

PENDEKATAN FUZZY MULTI-KRITERIA UNTUK
MENENTUKAN LINTASAN KRITIS PADA JARINGAN
PROYEK

SKRIPSI

Oleh:

RISSA PRIMAHARANI

0610940053-94



PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011

**PENDEKATAN FUZZY MULTI-KRITERIA UNTUK
MENENTUKAN LINTASAN KRITIS PADA JARINGAN
PROYEK**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Matematika

Oleh:

RISSA PRIMA HARANI
0610940053-94



**PROGRAM STUDI MATEMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

LEMBAR PENGESAHAN SKRIPSI

**PENDEKATAN FUZZY MULTI-KRITERIA UNTUK
MENENTUKAN LINTASAN KRITIS PADA JARINGAN
PROYEK**

Oleh:

**RISSA PRIMA HARANI
0610940053-94**

Setelah dipertahankan di depan majelis pengaji
pada tanggal 18 April 2011
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam Bidang Matematika

Dosen Pembimbing I

Dosen Pembimbing II

Prof. Dr. Marjono, M.Phil
NIP. 196211161988031004

Prof. Dr. Agus Widodo, M.Kes
NIP. 195305231983031002

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf A., MSc
NIP. 196709071992031001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Rissa Primaharani
NIM : 0610940053-94
Jurusan : Matematika
Penulis Skripsi berjudul : Pendekatan *Fuzzy* Multi-Kriteria
Untuk Menentukan Lintasan
Kritis Pada Jaringan Proyek

Dengan ini menyatakan bahwa:

1. Isi skripsi yang saya buat benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka skripsi ini.
2. Apabila di kemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 18 April 2011
Yang menyatakan,

(Rissa Primaharani)
NIM. 0610940053-94

UNIVERSITAS BRAWIJAYA



PENDEKATAN FUZZY MULTI-KRITERIA UNTUK MENENTUKAN LINTASAN KRITIS PADA JARINGAN PROYEK

ABSTRAK

Skripsi ini difokuskan pada identifikasi lintasan kritis dalam jaringan proyek yang kompleks. Terdapat inovasi baru dalam mengidentifikasi penjadwalan proyek yang memadukan logika *fuzzy* dan teknik *Multi Criteria Decision Making* (MCDM). Tujuannya adalah untuk menentukan lintasan kritis yang tidak hanya mempertimbangkan durasi dari aktivitas, tetapi juga parameter penting tambahan seperti: variabilitas durasi, biaya, perlindungan yang tersedia dan risiko revisi desain utama. *Fuzzy PERT* digunakan untuk menentukan nilai *float* (waktu luang) pada masing-masing aktivitas, yang nantinya nilai *float* pada masing-masing aktivitas tersebut diperlukan untuk menentukan nilai parameter MCDM pada masing-masing lintasan. Untuk lingkup ini, parameter tersebut dievaluasi mengikuti pendekatan matematis berdasarkan logika *fuzzy* dipadukan menggunakan ‘*Technique for Order Preference by Similarity to Ideal Solution*’ (TOPSIS) untuk merangkingkan derajat kekritisan pada setiap lintasan. Jika nilai derajat kekritisan lintasan mendekati 1 maka lintasan tersebut semakin kritis. Dengan mengetahui tingkat kekritisan masing-masing lintasan pada suatu jaringan proyek, maka pelaksana proyek dapat berhati-hati dalam memonitor selama fase proyek berlangsung sehingga penyelesaian proyek dapat berjalan tepat waktu.

Kata Kunci: lintasan kritis, *fuzzy PERT*, MCDM, TOPSIS

UNIVERSITAS BRAWIJAYA



A FUZZY MULTI-CRITERIA APPROACH FOR CRITICAL PATH IDENTIFICATION IN PROJECT NETWORK

ABSTRACT

This paper focuses on the identification of the critical path in a complex projects networks. There is a new innovation in identifying project scheduling that combines fuzzy logic and Multi Criteria Decision Making (MCDM) techniques. The aim is to determine the critical path taking into account not only the expected duration of the tasks, but also additional critical parameters such as: ‘duration variability’, ‘costs’, ‘available protection’ and ‘risk of major design revisions’. Fuzzy PERT is used to determine the value to each activity is needed to determine the value of MCDM parameter on each path. To this scope, all the previously mentioned parameters are evaluated following a mathematical approach based on Fuzzy Logic and are aggregated using the ‘Technique for Order Preference by Similarity to Ideal Solution’ (TOPSIS) to define the resulting critical path. If the criticality degree value of the path close to 1 then it is being more critical. Due to their high degree of criticality, should also be carefully monitored during the ongoing phase of the project so that the project the completion time of the project to run on time.

Keywords: critical path; fuzzy PERT; MCDM; TOPSIS

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Assalamu'alaikum wr. wb

Segala puji dan syukur ke hadirat Allah SWT yang telah memberikan bimbingan dan petunjuk-Nya, sehingga penulisan skripsi yang berjudul "*Pendekatan Fuzzy Multi-Kriteria Untuk Menentukan Lintasan Kritis Pada Jaringan Proyek*" dapat terselesaikan dengan baik dan lancar. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW sebagai suri tauladan bagi penulis. Melalui kesempatan ini penulis menyampaikan rasa terima kasih yang sedalam-dalamnya dan permohonan maaf yang setulus-tulusnya atas kesalahan baik yang disengaja ataupun tidak disengaja selama penulisan skripsi ini kepada:

1. Prof. Dr. Marjono, M.Phil, selaku pembimbing I sekaligus Dekan Fakultas MIPA atas segala bimbingan, nasihat, motivasi serta kesabaran yang telah diberikan selama penulisan skripsi ini.
2. Prof. Dr. Agus Widodo, M.Kes., selaku pembimbing II atas segala bimbingan, nasihat, motivasi serta kesabaran yang telah diberikan selama penulisan skripsi ini.
3. Drs. Bambang Sugandi, M.Si, Dra. Ari Andari,M.S, dan Kwardiniya A.,S.Si.,M.Si selaku dosen penguji atas segala saran yang diberikan untuk perbaikan skripsi ini.
4. Seluruh bapak/ibu dosen Jurusan Matematika yang telah memberikan ilmunya kepada penulis, serta segenap staf dan karyawan TU Jurusan Matematika atas segala bantuannya.
5. Ayah, Ibunda, adikku tersayang dan sahabat terdekatku atas segala doa, kasih sayang dan dukungan, yang telah diberikan.
6. Teman-teman atas kebersamaan, kekompakan dan bantuan yang telah diberikan.
7. Semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran melalui email penulis r13.554.fighter@gmail.com. Akhir kata, semoga skripsi ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum wr. wb.

Malang, 11 April 2011

Penulis

UNIVERSITAS BRAWIJAYA



DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	iii
LEMBAR PERNYATAAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xviii
DAFTAR LAMPIRAN	xxi
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penulisan.....	2
1.5 Manfaat Penulisan	2
 BAB II TINJAUAN PUSTAKA	 3
2.1 Metode PERT (Project Evaluation and Review Technique)	3
2.2 Logika Fuzzy	9
2.3 Jaringan Kerja Fuzzy.....	14
2.4 Kriteria Keputusan dengan Menggunakan Multi Criteria Decision Making (MCDM)	20
2.5 Identifikasi Kekritisannya dengan Menggunakan Metode <i>Fuzzy Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)</i>	23
 BAB III PEMBAHASAN	 27
3.1 Algoritma Dalam Menentukan Lintasan Kritis.....	27
3.2 Data Umum Proyek.....	28
3.3 Algoritma Program	34
3.4 Jaringan Proyek.....	45
3.5 Menentukan Nilai Waktu <i>Float Fuzzy (Fuzzy Float Time)</i>	46

3.6 Kriteria Keputusan / <i>Multy Criteria Decision Making</i> (MCDM).....	52
3.7 Identifikasi Lintasan Kritis.....	66
BAB IV KESIMPULAN DAN SARAN	69
4.1 Kesimpulan	69
4.2 Saran.....	69
DAFTAR PUSTAKA	71
LAMPIRAN	73



BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir, berbagai aplikasi manajemen proyek telah sangat berkembang. Manajemen proyek tersebut menitikberatkan pada penjadwalan dan pengendalian kegiatan (penugasan) sehingga proyek dapat diselesaikan dengan waktu seminimal mungkin (Chen-Tung Chen dan Sue-Fen Huang, 2007). Sebagian besar proyek yang mencakup risiko, sulit untuk dikelola, karena mencakup banyak kegiatan terkait yang kompleks. Oleh karena itu beberapa teknik jaringan dikembangkan untuk mengetahui lintasan kritis yang menentukan waktu penyelesaian proyek.

Critical Path Methode (CPM) adalah metode paling mudah untuk memodelkan pelaksanaan sebuah proyek dan telah diterapkan di beberapa bidang industri. Perlu diketahui bahwa durasi pada metode CPM telah ditentukan dan diketahui, tetapi metode tersebut jarang memuaskan, karena aktivitas jaringan proyek sering tidak tentu atau berubah-ubah. Akibat dari ketidakpastian durasi, pendekatan statistik pertama kali diperkenalkan oleh Malcom et al dalam makalah yang terkenal tentang *Project Evaluation and Review Technique* (PERT). Namun, meskipun durasi penugasan merupakan variabel acak yang bebas, masalah dalam menemukan distribusi waktu penyelesaian proyek cenderung rumit dan sering tidak layak.

Beberapa teknik yang diusulkan sebelumnya, ternyata sulit untuk digunakan, terutama untuk jaringan proyek skala besar, dan tidak bisa diimplementasikan pada situasi nyata. Berdasarkan banyaknya jumlah data yang tersedia, metode yang akan diusulkan berdasar pada teknik *Multy Criteria Decision Making* (MCDM). Panjang setiap lintasan dan indikator kekritisan tambahan, dievaluasi dengan menggunakan prosedur matematis berdasarkan *Fuzzy*

Triangular Number (FTN), sedangkan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) diterapkan untuk untuk me-rangking-kan kekritisan lintasan pada suatu jaringan proyek (Zammori dkk, 2008).

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka pokok permasalahan yang dibahas dalam penulisan skripsi ini adalah bagaimana mengetahui lintasan kritis dari suatu jaringan proyek dengan menggunakan logika fuzzy, *Multy Criteria Decision Making* (MCDM) dan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)?

1.3 Batasan Masalah

Batasan masalah pada skripsi ini adalah digunakan data proyek pembangunan jalan raya yang menghubungkan dua pusat kota, meliputi durasi, biaya, maksimal presentase penurunan durasi, maksimal presentase peningkatan biaya dan nilai level kemungkinan terjadi perulangan sesuai presentase d_z .

1.4 Tujuan Penulisan

Tujuan penulisan skripsi ini adalah untuk menentukan lintasan kritis dari suatu jaringan proyek dengan menggunakan logika fuzzy, *Multy Criteria Decision Making* (MCDM) dan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS).

1.5 Manfaat Penulisan

Manfaat penulisan skripsi ini adalah:

1. Pelaksana proyek dapat mengetahui aktivitas-aktivitas yang berpotensi kritis dan yang pasti kritis pada suatu jaringan proyek.
2. Mengetahui lintasan kritis pada suatu jaringan proyek dengan memperhatikan kriteria-kriteria yang mempengaruhi berjalannya pembangunan proyek.
3. Mengetahui waktu penyelesaian proyek.

BAB II

TINJAUAN PUSTAKA

2.1 Metode PERT (Project Evaluation and Review Technique)

Metode PERT dikembangkan oleh *Navy Special Project Office* dalam kerjasama dengan perusahaan konsultasi manajemen. Dalam proyek ini, sistem PERT membantu manajemen menjawab permasalahan-permasalahan yang berkaitan dengan kapan proyek akan diselesaikan, kapan orang yang terlibat dalam proyek harus mulai dan menyelesaikan skedul kerjanya, bagian proyek mana yang harus diprioritaskan sehingga bagian yang lain berjalan sesuai rencana (Agustini dan Rahmadi, 2004).

2.1.1 Jaringan PERT

Satu syarat untuk dapat membentuk jaringan PERT adalah data urutan kegiatan proyek. Dari berbagai kegiatan yang akan dilakukan dalam suatu proyek, dapat disusun dalam bentuk jaringan PERT yang menunjukkan saling hubungan antara suatu kegiatan dengan kegiatan lain (Agustini dan Rahmadi, 2004).

Simbol-simbol yang digunakan untuk menggambarkan suatu *network* dalam PERT-type system adalah sebagai berikut:

- a) → Anak panah (*arrow*), menyatakan sebuah kegiatan atau aktifitas. Kegiatan yang dimaksud adalah kegiatan yang memerlukan *duration* (jangka waktu tertentu) dalam pemakaian sejumlah *resource* (sumber tenaga, peralatan, material, biaya). Panjang ataupun kemiringan anak panah tidak mempunyai arti apapun. Sehingga tidak perlu menggunakan skala. Kepala anak panah menjadi arah bahwa kegiatan dimulai dari permulaan dan menuju akhir.
- b) ----→ Anak panah terputus-putus, menyatakan kegiatan semu atau *dummy*. *Dummy* disini digunakan untuk membatasi mulainya kegiatan. Seperti halnya *arrow* panjang, ketebalan dan kemiringan *dummy* tidak perlu berskala. Perbedaan *dummy* dengan kegiatan biasa adalah *dummy* tidak mempunyai durasi (jangka waktu tertentu) karena tidak memakai atau menghabiskan sejumlah *resource*.

- c)  Lingkaran kecil = *node*, menentukan sebuah kejadian atau *event*. Kejadian di sini didefinisikan sebagai ujung atau pertemuan dari satu atau beberapa kegiatan.
- d)  Anak panah tebal menyatakan kegiatan pada lintasan / kegiatan kritis.

Menurut (Wijayanti, 2007), dalam pelaksanaannya simbol-simbol di atas digunakan dengan mengikuti aturan-aturan sebagai berikut:

- a) Di antara dua *event* yang sama, hanya boleh digambarkan satu anak panah.
- b) Nama suatu aktivitas dinyatakan dengan huruf atau dengan nomor *event*.
- c) Aktivitas harus mengalir dari *event* bermotor rendah ke *event* bermotor tinggi.
- d) Diagram hanya memiliki sebuah *initial event* dan sebuah *terminal event*.

2.2.2 Waktu Aktivitas

Untuk mengetahui nilai waktu yang dibutuhkan dalam menyelesaikan suatu aktivitas pada jaringan PERT, terlebih dahulu diperkirakan durasi yang dibutuhkan untuk melaksanakan proyek. Karena waktu penyelesaian proyek tidak pasti, maka diperkirakan waktu aktivitas dijabarkan oleh distribusi kemungkinan daripada menggunakan satu perkiraan saja.

Menurut (Render dkk, 2000), dalam jaringan PERT ditetapkan 3 perkiraan waktu untuk masing-masing jaringan aktivitas, meliputi:

- a) Waktu paling optimis (waktu terpendek). Waktu yang dibutuhkan oleh sebuah kegiatan jika semua hal berlangsung sesuai rencana, disimbolkan a.
- b) Waktu realistik (waktu yang paling tepat), merupakan waktu yang terletak diantara waktu optimis dan pesimis, disimbolkan b.
- c) Waktu paling pesimis (waktu terpanjang). Waktu yang dibutuhkan oleh sebuah kegiatan dengan asumsi kondisi yang ada sangat tidak diharapkan, disimbolkan c.

Setelah memperkirakan tiga waktu diatas, maka menurut (Levin dkk, 1982) dikombinasikan satu nilai waktu, menggunakan berat rata-rata:

$$t = \frac{a+4b+c}{6} \quad (2.1)$$

di mana t merupakan waktu yang diharapkan untuk tiap aktivitas, sedangkan standar deviasi dari tiap aktivitas dapat dicari dengan rumus:

$$\text{standar deviasi dari aktivitas} = \frac{c-a}{6}. \quad (2.2)$$

2.2.3 Menentukan Lintasan Kritis

Menurut Ariningtias (2008) untuk memudahkan perhitungan dalam penentuan waktu digunakan notasi sebagai berikut:

- a) $te_{(i,j)}$: *activity duration time*, yaitu waktu yang diperlukan untuk suatu aktivitas (i,j).
- b) $TE_{(i)}$: *earliest event occurrence time*, yaitu waktu paling awal peristiwa atau kejadian (i) dapat terjadi, di mana ada aturan dasar bahwa semua aktivitas dapat dimulai bila aktivitas terdahulu selesai.
- c) $TL_{(i)}$: *latest event occurrence time*, yaitu waktu yang paling akhir atau paling lambat yang masih diperbolehkan bagi suatu kejadian (i) dapat terjadi.
- d) $TF_{(i,j)}$: *total float* atau waktu *slack*, yaitu jumlah waktu yang diperkenankan suatu aktivitas boleh ditunda tanpa mempengaruhi jadwal penyelesaian proyek secara keseluruhan.
- e) $ES_{(i,j)}$: *earliest activity start time*, yaitu waktu awal tercepat dimulainya aktivitas (i,j) dan diasumsikan bahwa semua aktivitas pendahulu dimulai seawall mungkin.
- f) $EF_{(i,j)}$: *earliest activity finist time*, yaitu waktu selesai paling awal suatu aktivitas (i,j). Besarnya EF aktivitas terdahulu merupakan ES aktivitas berikutnya.
- g) $LS_{(i,j)}$: *lates activity start time*, yaitu waktu paling akhir aktivitas (i,j) boleh dimulai tanpa memperlambat waktu penyelesaian proyek secara keseluruhan.
- h) $LF_{(i,j)}$: *lates activity finish time*, yaitu waktu paling akhir suatu aktivitas (i,j) boleh diselesaikan tanpa memperlambat waktu penyelesaian proyek secara keseluruhan.

Berikut ini adalah langkah-langkah untuk menentukan lintasan kritis:

1. Perhitungan Maju (Forward Pass)

Perhitungan maju digunakan untuk mengetahui kurun waktu penyelesaian proyek tercepat. Cara perhitungan ini dimulai dari node awal ke node akhir (Taha, 1995).

Menurut Ariningtias (2008) aturan-aturan perhitungan maju dalam menyusun jaringan kerja sebagai berikut:

- Saat tercepat terjadinya kejadian awal ditentukan pada hari ke nol, sehingga berlaku:

$$TE_{(1)} = 0 \quad (2.3)$$

- Kecuali aktivitas awal, maka suatu aktivitas baru dapat dimulai bila aktivitas yang mendahulunya telah selesai.
- Waktu selesai paling awal suatu aktivitas adalah sama dengan waktu mulai paling awal ditambah kurun waktu aktivitas yang bersangkutan.

$$EF_{(i,j)} = ES_{(i,j)} + te_{(i,j)} \quad (2.4)$$

karena

$$TE_{(j)} = EF_{(i,j)} \quad (2.5)$$

maka

$$TE_{(j)} = TE_{(i,j)} + te_{(i,j)} \quad (2.6)$$

- Bila suatu aktivitas memiliki dua atau lebih aktivitas terdahulu yang menggabung, maka aktivitas tersebut sama dengan EF yang terbesar dari aktivitas terdahulu. Penjelasan tersebut di atas dapat dirumuskan sebagai berikut:

$$TE_{(j)} = \max \{EF_{(i,j)}\}. \quad (2.7)$$

2. Perhitungan Mundur (Backward Pass)

Menurut (Ariningtias, 2004), tujuan dari perhitungan mundur adalah untuk mengetahui waktu paling akhir suatu aktivitas dapat dimulai dan diakhiri tanpa menunda kurun waktu penyelesaian proyek secara keseluruhan. Cara perhitungan ini dimulai dari node akhir dan bergerak ke node awal.

Aturan-aturan perhitungan mundur dalam menyusun jaringan kerja sebagai berikut:

- Pada kejadian akhir berlaku:

$$TL_{n(j)} = TE_{n(j)} \quad (2.8)$$

- b) Waktu mulai paling akhir suatu aktivitas adalah sama dengan waktu selesai paling akhir dikurangi kurun waktu berlangsungnya aktivitas yang bersangkutan.

$$LS_{(i,j)} = LF_{(i,j)} - te_{(i,j)} \quad (2.9)$$

Karena

$$LF_{(i,j)} = TL_{(i,j)} \quad (2.10)$$

Maka

$$LS_{(i,j)} = TL_{(i,j)} - te_{(i,j)} \quad (2.11)$$

$$TL_{(i)} = LS_{(i,j)} \quad (2.12)$$

- c) Bila suatu aktivitas memecah menjadi dua atau lebih aktivitas berikutnya, maka LF aktivitas tersebut sama dengan LS aktivitas berikutnya yang terkecil, dan dapat dirumuskan sebagai berikut:

$$TL_{(i)} = \min \{LS_{(i,j)}\}. \quad (2.13)$$

3. Perhitungan Kelonggaran Waktu (*Total Float atau Slack*)

Total float atau waktu *slack* adalah jumlah waktu yang diperkenankan suatu aktivitas boleh ditunda tanpa mempengaruhi jadwal penyelesaian proyek secara keseluruhan. Jumlah waktu tersebut diperoleh jika semua aktivitas terdahulu dimulai seawal mungkin (Soeharto, 1995). *Total float* atau waktu *slack* suatu aktivitas sama dengan waktu selesai paling akhir, dikurangi waktu selesai paling awal atau maktu mulai paling akhir, dikurangi waktu mulai paling awal dari aktivitas tersebut.

$$TF_{(i,j)} = LS_{(i,j)} - ES_{(i)} = LF_{(j)} - EF_{(i,j)} \quad (2.14)$$

Total float atau waktu slack juga dapat dirumuskan sebagai waktu paling akhir terjadinya kejadian atau node akhir TL(j), dikurangi waktu paling awal terjadinya node atau kejadian sebelumnya TE(i), kurun waktu aktivitas yang bersangkutan.

$$S_{(i,j)} = TL_{(j)} - TE_{(i)} - te_{(i,j)} \quad (2.15)$$

Keuntungan utama dari total float atau waktu slack kejadian adalah membantu untuk menentukan jalur kritis. Apabila suatu aktivitas tidak mempunyai total float atau waktu slack maka aktivitas tersebut tidak dapat ditunda pelaksanaannya karena penundaan yang dilakukan akan menyebabkan penundaan waktu penyelesaian proyek secara keseluruhan. Aktivitas-aktivitas yang tidak mempunyai waktu slack disebut aktivitas kritis. Sedangkan, rangkaian aktivitas-aktivitas kritis yang menyusun jaringan kerja disebut lintasan kritis (Render dkk, 2000).

4. Perhitungan Lintasan Kritis (*Critical Path*)

Langkah yang paling penting dan merupakan bagian utama dari pengendalian proyek adalah menentukan lintasan kritis. Penentuan lintasan kritis ini dapat diketahui setelah penentuan total float atau waktu slack (Render dkk, 2000). Jalur kritis sangat penting dan memerlukan perhatian khusus karena apabila pelaksanaan dari aktivitas-aktivitas kritis ini mengalami penundaan maka akan menyebabkan keterlambatan proyek secara keseluruhan (Soeharto, 1995).

Menurut Ariningtias (2004), semua aktivitas kritis mempunyai total float atau waktu slack sebesar nol.

$$TF_{(i,j)} = S_{(i,j)} = 0 \quad (2.16)$$

Menggunakan hasil dari perhitungan maju dan perhitungan mundur, maka sebuah aktivitas (i,j) berada pada lintasan kritis bila memenuhi kondisi berikut:

$$\begin{aligned} TE_{(i)} &= TL_{(i)} \\ TE_{(j)} &= TL_{(j)} \\ TE_{(j)} - TE_{(i)} &= TL_{(j)} - TL_{(i)} = te_{(i,j)}. \end{aligned} \quad (2.17)$$

5. Probabilitas Waktu Penyelesaian Proyek

Menurut Marsudi (2008), probabilitas waktu penyelesaian proyek adalah sebagai berikut;

- Variansi untuk aktivitas

$$\sigma_{ET}^2 = \left(\frac{c-a}{6}\right)^2 \quad (2.18)$$

di mana

T = waktu penyelesaian event

TE = nilai ekspektasi dari T

σ_{TE}^2 = variansi dari T

maka

$$T \sim N(TE, \sigma_{TE}^2). \quad (2.19)$$

Catatan:

$\sigma_{TE(j)}^2$ dihitung dengan menjumlahkan σ_{ET}^2 dalam menentukan $TE(j)$.

