

**GENERATING MINIMUM RECTILINEAR  
STEINER SPANNING TREE**

**Case Study: Water Pipeline at Central Building, Campus III USD**

**S k r i p s i**



Disusun Oleh:

Melia Angelita Jaya, Lim

NIM : 005314005

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2004**



**GENERATING MINIMUM RECTILINEAR  
STEINER SPANNING TREE**

**Case Study: Water Pipeline at Campus III USD**

S k r i p s i

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
Jurusan Teknik Informatika



Disusun Oleh:  
Melia Angelita Jaya, Lim  
NIM: 005314005

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2004**

**HALAMAN PERSETUJUAN**

**SKRIPSI**

**GENERATING MINIMUM RECTILINEAR  
STEINER SPANNING TREE**

**Case Study: Water Pipeline at Central Building, Campus III USD**

Disusun Oleh:

Melia Angelita Jaya, Lim

NIM: 005314005

Telah disetujui oleh:

Pembimbing I



Drs. Jong Jek Siang, M.Sc.

Tanggal 21 Oktober 2004

Pembimbing II



Puspaningtyas S. Adi, S.T.

Tanggal 22 Oktober 2004

## SKRIPSI

### GENERATING MINIMUM RECTILINEAR STEINER SPANNING TREE

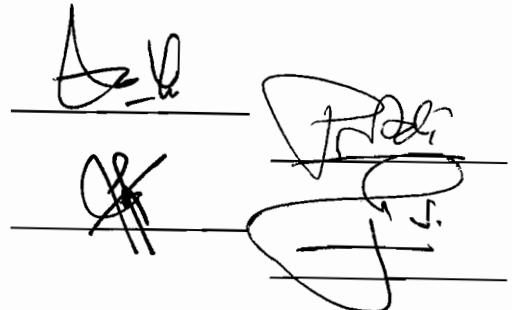
**Case Study: Water Pipeline at Central Building, Campus III USD**

Dipersiapkan dan ditulis oleh:  
Melia Angelita Jaya, Lim  
NIM: 005314005

Telah dipertahankan di depan Panitia Penguji  
Pada tanggal 18 Oktober 2004  
Dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Ketua A.M. Polina, S.Kom., M.Sc.  
Sekretaris Puspaningtyas S. Adi , S.T.  
Anggota Drs. J.J.Siang, M.Sc.  
Anggota Bambang Soelistijanto, S.T.



Yogyakarta, 18 Oktober 2004

Fakultas teknik

Universitas Sanata Dharma

Dekan



Ir. Greg Heliarko, S.J., S.S., B.S.T., M.A, M.Sc.

## **ABSTRACT**

### **GENERATING MINIMUM RECTILINEAR STEINER SPANNING TREE**

#### **Case Study: Water Pipeline at Central Building, Campus III USD**

In architecture, one of process that needs accurate calculation is generating water pipeline. In this case, placing water pipe and their connecting usually use vertical and horizontal direction. That will cause increasing the length of pipes needed and will eventually cause of the increasing costs too. Thus, if the length of water pipeline could be minimized, the cost could be reduced.

This research implements Minimum Rectilinear Steiner Spanning Tree with Iterated-1 Steiner Algorithm that enable to result the minimum water pipeline using several points namely Steiner points as branch. The shortest path among points (terminals and Steiner points) is calculated by using Prim Algorithm.

The result of this research is a program with points as destination place of the pipe that placed in the first quadrant with integer coordinate (x, y) as inputs and the output is a minimum water pipeline using Minimum Rectilinear Steiner Spanning Tree. In the experiment using 15 samples with 3 – 5 points as terminal, the average ratio for comparison between Minimum Rectilinear Steiner Spanning Tree and Minimum Rectilinear Spanning Tree is 0.822 and the average ratio for comparison between Minimum Rectilinear Steiner Spanning Tree and Minimum Spanning Tree is 1.064.

## **INTISARI**

### **PEMBENTUKAN MINIMUM RECTILINEAR STEINER SPANNING TREE**

**Studi Kasus: Jaringan Pipa Air di Gedung Pusat, Kampus III USD**

Dalam bidang arsitektur, salah satu proses yang memerlukan perhitungan cermat adalah pembentukan jaringan pipa air. Pada jaringan pipa air, penempatan pipa air dan sambungannya cenderung mengikuti arah vertikal dan horizontal. Hal ini dapat meningkatkan panjang pipa yang diperlukan dan akhirnya juga dapat meningkatkan biaya yang diperlukan. Jadi, jika panjang pipa pada jaringan pipa air dibuat seminimal mungkin maka jumlah biaya dapat dikurangi.

Tugas akhir ini mengimplementasikan Minimum Rectilinear Steiner Spanning Tree dengan Algoritma Iterated-1 Steiner yang dapat menghasilkan jalur pipa air yang minimum dengan menggunakan beberapa titik yang disebut steiner point sebagai titik percabangan. Jalur terpendek antar titik (terminal dan steiner point) dihitung menggunakan Algoritma Prim.

Hasil tugas akhir ini adalah sebuah program dengan titik – titik tempat tujuan pipa yang terletak pada kuadran I dengan koordinat integer (x,y) sebagai input dan outputnya adalah sebuah jaringan pipa air yang minimal menggunakan Minimum Rectilinear Steiner Spanning Tree. Pada percobaan menggunakan 15 buah sampel dengan jumlah terminal sebanyak 3 – 5, rasio rata-rata untuk perbandingan antara Minimum Rectilinear Steiner Spanning Tree dengan Minimum Rectilinear Spanning Tree adalah 0,822 dan rasio rata-rata untuk perbandingan antara Minimum Rectilinear Steiner Spanning Tree dengan Minimum Spanning Tree adalah 1,064.

## **PERNYATAAN KEASLIAN KARYA**

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka sebagaimana layaknya karya ilmiah.

Yogyakarta, awal Oktober 2004

Penulis



Melia Angelita Jaya, Lim

*Dedication for ...*

*Guru Shakyamuni Buddha,*

*Bhagavati Tara Devi,*

*All Buddha, Bodhisattva and Mahasattva*

*Papa & Mama, I Love you so much...*

*My little sister, you're very kind & sweet*

*Yayangku, Thanks for everything*

*All Sentient beings ...*

*May have happiness and its causes*

I am only one, but I am still one.

I cannot do everything, but still I can do something.

And because I cannot do everything

I will not refuse to do the something that I can do.

-- Helen Keller --

## **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan ke hadirat Tuhan yang telah memberi inspirasi, petunjuk dan kemurahan-Nya sehingga tugas akhir ini dapat diselesaikan. Tugas akhir ini disusun sebagai salah satu syarat guna memperoleh gelar Sarjana Teknik di Universitas Sanata Dharma Yogyakarta.

Proses penyusunan tugas akhir ini melibatkan banyak pribadi yang berkenan memberikan bantuan-bantuan yang amat berarti. Untuk itu, dalam kesempatan ini penulis menghaturkan ungkapan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Agnes Maria Polina S.Kom.,Msc. selaku Ketua Jurusan Teknik Informatika. Terima kasih atas semangat dan dukungan yang telah diberikan, khususnya selama pembuatan tugas akhir ini.
2. Bpk. Drs. Jong Jek Siang, M.Sc. selaku Dosen Pembimbing I yang telah memberikan kepercayaan kepada penulis untuk mengambil judul ini. Terima kasih atas kesabaran, curahan perhatian, dukungan dan bimbingan yang telah diberikan selama ini.
3. Bpk. Puspaningtyas S. Adi, S.T. selaku Dosen Pembimbing II. Terima kasih atas kesabaran, curahan perhatian, dukungan dan bimbingan yang telah diberikan selama ini.
4. Laboran Teknik Informatika, khususnya Bpk. Bele yang telah banyak membantu selama masa perkuliahan dan proses pendadaran. Terima kasih atas perhatian, waktu dan bantuannya.
5. Segenap dosen Fakultas Teknik Informatika. Terima kasih atas ilmu, dukungan, dan bimbingan yang telah diberikan selama ini.

6. Ibu Woro dan Bpk. Heru selaku petugas Biro Sarana dan Prasarana Kampus III Paingan yang telah berkenan memberikan data tentang jaringan pipa air dan penjelasan yang terperinci tentang cara membaca denah yang diberikan.
7. Mama dan papa tersayang, yang doanya tidak pernah terputus dan selalu hadir setiap saat. *I love you more than anything.*
8. Adikku yang manis, *mei-mei*, *Thanks a lot* karena selalu bersedia mendengarkan setiap *curhatku*, memberikan semangat dan pelukan di saat-saat sulit, *You're the best sister that I ever meet.*
9. *Yayankku*, Yanto Masyap. *Thanks for everything. May what we do bring us more close to enlightenment.*
10. Keluarga Besar Kost Zusi Arib, Bapak & Ibu kost yang sangat perhatian, anak-anak kost yang selalu ramai: Mba Tari, Mba Sarah, Tuti (*Thanks a lot* atas bantuan dan dampingannya), Mili, Ntrie, Alin ‘Iyut Pipi’, Cici ‘Tuk2’, Mitha, Imeth, Meme, dan yang baru saja bergabung: Duma, Evi, Siska, Pipin, serta semuanya yang belum disebutkan namanya, terima kasih atas dukungannya selama ini. *We are like family. I'll miss you all.*
11. Anak-anak GMCBP yang dulu dan sekarang, terima kasih atas dukungannya. *Spesial thanks for YM. Bhante Bodhi yang sudah mau berbagi cerita dan memberikan semangat dalam pengajaran skripsi ini.*
12. Teman-teman di jurusan Teknik Informatika; teman-teman seperjuangan: Iin (*My twin sister, Kapan nginep lagi? ☺*), Wawan Serva (*Thanks sudah ditemenin ☺*), Dewi (*Thanks atas doa dan semangatnya ☺*), Ninuk, teman-

teman angkatan 2000: Rosa, Cha2, Nanda, Nicko, Ismi, Ruby, Jerry, Evy, dan semuanya yang tidak dapat disebutkan satu persatu, serta kakak dan adik tingkat, terima kasih atas dukungannya selama ini.

13. Teman-teman *se-kampung*, yang selalu menyemangatiku selama pembuatan skripsi ini: Lenny, Wirla, Helen, Steven, Agung, Vika, Yohanes, Fidi, Fadli, sukses selalu untuk kalian semua!
14. Semua pihak yang telah membantu sampai skripsi ini terselesaikan, yang sudah maupun belum penulis sebutkan. *May Happiness come to you ☺*.

Penulis menyadari penyusunan skripsi ini masih banyak memerlukan evaluasi sebagai dasar untuk pengembangan studi-studi ilmiah serupa di masa yang akan datang. Oleh sebab itu penulis terbuka dalam menerima berbagai kritik dan saran dari seluruh pembaca.

Semoga siapapun yang membaca dan menggunakan karya ini mendapatkan sesuatu yang baik dan bermanfaat.

Yogyakarta, awal Oktober 2004

Penulis

Melia Angelita Jaya, Lim

DAFTAR ISI

## HALAMAN JUDUL

## HALAMAN PERSETUJUAN

## HALAMAN PENGESAHAN

## ABSTRAK

## HALAMAN PERSEMBAHAN

KATA MUTIARA

KATA PENGANTAR

DAFTAR ISI

## DAFTAR GAMBA

DAFTAR TABEL

## BAB I PENDAHULUAN

1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan dan Manfaat	3
1.5. Sistematika Penulisan	3

## BAB II LANDASAN TEORI

2.1. Graf	5
2.1.1. Derajat	6
2.1.2. Sirkuit	6
2.1.3. Pohon (Tree)	6
2.1.3.1. Pohon Rentang Minimum (Minimum Spanning Tree/MST)	7
2.1.3.2. Penyelesaian Pohon Rentang Minimum	7
2.2. Steiner Tree	8
2.2.1. Rectilinear Steiner Tree Problem	9
2.2.2. Penyelesaian Rectilinear Steiner Tree Problem	11



### **BAB III PERANCANGAN SISTEM**

3.1. Analisis Sistem	26
3.1.1. Identifikasi Masalah	26
3.1.2. Ruang Lingkup Masalah	26
3.2. Tahap Desain Model	27
3.2.1. Menentukan Model Pemecahan Masalah	27
3.3. Perancangan Antar Muka	27
3.3.1. Tampilan Awal	30
3.3.2. Tampilan Utama	30
3.3.3. Tampilan Menu New	32
3.3.4. Tampilan Form Edit	33
3.3.5. Tampilan Form Add	34
3.3.6. Tampilan Form Delete	35
3.3.7. Tampilan Toolbar	35
3.3.8. Tampilan Form Process	36
3.3.9. Perancangan Penyimpanan	37
3.4. Perancangan Proses	38
3.4.1. Lingkungan Perangkat Lunak	38
3.4.2. Lingkungan Perangkat Keras	39
3.4.3 Struktur Data	39
3.4.4 Algoritma dan Flowchart	41

### **BAB IV HASIL DAN PEMBAHASAN**

4.1. Implementasi Program	44
4.1.1. Algoritma Prim	44
4.1.2. Hanan Poin	48
4.1.3 Algoritma Iterated 1-Steiner	49
4.1.4 Proses Penyimpanan Data Terminal	52
4.1.5 Proses Open Data Terminal	53
4.2. Implementasi Antar Muka	55

4.2.1 Implementasi Tampilan pembuka	55
4.2.2 Implementasi Tampilan Utama	56
4.2.3 Implementasi Tampilan Form New	58
4.2.4 Implementasi Tampilan Open	59
4.2.5 Implementasi Tampilan Form Edit	60
4.2.6 Implementasi Tampilan Form Add	61
4.2.7 Implementasi Tampilan Form Delete	62
4.2.8 Implementasi Tampilan Form Proses & hasil	63
4.3. Pembahasan	65
4.3.1 Studi Kasus Jaringan Pipa Air Kampus III USD	65
4.3.2 Uji Coba Perhitungan	75
4.4. Kelebihan Program	81
4.5. Kekurangan Program	82
4.6. Alternatif Pengembangan	82
<b>BAB V PENUTUP</b>	
5.1 Kesimpulan	83
5.2 Saran	84
<b>DAFTAR PUSTAKA</b>	85
<b>LAMPIRAN A</b>	
Data & Hasil Perhitungan Jaringan Pipa Air Kampus III USD	
<b>LAMPIRAN B</b>	
Listing Program	

## **DAFTAR GAMBAR**

Gambar 2.1.	Graf	5
Gambar 2.2.	Pohon	7
Gambar 2.3.	Minimum Rectilinear Spanning Tree (Kiri) dan Minimum Rectilinear Steiner Tree (Kanan)	10
Gambar 2.4.	Hanan Grid	10
Gambar 3.1.	Flowchart antarmuka	28
Gambar 3.2.	Rancangan Tampilan menu startup	30
Gambar 3.3.	Rancangan Tampilan Utama	30
Gambar 3.4.	Rancangan Struktur Menu	32
Gambar 3.5.	Rancangan Form Skala	32
Gambar 3.6	Rancangan Lembar Kerja	33
Gambar 3.7.	Rancangan Edit Data Terminal	34
Gambar 3.8.	Rancangan Form Add	34
Gambar 3.9.	Rancangan Form Delete	35
Gambar 3.10.	Rancangan Toolbar	35
Gambar 3.11.	Rancangan Form Process	37
Gambar 3.12.	Flowchart Algoritma IIS	42
Gambar 3.13.	Flowchar Algoritma Prim	43
Gambar 4.1.	Form Tampilan Awal	56
Gambar 4.2.	Tampilan Utama Awal	56
Gambar 4.3.	Form Skala & Satuan	58
Gambar 4.4.	Lembar Kerja	59
Gambar 4.5.	Menu Open	59
Gambar 4.6.	Input Jumlah Terminal	60
Gambar 4.7.	Form Edit Terminal	60
Gambar 4.8.	Validasi Kesalahan	61
Gambar 4.9.	Form Add	62
Gambar 4.10	Form Delete	62

Gambar 4.11	Menu Process dan Hasilnya	63
Gambar 4.12	Jaringan Pipa Air untuk Air Bersih	67
Gambar 4.13	Jaringan Pipa Air untuk Air Kotor	70
Gambar 4.14	Jaringan Pipa Air untuk Air Kotoran	73
Gambar 4.15	Gambar Data Uji Coba	77
Gambar 4.16	Data Perbandingan Jarak	79
Gambar 4.17	Minimal Rectilinear Spanning Tree (kiri) dan Minimal Spanning Tree (kanan)	80

## **DAFTAR TABEL**

Tabel 4.1.	Data Air Bersih	66
Tabel 4.2.	Data Air Kotor	68
Tabel 4.3	Data Air Kotoran	71
Tabel 4.4	Data Uji Coba	75

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang Masalah**

Dalam bidang arsitektur, banyak proses yang memerlukan perencanaan dan perhitungan yang matang sebelum diimplementasikan dalam bentuk nyata. Salah satu proses yang memerlukan perhitungan cermat adalah pembentukan jaringan pipa air.

Dalam pembentukan jaringan pipa air, terdapat beberapa elemen penting yang terkait, misalnya sumber air (sumur, penampungan air), tempat – tempat yang membutuhkan air (seperti wastafel, kamar mandi), atau tempat pembuangan air kotor. Tempat – tempat tersebut dihubungkan dengan pipa air sehingga air bersih atau air kotor dapat mengalir.

Pada pembentukan jaringan pipa air, biaya yang dikeluarkan sebanding dengan panjang pipa yang digunakan sehingga semakin panjang pipa, semakin besar biayanya. Oleh karena itu dibutuhkan suatu mekanisme perhitungan untuk membantu membentuk jaringan pipa air dengan panjang pipa seminimal mungkin sehingga dapat menekan jumlah biaya yang harus dikeluarkan.

Salah satu metode yang dapat digunakan untuk meminimalkan panjang pipa pada jaringan pipa air adalah dengan menggunakan Rectilinear Steiner Spanning Tree, yaitu bagaimana menghubungkan semua titik/terminal (tidak ada terminal yang terpencil) dengan menambahkan beberapa titik bantu (*Steiner Points*), sehingga menghasilkan panjang lintasan seminimal mungkin. Pada

pembentukan jaringan ini, pengukuran jarak yang digunakan adalah rectilinear, yaitu setiap node hanya dihubungkan dengan garis vertikal dan horizontal.

Dengan metode ini, sumber air, tempat yang membutuhkan air dan tempat pembuangan air dianggap sebagai terminal yang akan dihubungkan dengan pipa air. Peletakkan titik percabangan yang tepat (steiner point) akan dapat meminimalkan panjang pipa air yang dibutuhkan.

Dari uraian tersebut, maka penulis tertarik untuk mengambil topik “Pembentukan Minimum Rectilinear Steiner Spanning Tree”. Untuk studi kasusnya menggunakan salah satu jaringan pipa air pada Kampus III Universitas Sanata Dharma.

## 1.2. Rumusan Masalah

Dari latar belakang masalah yang telah dikemukakan oleh penulis, akan menjadi menarik kiranya jika dibuat rumusan masalah sebagai berikut: ‘*Bagaimana mendesain sebuah aplikasi sistem bantu untuk simulasi pada perencanaan pembuatan jaringan pipa air yang optimal menggunakan Rectilinear Steiner Spanning Tree ?*’

## 1.3. Batasan Masalah

Dalam desain ini, batasan masalah meliputi hal – hal sebagai berikut:

1. Sistem dibatasi pada aspek simulasi pembentukan jaringan pipa air menggunakan Rectilinear Steiner Spanning Tree.
2. Algoritma yang akan digunakan adalah Iterated 1-Steiner (I1S).

3. Bahasa Pemrograman yang digunakan adalah Visual Basic 6.0
4. Seluruh bidang / daerah adalah daerah bebas, dalam arti dapat dilalui oleh pipa air atau dibuat titik percabangan.

#### **1.4 Tujuan dan Manfaat**

Tujuan dan manfaat dari penulisan ini adalah membuat aplikasi perangkat lunak yang berfungsi sebagai simulasi sistem bantu untuk mengambil keputusan pada pembentukan jaringan pipa air dengan menggunakan Rectilinear Steiner Spanning Tree

#### **1.5. Sistematika Penulisan**

Sistematika penulisan tugas akhir ini terbagi atas lima bab dengan garis besar sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat, dan sistematika penulisan

#### **BAB II LANDASAN TEORI**

Bab ini berisi landasan teori yang dipakai untuk pembahasan penulisan tugas akhir ini.

#### **BAB III PERANCANGAN SISTEM**

Bab ini membahas tentang gambaran perancangan sistem aplikasi yang dibuat.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi implementasi dari rancangan sistem yang dibuat pada bab sebelumnya serta pembahasan tentang program yang dibuat.

## **BAB V PENUTUP**

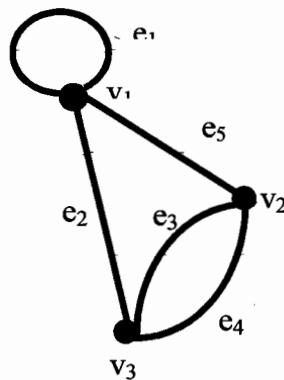
Bab ini berisi kesimpulan dan saran dalam penulisan tugas akhir ini.

## BAB II

### LANDASAN TEORI

#### 2.1 Graf

Suatu graf terdiri dari dua himpunan yang berhingga, yaitu himpunan titik – titik tidak kosong (simbol  $V(G)$ ) dan himpunan garis – garis (simbol  $E(G)$ ). Setiap garis berhubungan dengan satu atau dua titik. Titik – titik tersebut dinamakan titik ujung. Garis yang hanya berhubungan dengan satu titik ujung disebut loop. Dua garis yang berbeda yang menghubungkan titik yang sama disebut garis pararel.



Gambar 2.1. *Graf*

Pada Gambar 2.1 dapat dilihat sebuah graf dengan  $V(G) = \{ v_1, v_2, v_3 \}$  dan  $E(G) = \{ e_1, e_2, e_3, e_4, e_5 \}$ . Titik – titik ujung garis adalah:

Garis	Titik Ujung
$e_1$	$\{v_1\}$
$e_2$	$\{v_1, v_3\}$
$e_3$	$\{v_2, v_3\}$
$e_4$	$\{v_2, v_3\}$
$e_5$	$\{v_1, v_2\}$

Dari gambar 2.1 dapat dilihat bahwa garis  $e_1$  merupakan loop, sedangkan garis  $e_3$  dan  $e_4$  merupakan garis paralel karena keduanya menghubungkan titik  $v_2$  dengan  $v_3$ .

### **2.1.1. Derajat**

Misalkan  $v$  adalah titik dalam suatu graf  $G$ . Derajat titik  $v$  adalah jumlah garis yang berhubungan dengan titik  $v$ . Pada loop, garisnya dihitung dua kali. Pada gambar 2.1 derajat titik  $v_1 = 3$  karena garis yang berhubungan dengan  $v_1$  adalah  $e_2$  dan loop  $e_1$ .

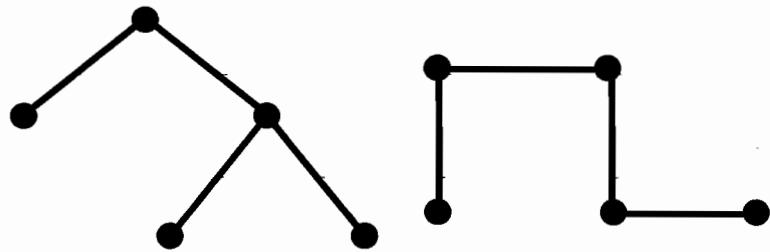
### **2.1.2. Sirkuit**

Suatu sirkuit adalah barisan titik – titik yang dihubungkan dengan garis secara berselang seling dengan semua garis dan titiknya berbeda, namun dimulai dan diakhiri pada titik yang sama.

Contoh sirkuit pada Gambar 2.1 adalah  $v_1 e_2 v_3 e_3 v_2 e_5 v_1$

### **2.1.3. Pohon (Tree)**

Misalkan  $G$  adalah suatu graf sederhana, yaitu graf yang tidak memiliki garis paralel dan loop.  $G$  dapat disebut pohon bila dan hanya bila  $G$  tidak memuat sirkuit dan terhubung.



Gambar 2.2 *Pohon*

### 2.1.3.1 Pohon Rentang Minimum (Minimum Spanning Tree/MST)

Setiap garis dalam graf berhubungan dengan suatu label yang menyatakan bobot garis tersebut. Dalam aplikasinya, bobot suatu garis lebih tepat dibaca sebagai “jarak”, “biaya”, “panjang”, ”kapasitas”, dan lain – lain.

Graf berlabel adalah suatu graf tanpa garis paralel di mana setiap garisnya berhubungan dengan suatu bilangan riil tak negatif yang menyatakan bobot garis tersebut.

Aplikasi yang sering dipakai dalam graf berlabel adalah mencari pohon rentang dengan bobot seminimum mungkin (sering disebut pohon rentang minimum/minimum spanning tree), yaitu mencari jalur yang menghubungkan semua titik sehingga total bobot garis menjadi seminimum mungkin.

### 2.1.3.2 Penyelesaian Pohon Rentang Minimum

Salah satu metode untuk mencari pohon rentang minimum adalah dengan menggunakan Algoritma Prim. Algoritma ini dimulai dengan graf kosong.

Misalkan ingin dicari pohon rentang minimum  $T$  dari graf  $G$  dengan Algoritma Prim, maka tahap – tahapnya:

0. Inisialisasi: Mula – mula  $T$  adalah graf kosong
1. Ambil sembarang  $v \in V(G)$ . Masukkan  $v$  ke dalam  $V(T)$
2.  $V(G) = V(G) - \{v\}$ .
3. Untuk  $i = 1, 2, \dots, n-1$  lakukan:
  - a. Pilihlah garis  $e \in E(G)$  dan  $e \notin E(T)$  dengan syarat
    - i.  $e$  berhubungan dengan satu titik dalam  $T$  dan tidak membentuk sirkuit.
    - ii.  $E$  mempunyai bobot terkecil dibandingkan dengan semua garis yang berhubungan dengan titik – titik dalam  $T$ . Misalkan  $w$  adalah titik ujung  $e$  yang tidak berada dalam  $T$ .
  - b. Tambahkan  $e$  ke  $E(T)$  dan  $w$  ke  $V(T)$
  - c.  $V(G) = V(G) - \{w\}$

## **2.2. Steiner Tree**

*Steiner Tree Problem* mempunyai sejarah yang panjang dalam matematika terapan. Masalah dimulai ketika Fermat (1601 – 1665) bertanya “Bagaimana saya menemukan sebuah titik sehingga dapat menghasilkan jarak total terpendek untuk setiap tiga titik yang diberikan pada suatu bidang?” Pertanyaan ini akhirnya berhasil dipecahkan oleh Torricelli.

Pada tahun 1934 Tarnik dan Kossler mempunyai masalah untuk mencari jarak terpendek yang menghubungkan titik – titik sebanyak  $n$  buah pada suatu

bidang. Masalah mereka kemudian dipopulerkan dalam sebuah buku berjudul “What is Mathematics?” (1941) dengan nama Steiner.

Sebenarnya, *Steiner Tree problem* adalah masalah mencari jarak terpendek yang menjangkau semua kumpulan titik yang diberikan. Titik – titik tersebut dikenal sebagai terminal. Pemecahan masalah ini mengijinkan dibuatnya titik dan garis baru sehingga dapat membuat jaringan dengan biaya seminimal mungkin. Titik – titik yang ditambahkan dikenal sebagai *steiner point* (titik steiner). Jadi sebenarnya *Steiner Tree Problem* adalah generalisasi dari pertanyaan Fermat yang mengijinkan ditambahkannya titik – titik bantu secara bebas pada sejumlah titik tertentu (tidak hanya tiga).

*Steiner Tree Problem* terus mengalami perkembangan. Tahun 1965 Hannan memperkenalkan masalah *rectilinear* dan pada tahun 1971 Hakimi dan Levin memperkenalkan *steiner tree problem* pada jaringan.

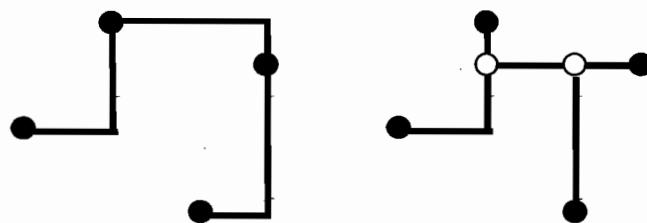
### 2.2.1 *Rectilinear Steiner Tree Problem*

*Rectilinear Steiner Spanning Tree Problem* dapat didefinisikan sebagai berikut: diberikan sekumpulan terminal  $P$  sejumlah  $n$  pada bidang kartesian, temukan sekumpulan titik  $S$  sebagai kandidat *steiner point*, kemudian dicari hubungan yang paling pendek dari terminal – terminal dan *steiner point* (yang dipilih) menggunakan garis vertikal atau horizontal, sehingga  $P \cup S$  mempunyai biaya yang paling minimum.

Pada *Rectilinear Steiner Tree problem*, jarak didefinisikan sebagai matriks L1 (Rectilinear Metric/Manhattan Metric). Jarak antara dua titik merupakan penjumlahan dari selisih koordinat x dan koordinat y.

