

**ANALISIS ALGORITMA PADA PENJADWALAN PROYEK  
PEMBANGUNAN DENGAN METODE CPM MODEL AOA**

**SKRIPSI**

oleh:  
**NURJIANAH**  
**NIM. 07610033**



**JURUSAN MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2011**

**ANALISIS ALGORITMA PADA PENJADWALAN PROYEK  
PEMBANGUNAN DENGAN METODE CPM MODEL AOA**

**SKRIPSI**

**Diajukan Kepada:**

**Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Sains (S.Si)**

**oleh:**

**NURJIANAH  
NIM. 07610033**

**JURUSAN MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2011**

**SURAT PERNYATAAN  
KEASLIAN TULISAN**

Saya yang bertanda tangan dibawah ini:

Nama : Nurjiana  
NIM : 07610033  
Jurusan : Matematika  
Fakultas : Sains dan Teknologi  
Judul : Analisis Algoritma pada Penjadwalan Proyek Pembangunan  
dengan Metode CPM Model AOA

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Maret 2011

Yang membuat pernyataan,

Nurjiana  
NIM. 07610033

**ANALISIS ALGORITMA PADA PENJADWALAN PROYEK  
PEMBANGUNAN DENGAN METODE CPM MODEL AOA**

**SKRIPSI**

oleh:  
**NURJIANAH**  
**NIM. 07610033**

Telah diperiksa dan disetujui untuk diuji :

Dosen Pembimbing I,

Dosen Pembimbing II,

Mohammad Jamhuri, M.Si  
NIP. 19810502 200501 1 004

Abdussakir, M.Pd  
NIP. 19751006 200312 1 001

Tanggal, 26 Maret 2011

Mengetahui,  
Ketua Jurusan Matematika

Abdussakir, M.Pd  
NIP. 19751006 200312 1 001

**ANALISIS ALGORITMA PADA PENJADWALAN PROYEK  
PEMBANGUNAN DENGAN METODE CPM MODEL AOA**

**SKRIPSI**

oleh:  
**Nurjianah**  
**NIM. 07610033**

Telah dipertahankan di Depan Dewan Penguji Skripsi dan  
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk  
Memperoleh Gelar Sarjana Sains (S.Si)

**Tanggal, 1 April 2011**

**Susunan Dewan Penguji**

**Tanda Tangan**

1. Penguji Utama : Sri Harini, M.Si  
NIP. 19731014 200112 2 002
2. Ketua : Abdul Aziz, M.Si  
NIP. 19760318 200604 1 002
3. Sekretaris : Mohammad Jamhuri, M.Si  
NIP. 19810502 200501 1 004
4. Anggota : Abdussakir, M.Pd  
NIP. 19751006 200312 1 001

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Mengetahui dan Mengesahkan**  
**Ketua Jurusan Matematika**

**Abdussakir, M.Pd**  
**NIP. 19751006 200312 1 001**

## MOTTO:

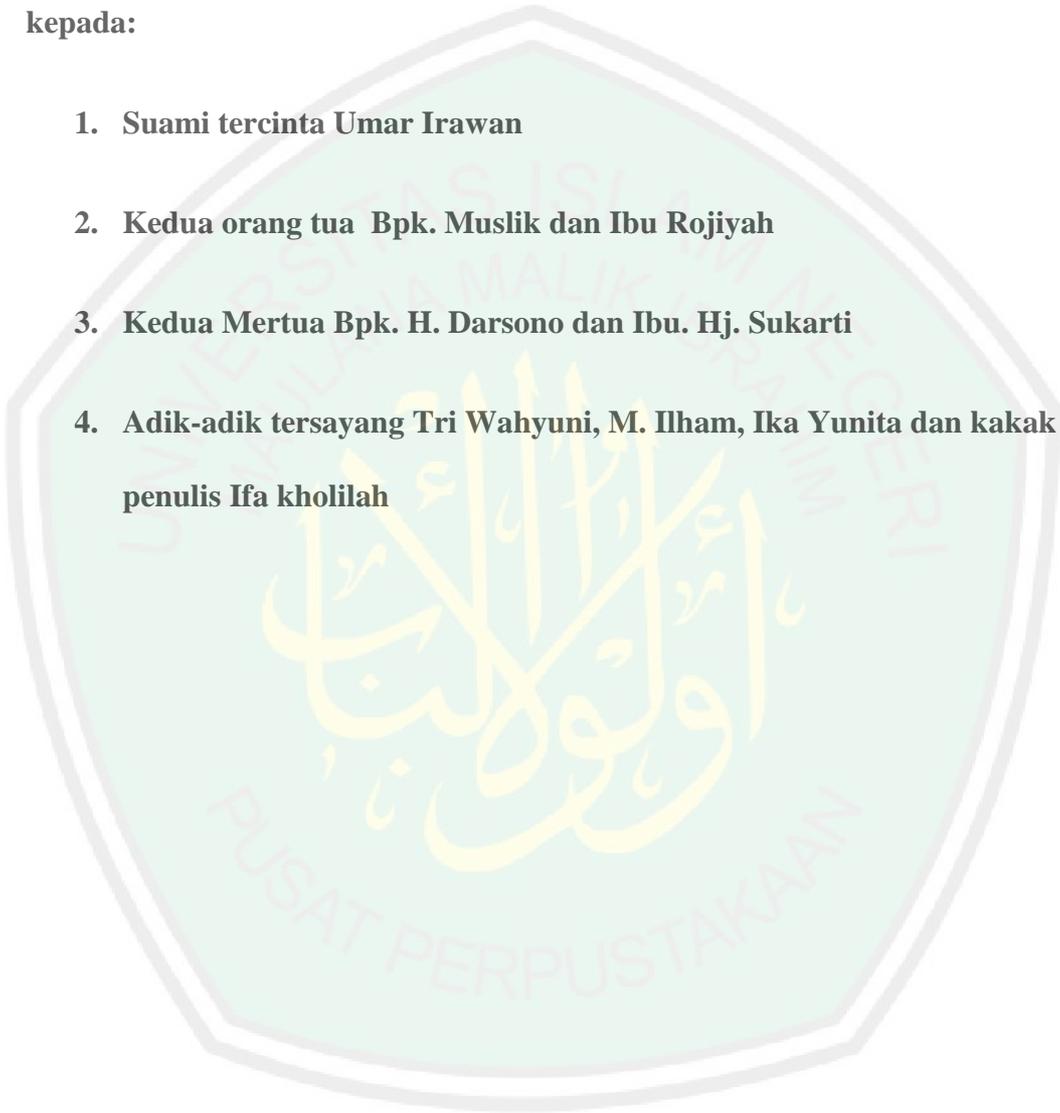
Sebaik-baik manusia adalah yang paling bermanfaat  
bagi sesamanya jadi pergunakanlah waktu dengan  
sebaik-baiknya



## **PERSEMBAHAN**

**Dengan segenap rasa syukur Alhamdulillah, karya ini dipersembahkan kepada:**

- 1. Suami tercinta Umar Irawan**
- 2. Kedua orang tua Bpk. Muslik dan Ibu Rojiyah**
- 3. Kedua Mertua Bpk. H. Darsono dan Ibu. Hj. Sukarti**
- 4. Adik-adik tersayang Tri Wahyuni, M. Ilham, Ika Yunita dan kakak penulis Ifa kholilah**



## KATA PENGANTAR



Alhamdulillah segala puji dan syukur hanya ditujukan kepada Allah SWT yang telah melimpahkan nikmat terbaik berupa iman dan Islam, juga yang selalu melimpahkan rahmat, taufik, hidayah serta inayah-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Analisis Algoritma pada Penjadwalan Proyek Pembangunan dengan Metode CPM Model AOA” sebagai salah satu syarat dalam menyelesaikan pendidikan S1 dan memperoleh gelar Sarjana Sains (S.Si).

Sholawat serta salam semoga tetap tercurahkan kepada kekasih hati baginda Rasulullah Muhammad SAW, yang telah menunjukkan jalan kebenaran dan keselamatan, yakni ajaran Islam yang menjadi rahmat bagi seluruh umat manusia dan sekalian alam.

Selama penulisan skripsi ini penulis telah banyak mendapat bimbingan, masukan, motivasi dan arahan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Prof. Dr. H. Imam Suprayogo selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Prof. Drs. Sutiman B. Sumitro, SU, DSc selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang .

3. Abdussakir, M.Pd selaku ketua Jurusan Matematika Fakultas Saintek Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang dan selaku Dosen Pembimbing Integrasi Sains Matematika dan Islam yang telah banyak memberi arahan kepada penulis .
4. M. Jamhuri, M.Si sebagai dosen pembimbing Matematika yang telah banyak memberikan tuntunan dan arahan sehingga penulisan skripsi ini dapat terselesaikan.
5. Segenap dosen Jurusan Matematika Fakultas Sains dan Teknologi yang telah banyak membantu dalam penyelesaian skripsi ini.
6. Suami tercinta Umar Irawan yang selalu memberikan semangat, sehingga penulisan skripsi ini dapat terselesaikan.
7. Kedua orang tua penulis Ayahanda Muslik dan Bunda Rojiah yang dengan restunya, doanya, harapan-harapan serta pengorbanannya menjadikan penulis untuk tidak menyerah pada keadaan bagaimanapun, termasuk dalam penyelesaian skripsi ini.
8. Keluarga besar penulis yang dengan doa serta dukungannya menjadikan penulis semakin bersemangat dalam penulisan skripsi ini.
9. Teman-teman terbaik penulis Puspita, Any Tsalasatul, Reni, Fitrotin, Zuhairini.
10. Teman-teman jurusan matematika 2007 yang telah banyak membantu dalam penyelesaian penulisan skripsi ini.
11. Semua pihak yang terlibat baik secara langsung maupun tidak langsung pada proses terselesaikannya penulisan skripsi ini.

Semoga Allah SWT membalas kebaikan semuanya. Amin.

Harapan penulis semoga skripsi ini dapat bermanfaat bagi penulis khususnya dan bagi pembaca pada umumnya. Amin.

Malang, 26 Maret 2011

Penulis



## DAFTAR ISI

<b>HALAMAN SAMPUL</b>	
<b>HALAMAN PENGAJUAN</b>	
<b>HALAMAN PERNYATAAN KEASLIAN TULISAN</b>	
<b>HALAMAN PERSETUJUAN</b>	
<b>HALAMAN PENGESAHAN</b>	
<b>HALAMAN MOTTO</b>	
<b>HALAMAN PERSEMBAHAN</b>	
<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>viii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>BAB I : PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah .....	6
1.4. Tujuan Penelitian .....	6
1.5. Manfaat Penelitian .....	6
1.6. Sistematika Penulisan .....	7
<b>BAB II :KAJIAN PUSTAKA .....</b>	<b>8</b>
2.1. Penjadwalan Proyek dengan CPM ( <i>Critical Path Method</i> ).....	8
2.2. Simbol-Simbol yang Digunakan.....	11
2.3. Penentuan Waktu .....	14
2.4. Asumsi dan Cara Perhitungan.....	16
2.4.1. Perhitungan Maju ( <i>Forward Computation</i> ) .....	17
2.4.2. Perhitungan Mundur ( <i>Backward Computation</i> ).....	18
2.4.3. Perhitungan Kelonggaran Waktu ( <i>Float or Slack</i> ).....	19
2.5. Lintasan Kritis ( <i>Critical Path</i> ).....	20
2.6. Algoritma Pemrograman .....	21

2.6.1. Tipe Data Record .....	22
2.6.2. If (Pemilihan) .....	23
2.6.3. Loop (Perulangan) .....	23
2.6.4. Function (Fungsi) .....	24
2.6.5. Array .....	25
2.6.6. Menampilkan Array dengan MSFlexGrid .....	25
2.6.7. Fungsi-Fungsi Grafik .....	30
2.7. Bagan Alir ( <i>Flowchart</i> ) .....	31
2.8. Kajian Agama untuk Penjadwalan Proyek.....	35
<b>BAB III : METODE PENELITIAN.....</b>	<b>39</b>
3.1. Jenis Data.....	39
3.2. Sumber Data .....	39
3.3. Teknis Analisis Data.....	39
3.4. Pemrograman .....	40
<b>BAB IV : PEMBAHASAN .....</b>	<b>42</b>
4.1. Menentukan Lintasan Kritis.....	42
4.1.1. Perhitungan Maju .....	42
4.1.2. Perhitungan Mundur .....	53
4.1.3. Perhitungan Kelonggaran Waktu ( <i>Float</i> ) .....	64
4.2. Menentukan Waktu Optimal Penjadwalan Proyek.....	77
4.3. Penyusunan Diagram Jaringan Kerja.....	78
4.4. Simulasi Program.....	83
4.5. Interpretasi Program .....	87
<b>BAB V : PENUTUP .....</b>	<b>89</b>
5.1. Kesimpulan.....	89
5.2. Saran.....	90

**DAFTAR PUSTAKA**

**LAMPIRAN-LAMPIRAN**

## DAFTAR GAMBAR

Gambar 2.1 Pemakaian <i>Dummy</i> .....	12
Gambar 2.2 Kegiatan A Harus Diselesaikan Dahulu Sebelum Kegiatan B Dapat Dimulai.....	13
Gambar 2.3 Kegiatan C, D, dan E Selesai Sebelum Kegiatan F Dapat Dimulai..	13
Gambar 2.4 Kegiatan Kegiatan G dan H Selesai Sebelum Kegiatan I dan J.....	13
Gambar 2.5 Kegiatan K dan L Harus Selesai Sebelum Kegiatan M Dapat Dimulai Tetapi Kegiatan N Sudah Boleh Dimulai Bila Kegiatan L Sudah Selesai.....	14
Gambar 2.6 Kegiatan P, Q, dan R Mulai dan Selesai pada <i>event</i> yang Sama.....	14
Gambar 2.7 Lingkaran Kejadian ( <i>event</i> ).....	17
Gambar 2.8 Saat Paling Cepat untuk Menyelesaikan Kegiatan.....	17
Gambar 2.9 <i>Merge Event</i> .....	18
Gambar 2.10 Saat Paling Lambat untuk Memulai Kegiatan.....	18
Gambar 2.11 <i>Burst Event</i> .....	19
Gambar 2.12 Beberapa Simbol dalam Bagan Alir.....	32
Gambar 2.13 Bagan Alir Menghitung Luas Persegi Panjang .....	34
Gambar 4.1 Bagan Alir Perhitungan Maju .....	45
Gambar 4.2 Bagan Alir Saat Tercepat Dimulainya Kegiatan ES .....	48
Gambar 4.3 Bagan Alir Saat Tercepat Diselesaikannya Kegiatan EF.....	52
Gambar 4.4 Bagan Alir Perhitungan Mundur.....	56
Gambar 4.5 Bagan Alir Saat Paling Lambat Diselesaikannya Kegiatan LF.....	59
Gambar 4.6 Bagan Alir Saat Paling Lambat Dimulainya Kegiatan LS.....	63
Gambar 4.7 Bagan Alir Total Float TF.....	67
Gambar 4.8 Bagan Alir Free Float FF.....	68
Gambar 4.9 Bagan Alir Lintasan.....	70
Gambar 4.10 Bagan Alir Waktu Maksimal.....	72
Gambar 4.11 Bagan Alir Waktu Minimal.....	73
Gambar 4.12 Bagan Alir Fungsi Find.....	74

Gambar 4.13 Bagan Alir Fungsi Find1.....	75
Gambar 4.14 Bagan Alir Fungsi Find2n.....	76
Gambar 4.15 Bagan Alir untuk Menentukan Koordinat Gambar.....	80
Gambar 4.16 Bagan Alir Koordinat Maksimal.....	82
Gambar 4.17 Tampilan Awal Program.....	83
Gambar 4.18 Notepad untuk Menulis Data.....	84
Gambar 4.19 Tampilan untuk Input Program dari Notepad.....	84
Gambar 4.20 Tampilan Hasil Perhitungan.....	85
Gambar 4.21 Potongan Gambar ke-1 Diagram Jaringan Kerja.....	86
Gambar 4.22 Potongan Gambar ke-2 Diagram Jaringan Kerja.....	86
Gambar 4.23 Potongan Gambar ke-3 Diagram Jaringan Kerja.....	87



## ABSTRAK

Nurjiana. 2011. *Analisis Algoritma pada Penjadwalan Proyek Pembangunan dengan Metode CPM Model AOA*. Skripsi, Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

**Pembimbing :** (1) M. Jamhuri, M.Si  
(2) Abdussakir, M.Pd

**Kata kunci :** *Network, Critical Path Method, Activity On Arrow, Penjadwalan*

Riset operasi diartikan sebagai peralatan manajemen yang menyatukan ilmu pengetahuan, matematika dan logika dalam rangka memecahkan masalah-masalah yang dihadapi sehari-hari, sehingga akhirnya permasalahan tersebut dapat dipecahkan secara optimal. Riset operasi sangat membantu menyelesaikan penjadwalan pembangunan proyek yang sangat diperlukan untuk mengoptimalkan waktu.

Salah satu teknik yang populer dalam penjadwalan adalah perencanaan jaringan kerja (*network planning*). Perencanaan jaringan kerja adalah salah satu model yang banyak digunakan dalam penyelenggaraan proyek, yang dapat memberikan informasi tentang kegiatan-kegiatan yang ada dalam diagram jaringan itu. Metode CPM (*Critical Path Method*) model AOA (*activity on arrow*) adalah salah satu teknik dalam perencanaan jaringan kerja. Metode ini menganalisis waktu dan biaya dalam menyelesaikan suatu pekerjaan dalam pembangunan proyek. Karena dalam analisisnya selalu ditemukan perhitungan yang rumit maka dalam skripsi ini penulis akan membuat suatu program yang tujuannya membantu mempermudah dalam penyusunan diagram jaringan kerja (*network*), penentuan lintasan kritis serta pengoptimalan hari pada suatu penjadwalan proyek.

Hasil pemrograman diharapkan dapat membantu pihak terkait dalam menganalisis penjadwalan proyek, sehingga menjadi lebih cepat dan mudah perhitungannya. Tetapi karena keterbatasan penulis maka penyusunan program ini belum sempurna, sehingga program ini dapat dikembangkan lagi agar lebih mendetail dan lebih lengkap.

## ABSTRACT

Nurjiana, 2011. *The Analysis of The Algorithm of Development Project Scheduling with AOA Model CPM Method*. Thesis. Mathematic Department, Faculty of Science and Technology, Islam State University (UIN) of Maulana Malik Ibrahim Malang.

**Promotor :** (1) M. Jamhuri, M.Si  
(2) Abdussakir, M.Pd

**Keywords:** *Network, Critical Path Method, Activity on Arrow, Scheduling*

Operational Research is one management equipment to integrate science, mathematic and logic to solve the daily problems such that the problem can be solved optimal. It may help the implementation of project development schedule to optimize the timing.

A popular technique in the scheduling is network planning. Indeed, work network planning is the most widely used model in the project implementation which is successfully giving the information about the activities in the network diagram. AOA (activity on arrow) Model CPM (Critical Path Method) Method is a technique of work network planning. The method is analyzing the timing and cost of the execution of project development work. Due to its complex calculations, the thesis only discusses the program which is made to facilitate the setup of work network diagram, critical path determination, and the optimization of day in a project schedule.

The programming should be important to help the faster and easier calculated analysis of project schedule. However, the author understands the limitation of this less perfect program. Therefore, the detail and more complete program must be developed in the future.

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pesatnya perkembangan ilmu pengetahuan dan teknologi membuat matematika menjadi sangat penting artinya, bahkan dapat dikatakan bahwa perkembangan ilmu pengetahuan dan teknologi tersebut tidak lepas dari peranan matematika. Tidak dapat dipungkiri bahwa matematika telah menjadi elemen dasar bagi perkembangan ilmu pengetahuan dan teknologi.

Riset operasi adalah salah satu dari cabang ilmu matematika. Menurut Miller dan MK Starr, riset operasi merupakan peralatan manajemen yang menyatukan ilmu pengetahuan, matematika dan logika dalam kerangka pemecahan masalah yang dihadapi sehari-hari, sehingga akhirnya permasalahan tersebut dapat diselesaikan secara optimal (Zulfikarijah, 2004:7).

Riset operasi sangat membantu dalam menyelesaikan manajemen proyek, salah satunya digunakan untuk menyelesaikan penjadwalan pembangunan proyek yang sangat diperlukan untuk mengoptimalkan waktu. Suatu contoh yaitu pembangunan gedung perkantoran. Banyak yang harus diperhatikan dalam proses pembangunan seperti perencanaan, penjadwalan, pengorganisasiandan masih banyak lainnya yang memerlukan ketelitian yang sangat tinggi. Sehingga diperlukan manajemen proyek yang memiliki efisiensi dan efektifitas dalam proses panjadwalan suatu proyek pembangunan yang akhirnya dapat diperoleh waktu optimal dalam penyelesaian proses pembangunan.

Di dalam Islam juga diajarkan agar pentingnya menggunakan waktu dengan sebaik-baiknya. Sebagaimana dalam suatu perkataan sahabat:

الْوَقْتُ كَالسَّيْفِ إِنْ لَمْ تَقْطَعْهُ قَطَعَكَ

Artinya:

*Waktu bagaikan pedang, jika kamu tidak memotongnya ia yang akan memotongmu.*

Melalaikan waktu adalah suatu kebinasaan. Bisa jadi tanpa disadari waktu akan membunuh manusia, jikalau manusia lalai dengan waktu, karena waktu begitu cepat berlalu dan tidak akan pernah kembali, terus berlari tanpa henti, terus berputar tanpa kenal lelah dan tidak menghiraukan apa yang ditinggalkannya. Jadi jika manusia tidak memanfaatkan waktu dengan sebaik baiknya, maka manusia akan binasa sebagaimana binasanya seseorang yang terkena sabetan pedang jika dia tidak segera menghindar dan melawannya maka dia (pedang) akan memotong dan menghancurkannya, karena waktu bagaikan pedang yang membunuh.

Di dalam suatu manajemen proyek, pengendalian proyek mempunyai manfaat yang penting untuk meminimumkan biaya produksi sehingga dapat menghasilkan keuntungan yang optimal. Suatu proyek yang dapat diselesaikan tepat waktu atau lebih cepat daripada jadwal yang ditetapkan akan memberikan banyak keuntungan bagi perusahaan melalui penghematan sumberdaya yang digunakan seperti upah tenaga kerja, biaya sewa peralatan dan sebagainya.

Dalam penjadwalan proyek terdapat berbagai teknik yang dapat digunakan. Teknik yang populer dalam penjadwalan proyek dapat dikelompokkan ke dalam dua metode, yaitu bagan balok dan perencanaan jaringan kerja.

Metode bagan balok yang paling umum dipakai adalah Gantt chart, yang diperkenalkan oleh Henry L. Gantt pada tahun 1916. Bagan ini menggambarkan elemen-elemen kegiatan dari suatu proyek, dalam susunan vertikal dan kronologis waktu pelaksanaan proyek, dalam arah horisontal, dengan menggunakan skala waktu yang proporsional (Herjanto, 2008:358).

Sedangkan perencanaan jaringan kerja (*network planning*) adalah salah satu model yang banyak digunakan dalam penyelenggaraan proyek, yang dapat memberikan informasi tentang kegiatan-kegiatan yang ada dalam diagram jaringan itu. Dengan perencanaan jaringan kerja dapat dilakukan analisis terhadap jadwal waktu selesainya proyek, masalah yang mungkin timbul kalau terjadi keterlambatan, probabilitas selesainya proyek, biaya yang diperlukan dalam rangka mempercepat penyelesaian proyek dan sebagainya. Dua teknik perencanaan jaringan kerja yang populer adalah CPM (*Critical Path Method*) diperkenalkan oleh JE Kelly dari Remington Rand dan MR Walker dari duPont pada tahun 1957 dan PERT (*Program Evaluation And Review Technique*) yang dikembangkan oleh US Navy bekerjasama dengan Booz, Allen & Hamilton pada tahun 1958 (Herjanto, 2008:358).

Metode PERT (*Program Evaluation and Review Technique*) dan CPM (*Critical Path Method*) digunakan untuk menentukan jalur kritis suatu proyek. Jalur kritis/lintasan kritis adalah suatu jalur yang memiliki kegiatan dengan total waktu yang paling lama dibandingkan dengan semua lintasan lain yang mungkin. Jumlah waktu pada lintasan kritis sama dengan umur proyek. Lintasan kritis adalah lintasan yang terdiri dari kegiatan-kegiatan kritis, lintasan kritis merupakan

suatu hal yang selalu menjadi perhatian dalam penjadwalan proyek disamping umur proyek, karena terlambat atau tidaknya proyek tergantung dari terlambat atau tidaknya kegiatan yang berada pada lintasan kritis itu.

Metode PERT dan CPM memecahkan masalah dengan perhitungan maju, perhitungan mundur dan perhitungan kelonggaran waktu (*float*). *Float* merupakan lintasan yang tidak kritis yang bisa memperlambat waktu. Setiap jaringan memiliki titik inisiasi sebagai titik awal dan titik terminasi sebagai tanda berakhirnya suatu proses produksi.

Bagian terpenting dalam keberhasilan pengembangan penerapan riset operasi adalah kemajuan yang terjadi dalam bidang teknologi, khususnya komputer. Perkembangan teknologi komputer yang cukup pesat telah merambah ke hampir semua sektor kehidupan manusia dan dapat pula digunakan sebagai salah satu alat bantu untuk menyelesaikan permasalahan-permasalahan matematika sehingga permasalahan yang sebelumnya sulit atau bahkan tidak dapat dipecahkan karena perhitungannya yang rumit menjadi lebih mudah penyelesaiannya. Oleh sebab itu, penulis akan membuat suatu program yang tujuannya membantu mempermudah dalam penyusunan diagram jaringan kerja (*network*), penentuan lintasan kritis serta pengoptimalan hari pada suatu penjadwalan proyek.

