



**OPTIMASI JARINGAN LISTRIK DENGAN
ALGORITMA *PRIM* DAN APLIKASI PROGRAM
MATLAB
(STUDI KASUS PLN KOTA PEKALONGAN)**

skripsi
disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Program Studi Matematika

PERPUSTAKAAN
UNNES

oleh
Harni Ulfiana
4150404500

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG**

2009

PERNYATAAN KEASLIAN TULISAN

Dengan ini saya menyatakan bahwa isi skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya tidak terdapat karya yang diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam skripsi ini dan disebutkan dalam daftar pustaka.

Semarang, Februari 2009

Harni Ulfiana
NIM. 4150404500



PENGESAHAN

Skripsi ini telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA
UNNES pada tanggal

Panitia

Ketua

Sekretaris

Drs. Imam Kasmadi S., M.S
130781011

Drs. Edy Soedjoko, M.Pd
131693657

Penguji

Dr. Dwijanto, M.S
131404323

Penguji/Pembimbing I

Penguji/Pembimbing II

Mulyono, S.Si, M.Si
132158717

Endang Sugiharti, S.Si, M.Kom
132231407

ABSTRAK

Ulfiana, Harni. 2009. *Optimasi Jaringan Listrik dengan Algoritma Prim dan Program MATLAB (Studi Kasus PLN Kota Pekalongan)*. Skripsi, Jurusan Matematika FMIPA Unnes. Mulyono, S.Si, M.Si dan Endang Sugiharti, S.Si, M.Kom.

Kata Kunci: Pohon Rentang Minimum, Algoritma Prim, Program MATLAB

PLN Kota Pekalongan memerlukan perencanaan yang tepat dalam membangun jaringan distribusi listrik di wilayah Kota Pekalongan agar biaya yang digunakan seminimal mungkin. Dengan meminimumkan panjang kabel terpasang maka dapat meminimumkan biaya dalam pembangunan jaringan distribusi listrik. Untuk itu akan dicari pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan yang menghasilkan total panjang kabel terpasang minimum.

Dalam penelitian ini, yang menjadi permasalahan adalah mencari pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan Program MATLAB serta membandingkan hasil pohon rentang minimum yang diperoleh dengan algoritma Prim dan *binary integer programming* dengan bantuan program MATLAB. Tujuan dari penelitian ini adalah untuk mengetahui pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binary integer programming* dengan bantuan program MATLAB serta untuk mengetahui perbandingan hasil pohon rentang minimum yang diperoleh dengan algoritma Prim dan *binary integer programming* dengan bantuan program MATLAB.

Metode yang dilakukan dalam penelitian ini adalah menemukan masalah, merumuskan masalah, pengambilan data, analisis, dan pemecahan masalah serta penarikan kesimpulan.

Berdasarkan data-data yang diperoleh yaitu peta jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 dan data panjang kabel yang digunakan di wilayah tersebut kemudian dibuat gambar jaringan dan dicari pohon rentang minimum dari gambar jaringan tersebut. Pohon rentang minimum yang diperoleh baik dengan algoritma Prim maupun dengan *binary integer programming* dengan bantuan program MATLAB mempunyai panjang yang sama yaitu 11.312 m dan sisi-sisi yang terpilih dalam pohon rentang minimum yang diperoleh dengan algoritma Prim dan *binary integer programming* dengan bantuan program MATLAB juga sama. Jadi algoritma Prim dan *binary integer programming* dengan bantuan program MATLAB menghasilkan nilai optimum yang sama. Dengan demikian, pencarian pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dapat menghemat penggunaan kabel sepanjang 5.862 m dari total panjang kabel sebelumnya 17.174 m.

Pencarian pohon rentang minimum dalam penelitian ini dilakukan dengan bantuan program MATLAB. Namun demikian metode yang digunakan masih semi manual, di mana output yang dihasilkan tidak langsung berupa gambar pohon rentang minimum. Untuk itu diharapkan pada penelitian-penelitian

selanjutnya yang mengkaji tentang pencarian pohon rentang minimum dapat membuat program yang secara langsung memberikan output berupa gambar pohon rentang minimumnya dan menggunakan *software* komputer yang lain sehingga dapat menambah pengetahuan tentang *software-software* komputer yang dapat digunakan untuk mencari pohon rentang minimum. Dalam penelitian ini diasumsikan bahwa besarnya tegangan kabel adalah seragam, padahal dalam kenyataannya besarnya tegangan tidak selalu seragam. Untuk itu diharapkan pada penelitian-penelitian selanjutnya yang mengkaji tentang jaringan distribusi listrik dapat mempertimbangkan besarnya tegangan listrik, karena pendistribusian listrik untuk wilayah perumahan dan wilayah industri menggunakan tegangan yang berbeda.



MOTTO DAN PERSEMBAHAN

Motto

Jangan bersedih atas pekerjaan yang belum sempat engkau selesaikan, sebab pekerjaan orang-orang besar itu tidak pernah ada ujungnya

Persembahan

Skripsi ini saya persembahkan untuk

- 1. Bapak dan Ibu tercinta yang senantiasa mendoakanku*
- 2. Adik-adikku tersayang yang selalu menyemangati*
- 3. Sahabat dan teman-temanku semua yang selalu membantu dan mendukungku*

PRAKATA

Syukur Alhamdulillah penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Optimasi Jaringan Listrik dengan Algoritma Prim dan Aplikasi Program MATLAB (Studi Kasus PLN Kota Pekalongan)” ini dengan baik.

Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Sains pada Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang.

Dalam kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam menyelesaikan skripsi ini. Ucapan terima kasih disampaikan kepada:

1. Prof. Dr. H. Sudijono Sastroatmodjo, M.Si, Rektor Universitas Negeri Semarang,
2. Drs. Kasmadi Imam S., M.S, Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang,
3. Drs. Edy Soedjoko, M.Pd, Ketua Jurusan Matematika Universitas Negeri Semarang,
4. Mulyono, S.Si, M.Si, Pembimbing Utama yang telah memberikan bimbingan dan pengarahan kepada penulis,
5. Endang Sugiharti, S.Si, M.Kom, Pembimbing Pendamping yang telah memberikan bimbingan dan pengarahan kepada penulis,

6. Pimpinan PLN Kota Pekalongan yang telah memberikan ijin kepada penulis untuk melaksanakan penelitian ini,
7. Kepala UPJ PLN Kota Pekalongan yang telah memberikan pengarahan kepada penulis,
8. Semua pihak yang telah membantu dalam menyelesaikan skripsi ini.

Harapan penulis semoga skripsi ini dapat berguna bagi semua pihak.

Semarang, Februari 2009

Penulis



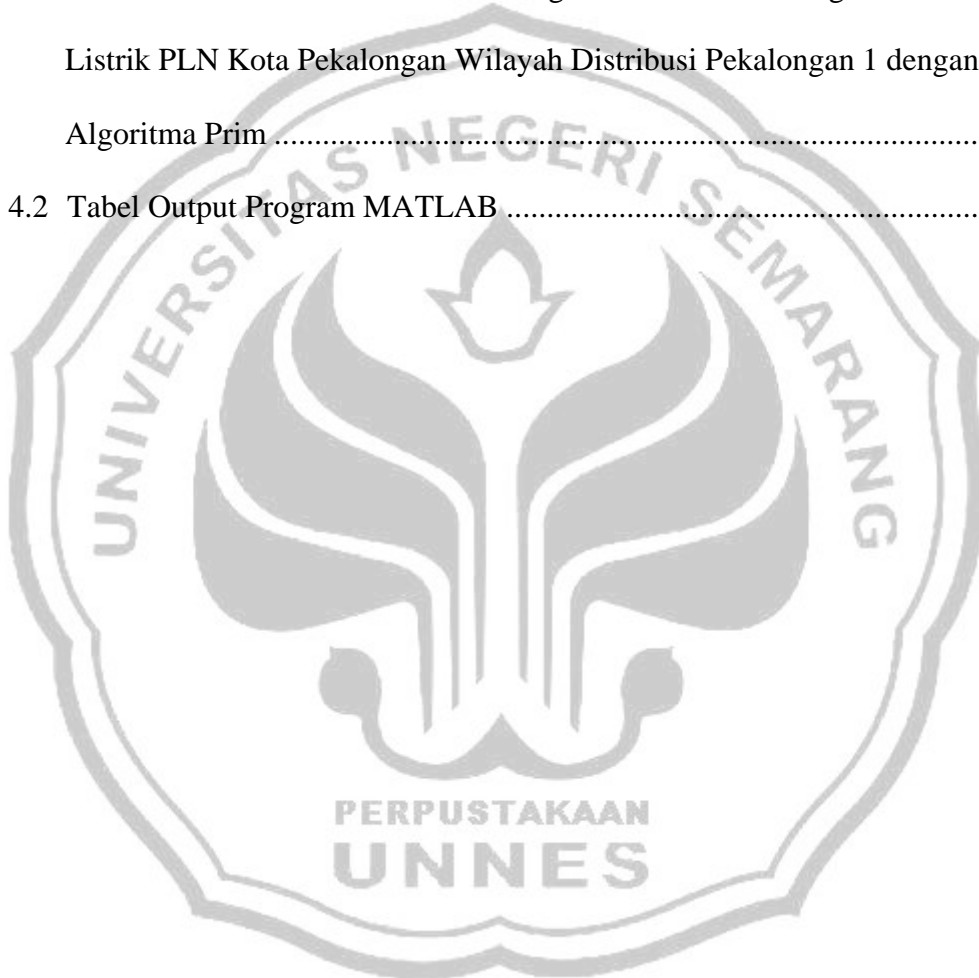
DAFTAR ISI

	Halaman
Judul	i
Pernyataan Keaslian Tulisan	ii
Pengesahan	iii
Abstrak	iv
Motto dan Persembahan	vi
Prakata	vii
Daftar Isi	ix
Daftar Tabel	xi
Daftar Gambar	xii
Daftar Lampiran	xiii
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Skripsi	5
 BAB II LANDASAN TEORI	 7
2.1 Riset Operasi	7

2.2 Program Linear	11
2.3 Program 0-1 (<i>Binary Integer Programming</i>).....	15
2.4 Analisis Jaringan.....	15
2.5 Graf (<i>Graph</i>).....	18
2.6 Pohon Rentang Minimum.....	20
2.7 Algoritma Prim	22
2.8 MATLAB	25
2.9 Aplikasi MATLAB.....	27
BAB III METODE PENELITIAN	34
3.1 Menemukan Masalah.....	34
3.2 Merumuskan Masalah.....	34
3.3 Pengambilan Data	35
3.4 Analisis dan Pemecahan Masalah.....	35
3.5 Penarikan Kesimpulan.....	35
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	36
4.1 Hasil Penelitian.....	36
4.2 Pembahasan	48
BAB V PENUTUP.....	52
5.1 Simpulan.....	52
5.2 Saran	53
Daftar Pustaka	54
Lampiran	55

