lampiran 9.1

Flow Chart Program Iteras 1 (A)



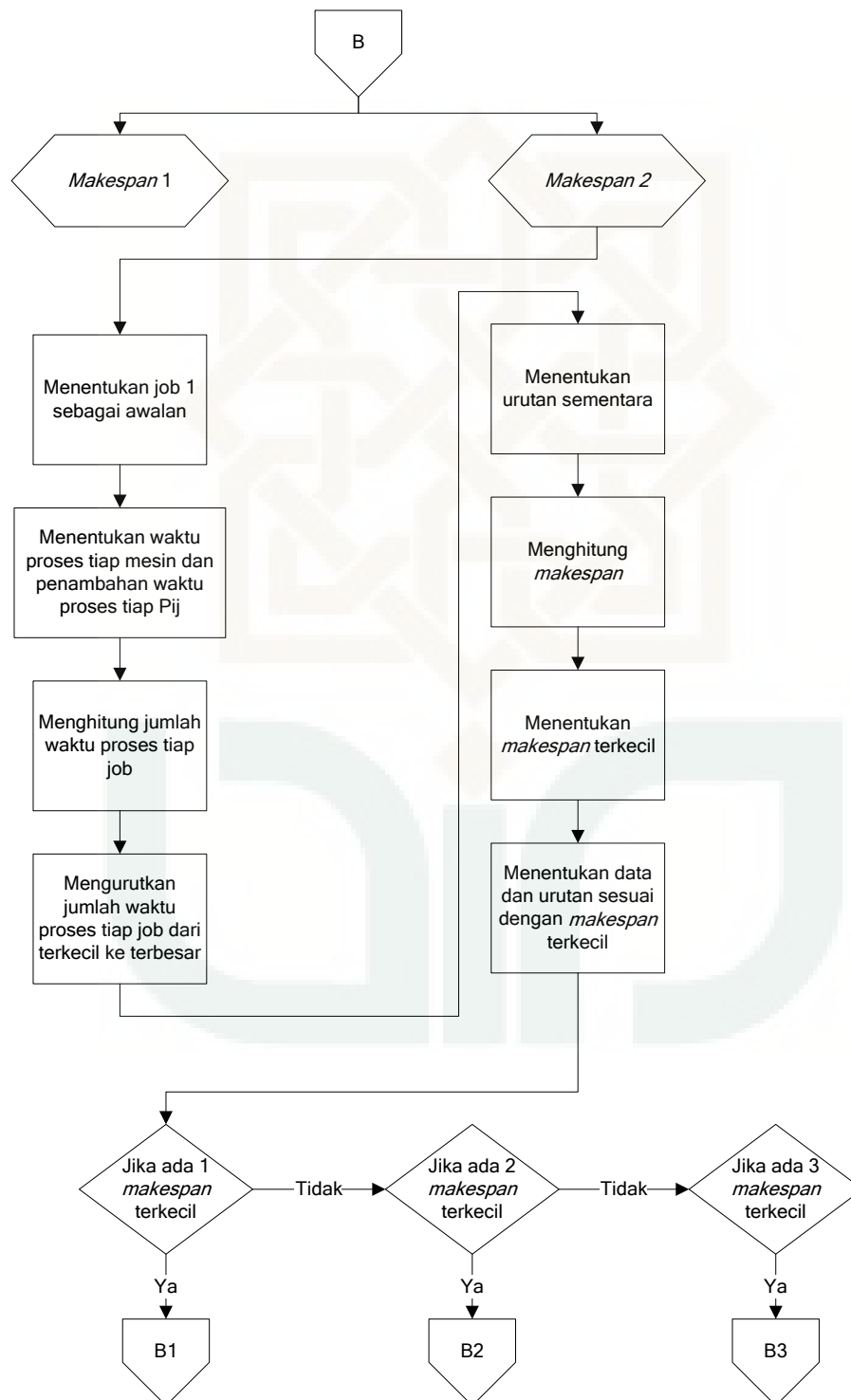
Lampiran 9.2

Flow Chart Program Iterasi 1 (B pada *makespan* 1)



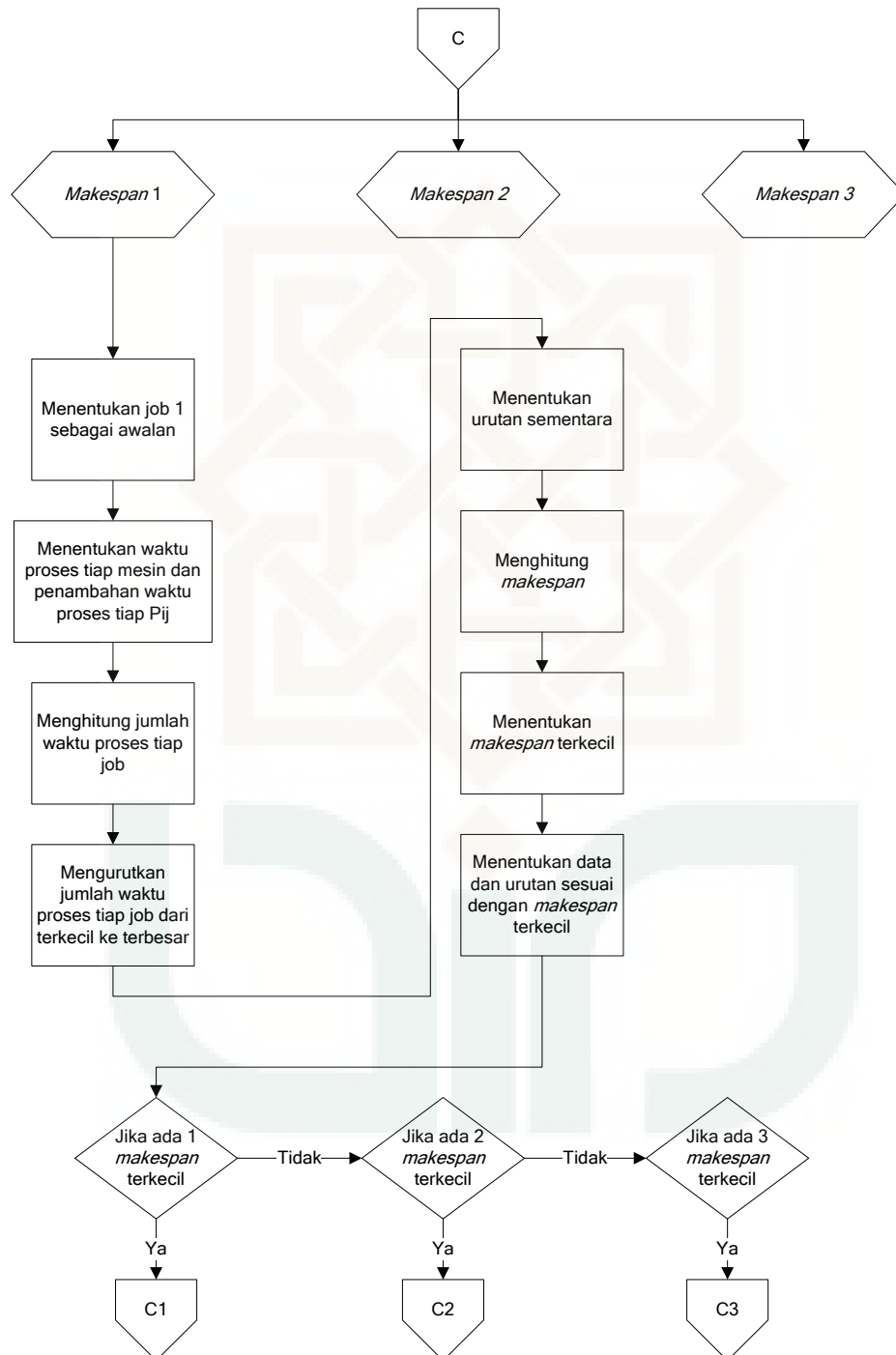
Lampiran 9.3

Flow Chart Program Iterasi 1 (B pada *makespan* 2)



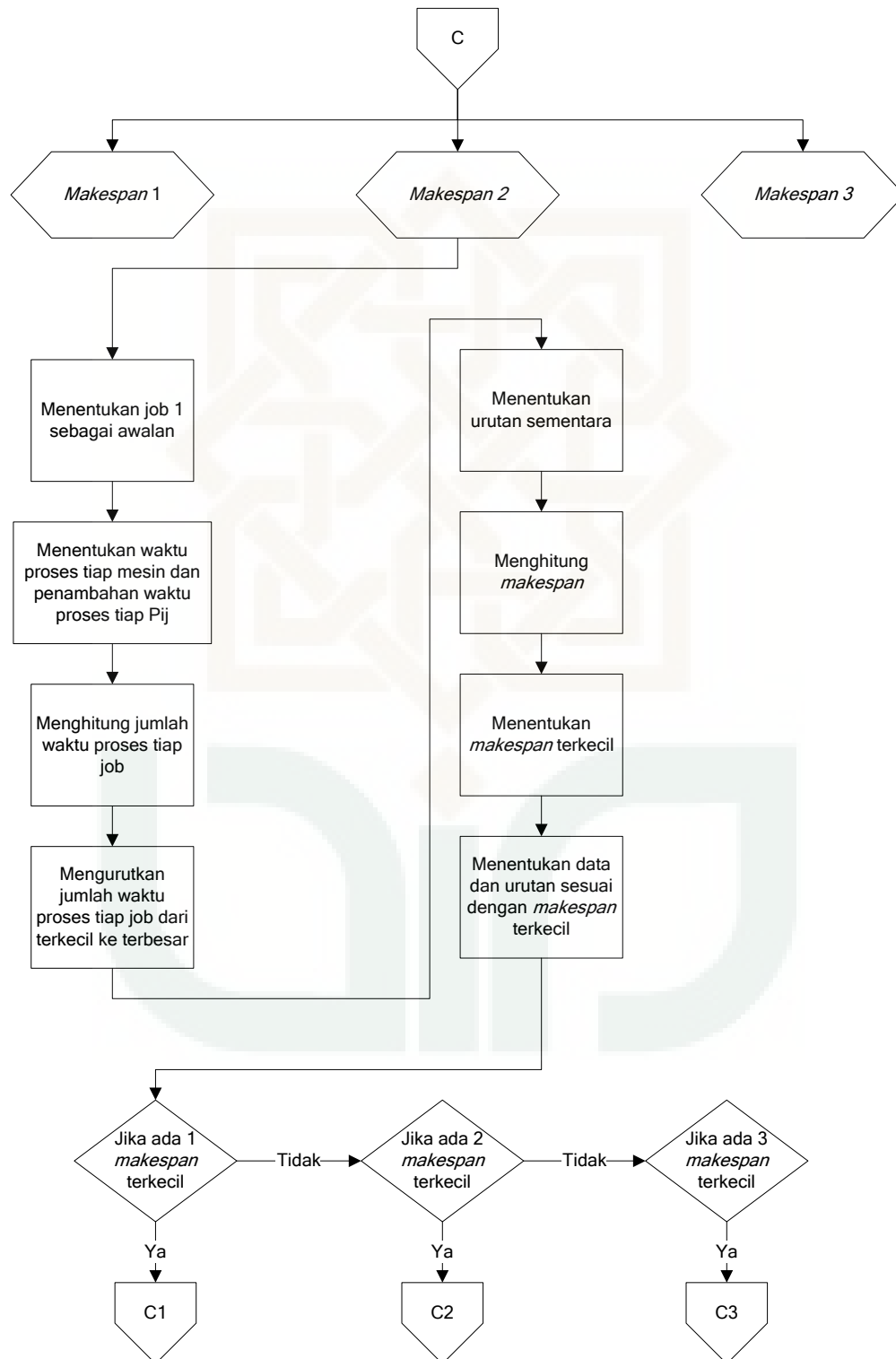
Lampiran 9.4

Flow Chart Program Iterasi 1 (C pada *makespan* 1)



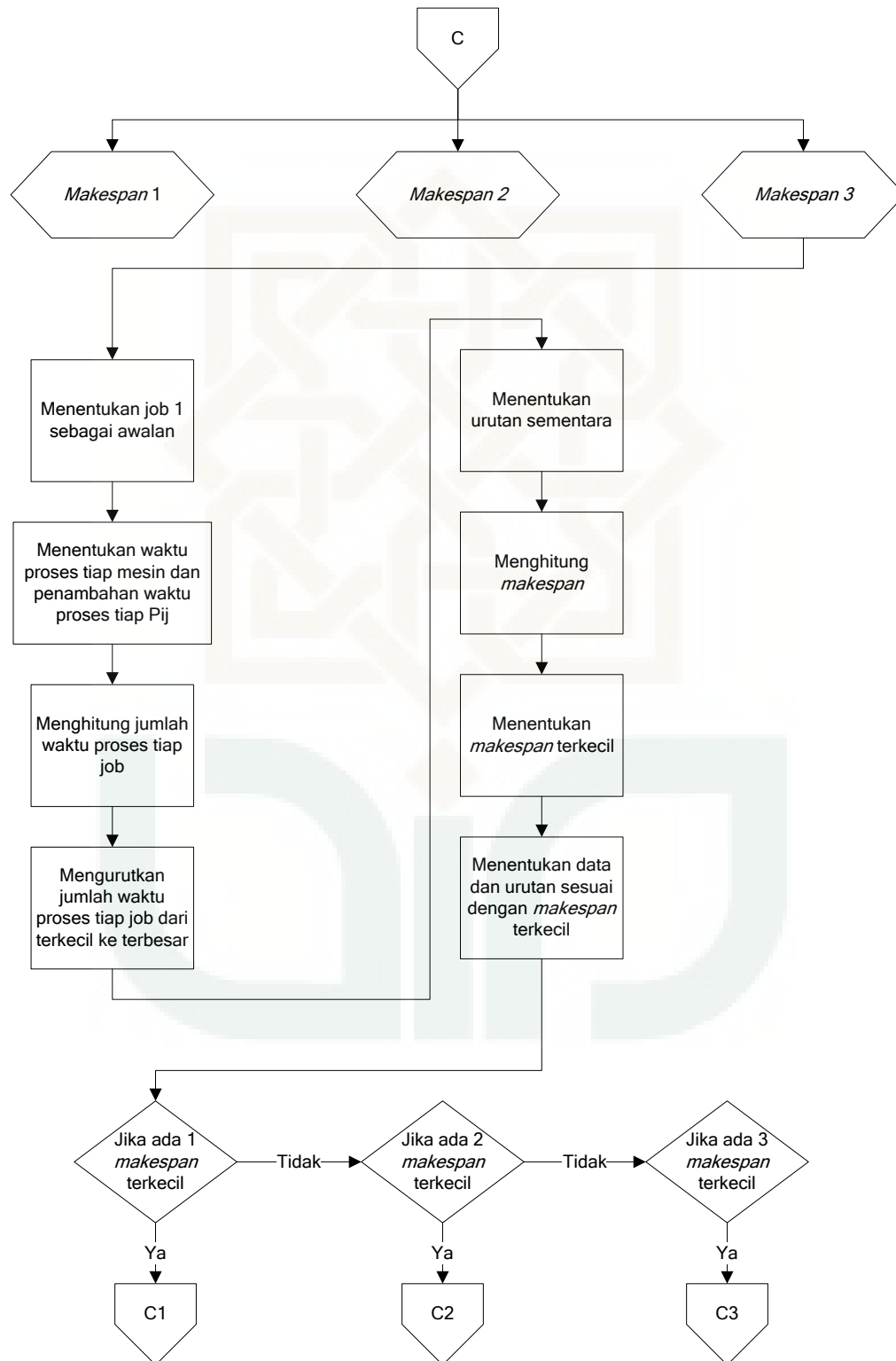
Lampiran 9.5

Flow Chart Program Iterasi 1 (C pada *makespan* 2)