Dalam penjadwalan proyek, teknik perencanaan jaringan kerja yang populer adalah PERT (*Program Evaluation And Review Technique*) dan CPM (*Critical Path Method*). Perbedaan keduanya, PERT hanya mempertimbangkan waktu penyelesaiannya saja dengan waktu kegiatan yang relatif (tidak pasti) dan

CPM berusaha mengoptimalkan biaya total proyek untuk jangka waktu penyelesaian yang dapat dicapai dengan mempertimbangkan waktu yang pasti. Dalam penggambaran diagram jaringan kerja CPM menggunakan AOA (*Activity On Arrow*), yang menggunakan anak panah sebagai simbol dari kegiatan. Sedangkan PERT menggunakan pendekatan AON (*Activity On Node*), yang menggunakan lingkaran (*node*) sebagai simbol kegiatan. Karena PERT hanya mempertimbangkan waktu penyelesaiannya saja, sedangkan CPM menganalisis waktu dan biaya dalam menyelesaikan pekerjaan, maka dalam analisis program penulis memilih menggunakan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*).

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas, maka penulis dapat merumuskan masalah dalam penelitian ini, yaitu sebagai berikut:

1. Bagaimana algoritma, bagan alir dan susunan program komputer untuk menyusun program penyusunan diagram jaringan kerja dan penentuan lintasan kritis dalam penjadwalan proyek dengan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*)?
2. Bagaimana analisis algoritma dari perhitungan maju, perhitungan mundur dan perhitungan kelonggaran waktu (*float*) untuk penentuan lintasan kritis dalam penjadwalan proyek dengan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*)?

### 1.3. Batasan Masalah

Dalam penelitian ini penulis hanya menganalisis waktu dalam penjadwalan proyek pembangunan dan dalam analisis algoritma penulis menganalisis banyak kerja/operasi (penjumlahan, perbandingan) yang dilakukan oleh algoritma. Dalam pembuatan program penulis menggunakan bahasa pemrograman Visual Basic 6.0.

### 1.4. Tujuan

1. Membuat algoritma, bagan alir dan susunan program komputer untuk menyusun program penyusunan diagram jaringan kerja dan penentuan lintasan kritis dalam penjadwalan proyek dengan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*).
2. Menganalisis algoritma dari perhitungan maju, perhitungan mundur dan perhitungan kelonggaran waktu (*float*) untuk penentuan lintasan kritis dalam penjadwalan proyek dengan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*)

### 1.5. Manfaat

1. Bagi penulis, menambah wawasan serta meningkatkan keterampilan dalam mengaplikasikan pengetahuan pada program komputer.
2. Bagi pihak akademis, sebagai bahan pengembangan dan perbaikan keilmuan, dalam memadukan antara sains dan teknologi.
3. Bagi pengguna metode CPM model AOA, dapat mempermudah proses perhitungan dalam menjadwalkan suatu proyek, sehingga proses perhitungannya menjadi lebih cepat dan mudah.

## **1.6. Sistematika Penulisan**

Untuk mempermudah dalam memahami skripsi ini, penulis menggunakan sistematika penulisan lima bab, masing-masing bab dijelaskan sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab ini berisi: latar belakang, rumusan masalah, fokus penelitian, tujuan penelitian, manfaat peneltiandan sistematika pembahasan.

### **BAB II KAJIAN PUSTAKA**

Pada bab ini memberikan kajian-kajian yang menjadi landasan masalah yang dibahas, yaitu penjadwalan proyek dengan CPM (*Critical Path Method*), simbol-simbol yang digunakan dalam penggambaran diagram *network*, penentuan waktu, asumsi dan cara perhitungan, algoritma pemrogramandan sebagainya.

### **BAB III METODE PENELITIAN**

Pada bab ini berisi: jenis data, sumber data, teknis analisis data, tahap pemrograman.

### **BAB IV PEMBAHASAN**

Pada bab ini membahas mengenai algoritma, bagan alirdan program komputer untuk menyusun program penyusunan diagram jaringan kerja, penentuan lintasan kritis dan waktu optimal penjadwalan proyek.

### **BAB V PENUTUP**

Pada bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran bagi pembaca yang akan melanjutkan penelitian dalam skripsi ini.

## BAB II

### KAJIAN PUSTAKA

#### 2.1. Penjadwalan Proyek Dengan CPM (*Critical Path Method*)

Suatu proyek mendefinisikan satu kombinasi kegiatan-kegiatan yang saling berkaitan yang harus dilakukan dalam urutan tertentu sebelum keseluruhan tugas dapat diselesaikan. Kegiatan-kegiatan ini saling berkaitan dalam satu urutan yang logis dalam arti bahwa beberapa kegiatan tidak dapat dimulai sampai kegiatan-kegiatan lainnya diselesaikan. Suatu kegiatan dalam suatu proyek biasanya dipandang sebagai suatu tugas yang memerlukan waktu dan sumber daya untuk penyelesaiannya. Pada umumnya, suatu proyek adalah satu usaha satu kali, yaitu urutan kegiatan yang sama kemungkinan tidak diulangi di masa mendatang (Taha, 1996:75).

Manajemen proyek telah berkembang sebagai satu bidang baru dengan dikembangkannya dua teknik analitis untuk perencanaan, penjadwalan dan pengendalian proyek. Keduanya adalah CPM (*Critical Path Method*) diperkenalkan oleh JE Kelly dari Remington Rand dan MR Walker dari duPont pada tahun 1957 dan PERT (*Program Evaluation And Review Technique*) yang dikembangkan oleh US Navy bekerjasama dengan Booz, Allen & Hamilton pada tahun 1958. Kedua teknik ini mempunyai kemiripan yang besar. Keduanya memvisualisasikan proyek dalam bentuk diagram jaringan kerja (*network diagram*). CPM maupun PERT, keduanya banyak dipergunakan dalam pengendalian proyek, misalnya dalam kegiatan militer, konstruksi, instalasi, pabrikasi dan pengairan. Perbedaan antara keduanya dalam penggambaran

diagram jaringan kerja adalah CPM menggunakan *activity on arrow* (AOA), yang menggunakan anak panah sebagai simbol dari kegiatan. Sedangkan PERT menggunakan pendekatan *activity on node* (AON), yang menggunakan lingkaran (*node*) sebagai simbol kegiatan (Herjanto, 2008:359-360).

PERT dan CPM pada dasarnya merupakan metode-metode yang berorientasi pada waktu dalam arti bahwa keduanya mengarah pada penentuan suatu jadwal. Walaupun dikembangkan secara independen, keduanya secara mengejutkan sangat serupa. Kemungkinan perbedaan yang terpenting adalah bahwa pada awalnya estimasi waktu untuk kegiatan-kegiatan diasumsikan bersifat deterministik dalam CPM dan Probabilistik dalam PERT. Saat ini, PERT dan CPM pada kenyataannya membentuk satu teknik dan perbedaannya jika ada, hanyalah bersifat historis. Konsekwensinya, kedua teknik ini disebut sebagai teknik-teknik “penjadwalan proyek” (Taha, 1996:76).

Penjadwalan proyek dengan PERT – CPM terdiri dari tiga tahap dasar : perencanaan, penjadwalan dan pengendalian.

Tahap perencanaan dimulai dengan memecahkan proyek ke dalam beberapa kegiatan yang berbeda. Estimasi waktu untuk kegiatan-kegiatan ini lalu ditentukandan suatu diagram jaringan (*network*) dikembangkan dengan masing-masing busur (panah) mewakili satu kegiatan antara kegiatan-kegiatan dalam proyek tersebut. Pengembangan diagram jaringan sebagai satu tahap perencanaan memiliki keuntungan berupa dipelajarinya tugas-tugas yang berbeda secara terperinci, sambil kemungkinan menyarankan perbaikan-perbaikan sebelum proyek tersebut benar-benar dilaksanakan. Yang lebih penting lagi adalah

penggunaannya untuk mengembangkan suatu jadwal untuk proyek tersebut (Taha, 1996:76).

Tujuan akhir dari tahap penjadwalan adalah dikembangkannya suatu bagan waktu yang memperlihatkan waktu awal dan akhir untuk setiap kegiatan disamping hubungannya dengan kegiatan-kegiatan lain dalam proyek tersebut. Di samping itu, jadwal tersebut harus menunjukkan kegiatan-kegiatan kritis (dalam hal waktu) yang memerlukan perhatian khusus jika proyek tersebut ingin diselesaikan pada waktunya. Untuk kegiatan-kegiatan yang tidak kritis, jadwal tersebut harus menunjukkan jumlah kesenggangan atau waktu mengambang yang dapat dipergunakan secara menguntungkan ketika kegiatan-kegiatan tersebut ditunda atau ketika sumber daya yang terbatas ingin dipergunakan secara efisien (Taha, 1996:76).

Tahap terakhir dalam manajemen proyek adalah pengendalian proyek. Ini mencakup penggunaan diagram panah tersebut dan bagan waktu untuk membuat laporan kemajuan secara rutin. Jaringan tersebut dapat terus diperbaiki dan dianalisis dan jika perlu, suatu jadwal baru ditentukan untuk bagian yang tersisa dari proyek tersebut (Taha, 1996:76).

Pada dasarnya diagram jaringan (*network*) adalah suatu perencanaan, penjadwalan dan pengendalian suatu proses produksi. Oleh karena itu *network* dapat dijadikan sebagai sarana untuk mengkoordinasi semua pekerjaan berdasarkan sumber daya yang diinginkan sehingga proses produksi dapat diselesaikan dengan cepat dan total biaya minimum (Ervianto, 2005:233-234).

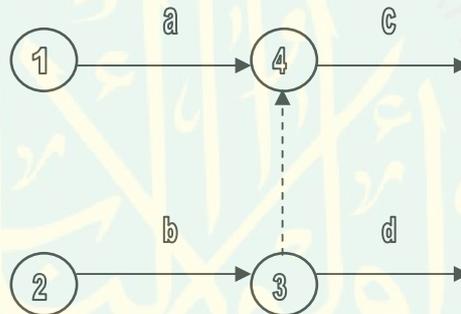
## 2.2. Simbol-Simbol yang Digunakan

Dalam menggambarkan suatu diagram *network* digunakan simbol-simbol sebagai berikut:

- Anak panah (*arrow*), menyatakan suatu kegiatan atau aktivitas yang memerlukan durasi. Aktivitas merupakan kumpulan gerakan yang merupakan bagian dari aktivitas proses produksi yang mengkonsumsi waktu dan sumber daya serta mempunyai titik awal dan titik akhir. Aktivitas memerlukan gerakan, waktu, uang, energi manusia, fasilitas, peralatan dan berbagai bahan lainnya. Sepanjang panah akan tercantum waktu yang diperlukan untuk menyelesaikan aktivitas.
- Lingkaran kecil (*node*), menyatakan suatu kejadian atau peristiwa atau *event*. Kejadian merupakan suatu kondisi tertentu yang terjadi pada suatu saat tertentu atau suatu waktu tertentu atau suatu penyelesaian dari suatu aktivitas atau suatu kumpulan aktivitas yang tidak menyebutkan waktu penyelesaian. Awal dan akhir suatu aktivitas dinyatakan oleh suatu kejadian atau *event*. Kejadian dinyatakan dalam angka/huruf tunggal satu digit atau angka tunggal dua digit (misal 1, atau 2 atau 3, atau 4 atau 11 atau 12). Sedangkan aktivitas dinyatakan dalam angka/huruf pasangan yang ditandai oleh koma (,) seperti (1,2), (2,4) atau (4,6). Angka awal setiap aktivitas dimulai dari angka terkecil menuju kepada angka terbesar.
- - - → Anak panah putus-putus, menyatakan kegiatan semu atau *dummy*. *Dummy* hanya berguna untuk menunjukkan hubungan antar kegiatan. *Dummy* tidak

mempunyai durasi karena tidak memakai atau menghabiskan sejumlah *resource*.

Fungsi *dummy* di atas adalah memindahkan seketika itu juga (sesuai dengan arah panah). Misalkan 4 macam kegiatan yang berbeda A, B, C dan D merupakan 4 aktivitas yang harus diselesaikan. Aktivitas A dan B harus mendahului aktivitas C tetapi hanya aktivitas B yang mendahului aktivitas D, Hal tersebut dapat dilakukan dengan cara menambah aktivitas semu yang ditunjukkan dengan garis putus-putus seperti ilustrasi di bawah ini :



Gambar 2. 1 Pemakaian *Dummy*

Syarat-syarat pembuatan diagram *network* (Ervianto, 2005:233-234):

1. Dalam penggambaran, diagram *network* harus jelas dan mudah dibaca.
2. Harus dimulai dari *event*/kejadian dan diakhiri pada *event*/kejadian.
3. Kegiatan disimbolkan dengan anak panah yang digambar garis lurus dan boleh patah.
4. Di antara dua kejadian, hanya boleh ada satu anak panah.
5. Penggunaan kegiatan semu (*dummy*) ditunjukkan dengan garis putus-putus dan jumlahnya seperlunya saja.

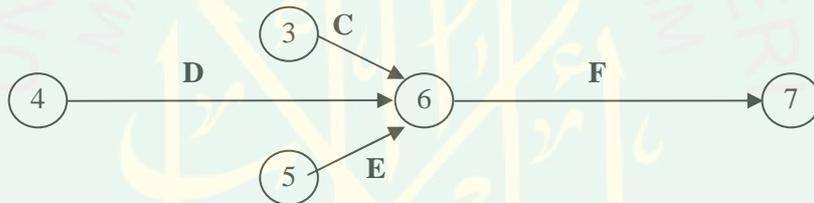
Adapun logika ketergantungan kegiatan-kegiatan pada diagram *network* sebagai berikut:

- a. Jika kegiatan A harus diselesaikan dahulu sebelum kegiatan B dapat dimulai, maka hubungan antara kedua kegiatan tersebut dapat digambarkan sebagai berikut. Kegiatan A dapat ditulis (1,2) dan kegiatan B dapat ditulis (2,3).



**Gambar 2. 2** Kegiatan A Harus Diselesaikan Dahulu Sebelum Kegiatan B Dapat Dimulai

- b. Jika kegiatan C, D dan E selesai sebelum kegiatan F dapat dimulai, maka :



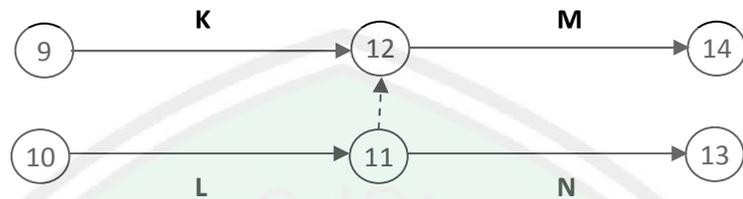
**Gambar 2. 3** Kegiatan C, D dan E Selesai Sebelum Kegiatan F Dapat Dimulai

- c. Jika kegiatan G dan H selesai sebelum kegiatan I dan J, maka :



**Gambar 2. 4** Kegiatan-Kegiatan G dan H Selesai Sebelum Kegiatan I dan J.

- d. Jika kegiatan K dan L harus selesai sebelum kegiatan M dapat dimulai, tetapi kegiatan N sudah boleh dimulai bila kegiatan L sudah selesai, maka:



**Gambar 2. 5** Kegiatan K dan L Harus Selesai Sebelum Kegiatan M Dapat Dimulai, Tetapi Kegiatan N Sudah Boleh Dimulai Bila Kegiatan L Sudah Selesai.

Fungsi *dummy* di atas adalah memindahkan seketika itu juga (sesuai dengan arah panah) keterangan tentang selesainya kegiatan L dari *event* nomor 11 ke *event* nomor 12.

- e. Jika kegiatan P, Q dan R mulai dan selesai pada *event* yang sama, maka harus digambar dengan menggunakan *dummy* sebagai berikut.



**Gambar 2. 6** Kegiatan P, Q dan R Mulai dan Selesai pada *Event* yang Sama

(Dimiyati dan Dimiyati, 1994:178-179).

### 2.3. Penentuan Waktu

Setelah *network* suatu proyek dapat digambarkan, langkah berikutnya adalah mengestimasi (menduga) waktu yang diperlukan untuk masing-masing aktivitas dan menganalisis seluruh diagram *network* untuk menentukan waktu terjadinya masing-masing kejadian (*event*) (Dimiyati dan Dimiyati, 1994:180).

Dalam mengestimasi dan menganalisis waktu ini, akan kita dapatkan satu atau beberapa lintasan tertentu dari kegiatan-kegiatan pada *network* tersebut yang

menentukan jangka waktu penyelesaian seluruh proyek. Lintasan ini disebut lintasan kritis (*critical path*). Di samping lintasan kritis ini terdapat lintasan-lintasan lain yang mempunyai jangka waktu yang lebih pendek daripada lintasan kritis. Dengan demikian, maka lintasan yang tidak kritis ini mempunyai waktu untuk bisa terlambat, yang dinamakan *float* (Dimiyati dan Dimiyati, 1994:180).

Dalam perhitungan penentuan waktu proyek dikenal beberapa istilah, sebagai berikut:

- a. TE (*earliest event occurrence time*), yaitu saat tercepat terjadinya *event*.
- b. TL (*latest event occurrence time*), yaitu saat paling lambat terjadinya *event*.
- c. ES (*earliest activity start time*), yaitu saat tercepat dimulainya kegiatan.
- d. EF (*earliest activity finish time*), yaitu saat tercepat diselesaikannya kegiatan.
- e. LS (*latest activity start time*), yaitu saat paling lambat dimulainya kegiatan.
- f. LF (*latest activity finish time*), yaitu saat paling lambat diselesaikannya kegiatan.
- g. t (*activity duration time*), yaitu waktu yang diperlukan untuk suatu kegiatan (biasanya dinyatakan dalam hari).
- h. TF (*total float/Slack*), merupakan jumlah waktu yang mana waktu penyelesaian suatu kegiatan dapat diundur tanpa mempengaruhi saat tercepat dari penyelesaian proyek secara keseluruhan.
- i. FF (*free float/Slack*), merupakan jumlah waktu yang mana penyelesaian suatu kegiatan dapat diukur tanpa mempengaruhi saat tercepat dari dimulainya kegiatan yang lain (Dimiyati dan Dimiyati, 1994:181).

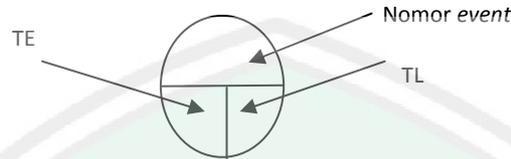
## 2.4. Asumsi dan Cara Perhitungan

Dalam melakukan perhitungan penentuan waktu ini digunakan tiga buah asumsi dasar yaitu :

1. Proyek hanya memiliki satu *initial event* dan satu *terminal event*.
2. Saat tercepat terjadinya *initial event* adalah hari ke nol.
3. Saat paling lambat terjadinya *terminal event* adalah  $TL = TE$  untuk *event* ini.

Adapun cara perhitungan penentuan waktu terdiri dari dua cara, yaitu cara perhitungan maju (*forward computation*) dan perhitungan mundur (*backward computation*). Pada perhitungan maju, perhitungan bergerak mulai dari *initial event* menuju ke *terminal event*. Maksudnya ialah menghitung saat yang paling cepat terjadinya *events* dan saat paling cepat dimulainya serta diselesaikannya kegiatan-kegiatan (TE, ES dan EF). Pada perhitungan mundur, perhitungan bergerak dari *terminal event* menuju ke *initial event*. Tujuannya ialah untuk menghitung saat paling lambat terjadinya *events* dan saat paling lambat dimulainya dan diselesaikannya kegiatan-kegiatan (TL, LS dan LF). Dengan selesainya perhitungan ini barulah *float* dapat dihitung (Dimiyati dan Dimiyati, 1994:181).

Untuk melakukan perhitungan maju dan perhitungan mundur ini, lingkaran kejadian(event) dibagi atas tiga bagian sebagai berikut:



Gambar 2. 7 Lingkaran Kejadian (event)

#### 1.4.1. Perhitungan Maju (Forward Computation)

Ada tiga langkah yang dilakukan pada perhitungan maju, yaitu :

- a. Saat tercepat terjadinya *initial event* ditentukan pada hari ke nol maka untuk *initial event* berlaku  $TE(i) = 0$  (adanya asumsi awal bahwa proyek tidak berhubungan dengan proyek yang lain).
- b. Saat paling cepat untuk menyelesaikan kegiatan (i,j) sama dengan saat paling cepat untuk memulai kegiatan ditambah dengan durasi kegiatan tersebut.

Apabila *initial event* terjadi pada hari yang ke-nol, maka:

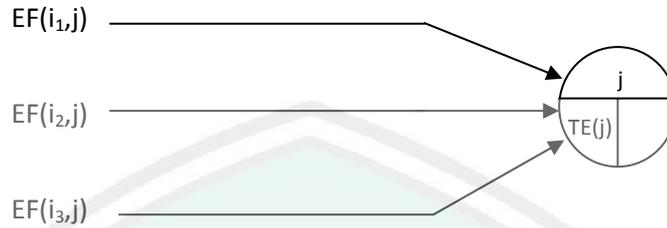


Gambar 2. 8 Saat Paling Cepat untuk Menyelesaikan Kegiatan

$$ES(i,j) = TE(i) = 0 \tag{2.1}$$

$$EF(i,j) = ES(i,j) + t(i,j) = TE(i) + t(i,j) \tag{2.2}$$

c. *Merge event*, merupakan *event* yang menggabungkan beberapa kegiatan



Gambar 2. 9 Merge Event

Didefinisikan:

$$TE(j) = maks \{EF(i_{1,j}), EF(i_{2,j}), \dots, EF(i_{n,j})\} \tag{2.3}$$

(Dimiyati dan Dimiyati, 1994:182-183).

#### 2.4.2. Perhitungan Mundur (*Backward Computation*).

Ada tiga langkah yang dilakukan pada perhitungan mundur, yaitu :

- a. Pada *terminal event* berlaku  $TL = TE$
- b. Saat paling lambat untuk memulai suatu kegiatan sama dengan saat paling lambat untuk menyelesaikan kegiatan itu dikurangi dengan durasi kegiatan tersebut.



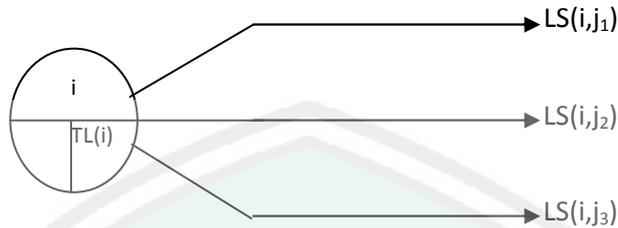
Gambar 2. 10 Saat Paling Lambat untuk Memulai Kegiatan

$$LS = LF - t(i,j) \tag{2.4}$$

$$LF(i,j) = TL \text{ dimana } TL = TE \tag{2.5}$$

$$LS(i,j) = TL(j) - t(i,j) \tag{2.6}$$

c. *Burst event*, merupakan event yang mengeluarkan beberapa kegiatan.



Gambar 2. 11 *Burst Event*

Didefinisikan:

$$TL(i) = \min \{ LS(i,j_1), LS(i,j_2), \dots, LS(i,j_n) \} \quad (2.7)$$

(Dimiyati dan Dimiyati, 1994:185).

#### 1.4.3. Perhitungan Kelonggaran Waktu (*Float or Slack*).

Setelah melakukan perhitungan maju dan mundur langkah selanjutnya adalah melakukan perhitungan kelonggaran waktu (*float/slack*). Dari aktifitas (*i,j*), yang terdiri atas *total float* dan *free float*.

*Total float* dihitung dengan cara mencari selisih antara saat paling lambat dimulainya aktivitas dengan saat paling cepat dimulainya aktivitas (LS-ES), atau bisa juga dengan mencari selisih antara saat paling lambat diselesaikannya aktivitas dengan saat paling cepat diselesaikannya aktivitas (LF-EF). Dalam hal ini cukup dipilih salah satu saja.

Jika akan menggunakan persamaan  $TF = LS - ES$ , maka *total float* aktivitas (*i,j*) adalah:

$$TF(i,j) = LS(i,j) - ES(i,j)$$

Dari perhitungan mundur diketahui bahwa  $LS(i,j) = TL(j) - t(i,j)$ , sedangkan dari perhitungan maju  $ES(i,j) = TE(i)$ . Maka:

$$TF(i,j) = TL(j) - t(i,j) - TE(i) \quad (2.8)$$

*Free float* aktivitas (i,j) dihitung dengan cara mencari selisih antara saat tercepat terjadinya *event* di ujung aktivitas dengan saat tercepat diselesaikannya aktivitas(i,j) tersebut. Atau:

$$FF(i,j) = TE(j) - EF(i,j)$$

Dari perhitungan maju didapat  $EF(i,j) = TE(i) - t(i,j)$ , maka:

$$FF(i,j) = TE(j) - TE(i) - t(i,j) \quad (2.9)$$

(Dimiyati dan Dimiyati, 1994:187).

## 2.5. Lintasan Kritis (*Critical Path*)

Suatu kegiatan yang tidak mempunyai kelonggaran (*float*) disebut kegiatan kritis. Dengan kata lain, kegiatan kritis mempunyai  $TF(i,j) = FF(i,j) = 0$ . Kegiatan-kegiatan kritis ini akan membentuk lintasan kritis yang biasanya dimulai dari *initial event* sampai ke *terminal event*, jadi lintasan kritis merupakan suatu lintasan di mana di dalamnya terdapat kegiatan-kegiatan yang saling berurutan pelaksanaannya dari tanggal di mulainya proyek sampai berakhirnya proyek, akibatnya pelaksanaan kegiatan-kegiatan yang ada pada lintasan ini tidak boleh mengalami penundaan, karena penundaan akan menyebabkan terlambatnya proyek yang dijalankan.