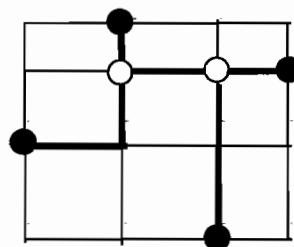
$$d(a,b) = |X_a - X_b| + |Y_a - Y_b|$$

Pada gambar 2.3 dapat dilihat perbedaan antara Pohon Rentang Minimum (MST) yang menggunakan pengukuran jarak secara rectilinear dan Minimum Rectilinear Steiner Tree (MRST). Pada MRST terdapat 2 buah steiner point (point yang tidak berisi) yang ditambahkan sehingga jarak antar terminal menjadi lebih pendek dibandingkan dengan MST.



Gambar 2.3 *Minimum Rectilinear Spanning Tree (Kiri)* dan *Minimum Rectilinear Steiner Tree (kanan)*

Menurut Hannan, *Rectilinear Steiner Tree* yang optimum didapat dengan menggunakan *steiner point* yang lokasinya merupakan perpotongan antara garis vertikal dan horizontal yang melewati terminal (selanjutnya disebut *Hanan Grid*), sedangkan titik – titik yang dihasilkan disebut *Hanan Point*.



Gambar 2.4. *Hanan Grid*

Secara umum, ciri – ciri *Rectilinear Steiner Tree* adalah sebagai berikut:

1. Derajat maksimal terminal adalah empat.
2. Derajat *steiner point* adalah tiga atau empat.
3. Hanya membutuhkan paling banyak  $n - 2$  steiner point dengan  $n$  adalah jumlah terminal yang diberikan.
4. Grid bergaris dibuat secukupnya. Cara yang dianjurkan untuk membuat grid ini adalah menggunakan *Hanan Grid*.

### **2.2.2 Penyelesaian *Rectilinear Steiner Tree Problem***

Salah satu algoritma yang digunakan dalam pemecahan *Rectilinear Steiner Tree problem* adalah Iterated 1-Steiner (I1S) dari Kahng dan Robins. Algoritmanya didefinisikan sebagai berikut:

Untuk 2 kumpulan titik, yaitu  $P$  (terminal yang diinputkan) dan  $S$  (steiner point), Minimum Rectilinear Steiner Tree (MRST) menyimpan  $S$  dengan memperhatikan  $P$  sebagai:

$$\Delta \text{ MST}(P, S) = \text{cost}(\text{MST}(P)) - \text{cost}(\text{MST}(P \cup S))$$

*Hanan Point* ( $H(P)$ ) merupakan titik perpotongan dari semua garis vertikal dan horizontal yang melewati terminal – terminal pada  $P$ .  $H(P)$  digunakan untuk menunjukkan kumpulan kandidat *Steiner Point*. Dalam sebuah himpunan dengan  $n$  terminal terdapat kemungkinan *Hanan Point* sebanyak  $n^2$ .

Untuk semua kumpulan titik  $P$ , sebuah *1-Steiner point*  $x \in H(P)$  memaksimalkan  $\Delta \text{ MST}(P, \{x\}) > 0$ . Metode I1S akan diulang untuk menemukan 1-Steiner point dan memasukkannya pada  $S$ . Biaya MST melalui PUS akan

menurun dengan ditambahkannya *steiner point* dan penambahan steiner point akan berhenti jika tidak ada lagi  $x$  yang memenuhi  $\Delta \text{MST}((\text{PUS}), \{x\}) > 0$ .

Meskipun steiner tree dapat memuat paling banyak  $n - 2$  steiner point, IIS dapat menambah lebih dari  $n - 2$  steiner point. Oleh karena itu, pada setiap langkah, steiner point yang mempunyai derajat  $\leq 2$  akan dihapus pada PUS.

**Input :** Sekumpulan titik  $P$  yang terdiri dari  $n$  terminal

**Output :** *Rectilinear Steiner Spanning Tree* yang melewati  $P$

$S = \emptyset$

While  $T = \{ x \in H(P) \mid \Delta \text{MST}(\text{PUS}, \{x\}) > 0 \} \neq \emptyset$  Do

{     Carilah  $x \in T$  dengan maksimal  $\Delta \text{MST}(\text{PUS}, \{x\})$

$S = S \cup \{x\}$

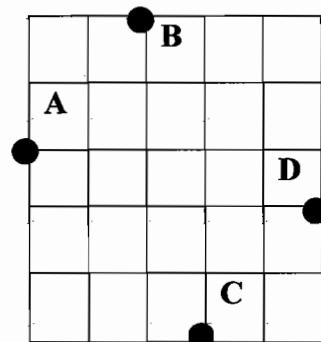
Hapus titik – titik  $S$  yang berderajat  $\leq 2$  pada MST (PUS)

}

Output MST (PUS)

Contoh Soal.

Diketahui 4 buah titik, A(0,3), B(2,5), C(3,0) dan D(5,2). Carilah jarak terpendek diantara keempat titik tersebut menggunakan *Rectilinear Steiner Tree*. (Pada contoh soal ini, koordinat (0,0) berada pada pojok kiri bawah)



### Penyelesaian (iterasi 1)

Input: 4 buah titik A(0,3), B(2,5), C(3,0) dan D(5,2).

$$S = \emptyset$$

Mencari Hannan Steiner Point H(P)

	3	5	0	2
0	(0,3)	(0,5)	(0,0)	(0,2)
2	(2,3)	(2,5)	(2,0)	(2,2)
3	(3,3)	(3,5)	(3,0)	(3,2)
5	(5,3)	(5,5)	(5,0)	(5,2)

Terbentuk 12 Hanan Point,  $x = \{ (0,5), (0,0), (0,2), (2,3), (2,0), (2,2), (3,3), (3,5), (3,2), (5,3), (5,5), (5,0) \}$

Mencari jarak antar terminal A(0,3), B(2,5), C(3,0) dan D(5,2).

$$d(a,b) = |X_a - X_b| + |Y_a - Y_b|$$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

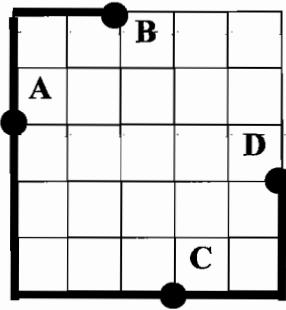
Mencari MST menggunakan Algoritma Prim

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula- mula	-	A	-

1	AB	B	-
2	AC	C	Pilih antara AC,AD,BC,BD
3	CD	D	-

Jadi  $MST(P)$  yang terbentuk:  $AB - AC - CD = 4 + 6 + 4 = 14$

Karena  $S = \emptyset$  maka  $MST(PUS) = MST(P) = 14$



Mencari jarak antar terminal dan Hanan Point

a.  $x = (0,5)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-0| + |3-5| = 0 + 2 = 2$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-0| + |5-5| = 2 + 0 = 2$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-0| + |0-5| = 3 + 5 = 8$$

$$Dx = |5-0| + |2-5| = 5 + 3 = 8$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	Ax	x	-
2	Bx	B	-
3	AC	C	Pilih antara AC,AD,BC,BD
4	CD	D	-

$$\text{Jadi } \text{MST}(\text{PUSUx}) = Ax - Bx - AC - CD = 2 + 2 + 6 + 4 = 14$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 14 = 0$$

b.  $x = (0,0)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-0| + |3-0| = 0 + 3 = 3$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-0| + |5-0| = 2 + 5 = 7$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-0| + |0-0| = 3 + 0 = 3$$

$$Dx = |5-0| + |2-0| = 5 + 2 = 7$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula- mula	-	A	-
1	Ax	x	-
2	Cx	C	-
3	AB	B	-
4	CD	D	-

$$\text{Jadi } \text{MST}(\text{PUSUx}) = Ax + Cx + AB + CD = 3 + 3 + 4 + 4 = 14$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 14 = 0$$

c.  $x = (0,2)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-0| + |3-2| = 0 + 1 = 1$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-0| + |5-2| = 2 + 3 = 5$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-0| + |0-2| = 3 + 2 = 5$$

$$Dx = |5-0| + |2-2| = 5 + 0 = 5$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula- mula	-	A	-
1	Ax	x	-
2	AB	B	-
3	Cx	C	Pilih antara Cx,Dx
4	CD	D	-

$$\text{Jadi } \text{MST}(\text{PUSUx}) = Ax + AB + Cx + CD = 1 + 4 + 5 + 4 = 14$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 14 = 0$$

d.  $x = (2,3)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-2| + |3-3| = 2 + 0 = 2$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-2| + |5-3| = 0 + 2 = 2$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-2| + |0-3| = 1 + 3 = 4$$

$$Dx = |5-2| + |2-3| = 3 + 1 = 4$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	Ax	x	-
2	Bx	B	-
3	Cx	C	Pilih antara Cx,Dx
4	CD	D	Pilih antara CD,Dx

$$\text{Jadi } \text{MST}(\text{PUSUx}) = Ax + Bx + Cx + CD = 2 + 2 + 4 + 4 = 12$$

$$\Delta \text{ MST } ((\text{PUS}), \{x\}) = \text{cost} (\text{MST } (\text{PUS})) - \text{cost} (\text{MST}(\text{PUSU}x)) = 14 - 12 = 2$$

e.  $x = (2,0)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-2| + |3-0| = 2 + 3 = 5$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-2| + |5-0| = 0 + 5 = 5$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-2| + |0-0| = 1 + 0 = 1$$

$$Dx = |5-2| + |2-0| = 3 + 2 = 5$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	AB	B	-
2	Ax	x	Pilih antara Ax,Bx
3	Cx	C	-
4	CD	D	-

$$\text{Jadi } \text{MST}(\text{PUSU}x) = AB + Ax + Cx + CD = 4 + 5 + 1 + 4 = 14$$

$$\Delta \text{ MST } ((\text{PUS}), \{x\}) = \text{cost} (\text{MST } (\text{PUS})) - \text{cost} (\text{MST}(\text{PUSU}x)) = 14 - 14 = 0$$

f.  $x = (2,2)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-2| + |3-2| = 2 + 1 = 3$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-2| + |5-2| = 0 + 3 = 3$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-2| + |0-2| = 1 + 2 = 3$$

$$Dx = |5-2| + |2-2| = 3 + 0 = 3$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	Ax	x	-
2	Bx	B	Pilih antara Bx,Cx,Dx
3	Cx	C	Pilih antara Cx,Dx
4	Dx	D	-

$$\text{Jadi MST(PUSUx)} = Ax + Bx + Cx + Dx = 3 + 3 + 3 + 3 = 12$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 12 = 2$$

g.  $x = (3,3)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-3| + |3-3| = 3 + 0 = 3$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-3| + |5-3| = 1 + 2 = 3$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-3| + |0-3| = 0 + 3 = 3$$

$$Dx = |5-3| + |2-3| = 2 + 1 = 3$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	Ax	x	-
2	Bx	B	Pilih antara Bx,Cx,Dx
3	Cx	C	Pilih antara Cx,Dx
4	Dx	D	-

$$\text{Jadi MST(PUSUx)} = Ax + Bx + Cx + Dx = 3 + 3 + 3 + 3 = 12$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 12 = 2$$

h.  $x = (3,5)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-3| + |3-5| = 3 + 2 = 5$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-3| + |5-5| = 1 + 0 = 1$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-3| + |0-5| = 0 + 5 = 5$$

$$Dx = |5-3| + |2-5| = 2 + 3 = 5$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	AB	B	-
2	Bx	x	-
3	Cx	C	Pilih antara Cx,Dx
4	CD	D	-

$$\text{Jadi } \text{MST}(\text{PUSUx}) = AB + Bx + Cx + CD = 4 + 1 + 5 + 4 = 14$$

$$\Delta \text{MST} ((\text{PUS}), \{x\}) = \text{cost} (\text{MST} (\text{PUS})) - \text{cost} (\text{MST}(\text{PUSUx})) = 14 - 14 = 0$$

i.  $x = (3,2)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-3| + |3-2| = 3 + 1 = 4$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-3| + |5-2| = 1 + 3 = 4$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-3| + |0-2| = 0 + 2 = 2$$

$$Dx = |5-3| + |2-2| = 2 + 0 = 2$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	AB	B	Pilih antara AB, Ax
2	Ax	x	Pilih antara Ax, Bx
3	Cx	C	Pilih antara Cx, Dx
4	Dx	D	-

$$\text{Jadi MST(PUSUx)} = AB + Ax + Cx + Dx = 4 + 4 + 2 + 2 = 12$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 12 = 2$$

j.  $x = (5,3)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-5| + |3-3| = 5 + 0 = 5$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-5| + |5-3| = 3 + 2 = 5$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-5| + |0-3| = 2 + 3 = 5$$

$$Dx = |5-5| + |2-3| = 0 + 1 = 1$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula-mula	-	A	-
1	AB	B	-
2	Ax	x	Pilih antara Ax,Bx
3	Dx	D	-
4	CD	C	-

$$\text{Jadi MST(PUSUx)} = AB + Ax + Dx + CD = 4 + 5 + 1 + 4 = 14$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 14 = 0$$

k.  $x = (5,5)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-5| + |3-5| = 5 + 2 = 7$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-5| + |5-5| = 3 + 0 = 3$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-5| + |0-5| = 2 + 5 = 7$$

$$Dx = |5-5| + |2-5| = 0 + 3 = 3$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula- mula	-	A	-
1	AB	B	-
2	Bx	x	-
3	Dx	D	-
4	CD	C	-

$$\text{Jadi } \text{MST}(\text{PUSUx}) = AB + Bx + Dx + CD = 4 + 3 + 3 + 4 = 14$$

$$\Delta \text{MST}((\text{PUS}), \{x\}) = \text{cost}(\text{MST}(\text{PUS})) - \text{cost}(\text{MST}(\text{PUSUx})) = 14 - 14 = 0$$

1.  $x = (5,0)$

$$AB = |0-2| + |3-5| = 2 + 2 = 4$$

$$AC = |0-3| + |3-0| = 3 + 3 = 6$$

$$AD = |0-5| + |3-2| = 5 + 1 = 6$$

$$Ax = |0-5| + |3-0| = 5 + 3 = 8$$

$$BC = |2-3| + |5-0| = 1 + 5 = 6$$

$$BD = |2-5| + |5-2| = 3 + 3 = 6$$

$$Bx = |2-5| + |5-0| = 3 + 5 = 8$$

$$CD = |3-5| + |0-2| = 2 + 2 = 4$$

$$Cx = |3-5| + |0-0| = 2 + 0 = 2$$

$$Dx = |5-5| + |2-0| = 0 + 2 = 2$$

Iterasi	Garis yang terpilih	Titik yang ditambahkan	Keterangan
Mula- mula	-	A	-
1	AB	B	-
2	AC	C	Pilih antara

			AC,AD,BC,BD
3	Cx	x	-
4	Dx	D	-

Jadi  $MST(PUS \cup x) = AB + AC + Cx + Dx = 4 + 6 + 2 + 2 = 14$

$\Delta MST((PUS), \{x\}) = \text{cost}(MST(PUS)) - \text{cost}(MST(PUS \cup x)) = 14 - 14 = 0$

While  $T = \{x \in H(P) \mid \Delta MST(PUS), \{x\} > 0\} \neq \emptyset$  Do

{ Carilah  $x \in T$  dengan maksimal  $\Delta MST(PUS), \{x\}$

$S = S \cup \{x\}$

Hapus titik – titik  $S$  yang berderajat  $\leq 2$  pada  $MST(PUS)$

}

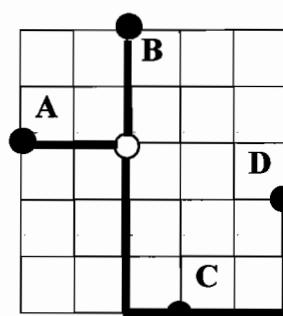
$T = \{x \in H(P) \mid \Delta MST(PUS), \{x\} > 0\}$  ada 4 buah sehingga memenuhi  $T = \{x \in H(P) \mid \Delta MST(PUS), \{x\} > 0\} \neq \emptyset$  maka

1. Carilah  $x \in T$  dengan maksimal  $\Delta MST(PUS), \{x\}$

Dipilih (2,3) (Pilih antara (2,3), (2,3),(3,3) dan (3,2)).

2.  $S = S \cup \{x\} = \emptyset + \{(2,2)\} = \{(2,2)\}$

3. Hapus titik – titik yang berderajat  $\leq 2$  pada  $MST(PUS)$



## **BAB III**

### **PERANCANGAN SISTEM**

#### **3.1. Analisis Sistem**

Analisis sistem dilakukan dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi serta kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan – perbaikannya. Adapun langkah – langkah analisis sistem adalah:

##### **3.1.1. Identifikasi masalah**

Pengidentifikasi masalah merupakan langkah pertama yang dilakukan dalam analisis sistem. Pada tugas akhir ini, masalah yang ditemukan adalah bagaimana membentuk jaringan pipa air dengan biaya seminimal mungkin. Perhitungan dengan sistem manual cukup lama dan memerlukan tingkat ketelitian yang tinggi, apalagi jika jumlah terminalnya banyak.

##### **3.1.2. Ruang Lingkup Masalah**

Pada aplikasi program bantu ini, faktor – faktor dalam pembentukan jaringan pipa air dilambangkan dengan tiga hal utama, yaitu:

- a. Terminal, merupakan suatu tempat sumber air atau tempat yang membutuhkan air, seperti sumur, *westafel*, *toilet* atau kamar mandi atau tempat pembuangan air kotor.
- b. Garis, merupakan pipa air.

- c. Steiner poin, merupakan titik percabangan pipa air, dalam hal ini dianggap percabangan berbentuk T atau +.

Pipa air dianggap hanya mempunyai dua arah, vertikal dan horizontal.

### **3.2. Tahap Desain Model**

#### **3.2.1 Menentukan Model Pemecahan Masalah**

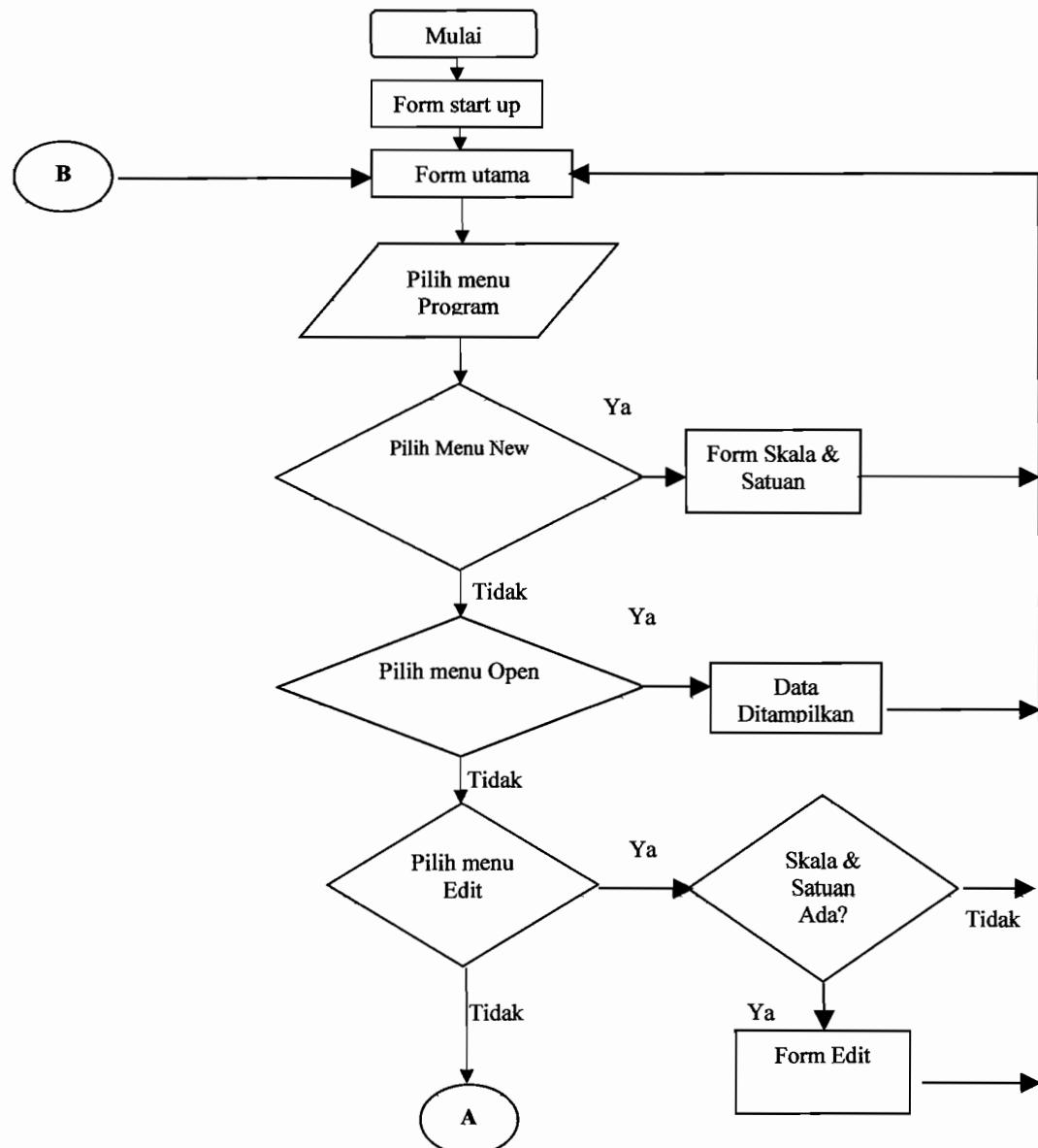
Pada dasarnya masalah pembentukan jaringan pipa air merupakan masalah yang menyangkut bagaimana membuat jaringan pipa air dengan biaya seminimal mungkin. Salah satu metode yang dapat digunakan untuk mencari jalur yang menghubungkan semua terminal sehingga jaraknya menjadi seminimum mungkin adalah dengan menggunakan Algoritma Prim.

Namun demikian, Algoritma Prim tidak mengijinkan adanya penambahan titik bantu, padahal penambahan titik – titik bantu dalam mencari jarak terpendek dapat memperpendek jarak yang dihasilkan. Oleh karena itu digunakan Minimum Steiner Tree sebagai metode untuk memecahkan permasalahan karena metode ini mengijinkan adanya penambahan titik bantu. Penempatan pipa air dan sambungannya cenderung mengikuti arah vertikal dan horizontal sehingga metode yang digunakan adalah Minimum Rectilinear Steiner Spanning Tree.

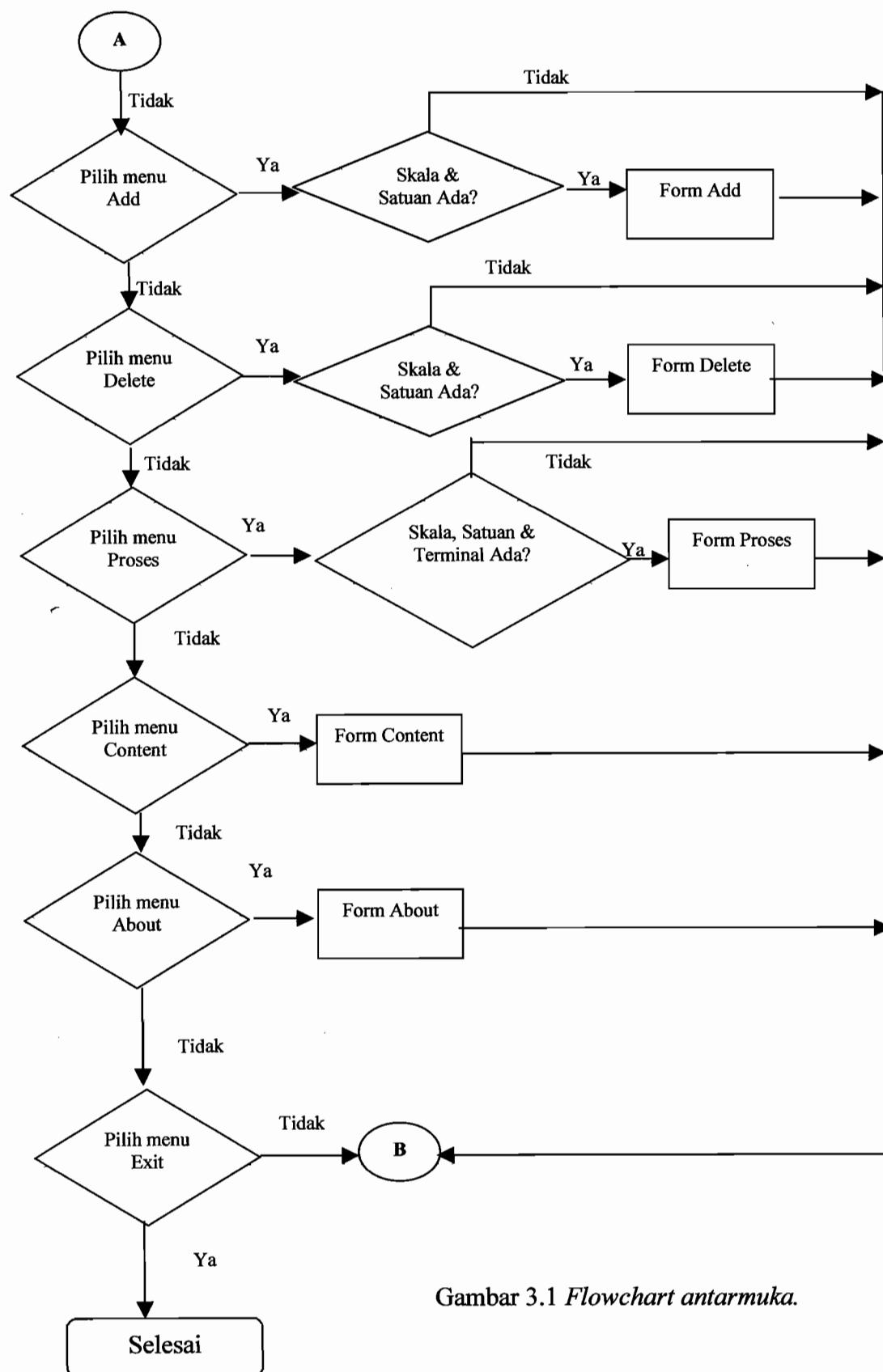
### **3.3. Perancangan Antarmuka**

Perancangan antarmuka digunakan untuk memberikan gambaran tentang sistem yang akan dibuat sehingga pengguna mudah berinteraksi dengan sistem. Pada tahap ini akan dibuat rancangan antarmuka perangkat lunak yang akan

dibuat. Antarmuka ini yang nantinya akan berinteraksi langsung dengan pengguna. Pada perancangan antarmuka tampilan proses dibagi menjadi beberapa *form*, dengan diagram alir sebagai berikut :



Gambar 3.1 *Flowchart* antarmuka.



Gambar 3.1 Flowchart antarmuka.

### 3.3.1. Tampilan Awal

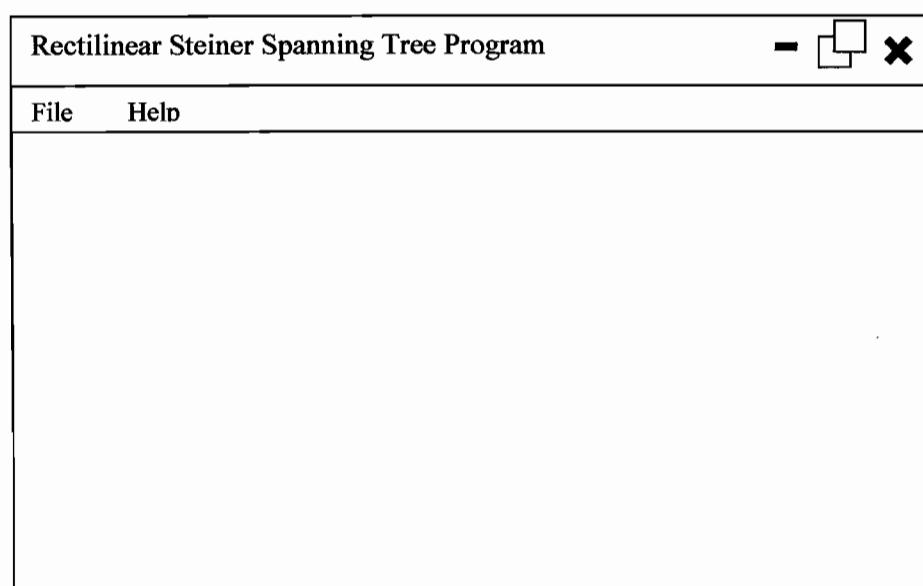
Tampilan awal berupa *form start up* untuk masuk ke menu utama. *Form start up* merupakan *form* yang secara otomatis menutup dengan menggunakan kontrol waktu.



Gambar 3.2 *Rancangan tampilan menu startup.*

### 3.3.2. Tampilan Utama

Gambar 3.2 berikut adalah merupakan rancangan tampilan utama di mana pengguna akan berinteraksi dengan program pada tampilan utama ini.