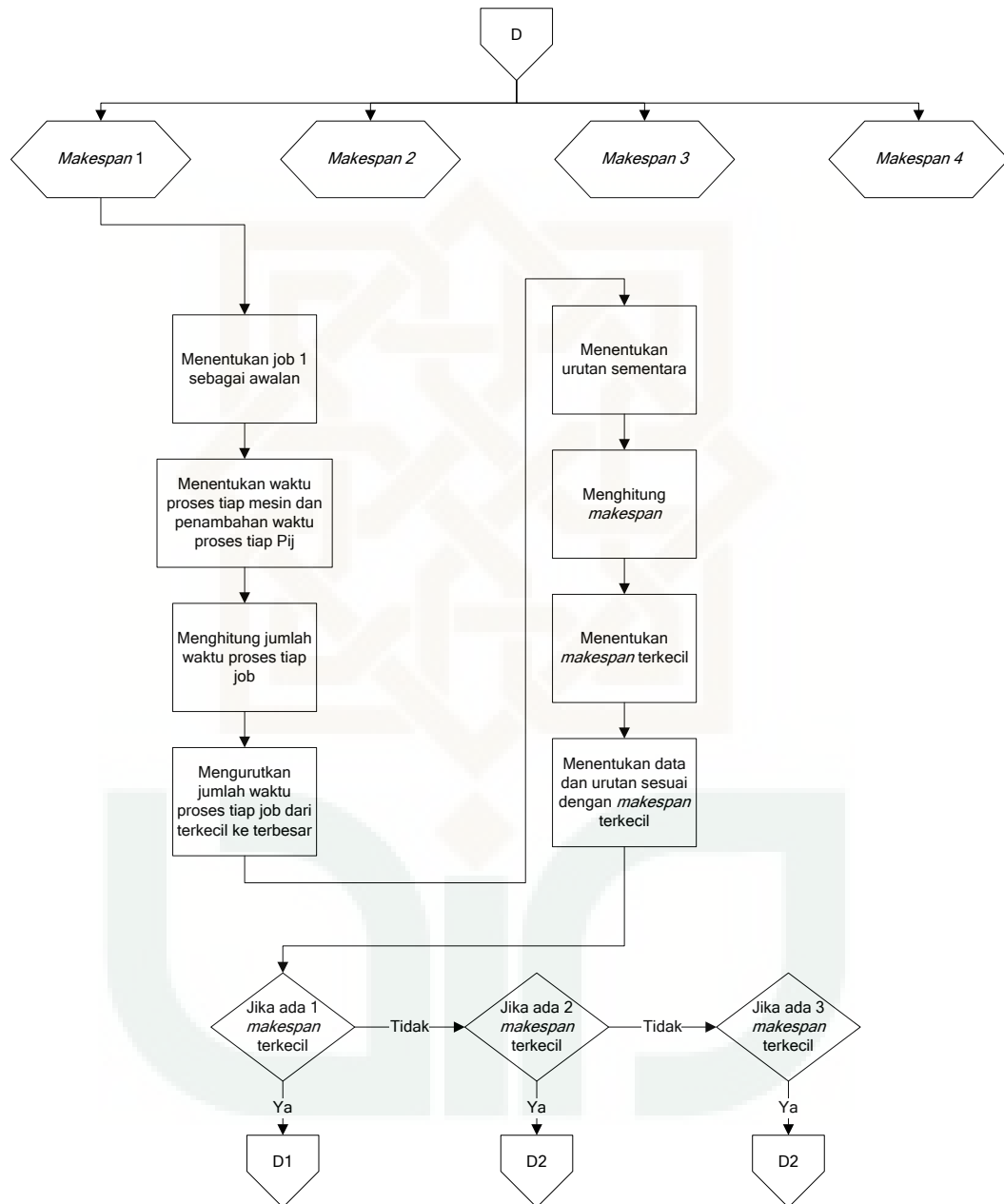
Lampiran 9.6

Flow Chart Program Iterasi 1 (C pada *makespan* 3)



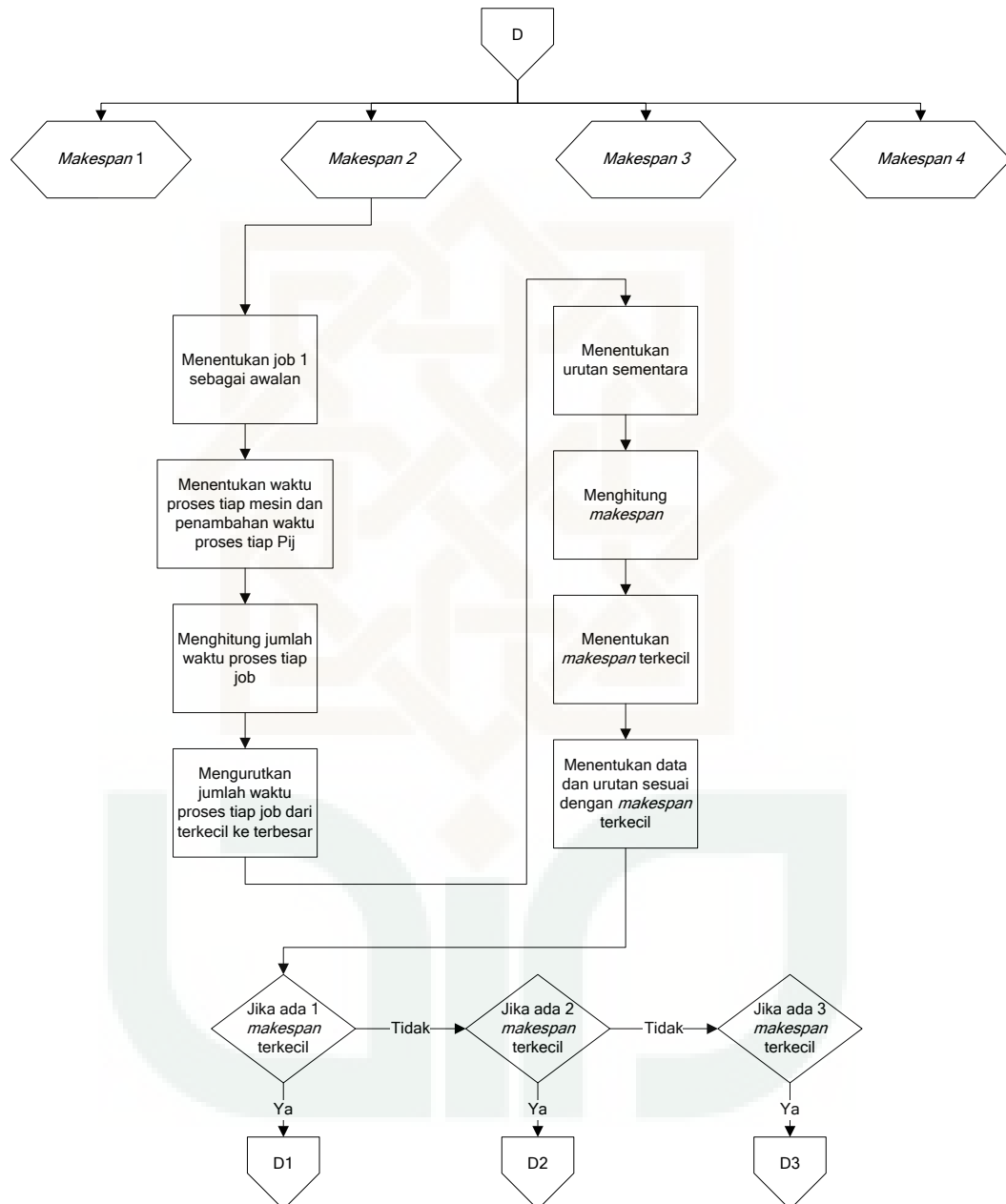
Lampiran 9.7

Flow Chart Program Iterasi 1 (D pada *makespan* 1)



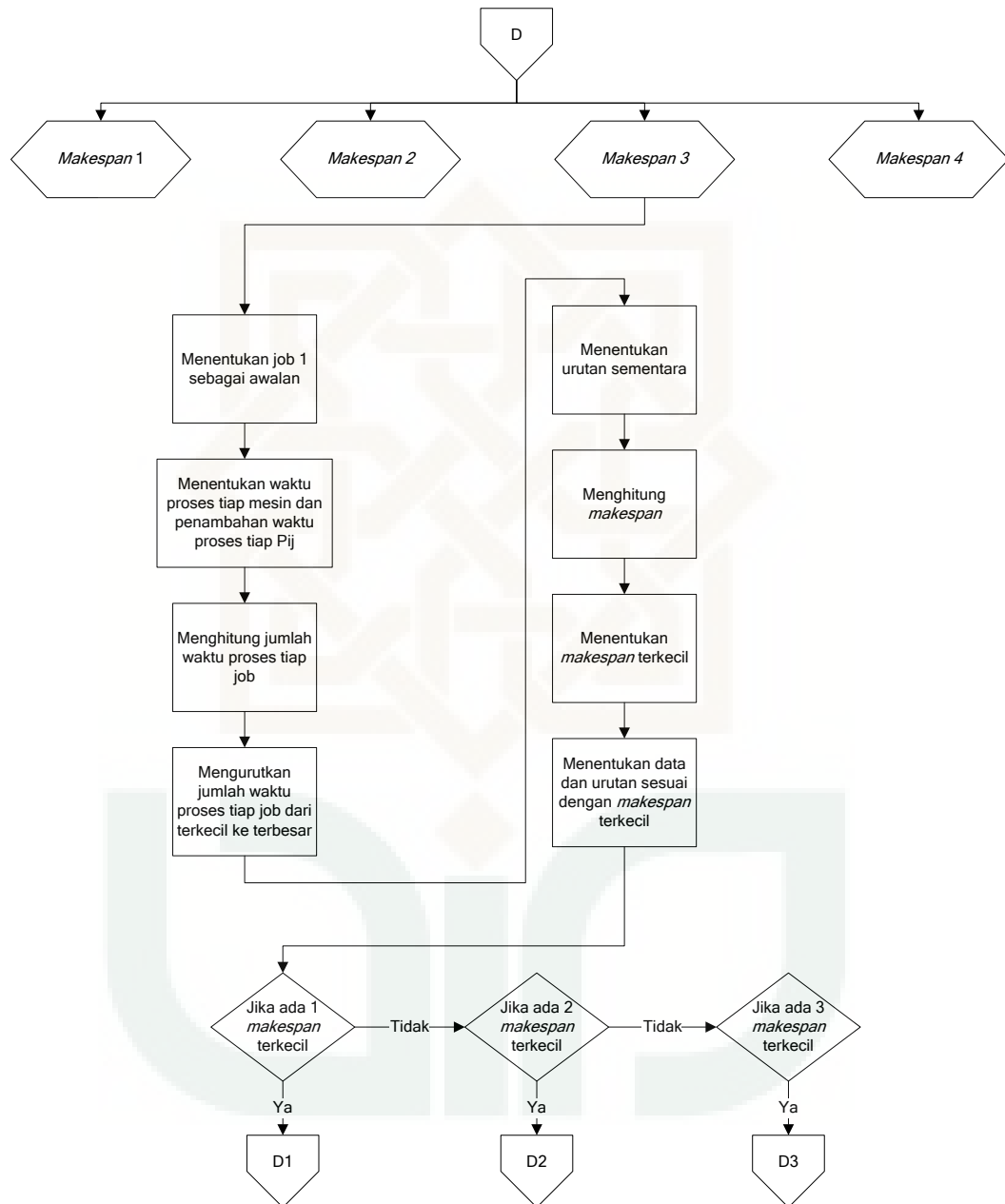
Lampiran 9.8

Flow Chart Program Iterasi 1 (D pada *makespan* 2)



Lampiran 9.9

Flow Chart Program Iterasi 1 (D pada *makespan* 3)



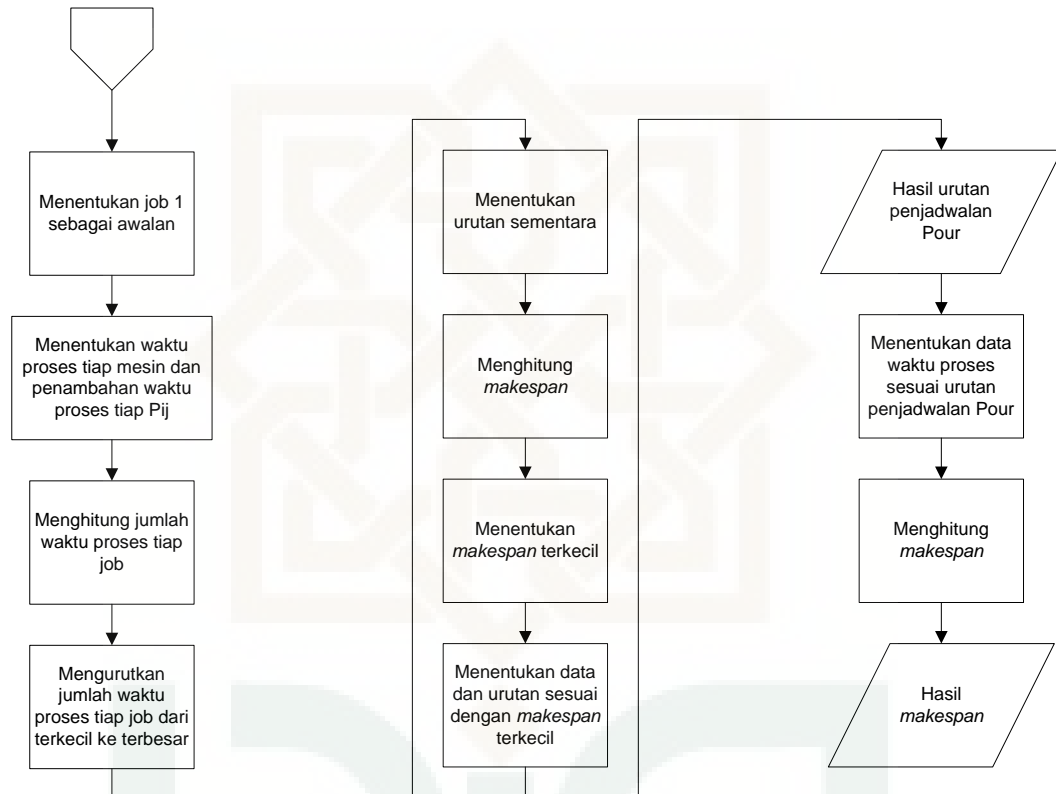
Lampiran 9.10

Flow Chart Program Iterasi 1 (D pada *makespan* 4)



Lampiran 9.11

Flow Chart Program Iterasi 2 (A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, D3)



Lampiran 10

Coding/Program Matlab Metode Pour

```
clear all
clc

tic
load datarandom1           %input data random waktu proses

[J M]=size(data);
Data=data; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0;
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end

Dat(1,:)=data(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
%menentukan waktu minimum tiap mesin dan penambahan waktu proses
tiap Pij
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
            end
        end
        data2=[Dat] ;
    end
end
```

```

        end
    end
end

Total=sum(Data_baru,2); %menghitung jumlah waktu proses tiap
job
Total1=sort(Total);urutan=Urut(2:J); %mengurutkan jumlah %waktu
proses tiap job dari terkecil ke terbesar
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru; %menentukan urutan sementara
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E) %menentukan makespan terkecil
[d1 d2]=size(s);

```

```

%menentukan data dan urutan sesuai dengan makespan terkecil
d1==4
    [p,q]=urut_4_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==4
    [data2,urut]=data_terkecil_4job(s,p,q);

[x y]=size(data2);

if x==1;    %jika ada 1 makespan terkecil
    data_awal1=data;
    data1=data2(1,:);
    data1=reshape(data1,J,M);
    data1(1,:)=[];
    urut1a=urut(1,:);
    urut1a1=urut1a(2:4);
    urutan_pour=urut(1,1);

    [makespanela,urutan_pour1a,makespane2b,urutan_pour2b,makespane3c,u
    rutan_pour3c]=permasalahan_3job_pada4job_4sama1(data1,urutan_pour,
    data_awal1,urut1a1);

    Hasilnya=[makespanela;makespane2b;makespane3c];
    s=sort(Hasilnya);    %menentukan makespan terkecil
    makespane=s(1,1)    %menampilkan hasil makespan terkecil

    toc
elseif x==2;    %jika ada 2 makespan terkecil
    for i=1;
        data_awal1=data;
        data1=data2(i,:);
        data1=reshape(data1,J,M);
        data1(1,:)=[];
        urut1a=urut(i,:);
        urut1a1=urut1a(2:4);
        urutan_pour=urut(1,1);

        [makespanela,urutan_pour1a,makespane2b,urutan_pour2b,makespane3c,u
        rutan_pour3c]=permasalahan_3job_pada4job_4sama1(data1,urutan_pour,
        data_awal1,urut1a1);
    end

    for i=2;
        data_awal2=data;
        data3=data2(i,:);
        data3=reshape(data3,J,M);
        data3(1,:)=[];
        urut2a=urut(i,:);
        urut2a1=urut2a(2:4);
        urutan_pour2=urut2a(1,1);

        [makespanela1,urutan_pour1a1,makespane2b1,urutan_pour2b1,makespane
        3c1,urutan_pour3c1]=permasalahan_3job_pada4job_4sama2(data3,urutan
        _pour2,data_awal2,urut2a1);
    end