## 2.6. Algoritma Pemrograman

Telah diketahui bersama bahwa komputer merupakan alat bantu yang dapat memecahkan berbagai masalah yang ditemui oleh seorang *user*, masalah yang mudah sampai dengan masalah yang sangat kompleks. Dalam usaha untuk memecahkan suatu masalah tersebut, seorang *user* dituntut dapat mengubah atau menyajikan masalah yang ada dalam suatu model atau konsep yang tepat bagi komputer sehingga dapat digunakan untuk memecahkan permasalahan tersebut dengan mudah. Model atau konsep tersebut biasanya dibuat dalam suatu konsep atau model matematika.

Untuk mendapatkan suatu model matematika yang tepat sangatlah tidak mudah. Oleh sebab itu, diperlukan studi lebih lanjut baik dalam bidang matematika atau dalam bidang ilmu komputer itu sendiri atau yang lainnya.

Usaha yang dilakukan agar solusi suatu masalah dapat diperoleh adalah membuat suatu proses atau prosedur yang merupakan urutan dari langkah-langkah atau instruksi-instruksi yang berintegrasi. Langkah-langkah instruksi tersebut secara umum dikenal dengan ALGORITMA. Istilah tersebut pertama kali dikenal oleh seorang Ahli Matematika yang bernama Abu Ja'far Muhammad Ibnu Musa Al Khawarizmi. Selain itu ada juga definisi menurut suatu kamus bahasa bahwa Algoritma adalah suatu metode khusus yang digunakan untuk menyelesaikan suatu masalah yang nyata (Yulikuspartono, 2004:2).

Setelah langkah-langkah instruksi atau Algoritma tersebut selesai disusun, berikutnya *user* harus menyajikan langkah-langkah tersebut dalam satu bahasa yang dikenal oleh komputer. Penyajian tersebut harus teratur dan memenuhi

kaidah persyaratan yang ditentukan dalam suatu bahasa komputer. Kegiatan menulis atau membuat langkah-langkah instruksi tersebut dalam suatu bahasa komputer dikenal dengan nama PEMROGRAMAN. Hasil yang telah dicapai oleh penulisan tersebut disebut dengan PROGRAM. Jadi suatu algoritma pada hakikatnya merupakan suatu prosedur yang tepat untuk dapat memecahkan masalah dengan menggunakan bantuan komputer serta suatu bahasa pemrograman tertentu (Yulikuspartono, 2004:2).

### 2.6.1. Tipe Data Record

Visual Basic 6.0 mengizinkan pengguna/pemrogram membuat tipe data sendiri. Ada beberapa jenis tipe data buatan sendiri, salah satunya adalah tipe data **Record**. Tipe data **Record** mengelompokkan variabel-variabel yang berhubungan dalam satu nama yang sama. Masing-masing variabel yang berada dalam suatu tipe data **Record** dapat memiliki tipe data yang berbeda-beda.

Untuk membuat tipe data **Record**, struktur penulisan yang digunakan adalah sebagai berikut:

```
Private/Public Type NamaTipeData
```

```
    Var1 as TipeData
```

```
    Var2 as TipeData
```

```
    ...
```

```
End Type (Ramadhan, 2004:25).
```

### 2.6.2. If (Pemilihan)

If (pemilihan) adalah suatu kondisi dimana satu atau lebih baris kode program (*statement*) akan dieksekusi jika memenuhi kondisi tertentu. Jika kondisi tidak terpenuhi, program akan mengeksekusi baris kode program yang lain.

#### a. *Statement If Tunggal*

```
If <condition> Then <Statement>
```

#### b. *Statement If Ganda*

```
If <condition1> Then
  <Statement1>
Else
  <Statement2>
End If
```

#### c. *Statement If Majemuk*

```
If <condition1> Then
  <Statement1>
ElseIf <condition2> Then
  <Statement2>
Else
  <Statement3>
End If
```

#### d. *Statement Select Case*

```
Select Case <name_var>
  Case <condition1> : <Statement1>
  Case <condition2> : <Statement2>
  Case Else: <Statement3>
End Select
```

### 2.6.3. Loop (Perulangan)

Loop (perulangan) adalah suatu kondisi dimana satu atau beberapa baris kode program (*Statement*) dieksekusi secara berulang-ulang.

#### a. *Statement For*

```
For <name_var> = <first> to <last> [Step <diff>]
  <Statement>
  [Exit For]
Next
```

b. *Statement Do While*

```
Do While <condition>  
  <statement>  
  [Exit Do]  
Loop
```

c. *Statement Do-Loop While*

```
Do  
  <statement>  
  [Exit Do]  
Loop While <condition>
```

d. *Statement Do Until*

```
Do Until <condition>  
  <statement>  
  [Exit Do]  
Loop
```

e. *Statement Do-Loop Until*

```
Do  
  <statement>  
  [Exit Do]  
Loop Until <condition>
```

f. *While-Wend*

```
While <condition>  
  <statement>  
Wend
```

#### 2.6.4. Function (Fungsi)

Fungsi adalah sub program yang mengembalikan nilai. Suatu Fungsi memiliki aturan penulisan sebagai berikut;

```
<Private/Public>Function <N_Fungsi> (<Arg> As <TipeArg> As  
<Tp_Dta>  
  ....  
<N_Fungsi> = <Nilai_Kembalian>  
End Funtion
```

### 2.6.5. Array

Array merupakan kumpulan data yang bertipe sama yang menggunakan nama sama. Dengan menggunakan array, sejumlah variabel dapat memakai nama yang sama.

Cara untuk membuat indeks dari Array dimulai dari satu:

```
Option Base 1
```

Cara membuat indeks dari Array bersifat statik (ukurannya tetap) adalah sebagai berikut:

```
Dim <name_var> (indeks) As <type_var>
```

Sedangkan cara membuat indeks dari Array bersifat dinamik (ukurannya dapat berubah) adalah sebagai berikut:

```
Dim <name_var> () As <type_var>
```

Berikutnya cara untuk mengubah jumlah indeks yang dipunyai oleh suatu Array:

```
Redim <name_var> (A)
```

Yang terakhir cara untuk mengubah jumlah indeks dari Array dan membuat nilai sebelumnya tidak hilang:

```
Redim Preserve <name_var> (B)
```

### 2.6.6. Menampilkan Array dengan MSFlexGrid

VB menyediakan mesin objek yang cocok digunakan untuk menampilkan data array karena dapat menampilkan kotak atau grid dan membentuk matrik dua dimensi, yang memiliki subskrips yang sama dengan array itu sendiri, yaitu kontrol Microsoft FlexGrid (MSFlexGrid). Kotak yang selanjutnya disebut sebagai sel dapat diisi data berupa string atau gambar.

### **a. Instalasi Kontrol MSFlexGrid**

Agar dapat digunakan dalam penulisan program, maka file MSFlexGrid.ocx harus ditambahkan pada proyek yang sedang dikerjakan. Caranya klik dari menu Project, Components sehingga kotak dialog Components muncul. Pada kotak dialog, pilih tab Control yang menyediakan daftar kontrol yang dapat di instal. Pilih kontrol dengan nama **Microsoft FlexGrid Control 6.0 (SP3)**.

### **b. Properti Penting yang Perlu Diketahui**

#### **i. Properti Row, Rows, Cols dan Cols**

Properti Rows dan Cols untuk menentukan jumlah baris dan jumlah kolom pada grid yang ditampilkan. Properti Row dan properti Col menetapkan lokasi sel yang aktif, kolom dihitung dari kiri ke kanan dan sedang baris dihitung dari atas ke bawah. Baris atau kolom pertama berindeks 0, indeks akan bertambah untuk setiap baris di bawahnya sampai dengan indeks maksimum untuk baris, yaitu Rows-1 dan indeks kolom bertambah sampai indeks kolom maksimum, yaitu Cols-1.

Baris berindeks 0 (paling atas) adalah baris judul horizontal, hanya bergeser jika digunakan scrollbar horizontal. Kolom berindeks 0 (paling kiri) adalah kolom judul vertikal, hanya bergeser jika digunakan scrollbar vertikal. Pengaturan properti Row dan Col dapat dilakukan dalam kode program atau saat program dijalankan, yaitu melalui tombol mouse atau tombol panah.

#### **ii. Properti ColSel, Rowsel**

Properti ColSel menetapkan kolom awal atau akhir daerah sel yang dipilih.

Properti RowSel menetapkan baris awal atau akhir daerah sel yang dipilih.

Cara penulisannya:

```
Object. ColSel[=value]
Object. RowSel[=value]
```

Mengubah properti Row dan Col secara otomatis mengubah isi properti RowSel dan ColSel. Sel yang dipilih adalah sel yang sekarang digunakan. Jika memilih suatu blok sel, maka pertama harus menetapkan Row dan Col, kemudian dilanjutkan ke RowSel dan ColSel.

### iii. Properti Text dan WordWrap

Properti Text adalah isi sel berupa data string yang alamatnya ditentukan dari properti Col dan Row. String yang terlalu panjang untuk ditampilkan dapat diminta berpindah baris jika properti Wordrap diadika True.

### iv. Properti ColWidth dan RowHeight

Jika sel untuk menampilkan teks terlalu kecil atau besar, maka lebar kolom dapat diubah dengan properti ColWidth, sedangkan tinggi baris diubah dengan properti RowHeight. Satuan ukuran dalam twips.

```
Object. ColWidth(indeks_kol)[=value]
Object. RowHeight (indeks_bers)[=value]
```

Jika indeks\_bers= -1 maka merujuk pada semua baris. Jika value=0 maka sel yang bersangkutan akan disembunyikan. Jika value=-1 maka mengikuti nilai default yang tergantung pada ukuran font yang digunakan.

### v. Properti TextMatrix

Fungsinya sama dengan propert Text tetapi tidak perlu menyatakan terlebih dahulu properti Row dan Col secara terpisah, karena fungsinya digantikan oleh parameter indeks\_bers dan indeks\_kol di dalamnya. Contoh penulisan:

```
Object.TextMatrix (indeks_bers, indeks_kol)[=string]
```

Pemakaian properti ini tidak mengubah nilai properti Row dan Col, artinya secara fisik tidak masuk ke sel tersebut yang dalam properti Row dan Col.

#### **vi. Properti ColAlignment**

Mengatur penempatan data di kolom.

```
Object. ColAlignment (indeks)[=nilai]
```

Isi setiap kolom dapat diatur tersendiri dengan properti ColAlignment dan berbeda dengan kolom yang lainnya. Untuk mengatur kolom saat design time, gunakan properti FormatString. Jika mengatur sel dan bukan kolom, gunakan properti CellAlignment.

#### **vii. Properti BackColor, BackColorBkg, BackColorFixed, BackColorSel**

Mengatur warna latar belakang dari elemen MSFlexGrid, penulisannya:

```
Object. BackColor[=color]
Object. BackColorFixed[=color]
Object. BackColorSel[=color]
Object. BackColorBkg[=color]
```

#### **viii. Properti CellBackColor, CellForeColor**

CellBackColor digunakan untuk mengubah atau mengatur latar belakang warna sel secara individu, sedangkan warna depan sel secara individu digunakan CellForeColor. Hanya aktif jika digunakan dalam program.

```
Object. CellBackColor[=color]
Object. CellBackColorFixed[=color]
```

#### **ix. Properti CellHeight, CellLeft, CellTop, CellWidth**

Menentukan posisi dan ukuran dari sel yang sedang aktif. Unit yang digunakan adalah twips. Hanya dapat digunakan dalam pemrograman.

```
Object. CellHeight [=value]
Object. CellLeft [=value]
Object. CellTop [=value]
Object. CellWidth [=value]
```

Properti ini sangat berguna untuk mengemulasi sel agar dapat diedit. Dengan memanfaatkan event KeyPress event maka suatu kontrol TextBox atau yang sejenis ditempatkan di atas sel yang sedang aktif untuk menerima perintah edit data dari sel tersebut.

#### **x. Properti ScrollBars**

Memberi nilai yang menetapkan apakah MSFlexGrid mempunyai scroll bar horizontal dan/atau vertical atau tidak ada sama sekali. Penulisan:

```
Object. ScrollBars[=nilai]
```

ScrollBars akan muncul jika jumlah sel melebihi garis batas kontrol dan jika nilai yang menetapkan ScrollBars=0, maka scroll bar tidak akan keluar meskipun isinya melebihi garis batas.

#### **xi. Properti FixedCols, FixedRows**

FixedCols – menetapkan jumlah kolom tetap (kolom judul).

FixedRows – menetapkan jumlah baris tetap (baris judul).

Tanpa penetapan khusus, MSFlexGrid mempunyai satu kolom/baris tetap untuk judul. Cara penulisan:

```
Object. FixedCols[=value]  
Object. FixedRows[=value]
```

Kolom tetap judul pada posisi diam jika scrollbar horizontal aktif, sedangkan baris tetap judul pada posisi diam jika scrollbar vertikal aktif. Kolom atau baris tetap judul dapat dihilangkan dengan memberi nilai value=nol

#### **xii. Properti TextStyle, ...Band, ...Fixed dan ...Header**

Mengatur tampilan style tiga dimensi dari teks di dalam sel yang di pilih.

**TextStyle** – menentukan style sel regular MSFlexGrid.

**TextStyleBand** – menentukan style dari bands.

**TextStyleFixed** – menentukan style dari baris / kolom judul.

**TextStyleHeader** – menentukan style headers.

```
Object.TextStyle[=style]
Object.TextStyleBand(BandNumber)[=style]
Object.TextStyleFixed[=style]
Object.TextStyleHeader(bandNumber)[=style] (Dewobroto, 2004:140-144).
```

### 2.6.7. Fungsi-Fungsi Grafik

Fungsi-fungsi grafik adalah fungsi-fungsi untuk membuat grafik (misalnya garis, kotak, lingkaran dan lain-lain). Agar dapat menggunakan fungsi-fungsi grafik, maka harus menggunakan prosedur Activate atau Paint. Sedangkan Toolbox sudah menyediakan 2 buah kontrol untuk membuat grafik pada form (yaitu kontrol Line dan Shape). Dengan kontrol Shape, dapat dibuat lingkaran, segitiga dan lain-lain.

#### a. Line

Line adalah fungsi untuk membuat garis atau kotak. Bentuk umum: Line (x1,y1)-(x2,y2), warna, mode. Sedangkan mode terdiri dari B dan BF. Jika ingin membuat kotak, gunakan B (kotak tanpa warna isian) atau BF (kotak dengan warna isian)

Contoh:

```
Private Sub Form_Paint()
    Line (100,100)-(100,2000), vbBlue
End Sub
```

akan menghasilkan garis vertikal

## b. Circle

Circle adalah fungsi untuk membuat lingkaran. Bentuk umum: Circle (x,y), Radius, Warna.

Contoh:

```
Private Sub Form_Paint()  
    Circle (1200,1200),1000,vbBlue  
End Sub
```

akan menghasilkan suatu lingkaran dengan warna garis biru (Pardosi, 2005:160-162).

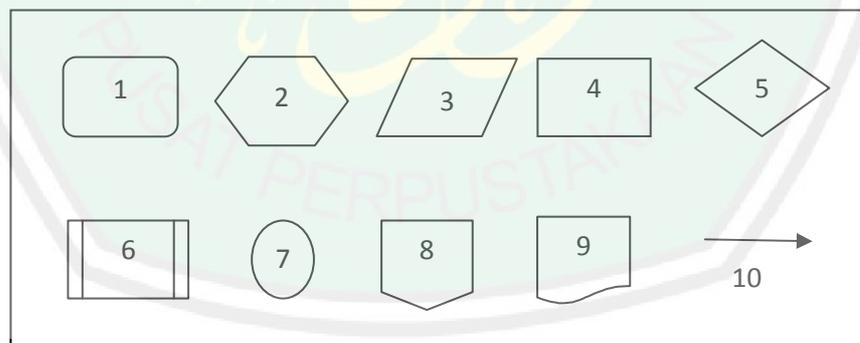
## 2.7. Bagan Alir (*Flowchart*)

Dalam mendeskripsikan prosedur penyelesaian permasalahan dalam bentuk algoritma, seringkali dijumpai kesulitan jika hanya menggunakan kata-kata atau kalimat saja. Dengan demikian, deskripsinya cenderung sulit dipahami dan memungkinkan timbulnya kesalahan interpretasi bagi orang lain. Selain itu, diperlukan banyak keterangan-keterangan yang seringkali justru sulit dimengerti. Belum lagi setiap orang tentu memiliki cara-cara tersendiri yang berbeda sehingga tidak pernah ditemukan keseragaman dalam format penulisannya (Sutanta, 2004:27).

Untuk alasan-alasan tersebut, maka penggunaan *flowchart* akan banyak membantu dan menguntungkan. Dengan *flowchart*, langkah prosedur penyelesaian permasalahan dapat diekspresikan dengan serangkaian simbol grafis yang baku dan lebih mudah digunakan. Penggunaan *flowchart* akan menghindarkan sejak dini timbulnya kesalahan interpretasi bagi orang lain yang merupakan awal kegagalan suatu prosedur yang dikembangkan (Sutanta, 2004:27).

Banyak manfaat akan diperoleh apabila sering menggunakan *flowchart* dalam mengembangkan prosedur pemecahan masalah komputasi. Pertama, akan terbiasa berpikir secara sistematis dan terstruktur dalam setiap kesempatan. Kedua, dengan menggunakan *flowchart*, akan lebih mudah mengecek dan menemukan bagian-bagian prosedur yang tidak valid dan bertele-tele. Selain itu, prosedur yang dikembangkan akan lebih mudah dipahami oleh orang lain, sehingga tidak menimbulkan kesalahan interpretasi apabila mau menerapkan prosedur yang dikembangkan (Sutanta, 2004:27).

*Flowchart* dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus. Beberapa simbol yang sering digunakan dalam *flowchart* adalah seperti ditunjukkan dalam gambar sebagai berikut:



**Gambar 2. 12** Beberapa Simbol dalam Bagan Alir.

Keterangan:

1. Dimulai atau diakhiri suatu proses
2. Persiapan proses/pemberian nilai awal
3. Pembacaan atau pencetakan data
4. Proses
5. Proses pengecekan kondisi
6. Proses yang telah didefinisikan
7. Penghubung pada satu halaman
8. Penghubung pada halaman lain
9. Dokumen
10. Arah aliran proses

Secara umum, proses penyelesaian suatu permasalahan terdiri dari lima langkah utama, yaitu:

1. Dimulainya suatu proses
2. Membaca data masukan
3. Proses penyelesaian permasalahan
4. Mencetak hasil pengolahan/informasi
5. Diakhirinya semua proses pengolahan

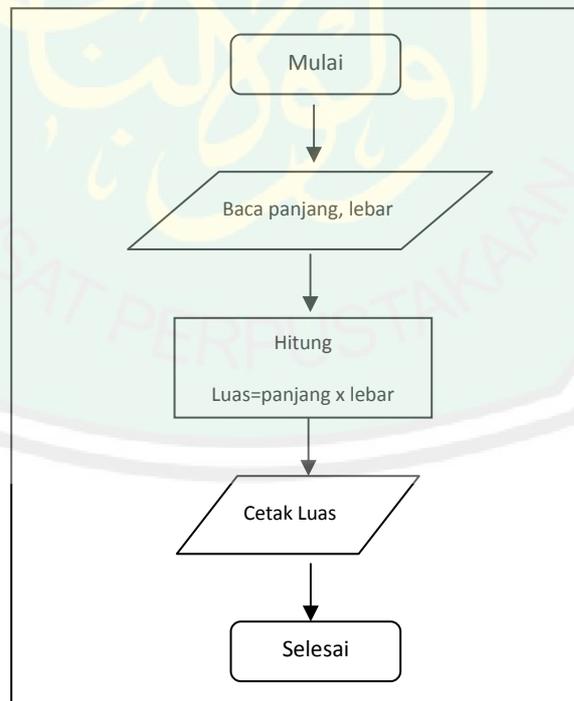
Sebagai contoh, **Gambar 2.13** menunjukkan contoh *flowchart* untuk menyelesaikan permasalahan menghitung luas empat persegipanjang. Langkah-langkah untuk menghitung luas empat persegi panjang adalah dimulai pada titik mulai, kemudian membaca data “panjang” dan “lebar” empat persegi panjang yang akan dihitung. Berdasarkan data masukan tersebut dapat dihitung “luas”

empat persegi panjang dengan cara mengalikan “panjang” dengan “lebar”, hasilnya kemudian dicetak. Proses berakhir pada titik selesai. Proses untuk menghitung luas persegi panjang pada **Gambar 2.13** dapat dituliskan dalam serangkaian langkah, yaitu sebagai berikut:

Panjang, Lebar merupakan variabel untuk menyimpan data masukan.

Luas merupakan variabel untuk menyimpan hasil perhitungan yang diperoleh.

1. Mulai
2. Baca data ( panjang, lebar)
3. Hitung luas (luas = panjang x lebar)
4. Cetak hasil (luas)
5. Selesai



**Gambar 2. 13** Flowchart Menghitung Luas Persegi Panjang

(Sutanta, 2004:27-30).

## 2.8. Kajian Agama untuk Penjadwalan Proyek

Perencanaan dan penjadwalan proyek diperlukan sebelum proses pembangunan suatu proyek dilaksanakan. Hal tersebut karena untuk mendapatkan hasil pembangunan proyek dengan sebaik-baiknya yang sesuai dengan keinginan dan tujuan. Hal ini dijelaskan dalam surat al-Hasyr ayat 18:

..... وَتَنْظُرْ نَفْسٌ مَّا قَدَّمَتْ لِغَدٍ ط

Artinya :

*Dan hendaklah setiap diri memperhatikan apa yang telah diperbuatnya untuk hari esok (akhirat).*

Manajemen proyek digunakan untuk menyelesaikan suatu penjadwalan proyek yang sangat diperlukan untuk mengoptimalkan waktu. Penjadwalan proyek tidak terlepas dari arti penting suatu kata “waktu”. Karena secara tidak langsung maksud dari suatu penjadwalan proyek adalah merencanakan lama waktu yang dibutuhkan untuk menyelesaikan suatu proyek. Kegiatan merencanakan yang diikuti kegiatan menambah dan mengurangi durasi kegiatan sesuai dengan kebutuhan sehingga didapatkan waktu yang optimal. Dijelaskan dalam surat al- Asr ayat 1-2:

وَالْعَصْرِ ﴿١﴾ إِنَّ الْإِنْسَانَ لِفِي خُسْرٍ ﴿٢﴾

Artinya:

*Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian.*

Ayat di atas mengisyaratkan pentingnya memperhatikan waktu (perhitungan) dalam menjalankan suatu rencana. Surat tersebut memberikan gambaran kerugian yang berlimpah bagi mereka yang tidak mempergunakan waktunya dengan baik.

Dalam suatu pekataan sahabat juga dijelaskan:

الْوَقْتُ كَالسَّيْفِ إِنَّمَا تَقْطَعُهُ قَطْعَكَ

Artinya:

*Waktu bagaikan pedang, jika kamu tidak memotongnya ia yang akan memotongmu.*

Melalaikan waktu adalah suatu kebinasaan. Bisa jadi tanpa disadari waktu akan membunuh manusia, jikalau manusia lalai dengan waktu, karena waktu begitu cepat berlalu dan tidak akan pernah kembali, terus berlari tanpa henti, terus berputar tanpa kenal lelah dan tidak menghiraukan apa yang ditinggalkannya. Jadi jika manusia tidak memanfaatkan waktu dengan sebaik baiknya, maka manusia akan binasa sebagaimana binasanya seseorang yang terkena sabetan pedang jika dia tidak segera menghindar dan melawannya maka dia (pedang) akan memotong dan menghancurkannya, karena waktu bagaikan pedang yang membunuh.

Merencanakan penjadwalan suatu proyek merupakan wewenang manusia sebagai bentuk usaha untuk mencapai tujuan akhir sesuai dengan keinginan. Sejauh ini manusia sebagai ciptaan-Nya hanya berwenang merencanakan dan berusaha atas segala hal yang belum terjadi. Hal di atas tidak terlepas dari peran penting manusia sebagai khalifah yang mempunyai peran menjaga, memelihara dan melestarikan segala sesuatu di muka bumi. Dalam hal ini Allah SWT telah berfirman:

وَإِذْ قَالَ رَبُّكَ لِلْمَلَائِكَةِ إِنِّي جَاعِلٌ فِي الْأَرْضِ خَلِيفَةً

Artinya: *Ingatlah ketika Tuhanmu berfirman kepada Para Malaikat: "Sesungguhnya aku hendak menjadikan seorang khalifah di muka bumi" (Al-Baqarah:30).*

Dijelaskan dalam tafsir Ibnu Katsir Ad-Dimasyqi (1999:104) "Sesungguhnya aku hendak menjadikan seorang khalifah di muka bumi. "Yaitu suatu kaum yang akan menggantikan satu kaum sama lain, kurun demi kurundan generasi demi generasi. Mereka yang akan menciptakan suatu kemaslahatan dan bukan mereka yang akan menimbulkan kerusakan di muka bumi. Termasuk bagaimana merencanakan segala sesuatu yang didasarkan pengetahuan tentangnya.

Tahap perencanaan penjadwalan proyek meliputi langkah mempresentasikan kegiatan dalam suatu rangkaian network. Tahap yang mengisyaratkan kepada pelaksana proyek agar sesegera mungkin melaksanakan kegiatan tanpa menunda-nunda sesuai dengan urutan kerja. Dalam surat asy-Syarh ayat 7-8 dijelaskan:

فَإِذَا فَرَغْتَ فَانصَبْ ۖ وَإِلَىٰ رَبِّكَ فَارْغَبْ ۚ

Artinya: *Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan hanya kepada Tuhanmulah hendaknya kamu berharap.*

Quraish Shihab dalam tafsir Al-Misbah (2003:364) menjelaskan maksud dari ayat di atas adalah kesungguhan bekerja disertai dengan harapan serta optimisme akan kehadiran bantuan Ilahi. Maka apabila engkau telah selesai yaitu sedang di dalam keluangan setelah tadinya engkau sibuk maka bekerjalah dengan sungguh-sungguh hingga engkau letih atau hingga tegak dan nyata suatu persoalan baru dan hanya kepada Tuhanmu saja tidak kepada siapapun selain-Nya

dan berkeinginan penuh guna memperoleh bantuan-Nya dalam menghadapi setiap kesulitan serta melakukan suatu aktivitas.