DAFTAR TABEL

Tabel	Halaman
2.1 Tabel Iterasi	24
4.1 Tabel Iterasi Pencarian Pohon Rantang Minimum dari Jaringan Distribusi Listrik PLN Kota Pekalongan Wilayah Distribusi Pekalongan 1 dengan Algoritma Prim	38
4.2 Tabel Output Program MATLAB	47



DAFTAR GAMBAR

Gambar	Halaman
2.1 Contoh Jaringan	16
2.2 Pola Jalan dan Tempat-Tempat Persinggahan di Kawasan Alam Hijau	17
2.3 Graf G	18
2.4 Pohon Rentang Minimum yang Diperoleh dengan Algoritma Prim	25
2.5 Desktop MATLAB	26
2.6 Pohon Rentang Minimum yang Diperoleh dengan Program MATLAB	33



DAFTAR LAMPIRAN

Lampiran	Halaman
1. Data Hasil Penelitian	55
2. Gambar Jaringan Distribusi Listrik PLN Kota Pekalongan Wilayah Distribusi Pekalongan 1	58
3. Hasil Iterasi Pencarian Pohon Rentang Minimum dari Jaringan Distribusi Listrik PLN Kota Pekalongan Wilayah Distribusi Pekalongan 1 dengan Algoritma Prim	59
4. Gambar Pohon Rentang Minimum dari Jaringan Distribusi Listrik PLN Kota Pekalongan Wilayah Distribusi Pekalongan 1	62
5. Input Program MATLAB	63
6. Output Program MATLAB.....	69

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan zaman dan teknologi yang semakin canggih, hampir setiap kebutuhan dalam ilmu pengetahuan dan teknologi membutuhkan peranan matematika. Tidak dapat dipungkiri bahwa matematika telah menjadi elemen dasar bagi perkembangan ilmu pengetahuan dan teknologi. Sehingga hampir dipastikan bahwa setiap bagian dari ilmu pengetahuan dan teknologi baik dalam unsur kajian umum, ilmu murni, maupun terapannya memerlukan peranan ilmu matematika sebagai ilmu bantunya.

Riset operasi adalah bagian dari aplikasi matematika untuk memecahkan masalah optimasi. Banyak model riset operasi yang sudah dikembangkan yang berhubungan dengan matematika. Salah satunya adalah program linear. Program linear merupakan model dari riset operasi yang banyak digunakan dalam bidang industri, transportasi, perdagangan, ekonomi dan berbagai bidang lainnya.

Selain itu, model lain yang juga banyak dikembangkan adalah model analisis jaringan. Jaringan merupakan suatu istilah yang sudah dikenal luas dalam kehidupan sehari-hari. Jaringan kerja muncul pada sejumlah perencanaan dan dalam berbagai bidang. Jaringan transportasi, listrik, dan komunikasi merupakan sesuatu yang kita jumpai sehari-hari. Pengembangan jaringan juga digunakan secara luas untuk masalah-masalah seperti produksi, distribusi, perencanaan

proyek, penempatan fasilitas, manajemen sumberdaya, perencanaan keuangan dan lain sebagainya.

Persoalan jaringan dapat dibagi menjadi 3 (tiga) macam, yaitu: (a) persoalan rute terpendek (*shortest route*); (b) persoalan minimasi jaringan atau pohon rentang minimum (*minimum spanning tree*); dan (c) persoalan aliran maksimum (*maximal flow*). Persoalan jaringan yang akan dibahas dalam skripsi ini adalah persoalan pohon rentang minimum, yaitu bagaimana menentukan pengadaan busur-busur yang menghubungkan semua *node* yang ada pada jaringan, sehingga diperoleh total panjang busur yang minimum. Persoalan pohon rentang minimum memiliki sejumlah penerapan praktis yang penting. Sebagai contoh, dalam perencanaan jaringan transportasi, jaringan komunikasi, jaringan listrik, atau jaringan distribusi.

Aplikasi pohon rentang minimum dalam skripsi ini adalah pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan untuk memperoleh total panjang kabel yang digunakan minimum. Dengan meminimumkan panjang kabel yang digunakan maka dapat meminimumkan biaya dalam pembangunan jaringan distribusi listrik di wilayah Kota Pekalongan. Untuk itu diperlukan perencanaan yang tepat dalam membuat jaringan listrik agar biaya yang digunakan seminimal mungkin.

Terdapat 2 (dua) algoritma untuk mendapatkan pohon rentang minimum yaitu algoritma Prim dan algoritma Kruskal, di sini penulis akan menggunakan algoritma Prim. Selain algoritma Prim, penulis juga menggunakan *software* MATLAB.

Penggunaan *software* dalam menyelesaikan masalah optimasi sangatlah penting. Terutama bila melibatkan banyak iterasi dalam menemukan solusi optimum dari suatu permasalahan. MATLAB termasuk salah satu *software* yang banyak digunakan untuk menyelesaikan masalah optimasi. Program MATLAB yang digunakan merupakan implementasi dari *binary integer programming* yang merupakan salah satu model matematika untuk menyelesaikan masalah pencarian pohon rentang minimum. Dengan menggunakan *binary integer programming* maka dihasilkan variabel keputusan yang bernilai 0 atau 1. Sehingga penggunaan *bonnary integer programming* dengan bantuan program MATLAB diharapkan dapat membantu menyelesaikan permasalahan dalam skripsi ini yaitu menentukan pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan lebih efisien.

1.2. Rumusan Masalah

Dari uraian di atas diperoleh rumusan masalah dalam skripsi ini adalah sebagai berikut.

1. Bagaimana hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binaru integer programming* menggunakan bantuan program MATLAB?
2. Bagaimana perbandingan hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binaru integer programming* menggunakan bantuan program MATLAB?

1.3. Pembatasan Masalah

Dalam skripsi ini dibahas mengenai pencarian pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan di wilayah distribusi Pekalongan 1 Kota Pekalongan. Dipilih wilayah distribusi Pekalongan 1 karena wilayah tersebut merupakan wilayah perumahan, di mana kebanyakan dari pelanggannya menggunakan listrik hanya untuk kebutuhan rumah tangga saja. Dalam skripsi ini diasumsikan bahwa besarnya tegangan di wilayah distribusi Pekalongan 1 adalah seragam. Pencarian pohon rentang minimum di sini dilakukan dengan menggunakan algoritma Prim dan aplikasi program MATLAB. Program MATLAB yang digunakan merupakan implementasi dari *binary integer programming* dengan menggunakan fungsi `bintprog` untuk mengetahui sisi-sisi yang terpilih dalam pohon rentang minimum dan panjang pohon rentang minimum.

1.4. Tujuan

Tujuan yang ingin dicapai dari penyusunan skripsi ini adalah sebagai berikut.

1. Untuk mengetahui hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binaru integer programming* menggunakan bantuan program MATLAB.
2. Untuk mengetahui perbandingan hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binaru integer programming* menggunakan bantuan program MATLAB.

1.5. Manfaat

Manfaat dari penyusunan skripsi ini adalah sebagai berikut.

1. Diharapkan dapat menambah wawasan dan pengetahuan tentang masalah-masalah dalam suatu jaringan.
2. Diharapkan dapat menjadi referensi dalam menerapkan algoritma Prim dan MATLAB untuk penyelesaian masalah pohon rentang minimum.

1.6. Sistematika Skripsi

Penulisan skripsi ini secara garis besar dibagi menjadi 3 (tiga) bagian yaitu sebagai berikut.

1. Bagian Awal Skripsi

Bagian ini berisi halaman judul, abstrak, halaman pengesahan, halaman motto dan persembahan, kata pengantar, daftar isi, daftar tabel, daftar gambar, dan daftar lampiran.

2. Bagian Isi Skripsi

Bagian ini berisi 5 bab, yaitu:

- a. Bab I Pendahuluan

Bab ini menguraikan tentang gambaran umum skripsi yaitu latar belakang, rumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika skripsi.

- b. Bab II Landasan Teori

Bab ini menguraikan tentang definisi maupun pemikiran-pemikiran yang dijadikan kerangka teoritis dalam mendasari pemecahan dari permasalahan

dalam skripsi ini yaitu riset operasi, program linear, analisis jaringan, pohon rentang minimum, algoritma Prim, program MATLAB, dan aplikasi MATLAB.

c. Bab III Metode Penelitian

Bab ini menguraikan langkah-langkah kerja yang akan ditempuh, meliputi menemukan masalah, merumuskan masalah, pengambilan data, analisis, dan pemecahan masalah serta penarikan simpulan.

d. Bab IV Hasil Penelitian dan Pembahasan

Bab ini menguraikan hasil penelitian dan pembahasan dari permasalahan yang dikemukakan. Pada hasil penelitian berisi hasil pencarian pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan menggunakan algoritma Prim dan *binaru integer programming* dengan bantuan program MATLAB. Pada pembahasan berisi analisis hasil pencarian yang kemudian dibandingkan antara hasil pohon rentang minimum yang diperoleh dengan algoritma Prim dan *binaru integer programming* menggunakan bantuan program MATLAB.

e. Bab V Penutup

Bab ini mengemukakan simpulan dan saran.

3. Bagian Akhir Skripsi

Bagian ini berisi daftar pustaka dan lampiran.

BAB II

LANDASAN TEORI

2.1. Riset Operasi

2.1.1. Asal-Usul Riset Operasi

Awal dari kegiatan yang dinamakan riset operasi umumnya bersumber dari jasa-jasa militer pada awal Perang Dunia II. Pada masa perang ada kebutuhan mendesak untuk mengalokasikan sumber-sumber daya yang terbatas kepada berbagai operasi militer dan kepada kegiatan-kegiatan dalam setiap operasi dengan cara yang efektif. Oleh karena itu, pimpinan militer Inggris dan kemudian juga Amerika Serikat meminta sejumlah ahli sains untuk menerapkan pendekatan ilmiah untuk menghadapi masalah ini dan masalah-masalah strategis dan taktis yang lain. Mereka diminta untuk melakukan riset mengenai operasi militer. Tim-tim ahli sains inilah merupakan tim-tim riset operasi yang pertama.