- Peluang event T dengan jadual SD adalah

$$P(T \leq SD) = \int_{-\infty}^{SD} N(TE, \sigma_{TE}^2) dt$$

$$= \int_{-\infty}^{(SD-TE)/\sigma_{TE}} N(0,1)dz. \quad (2.20)$$

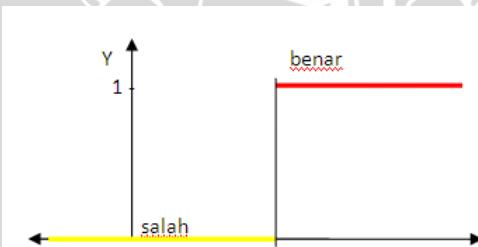
2.2 Logika Fuzzy

2.2.1 Pengertian Logika Fuzzy

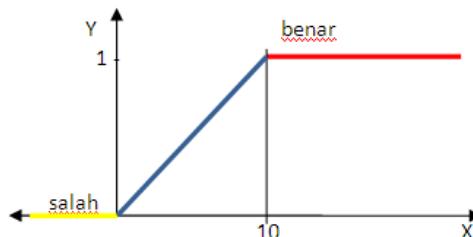
Sebelum munculnya teori logika fuzzy (*Fuzzy Logic*), dikenal sebuah logika tegas (*Crisp Logic*) yang memiliki nilai benar atau salah secara tegas. Sebaliknya Logika Fuzzy merupakan sebuah logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar dan salah. Dalam teori logika fuzzy sebuah nilai bernilai benar dan salah secara bersamaan namun berapa besar kebenaran suatu nilai tergantung kepada bobot keanggotaan yang dimiliknya (Wahyu, 2008).

2.2.2 Perbedaan Logika Fuzzy dengan Logika tegas

Menurut Wahyu (2008), perbedaan antara kedua jenis logika tersebut adalah : logika tegas memiliki nilai salah=0.0 dan benar=1.0, sedangkan logika fuzzy memiliki nilai antara 0.0 sampai 1.0. Secara grafik, perbedaan antara logika tegas dan logika fuzzy ditunjukkan oleh gambar di bawah ini:



(a) Logika Tegas

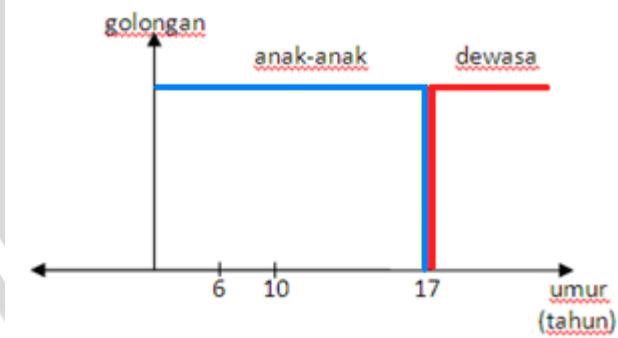


(b) Logika Fuzzy

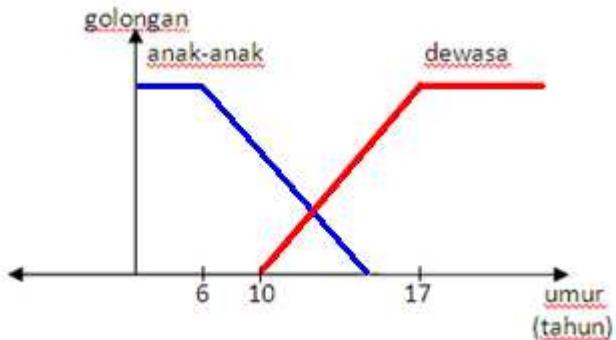
Gambar 2.1 Perbedaan Logika Tegas dan Logika Fuzzy

Di dalam Gambar 2.1.a) apabila X lebih dari atau sama dengan 10 baru dikatakan benar yaitu bernilai $Y=1$, sebaliknya nilai X yang kurang dari 10 adalah salah, yaitu $Y=0$. Maka angka 9 atau 8 atau 7 dan seterusnya adalah dikatakan salah. Didalam gambar 2.1.b) nilai $X=9$ atau 8 atau 7 atau nilai diantara 0 dan 10 adalah dikatakan ada benarnya dan ada salahnya juga.

Dalam contoh kehidupan, manusia dikatakan sudah dewasa apabila berumur lebih dari 17 tahun, maka siapapun yang kurang dari umur tersebut didalam logika tegas akan dikatakan sebagai belum dewasa atau anak-anak, sedangkan dalam hal ini pada logika fuzzy umur dibawah 17 tahun dapat saja dikategorikan dewasa tapi tidak penuh, misal untuk umur 16 tahun atau 15 tahun atau 14 tahun atau 13 tahun. Secara grafik dapat digambarkan sebagai berikut:



a) Logika tegas penentuan umur manusia



b) Logika *fuzzy* penentuan umur manusia

Gambar 2.2 Perbandingan Logika Tegas dan Logika *fuzzy* dalam penentuan golongan umur manusia

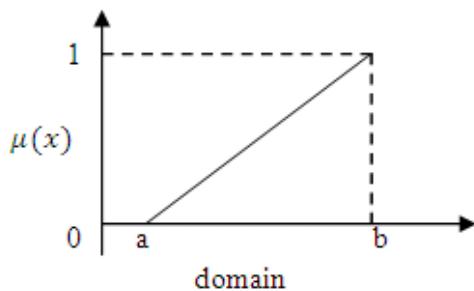
2.2.3 Fungsi keanggotaan Logika *Fuzzy*

Menurut Handayani (2010) fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam derajat keanggotannya yang memiliki interval 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan.

a. Representasi linier

Pada representasi linier, pemetaan *input* ke derajat keanggotannya digambarkan sebagai suatu garis lurus. Bentuk ini adalah bentuk yang paling sederhana. Ada dua keadaan himpunan *fuzzy* yang linier;

- Representasi linier naik yaitu kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi seperti terlihat pada Gambar 2.3.



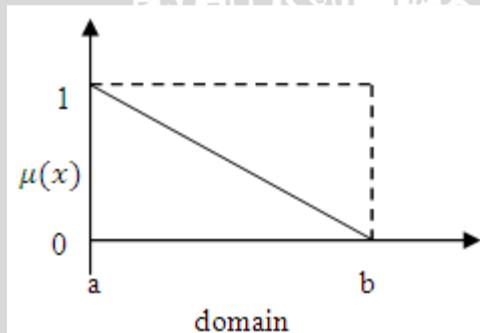
Gambar 2.3 Representasi Linier Naik

Fungsi keanggotaan untuk representasi linier naik didefinisikan sebagai

$$\mu(x) = \begin{cases} 0; & x < a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x > b \end{cases} \quad (2.21)$$

di mana $\{a, b, c\}$ merupakan FTN (*Fuzzy Triangular Number*).

- ii). Representasi linier turun merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah seperti terlihat pada Gambar 2.4.



Gambar 2.4 Representasi Linier Turun

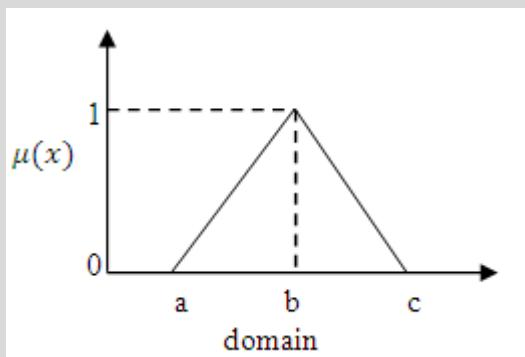
Fungsi keanggotaan untuk representasi linier turun didefinisikan sebagai

$$\mu(x) = \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x > b \end{cases} \quad (2.22)$$

di mana $\{a,b,c\}$ merupakan FTN (*Fuzzy Triangular Number*).

b. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis linier seperti terlihat pada Gambar 2.5.



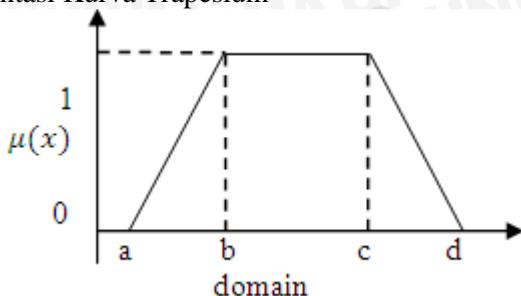
Gambar 2.5 Kurva Segitiga

Fungsi keanggotaan untuk kurva segitiga didefinisikan sebagai

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x > c \\ \frac{x-a}{b-a}; & a < x \leq b \\ \frac{c-x}{c-b}; & b < x \leq c \end{cases} \quad (2.23)$$

di mana $\{a,b,c\}$ merupakan FTN (*Fuzzy Triangular Number*).

c. Representasi Kurva Trapesium



Gambar 2.6 Kurva Trapesium

Fungsi keanggotaan untuk kurva trapesium didefinisikan sebagai

$$\mu(x) = \begin{cases} 0; & x < a \text{ atau } x > d \\ \frac{x-a}{b-a}; & a \leq x < b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c < x \leq d \end{cases} \quad (2.24)$$

di mana $\{a,b,c\}$ merupakan FTN (*Fuzzy Triangular Number*).

d. Menurut Laiskodat (2011), fungsi keanggotaan kurva-S didefinisikan sebagai berikut

$$f(x; \alpha, \beta, \gamma) = \begin{cases} 0 & x < \alpha \\ 2 \left(\frac{x-\alpha}{\gamma-\alpha} \right)^2 & \alpha \leq x \leq \beta \\ 1 - 2 \left(\frac{\gamma-x}{\gamma-\alpha} \right)^2 & \beta < x \leq \gamma \\ 1 & x > \gamma \end{cases}. \quad (2.25)$$

2.3 Jaringan Kerja Fuzzy

Menurut Zammori (2008) dalam menganalisa jaringan proyek, durasi dari tiap aktivitas digambarkan dalam bentuk *Fuzzy Triangular Number* (FTN). FTN akan dinyatakan oleh 3 nilai $\{a,b,c\}$ yang mewakili nilai lebih rendah, nilai inti dan nilai yang lebih tinggi. Secara khusus, durasi optimis merupakan waktu yang

diperlukan jika tidak ada masalah yang timbul dan jika sumber daya berusaha memenuhi target durasi. Sedemikian halnya dengan durasi yang diharapkan merupakan durasi yang biasanya penugasan akan selesai secara normal, sedangkan durasi pesimis merupakan durasi yang mencakup keselamatan untuk menangani ketidakpastian yang terlibat dalam pekerjaan itu sendiri.

Berikut merupakan rumus dari hubungan dari kegiatan pada masing-masing aktivitas:

a) Perhitungan Maju (*Forward Step*)

$$\begin{aligned} EOT_{start} &= \{0,0,0\} \\ EOT_j &= \max_{i \in P} EOT_i (+) D_{ij}. \end{aligned} \quad (2.26)$$

b) Perhitungan mundur (*Backward Step*)

$$\begin{aligned} LOT_{end} &= EOT_{end} \\ LOT_i &= \min_{j \in S} \{LOT_j (-) D_{ij}\} \end{aligned} \quad (2.27)$$

$$F_{ij} = LOT_j (-) [EOT_i (+) D_{ij}] \quad (2.28)$$

di mana,

- i dan j merupakan dua titik yang berhubungan pada suatu jaringan.
- D_{ij} merupakan durasi fuzzy pada suatu penugasan yang menghubungkan titik i dan j .
- EOT_k merupakan fuzzy *Early Occurrence Time* pada aktivitas ke- k .
- LOT_k merupakan fuzzy *Latest Occurrence Time* pada aktivitas ke- k .
- P_j merupakan himpunan kegiatan sebelum kegiatan ke- j .
- S_i merupakan himpunan kegiatan setelah kegiatan ke- i .
- $(+), (-), \min, \max$ masing-masing merupakan penjumlahan, pengurangan, minimum, maksimum.

FTN dapat digunakan dalam operasi matematika dengan cara yang sedehana. Pengembangan fuzzy dari aritmatika dasar operasi pada FTN didefinisikan sebagai berikut:

- Penjumlahan fuzzy: $A_1 (+) A_2 = (a_1 + a_2, b_1 + b_2, c_1 + c_2)$
- Pengurangan fuzzy: $A_1 (-) A_2 = (a_1 - c_2, b_1 - b_2, c_1 - a_2)$

- c) Perkalian skalar: $k \times A_1 = (k \times a_1, k \times b_1, k \times c_1), k \in R$
- d) Perkalian fuzzy: $A_1(\times)A_2 = (a_1 \times a_2, b_1 \times b_2, c_1 \times c_2)$
- e) Pembagian fuzzy: $A_1(/)A_2 = \left(\frac{a_1}{c_2}, \frac{b_1}{b_2}, \frac{c_1}{a_2} \right)$
- f) Invers fuzzy: $(A_1)^{-1} = \left(\frac{1}{c_1}, \frac{1}{b_1}, \frac{1}{a_1} \right)$
- g) Rata-rata fuzzy: $\tilde{F}\{A_1, A_2, \dots, A_n\} = \left(\frac{\sum_{i=1}^n a_i}{n}, \frac{\sum_{i=1}^n b_i}{n}, \frac{\sum_{i=1}^n c_i}{n} \right)$

di mana A_1, A_2, \dots, A_n merupakan FTN.

Tidak ada jaminan bahwa batas maksimum (minimum) dari himpunan FTN mempertahankan bentuk segitiga. Untuk mengatasi kekurangan tersebut, terdapat rumus pendekatan yang berhasil diterapkan pada jaringan proyek. Oleh karena itu, digunakan rumus sebagai berikut:

$$\widehat{\max}(A_1, A_2) \equiv F_{\max} \equiv \{\max(a_1, a_2), \max(b_1, b_2), \max(c_1, c_2)\} \quad (2.29)$$

$$\widehat{\min}(A_1, A_2) \equiv F_{\min} \equiv \{\min(a_1, a_2), \min(b_1, b_2), \min(c_1, c_2)\} \quad (2.30)$$

di mana $A_1 \equiv \{a_1, b_1, c_1\}$ dan $A_2 \equiv \{a_2, b_2, c_2\}$ adalah dua FTN.

F_{ij} merupakan FTN yang mewakili nilai float pada suatu aktivitas T_{ij} di mana nilai tengahnya selalu lebih besar atau sama dengan nol. Jika aktivitas T_{ij} ‘pasti kritis’, maka nilai F_{ij}^* merupakan simetris FTN dengan nilai inti sama dengan nol, sehingga saat defuzzifikasi (proses mengubah bilangan fuzzy untuk mendapatkan kembali bentuk tegasnya) menggunakan pendekatan defuzzifikasi dasar *Graded Mean Interaction*:

$$GM(F) = \frac{a+4b+c}{6} \quad (2.31)$$

F_{ij}^* akan selalu sama dengan nol. Jika aktivitas T_{ij} hanya ‘berpotensi kritis’ maka waktu slack F_{ij}^* merupakan asimetrik FTN dan bagian positif selalu lebih besar dari bagian negatifnya.

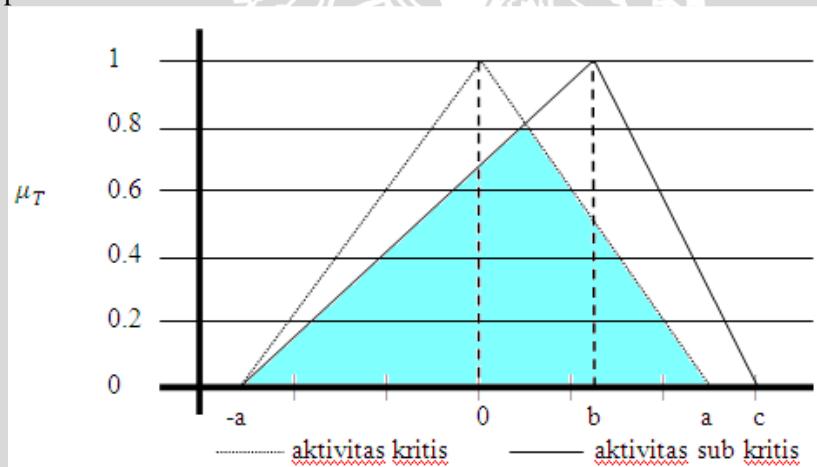
Nilai *float* atau *slack* merupakan jumlah waktu penyelesaian suatu aktivitas yang dapat diundur tanpa mempengaruhi saat paling cepat dari penyelesaian proyek secara keseluruhan. Pada kenyataannya, dalam menentukan kekritisan float (Float Critically), diberikan dua *Zero fuzzy Triangular Number* (ZTN) seperti $F_1 \equiv \{-a, b, c\}$ dan $F_2 \equiv \{-a, b, d\}$ yang keduanya memiliki derajat kekritisan yang

hampir sama. Jika d lebih besar nilainya dibandingkan dengan c , maka F_1 derajat kekritisannya lebih besar daripada F_2 . Untuk mengatasi masalah ini kita bisa mengambil keuntungan dari kondisi di mana nilai float dari aktivitas kritis adalah ZTN yang simetris, sehingga, jika $F_{ij} \equiv \{-a, b, c\}$ merupakan waktu float dari aktivitas T_{ij} , kekritisan dapat dihitung dengan membandingkan dengan $F_{ij}^* \equiv \{-a, 0, a\}$, di mana *float* dari ‘kekritisannya pasti’ yang sesuai dari aktivitas T_{ij} .

Atas dasar pertimbangan tersebut, Kekritisian *Float* (*Float Criticality*) dapat dinyatakan sebagai:

$$FC_{ij} = \begin{cases} 0 & \text{jika } \mu_{F_{ij}}(0) = 0 \\ \frac{\text{Area}(F_{ij}^* \cap F_{ij})}{\text{Area}(F_{ij})} & \text{jika } \mu_{F_{ij}}(0) > 0. \end{cases} \quad (2.32)$$

Float Criticality akan bernilai nol jika F_{ij} merupakan positif FTN, dan *Float Criticality* akan bernilai satu jika F_{ij} merupakan simetrik ZTN. Sebuah gambaran penjelasan dari persamaan (2.32) diberikan pada Gambar 2.7.



Gambar 2.7 Kekritisian *Float* (FC_{ij})

Untuk mengetahui fungsi keanggotaan dari kekritisan *float* (FC_{ij}) seperti yang tertera pada persamaan (2.32), maka pertama-tama perlu dihitung luas $\text{Area}(F_{ij}^* \cap F_{ij})$ menggunakan fungsi keanggotaan kurva triangular pada persamaan (2.24):

$$\mu_F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c+x}{c-b} & b < x \leq c \\ 1 & x > c \end{cases} \quad (2.33)$$

di mana $\{a,b,c\}$ merupakan *Fuzzy Triangular Number* (FTN), sehingga fungsi keanggotaan dari nilai float $F_{ij}^* \equiv \{-a, 0, a\}$ adalah:

$$\mu_{F^*}(x) = \begin{cases} 0 & x < -a \text{ dan } x > a \\ \frac{x+a}{a} & -a \leq x \leq 0 \\ \frac{a-x}{a} & 0 < x \leq a \end{cases} \quad (2.34)$$

Dan fungsi keanggotaan dari nilai float $F_{ij} \equiv \{-a, b, c\}$ adalah:

$$\mu_F(x) = \begin{cases} 0 & x < -a \text{ dan } x > c \\ \frac{x+a}{b+a} & -a \leq x \leq b \\ \frac{c-x}{c-b} & b < x \leq c \end{cases} \quad (2.35)$$

Untuk mengetahui luas segitiga $Area(F_{ij}^* \cap F_{ij})$, maka harus diketahui panjang alas dan tinggi dari area segitiga tersebut. Dari Gambar 2.7 bisa diketahui panjang alas segitiga yang diarsir adalah $2a$, sedangkan tinggi dari area yang diarsir adalah nilai dari fungsi keanggotaan pada nilai x yang merupakan perpotongan antara sisi segitiga F_{ij}^* yang nilai x -nya pada selang $0 \leq x \leq a$, dan sisi segitiga F_{ij} yang nilai x -nya pada selang $-a \leq x \leq b$.

Karena nilai x dari perpotongan sisi F_{ij}^* dan F_{ij} bernilai sama, maka nilai x dari perpotongan nilai kedua sisi tersebut adalah:

$$\begin{aligned} \frac{a-x}{a} &= \frac{x+a}{b+a} \\ (a-x)(b+a) &= a(x+a) \\ ab + a^2 - bx - ax &= ax + a^2 \\ ab &= bx + 2ax \\ ab &= x(b+2a) \\ x &= \frac{ab}{b+2a} \end{aligned} \quad (2.36)$$

Dari nilai x yang telah diketahui pada persamaan (2.36) dapat diketahui nilai fungsi keanggotaan yang merupakan tinggi dari segitiga yang diarsir. Berikut adalah perhitungan nilai fungsi keanggotaan dari x :

$$\mu_{F^*}(x) = \frac{a-x}{a}$$

$$\begin{aligned}
\frac{a-x}{a} &= \frac{a - \frac{ab}{b+2a}}{a} \\
&= \frac{\frac{a(b+a) - ab}{b+2a}}{a} \\
&= \frac{\frac{ab + 2a^2 - ab}{b+2a}}{a} \\
&= \frac{2a^2}{a(b+a)} = \frac{2a}{(b+2a)}. \tag{2.37}
\end{aligned}$$

Dari persamaan (2.37) dapat diketahui tinggi dari segitiga $\text{Area}(F_{ij}^* \cap F_{ij})$ dengan panjang alas $2a$ sehingga dapat dihitung luas segitiga yang diarsir adalah sebagai berikut:

$$\begin{aligned}
\text{Area}(F_{ij}^* \cap F_{ij}) &= \frac{1}{2} 2a \frac{2a}{(b+2a)} \\
&= \frac{2a^2}{(b+2a)} \tag{2.38}
\end{aligned}$$

di mana luas $\text{Area}(F_{ij})$ adalah:

$$\begin{aligned}
\text{Area}(F_{ij}) &= \frac{1}{2} (a+c) 1 \\
&= \frac{1}{2} (a+c) \tag{2.39}
\end{aligned}$$

Dari persamaan (2.38) dan persamaan (2.39) dapat diketahui nilai kekritisan float jika $\mu_{F_{ij}}(0) > 0$ adalah:

$$\begin{aligned}
\frac{\text{Area}(F_{ij}^* \cap F_{ij})}{\text{Area}(F_{ij})} &= \frac{\frac{2a^2}{b+2a}}{\frac{1}{2}(a+c)} \\
&= \frac{2a^2}{\frac{1}{2}(a+c)(b+2a)} \\
&= \frac{4a^2}{(a+c)(b+2a)} \tag{2.40}
\end{aligned}$$

sehingga Kekritisan Float (Float Criticality) dapat dinyatakan menjadi:

$$FC_{ij} = \begin{cases} 0 & \text{jika } \mu_{F_{ij}}(0) = 0 \\ \frac{\text{Area}(F_{ij}^* \cap F_{ij})}{\text{Area}(F_{ij})} = \frac{4a^2}{(a+c)(b+2a)} & \text{jika } \mu_{F_{ij}}(0) > 0 \end{cases} \quad (2.41)$$

di mana $\{a, b, c\}$ merupakan *Fuzzy Triangular Number* (FTN).

2.4 Kriteria Keputusan dengan Menggunakan *Multi Criteria Decision Making* (MCDM)

Menurut (Zammori, 2008), untuk menerapkan MCDM, perlu mengidentifikasi semua alternatif kelayakan dan criteria keputusan untuk membedakan tiap kriteria keputusan tersebut. Berikut adalah alternatif berkaitan dengan lintasan proyek, kriteria keputusan diperkenalkan sebagai berikut:

2.4.1 Panjang Lintasan / *Path Length* (PL)

Durasi dan variabilitas dari lintasan biasanya digunakan untuk mengidentifikasi lintasan kritis. Karena FTN telah mengandung informasi mengenai durasi dan variabilitas, maka evaluasinya sangat mudah dan hanya dapat diperoleh dari menjumlahkan durasi kegiatan dari lintasan proyek, yaitu:

$$PL_k = \sum_{ij \in k} D_{ij} = \left\{ \sum_{ij \in k} a_{ij}, \sum_{ij \in k} b_{ij}, \sum_{ij \in k} c_{ij} \right\} \quad (2.42)$$

di mana $D_{ij} \equiv (a_{ij}, b_{ij}, c_{ij})$ merupakan durasi dari penugasan umum pada lintasan ke- k .