## **BAB III**

### **METODE PENELITIAN**

#### **3.1. Jenis Data**

Data yang digunakan sebagai input program adalah data uraian-uraian pekerjaan ( $X_i$ ), pekerjaan sebelumnya ( $X_{i-1}$ ) serta durasi pekerjaan ( $t$ ). Maka jenis data yang digunakan adalah data kuantitatif, yaitu data berupa angka-angka yang berkaitan dengan permasalahan yang dibahas.

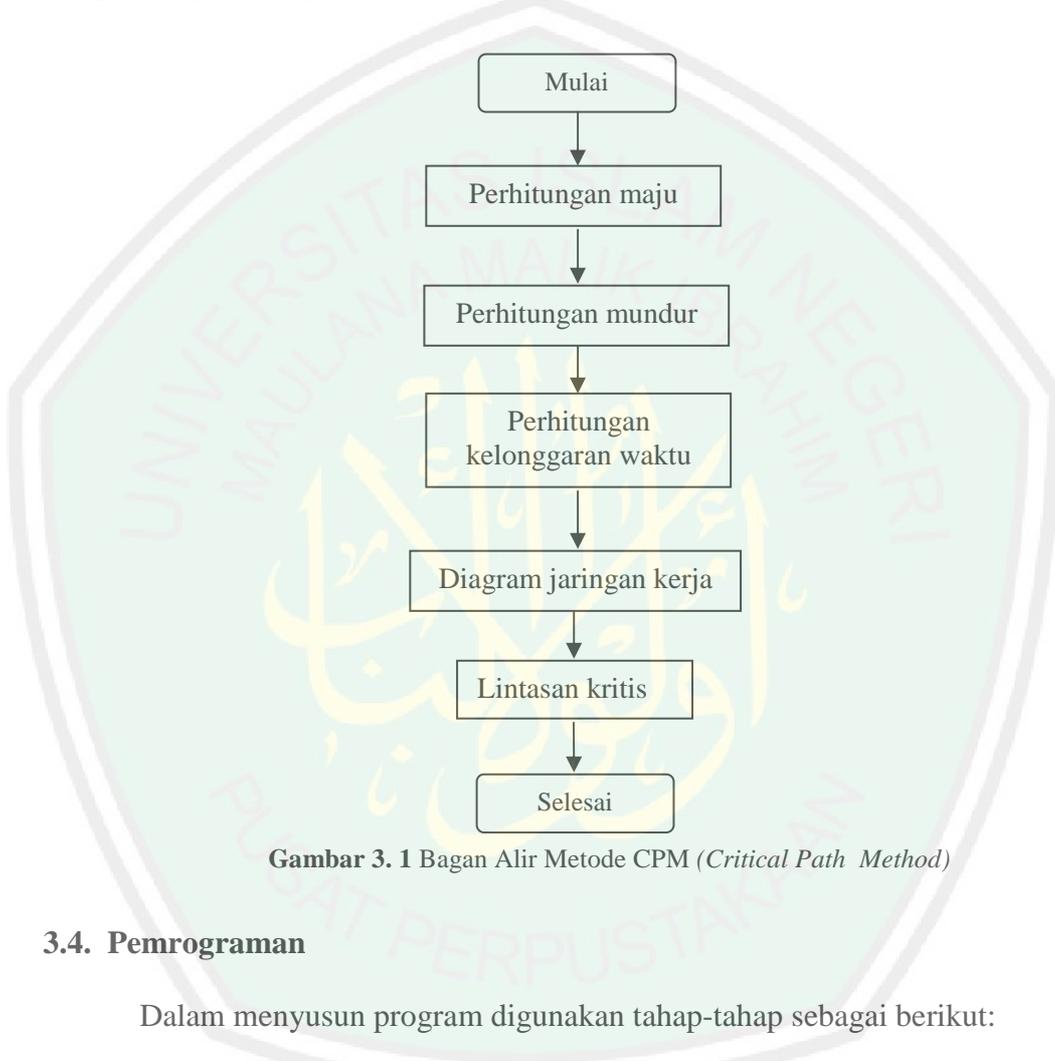
#### **3.2. Sumber Data**

Data yang diperoleh adalah data sekunder, yaitu data yang diperoleh dari CV. Sekar Tanjung selaku sub kontraktor pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield Malang.

### 3.3. Teknis Analisis Data

Dalam pengolahan data, penulis menggunakan metode CPM model AOA.

Adapun langkah-langkahnya dalam bagan alir adalah:



Gambar 3. 1 Bagan Alir Metode CPM (*Critical Path Method*)

### 3.4. Pemrograman

Dalam menyusun program digunakan tahap-tahap sebagai berikut:

1. Tahap pemilihan, yaitu memilih metode yang akan digunakan dalam proses pengolahan data. Dalam hal ini metode yang digunakan adalah metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*).
2. Tahap penyusunan Algoritma, yaitu menyusun langkah-langkah kerja atau prosedur pemrograman yang ingin di desain sesuai dengan kebutuhan.

3. Tahap pemrograman, yaitu menerjemahkan langkah-langkah algoritma ke dalam bahasa pemrograman. Bahasa pemrograman yang digunakan adalah Visual Basic 6.0.
4. Tahap Dokumentasi, yaitu menyimpan program dalam sistem komputer dan menjalankan program untuk mendapatkan output sesuai dengan maksud pemrograman.
5. Tahap uji coba program, yaitu menguji coba tingkat kebenaran dan kesalahan serta kelebihan dan kelemahan program yang telah dibuat sehingga memungkinkan untuk dikembangkan pada masa yang akan datang.
6. Tahap interpretasi, yaitu penafsiran terhadap hasil program untuk dipergunakan dalam mengambil keputusan.

## BAB IV

### PEMBAHASAN

#### 4.1. Menentukan Lintasan Kritis

Dalam menentukan lintasan kritis ada 3 tahapan yang harus dilakukan, yaitu perhitungan maju, perhitungan mundur, perhitungan kelonggaran waktu (*float*):

##### 4.1.1. Perhitungan Maju

Algoritma Perhitungan Maju:

1. Data awal:  
Pengkodean kegiatan-kegiatan
2. Menentukan saat tercepat terjadinya *event* atau kejadian  $TE(i)$
3. Menentukan saat tercepat memulai kegiatan  $ES(i,j)$
4. Menentukan saat paling cepat untuk menyelesaikan kegiatan  $EF(i,j)$ , sama dengan saat paling cepat untuk memulai kegiatan  $ES(i,j)$  ditambah dengan durasi kegiatan tersebut  $t(i,j)$
5. Jika terdapat *merge event* (kejadian yang menggabungkan beberapa kegiatan), maka untuk mengetahui  $TE(j)$  dicari waktu yang paling maksimal dari  $EF(i_1,j)$ ,  $EF(i_2,j)$ , ...,  $EF(i_n,j)$ .

Contoh:

Berdasarkan data pada **lampiran1** untuk menentukan TE, ES dan EF digunakan persamaan (2.1), (2.2) dan (2.3) sebagai berikut:

$$ES(i,j) = TE(i) = 0$$

$$EF(i,j) = ES(i,j) + t(i,j) = TE(i) + t(i,j)$$

$$TE(j) = \text{maks} \{EF(i_1,j), EF(i_2,j), \dots, EF(i_n,j)\}$$

$$ES(A,B) = TE(A) = 0$$

$$EF(A,B) = TE(A) + t(A,B) = 0 + 1 = 1$$

$$ES(B,C) = TE(B) = 1$$

$$EF(B,C) = TE(B) + t(B,C) = 1 + 6 = 7$$

$$ES(C,D) = TE(C) = 7$$

$$EF(C,D) = TE(C) + t(C,D) = 7 + 2 = 9$$

$$ES(D,E) = TE(D) = 9$$

$$EF(D,E) = TE(D) + t(D,E) = 9 + 6 = 15$$

$$ES(E,F) = TE(E) = 15$$

$$EF(E,F) = TE(E) + t(E,F) = 15 + 1 = 16$$

$$ES(F,G) = TE(F) = 16$$

$$EF(E,G) = TE(E) + t(E,G) = 15 + 4 = 19$$

$$EF(F,G) = TE(F) + t(F,G) = 16 + 0 = 16$$

$$ES(G,J) = TE(G) = \text{maks} \{EF(E,G); EF(F,G)\}$$

$$= \text{maks} \{19, 16\}$$

$$= 19$$

$$EF(E,H) = TE(E) + t(E,H) = 15 + 4 = 19$$

$$ES(H,M) = TE (H) = 19$$

$$EF (G, I) = TE(G) + t (G, I) = 19 + 12 = 31$$

$$ES(I,M) = TE (I) = 31$$

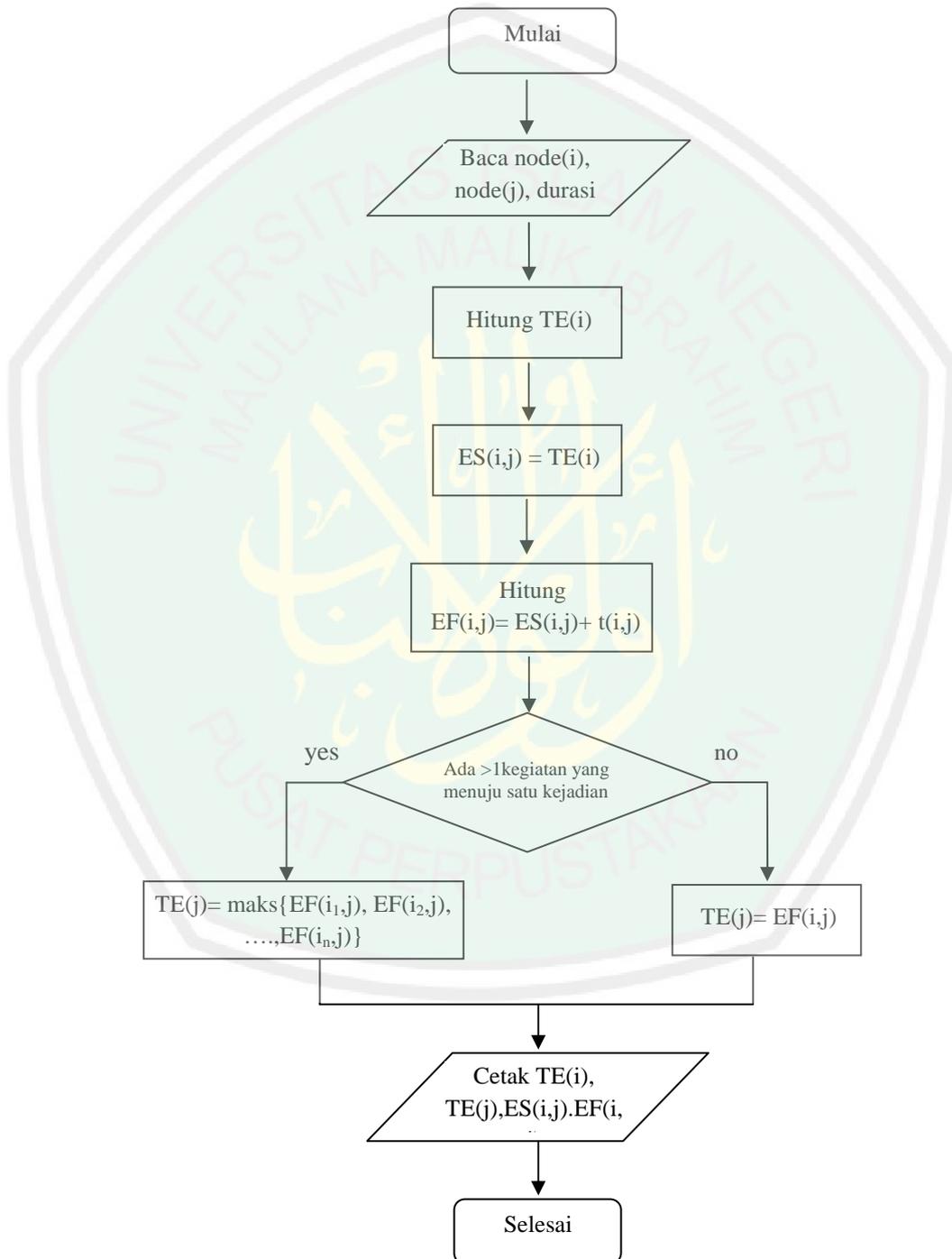
$$EF (G, J) = TE(G) + t (G, J) = 19 + 4 = 23$$

$$ES(J,K) = TE (J) = 23$$

Dan seterusnya sampai seluruh kegiatan dan kejadian selesai.



Langkah-langkah untuk melakukan perhitungan maju dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



Gambar 4. 1 Bagan Alir Perhitungan Maju

Bagan alir di atas menunjukkan langkah-langkah untuk perhitungan maju. Dimulai pada titik mulai, kemudian membaca data “node(i)”, ”node(j)” dan “durasi”. “node(i)” dan ”node(j)” menyatakan kegiatan (i,j), sedangkan durasi menyatakan waktu penyelesaian kegiatan (i,j). Langkah selanjutnya menghitung saat tercepat terjadinya kejadian  $TE(i)$ . Setelah itu menghitung saat tercepat dimulainya kegiatan  $ES(i,j)$  dimana  $ES(i,j) = TE(i)$ . Proses selanjutnya menghitung saat tercepat diselesaikannya kegiatan  $EF(i,j)$  yang diperoleh dari penjumlahan antara saat tercepat dimulainya kegiatan  $ES(i,j)$  dan durasi kegiatan  $t(i,j)$ . Jika terdapat suatu kondisi dimana terdapat beberapa kegiatan (i,j) yang menuju ke satu kejadian yang sama maka saat tercepat terjadinya kejadian  $TE(j)$  diperoleh dari waktu maksimal dari saat tercepat diselesaikannya kegiatan 1  $EF(i1,j)$ , saat tercepat diselesaikannya kegiatan 2  $EF(i2,j)$ , sampai saat tercepat diselesaikannya kegiatan n  $EF(in,j)$ . Jika suatu kondisi tersebut tidak terpenuhi maka saat tercepat terjadinya kejadian  $TE(j)$  sama dengan saat tercepat diselesaikannya kegiatan  $EF(i,j)$ . Langkah berikutnya mencetak semua hasil perhitungan. Selanjutnya proses berakhir pada titik selesai.

**a. Saat tercepat dimulainya kegiatan  $ES(i,j)$**

Algoritma saat tercepat dimulainya kegiatan  $ES(i,j)$ :

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk  $i=1$
4. Membuat variabel D sebagai fungsi untuk mencari node(i) di kolom 1 mulai dari  $i-1$

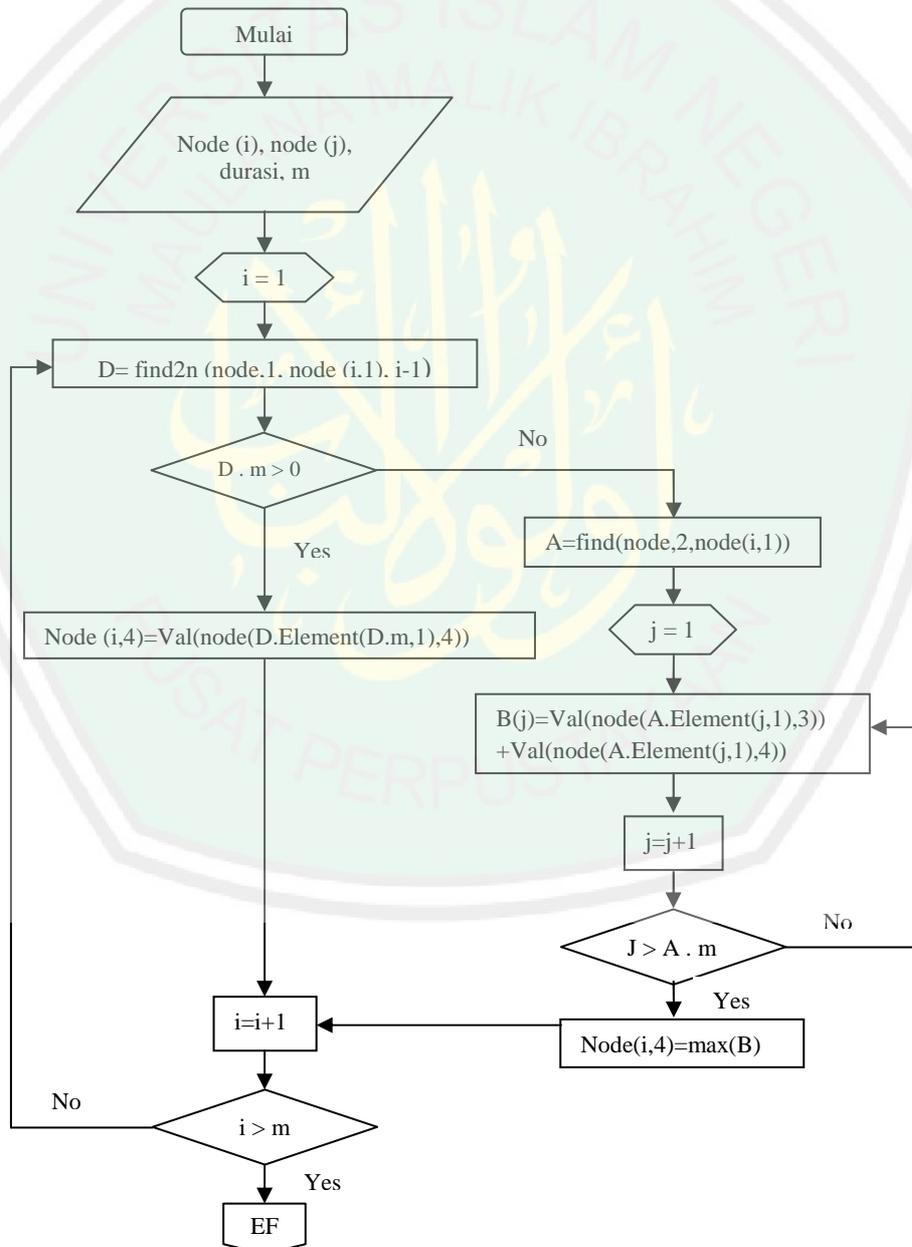
5. Menentukan nilai di kolom 4 jika  $i$  ditemukan
6. Membuat variabel  $A$  sebagai fungsi untuk mencari  $\text{node}(i)$  di kolom 2 jika  $i$  tidak ditemukan
7. Pemberian nilai awal untuk  $j=1$
8. Menjumlahkan nilai di kolom 3 dan di kolom 4
9. Mencari nilai maksimal jika ada  $j$  yang sama
10. Mencari  $j$  selanjutnya sampai banyaknya baris
11. Mencari  $i$  selanjutnya sampai banyaknya baris
12. Menghitung saat tercepat diselesaikannya kegiatan  $EF(i,j)$

Banyak kerja/operasi (penjumlahan, perbandingan) yang dilakukan oleh algoritma saat tercepat dimulainya kegiatan  $ES(i,j)$  untuk perulangan  $i$  (1, 2, ...,  $n$ ) adalah:

- a. Dalam variabel  $D$  untuk mencari  $\text{node}(i)$  di kolom 1 mulai dari  $i=1$  sampai  $i=n-1$ ,  $D$  melakukan kerja sebanyak  $(n-1)n$  kali. Diperoleh dari perkalian  $(n-1)$  sebagai batas atas  $i$  dalam variabel  $D$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .
- b. Dalam variabel  $A$  untuk mencari  $\text{node}(i)$  di kolom 2 mulai dari  $i=1$  sampai  $i=n$ ,  $A$  melakukan kerja sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas  $i$  dalam variabel  $A$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .
- c. Perulangan  $j=1$  sampai  $j=n$ ,  $j$  melakukan perulangan sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas perulangan  $j$  dalam variabel  $D$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .

Jadi, di dalam perhitungan  $ES(i,j)$  terdapat kerja algoritma sebanyak  $(3n^2 - n)$  kali. Diperoleh dari penjumlahan banyak kerja dari variabel D, variabel A dan perulangan j.

Algoritma untuk menghitung saat tercepat dimulainya kegiatan  $ES(i,j)$  dapat digambarkan dalam bentuk bagan alir sebagai berikut:



Gambar 4. 2 Bagan Alir Saat Tercepat Dimulainya Kegiatan ES

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung saat tercepat dimulainya kegiatan  $ES(i,j)$ . Dimulai pada titik mulai, kemudian membaca data “node(i)”, “node(j)”, “durasi” dan “m”. “node(i)” dan “node(j)” menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan “m” menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk  $i=1$ . Selanjutnya didefinisikan variabel D sebagai fungsi untuk mencari node(i) di kolom 1. Jika D ada maka di kolom 4 diisi dengan hasil yang sama dengan hasil di baris sebelumnya. Jika D tidak ada maka dibuat suatu fungsi lagi yaitu fungsi find yang didefinisikan sebagai variabel A, maksudnya untuk mencari node(i) di kolom 2. Diberikan nilai awal  $j=1$  dan didefinisikan variabel B sebagai penjumlahan hasil di kolom 3 dan hasil di kolom 4. Setelah itu dicari j berikutnya, jika dalam j berikutnya tadi terdapat hasil yang sama dengan j sebelumnya maka dalam kolom 4 dipilih hasil yang maksimal. Jika tidak terdapat j yang sama maka proses di ulangi sampai banyaknya baris data. Langkah selanjutnya dicari i berikutnya sampai banyaknya baris data. Jika sudah selesai sampai banyaknya baris data maka saat tercepat diselesaikannya kegiatan  $EF(i,j)$  dapat diproses.

**b. Saat tercepat diselesaikannya kegiatan  $EF(i,j)$**

Algoritma saat tercepat diselesaikannya kegiatan  $ES(i,j)$ :

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk  $i=1$
4. Membuat variabel A sebagai fungsi untuk mencari node(i) di kolom 2

5. Pemberian nilai awal untuk  $j=1$
6. Menjumlahkan nilai di kolom 3 dan di kolom 4
7. Mencari nilai maksimal jika ada  $j$  yang sama
8. Mencari  $j$  selanjutnya sampai banyaknya baris
9. Mencari  $i$  selanjutnya sampai banyaknya baris
10. Menghitung saat paling lambat diselesaikannya kegiatan  $LF(i,j)$

Banyak kerja/operasi (penjumlahan, perbandingan) yang dilakukan oleh algoritma saat tercepat diselesaikannya kegiatan  $EF(i,j)$  untuk perulangan  $i$  (1, 2, ...,  $n$ ) adalah:

- a. Dalam variabel  $A$  untuk mencari  $node(j)$  di kolom 2 mulai dari  $i=1$  sampai  $i=n$ ,  $A$  melakukan kerja sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas  $i$  dalam variabel  $A$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .
- b. Perulangan  $j=1$  sampai  $j=n$ ,  $j$  melakukan perulangan sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas perulangan  $j$  dalam variabel  $D$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .

Jadi, di dalam perhitungan  $EF(i,j)$  terdapat kerja algoritma sebanyak  $(2n^2)$  kali. Diperoleh dari penjumlahan banyak kerja dari variabel  $A$  dan perulangan  $j$ .

Total keseluruhan banyak kerja yang dilakukan oleh algoritma dalam perhitungan maju adalah sebanyak  $(5n^2 - n)$ . Diperoleh dari penjumlahan banyak kerja dari saat tercepat dimulainya kegiatan  $ES(i,j)$  dan saat tercepat diselesaikannya kegiatan  $EF(i,j)$ . Jadi, dengan menggunakan data pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield Malang dengan banyak

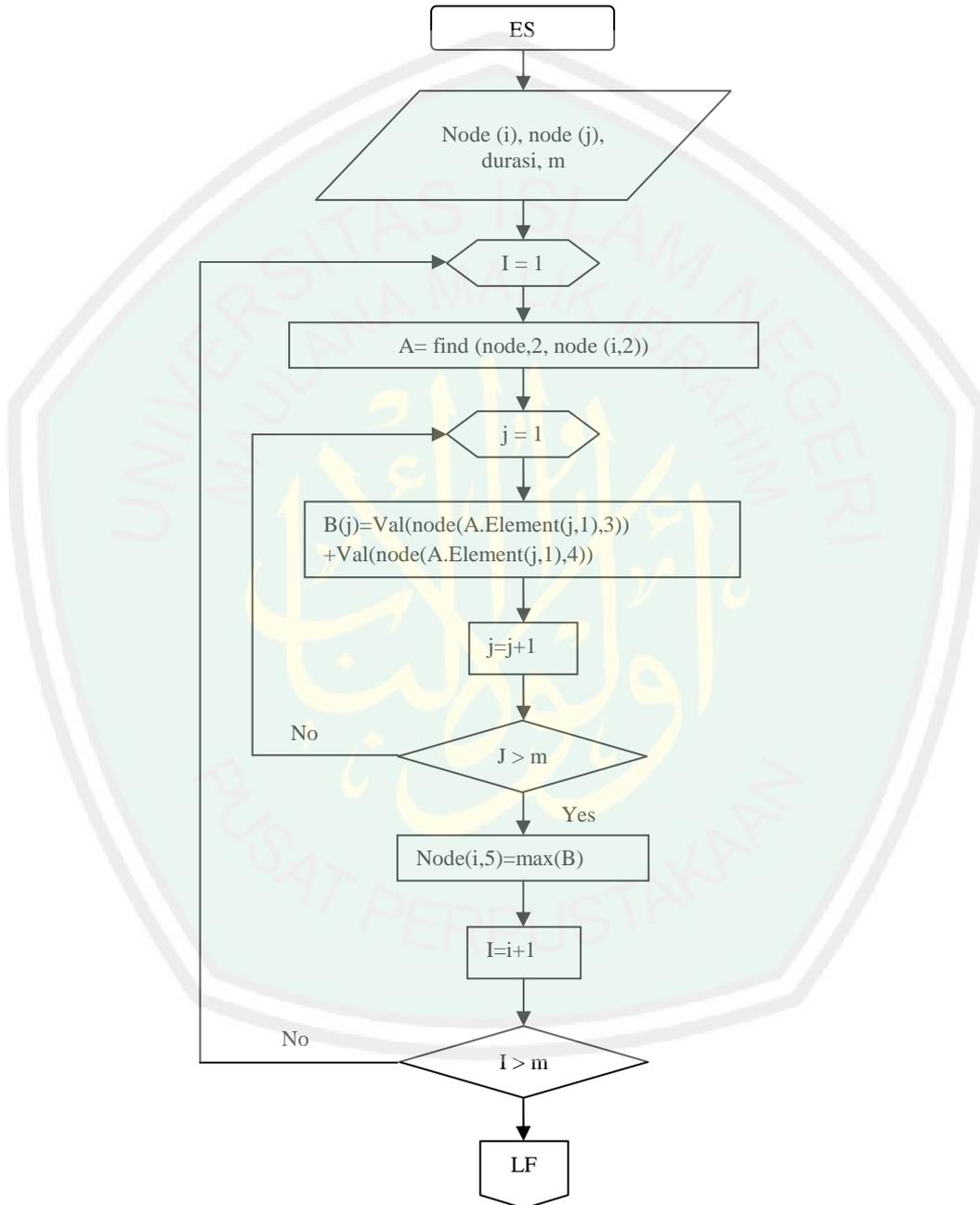
data/ banyak  $n = 80$ , maka banyak kerja algoritma dalam perhitungan maju sebanyak 31.920 kali.