Didorong oleh keberhasilan riset operasi dalam militer, pihak industri lambat laun tertarik dalam bidang ini. Dengan meningkatnya kemajuan industri setelah perang, masalah-masalah yang disebabkan oleh meningkatnya kerumitan dan spesialisasi organisasi mulai mencuat kembali. Semakin banyak orang menyadari, bahwa masalah-masalah ini pada dasarnya sama dengan yang dihadapi pihak militer pada saat itu. Dengan demikian riset operasi mulai memasuki bidang industri, bisnis, dan pemerintahan sipil. Menjelang tahun 1959, riset operasi sudah masuk di Inggris dan sedang dalam proses masuk di Amerika Serikat. Sejak saat itu bidang ilmu ini berkembang sangat cepat.

2.1.2. Sifat-Sifat Riset Operasi

Riset operasi meliputi "riset mengenai operasi". Nama ini menyatakan sesuatu mengenai pendekatan dan bidang aplikasi dari bidang ini. Maka, riset operasi diterapkan kepada masalah-masalah mengenai bagaimana melaksanakan dan mengkoordinasikan operasi atau kegiatan-kegiatan dalam suatu organisasi (Hillier, 1990: 4).

Secara khusus, proses ini dimulai dengan mengamati dan merumuskan masalah dan kemudian membangun suatu model ilmiah yang berusaha untuk mengabstraksikan inti dari persoalan yang sebenarnya. Riset operasi juga berkaitan dengan manajemen organisasi secara praktis. Oleh karena itu, agar berhasil baik riset operasi harus memberikan kesimpulan-kesimpulan yang positif dan dapat dimengerti bagi para pengambil keputusan bila perlu.

Ciri lain dari riset operasi adalah pandangannya yang luas dan berupaya untuk mendapatkan penyelesaian yang terbaik atau optimum bagi masalah yang dihadapi.

2.1.3. Tahap-Tahap dalam Riset Operasi

Pembentukan model yang cocok hanyalah salah satu tahap dari aplikasi riset operasi. Pola dasar penerapan riset operasi terhadap suatu masalah dapat dipisahkan menjadi beberapa tahap, yaitu sebagai berikut.

1. Merumuskan masalah

Sebelum solusi terhadap suatu persoalan dipikirkan, pertama kali suatu definisi persoalan yang tepat harus dirumuskan. Sering dilaporkan oleh organisasi-organisasi bahwa kegagalan dalam penyelesaian masalah

diakibatkan karena kesalahan mendefinisikan persoalan. Dalam perumusan masalah ada tiga pertanyaan penting yang harus dijawab:

- a. variabel keputusan yaitu unsur-unsur dalam persoalan yang dapat dikendalikan oleh pengambil keputusan, sering disebut sebagai instrumen.
- b. tujuan (*objective*), penetapan tujuan membantu pengambil keputusan memusatkan perhatian pada persoalan dan pengaruhnya terhadap organisasi, tujuan diekspresikan dalam variabel keputusan.
- c. kendala (*constrain*) adalah pembatas-pembatas terhadap alternatif tindakan yang tersedia.

2. Pembentukan model

Sesuai dengan definisi persoalannya, pengambil keputusan menentukan model yang paling cocok untuk mewakili sistem. Model merupakan ekspresi kuantitatif dari tujuan dan kendala-kendala persoalan dalam variabel keputusan. Jika model yang dihasilkan cocok dengan salah satu model matematika yang biasa (misalnya linear), maka solusinya dapat dengan mudah diperoleh dengan program linear. Jika hubungan model matematika begitu rumit untuk penerapan solusi analitik, maka suatu model probabilitas mungkin lebih cocok. Beberapa kasus membutuhkan penggunaan kombinasi model matematik dan probabilitas. Ini tentu saja tergantung pada sifat-sifat dan kerumitan sistem yang dipelajari. Model lain yang dapat digunakan adalah model jaringan.

3. Mencari penyelesaian masalah

Pada tahap ini bermacam-macam teknik dan metode solusi kuantitatif yang merupakan bagian utama dari riset operasi memasuki proses. Penyelesaian masalah sesungguhnya merupakan aplikasi satu atau lebih teknik-teknik ini terhadap model. Seringkali, solusi terhadap model berarti nilai-nilai variabel keputusan yang mengoptimalkan salah satu fungsi tujuan dengan nilai fungsi tujuan lain yang diterima. Di samping solusi model, perlu juga mendapat informasi tambahan mengenai tingkah laku solusi yang disebabkan karena perubahan parameter sistem. Ini biasanya dinamakan sebagai analisis sensitivitas. Analisis ini terutama diperlukan jika parameter sistem tak dapat diduga secara tepat.

4. Validasi model

Asumsi-asumsi yang digunakan dalam pembentukan model harus valid (absah). Dengan kata lain, model harus diperiksa apakah ia mencerminkan berjalannya sistem yang diwakili. Suatu metode yang biasa digunakan untuk menguji validitas model adalah membandingkan *performance*-nya dengan data masa lalu yang tersedia. Model dikatakan valid jika dengan kondisi input yang serupa, ia dapat menghasilkan kembali *performance* seperti masa lampau. Masalahnya adalah bahwa tak ada yang menjamin *performance* masa depan akan berlanjut meniru cerita lama.

5. Penerapan hasil akhir

Tahap terakhir adalah menerapkan hasil model yang telah diuji. Hal ini membutuhkan suatu penjelasan yang hati-hati tentang solusi yang digunakan

dan hubungannya dengan realitas. Suatu tahap kritis pada tahap ini adalah mempertemukan ahli riset operasi (pembentuk model) dengan mereka yang bertanggung jawab terhadap pelaksanaan sistem.

(Mulyono, 2002: 7-8)

2.2. Program Linear

Program linear merupakan suatu model dari riset operasi yang merupakan bentuk khusus dari permasalahan optimasi. Permasalahan optimasi meliputi pemaksimalan atau meminimuman suatu fungsi tujuan yang dibatasi oleh berbagai kendala keterbatasan sumber daya dan kendala persyaratan-persyaratan tertentu yang harus dipenuhi. Contoh untuk permasalahan yang memaksimalkan adalah masalah keuntungan sedangkan contoh untuk permasalahan meminimumkan adalah masalah biaya, sediaan, dan lain-lain. Kendala-kendala yang sering dijumpai adalah keterbatasan bahan mentah, tenaga kerja, dan sebagainya, yang dapat diekspresikan dalam bentuk sejumlah persamaan atau pertidaksamaan linear dalam variabel atau peubahnya.

Program linear adalah suatu cara untuk menyelesaikan persoalan alokasi sumber-sumber yang terbatas diantara beberapa aktivitas yang bersaing, dengan cara yang terbaik yang mungkin dilakukan (Dimyati dan Dimyati, 1999:17). Program linear memakai suatu model matematis untuk menggambarkan masalah yang dihadapi. Kata "linear" berarti semua fungsi matematis dalam model ini harus fungsi-fungsi linear. Kata "pemrograman" merupakan sinonim untuk kata "perencanaan". Maka, program linear adalah perencanaan kegiatan-kegiatan untuk

memperoleh hasil yang optimal, yaitu suatu hasil yang mencapai tujuan yang ditentukan dengan cara yang paling baik (sesuai model matematis) di antara semua alternatif yang mungkin (Hillier, 1990:27).

Menurut Suyitno (1999:2), pemecahan masalah program linear melalui tahap-tahap sebagai berikut.

1. Memahami masalah di bidang yang bersangkutan.
2. Menyusun model matematika.
3. Menyelesaikan model matematika (mencari jawaban model).
4. Menafsirkan jawaban model menjadi jawaban atas masalah yang nyata.

Karakteristik-karakteristik yang biasanya digunakan dalam persoalan program linear adalah sebagai berikut.

1. Variabel keputusan, adalah variabel yang menguraikan secara lengkap keputusan-keputusan yang akan dibuat atau berarti pula sebagai kumpulan variabel yang akan dicari untuk ditentukan nilainya.
2. Fungsi tujuan, merupakan fungsi dari variabel keputusan yang akan dioptimumkan. Fungsi tujuan merupakan pernyataan matematika yang menyatakan hubungan Z (nilai fungsi tujuan) dengan jumlah dari perkalian semua koefisien fungsi tujuan.
3. Pembatas, merupakan kendala yang dihadapi sehingga kita tidak bisa menentukan harga-harga variabel keputusan secara sembarang. Koefisien dari variabel keputusan pada pembatas disebut koefisien teknologis, sedangkan bilangan yang ada di sisi kanan setiap pembatas disebut ruas kanan pembatas.

4. Pembatas tanda, adalah pembatas yang menjelaskan apakah variabel keputusannya diasumsikan hanya berharga nonnegatif atau variabel keputusan tersebut boleh berharga positif atau negatif (tidak terbatas dalam tanda).

Tidak semua masalah optimasi dapat diselesaikan dengan metode program linear. Beberapa prinsip utama yang mendasari penggunaan metode program linear adalah sebagai berikut.

1. Adanya sasaran, sasaran dalam model matematika masalah program linear berupa fungsi tujuan yang akan dicari nilai optimumnya (maksimum/minimum).
2. Adanya keterbatasan sumber daya, sumber daya dapat berupa waktu, tenaga kerja, biaya, bahan, dan sebagainya; sumber daya yang terbatas disebut kendala atau pembatas.
3. Masalah harus dapat dituangkan dalam bahasa matematika yang disebut model matematika yang memuat fungsi tujuan dan fungsi kendala; fungsi tujuan harus berupa fungsi linear dan fungsi kendala berupa pertidaksamaan atau persamaan linear.
4. Antar variabel yang membentuk fungsi tujuan dan kendala ada keterkaitan, artinya perubahan pada satu variabel akan mempengaruhi nilai variabel yang lain.

Petunjuk untuk menyusun model matematika adalah sebagai berikut.

1. Menentukan tipe dari masalah (maksimasi atau minimasi).

2. Mendefinisian variabel keputusan; koefisien kontribusi digunakan untuk menentukan tipe masalah dan untuk membantu mengidentifikasi variabel keputusan.
3. Merumuskan fungsi tujuan; setelah menentukan tipe masalah dan variabel keputusan selanjutnya mengkombinasikan informasi ke rumusan fungsi tujuan.
4. Merumuskan kendala; tahap ini lebih merupakan seni dari pada ilmu pengetahuan, ada dua pendekatan dasar, yaitu:
 - a. Pendekatan ruas kanan, merupakan besar maksimum dari sumber daya yang tersedia dalam masalah maksimum maupun minimum dari sumber daya yang tersedia dalam masalah minimum.
 - b. Pendekatan ruas kiri, merupakan koefisien teknis dari daftar dalam tabel atau baris-baris, meletakkan semua nilai sebagai koefisien teknis dan daftarnya dalam baris dan kolom.

Model matematika dalam program linear dirumuskan sebagai berikut.