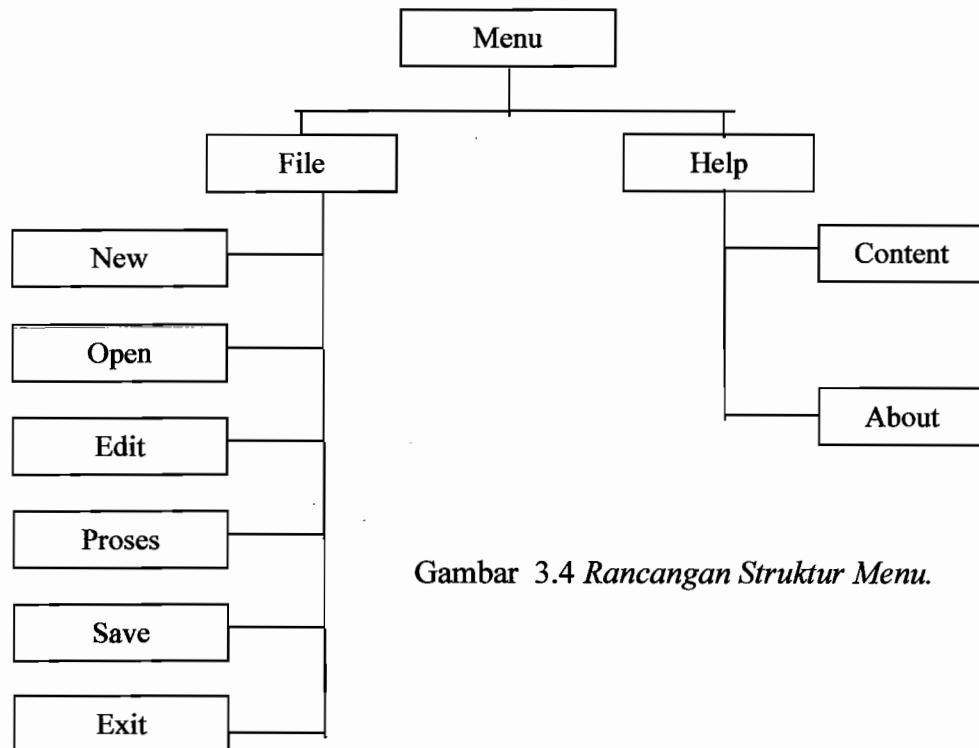


Gambar 3.3 *Rancangan tampilan utama.*

Pada tampilan utama ini terdapat dua menu pilihan, yaitu:

1. Menu File, terdiri dari enam sub menu, yaitu:
  - a. Menu New, digunakan untuk memanggil lembar kerja baru.
  - b. Menu Open, digunakan untuk membuka data yang sudah ada/disimpan sebelumnya.
  - c. Menu Edit, digunakan untuk mengubah data terminal, misalnya: menambah, mengurangi, atau mengganti nama dan koordinat terminal.
  - d. Menu Proses, digunakan untuk mengolah data yang telah diinputkan.
  - e. Menu Save, digunakan untuk menyimpan input data terminal.
  - f. Menu Exit, digunakan untuk keluar dari program
2. Menu Help, terdiri dari dua sub menu, yaitu:
  - a. Menu Content, merupakan menu panduan untuk menjalankan program
  - b. Menu About, merupakan menu informasi penyusun program.

Berikut struktur menu yang digunakan:



Gambar 3.4 Rancangan Struktur Menu.

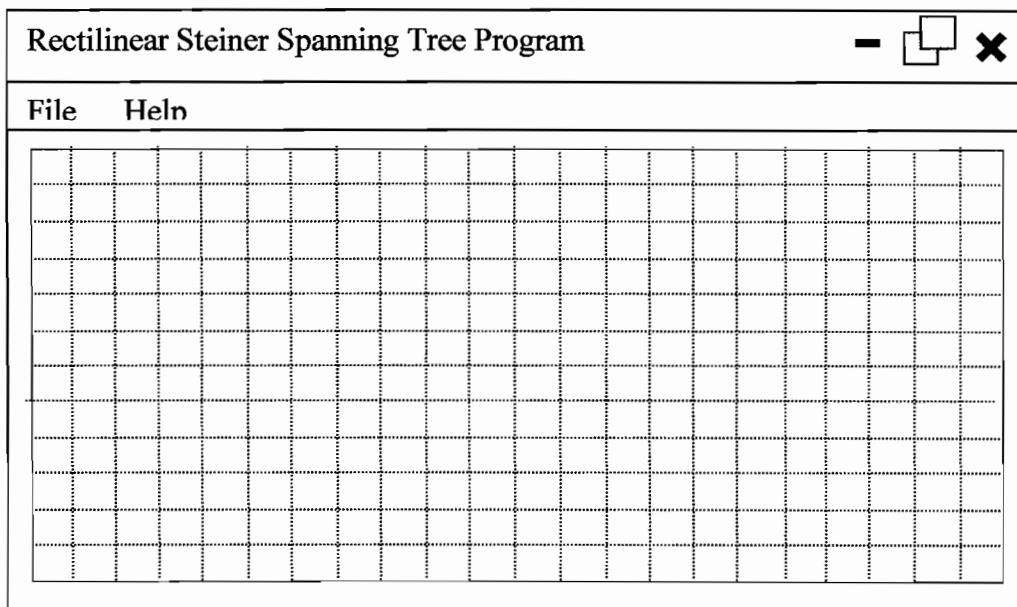
### 3.3.3. Tampilan Menu New

Saat menu File – New dipanggil, akan ditampilkan Form Skala & Satuan yang digunakan untuk menginputkan skala dan satuan yang akan digunakan pada gambar.

Form Skala & Satuan		<input type="button" value="X"/>
Satuan	<input type="text"/>	
Perbandingan Skala	1 : <input type="text"/>	
		<input type="button" value="OK"/> <input type="button" value="Cancel"/>

Gambar 3.5 Rancangan Form Skala

Setelah data skala diinputkan, pada tampilan utama (Gambar 3.3) akan dilengkapi dengan sebuah lembar kerja yang bergaris (menggunakan grid – grid).



Gambar 3.6 Rancangan Lembar Kerja

Lembar kerja ini digunakan untuk memudahkan pengguna dalam menginputkan data terminal. Pengguna dapat mengetahui koordinat daerah yang ditunjuk oleh *kursor mouse* serta dapat langsung menambahkan terminal di daerah yang ditunjuk oleh *kursor mouse* tersebut serta menghapus terminal yang telah ada. Koordinat yang diinputkan harus berupa 5 dan kelipatannya.

### 3.3.4. Tampilan Form Edit

Form ini digunakan untuk melihat dan mengubah data terminal, baik jumlah maupun nama serta koordinatnya. Selain menggunakan mouse, pengguna juga dapat menambahkan data terminal secara konvensional dengan

menginputkan nama terminal serta koordinat terminal (sumbu x dan y) yang nilainya merupakan 5 atau kelipatannya.

Rancangan Form Edit Data Terminal. Form ini menampilkan tabel dengan 4 kolom: Nomor, Nama, Pos X, dan Pos Y. Terdapat 3 baris kosong dalam tabel. Di sisi kanan, terdapat menu vertikal dengan opsi Add, Delete, Cancel, Finish, dan Exit. Form ini memiliki judul 'Data Terminal' di bagian atas kiri dan ikon tutup di bagian atas kanan.

Nomor	Nama	Pos X	Pos Y

Add  
 Delete  
 Cancel  
 Finish  
 Exit

Gambar 3.7 Rancangan Form Edit Data Terminal

### 3.3.5. Tampilan Form Add

Pengguna dapat menambahkan titik secara langsung dengan mengklik grid pada lembar kerja. Secara otomatis, program akan menampilkan Form Add. Pengguna hanya diminta untuk mengisi nama terminal, sedangkan pos X dan Pos Y secara otomatis akan ditampilkan berdasarkan koordinat tempat pengguna mengklik mouse.

Rancangan Form Add. Form ini memiliki tiga input text berturut-turut: Nama Terminal, PosX, dan PosY. Di bawahnya terdapat dua tombol: OK dan Cancel. Form ini memiliki judul 'Form Add' di bagian atas kiri dan ikon tutup di bagian atas kanan.

Nama Terminal  
 PosX  
 PosY

OK      Cancel

Gambar 3.8 Rancangan Form Add

### 3.3.6. Tampilan Form Delete

Jika pengguna ingin menghapus salah satu terminal, pengguna hanya perlu mengklik koordinat terminal yang ingin dihapus. Secara otomatis, Form ini akan untuk mengkonfirmasi nama terminal pada pengguna, apakah benar nama terminal tersebut akan dihapus dari lembar kerja.

Rancangan Form Delete yang menunjukkan tampilan dialog untuk mengonfirmasi penghapusan terminal. Form ini memiliki judul 'Form Delete' di bagian atas kiri dan ikon tutup (X) di bagian atas kanan. Di bagian tengah kiri terdapat label 'Nama Terminal' di samping sebuah kotak input. Di bagian bawah terdapat dua tombol: 'OK' dan 'Cancel' yang dikelompokkan dalam sebuah kotak.

Gambar 3.9 *Rancangan Form Delete*

### 3.3.7. Tampilan Toolbar

Toolbar berisi tombol – tombol yang digunakan untuk memudahkan interaksi dengan program. Toolbar ini berupa icon-icon dan diletakkan di bawah menu program sehingga dapat memudahkan user dalam memilih menu program. Tombol – tombol yang ada pada kotak tersebut hampir mewakili seluruh menu. Adapun rancangan Toolbor dapat dilihat pada gambar 3.10.



Gambar 3.10 *Rancangan Toolbor*

### 3.3.8. Tampilan Form Process

Menu File – Process dipilih jika ingin mengeksekusi program. Jika menu ini dipilih, maka pada lembar kerja akan ditampilkan jalur Rectilinear Steiner Spanning Tree Program yang diperoleh, yaitu adanya garis yang menghubungkan antar terminal dan steiner poin yang dihasilkan. Selain hasil pencarian yang berupa gambar, hasil pencarian yang berupa teks juga dapat dilihat pada form proses dan hasil (Gambar 3.11). Informasi yang akan ditampilkan adalah data terminal, data steiner poin, data jalur yang dilalui terminal, steiner poin, serta jumlah jarak yang dihasilkan. Pengguna dapat menyimpan dan mencetak hasil perhitungan serta mencetak hasil layout (gambar) yang ada pada lembar kerja. Selain itu, pengguna juga dapat melihat data perhitungan setiap iterasi pembentukan Rectilinear Steiner Tree atau perhitungan jarak antar terminal tanpa rectilinear metric dengan memilih *Hasil Perhitungan* yang diinginkan dan mengklik tombol Go untuk memprosesnya.

Form Proses & Hasil				X																				
<b>Hasil Perhitungan</b>																								
<input type="text"/>		<input type="checkbox"/>	GO																					
<b>Data Terminal &amp; Steiner Poin</b>																								
<table border="1"> <thead> <tr> <th>Nomor</th> <th>Nama</th> <th>Pos X</th> <th>Pos Y</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>					Nomor	Nama	Pos X	Pos Y																
Nomor	Nama	Pos X	Pos Y																					
<b>Data Jalur</b>		Skala 1: n																						
<table border="1"> <thead> <tr> <th>Nomor</th> <th>Awal</th> <th>Akhir</th> <th>Jarak</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>		Nomor	Awal	Akhir	Jarak																			
Nomor	Awal	Akhir	Jarak																					
Jarak Total		<input type="text"/>	satuan																					
<b>Proses Hasil</b>																								
<input type="text"/>		<input type="checkbox"/>	PROSES	<input type="button" value="EXIT"/>																				

Gambar 3.11 Rancangan Form Process

### 3.3.9. Perancangan Penyimpanan

Data yang diinputkan pengguna dapat disimpan (menu save) sehingga dapat digunakan lagi (menu open). Data yang akan disimpan adalah data tentang

skala, satuan, dan terminal (nama, Pos X dan Pos Y). Data ini akan disimpan pada file berekstensi .txt.

### **3.4 Perancangan Proses**

#### **3.4.1. Lingkungan Perangkat Lunak**

Perangkat lunak yang akan digunakan di dalam implementasi pembuatan aplikasi sistem bantu untuk simulasi pada perencanaan pembuatan jaringan pipa air menggunakan Rectilinear Steiner Spanning Tree adalah sebagai berikut:

- a. Sistem Operasi : Windows 1998
- b. Pengelolaan Antar Muka : Visual Basic 6.0

Perangkat lunak yang dipilih, disesuaikan dengan kompatibilitas dan fasilitas yang digunakan untuk mendukung pembangunan aplikasi. Aplikasi Tugas Akhir ini dibangun dengan menggunakan bahasa pemrograman Visual Basic 6.0.

Alasan pemilihan Visual Basic 6.0 sebagai alat bantu pengembangan adalah kemudahan bahasa pemrograman, karena bahasa pemrograman Basic mudah untuk dipelajari. Bahasa pemrograman ini juga sudah cukup populer di kalangan pemrogram komputer sehingga dengan penggunaan bahasa pemrograman ini, program aplikasi yang akan dibuat mudah untuk dikembangkan oleh *programmer* lainnya.

Selain itu, tersedianya fasilitas GUI yang mendukung untuk pengembangan aplikasi dan mampu menyelesaikan masalah. Sarana pengembangan yang bersifat visual dan bersifat *mouse-driven* (digerakkan dengan mouse) memungkinkan pembuatan aplikasi yang lebih *user friendly*.

Perangkat lunak ini juga kompatibel dengan sistem operasi yang digunakan yaitu Windows 98. Sebenarnya, Visual Basic 6.0 kompatibel dengan beberapa sistem operasi yang banyak digunakan orang yaitu Microsoft Windows 9x, Windows 2000, XP, NT yang merupakan sistem operasi 32 Bit.

### **3.4.2. Lingkungan perangkat keras**

Lingkungan perangkat keras yang sesuai dengan pembentukan jaringan pipa air dengan menggunakan Rectilinear Steiner Spanning Tree ini adalah:

- a. Komputer Pentium III, memori minimal 128 MB dan hard disk minimal 500 MB
- b. Alat masukan berupa *keyboard* dan *mouse*
- c. Alat keluaran berupa monitor dan printer.

### **3.4.3. Struktur Data**

Pada proses pembuatan aplikasi diperlukan variabel-variabel untuk menyimpan data yang diperlukan pada saat program dijalankan. Variabel - variabel input yang digunakan adalah:

```
Type noktah
    nomor As Integer
    nama As String
    posX As Integer
    posY As Integer
    derajat As Integer
End Type
```

Tipe data Noktah digunakan untuk menyimpan data terminal atau steiner poin.

```
Type jalur
    nomor As Integer
    awal As String
    akhir As String
    jarak As Integer
End Type
```

Tipe data Jalur digunakan untuk menyimpan data jalur yang dibentuk pada waktu perhitungan

```
Type edit
    kolom As Integer
    baris As Integer
    ubah As String
End Type
```

Tipe data Edit digunakan untuk menyimpan perubahan data yang terjadi pada MSFlexGrid pada Form Edit Terminal.

```
Type simpanPrim
    dataPrim(1 To 100) As noktah
    dataJumlah As Integer
    dataJalur(1 To 200) As jalur
    dataJarak As Integer
    dataJumlahJalur As Integer
End Type
```

Tipe data simpanPrim digunakan untuk menyimpan semua data yang berubah saat terjadinya iterasi pada Algoritma MRST.

```
Public terminal(1 To 50) As noktah
Public jml As Integer
Public skala As Integer
Public satuan As String
```

Variabel terminal bertipe data noktah, digunakan untuk menyimpan data terminal. Data ini berupa array dengan kapasitas 50 data (dari 1 hingga 50). Variabel jml digunakan untuk menyimpan jumlah terminal yang telah dimasukkan, variabel skala digunakan untuk menyimpan data perbandingan skala, sedangkan data

satuan digunakan untuk menyimpan jenis satuan yang digunakan, misalnya: meter, kilometer, dan sebagainya.

```
Public simpan(1 To 50) As simpanPrim
Public Seluruh As Integer
Public tiga As Integer
```

Data simpan bertipe data simpanPrim, merupakan array yang dapat menampung 50 data (dari 1 hingga 50), yang digunakan untuk menyimpan data iterasi yang terjadi pada algoritma MRST. Data tiga bertipe data integer, digunakan untuk mengetahui jumlah iterasi yang terjadi pada MRST, sedangkan data Seluruh (juga bertipe data integer) digunakan pada form proses dan hasil, untuk mengetahui iterasi yang dipilih pengguna untuk ditampilkan hasilnya atau diproses (disimpan atau dicetak).

```
Public JalurPB(1 To 100) As jalur
Public tigaPB As Integer
```

Kedua data tersebut digunakan dalam perhitungan Algoritma Prim tanpa Rectilinear Metric.

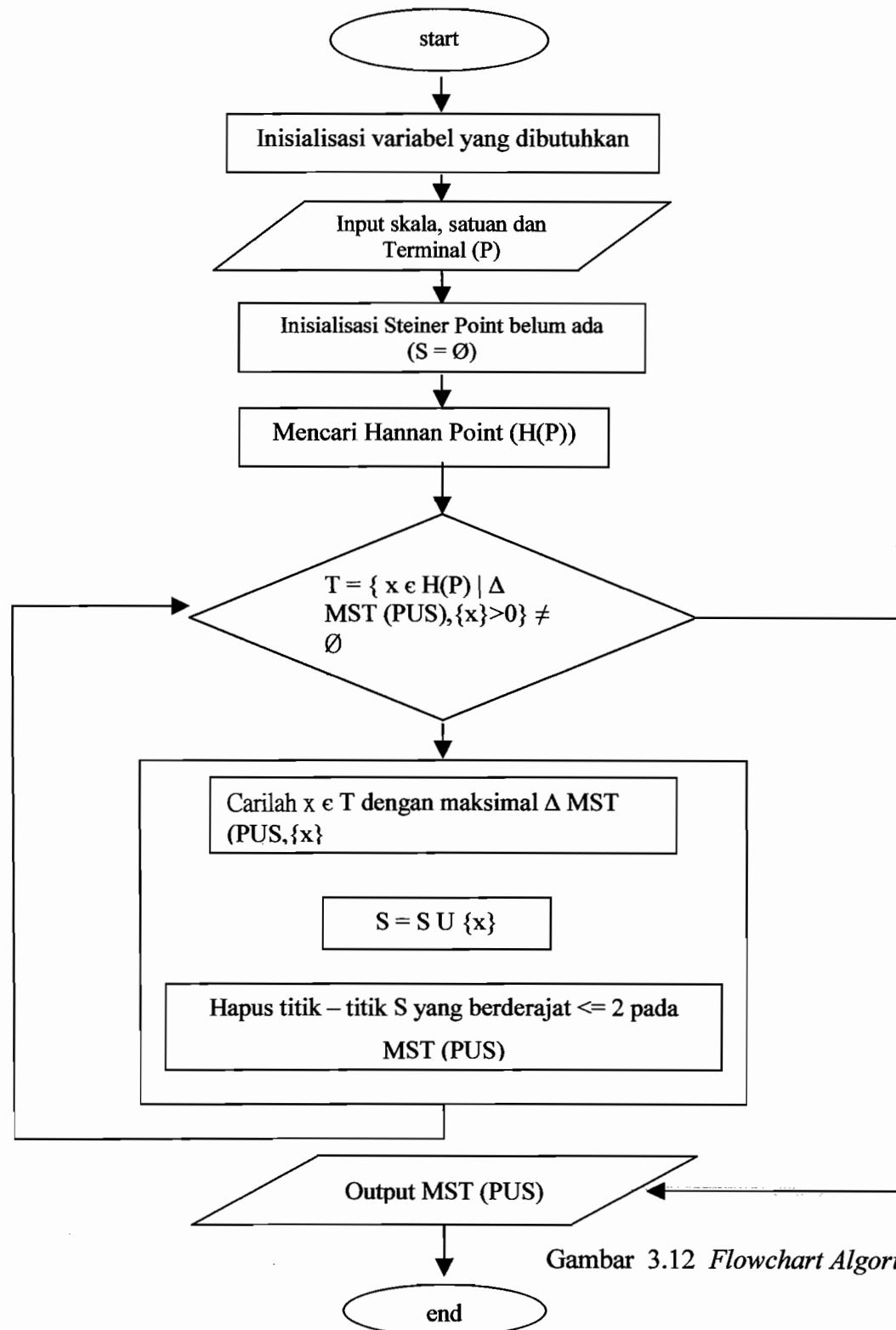
```
Public PB As Integer
```

Data ini digunakan sebagai penanda apakah suatu perhitungan menggunakan Rectilinear Metric atau tidak.

#### **3.4.4 Algoritma dan Flowchart**

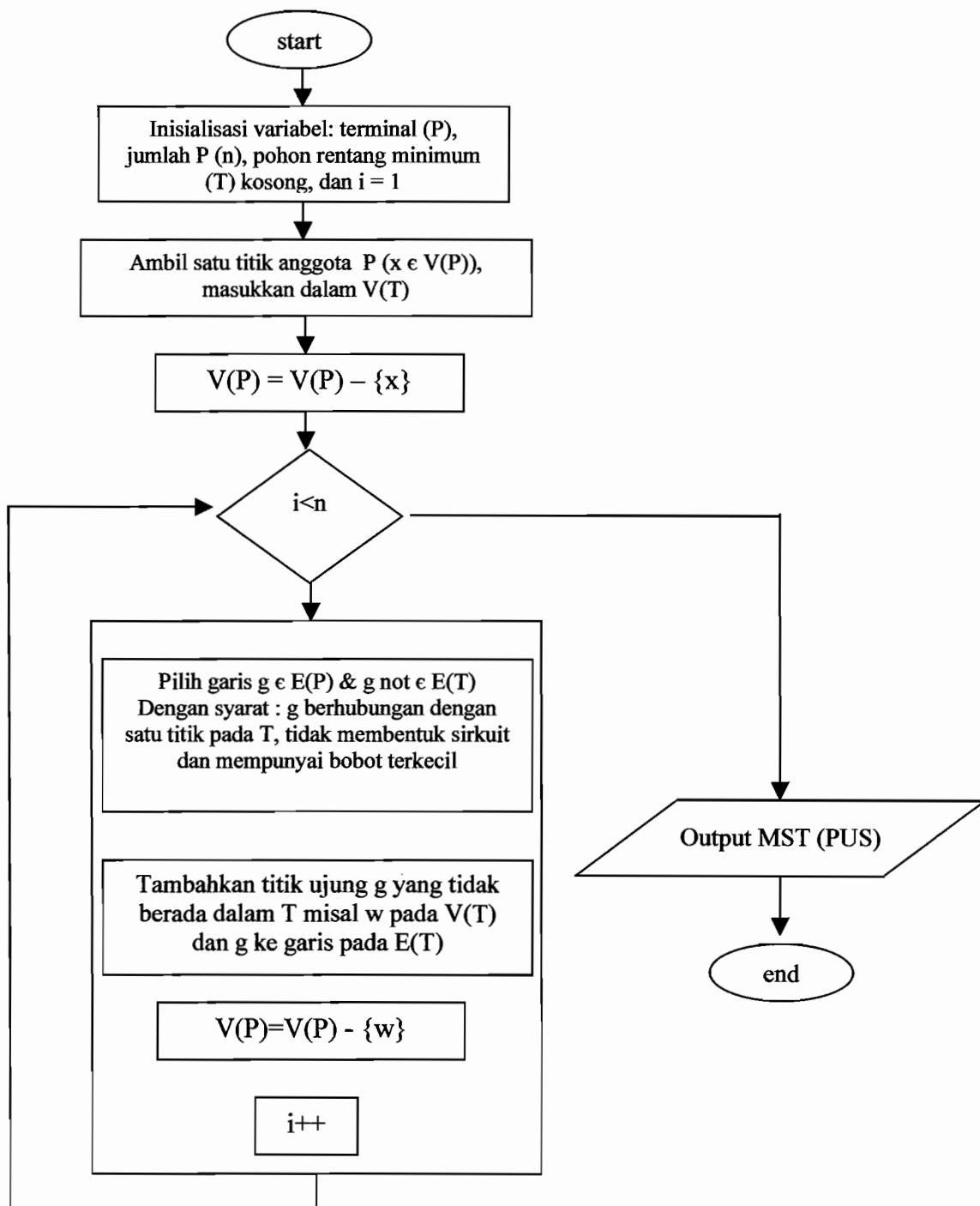
Algoritma yang akan digunakan dalam aplikasi pembentukan jaringan pipa air ini adalah Iterated 1-Steiner (I1S) dari Kahng dan Robins. Dalam algoritma ini terdapat perhitungan untuk mencari Minimum Spanning Tree (MST). Algoritma

yang digunakan untuk mengitung MST adalah algoritma Prim. Secara garis besar, flowchart algoritma IIS dapat dilihat pada Gambar 3.12.



Gambar 3.12 Flowchart Algoritma IIS

Pada gambar 3.13 akan ditampilkan flowchart Minimum Spanning Tree dengan algoritma Prim.



Gambar 3.13 Flowchart Algoritma Prim

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Implementasi Program**

##### **4.1.1. Algoritma Prim**

Untuk menghitung jarak terpendek antar titik (terminal dan *steiner point*) digunakan algoritma Prim. Implementasi Algoritma Prim dapat dilihat pada Function FungsiPrim.

```
Private Function FungsiPrim(point() As noktah, jumlah As Integer,  
ulang As Integer) As Integer
```

Fungsi FungsiPrim mempunyai tiga buah parameter, yaitu:

- a. variabel point, dengan tipe data noktah. Digunakan untuk menampung data titik (terminal dan *steiner point*) yang akan dihitung menggunakan algoritma Prim
- b. variabel jumlah, dengan tipe data integer. Digunakan untuk menampung jumlah titik pada perhitungan algoritma prim
- c. variabel ulang, dengan tipe data integer. Digunakan untuk menampung keterangan tentang iterasi ke-n pada perhitungan.

Nilai balik FungsiPrim bertipe data integer, yang merupakan jarak total yang dihasilkan dari perhitungan algoritma ini.

Beberapa variabel yang dibutuhkan dalam perhitungan Algoritma Prim, antara lain variabel Jarak yang digunakan untuk menghitung jarak antar titik (point), dan variabel Matriks yang berisi jarak antar titik yang akan mengalami

perubahan akibat perhitungan. Variabel angka digunakan untuk menyimpan jalur yang sudah dilalui.

```
Dim i, j As Integer ' untuk perulangan for
Dim d, e, f As Integer ' untuk cek sirkuit
Dim jarak(1 To 2500, 1 To 2500) As Integer
Dim Matriks(1 To 2500, 1 To 2500) As Integer
Dim angka(100) As Integer
```

Setiap perhitungan dengan Algoritma Prim juga akan menghitung derajat masing-masing poin. Derajat akan digunakan untuk pengecekan *steiner pointt*.

```
'set derajat = 0 & kosongkan array
For i = 1 To jumlah
    point(i).derajat = 0
Next i
```

Sebelum memulai perhitungan, semua data yang ada pada array harus dikosongkan terlebih dahulu

```
Erase Matriks
Erase jarak
Erase angka
Erase simpan(ulang).dataJalur
```

Untuk mencari jarak terpendek kumpulan titik yang diberikan, langkah awal yang dilakukan adalah mencari jarak antar titik yang ada. Pencarian jarak antar titik dilakukan dengan Rectilinear Metric, yaitu

$$d(a,b) = |X_a - X_b| + |Y_a - Y_b|$$

Pada perhitungan ini, jarak antar titik yang sama dianggap tidak ada. Jarak yang dihasilkan disimpan pada variabel jarak dan juga dicopykan pada variabel Matriks. Variabel Matriks digunakan pada proses perhitungan.

```
For i = 1 To jumlah
    For j = 1 To jumlah
        If i <> j Then
            jarak(i, j) = Abs(point(i).posX - point(j).posX) +
            Abs(point(i).posY - point(j).posY)

            'Mengcopy jarak antar titik ke variabel uji
            Matriks(i, j) = jarak(i, j)
        End If
    Next j
Next i
```

Untuk memulai perhitungan, dibutuhkan beberapa variabel bantu dan pengesetan awal data

```
'variabel bantu
Dim k, count, x, y, cari, JarakTotal As Integer
Dim T(1 To 100) As Integer

JarakTotal = 0  'set jarak awal = 0

'untuk data jalur
g = 0

'set angka = -1
For e = 1 To jumlah
    angka(e) = -1
Next e

d = 1
angka(d) = 1

'Sembarang titik yang digunakan adalah titik nomor 1
k = 1

'jumlah titik
count = 0

'pohon rentang minimum
T(1) = 1
```

Perhitungan untuk mencari jarak terpendek antar titik yang berhubungan.

```
Do Until count = jumlah - 1  'perulangan dilakukan hingga jumlah titik - 1

'variabel cari diset nilai tertinggi tipe data integer
cari = 32767

For i = 1 To k
    For j = 1 To jumlah
        If cari > Matriks(T(i), j) And Matriks(T(i), j) > 0
Then
            cari = Matriks(T(i), j)
            x = T(i)
            y = j
        End If
    Next j
Next i
```

```

For i = 1 To jumlah
    For j = 1 To jumlah
        If i = x And j = y Then
            JarakTotal = JarakTotal + jarak(i, j)

        'input nilai jalur
        g = g + 1
        simpan(ulang).dataJalur(g).nomor = g
        simpan(ulang).dataJalur(g).awal = x
        simpan(ulang).dataJalur(g).akhir = y
        simpan(ulang).dataJalur(g).jarak = jarak(i, j)

    'Derajat titik
    point(i).derajat = point(i).derajat + 1
    point(j).derajat = point(j).derajat + 1

    'cek sirkuit
    e = 0
    For f = 1 To d
        If angka(f) = i Then
            e = 1
        End If
    Next f

    If e = 0 Then
        d = d + 1
        angka(d) = i
    End If

    e = 0

    For f = 1 To d
        If angka(f) = j Then
            e = 1
        End If
    Next f

    If e = 0 Then
        d = d + 1
        angka(d) = j
    End If

    k = k + 1
    T(k) = j

    'cek sirkuit yang terjadi
    Dim a, b As Integer
    For a = 1 To jumlah
        For b = 1 To jumlah
            If angka(a) <> -1 And angka(b) <> -1 Then
                Matriks(angka(a), angka(b)) = -1
            End If
        Next b
    Next a

    count = count + 1
End If
Next j
Next i

Loop

```

Nilai balik fungsiPrim adalah jarak total yang didapat. Sedangkan jumlah jalur pada iterasi tersebut adalah nilai akhir dari variabel g

```
FungsiPrim = JarakTotal
simpan(ulang).dataJumlahJalur = g
```

#### 4.1.2. Hanan Poin

Menurut Hannan, *Rectilinear Steiner Tree* yang optimum didapat dengan menggunakan *steiner point* yang lokasinya merupakan perpotongan antara garis vertikal dan horizontal yang melewati terminal. Garis – garis yang merupakan perpotongan antara garis vertikal dan horizontal yang melewati terminal disebut Hanan Grid sedangkan titik – titik yang dihasilkan disebut Hanan Point.