Misalkan $PL_k \equiv \{a_k, b_k, c_k\}$ adalah panjang lintasan fuzzy pada lintasan ke- k dan $P_{CT} \equiv \{a_{CT}, b_{CT}, c_{CT}\}$ merupakan waktu penyelesaian proyek fuzzy. Karena lintasan kritis nyata akan didefinisikan dengan penerapan MCDM pada bilangan fuzzy, maka perlu mendefinisikan kekritisannya lintasan ke- k atau *Path Criticality* (PCr) sebagai FTN juga, seperti persamaan berikut:

$$PCr_k = \frac{PL_k}{P_{CT}} = \left\{ \frac{a_k}{c_{CT}}, \frac{b_k}{b_{CT}}, \frac{c_k}{a_{CT}} \right\}. \quad (2.43)$$

2.4.2 Jumlah Biaya / *Expected Path Cost* (PC)

Biaya penyelesaian yang diharapkan merupakan parameter penting lain yang harus dipertimbangkan. FTN dapat digunakan dengan efektif untuk mewakili keseluruhan biaya setiap kegiatan.

Selanjutnya, biaya dari lintasan ke- k dapat dengan mudah dievaluasi dengan rumus sebagai berikut;

$$PC_k = \sum_{ij \in k} C_{ij} = \{\sum_{ij \in k} a_{ij}, \sum_{ij \in k} b_{ij}, \sum_{ij \in k} c_{ij}\} \quad (2.44)$$

di mana $C_{ij} \equiv (a_{ij}, b_{ij}, c_{ij})$ merupakan biaya dari penugasan umum pada lintasan ke- k .

2.4.3 Perlindungan yang Tersedia / Available Protection (AP)

Dalam beberapa kasus, dimungkinkan untuk meningkatkan biaya dari penugasan untuk mengurangi durasinya, dengan tujuan tambahan dan/atau alternatif sumber daya (seperti lembur, lebih banyak pekerja, tim proyek yang berdedikasi, *outsourcing*, dll). Kesempatan ini sering digunakan untuk mempercepat penyelesaian proyek dan untuk mendapatkan dorongan yang khusus ditawarkan jika proyek selesai sebelum tanggal target. Bahkan ketika proyek ini terlambat, manajer proyek dapat memutuskan untuk melindungi waktu jatuh tempo dengan mempercepat kegiatan pada lintasan kritis. Oleh karena itu, tersedia *Available Protection* (AP) pada setiap lintasan yang harus digunakan sebagai parameter tambahan untuk mengidentifikasi lintasan kritis yang sebenarnya. Nilai parameter AP dapat dihitung menggunakan persamaan berikut:

$$AP_k = \sum_{ij \in k} (\Delta D_{ij} / \Delta C_{ij}) \quad (2.45)$$

di mana ΔD_{ij} dan ΔC_{ij} merupakan FTN mewakili pengurangan maksimum dari waktu dan koresponden peningkatan biaya untuk penugasan T_{ij} sepanjang lintasan ke- k .

Seperti proyek biasanya, untuk mengevaluasi ΔD_{ij} dan ΔC_{ij} dibutuhkan evaluasi dua parameter tegas p_{ij} dan c_{ij} yang mewakili masing-masing maksimum persen pengurangan waktu dan koresponden persen peningkatan biaya.

Selanjutnya ΔD_{ij} dan ΔC_{ij} diperoleh dengan:

$$\Delta C_{ij} = C_{ij} \times c_{ij} \quad (2.46)$$

$$\Delta D_{ij} = (D_{ij} \times p_{ij} (+) \tilde{F}_{ij}) \times w_{ij} \quad (2.47)$$

di mana ΔC_{ij} adalah biaya *fuzzy*, ΔD_{ij} adalah durasi yang diharapkan, \tilde{F}_{ij} adalah waktu *float* yang dikoreksi dan w_{ij} adalah parameter bobot, sedangkan, w_{ij} dapat didefinisikan sebagai:

$$w_{ij} = \left[(\sum_{ab \in k^*} D_{ab}) (+) \left(\frac{1}{2} \times D_{ij} \right) \right] (/) PL_k \quad (2.48)$$

di mana k^* dinotasikan sebagai bagian dari lintasan dari semua aktivitas yang mendahului T_{ij} pada lintasan ke- k .

Namun jika waktu *slack* merupakan ZTN, diperlukan sebuah koreksi nilai *slack*. Sebuah ZTN memiliki bagian positif dan bagian negatif juga, tetapi *Availeble Protection* (AP) hanya tersedia jika waktu *slack* adalah positif FTN. Oleh karena itu hanya bagian positif dari F_{ij} yang akan dihitung, bobot dari kemungkinan F_{ij} menjadi positif. Kuantitas ini yang dapat dinotasikan sebagai ‘*float* dikoreksi’ \tilde{F}_{ij} dapat dihitung dengan persamaan:

$$\tilde{F}_{ij} = F_{ij}^+ \times (1 - FC_{ij}) \quad (2.49)$$

di mana $(1 - FC_{ij})$ merupakan bilangan tegas yang dinyatakan sebagai kemungkinan dari F_{ij} menjadi positif, dan F_{ij}^+ adalah bagian positif dari F_{ij} . Selanjutnya F_{ij}^+ dapat diperkirakan menggunakan persamaan (2.50):

$$F_{ij}^+ \equiv \begin{cases} \{0, b, c\} & \text{jika } F_{ij} \equiv \{-a, b, c\} \\ \{a, b, c\} & \text{selainnya} \end{cases} \quad (2.50)$$

2.4.4 Risk of Major Design Revision (RMR)

Terkadang masalah muncul pada akhir suatu penugasan, dikarenakan kesalahan pada hasil *output*, hal ini menyebabkan kegiatan berikutnya tidak dapat dimulai. Karena itu, orang yang menangani proyek harus melindungi proyek dari pengulangan aktivitas, menganjurkan penyelasaian komunikasi formal dan informal antar sumberdaya, dan mengembangkan sebuah sistem peringatan awal. Pada lingkup ini, terdapat indeks yang berdasar pada rumus yang umum digunakan dalam penilaian risiko:

$$Risk = Probability \times effects$$

Jika diasumsikan bahwa, (i) tidak lebih dari satu aktivitas dapat dimulai kembali saat terjadi pengulangan dan (ii) kemungkinan untuk memulai kembali aktivitas yang sama dua kali dapat hampir

diabaikan, *Recycling Critically* (RC) dari penugasan T_{ij} dapat didefinisikan sebagai:

$$RC_{ij} = \sum_z (p_z \times d_z \times D_{ij}) - \tilde{F}_{ij} \quad (2.51)$$

di mana p_z merupakan level kemungkinan saat pengulangan terjadi sesuai presentase d_z pada penugasan T_{ij} yang harus diulang, dan $\sum_z p_z = 1$.

Persamaan (2.52) menegaskan bahwa kekritisan sama dengan waktu tambahan yang diharapkan (karena aktivitas daur ulang dipotong dengan waktu float yang dikoreksi yang sebetulnya tersedia untuk aktivitas T_{ij} yang sesuai. Dari RC, *Risk of Major Design Revision* (RMR) dari lintasan ke-k akhirnya dapat dijabarkan menjadi:

$$RMR_k = \sum_{ij \in k} (RC_{ij}). \quad (2.52)$$

2.5 Identifikasi Kekritisan dengan Menggunakan Metode *Fuzzy Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)

Menurut Huang dkk (2010) Untuk menentukan peringkat kekritisan lintasan, terlebih dahulu didefinisikan indeks-indeks yang digabungkan menggunakan teknik MCDM. Untuk bidang ini, TOPSIS dipilih untuk mempermudah dalam perluasan kasus *fuzzy* dan juga mempermudah perhitungan. TOPSIS merupakan alat dari MCDM yang didasarkan asumsi bahwa diberikan suatu himpunan alternatif yang layak, di mana yang terbaik adalah yang paling dekat dengan solusi ideal-positif dan paling jauh dari solusi ideal negatif. Solusi positif-ideal merupakan solusi yang memaksimalkan kriteria keuntungan dan meminimumkan criteria biaya sedangkan solusi ideal-negatif memaksimalkan kriteria biaya dan meminimumkan kriteria keuntungan.

Misalkan masalah MCDM mempunyai n alternatif, (A_1, \dots, A_n) dan m kriteria keputusan (C_1, \dots, C_m) . Untuk mengetahui peringkat dari n alternatif, TOPSIS membutuhkan inputan berupa matriks $X_{nm} = [x_{ij}]$, yang berisi evaluasi yang diberikan oleh alternatif ke- i

dengan parameter keputusan j dan vektor $W_m[w_j]$ yang mempengaruhi bobot yang diberikan pada masing-masing dari m parameter keputusan.

Kemudian proses *fuzzy TOPSIS* diselesaikan dengan tahapan berikut:

- Tahap 1: Bentuk matriks keputusan normalisasi $\bar{R}_{nm} = [\bar{r}_{ij}]$

Matriks keputusan dinormalkan dengan rumus berikut:

$$\tilde{r}_{ij} = \frac{\tilde{x}_{ij} - \min_i(\tilde{x}_{ij})}{\max_i(\tilde{x}_{ij}) - \min_i(\tilde{x}_{ij})}, \quad (2.53)$$

sehingga akan diperoleh matriks keputusan normalisasi \tilde{R} :

$$\tilde{R} = (\tilde{r}_{ij})_{n \times m} = \begin{bmatrix} \tilde{r}_{11} & \cdots & \tilde{r}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{r}_{n1} & \cdots & \tilde{r}_{nm} \end{bmatrix}. \quad (2.54)$$

- Tahap 2: Bentuk bobot matriks keputusan $\tilde{V}_{nm} = [\tilde{v}_{ij}]$

Setiap elemen dari matriks keputusan normal dikoreksi menggunakan bobot yang sesuai w_j :

$$\tilde{v}_{ij} = \tilde{w}_j \times \tilde{r}_{ij} \quad (2.55)$$

di mana \tilde{w}_j merupakan bobot vector m dari indeks atribut keputusan.

Oleh karena itu akan diperoleh matriks \tilde{V}_{nm} :

$$\tilde{V}_{nm} = \begin{bmatrix} \tilde{v}_{11} & \cdots & \tilde{v}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{v}_{n1} & \cdots & \tilde{v}_{nm} \end{bmatrix}. \quad (2.56)$$

- Tahap 3: Tetapkan solusi ideal positif dan ideal negatif

Untuk matriks keputusan normalisasi, solusi Ideal positif (\tilde{v}^+) dan negative (\tilde{v}^-) dapat dinyatakan menjadi dua dimensi- n yang didefinisikan sebagai berikut:

$$\tilde{v}^+ = \{\tilde{v}_1^+, \tilde{v}_2^+, \dots, \tilde{v}_n^+\} \quad (2.57)$$

$$\tilde{v}^- = \{\tilde{v}_1^-, \tilde{v}_2^-, \dots, \tilde{v}_n^-\} \quad (2.58)$$

Jadi, solusi Ideal positif (\tilde{v}^+) dan negative (\tilde{v}^-) yang diberi bobot dan dinormalkan dapat dihitung dengan:

$$\tilde{v}_i^+ = (\tilde{v}_1^+, \tilde{v}_2^+, \dots, \tilde{v}_m^+)$$

$$\tilde{v}_i^- = \tilde{w}_1 \times \tilde{r}_1^+, \tilde{w}_2 \times \tilde{r}_2^+, \dots, \tilde{w}_m \times \tilde{r}_m^+ \quad (2.59)$$

$$\begin{aligned}\tilde{v}_i^- &= (\tilde{v}_1^-, \tilde{v}_2^-, \dots, \tilde{v}_m^-) \\ \tilde{v}_i^- &= \tilde{w}_1 \times \tilde{r}_1^-, \tilde{w}_2 \times \tilde{r}_2^-, \dots, \tilde{w}_m \times \tilde{r}_m^-\end{aligned}\quad (2.60)$$

di mana

$$r^+ = (r_1^+, r_2^+, \dots, r_m^+) = ((1,1,1), (1,1,1), \dots, (1,1,1))_{1 \times m}$$

$$r^- = (r_1^-, r_2^-, \dots, r_m^-) = ((0,0,0), (0,0,0), \dots, (0,0,0))_{1 \times m}.$$

- Tahap 4: Mengukur jarak antara solusi alternatif dan solusi ideal. Untuk alternatif umum, jarak dapat dengan mudah diperoleh dengan menerapkan *Euler distance*:

$$D_i^+ = D(\tilde{v}^+, \tilde{v}_i) = \sqrt{\sum_{j=1}^m [d(\tilde{v}_j^+, \tilde{v}_{ij})]^2} \quad (2.61)$$

$$D_i^- = D(\tilde{v}^-, \tilde{v}_i) = \sqrt{\sum_{j=1}^m [d(\tilde{v}_j^-, \tilde{v}_{ij})]^2} \quad (2.62)$$

di mana

$$d(\tilde{A}_1, \tilde{A}_2) = \sqrt{\frac{1}{3} [(a_2 - a_1)^2 + (b_2 - b_1)^2 + (c_2 - c_1)^2]}. \quad (2.63)$$

- Tahap 5: Menentukan ranking akhir dari solusi.

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}. \quad (2.64)$$

Menurut derajat hubungan pendekatan C_i , dapat dirankingkan pilihan urutan dari semua alternatif (alternatif yang memuat nilai C_i yang lebih besar merupakan alternatif yang paling kritis).

UNIVERSITAS BRAWIJAYA



BAB III

PEMBAHASAN

3.1 Algoritma Dalam Menentukan Lintasan Kritis.

Dalam menentukan lintasan kritis pada suatu jaringan proyek digunakan langkah-langkah sebagai berikut:

1. Menghitung waktu luang suatu penugasan.
 - a) Menyusun jaringan kerja berdasarkan daftar rencana kegiatan pembangunan jalan raya yang disajikan pada Tabel 3.1.
 - b) Menentukan perhitungan maju dan perhitungan mundur menggunakan operasi *fuzzy* menggunakan persamaan (2.26) dan (2.27).
 - c) Menentukan perhitungan kelonggaran waktu (*Float Time*) menggunakan persamaan (2.28).
 - d) Menentukan perhitungan kelonggaran waktu positif (Positif *Float Time*) menggunakan persamaan (2.49).
 - e) Menentukan perhitungan kekritisan kelonggaran waktu (*Float Criticality*) menggunakan persamaan (2.40).
 - f) Menentukan perhitungan kelonggaran waktu yang dikoreksi (*Corrected Float Time*) menggunakan persamaan (2.28).
2. Menentukan kriteria keputusan menggunakan teknik *Multy Kriteria Decision Making* (MCDM).
 - a) Mengidentifikasi semua alternatif yang mungkin dalam lintasan proyek. Pada metode ini digunakan 5 kriteria dalam membuat keputusan.
 - b) Menghitung nilai masing-masing kriteria keputusan pada tiap lintasan proyek. Kelima kriteria tersebut dapat dihitung menggunakan persamaan (2.41), (2.42), (2.43), (2.44), dan (2.51).
3. Merangkingkan kekritisan tiap lintasan menggunakan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS).

3.2 Data Umum Proyek

Sebagai ilustrasi numerik untuk menentukan lintasan kritis diambil contoh penerapan studi kasus dari Zammori dkk (2008) tentang proyek pembangunan jalan raya yang menghubungkan 2

pusat perkotaan. Terdapat 3 kegiatan utama dalam pelaksanaan proyek pembangunan jalan raya, yang meliputi:

- a. Membangun papan jalan.
- b. Membangun 2 jembatan di atas jalan.
- c. Mengaspal jalan raya.

Beberapa pekerjaan tambahan juga diperlukan untuk menstabilkan pekerjaan yang berat dan membangun infrastuktur guna mengantisipasi hujan yang mungkin terjadi. Penugasan secara rinci disajikan pada Tabel 3.1, Tabel 3.2, Tabel 3.3 dan Tabel 3.4 berikut;

3.2.1 Data Urutan Aktivitas

Berikut ini adalah tabel urutan aktivitas pada jaringan proyek pembuatan jalan raya;

Tabel 3.1 Daftar Urutan Jenis Aktivitas dan aktivitas sebelumnya

NO	Aktivitas	node	ID	Aktivitas sebelumnya
1	Pemasangan tempat konstruksi	[1,2]	d1	-
2	Menstabilkan lereng yang curam	[2,3]	d2	d1
3	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	d1
4	Pemotonga jalan pada jalur pertama	[3,7]	d4	d2
5	Membuat tanggul pada jalur pertama	[5,6]	d5	d3
6	Pekerjaan tambahan pada jalur kedua	[5,8]	d6	d3
7	Aktivitas akhir pada jalur pertama	[7,10]	d7	d5,d5
8	Memasang papan jalan pada jalur pertama	[7,11]	d8	d4,d5
9	Membuat tanggul pada jalur kedua	[8,9]	d9	d6,d5
10	Kegiatan akhir pada jalur kedua	[10,22]	d10	d7,d9
11	Aktivitas akhir pada jalur kedua	[11,21]	d11	d8,d9

12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	d12	d1
13	Memasang tiang pancang pada jalur pertama	[12,13]	d13	d12
14	Memasang palang pada jalur pertama	[13,14]	d14	d13
15	Aktivitas akhir pada jembatan pertama	[14,15]	d15	d14
16	Penundaan	[4,16]	d16	d1
17	Pondasi pada jembatan kedua	[16,17]	d17	d16
18	Memasang tiang pancang jalur kedua	[17,18]	d18	d17
19	Memasang palang pada jalur kedua	[18,19]	d19	d18
20	Aktivitas akhir pada jembatan kedua	[19,20]	d20	d19
21	Pengaspalan jalan	[21,22]	d21	d11,d15,d20
22	Penutupan tempat konstruksi	[22,23]	d22	d16,d21

Sumber: Zammori (2008).

3.2.2 Data Perencanaan Durasi dan Biaya Proyek untuk Setiap Aktivitas

Berikut ini adalah tabel urutan durasi dan biaya aktivitas pada jaringan proyek pembuatan jalan raya;

Tabel 3.2 Daftar Durasi dan Biaya Masing-masing Aktivitas

NO	Aktivitas	node	Durasi	Biaya
1	Pemasangan tempat konstruksi	[1,2]	{12, 15,17}	{7,9,11}
2	Menstabilkan lereng yang curam	[2,3]	{23, 25,28}	{145, 150,153}
3	Pekerjaan tambahan pada jalur pertama	[2,5]	{23, 24,25}	{87, 95,104}
4	Pemotonga jalan pada jalur pertama	[3,7]	{27, 30,31}	{229, 234,240}

5	Membuat tanggul pada jalur pertama	[5,6]	{30, 32,33}	{317, 323,327}
6	Pekerjaan tambahan pada jalur kedua	[5,8]	{8, 9,11}	{28, 31,37}
7	Aktivitas akhir pada jalur pertama	[7,10]	{21, 24,25}	{73, 77,80}
8	Memasang papan jalan pada jalur pertama	[7,11]	{20, 22,26}	{275,280,285}
9	Membuat tanggul pada jalur kedua	[8,9]	{22, 24,25}	{198, 205,215}
10	Kegiatan akhir pada jalur kedua	[10,22]	{16, 17,19}	{36, 43,47}
11	Aktivitas akhir pada jalur kedua	[11,21]	{3, 5,6}	{145, 151,157}
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	{34, 35,39}	{357, 363,370}
13	Memasang tiang pancang pada jalur pertama	[12,13]	{19, 22,23}	{334, 340,345}
14	Memasang palang pada jalur pertama	[13,14]	{12, 15,18}	{124, 127,133}
15	Aktivitas akhir pada jembatan pertama	[14,15]	{7, 10,12}	{77, 83,88}
16	Penundaan	[4,16]	{34, 35,39}	{-}
17	Pondasi pada jembatan kedua	[16,17]	{18, 19,21}	{267, 270,275}
18	Memasang tiang pancang jalur kedua	[17,18]	{12, 13,14}	{145, 152,156}
19	Memasang palang pada jalur kedua	[18,19]	{7, 8,11}	{67, 74,78}
20	Aktivitas akhir pada jembatan kedua	[19,20]	{3, 4,6}	{35, 38,43}
21	Pengaspalan jalan	[21,22]	{12, 15,17}	{35, 38,43}
22	Penutupan tempat konstruksi	[22,23]	{9, 12,14}	{5, 7,8}

Sumber: Zammori (2008).

3.2.3 Data Maksimal Presentase Penurunan Durasi dan Peningkatan Biaya untuk Setiap Aktivitas

Berikut ini adalah tabel urutan presentase maksimal penurunan durasi dan maksimal peningkatan biaya masing-masing aktivitas pada jaringan proyek pembuatan jalan raya;

Tabel 3.3 Daftar Maksimal Presentase Penurunan Durasi (p_{ij}) dan Peningkatan Biaya (c_{ij}) pada Masing-masing Aktivitas

NO	Aktivitas	node	p_{ij}	c_{ij}
1	Pemasangan tempat konstruksi	[1,2]	50%	60%
2	Menstabilkan lereng yang curam	[2,3]	35%	45%
3	Pekerjaan tambahan pada jalur pertama	[2,5]	27%	30%
4	Pemotonga jalan pada jalur pertama	[3,7]	30%	35%
5	Membuat tanggul pada jalur pertama	[5,6]	40%	50%
6	Pekerjaan tambahan pada jalur kedua	[5,8]	28%	35%
7	Aktivitas akhir pada jalur pertama	[7,10]	20%	35%
8	Memasang papan jalan pada jalur pertama	[7,11]	28%	37%
9	Membuat tanggul pada jalur kedua	[8,9]	35%	40%
10	Kegiatan akhir pada jalur kedua	[10,22]	25%	34%
11	Aktivitas akhir pada jalur kedua	[11,21]	35%	35%
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	38%	45%
13	Memasang tiang pancang pada jalur pertama	[12,13]	30%	40%
14	Memasang palang pada jalur pertama	[13,14]	27%	35%

15	Aktivitas akhir pada jembatan pertama	[14,15]	26%	32%
16	Penundaan	[4,16]	[-]	[-]
17	Pondasi pada jembatan kedua	[16,17]	40%	50%
18	Memasang tiang pancang jalur kedua	[17,18]	29%	35%
19	Memasang palang pada jalur kedua	[18,19]	18%	23%
20	Aktivitas akhir pada jembatan kedua	[19,20]	35%	40%
21	Pengaspalan jalan	[21,22]	23%	34%
22	Penutupan tempat konstruksi	[22,23]	60%	60%

Sumber: Zammori (2008).