Fungsi tujuan

$$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \text{ dengan } c_1, c_2, \dots, c_n \text{ adalah konstanta}$$

Harus memenuhi fungsi kendala

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$$

atau

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$$

atau

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \text{ dengan } i = 1, 2, 3, \dots, m$$

2.3. Program 0-1 (*Binary Integer Programming*)

Penyelesaian sebuah kasus program linear mungkin menghasilkan nilai optimum variabel-variabel keputusan yang berupa bilangan pecahan. Model penyelesaian matematis yang memungkinkan hasil penyelesaian kasus program linear yang berupa bilangan pecahan diubah menjadi bilangan bulat tanpa meninggalkan optimalitas penyelesaian disebut program bilangan bulat (*integer programming*) (Siswanto, 2007: 217). Program bilangan bulat mencakup 2 (dua) teknik analisis, yaitu (1) teknik analisis untuk menghasilkan penyelesaian optimum berupa bilangan bulat dan (2) teknik analisis untuk menghasilkan penyelesaian optimum berupa bilangan biner (0 dan 1). Teknik yang kedua sering disebut sebagai program 0-1 (*binary integer programming*).

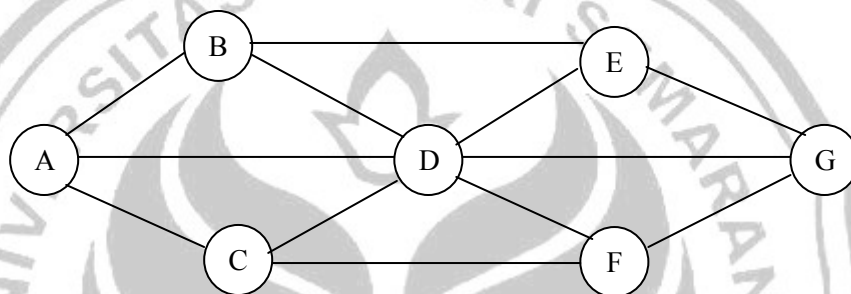
Sesuai dengan namanya, program 0-1 menghasilkan variabel-variabel keputusan yang bernilai 0 dan 1. Penggunaan bilangan biner 0-1 sebagai variabel keputusan membuat model program linear dapat digunakan untuk menyelesaikan soal-soal yang lebih bervariasi.

2.4. Analisis Jaringan

Jaringan merupakan suatu istilah yang sudah dikenal luas dalam kehidupan sehari-hari. Jaringan kerja muncul pada sejumlah perencanaan dan dalam berbagai bidang. Jaringan transportasi, listrik, dan komunikasi merupakan sesuatu yang kita jumpai sehari-hari. Pengembangan jaringan juga digunakan secara luas untuk masalah-masalah seperti produksi, distribusi, perencanaan

proyek, penempatan fasilitas, manajemen sumberdaya, perencanaan keuangan, dan lain sebagainya.

Suatu jaringan kerja terdiri atas suatu gugus titik dan garis yang menghubungkan pasangan titik tertentu. Titik-titik tersebut dinamakan simpul (*node*) dan digambarkan dengan lingkaran, sedangkan garis-garis yang menghubungkan simpul-simpul tersebut dinamakan busur (*branch*), sebagaimana ditunjukkan pada Gambar 2.1.



Gambar 2.1. Contoh jaringan

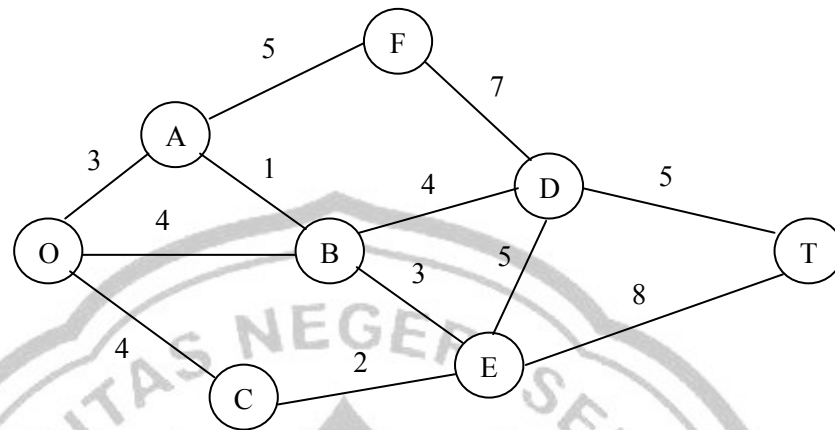
Persoalan jaringan dapat dibagi menjadi 3 (tiga) macam, yaitu:

1. Persoalan rute terpendek (*shortest route*)
2. Persoalan minimasi jaringan atau pohon rentang minimum (*minimum spanning tree*)
3. Persoalan aliran maksimum (*maximal flow*)

Contoh 2.1: (Dimiyati dan Dimiyati, 1999:161)

“Alam Hijau” adalah suatu kawasan hutan lindung yang telah ditata sedemikian rupa sehingga dapat digunakan untuk hiking oleh sejumlah terbatas pecinta alam. Tidak ada mobil yang diperkenankan masuk ke kawasan ini, tetapi

di sana disediakan sejenis tram yang dikelola oleh petugas setempat. Pola jalan dan tempat-tempat persinggahan yang bisa digunakan adalah sebagai berikut.



Gambar 2.2. Pola jalan dan tempat-tempat persinggahan di kawasan Alam Hijau

O adalah tempat masuk ke taman lindung tersebut, sedangkan huruf-huruf yang lain menyatakan tempat-tempat persinggahan. Angka-angka menunjukkan jarak dari satu tempat persinggahan ke persinggahan lainnya, dalam satuan km. Tram yang disediakan akan beroperasi (dijalankan) dari stasiun O ke stasiun T.

Dalam hal ini, ada 3 (tiga) persoalan yang dihadapi oleh manajer taman hutan lindung tersebut, yaitu:

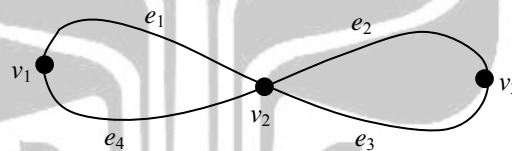
1. Menentukan rute mana yang mempunyai total jarak terkecil untuk mengoperasikan tram dari O ke T.
2. Jika di kawasan tersebut akan dipasang telepon untuk memperlancar komunikasi antar stasiun (termasuk stasiun O dan T), maka di manakah (pada jalur manakah) kabel telepon harus dipasang sehingga panjang total kabel telepon minimum.

3. Untuk menjaga kelestarian lingkungan kawasan tersebut telah dibuat suatu aturan yang menetapkan banyaknya perjalanan tram yang dapat dilakukan pada masing-masing jalur setiap harinya. Batasan yang ditetapkan berbeda antara jalur yang satu dengan jalur lainnya. Karena itu, pada hari-hari libur harus ditentukan rute berbagai perjalanan yang dapat memaksimalkan jumlah perjalanan tram yang dapat dilakukan setiap harinya, tanpa melanggar batasan-batasan yang telah ditentukan pada masing-masing jalur.

2.5. Graf

Graf (*Graph*) didefinisikan sebagai:

$$G = (V, E)$$



Gambar 2.3. Graf G

Dalam hal ini, V merupakan himpunan tidak kosong dari simpul-simpul (*vertices/node*) digambarkan dalam titik-titik, dan E adalah himpunan sisi-sisi (*edges/arcs*) digambarkan dalam garis-garis yang menghubungkan sepasang simpul.

Aplikasi graf sangat luas. Graf dipakai dalam berbagai disiplin ilmu maupun dalam kehidupan sehari-hari. Penggunaan graf di berbagai bidang tersebut adalah untuk memodelkan persoalan.

Beberapa terminologi dasar dalam graf yang harus diketahui adalah sebagai berikut.

1. Graf Berarah (*Directed Graph/Digraph*)

Graf berarah adalah graf yang setiap sisinya diberi orientasi arah. Dalam hal ini sisi yang ditulis (v_1, v_2) berbeda dengan sisi (v_2, v_1) .

2. Graf Berbobot (*Weight Graph*)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga.

3. Graf Lengkap (*Complete Graph*)

Graf lengkap adalah graf sederhana, tidak mengandung gelang (sisi yang kedua simpulnya sama) maupun sisi ganda (dua sisi yang memiliki simpul asal dan simpul tujuan yang sama), serta setiap sisinya mempunyai sisi ke simpul lain.

4. Graf Kosong (*Null Graph*)

Graf yang himpunan sisinya merupakan himpunan kosong disebut sebagai graf kosong.

5. Bertetangga (*Adjacent*)

Dua buah simpul pada graf tak berarah dikatakan bertetangga bila keduanya terhubung dengan sebuah sisi. Dapat dikatakan, jika ada v_1 dan v_2 yang bertetangga, maka harus ada sisi (v_1, v_2)

6. Bersisian (*Incident*)

Untuk sembarang sisi $e = (v_1, v_2)$, sisi e dikatakan bersisian dengan simpul v_1 dan simpul v_2 .

7. Jalan (*Walk*)

Misalkan G adalah sebuah graf. Sebuah jalan di G adalah barisan berhingga tak kosong $W = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k$ yang suku-sukunya bergantian titik dan sisi, sedemikian sehingga v_{i-1} dan v_i adalah titik-titik akhir sisi e_i , untuk $1 < i < k$. Titik v_0 dan titik v_k disebut titik awal dan titik akhir dari jalan W . Sedangkan titik v_1, v_2, \dots, v_{k-1} disebut titik-titik internal dari jalan W .

8. Jejak (*Trail*)

Jika semua sisi e_1, e_2, \dots, e_k dalam jalan W berbeda, maka W disebut jejak.

9. Lintasan (*Path*)

Jika semua sisi e_1, e_2, \dots, e_k dan semua titik v_1, v_2, \dots, v_k dalam jalan W berbeda, maka W disebut lintasan.

10. Sirkuit (*Circuit*)

Jejak yang titik awal dan titik akhirnya sama disebut sirkuit.

11. Siklus (*Cycle*)

Lintasan yang titik awal dan titik akhirnya sama disebut siklus.

2.6. Pohon Rentang Minimum

2.6.1. Pohon Rentang

Pohon rentang adalah suatu pohon T yang berasal dari sebuah graf G . Graf G adalah graf tak berarah terhubung yang bukan merupakan pohon (memiliki siklus). Sedangkan pohon T adalah graf yang merupakan graf pohon yang didapat dengan cara memutuskan siklus-siklus yang terdapat pada graf G . Jadi, pohon rentang adalah graf pohon yang himpunan semua

simpulnya merupakan *improper subset* dari himpunan simpul yang terdapat pada graf G , sedangkan himpunan semua sisinya merupakan *proper subset* dari himpunan semua sisi pada graf G .