Prosedur Sub Hanan ini digunakan untuk mencari Hanan Point dari kumpulan terminal yang diberikan.

```
Dim indeks As Integer
Dim i, j As Integer
indeks = 0

For i = 1 To jml
    For j = 1 To jml
        If i <> j Then
            indeks = indeks + 1
            HananPoint(indeks).nomor = indeks
            HananPoint(indeks).posX = terminal(i).posX
            HananPoint(indeks).posY = terminal(j).posY
        End If
    Next j
Next i
```

### 4.1.3. Algoritma Iterated-1 Steiner

Salah satu algoritma yang digunakan dalam pemecahan Minimum Rectilinear Steiner Spanning Tree adalah Iterated 1-Steiner (IIS) dari Kahng dan Robins. Untuk melakukan perhitungan diperlukan beberapa variable, yaitu:

```
Dim a, b As Integer
Dim T(1 To 200) As noktah
Dim jmlGab As Integer
Dim p As Integer

jmlGab = jml
```

Pada iterasi pertama ( $dat = 1$ ), variabel simpan diinputkan data-data terminal. Jumlah data sama dengan jumlah terminal yang ada. Kemudian dicari jarak terpendek antar terminal (sebelum adanya *steiner point*).

```
dat = 1
simpan(dat).dataJumlah = jml
For i = 1 To jml
    simpan(dat).dataPrim(i).nomor = i
    simpan(dat).dataPrim(i).nama = terminal(i).nama
    simpan(dat).dataPrim(i).posX = terminal(i).posX
    simpan(dat).dataPrim(i).posY = terminal(i).posY
Next i

simpan(dat).dataJarak = FungsiPrim(simpan(dat).dataPrim,
simpan(dat).dataJumlah, 1)
a = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah, 1)
```

Jumlah Hanan Poin yang mungkin dibuat dalam sebuah himpunan dengan  $n$  terminal adalah sebanyak  $n^2$ , dan  $n$  buah titik diantaranya merupakan terminal itu sendiri sehingga jumlah Hanan Poin menjadi  $n^2 - n$ . Variabel nilai merupakan banyaknya kemungkinan Hanan Poin.

Variabel data digunakan untuk mengecek data apakah ada penambahan *steiner point*, sedangkan variabel s digunakan untuk membantu dalam proses iterasi.

```
Dim data, nilai, s As Integer
data = 0
nilai = jml * jml - jml
```

Jika jarak sebelumnya lebih besar daripada jarak jika ditambahkan *steiner point*, maka data *steiner point* sementara akan dipindahkan pada variabel T. Jika seluruh hanan poin sudah dicek, maka variabel T dengan indeks terbesar merupakan *steiner point* yang diambil. Data akan disimpan pada variabel simpan; pertama-tama data pada variabel simpan (dari indeks sebelumnya) akan dicopykan ke variabel simpan indeks baru, selanjutkan data *steiner point* yang baru akan disimpan pada variabel simpan tersebut.

Iterasi akan terus dilakukan hingga tidak lagi ditemukan *steiner point* baru (jarak tidak lagi bertambah pendek dengan ditambahkannya *steiner point* baru).

```
Do
    For s = 1 To nilai
        simpan(dat).dataPrim(1 + simpan(dat).dataJumlah).posX =
HanhanPoint(s).posX
        simpan(dat).dataPrim(1 + simpan(dat).dataJumlah).posY =
HanhanPoint(s).posY
        b = FungsiPrim(simpan(dat).dataPrim,
simpan(dat).dataJumlah + 1, dat)

        If a > b Then
            data = data + 1
            T(data).posX = HanhanPoint(s).posX
            T(data).posY = HanhanPoint(s).posY
            simpan(dat + 1).dataJarak = b
            a = b
        End If

        b = 0
    Next s
```



```

If data <> 0 Then
    dat = dat + 1
    For i = 1 To simpan(dat - 1).dataJumlah
        simpan(dat).dataPrim(i).nomor = simpan(dat -
1).dataPrim(i).nomor
        simpan(dat).dataPrim(i).nama = simpan(dat -
1).dataPrim(i).nama
        simpan(dat).dataPrim(i).posX = simpan(dat -
1).dataPrim(i).posX
        simpan(dat).dataPrim(i).posY = simpan(dat -
1).dataPrim(i).posY
    Next i

    simpan(dat).dataPrim(1 + jmlGab).nomor = 1 + jmlGab
    simpan(dat).dataPrim(1 + jmlGab).nama = Chr(167) & 1 +
jmlGab - jml
    simpan(dat).dataPrim(1 + jmlGab).posX = T(data).posX
    simpan(dat).dataPrim(1 + jmlGab).posY = T(data).posY
    simpan(dat).dataJumlah = jmlGab + 1

    'mengembalikan data sebelumnya
    b = FungsiPrim(simpan(dat - 1).dataPrim, simpan(dat -
1).dataJumlah, dat - 1)

    jmlGab = jmlGab + 1
    data = 0
Else
    data = -1
End If
Loop Until data = -1

```

Meskipun steiner tree dapat memuat paling banyak  $n - 2$  *steiner point*, IIS dapat menambah lebih dari  $n - 2$  *steiner point*. Oleh karena itu akan dilakukan pengecekan, *steiner point* yang mempunyai derajat  $\leq 2$  akan dihapus.

```

b = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah,
dat)

Dim o As Integer
For p = simpan(1).dataJumlah + 1 To simpan(dat).dataJumlah
    If simpan(dat).dataPrim(p).derajat <= 2 Then
        For o = p To simpan(dat).dataJumlah - 1
            simpan(dat).dataPrim(o).nomor = o + 1
            simpan(dat).dataPrim(o).nama = Chr(167) & o -
simpan(1).dataJumlah
            simpan(dat).dataPrim(o).posX =
simpan(dat).dataPrim(o + 1).posX
            simpan(dat).dataPrim(o).posY =
simpan(dat).dataPrim(o + 1).posY
            simpan(dat).dataPrim(o).derajat =
simpan(dat).dataPrim(o + 1).derajat
        Next o
        simpan(dat).dataJumlah = simpan(dat).dataJumlah - 1
    End If
Next p

```

Setelah itu, data akan diset untuk menampilkan hasil akhir dari perhitungan MRST

```

'Jumlah data diset semua data
Seluruh = dat

b = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah,
dat)
simpan(dat).dataJarak = b

'jumlah jarak MRST
tiga = Seluruh

```

#### 4.1.4. Proses Penyimpanan Data Terminal

Untuk memudahkan pengguna, data terminal dapat disimpan dalam file berekstensi txt. File ini akan menyimpan data-data berupa:

1. satuan
2. skala
3. data terminal, yaitu nama, pos X dan Pos Y terminal

Pada prosedur sub SimpanTerminal, semua data akan diubah terlebih dahulu ke variabel bertipe data string. Antara data satuan, skala, dan data terminal dipisahkan dengan karakter enter (chr(10) dan chr(13)). Sedangkan karakter pemisah yang digunakan untuk pembatas antara nama, pos X dan Pos Y pada data terminal adalah karakter ¥ (char(165)). Karakter ini digunakan karena jarang sekali pengguna memberikan karakter tersebut untuk penamaan datanya.

```

Sub SimpanTerminal()
Dim data(1 To 100) As String
Dim data1 As String
Dim i As Integer

On Error GoTo errHandler
frmUtama.tutup_menu

'Mengubah semua data terminal ke string
For i = 1 To jml
    data(i) = terminal(i).nama + Chr(165) +
    CStr(terminal(i).posX) + Chr(165) + CStr(terminal(i).posY)
Next i

data1 = satuan + Chr(10) + Chr(13) + CStr(skala)

For i = 1 To jml
    data1 = data1 + Chr(10) + Chr(13) + data(i)
Next i

With frmUtama.CommonDialog1
    .Filter = "(*.txt)|*.txt|"
    .DialogTitle = "simpan file"
    .ShowSave
End With

Open frmUtama.CommonDialog1.FileName For Output As #2
Write #2, data1
Close #2

```

#### **4.1.5. Proses Open Data Terminal**

Untuk membuka data Terminal yang telah disimpan oleh pengguna, data-data yang telah diubah ke dalam variabel bertipe data string harus dipilah – pilah dan kemudian diubah kembali ke dalam tipe data yang bersesuaian. Salah satu

fungsi yang dapat digunakan untuk memilah data adalah fungsi split. Fungsi ini mengembalikan data berupa array berdasarkan karakter pembatas.

```

Sub BukaFile()
Dim data1 As String 'menampung data dari file yang dibuka
Dim cari, cari2 As String
Dim m, n, i As Integer
Dim data(1 To 100) As String

'membuka file dan menyimpan pada data1
Open frmUtama.CommonDialog1.FileTitle For Input As #1
    While Not EOF(1)
        Input #1, data1
    Wend
Close #1

cari = Chr(10) + Chr(13) 'karakter pembatas
cari2 = Chr(165)

Dim coba

coba = Split(data1, cari)
satuan = coba(0)
skala = coba(1)

For m = 2 To UBound(coba)
    coba2 = Split(coba(m), cari2)
    terminal(m - 1).nomor = m - 1
    terminal(m - 1).nama = coba2(0)
    terminal(m - 1).posX = CInt(coba2(1))
    terminal(m - 1).posY = CInt(coba2(2))
Next m

jml = UBound(coba) - 1

```

Pada Prosedur Sub BukaFile dapat dilihat bahwa karakter pembatas I, yaitu karakter pembatas yang memisahkan data satuan, skala, dan data terminal (`chr(10)+chr(13)`) disimpan pada variabel `cari`, sedangkan karakter pembatas II, yang memisahkan data-data nama, Pos X, dan Pos Y (`chr(165)`) disimpan pada variabel `cari2`.

```
coba = Split(data1, cari)
```

Variabel coba bertipe data variant. Dengan penggunaan fungsi Split, variabel coba menjadi bertipe data array. Nilai coba(0) adalah data-data yang ada sebelum ditemukannya karakter pembatas (variabel cari), coba(1) merupakan data-data yang ada antara karakter pembatas dengan karakter pembatas berikutnya, demikian seterusnya.

Misalkan, hasil penyimpanan data yang terjadi adalah sebagai berikut:

```
"km
1
w¥50¥20
b¥70¥20
n¥60¥40"
```

Maka saat fungsi `coba = Split(data1, cari)` dijalankan nilai `coba(0) = km`, `coba(1) = 1`, `coba(2)= w¥50¥20`, dan seterusnya.

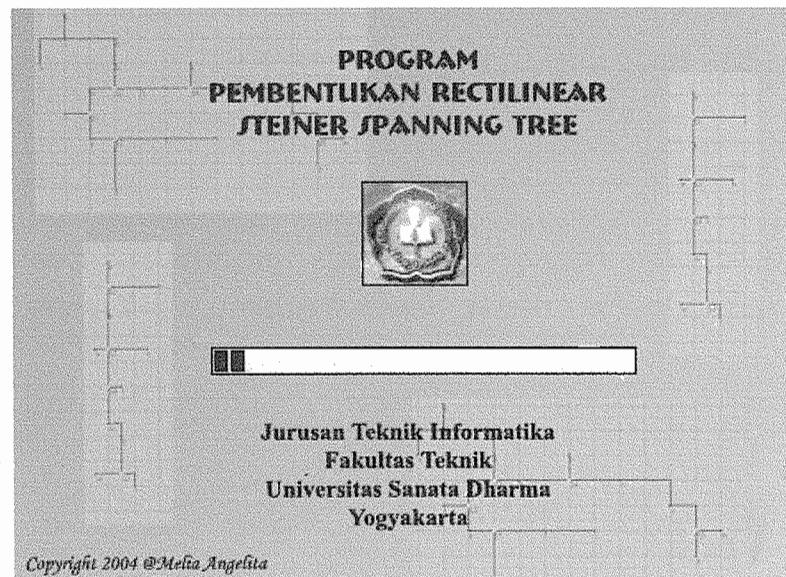
## 4.2 Implementasi Antarmuka

Antarmuka merupakan tampilan yang nantinya akan berinteraksi langsung dengan pengguna. Antarmuka Program Pembentukan Rectilinear Steiner Spanning Tree yang dibuat dapat diuraikan seperti di bawah ini.

### 4.2.1 Implementasi Tampilan Pembuka

Tampilan pembuka adalah tampilan yang pertama kali ditampilkan pada saat pertama kali program dijalankan. Tampilan pembuka berupa form sambutan

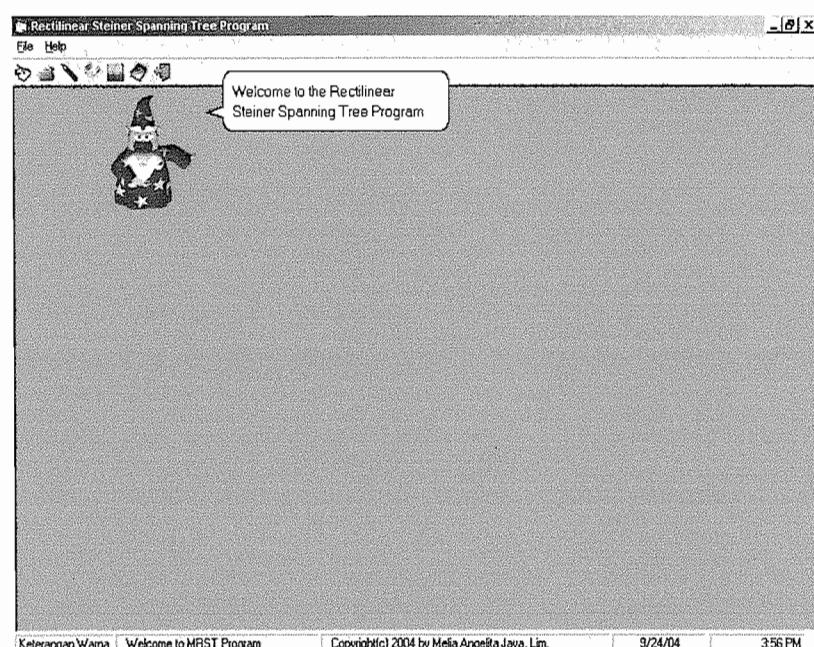
saat pengguna pertama kali masuk ke program. Form ini akan menutup sendiri sesuai dengan kontrol waktu.



Gambar 4.1 *Form tampilan awal*

#### 4.2.2. Implementasi Tampilan Utama

Saat pertama kali masuk pada tampilan utama, pengguna akan disambut dengan Merlin yang akan menjelaskan program secara singkat.



Gambar 4.2 *Tampilan Utama Awal*

Tampilan utama merupakan antarmuka induk yang berisi menu pilihan untuk menuju ke menu yang lain. Secara umum menu utama program pembentukan rectilinear Steiner Spanning Tree adalah sebagai berikut

1      *Menu File*

Berisi enam sub menu yaitu:

a.    *Menu New*

Menu ini berfungsi untuk memanggil lembar kerja baru. Pengguna akan diminta untuk menginputkan satuan dan skala perhitungan yang digunakan.

b.    *Menu Open*

Menu ini berfungsi digunakan untuk membuka data yang sudah ada/ disimpan sebelumnya.

c.    *Menu Edit*

Menu ini berfungsi untuk mengubah data terminal. Pengguna dapat menambah jumlah terminal, menghapus terminal, atau mengganti nama dan koordinat suatu terminal.

d.    *Menu Proses.*

Menu ini berfungsi untuk mengolah data terminal yang telah diinputkan.

e.    *Menu Save*

Menu ini berfungsi untuk menyimpan data terminal.

f.    *Menu Exit*

Menu ini berfungsi untuk keluar dari program

## 2 Menu Help

Berisi dua sub menu yaitu:

a. Menu Content

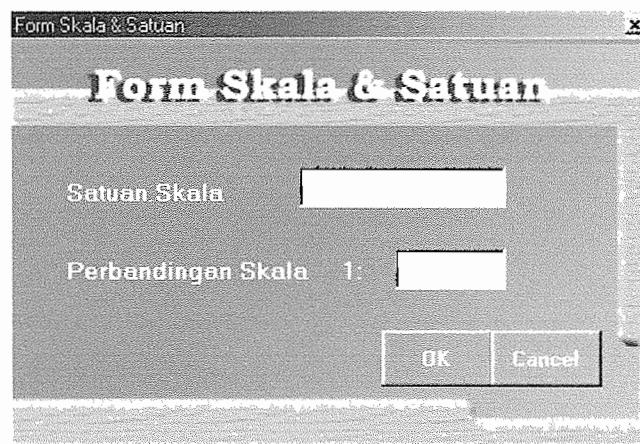
Menu ini berfungsi untuk menampilkan panduan untuk menjalankan program

b. Menu About

Merupakan menu untuk menampilkan informasi tentang penyusun program.

### 4.2.3. Implementasi Tampilan Form New

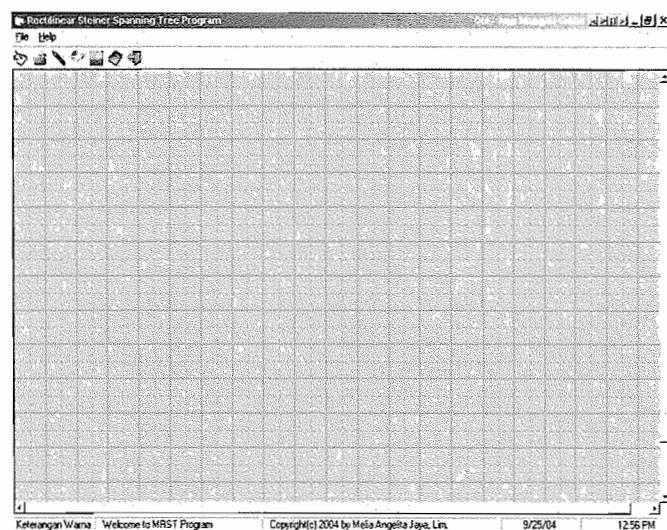
Form New merupakan antarmuka yang digunakan untuk memasukkan data baru. Pertama – tama pengguna harus memasukkan data satuan dan skala yang akan digunakan pada perhitungan.



Gambar 4.3 Form Skala & Satuan

Setelah selesai memasukkan data pada form skala dan satuan, akan ditampilkan lembar kerja berupa grid-grid. Kursor mouse akan menampilkan koordinat setiap

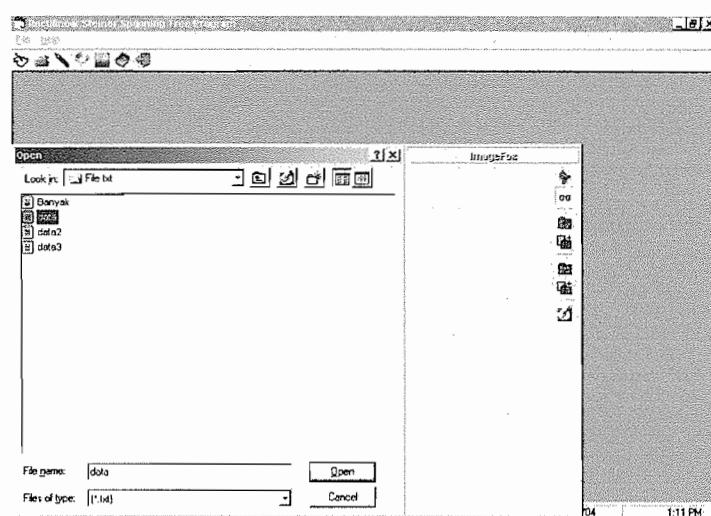
tempat yang ditunjuk sehingga dapat memudahkan pengguna untuk meletakkan suatu terminal atau menghapusnya.



Gambar 4.4 Lembar Kerja

#### 4.2.4. Implementasi Tampilan Open

Pengguna juga dapat mengambil kembali data yang telah disimpannya, dengan menjalankan menu open.

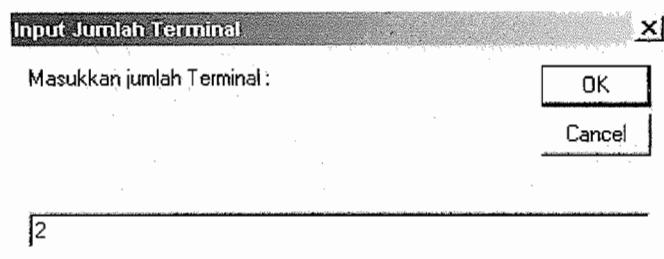


Gambar 4.5 Menu Open

#### 4.2.5. Implementasi Tampilan Form Edit

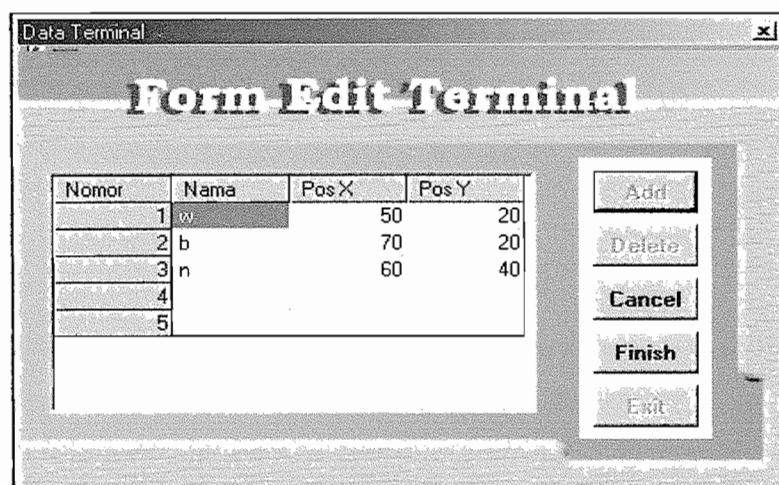
Form ini digunakan untuk melihat dan mengubah data terminal. Pengguna dapat menambah data terminal, menghapus terminal atau mengubah nama dan koordinat suatu terminal.

Jika pengguna ingin menambah terminal baru, klik tombol add pada form edit. User akan diminta memasukkan jumlah terminal yang ingin ditambahkan.



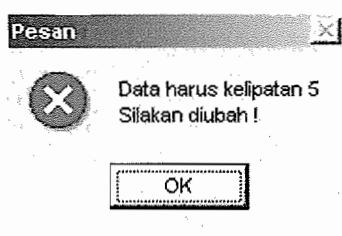
Gambar 4.6 Input Jumlah terminal

Selanjutnya pada form edit akan ditampilkan ruang kosong untuk menginputkan data terminal baru. tombol Add, Delete dan Exit akan dinonaktifkan, sedangkan tombol Cancel dan Finish akan diaktifkan. Tombol Cancel digunakan untuk membatalkan penambahan data, sedangkan tombol Finish dapat diklik jika pengguna selesai mennginputkan data terminal baru.



Gambar 4.7 Form Edit Terminal

Pengguna dapat mengklik ganda pada sel yang ingin diinputkan datanya. Pada sel Nama, pengguna dapat menginputkan data berupa karakter atau angka dengan panjang maksimal 5, sedangkan pada sel pos X dan Pos Y, pengguna hanya dapat memasukkan data integer berupa kelipatan 5. Nama dan koordinat yang diinputkan tidak boleh sama. Setelah selesai memasukkan data, pengguna harus mengklik sel lain.

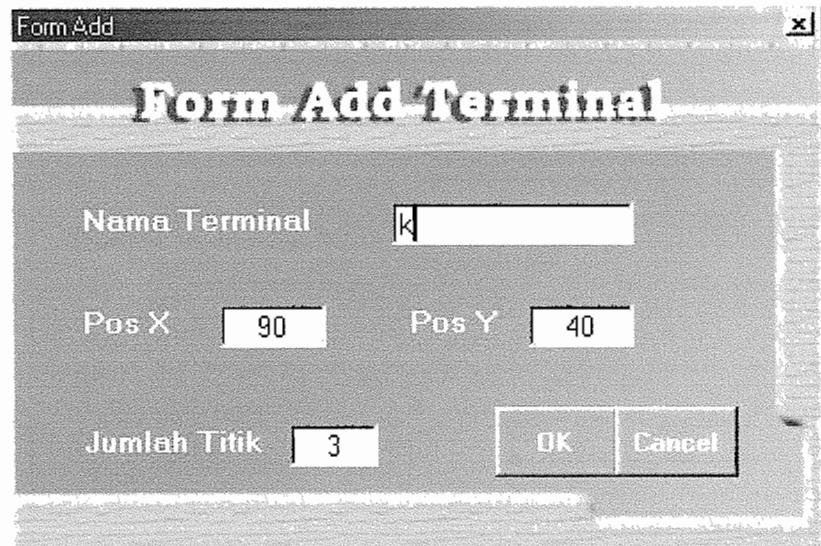


Gambar 4.8 *Validasi Kesalahan*

Untuk menghapus terminal, klik tombol Delete. Pengguna akan diminta untuk memasukkan nama terminal yang ingin dihapus. Sedangkan untuk mengubah nama terminal atau koordinatnya, pengguna cukup mengklik ganda pada sel yang ingin diganti, mengisi dengan data baru dan kemudian mengklik sel lain jika sudah selesai.

#### 4.2.6. Implementasi Tampilan Form Add

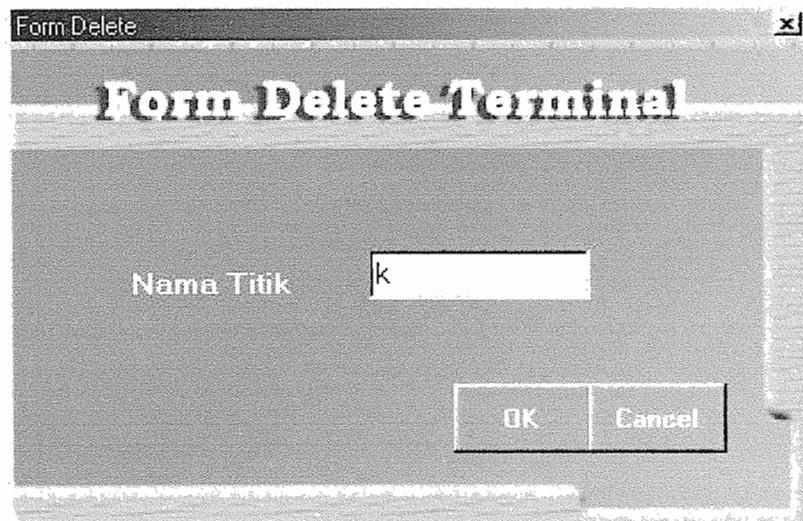
Pengguna juga dapat memasukkan titik pada lembar kerja yang telah disediakan dengan mengklik pada tempat yang koordinatnya bersesuaian. Secara otomatis akan ditampilkan form add. Pengguna hanya tinggal menginputkan nama terminalnya saja.



Gambar 4.9 Form Add

#### 4.2.7. Implementasi Tampilan Form Delete

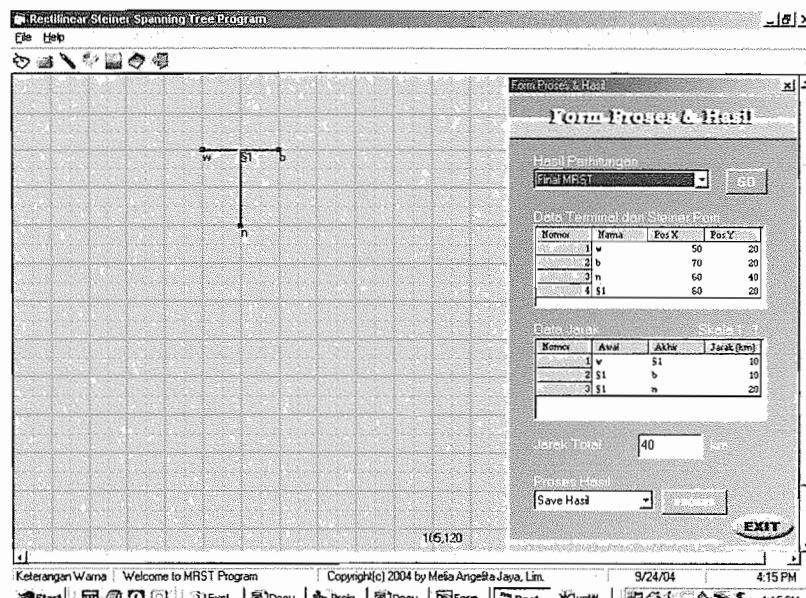
Untuk menghapus terminal, pengguna juga dapat mengklik terminal yang ada pada lembar kerja. Secara otomatis, form delete terminal akan ditampilkan.