```

```

end

Hasilnya=[makespanela;makespane2b;makespane3c;makespanela1;makespane2b1;makespane3c1];
        s=sort(Hasilnya);      %menentukan makespan terkecil
        makespane=s(1,1)      %menampilkan hasil makespan terkecil

        toc
elseif x==3;      %jika ada 3 makespan terkecil
    for i=1;
        data_awal1=data;
        data1=data2(i,:);
        data1=reshape(data1,J,M);
        data1(1,:)=[];
        urut1a=urut(i,:);
        urut1a1=urut1a(2:4);
        urutan_pour=urut(1,1);

        [makespanela,urutan_pour1a,makespane2b,urutan_pour2b,makespane3c,urutan_pour3c]=permasalahan_3job_pada4job_4sama1(data1,urutan_pour,data_awal1,urut1a1);
        end

        for i=2;
            data_awal2=data;
            data3=data2(i,:);
            data3=reshape(data3,J,M);
            data3(1,:)=[];
            urut2a=urut(i,:);
            urut2a1=urut2a(2:4);
            urutan_pour2=urut2a(1,1);

            [makespanela1,urutan_pour1a1,makespane2b1,urutan_pour2b1,makespane3c1,urutan_pour3c1]=permasalahan_3job_pada4job_4sama2(data3,urutan_pour2,data_awal2,urut2a1);
            end

            for i=3;
                data_awal3=data;
                data5=data2(i,:);
                data5=reshape(data5,J,M);
                data5(1,:)=[];
                urut3a=urut(i,:);
                urut3a1=urut3a(2:4);
                urutan_pour3=urut3a(1,1);

                [makespanela2,urutan_pour1a2,makespane2b2,urutan_pour2b2,makespane3c2,urutan_pour3c2]=permasalahan_3job_pada4job_4sama3(data5,urutan_pour3,data_awal3,urut3a1);
                end

        Hasilnya=[makespanela,makespane2b,makespane3c,makespanela1,makespane2b1,makespane3c1,makespanela2,makespane2b2,makespane3c2];

```

```

        s=sort(Hasilnya);    %menentukan makespan terkecil
        makespan=s(1,1)      %menampilkan hasil makespan
    terkecil

    toc
elseif x==4;    %jika ada 4 makespan terkecil
    for i=1;
        data_awal1=data;
        data1=data2(i,:);
        data1=reshape(data1,J,M);
        data1(1,:)=[];
        urut1a=urut(i,:);
        urut1a1=urut1a(2:4);
        urutan_pour=urut(1,1);

        [makespanela,urutan_pour1a,makespane2b,urutan_pour2b,makespane3c,u
        rutan_pour3c]=permasalahan_3job_pada4job_4sama1(data1,urutan_pour,
        data_awal1,urut1a1);
        end

        for i=2;
            data_awal2=data;
            data3=data2(i,:);
            data3=reshape(data3,J,M);
            data3(1,:)=[];
            urut2a=urut(i,:);
            urut2a1=urut2a(2:4);
            urutan_pour2=urut2a(1,1);

            [makespanela1,urutan_pour1a1,makespane2b1,urutan_pour2b1,makespane
            3c1,urutan_pour3c1]=permasalahan_3job_pada4job_4sama2(data3,urutan
            _pour2,data_awal2,urut2a1);
            end

            for i=3;
                data_awal3=data;
                data5=data2(i,:);
                data5=reshape(data5,J,M);
                data5(1,:)=[];
                urut3a=urut(i,:);
                urut3a1=urut3a(2:4);
                urutan_pour3=urut3a(1,1);

                [makespanela2,urutan_pour1a2,makespane2b2,urutan_pour2b2,makespane
                3c2,urutan_pour3c2]=permasalahan_3job_pada4job_4sama3(data5,urutan
                _pour3,data_awal3,urut3a1);
                end

            for i=4;
                data_awal4=data;
                data7=data2(i,:);
                data7=reshape(data7,J,M);
                data7(1,:)=[];
                urut4a=urut(i,:);
                urut4a1=urut4a(2:4);
                urutan_pour4=urut4a(1,1);

```

```
[makespanela3,urutan_pour1a3,makespane2b3,urutan_pour2b3,makespane
3c3,urutan_pour3c3]=permasalahan_3job_pada4job_4sama4(data7,urutan
_pour4,data_awal4,urut4a1);
```

```
end
```

```
Hasilnya=[makespanela;makespane2b;makespane3c;makespanela1;makespa
ne2b1;makespane3c1;makespanela2;makespane2b2;makespane3c2;makespan
e1a3;makespane2b3;makespane3c3];
```

```
    s=sort(Hasilnya); %menentukan makespan terkecil
```

```
    makespane=s(1,1) %menampilkan hasil makespan
```

```
terkecil
```

```
end
```

```
toc
```


Lampiran 10.1

Function Program Iterasi 1 (A)

```
function
[makespane1a,urutan_pour1a,makespane2b,urutan_pour2b,makespane3c,u
rutan_pour3c]=permasalahan_3job_pada4job_4samar(data1,urutan_pour,
data_awal,urut1a1)

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0;
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
%menentukan waktu minimum tiap mesin dan penambahan waktu proses
tiap Pij
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
```

```

        end
    end

    Total=sum(Data_baru,2); %menghitung jumlah waktu proses tiap
    job
    Total1=sort(Total);urutan=Urut(2:J); %mengurutkan jumlah %waktu
    proses tiap job dari terkecil ke terbesar
    urutan_baru(1)=i;
    for i2=1:J-1
        for i1=1:J-1
            a=Total(i1); b=Total1(i2);
            if a==b
                urutan_baru(i2+1)=urutan(i1);
            end
        end
    end
    Hasil(i,:)=urutan_baru; %menentukan urutan sementara
    for i1=1:J
        i3=urutan_baru(i1);
        Data(i1,:)=data(i3,:);
        Simpanan_Dat(i,:)=Data(:);
    end
    % Membuat Makespan

    for i1=1:M
        S(1,1)=0;
        if i1==1
            S(1,1)=0;
            for it=2:J
                S(it,1)=S(it-1,1)+Data(it-1,i1);
            end
            E(1:J-1,1)=S(2:J,1);
            E(J,1)=S(J,1)+Data(J,1);
        else
            for i2=1:J
                if i2==1
                    S(i2,i1)=E(i2,i1-1);
                    E(i2,i1)=S(i2,i1)+Data(i2,i1);
                else
                    S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                    E(i2,i1)=S(i2,i1)+Data(i2,i1);
                end
            end
        end
    end

    end
    makespane=E(J,M);
    %clear data_urut
    [Simpanan_E(i,:)]=makespane;
    Hasil_urutan=Hasil;
    Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E) %menentukan makespan terkecil
[d1 d2]=size(s);

%menentukan data dan urutan sesuai dengan makespan terkecil

```

```

d1==3
    [p,q]=urut_3_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==3;
    [data2,urut]=data_terkecil_3job(s,p,q);
    [x y]=size(data2);

if x==1
    for i=1
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

        [a b]=size(urut1a1) ;
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3
            for i=1:q
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2)
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3)
                    x1=urut1a1(1,3);
                end
                data_pour1(i,:)=x1;
            end
        end

        urutan_pour1=[urutan_pour;data_pour1(1,1)];
        urut1=data_pour1(2:3);
        urut1=urut1';

[makespan1a,urutan_pour1a]=iterasilala(data1,urut1,urutan_pour1,d
ata_awal1)
    end

    makespan2b=[];
    urutan_pour2b=[];
    makespan3c=[];
    urutan_pour3c=[];

elseif x==2
    for i=1
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

```

```

[a b]=size(urut1a1) ;
urutan=1:b;
urut11=urut(i,:);
[p q]=size(urut);

if q==3
    for i=1:q
        if urut11(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut11(1,i)==urutan(1,2)
            x1=urut1a1(1,2);
        elseif urut11(1,i)==urutan(1,3)
            x1=urut1a1(1,3);
        end
        data_pour1(i,:)=x1;
    end
end

urutan_pour1=[urutan_pour;data_pour1(1,1)];
urut1=data_pour1(2:3);
urut1=urut1';

[makespanela,urutan_pour1a]=iterasilala(data1,urut1,urutan_pour1,d
ata_awal1)
end

for i=2
    data_awal2=data_awal;
    data31=data2(i,:);
    urut2=urut(i,:);
    data3=reshape(data31,J,M);
    data3(1,:)=[];

    [a b]=size(urut1a1) ;
    urutan=1:b;
    urut12=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q
            if urut12(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut12(1,i)==urutan(1,2)
                x1=urut1a1(1,2);
            elseif urut12(1,i)==urutan(1,3)
                x1=urut1a1(1,3);
            end
            data_pour2(i,:)=x1;
        end
    end

    urutan_pour2=[urutan_pour;data_pour2(1,1)];
    urut2=data_pour2(2:3);

```

```

        urut2=urut2'

[makespane2b,urutan_pour2b]=iterasila2b(data3,urut2,urutan_pour2,d
ata_awal2)
    end

    makespane3c=[];
    urutan_pour3c=[];

elseif x==3
    for i=1
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

        [a b]=size(urut1a1) ;
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3
            for i=1:q
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2)
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3)
                    x1=urut1a1(1,3);
                end
                data_pour1(i,:)=x1;
            end
        end

        urutan_pour1=[urutan_pour;data_pour1(1,1)];
        urut1=data_pour1(2:3);
        urut1=urut1';

[makespanela,urutan_pour1a]=iterasila1a(data1,urut1,urutan_pour1,d
ata_awal1)
    end

    for i=2
        data_awal2=data_awal;
        data31=data2(i,:);
        urut2=urut(i,:);
        data3=reshape(data31,J,M);
        data3(1,:)=[];

        [a b]=size(urut1a1) ;
        urutan=1:b;
        urut12=urut(i,:);
        [p q]=size(urut);

```

```

        if q==3
            for i=1:q
                if urut12(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut12(1,i)==urutan(1,2)
                    x1=urut1a1(1,2);
                elseif urut12(1,i)==urutan(1,3)
                    x1=urut1a1(1,3);
                end
                data_pour2(i,:)=x1;
            end
        end

        urutan_pour2=[urutan_pour;data_pour2(1,1)];
        urut2=data_pour2(2:3);
        urut2=urut2';

[makespane2b,urutan_pour2b]=iterasila2b(data3,urut2,urutan_pour2,d
ata_awal2)

    for i=3
        data_awal3=data_awal;
        data51=data2(i,:);
        urut4=urut(i,:);
        data5=reshape(data51,J,M);
        data5(1,:)=[];

        [a b]=size(urut1a1)      ;
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3
            for i=1:q
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2)
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3)
                    x1=urut1a1(1,3);
                end
                data_pour3(i,:)=x1;
            end
        end

        urutan_pour3=[urutan_pour;data_pour3(1,1)];
        urut3=data_pour3(2:3);
        urut3=urut3';

[makespane3c,urutan_pour3c]=iterasila3c(data5,urut3,urutan_pour3,d
ata_awal3)
    end
end
end

```

Lampiran 10.2

Function Program Iterasi 1 (B)

```
function
[makespanela1,urutan_pour1a1,makespane2b1,urutan_pour2b1,makespane
3c1,urutan_pour3c1]=permasalahan_3job_pada4job_4sama2(data3,urutan
_pour2,data_awal2,urut2a1)

data1=data3;
urutan_pour=urutan_pour2;
data_awal=data_awal2;
urut1a1=urut2a1;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0;
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
%menentukan waktu minimum tiap mesin dan penambahan waktu proses
tiap Pij
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
```

```

        b=Data_baru(i2,k);
        if a1==b
            Data_baru(i2,k)=jml+Data_baru(i2,k);
            jml=Data_baru(i2,k);
            data2=[Dat] ;
        end
    end
end

Total=sum(Data_baru,2); %menghitung jumlah waktu proses tiap
job
Total1=sort(Total);urutan=Urut(2:J); %mengurutkan jumlah %waktu
proses tiap job dari terkecil ke terbesar
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
urutan_baru(i,:)=urutan_baru; %menentukan urutan sementara
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;

```



```

        Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E) %menentukan makespan terkecil
[d1 d2]=size(s);

%menentukan data dan urutan sesuai dengan makespan terkecil

d1==3
    [p,q]=urut_3_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==3;
    [data2,urut]=data_terkecil_3job(s,p,q);
    [x y]=size(data2);

if x==1
for i=1;
    data_awal1=data_awal;
    data11=data2(i,:);
    urut1=urut(i,:);
    data1=reshape(data11,J,M);
    data1(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut11=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q;
            if urut11(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut11(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut11(1,i)==urutan(1,3);
                x1=urut1a1(1,3);
            end
            data_pour1(i,:)=x1;
        end
    end

    urutan_pour1=[urutan_pour;data_pour1(1,1)];
    urut1=data_pour1(2:3);
    urut1=urut1';

[makespane1a1,urutan_pour1a1]=iterasilala_4sama2(data1,urut1,uruta
n_pour1,data_awal1)
end

makespane2b1=[];
urutan_pour2b1=[];
makespane3c1=[];
urutan_pour3c1=[];

```

```

elseif x==2
for i=1
    data_awal1=data_awal;
    data11=data2(i,:);
    urut1=urut(i,:);
    data1=reshape(data11,J,M);
    data1(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut11=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q
            if urut11(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut11(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut11(1,i)==urutan(1,3);
                x1=urut1a1(1,3);
            end
            data_pour1(i,:)=x1;
        end
    end

    urutan_pour1=[urutan_pour;data_pour1(1,1)];
    urut1=data_pour1(2:3);
    urut1=urut1';

[makespanela1,urutan_pour1a1]=iterasilala_4sama2(data1,urut1,uruta
n_pour1,data_awal1)
end

for i=2
    data_awal2=data_awal;
    data31=data2(i,:);
    urut2=urut(i,:);
    data3=reshape(data31,J,M);
    data3(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut12=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q
            if urut12(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut12(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut12(1,i)==urutan(1,3);

```

```

        x1=urut1a1(1,3);
    end
    data_pour2(i,:)=x1;
end
end

urutan_pour2=[urutan_pour;data_pour2(1,1)];
urut2=data_pour2(2:3);
urut2=urut2';

[makespane2b1,urutan_pour2b1]=iterasila2b_4sama2(data3,urut2,uruta
n_pour2,data_awal2)
end

makespane3c1=[];
urutan_pour3c1=[];

elseif x==3
    for i=1
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

        [a b]=size(urut1a1) ;
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3
            for i=1:q
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2);
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3)
                    x1=urut1a1(1,3);
                end
                data_pour1(i,:)=x1;
            end
        end

        urutan_pour1=[urutan_pour;data_pour1(1,1)];
        urut1=data_pour1(2:3);
        urut1=urut1';

        [makespanela1,urutan_pour1a1]=iterasilala_4sama2(data1,urut1,uruta
n_pour1,data_awal1)
        end

        for i=2;
            data_awal2=data_awal;
            data31=data2(i,:);