3.2.4 Data Level Kemungkinan Jika Terjadi Perulangan

Aktivitas Sesuai Presentase dari Suatu Aktivitas Harus Diulang

Berikut ini adalah tabel level kemungkinan terjadi perulangan sesuai presentase pada masing-masing aktivitas pada jaringan proyek pembuatan jalan raya;

Tabel 3.4 Daftar Level Kemungkinan (p_z) Jika Terjadi Perulangan Sesuai dengan Presentase (d_z) dari Aktivitas (T_{ij}) yang Harus Diulang

NO	Aktivitas	node	$p_z \rightarrow d_z$
1	Pemasangan tempat konstruksi	[1,2]	$[0.55 \rightarrow 0; 0.05 \rightarrow 1; 0.15 \rightarrow 0.5; 0.2 \rightarrow 0.33]$
2	Menstabilkan lereng yang curam	[2,3]	[-]
3	Pekerjaan tambahan pada jalur pertama	[2,5]	$[0.5 \rightarrow 0; 0.03 \rightarrow 1; 0.17 \rightarrow 0.5; 0.3 \rightarrow 0.33]$
4	Pemotonga jalan pada jalur pertama	[3,7]	$[0.6 \rightarrow 0; 0.05 \rightarrow 1; 0.15 \rightarrow 0.5; 0.2 \rightarrow 0.33]$
5	Membuat tanggul pada jalur pertama	[5,6]	$[0.45 \rightarrow 0; 0.1 \rightarrow 1; 0.2 \rightarrow 0.5; 0.25 \rightarrow 0.33]$
6	Pekerjaan tambahan pada jalur kedua	[5,8]	$[0.5 \rightarrow 0; 0.03 \rightarrow 1; 0.17 \rightarrow 0.5; 0.3 \rightarrow 0.33]$

7	Aktivitas akhir pada jalur pertama	[7,10]	$[0.5 \rightarrow 0; 0.03 \rightarrow 1; 0.17 \rightarrow 0.5; 0.3 \rightarrow 0.33]$
8	Memasang papan jalan pada jalur pertama	[7,11]	$[0.67 \rightarrow 0; 0.02 \rightarrow 1; 0.13 \rightarrow 0.5; 0.18 \rightarrow 0.33]$
9	Membuat tanggul pada jalur kedua	[8,9]	$[0.45 \rightarrow 0; 0.1 \rightarrow 1; 0.2 \rightarrow 0.5; 0.25 \rightarrow 0.33]$
10	Kegiatan akhir pada jalur kedua	[10,22]	$[0.7 \rightarrow 0; 0.05 \rightarrow 1; 0.1 \rightarrow 0.5; 0.15 \rightarrow 0.33]$
11	Aktivitas akhir pada jalur kedua	[11,21]	$[0.67 \rightarrow 0; 0.02 \rightarrow 1; 0.13 \rightarrow 0.5; 0.18 \rightarrow 0.33]$
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	$[0.5 \rightarrow 0; 0.03 \rightarrow 1; 0.17 \rightarrow 0.5; 0.3 \rightarrow 0.33]$
13	Memasang tiang pancang pada jalur pertama	[12,13]	$[0.75 \rightarrow 0; 0.03 \rightarrow 1; 0.07 \rightarrow 0.5; 0.15 \rightarrow 0.33]$
14	Memasang palang pada jalur pertama	[13,14]	$[0.6 \rightarrow 0; 0.05 \rightarrow 1; 0.1 \rightarrow 0.5; 0.25 \rightarrow 0.33]$
15	Aktivitas akhir pada jembatan pertama	[14,15]	$[0.67 \rightarrow 0; 0.05 \rightarrow 1; 0.15 \rightarrow 0.5; 0.13 \rightarrow 0.33]$
16	Penundaan	[4,16]	[-]
17	Pondasi pada jembatan kedua	[16,17]	$[0.65 \rightarrow 0; 0.02 \rightarrow 1; 0.13 \rightarrow 0.5; 0.2 \rightarrow 0.33]$
18	Memasang tiang pancang jalur kedua	[17,18]	$[0.75 \rightarrow 0; 0.03 \rightarrow 1; 0.07 \rightarrow 0.5; 0.15 \rightarrow 0.33]$
19	Memasang palang pada jalur kedua	[18,19]	$[0.6 \rightarrow 0; 0.05 \rightarrow 1; 0.1 \rightarrow 0.5; 0.25 \rightarrow 0.33]$
20	Aktivitas akhir pada jembatan kedua	[19,20]	$[0.67 \rightarrow 0; 0.05 \rightarrow 1; 0.15 \rightarrow 0.5; 0.13 \rightarrow 0.33]$
21	Pengaspalan jalan	[21,22]	$[0.7 \rightarrow 0; 0.03 \rightarrow 1; 0.1 \rightarrow 0.5; 0.17 \rightarrow 0.33]$
22	Penutupan tempat konstruksi	[22,23]	[-]

Sumber: Zammori (2008).

3.3 Algoritma Program

Pada umumnya, proyek-proyek besar mempunyai data aktivitas proyek yang cukup banyak, sehingga akan sangat membantu jika

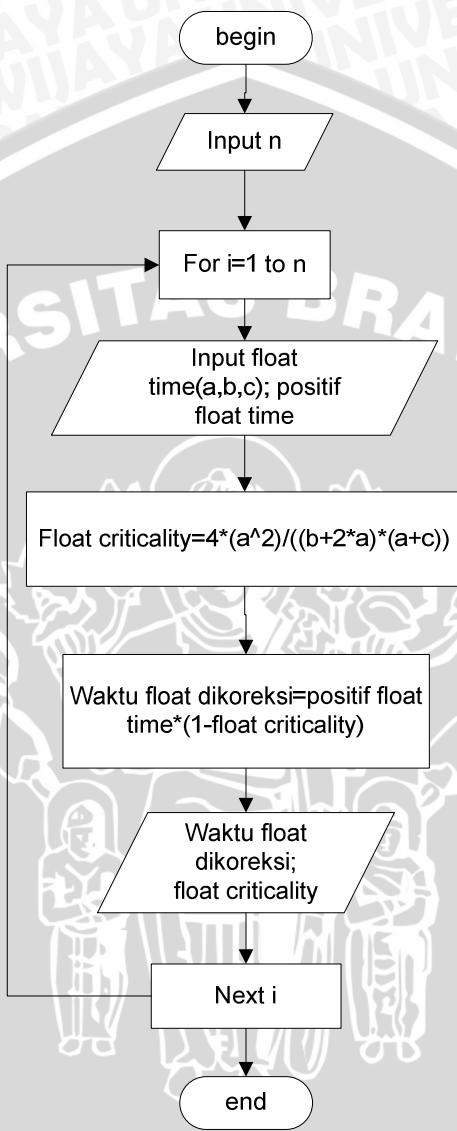
terdapat program yang dapat digunakan untuk mendapatkan nilai hasil perhitungan secara lebih cepat. Oleh sebab itu, berikut ini terdapat program-program yang dapat digunakan untuk menghitung;

- a. Nilai waktu luang, yang meliputi nilai kekritisan *float* dan nilai waktu *float* dikoreksi.
- b. Nilai kelima kriteria keputusan, yang meliputi panjang lintasan, kekritisan lintasan, jumlah biaya, perlindungan yang tersedia dan resiko perulangan desain utama.
- c. Nilai perangkingan kekritisan masing-masing lintasan, dimana perlu dihitung nilai matriks normalisasi, matriks normalisasi terbobot dan jarak *Euler* sehingga didapatkan nilai derajat kekritisan pada masing-masing lintasan.

Berikut ini terdapat 3 projek dalam menentukan lintasan kritis pelaksanaan proyek menggunakan bahasa pemrograman *Delphi*.

1. Diagram Alur (*Flowchart*) dari Program Untuk Menentukan Kekritisan Waktu *Float* dan Nilai Waktu *Float* Dikoreksi

Berikut ini adalah diagram alur program untuk menentukan kekritisan waktu *float* dan nilai waktu *float* dikoreksi;

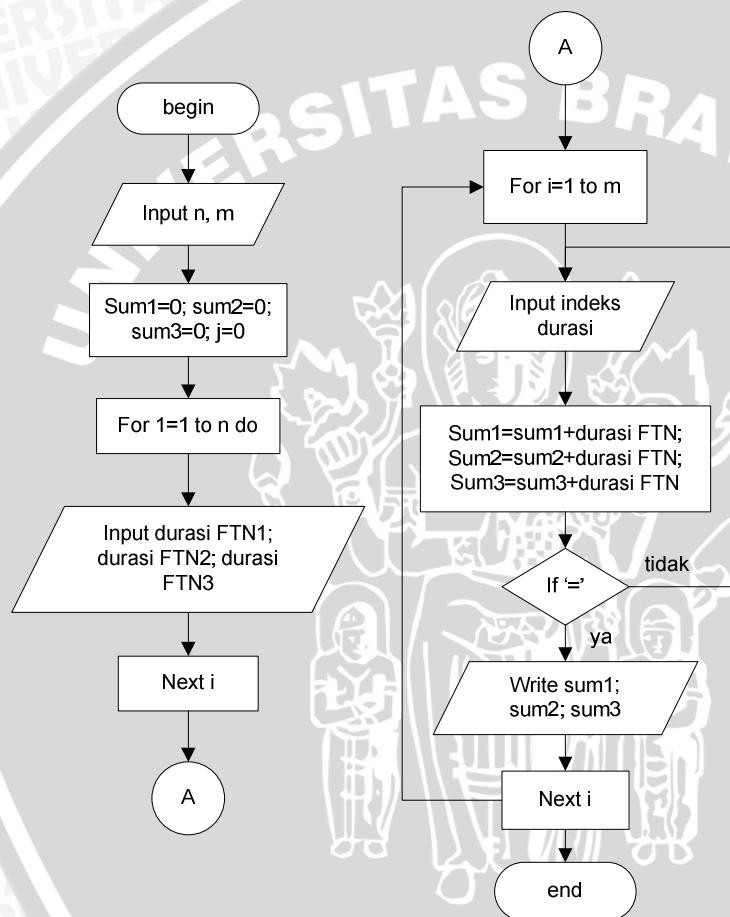


Gambar 3.1 Diagram Alur Program Kekritisian Waktu *float* dan Nilai waktu *float* dikoreksi

2 Menentukan Nilai Masing-masing Kriteria Keputusan

a) Diagram Alur (Flowchart) Program Panjang Lintasan

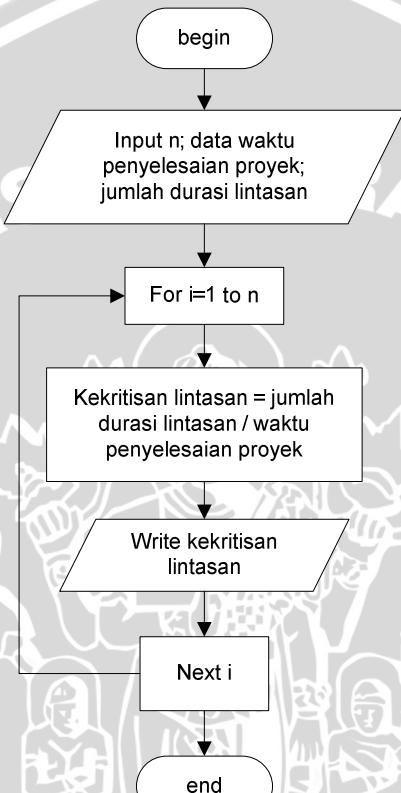
Berikut ini adalah diagram alur program jumlah durasi pada masing-masing lintasan;



Gambar 3.2 Diagram Alur Program Panjang Lintasan

b) Diagram Alur (Flowchart) Program Kekritisian Lintasan

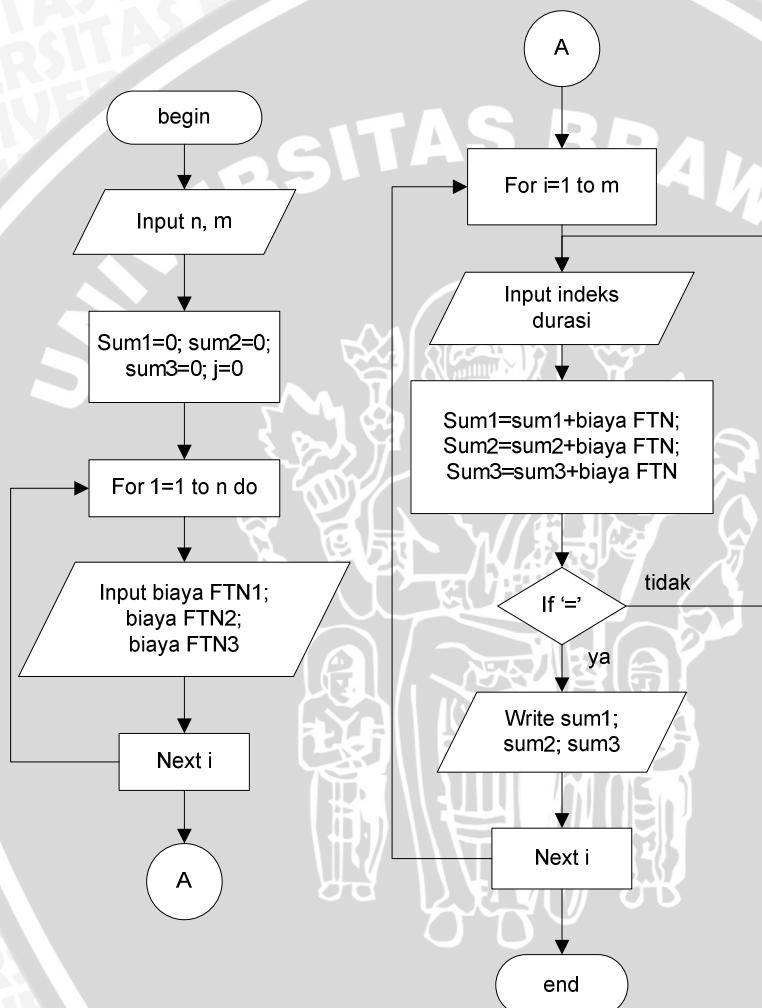
Berikut ini adalah diagram alur program kekritisan lintasan;



Gambar 3.3 Diagram Alur Program Kekritisian Lintasan

c) Diagram Alur (Flowchart) Program Jumlah Biaya

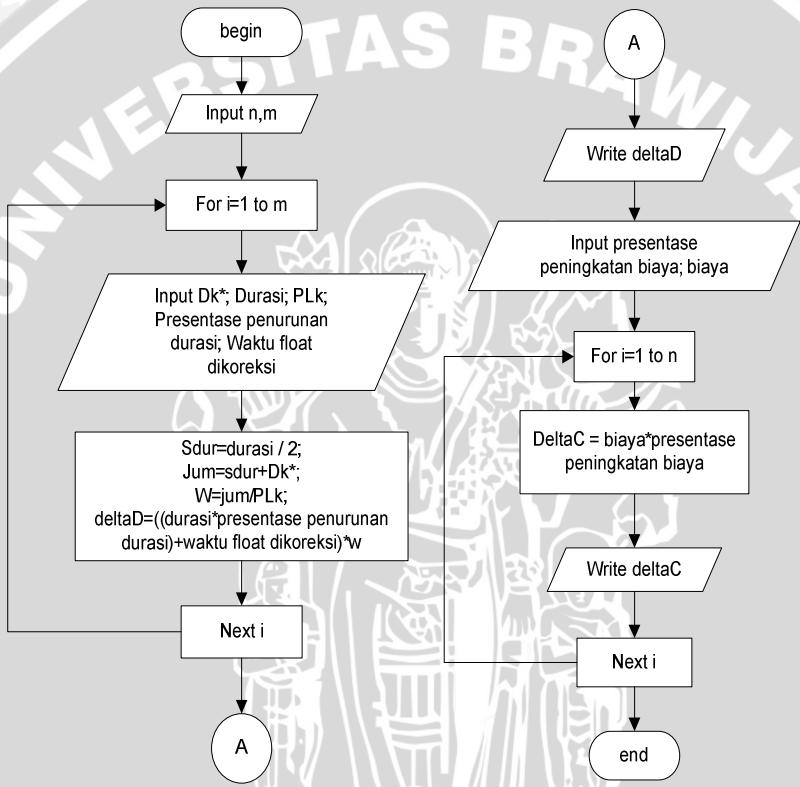
Berikut ini adalah diagram alur program jumlah biaya dari masing-masing lintasan;



Gambar 3.4 Diagram Alur Program Jumlah Biaya

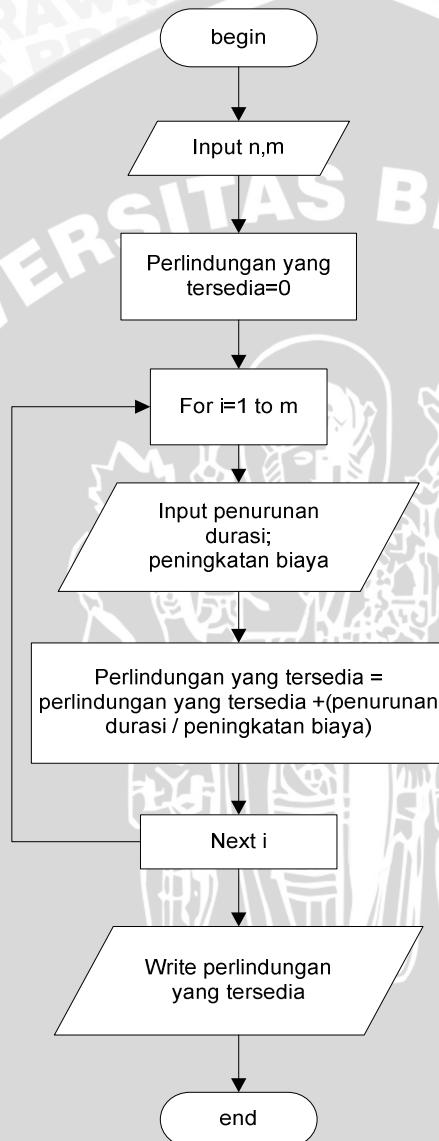
- d) Program Perlindungan yang Tersedia
- i) Diagram Alur (Flowchart) Program Menentukan nilai Peningkatan Durasi dan Peningkatan Biaya

Berikut ini adalah diagram alur program untuk menentukan nilai peningkatan durasi dan peningkatan biaya;



Gambar 3.5 Diagram Alur Program Maksimum Penurunan Durasi dan Peningkatan Biaya

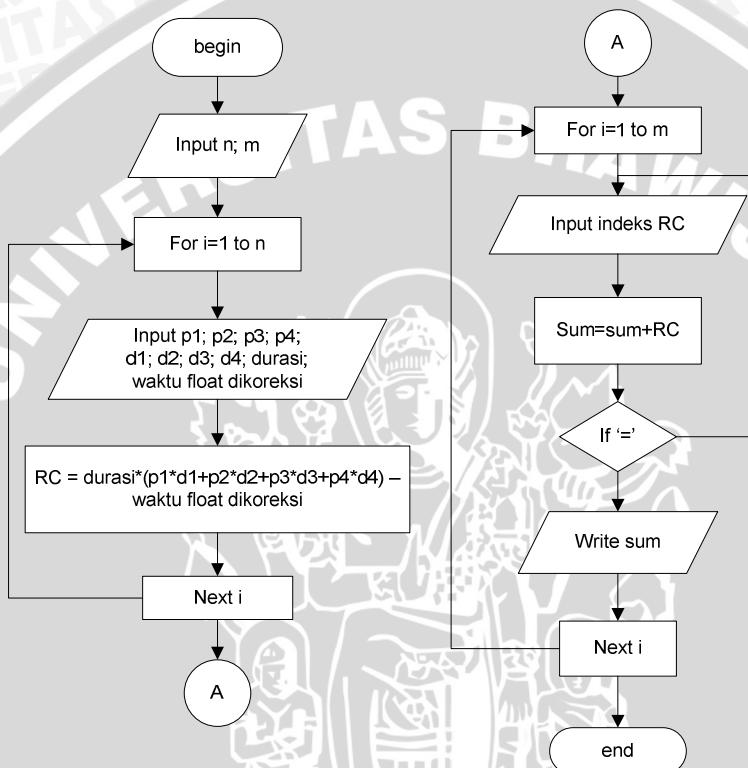
Setelah diketahui nilai ΔD dan ΔC , maka nilai AP_k dapat dihitung menggunakan program dengan diagram alur berikut:



Gambar 3.6 Diagram Alur Program Perlindungan yang Tersedia

e) **Diagram Alur (Flowchart) Program Resiko Revisi Desain Utama**

Berikut ini adalah diagram alur program resiko revisi desain utama;

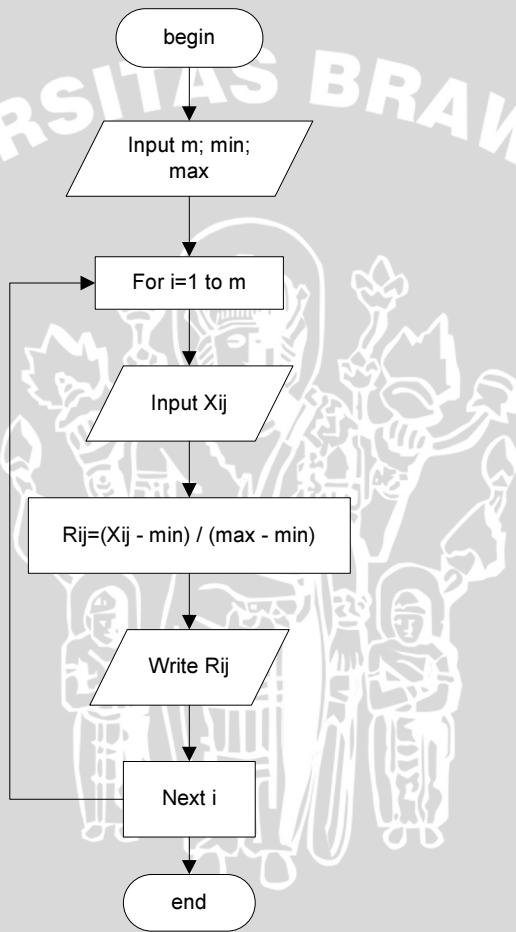


Gambar 3.7 Diagram Alur Program Resiko Revisi Desain Utama

3. Menetukan Derajat Kekritisian

a) Diagram Alur (Flowchart) Program Matriks Normalisasi

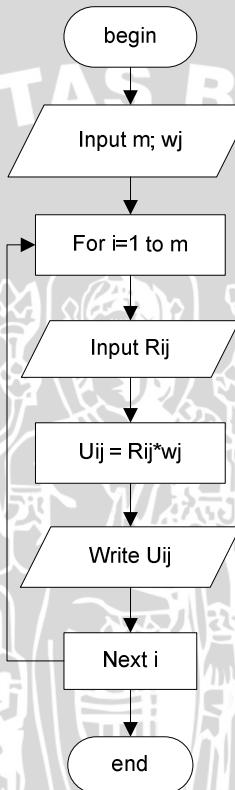
Berikut ini adalah diagram alur program matriks normalisasi;



Gambar 3.8 Diagram Alur Program Matriks Normalisasi

b) Diagram Alur (Flowchart) Program Matriks Normalisasi Terbobot

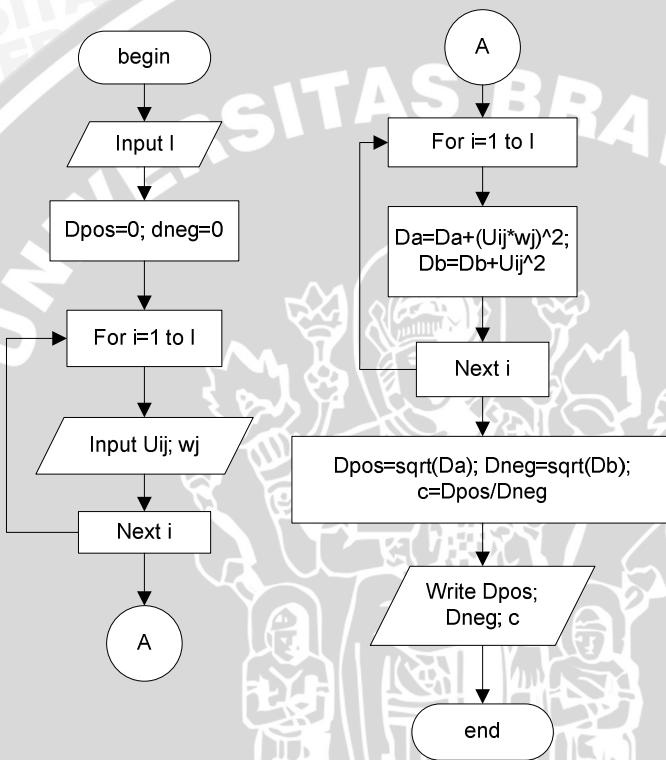
Berikut ini adalah diagram alur program matriks normalisasi terbobot;



Gambar 3.9 Diagram Alur Program Matriks Normalisasi Terbobot

c) Diagram Alur (Flowchart) Program *Euler Distance* dan Derajat Kekritisian

Berikut ini adalah diagram alur program *euler distance* dan derajat kekritisan dari masing-masing lintasan;



Gambar 3.10 Diagram Alur Program *Euler Distance* dan Derajat Kekritisian

3.4 Jaringan Proyek

Berikut ini merupakan diagram jaringan proyek pembangunan jalan;

UNIVERSITAS BRAWIJAYA



3.5 Menentukan Nilai Waktu *Float Fuzzy* (*Fuzzy Float Time*)

Untuk mendapatkan nilai *Fuzzy Float Time* (F_{ij}), perlu dihitung nilai *Early Occurrence Time* (EOT) dan nilai *Latest Occurrence Time* (LOT) terlebih dahulu. Berikut merupakan nilai EOT dan LOT pada tiap aktivitas dengan rincian perhitungan pada Lampiran 1 dan Lampiran 2.