Gambar 4.10 Form Delete

#### 4.2.8. Implementasi Tampilan Form Proses & Hasil

Jika pengguna ingin memproses data yang telah diinputkan, pengguna dapat mengklik menu Process.



Gambar 4.11 Menu Proses dan Hasilnya

Data hasil perhitungan Minimum Rectilinear Steiner Spanning tree akan ditampilkan pada form Proses dan Hasil, yaitu berupa:

1. Data Terminal dan *steiner point*, meliputi:
  - a. Nomor, merupakan nomor urut data
  - b. Nama, merupakan nama terminal atau nama *steiner point*. Nama terminal maksimal 5 karakter, dapat berupa angka atau huruf. Nama *steiner point* diawali dengan karakter § dan diikuti nomor urut *steiner point* tersebut.
  - c. Pos X, merupakan koordinat sumbu X dari terminal.
  - d. Pos Y, merupakan koordinat sumbu Y dari terminal.

2. Data Jarak, meliputi:

- a. Nomor, merupakan nomor urut data.
- b. Awal, merupakan nama terminal awal suatu jalur.
- c. Akhir, merupakan nama terminal akhir suatu jalur.
- d. Jarak, merupakan panjang jarak suatu jalur.

Sedangkan pada lembar kerja akan ditampilkan jalur yang dibentuk, dengan ketentuan:

- a. Titik warna merah, merupakan terminal yang diberikan
- b. Titik warna kuning, merupakan *steiner point*
- c. Garis hitam merupakan jalur yang dibentuk

Pada Form Proses dan Hasil juga terdapat combo box yang berisi hasil perhitungan. Pengguna dapat mengetahui hasil perhitungan dengan memilih salah satu hasil perhitungan dan kemudian mengklik tombol GO. Hasil perhitungan yang disediakan adalah:

1. Minimla Spanning Tree, merupakan perhitungan Minimum Spanning Tree antar terminal yang diberikan, dengan algoritma Prim, dan jarak yang dihitung tanpa Rectilinear Metric.
2. Iterasi ke: 1, merupakan hasil perhitungan Minimum Rectilinear Spanning Tree antar terminal yang diberikan, sebelum ditambahkan steiner point.
3. Iterasi ke: 2 – n-1, merupakan hasil penambahan *steiner point* berdasarkan perhitungan.

4. Final MRST, merupakan hasil akhir dari perhitungan Minimum Rectilinear Steiner Spanning Tree yang dihasilkan program.

Untuk setiap hasil perhitungan yang telah dilakukan, selain hasilnya dapat dilihat pada layar monitor, pengguna juga dapat memproses hasil tersebut dengan:

1. Save Hasil, yaitu menyimpan hasil perhitungan tersebut dalam bentuk file dokumen (\*.doc) sehingga dapat dibuka lagi pada saat program tidak dijalankan.
2. Print Hasil, yaitu langsung mencetak hasil perhitungan pada kertas (tanpa menyimpannya terlebih dahulu).
3. Print Layout, yaitu mencetak hasil layout lembar kerja pada kertas.

### **4.3. Pembahasan**

Program pembentukan Rectilinear Steiner spanning Tree dibuat menggunakan Algoritma Iterated 1-Steiner, sedangkan untuk pencarian jarak terpendek menggunakan Algoritma Prim. Jumlah terminal maksimal yang dapat diinputkan adalah sebanyak 50 titik.

#### **4.3.1. Studi Kasus Jaringan Pipa Air Kampus III USD**

Studi kasus yang digunakan adalah Jaringan Pipa Air Gedung Pusat Universitas Sanata Dharma, Kampus III Paingan (Lihat Denah Instalasi Air Bersih, Air Kotor & Air Kotoran (Toilet Lantai Dasar, 2, 3 & 4) pada Lampiran A). Terdapat 3 jenis jaringan pipa air pada tiap lantai, yaitu: jaringan pipa air untuk air bersih, air kotor dan air kotoran. Skala yang digunakan adalah 1 : 50

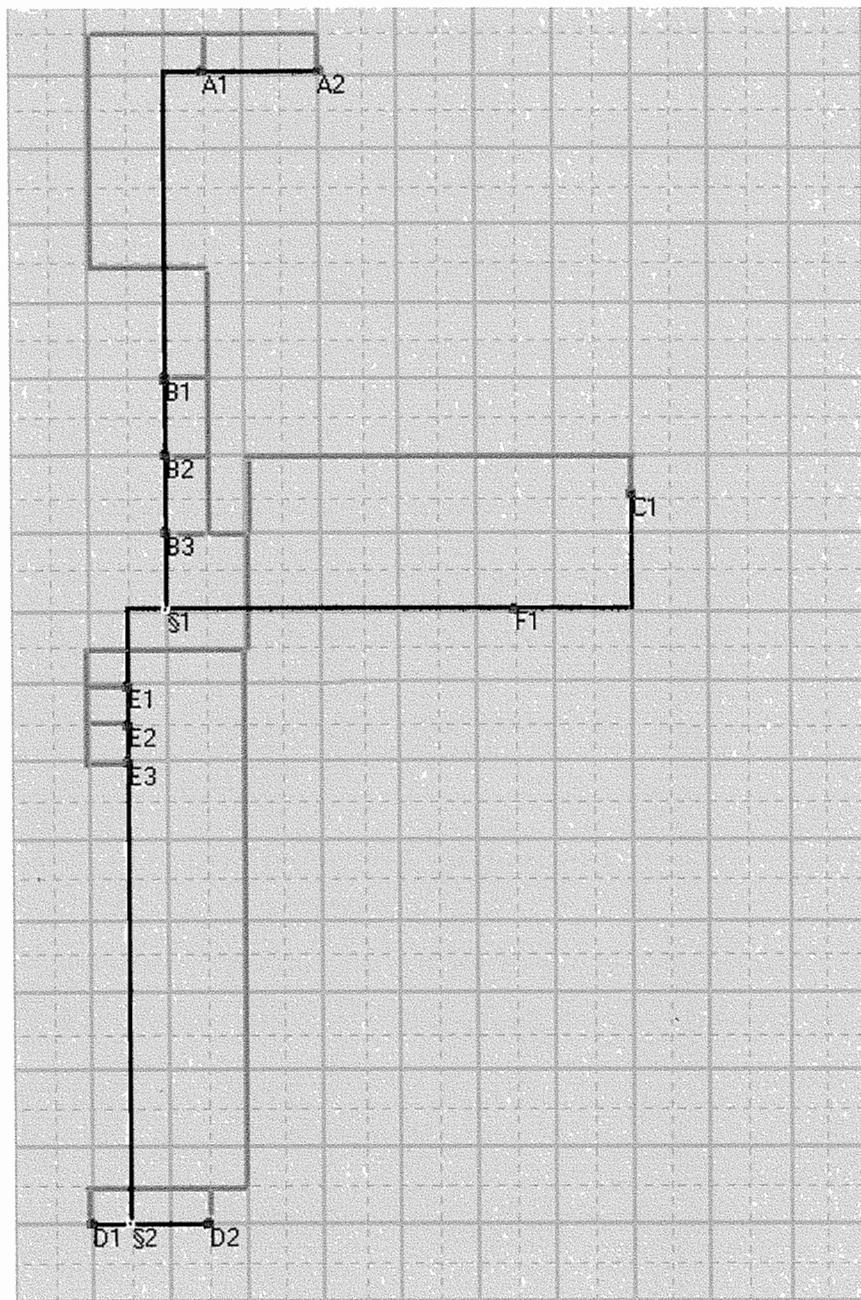
### a. Air Bersih

No	Nama	Pos X	Pos Y
1	A1	115	10
2	A2	130	10
3	B1	110	50
4	B2	110	60
5	B3	110	70
6	C1	170	65
7	D1	100	160
8	D2	115	160
9	E1	105	90
10	E2	105	95
11	E3	105	100
12	F1	155	80

Tabel 4.1 Data Air Bersih

Dari hasil perhitungan didapat:

1. Jarak total dihitung menggunakan Minimum Rectilinear Steiner Spanning Tree adalah 13250 mm, dengan 2 buah *steiner point*, yaitu:
  - a. §1 dengan koordinat (110,80)
  - b. §2 dengan koordinat(105,160)
2. Jarak total dihitung menggunakan Minimum Rectilinear Spanning Tree (tanpa *steiner point*) adalah 14000 mm.
3. Jarak total dihitung menggunakan Minimum Spanning Tree (tanpa menggunakan rectilinear metric) adalah 12400mm.



Gambar 4.12 Jaringan Pipa Air untuk Air Bersih  
(Warna Gelap: Program, Warna Terang: Denah pada Lampiran A)

Hasil perhitungan program dibandingkan dengan Denah Instalasi Air Bersih pada Lampiran A (dilambangkan dengan garis ). Dari Denah pada Lampiran A dapat dilihat beberapa data:

1. Jumlah titik percabangan ada sekitar 11 buah.
2. Panjang pipa air yang dibutuhkan adalah  $370 \times 50 = 18500$  mm

Berdasarkan data dan gambar dari keduanya dapat dilihat bahwa terdapat perbedaan antara Denah Instalasi Air Bersih pada lampiran A dengan hasil perhitungan Jaringan Pipa Air untuk Air Bersih berdasarkan perangkat lunak yang dibuat. Beberapa perbedaan yang dihasilkan:

- a. Titik percabangan yang dihasilkan (*steiner point*) pada perangkat lunak lebih sedikit dibandingkan pada denah lampiran A
- b. Pada hasil perhitungan menggunakan perangkat lunak, panjang pipa air yang dibutuhkan lebih sedikit dibandingkan dengan panjang pipa pada denah lampiran A.
- c. Gambar jaringan pipa air yang dihasilkan berbeda.

### **b. Air Kotor**

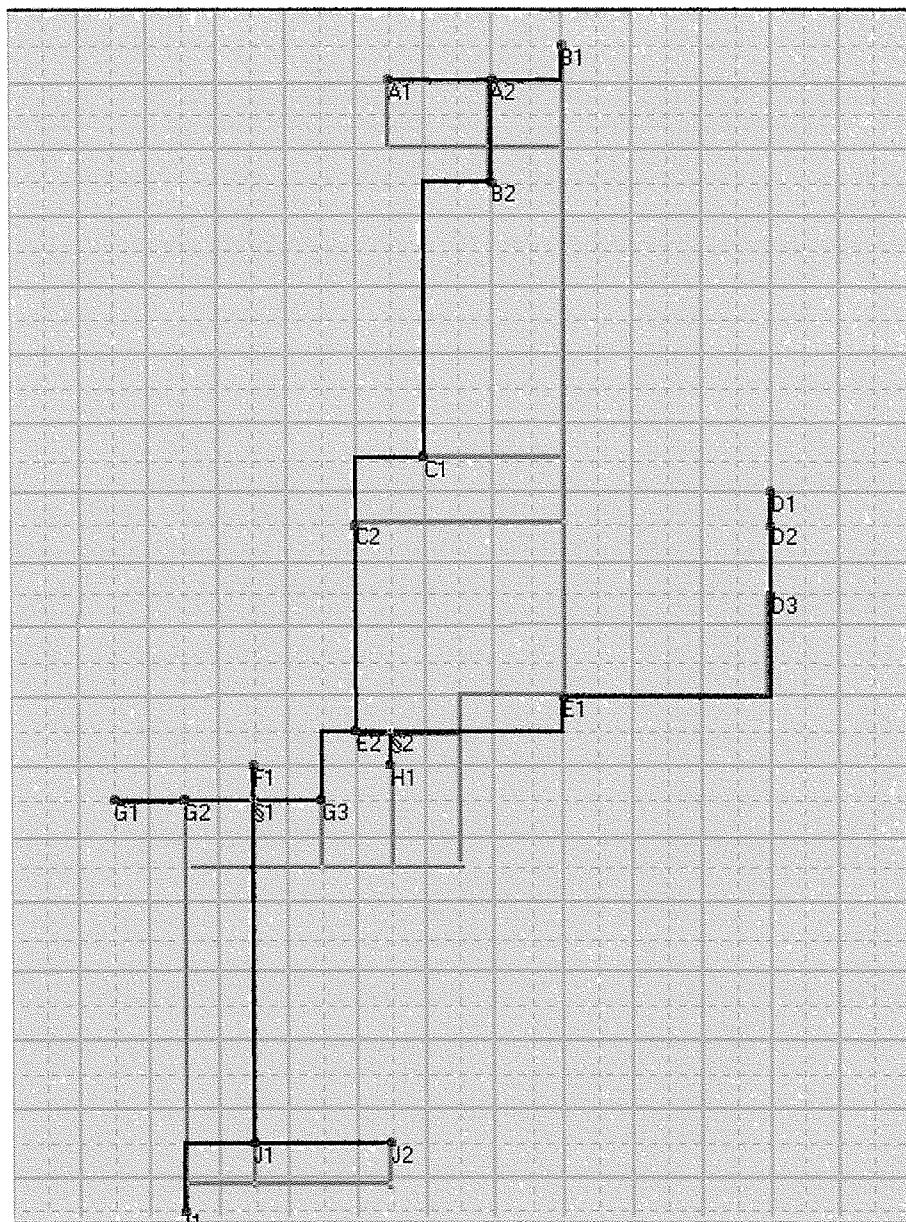
No	Nama	Pos X	Pos Y
1	A1	115	10
2	A2	130	10
3	B1	140	5
4	B2	130	25
5	C1	120	65
6	C2	110	75
7	D1	170	70
8	D2	170	75

9	D3	170	85
10	E1	140	100
11	E2	110	105
12	F1	95	110
13	G1	75	115
14	G2	85	115
15	G3	105	115
16	H1	115	110
17	I1	85	175
18	J1	95	165
19	J2	115	165

Tabel 4.2 *Data Air Kotor*

Dari hasil perhitungan didapat:

1. Jarak total dihitung menggunakan Minimum Rectilinear Steiner Spanning Tree adalah 19250 mm, dengan 2 buah *steiner point*, yaitu:
  - a. §1 dengan koordinat (95,115)
  - b. §2 dengan koordinat (115,105)
2. Jarak total dihitung menggunakan Minimum Rectilinear Spanning Tree (tanpa *steiner point*) adalah 20000 mm.
3. Jarak total dihitung menggunakan Minimum Spanning Tree (tanpa menggunakan rectilinear metric) adalah 16850 mm



Gambar 4.13 Jaringan Pipa Air untuk Air Kotor  
(Warna Gelap: Program, Warna Terang: Denah pada Lampiran A)

Sama seperti Denah Air bersih, hasil perhitungan program juga dibandingkan dengan Denah Instalasi Air Kotoran pada Lampiran A (dilambangkan dengan garis —••—•— ). Dari Denah pada Lampiran A dapat dilihat beberapa data:

1. Jumlah titik percabangan ada sekitar 11 buah.
2. Panjang pipa air yang dibutuhkan adalah  $490 \times 50 = 24500$  mm

Berdasarkan data dan gambar dari keduanya dapat dilihat bahwa terdapat perbedaan antara Denah Instalasi Air Kotor pada lampiran A dengan hasil perhitungan Jaringan Pipa Air untuk Air Kotor berdasarkan perangkat lunak yang dibuat. Beberapa perbedaan yang dihasilkan:

- a. Titik percabangan yang dihasilkan (*steiner point*) pada perangkat lunak lebih sedikit dibandingkan pada denah lampiran A
- b. Pada hasil perhitungan menggunakan perangkat lunak, panjang pipa air yang dibutuhkan lebih sedikit dibandingkan dengan panjang pipa pada denah lampiran A.
- c. Gambar jaringan pipa air yang dihasilkan berbeda.

### **c. Air Kotoran**

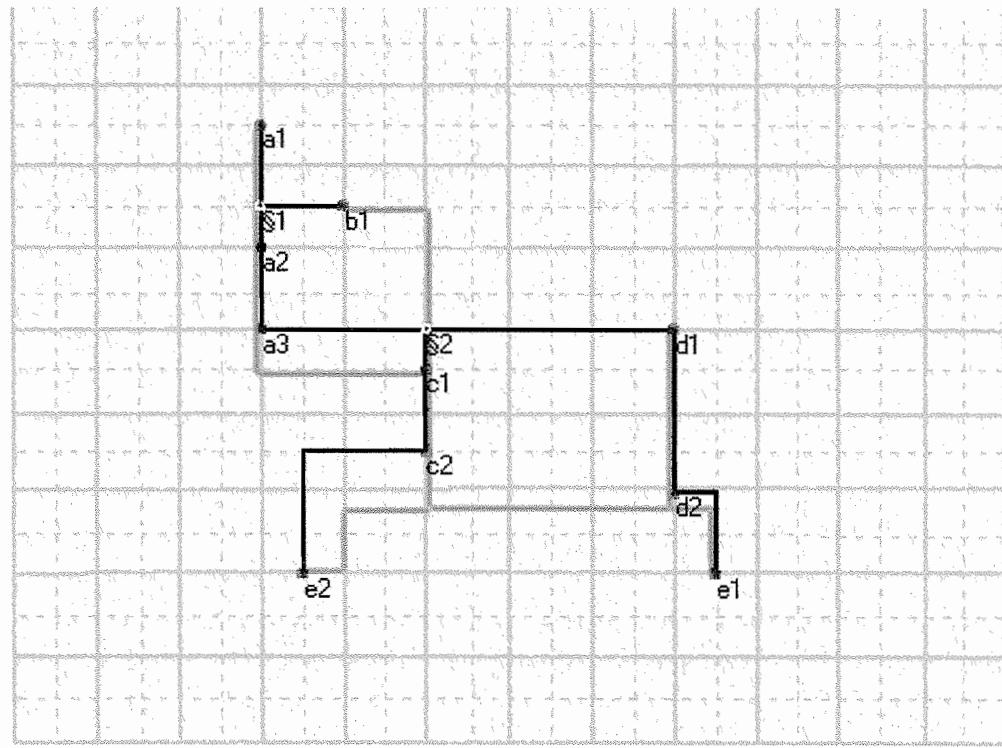
No	Nama	Pos X	Pos Y
1	A1	110	45
2	A2	110	60
3	A3	110	70
3	B1	120	55

4	C1	130	75
5	C2	130	85
6	D1	160	70
7	D2	160	90
8	E1	165	100
9	E2	115	100

Tabel 4.3. Data Air Kotoran

Dari hasil perhitungan didapat:

1. Jarak total dihitung menggunakan Minimum Rectilinear Steiner Spanning Tree adalah 8250 mm, dengan 2 buah *steiner point*, yaitu:
  - a. §1 dengan koordinat (110,55)
  - b. §2 dengan koordinat (130,70)
2. Jarak total dihitung menggunakan Minimum Rectilinear Spanning Tree (tanpa *steiner point*) adalah 8750 mm.
3. Jarak total dihitung menggunakan Minimum Spanning Tree (tanpa menggunakan rectilinear metric) adalah 7400 mm.



**Gambar 4.14 Jaringan Pipa Air untuk Air Kotoran  
(Warna Gelap: Program, Warna Terang: Denah pada Lampiran A)**

Hasil perhitungan program juga dibandingkan dengan Denah Instalasi Air Kotoran pada Lampiran A (dilambangkan dengan garis ). Dari Denah pada Lampiran A dapat dilihat beberapa data:

1. Jumlah titik percabangan ada 2 buah.
2. Panjang pipa air yang dibutuhkan adalah  $185 \times 50 = 9250$  mm

Berdasarkan data dan gambar dari keduanya dapat dilihat bahwa terdapat perbedaan antara Denah Instalasi Air Kotoran pada lampiran A dengan hasil perhitungan Jaringan Air Kotoran berdasarkan perangkat lunak yang dibuat.

Beberapa perbedaan yang dihasilkan:

- a. Pada hasil perhitungan menggunakan perangkat lunak, panjang pipa air yang dibutuhkan lebih sedikit dibandingkan dengan panjang pipa pada denah lampiran A.

- b. Gambar jaringan pipa air yang dihasilkan berbeda.

Pada data dan gambar Jaringan Pipa Air untuk Air Bersih, Air Kotor, dan Air Kotoran, terdapat beberapa perbedaan dibandingkan dengan hasil perhitungan jaringan pipa air oleh perangkat lunak. Perbedaan-perbedaan tersebut disebabkan oleh:

- a. Pada denah terdapat tiga buah jalur pipa air sehingga untuk daerah yang sama tidak mungkin dilalui dua kali atau lebih. Hal ini untuk mencegah terjadinya gangguan pada pipa air yang lain jika salah satu jalur pipa mengalami perbaikan.

Solusi untuk program perangkat lunak: jika suatu daerah dilalui oleh lebih dari sebuah jalur maka pengguna dapat menambahkan titik bantu pada daerah lain (sebagai terminal) sehingga jalur yang dibentuk akan mengikuti jalur yang diminta pengguna atau alternatif pengembangan program: adanya daerah pembatas untuk membatasi daerah yang tidak boleh dilalui oleh suatu garis.

- b. Denah pada Lampiran A dibuat berdasarkan perkiraan saja (berdasarkan kebiasaan, kemudahan dan segi tampilan), tidak dilakukan menurut perhitungan tertentu.



### 4.3.2. Uji Coba Perhitungan

Untuk mengetahui perbandingan jarak antara Minimal Spanning Tree, Minimal Rectilinear Spanning Tree dan Minimal Rectilinear Steiner Spanning Tree, program ini diujicobakan pada sepuluh data dengan hasil sebagai berikut:

No	Koordinat	Jml Ttk	Jml SP	A	B	C	C/A	C/B
1	A(30,10) B(80,10) C(50,20)	3	1	54	70	60	1,111	0,857
2	A(70,50) B(80,60) C(80,30)	3	1	36	50	40	1,111	0,800
3	A(50,30) B(60,20) C(80,40) D(70,50)	4	2	56	80	60	1,071	0,750
4	A(60,40) B(80,30) C(60,20)	3	1	42	50	40	0,952	0,800
5	A(70,20) B(70,40) C(110,30) D(30,30)	4	1	102	120	100	0,980	0,833
6	A(60,40) B(90,30) C (75,20)	3	1	43	60	50	1,163	0,833
7	A(70,30) B(90,30) C(80,60) D(80,20)	4	1	60	80	60	1,000	0,750
8	A(60,40) B(70,60) C(100,30) D(80,30)	3	1	64	80	70	1,167	0,875
9	A(50,40) B(60,30) C(70,40) D(60,50)	4	1	42	60	40	0,952	0,667
10	A(60,40) B(60,60) C(80,70) D(70,50)	3	1	50	70	60	1,200	0,857

11	A (55,55) B (35,40) C (40,60) D (15,55) E (50,50)	5	2	58	70	60	1,034	0,857
12	A (60,30) B (20,30) C (40,10) D (40,30)	4	0	60	60	60	1,000	1,000
13	A (30,20) B (30,30) C (50,40) D (35,45) E (45,55)	5	1	56	70	60	1,071	0,857
14	A (50,30) B (20,30) C (30,20) D (40,50) E (35,40)	5	2	61	85	65	1,066	0,765
15	A (30,30) B (40,10) C (50,20) D (70,20) E (65,30)	5	2	65	85	70	1,076	0,824
<b>RATA - RATA</b>							<b>1,064</b>	<b>0,822</b>

Tabel 4.4. Data Uji Coba

Keterangan:

Jml Ttk : Jumlah Terminal

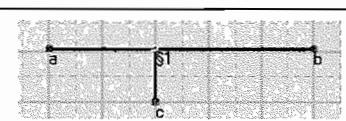
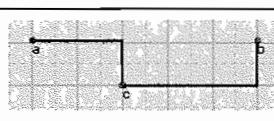
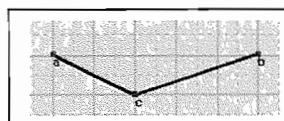
Jml SP : Jumlah *steiner point*

A : Minimum Spanning Tree

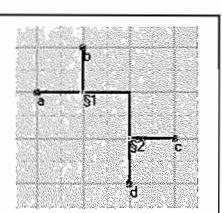
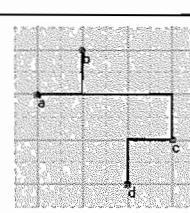
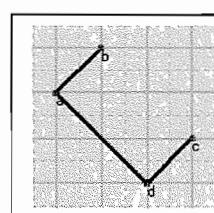
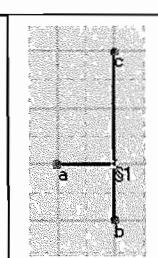
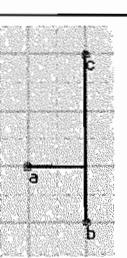
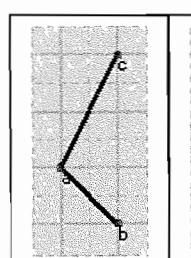
B : Minimum Rectilinear Spanning Tree

C : Minimum Rectilinear Steiner Spanning Tree

Pada Gambar 4.15 dapat dilihat bentuk gambar untuk masing – masing data uji coba (Kiri: Minimal Spanning Tree, Tengah: Minimal Rectilinear Spanning Tree, dan Kanan: Minimal Rectilinear Steiner Tree).