```

```

urut2=urut(i,:);
data3=reshape(data31,J,M);
data3(1,:)=[];

[a b]=size(urut1a1);
urutan=1:b;
urut12=urut(i,:);
[p q]=size(urut);

if q==3
    for i=1:q
        if urut12(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut12(1,i)==urutan(1,2);
            x1=urut1a1(1,2);
        elseif urut12(1,i)==urutan(1,3);
            x1=urut1a1(1,3);
        end
        data_pour2(i,:)=x1;
    end
end

urutan_pour2=[urutan_pour;data_pour2(1,1)];
urut2=data_pour2(2:3);
urut2=urut2';

[makespane2b1,urutan_pour2b1]=iterasila2b_4sama2(data3,urut2,uruta
n_pour2,data_awal2)

for i=3;
    data_awal3=data_awal;
    data51=data2(i,:);
    urut4=urut(i,:);
    data5=reshape(data51,J,M);
    data5(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut11=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q;
            if urut11(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut11(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut11(1,i)==urutan(1,3);
                x1=urut1a1(1,3);
            end
            data_pour3(i,:)=x1;
        end
    end

    urutan_pour3=[urutan_pour;data_pour3(1,1)];

```

```
        urut3=data_pour3(2:3);  
        urut3=urut3';  
  
        [makespane3c1,urutan_pour3c1]=iterasila3c_4sama2(data5,urut3,uruta  
n_pour3,data_awal3)  
    end  
end  
end
```



Lampiran 10.3

Function Program Iterasi 1 (C)

```
function
[makespan1a2,urutan_pour1a2,makespan2b2,urutan_pour2b2,makespan3c2,urutan_pour3c2]=permasalahan_3job_pada4job_4sama3(data5,urutan_pour3,data_awal3,urut3a1)

data1=data5;
urutan_pour=urutan_pour3;
data_awal=data_awal3;
urut1a1=urut3a1;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];
for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0;
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
%menentukan waktu minimum tiap mesin dan penambahan waktu proses
tiap Pij
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
```

```

                                jml=Data_baru(i2,k);
                                data2=[Dat] ;
                                end
                            end
                        end
                    end
                end

                Total=sum(Data_baru,2); %menghitung jumlah waktu proses tiap
                job
                Total1=sort(Total);urutan=Urut(2:J); %mengurutkan jumlah %waktu
                proses tiap job dari terkecil ke terbesar
                urutan_baru(1)=i;
                for i2=1:J-1
                    for i1=1:J-1
                        a=Total(i1); b=Total1(i2);
                        if a==b
                            urutan_baru(i2+1)=urutan(i1);
                        end
                    end
                end
                Hasil(i,:)=urutan_baru; %menentukan urutan sementara
                for i1=1:J
                    i3=urutan_baru(i1);
                    Data(i1,:)=data(i3,:);
                    Simpanan_Dat(i,:)=Data(:);
                end
                % Membuat Makespan

                for i1=1:M
                    S(1,1)=0;
                    if i1==1
                        S(1,1)=0;
                        for it=2:J
                            S(it,1)=S(it-1,1)+Data(it-1,i1);
                        end
                        E(1:J-1,1)=S(2:J,1);
                        E(J,1)=S(J,1)+Data(J,1);
                    else
                        for i2=1:J
                            if i2==1
                                S(i2,i1)=E(i2,i1-1);
                                E(i2,i1)=S(i2,i1)+Data(i2,i1);
                            else
                                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                                E(i2,i1)=S(i2,i1)+Data(i2,i1);
                            end
                        end
                    end
                end
                makespane=E(J,M);
                %clear data_urut
                [Simpanan_E(i,:)]=makespane;
                Hasil_urutan=Hasil;
                Data_diambil=Simpanan_Dat;
            end
        end
    end
end

```

```

s=sort(Simpanan_E) %menentukan makespan terkecil
[d1 d2]=size(s);

%menentukan data dan urutan sesuai dengan makespan terkecil

d1==3
    [p,q]=urut_3_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==3;
    [data2,urut]=data_terkecil_3job(s,p,q);
    [x y]=size(data2);

if x==1
    for i=1;
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

        [a b]=size(urut1a1);
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3
            for i=1:q;
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2);
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3);
                    x1=urut1a1(1,3);
                end
                data_pour1(i,:)=x1;
            end
        end

        urutan_pour1=[urutan_pour;data_pour1(1,1)];
        urut1=data_pour1(2:3);
        urut1=urut1';

        [makespan1a2,urutan_pour1a2]=iterasilala_4sama3(data1,urut1,uruta
n_pour1,data_awal1)
        end

        makespane2b2=[];
        urutan_pour2b2=[];
        makespane3c2=[];
        urutan_pour3c2=[];

elseif x==2
    for i=1

```



```

data_awal1=data_awal;
data11=data2(i,:);
urut1=urut(i,:);
data1=reshape(data11,J,M);
data1(1,:)=[];

[a b]=size(urut1a1);
urutan=1:b;
urut11=urut(i,:);
[p q]=size(urut);

if q==3
    for i=1:q
        if urut11(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut11(1,i)==urutan(1,2);
            x1=urut1a1(1,2);
        elseif urut11(1,i)==urutan(1,3);
            x1=urut1a1(1,3);
        end
        data_pour1(i,:)=x1;
    end
end

urutan_pour1=[urutan_pour;data_pour1(1,1)];
urut1=data_pour1(2:3);
urut1=urut1';

[makespanela2,urutan_pour1a2]=iterasilala_4sama3(data1,urut1,uruta
n_pour1,data_awal1)
end

for i=2
    data_awal2=data_awal;
    data31=data2(i,:);
    urut2=urut(i,:);
    data3=reshape(data31,J,M);
    data3(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut12=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q;
            if urut12(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut12(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut12(1,i)==urutan(1,3);
                x1=urut1a1(1,3);
            end
            data_pour2(i,:)=x1;
        end
    end
end

```

```

        end
    end

    urutan_pour2=[urutan_pour;data_pour2(1,1)];
    urut2=data_pour2(2:3);
    urut2=urut2';

    [makespane2b2,urutan_pour2b2]=iterasila2b_4sama3(data3,urut2,urutan_pour2,data_awal2)
end

    makespane3c2=[];
    urutan_pour3c2=[];

elseif x==3
    for i=1
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

        [a b]=size(urut1a1);
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3
            for i=1:q
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2);
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3);
                    x1=urut1a1(1,3);
                end
                data_pour1(i,:)=x1;
            end
        end

        urutan_pour1=[urutan_pour;data_pour1(1,1)];
        urut1=data_pour1(2:3);
        urut1=urut1';

        [makespanela2,urutan_pour1a2]=iterasilala_4sama3(data1,urut1,urutan_pour1,data_awal1)
    end

    for i=2
        data_awal2=data_awal;
        data31=data2(i,:);
        urut2=urut(i,:);
        data3=reshape(data31,J,M);
        data3(1,:)=[];
    end
end

```

```

[a b]=size(urut1a1);
urutan=1:b;
urut12=urut(i,:);
[p q]=size(urut);

if q==3
    for i=1:q;
        if urut12(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut12(1,i)==urutan(1,2);
            x1=urut1a1(1,2);
        elseif urut12(1,i)==urutan(1,3);
            x1=urut1a1(1,3);
        end
        data_pour2(i,:)=x1;
    end
end

urutan_pour2=[urutan_pour;data_pour2(1,1)];
urut2=data_pour2(2:3);
urut2=urut2';

[makespane2b2,urutan_pour2b2]=iterasila2b_4sama3(data3,urut2,uruta
n_pour2,data_awal2)

for i=3
    data_awal3=data_awal;
    data51=data2(i,:);
    urut4=urut(i,:);
    data5=reshape(data51,J,M);
    data5(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut11=urut(i,:);
    [p q]=size(urut);

    if q==3
        for i=1:q;
            if urut11(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut11(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut11(1,i)==urutan(1,3);
                x1=urut1a1(1,3);
            end
            data_pour3(i,:)=x1;
        end
    end

    urutan_pour3=[urutan_pour;data_pour3(1,1)];
    urut3=data_pour3(2:3);
    urut3=urut3';

```

```
[makespane3c2,urutan_pour3c2]=iterasila3c_4sama3(data5,urut3,uruta  
n_pour3,data_awal3)  
    end  
end  
end
```



Lampiran 10.4

Function Program Iterasi 1 (D)

```
function
[makespan1a3,urutan_pour1a3,makespan2b3,urutan_pour2b3,makespan3c3,urutan_pour3c3]=permasalahan_3job_pada4job_4sama4(data7,urutan_pour4,data_awal4,urut4a1)

data1=data7;
urutan_pour=urutan_pour4;
data_awal=data_awal4;
urut1a1=urut4a1;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0;
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
%menentukan waktu minimum tiap mesin dan penambahan waktu proses
tiap Pij
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
```

```

        Data_baru(i2,k)=jml+Data_baru(i2,k);
        jml=Data_baru(i2,k);
        data2=[Dat] ;
    end
end
end
end

Total=sum(Data_baru,2); %menghitung jumlah waktu proses tiap
job
Total1=sort(Total);urutan=Urut(2:J); %mengurutkan jumlah %waktu
proses tiap job dari terkecil ke terbesar
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
end
Hasil(i,:)=urutan_baru; %menentukan urutan sementara
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

```

```

s=sort(Simpanan_E) %menentukan makespan terkecil
[d1 d2]=size(s);

%menentukan data dan urutan sesuai dengan makespan terkecil

d1==3
[p,q]=urut_3_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==3;
[data2,urut]=data_terkecil_3job(s,p,q);
[x y]=size(data2);

if x==1;
for i=1;
data_awal1=data_awal;
data11=data2(i,:);
urut1=urut(i,:);
data1=reshape(data11,J,M);
data1(1,:)=[];

[a b]=size(urut1a1);
urutan=1:b;
urut11=urut(i,:);
[p q]=size(urut);

if q==3;
for i=1:q;
if urut11(1,i)==urutan(1,1);
x1=urut1a1(1,1);
elseif urut11(1,i)==urutan(1,2);
x1=urut1a1(1,2);
elseif urut11(1,i)==urutan(1,3);
x1=urut1a1(1,3);
end
data_pour1(i,:)=x1;
end
end

urutan_pour1=[urutan_pour;data_pour1(1,1)];
urut1=data_pour1(2:3);
urut1=urut1';

[makespan1a3,urutan_pour1a3]=iterasilala_4sama4(data1,urut1,uruta
n_pour1,data_awal1)
end

makespane2b3=[];
urutan_pour2b3=[];
makespane3c3=[];
urutan_pour3c3=[];

elseif x==2

```

```

for i=1;
data_awal1=data_awal;
data11=data2(i,:);
urut1=urut(i,:);
data1=reshape(data11,J,M);
data1(1,:)=[];

[a b]=size(urut1a1);
urutan=1:b;
urut11=urut(i,:);
[p q]=size(urut);

if q==3;
    for i=1:q;
        if urut11(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut11(1,i)==urutan(1,2);
            x1=urut1a1(1,2);
        elseif urut11(1,i)==urutan(1,3);
            x1=urut1a1(1,3);
        end
        data_pour1(i,:)=x1;
    end
end

urutan_pour1=[urutan_pour;data_pour1(1,1)];
urut1=data_pour1(2:3);
urut1=urut1';

[makespanela3,urutan_pour1a3]=iterasilala_4sama4(data1,urut1,uruta
n_pour1,data_awal1)
end

for i=2;
data_awal2=data_awal;
data31=data2(i,:);
urut2=urut(i,:);
data3=reshape(data31,J,M);
data3(1,:)=[];

[a b]=size(urut1a1);
urutan=1:b;
urut12=urut(i,:);
[p q]=size(urut);

if q==3;
    for i=1:q;
        if urut12(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut12(1,i)==urutan(1,2);
            x1=urut1a1(1,2);
        elseif urut12(1,i)==urutan(1,3);
            x1=urut1a1(1,3);
        end
    end
end

```



```

        data_pour2(i,:)=x1;
    end
end

urutan_pour2=[urutan_pour;data_pour2(1,1)];
urut2=data_pour2(2:3);
urut2=urut2';

[makespane2b3,urutan_pour2b3]=iterasila2b_4sama4(data3,urut2,urutan_pour2,data_awal2)
end

makespane3c3=[];
urutan_pour3c3=[];

elseif x==3;
    for i=1;
        data_awal1=data_awal;
        data11=data2(i,:);
        urut1=urut(i,:);
        data1=reshape(data11,J,M);
        data1(1,:)=[];

        [a b]=size(urut1a1);
        urutan=1:b;
        urut11=urut(i,:);
        [p q]=size(urut);

        if q==3;
            for i=1:q;
                if urut11(1,i)==urutan(1,1);
                    x1=urut1a1(1,1);
                elseif urut11(1,i)==urutan(1,2);
                    x1=urut1a1(1,2);
                elseif urut11(1,i)==urutan(1,3);
                    x1=urut1a1(1,3);
                end
                data_pour1(i,:)=x1;
            end
        end

        urutan_pour1=[urutan_pour;data_pour1(1,1)];
        urut1=data_pour1(2:3);
        urut1=urut1';

        [makespanela3,urutan_pour1a3]=iterasila1a_4sama4(data1,urut1,urutan_pour1,data_awal1)
    end

    for i=2;
        data_awal2=data_awal;
        data31=data2(i,:);
        urut2=urut(i,:);
        data3=reshape(data31,J,M);

```

```

data3(1,:)=[];

[a b]=size(urut1a1);
urutan=1:b;
urut12=urut(i,:);
[p q]=size(urut);

if q==3;
    for i=1:q;
        if urut12(1,i)==urutan(1,1);
            x1=urut1a1(1,1);
        elseif urut12(1,i)==urutan(1,2);
            x1=urut1a1(1,2);
        elseif urut12(1,i)==urutan(1,3);
            x1=urut1a1(1,3);
        end
        data_pour2(i,:)=x1;
    end
end

urutan_pour2=[urutan_pour;data_pour2(1,1)];
urut2=data_pour2(2:3);
urut2=urut2';

[makespane2b3,urutan_pour2b3]=iterasila2b_4sama4(data3,urut2,uruta
n_pour2,data_awal2)
end
for i=3
    data_awal3=data_awal;
    data51=data2(i,:);
    urut4=urut(i,:);
    data5=reshape(data51,J,M);
    data5(1,:)=[];

    [a b]=size(urut1a1);
    urutan=1:b;
    urut11=urut(i,:);
    [p q]=size(urut);

    if q==3;
        for i=1:q;
            if urut11(1,i)==urutan(1,1);
                x1=urut1a1(1,1);
            elseif urut11(1,i)==urutan(1,2);
                x1=urut1a1(1,2);
            elseif urut11(1,i)==urutan(1,3);
                x1=urut1a1(1,3);
            end
            data_pour3(i,:)=x1;
        end
    end

    urutan_pour3=[urutan_pour;data_pour3(1,1)];
    urut3=data_pour3(2:3);
    urut3=urut3';

```

```
[makespane3c3,urutan_pour3c3]=iterasila3c_4sama4(data5,urut3,uruta  
n_pour3,data_awal3)  
    end  
end
```



Lampiran 10.5

Function Program Iterasi 2 (A1)

```
function
[makespanela,urutan_pourla]=iterasilala(data1,urut1,urutan_pour1,d
ata_awal1)

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J;
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M;
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
end
end
```

```

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1;
    for i1=1:J-1;
        a=Total(i1); b=Total1(i2);
        if a==b;
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1;
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E)
[d1 d2]=size(s);

d1==2
[p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

```

```

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1)      ;
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour1a=[urutan_pour1;baru2];

[a b]=size(data_awal1)      ;
urutan=1:a;
urutan_pour1a=urutan_pour1a';
[p q]=size(urutan_pour1a);

if q==4
    for i=1:q
        if urutan_pour1a(1,i)==urutan(1,1);
            x1=data_awal1(1,:);
        elseif urutan_pour1a(1,i)==urutan(1,2);
            x1=data_awal1(2,:);
        elseif urutan_pour1a(1,i)==urutan(1,3);
            x1=data_awal1(3,:);
        elseif urutan_pour1a(1,i)==urutan(1,4);
            x1=data_awal1(4,:);
        end
        data_pour(i,:)=x1;
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    end
end