Dengan memakai program penjadwalan ini, kecepatan program dalam menghitung data pembangunan dengan kecepatan prosessor komputer 2.30 G.Hz dan dengan banyak data/ banyak  $n = 80$ , melakukan perhitungan maju dibutuhkan waktu 0,000014 detik. Waktu tersebut diperoleh dari (kecepatan prosessor/detik) dibagi dengan banyak kerja algoritma.



Algoritma untuk menghitung saat tercepat diselesaikannya kegiatan EF(i,j)

dapat digambarkan dalam bentuk bagan alir sebagai berikut:



Gambar 4.3 Bagan Alir Saat Tercepat Diselesaikannya Kegiatan EF

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung saat tercepat diselesaikannya kegiatan  $EF(i,j)$ . Dimulai pada titik mulai, kemudian membaca data "node(i)", "node(j)", "durasi" dan "m". "node(i)" dan "node(j)" menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan "m" menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk  $i=1$ . Selanjutnya didefinisikan variabel A sebagai fungsi untuk mencari node(j) di kolom 2. Diberikan nilai awal  $j=1$  dan didefinisikan variabel B sebagai penjumlahan hasil di kolom 3 dan hasil di kolom 4. Setelah itu dicari j berikutnya, jika dalam j berikutnya tadi terdapat hasil yang sama dengan j sebelumnya maka dalam kolom 5 dipilih hasil yang maksimal. Jika tidak terdapat j yang sama maka proses di ulangi sampai banyaknya baris data. Langkah selanjutnya dicari i berikutnya sampai banyaknya baris data. Jika sudah selesai sampai banyaknya baris data maka saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  dapat diproses.

Kode program untuk saat tercepat dimulainya kegiatan  $ES(i,j)$  dan saat tercepat diselesaikannya kegiatan  $EF(i,j)$  dapat dilihat pada **lampiran 2**.

#### 4.1.2 Perhitungan Mundur:

Algoritma perhitungan mundur:

1. Data awal.
2. Menentukan saat paling lambat terjadinya *event* atau kejadian  $TL(j)$ , pada *terminal event* berlaku  $TL=TE$ .
3. Menentukan saat paling lambat diselesaikannya kegiatan  $LF(i,j)$ .

4. Menentukan saat paling lambat untuk memulai suatu kegiatan  $LS(i,j)$ , sama dengan saat paling lambat untuk menyelesaikan kegiatan itu  $LF(i,j)$  dikurangi dengan durasi kegiatan tersebut  $t(i,j)$ .
5. Jika terdapat *burst event* (kejadian yang mengeluarkan beberapa kegiatan), maka untuk mengetahui  $TL(i)$  dicari waktu yang paling minimal dari  $LS(i,j_1), LS(i,j_2), \dots, LS(i,j_n)$ .

Contoh:

Dengan menggunakan data yang sama pada **Lampiran 1**:

Untuk menentukan  $TL$ ,  $LS$  dan  $LF$  digunakan persamaan (2.4), (2.5), (2.6) dan (2.7) sebagai berikut:

$$LS = LF - t$$

$$LF(i,j) = TL \text{ dimana } TL = TE$$

$$LS(i,j) = TL(j) - t(i,j)$$

$$TL(i) = \min \{ LS(i,j_1), LS(i,j_2), \dots, LS(i,j_n) \}.$$

$$LF(ZZI,ZZJ) = TL (ZZJ) = 74$$

$$LS( ZZI, ZZJ) = TL (ZZJ) - t (ZZI, ZZJ) = 74 - 4 = 70$$

$$LF(ZZH, ZZI) = TL (ZZI) = 70$$

$$LS (ZZH, ZZI) = TL (ZZI) - t (ZZH, ZZI) = 70 - 2 = 68$$

$$LF(ZZG,ZZH) = TL (ZZH) = 68$$

$$LS (ZZG,ZZH) = TL (ZZH) - t (ZZG,ZZH) = 68 - 0 = 68$$

$$LF(ZZF,ZZG) = TL (ZZG) = 68$$

$$LS (ZZF,ZZG) = TL (ZZG) - t (ZZF, ZZG) = 68 - 0 = 68$$

$$LF(ZZE,ZZF) = TL (ZZF) = 68$$

$$LS(ZZE, ZZG) = TL(ZZG) - t(ZZE, ZZG) = 68 - 6 = 62$$

$$LS(ZZE, ZZF) = TL(ZZF) - t(ZZE, ZZF) = 68 - 4 = 64$$

$$LF(ZZD, ZZE) = TL(ZZE) = \min \{ LS(ZZE, ZZG); LS(ZZE, ZZF) \} \\ = \min \{ 62, 64 \} = 62$$

$$LS(ZZD, ZZE) = TL(ZZE) - t(ZZD, ZZE) = 62 - 4 = 58$$

$$LF(ZZC, ZZD) = TL(ZZD) = 58$$

$$LS(ZZC, ZZD) = TL(ZZD) - t(ZZC, ZZD) = 58 - 0 = 58$$

$$LF(ZZB, ZZC) = TL(ZZC) = 58$$

$$LS(ZZB, ZZD) = TL(ZZD) - t(ZZB, ZZD) = 58 - 4 = 54$$

$$LS(ZZB, ZZC) = TL(ZZC) - t(ZZB, ZZC) = 58 - 4 = 54$$

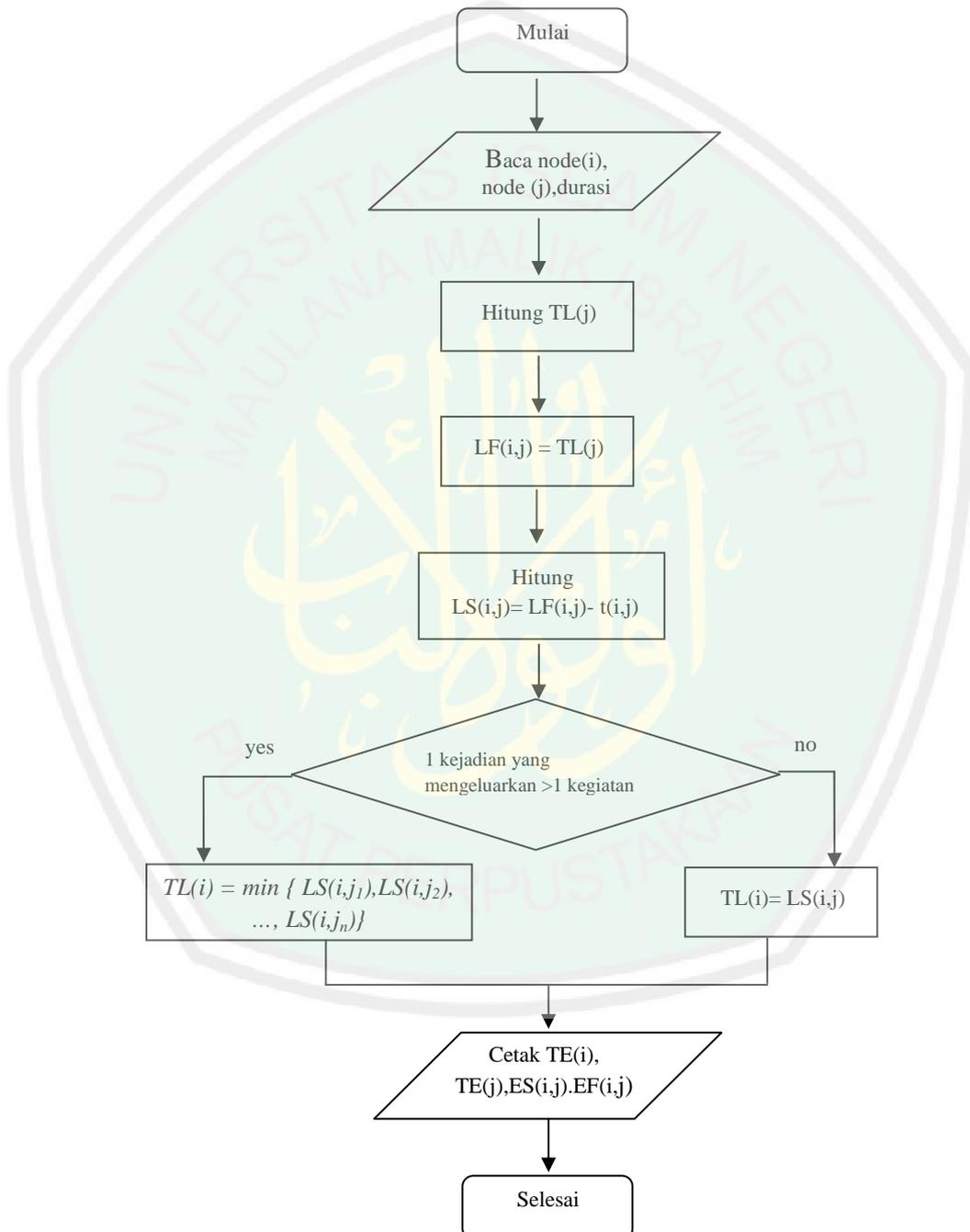
$$LF(ZZA, ZZB) = TL(ZZB) = \min \{ LS(ZZB, ZZD); LS(ZZB, ZZC) \} \\ = \min \{ 54, 54 \} \\ = 54$$

$$LS(ZZA, ZZB) = TL(ZZB) - t(ZZA, ZZB) = 54 - 0 = 54$$

$$LF(X, ZZA) = TL(ZZA) = 54$$

Dan seterusnya sampai seluruh kegiatan dan kejadian selesai dihitung atau sampai hari ke 0.

Langkah-langkah untuk melakukan perhitungan mundur dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



Gambar 4.4 Bagan Alir Perhitungan Mundur

Bagan alir di atas menunjukkan langkah-langkah untuk perhitungan mundur. Dimulai pada titik mulai, kemudian membaca data “node(i)”, “node(j)” dan “durasi”. “node(i)” dan “node(j)” menyatakan kegiatan (i,j), sedangkan durasi menyatakan waktu penyelesaian kegiatan (i,j). Langkah selanjutnya menghitung saat paling lambat terjadinya kejadian  $TL(j)$ . Setelah itu menghitung saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  dimana  $LF(i,j) = TL(j)$ . Proses selanjutnya menghitung saat paling lambat dimulainya kegiatan  $LS(i,j)$  yang diperoleh dari pengurangan antara saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  dengan durasi kegiatan  $t(i,j)$ . Jika terdapat suatu kondisi dimana terdapat satu kejadian yang mengeluarkan beberapa kegiatan (i,j) maka saat paling lambat terjadinya kejadian  $TL(j)$  diperoleh dari waktu minimal dari saat paling lambat dimulainya kegiatan 1  $LS(i1,j)$ , saat paling lambat dimulainya kegiatan 2  $LS(i2,j)$ , sampai saat paling lambat dimulainya kegiatan n  $LS(in,j)$ . Jika suatu kondisi tersebut tidak terpenuhi maka saat paling lambat terjadinya kejadian  $TL(j)$  sama dengan saat paling lambat dimulainya kegiatan  $LS(i,j)$ . Langkah berikutnya mencetak semua hasil perhitungan. Selanjutnya proses berakhir pada titik selesai.

**a. Saat paling lambat diselesaikannya kegiatan  $LF(i,j)$**

Algoritma saat paling lambat diselesaikannya kegiatan  $LF(i,j)$ :

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk  $i=n$ , perhitungan dilakukan mulai dari n menuju 1
4. Membuat variabel A sebagai fungsi untuk mencari node(j) di kolom 1
5. Pemberian nilai awal untuk  $j=1$  jika kondisi variabel A terpenuhi

6. Mengurangkan nilai di kolom 6 dengan nilai di kolom 3 dalam setiap baris yang dicari
7. Mencari nilai minimal jika ada  $j$  yang sama
8. Mencari  $j$  selanjutnya sampai banyaknya baris
9. Mengisikan nilai di kolom 6 sama dengan nilai di kolom 5 dalam setiap baris yang dicari jika kondisi dalam variabel  $A$  tidak terpenuhi
10. Mencari  $i$  selanjutnya sampai banyaknya baris
11. Menghitung saat paling lambat dimulainya kegiatan  $LS(i,j)$

Banyak kerja/operasi (penjumlahan, perbandingan) yang dilakukan oleh algoritma saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  untuk perulangan  $i$  (1, 2, ...,  $n$ ) adalah:

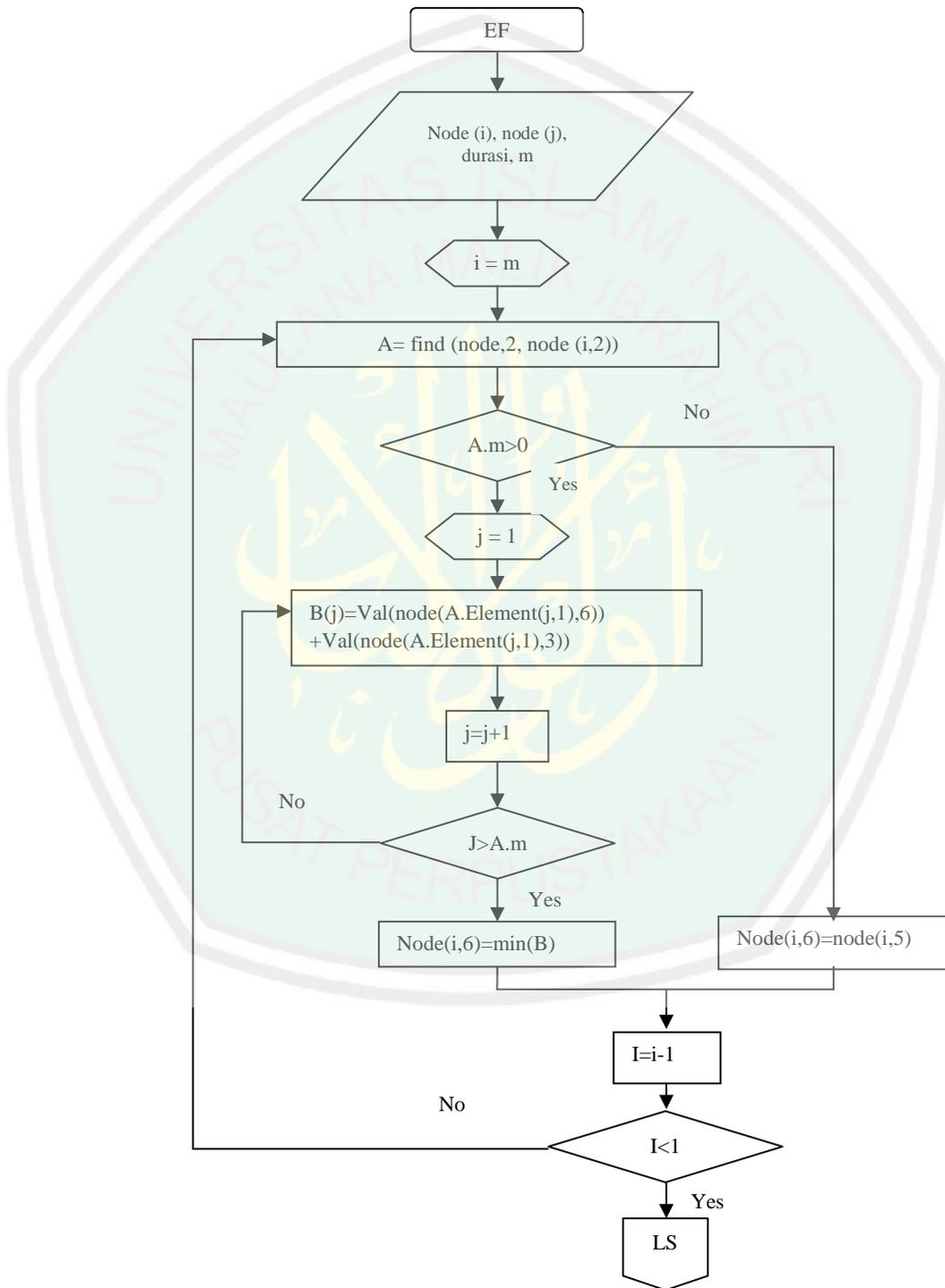
- c. Dalam variabel  $A$  untuk mencari  $node(j)$  di kolom 1 mulai dari  $i=n$  sampai  $i=1$ ,  $A$  melakukan kerja sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas  $i$  dalam variabel  $A$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .
- d. Perulangan  $j=1$  sampai  $j=n$ ,  $j$  melakukan perulangan sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas perulangan  $j$  dalam variabel  $D$  dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .

Jadi, di dalam perhitungan  $LF(i,j)$  terdapat kerja algoritma sebanyak  $(2n^2)$  kali.

Diperoleh dari penjumlahan banyak kerja dari variabel  $A$  dan perulangan  $j$ .

Algoritma untuk menghitung saat paling lambat diselesaikannya kegiatan

LF(i,j) dapat digambarkan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.5** Bagan Alir Saat Paling Lambat Diselesaikannya Kegiatan LF

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung saat paling lambat diselesaikannya kegiatan  $LF(i,j)$ . Dimulai pada titik saat tercepat diselesaikannya kegiatan  $EF(i,j)$ , kemudian membaca data "node(i)", "node(j)", "durasi" dan "m". "node(i)" dan "node(j)" menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan "m" menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk  $i=m$ , maksudnya perhitungan dilakukan dari baris data terakhir menuju baris data awal.. Selanjutnya didefinisikan variabel A sebagai fungsi untuk mencari node(j) di kolom 2. Jika A ada maka dicari j dengan pemberian nilai awal 1 untuk mencari variabel B dimana B adalah penjumlahan hasil di kolom 6 dan hasil di kolom 3. Setelah itu dicari j berikutnya, jika dalam j berikutnya tadi terdapat hasil yang sama dengan j sebelumnya maka dalam kolom 6 dipilih hasil yang minimal. Jika tidak terdapat j yang sama maka proses di ulangi sampai banyaknya baris data. Selanjutnya jika kondisi A tidak ada maka di kolom 6 di isi hasil yang sama dengan hasil kolom 5. Langkah selanjutnya dicari i berikutnya sampai banyaknya baris data. Jika sudah selesai sampai banyaknya baris data maka saat paling lambat dimulainya kegiatan LS (i,j) dapat diproses.

**b. Saat paling lambat dimulainya kegiatan LS(i,j)**

Algoritma saat paling lambat dimulainya kegiatan LS(i,j):

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk  $i=n$ , perhitungan dilakukan mulai dari  $n$  menuju 1
4. Membuat variabel A sebagai fungsi untuk mencari node(i) di kolom 1
5. Pemberian nilai awal untuk  $j=1$
6. Mengurangkan nilai di kolom 6 dengan nilai di kolom 3 dalam setiap baris yang dicari
7. Mengisikan hasil di kolom 7 dengan mencari nilai minimal jika ada  $j$  yang sama
8. Mencari  $i$  selanjutnya sampai banyaknya baris
9. Menghitung *total float* (TF)

Banyak kerja yang dilakukan oleh algoritma saat paling lambat dimulainya kegiatan LS(i,j) untuk perulangan  $i$  (1, 2, ..., n) adalah:

- e. Dalam variabel A untuk mencari node(i) di kolom 1 mulai dari  $i=1$  sampai  $i=n$ , A melakukan kerja sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas  $i$  dalam variabel A dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .
- f. Perulangan  $j=1$  sampai  $j=n$ ,  $j$  melakukan perulangan sebanyak  $(n^2)$  kali. Diperoleh dari perkalian  $n$  sebagai batas atas perulangan  $j$  dalam variabel D dan  $n$  sebagai batas atas  $i$  dalam perulangan  $i$ .

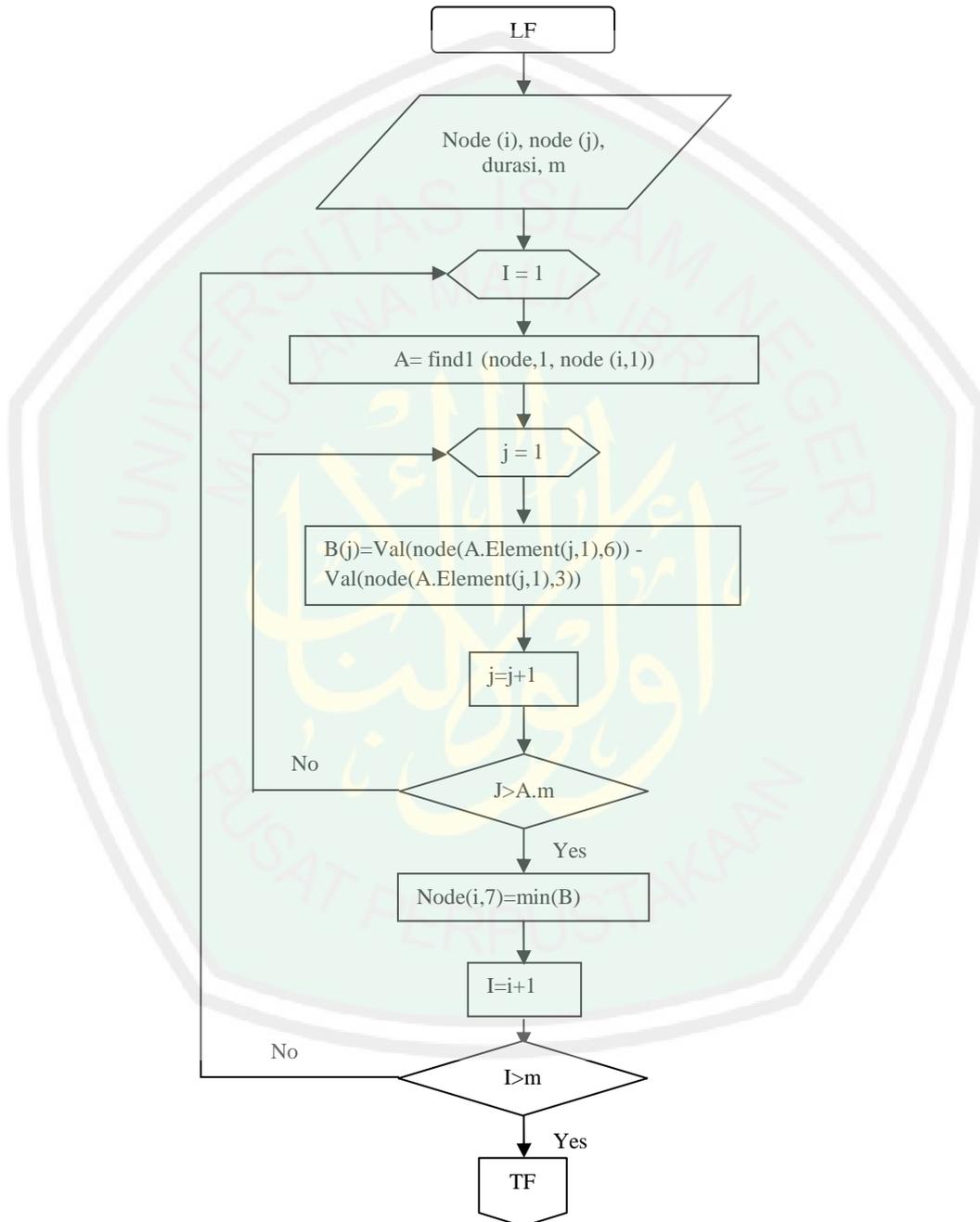
Jadi, di dalam perhitungan  $LS(i,j)$  terdapat kerja algoritma sebanyak  $(2n^2)$  kali. Diperoleh dari penjumlahan banyak kerja dari variabel A dan perulangan j.

Total keseluruhan banyak kerja yang dilakukan oleh algoritma dalam perhitungan mundur adalah sebanyak  $(4n^2)$ . Diperoleh dari penjumlahan banyak kerja dari saat paling lambat dimulainya kegiatan  $LS(i,j)$  dan saat paling lambat diselesaikannya kegiatan  $LF(i,j)$ . Jadi, dengan menggunakan data pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield Malang dengan banyak data/ banyak  $n = 80$ , maka banyak kerja algoritma dalam perhitungan mundur sebanyak 25.600 kali.

Dengan memakai program penjadwalan ini, kecepatan program dalam menghitung data pembangunan dengan kecepatan prosessor komputer 2.30 G.Hz dan dengan banyak data/ banyak  $n = 80$ , melakukan perhitungan mundur dibutuhkan waktu 0,000011 detik. Waktu tersebut diperoleh dari (kecepatan prosessor/detik) dibagi dengan banyak kerja algoritma.