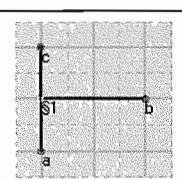
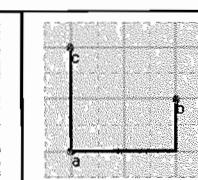
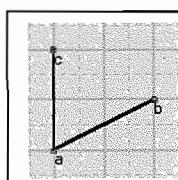


*Data Pertama*

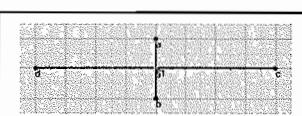
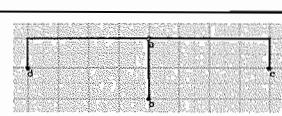
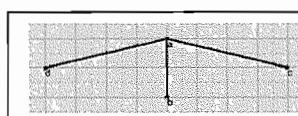


*Data Ketiga*

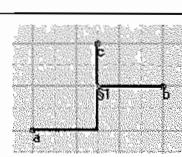
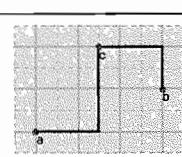
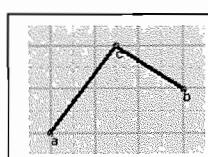
*Data Kedua*



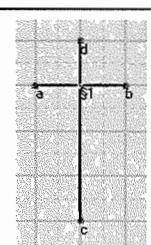
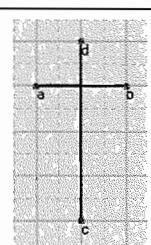
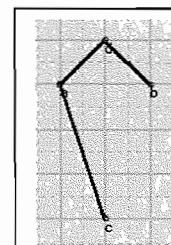
*Data Keempat*



*Data Kelima*

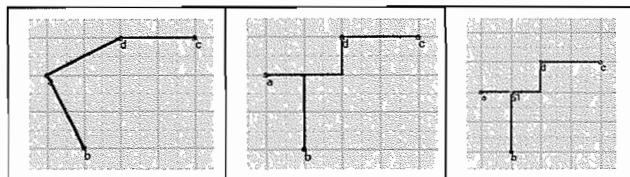


*Data Keenam*

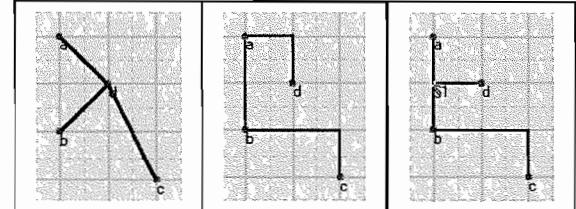


*Data Ketujuh*

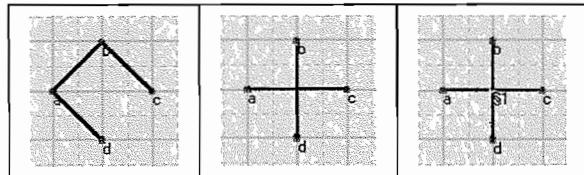
Gambar 4.15 Gambar Data Uji Coba



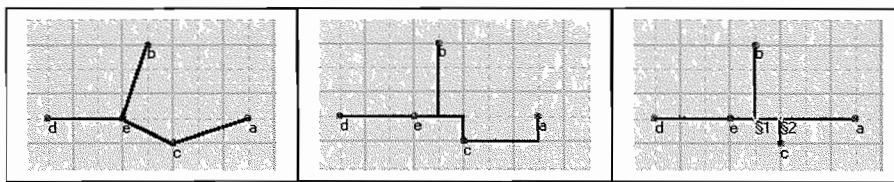
Data Kedelapan



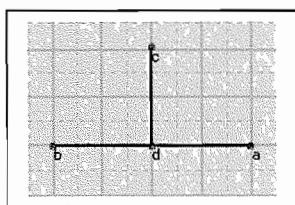
Data Kesepuluh



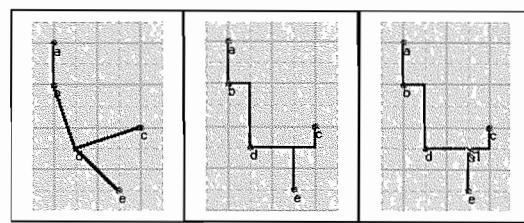
Data Kesembilan



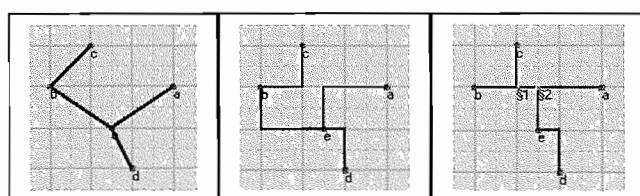
Data Kesebelas



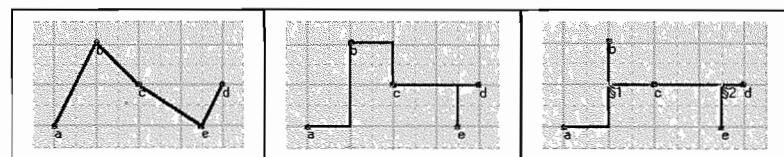
Data Keduabelas



Data Ketigabelas



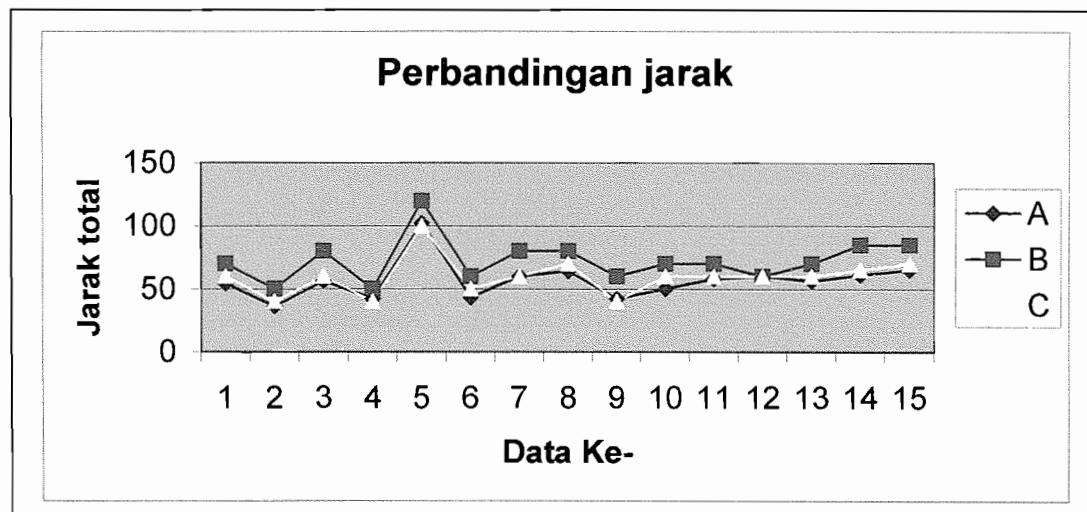
Data Keempatabelas



Data Kelimabelas

Gambar 4.15 Gambar Data Uji Coba

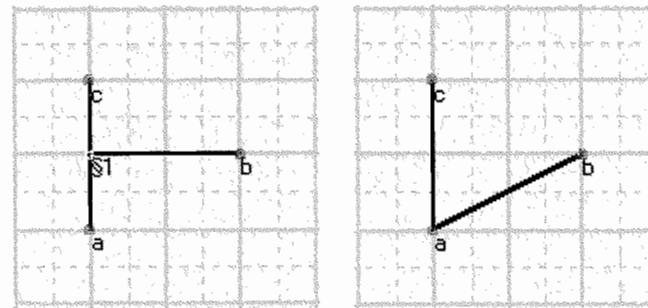
Untuk masing-masing data pada tabel 4.4 dapat dibuat grafik sebagai berikut:



Gambar 4.16 Data Perbandingan Jarak

Dari data uji coba (Tabel 4.4) dapat ditarik beberapa kesimpulan:

1. Pada umumnya, jarak total yang didapat dari hasil perhitungan Minimal Rectilinear Steiner Spanning Tree (Final MRST) berada antara jarak total pada perhitungan Minimum Spanning Tree dan Minimum Rectilinear Spanning Tree (tanpa *steiner point*). Namun, untuk beberapa kasus jarak Minimum Rectilinear Steiner Spanning Tree (Final MRST) dapat sama atau lebih pendek daripada perhitungan jarak Minimum Spanning Tree (Dapat dilihat pada data ke-4, 5, 7, 9). Hal ini terjadi jika jarak yang menghubungkan terminal dengan terminal dapat dipersingkat dengan jarak antara sebuah terminal dengan *steiner point*. Untuk lebih jelas dapat dilihat pada gambar 4.17.



Gambar 4.17 *Minimum Rectilinear Steiner Spanning Tree (Kiri) dan Minimum Spanning Tree (Kanan)*

Anggaplah 1 kotak = 10 satuan. Pada gambar 4.16 dapat dilihat bahwa jarak ac pada kedua gambar adalah 20 satuan. Jarak ab pada Minimum Spanning Tree (Kanan) adalah 22.361 satuan sehingga jarak total untuk Minimum Spanning Tree adalah  $20 + 20.361 = 40.361$  satuan sedangkan jarak §1b (kiri) hanya 20 satuan sehingga jarak total untuk Minimum Rectilinear Steiner Spanning Tree adalah  $20 + 20 = 40$  satuan.

2. Rasio Minimum Rectilinear Steiner Spanning Tree dengan Minimum Rectilinear Spanning Tree tidak ditentukan oleh banyaknya *Steiner points* yang diperoleh. Hasil ini dapat dilihat pada data 3 dan 7 serta data 11 dan 13. Data 3 dan 7 mempunyai rasio C/B (perbandingan Minimum Rectilinear Steiner Spanning Tree dengan Minimum Rectilinear Spanning Tree dengan) = 0,75 dan jumlah data sebanyak 4 buah, tetapi data 3 mempunyai 2 buah *steiner point* dan data 7 hanya mempunyai sebuah *steiner point*. Begitu pula data 11 dan 13, masing-masing mempunyai rasio C/B = 0,875 dan jumlah data sebanyak 5 buah, tetapi data 11 mempunyai *steiner point* sebanyak 2 dan data 13 hanya mempunyai *steiner point* sebanyak 1 buah.

3. Jumlah *steiner point* maksimal yang dapat ditambahkan adalah n-2, dengan n adalah jumlah terminal yang ada.
4. Rasio rata-rata perbandingan antara Minimum Rectilinear Steiner Spanning Tree dengan Minimum Rectilinear Spanning Tree (C/B) adalah 0.822. Hal ini berarti penambahan *steiner point* selalu mengurangi jumlah jarak total dibandingkan dengan perhitungan tanpa *steiner point* (Minimal Rectilinear Spanning Tree) dengan rata-rata pengurangan jarak sekitar 17,8 %.
5. Rasio perbandingan rata-rata antara Minimum Rectilinear Steiner Spanning Tree dengan Minimum Spanning Tree (C/A) adalah 1,064. Hal ini berarti pada umumnya jarak total dari perhitungan Minimum Spanning Tree lebih pendek dibandingkan dengan perhitungan Minimum Rectilinear Steiner Spanning Tree.

#### **4.4. Kelebihan Program**

Dari program pembentukan Rectilinear Steiner Spanning Tree yang telah dibuat, dapat diambil beberapa kelebihan:

1. Pengguna dapat melihat proses dan hasil per iterasi beserta tampilan layoutnya. Pengguna juga dapat menyimpan, mencetak data dan layout hasil perhitungan tiap iterasi.
2. Hasil dibuat dalam bentuk tabel sehingga memudahkan membaca data.
3. Input data dengan 2 cara, menggunakan input secara konvensional atau langsung pada lembar kerja.

4. Data terminal dapat disimpan sehingga dapat digunakan lagi jika diperlukan.

#### **4.5. Kekurangan Program**

1. Hanya dapat menginputkan terminal sebanyak 50 data, dan nilai pos X 260 serta pos Y 175.
2. Jalur yang dibentuk hanya ada satu jenis dan tidak dapat dipilih oleh pengguna sedangkan pada perhitungan rectilinear metric, dapat dihasilkan berbagai macam jalur (walaupun jumlah jaraknya sama).

#### **4.6. Alternatif Pengembangan**

1. Program Minimum Rectilinear Steiner Spanning Tree ini dapat dikembangkan untuk membuat desain fisik sirkuit VLSI, yaitu untuk penempatan cell dan fase routing (untuk menemukan layout koneksi kabel antar terminal pada blok sirkuit atau gerbang).
2. Pengembangan juga dapat dilakukan dengan pembuatan variasi program steiner tree yang lain, misalnya Euclidean Steiner Tree, Geometric Steiner Tree, QoS constrained Steiner Tree, dan sebagainya.

## BAB V

### PENUTUP

Pada bagian akhir dari penulisan skripsi ini dicantumkan beberapa kesimpulan dan saran dari hal-hal yang terkait dengan program pembentukan Minimum Rectilinear Steiner Spanning Tree berdasarkan bab-bab sebelumnya.

#### 5.1 Kesimpulan

Di dalam pembuatan program pembentukan Rectilinear Steiner spanning Tree dapat disimpulkan bahwa:

1. Algoritma Iterated 1-Steiner pada pembentukan Minimum Rectilinear Steiner Spanning Tree cocok digunakan dalam pembentukan jaringan pipa air karena dapat mengurangi jumlah jarak total yang dibentuk dibandingkan dengan Minimum Rectilinear Spanning Tree. Namun demikian, metode ini masih terlihat kaku karena belum ada batasan untuk daerah – daerah yang boleh dilewati atau tidak.
2. Dari hasil percobaan terhadap 15 data dengan jumlah terminal antara 3 - 5 didapat rasio rata-rata perbandingan antara Minimum Rectilinear Steiner Spanning Tree dengan Minimum Rectilinear Spanning Tree adalah 0.822 dan rasio perbandingan rata-rata antara Minimum Rectilinear Steiner Spanning Tree dengan Minimum Spanning Tree adalah 1,064.
3. Untuk kasus tertentu, yaitu jika jarak yang menghubungkan terminal dengan terminal dapat dipersingkat dengan jarak antara sebuah terminal dengan steiner poin, algoritma Iterated 1-Steiner bahkan dapat

menghasilkan jarak yang lebih minimal dibandingan penggunaan Minimal Spanning Tree (Keterangan lebih jelas dapat dilihat pada halaman 80)

4. Jumlah Steiner Poin maksimal yang dapat dihasilkan tidak lebih banyak daripada jumlah terminal yang ada, maksimal  $n - 2$  dengan  $n$  adalah jumlah terminal yang ada. Rasio Minimum Rectilinear Steiner Spanning Tree dengan Minimum Rectilinear Spanning Tree juga tidak ditentukan oleh banyaknya *Steiner points* yang diperoleh (Keterangan lebih jelas dapat dilihat pada halaman 80).
5. Steiner poin yang dihasilkan berderajat tiga atau empat. Jika steiner poin mempunyai derajat = 2 maka bagian tersebut biasanya merupakan titik pojok rectilinear metric sehingga tidak perlu ditambahkan steiner poin.

## 5.2 Saran

Sebagai pertimbangan untuk pengembangan program ini, ada beberapa saran yang dapat penulis sampaikan:

1. Dalam program ini, grid – grid yang sudah ditentukan, begitu juga batas maksimal pos X dan Pos Y. Maka penulis menyarankan agar jumlah grid, pos X dan Pos Y dapat diinputkan secara fleksibel menurut kepentingan pengguna dan juga disediakan berbagai kemungkinan alternatif jalur.
2. Program ini dapat dikembangkan lebih lanjut, misalnya untuk mencari jarak pada suatu rangkaian elektronika, di mana ada bagian – bagian/daerah-daerah tertentu yang tidak dapat dilewati garis dan desain fisik sirkuit VSLI, yaitu untuk penempatan cell dan fase routing.

## **DAFTAR PUSTAKA**

..., *Steiner Tree More is less*, <http://www.diku.dk/geosteiner/>

Agustinus Nalwan, *Membuat Program Profesional Secara Cepat dengan VB*, PT. Elex Media Komputindo, Jakarta, 2004

Ario Suryo Kusumo, *Buku Latihan Microsoft Visual Basic 6.0*, PT. Elex Media Komputindo, Jakarta, 2001

Elaheh Bozorgzadeh, dkk., *Creating and Exploiting Flexibility in Rectilinear Steiner Trees* <http://www.cs.ucla.edu/~elib/publications/journal/flex-tcad.pdf>.

Ion I. Mandoiu, dkk., *A New Heuristic for rectilinear Steiner Trees*, [www.engr.uconn.edu/~ion/FILES/pub/iccad99.ps](http://www.engr.uconn.edu/~ion/FILES/pub/iccad99.ps)

Jeff Griffith, dkk., *Closing the Gap: Near - Optimal Steiner Trees in Polynomial Time* , [www.cs.virginia.edu/~robins/papers/steiner\\_journal\\_final.ps](http://www.cs.virginia.edu/~robins/papers/steiner_journal_final.ps).

Jonathan Coombs, *Efficiently Solving the Euclidean Steiner Tree Problem in Two Dimensions*, <http://www.css.tayloru.edu/~jcoombs/proj/srsem/newproj/docs/docs3.htm>.

Jong Jek Siang, Drs.,M.Sc., *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer*, Yogyakarta: Penerbit ANDI, 2002.

Pamungkas, Ir., *Tip&Trik Microsoft Visual Basic 6.0*, PT. Elex Media Komputindo, Jakarta, 2001

Swawki Areibi, *An Efficient rectilinear steiner tree alg. 4 VLSI Global Routing*  
[http://wolfman.eos.uoguelph.ca/~sareibi/PUBLICATIONS\\_dr/abs-conferences/CCECE\\_01\\_paper.pdf](http://wolfman.eos.uoguelph.ca/~sareibi/PUBLICATIONS_dr/abs-conferences/CCECE_01_paper.pdf)

Wiryanto Dewobroto, *Aplikasi Sain dan Teknik dengan Visual Basic 6.0*, PT. Elex  
Media Komputindo, Jakarta, 2003

Yumei Huo, *Steiner Tree Problem in Graphs*,  
<http://www.cs.njit.edu/~czumaj/TEACHING/CIS786/Fall2002/Homework1/Steiner-Tree.html>

# **LAMPIRAN A**

## **( Data & Hasil Perhitungan Jaringan Pipa Air Kampus III USD)**

## HASIL PERHITUNGAN

---

### Data Terminal dan Steiner Poin

Nomor	Nama	Pos X	Pos Y	Derajat
1	A1	115	10	2
2	A2	130	10	1
3	B1	110	50	2
4	B2	110	60	2
5	B3	110	70	2
6	C1	170	65	1
7	D1	100	160	1
8	D2	115	160	1
9	E1	105	90	2
10	E2	105	95	2
11	E3	105	100	2
12	F1	155	80	2
13	§1	110	80	3
14	§2	105	160	3

Skala 1 : 50

### Rute Jalur

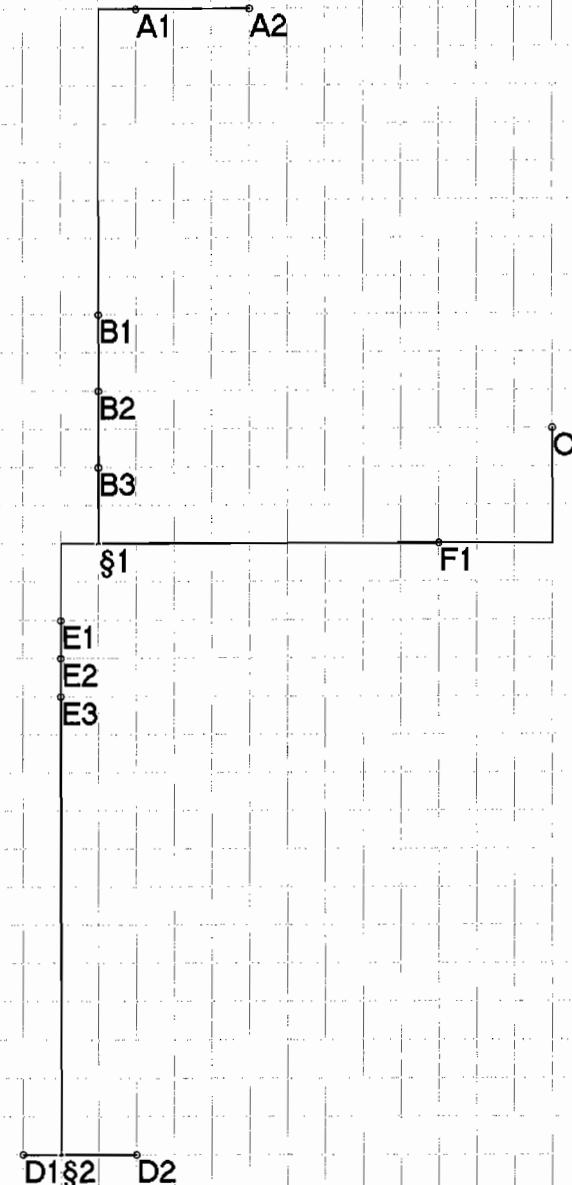
No.	Awal	Akhir	Jarak (mm)
1	A1	A2	750
2	A1	B1	2250
3	B1	B2	500
4	B2	B3	500
5	B3	§1	500
6	§1	E1	750
7	E1	E2	250
8	E2	E3	250
9	§1	F1	2250
10	F1	C1	1500
11	E3	§2	3000
12	§2	D1	250
13	§2	D2	500

Jumlah Titik : 14

Jumlah Steiner Point : 2

Jarak Total : 13250 mm

Print at : 10/9/04 4:49:31 AM



**HASIL PERHITUNGAN**  
=====

**Data Terminal dan Steiner Poin**

Nomor	Nama	Pos X	Pos Y	Derajat
1	A1	115	10	1
2	A2	130	10	3
3	B1	140	5	1
4	C1	120	65	2
5	C2	110	75	2
6	D1	170	70	1
7	D2	170	75	2
8	D3	170	85	2
9	F1	95	110	1
10	G1	75	115	1
11	G2	85	115	2
12	G3	105	115	2
13	H1	115	110	1
14	I1	85	175	1
15	J1	95	165	3
16	E1	140	100	2
17	E2	110	105	3
18	J2	115	165	1
19	B2	130	25	2
20	§1	95	115	4
21	§2	115	105	3

Skala 1 : 50

**Rute Jalur**

No.	Awal	Akhir	Jarak (mm)
1	A1	A2	750
2	A2	B1	750
3	A2	B2	750
4	B2	C1	2500
5	C1	C2	1000
6	C2	E2	1500
7	E2	§2	250
8	§2	H1	250
9	E2	G3	750
10	G3	§1	500
11	§1	F1	250
12	§1	G2	500
13	G2	G1	500
14	§2	E1	1500
15	E1	D3	2250
16	D3	D2	500
17	D2	D1	250
18	§1	J1	2500
19	J1	I1	1000
20	J1	J2	1000

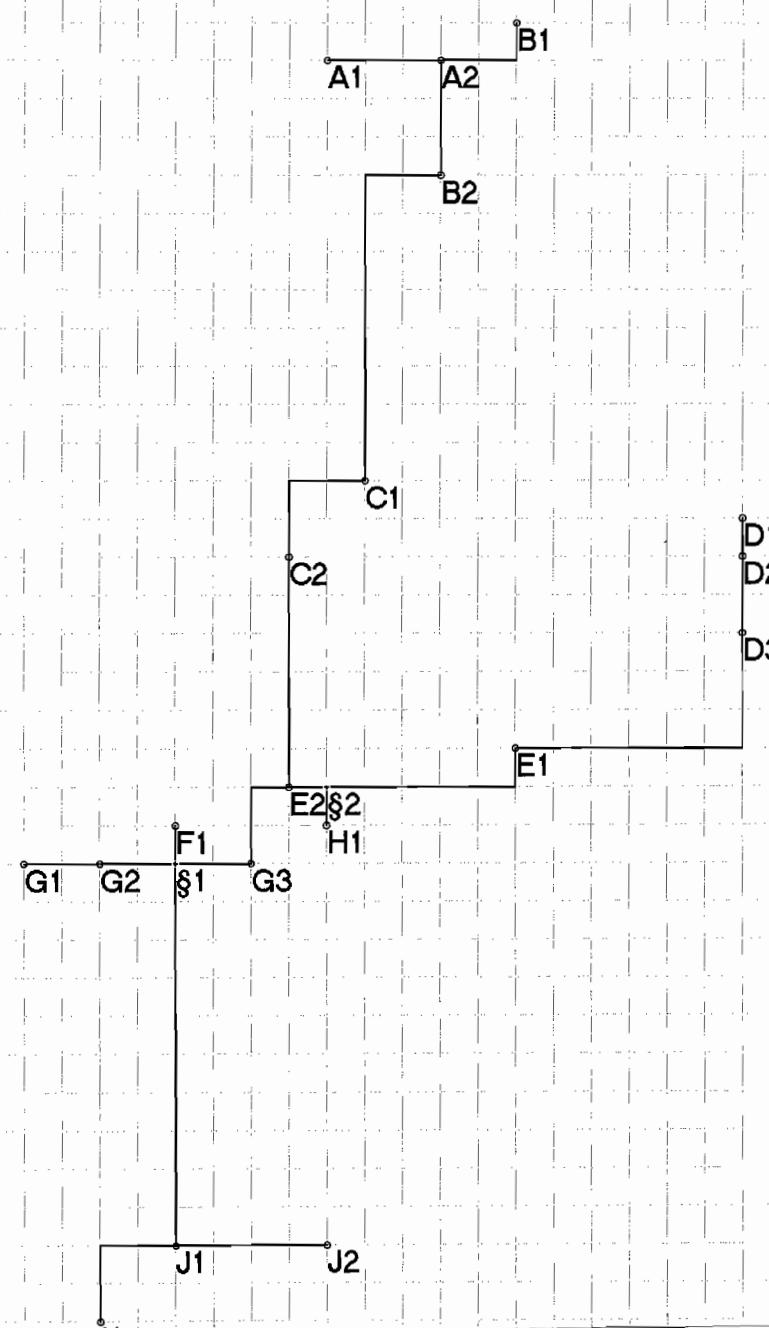
Jumlah Titik : 21

Jumlah Steiner Point : 2

Jarak Total : 19250 mm

**PRINT LAYOUT PERHITUNGAN**

Print at : 10/9/04 4:55:14 AM



## HASIL PERCINTUNGAN

---

### Data Terminal dan Steiner Poin

Nomor	Nama	Pos X	Pos Y	Derajat
1	a1	110	45	1
2	a2	110	60	2
3	a3	110	70	2
4	b1	120	55	1
5	c1	130	75	2
6	c2	130	85	2
7	d1	160	70	2
8	d2	160	90	2
9	e1	165	100	1
10	e2	115	100	1
11	§1	110	55	3
12	§2	130	70	3

Skala 1 : 50

### Rute Jalur

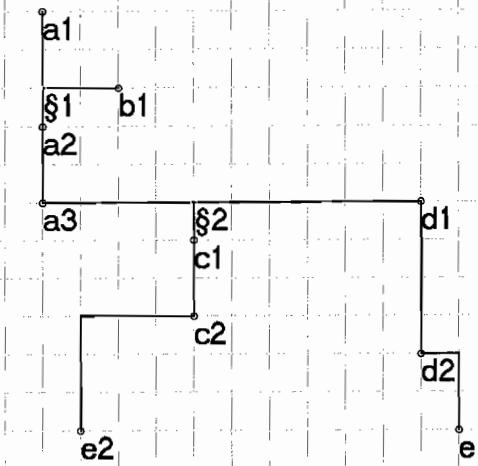
No.	Awal	Akhir	Jarak (mm)
1	a1	§1	500
2	§1	a2	250
3	§1	b1	500
4	a2	a3	500
5	a3	§2	1000
6	§2	c1	250
7	c1	c2	500
8	§2	d1	1500
9	d1	d2	1000
10	d2	e1	750
11	c2	e2	1500

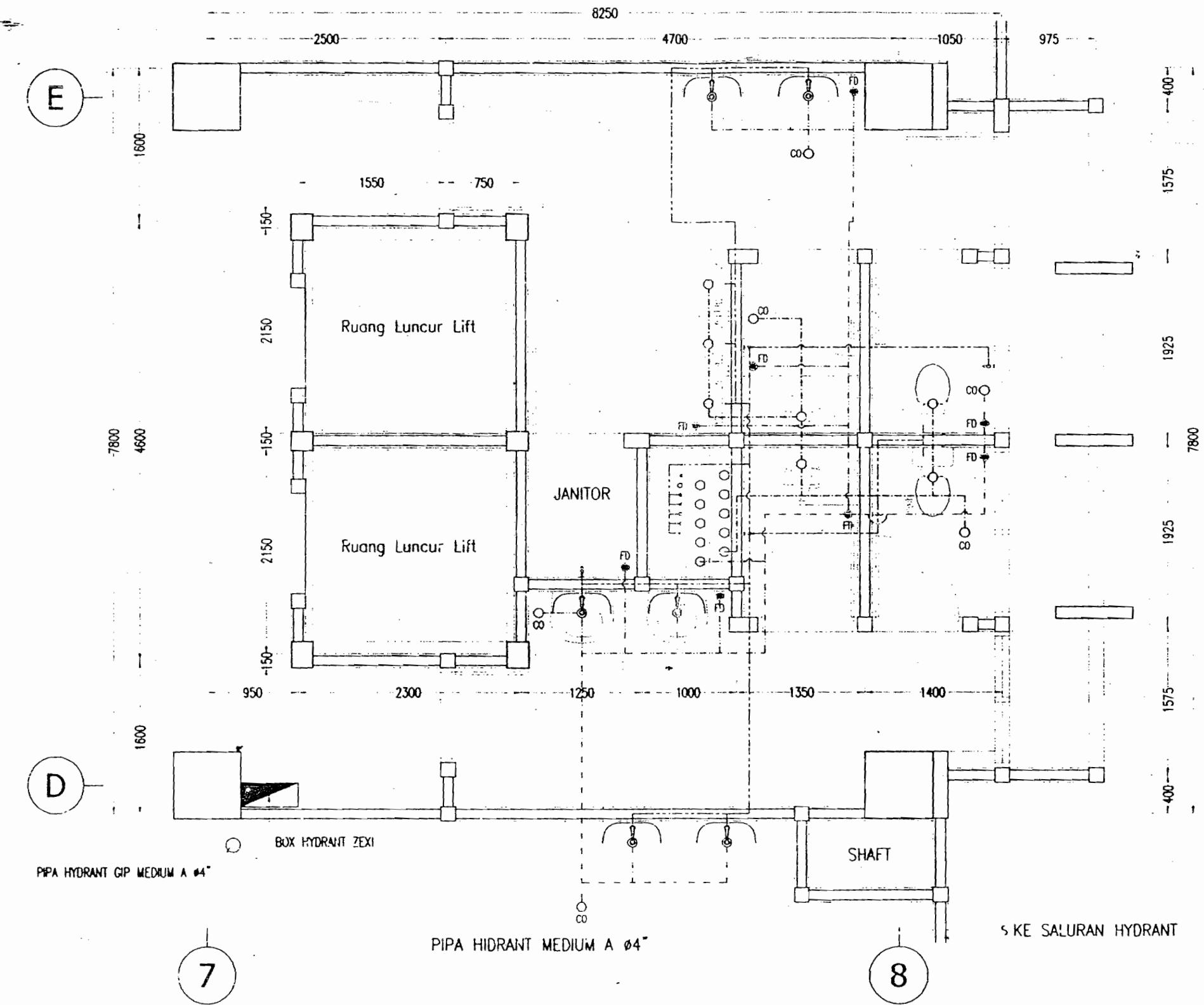
Jumlah Titik : 12

Jumlah Steiner Point : 2

Jarak Total : 8250 mm

Print at : 10/9/04 4:59:03 AM





# DENAH INSTALASI AIR BERSIH, AIR KOTOR & AIR KOTORAN ( TOILET LT. DASAR, 2, 3 & 4 )

SKALA 1 : 50

# **LAMPIRAN B**

## **(Listing Program)**

## **MODULE I**

```
Type noktah 'Data Terminal
    nomor As Integer
    nama As String
    posX As Integer
    posY As Integer
    derajat As Integer
End Type

Type jalur 'Jalur yang dibentuk dari hasil perhitungan
    nomor As Integer
    awal As String
    akhir As String
    jarak As Integer
End Type

Type edit ' Tipe data untuk perubahan pada form add2/MSFlexGrid
    kolom As Integer
    baris As Integer
    ubah As String
End Type

Public terminal(1 To 50) As noktah 'Menyimpan data terminal (input user)
Public jml As Integer ' Jumlah terminal
Public skala As Integer 'Keterangan nilai skala
Public satuan As String 'Keterangan satuan skala

Type simpanPrim 'simpan data iterasi MRST
    dataPrim(1 To 50) As noktah
    dataJumlah As Integer
    dataJalur(1 To 100) As jalur
    dataJarak As Integer
    dataJumlahJalur As Integer
End Type

Public simpan(1 To 50) As simpanPrim 'Simpan data iterasi MRST
Public Seluruh As Integer 'Iterasi ke-n yg ingin diproses
Public tiga As Integer 'jumlah iterasi

'simpan data Jalur Biasa
Public JalurPB(1 To 100) As jalur
Public tigaPB As Integer 'banyak jalur yang dihasilkan

'penanda MRST atau Prim biasa
Public PB As Integer
```