```

```

else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespan1a=E(J,M);

```



Lampiran 10.6

Function Program Iterasi 2 (A2)

```
function
[makespane2b,urutan_pour2b]=iterasila2b(data3,urut2,urutan_pour2,d
ata_awal2)

data1=data3;
urut1=urut2;
urutan_pour=urutan_pour2;
[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
```



```

        end
    end
end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E)
[d1 d2]=size(s);

```

```

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1);
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour2b=[urutan_pour;baru2];

[a b]=size(data_awal2)      ;
urutan=1:a;
urutan_pour2b=urutan_pour2b';
[p q]=size(urutan_pour2b);

if q==4
    for i=1:q
        if urutan_pour2b(1,i)==urutan(1,1);
            x1=data_awal2(1,:);
        elseif urutan_pour2b(1,i)==urutan(1,2);
            x1=data_awal2(2,:);
        elseif urutan_pour2b(1,i)==urutan(1,3);
            x1=data_awal2(3,:);
        elseif urutan_pour2b(1,i)==urutan(1,4);
            x1=data_awal2(4,:);
        end
        data_pour(i,:)=x1
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J

```

```

        S(it,1)=S(it-1,1)+Data(it-1,i1);
    end
    E(1:J-1,1)=S(2:J,1);
    E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespane2b=E(J,M);

```

Lampiran 10.7

Function Program Iterasi 2 (A3)

```
function
[makespane3c,urutan_pour3c]=iterasila3c(data5,urut3,urutan_pour3,d
ata_awal3)

data1=data5;
urut1=urut3;
urutan_pour=urutan_pour3;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
```

```

                                jml=Data_baru(i2,k);
                                data2=[Dat] ;
                                end
                            end
                        end
                    end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

```

```

s=sort(Simpanan_E)
[d1 d2]=size(s);

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1)    ;
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour3c=[urutan_pour;baru2];

[a b]=size(data_awal3)    ;
urutan=1:a;
urutan_pour3c=urutan_pour3c';
[p q]=size(urutan_pour3c);

if q==4
    for i=1:q
        if urutan_pour3c(1,i)==urutan(1,1);
            x1=data_awal3(1,:);
        elseif urutan_pour3c(1,i)==urutan(1,2);
            x1=data_awal3(2,:);
        elseif urutan_pour3c(1,i)==urutan(1,3);
            x1=data_awal3(3,:);
        elseif urutan_pour3c(1,i)==urutan(1,4);
            x1=data_awal3(4,:);
        end
        data_pour(i,:)=x1
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
    end
end

```

```

for it=2:J
    S(it,1)=S(it-1,1)+Data(it-1,i1);
end
E(1:J-1,1)=S(2:J,1);
E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespane3c=E(J,M);

```

Lampiran 10.8

Function Program Iterasi 2 (B1)

```
function
[makespanela1,urutan_pour1a1]=iterasilala_4sama2(data1,urut1,uruta
n_pour1,data_awal1)

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
end
end
```



```

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E)
[d1 d2]=size(s);

d1==2
[p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

```

```

d1==2

[a b]=size(urut1) ;
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2);
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour1a1=[urutan_pour1;baru2];

[a b]=size(data_awal1) ;
urutan=1:a;
urutan_pour1a=urutan_pour1a1';
[p q]=size(urutan_pour1a);

if q==4
    for i=1:q
        if urutan_pour1a(1,i)==urutan(1,1);
            x1=data_awal1(1,:);
        elseif urutan_pour1a(1,i)==urutan(1,2)
            x1=data_awal1(2,:);
        elseif urutan_pour1a(1,i)==urutan(1,3)
            x1=data_awal1(3,:);
        elseif urutan_pour1a(1,i)==urutan(1,4)
            x1=data_awal1(4,:);
        end
        data_pour(i,:)=x1;
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J

```

```

        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end

makespanela1=E(J,M);

```



Lampiran 10.9

Function Program Iterasi 2 (B2)

```
function
[makespane2b1,urutan_pour2b1]=iterasila2b_4sama2(data3,urut2,uruta
n_pour2,data_awal2)

data1=data3;
urut1=urut2;
urutan_pour=urutan_pour2;
[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0;
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end
    Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
```

```

        end
    end
end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E)
[d1 d2]=size(s);

```

```

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1);
    urutan=1:b;
    [p q]=size(urut);

    if q==2
        for i=1:q
            if urut(1,i)==urutan(1,1);
                x1=urut1(1,1);
            elseif urut(1,i)==urutan(1,2)
                x1=urut1(1,2);
            end
            baru2(i,:)=x1;
        end
    end

    urutan_pour2b1=[urutan_pour;baru2];

[a b]=size(data_awal2)      ;
    urutan=1:a;
    urutan_pour2b1=urutan_pour2b1';
    [p q]=size(urutan_pour2b1);

    if q==4
        for i=1:q
            if urutan_pour2b1(1,i)==urutan(1,1);
                x1=data_awal2(1,:);
            elseif urutan_pour2b1(1,i)==urutan(1,2);
                x1=data_awal2(2,:);
            elseif urutan_pour2b1(1,i)==urutan(1,3);
                x1=data_awal2(3,:);
            elseif urutan_pour2b1(1,i)==urutan(1,4);
                x1=data_awal2(4,:);
            end
            data_pour(i,:)=x1
        end
    end

    data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1;
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);

```

```

end
    E(1:J-1,1)=S(2:J,1);
    E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespane2b1=E(J,M);

```

Lampiran 10.10

Function Program Iterasi 2 (B3)

```
function
[makespane3c1,urutan_pour3c1]=iterasila3c_4sama2(data5,urut3,urutan_pour3,data_awal3)

data1=data5;
urut1=urut3;
urutan_pour=urutan_pour3;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
```



```

                                jml=Data_baru(i2,k);
                                data2=[Dat] ;
                                end
                            end
                        end
                    end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

```

```

s=sort(Simpanan_E);
[d1 d2]=size(s);

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q)

    [a b]=size(urut1)    ;
    urutan=1:b;
    [p q]=size(urut);

    if q==2
        for i=1:q
            if urut(1,i)==urutan(1,1);
                x1=urut1(1,1);
            elseif urut(1,i)==urutan(1,2)
                x1=urut1(1,2);
            end
            baru2(i,:)=x1;
        end
    end

    urutan_pour3c1=[urutan_pour;baru2];

[a b]=size(data_awal3)    ;
    urutan=1:a;
    urutan_pour3c1=urutan_pour3c1';
    [p q]=size(urutan_pour3c1);

    if q==4
        for i=1:q
            if urutan_pour3c1(1,i)==urutan(1,1);
                x1=data_awal3(1,:);
            elseif urutan_pour3c1(1,i)==urutan(1,2);
                x1=data_awal3(2,:);
            elseif urutan_pour3c1(1,i)==urutan(1,3);
                x1=data_awal3(3,:);
            elseif urutan_pour3c1(1,i)==urutan(1,4);
                x1=data_awal3(4,:);
            end
            data_pour(i,:)=x1
        end
    end

    data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1

```

```

S(1,1)=0;
for it=2:J
    S(it,1)=S(it-1,1)+Data(it-1,i1);
end
E(1:J-1,1)=S(2:J,1);
E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespane3c1=E(J,M);

```

Lampiran 10.11

Function Program Iterasi 2 (C1)

```
function
[makespan1a2,urutan_pour1a2]=iterasilala_4sama3(data1,urut1,uruta
n_pour1,data_awal1)

[J M]=size(data1)
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
end
end
```

```

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E);
[d1 d2]=size(s);

d1==2
[p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

```

```

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1);
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1
    end
end

urutan_pour1a2=[urutan_pour1;baru2];

[a b]=size(data_awal1);
urutan=1:a;
urutan_pour1a=urutan_pour1a2';
[p q]=size(urutan_pour1a);

if q==4
    for i=1:q
        if urutan_pour1a(1,i)==urutan(1,1);
            x1=data_awal1(1,:);
        elseif urutan_pour1a(1,i)==urutan(1,2)
            x1=data_awal1(2,:);
        elseif urutan_pour1a(1,i)==urutan(1,3)
            x1=data_awal1(3,:);
        elseif urutan_pour1a(1,i)==urutan(1,4)
            x1=data_awal1(4,:);
        end
        data_pour(i,:)=x1;
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
    end
end

```

```

        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespanela2=E(J,M);

```

Lampiran 10.12

Function Program Iterasi 2 (C2)

```
function
[makespane2b2,urutan_pour2b2]=iterasila2b_4sama3(data3,urut2,uruta
n_pour2,data_awal2)

data1=data3;
urut1=urut2;
urutan_pour=urutan_pour2;
[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end
    Simpanan_urutan(i,:)=Urut
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
```



```

        end
    end
end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane
Hasil_urutan=Hasil
Data_diambil=Simpanan_Dat
end

s=sort(Simpanan_E)
[d1 d2]=size(s);

```

```

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1)    ;
    urutan=1:b;
    [p q]=size(urut);

    if q==2
        for i=1:q
            if urut(1,i)==urutan(1,1);
                x1=urut1(1,1);
            elseif urut(1,i)==urutan(1,2)
                x1=urut1(1,2);
            end
            baru2(i,:)=x1;
        end
    end

    urutan_pour2b2=[urutan_pour;baru2];

[a b]=size(data_awal2)    ;
    urutan=1:a;
    urutan_pour2b2=urutan_pour2b2';
    [p q]=size(urutan_pour2b2);

    if q==4
        for i=1:q
            if urutan_pour2b2(1,i)==urutan(1,1);
                x1=data_awal2(1,:);
            elseif urutan_pour2b2(1,i)==urutan(1,2);
                x1=data_awal2(2,:);
            elseif urutan_pour2b2(1,i)==urutan(1,3);
                x1=data_awal2(3,:);
            elseif urutan_pour2b2(1,i)==urutan(1,4);
                x1=data_awal2(4,:);
            end
            data_pour(i,:)=x1
        end
    end

    data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);

```

```

end
    E(1:J-1,1)=S(2:J,1);
    E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespane2b2=E(J,M);

```

Lampiran 10.13

Function Program Iterasi 2 (C3)

```
function
[makespane3c2,urutan_pour3c2]=iterasila3c_4sama3(data5,urut3,urutan_pour3,data_awal3)

data1=data5;
urut1=urut3;
urutan_pour=urutan_pour3;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
```

```

                                jml=Data_baru(i2,k);
                                data2=[Dat] ;
                                end
                            end
                        end
                    end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

```

```

s=sort(Simpanan_E);
[d1 d2]=size(s);

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1);
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour3c2=[urutan_pour;baru2];

[a b]=size(data_awal3) ;
urutan=1:a;
urutan_pour3c2=urutan_pour3c2';
[p q]=size(urutan_pour3c2);

if q==4
    for i=1:q
        if urutan_pour3c2(1,i)==urutan(1,1);
            x1=data_awal3(1,:);
        elseif urutan_pour3c2(1,i)==urutan(1,2);
            x1=data_awal3(2,:);
        elseif urutan_pour3c2(1,i)==urutan(1,3);
            x1=data_awal3(3,:);
        elseif urutan_pour3c2(1,i)==urutan(1,4);
            x1=data_awal3(4,:);
        end
        data_pour(i,:)=x1
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;

```

```

if i1==1
    S(1,1)=0;
    for it=2:J
        S(it,1)=S(it-1,1)+Data(it-1,i1);
    end
    E(1:J-1,1)=S(2:J,1);
    E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end

makespane3c2=E(J,M);

```

Lampiran 10.14

Function Program Iterasi 2 (D1)

```
function
[makespan1a3,urutan_pour1a3]=iterasilala_4sama4(data1,urut1,uruta
n_pour1,data_awal1)

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

    Simpanan_urutan(i,:)=Urut
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
end
end
```



```

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E)
[d1 d2]=size(s);

d1==2
[p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

```

```

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1) ;
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour1a3=[urutan_pour1;baru2];

[a b]=size(data_awal1);
urutan=1:a;
urutan_pour1a3=urutan_pour1a3';
[p q]=size(urutan_pour1a3);

if q==4
    for i=1:q
        if urutan_pour1a3(1,i)==urutan(1,1);
            x1=data_awal1(1,:);
        elseif urutan_pour1a3(1,i)==urutan(1,2)
            x1=data_awal1(2,:);
        elseif urutan_pour1a3(1,i)==urutan(1,3)
            x1=data_awal1(3,:);
        elseif urutan_pour1a3(1,i)==urutan(1,4)
            x1=data_awal1(4,:);
        end
        data_pour(i,:)=x1;
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
    end
end

```

```

        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespanela3=E(J,M);

```

Lampiran 10.15

Function Program Iterasi 2 (D2)

```
function
[makespane2b3,urutan_pour2b3]=iterasila2b_4sama4(data3,urut2,uruta
n_pour2,data_awal2)

data1=data3;
urut1=urut2;
urutan_pour=urutan_pour2;
[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end
    Simpanan_urutan(i,:)=Urut
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
                jml=Data_baru(i2,k);
                data2=[Dat] ;
            end
        end
    end
end
```

```

        end
    end
end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan

for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

s=sort(Simpanan_E);
[d1 d2]=size(s);

```

```

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

    [a b]=size(urut1);
    urutan=1:b;
    [p q]=size(urut);

    if q==2
        for i=1:q
            if urut(1,i)==urutan(1,1);
                x1=urut1(1,1);
            elseif urut(1,i)==urutan(1,2)
                x1=urut1(1,2);
            end
            baru2(i,:)=x1;
        end
    end

    urutan_pour2b3=[urutan_pour;baru2];

    [a b]=size(data_awal2)      ;
    urutan=1:a;
    urutan_pour2b3=urutan_pour2b3';
    [p q]=size(urutan_pour2b3);

    if q==4
        for i=1:q
            if urutan_pour2b3(1,i)==urutan(1,1);
                x1=data_awal2(1,:);
            elseif urutan_pour2b3(1,i)==urutan(1,2);
                x1=data_awal2(2,:);
            elseif urutan_pour2b3(1,i)==urutan(1,3);
                x1=data_awal2(3,:);
            elseif urutan_pour2b3(1,i)==urutan(1,4);
                x1=data_awal2(4,:);
            end
            data_pour(i,:)=x1
        end
    end

    data=data_pour;

    [J M]=size(data); Data=data;
    for i1=1:M
        S(1,1)=0;
        if i1==1
            S(1,1)=0;
            for it=2:J
                S(it,1)=S(it-1,1)+Data(it-1,i1);
            end
        end
    end

```

```

end
    E(1:J-1,1)=S(2:J,1);
    E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end;
    end
end
end
makespane2b3=E(J,M);