Algoritma untuk menghitung saat paling lambat diselesaikannya kegiatan

$LF(i,j)$  dapat digambarkan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.6** Bagan Alir Saat Paling Lambat Dimulainya Kegiatan LS

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung saat paling lambat dimulainya kegiatan  $LS(i,j)$ . Dimulai pada titik saat paling lambat diselesaikannya kegiatan  $LF(i,j)$ , kemudian membaca data “node(i)”, “node(j)”, “durasi” dan “m”. “node(i)” dan “node(j)” menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan “m” menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk  $i=1$ . Selanjutnya didefinisikan variabel A sebagai fungsi untuk mencari node(i) di kolom 1. Jika A ada maka dicari j dengan pemberian nilai awal 1 untuk mencari variabel B dimana B adalah pengurangan hasil di kolom 6 dan hasil di kolom 3. Setelah itu dicari j berikutnya, jika dalam j berikutnya tadi terdapat hasil yang sama dengan j sebelumnya maka dalam kolom 7 dipilih hasil yang minimal. Jika tidak terdapat j yang sama maka proses di ulangi sampai banyaknya baris data. Langkah selanjutnya dicari i berikutnya sampai banyaknya baris data. Jika sudah selesai sampai banyaknya baris data maka perhitungan kelonggaran waktu *total float* (TF) dapat diproses.

Kode program untuk saat paling lambat dimulainya kegiatan  $LS(i,j)$  dan saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  dapat dilihat pada **lampiran 2**.

#### 4.1.3 Perhitungan Kelonggaran Waktu (*Float*)

*Float* terbagi atas dua jenis, yaitu *total float* dan *free float*.

Algoritma perhitungan *total float*:

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk  $i=1$

4. Mengisikan hasil di kolom 8 dengan mengurangi nilai di kolom 6 dengan nilai di kolom 3 lalu menambahkan nilai di kolom 4
5. Mencari  $i$  berikutnya sampai banyaknya baris
6. Menghitung *free float* (FF)

Algoritma perhitungan *free float*:

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk  $i=1$
4. Mengisikan hasil di kolom 9 dengan mengurangi nilai di kolom 5 dengan nilai di kolom 3 lalu menambahkan nilai di kolom 4
5. Mencari  $i$  berikutnya sampai banyaknya baris
6. Menentukan lintasan

Banyak kerja/operasi (penjumlahan, perbandingan) yang dilakukan oleh algoritma *total float* dan *free float* adalah masing-masing sebanyak  $n$  kali karena hanya terdapat 1 perulangan, yaitu perulangan  $i$  (1, 2, ...,  $n$ ). Jadi total keseluruhan banyak kerja yang dilakukan dalam perhitungan kelonggaran waktu adalah  $2n$ . Diperoleh dari penjumlahan banyak kerja dari saat *total float* dan *free float*. Dengan menggunakan data pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield Malang dengan banyak data/ banyak  $n = 80$ , maka banyak kerja algoritma dalam perhitungan kelonggaran waktu sebanyak 160 kali.

Dengan memakai program penjadwalan ini, kecepatan program dalam menghitung data pembangunan dengan kecepatan prosesor komputer 2.30 G.Hz dan dengan banyak data/ banyak  $n = 80$ , melakukan perhitungan kelonggaran

waktu dibutuhkan waktu 0,000070 detik. Waktu tersebut diperoleh dari (kecepatan prosessor/detik) dibagi dengan banyak kerja algoritma.

Contoh:

Dengan menggunakan data yang sama pada **Lampiran 1**:

Untuk menentukan FF dan TF digunakan persamaan (2.8) dan (2.9) sebagai berikut:

$$\text{Free Float} \Rightarrow FF(i,j) = TE(j) - TE(i) - t(i,j)$$

$$\text{Total Float} \Rightarrow TF(i,j) = TL(j) - TE(i) - t(i,j)$$

$$\text{Kegiatan X1 : } FF(A,B) = TE(B) - TE(A) - t(A,B) = 1 - 0 - 1 = 0$$

$$TF(A,B) = TL(B) - TE(A) - t(A,B) = 1 - 0 - 1 = 0$$

$$\text{Kegiatan X2 : } FF(B,C) = TE(C) - TE(B) - t(B,C) = 7 - 1 - 6 = 0$$

$$TF(B,C) = TL(C) - TE(B) - t(B,C) = 7 - 1 - 6 = 0$$

$$\text{Kegiatan X3 : } FF(C,D) = TE(D) - TE(C) - t(C,D) = 9 - 7 - 2 = 0$$

$$TF(C,D) = TL(D) - TE(C) - t(C,D) = 9 - 7 - 2 = 0$$

$$\text{Kegiatan X5 : } FF(D,E) = TE(E) - TE(D) - t(D,E) = 15 - 9 - 6 = 0$$

$$TF(D,E) = TL(E) - TE(D) - t(D,E) = 15 - 9 - 6 = 0$$

$$\text{Kegiatan X6 : } FF(E,F) = TE(F) - TE(E) - t(E,F) = 16 - 15 - 1 = 0$$

$$TF(E,F) = TL(F) - TE(E) - t(E,F) = 19 - 15 - 1 = 3$$

$$\text{Kegiatan X14: } FF(E,G) = TE(G) - TE(E) - t(E,G) = 19 - 15 - 4 = 0$$

$$TF(E,G) = TL(G) - TE(E) - t(E,G) = 19 - 15 - 4 = 0$$

$$\text{Kegiatan D1: } FF(F,G) = TE(G) - TE(F) - t(F,G) = 19 - 16 - 0 = 3$$

$$TF(F,G) = TL(G) - TE(F) - t(F,G) = 19 - 16 - 0 = 3$$

$$\text{Kegiatan X51: } FF(E,H) = TE(H) - TE(E) - t(E,H) = 19 - 15 - 4 = 0$$

$$TF (E,H) = TL (H) - TE (E) - t (E,H) = 31 - 15 - 4 = 12$$

Kegiatan X15:  $FF (G,I) = TE (I) - TE (G) - t (G,I) = 31 - 19 - 12 = 0$

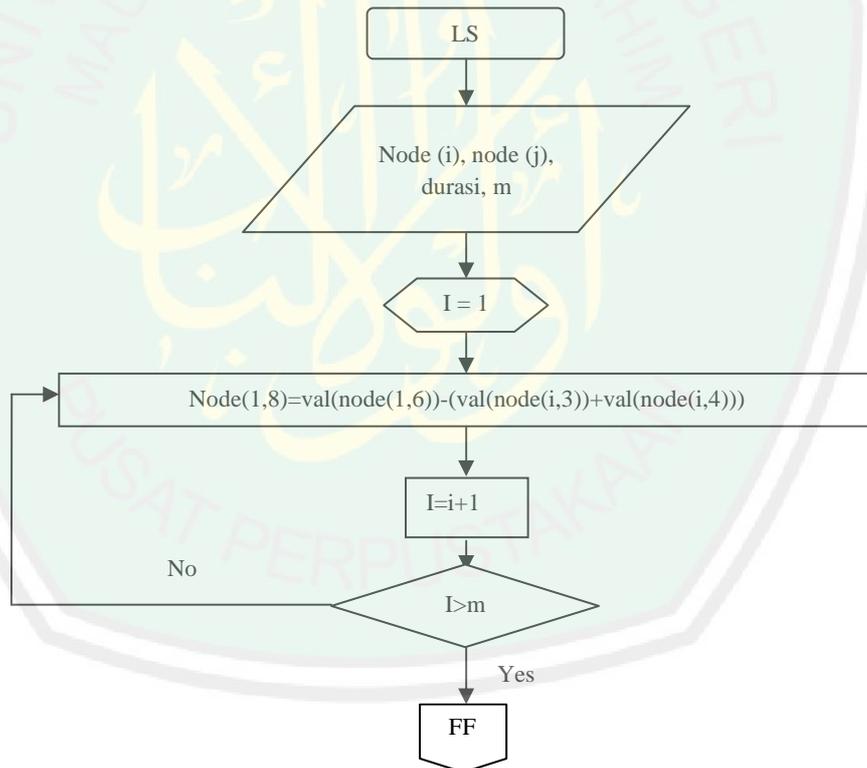
$$TF (G,I) = TL (I) - TE (G) - t (G,I) = 35 - 19 - 12 = 4$$

Kegiatan X7 :  $FF (G,J) = TE (J) - TE (G) - t (G,J) = 23 - 19 - 4 = 0$

$$TF (G,J) = TL (J) - TE (G) - t (G,J) = 23 - 19 - 4 = 0$$

Dan seterusnya sampai seluruh kegiatan selesai.

Algoritma untuk menghitung *total float* (TF) dapat digambarkan dalam bentuk bagan alir sebagai berikut:

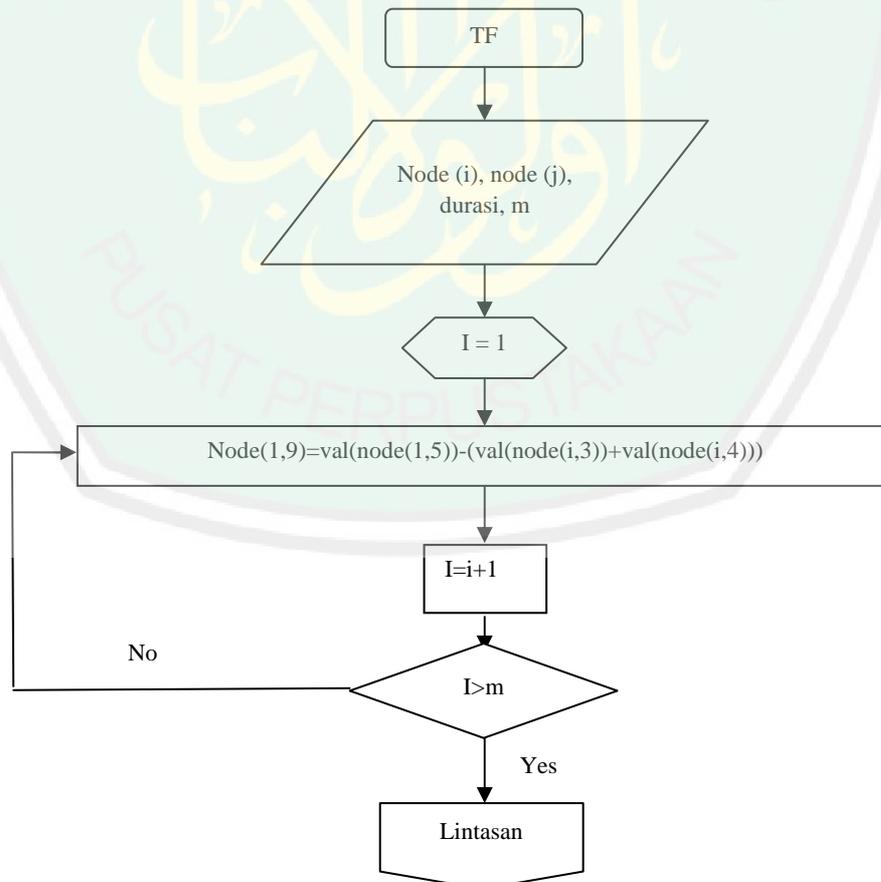


**Gambar 4.7** Bagan Alir *Total Float* (TF)

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung *total float* (TF). Dimulai pada titik saat paling lambat dimulainya kegiatan LS(i,j), kemudian membaca data “node(i)”, ”node(j)”, “durasi” dan “m”. “node(i)” dan

”node(j)” menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan “m” menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk  $i=1$ . Apabila  $i$  ada maka di kolom 8 diisi hasil pengurangan antara hasil di kolom 6 dengan hasil di kolom 3 setelah itu ditambah hasil di kolom 4. Langkah selanjutnya dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai sampai banyaknya baris maka perhitungan *free float* (FF) dapat diproses. Jika belum maka harus dilakukan pencarian  $i$  lagi sampai selesai banyaknya baris.

Algoritma untuk menghitung *free float* (FF) dapat digambarkan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.8** Bagan Alir *Free Float* (FF)

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung *free float*(FF). Dimulai pada titik saat paling lambat dimulainya kegiatan LS(i,j), kemudian membaca data “node(i)”, “node(j)”, “durasi” dan “m”. “node(i)” dan “node(j)” menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan “m” menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk i=1. Apabila i ada maka di kolom 9 diisi hasil pengurangan antara hasil di kolom 5 dengan hasil di kolom 3 setelah itu di tambah hasil di kolom 4. Langkah selanjutnya dicari i berikutnya sampai banyaknya baris. Jika sudah selesai sampai banyaknya baris maka penentuan lintasan kritis dapat diproses. Jika belum maka harus dilakukan pencarian i lagi sampai selesai banyaknya baris.

Setelah melakukan perhitungan maju, perhitungan mundur dan perhitungan kelonggaran waktu (*float*) maka dapat di tentukan kegiatan- kegiatan kritis yang akhirnya akan membentuk suatu lintasan kritis.

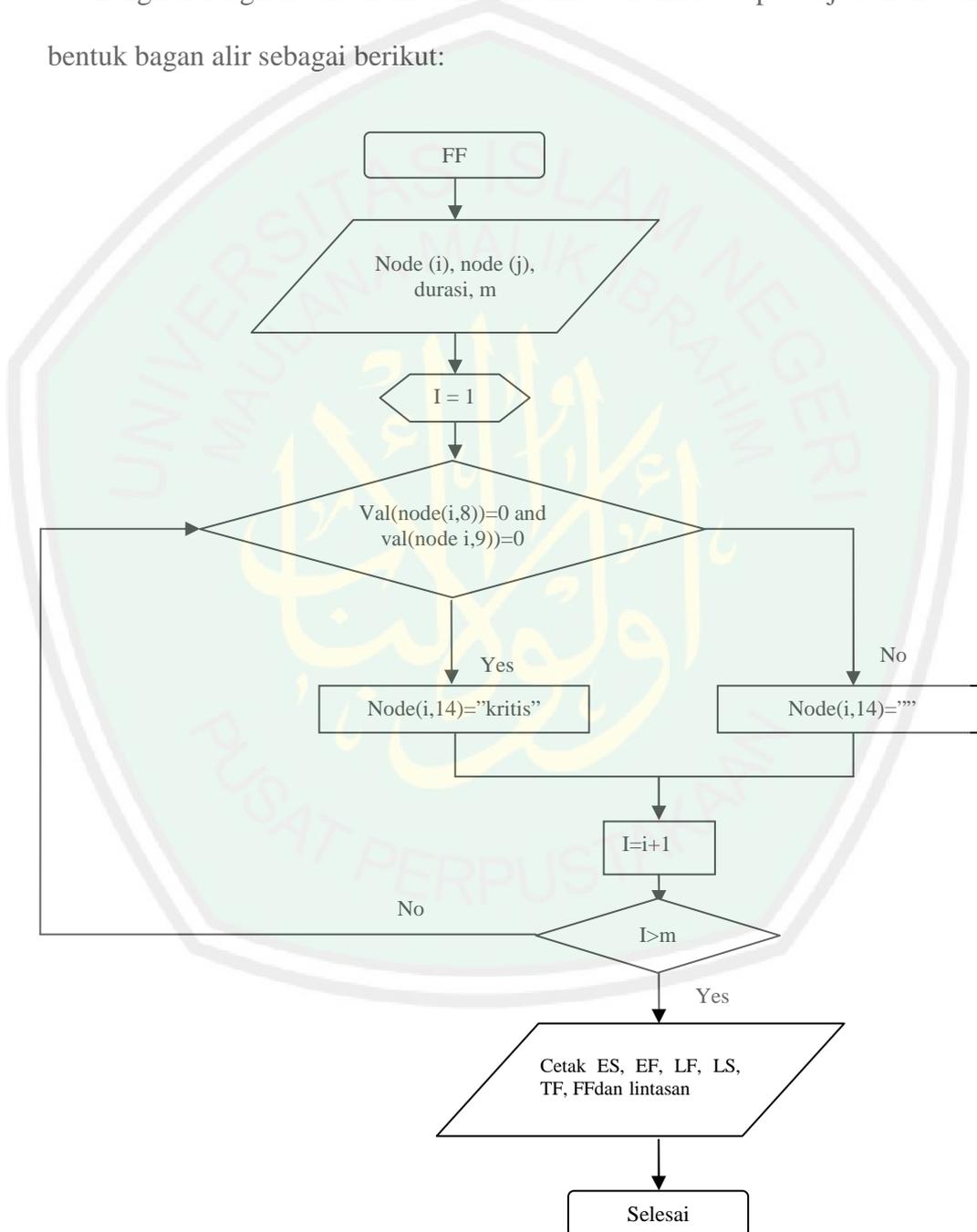
Algoritma penentuan lintasan:

1. Mulai
2. Baca data (node(i), node(j), durasi, m)
3. Pemberian nilai awal untuk i=1
4. Mengisikan kolom 14 dengan keterangan “kritis” jika terdapat suatu kondisi pada kolom 8 dan pada kolom 9 bernilai 0 dan mengisikan keterangan “:]” jika di kolom 8 dan di kolom 9 tidak bernilai 0
5. Mencari i berikutnya sampai banyaknya baris

6. Cetak hasil (ES, EF, LF, LS, TF, FF dan lintasan)

7. Selesai .

Langkah-langkah untuk menentukan lintasan kritis dapat dijelaskan dalam bentuk bagan alir sebagai berikut:

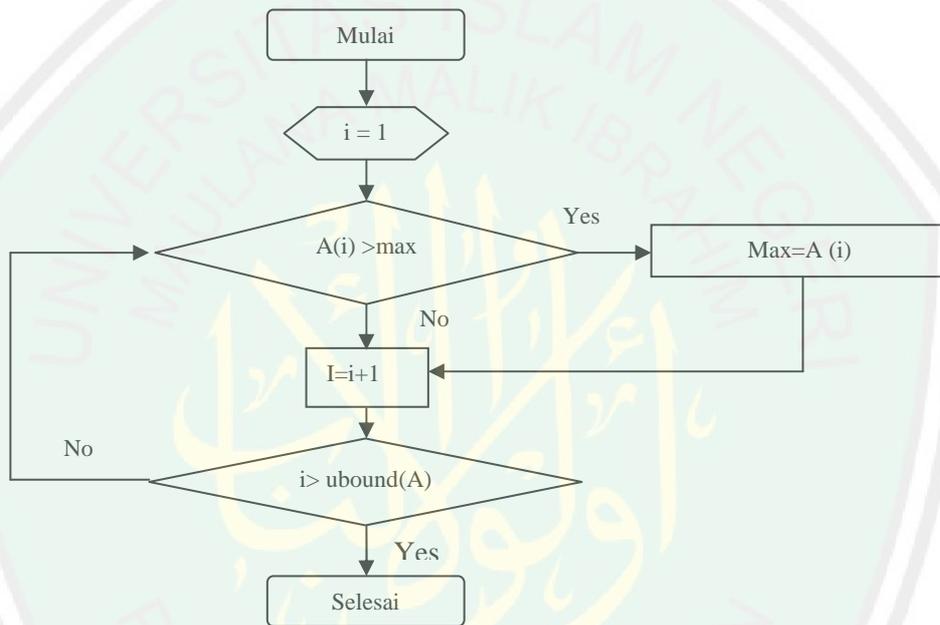


Gambar 4.9 Bagan Alir Lintasan

Bagan alir di atas menunjukkan langkah-langkah untuk menghitung *free float* (FF). Dimulai pada titik saat paling lambat dimulainya kegiatan LS(i,j), kemudian membaca data “node(i)”, “node(j)”, “durasi” dan “m”. “node(i)” dan “node(j)” menyatakan kegiatan (i,j), durasi menyatakan waktu penyelesaian kegiatan (i,j) dan “m” menyatakan banyaknya baris data yang akan diproses. Pemberian nilai awal untuk  $i=1$ . Jika terdapat suatu kondisi di kolom 8 dan di kolom 9 bernilai 0 maka di kolom 14 diberi keterangan “kritis” jika di kolom 8 dan di kolom 9 tidak bernilai 0 maka di kolom 14 diberi keterangan “:]”. Langkah selanjutnya dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai sampai banyaknya baris maka mencetak hasil (ES, EF, LF, LS, TF, FF dan lintasan). Jika belum maka harus dilakukan pencarian  $i$  lagi sampai selesai banyaknya baris. Proses berakhir pada titik selesai.

Kode program untuk menentukan total float (TF), free float (FF) dan lintasan dapat dilihat pada **lampiran 2**.

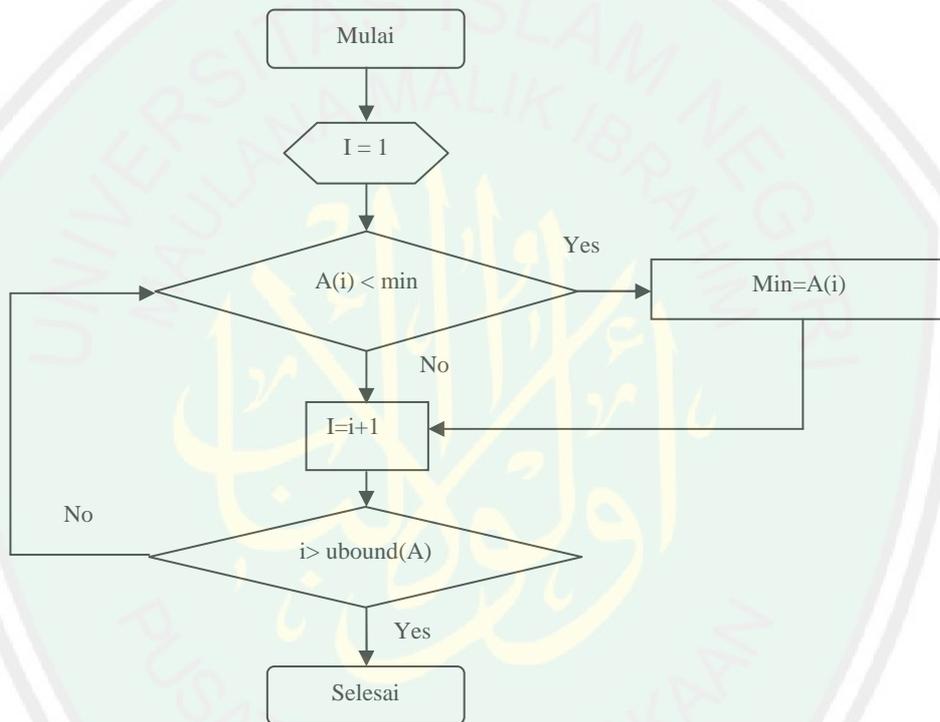
Di dalam perhitungan maju yaitu di dalam menghitung saat tercepat dimulainya kegiatan  $ES(i,j)$  dan saat tercepat diselesaikannya kegiatan  $EF(i,j)$  terdapat suatu kondisi untuk menentukan waktu maksimal. Langkah-langkah untuk menentukan waktu maksimal dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.10** Bagan Alir Waktu Maksimal

Bagan alir di atas menunjukkan langkah-langkah untuk menentukan waktu maksimal. Dimulai pada titik mulai. Pemberian nilai awal untuk  $i=1$ . Jika dalam beberapa baris terdapat hasil yang sama maka dipilih nilai yang maksimal dari beberapa baris tersebut. Jika tidak ada maka dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka proses dapat diakhiri.

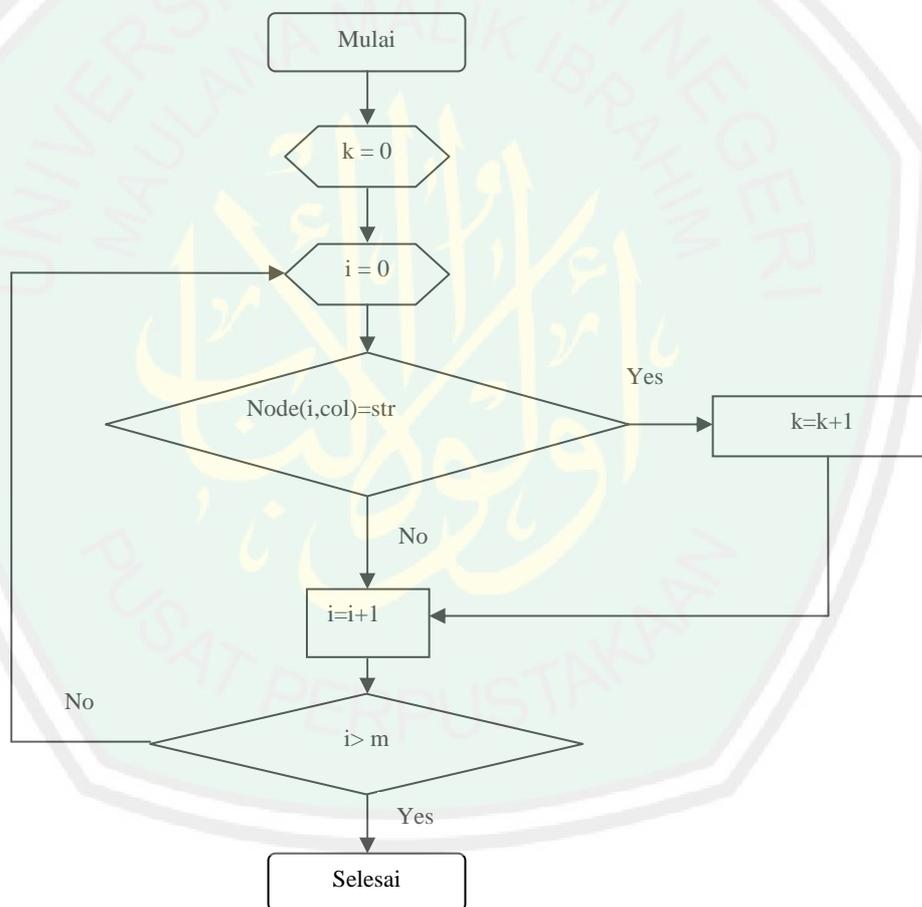
Di dalam perhitungan mundur yaitu di dalam menghitung saat paling lambat dimulainya kegiatan  $LS(i,j)$  dan saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  terdapat suatu kondisi untuk menentukan waktu minimal. Langkah-langkah untuk menentukan waktu minimal dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.11** Flowchart Waktu Minimal

Bagan alir di atas menunjukkan langkah-langkah untuk menentukan waktu minimal. Dimulai pada titik mulai. Pemberian nilai awal untuk  $i=1$ . Jika dalam beberapa baris terdapat hasil yang sama maka dipilih nilai yang minimal dari beberapa baris tersebut. Jika tidak ada maka dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka proses dapat diakhiri.