2.6.2. Pohon Rentang Minimum

Jika G adalah graf berbobot, maka bobot dari pohon rentang T dari G didefinisikan sebagai jumlah bobot pada semua sisi di T . Pohon rentang yang berbeda memiliki bobot yang berbeda pula. Di antara semua pohon rentang di G , pohon yang memiliki bobot paling minimum dinamakan pohon rentang minimum.

Persoalan pohon rentang minimum menyangkut pemilihan seperangkat penghubung, yang menghubungkan semua *node* dalam suatu jaringan, sedemikian rupa sehingga jumlah panjang dari semua penghubung terpilih tadi adalah sekecil mungkin.

Persoalan pohon rentang minimum merupakan variasi dari persoalan rute terpendek yang perbedaannya terletak pada lintasan yang dicari. Pada rute terpendek, kita mencari lintasan dari sumber ke tujuan yang memberikan total jarak minimum, sedangkan pada pohon rentang minimum yang dipersoalkan adalah menentukan busur-busur yang menghubungkan *node* yang ada pada jaringan, sehingga diperoleh panjang busur total yang minimum (Dimiyati dan Dimiyati, 1999:165-166). Penghubung-penghubung dalam persoalan rute terpendek sudah ada, tinggal mencari lintasan terpendeknya. Sedangkan pada persoalan pohon rentang minimum sebaliknya, penghubung-penghubungnya

belum ada dan bagaimana menentukan penghubung-penghubung itu sehingga diperoleh total panjang yang minimum.

2.7. Algoritma Prim

Salah satu metode untuk mencari pohon rentang minimum ditemukan oleh Robert C. Prim. Metode ini disebut algoritma Prim. Algoritma Prim membentuk pohon rentang minimum dengan langkah per langkah. Pada setiap langkah kita mengambil sisi graf G yang memiliki bobot minimum namun yang terhubung dengan pohon rentang T yang telah terbentuk.

Untuk mencari pohon rentang minimum T dari graf G dengan algoritma Prim, mula-mula dipilih satu titik sembarang (misal v_1). Kemudian ditambahkan satu garis yang berhubungan dengan v_1 dengan bobot yang paling minimum (misal e_1) dan titik ujung lainnya ke T sehingga T terdiri dari sebuah garis e_1 dan dua buah titik-titik ujung garis e_1 (salah satunya adalah v_1). Pada setiap langkah selanjutnya, dipilih sebuah garis dalam $E(G)$ yang bukan anggota $E(T)$ dengan sifat: (1) garis tersebut berhubungan dengan salah satu titik anggota $V(T)$ dan (2) garis tersebut mempunyai bobot yang paling kecil. Langkah tersebut diulang hingga diperoleh $(n-1)$ garis dalam $E(T)$, dengan n adalah jumlah titik dalam G .

Misalkan G adalah graf berlabel dengan n titik dan T adalah pohon rentang minimum yang akan dibentuk (mula-mula kosong). Secara formal algoritma Prim adalah sebagai berikut:

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 siklus.
 - 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. $V(G) = V(G) - \{w\}$

(Siang, 2002: 269-270)

Contoh 2.2:

Gunakan algoritma Prim untuk mencari pohon rentang minimum dari Gambar 2.2 dimulai dari titik O.

Misalkan T adalah pohon rentang minimum yang akan dibuat. Mula-mula $V(T) = \{O\}$ dan $E(T) = \{\}$.

Pada iterasi pertama, pilih garis di dalam $E(G)$ dan di luar $E(T)$ yang berhubungan dengan titik-titik dalam $V(T)$ dengan bobot terkecil. Garis-garis di luar $E(T)$ yang berhubungan dengan titik-titik dalam $V(T)$ adalah garis OA (dengan bobot 3), garis OB (dengan bobot 4), dan garis OC (dengan bobot 4). Pilih garis dengan bobot terkecil, yaitu OA. Tambahkan OA ke $E(T)$ dan titik ujung OA (=A) ke $V(T)$. Sehingga $V(T) = \{O, A\}$ dan $E(T) = \{OA\}$.

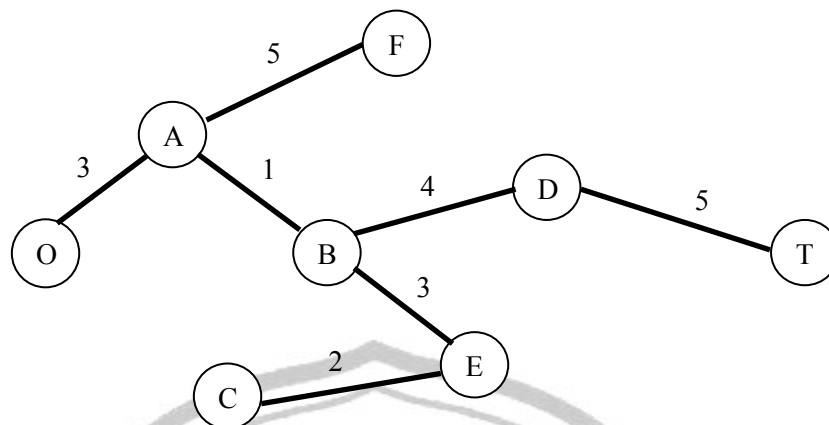
Pada iterasi kedua, pilih garis di dalam $E(G)$ dan di luar $E(T)$ yang berhubungan dengan titik-titik dalam $V(T)$ dengan bobot terkecil. Garis-garis di luar $E(T)$ yang berhubungan dengan titik-titik dalam $V(T)$ adalah garis AF (dengan bobot 5), garis AB (dengan bobot 1), garis OB (dengan bobot 4), dan garis OC (dengan bobot 4). Pilih garis dengan bobot terkecil, yaitu AB . Tambahkan AB ke $E(T)$ dan titik ujung $AB (=B)$ ke $V(T)$. Sehingga $V(T) = \{O, A, B\}$ dan $E(T) = \{OA, AB\}$.

Proses iterasi yang sama diulang-ulang hingga $V(T)$ memuat semua titik dalam G atau jumlah iterasi adalah $(n-1)$, dengan n adalah jumlah titik dalam G . Didapatkan hasil sebagai berikut.

Tabel 2.1. Tabel Iterasi

Iterasi	Sisi yang terpilih	Bobot
0	-	0
1	OA	3
2	AB	1
3	BE	3
4	EC	2
5	BD	4
6	DT	5
7	AF	5

Pohon rentang yang terbentuk setelah semua iterasi dilalui adalah pohon rentang minimum pada Gambar 2.3 dengan panjang 23.





Gambar 2.3. Pohon rentang minimum yang diperoleh dengan algoritma Prim

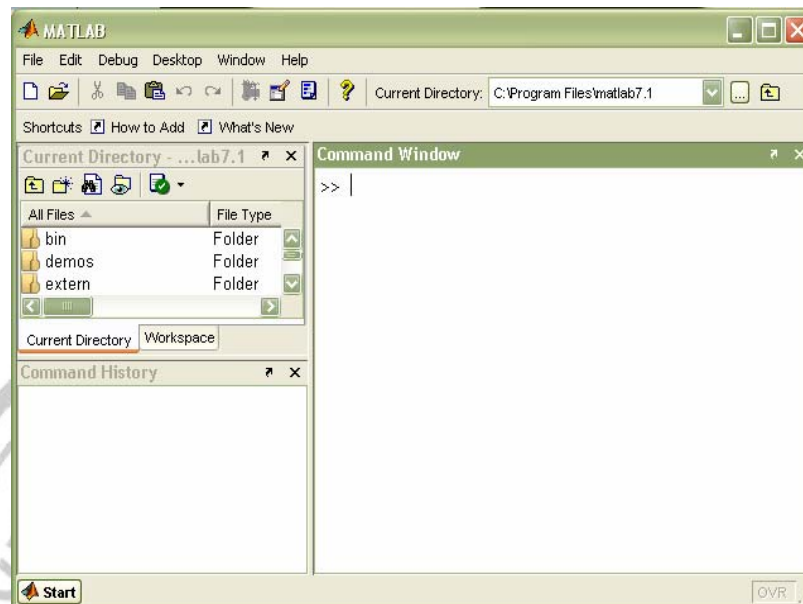
2.8. MATLAB

Penggunaan *software* dalam menyelesaikan masalah optimasi sangatlah penting. Terutama bila melibatkan banyak iterasi dalam menemukan solusi optimum dari suatu permasalahan. MATLAB termasuk salah satu *software* yang banyak digunakan untuk menyelesaikan masalah optimasi.

Nama MATLAB merupakan singkatan dari Matrix Laboratory. MATLAB adalah suatu bahasa pemrograman yang diperuntukkan untuk komputasi teknis. Masalah yang bisa diselesaikan dengan bantuan MATLAB terutama adalah masalah yang bisa diformulasikan dalam bentuk matriks dan vektor. Dalam MATLAB ada banyak *toolbox*. *Toolbox* adalah kumpulan fungsi dalam MATLAB yang digunakan untuk menyelesaikan masalah tertentu.

MATLAB dapat dioperasikan dalam berbagai sistem, dalam skripsi ini digunakan sistem Windows. Untuk membuka MATLAB dapat dilakukan dengan cara *double* klik pada *shortcut* MATLAB  yang ada pada desktop komputer atau dapat dilakukan dengan cara klik  lalu pilih All Program setelah itu

arahkan pada MATLAB dan klik. Kemudian akan muncul tampilan seperti pada Gambar 2.4. Tampilan tersebut dinamakan desktop MATLAB.



Gambar 2.4. Desktop MATLAB

Beberapa Bagian dari Desktop MATLAB

1. Command Window

Window ini adalah *window* utama dari MATLAB. Di sini adalah tempat untuk menjalankan fungsi, mendeklarasikan variabel, menjalankan proses-proses, serta melihat isi variabel.

2. Command History

Window ini berfungsi untuk menyimpan perintah-perintah apa saja yang sebelumnya dilakukan oleh pengguna terhadap MATLAB.

3. Current Directory

Window ini menampilkan isi dari *directory* kerja saat menggunakan MATLAB. Kita dapat mengganti *directory* ini sesuai dengan tempat *directory* kerja yang

diinginkan. *Default* dari alamat *directory* berada dalam folder *works* tempat program files MATLAB berada.

4. Workspace

Workspace berfungsi untuk menampilkan seluruh variabel-variabel yang sedang aktif pada saat pemakaian MATLAB. Apabila variabel berupa data matrikss berukuran besar maka *user* dapat melihat isi dari seluruh data dengan melakukan *double* klik pada variabel tersebut. MATLAB secara otomatis akan menampilkan *window array editor* yang berisikan data pada setiap variabel yang dipilih *user*.