Tabel 3.5 Daftar Nilai EOT dan LOT pada Masing-masing Aktifitas

NO	EOT	EOT_j	LOT	LOT_i
1	EOT1	[0,0,0]	LOT1	[-28,0,28]
2	EOT2	[12,15,17]	LOT2	[-11,15,40]
3	EOT3	[35,40,45]	LOT3	[18,43,67]
4	EOT4	[12,15,17]	LOT4	[-11,18,45]
5	EOT5	[35,39,42]	LOT5	[16,39,63]
6	EOT6	[65,71,75]	LOT6	[49,71,93]
7	EOT7	[65,71,76]	LOT7	[49,73,94]
8	EOT8	[65,71,75]	LOT8	[50,71,93]
9	EOT9	[87,95,100]	LOT9	[75,95,115]
10	EOT10	[87,95,101]	LOT10	[79,98,115]
11	EOT11	[87,95,102]	LOT11	[75,95,116]
12	EOT12	[46,50,56]	LOT12	[28,53,81]
13	EOT13	[65,72,79]	LOT13	[51,75,100]
14	EOT14	[77,87,97]	LOT14	[69,90,112]
15	EOT15	[84,97,109]	LOT15	[81,100,119]
16	EOT16	[46,50,56]	LOT16	[29,56,79]
17	EOT17	[64,69,77]	LOT17	[50,75,97]
18	EOT18	[76,82,91]	LOT18	[64,88,109]
19	EOT19	[83,90,102]	LOT19	[75,96,116]
20	EOT20	[86,97,109]	LOT20	[81,100,119]
21	EOT21	[90,100,109]	LOT21	[81,100,119]
22	EOT22	[103,115,126]	LOT22	[98,115,131]
23	EOT23	[112,127,140]	LOT23	[112,127,140]

Seperti yang terlihat pada Tabel 3.5 di atas, dapat diketahui bahwa waktu penyelesaian proyek tersebut adalah $P_{CT} = [112.127,140]$. Dengan menggunakan persamaan (2.28), dapat diketahui nilai *Fuzzy Float Time* (F_{ij}) dengan menggunakan bantuan bahasa pemrograman delphi untuk mempermudah perhitungan operasi fuzzy yang secara rinci dijabarkan pada Lampiran 3. Berikut merupakan nilai *Fuzzy Float Time* (F_{ij}) pada masing-masing aktivitas;

Tabel 3.6 Daftar Nilai *Fuzzy Float Time* (F_{ij}) pada Masing-masing Aktivitas

NO	Aktivitas	<i>node</i>	F_{ij}
1	Pemasangan tempat konstruksi	[1,2]	[28,0,28]
2	Menstabilkan lereng yang curam	[2,3]	[27,3,32]
3	Pekerjaan tambahan pada jalur pertama	[2,5]	[-26,0,28]
4	Pemotonga jalan pada jalur pertama	[3,7]	[-27,3,32]
5	Membuat tanggul pada jalur pertama	[5,6]	[-26,6,33]
6	Pekerjaan tambahan pada jalur kedua	[5,8]	[-3,23,50]
7	Aktivitas akhir pada jalur pertama	[7,10]	[-22,3,29]
8	Memasang papan jalan pada jalur pertama	[7,11]	[-27,2,31]
9	Membuat tanggul pada jalur kedua	[8,9]	[-25,0,28]
10	Kegiatan akhir pada jalur kedua	[10,22]	[-22,3,28]
11	Aktivitas akhir pada jalur kedua	[11,21]	[-27,0,29]
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	[-28,3,35]
13	Memasang tiang pancang pada jalur pertama	[12,13]	[-28,3,35]

14	Memasang palang pada jalur pertama	[13,14]	[-28,3,35]
15	Aktivitas akhir pada jembatan pertama	[14,15]	[-28,3,35]
16	Penundaan	[4,16]	[-27,6,33]
17	Pondasi pada jembatan kedua	[16,17]	[-27,6,33]
18	Memasang tiang pancang jalur kedua	[17,18]	[-27,6,33]
19	Memasang palang pada jalur kedua	[18,19]	[-27,6,33]
20	Aktivitas akhir pada jembatan kedua	[19,20]	[-27,6,33]
21	Pengaspalan jalan	[21,22]	[-28,0,29]
22	Penutupan tempat konstruksi	[22,23]	[-28,0,28]

Seperti yang diketahui pada Tabel 3.6, semua jarigan lintasn berpotensi kritis, karena waktu luang dari masing-masing aktivitas menggambarkan *Zero Triangular Number* (ZTN). Nilai $F_{1,2} \equiv \{-28,0,28\}$ dan $F_{22,23} \equiv \{-28,0,28\}$ merupakan simetris ZTN sehingga nilai kekritisan waktu luangnya sama dengan satu. Dari nilai *Fuzzy Float Time* (F_{ij}) yang telah didapatkan di atas, nilai F_{ij}^+ yang merupakan bagian positif dari F_{ij} juga dapat diperoleh dengan mudah menggunakan persamaan (2.49), sehingga didapatkan nilai F_{ij}^+ sebagai berikut:

Tabel 3.7 Daftar Nilai Positif F_{ij} (F_{ij}^+) pada Masing-masing Aktivitas

NO	Aktivitas	node	F_{ij}^+
1	Pemasangan tempat konstruksi	[1,2]	[0,0,28]
2	Menstabilkan lereng yang curam	[2,3]	[0,3,32]
3	Pekerjaan tambahan pada jalur pertama	[2,5]	[0,0,28]
4	Pemotonga jalan pada jalur pertama	[3,7]	[0,3,32]

5	Membuat tanggul pada jalur pertama	[5,6]	[0,6,33]
6	Pekerjaan tambahan pada jalur kedua	[5,8]	[0,23,50]
7	Aktivitas akhir pada jalur pertama	[7,10]	[0,3,29]
8	Memasang papan jalan pada jalur pertama	[7,11]	[0,2,31]
9	Membuat tanggul pada jalur kedua	[8,9]	[0,0,28]
10	Kegiatan akhir pada jalur kedua	[10,22]	[0,3,28]
11	Aktivitas akhir pada jalur kedua	[11,21]	[0,0,29]
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	[0,3,35]
13	Memasang tiang pancang pada jalur pertama	[12,13]	[0,3,35]
14	Memasang palang pada jalur pertama	[13,14]	[0,3,35]
15	Aktivitas akhir pada jembatan pertama	[14,15]	[0,3,35]
16	Penundaan	[4,16]	[0,6,33]
17	Pondasi pada jembatan kedua	[16,17]	[0,6,33]
18	Memasang tiang pancang jalur kedua	[17,18]	[0,6,33]
19	Memasang palang pada jalur kedua	[18,19]	[0,6,33]
20	Aktivitas akhir pada jembatan kedua	[19,20]	[0,6,33]
21	Pengaspalan jalan	[21,22]	[0,0,29]
22	Penutupan tempat konstruksi	[22,23]	[0,0,28]

Nilai Kekritisian *Float* (*Float Criticality*) dapat dihitung menggunakan persamaan (2.40) dengan menggunakan bantuan

bahasa pemrograman Delphi. Output nilai FC_{ij} ditampilkan pada Lampiran 4 menghasilkan nilai sebagai berikut:

Tabel 3.8 Daftar nilai *Float Criticality* (FC_{ij}) pada Masing-masing Aktivitas

NO	Aktivitas	node	FC_{ij}
1	Pemasangan tempat konstruksi	[1,2]	1
2	Menstabilkan lereng yang curam	[2,3]	0,867
3	Pekerjaan tambahan pada jalur pertama	[2,5]	0,963
4	Pemotonga jalan pada jalur pertama	[3,7]	0,867
5	Membuat tanggul pada jalur pertama	[5,6]	0,79
6	Pekerjaan tambahan pada jalur kedua	[5,8]	0,023
7	Aktivitas akhir pada jalur pertama	[7,10]	0,808
8	Memasang papan jalan pada jalur pertama	[7,11]	0,898
9	Membuat tanggul pada jalur kedua	[8,9]	0,943
10	Kegiatan akhir pada jalur kedua	[10,22]	0,824
11	Aktivitas akhir pada jalur kedua	[11,21]	0,964
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	0,844
13	Memasang tiang pancang pada jalur pertama	[12,13]	0,844
14	Memasang palang pada jalur pertama	[13,14]	0,844
15	Aktivitas akhir pada jembatan pertama	[14,15]	0,844
16	Penundaan	[4,16]	0,81
17	Pondasi pada jembatan kedua	[16,17]	0,81
18	Memasang tiang pancang jalur kedua	[17,18]	0,81

19	Memasang palang pada jalur kedua	[18,19]	0,81
20	Aktivitas akhir pada jembatan kedua	[19,20]	0,81
21	Pengaspalan jalan	[21,22]	0,982
22	Penutupan tempat konstruksi	[22,23]	1

Nilai kekritisan *float* berkisar antara 0 sampai 1. Jika nilai kekritisan *float* mendekati 1 maka aktivitas tersebut mendekati kritis. Nilai kekritisan *float* pada aktivitas 1 dan 22 bernilai 1, hal ini menunjukkan bahwa kedua aktivitas tersebut tidak mempunyai waktu luang.

Dengan menggunakan persamaan (2.48) dapat diketahui nilai *float* dikoreksi (\tilde{F}_{ij}) melalui bantuan bahasa pemrograman Delphi dan outputnya ditampilkan pada Lampiran 4. Berikut adalah hasil nilai *float* dikoreksi (\tilde{F}_{ij}):

Tabel 3.9 Daftar nilai *float* dikoreksi (\tilde{F}_{ij}) pada Masing-masing Aktivitas

NO	Aktivitas	node	\tilde{F}_{ij}
1	Pemasangan tempat konstruksi	[1,2]	[0;0;0]
2	Menstabilkan lereng yang curam	[2,3]	[0;0,4;4,25]
3	Pekerjaan tambahan pada jalur pertama	[2,5]	[0;0;1,3]
4	Pemotonga jalan pada jalur pertama	[3,7]	[0;0,4;4,3]
5	Membuat tanggul pada jalur pertama	[5,6]	[0;1,26;6,93]
6	Pekerjaan tambahan pada jalur kedua	[5,8]	[0;22,5;48,8]
7	Aktivitas akhir pada jalur pertama	[7,10]	[0;0,58;5,58]
8	Memasang papan jalan pada jalur pertama	[7,11]	[0;0,2;3,17]
9	Membuat tanggul pada jalur kedua	[8,9]	[0;0;1,6]

10	Kegiatan akhir pada jalur kedua	[10,22]	[0;0,5;4,9]
11	Aktivitas akhir pada jalur kedua	[11,21]	[0;0;1]
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	[0;0,5;5,5]
13	Memasang tiang pancang pada jalur pertama	[12,13]	[0;0,5;5,5]
14	Memasang palang pada jalur pertama	[13,14]	[0;0,5;5,5]
15	Aktivitas akhir pada jembatan pertama	[14,15]	[0;0,5;5,5]
16	Penundaan	[4,16]	[0;1,1;6,3]
17	Pondasi pada jembatan kedua	[16,17]	[0;1,1;6,3]
18	Memasang tiang pancang jalur kedua	[17,18]	[0;1,1;6,3]
19	Memasang palang pada jalur kedua	[18,19]	[0;1,1;6,3]
20	Aktivitas akhir pada jembatan kedua	[19,20]	[0;1,1;6,3]
21	Pengaspalan jalan	[21,22]	[0;0;0,5]
22	Penutupan tempat konstruksi	[22,23]	[0;0;0]

3.6 Kriteria Keputusan / Multy Kriteria Decision Making (MCDM)

3.6.1 Panjang Lintasan / Path length (PL)

Kriteria pertama yang mempengaruhi derajat kekritisan suatu lintasan adalah panjang lintasan. Jumlah durasi pada masing-masing lintasan dihitung dengan persamaan (2.41). Nilai panjang lintasan dapat dihitung dengan menggunakan bantuan bahasa pemrograman Delphi. Output nilai *PL* menggunakan ditampilkan pada Lampiran 5 menghasilkan nilai sebagai berikut:

Tabel 3.10 Daftar Panjang Lintasan (*PL*) pada Masing-masing Aktivitas

NO	Lintasan Nodes	Lintasan Durasi	Panjang Lintasan
1.	1-2-3-7-10-22-23	d1-d2-d4-d7-d10-d22	[108,123,134]
2.	1-2-3-7-11-21-22-23	d1-d2-d4-d8-d11-d8-d11-d21-d22	[106,124,139]
3.	1-2-5-6-7-10-22-23	d1-d3-d5-d7-d10-d22	[111,124,133]
4.	1-2-5-6-7-11-21-22-22	d1-d3-d5-d8-d11-d21-d22	[109,125,138]
5.	1-2-5-6-8-9-10-22-23	d1-d3-d5-d9-d10-d22	[112,124,133]
6.	1-2-5-6-8-9-11-21-22-23	d1-d3-d5-d9-d11-d21-d22	[111,127,137]
7.	1-2-5-8-9-10-22-23	d1-d3-d6-d9-d10-d22	[90,101,111]
8.	1-2-5-8-9-11-21-22-23	d1-d3-d6-d9-d11-d21-d22	[89,104,115]
9.	1-2-4-12-13-14-15-20-21-22-23	d1-d12-d13-d14-d15-d21-d22	[105,124,140]
10.	1-2-4-16-17-18-19-20-21-22-23	d1-d16-d17-d18-d19-d20-d21-d22	[107,121,139]

3.6.2 Kekritisan Lintasan / *Path Criticality* (PCr)

Untuk mendapatkan nilai kekritisan lintasan digunakan persamaan (2.42) yang dapat dengan mudah didapatkan nilainya menggunakan bantuan bahasa pemrograman Delphi. *Output* perhitungan program ditampilkan pada Lampiran 6 dengan hasil sebagai berikut:

Tabel 3.11 Daftar Kekritisan Lintasan (*PCr*) pada Masing-masing Aktivitas

NO	Lintasan Nodes	Lintasan Durasi	Kekritisan Lintasan
1.	1-2-3-7-10-22-23	d1-d2-d4-d7-d10-d22	[0,77;0,97;1,2]
2.	1-2-3-7-11-21-22-23	d1-d2-d4-d8-d11-d8-d11-d21-d22	[0,76;0,98;1,24]

3.	1-2-5-6-7-10-22-23	d1-d3-d5-d7-d10-d22	[0,79;0,98;0,19]
4.	1-2-5-6-7-11-21-22-22	d1-d3-d5-d8-d11-d21-d22	[0,78;0,98;1,23]
5.	1-2-5-6-8-9-10-22-23	d1-d3-d5-d9-d10-d22	[0,8;0,98;1,19]
6.	1-2-5-6-8-9-11-21-22-23	d1-d3-d5-d9-d11-d21-d22	[0,79;1;1,22]
7.	1-2-5-8-9-10-22-23	d1-d3-d6-d9-d10-d22	[0,64;0,8;0,99]
8.	1-2-5-8-9-11-21-22-23	d1-d3-d6-d9-d11-d21-d22	[0,64;0,82;1,03]
9.	1-2-4-12-13-14-15-20-21-22-23	d1-d12-d13-d14-d15-d21-d22	[0,75;0,98;1,25]
10.	1-2-4-16-17-18-19-20-21-22-23	d1-d16-d17-d18-d19-d20-d21-d22	[0,76;0,95;1,24]

3.6.3 Jumlah Biaya yang Diharapkan / *Expected Path Cost (PC)*

Jumlah biaya pada masing-masing lintasan dihitung berdasarkan persamaan (2.43) dihitung menggunakan bahasa pemrograman Delphi. Dari program jumlah biaya yang ditampilkan pada Lampiran 7 didapatkan hasil sebagai berikut:

Tabel 3.12 Daftar Jumlah Biaya (*PC*) pada Masing-masing Aktivitas

NO	Lintasan Nodes	Lintasan Durasi	Jumlah Biaya
1.	1-2-3-7-10-22-23	d1-d2-d4-d7-d10-d22	[495,520,539]
2.	1-2-3-7-11-21-22-23	d1-d2-d4-d8-d11-d8-d11-d21-d22	[1084,1116,1148]
3.	1-2-5-6-7-10-22-23	d1-d3-d5-d7-d10-d22	[525,554,577]
4.	1-2-5-6-7-11-21-22-22	d1-d3-d5-d8-d11-d21-d22	[1114,1150,1186]
5.	1-2-5-6-8-9-10-22-23	d1-d3-d5-d9-d10-d22	[650,682,712]

6.	1-2-5-6-8-9-11-21-22-23	d1-d3-d5-d9-d11-d21-d22	[1037,1075,1116]
7.	1-2-5-8-9-10-22-23	d1-d3-d6-d9-d10-d22	[361,390,422]
8.	1-2-5-8-9-11-21-22-23	d1-d3-d6-d9-d11-d21-d22	[748,783,826]
9.	1-2-4-12-13-14-15-20-21-22-23	d1-d12-d13-d14-d15-d21-d22	[1182,1214,1249]
10.	1-2-4-16-17-18-19-20-21-22-23	d1-d16-d17-d18-d19-d20-d21-d22	[804,835,865]

3.6.4 Perlindungan yang Tersedia / Available Protection (AP)

Dalam menentukan nilai perlindungan yang tersedia, harus diketahui terlebih dahulu nilai ΔD pada masing-masing lintasan dan nilai ΔC menggunakan persamaan (2.45) dan (2.46). Berikut ini merupakan nilai ΔD pada masing-masing lintasan dan nilai ΔC dimana nilainya dihitung menggunakan bahasa pemrograman Delphi yang ditampilkan pada Lampiran 8 dan 9;

Tabel 3.13 Daftar Nilai ΔD pada Lintasan 1

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,27 ; 0,46 ; 0,67]
2	Menstabilkan lereng yang curam	[2,3]	d2	[1,4 ; 2,16 ; 4,03]
3	Pemotonga jalan pada jalur pertama	[3,7]	d4	[2,93 ; 4,28 ; 7,62]
4	Aktivitas akhir pada jalur pertama	[7,10]	d7	[2,08 ; 3,37 ; 8,08]
5	Kegiatan akhir pada jalur kedua	[10,22]	d10	[2,39 ; 3,49 ; 8,8]
6	Penutupan tempat konstruksi	[22,23]	d22	[3,4455 ; 5,85 ; 8,556]

Tabel 3.14 Daftar Nilai ΔD pada Lintasan 2

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,27 ; 0,454 ; 0,682]

2	Menstabilkan lereng yang curam	[2,3]	d2	[1,412 ; 2,03 ; 3,78]
3	Pemotonga jalan pada jalur pertama	[3,7]	d4	[2,93 ; 4,17 ; 7,762]
4	Memasang papan jalan pada jalur pertama	[7,11]	d8	[3,009 ; 4,155 ; 8,774]
5	Aktivitas akhir pada jalur kedua	[11,21]	d11	[0,654 ; 1,334 ; 3,071]
6	Pengaspalan jalan	[21,22]	d21	[1,874 ; 2,91 ; 4,85]
7	Penutupan tempat konstruksi	[22,23]	d22	[4,09 ; 6,85 ; 10,46]

Tabel 3.15 Daftar Nilai ΔD pada Lintasan 3

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,271 ; 0,454 ; 0,651]
2	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	[1,097 ; 1,411 ; 2,14]
3	Membuat tanggul pada jalur pertama	[5,6]	d5	[4,511 ; 6,236 ; 10,61]
4	Aktivitas akhir pada jalur pertama	[7,10]	d7	[2,384 ; 3,601 ; 8,34]
5	Kegiatan akhir pada jalur kedua	[10,22]	d10	[2,677 ; 3,7 ; 8,998]
6	Penutupan tempat konstruksi	[22,23]	d22	[3,84 ; 6,155 ; 8,703]

Tabel 3.16 Daftar Nilai ΔD pada Lintasan 4

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,261 ; 0,45 ; 0,663]
2	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	[0,27 ; 0,388 ; 20,623]
3	Membuat tanggul pada jalur pertama	[5,6]	d5	[4,348 ; 6,186 ; 10,804]

4	Memasang papan jalan pada jalur pertama	[7,11]	d8	[2,638 ; 3,66 ; 7,766]
5	Aktivitas akhir pada jalur kedua	[11,21]	d11	[0,453 ; 0,959 ; 2,19]
6	Pengaspalan jalan	[21,22]	d21	[1,52 ; 2,443 ; 4,026]
7	Penutupan tempat konstruksi	[22,23]	d22	[3,267 ; 5,702 ; 8,631]

Tabel 3.17 Daftar Nilai ΔD pada Lintasan 5

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,271 ; 0,454 ; 0,645]
2	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	[1,097 ; 1,411 ; 2,12]
3	Membuat tanggul pada jalur pertama	[5,6]	d5	[4,511 ; 6,236 ; 10,514]
4	Membuat tanggul pada jalur kedua	[8,9]	d9	[3,94 ; 5,08 ; 7,35]
5	Kegiatan akhir pada jalur kedua	[10,22]	d10	[2,436 ; 3,39 ; 8,228]
6	Penutupan tempat konstruksi	[22,23]	d22	[3,512 ; 5,69 ; 8,025]

Tabel 3.18 Daftar Nilai ΔD pada Lintasan 6

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,27 ; 0,443 ; 0,651]
2	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	[1,065 ; 1,378 ; 2,14]
3	Membuat tanggul pada jalur pertama	[5,6]	d5	[4,38 ; 6,089 ; 10,609]
4	Membuat tanggul pada jalur kedua	[8,9]	d9	[3,822 ; 4,96 ; 7,413]
5	Aktivitas akhir pada jalur kedua	[11,21]	d11	[0,471 ; 0,971 ; 2,123]

6	Pengaspalan jalan	[21,22]	d21	[1,571 ; 2,458 ; 3,913]
7	Penutupan tempat konstruksi	[22,23]	d22	[3,37 ; 5,73 ; 8,476]

Tabel 3.19 Daftar Nilai ΔD pada Lintasan 7

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,324 ; 0,557 ; 0,803]
2	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	[1,315 ; 1,732 ; 2,639]
3	Pekerjaan tambahan pada jalur kedua	[5,8]	d6	[0,484 ; 7,06 ; 19,311]
4	Membuat tanggul pada jalur kedua	[8,9]	d9	[3,677 ; 4,99 ; 7,533]
5	Kegiatan akhir pada jalur kedua	[10,22]	d10	[2,738 ; 3,457 ; 8,846]
6	Penutupan tempat konstruksi	[22,23]	d22	[3,478 ; 5,917 ; 8,68]

Tabel 3.20 Daftar Nilai ΔD pada Lintasan 8

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,313 ; 0,541 ; 0,812]
2	Pekerjaan tambahan pada jalur pertama	[2,5]	d3	[1,269 ; 1,682 ; 2,668]
3	Pekerjaan tambahan pada jalur kedua	[5,8]	d6	[0,467 ; 6,856 ; 19,528]
4	Membuat tanggul pada jalur kedua	[8,9]	d9	[3,549 ; 4,846 ; 7,617]
5	Aktivitas akhir pada jalur kedua	[11,21]	d11	[0,425 ; 0,85 ; 1,951]
6	Pengaspalan jalan	[21,22]	d21	[1,512 ; 2,505 ; 4,187]
7	Penutupan tempat konstruksi	[22,23]	d22	[3,31 ; 5,954 ; 9,155]

Tabel 3.21 Daftar Nilai ΔD pada Lintasan 9

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,257 ; 0,454 ; 0,688]
2	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	d12	[2,676 ; 3,617 ; 7,064]
3	Memasang tiang pancang pada jalur pertama	[12,13]	d13	[2,26 ; 3,43 ; 7,97]
4	Memasang palang pada jalur pertama	[13,14]	d14	[1,643 ; 2,917 ; 8,683]
5	Aktivitas akhir pada jembatan pertama	[14,15]	d15	[1,047 ; 2,3 ; 8,456]
6	Pengaspalan jalan	[21,22]	d21	[1,774 ; 2,907 ; 4,935]
7	Penutupan tempat konstruksi	[22,23]	d22	[3,876 ; 6,852 ; 10,64]

Tabel 3.22 Daftar Nilai ΔD pada Lintasan 10

NO	Aktivitas	node	ID	ΔD
1	Pemasangan tempat konstruksi	[1,2]	d1	[0,259 ; 0,465 ; 0,675]
2	Penundaan	[4,16]	d16	[0 ; 0,295 ; 2,149]
3	Pondasi pada jembatan kedua	[16,17]	d17	[2,85 ; 4,278 ; 9,136]
4	Memasang tiang pancang jalur kedua	[17,18]	d18	[1,753 ; 3,039 ; 8,133]
5	Memasang palang pada jalur kedua	[18,19]	d19	[0,721 ; 1,805 ; 7,467]
6	Aktivitas akhir pada jembatan kedua	[19,20]	d20	[0,638 ; 1,901 ; 8,243]
7	Pengaspalan jalan	[21,22]	d21	[1,827 ; 2,894 ; 4,802]
8	Penutupan tempat konstruksi	[22,23]	d22	[3,982 ; 6,843 ; 10,363]