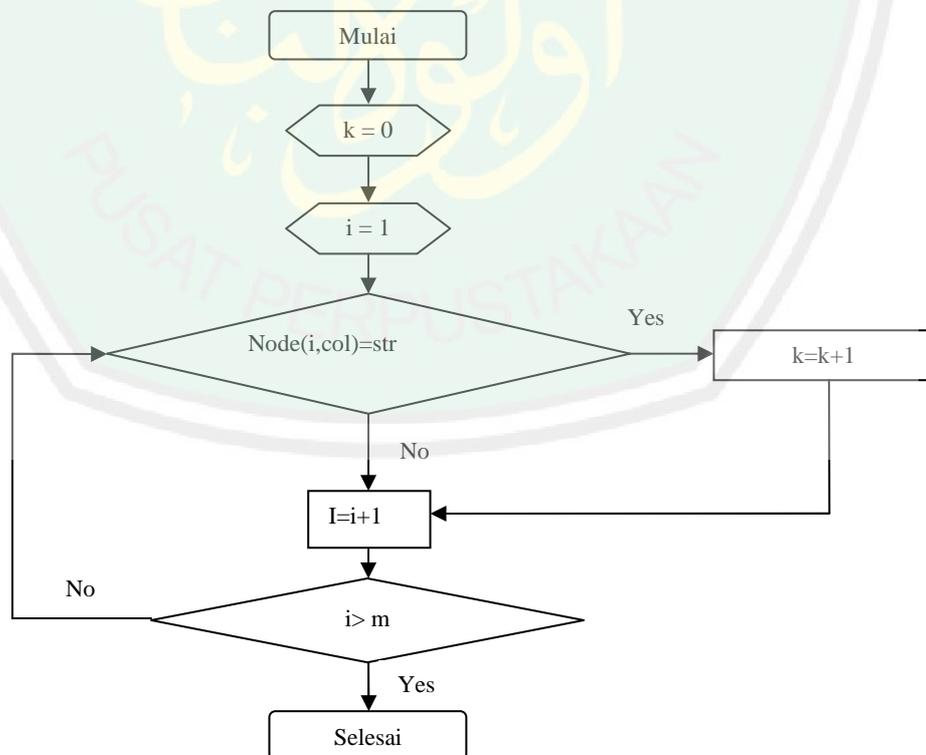
Di dalam perhitungan maju dan perhitungan mundur yaitu di dalam menghitung saat tercepat dimulainya kegiatan  $ES(i,j)$ , saat tercepat diselesaikannya kegiatan  $EF(i,j)$  dan saat paling lambat diselesaikannya kegiatan  $LF(i,j)$  terdapat suatu fungsi find untuk mencari letak baris yang dicari. Langkah-langkah untuk menentukan fungsi find dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.12** Bagan Alir Fungsi Find

Bagan alir di atas menunjukkan langkah-langkah untuk menentukan fungsi find untuk mencari letak baris yang dicari. Dimulai pada titik mulai. Pemberian nilai awal untuk  $k=0$  dan  $i=0$ . Jika terdapat suatu kondisi bahwa  $i$  ada maka dicari  $k$  berikutnya, jika tidak ada maka dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka proses dapat diakhiri. Jika belum selesai maka harus diproses sampai selesai banyaknya baris.

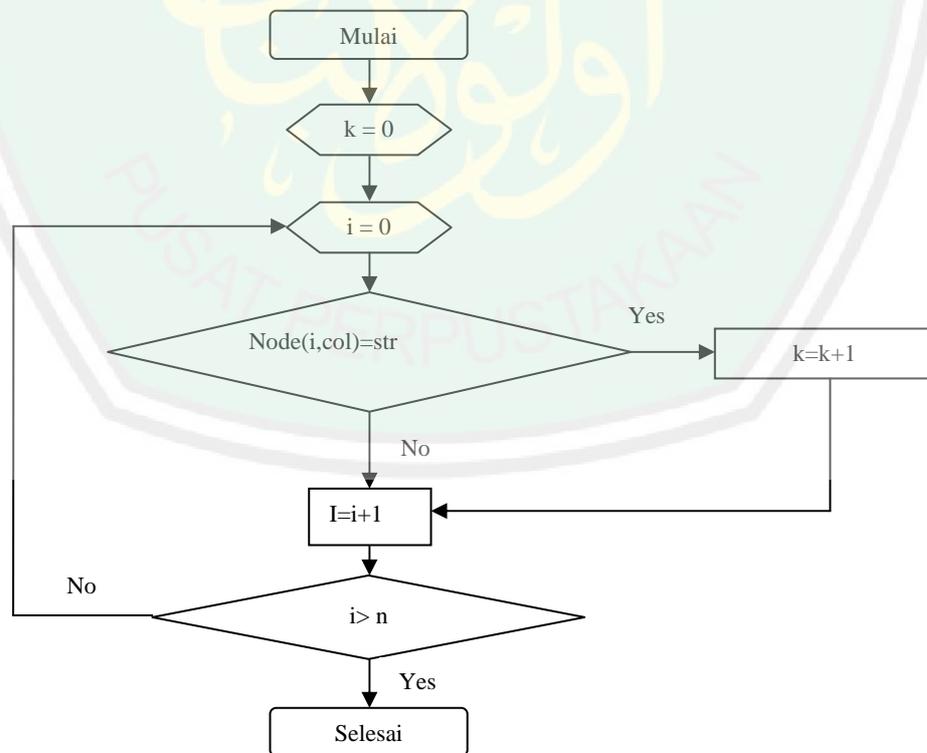
Di dalam perhitungan mundur yaitu di dalam menghitung saat paling lambat diselesaikannya kegiatan  $LS(i,j)$  terdapat suatu fungsi find1 untuk mencari letak baris yang dicari. Langkah-langkah untuk menentukan fungsi find1 dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.13** Bagan Alir Fungsi Find1

Bagan alir di atas menunjukkan langkah-langkah untuk menentukan fungsi find1 untuk mencari letak baris yang dicari. Dimulai pada titik mulai. Pemberian nilai awal untuk  $k=0$  dan  $i=1$ . Jika terdapat suatu kondisi bahwa  $i$  ada maka dicari  $k$  berikutnya, jika tidak ada maka dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka proses dapat diakhiri. Jika belum selesai maka harus diproses sampai selesai banyaknya baris.

Di dalam perhitungan mundur yaitu di dalam menghitung saat tercepat diselesaikannya kegiatan  $ES(i,j)$  terdapat suatu fungsi find2n untuk mencari letak baris yang dicari. Langkah-langkah untuk menentukan fungsi find2n dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.14** Bagan Alir Fungsi Find2n

Bagan alir di atas menunjukkan langkah-langkah untuk menentukan fungsi  $find2n$  untuk mencari letak baris yang dicari. Dimulai pada titik mulai. Pemberian nilai awal untuk  $k=0$  dan  $i=0$ . Jika terdapat suatu kondisi bahwa  $i$  ada maka dicari  $k$  berikutnya, jika tidak ada maka dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka proses dapat diakhiri. Jika belum selesai maka harus diproses sampai selesai banyaknya baris.

Kode program untuk menentukan waktu maksimal, waktu minimal, fungsi  $find$ , fungsi  $find1$ , fungsi  $find2n$  dapat dilihat pada **lampiran 2**.

Suatu kegiatan yang tidak mempunyai kelonggaran (*float*) disebut kegiatan kritis. Dengan kata lain, kegiatan kritis mempunyai  $FF = TF = 0$ . Pada contoh di atas, kegiatan kritisnya adalah kegiatan-kegiatan  $X1, X2, X3, X5, X14, X7, \dots$ dst. Kegiatan-kegiatan kritis ini akan membentuk **lintasan kritis** yang biasanya dimulai dari *initial event* sampai ke *terminal event*.

#### **4.2. Menentukan Waktu Optimal Penjadwalan Proyek.**

Dari perhitungan maju dan perhitungan mundur akan didapat waktu optimal pengerjaan proyek. Apabila dari perhitungan maju pada *terminal event* di dapat jumlah hari pengerjaan proyek dan jika hari itu dihitung dengan perhitungan mundur kembali pada hari ke-0, maka jumlah hari pada *terminal event* tersebut adalah waktu optimal pengerjaan proyek.

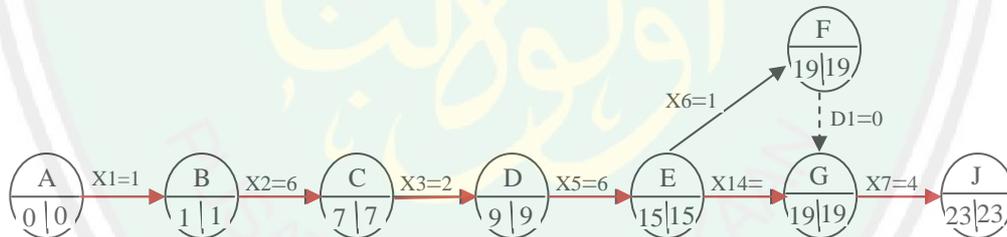
Pada contoh kasus di atas setelah dilakukan perhitungan maju dan perhitungan mundur didapat waktu optimalnya adalah 74 hari. Jadi, waktu penyelesaian pembangunan rumah tersebut dapat diselesaikan selama 74 hari.

### 4.3 Penyusunan Diagram Jaringan Kerja

Algoritma penyusunan diagram kerja:

1. Data.
2. Menggambarkan kejadian-kejadian dengan lingkaran berdasarkan urutan kejadian.
3. Menggambarkan kegiatan-kegiatan dengan anak panah berdasarkan urutan kegiatan
4. Memberi kode dan durasi pada setiap kegiatan (anak panah).
5. Membagi kejadian (lingkaran) menjadi 3 bagian.
6. Memberi simbol, nilai TE dan TL pada setiap kejadian (lingkaran)
7. Menentukan lintasan kritis.

Contoh:



Contoh diagram jaringan kerja (*network*) di atas merupakan potongan dari keseluruhan diagram jaringan kerja dari proyek pembangunan rumah tipe-49 perumahan Russelia Greenfield Malang. Garis panah merah menunjukkan lintasan kritis, dimana lintasan tersebut pengerjaan kegiatannya tidak boleh di tunda/terlambat. Karena apabila di tunda/terlambat dapat mempengaruhi waktu penyelesaian proyek secara keseluruhan. Sehingga waktu penyelesaian proyek menjadi lebih banyak dari waktu yang ditetapkan sebelumnya, yaitu 74 hari.

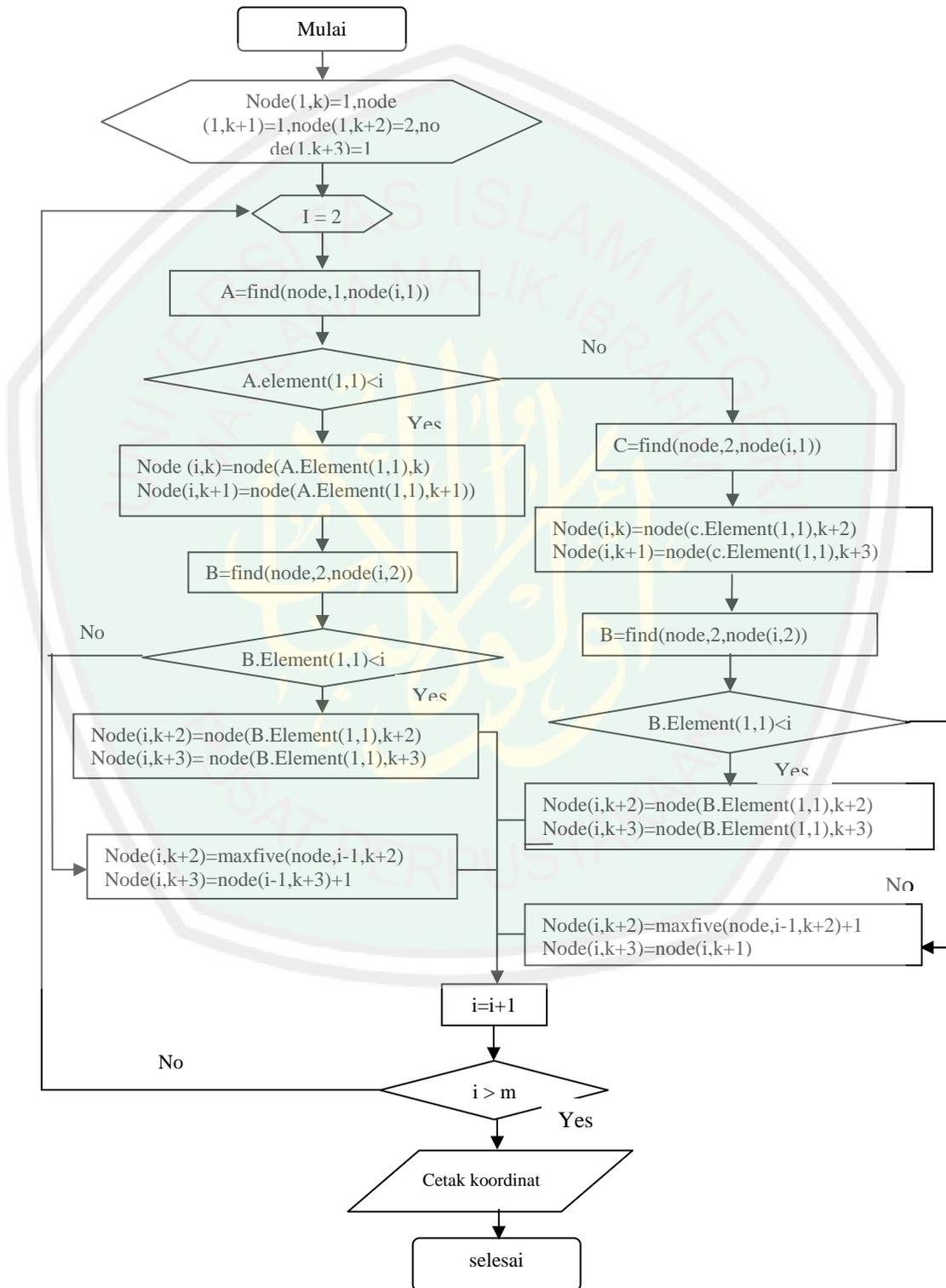
Dalam pembangunan suatu proyek setiap kegiatan-kegiatannya harus dikerjakan dengan hasil yang sempurna, agar kegiatan-kegiatan yang mengikutinya juga dapat dikerjakan tanpa kendala dari kegiatan sebelumnya. Sehingga tidak akan ada suatu keterlambatan dalam pengerjaannya. Hal ini dijelaskan dalam suatu ayat:

فَإِذَا فَرَغْتَ فَانصَبْ ﴿٧﴾

Artinya: *Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain (asy-Syarh:7-8).*

Ayat tersebut mengisyaratkan agar tidak menunda-nunda suatu pekerjaan hingga terdapat waktu luang (menganggur) antara pekerjaan yang satu dengan yang lain. Dan dalam mengerjakan suatu pekerjaan diharapkan dapat dikerjakan dengan sungguh-sungguh dengan memanfaatkan waktu yang ada, yang nanti akhirnya dapat diperoleh suatu hasil pekerjaan yang optimal.

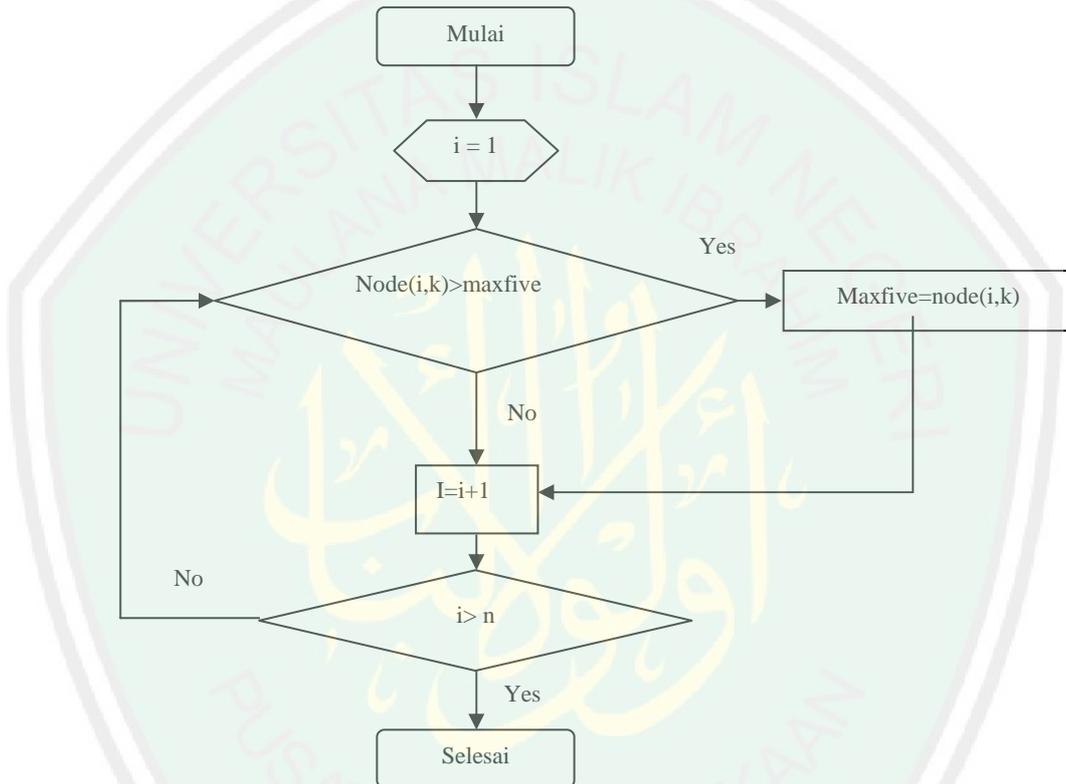
Langkah-langkah untuk menentukan koordinat lingkaran dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



Gambar 4.15 Bagan Alir untuk Menentukan Koordinat Gambar

Dalam menggambar diagram jaringan kerja maka harus ditentukan koordinat dari kejadian (lingkaran). Bagan alir di atas menunjukkan langkah-langkah untuk menentukan koordinat lingkaran. Dimulai pada titik mulai. Pertama ditentukan koordinat dua lingkaran yaitu  $x_1, y_1, x_2$  dan  $y_2$ . Selanjutnya dilakukan perulangan dengan  $i=2$  dan didefinisikan variabel A sebagai fungsi untuk mencari node(i) di kolom 1. Jika A ada koordinatnya mengikuti koordinat  $x_1$  dan  $y_1$  sebelumnya. Selanjutnya didefinisikan variabel B sebagai fungsi untuk mencari node(j) di kolom 2. Jika B ada koordinat  $x_2$  dan  $y_2$  tetap, jika B tidak ada maka  $x_2$  dan  $y_2$  koordinatnya di tambah 1 dari koordinat yang maksimal. Jika A tidak ada maka dibuat variabel C untuk mencari node(i) di kolom 2. Jika C ada  $x_1$  sama dengan  $x_2$  koordinat sebelumnya dan  $y_2$  sama dengan  $y_2$  koordinat sebelumnya. Langkah selanjutnya didefinisikan variabel B untuk mencari node(j) di kolom 2. Jika B ada  $x_2$  dan  $y_2$  sama dengan  $x_2$  dan  $y_2$  sebelumnya. Jika B tidak ada maka  $x_2$  koordinatnya di tambah 1 dari koordinat yang maksimal dan  $y_2$  sama dengan koordinat  $y_1$ . Langkah selanjutnya dicari i berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka mencetak koordinat. Jika belum selesai maka harus diproses sampai selesai banyaknya baris. Proses diakhiri pada titik selesai.

Dalam menggambar diagram jaringan kerja maka dibutuhkan koordinat maksimal untuk menentukan koordinat berikutnya. Langkah-langkah untuk menentukan koordinat maksimal dapat dijelaskan dalam bentuk bagan alir sebagai berikut:



**Gambar 4.16** Bagan Alir untuk Menentukan Koordinat Maksimal

Bagan alir di atas menunjukkan langkah-langkah untuk menentukan koordinat maksimal. Dimulai pada titik mulai. Dicari  $i$  dengan pemberian nilai awal untuk  $i=1$ . Jika  $i$  ada dipilih koordinat yang maksimal. Jika tidak ada maka dicari  $i$  berikutnya sampai banyaknya baris. Jika sudah selesai diproses sampai banyaknya baris maka proses dapat diakhiri.

Kode program untuk menentukan koordinat lingkaran dan koordinat maksimal dapat dilihat pada **lampiran 2**.

#### 4.4 Simulasi program

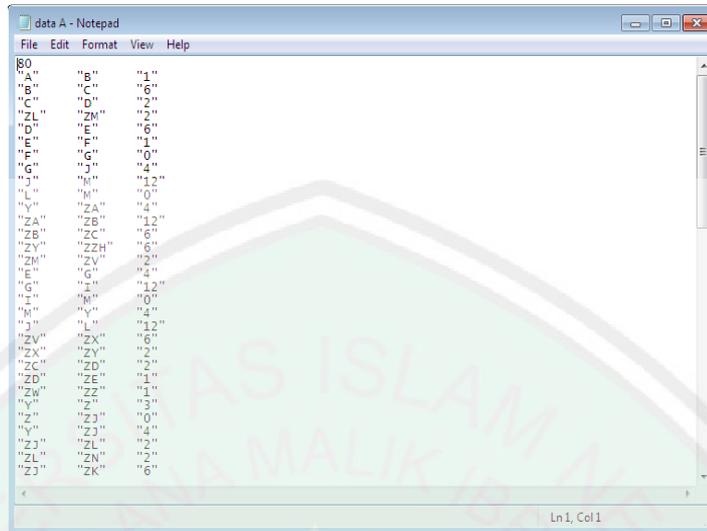
- a. Buka program dan klik start seperti pada gambar di bawah ini untuk memulai menggunakan program.



Gambar 4.17 Tampilan Awal Program

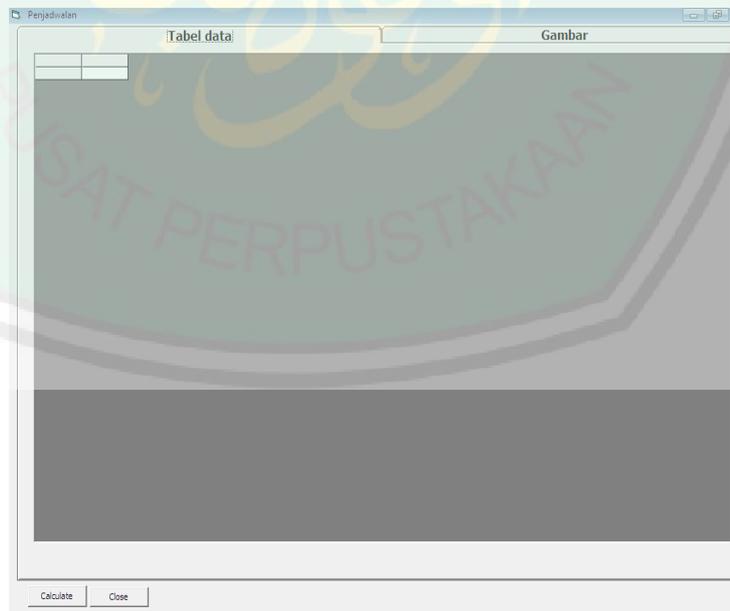
Tombol start adalah tombol untuk memulai menggunakan program dan tombol exit adalah tombol untuk keluar dari program

- b. Ketik data yang berupa jenis-jenis kegiatan (i,j) dan durasinya dalam notepad, baris pertama isikan banyak data dan baris selanjutnya isikan kegiatan (i,j) dan durasi, kegiatan i dan j ditulis dalam dua tanda petik, seperti pada gambar di bawah ini:



**Gambar 4.18** Notepad untuk Menulis Data

- c. Simpan data tersebut dan buka program, setelah mengklik menu start pada langkah awal tadi akan muncul tampilan seperti di bawah ini . Klik calculate dan ambil data yang sudah dituliskan dalam notepad .