MATLAB menyediakan fungsi *help* yang berisikan tutorial mengenai MATLAB. *User* dapat menjalankan fungsi ini dengan menekan tombol pada *toolbar* atau menuliskan perintah 'helpwin' pada *command window*. MATLAB juga menyediakan fungsi *demos* yang berisikan video tutorial MATLAB serta contoh-contoh program yang bisa dibuat dengan MATLAB.

2.9. Aplikasi MATLAB

Banyak masalah yang bisa diselesaikan dengan bantuan MATLAB terutama yang dapat diformulasikan dalam sebuah matrikss dan vektor. Di antaranya adalah masalah optimasi. MATLAB mempunyai fungsi-fungsi yang sudah siap untuk berbagai masalah optimasi. Kita tinggal memilih fungsi yang sesuai dengan permasalahan yang kita punyai. Kemudian kita menuliskan permasalahan tersebut dalam format MATLAB.

Contoh 2.3:

Gunakan program MATLAB untuk mencari pohon rentang minimum dari Gambar 2.2.

Mencari pohon rentang minimum termasuk ke dalam masalah optimasi yaitu mencari total jarak minimum sedemikian hingga semua titik dalam jaringan tersebut terhubung dan tidak membentuk siklus. Permasalahan ini dapat diselesaikan dengan program 0-1.

Di dalam MATLAB terdapat fungsi untuk menyelesaikan masalah program 0-1 yaitu fungsi *bintprog*. Format untuk menyelesaikan masalah program 0-1 dengan fungsi *bintprog* adalah sebagai berikut.

$$\min f(x)$$

$$\text{st } A(x) \leq b$$

$$A_{\text{eq}}(x) = \text{beq}$$

di mana $f(x)$ adalah fungsi tujuan,

$A(x)$ dan $A_{\text{eq}}(x)$ adalah fungsi kendala, dan

b dan beq adalah nilai ruas kanan dari $A(x)$ dan $A_{\text{eq}}(x)$.

Sintaksnya adalah sebagai berikut.

`x = bintprog (f, A, b) atau`

`x = bintprog (f, A, b, Ae, be)`

Untuk menyelesaikan permasalahan pada Contoh 2.3 maka dari permasalahan tersebut dibentuk model matematikanya.

Misalkan,

x_1 : sisi dari titik O ke A,

x_2 : sisi dari titik O ke B,

x_3 : sisi dari titik O ke C,

x_4 : sisi dari titik A ke F,

x_5 : sisi dari titik A ke B,

x_6 : sisi dari titik B ke D,

x_7 : sisi dari titik B ke E,

x_8 : sisi dari titik C ke E,

x_9 : sisi dari titik D ke F,

x_{10} : sisi dari titik D ke E,

x_{11} : sisi dari titik D ke T dan

x_{12} : sisi dari titik E ke T.

Maka model matematikanya adalah sebagai berikut.

Fungsi tujuan :

$$\text{Minimum } 3x_1 + 4x_2 + 4x_3 + 5x_4 + x_5 + 3x_6 + 2x_7 + 4x_8 + 7x_9 + 5x_{10} + 5x_{11} + 8x_{12}$$

Fungsi kendala:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} = 7$$

$$x_1 + x_2 + x_5 \leq 2$$

$$x_4 + x_5 + x_6 + x_9 \leq 3$$

$$x_2 + x_3 + x_7 + x_8 \leq 3$$

$$x_6 + x_7 + x_{10} \leq 2$$

$$x_{10} + x_{11} + x_{12} \leq 2$$

$$x_1 + x_3 + x_5 + x_7 + x_8 \leq 4$$

Kemudian fungsi tujuan dan fungsi kendala tersebut ditulis dalam bentuk matriks sebagai berikut.

Fungsi tujuan:

$$\text{Minimum } [3 \ 4 \ 4 \ 5 \ 1 \ 4 \ 2 \ 3 \ 7 \ 5 \ 5 \ 8] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix}$$

Fungsi kendala:

$$[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = 7$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} \leq \begin{bmatrix} 2 \\ 3 \\ 3 \\ 2 \\ 2 \\ 4 \end{bmatrix}$$

Setelah itu diketikkan dalam *command window* sesuai dengan format untuk menyelesaikan program 0-1 yaitu sebagai berikut.

```
>> f=[3 4 4 5 1 4 2 3 7 5 5 8];
>> Ae=[1 1 1 1 1 1 1 1 1 1 1 1];
>> be=7;
>> A=[1 1 0 0 1 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 1 0 0 0
0 1 1 0 0 0 1 1 0 0 0 0
0 0 0 0 0 1 1 0 0 1 0 0
0 0 0 0 0 0 0 0 0 1 1 1
1 0 1 0 1 0 1 1 0 0 0 0];
>> b=[2
3
3
2
2
```

4];

Untuk mendapatkan solusi dari permasalahan digunakan fungsi *bintprog*.

```
>> [x,fx]=bintprog(f,A,b,Ae,be)
```

Sehingga diperoleh hasil sebagai berikut.

Optimization terminated.

```
x = 1
```

```
0
```

```
0
```

```
1
```

```
1
```

```
1
```

```
1
```

```
1
```

```
0
```

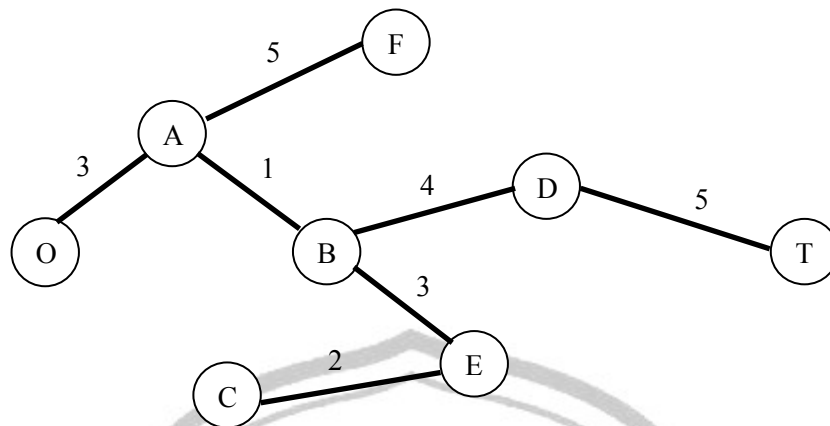
```
0
```

```
1
```

```
0
```

```
fx = 23
```

Dari hasil *Optimization terminated* diperoleh pohon rentang minimum dengan panjang 23. Sisi-sisi yang terpilih yang membentuk pohon rentang minimum ditunjukkan dengan variabel x yang bernilai 1, sedangkan variabel x yang bernilai 0 menunjukkan sisi yang tidak terpilih. Sehingga dapat dilihat bahwa sisi-sisi yang terpilih adalah x_1 (OA), x_4 (AF), x_5 (AB), x_6 (BD), x_7 (BE), x_8 (CE) dan x_{11} (DT). Jadi pohon rentang minimum yang dihasilkan adalah sebagai berikut.



Gambar 2.6. Pohon rentang minimum yang diperoleh dengan program MATLAB

BAB III

METODE PENELITIAN

Pada penelitian ini metode yang digunakan adalah studi kasus. Langkah-langkah yang dilakukan adalah sebagai berikut.

3.1. Menemukan Masalah

Dalam tahap ini diidentifikasi masalah dalam pendistribusian listrik di wilayah distribusi Pekalongan 1 Kota Pekalongan. Kemudian dicari sumber pustaka dan dipilih bagian dari sumber pustaka tersebut yang dapat dijadikan sebagai masalah yang akan dikaji dalam skripsi ini.

3.2. Merumuskan Masalah

Masalah yang ditemukan kemudian dirumuskan ke dalam pertanyaan yang harus diselesaikan yaitu sebagai berikut.

1. Bagaimana hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binary integer programming* menggunakan bantuan program MATLAB?
2. Bagaimana perbandingan hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan dengan algoritma Prim dan *binary integer programming* menggunakan bantuan program MATLAB?

3.3. Pengambilan Data

Dalam penelitian ini, data diperoleh dengan metode dokumentasi yaitu metode pengumpulan data dengan cara mengambil data sekunder yang diperoleh dari PT. PLN Kota Pekalongan. Data tersebut berupa peta jaringan distribusi

listrik di wilayah distribusi Pekalongan 1 Kota Pekalongan dan data panjang kabel yang digunakan di wilayah tersebut.

3.4. Analisis dan Pemecahan Masalah

Dari berbagai sumber pustaka yang menjadi bahan kajian diperoleh suatu pemecahan masalah di atas. Selanjutnya dilakukan langkah-langkah sebagai berikut.

1. Membuat gambar jaringan distribusi listrik PLN Kota Pekalongan berdasarkan peta jaringan distribusi wilayah distribusi Pekalongan 1.
2. Mencari pohon rentang minimum dengan menggunakan algoritma Prim dan *binary integer programming* menggunakan bantuan program MATLAB berdasarkan gambar jaringan distribusi yang telah dibuat.
3. Membandingkan hasil pohon rentang minimum dengan algoritma Prim dan hasil pohon rentang minimum dengan *binary integer programming* menggunakan bantuan program MATLAB.

3.5. Penarikan Kesimpulan

Langkah terakhir dalam metode penelitian adalah penarikan kesimpulan yang diperoleh dari hasil langkah pemecahan masalah.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

Penelitian ini mengkaji tentang jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 Kota Pekalongan. Yang menjadi permasalahan dalam penelitian ini adalah bagaimana mencari pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 dengan menggunakan algoritma Prim dan *binary integer programming* menggunakan bantuan program MATLAB.

Dalam penelitian ini akan dicari total panjang kabel distribusi yang bernilai minimum yang memuat semua titik. Data yang diperoleh yaitu gambar peta jaringan distribusi listrik di wilayah distribusi Pekalongan 1 Kota Pekalongan dan jarak/panjang kabel yang digunakan di wilayah tersebut. Dalam hal ini pendistribusian tidak sampai langsung kepada pelanggan, hanya sampai pada ujung jalan yang menuju ke pelanggan. Berdasarkan data yang diperoleh kemudian disajikan pada Lampiran 1.

Dari data tersebut kemudian dibuat gambar jaringan dengan titik sumbernya adalah gardu induk PLN yang berada di Jl. Urip Sumoharjo, kemudian dihubungkan ke semua titik yang merupakan titik perpotongan ujung-ujung kabel pada ujung jalan. Gambar jaringan tersebut disajikan pada Lampiran 2.