Tabel 3.23 Daftar Nilai ΔC pada masing-masing Aktivitas

NO	Aktivitas	node	ΔC
1	Pemasangan tempat konstruksi	[1,2]	[4,2 ; 5,4 ; 6,6]
2	Menstabilkan lereng yang curam	[2,3]	[65,25 ; 67,5 ; 68,85]
3	Pekerjaan tambahan pada jalur pertama	[2,5]	[26,1 ; 28,5 ; 31,2]
4	Pemotonga jalan pada jalur pertama	[3,7]	[80,15 ; 81,9 ; 84]
5	Membuat tanggul pada jalur pertama	[5,6]	[158,5 ; 161,5 ; 163,5]
6	Pekerjaan tambahan pada jalur kedua	[5,8]	[9,8 ; 10,85 ; 12,95]
7	Aktivitas akhir pada jalur pertama	[7,10]	[25,5 ; 26,95 ; 28]
8	Memasang papan jalan pada jalur pertama	[7,11]	[101,75 ; 103,6 ; 105,45]
9	Membuat tanggul pada jalur kedua	[8,9]	[79,2 ; 82 ; 86]
10	Kegiatan akhir pada jalur kedua	[10,22]	[12,24 ; 14,62 ; 15,98]
11	Aktivitas akhir pada jalur kedua	[11,21]	[50,75 ; 52,85 ; 54,95]
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	[160,65 ; 163,35 ; 166,5]
13	Memasang tiang pancang pada jalur pertama	[12,13]	[133,6 ; 136 ; 138]
14	Memasang palang pada jalur pertama	[13,14]	[43,4 ; 44,45 ; 46,55]
15	Aktivitas akhir pada jembatan pertama	[14,15]	[24,64 ; 26,56 ; 28,16]
16	Penundaan	[4,16]	[0 ; 0 ; 0]
17	Pondasi pada jembatan kedua	[16,17]	[133,5 ; 135 ; 137,5]
18	Memasang tiang pancang jalur kedua	[17,18]	[50,75 ; 53,2 ; 54,6]

19	Memasang palang pada jalur kedua	[18,19]	[15,41 ; 17,02 ; 17,94]
20	Aktivitas akhir pada jembatan kedua	[19,20]	[14 ; 15,2 ; 17,2]
21	Pengaspalan jalan	[21,22]	[94,52 ; 96,9 ; 99,96]
22	Penutupan tempat konstruksi	[22,23]	[3 ; 4,2 ; 4,8]

Setelah didapatkan nilai ΔD dan ΔC diatas, maka dapat dihitung nilai AP dengan persamaan (2.44). Menggunakan bahasa pemrograman Delphi, didapatkan *output* nilai AP yang ditampilkan pada Lampiran 10 dengan hasil sebagai berikut:

Tabel 3.24 Daftar Nilai AP Pada Masing-masing Lintasan

NO	Lintasan Nodes	Lintasan Durasi	AP
1.	1-2-3-7-10-22-23	d1-d2-d4-d7-d10-d22	[1,039 ; 1,93 ; 4,21]
2.	1-2-3-7-11-21-22-23	d1-d2-d4-d8-d11-d8-d11-d21-d22	[1,008 ; 1,891 ; 4,002]
3.	1-2-5-6-7-10-22-23	d1-d3-d5-d7-d10-d22	[1,161 ; 2,024 ; 4,267]
4.	1-2-5-6-7-11-21-22-22	d1-d3-d5-d8-d11-d21-d22	[0,804 ; 1,57 ; 4,055]
5.	1-2-5-6-8-9-10-22-23	d1-d3-d5-d9-d10-d22	[1,034 ; 1,821 ; 3,741]
6.	1-2-5-6-8-9-11-21-22-23	d1-d3-d5-d9-d11-d21-d22	[0,873 ; 1,637 ; 3,306]
7.	1-2-5-8-9-10-22-23	d1-d3-d6-d9-d10-d22	[1,067 ; 2,52 ; 5,974]
8.	1-2-5-8-9-11-21-22-23	d1-d3-d6-d9-d11-d21-d22	[0,878 ; 2,31 ; 5,519]
9.	1-2-4-12-13-14-15-20-21-22-23	d1-d12-d13-d14-d15-d21-d22	[0,969 ; 1,945 ; 4,409]
10.	1-2-4-16-17-18-19-20-21-22-23	d1-d16-d17-d18-d19-d20-d21-d22	[1,017 ; 2,065 ; 4,968]

3.6.5 Resiko Revisi Desain Utama / Risk Major Revision (RMR)

Untuk mendapatkan nilai resiko revisi desain utama, perlu dihitung terlebih dahulu nilai kekritisan pengulangan (*Recycling Criticality*) pada masing-masing aktivitas menggunakan persamaan (2.50). Berikut merupakan nilai kekritisan perulangan pada tiap aktivitas dengan bantuan bahasa pemrograman Delphi yang *output*-nya ditampilkan pada Lampiran 10:

Tabel 3.25 Daftar Nilai Kekritisannya Pengulangan (*Recycling Criticality*) Pada Masing-masing Aktivitas

NO	Aktivitas	node	RC_{ij}
1	Pemasangan tempat konstruksi	[1,2]	[2,292 ; 2,865 ; 3,247]
2	Menstabilkan lereng yang curam	[2,3]	[-4,25 ; -0,4 ; 0]
3	Pekerjaan tambahan pada jalur pertama	[2,5]	[3,622 ; 5,136 ; 5,35]
4	Pemotonga jalan pada jalur pertama	[3,7]	[0,857 ; 5,33 ; 5,921]
5	Membuat tanggul pada jalur pertama	[5,6]	[1,545 ; 7,78 ; 9,3225]
6	Pekerjaan tambahan pada jalur kedua	[5,8]	[-47,088 ; -20,574 ; 2,354]
7	Aktivitas akhir pada jalur pertama	[7,10]	[-2,4405 ; 3,008 ; 3,7375]
8	Memasang papan jalan pada jalur pertama	[7,11]	[-0,282 ; 2,9768 ; 3,7544]
9	Membuat tanggul pada jalur kedua	[8,9]	[4,615 ; 6,78 ; 7,0625]
10	Kegiatan akhir pada jalur kedua	[10,22]	[-2,508 ; 2,0415 ; 2,8405]
11	Aktivitas akhir pada jalur kedua	[11,21]	[-0,5668 ; 0,722 ; 0,8664]
12	Membuat pondasi jembatan diatas jalan yang pertama	[4,12]	[1,776 ; 6,99 ; 8,346]
13	Memasang tiang pancang pada jalur pertama	[12,13]	[-3,3245 ; 2,019 ; 2,6335]

14	Memasang palang pada jalur pertama	[13,14]	[-3,31 ; 2,2375 ; 3,285]
15	Aktivitas akhir pada jembatan pertama	[14,15]	[-4,4892 ; 0,944 ; 1,7328]
16	Penundaan	[4,16]	[-6,3 ; -1,1 ; 0]
17	Pondasi pada jembatan kedua	[16,17]	[-3,582 ; 1,769 ; 3,171]
18	Memasang tiang pancang jalur kedua	[17,18]	[-4,926 ; 0,3885 ; 1,603]
19	Memasang palang pada jalur kedua	[18,19]	[-5,0225 ; 0,36 ; 2,0075]
20	Aktivitas akhir pada jembatan kedua	[19,20]	[-5,7963 ; -0,4284 ; 1,0074]
21	Pengaspalan jalan	[21,22]	[1,1332 ; 2,0415 ; 2,3137]
22	Penutupan tempat konstruksi	[22,23]	[0 ; 0 ; 0]

Setelah diketahui nilai kekritisan pengulangan (*Recycling Criticality*) pada masing-masing aktivitas maka dengan persamaan (2.51) dapat diketahui nilai resiko revisi desain utama pada masing-masing lintasan. Dengan bantuan bahasa pemrograman Delphi didapatkan output program resiko revisi desain utama yang ditampilkan pada Lampiran 10 dengan hasil sebagai berikut:

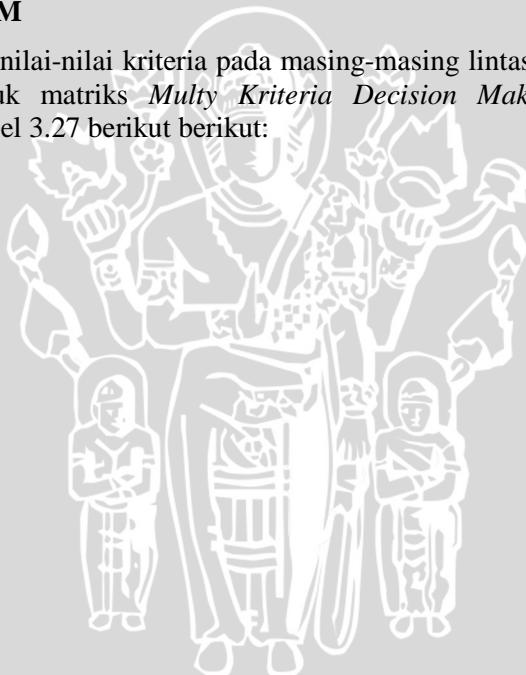
Tabel 3.26 Daftar Nilai Resiko Revisi Desain Utama pada Masing-masing Lintasan

NO	Lintasan Nodes	Lintasan Durasi	RMR
1.	1-2-3-7-10-22-23	d1-d2-d4-d7-d10-d22	[-6,05 ; 12,845 ; 15,746]
2.	1-2-3-7-11-21-22-23	d1-d2-d4-d8-d11-d8-d11-d21-d22	[-0,819 ; 13,535 ; 16,103]
3.	1-2-5-6-7-10-22-23	d1-d3-d5-d7-d10-d22	[2,511 ; 21,011 ; 24,498]
4.	1-2-5-6-7-11-21-22-22	d1-d3-d5-d8-d11-d21-d22	[7,741 ; 21,704 ; 24,854]
5.	1-2-5-6-8-9-10-22-23	d1-d3-d5-d9-d10-d22	[9,566 ; 24,783 ; 27,823]

6.	1-2-5-6-8-9-11-21-22-23	d1-d3-d5-d9-d11-d21-d22	[12,638 ; 25,505 ; 28,162]
7.	1-2-5-8-9-10-22-23	d1-d3-d6-d9-d10-d22	[-39,067 ; -3,572 ; 20,854]
8.	1-2-5-8-9-11-21-22-23	d1-d3-d6-d9-d11-d21-d22	[35,995 ; -2,85 ; 21,194]
9.	1-2-4-12-13-14-15-20-21-22-23	d1-d12-d13-d14-d15-d21-d22	[-5,923 ; 17,097 ; 21,558]
10.	1-2-4-16-17-18-19-20-21-22-23	d1-d16-d17-d18-d19-d20-d21-d22	[-22,202 ; 5,896 ; 13,35]

3.6.6 Matriks MCDM

Setelah diketahui nilai-nilai kriteria pada masing-masing lintasan, maka dapat dibentuk matriks *Multy Kriteria Decision Making* (MCDM) seperti Tabel 3.27 berikut berikut:



UNIVERSITAS BRAWIJAYA



3.7 Identifikasi Lintasan Kritis

3.7.1 Matriks Normalisasi

Dari mariks *Multy Kriteria Decision Making* (MCDM) yang telah diketahui sebelumnya, maka dengan menggunakan persamaan (2.52) dan bantuan program Matriks Normalisasi dengan output pada Lampiran 12. Berikut ini merupakan hasil output program Matriks Normalisasi yang ditampilkan pada Tabel 3.28:



UNIVERSITAS BRAWIJAYA



3.7.2 Matriks Normalisasi Terbobot

Matriks normalisasi terbobot dapat dihitung menggunakan persamaan (2.54) dan dihitung menggunakan program matriks normalisasi terbobot dengan bahasa pemrograman Delphi yang nilai outputnya ditampilkan pada Lampiran 13. Berikut merupakan output nilai matriks normalisasi terbobot yang ditampilkan pada Tabel 2.29:



UNIVERSITAS BRAWIJAYA



3.7.3 Jarak Euler (*Euler Distance*) dan Derajat Kekritisian Lintasan

Nilai Jarak Euler dapat diketahui menggunakan persamaan (2.60) dan (2.61) yang kemudian dapat dihitung nilai derajat kekritisan lintasan menggunakan persamaan (2.63). Nilai perhitungannya dapat dengan mudah dihitung menggunakan program *Euler Distance* dan Derajat Kekritisian dengan bahasa pemrograman Delphi yang *output* programnya ditampilkan pada Lampiran 14. Berikut ini merupakan *output* program *Euler Distance* dan Derajat Kekritisian:

$$D^+ = (0,99; 0,686; 0,928; 0,74; 0,838; \\ 0,789; 1,26; 1,047; 0,55; 0,752)$$

$$D^- = (0,85; 1,169; 0,945; 1,24; 1,022; \\ 1,237; 0,856; 0,8; 0,128; 1,028)$$

Setelah diketahui nilai *Euler Distance*, maka diketahui nilai derajat kekritisan pada tiap lintasan sebagai berikut:

$$C = (0,462; 0,63; 0,505; 0,63; 0,55; \\ 0,611; 0,405; 0,433; 0,7; 0,578)$$

Dari hasil diatas, dapat diketahui skema kekritisan sebagai berikut $C_9 > C_2 > C_4 > C_6 > C_{10} > C_5 > C_3 > C_1 > C_8 > C_7$. Sehingga dapat diketahui bahwa lintasan paling Kritis adalah lintasan ke-9.

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Metode yang digunakan dalam menentukan lintasan kritis pada suatu jaringan proyek menghasilkan lintasan yang paling kritis adalah lintasan ke-9 dengan derajat kekritisan 0,7. Dengan mengetahui tingkat kekritisan masing-masing lintasan pada suatu jaringan proyek, maka pelaksana proyek dapat berhati-hati dalam memonitor selama fase proyek berlangsung sehingga penyelesaian proyek dapat berjalan tepat waktu.

4.2 Saran

Dalam skripsi ini hanya digunakan metode yang digunakan dalam menentukan lintasan kritis. Untuk pengembangannya dapat digunakan metode lain untuk menentukan lintasan kritis suatu jaringan proyek besar, sehingga dapat dibandingkan hasil identifikasi lintasan kritis dengan data yang tersedia.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Agustini,D.H. dan Y.E. Rahmadi. 2004. *Riset Operasional Konsep-konsep Dasar*. Jakarta : Rineka Cipta. 188-189.
- Arinjtingias, DD. T., 2004. *Perbandingan Antara Metode PERT (Project Evaluation and Review Technique) dan Jaringan Kerja Fuzzy untuk menentukan lintasan Kritis*. Malang : Universitas Brawijaya.
- Chen Tung Chen dan Sue Fen Huang, 2007. *Applying fuzzy method for measuring criticality In project network*. Taiwan : Elsevier.
- Huang J.W., Chen R., Wang X.X., Zhou Y.H. 2010.*Study on The Application of Fuzzy TOPSIS to the Multi-objective Decision Making*. China : IEEE.
- Laiskodat, Maryvin. 2011. *Aplikasi Untuk Mendeteksi Saluran Penceraaan dengan Fuzzy Multi Criteria Decision Making*. Jakarta : Universitas Pembangunan Nasional.
- Levin, R.I., C.A. Kirkpatrick, D.S. Rubin. 1982. *Quantitative Approaches to Management*. Electra : Monotype Composition Company, Inc. 608-617.
- Marsudi, 2008. *Modul Responsi Riset Operasi II*. Malang : Laboratorium matematika Jurusan matematika FMIPA UB.
- Render, B. dan stair, R. M., 2000. *Quantitative Analysis for Management*. 7th edition. New Jersey : Prentice Hall Englewood Cliffs.
- Soeharto, I., 1995. *Manajemen Proyek Dari Konseptual Sampai Operasional*. Jakarta : Erlangga.

Taha, H. A., 1997. *Riset Operasi Suatu Pengantar*. Jilidn 2. Edisi kelima. Jakarta : Binarupa Aksara.

Wahyu. 2008. *Pengembangan Konten E-learning dalam Bidang Artificial Intelligence*. Malang : Universitas Widya Gama.

Wiyanti, Wiwik. 2007. *Manajemen Waktu Penjadwalan Proyek Pembangunan Gedung*. Semarang : Universitas Diponegoro. 16-21.

Zammori, F.A., M. Braglia, M. Frosolini. 2008. *A fuzzy multi-criteria approach for critical path definition*. Italy : Elsevier.



LAMPIRAN

Lampiran1. Hasil Perhitungan Nilai *Early Occurrence Time* (EOT) menggunakan bantuan Microsoft excel.

EOT	max	EOTj
EOT1	[0,0,0]	[0,0,0]
EOT2	{[12,15,17]}	[12,15,17]
EOT3	{[12,15,17],[35,40,45]}	[35,40,45]
EOT4	{[12,15,17]}	[12,15,17]
EOT5	{[12,15,17],[35,39,42]}	[35,39,42]
EOT6	{[12,15,17],[35,39,42],[65,71,75]}	[65,71,75]
EOT7	{[12,15,17],[35,39,42],[35,40,45], [62,70,76],[65,71,75]}	[65,71,76]
EOT8	{[12,15,17],[35,39,42],[43,48,53], [65,71,75]}	[65,71,75]
EOT9	{[12,15,17],[35,39,42],[43,48,53], [65,71,75],[87,95,100]}	[87,95,100]
EOT10	{[12,15,17],[35,39,42],[35,40,45], [43,48,53],[62,70,76],[65,71,75], [87,95,100],[86,95,101]}	[87,95,101]
EOT11	{[12,15,17],[35,39,42],[35,40,45], [43,48,53],[62,70,76],[65,71,75], [87,95,100],[85,93,102]}	[87,95,102]
EOT12	{[12,15,17],[46,50,56]}	[46,50,56]
EOT13	{[12,15,17],[46,50,56],[65,72,79]}	[65,72,79]
EOT14	{[12,15,17],[46,50,56],[65,72,79], [77,87,97]}	[77,87,97]
EOT15	{[12,15,17],[46,50,56],[65,72,79], [77,87,97],[84,97,109]}	[84,97,109]
EOT16	{[12,15,17],[46,50,56]}	[46,50,56]
EOT17	{[12,15,17],[46,50,56],[64,69,77]}	[64,69,77]
EOT18	{[12,15,17],[46,50,56],[64,69,77], [76,82,91]}	[76,82,91]
EOT19	{[12,15,17],[46,50,56],[64,69,77], [76,82,91],[83,90,102]}	[83,90,102]

EOT20	$\{[12,15,17],[46,50,56],[64,69,77], [65,72,79],[76,82,91],[77,87,97], [83,90,102],[84,97,109],[86,94,108]\}$	[86,97,109]
EOT21	$\{[12,15,17],[35,39,42],[35,40,45], [43,48,53],[46,50,56],[62,70,76], [64,69,77],[65,71,75],[65,72,79], [76,82,91],[77,87,97],[83,90,102], [84,97,109],[86,94,108],[87,95,100], [85,93,102],[90,100,108]\}$	[90,100,109]
EOT22	[102,115,126],[103,112,120]	[103,115,126]
EOT23	-	[112,127,140]

Lampiran2. Hasil Perhitungan Nilai *Latest Occurrence Time* (LOT) menggunakan bantuan Microsoft excel.

LOT	min	LOTi
LOT1	-	[-28,0,28]
LOT2	$[98,115,131],[81,100,119],[79,98,115], [75,96,116],[69,90,112],[64,88,109], [54,74,94],[50,75,97],[50,71,93], [51,75,100],[49,71,96],[39,62,85], [29,56,79],[28,53,81],[18,41,67], [16,39,63],[-11,18,47],[-10,21,45], [-9,15,40],[-10,18,44]$	[-11,15,40]
LOT3	$[98,115,131],[81,100,119],[79,98,115], [75,95,116],[54,74,94],[49,73,96], [18,43,67]$	[18,43,67]
LOT4	$[98,115,131],[81,100,119],[75,96,116], [69,90,112],[64,88,109],[50,75,97], [51,75,100],[29,56,79],[28,53,81], [-11,18,47],[-10,21,45]$	[-11,18,45]
LOT5	$[98,115,131],[81,100,119],[79,98,115], [75,95,116],[54,74,94],[50,71,93], [49,71,96],[16,39,63],[39,62,85]$	[16,39,63]

LOT6	[98,115,131],[81,100,119],[79,98,115], [75,95,116],[54,74,94],[50,71,93], [49,71,96]	[49,71,93]
LOT7	[98,115,131],[81,100,119],[79,98,115], [75,95,116],[54,74,94],[49,73,96]	[49,73,94]
LOT8	[98,115,131],[81,100,119],[79,98,115], [75,95,116],[50,71,93]	[50,71,93]
LOT9	[98,115,131],[81,100,119],[79,98,115], [75,95,116]	[75,95,115]
LOT10	[98,115,131],[79,98,115]	[79,98,115]
LOT11	[98,115,131],[81,100,119],[75,95,116]	[75,95,116]
LOT12	[98,115,131],[81,100,119],[69,90,112], [51,75,100],[28,53,81]	[28,53,81]
LOT13	[98,115,131],[81,100,119],[69,90,112], [51,75,100]	[51,75,100]
LOT14	[98,115,131],[81,100,119],[69,90,112]	[69,90,112]
LOT15	[98,115,131],[81,100,119]	[81,100,119]
LOT16	[98,115,131],[81,100,119],[75,96,116], [64,88,109],[50,75,97],[29,56,79]	[29,56,79]
LOT17	[98,115,131],[81,100,119],[75,96,116], [64,88,109],[50,75,97]	[50,75,97]
LOT18	[98,115,131],[81,100,119],[75,96,116], [64,88,109]	[64,88,109]
LOT19	[98,115,131],[81,100,119],[75,96,116]	[75,96,116]
LOT20	[98,115,131],[81,100,119]	[81,100,119]
LOT21	[98,115,131],[81,100,119]	[81,100,119]
LOT22	[98,115,131]	[98,115,131]
LOT23	-	[112,127,140]

Lampiran3. Hasil Perhitungan Nilai *Fuzzy Float Time* (F_{ij}) menggunakan bantuan Microsoft excel.

<i>node</i>	Dij+EOT	F_{ij} =LOT-(Dij+EOT)
1-2	[12,15,17]	[-28,0,28]
2-3	[35,40,45]	[-27,3,32]
2-5	[35,39,42]	[-26,0,28]
3-7	[62,70,76]	[-27,3,32]
5-6	[65,71,75]	[-26,0,28]
5-8	[43,48,53]	[-3,23,50]
7-10	[86,95,101]	[-22,3,29]
7-11	[85,93,102]	[-27,2,31]
8-9	[87,95,100]	[-25,0,28]
10-22	[103,112,120]	[-22,3,28]
11-21	[90,100,108]	[-27,0,29]
4-12	[46,50,56]	[-28,3,35]
12-13	[65,72,79]	[-28,3,35]
13-14	[77,87,97]	[-28,3,35]
14-15	[84,97,109]	[-28,3,35]
4-16	[46,50,56]	[-27,6,33]
16-17	[64,69,77]	[-27,6,33]
17-18	[76,82,91]	[-27,6,33]
18-19	[83,90,102]	[-27,6,33]
19-20	[86,94,108]	[-27,6,33]
21-22	[102,115,126]	[-28,0,29]
22-23	[112,127,140]	[-28,0,28]

Dengan menggunakan *software DELPHI* didapatkan hasil perhitungan MCDM sebagai berikut.

Lampiran 4. Hasil Perhitungan Nilai *Float Criticality* (FC) dan Nilai *float* dikoreksi (\tilde{F}_{ij}) Pada Masing-masing Aktivitas.