## **MODULE II**

```
Sub errHandler()
    errdesc = Err.Description
    errnum = Err.Number
    Beep
    MsgBox "Nomor kesalahan " & errnum & Chr(10) & Chr(13) & "Terjadi karena: " & errdesc,
    vbCritical, "Pesan Kesalahan"
    Exit Sub
End Sub
```

## **FORM TAMPILAN AWAL**

```
Private Sub Label1_Click()
    Unload Me
End Sub

Private Sub Label2_Click()
    Unload Me
    frmUtama.Show
End Sub
```

```

Private Sub Form_Load()
    Left = (Screen.Width - Width) \ 2
    Top = (Screen.Height - Height) \ 2
End Sub

Private Sub Timer1_Timer()
    ProgLoad.Value = ProgLoad.Value + 2
    If ProgLoad.Value = 100 Then
        Unload Me
        frmUtama.Show
    End If
End Sub

```

**FORM UTAMA**

```

Sub tutup_menu()
mnuFile.Enabled = False
mnuHelp.Enabled = False
Toolbar1.Enabled = False
frmNew.Enabled = False
End Sub

Sub buka_menu()
mnuFile.Enabled = True
mnuHelp.Enabled = True
Toolbar1.Enabled = True
frmNew.Enabled = True
End Sub

Private Sub MDIForm_Load()
'panggilAsisten
End Sub

Private Sub mnuAbout_Click()
frmUtama.Enabled = False
frmAbout.Show
End Sub

Private Sub mnuContent_Click()
With cdlHelp
    .HelpFile = App.Path & "\helpmrst.hlp"
    .HelpCommand = &HF
    .ShowHelp
End With
End Sub

Private Sub mnuEdit_Click()
    tutup_menu
    frmAdd2.Top = 0
    frmAdd2.Left = 0
    frmAdd2.Show
End Sub

Private Sub mnuExit_Click()
y = MsgBox("Anda yakin ingin keluar dari program?", vbYesNo + vbQuestion, "Exit")
If y = vbYes Then Unload Me
End Sub

Private Sub mnuNew_Click()
mnuFile.Enabled = False
mnuHelp.Enabled = False
Toolbar1.Enabled = False
frmSkala.Top = 0
frmSkala.Left = 0
frmSkala.Show
End Sub

Private Sub mnuOpen_Click()
CommonDialog1.CancelError = True
On Error GoTo errHandler
    ' Set flags
    frmUtama.CommonDialog1.Flags = cdlOFNHideReadOnly

```

```

' Set filters
frmUtama.CommonDialog1.Filter = "(*.txt)|*.txt|"
' Specify default filter
frmUtama.CommonDialog1.FilterIndex = 2

' Membuka dialog box
frmUtama.CommonDialog1.ShowOpen

frmOpcn.BukaFile
frmNew.Enabled = True

errHandler:
mnuFile.Enabled = True
mnuHelp.Enabled = True
Toolbar1.Enabled = True
End Sub
Private Sub mnuProcess_Click()
PB = 0
If jml < 1 Then
    Dim y As Integer
    y = MsgBox("Data Terminal Belum Ada", vbCritical + vbOKOnly, "Pesan")
    Exit Sub
Else
    frmNew.MousePointer = 11
    tutup_menu
    frmRun.Top = 0
    frmRun.Left = 7350
    Load frmRun
    frmRun.Show
End If
End Sub
Private Sub mnuSave_Click()
If jml < 1 Then
    Dim y As Integer
    y = MsgBox("Data Terminal Belum Ada", vbCritical + vbOKOnly, "Pesan")
    Exit Sub
Else
    frmsave.Visible = False
    frmsave.SimpanTerminal
End If

End Sub
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

Select Case Button.Index
    Case 1:
        mnuNew_Click
    Case 2:
        mnuOpen_Click
    Case 3:
        If skala < 1 Then
            y = MsgBox(" Skala dan Satuan belum diinputkan", vbOKOnly + vbCritical,
"Peringatan")
        Else
            mnuEdit_Click
        End If
    Case 4:
        mnuProcess_Click
    Case 5:
        mnuSave_Click
    Case 6:
        mnuContent_Click
    Case 7:
        mnuExit_Click
End Select
End Sub

Private Sub panggilAsisten()
Dim asisten As IAgentCtlCharacterEx
Const datapath = "merlin.acs"
    Agent1.Characters.Load "merlin", datapath
    Set asisten = Agent1.Characters("merlin")

```

```

asisten.LanguageID = &H409
asisten.MoveTo 65, 50, 10
asisten.Show
asisten.Play ("greet")
asisten.Speak "Welcome to the Rectilinear Steiner Spanning Tree Program"
asisten.Speak "You can put some points as terminals and find the shortest path with
rectilinear metric and steiner points"
asisten.Speak "Thank you"
asisten.Play ("lookdown")
asisten.Hide

End Sub

```

## **FORM SKALA & SATUAN**

```

Private Sub panggilNew()
    frmNew.Top = 0
    frmNew.Left = 0
    frmNew.Cls
    frmNew.Refresh
    Load frmNew
End Sub
Private Sub cmdCancel_Click()
If jml < 1 Then
    Unload Me
Else
    'Meload Lembar Kerja
    panggilNew
    Unload Me
End If

frmUtama.mnuFile.Enabled = True
frmUtama.mnuHelp.Enabled = True
frmUtama.Toolbar1.Enabled = True

End Sub

Private Sub cmdOK_Click()

If txtSat.Text = "" Then
    teks = MsgBox("Data harus lengkap", vbCritical + vbOKOnly, "Pesan")
    txtSat.SetFocus
ElseIf txtSkala.Text = "" Then
    teks = MsgBox("Data harus lengkap", vbCritical + vbOKOnly, "Pesan")
    txtSkala.SetFocus
ElseIf txtSkala.Text = 0 Or txtSkala.Text > 100 Then
    teks = MsgBox("Skala Tidak Boleh Nol atau" & Chr(10) & Chr(13) & "Skala Tidak Boleh
> 100", vbCritical + vbOKOnly, "Pesan")
    txtSkala.Text = ""
    txtSkala.SetFocus
Else
    jml = 0

    'menyimpan input nilai satuan dan skala
    skala = txtSkala.Text
    satuan = txtSat.Text

    'menampilkan menu
    If skala > 0 Then
        frmUtama.mnuEdit.Enabled = True
    End If

    'Menonaktifkan form skala
    Unload Me

    'Mengecek jumlah Titik dan mengaktifkan menu Run dan save
    If jml > 0 Then
        frmUtama.mnuProcess.Enabled = True
        frmUtama.mnuSave.Enabled = True
        frmUtama.mnuEdit.Enabled = True
    End If

    'Mengosongkan data

```

```

simpan(1).dataJalur(1).nomor = 0
Erase JalurPB
Seluruh = 0

'Meload Lembar Kerja
panggilNew

End If
frmUtama.mnuFile.Enabled = True
frmUtama.mnuHelp.Enabled = True
frmUtama.Toolbar1.Enabled = True
Exit Sub

End Sub

Private Sub txtSkala_KeyPress(KeyAscii As Integer)
If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then
    Beep
    KeyAscii = 0
End If

End Sub

```

## Form OPEN

```

Sub BukaFile()
Dim data1 As String 'menampung data dari file yang dibuka
Dim cari, cari2 As String
Dim m, n, i As Integer
Dim data(1 To 100) As String
On Error GoTo errHandler
frmUtama.tutup_menu

'membuka file dan menyimpan pada data1
Open frmUtama.CommonDialog1.FileTitle For Input As #1
    While Not EOF(1)
        Input #1, data1
    Wend
Close #1

cari = Chr(10) + Chr(13) 'karakter pembatas
cari2 = Chr(165)

Dim coba

coba = Split(data1, cari)
satuan = coba(0)
skala = coba(1)

For m = 2 To UBound(coba)
    coba2 = Split(coba(m), cari2)
    terminal(m - 1).nomor = m - 1
    terminal(m - 1).nama = coba2(0)
    terminal(m - 1).posX = CInt(coba2(1))
    terminal(m - 1).posY = CInt(coba2(2))
Next m

jml = UBound(coba) - 1

'Mengecek jumlah Titik dan mengaktifkan menu Run dan save
If jml > 0 Then
    frmUtama.mnuProcess.Enabled = True
    frmUtama.mnuSave.Enabled = True
    frmUtama.mnuEdit.Enabled = True
End If

'verififikasi data
simpan(1).dataJalur(1).nomor = 0
Erase JalurPB

```



```

Seluruh = 0
frmNew.Top = 0
frmNew.Left = 0
frmNew.Cls

ErrorHandler:
End Sub

```

### **Form NEW (LEMBAR KERJA)**

```

Public Sub Form_Activate()

Dim i, selisih As Integer
If jml <> 0 Then
    For i = 1 To jml
        Me.Circle (terminal(i).posX, terminal(i).posY), 0.4, RGB(255, 0, 0)
        Print terminal(i).nama
    Next i
End If

If simpan(1).dataJalur(1).nomor > 0 And PB = 0 Then
    For i = 1 To simpan(Seluruh).dataJumlahJalur
        Me.Line (simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posX,
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posY)-
(simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posX,
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posY), , B
        Me.Line (simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posX,
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posY)-
(simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posX,
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posY), , B
    Next i
End If

If PB = 1 And JalurPB(1).nomor > 0 Then
    For i = 1 To tigaPB
        Me.Line (terminal(JalurPB(i).awal).posX, terminal(JalurPB(i).awal).posY)-
(terminal(JalurPB(i).akhir).posX, terminal(JalurPB(i).akhir).posY), vbBlack
    Next i
End If

If PB = 0 And Seluruh <> 0 Then
    For i = simpan(1).dataJumlah + 1 To simpan(Seluruh).dataJumlah
        Me.Circle (simpan(Seluruh).dataPrim(i).posX, simpan(Seluruh).dataPrim(i).posY),
0.4, vbYellow
        Print simpan(Seluruh).dataPrim(i).nama
    Next i
End If

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
'If (x Mod 5) = 0 And (y Mod 5) = 0 Then
Label1.Top = y + 0.5
Label1.Left = x + 2.5
Label1.Caption = CInt(x) & "," & CInt(y)
Label2.Caption = CInt(x)
Label3.Caption = CInt(y)
'End If
Form_Activate
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
If (x Mod 5) = 0 And (y Mod 5) = 0 Then
    If Button = 1 Then ' button = 1 : klik kiri
    'PopupMenu frmUtama.mnuPerintah
    Dim j As Integer
    j = 0

```

```

        For i = 1 To jml
            If Label2.Caption = terminal(i).posX And Label3.Caption = terminal(i).posY
Then
            Load frmdelete
            frmdelete.txtNamaTtk = terminal(i).nama
            j = 1
        End If
Next i

        If j = 0 Then
            frmAdd.Show
            frmAdd.txtPosX.Text = Label2.Caption
            frmAdd.txtPosY.Text = Label3.Caption
        End If

    End If
End If
End Sub

```

## **Form ADD**

```

Private Sub cmdCancel_Click()
'Menonaktifkan form Add
Unload Me

'Mengecek jumlah Titik dan mengaktifkan menu Run dan save
If jml > 0 Then
    frmUtama.mnuProcess.Enabled = True
    frmUtama.mnuSave.Enabled = True
End If

'Meload Lembar Kerja
frmNew.Top = 0
frmNew.Left = 0
Load frmNew

End Sub

Private Sub cmdOK_Click()

If cekTitik() = False Then
    If txtNamaTtk.Text = "" Then
        y = MsgBox(" Input Nama Terminal", vbCritical + vbOKOnly, "Pesan")
    Else
        jml = jml + 1
        terminal(jml).nomor = jml
        terminal(jml).nama = txtNamaTtk.Text
        terminal(jml).posX = txtPosX.Text
        terminal(jml).posY = txtPosY.Text
        txtNamaTtk.Text = ""
        txtPosX.Text = ""
        txtPosY.Text = ""
        txtNamaTtk.SetFocus
        txtJml.Text = jml

        If jml > 0 Then
            frmUtama.mnuProcess.Enabled = True
            frmUtama.mnuSave.Enabled = True
        End If

        Unload Me

        'Verifikasi Data & tampilan
        simpan(1).dataJalur(1).nomor = 0
        Erase JalurPB
        Seluruh = 0
        frmNew.Cls
        frmNew.Form_Activate
    End If
Else
    txtNamaTtk.SetFocus
End If

```

```

End Sub

Private Sub Form_Load()
txtJml.Text = jml
End Sub

Private Function cekTitik() As Boolean
Dim i As Integer
Dim a As Boolean
a = False

For i = 1 To jml
    If (StrComp(txtNamaTtk.Text, terminal(i).nama) = 0) Then
        MsgBox "Nama Titik yang Anda masukkan sudah ada", 16, "Pesan"
        a = True
        txtNamaTtk.Text = ""
        txtNamaTtk.SetFocus
        txtJml.Text = jml
        Exit For
    Else
    End If
Next i
cekTitik = a
End Function

Private Sub Form_Unload(Cancel As Integer)
frmUtama.buka_menu
End Sub

Private Sub txtPosX_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then
    Beep
    KeyAscii = 0
End If

End Sub

Private Sub txtPosY_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then
    Beep
    KeyAscii = 0
End If

End Sub

```

### **Form DELETE**

```

Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
On Error GoTo errHandler
Dim i, j As Integer
j = 0

For i = 1 To jml
    If (StrComp(terminal(i).nama, txtNamaTtk.Text) = 0) Then
        j = i
    End If
Next i

If j = 0 Then
    MsgBox "Titik yang Anda masukkan tidak ada", 16, "Pesan"
    txtNamaTtk.Text = ""
    txtNamaTtk.SetFocus
End If

If j > 0 Then
    Dim pesan As Integer

```

```

pesan = MsgBox("Anda yakin ingin menghapus titik?", 4 + 256 + 32, "Hapus Titik")
If pesan = 6 Then
    For i = j To jml
        terminal(i).nomor = terminal(i).nomor
        terminal(i).nama = terminal(i + 1).nama
        terminal(i).posX = terminal(i + 1).posX
        terminal(i).posY = terminal(i + 1).posY
    Next i
    jml = jml - 1

    'verifikasi data
    simpan(1).dataJalur(1).nomor = 0
    Erase JalurPB
    Seluruh = 0
Else
    Unload Me
End If

txtNamaTtk.Text = ""
txtNamaTtk.SetFocus
Else
End If

Unload Me
frmNew.Cls
frmNew.Form_Activate

ErrorHandler:
End Sub

Private Sub Form_Unload(Cancel As Integer)
frmUtama.buka_menu
End Sub

```

### **Form Edit**

```

Dim perubahan(1 To 100) As edit
Dim nilai, a As Integer
Dim cek As Integer

Private Sub Buka()
cmdDelete.Enabled = True
cmdAdd.Enabled = True
cmdExit.Enabled = True
cmdFinish.Enabled = False
cmdCancel.Enabled = False
End Sub

Private Sub tutup()
cmdDelete.Enabled = False
cmdExit.Enabled = False
cmdAdd.Enabled = False
cmdFinish.Enabled = True
cmdCancel.Enabled = True
End Sub

Private Sub cmdAdd_Click()

On Error GoTo ErrorHandler:

a = InputBox("Masukkan jumlah Terminal : ", "Input Jumlah Terminal")

If StrComp(a, "") > 0 Or StrComp(a, "") < 0 Then
    tutup
    With MSFlexGrid1
        .Rows = jml + 1 + a
        .Cols = 4

        For i = 1 To jml + a
            .TextMatrix(i, 0) = i
        Next i
    End With

'untuk verifikasi data tidak boleh kosong

```

```

For z = jml + 1 To jml + a
    terminal(z).posX = -1
    terminal(z).posY = -1
Next z

' memindahkan TextBox ke tempat yang aktif sesegera mungkin
MoveTextBox
End If
Exit Sub

errHandler:
End Sub

Private Sub cmdCancel_Click()
For z = jml + 1 To jml + a
    terminal(z).posX = 0
    terminal(z).posY = 0
Next z

a = 0
Buka
Tampil_Data

End Sub

Private Sub cmdDelete_Click()
Dim b As String

b = ""
b = InputBox("Masukkan Nama Terminal: ", "Delete Terminal")

If StrComp(b, "") > 0 Or StrComp(b, "") < 0 Then
    SaveChange
    Dim i, j As Integer      'bantuan perulangan
    j = 0
    For i = 1 To jml
        If (StrComp(terminal(i).nama, b) = 0) Then
            j = i
        End If
    Next i

    If j = 0 Then
        y = MsgBox("Titik yang Anda masukkan tidak ada", vbCritical + vbOKOnly, "Pesan")
        Exit Sub
    End If

    If j > 0 Then
        Dim pesan As Integer
        pesan = MsgBox("Anda yakin ingin menghapus titik?", 4 + 256 + 32, "Hapus Titik")
        If pesan = 6 Then
            For i = j To jml
                terminal(i).nomor = i
                terminal(i).nama = terminal(i + 1).nama
                terminal(i).posX = terminal(i + 1).posX
                terminal(i).posY = terminal(i + 1).posY
            Next i
            jml = jml - 1
            b = 1
            simpan(1).dataJalur(1).nomor = 0
            Erase JalurPB
            Seluruh = 0
            End If
        End If

        Tampil_Data
    End If
End Sub

Private Sub cmdExit_Click()

If nilai > 0 Then
    y = MsgBox(" Simpan ulang Data?", vbYesNo + vbQuestion, "Pesan")

```

```

If y = vbYes Then
    SaveChange
    If cek = 1 Then Exit Sub

    'verifikasi data
    simpan(1).dataJalur(1).nomor = 0
    Erase JalurPB
    Seluruh = 0
Else
    nilai = 0
    Unload Me
End If

End If

'verifikasi data
simpan(1).dataJalur(1).nomor = 0
Erase JalurPB
Seluruh = 0
frmNew.Cls
frmNew.Form_Activate

If jml > 0 Then
    frmUtama.mnuProcess.Enabled = True
    frmUtama.mnuSave.Enabled = True
End If

Unload Me
End Sub

Private Sub SaveChange()
Dim ubahstring As Integer
Dim ubah(1 To 100) As noktah
Dim z As Integer

'Mengecek data perubahan tidak lengkap
cek = 0
If nilai > 0 Then
    For i = 1 To nilai
        If perubahan(i).ubah = "" Then
            For z = i To nilai - 1
                perubahan(z).baris = perubahan(z + 1).baris
                perubahan(z).kolom = perubahan(z + 1).kolom
                perubahan(z).ubah = perubahan(z + 1).ubah
            Next z
            nilai = nilai - 1
            y = MsgBox("Data tidak Lengkap " & Chr(13) & Chr(10) & "Silakan dilengkapi !
", vbCritical + vbOKOnly, "Pesan")
            cek = 1
        End If
        If cek = 1 Then Exit For
    Next i
End If

If cek = 1 Then Exit Sub

'cek perubahan
For i = 1 To nilai
    If perubahan(i).kolom = 2 Or perubahan(i).kolom = 3 Then
        If perubahan(i).ubah Mod 5 <> 0 Then
            cek = 1

            For z = i To nilai - 1
                perubahan(z).baris = perubahan(z + 1).baris
                perubahan(z).kolom = perubahan(z + 1).kolom
                perubahan(z).ubah = perubahan(z + 1).ubah
            Next z
            nilai = nilai - 1
            y = MsgBox("Data harus kelipatan 5 " & Chr(13) & Chr(10) & "Silakan diubah !
", vbCritical + vbOKOnly, "Pesan")
            End If
        End If
    End If

```

```

        If cek = 1 Then Exit For
    Next i

    If cek = 1 Then Exit Sub

    For i = 1 To nilai
        If perubahan(i).kolom = 1 Then
            terminal(perubahan(i).baris).nama = perubahan(i).ubah
        ElseIf perubahan(i).kolom = 2 Then
            ubahstring = CInt(perubahan(i).ubah)
            terminal(perubahan(i).baris).posX = ubahstring
        Else
            ubahstring = CInt(perubahan(i).ubah)
            terminal(perubahan(i).baris).posY = ubahstring
        End If
    Next

    'Mengecek nilai kosong
    For i = jml + 1 To jml + a
        If terminal(i).nama = "" Or terminal(i).posX < 0 Or terminal(i).posY < 0 Then
            cek = 1
            y = MsgBox("Data tidak Lengkap" & Chr(13) & Chr(10) & "Silakan dilengkapi ! ", vbCritical + vbOKOnly, "Pesanan")
        End If
        If cek = 1 Then Exit For
    Next i

    If cek = 1 Then Exit Sub

    'Menyimpan jika ada perubahan jumlah
    If a > 0 Then
        For i = jml + 1 To jml + a
            terminal(i).nomor = i
        Next i
    End If

    jml = jml + a

    'mensemprot kembali nilai data
    a = 0
    nilai = 0

    For i = 1 To jml
        For j = 1 To jml
            If i < j Then
                If StrComp(terminal(i).nama, terminal(j).nama) = 0 And terminal(i).posX = terminal(j).posX And terminal(i).posY = terminal(j).posY Then
                    y = MsgBox("Nama dan Koordinat terminal ke: " & i & " sama dengan nama terminal ke: " & j, vbCritical + vbOKOnly, "Data Kembar")
                    cek = 1
                ElseIf StrComp(terminal(i).nama, terminal(j).nama) = 0 Then
                    y = MsgBox("Nama terminal ke: " & i & " sama dengan nama terminal ke: " & j, vbCritical + vbOKOnly, "Data Kembar")
                    cek = 1
                ElseIf terminal(i).posX = terminal(j).posX And terminal(i).posY = terminal(j).posY Then
                    y = MsgBox("Koordinat terminal ke: " & i & " sama dengan nama terminal ke: " & j, vbCritical + vbOKOnly, "Data Kembar")
                    cek = 1
                Else
                    End If
                End If
            If cek = 1 Then Exit For
            Next j
        If cek = 1 Then Exit For
    Next i

    If cek = 1 Then Exit Sub

    'verifikasi data
    frmNew.Refresh
    simpan(1).dataJalur(1).nomor = 0

```

```

Erase JalurPB
Seluruh = 0
frmNew.Refresh
cek = 0
End Sub

Private Sub cmdFinish_Click()

SaveChange
If cek = 1 Then
    cek = 0
    Exit Sub
End If

'verifikasi data
simpan(1).dataJalur(1).nomor = 0
Erase JalurPB
Seluruh = 0
Buka

End Sub

Private Sub Form_Activate()
nilai = 0
a = 0
Tampil_Data
End Sub

Private Sub moveTextBoxInt()
With MSFlexGrid1
    Text4.Left = .CellLeft + .Left
    Text4.Top = .CellTop + .Top
    Text4.Height = .CellHeight
    Text4.Width = .CellWidth
End With
Text2.Text = MSFlexGrid1.Row
Text3.Text = MSFlexGrid1.Col
End Sub

Private Sub MoveTextBox()
With MSFlexGrid1
    Text1.Left = .CellLeft + .Left
    Text1.Top = .CellTop + .Top
    Text1.Height = .CellHeight
    Text1.Width = .CellWidth
End With
Text2.Text = MSFlexGrid1.Row
Text3.Text = MSFlexGrid1.Col
End Sub

Private Sub Form_Unload(Cancel As Integer)
frmUtama.buka_menu
End Sub

Private Sub MSFlexGrid1_Click()
moveTextBoxInt
Text4.Text = MSFlexGrid1.Text
Text4.Enabled = False
Text4.Visible = False

MoveTextBox
Text1.Text = MSFlexGrid1.Text
Text1.Visible = False
Text1.Enabled = False
End Sub

Private Sub MSFlexGrid1_DblClick()
If MSFlexGrid1.Col = 1 Then
    MoveTextBox
    Text1.Text = MSFlexGrid1.Text
    Text1.Visible = True
    Text1.Enabled = True
    Text1.SetFocus
    nilai = nilai + 1
Else

```

```

moveTextBoxInt
Text4.Text = MSFlexGrid1.Text
Text4.Visible = True
Text4.Enabled = True
Text4.SetFocus
nilai = nilai + 1
End If
End Sub

Private Sub MSFlexGrid1_LeaveCell()
Text1.Enabled = False
Text4.Enabled = False
Text1.Visible = False
If nilai > 0 Then
    If MSFlexGrid1.Col = 1 Then
        perubahan(nilai).baris = Text2.Text
        perubahan(nilai).kolom = Text3.Text
        perubahan(nilai).ubah = Text1.Text
        MSFlexGrid1.Text = Text1.Text
        MSFlexGrid1.SetFocus
    Else
        perubahan(nilai).baris = Text2.Text
        perubahan(nilai).kolom = Text3.Text
        perubahan(nilai).ubah = Text4.Text
        MSFlexGrid1.Text = Text4.Text
        MSFlexGrid1.SetFocus
    End If
End If
End Sub

Private Sub Tampil_Data()
With MSFlexGrid1
    .Rows = jml + 1
    .Cols = 4
    .ColWidth(0) = 500

    'menuliskan judul
    .TextMatrix(0, 0) = " No. "
    .TextMatrix(0, 1) = " Nama "
    .TextMatrix(0, 2) = " Pos X "
    .TextMatrix(0, 3) = " Pos Y "

    If jml > 0 Then
        For i = 1 To jml
            .TextMatrix(i, 0) = terminal(i).nomor
            .TextMatrix(i, 1) = terminal(i).nama
            .TextMatrix(i, 2) = terminal(i).posX
            .TextMatrix(i, 3) = terminal(i).posY
        Next i
    End If
End With
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then
    Beep
    KeyAscii = 0
End If
End Sub

```

## Form RUN

```

Option Explicit

Dim HananPoint(1 To 2500) As noktah      'simpan Hanan Poin
Dim g As Integer                          'data jumlah jarak MRST
Dim dat As Integer                        'jumlah perulangan mencari
steiner point
Dim gPB As Integer                        'data jumlah jarak Prim
Dim PrimBiasa As Integer                 'hasil perhitungan Prim Biasa

```

```

*****
***          Algoritma Prim           ***
*****
Private Function FungsiPrim(point() As noktah, jumlah As Integer, ulang As Integer) As
Integer
Dim i, j As Integer  ' untuk perulangan for
Dim d, e, f As Integer  ' untuk cek sirkuit
Dim jarak(1 To 2500, 1 To 2500) As Integer
Dim Matriks(1 To 2500, 1 To 2500) As Integer
Dim angka(100) As Integer

'set derajat = 0 & kosongkan array
For i = 1 To jumlah
    point(i).derajat = 0
Next i

Erase Matriks
Erase jarak
Erase angka
Erase simpan(ulang).dataJalur

'Hitung jarak
For i = 1 To jumlah
    For j = 1 To jumlah
        If i <> j Then
            jarak(i, j) = Abs(point(i).posX - point(j).posX) + Abs(point(i).posY -
point(j).posY)

            'Mengcopy jarak antar titik ke variabel uji
            Matriks(i, j) = jarak(i, j)
        End If
    Next j
Next i

'variabel bantu
Dim k, count, x, y, cari, JarakTotal As Integer
Dim T(1 To 100) As Integer

JarakTotal = 0  'set jarak awal = 0

'until data jalur
g = 0

'set angka = -1
For e = 1 To jumlah
    angka(e) = -1
Next e

d = 1
angka(d) = 1

'Sembarang titik yang digunakan adalah titik nomor 1
k = 1

'jumlah titik
count = 0

'pohon rentang minimum
T(1) = 1

'Mencari titik dengan bobot terkecil
Do Until count = jumlah - 1  'perulangan dilakukan hingga jumlah titik - 1

'variabel cari diset nilai tertinggi tipe data integer
cari = 32767

For i = 1 To k
    For j = 1 To jumlah
        If cari > Matriks(T(i), j) And Matriks(T(i), j) > 0 Then
            cari = Matriks(T(i), j)
            x = T(i)

```

```

        y = j
    End If
Next j
Next i

'Set tidak membentuk sirkuit
For i = 1 To jumlah
    For j = 1 To jumlah
        If i = x And j = y Then
            JarakTotal = JarakTotal + jarak(i, j)

        'input nilai jalur
        g = g + 1
        simpan(ulang).dataJalur(g).nomor = g
        simpan(ulang).dataJalur(g).awal = x
        simpan(ulang).dataJalur(g).akhir = y
        simpan(ulang).dataJalur(g).jarak = jarak(i, j)

        'Derajat titik
        point(i).derajat = point(i).derajat + 1
        point(j).derajat = point(j).derajat + 1

        'cek sirkuit
        e = 0
        For f = 1 To d
            If angka(f) = i Then
                e = 1
            End If
        Next f

        If e = 0 Then
            d = d + 1
            angka(d) = i
        End If

        e = 0

        For f = 1 To d
            If angka(f) = j Then
                e = 1
            End If
        Next f

        If e = 0 Then
            d = d + 1
            angka(d) = j
        End If

        k = k + 1
        T(k) = j

        'cek sirkuit yang terjadi
        Dim a, b As Integer
        For a = 1 To jumlah
            For b = 1 To jumlah
                If angka(a) <> -1 And angka(b) <> -1 Then
                    Matriks(angka(a), angka(b)) = -1
                End If
            Next b
        Next a

        count = count + 1
    End If
Next j
Next i

Loop

FungsiPrim = JarakTotal
simpan(ulang).dataJumlahJalur = g

End Function