```

Lampiran 10.16

Function Program Iterasi 2 (D3)

```
function
[makespane3c3,urutan_pour3c3]=iterasila3c_4sama4(data5,urut3,urutan_pour3,data_awal3)

data1=data5;
urut1=urut3;
urutan_pour=urutan_pour3;

[J M]=size(data1);
Data=data1; Data_baru=[];urutan=[];

for i=1:J;
% membuat data baru untuk data ke-i di anggap sebagai awalan
i2=0
for i1=1:J
    if i==i1
        continue
    elseif i1>i
        i2=i2+1;
        Data_baru(i2,:)=data1(i1,:);
        urutan(i2)=i1;
    end
    if i1==J && i2~=J-1
        Data_baru(i2+1:J-1,:)=data1(1:i-1,:);
        urutan(i2+1:J-1)=1:i-1;
    end
end
end

Dat(1,:)=data1(i,:);
Dat(2:J,:)=Data_baru;
Urut(1)=i; Urut(2:J)=urutan;
if i==J
    Data_baru(1:J-1,:)=data1(1:J-1,:);
    Dat(2:J,:)=Data_baru;
    Urut(2:J)=1:J-1;
end

Simpanan_urutan(i,:)=Urut;
% menentukan waktu minimum tiap mesin dan membuat data baru nya
for k=1:M
    waktu_min(k)=min(Data_baru(:,k));
    ab=sort(Data_baru(:,k));
    a=waktu_min(k); jml=0;
    for i1=1:J-1
        a1=ab(i1);
        for i2=1:J-1
            b=Data_baru(i2,k);
            if a1==b
                Data_baru(i2,k)=jml+Data_baru(i2,k);
            end
        end
    end
end
```



```

                                jml=Data_baru(i2,k);
                                data2=[Dat] ;
                                end
                            end
                        end
                    end

% Menentukan Urutan Job
Total=sum(Data_baru,2);
Total1=sort(Total);urutan=Urut(2:J);
urutan_baru(1)=i;
for i2=1:J-1
    for i1=1:J-1
        a=Total(i1); b=Total1(i2);
        if a==b
            urutan_baru(i2+1)=urutan(i1);
        end
    end
end
Hasil(i,:)=urutan_baru;
for i1=1:J
    i3=urutan_baru(i1);
    Data(i1,:)=data1(i3,:);
    Simpanan_Dat(i,:)=Data(:);
end
% Membuat Makespan
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
        for it=2:J
            S(it,1)=S(it-1,1)+Data(it-1,i1);
        end
        E(1:J-1,1)=S(2:J,1);
        E(J,1)=S(J,1)+Data(J,1);
    else
        for i2=1:J
            if i2==1
                S(i2,i1)=E(i2,i1-1);
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            else
                S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
                E(i2,i1)=S(i2,i1)+Data(i2,i1);
            end
        end
    end
end
makespane=E(J,M);
%clear data_urut
[Simpanan_E(i,:)]=makespane;
Hasil_urutan=Hasil;
Data_diambil=Simpanan_Dat;
end

```

```

s=sort(Simpanan_E);
[d1 d2]=size(s);

d1==2
    [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan);

d1==2
    [data,urut]=data_terkecil_2job(s,p,q);

[a b]=size(urut1)    ;
urutan=1:b;
[p q]=size(urut);

if q==2
    for i=1:q
        if urut(1,i)==urutan(1,1);
            x1=urut1(1,1);
        elseif urut(1,i)==urutan(1,2)
            x1=urut1(1,2);
        end
        baru2(i,:)=x1;
    end
end

urutan_pour3c3=[urutan_pour;baru2];

[a b]=size(data_awal3)    ;
urutan=1:a;
urutan_pour3c3=urutan_pour3c3';
[p q]=size(urutan_pour3c3);

if q==4
    for i=1:q
        if urutan_pour3c3(1,i)==urutan(1,1);
            x1=data_awal3(1,:);
        elseif urutan_pour3c3(1,i)==urutan(1,2);
            x1=data_awal3(2,:);
        elseif urutan_pour3c3(1,i)==urutan(1,3);
            x1=data_awal3(3,:);
        elseif urutan_pour3c3(1,i)==urutan(1,4);
            x1=data_awal3(4,:);
        end
        data_pour(i,:)=x1
    end
end

data=data_pour;

[J M]=size(data); Data=data;
for i1=1:M
    S(1,1)=0;
    if i1==1
        S(1,1)=0;
    end
end

```

```

for it=2:J
    S(it,1)=S(it-1,1)+Data(it-1,i1);
end
E(1:J-1,1)=S(2:J,1);
E(J,1)=S(J,1)+Data(J,1);
else
    for i2=1:J
        if i2==1
            S(i2,i1)=E(i2,i1-1);
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        else
            S(i2,i1)=max(E(i2,i1-1),E(i2-1,i1));
            E(i2,i1)=S(i2,i1)+Data(i2,i1);
        end
    end
end
end
makespane3c3=E(J,M);

```

Lampiran 10.17

Function Mengurutkan 2 Job

%Rumus mengurutkan untuk 2 job

```
function [p,q]=urut_2_job(s,Simpanan_E,Data_diambil,Hasil_urutan)

if s(1,1)==Simpanan_E(1);
    x5=Data_diambil(1,:);
    e=Hasil_urutan(1,:);
    Data_diambil(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
elseif s(1,1)==Simpanan_E(2);
    x5=Data_diambil(2,:);
    e=Hasil_urutan(2,:);
    Data_diambil(2,:)=[];
    Hasil_urutan(2,:)=[];
    s(1,:)=[];
    Simpanan_E(2,:)=[];
end

if s(1,1)==Simpanan_E(1);
    x6=Data_diambil(1,:);
    f=Hasil_urutan(1,:);
    Data_diambil(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
end

p=[x5; x6];
q=[e; f];
```

Lampiran 10.18

Function Mengurutkan 3 Job

```
%Rumus mengurutkan 3 job
function [p,q]=urut_3_job(s,Simpanan_E,Data_diambil,Hasil_urutan)

if s(1,1)==Simpanan_E(1);
    x4=Data_diambil(1,:);
    d=Hasil_urutan(1,:);
    Data_diambil(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
elseif s(1,1)==Simpanan_E(2);
    x4=Data_diambil(2,:);
    d=Hasil_urutan(2,:);
    Data_diambil(2,:)=[];
    Hasil_urutan(2,:)=[];
    s(1,:)=[];
    Simpanan_E(2,:)=[];
elseif s(1,1)==Simpanan_E(3);
    x4=Data_diambil(3,:);
    d=Hasil_urutan(3,:);
    Data_diambil(3,:)=[];
    Hasil_urutan(3,:)=[];
    s(1,:)=[];
    Simpanan_E(3,:)=[];
end

if s(1,1)==Simpanan_E(1);
    x5=Data_diambil(1,:);
    e=Hasil_urutan(1,:);
    Data_diambil(1,:)=[];
    Hasil_urutan(1,:)=[];
    s(1,:)=[];
    Simpanan_E(1,:)=[];
elseif s(1,1)==Simpanan_E(2);
    x5=Data_diambil(2,:);
    e=Hasil_urutan(2,:);
    Data_diambil(2,:)=[];
    Hasil_urutan(2,:)=[];
    s(1,:)=[];
    Simpanan_E(2,:)=[];
end
```

```
if s(1,1)==Simpanan_E(1);  
    x6=Data_diambil(1,:);  
    f=Hasil_urutan(1,:);  
    Data_diambil(1,:)=[];  
    Hasil_urutan(1,:)=[];  
    s(1,:)=[];  
    Simpanan_E(1,:)=[];  
end  
  
p=[x4; x5; x6];  
q=[d; e; f];
```



Lampiran 10.19

Function Mengurutkan 4 Job

%Rumus mengurutkan 4 job

```
function [p,q]=urut_4_job(s,Simpanan_E,Data_diambil,Hasil_urutan)
```

```
    if s(1,1)==Simpanan_E(1);
        x3=Data_diambil(1,:);
        c=Hasil_urutan(1,:);
        Data_diambil(1,:)=[];
        Hasil_urutan(1,:)=[];
        s(1,:)=[];
        Simpanan_E(1,:)=[];
    elseif s(1,1)==Simpanan_E(2);
        x3=Data_diambil(2,:);
        c=Hasil_urutan(2,:);
        Data_diambil(2,:)=[];
        Hasil_urutan(2,:)=[];
        s(1,:)=[];
        Simpanan_E(2,:)=[];
    elseif s(1,1)==Simpanan_E(3);
        x3=Data_diambil(3,:);
        c=Hasil_urutan(3,:);
        Data_diambil(3,:)=[];
        Hasil_urutan(3,:)=[];
        s(1,:)=[];
        Simpanan_E(3,:)=[];
    elseif s(1,1)==Simpanan_E(4);
        x3=Data_diambil(4,:);
        c=Hasil_urutan(4,:);
        Data_diambil(4,:)=[];
        Hasil_urutan(4,:)=[];
        s(1,:)=[];
        Simpanan_E(4,:)=[];
    end

    if s(1,1)==Simpanan_E(1);
        x4=Data_diambil(1,:);
        d=Hasil_urutan(1,:);
        Data_diambil(1,:)=[];
        Hasil_urutan(1,:)=[];
        s(1,:)=[];
        Simpanan_E(1,:)=[];
    elseif s(1,1)==Simpanan_E(2);
        x4=Data_diambil(2,:);
        d=Hasil_urutan(2,:);
        Data_diambil(2,:)=[];
        Hasil_urutan(2,:)=[];
        s(1,:)=[];
        Simpanan_E(2,:)=[];
    elseif s(1,1)==Simpanan_E(3);
        x4=Data_diambil(3,:);
        d=Hasil_urutan(3,:);
```

```

        Data_diambil(3,:)=[];
        Hasil_urutan(3,:)=[];
        s(1,:)=[];
        Simpanan_E(3,:)=[];
    end

    if s(1,1)==Simpanan_E(1);
        x5=Data_diambil(1,:);
        e=Hasil_urutan(1,:);
        Data_diambil(1,:)=[];
        Hasil_urutan(1,:)=[];
        s(1,:)=[];
        Simpanan_E(1,:)=[];
    elseif s(1,1)==Simpanan_E(2);
        x5=Data_diambil(2,:);
        e=Hasil_urutan(2,:);
        Data_diambil(2,:)=[];
        Hasil_urutan(2,:)=[];
        s(1,:)=[];
        Simpanan_E(2,:)=[];
    end

    if s(1,1)==Simpanan_E(1);
        x6=Data_diambil(1,:);
        f=Hasil_urutan(1,:);
        Data_diambil(1,:)=[];
        Hasil_urutan(1,:)=[];
        s(1,:)=[];
        Simpanan_E(1,:)=[];
    end

    p=[x3; x4; x5; x6]
    q=[c; d; e; f]

```


Lampiran 10.20

Mengambil Data Terkecil untuk 2 Job

```
%Rumus mengambil data terkecil untuk 2 job

function [data, urut]=data_terkecil_2job(s,p,q)

if s(1,1)<s(2,1)
    data=p(1,:);
    urut=q(1,:);
elseif s(2,1)==s(2,1)
    data=[p(1,:); p(2,:)];
    urut=[q(1,:); q(2,:)];
end
```

Lampiran 10.21

Mengambil Data Terkecil untuk 3 Job

```
%Rumus mengambil data terkecil untuk 3 job

function [data2, urut]=data_terkecil_3job(s,p,q)

if s(1,1)<s(2,1)
    data2=p(1,:);
    urut=q(1,:);
elseif s(2,1)<s(3,1)
    data2=[p(1,:); p(2,:)];
    urut=[q(1,:); q(2,:)];
elseif s(2,1)==s(3,1)
    data2=[p(1,:); p(2,:); p(3,:)];
    urut=[q(1,:); q(2,:); q(3,:)];
end
```

Lampiran 10.22

Mengambil Data Terkecil untuk 4 Job

```
%Rumus mengambil data terkecil untuk 4 job

function [data2, urut]=data_terkecil_4job(s,p,q)

if s(1,1)<s(2,1)
    data2=p(1,:);
    urut=q(1,:);
elseif s(2,1)<s(3,1)
    data2=[p(1,:); p(2,:)];
    urut=[q(1,:); q(2,:)];
elseif s(3,1)<s(4,1)
    data2=[p(1,:); p(2,:); p(3,:)];
    urut=[q(1,:); q(2,:); q(3,:)];
elseif s(3,1)==s(4,1)
    data2=[p(1,:); p(2,:); p(3,:); p(4,:)];
    urut=[q(1,:); q(2,:); q(3,:); q(4,:)];
end
```

Lampiran 11

Rekapitulasi Hasil Perhitungan Metode Enumerasi

No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}	F _{max}
1	26	21	26	24	21	21	27	27	21	22	21	26	25	25	25	26	27	27	21	21	22	21	21	22
2	31	31	31	31	33	33	30	30	30	30	30	30	31	31	29	29	29	29	35	35	35	35	35	35
3	33	36	33	33	34	33	35	32	40	36	37	37	31	35	28	31	35	32	36	36	35	40	32	35
4	36	31	36	33	31	31	32	36	31	32	31	36	31	32	33	36	32	36	31	31	32	31	31	32
5	35	35	34	34	37	37	36	36	36	36	36	36	33	33	33	33	33	33	38	38	38	38	38	38
6	32	33	32	29	30	30	29	32	30	30	33	32	29	29	29	32	29	32	30	29	29	30	29	29
7	40	40	38	38	38	38	35	38	36	35	40	40	34	33	36	38	33	38	36	34	33	36	34	33
8	31	31	31	31	31	31	31	31	31	31	31	31	36	36	36	36	36	36	31	31	31	31	36	36
9	27	27	27	27	27	27	31	31	27	27	27	27	32	32	32	32	32	32	28	28	28	28	28	28
10	22	22	23	26	25	26	24	22	23	24	22	22	26	26	26	23	24	23	25	26	25	25	26	26
11	36	35	41	38	33	38	33	36	33	32	33	36	38	35	38	41	38	38	33	38	32	33	38	35
12	41	41	45	43	37	37	34	39	34	34	35	41	37	36	41	45	38	39	33	37	31	33	37	31
13	35	33	33	30	29	28	30	33	29	28	33	35	27	27	28	33	28	33	29	27	27	29	27	27
14	33	37	29	32	37	33	33	33	37	37	37	33	32	33	32	29	33	29	37	33	37	37	33	33
15	28	28	28	28	28	30	31	28	28	32	26	26	35	34	32	32	32	32	32	32	32	31	36	36
16	42	42	39	39	39	39	40	40	40	40	40	40	37	37	37	38	37	37	34	34	34	34	34	34
17	38	38	40	41	41	41	32	36	36	32	36	36	39	35	41	38	34	36	39	41	35	36	39	35
18	35	32	34	34	32	31	38	38	33	35	33	35	37	37	37	37	37	37	32	32	35	32	34	34
19	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
20	41	41	43	43	41	41	36	37	38	36	39	41	38	38	39	39	38	38	38	40	36	38	36	36

21	33	37	33	32	34	34	30	33	34	34	37	35	27	29	30	33	30	33	31	31	31	33	31	31
22	39	35	41	36	34	36	36	40	34	35	34	39	36	35	36	41	38	40	34	36	35	34	36	35
23	34	32	34	35	32	35	40	35	37	38	35	35	41	41	36	36	40	38	38	38	38	38	39	41
24	37	40	38	39	40	42	38	37	40	38	40	37	40	38	39	37	38	37	40	42	38	40	40	40
25	37	34	36	36	34	33	35	35	34	32	34	37	34	34	34	34	34	34	34	33	32	34	31	31
26	30	30	30	30	31	33	26	24	28	26	28	25	31	27	29	28	27	27	33	33	28	28	31	30
27	43	40	43	39	40	39	44	43	40	44	40	43	39	44	39	43	44	43	40	39	44	40	39	44
28	34	34	34	34	34	34	30	30	30	30	30	30	30	30	30	30	30	30	32	32	32	32	32	32
29	32	32	32	32	32	32	34	34	34	34	34	34	35	35	35	35	35	35	33	33	35	35	36	36
30	36	33	38	34	31	31	32	38	29	31	33	36	31	31	36	38	34	38	29	31	31	29	31	31
31	38	38	40	39	38	38	37	41	35	37	36	37	39	39	43	43	40	41	34	36	37	33	39	37
32	38	34	33	32	33	32	29	32	31	29	33	32	32	29	32	33	29	32	33	32	29	31	32	29
33	28	28	29	32	32	32	23	27	24	23	27	27	28	28	32	28	25	27	28	29	28	28	28	28
34	33	32	33	33	32	33	33	33	32	33	32	33	33	33	33	33	33	33	32	33	33	32	33	33
35	36	36	37	37	33	33	38	38	38	38	38	38	39	39	39	39	39	39	35	35	35	35	35	35
36	44	44	44	44	44	44	41	41	43	42	43	43	42	41	42	42	41	41	44	44	43	43	43	43
37	35	30	37	37	30	30	33	33	28	28	28	35	35	35	35	35	35	35	28	30	28	28	28	28
38	24	25	25	25	26	27	20	20	25	20	25	23	23	20	23	21	20	20	26	27	21	25	23	22
39	43	39	43	38	39	38	39	43	39	39	39	43	38	39	38	43	39	43	39	38	39	39	38	39
40	30	33	30	31	33	31	35	31	35	38	33	30	31	36	31	30	35	30	33	31	38	36	34	36
41	38	38	38	38	38	38	35	37	37	37	39	39	34	35	36	36	35	37	36	36	36	37	36	36
42	39	33	39	36	32	36	39	39	33	33	33	39	36	36	36	39	39	39	32	36	33	32	36	36
43	27	28	27	27	28	27	30	27	28	30	28	27	28	30	27	27	30	27	28	27	30	28	28	30
44	31	31	31	31	31	31	28	28	30	29	29	29	30	30	30	30	30	30	32	32	29	30	31	31
45	40	40	40	32	34	32	35	40	35	35	40	40	32	32	32	40	35	40	34	32	32	34	32	32