NO	<i>Listing Program</i>
	<pre>procedure TForm1.BitBtn1Click(Sender: TObject); var i:integer; begin for i:=1 to n do begin f1[i]:=StrToInt(StringGrid1.Cells[i-1,1]); f2[i]:=StrToInt(StringGrid1.Cells[i-1,2]); f3[i]:=StrToInt(StringGrid1.Cells[i-1,3]); end; for i:=1 to n do begin pf1[i]:=StrToInt(StringGrid2.Cells[i-1,1]); pf2[i]:=StrToInt(StringGrid2.Cells[i-1,2]); pf3[i]:=StrToInt(StringGrid2.Cells[i-1,3]); end; StringGrid3.ColCount:=n; StringGrid3.RowCount:=2; for i:=1 to n do begin fc[i]:=(4*f1[i]*f1[i])/((f2[i]+2*f1[i])*(f1[i]+f3[i])); end; for i:=1 to n do begin stringgrid3.cells[i-1,1]:=floatToStr(fc[i]); end; end; procedure TForm1.Button2Click(Sender: TObject); var i:integer; begin n:=StrToInt(Edit1.Text); StringGrid1.ColCount:=n; StringGrid1.RowCount:=4; StringGrid2.ColCount:=n; StringGrid2.RowCount:=4;</pre>

```
for i:=1 to n do
begin
  StringGrid1.Cells[i-1,0]:=IntToStr(i);
  StringGrid2.Cells[i-1,0]:=IntToStr(i);
end;
end;

procedure TForm1.Button1Click(Sender: TObject);
var i:integer;
begin
  StringGrid4.ColCount:=n; StringGrid4.RowCount:=4;
  for i:=1 to n do
  begin
    fc[i]:=StrToFloat(StringGrid3.Cells[i-1,1]);
  end;
  for i:=1 to n do
  begin
    cf[i]:=1-fc[i];
  end;
  for i:=1 to n do
  begin
    fk1[i]:=pf1[i]*cf[i]; fk2[i]:=pf2[i]*cf[i]; fk3[i]:=pf3[i]*cf[i];
  end;
  for i:=1 to n do
  begin
    stringgrid4.cells[i-1,1]:=floatToStr(fk1[i]);
    stringgrid4.cells[i-1,2]:=floatToStr(fk2[i]);
    stringgrid4.cells[i-1,3]:=floatToStr(fk3[i]);
  end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  application.terminate;
end;end.
```

Float

Input Float time				
Banyak Data:	22	OK		
Float Time (Fi)				
7	8	9	10	11
22	27	25	22	27
3	2	0	3	0
29	31	28	28	29
<input type="button" value="<"/> <input type="button" value=""/> <input type="button" value=">"/>				
<input checked="" type="checkbox"/> FC				
Positif Float Time (F+)				
4	5	6	7	8
0	0	0	0	0
3	6	23	3	2
32	33	50	29	31
<input type="button" value="<"/> <input type="button" value=""/> <input type="button" value=">"/>				
<input checked="" type="checkbox"/> OK				

Output Data				
Float Criticality (FCii)				
8	9	10	11	12
0.897783251	0.943396226	0.823829787	0.964285714	0.84369114
<input type="button" value="<"/> <input type="button" value=""/> <input type="button" value=">"/>				
<input type="button" value="Hitung ~Fi"/>				
Waktu Float Dikoreksi (~Fi)				
12	13	14	15	16
0	0	0	0	0
0.468926553	0.468926553	0.468926553	0.468926553	1.14
5.470809792	5.470809792	5.470809792	5.470809792	6.27
<input type="button" value="<"/> <input type="button" value=""/> <input type="button" value=">"/>				
<input type="button" value="Close"/>				

Lampiran 5. Hasil Perhitungan Nilai Panjang Lintasan Pada Masing-masing Lintasan.

NO	<i>Listing Program</i>
	<pre> procedure TForm1.Button1Click(Sender: TObject); var i:integer; begin n:=StrToInt(Edit1.Text); m:=StrToInt(Edit2.Text); StringGrid1.ColCount:=n; StringGrid1.RowCount:=4; StringGrid3.ColCount:=m; StringGrid3.RowCount:=4; for i:=1 to n do begin StringGrid1.Cells[i-1,0]:=IntToStr(i); ComboBox1.Items.Add('d'+IntToStr(i)); end; for i:=1 to n do begin StringGrid3.Cells[i-1,0]:=IntToStr(i); end; end; procedure TForm1.Button3Click(Sender: TObject); begin form1.Visible:=false; form2.Visible:=true; end;</pre>

```

procedure TForm1.Button4Click(Sender: TObject);
var i:integer;
s:array[1..100]of integer;
begin
j:=j+1; i:=ComboBox1.ItemIndex;
if RadioGroup1.ItemIndex=1 then
begin
kal1:=kal1+ComboBox1.Items[i]+'\';
s[j]:=i;
begin
sum1:=sum1+StrToInt(StringGrid1.Cells[s[j],1]);
sum2:=sum2+StrToInt(StringGrid1.Cells[s[j],2]);
sum3:=sum3+StrToInt(StringGrid1.Cells[s[j],3]);
end; end;
else
begin
kal1:=kal1+ComboBox1.Items[i]+'\';
s[j]:=i;
begin
sum1:=sum1+StrToInt(StringGrid1.Cells[s[j],1]);
sum2:=sum2+StrToInt(StringGrid1.Cells[s[j],2]);
sum3:=sum3+StrToInt(StringGrid1.Cells[s[j],3]);
end; end;
Edit3.Text:=kal1;
stringgrid2.cells[0,1]:=IntToStr(sum1);
stringgrid2.cells[1,1]:=IntToStr(sum2);
stringgrid2.cells[2,1]:=IntToStr(sum3);
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
w:=1; j:=0; sum1:=0; sum2:=0; sum3:=0;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
j:=0; kal1:=""; Edit3.Text:=""; RadioGroup1.ItemIndex:=-1;
ComboBox1.ItemIndex:=-1; sum1:=0; sum2:=0; sum3:=0;
end;

```

```
procedure TForm1.Button6Click(Sender: TObject);
var i:integer;
begin
w:=1;
for i:=1 to n do
begin
  StringGrid2.Cells[i-1,1]:=""; StringGrid2.Cells[i-1,2]:="";
  StringGrid2.Cells[i-1,3]:="";
end; end;

procedure TForm1.Button7Click(Sender: TObject);
var i:integer;
begin
edit1.Text:="";
for i:=1 to n do
begin
  StringGrid1.Cells[i-1,1]:=""; StringGrid1.Cells[i-1,2]:="";
  StringGrid1.Cells[i-1,3]:="";
end;
for i:=1 to m do
begin
  StringGrid2.Cells[i-1,1]:=""; StringGrid2.Cells[i-1,2]:="";
  StringGrid2.Cells[i-1,3]:="";
end; end;
end.
```

Panjang Lintasan

Input Data																																		
Banyak Data	22	Banyak Lintasan	10	OK	hapus data																													
Durasi FTN	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>12</td><td>23</td><td>23</td><td>27</td><td>30</td></tr> <tr><td>15</td><td>25</td><td>24</td><td>30</td><td>32</td></tr> <tr><td>17</td><td>28</td><td>25</td><td>31</td><td>33</td></tr> </table>					1	2	3	4	5	12	23	23	27	30	15	25	24	30	32	17	28	25	31	33									
1	2	3	4	5																														
12	23	23	27	30																														
15	25	24	30	32																														
17	28	25	31	33																														
	<input type="button" value="<"/> <input type="button" value=">"/>																																	
Perhitungan																																		
d22	<input type="radio"/> tanda <input checked="" type="radio"/> + <input type="radio"/> = <input type="radio"/> ok	<input type="button" value="close"/>																																
$d1+d16+d17+d18+d19+d20+d21+d22=$				<input type="button" value="hapus perhitungan"/>																														
Output Data																																		
Panjang Lintasan	<table border="1"> <tr><td>107</td><td>121</td><td>139</td><td><input type="button" value="^"/></td><td><input type="button" value="<"/></td><td><input type="button" value=">"/></td><td><input type="button" value=">>"/></td></tr> </table>					107	121	139	<input type="button" value="^"/>	<input type="button" value="<"/>	<input type="button" value=">"/>	<input type="button" value=">>"/>	<input type="button" value="hapus output data"/>																					
107	121	139	<input type="button" value="^"/>	<input type="button" value="<"/>	<input type="button" value=">"/>	<input type="button" value=">>"/>																												
<table border="1"> <tr><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td></td></tr> <tr><td>112</td><td>111</td><td>90</td><td>89</td><td>105</td><td>107</td><td></td></tr> <tr><td>124</td><td>127</td><td>101</td><td>104</td><td>124</td><td>121</td><td></td></tr> <tr><td>133</td><td>137</td><td>111</td><td>115</td><td>140</td><td>139</td><td></td></tr> </table>						5	6	7	8	9	10		112	111	90	89	105	107		124	127	101	104	124	121		133	137	111	115	140	139		<input type="button" value="<"/>
5	6	7	8	9	10																													
112	111	90	89	105	107																													
124	127	101	104	124	121																													
133	137	111	115	140	139																													
						<input type="button" value=">"/>																												

Lampiran 6. Hasil Perhitungan Nilai Kekritisian Lintasan Pada Masing-masing Lintasan.

NO	<i>Listing Program</i>
	<pre> procedure TForm3.Button1Click(Sender: TObject); var i:integer; begin j:=0; n:=StrToInt(Edit1.Text); StringGrid1.ColCount:=n; StringGrid1.RowCount:=4; StringGrid2.ColCount:=n; StringGrid2.RowCount:=4; for i:=1 to n do </pre>

```

begin
  StringGrid1.Cells[i-1,0]:=IntToStr(i);
  StringGrid2.Cells[i-1,0]:=IntToStr(i);
end;
end;

procedure TForm3.Button2Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to n do
  begin
    a[i]:=StrToInt(StringGrid1.Cells[i-1,1]);
    b[i]:=StrToInt(StringGrid1.Cells[i-1,2]);
    c[i]:=StrToInt(StringGrid1.Cells[i-1,3]);
  end;
end;

procedure TForm3.Button4Click(Sender: TObject);
begin
  form3.Visible:=false; form2.Visible:=true;
end;

procedure TForm3.Button3Click(Sender: TObject);
var i,j:integer;
begin
  aCT:=strtoint(edit2.Text); bCT:=strtoint(edit3.Text);
  cCT:=strtoint(edit4.Text);
  for i:=1 to n do
  begin
    PC1[i-1]:=a[i]/cCT; PC2[i-1]:=b[i]/bCT;
    PC3[i-1]:=c[i]/aCT;
  end;
  for i:=1 to n do
  begin
    StringGrid2.Cells[i-1,1]:=floattostr(PC1[i-1]);
    StringGrid2.Cells[i-1,2]:=floattostr(PC2[i-1]);
    StringGrid2.Cells[i-1,3]:=floattostr(PC3[i-1]);
  end; end;
end.

```

Kekritisian Lintasan

Input Data					
Banyak Lintasan	10	ok			
lintasan ke-	1	2	3	4	5
	108	106	111	109	112
	123	124	124	125	124
	134	139	133	138	133
	<  >				OK
PCt	112	127	140	Proses	

Output Data					
1	2	3	4	5	6
0,771428571	0,757142857	0,792857142	0,778571428	0,8	0
0,968503937	0,976377952	0,976377952	0,984251968	0,976377952	1
1,196428571	1,241071428	1,1875	1,232142857	1,1875	1
<  >				hapus	close

Lampiran 7. Hasil Perhitungan Nilai Jumlah Biaya yang Diharapkan (PC) Pada Masing-masing Lintasan.

NO	<i>Listing Program</i>
	<pre> procedure TForm5.Button1Click(Sender: TObject); var i,h:integer; begin j:=0; n:=StrToInt(Edit1.Text); m:=StrToInt(Edit2.Text); StringGrid1.ColCount:=n; StringGrid1.RowCount:=4; StringGrid3.ColCount:=m; StringGrid3.RowCount:=4; for i:=1 to n do begin StringGrid1.Cells[i-1,0]:=IntToStr(i); ComboBox1.Items.Add('d'+IntToStr(i)); end; for h:=1 to m do </pre>

```

begin
  StringGrid3.Cells[h-1,0]:=IntToStr(h);
end;
end;

procedure TForm5.Button2Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to n do
  begin
    a[i]:=StrToInt(StringGrid1.Cells[i-1,1]);
    b[i]:=StrToInt(StringGrid1.Cells[i-1,2]);
    c[i]:=StrToInt(StringGrid1.Cells[i-1,3]);
  end;
end;

procedure TForm5.Button4Click(Sender: TObject);
var i:integer;
  s:array[1..100]of integer;
begin
  j:=j+1; i:=ComboBox1.ItemIndex;
  if RadioGroup1.ItemIndex=1 then
    begin
      kal1:=kal1+ComboBox1.Items[i]+ '=';
      s[j]:=i;
      begin
        sum1:=sum1+StrToInt(StringGrid1.Cells[s[j],1]);
        sum2:=sum2+StrToInt(StringGrid1.Cells[s[j],2]);
        sum3:=sum3+StrToInt(StringGrid1.Cells[s[j],3]);
      end; end;
  else
    begin
      kal1:=kal1+ComboBox1.Items[i]+ '+';
      s[j]:=i;
      begin
        sum1:=sum1+StrToInt(StringGrid1.Cells[s[j],1]);
        sum2:=sum2+StrToInt(StringGrid1.Cells[s[j],2]);
        sum3:=sum3+StrToInt(StringGrid1.Cells[s[j],3]);
      end;
    end;
end;

```

```
end;
Edit3.Text:=kal1;
stringgrid2.cells[0,1]:=IntToStr(sum1);
stringgrid2.cells[1,1]:=IntToStr(sum2);
stringgrid2.cells[2,1]:=IntToStr(sum3);
end;

procedure TForm5.Button3Click(Sender: TObject);
begin
form5.Visible:=false; form2.Visible:=true;
end;

procedure TForm5.Button7Click(Sender: TObject);
begin
kal1:=""; RadioGroup1.ItemIndex:=-1;
ComboBox1.ItemIndex:=-1; sum1:=0; sum2:=0; sum3:=0;
edit3.Text:="";
end;

procedure TForm5.FormCreate(Sender: TObject);
begin
kal1:=""; j:=0; sum1:=0; sum2:=0; sum3:=0;
end;

procedure TForm5.Button5Click(Sender: TObject);
var i:integer;
begin
edit1.Text:=""; edit2.Text:="";
for i:=1 to n do
begin
StringGrid1.Cells[i-1,1]:=""; StringGrid1.Cells[i-1,2]:="";
StringGrid1.Cells[i-1,3]:="";
end; end;

procedure TForm5.Button8Click(Sender: TObject);
begin
w:=1;
for i:=1 to n do
begin
```

```

StringGrid2.Cells[i-1,1]:=""; StringGrid2.Cells[i-1,2]:="";
StringGrid2.Cells[i-1,3]:="";
end;
end;
end.

```

Jumlah Biaya

Input Data

Banyak Data	22	Banyak Lintasan	10	ok
-------------	----	-----------------	----	----

Biaya	1	2	3	4	5
	7	145	87	229	317
	9	150	95	234	323
	11	153	104	240	327

hapus data

OK

Perhitungan

d22	tanda	ok
d1+d16+d17+d18+d19+d20+d21+d22=		hapus perhitungan

close

Output Data

Jumlah Biaya	1	2	3	4	5	6	7
	804	835	865				

hapus output

1	2	3	4	5	6	7
495	1084	525	1114	650	1037	361
520	1116	554	1150	682	1075	390
539	1148	557	1186	712	1116	422

Lampiran 8. Hasil Perhitungan Nilai Maksimal Pertambahan Biaya dan penurunan durasi pada Masing-masing Aktivitas.

NO	<i>Listing Program</i>
	<pre> procedure TForm6.FormCreate(Sender: TObject); begin j:=0; end; procedure TForm6.Button1Click(Sender: TObject); begin dk1:=strtoint(edit9.Text); dk2:=strtoint(edit10.Text); dk3:=strtoint(edit11.Text); jum1:=dk1+sdur1; jum2:=dk2+sdur2; jum3:=dk3+sdur3; edit5.Text:=floattostr(jum1); edit6.Text:=floattostr(jum2); edit7.Text:=floattostr(jum3); end; procedure TForm6.BitBtn3Click(Sender: TObject); var i:integer; begin StringGrid3.RowCount:=2; StringGrid3.ColCount:=3; for i:=1 to 3 do begin StringGrid3.cells[i-1,0]:=inttostr(i); end; jd1:=strtofloat(edit5.text); jd2:=strtofloat(edit6.text); jd3:=strtofloat(edit7.text); pl1:=strtoint(edit2.Text); pl2:=strtoint(edit3.Text); pl3:=strtoint(edit4.Text); w1:=(jd1/pl3); w2:=(jd2/pl2); w3:=(jd3/pl1); stringgrid3.cells[0,1]:=floattostr(w1); stringgrid3.cells[1,1]:=floattostr(w2); stringgrid3.cells[2,1]:=floattostr(w3); end; procedure TForm6.BitBtn4Click(Sender: TObject); var i:integer; begin StringGrid4.RowCount:=2; StringGrid4.ColCount:=3; for i:=1 to 3 do </pre>

```

begin
  StringGrid4.Cells[i-1,0]:=inttostr(i);
end;
d1:=strtoint(edit12.Text); d2:=strtoint(edit13.Text);
d3:=strtoint(edit14.Text); p:=strtofloat(edit15.Text);
wf1:=strtofloat(edit16.Text); wf2:=strtofloat(edit17.Text);
wf3:=strtofloat(edit18.Text); deltaD1:=((d1*p)+wf1)*w1;
deltaD2:=((d2*p)+wf2)*w2; deltaD3:=((d3*p)+wf3)*w3;
stringgrid4.Cells[0,1]:=floattostr(deltaD1);
stringgrid4.Cells[1,1]:=floattostr(deltaD2);
stringgrid4.Cells[2,1]:=floattostr(deltaD3);
end;

procedure TForm6.BitBtn1Click(Sender: TObject);
var i:integer;
begin
  n:=strtoint(form4.Edit5.Text);
  for i:=1 to n do
  begin
    c1[i]:=StrToFloat(StringGrid1.Cells[i-1,1]);
  end;
end;

procedure TForm6.BitBtn5Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to n do
  begin
    deltaC1[i]:=strtoint(form5.StringGrid1.Cells[i-1,1])*c1[i];
    deltaC2[i]:=strtoint(form5.StringGrid1.Cells[i-1,2])*c1[i];
    deltaC3[i]:=strtoint(form5.StringGrid1.Cells[i-1,3])*c1[i];
    stringgrid2.Cells[i-1,1]:=floatToStr(deltaC1[i]);
    stringgrid2.Cells[i-1,2]:=floatToStr(deltaC2[i]);
    stringgrid2.Cells[i-1,3]:=floatToStr(deltaC3[i]);
  end; end;

procedure TForm6.Button2Click(Sender: TObject);
begin
  dur1:=strtofloat(edit12.Text); dur2:=strtofloat(edit13.Text);

```

```

dur3:=strtofloat(edit14.Text); sdur1:=dur1/2;
sdur2:=dur2/2; sdur3:=dur3/2;
edit8.text:=floattostr(sdur1); edit19.text:=floattostr(sdur2);
edit20.text:=floattostr(sdur3);
end;

procedure TForm6.BitBtn2Click(Sender: TObject);
begin
  deltaD1:=0; deltaD2:=0; deltaD3:=0; w1:=0; w2:=0; w3:=0;
end;
end.

```

Maksimum Penurunan Durasi

Perhitungan		Input Data		Output Hasil Peningkatan Biaya							
Dk*	<input type="text"/>	<input type="text"/>	<input type="text"/>	Presentase Penurunan Durasi	<input type="text"/>						
Durasi	<input type="text"/>	<input type="text"/>	<input type="text"/>	Waktu Float Dikoreksi	<input type="text"/>						
1/2 D	<input type="text"/>	<input type="text"/>	<input type="text"/>	delta D	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>						
Jumlah D	<input type="text"/>	<input type="text"/>	<input type="text"/>	Beta	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>						
Inputan PLk	<input type="text"/>	<input type="text"/>	<input type="text"/>	Nilai w							
Nilai w	<input type="text"/> <input type="text"/>	<input type="text"/> <input type="text"/>	<input type="text"/> <input type="text"/>								
Input Data		Maksimal Presentase Peningkatan Biaya									
		19	20	21	22	1	2	3	4	5	
		0.23	0.4	0.34	0.6	4,2	65,25	26,1	80,15	158,5	
		< >				5,4	67,5	28,5	81,9	161,5	
		< >				6,6	68,85	31,2	84	163,5	
		< >				OK Proses					

Maksimum Penurunan Durasi

Perhitungan			Input Data		
Dk*	98	109	125	Presentase Penurunan Durasi	0,6
Durasi	9	12	14	Waktu Float Dikoreksi	0 0 0
1/2 D	4,5	6	7	delta D	1 2 3 3,982014385 6,842975205 10,36261682
Jumlah D	102,5	115	132	Beta	
Inputkan PLk	107	121	139	Nilai w	
Nilai w	1	2	3		
	0,737410071	0,950413223	1,233644859		

Output Hasil Peningkatan Biaya

1	2	3	4	5

Lampiran 9. Hasil Perhitungan Nilai Perlindungan yang Tersedia pada Masing-masing Lintasan.

NO	<i>Listing Program</i>
	<pre> procedure TForm4.Button1Click(Sender: TObject); var i:integer; begin n:=strtoint(edit5.Text); m:=StrToInt(Edit1.Text); StringGrid1.RowCount:=4; StringGrid1.ColCount:=m; StringGrid5.RowCount:=4; StringGrid5.ColCount:=m; form6.Visible:=true; form6.StringGrid1.RowCount:=2; form6.StringGrid1.ColCount:=n; form6.StringGrid2.RowCount:=4; form6.StringGrid2.ColCount:=n; for i:=1 to m do begin StringGrid1.Cells[i-1,0]:=IntToStr(i); StringGrid5.Cells[i-1,0]:=IntToStr(i); end; for i:=1 to n do begin Form6.StringGrid1.Cells[i-1,0]:=IntToStr(i); Form6.StringGrid2.Cells[i-1,0]:=IntToStr(i); end; end;</pre>
	<pre> procedure TForm4.Button4Click(Sender: TObject); var i:integer;</pre>

```

begin
  for i:=1 to m do
    begin
      pd1[i]:=StrTofloat(StringGrid1.Cells[i-1,1]);
      pd2[i]:=StrTofloat(StringGrid1.Cells[i-1,2]);
      pd3[i]:=StrTofloat(StringGrid1.Cells[i-1,3]);
      pb1[i]:=StrTofloat(StringGrid5.Cells[i-1,1]);
      pb2[i]:=StrTofloat(StringGrid5.Cells[i-1,2]);
      pb3[i]:=StrTofloat(StringGrid5.Cells[i-1,3]);
    end;
    for i:=1 to m do
      begin
        pt1[i]:=pd1[i]/pb3[i]; pt2[i]:=pd2[i]/pb2[i];
        pt3[i]:=pd3[i]/pb1[i];
      end;
      for i:=1 to m do
        begin
          jum1:=jum1+pt1[i]; jum2:=jum2+pt2[i];
          jum3:=jum3+pt3[i];
        end;
        edit2.Text:=floattostr(jum1); edit3.Text:=floattostr(jum2);
        edit4.Text:=floattostr(jum3);
      end;

procedure TForm4.BitBtn1Click(Sender: TObject);
begin
  jum1:=0; jum2:=0; jum3:=0;
end;

procedure TForm4.Button2Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to m do
    begin
      StringGrid1.Cells[i-1,1]:=""; StringGrid1.Cells[i-1,2]:="";
      StringGrid1.Cells[i-1,3]:=""; StringGrid5.Cells[i-1,1]:="";
      StringGrid5.Cells[i-1,2]:=""; StringGrid5.Cells[i-1,3]:="";
    end; end;
end.

```

Perlindungan yang tersedia

Input Data

Banyak Aktivitas pada Lintasan ke-k

Banyak Aktivitas Proyek

Penurunan Durasi

4	5	6	7	
0,721	0,638	1,827	3,982	
1,805	1,901	2,894	6,843	
7,467	8,243	4,802	10,363	

Peningkatan Biaya

4	5	6	7	
15,41	14	94,52	3	
17,02	15,2	96,9	4,2	
17,94	17,2	99,96	4,8	

Perlindungan yang Tersedia

Lampiran 10. Hasil Perhitungan Nilai *Risk of Major Design Revision* (RMR) Pada Masing-masing Lintasan.