```

```

Private Function fungsiPrimBiasa(point() As noktah, jumlah As Integer) As Integer
Dim i, j As Integer ' untuk perulangan for
Dim d, e, f As Integer ' untuk cek sirkuit
Dim jarak(1 To 100, 1 To 100) As Integer
Dim Matriks(1 To 100, 1 To 100) As Integer
Dim angka(100) As Integer

'set derajat = 0 & kosongkan array
For i = 1 To jumlah
    point(i).derajat = 0
Next i

Erase Matriks
Erase jarak
Erase angka
Erase JalurPB

'Hitung jarak
For i = 1 To jumlah
    For j = 1 To jumlah
        If i <> j Then
            jarak(i, j) = Sqr((point(i).posX - point(j).posX) ^ 2 + (point(i).posY - point(j).posY) ^ 2)

            'Mengcopy jarak antar titik ke variabel uji
            Matriks(i, j) = jarak(i, j)
        End If
    Next j
Next i

'variabel bantu
Dim k, count, x, y, cari, JarakTotal As Integer
Dim T(1 To 200) As Integer

JarakTotal = 0 'set jarak awal = 0

'until data jarak Prim Biasa
gPB = 0

'set angka = -1
For e = 1 To jumlah
    angka(e) = -1
Next e

d = 1
angka(d) = 1

'Sembarang titik yang digunakan adalah titik nomor 1
k = 1

'jumlah titik
count = 0

'pohon rentang minimum
T(1) = 1

'Mencari titik dengan bobot terkecil
Do Until count = jumlah - 1 'perulangan dilakukan hingga jumlah titik - 1

'variabel cari diset nilai tertinggi tipe data integer
cari = 32767

For i = 1 To k
    For j = 1 To jumlah
        If cari > Matriks(T(i), j) And Matriks(T(i), j) > 0 Then
            cari = Matriks(T(i), j)
            x = T(i)
            y = j
        End If
    Next j
Next i

```

```

'Set tidak membentuk sirkuit
For i = 1 To jumlah
    For j = 1 To jumlah
        If i = x And j = y Then
            JarakTotal = JarakTotal + jarak(i, j)

            'input nilai jalur
            gPB = gPB + 1
            JalurPB(gPB).nomor = gPB
            JalurPB(gPB).awal = i
            JalurPB(gPB).akhir = j
            JalurPB(gPB).jarak = jarak(i, j)

            'cek sirkuit
            e = 0
            For f = 1 To d
                If angka(f) = i Then
                    e = 1
                End If
            Next f

            If e = 0 Then
                d = d + 1
                angka(d) = i
            End If

            e = 0

            For f = 1 To d
                If angka(f) = j Then
                    e = 1
                End If
            Next f

            If e = 0 Then
                d = d + 1
                angka(d) = j
            End If

            k = k + 1
            T(k) = j

            'cek sirkuit yang terjadi
            Dim a, b As Integer
            For a = 1 To jumlah
                For b = 1 To jumlah
                    If angka(a) <> -1 And angka(b) <> -1 Then
                        Matriks(angka(a), angka(b)) = -1
                    End If
                Next b
            Next a

            count = count + 1
        End If
    Next j
Next i
Loop

tigaPB = gPB
fungsiPrimBiasa = JarakTotal
End Function
*****
***          Hanan Point          ***
*****
Private Sub Hanan()
Dim indeks As Integer
Dim i, j As Integer
indeks = 0

For i = 1 To jml
    For j = 1 To jml
        If i <> j Then
            indeks = indeks + 1
            HananPoint(indeks).nomor = indeks

```

```

        HananPoint(indeks).posX = terminal(i).posX
        HananPoint(indeks).posY = terminal(j).posY
    End If
Next j
Next i
End Sub

Private Sub CmdGo_Click()
Dim i, x, a As Integer
Dim Enter As String

For i = 0 To tiga
    If StrComp(combo1.Text, combo1.List(i)) = 0 Then
        Seluruh = i
    End If
Next i

If Seluruh = 0 Then
    PB = 1
    With MSFlexGrid1
        If jml > 0 Then
            .Rows = jml + 1

            For i = 1 To jml
                .TextMatrix(i, 0) = terminal(i).nomor
                .TextMatrix(i, 1) = terminal(i).nama
                .TextMatrix(i, 2) = terminal(i).posX
                .TextMatrix(i, 3) = terminal(i).posY
            Next i
        End If
    End With

    With MSFlexGrid2
        If jml > 0 Then
            .Rows = tigaPB + 1
            For i = 1 To tigaPB
                .TextMatrix(i, 0) = JalurPB(i).nomor
                .TextMatrix(i, 1) = terminal(JalurPB(i).awal).nama
                .TextMatrix(i, 2) = terminal(JalurPB(i).akhir).nama
                .TextMatrix(i, 3) = JalurPB(i).jarak * skala
            Next i
        End If
    End With

    Text1.Text = fungsiprimBiasa(terminal, jml) * skala
End If

If Seluruh > 0 Then
    PB = 0

    With MSFlexGrid1
        If jml > 0 Then
            .Rows = simpan(Seluruh).dataJumlah + 1
            For i = 1 To simpan(Seluruh).dataJumlah
                .TextMatrix(i, 0) = simpan(Seluruh).dataPrim(i).nomor
                .TextMatrix(i, 1) = simpan(Seluruh).dataPrim(i).nama
                .TextMatrix(i, 2) = simpan(Seluruh).dataPrim(i).posX
                .TextMatrix(i, 3) = simpan(Seluruh).dataPrim(i).posY
            Next i
        End If
    End With

    With MSFlexGrid2
        If jml > 0 Then
            .Rows = simpan(Seluruh).dataJumlahJalur + 1
            If jml > 0 Then
                .Rows = simpan(Seluruh).dataJumlahJalur + 1
                For i = 1 To simpan(Seluruh).dataJumlahJalur
                    .TextMatrix(i, 0) = simpan(Seluruh).dataJalur(i).nomor
                    .TextMatrix(i, 1) = simpan(Seluruh).dataJalur(i).awal).nama
                    .TextMatrix(i, 2) = simpan(Seluruh).dataJalur(i).akhir).nama
                    .TextMatrix(i, 3) = simpan(Seluruh).dataJalur(i).jarak * skala
                Next i
            End If
        End If
    End With

```

```

        Next i
    End If
End If
End With
Text1.Text = simpan(Seluruh).dataJarak * skala
End If

frmNew.Cls
frmNew.Refresh
frmNew.Form_Activate

End Sub

Private Sub cmdProses_Click()
Dim i As Integer
Dim proses As Integer

For i = 0 To 2
    If StrComp(Combo2.Text, Combo2.List(i)) = 0 Then
        proses = i
    End If
Next i

If Seluruh = 0 Then
    If proses = 0 Then
        frmResultPB.Hide
        frmResultPB.SaveHasilPB
        Unload frmResultPB
    ElseIf proses = 1 Then
        frmResultPB.Hide
        frmResultPB.PrintHasilPB
        Unload frmResultPB
    Else
        Load frmPrintPB
        Unload frmPrintPB
    End If
Else
    If proses = 0 Then
        frmResult.Hide
        frmResult.SaveHasil
        Unload frmResult
    ElseIf proses = 1 Then
        frmResult.Hide
        frmResult.PrintHasil
        Unload frmResult
    Else
        Load FrmPrint
        Unload FrmPrint
    End If
End If
End Sub

Private Sub Form_Deactivate()
'Unload Me
End Sub

Private Sub Form_Load()

Cls
Dim SteinerPoint(1 To 50) As noktah
Dim i As Integer

'set derajat = 0
For i = 1 To jml
    terminal(i).derajat = 0
Next i

'Mencari Hanan Point
Hanan

'=====
' Algoritma MRST
'=====
Dim a, b As Integer

```

```

Dim T(1 To 200) As noktah
Dim jmlGab As Integer
Dim p As Integer

jmlGab = jml

'menginputkan variabel simpan
dat = 1
simpan(dat).dataJumlah = jml
For i = 1 To jml
    simpan(dat).dataPrim(i).nomor = i
    simpan(dat).dataPrim(i).nama = terminal(i).nama
    simpan(dat).dataPrim(i).posX = terminal(i).posX
    simpan(dat).dataPrim(i).posY = terminal(i).posY
Next i

simpan(dat).dataJarak = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah, 1)
a = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah, 1)

'hitung
Dim data, nilai, s As Integer
data = 0
nilai = jml * jml - jml

Do
    For s = 1 To nilai
        simpan(dat).dataPrim(1 + simpan(dat).dataJumlah).posX = HananPoint(s).posX
        simpan(dat).dataPrim(1 + simpan(dat).dataJumlah).posY = HananPoint(s).posY
        b = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah + 1, dat)

        If a > b Then
            data = data + 1
            T(data).posX = HananPoint(s).posX
            T(data).posY = HananPoint(s).posY
            simpan(dat + 1).dataJarak = b
            a = b
        End If
    b = 0
    Next s

    If data <> 0 Then
        dat = dat + 1
        For i = 1 To simpan(dat - 1).dataJumlah
            simpan(dat).dataPrim(i).nomor = simpan(dat - 1).dataPrim(i).nomor
            simpan(dat).dataPrim(i).nama = simpan(dat - 1).dataPrim(i).nama
            simpan(dat).dataPrim(i).posX = simpan(dat - 1).dataPrim(i).posX
            simpan(dat).dataPrim(i).posY = simpan(dat - 1).dataPrim(i).posY
        Next i

        simpan(dat).dataPrim(1 + jmlGab).nomor = 1 + jmlGab
        simpan(dat).dataPrim(1 + jmlGab).nama = Chr(167) & 1 + jmlGab - jml
        simpan(dat).dataPrim(1 + jmlGab).posX = T(data).posX
        simpan(dat).dataPrim(1 + jmlGab).posY = T(data).posY
        simpan(dat).dataJumlah = jmlGab + 1

        'mengembalikan data sebelumnya
        b = FungsiPrim(simpan(dat - 1).dataPrim, simpan(dat - 1).dataJumlah, dat - 1)

        jmlGab = jmlGab + 1
        data = 0
    Else
        data = -1
    End If
Loop Until data = -1

'hapus steiner point yang kurang dari atau sama dengan 2
b = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah, dat)

Dim o As Integer
For p = simpan(1).dataJumlah + 1 To simpan(dat).dataJumlah

```

```

If simpan(dat).dataPrim(p).derajat <= 2 Then
    For o = p To simpan(dat).dataJumlah - 1
        simpan(dat).dataPrim(o).nomor = o + 1
        simpan(dat).dataPrim(o).nama = Chr(167) & o - simpan(1).dataJumlah
        simpan(dat).dataPrim(o).posX = simpan(dat).dataPrim(o + 1).posX
        simpan(dat).dataPrim(o).posY = simpan(dat).dataPrim(o + 1).posY
        simpan(dat).dataPrim(o).derajat = simpan(dat).dataPrim(o + 1).derajat
    Next o
    simpan(dat).dataJumlah = simpan(dat).dataJumlah - 1

End If
Next p

'Jumlah data diset semua data
Seluruh = dat

b = FungsiPrim(simpan(dat).dataPrim, simpan(dat).dataJumlah, dat)
simpan(dat).dataJarak = b

'jumlah jarak MRST
tiga = Seluruh

'ubah pointer mouse
frmNew.MousePointer = 0

'aktifkan menu result
frmUtama.buka_menu

'combo box GO
combol.AddItem ("Minimal Spanning Tree")
For i = 1 To tiga - 1
    combol.AddItem ("Iterasi ke: " & i)
Next i
combol.AddItem ("Final MRST")
combol.Text = combol.List(tiga)

'combo box PROSES
Combo2.AddItem ("Save Hasil")
Combo2.AddItem ("Print Hasil")
Combo2.AddItem ("Print Layout")
Combo2.Text = Combo2.List(0)

'MSFlexGrid
With MSFlexGrid1
    .Rows = 1
    .Cols = 4
    .ColWidth(0) = 500
    'menuliskan judul
    .TextMatrix(0, 0) = " No."
    .TextMatrix(0, 1) = " Nama "
    .TextMatrix(0, 2) = " Pos X "
    .TextMatrix(0, 3) = " Pos Y "

    If jml > 0 Then
        .Rows = simpan(tiga).dataJumlah + 1
        For i = 1 To simpan(tiga).dataJumlah
            .TextMatrix(i, 0) = simpan(tiga).dataPrim(i).nomor
            .TextMatrix(i, 1) = simpan(tiga).dataPrim(i).nama
            .TextMatrix(i, 2) = simpan(tiga).dataPrim(i).posX
            .TextMatrix(i, 3) = simpan(tiga).dataPrim(i).posY
        Next i
    End If
End With

With MSFlexGrid2
    .Rows = 1
    .Cols = 4
    .ColWidth(0) = 500
    'menuliskan judul
    .TextMatrix(0, 0) = " No"
    .TextMatrix(0, 1) = " Awal "
    .TextMatrix(0, 2) = " Akhir "
    .TextMatrix(0, 3) = " Jarak (" & satuan & ")"

```

```

If jml > 0 Then
    .Rows = simpan(tiga).dataJumlahJalur + 1
    For i = 1 To simpan(tiga).dataJumlahJalur
        .TextMatrix(i, 0) = simpan(tiga).dataJalur(i).nomor
        .TextMatrix(i, 1) = simpan(tiga).dataPrim(simpan(tiga).dataJalur(i).awal).nama
        .TextMatrix(i, 2) = simpan(tiga).dataPrim(simpan(tiga).dataJalur(i).akhir).nama
        .TextMatrix(i, 3) = simpan(tiga).dataJalur(i).jarak * skala
    Next i
End If
End With
Text1.Text = simpan(Seluruh).dataJarak * skala
lblSkala.Caption = "Skala 1 : " & skala
lblSatuan.Caption = satuan

'Memanggil fungsi Prim Pasa
PrimBiasa = fungsiPrimBiasa(terminal, jml)

End Sub

Private Sub lblExit_Click()
Unload Me
End Sub

```

### Form RESULT

```

Sub simpanHasil()
Dim grs1, grs2, spasi, Enter As String
Dim i, b As Integer

Enter = Chr(13) & Chr(10)
grs1 = String(22, "=")
grs2 = String(100, "-")
spasi = Space(10)

Text1.Text = Enter & spasi & Space(30) & Chr(9) & " HASIL PERHITUNGAN " & Enter &
Space(29) & spasi & grs1 & Enter & Enter

Text1.Text = Text1.Text & spasi & "Data Terminal dan Steiner Poin" & Enter
Text1.Text = Text1.Text & spasi & grs2 & Enter & spasi & "Nomor " & Chr(9) & "Nama" &
Chr(9) & Chr(9) & "Pos X" & Chr(9) & "Pos Y" & Chr(9) & "Derajat" & Enter & spasi & grs2
& Enter

For i = 1 To simpan(Seluruh).dataJumlah
    Text1.Text = Text1.Text & spasi & simpan(Seluruh).dataPrim(i).nomor & spasi & Chr(9)
    & simpan(Seluruh).dataPrim(i).nama & Chr(9) & Chr(9) & simpan(Seluruh).dataPrim(i).posX
    & Chr(9) & simpan(Seluruh).dataPrim(i).posY & Chr(9) & simpan(Seluruh).dataPrim(i).derajat & Enter
Next i

Text1.Text = Text1.Text & spasi & grs2 & Enter & Enter

'menampilkan skala
Text1.Text = Text1.Text & spasi & "Skala 1 : " & skala & Enter

' Menampilkan Rute Jalur
Text1.Text = Text1.Text & Enter & spasi & " Rute Jalur " & Enter
Text1.Text = Text1.Text & spasi & grs2 & Enter & spasi & " No. " & Chr(9) & spasi &
"Awal " & Chr(9) & Chr(9) & Chr(9) & "Akhir" & Chr(9) & Chr(9) & "Jarak (" & satuan &
")" & Enter & spasi & grs2 & Enter

For i = 1 To simpan(Seluruh).dataJumlahJalur
    Text1.Text = Text1.Text & spasi & i & Chr(9) & spasi & Chr(9) &
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).nama & Chr(9) & Chr(9) &
Chr(9) & simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).nama & Chr(9) &
Chr(9) & simpan(Seluruh).dataJalur(i).jarak * skala & Enter
Next i
Text1.Text = Text1.Text & spasi & grs2 & Enter

```

```

'Keterangan Lain
b = simpan(Seluruh).dataJumlah
Text1.Text = Text1.Text & Enter & spasi & " Jumlah Titik : " & b & Enter

b = simpan(Seluruh).dataJumlah - simpan(1).dataJumlah
Text1.Text = Text1.Text & Enter & spasi & " Jumlah Steiner Point : " & b & Enter

b = simpan(Seluruh).dataJarak
Text1.Text = Text1.Text & Enter & spasi & " Jarak Total : " & b * skala & " " & satuan

RichTextBox1.TextRTF = Text1.Text
End Sub

Sub PrintHasil()
On Error GoTo handle
CommonDialog1.Flags = cdlPDRetnDC + cdlPDPageNums

If RichTextBox1.SelLength = 0 Then
    CommonDialog1.Flags = CommonDialog1.Flags + cdlPDAllPages
Else
    CommonDialog1.Flags = CommonDialog1.Flags + cdlPDSelection
End If
CommonDialog1.ShowPrinter
RichTextBox1.SelPrint CommonDialog1.hDC
handle:
End Sub

Sub SaveHasil()
Dim namafile As String

On Error GoTo batal

With CommonDialog1
    .Filter = "(*.doc)|*.doc|"
    .DialogTitle = "simpan file"
    .ShowSave
End With

namafile = CommonDialog1.FileName
RichTextBox1.SaveFile namafile

batal:
End Sub

```

Private Sub Form\_Load()

simpanHasil

End Sub

## **Form RESULTPB**

```

Sub simpanHasilPB()
Dim grs1, grs2, spasi, Enter As String
Dim i, b As Integer
Text1.Text = ""
Enter = Chr(13) & Chr(10)
grs1 = String(22, "=")
grs2 = String(100, "-")
spasi = Space(10)

Text1.Text = Enter & spasi & Space(30) & Chr(9) & " HASIL PERHITUNGAN " & Enter &
Space(29) & spasi & grs1 & Enter & Enter

Text1.Text = Text1.Text & spasi & "Data Terminal" & Enter
Text1.Text = Text1.Text & spasi & grs2 & Enter & spasi & "Nomor " & Chr(9) & "Nama" &
Chr(9) & Chr(9) & "Pos X" & Chr(9) & "Pos Y" & Enter & spasi & grs2 & Enter

For i = 1 To jml
Text1.Text = Text1.Text & spasi & terminal(i).nomor & spasi & Chr(9) & terminal(i).nama &
Chr(9) & Chr(9) & terminal(i).posX & Chr(9) & terminal(i).posY & Enter

```

```

Next i
Text1.Text = Text1.Text & spasi & grs2 & Enter & Enter

'menampilkan skala
Text1.Text = Text1.Text & spasi & "Skala 1 : " & skala & Enter

' Menampilkan Rute Jalur
Text1.Text = Text1.Text & Enter & spasi & "Rute Jalur " & Enter
Text1.Text = Text1.Text & spasi & grs2 & Enter & spasi & "No. " & Chr(9) & spasi &
"Awai " & Chr(9) & Chr(9) & "Akhir" & Chr(9) & Chr(9) & "Jarak (" & satuan & ")" & Enter
& spasi & grs2 & Enter

For i = 1 To tigaPB
    Text1.Text = Text1.Text & spasi & i & Chr(9) & spasi &
    Text1.Text = JarakPB(i).awal.nama & Chr(9) & Chr(9) & Chr(9) & Chr(9) &
    Text1.Text = JarakPB(i).akhir.nama & Chr(9) & Chr(9) & JarakPB(i).jarak * skala & Enter
    Ax1 i
Text1.Text = Text1.Text & spasi & grs2 & Enter

'Keterangan Lain
b = jml
Text1.Text = Text1.Text & Enter & spasi & "Jumlah Titik : " & b & Enter

b = lrmRun.Text1.Text
Text1.Text = Text1.Text & Enter & spasi & "Jarak Total : " & b & " " & satuan

RichTextBox1.TextRTF = Text1.Text
End Sub

Sub PrintHasilPB()
On Error GoTo handle
CommonDialog1.Flags = cdlPDRetnDC + cdlPDPageNums
If RichTextBox1.SelLength = 0 Then
    CommonDialog1.Flags = CommonDialog1.Flags + cdlPDAllPages
Else
    CommonDialog1.Flags = CommonDialog1.Flags + cdlPDSelection
End If
CommonDialog1.ShowPrinter
RichTextBox1.SelPrint CommonDialog1.hDC
handle:
End Sub

Sub SaveHasilPB()
Dim namafile As String

On Error GoTo batal

With CommonDialog1
    .Filter = "(*.doc)|*.doc|"
    .DialogTitle = "simpan file"
    .ShowSave
End With

namafile = CommonDialog1.FileName
RichTextBox1.SaveFile namafile

batal:
End Sub

Private Sub Form_Load()
simpanHasilPB
End Sub

```

### **Form Print**

```

Private Sub Form_Load()

Dim i, selisih As Integer
Text1.Text = Now

On Error GoTo handle

```

```

Next i
Text1.Text = Text1.Text & spasi & grs2 & Enter & Enter

'menampilkan skala
Text1.Text = Text1.Text & spasi & "Skala 1 : " & skala & Enter

' Menampilkan Rute Jalur
Text1.Text = Text1.Text & Enter & spasi & " Rute Jalur " & Enter
Text1.Text = Text1.Text & spasi & grs2 & Enter & spasi & " No. " & Chr(9) & spasi &
"Awal " & Chr(9) & Chr(9) & "Akhir" & Chr(9) & Chr(9) & "Jarak (" & satuan & ")" & Enter
& spasi & grs2 & Enter

For i = 1 To tigaPB
    Text1.Text = Text1.Text & spasi & i & Chr(9) & spasi &
terminal(JalurPB(i).awal).nama & Chr(9) & Chr(9) & Chr(9) & terminal(JalurPB(i).akhir).nama & Chr(9) & Chr(9) & JalurPB(i).jarak * skala & Enter
Next i
Text1.Text = Text1.Text & spasi & grs2 & Enter

'Keterangan Lain
b = jml
Text1.Text = Text1.Text & Enter & spasi & " Jumlah Titik : " & b & Enter

b = frmRun.Text1.Text
Text1.Text = Text1.Text & Enter & spasi & " Jarak Total : " & b & " " & satuan

RichTextBox1.TextRTF = Text1.Text
End Sub

Sub PrintHasilPB()
On Error GoTo handle
CommonDialog1.Flags = cdlPDReturnDC + cdlPDPageNums
If RichTextBox1.SelLength = 0 Then
    CommonDialog1.Flags = CommonDialog1.Flags + cdlPDAllPages
Else
    CommonDialog1.Flags = CommonDialog1.Flags + cdlPDSelection
End If
CommonDialog1.ShowPrinter
RichTextBox1.SelPrint CommonDialog1.hDC
handle:
End Sub

Sub SaveHasilPB()
Dim namafile As String

On Error GoTo batal

With CommonDialog1
    .Filter = "(*.doc)|*.doc|"
    .DialogTitle = "simpan file"
    .ShowSave
End With

namafile = CommonDialog1.FileName
RichTextBox1.SaveFile namafile

batal:
End Sub

Private Sub Form_Load()
simpanHasilPB
End Sub

```

### **Form Print**

```

Private Sub Form_Load()

Dim i, selisih As Integer
Text1.Text = Now

On Error GoTo handle

```

```

'CommonDialog1.ShowPrinter

Printer.Orientation = vbPRORLandscape
Printer.ScaleMode = 6
Printer.CurrentX = 6
Printer.CurrentY = 6

Printer.FontName = "times new roman"
Printer.FontBold = True
Printer.FontSize = 12
Printer.Print Tab(10); "PRINT LAYOUT PERHITUNGAN"
Printer.Print Tab(10); "Print at : "; Text1.Text

i = 15
While i < frmNew.ScaleHeight + 15
    X1 = 15
    X2 = frmNew.ScaleWidth + 15
    Y1 = i
    Y2 = i
    Printer.DrawWidth = 4
    Printer.DrawStyle = 2
    Printer.Line (X1, Y1)-(X2, Y2), RGB(0, 255, 255)
    i = i + 5
Wend

i = 15
While i < frmNew.ScaleWidth + 15
    X1 = i
    X2 = i
    Y1 = 15
    Y2 = frmNew.ScaleHeight + 15
    Printer.DrawWidth = 4
    Printer.DrawStyle = 2
    Printer.Line (X1, Y1)-(X2, Y2), RGB(0, 255, 255)
    i = i + 5
Wend

Printer.FontName = "MS Sans Serif"
Printer.FontBold = False
Printer.FontSize = 12

If jml <> 0 Then
    For i = 1 To jml
        Printer.DrawWidth = 5
        Printer.Circle (terminal(i).posX + 15, terminal(i).posY + 15), 0.4, RGB(255, 0,
0)
        Printer.Print terminal(i).nama
    Next i
End If

If simpan(1).dataJalur(1).nomor > 0 And PB = 0 Then
    For i = 1 To simpan(Seluruh).dataJumlahJalur
        Printer.DrawWidth = 5
        Printer.DrawStyle = 0
        Printer.Line (simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posX +
15, simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posY + 15)-
(simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posX + 15,
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posY + 15), , B
        Printer.Line (simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posX +
15, simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).awal).posY + 15)-
(simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posX + 15,
simpan(Seluruh).dataPrim(simpan(Seluruh).dataJalur(i).akhir).posY + 15), , B
        Next i
    End If

If Seluruh <> 0 Then
    For i = simpan(1).dataJumlah + 1 To simpan(Seluruh).dataJumlah
        Printer.DrawWidth = 5
        Printer.Circle (simpan(Seluruh).dataPrim(i).posX + 15, 0.4, vbYellow
simpan(Seluruh).dataPrim(i).posY + 15), 0.4, vbYellow
        Printer.Print simpan(Seluruh).dataPrim(i).nama
    Next i
End If

```

```

        Printer.Print terminal(i).nama
    Next i
End If

For i = 1 To tigaPB
    Printer.Line      (terminal(JalurPB(i).awal).posX      +      15,
terminal(JalurPB(i).awal).posY      +      15)-(terminal(JalurPB(i).akhir).posX      +      15,
terminal(JalurPB(i).akhir).posY + 15), vbMagenta
    Next i

Printer.EndDoc

Exit Sub

handle:
    Unload Me
End Sub

```

### **Form ABOUT**

```

Private Sub CommandButton1_Click()
    Unload Me
    frmUtama.Enabled = True
    frmUtama.Show
End Sub

```



```

If PB = 1 And JalurPB(1).nomor > 0 Then
    For i = 1 To tigaPB
        Printer.Line (terminal(JalurPB(i).awal).posX + 15,
terminal(JalurPB(i).awal).posY + 15)-(terminal(JalurPB(i).akhir).posX + 15,
terminal(JalurPB(i).akhir).posY + 15), vbMagenta
    Next i
End If

Printer.EndDoc

Exit Sub

handle:
    Unload Me
End Sub

Private Sub Form_Unload(Cancel As Integer)
frmUtama.buka_menu
End Sub

Form PrintPB

Private Sub Form_Load()
Dim i, selisih As Integer
Text1.Text = Now

On Error GoTo handle
'CommonDialog1.ShowPrinter

Printer.Orientation = vbPRORLandscape
Printer.ScaleMode = 6
Printer.CurrentX = 6
Printer.CurrentY = 6

Printer.FontName = "times new roman"
Printer.FontBold = True
Printer.FontSize = 12
Printer.Print Tab(10); "PRINT LAYOUT JARAK PRIM"
Printer.Print Tab(10); "Print at : "; Text1.Text

i = 15
While i < frmNew.ScaleHeight + 15
    X1 = 15
    X2 = frmNew.ScaleWidth + 15
    Y1 = i
    Y2 = i
    Printer.DrawWidth = 4
    Printer.DrawStyle = 2
    Printer.Line (X1, Y1)-(X2, Y2), RGB(0, 255, 255)
    i = i + 5
Wend

i = 15
While i < frmNew.ScaleWidth + 15
    X1 = i
    X2 = i
    Y1 = 15
    Y2 = frmNew.ScaleHeight + 15
    Printer.DrawWidth = 4
    Printer.DrawStyle = 2
    Printer.Line (X1, Y1)-(X2, Y2), RGB(0, 255, 255)
    i = i + 5
Wend

Printer.FontName = "MS Sans Serif"
Printer.FontBold = False
Printer.FontSize = 12

If jml <> 0 Then
    For i = 1 To jml
        Printer.DrawWidth = 5
        Printer.Circle (terminal(i).posX + 15, terminal(i).posY + 15), 0.4, RGB(255, 0,
0)
    Next i
End If

```