46	34	30	34	34	30	30	38	38	32	32	32	34	38	38	38	38	38	38	31	31	31	31	31	31
47	38	38	34	34	33	33	32	32	32	32	36	37	28	28	28	32	28	30	30	27	27	29	27	27
48	33	32	35	34	31	34	35	34	34	34	34	34	35	36	34	35	37	37	32	34	33	33	34	36
49	27	31	27	27	31	31	31	27	31	31	31	27	31	31	27	27	31	27	31	31	31	31	31	31
50	38	34	38	33	33	31	41	39	37	38	35	38	36	38	34	38	41	39	33	32	38	34	33	38
51	32	28	30	26	28	26	30	30	28	26	28	32	26	26	26	30	28	30	28	26	26	28	26	26
52	31	32	31	30	32	30	29	31	32	29	32	31	30	29	30	31	29	31	32	30	29	32	30	29
53	43	43	44	45	47	47	39	39	39	39	41	39	43	43	45	42	42	42	45	45	45	45	45	45
54	37	35	37	36	33	33	37	37	35	35	35	37	36	36	36	37	37	37	33	33	33	33	33	33
55	30	30	30	31	30	31	33	31	30	33	30	30	31	33	31	30	33	31	30	31	33	30	31	33
56	40	41	41	44	45	45	35	40	37	36	41	40	39	39	44	40	36	40	40	41	40	40	40	40
57	34	34	34	33	33	33	31	34	32	31	34	34	33	30	33	34	31	34	33	33	30	32	33	30
58	39	39	40	36	36	36	38	38	38	38	38	38	34	33	37	40	39	39	33	34	32	32	34	33
59	27	27	27	36	29	29	29	29	29	29	29	29	31	31	29	27	29	29	31	31	31	31	31	31
60	36	34	36	33	33	33	36	37	34	36	35	36	33	36	33	36	36	37	33	33	36	34	33	36
61	38	34	38	38	38	38	35	37	35	35	36	36	35	35	39	39	36	37	34	34	34	34	34	34
62	29	30	29	28	30	28	29	29	30	29	30	29	28	29	28	29	29	29	30	28	29	30	28	29
63	26	26	25	25	25	25	32	32	32	32	32	32	26	28	26	26	31	31	25	25	29	29	25	28
64	32	32	29	28	32	28	37	37	32	37	32	32	30	33	30	30	34	33	32	28	37	32	30	33
65	36	42	36	36	42	36	42	37	44	44	42	37	36	42	36	36	42	36	42	38	44	44	38	42
66	27	25	25	23	25	23	27	27	25	25	25	27	23	23	23	25	25	25	25	23	25	25	23	23
67	42	38	42	43	39	39	36	42	37	36	38	42	39	37	43	42	36	42	37	39	36	37	39	36
68	29	31	24	23	30	24	31	31	33	33	31	31	24	25	24	24	26	24	30	26	32	32	26	26
69	28	31	28	30	32	32	27	26	31	27	31	27	30	27	30	26	27	26	32	32	27	31	30	27
70	38	36	41	41	36	41	34	38	35	34	35	38	41	34	41	41	34	38	36	41	34	35	41	34

71	31	30	33	36	33	35	33	31	30	32	30	31	38	38	36	35	35	35	35	35	35	35	37	37
72	36	35	36	37	36	37	35	36	35	35	35	36	37	36	37	36	35	36	36	37	35	35	37	36
73	40	43	38	35	39	38	33	38	39	36	43	40	34	33	34	38	33	38	39	34	33	39	34	33
74	35	35	35	30	31	30	37	35	37	37	35	35	32	32	30	35	37	35	32	32	33	32	32	32
75	34	38	34	34	38	34	34	34	38	34	38	34	34	34	34	34	34	34	38	34	34	38	34	34
76	37	37	43	43	41	41	33	35	33	33	34	36	39	39	42	39	39	39	37	40	37	37	37	37
77	38	38	38	38	38	38	35	35	35	35	35	35	34	34	34	34	34	34	38	38	36	36	36	36
78	42	42	42	42	42	42	39	39	39	39	40	40	36	36	36	36	36	36	41	41	40	40	40	40
79	39	39	42	42	39	42	40	40	39	40	39	39	42	41	42	42	41	41	39	42	40	39	42	41
80	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	24	24	24	24	24	24
81	29	29	27	30	32	30	31	31	31	31	31	31	30	32	30	27	29	29	32	30	34	34	30	32
82	30	30	30	34	30	34	32	30	30	32	30	30	35	34	34	30	32	30	30	34	32	30	35	34
83	36	36	36	35	35	35	36	36	36	36	36	36	35	35	35	36	36	36	35	35	35	35	35	35
84	42	42	43	42	40	40	47	47	47	47	47	47	47	47	47	47	48	48	39	39	39	39	39	40
85	36	30	38	35	29	33	40	38	34	34	34	36	39	39	37	38	42	42	31	33	33	31	33	37
86	35	40	34	35	40	36	38	34	42	38	40	37	35	38	35	34	38	34	40	38	38	42	35	38
87	29	29	29	29	29	29	27	27	27	27	27	27	32	30	30	30	30	30	31	31	29	29	32	32
88	33	29	33	30	29	30	38	33	33	34	29	33	35	35	30	33	38	33	30	30	34	30	31	35
89	36	36	38	39	36	39	37	36	36	37	36	36	39	37	39	38	37	36	36	39	37	36	39	37
90	27	27	27	27	27	27	31	31	31	31	31	31	26	26	26	26	29	29	28	28	28	28	28	28
91	33	33	33	33	33	33	26	26	26	26	26	26	28	25	32	32	25	25	29	29	26	26	28	26
92	36	36	36	36	36	36	39	39	37	37	35	35	40	40	40	40	40	40	38	38	38	38	38	38
93	37	39	32	32	39	33	38	38	39	40	39	37	32	37	32	32	37	35	39	33	40	39	34	37
94	39	39	36	36	37	36	35	35	35	35	35	35	31	31	31	32	31	31	37	34	33	33	30	30
95	29	30	29	30	31	30	32	29	33	32	30	29	30	32	30	29	32	29	31	30	32	33	30	32

96	32	33	28	30	38	32	35	35	36	36	35	35	30	33	30	28	28	28	38	35	41	41	35	35
97	28	28	26	26	31	29	27	27	27	27	27	27	25	25	25	25	25	25	31	29	30	30	28	28
98	23	23	23	23	23	23	21	22	20	21	21	21	23	21	24	24	21	22	22	22	21	20	23	21
99	39	36	37	37	33	33	31	32	32	28	35	39	29	29	30	33	29	32	32	30	25	32	26	25
100	32	34	32	36	34	36	36	32	34	37	34	32	36	39	36	32	36	32	34	36	37	35	36	39
101	32	29	34	38	32	32	29	32	25	25	26	32	35	35	38	35	31	32	29	30	29	29	31	29
102	17	17	19	19	17	19	20	20	17	20	17	17	21	21	21	21	21	21	17	19	20	17	21	21
103	41	41	42	41	40	40	44	44	44	44	44	44	44	44	44	44	45	45	37	37	37	37	37	37
104	39	42	39	33	37	36	37	39	40	40	42	39	31	34	33	39	37	39	37	34	35	37	34	34
105	26	26	26	26	26	26	28	28	29	29	28	28	27	27	25	25	27	26	29	29	29	29	29	29
106	40	36	40	41	36	41	40	40	36	40	36	40	41	40	41	40	40	40	36	41	40	36	41	40
107	33	32	34	37	32	37	39	36	32	39	32	33	37	40	37	34	39	37	32	37	39	33	37	40
108	29	29	29	29	29	29	26	27	26	26	27	27	28	27	29	29	27	27	28	28	27	27	28	27
109	35	35	37	37	37	37	30	32	33	30	35	35	32	32	34	32	32	32	33	35	32	33	32	32
110	35	32	35	34	32	32	29	33	32	29	32	35	32	29	32	35	29	33	32	32	29	32	32	29
111	32	34	32	31	32	32	32	32	34	32	34	33	30	32	30	32	32	32	32	32	32	34	30	32
112	37	36	37	41	36	41	41	37	36	41	36	37	41	41	41	37	41	37	36	41	41	36	41	41
113	32	35	32	27	32	30	32	32	35	35	35	32	27	30	27	32	32	32	32	30	32	33	30	30
114	28	26	28	28	26	26	28	28	26	26	26	28	28	28	28	28	28	28	26	26	26	26	26	26
115	31	30	28	28	28	28	34	34	33	33	33	33	31	31	31	31	31	31	28	28	29	28	28	28
116	37	36	36	33	36	33	40	41	37	40	38	38	35	40	36	36	40	41	36	32	40	37	35	40
117	21	21	21	21	21	21	26	22	27	26	23	23	24	26	20	20	26	22	25	25	26	27	24	26
118	32	29	32	32	29	30	37	36	30	37	29	32	37	37	36	36	37	36	30	30	37	30	37	37
119	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	38	38	38	38	38	38
120	38	38	38	38	38	38	36	36	35	35	38	38	36	36	36	36	36	36	32	32	32	32	32	32

121	29	32	29	30	33	33	30	29	33	33	32	29	31	31	30	29	30	29	34	34	34	34	34	34
122	30	36	30	30	36	35	34	30	37	34	36	30	33	34	30	30	34	30	36	36	34	37	33	34
123	37	37	39	40	38	40	38	38	38	38	38	38	40	41	40	39	40	40	38	40	39	39	40	41
124	36	36	36	36	36	37	37	33	36	37	33	32	41	37	37	37	37	33	40	40	37	36	41	37
125	36	35	36	39	35	39	41	37	37	41	34	36	43	41	39	39	41	37	39	39	41	37	43	41
126	37	44	37	40	44	40	37	37	44	37	44	37	40	37	40	37	37	37	44	40	37	44	40	37
127	24	23	24	24	23	23	25	24	24	24	23	24	25	25	24	24	25	24	24	24	24	24	24	24
128	35	33	36	33	32	33	38	39	36	36	36	36	34	36	35	36	39	39	31	33	36	33	33	36
129	37	37	37	37	37	37	33	33	33	33	33	33	37	36	37	37	36	36	37	37	33	33	37	36
130	26	26	26	29	29	29	26	26	26	26	26	26	29	29	29	26	26	26	29	29	29	29	29	29
131	40	40	40	40	40	40	41	41	41	41	41	41	41	41	41	41	41	41	40	40	41	41	41	41
132	34	34	32	32	34	32	30	30	33	30	33	33	31	30	31	31	30	30	34	32	30	33	31	30
133	37	39	36	36	38	38	35	36	38	36	39	37	35	35	35	36	35	36	38	37	35	38	35	35
134	38	38	39	44	42	44	37	38	38	37	38	38	44	38	44	39	37	38	42	44	37	38	44	38
135	33	33	33	33	33	33	32	37	33	32	38	38	27	32	32	32	32	37	28	28	32	33	27	32
136	27	31	27	27	31	30	30	29	34	34	31	29	29	30	27	27	30	27	33	33	34	34	33	33
137	29	27	29	32	28	31	30	29	27	30	27	29	33	33	32	30	30	29	29	31	30	29	33	32
138	32	37	32	32	37	32	39	34	40	39	37	35	33	39	32	32	39	34	37	34	39	40	33	39
139	42	42	43	43	42	42	41	41	41	41	41	41	46	46	46	46	46	46	39	39	39	39	39	39
140	34	34	32	32	36	35	33	33	33	33	33	33	31	30	31	30	30	30	36	34	35	35	34	34
141	35	38	35	35	37	37	35	34	39	38	38	36	34	35	33	34	35	34	38	38	37	39	37	37
142	23	26	25	27	26	30	25	23	26	28	26	23	29	29	27	27	27	27	28	30	28	28	32	32
143	21	21	21	21	21	21	26	23	26	26	23	23	24	24	21	21	26	23	24	24	24	24	24	24
144	28	28	28	28	28	28	23	25	24	24	28	28	23	23	25	25	23	25	24	24	24	24	24	24
145	32	33	32	29	30	30	29	32	30	30	33	32	29	29	29	32	29	32	30	29	29	30	29	29

146	40	40	38	38	38	38	35	38	36	35	40	40	34	33	36	38	33	38	36	34	33	36	34	33
147	31	31	31	31	31	31	31	31	31	31	31	31	36	36	36	36	36	36	31	31	31	31	36	36
148	27	27	27	27	27	27	31	31	27	27	27	27	32	32	32	32	32	32	28	28	28	28	28	28
149	22	22	23	26	25	26	24	22	23	24	22	22	26	26	26	23	24	23	25	26	25	25	26	26
150	36	35	41	38	33	38	33	36	33	32	33	36	38	35	38	41	38	38	33	38	32	33	38	35
151	41	41	45	43	37	37	34	39	34	34	35	41	37	36	41	45	38	39	33	37	31	33	37	31
152	35	33	33	30	29	28	30	33	29	28	33	35	27	27	28	33	28	33	29	27	27	29	27	27
153	33	37	29	32	37	33	33	33	37	37	37	33	32	33	32	29	33	29	37	33	37	37	33	33
154	28	28	28	28	28	30	31	28	28	32	26	26	35	34	32	32	32	32	32	32	32	31	36	36
155	42	42	39	39	39	39	40	40	40	40	40	40	37	37	37	38	37	37	34	34	34	34	34	34
156	38	38	40	41	41	41	32	36	36	32	36	36	39	35	41	38	34	36	39	41	35	36	39	35
157	35	32	34	34	32	31	38	38	33	35	33	35	37	37	37	37	37	37	32	32	35	32	34	34
158	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
159	41	41	43	43	41	41	36	37	38	36	39	41	38	38	39	39	38	38	38	40	36	38	36	36
160	33	37	33	32	34	34	30	33	34	34	37	35	27	29	30	33	30	33	31	31	31	33	31	31
161	39	35	41	36	34	36	36	40	34	35	34	39	36	35	36	41	38	40	34	36	35	34	36	35
162	34	32	34	35	32	35	40	35	37	38	35	35	41	41	36	36	40	38	38	38	38	38	39	41
163	37	40	38	39	40	42	38	37	40	38	40	37	40	38	39	37	38	37	40	42	38	40	40	40
164	37	34	36	36	34	33	35	35	34	32	34	37	34	34	34	34	34	34	34	33	32	34	31	31
165	30	30	30	30	31	33	26	24	28	26	28	25	31	27	29	28	27	27	33	33	28	28	31	30
166	43	40	43	39	40	39	44	43	40	44	40	43	39	44	39	43	44	43	40	39	44	40	39	44
167	34	34	34	34	34	34	30	30	30	30	30	30	30	30	30	30	30	30	32	32	32	32	32	32
168	32	32	32	32	32	32	34	34	34	34	34	34	35	35	35	35	35	35	33	33	35	35	36	36
169	36	33	38	34	31	31	32	38	29	31	33	36	31	31	36	38	34	38	29	31	31	29	31	31
170	38	38	40	39	38	38	37	41	35	37	36	37	39	39	43	43	40	41	34	36	37	33	39	37