NO	<i>Listing Program</i>
	<pre>TForm8.Button1Click(Sender: TObject); var i:integer; begin pd[1]:=strtofloat(edit7.text)*strtofloat(edit8.text);</pre>

```

pd[2]:=strtofloat(edit9.text)*strtobool(edit10.text);
pd[3]:=strtobool(edit11.text)*strtobool(edit12.text);
pd[4]:=strtobool(edit13.text)*strtobool(edit14.text);
for i:=1 to 4 do
begin
  durA[i]:=strtobool(edit1.text)*pd[i];
  durB[i]:=strtobool(edit2.text)*pd[i];
  durC[i]:=strtobool(edit3.text)*pd[i];
end;
for i:=1 to 4 do
begin
  jumdurA:=jumdurA+durA[i];
  jumdurB:=jumdurB+durB[i];
  jumdurC:=jumdurC+durC[i];
end;
end;

procedure TForm8.BitBtn1Click(Sender: TObject);
begin
  RC1:=jumdurA-strtobool(edit21.Text);
  RC2:=jumdurB-strtobool(edit20.Text);
  RC3:=jumdurC-strtobool(edit19.Text);
  edit4.Text:=floattostr(RC1); edit5.Text:=floattostr(RC2);
  edit6.Text:=floattostr(RC3);
end;

procedure TForm8.Button2Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to n do
  begin
    a[i]:=StrTofloat(StringGrid1.Cells[i-1,1]);
    b[i]:=StrTofloat(StringGrid1.Cells[i-1,2]);
    c[i]:=StrTofloat(StringGrid1.Cells[i-1,3]);
  end;
  for i:=1 to n do
  begin
    StringGrid2.Cells[i-1,0]:=IntToStr(i);
    ComboBox1.Items.Add('d'+IntToStr(i));
  end;
end;

```

```

end; end;

procedure TForm8.Button4Click(Sender: TObject);
var i:integer;
    s:array[1..100]of integer;
begin
j:=j+1; i:=ComboBox1.ItemIndex;
if RadioGroup1.ItemIndex=1 then
begin
kal1:=kal1+ComboBox1.Items[i]+ '=';
s[j]:=i;
begin
sum1:=sum1+StrTofloat(StringGrid1.Cells[s[j],1]);
sum2:=sum2+StrTofloat(StringGrid1.Cells[s[j],2]);
sum3:=sum3+StrTofloat(StringGrid1.Cells[s[j],3]);
end; end
else
begin
kal1:=kal1+ComboBox1.Items[i]+ '+';
s[j]:=i;
begin
sum1:=sum1+StrTofloat(StringGrid1.Cells[s[j],1]);
sum2:=sum2+StrTofloat(StringGrid1.Cells[s[j],2]);
sum3:=sum3+StrTofloat(StringGrid1.Cells[s[j],3]);
end; end;
Edit15.Text:=kal1; edit16.text:=floatToStr(sum1);
edit17.text:=floafToStr(sum2);
edit18.text:=floatToStr(sum3);
end;

procedure TForm8.Button3Click(Sender: TObject);
var i:integer;
begin
n:=strtoint(edit22.Text); m:=strtoint(edit23.text);
StringGrid1.ColCount:=n; StringGrid1.RowCount:=4;
StringGrid2.ColCount:=m; StringGrid2.RowCount:=4;
for i:=1 to n do
begin
StringGrid1.cells[i-1,0]:=inttostr(i);

```

```

end;
for i:=1 to m do
begin
  StringGrid2.cells[i-1,0]:=inttostr(i);
end;
end;

procedure TForm8.Button5Click(Sender: TObject);
begin
kal1:=""; RadioGroup1.ItemIndex:=-1;
ComboBox1.ItemIndex:=-1; edit15.Text:="";
sum1:=0; sum2:=0; sum3:=0;
end;

procedure TForm8.Button6Click(Sender: TObject);
begin
edit16.Text:=""; edit17.Text:=""; edit18.Text:="";
end;

procedure TForm8.Button7Click(Sender: TObject);
begin
edit1.Text:=""; edit2.Text:=""; edit3.Text:=""; edit19.Text:="";
edit20.Text:=""; edit21.Text:=""; jumdurA:=0; jumdurB:=0;
jumdurC:=0;
end;
end.

```

Resiko Pengulangan

Inputkan pz-->dz	Inputan RCij	Perhitungan																				
Banyak Aktivitas <input type="text" value="22"/>	<input type="button" value="OK"/>	Masukkan Linkasan :																				
Banyak Lintasan <input type="text" value="10"/>	<table border="1"> <tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr> <tr><td>-6,3</td><td>-3,582</td><td>-4,926</td><td>5,0225</td><td>-5,7963</td></tr> <tr><td>-1,1</td><td>1,769</td><td>0,3885</td><td>0,36</td><td>-0,4284</td></tr> <tr><td>0</td><td>3,171</td><td>1,603</td><td>2,0075</td><td>1,0074</td></tr> </table>	16	17	18	19	20	-6,3	-3,582	-4,926	5,0225	-5,7963	-1,1	1,769	0,3885	0,36	-0,4284	0	3,171	1,603	2,0075	1,0074	<input type="button" value="ok"/>
16	17	18	19	20																		
-6,3	-3,582	-4,926	5,0225	-5,7963																		
-1,1	1,769	0,3885	0,36	-0,4284																		
0	3,171	1,603	2,0075	1,0074																		
pz ----> dz	<input type="button" value="OK"/>	<input type="button" value="Hapus"/>																				
<table border="1"> <tr><td>0,65</td><td>---> 0</td></tr> <tr><td>0,02</td><td>---> 1</td></tr> <tr><td>0,13</td><td>---> 0,5</td></tr> <tr><td>0,2</td><td>---> 0,33</td></tr> </table>	0,65	---> 0	0,02	---> 1	0,13	---> 0,5	0,2	---> 0,33	<input type="button" value="OK"/>													
0,65	---> 0																					
0,02	---> 1																					
0,13	---> 0,5																					
0,2	---> 0,33																					
Dij : <table border="1"> <tr><td>18</td><td>19</td><td>21</td></tr> </table>	18	19	21																			
18	19	21																				
Fj dikoreksi :																						
RCij :																						
<input type="button" value="Hitung"/>																						
<input type="button" value="Hapus"/>																						

Inputan RMRk																				
<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>-6,0495</td><td>-0,8186</td><td>2,5105</td><td>7,7414</td><td>9,566</td></tr> <tr><td>12,8445</td><td>13,5353</td><td>21,0105</td><td>21,7035</td><td>24,7825</td></tr> <tr><td>15,746</td><td>16,1025</td><td>24,4975</td><td>24,854</td><td>27,8225</td></tr> </table>	1	2	3	4	5	-6,0495	-0,8186	2,5105	7,7414	9,566	12,8445	13,5353	21,0105	21,7035	24,7825	15,746	16,1025	24,4975	24,854	27,8225
1	2	3	4	5																
-6,0495	-0,8186	2,5105	7,7414	9,566																
12,8445	13,5353	21,0105	21,7035	24,7825																
15,746	16,1025	24,4975	24,854	27,8225																
Hasil RMR																				
<table border="1"> <tr><td>-22,2016</td><td>5,8956</td><td>13,3496</td></tr> <tr><td><input type="button" value="Hapus"/></td><td></td><td></td></tr> </table>	-22,2016	5,8956	13,3496	<input type="button" value="Hapus"/>																
-22,2016	5,8956	13,3496																		
<input type="button" value="Hapus"/>																				

Lampiran 11. Hasil Perhitungan Nilai Matriks Normalisasi.

NO	<i>Listing Program</i>
	<pre>procedure TForm2.Button1Click(Sender: TObject); var i:integer; begin m:=strtoint(edit1.text); StringGrid1.RowCount:=m+1; StringGrid1.ColCount:=4; StringGrid2.RowCount:=m+1; StringGrid2.ColCount:=4; Form3.StringGrid2.RowCount:=m+1; Form3.StringGrid2.ColCount:=4; Form3.StringGrid3.RowCount:=2; Form3.StringGrid3.ColCount:=3; StringGrid3.RowCount:=2; StringGrid3.ColCount:=3; StringGrid4.RowCount:=2; StringGrid4.ColCount:=3; for i:=1 to m do begin StringGrid1.Cells[0,i]:=IntToStr(i); StringGrid2.Cells[0,i]:=IntToStr(i); form3.StringGrid2.Cells[0,i]:=IntToStr(i); end; for i:=1 to 3 do begin StringGrid1.Cells[i,0]:='J'+inttostr(i); StringGrid2.Cells[i,0]:='J'+inttostr(i); form3.StringGrid2.Cells[i,0]:='J'+inttostr(i); end; end; procedure TForm2.Button2Click(Sender: TObject); var i:integer; begin for i:=1 to m do begin x1[i]:=strtofloat(stringgrid1.Cells[1,i]); x2[i]:=strtofloat(stringgrid1.Cells[2,i]); x3[i]:=strtofloat(stringgrid1.Cells[3,i]); end; for i:=1 to 3 do begin</pre>

```
min[i]:=strtofloat(stringgrid3.cells[i-1,1]);
max[i]:=strtofloat(stringgrid4.cells[i-1,1]);
end;
for i:=1 to m do
begin
  ra1[i]:=x1[i]-min[1]; ra2[i]:=x2[i]-min[2];
  ra3[i]:=x3[i]-min[3]; rb1[i]:=max[1]-min[1];
  rb2[i]:=max[2]-min[2]; rb3[i]:=max[3]-min[3];
end;
for i:=1 to m do
begin
  r1[i]:=ra1[i]/rb1[i]; r2[i]:=ra2[i]/rb2[i];
  r3[i]:=ra3[i]/rb3[i];
end;
for i:=1 to m do
begin
  stringgrid2.Cells[1,i]:=floattostr(r1[i]);
  stringgrid2.Cells[2,i]:=floattostr(r2[i]);
  stringgrid2.Cells[3,i]:=floattostr(r3[i]);
end;
end;

procedure TForm2.BitBtn1Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to m do
  begin
    r1[i]:=0; r2[i]:=0; r3[i]:=0;
  end; end;
end.
```

Matriks Normalisasi

Inputkan Data				Output Data																		
Banyak Lintasan	10	OK	$\sim R_{ij}$																			
X _{ij}	<table border="1"> <tr><td></td><td>J1</td><td>J2</td><td>J3</td></tr> <tr><td>1</td><td>-6,05</td><td>12,845</td><td>15,746</td></tr> <tr><td>2</td><td>-0,819</td><td>13,535</td><td>16,103</td></tr> <tr><td>3</td><td>2,511</td><td>21,011</td><td>24,498</td></tr> </table>				J1	J2	J3	1	-6,05	12,845	15,746	2	-0,819	13,535	16,103	3	2,511	21,011	24,498			
	J1	J2	J3																			
1	-6,05	12,845	15,746																			
2	-0,819	13,535	16,103																			
3	2,511	21,011	24,498																			
min (i) X _{ij}	<table border="1"> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>-39,067</td><td>-3,572</td></tr> <tr><td></td><td></td><td>13,35</td></tr> </table>							-39,067	-3,572			13,35										
	-39,067	-3,572																				
		13,35																				
max (i) X _{ij}	<table border="1"> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>35,995</td><td>25,505</td></tr> <tr><td></td><td></td><td>28,162</td></tr> </table>							35,995	25,505			28,162										
	35,995	25,505																				
		28,162																				
				Proses																		

Lampiran 12. Hasil Perhitungan Nilai Matriks Normalisasi Terbobot.

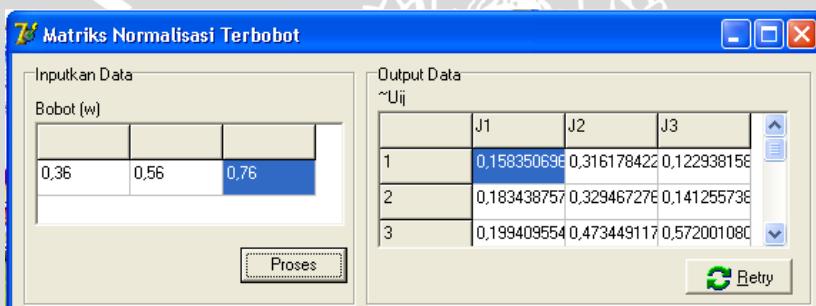
NO	<i>Listing Program</i>
	<pre> procedure TForm3.Button1Click(Sender: TObject); var i:integer; begin m:=strtoint(form2.edit1.text); w1:=strtofloat(stringgrid3.cells[0,1]); w2:=strtofloat(stringgrid3.cells[1,1]); w3:=strtofloat(stringgrid3.cells[2,1]); for i:=1 to m do begin x1[i]:=strtofloat(form2.stringgrid2.Cells[1,i]); x2[i]:=strtofloat(form2.stringgrid2.Cells[2,i]); x3[i]:=strtofloat(form2.stringgrid2.Cells[3,i]); end; for i:=1 to m do begin u1[i]:=x1[i]*w1; u2[i]:=x2[i]*w2; u3[i]:=x3[i]*w3; end; for i:=1 to m do begin </pre>

```

stringgrid2.Cells[1,i]:=floattostr(u1[i]);
stringgrid2.Cells[2,i]:=floattostr(u2[i]);
stringgrid2.Cells[3,i]:=floattostr(u3[i]);
end; end;

procedure TForm3.BitBtn1Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to m do
  begin
    u1[i]:=0; u2[i]:=0; u3[i]:=0;
  end; end;
end.

```



Lampiran 13. Hasil Perhitungan Nilai *Euler Distance* dan Derajat Kekritisian Lintasan.

NO	<i>Listing Program</i>
	<pre> procedure TForm4.Button1Click(Sender: TObject); var i:integer; begin l:=strToInt(edit1.text); StringGrid1.RowCount:=4; StringGrid1.ColCount:=l+1; StringGrid2.RowCount:=4; StringGrid2.ColCount:=l+1; for i:=1 to l do begin StringGrid1.Cells[i,0]:=IntToStr(i); StringGrid2.Cells[i,0]:=IntToStr(i); end; </pre>

```

for i:=1 to 3 do
begin
  StringGrid1.Cells[0,i]:='u'+IntToStr(i);
  StringGrid2.Cells[0,i]:='w'+IntToStr(i);
end; end;

procedure TForm4.Button2Click(Sender: TObject);
var i:integer;
begin
  for i:=1 to 1 do
  begin
    u1[i]:=strtofloat(stringgrid1.Cells[i,1]);
    u2[i]:=strtofloat(stringgrid1.Cells[i,2]);
    u3[i]:=strtofloat(stringgrid1.Cells[i,3]);
  end;
  for i:=1 to 1 do
  begin
    w1[i]:=strtofloat(stringgrid2.Cells[i,1]);
    w2[i]:=strtofloat(stringgrid2.Cells[i,2]);
    w3[i]:=strtofloat(stringgrid2.Cells[i,3]);
  end;
  for i:=1 to 1 do
  begin
    da[i]:=(sqr(u1[i]-w1[i])+sqr(u2[i]-w2[i])+sqr(u3[i]-
w3[i]))/3;
  end;
  for i:=1 to 1 do
  begin
    dp:=dp+da[i];
  end;
  dpos:=sqrt(dp); edit4.Text:=floattostr(dpos);
  for i:=1 to 1 do
  begin
    db[i]:=(sqr(u1[i])+sqr(u2[i])+sqr(u3[i]))/3;
  end;
  for i:=1 to 1 do
  begin
    dn:=dn+db[i];
  end;

```

```

dneg:=sqrt(dn); edit5.Text:=floattostr(dneg);
end;

procedure TForm4.Button3Click(Sender: TObject);
begin
  c:=dneg/(dpos+dneg); edit2.Text:=floattostr(c);
end;

procedure TForm4.BitBtn1Click(Sender: TObject);
var i:integer;
begin
  c:=0; dneg:=0; dpos:=0;
  for i:=0 to 1 do
  begin
    da[i]:=0; db[i]:=0;
  end;
  dp:=0; dn:=0;
  for i:=1 to 1 do
  begin
    stringgrid1.Cells[i,1]:=""; stringgrid1.Cells[i,2]:="";
    stringgrid1.Cells[i,3]:="";
  end; end;
end.

```

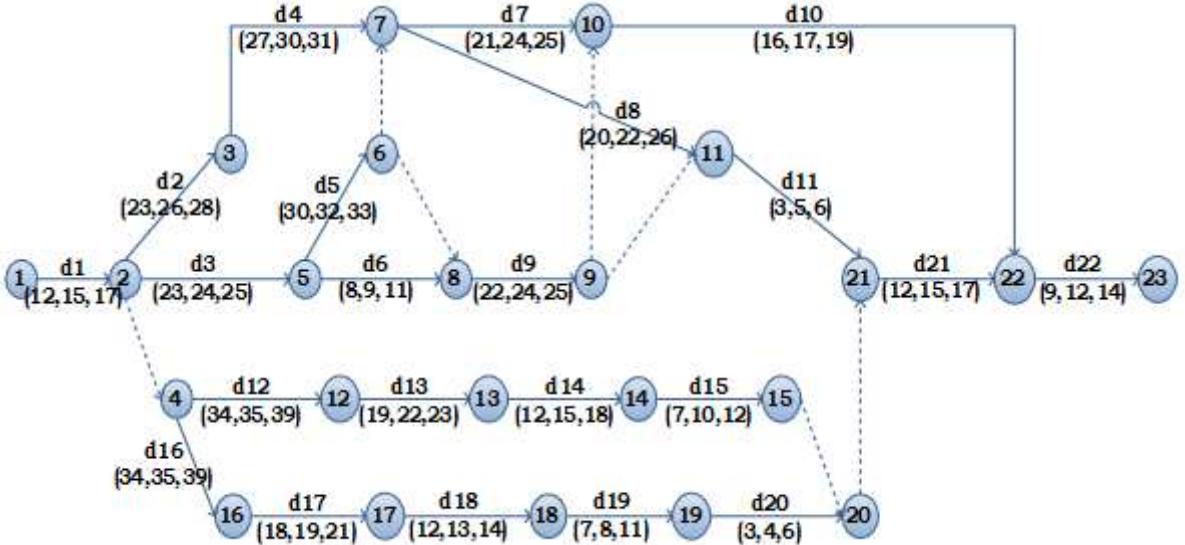
Euler Distance

Nilai Jarak	<input type="text" value="5"/>	<input type="button" value="OK"/>		
~Ui pada Lintasan Ke-k				
	3	4	5	
u1	0,378	0,376	0,081	
u2	0,486	0,432	0,182	
u3	0,536	0,598	0	
	<input type="button" value="<"/>	<input type="button" value=">"/>		

	1	2	3	4
w1	0,56	0,23	0,7	0,6
w2	0,66	0,43	0,9	0,8
w3	0,93	0,63	1	0,9
	<input type="button" value="<"/>	<input type="button" value=">"/>		

Euler distance (jarak euler) dan derajat kekrisan:

D+ (i)	0,56538933333	D- (i)	1,057496	<input type="button" value="Kekrisan"/>	C (i)	0,65161473720	<input type="button" value="Retry"/>
--------	---------------	--------	----------	---	-------	---------------	--------------------------------------



Gambar 3.11 Jaringan Proyek

Tabel 3.27 Matriks *Multy Kriteria Decision Making* (MCDM)

Panjang Lintasan	Kekritisan Lintasan	Jumlah Biaya	Perlindungan yang Tersedia	Resiko Revisi Desain Utama
[108,123,134]	[0,77; 0,97; 1,2]	[495,520,539]	[1,039 ; 1,93 ; 4,21]	[-6,05 ; 12,845 ; 15,746]
[106,124,139]	[0,76; 0,98; 1,24]	[1084,1116,1148]	[1,008 ; 1,891 ; 4,002]	[-0,819 ; 13,535 ; 16,103]
[111,124,133]	[0,79; 0,98; 0,19]	[525,554,577]	[1,161 ; 2,024 ; 4,267]	[2,511 ; 21,011 ; 24,498]
[109,125,138]	[0,78; 0,98; 1,23]	[1114,1150,1186]	[0,804 ; 1,57 ; 4,055]	[7,741 ; 21,704 ; 24,854]
[112,124,133]	[0,8; 0,98; 1,19]	[650,682,712]	[1,034 ; 1,821 ; 3,741]	[9,566 ; 24,783 ; 27,823]
[111,127,137]	[0,79; 1; 1,22]	[1037,1075,1116]	[0,873 ; 1,637 ; 3,306]	[12,638 ; 25,505 ; 28,162]
[90,101,111]	[0,64; 0,8; 0,99]	[361,390,422]	[1,067 ; 2,52 ; 5,974]	[-39,067 ; -3,572 ; 20,854]
[89,104,115]	[0,64; 0,82; 1,03]	[748,783,826]	[0,878 ; 2,31 ; 5,519]	[35,995 ; -2,85 ; 21,194]
[105,124,140]	[0,75; 0,98; 1,25]	[1182,1214,1249]	[0,969 ; 1,945 ; 4,409]	[-5,923 ; 17,097 ; 21,558]
[107,121,139]	[0,76; 0,95; 1,24]	[804,835,865]	[1,017 ; 2,065 ; 4,968]	[-22,202 ; 5,896 ; 13,35]

Tabel 3.28 Matriks Normalisasi

[0,826 ; 0,846 ; 0,793]	[0,813 ; 0,85 ; 0,953]	[0,163 ; 0,158 ; 0,141]	[0,658 ; 0,379 ; 0,339]	[0,44 ; 0,565 ; 0,162]
[0,739 ; 0,885 ; 0,966]	[0,75 ; 0,9 ; 0,991]	[0,881 ; 0,881 ; 0,878]	[0,571 ; 0,338 ; 0,261]	[0,51 ; 0,588 ; 0,186]
[0,957 ; 0,9885 ; 0,758]	[0,938 ; 0,9 ; 0]	[0,12 ; 0,12 ; 0,187]	[1 ; 0,478 ; 0,36]	[0,554 ; 0,845 ; 0,753]
[0,87 ; 0,923 ; 0,931]	[0,875 ; 0,9 ; 0,981]	[0,917 ; 0,922 ; 0,928]	[0 ; 0 ; 0,281]	[0,624 ; 0,869 ; 0,777]
[1 ; 0,885 ; 0,759]	[1 ; 0,9 ; 0,943]	[0,352 ; 0,354 ; 0,351]	[0,644 ; 0,264 ; 0,163]	[0,648 ; 0,975 ; 0,977]
[0,957 ; 1 ; 0,897]	[0,938 ; 1 ; 0,972]	[0,823 ; 0,831 ; 0,84]	[0,193 ; 0,071 ; 0]	[0,689 ; 1 ; 1]
[0,043 ; 0 ; 0]	[0 ; 0 ; 0,755]	[0 ; 0 ; 0]	[0,737 ; 1 ; 1]	[0 ; 0 ; 0,507]
[0 ; 0,115 ; 0,138]	[0 ; 0 ; 0,792]	[0,471 ; 0,477 ; 0,489]	[0,21 ; 0,779 ; 0,83]	[1 ; 0,025 ; 0,53]
[0,696 ; 0,885 ; 1]	[0,688 ; 0,9 ; 1]	[1 ; 1 ; 1]	[0,462 ; 0,395 ; 0,413]	[0,442 ; 0,712 ; 0,554]
[0,783 ; 0,769 ; 0,966]	[0,75 ; 0,75 ; 0,991]	[0,54 ; 0,54 ; 0,54]	[0,56 ; 0,521 ; 0,623]	[0,225 ; 0,33 ; 0]

Tabel 3.29 Matriks Normalisasi Terbobot

[0,463 ; 0,558 ; 0,738]	[0,187 ; 0,366 ; 0,6]	[0,114 ; 0,142 ; 0,141]	[0,415 ; 0,315 ; 0,325]	[0,158 ; 0,316 ; 0,123]
[0,414 ; 0,584 ; 0,899]	[0,173 ; 0,387 ; 0,624]	[0,616 ; 0,793 ; 0,878]	[0,36 ; 0,28 ; 0,25]	[0,183 ; 0,33 ; 0,141]
[0,536 ; 0,584 ; 0,706]	[0,216 ; 0,387 ; 0]	[0,14 ; 0,18 ; 0,187]	[0,63 ; 0,397 ; 0,346]	[0,199 ; 0,475 ; 0,572]
[0,487 ; 0,609 ; 0,866]	[0,201 ; 0,387 ; 0,618]	[0,642 ; 0,83 ; 0,924]	[0 ; 0 ; 0,27]	[0,224 ; 0,487 ; 0,59]
[0,56 ; 0,584 ; 0,706]	[0,23 ; 0,387 ; 0,594]	[0,246 ; 0,319 ; 0,351]	[0,406 ; 0,22 ; 0,157]	[0,233 ; 0,546 ; 0,743]
[0,536 ; 0,66 ; 0,834]	0,216 ; 0,43 ; 0,612]	[0,576 ; 0,748 ; 0,84]	[0,121 ; 0,056 ; 0]	[0,248 ; 0,56 ; 0,76]
[0,024 ; 0 ; 0]	[0 ; 0 ; 0,475]	[0 ; 0 ; 0]	[0,464 ; 0,83 ; 0,96]	[0 ; 0 ; 0,385]
[0 ; 0,076 ; 0,128]	[0 ; 0 ; 0,5]	[0,33 ; 0,43 ; 0,489]	[0,131 ; 0,647 ; 0,796]	[0,36 ; 0,014 ; 0,402]
[0,39 ; 0,584 ; 0,93]	[0,158 ; 0,387 ; 0,63]	[0,7 ; 0,9 ; 1]	[0,291 ; 0,328 ; 0,397]	[0,159 ; 0,398 ; 0,421]
[0,439 ; 0,508 ; 0,898]	[0,173 ; 0,323 ; 0,624]	[0,378 ; 0,486 ; 0,536]	[0,376 ; 0,432 ; 0,598]	[0,081 ; 0,182 ; 0]