**Gambar 4.19** Tampilan Untuk Input Program dari Notepad

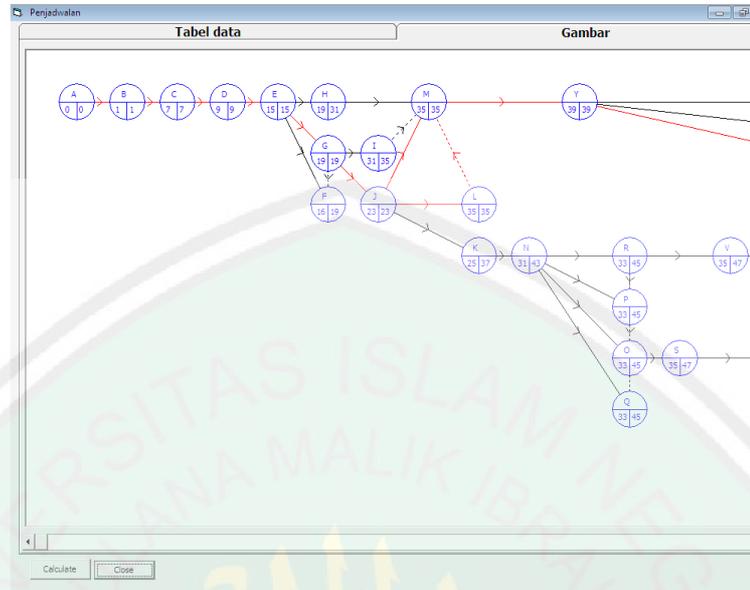
- d. Setelah langkah di atas akan tampil hasil perhitungan ES, EF,LF,LS,TF,FF dan lintasan, seperti gambar di bawah ini:

No	Node [i]	Node [j]	Durasi	ES	EF	LF	LS	TF	FF	X1	Y1	X2	Y2	Lintasan
1.	A	B	1	0	1	1	0	0	0	1	1	2	1	1 Kritis
2.	B	C	6	1	7	7	1	0	0	2	1	3	3	1 Kritis
3.	C	D	2	7	9	9	7	0	0	3	1	4	4	1 Kritis
4.	D	E	6	9	15	15	9	0	0	4	1	5	5	1 Kritis
5.	E	H	4	15	19	31	15	12	0	5	1	6	6	1-3
6.	E	G	4	15	19	19	15	0	0	5	1	6	6	2 Kritis
7.	E	F	1	15	16	19	15	3	0	5	1	6	6	3-1
8.	F	G	0	16	19	19	19	3	3	6	3	6	6	2-1
9.	G	I	12	19	31	35	19	4	0	6	2	7	7	2-1
10.	G	J	4	19	23	23	19	0	0	6	2	7	7	3 Kritis
11.	H	M	4	19	23	35	19	12	12	6	1	8	8	1-1
12.	I	M	0	31	35	35	35	4	4	7	2	8	8	1-1
13.	J	L	12	23	35	35	23	0	0	7	3	9	9	3 Kritis
14.	J	K	2	23	25	37	23	12	0	7	3	9	9	4-1
15.	J	M	12	23	35	35	23	0	0	7	3	8	8	1 Kritis
16.	K	N	6	25	31	43	23	12	0	9	4	10	10	4-1
17.	L	M	0	35	35	35	35	0	0	8	3	8	8	1 Kritis
18.	M	Y	4	35	39	39	35	0	0	8	1	11	11	1 Kritis
19.	N	R	2	31	33	45	43	12	0	10	4	12	12	4-1
20.	N	P	2	31	33	45	43	12	0	10	4	12	12	5-1
21.	N	O	2	31	33	45	43	12	0	10	4	12	12	6-1
22.	N	Q	2	31	33	45	43	12	0	10	4	12	12	7-1
23.	O	S	2	33	35	47	45	12	0	12	6	13	13	6-1
24.	P	R	0	33	33	45	45	12	0	12	5	12	12	4-1
25.	Q	R	0	33	33	45	45	12	0	12	5	12	12	4-1
26.	R	V	2	33	35	47	45	12	0	12	4	14	14	4-1
27.	S	T	2	35	37	49	47	12	0	13	6	15	15	6-1
28.	T	U	1	37	38	50	49	12	0	15	6	16	16	6-1
29.	U	X	0	38	38	50	50	12	0	16	6	17	17	6-1
30.	V	W	2	35	37	49	47	12	0	14	4	18	18	4-1
31.	W	Y	1	37	38	50	49	12	0	16	4	17	17	6-1
32.	X	Z	4	38	42	54	50	12	0	17	6	19	19	6-1
33.	X	ZB	2	38	42	54	50	14	2	17	6	19	19	7-1
34.	Y	Z	3	39	42	48	39	6	0	11	1	20	20	1-1
35.	Y	ZJ	4	39	43	48	39	5	0	11	1	20	20	2-1
36.	Y	ZA	4	39	43	43	39	0	0	11	1	20	20	3 Kritis

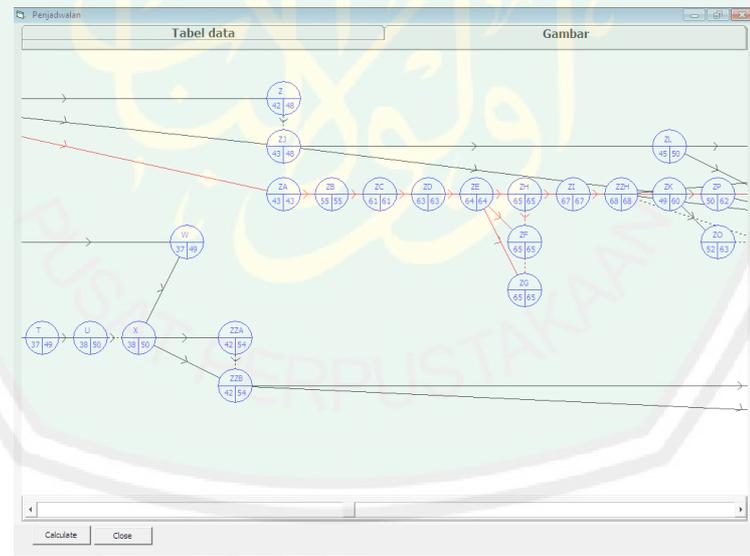
Gambar 4.20 Tampilan Hasil Perhitungan

ES adalah saat tercepat dimulainya kegiatan, EF (saat tercepat diselesaikannya kegiatan), LS (saat paling lambat dimulainya kegiatan), LF (saat paling lambat diselesaikannya kegiatan), TF (waktu mengambang), FF (waktu mengambang) dan lintasan berisi keterangan yang menunjukkan kegiatan-kegiatan kritis .

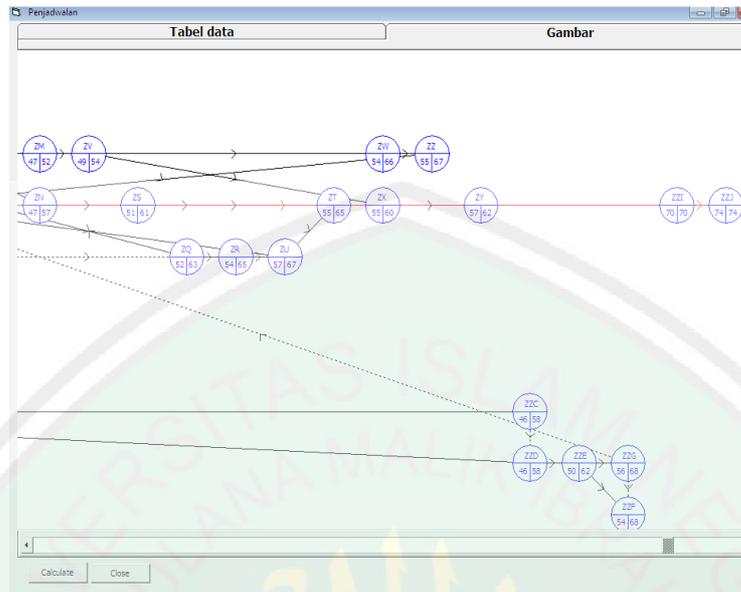
- e. Langkah selanjutnya klik menu gambar untuk mengetahui diagram jaringan kerja dari data yang diproses tersebut dan hasilnya seperti gambar di bawah ini:



**Gambar 4.21** Potongan Gambar ke-1 Diagram Jaringan Kerja



**Gambar 4.22** Potongan Gambar ke-2 Diagram Jaringan Kerja



**Gambar 4.23** Potongan Gambar ke-3 Diagram Jaringan Kerja

Gambar diagram jaringan kerja di atas terdiri dari 3 potongan gambar dikarenakan gambar terlalu lebar, jadi gambar tidak bisa ditampilkan dalam satu tampilan. Gambar panah adalah simbol kegiatan dan lingkaran adalah simbol kejadian. Garis merah menunjukkan lintasan kritis dimana kegiatan-kegiatan yang ada dalam lintasan tersebut tidak boleh ditunda pengerjaannya karena jika tertunda maka akan mempengaruhi waktu penyelesaian pembangunan proyek yang sudah dijadwalkan sebelumnya. Jumlah hari pada akhir kejadian menunjukkan waktu penyelesaian proyek secara keseluruhan.

#### 4.5 Interpretasi program

Program untuk penjadwalan proyek dengan menggunakan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*) ini memudahkan dalam melakukan proses perhitungan dan penggambaran dalam penjadwalan proyek. Karena untuk menjadwalkan suatu proyek dibutuhkan ketelitian yang sangat tinggi dalam proses perhitungannya, untuk mendapatkan waktu optimal harus

dilakukan beberapa langkah-langkah perhitungan, yaitu perhitungan maju, perhitungan mundur dan perhitungan *float* (waktu mengambang).

Dengan jumlah data yang banyak, perhitungan manual akan memberikan kesulitan bagi seseorang yang akan membuat suatu penjadwalan proyek, karena apabila telah melakukan perhitungan maju dan dalam perhitungan tersebut diperoleh total hari penyelesaian proyek, belum tentu jumlah hari tersebut hasilnya benar, untuk mengetahui benar atau tidaknya maka harus dilakukan perhitungan mundur. Apabila dalam perhitungan maju sudah diperoleh total hari maka total hari tersebut digunakan sebagai acuan awal dalam melakukan perhitungan mundur dan jika dari perhitungan mundur tadi tidak diperoleh 0 pada *initial event* maka ada kemungkinan perhitungan yang dilakukan salah, maka harus diteliti dimana letak kesalahan dan harus dilakukan perhitungan ulang sampai benar.

Dalam skripsi ini untuk uji coba program penulis menggunakan data pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield Malang. Setelah dilakukan perhitungan menggunakan program diperoleh total hari penyelesaian pembangunan proyek 74 hari dan penulis mencoba melakukan perhitungan manual untuk membandingkan hasil yang diperoleh program. Dari perhitungan manual tersebut juga diperoleh total hari penyelesaian proyek adalah 74 hari. Jadi program untuk penjadwalan proyek dengan menggunakan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*) ini dapat dipercaya kebenarannya.

## **BAB V**

### **PENUTUP**

#### **1.1 Kesimpulan**

Berdasarkan pembahasan pada BAB IV maka dapat disimpulkan beberapa hal sebagai berikut:

1. Algoritma dan bagan alir dari program penjadwalan proyek ini terdiri dari input yang berupa: kegiatan-kegiatan pembangunan proyek berdasarkan urutannya dan durasi/waktu setiap kegiatan. Yang di dalam proses hitungannya menggunakan metode CPM model AOA dan sebagai hasilnya (outputnya) adalah nilai ES (saat tercepat dimulainya kegiatan), EF (saat tercepat diselesaikannya kegiatan), LS (saat paling lambat dimulainya kegiatan), LF (saat paling lambat diselesaikannya kegiatan), TF (waktu mengambang) dan FF (waktu mengambang) yang digunakan untuk menentukan waktu optimal penjadwalan proyek. Dan susunan program dapat dilihat pada **lampiran 2**.
2. Dalam analisis algoritma dengan menghitung banyak kerja/operasi (penjumlahan, perbandingan) yang dilakukan oleh algoritma dalam perhitungan maju sebanyak  $(5n^2 - n)$  kali, perhitungan mundur sebanyak  $(4n^2)$  kali dan perhitungan kelonggaran waktu sebanyak  $(2n)$  kali. Jadi dengan menggunakan data pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield Malang dengan banyak data/ banyak  $n = 80$ , maka banyak kerja algoritma dalam perhitungan maju sebanyak 31.920 kali, perhitungan mundur sebanyak 25.600 kali dan perhitungan kelonggaran

waktu sebanyak 160 kali. Dalam pemakaian program, dengan menggunakan komputer yang mempunyai kecepatan prosesor 2.30 GHz untuk melakukan perhitungan maju dibutuhkan waktu 0,000014 detik. Perhitungan mundur dibutuhkan waktu 0,000011 detik. Dan perhitungan kelonggaran waktu dibutuhkan waktu 0,000070 detik.

### **1.2 Saran**

Karena keterbatasan penulis, pada program analisis penjadwalan proyek ini penulis hanya membuat program untuk analisis waktunya saja, bagi pembaca yang ingin mengembangkan program ini dapat membuat program untuk analisis biayanya dengan menggunakan metode CPM (*Critical Path Method*) model AOA (*Activity On Arrow*).

## DAFTAR PUSTAKA

- Ad-Dimasyqi, Ibnu Katsir. 2000. *Tafsir Ibnu Kasir Juz 4*. Bandung: Sinar baru Algensindo.
- Dewobroto, Wiryanto. 2004. *Aplikasi Sain dan Teknik dengan Visual Basic 6.0*. Jakarta: PT. Elex Media Komputindo.
- Dimiyati, Ahmaddan Dimiyati, Tjutju. 1994. *Operation Research Model-model Pengambilan Keputusan*. Bandung: PT. Sinar Baru Algensindo.
- Ervianto, Wulfram. 2005. *Manajemen Proyek Konstruksi (Edisi revisi)*. Yogyakarta: CV. Andi Offset.
- Herjanto, Eddy. 2008. *Manajemen Operasi*. Jakarta : PT. Grasindo.
- Pardosi, Mico.2005. *Bahasa Pemrograman Windows dan Internet Microsoft Visual Basic 6.0*. Surabaya: Dua Selaras.
- Ramadhan, Arief. 2004. *36 Jam Belajar Komputer Visual Basic 6.0*. Jakarta: PT. Elex Media Komputindo.
- Shihab, M. Quraish. 2004. *TAFSIR AL-MISBAH Pesan, Kesan dan Keserasian Al-Qur'an*. Jakarta: Lentera Hati.
- Sutanta, Edhy. 2004. *Algoritma Teknik Penyelesaian Permasalahan Untuk Komputasi*. Yogyakarta: Graha Ilmu.
- Taha, Hamdy.A. 1996. *Operations Research*. Jakarta: Binarupa Aksara.
- Yulikuspartono. 2004. *Pengantar Logika dan Algoritma*. Yogyakarta: CV. Andi Offset.
- Zulfikarijah, Fien. 2004. *Operation Research*. Malang: Bayumedia Publishing.

Data Pembangunan rumah tinggal tipe-49 perumahan Russelia Greenfield berdasarkan logika ketergantungannya.

	JENIS PEKERJAAN	KODE	DURASI /hari	KEGIATAN SEBELUM	NOTASI
<b>I</b>	<b>PEKERJAAN PERSIAPAN</b>				
1	Bowplank	X1	1	-	A-B
<b>II</b>	<b>PEKERJAAN TANAH</b>				
1	Galian tanah	X2	6	X1	B-C
2	Urugan pasir pondasi	X3	2	X2	C-D
3	Urugan pasir lantai	X4	2	X25	ZL-ZM
<b>III</b>	<b>PEKERJAAN PASANGAN</b>				
1	Pondasi batu kali	X5	6	X3	D-E
2	Rollag batu bata	X6	1	X5	E-F
3	Dummy	D1	0	X6	F-G
4	Pasangan bata tasram	X7	4	X14	G-J
5	Pasangan bata	X8	12	X7	J-M
6	Dummy	D2	0	X17	L-M
7	Plesteran dinding tasram	X9	4	X16	Y-ZA
8	Plesteran dinding	X10	12	X9	ZA-ZB
9	Benangan sudutan	X11	6	X10	ZB-ZC
10	Ban-banan	X12	6	X65	ZY-ZZH
11	Patlag	X13	2	X4	ZM-ZV
<b>IV</b>	<b>PEKERJAAN BETON</b>				
1	Sloof 15/20	X14	4	X5	E-G
2	Kolom praktis 15/15	X15	12	X14	G-I
3	Dummy	D3	0	X15	I-M
4	Ringbalk & gewel 15/15	X16	4	X8, D2, D3, X53	M-Y
5	Plat leuvel	X17	12	X7	J-L
<b>V</b>	<b>PEKERJAAN LANTAI</b>				
1	Keramik 40x40 motif	X18	6	X13	ZV-ZX
2	Keramik teras 40x40 motif	X19	2	X18	ZX-ZY
3	Keramik dinding KM 25x45	X20	2	X11	ZC-ZD
4	Keramik lantai KM 20x20	X21	1	X20	ZD-ZE
5	Keramik dapur 20x25	X22	1	X55	ZW-ZZ

6	Ramin	X23	3	X16	Y-Z
7	Dummy	D4	0	X23	Z-AJ
<b>VI</b>	<b>PEKERJAAN ATAP DAN PLAFOND</b>				
1	Galvalum	X24	4	X16	Y-AJ
2	Genteng flat supersonic	X25	2	X24, D4	ZJ-ZL
3	Wuwungan	X26	2	X25	ZL-ZN
4	Rangka plafond + eternit	X27	6	X24, D4	ZJ-ZK
5	List plafong gypsum	X28	3	X27	ZK-ZO
6	Dummy	D5	0	X28	ZO-ZQ
7	Lisplang	X29	4	X26	ZN-ZS
<b>VII</b>	<b>PEKERJAAN KAYU, PINTU DAN JENDELA</b>				
1	Kusen baja	X30	2	X7	J-K
2	Kusen kayu	X31	6	X30	K-N
3	Daun pintu (p1)	X32	2	X31	N-R
4	Dummy	D6	0	X34	Q-R
5	Daun pintu (p2)	X33	2	X31	N-P
6	Dummy	D7	0	X33	P-R
7	Daun pintu (p3)	X34	2	X31	N-Q
8	Daun jendela (j1)	X35	2	X31	N-O
9	Selot pintu	X36	2	X39	V-W
10	Sikutan jendela	X37	1	X41	T-U
11	Dummy	D8	0	X37	U-X
12	Handel pintu utama	X38	1	X36	W-X
13	Engsel pintu	X39	2	X32, D5, D6	R-V
14	Engsel jendela	X40	2	X35	O-S
15	Gerendel jendela	X41	2	X40	S-T
<b>VIII</b>	<b>PEKERJAAN CAT</b>				
1	Cat tembok luar	X42	4	X44	ZZE-ZZF
2	Dummy	D9	0	X42	ZZF-ZZG
3	Cat tembok dalam	X43	6	X44	ZZE-ZZG
4	Dummy	D10	0	X43, D9	ZZG-ZZH
5	Cat plafond	X44	4	X48, D12	ZZD-ZZE
6	Cat kusen baja	X45	2	X38, D8	X-ZZB
7	Cat kusen	X46	4	X38, D8	X-ZZA
8	Dummy	D11	0	X46	ZZA-ZZB
9	Cat rangka jendela	X47	4	X45, D10	ZZB-ZZC
10	Dummy	D12	0	X47	ZZC-ZZD
11	Cat daun pintu	X48	4	X45, D10	ZZB-ZZD
12	Cat lisplang	X49	2	X50	ZT-ZU
13	Cat genteng	X50	4	X29	ZS-ZT

<b>IX</b>	<b>PEKERJAAN SANITAIR</b>				
1	Septictank dan resepan	X51	4	X5	E-H
2	Titik kran	X52	1	X54	ZI-ZZH
3	Pasang PVC 3",4"	X53	4	X51	H-M
4	Closet duduk	X54	2	X58, D12, D13	ZH-ZI
5	Kitchen zing	X55	5	X13	ZV-ZW
6	Avour bak mandi	X56	1	X21	ZE-ZG
7	Avour kitchen zing	X57	1	X65	ZZ-ZZH
8	Avour lantai KM	X58	1	X21	ZE-ZH
9	Dummy	D13	0	X56	ZG-ZH
10	Tempat sabun	X59	1	X21	ZE-ZF
11	Dummy	D14	0	X59	ZF-ZH
<b>X</b>	<b>PEKERJAAN LISTRIK</b>				
1	Kotak panel	X60	1	X27	ZK-ZP
2	Saklar lampu	X61	2	X63	ZQ-ZR
3	Stop kontak	X62	2	X61	ZR-ZU
4	Arde	X63	1	X60	ZP-ZQ
5	Stop kontak antena	X64	1	X62, X49	ZU-ZZH
<b>XI</b>	<b>LAIN-LAIN</b>				
1	Taman	X65	2	X12,X52, X57,X64,D10	ZZH-ZZI
2	Carport	X66	4	X65	ZZI-ZZJ

Sumber: CV. Sekar Tanjung (Kontraktor dan Perdagangan Umum).

### **Kode program untuk menentukan saat tercepat dimulainya kegiatan ES**

```
For i = 1 To UBound(node, 1)
  D = find2n(node, 1, node(i, 1), i - 1)
  If D.m > 0 Then
    node(i, 4) = Val(node(D.Element(D.m, 1), 4))
  Else
    A = find(node, 2, node(i, 1))
    ReDim B(A.m)
    For j = 1 To A.m
      B(j) = Val(node(A.Element(j, 1), 3)) + Val(node(A.Element(j,
1), 4))
    Next j
    node(i, 4) = max(B)
  End If
Next i
```

### **Kode program untuk menentukan saat tercepat diselesaikannya kegiatan EF**

```
For i = 1 To UBound(node, 1)
  A = find(node, 2, node(i, 2))
  ReDim B(A.m)
  For j = 1 To A.m
    B(j) = Val(node(A.Element(j, 1), 3)) + Val(node(A.Element(j,
1), 4))
  Next j
  node(i, 5) = max(B)
Next i
```

### **Kode program untuk menentukan saat paling lambat diselesaikannya kegiatan LF**

```
For i = UBound(node, 1) To 1 Step -1
  A = find(node, 1, node(i, 2))
  If A.m > 0 Then
    ReDim B(A.m)
    For j = 1 To A.m
      B(j) = Val(node(A.Element(j, 1), 6)) -
Val(node(A.Element(j, 1), 3))
    Next j
    node(i, 6) = min(B)
  Else
    node(i, 6) = node(i, 5)
  End If
Next i
```

### **Kode program untuk menentukan saat paling lambat dimulainya kegiatan LS**

```
For i = 1 To UBound(node, 1)
  A = find1(node, 1, node(i, 1))
  ReDim B(A.m)
  For j = 1 To A.m
    B(j) = Val(node(A.Element(j, 1), 6)) - Val(node(A.Element(j, 1), 3))
  Next j
  node(i, 7) = min(B)
Next i
```

### **Kode program untuk menentukan *free float* (FF)**

```
For i = 1 To UBound(node, 1)
  node(i, 8) = Val(node(i, 6)) - (Val(node(i, 3)) + Val(node(i, 4)))
Next i
```

### **Kode program untuk menentukan *total float* (TF)**

```
For i = 1 To UBound(node, 1)
  node(i, 9) = Val(node(i, 5)) - (Val(node(i, 3)) + Val(node(i, 4)))
Next i
```

### **Kode program untuk menentukan lintasan**

```
For i = 1 To UBound(node, 1)
  If Val(node(i, 8)) = 0 And Val(node(i, 9)) = 0 Then
    node(i, 14) = "Kritis"
  Else
    node(i, 14) = ":]"
  End If
Next i
```

### **Kode program untuk menentukan waktu maksimal**

```
Public Function max(A() As Integer) As Integer
  max = A(1)
  For i = 1 To UBound(A)
    If A(i) > max Then max = A(i)
  Next i
End Function
```

### **Kode program untuk menentukan waktu minimal**

```
Public Function min(A() As Integer) As Integer
  min = A(1)
  For i = 1 To UBound(A)
    If A(i) < min Then min = A(i)
  Next i
End Function
```

### **Kode program untuk fungsi find**

```
Public Function find(node() As String, col As Integer, str As String) As matrix
    K = 0
    For i = 0 To UBound(node, 1)
        If node(i, col) = str Then
            K = K + 1
        End If
    Next i
    With find
    End Function
```

### **Kode program untuk fungsi find1**

```
Public Function find1(node() As String, col As Integer, str As String) As matrix
    K = 0
    For i = 1 To UBound(node, 1)
        If node(i, col) = str Then
            K = K + 1
        End If
    Next i
    End Function
```

### **Kode program untuk fungsi find2n**

```
Public Function find2n(node() As String, col As Integer, str As String, n As Integer) As matrix
    K = 0
    For i = 0 To n
        If node(i, col) = str Then
            K = K + 1
        End If
    Next i
    End Function
```

### **Kode program untuk menentukan koordinat lingkaran**

```
Public Function NodeK(ByRef node() As String, K As Integer) As matrix
    Dim A As matrix
    Dim B As matrix
    Dim c As matrix
    node(1, K) = 1: node(1, K + 1) = 1: node(1, K + 2) = 2: node(1, K + 3) = 1
    For i = 2 To UBound(node, 1)
        A = find(node, 1, node(i, 1))
        If A.Element(1, 1) < i Then
            node(i, K) = node(A.Element(1, 1), K)
            node(i, K + 1) = node(A.Element(1, 1), K + 1)

            B = find(node, 2, node(i, 2))
            If B.Element(1, 1) < i Then
                node(i, K + 2) = node(B.Element(1, 1), K + 2)
            End If
        End If
    Next i
    End Function
```

```

        node(i, K + 3) = node(B.Element(1, 1), K + 3)
    Else
        node(i, K + 2) = MaxFive(node, i - 1, K + 2)
        node(i, K + 3) = node(i - 1, K + 3) + 1
    End If
Else
    c = find(node, 2, node(i, 1))
    node(i, K) = node(c.Element(1, 1), K + 2)
    node(i, K + 1) = node(c.Element(1, 1), K + 3)
    B = find(node, 2, node(i, 2))
    If B.Element(1, 1) < i Then
        node(i, K + 2) = node(B.Element(1, 1), K + 2)
        node(i, K + 3) = node(B.Element(1, 1), K + 3)
    Else
        node(i, K + 2) = MaxFive(node, i - 1, K + 2) + 1
        node(i, K + 3) = node(i, K + 1)
    End If
End If
Next i
End Function

```

### **Kode program untuk menentukan koordinat maksimal**

```

Public Function MaxFive(node() As String, n As Integer, K As
Integer) As Integer
    MaxFive = node(1, K)
    For i = 1 To n
        If node(i, K) > MaxFive Then MaxFive = node(i, K)
    Next i
End Function

```