171	34	34	33	32	33	32	29	32	31	29	33	32	32	29	32	33	29	32	33	32	29	31	32	29
172	28	28	29	32	32	32	23	27	24	23	27	27	28	28	32	28	25	27	28	29	28	28	28	28
173	33	32	33	33	32	33	33	33	32	33	32	33	33	33	33	33	33	33	32	33	33	32	33	33
174	29	28	29	27	28	26	29	29	28	28	28	29	27	27	27	29	29	29	28	26	28	28	26	27
175	36	36	37	37	33	33	38	38	38	38	38	38	39	39	39	39	39	39	35	35	35	35	35	35
176	37	36	37	34	36	34	40	37	39	40	36	37	34	40	34	37	40	37	36	34	40	39	34	40
177	31	32	31	32	32	32	29	31	32	29	32	31	32	29	32	31	29	31	32	32	29	32	32	29
178	46	43	45	45	43	42	39	44	43	36	43	46	41	38	43	44	38	44	43	42	36	43	41	36
179	30	33	30	31	33	31	35	31	35	38	33	30	31	36	31	30	35	30	33	31	38	36	34	36
180	38	38	38	38	38	38	35	37	37	37	39	39	34	35	36	36	35	37	36	36	36	37	36	36
181	39	33	39	36	32	36	39	39	33	33	33	39	36	36	36	39	39	39	32	36	33	32	36	36
182	27	28	27	27	28	27	30	27	28	30	28	27	28	30	27	27	30	27	28	27	30	28	28	30
183	31	31	31	31	31	31	28	28	30	29	29	29	30	30	30	30	30	30	32	32	29	30	31	31
184	40	40	40	32	34	32	35	40	35	35	40	40	32	32	32	40	35	40	34	32	32	34	32	32
185	34	30	34	34	30	30	38	38	32	32	32	34	38	38	38	38	38	38	31	31	31	31	31	31
186	38	38	34	34	33	33	32	32	32	32	36	37	28	28	28	32	28	30	30	27	27	29	27	27
187	33	32	35	34	31	34	35	34	34	34	34	34	35	36	34	35	37	37	32	34	33	33	34	36
188	27	31	27	27	31	31	31	27	31	31	31	27	31	31	27	27	31	27	31	31	31	31	31	31
189	38	34	38	33	33	31	41	39	37	38	35	38	36	38	34	38	41	39	33	32	38	34	33	38
190	32	28	30	26	28	26	30	30	28	26	28	32	26	26	26	30	28	30	28	26	26	28	26	26
191	31	32	31	30	32	30	29	31	32	29	32	31	30	29	30	31	29	31	32	30	29	32	30	29
192	43	43	44	45	47	47	39	39	39	39	41	39	43	43	45	42	42	42	45	45	45	45	45	45
193	37	35	37	36	33	33	37	37	35	35	35	37	36	36	36	37	37	37	33	33	33	33	33	33
194	30	30	30	31	30	31	33	31	30	33	30	30	31	33	31	30	33	31	30	31	33	30	31	33
195	40	41	41	44	45	45	35	40	37	36	41	40	39	39	44	40	36	40	40	41	40	40	40	40

196	34	34	34	33	33	33	31	34	32	31	34	34	33	30	33	34	31	34	33	33	30	32	33	30
197	39	39	40	36	36	36	38	38	38	38	38	38	34	33	37	40	39	39	33	34	32	32	34	33
198	27	27	27	29	29	29	29	29	29	29	29	29	31	31	29	27	29	29	31	31	31	31	31	31
199	36	34	36	33	33	33	36	37	34	36	35	36	33	36	33	36	36	37	33	33	36	34	33	36
200	38	38	38	38	38	38	35	37	35	35	36	36	35	35	39	39	36	37	34	34	34	34	34	34
201	29	30	29	28	30	28	29	29	30	29	30	29	28	29	28	29	29	29	30	28	29	30	28	29
202	26	26	25	25	25	25	32	32	32	32	32	32	26	28	26	26	31	31	25	25	29	29	25	28
203	32	32	29	28	32	28	37	37	32	37	32	32	30	33	30	30	34	33	32	28	37	32	30	33
204	36	42	36	36	42	36	42	37	44	44	42	37	36	42	36	36	42	36	42	38	44	44	38	42
205	33	35	32	34	35	34	33	31	35	33	35	33	34	33	34	32	33	31	35	34	33	35	34	33
206	27	25	25	23	25	23	27	27	25	25	25	27	23	23	23	25	25	25	25	23	25	25	23	23
207	29	31	24	23	30	24	31	31	33	33	31	31	24	25	24	24	26	24	30	26	32	32	26	26
208	28	31	28	30	32	32	27	26	31	27	31	27	30	27	30	26	27	26	32	32	27	31	30	27
209	38	36	41	41	36	41	34	38	35	34	35	38	41	34	41	41	34	38	36	41	34	35	41	34
210	36	35	36	37	36	37	35	36	35	35	35	36	37	36	37	36	35	36	36	37	35	35	37	36
211	31	31	31	34	33	34	32	32	32	32	32	32	35	36	34	32	33	33	34	34	35	35	35	36
212	40	43	38	35	39	38	33	38	39	36	43	40	34	33	34	38	33	38	39	34	33	39	34	33
213	35	35	35	30	31	30	37	35	37	37	35	35	32	32	30	35	37	35	32	32	33	32	32	32
214	34	38	34	34	38	34	34	34	38	34	38	34	34	34	34	34	34	34	38	34	34	38	34	34
215	37	37	43	43	41	41	33	35	33	33	34	36	39	39	42	39	39	39	37	40	37	37	37	37
216	38	38	38	38	38	38	35	35	35	35	35	35	34	34	34	34	34	34	38	38	36	36	36	36
217	42	42	42	42	42	42	39	39	39	39	40	40	36	36	36	36	36	36	41	41	40	40	40	40
218	39	39	42	42	39	42	40	40	39	40	39	39	42	41	42	42	41	41	39	42	40	39	42	41
219	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	24	24	24	24	24	24
220	29	29	27	30	32	30	31	31	31	31	31	31	30	32	30	27	29	29	32	30	34	34	30	32

221	30	30	30	34	30	34	32	30	30	32	30	30	35	34	34	30	32	30	30	34	32	30	35	34
222	36	36	36	35	35	35	36	36	36	36	36	36	35	35	35	36	36	36	35	35	35	35	35	35
223	42	42	43	42	40	40	47	47	47	47	47	47	47	47	47	47	48	48	39	39	39	39	39	40
224	36	30	38	35	29	33	40	38	34	34	34	36	39	39	37	38	42	42	31	33	33	31	33	37
225	35	40	34	35	40	36	38	34	42	38	40	37	35	38	35	34	38	34	40	38	38	42	35	38
226	29	29	29	29	29	29	27	27	27	27	27	27	32	30	30	30	30	30	31	31	29	29	32	32
227	33	29	33	30	29	30	38	33	33	34	29	33	35	35	30	33	38	33	30	30	34	30	31	35
228	36	36	38	39	36	39	37	36	36	37	36	36	39	37	39	38	37	36	36	39	37	36	39	37
229	27	27	27	27	27	27	31	31	31	31	31	31	26	26	26	26	29	29	28	28	28	28	28	28
230	33	33	33	33	33	33	26	26	26	26	26	26	28	25	32	32	25	25	29	29	26	26	28	26
231	36	36	36	36	36	36	39	39	37	37	35	35	40	40	40	40	40	40	38	38	38	38	38	38
232	37	39	32	32	39	33	38	38	39	40	39	37	32	37	32	32	37	35	39	33	40	39	34	37
233	39	39	36	36	37	36	35	35	35	35	35	35	31	31	31	32	31	31	37	34	33	33	30	30
234	29	30	29	30	31	30	32	29	33	32	30	29	30	32	30	29	32	29	31	30	32	33	30	32
235	32	33	28	30	38	32	35	35	36	36	35	35	30	33	30	28	28	28	38	35	41	41	35	35
236	28	28	26	26	31	29	27	27	27	27	27	27	25	25	25	25	25	25	31	29	30	30	28	28
237	23	23	23	23	23	23	21	22	20	21	21	21	23	21	24	24	21	22	22	22	21	20	23	21
238	39	36	37	37	33	33	31	32	32	28	35	39	29	29	30	33	29	32	32	30	25	32	26	25
239	37	37	37	37	37	37	35	35	35	35	35	35	38	36	36	36	34	34	39	39	37	37	38	36
240	32	34	32	36	34	36	36	32	34	37	34	32	36	39	36	32	36	32	34	36	37	35	36	39
241	35	30	37	37	30	30	33	33	28	28	28	35	35	35	35	35	35	35	28	30	28	28	28	28
242	24	25	25	25	26	27	20	20	25	20	25	23	23	20	23	21	20	20	26	27	21	25	23	22
243	43	39	43	38	39	38	39	43	39	39	39	43	38	39	38	43	39	43	39	38	39	39	38	39
244	44	44	44	44	44	44	41	41	43	42	43	43	42	41	42	42	41	41	44	44	43	43	43	43
245	42	38	42	43	39	39	36	42	37	36	38	42	39	37	43	42	36	42	37	39	36	37	39	36

246	32	29	34	38	32	32	29	32	25	25	26	32	35	35	38	35	31	32	29	30	29	29	31	29
247	17	17	19	19	17	19	20	20	17	20	17	17	21	21	21	21	21	21	17	19	20	17	21	21
248	41	41	42	41	40	40	44	44	44	44	44	44	44	44	44	44	45	45	37	37	37	37	37	37
249	39	42	39	33	37	36	37	39	40	40	42	39	31	34	33	39	37	39	37	34	35	37	34	34
250	26	26	26	26	26	26	28	28	29	29	28	28	27	27	25	25	27	26	29	29	29	29	29	29
251	40	36	40	41	36	41	40	40	36	40	36	40	41	40	41	40	40	40	36	41	40	36	41	40
252	33	32	34	37	32	37	39	36	32	39	32	33	37	40	37	34	39	37	32	37	39	33	37	40
253	29	29	29	29	29	29	26	27	26	26	27	27	28	27	29	29	27	27	28	28	27	27	28	27
254	35	35	37	37	37	37	30	32	33	30	35	35	32	32	34	32	32	32	33	35	32	33	32	32
255	35	32	35	34	32	32	29	33	32	29	32	35	32	29	32	35	29	33	32	32	29	32	32	29
256	32	34	32	31	32	32	32	32	34	32	34	33	30	32	30	32	32	32	32	32	32	34	30	32
257	37	36	37	41	36	41	41	37	36	41	36	37	41	41	41	37	41	37	36	41	41	36	41	41
258	32	35	32	27	32	30	32	32	35	35	35	32	27	30	27	32	32	32	32	30	32	33	30	30
259	28	26	28	28	26	26	28	28	26	26	26	28	28	28	28	28	28	28	26	26	26	26	26	26
260	31	30	28	28	28	28	34	34	33	33	33	33	31	31	31	31	31	31	28	28	29	28	28	28
261	37	36	36	33	36	33	40	41	37	40	38	38	35	40	36	36	40	41	36	32	40	37	35	40
262	21	21	21	21	21	21	26	22	27	26	23	23	24	26	20	20	26	22	25	25	26	27	24	26
263	32	29	32	32	29	30	37	36	30	37	29	32	37	37	36	36	37	36	30	30	37	30	37	37
264	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	38	38	38	38	38	38
265	38	38	38	38	38	38	36	36	35	35	38	38	36	36	36	36	36	36	32	32	32	32	32	32
266	29	32	29	30	33	33	30	29	33	33	32	29	31	31	30	29	30	29	34	34	34	34	34	34
267	30	36	30	30	36	35	34	30	37	34	36	30	33	34	30	30	34	30	36	36	34	37	33	34
268	37	37	39	40	38	40	38	38	38	38	38	38	40	41	40	39	40	40	38	40	39	39	40	41
269	32	35	33	33	35	37	32	30	35	32	35	31	34	32	33	30	32	30	36	37	32	35	34	32
270	35	33	35	33	31	31	39	35	37	37	33	35	37	37	33	35	39	35	35	35	35	35	35	35

271	27	27	31	31	27	29	25	27	25	25	25	27	29	29	31	31	29	29	25	29	25	25	29	27
272	42	37	42	39	37	39	44	44	39	44	39	42	40	44	40	42	44	44	37	39	44	39	40	44
273	36	36	35	35	39	39	28	28	31	28	35	31	28	27	31	28	27	27	36	35	31	31	31	31
274	30	29	30	30	29	30	37	34	33	36	31	31	34	37	31	31	37	35	30	30	36	33	33	37
275	39	39	40	41	39	39	34	36	34	34	34	34	41	41	43	43	40	40	37	38	37	37	41	38
276	35	35	35	35	38	35	36	36	41	36	36	36	33	34	31	30	34	30	40	40	41	41	35	35
277	32	31	32	31	30	31	28	32	27	28	31	32	31	28	31	32	28	32	30	31	28	27	31	28
278	35	35	35	35	35	35	33	33	33	33	33	33	32	32	32	32	32	32	32	32	32	32	32	32
279	35	39	35	35	39	35	41	36	45	41	40	40	36	41	34	35	41	36	40	40	41	45	36	41
280	35	33	33	33	33	31	35	36	33	34	33	35	33	34	34	34	34	36	33	31	34	33	32	34
281	39	40	40	40	41	41	36	38	39	37	40	39	37	37	39	38	37	38	39	40	38	39	38	38
282	33	36	34	41	37	42	35	33	36	35	36	33	41	37	41	33	35	33	37	42	35	36	41	38
283	30	30	31	30	29	30	29	29	29	29	29	29	30	29	30	31	30	30	29	30	28	28	30	29
284	34	34	35	35	29	30	34	34	34	34	34	34	35	35	35	35	35	35	29	30	29	29	30	30
285	42	40	44	42	38	42	41	42	39	39	39	42	42	41	42	44	43	43	38	42	39	37	42	41
286	33	35	30	32	35	32	32	32	35	34	35	33	32	31	32	30	29	30	35	32	34	35	32	31
287	36	38	36	36	38	41	36	33	38	38	38	33	39	36	36	36	36	36	41	41	38	38	41	41
288	25	25	23	27	29	27	25	25	25	25	25	25	27	27	27	23	23	23	29	27	29	29	27	27
289	38	35	35	35	35	35	42	42	38	38	41	41	36	36	36	36	36	39	33	31	37	36	31	34
290	32	32	33	33	32	32	28	31	26	28	27	28	33	30	36	36	30	31	29	30	28	26	33	28
291	29	26	28	24	25	22	31	31	28	28	26	29	26	26	26	28	30	28	25	23	27	25	23	24
292	42	42	42	42	42	42	40	40	40	40	40	40	38	38	38	38	38	38	41	41	41	41	41	41
293	32	32	39	37	32	37	34	36	29	34	29	31	38	35	40	40	37	37	30	37	34	29	38	35
294	40	40	39	39	42	42	38	38	38	38	39	38	36	34	36	34	33	33	42	41	40	40	40	40
295	27	29	27	30	32	32	28	26	30	28	29	26	31	28	30	26	28	26	33	33	28	30	31	28

296	33	31	33	33	31	33	34	33	31	34	31	33	34	34	33	33	34	33	31	33	34	31	34	34
297	35	35	35	35	35	35	33	38	30	32	36	36	32	32	37	37	32	38	29	29	32	30	31	32
298	41	40	41	35	35	35	39	41	38	39	40	41	35	39	35	41	39	41	35	35	39	38	35	39
299	34	34	35	33	33	33	32	34	31	32	33	33	33	32	34	35	32	34	32	33	32	31	33	32
300	24	24	29	29	24	27	21	23	21	21	21	23	27	26	29	29	26	26	22	27	21	21	27	24

Keterangan :

	Urutan			Urutan
Nomor 1	1-2-3-4		Nomor 13	3-4-1-2
Nomor 2	1-2-4-3		Nomor 14	3-4-2-1
Nomor 3	1-3-2-4		Nomor 15	3-1-4-2
Nomor 4	1-3-4-2		Nomor 16	3-1-2-4
Nomor 5	1-4-2-3		Nomor 17	3-2-4-1
Nomor 6	1-4-3-2		Nomor 18	3-2-1-4
Nomor 7	2-3-4-1		Nomor 19	4-1-2-3
Nomor 8	2-3-1-4		Nomor 20	4-1-3-2
Nomor 9	2-4-1-3		Nomor 21	4-2-3-1
Nomor 10	2-4-3-1		Nomor 22	4-2-1-3
Nomor 11	2-1-4-3		Nomor 23	4-3-1-2
Nomor 12	2-1-3-4		Nomor 24	4-3-2-1