Dari gambar jaringan tersebut diketahui terdapat 90 titik dan 111 busur yang menghubungkan setiap titik. Berdasarkan gambar jaringan tersebut akan

dicari pohon rentang minimum dengan algoritma Prim dan *binary integer programming* menggunakan bantuan program MATLAB.

4.1.1. Hasil penelitian dengan algoritma Prim

Untuk menentukan pohon rentang minimum dengan algoritma Prim langkah-langkah yang dilakukan adalah sebagai berikut.

Misalkan T adalah pohon rentang minimum yang akan dibuat. Mula-mula pilih titik A_1 sebagai titik awal sehingga $V(T) = \{A_1\}$ dan $E(T) = \{\}$.

Pada iterasi pertama, pilih sisi yang berhubungan dengan titik di $V(T)$ dengan bobot terkecil dan tidak membentuk siklus. Karena hanya ada satu sisi yang berhubungan dengan titik A_1 yaitu sisi X_1 (dengan bobot 90) maka tambahkan X_1 ke $E(T)$ dan tambahkan titik ujung X_1 yaitu A_2 ke $V(T)$. Sehingga $V(T) = \{A_1, A_2\}$ dan $E(T) = \{X_1\}$.

Pada iterasi kedua, pilih sisi yang berhubungan dengan titik-titik di $V(T)$ dengan bobot terkecil dan tidak membentuk siklus. Sisi-sisi yang berhubungan dengan titik-titik di $V(T)$ adalah X_2 (dengan bobot 189) dan X_{17} (dengan bobot 207). Sisi dengan bobot terkecil dan tidak membentuk siklus adalah X_2 . Tambahkan X_2 ke $E(T)$ dan titik ujung X_2 ke $V(T)$ yaitu A_3 . Sehingga $V(T) = \{A_1, A_2, A_3\}$ dan $E(T) = \{X_1, X_2\}$.

Pada iterasi ketiga, pilih sisi yang berhubungan dengan titik di $V(T)$ dengan bobot terkecil dan tidak membentuk siklus. Sisi-sisi yang berhubungan dengan titik-titik di $V(T)$ adalah X_3 (dengan bobot 42), X_{17} (dengan bobot 207), dan X_{41} (dengan bobot 108). Sisi dengan bobot terkecil dan tidak membentuk siklus adalah X_3 . Tambahkan X_3 ke $E(T)$ dan titik ujung X_3 yaitu A_4 ke $V(T)$. sehingga $V(T) =$

$\{A1, A2, A3, A4\}$ dan $E(T) = \{X1, X2, X3\}$.

Proses iterasi dilakukan terus menerus sehingga $V(T)$ memuat semua titik.

Sebagian hasil dapat dilihat pada tabel berikut ini.

Tabel 4.1. Tabel iterasi pencarian pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 dengan algoritma Prim

Iterasi ke-	Sisi yang terpilih	Titik yang terkait	Bobot
1	X1	A1-A2	90
2	X2	A2-A3	189
3	X3	A3-A4	42
4	X41	A3-A38	108
5	X4	A4-A5	175
6	X5	A5-A6	42
7	X6	A6-A7	154
8	X46	A7-A43	91
9	X47	A43-A44	98
10	X70	A43-A61	126
11	X71	A61-A62	140
12	X72	A62-A63	85
13	X79	A63-A67	39
14	X80	A67-A68	44
15	X81	A68-A69	85
16	X78	A62-A8	135
17	X73	A63-A64	138
18	X82	A64-A70	39
19	X74	A64-A65	99
20	X84	A65-A72	50
21	X85	A72-A73	66
22	X75	A65-A66	94
23	X87	A66-A10	105
24	X83	A70-A71	116
25	X32	A32-A7	144
...

Hasil selengkapnya dapat dilihat pada Lampiran 3.

4.1.2. Hasil penelitian dengan *binary integer programming* menggunakan bantuan program MATLAB

Untuk menentukan pohon rentang minimum dengan *binary integer programming* menggunakan bantuan program MATLAB langkah-langkah yang dilakukan adalah sebagai berikut.

1. Menyusun model matematika berdasarkan gambar jaringan pada Lampiran 2.

Model matematika tersebut meliputi fungsi tujuan dan fungsi kendala.

Misalkan,

X_1 : sisi dari titik A1 ke A2,

X_2 : sisi dari titik A2 ke A3,

X_3 : sisi dari titik A3 ke A4,

X_4 : sisi dari titik A4 ke A5,

X_5 : sisi dari titik A5 ke A6,

X_6 : sisi dari titik A6 ke A7,

X_7 : sisi dari titik A7 ke A8,

X_8 : sisi dari titik A8 ke A9,

X_9 : sisi dari titik A9 ke A10,

X_{10} : sisi dari titik A10 ke A11,

X_{11} : sisi dari titik A11 ke A12,

X_{12} : sisi dari titik A12 ke A13,

X_{13} : sisi dari titik A13 ke A14,

X_{14} : sisi dari titik A14 ke A15,

X_{15} : sisi dari titik A15 ke A16,

X_{16} : sisi dari titik A16 ke A17,

X_{17} : sisi dari titik A2 ke A18,

X_{18} : sisi dari titik A18 ke A19,

X_{19} : sisi dari titik A19 ke A20,

X_{20} : sisi dari titik A20 ke A21,

X_{21} : sisi dari titik A21 ke A22,

X_{22} : sisi dari titik A22 ke A23,

X_{23} : sisi dari titik A23 ke A24,

X_{24} : sisi dari titik A24 ke A25,

X_{25} : sisi dari titik A25 ke A26,

X_{26} : sisi dari titik A26 ke A27,

X_{27} : sisi dari titik A27 ke A28,

X_{28} : sisi dari titik A28 ke A29,

X_{29} : sisi dari titik A29 ke A30,

X_{30} : sisi dari titik A30 ke A31,

X_{31} : sisi dari titik A31 ke A32,

X_{32} : sisi dari titik A32 ke A7,

X_{33} : sisi dari titik A19 ke A33,

X_{34} : sisi dari titik A33 ke A34,

X_{35} : sisi dari titik A34 ke A35,

X_{36} : sisi dari titik A35 ke A31,

X_{37} : sisi dari titik A21 ke A36,

X_{38} : sisi dari titik A36 ke A37,

X_{39} : sisi dari titik A37 ke A27,

X_{40} : sisi dari titik A37 ke A24,

X_{41} : sisi dari titik A3 ke A38,

X_{42} : sisi dari titik A38 ke A39,

X_{43} : sisi dari titik A39 ke A40,

X_{44} : sisi dari titik A40 ke A41

X_{45} : sisi dari titik A41 ke A42,

X_{46} : sisi dari titik A7 ke A43,

X_{47} : sisi dari titik A43 ke A44,

X_{48} : sisi dari titik A44 ke A45,

X_{49} : sisi dari titik A45 ke A46,

X_{50} : sisi dari titik A46 ke A47,

X_{51} : sisi dari titik A47 ke A48,

X_{52} : sisi dari titik A48 ke A49,

X_{53} : sisi dari titik A49 ke A50,

X_{54} : sisi dari titik A50 ke A51,

X_{55} : sisi dari titik A51 ke A52,

X_{56} : sisi dari titik A52 ke A53,

X_{57} : sisi dari titik A53 ke A54,

X_{58} : sisi dari titik A54 ke A55,

X_{59} : sisi dari titik A55 ke A11,

X_{60} : sisi dari titik A11 ke A56,

X_{61} : sisi dari titik A56 ke A57,

X_{62} : sisi dari titik A57 ke A58,

X_{63} : sisi dari titik A58 ke A28,

X_{64} : sisi dari titik A59 ke A60,

X_{65} : sisi dari titik A60 ke A30,

X_{66} : sisi dari titik A8 ke A60,

X_{67} : sisi dari titik A60 ke A58,

X_{68} : sisi dari titik A9 ke A59,

X_{69} : sisi dari titik A59 ke A57,

X_{70} : sisi dari titik A43 ke A61,

X_{71} : sisi dari titik A61 ke A62,

X_{72} : sisi dari titik A62 ke A63,

X_{73} : sisi dari titik A63 ke A64,

X_{74} : sisi dari titik A64 ke A65,

X_{75} : sisi dari titik A65 ke A66,

X_{76} : sisi dari titik A46 ke A61,

X_{77} : sisi dari titik A47 ke A62,

X_{78} : sisi dari titik A62 ke A8,

X_{79} : sisi dari titik A63 ke A67,

X_{80} : sisi dari titik A67 ke A68,

X_{81} : sisi dari titik A68 ke A69,

X_{82} : sisi dari titik A64 ke A70,

X_{83} : sisi dari titik A70 ke A71,

X_{84} : sisi dari titik A65 ke A72,

X_{85} : sisi dari titik A72 ke A73,

X_{86} : sisi dari titik A49 ke A66,

X_{87} : sisi dari titik A66 ke A10,

X_{88} : sisi dari titik A54 ke A74,

X_{89} : sisi dari titik A74 ke A75,

X_{90} : sisi dari titik A75 ke A76,

X_{91} : sisi dari titik A76 ke A77,

X_{92} : sisi dari titik A77 ke A78,

X_{93} : sisi dari titik A78 ke A79,

X_{94} : sisi dari titik A54 ke A80,

X_{95} : sisi dari titik A80 ke A81,

X_{96} : sisi dari titik A81 ke A12,

X_{97} : sisi dari titik A81 ke A82,

X_{98} : sisi dari titik A82 ke A83,

X_{99} : sisi dari titik A83 ke A84,

X_{100} : sisi dari titik A84 ke A85,

X_{101} : sisi dari titik A85 ke A86,

X_{102} : sisi dari titik A85 ke A17,

X_{103} : sisi dari titik A84 ke A16,

X_{104} : sisi dari titik A83 ke A87,

X_{105} : sisi dari titik A82 ke A76,

X_{106} : sisi dari titik A82 ke A14,

X_{107} : sisi dari titik A14 ke A88,

X_{108} : sisi dari titik A88 ke A89,

X_{109} : sisi dari titik A89 ke A90,

X_{110} : sisi dari titik A15 ke A88, dan

X_{111} : sisi dari titik A16 ke A89.

Maka model matematikanya adalah sebagai berikut.

Fungsi tujuan:

Bentuk umum: $\sum c_i X_i$

dengan c_i adalah bobot pada sisi ke i

X_i adalah sisi ke i

$i = 1, 2, \dots, 111$ (banyaknya sisi)

MINIMUMKAN $90X_1 + 189X_2 + 42X_3 + 175X_4 + 42X_5 + 154X_6 + 231X_7 + 158X_8 + 270X_9 + 154X_{10} + 121X_{11} + 47X_{12} + 180X_{13} + 198X_{14} + 186X_{15} + 204X_{16} + 207X_{17} + 63X_{18} + 72X_{19} + 180X_{20} + 117X_{21} + 144X_{22} + 90X_{23} + 90X_{24} + 261X_{25} + 161X_{26} + 112X_{27} + 56X_{28} + 57X_{29} + 154X_{30} + 77X_{31} + 144X_{32} + 171X_{33} + 153X_{34} + 117X_{35} + 182X_{36} + 117X_{37} + 36X_{38} + 369X_{39} + 189X_{40} + 108X_{41} + 189X_{42} + 126X_{43} + 90X_{44} + 63X_{45} + 91X_{46} + 98X_{47} + 217X_{48} + 112X_{49} + 84X_{50} + 98X_{51} + 256X_{52} + 209X_{53} + 55X_{54} + 165X_{55} + 115X_{56} + 120X_{57} + 94X_{58} + 171X_{59} + 187X_{60} + 228X_{61} + 204X_{62} + 190X_{63} + 167X_{64} + 217X_{65} + 245X_{66} + 252X_{67} + 198X_{68} + 278X_{69} + 126X_{70} + 140X_{71} + 85X_{72} + 138X_{73} + 99X_{74} + 94X_{75} + 224X_{76} + 252X_{77} + 133X_{78} + 39X_{79} + 44X_{80} + 85X_{81} + 39X_{82} + 116X_{83} + 50X_{84} + 66X_{85} + 457X_{86} + 105X_{87} + 130X_{88} + 35X_{89} + 160X_{90} + 85X_{91} + 120X_{92} + 170X_{93} + 94X_{94} + 149X_{95} + 116X_{96} + 245X_{97} + 372X_{98} + 122X_{99} + 192X_{100} + 289X_{101} + 192X_{102} +$

$$216X_{103} + 321X_{104} + 187X_{105} + 249X_{106} + 162X_{107} + 192X_{108} + 222X_{109} + 90X_{110} + 102X_{111}$$

Fungsi kendala:

$$\text{Bentuk umum: } \sum X_i = k \text{ dan } \sum X_i \leq k$$

di mana $X_i = 0$ atau 1 untuk $i = 1, 2, \dots, 111$

$$k = n-1$$

n = banyaknya titik dalam persamaan/pertidaksamaan yang dibuat

Fungsi kendala yang berbentuk persamaan dibuat berdasarkan sisi-sisi yang tidak membentuk siklus pada gambar jaringan distribusi listrik, sehingga sisi-sisi dalam persamaan yang dibuat harus terpilih semua. Sedangkan fungsi kendala yang berbentuk pertidaksamaan dibuat berdasarkan sisi-sisi yang membentuk siklus pada gambar jaringan distribusi listrik, sehingga salah satu sisi harus terhapus untuk memutus siklusnya.

$$X_1 = 1$$

$$X_{41} + X_{42} + X_{43} + X_{44} + X_{45} = 5$$

$$X_{79} + X_{80} + X_{81} = 3$$

$$X_{82} + X_{83} = 2$$

$$X_{84} + X_{85} = 2$$

$$X_{91} + X_{92} + X_{93} = 3$$

$$X_{101} = 1$$

$$X_{104} = 1$$

$$X_{109} = 1$$

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} + X_{12} + X_{13} + X_{14} +$$

$$\begin{aligned}
&X_{15} + X_{16} + X_{17} + X_{18} + X_{19} + X_{20} + X_{21} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} + X_{27} + \\
&X_{28} + X_{29} + X_{30} + X_{31} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} + X_{37} + X_{38} + X_{39} + X_{40} + \\
&X_{41} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} + X_{47} + X_{48} + X_{49} + X_{50} + X_{51} + X_{52} + X_{53} + \\
&X_{54} + X_{55} + X_{56} + X_{57} + X_{58} + X_{59} + X_{60} + X_{61} + X_{62} + X_{63} + X_{64} + X_{65} + X_{66} + \\
&X_{67} + X_{68} + X_{69} + X_{70} + X_{71} + X_{72} + X_{73} + X_{74} + X_{75} + X_{76} + X_{77} + X_{78} + X_{79} + \\
&X_{80} + X_{81} + X_{82} + X_{83} + X_{84} + X_{85} + X_{86} + X_{87} + X_{88} + X_{89} + X_{90} + X_{91} + X_{92} + \\
&X_{93} + X_{94} + X_{95} + X_{96} + X_{97} + X_{98} + X_{99} + X_{100} + X_{101} + X_{102} + X_{103} + X_{104} + \\
&X_{105} + X_{106} + X_{107} + X_{108} + X_{109} + X_{110} + X_{111} = 89
\end{aligned}$$

$$X_2 + X_3 + X_4 + X_5 + X_6 + X_{17} + X_{18} + X_{31} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} \leq 12$$

$$\begin{aligned}
&X_{19} + X_{20} + X_{27} + X_{28} + X_{29} + X_{30} + X_{33} + X_{34} + X_{35} + X_{36} + X_{37} + X_{38} + X_{39} \\
&\leq 12
\end{aligned}$$

$$X_{21} + X_{22} + X_{23} + X_{37} + X_{38} + X_{40} \leq 5$$

$$X_{24} + X_{25} + X_{26} + X_{39} + X_{40} \leq 4$$

$$X_{47} + X_{48} + X_{49} + X_{70} + X_{77} \leq 4$$

$$X_{50} + X_{71} + X_{77} + X_{78} \leq 3$$

$$X_{51} + X_{52} + X_{72} + X_{73} + X_{74} + X_{75} + X_{78} + X_{86} \leq 7$$

$$X_7 + X_{46} + X_{70} + X_{71} + X_{76} \leq 4$$

$$X_8 + X_9 + X_{72} + X_{73} + X_{74} + X_{75} + X_{76} + X_{87} \leq 7$$

$$X_{10} + X_{53} + X_{54} + X_{55} + X_{56} + X_{57} + X_{58} + X_{59} + X_{86} + X_{87} \leq 9$$

$$X_7 + X_{30} + X_{31} + X_{32} + X_{65} + X_{66} \leq 5$$

$$X_8 + X_{64} + X_{66} + X_{68} \leq 3$$

$$X_{28} + X_{29} + X_{63} + X_{65} + X_{67} \leq 4$$

$$X_{62} + X_{64} + X_{67} + X_{69} \leq 3$$

$$X_9 + X_{10} + X_{60} + X_{61} + X_{68} + X_{69} \leq 5$$

$$X_{11} + X_{58} + X_{59} + X_{94} + X_{95} + X_{96} \leq 5$$

$$X_{88} + X_{89} + X_{90} + X_{94} + X_{95} + X_{97} + X_{105} \leq 6$$

$$X_{12} + X_{13} + X_{96} + X_{97} + X_{106} \leq 4$$

$$X_{14} + X_{15} + X_{98} + X_{99} + X_{103} + X_{106} \leq 5$$

$$X_{16} + X_{100} + X_{102} + X_{103} \leq 3$$

$$X_{14} + X_{107} + X_{110} \leq 2$$

$$X_{15} + X_{108} + X_{110} + X_{111} \leq 3$$

$$X_8 + X_{30} + X_{31} + X_{32} + X_{46} + X_{64} + X_{65} + X_{68} + X_{70} + X_{71} + X_{78} \leq 10$$

2. Memasukkan model matematika tersebut sesuai dengan format ke dalam MATLAB. Input dalam program MATLAB adalah dalam bentuk matriks, sehingga model matematika di atas terlebih dahulu dibuat dalam bentuk matriks sebagaimana dalam penyelesaian Contoh 2.3. Kemudian inputnya diketikkan dalam *command window* sesuai dengan format. Inputnya dapat dilihat pada Lampiran 5.
3. Mencari solusi optimum dengan fungsi `bintprog`. Sintaknya adalah

$$[x, fx] = \text{bintprog}(f, a, b, ae, be)$$

Keterangan:

x : sisi-sisi dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 di mana $x_i = 0$ atau 1 untuk $i = 1, 2, \dots, 111$

fx : panjang pohon rentang minimum yang diperoleh

f : koefisien dari fungsi tujuan

a : koefisien dari fungsi kendala yang berbentuk pertidaksamaan

b : nilai ruas kanan dari fungsi kendala yang berbentuk pertidaksamaan

ae : koefisien dari fungsi kendala yang berbentuk persamaan

be : nilai ruas kanan dari fungsi kendala yang berbentuk persamaan

4. Membaca dan menganalisis hasil yang diperoleh. Sebagian hasil output dapat dilihat pada tabel berikut ini.

Tabel 4.2. Tabel output program MATLAB

```

Optimization terminated.
x =
    1
    1
    1
    1
    1
    1
    1
    0
    1
    0
    1
    1
    1
    1
    1
    0
    1
    1
    1
    1
    1
    ...

fx =
    11312
  
```

Output selengkapnya dapat dilihat pada Lampiran 6.

4.2. Pembahasan

4.2.1. Pencarian pohon rentang minimum dari jaringan distribusi listrik PLN

Kota Pekalongan wilayah distribusi Pekalongan 1 dengan algoritma Prim

Misalkan A_i adalah titik perpotongan ujung-ujung kabel pendistribusian listrik di wilayah distribusi Pekalongan 1 Kota Pekalongan, di mana $i = 1, 2, \dots, 90$. Sedangkan keterhubungan ujung kabel yang satu dengan yang lainnya dinyatakan dengan sisi yang dimisalkan dengan X_i di mana $i = 1, 2, \dots, 111$ dan jarak dari ujung kabel yang satu dengan ujung yang lainnya dinyatakan dengan bobot.

Pencarian pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 dengan algoritma Prim berdasarkan gambar jaringan yang telah dibuat adalah dengan cara menelusuri dari titik awal yaitu A_1 sampai dengan titik terakhir yaitu titik A_{90} , dengan mempertimbangkan bobot yang minimum yang terlewati dan tidak membentuk siklus. Proses iterasi yang dilakukan untuk mencari pohon rentang minimum dengan algoritma Prim dapat dilihat pada Lampiran 3.

Pohon rentang minimum yang dihasilkan mempunyai panjang 11.312 m. Hasil tersebut diperoleh dengan menjumlahkan bobot pada sisi-sisi yang terpilih yang membentuk pohon rentang minimum. Gambar pohon rentang minimum yang dihasilkan dengan algoritma Prim dapat dilihat pada Lampiran 4.

4.2.2. Pencarian pohon rentang minimum dari jaringan distribusi listrik PLN

Kota Pekalongan wilayah distribusi Pekalongan 1 dengan *binary integer programming* menggunakan bantuan program MATLAB

Berdasarkan model matematika yang telah dibuat kemudian dimasukkan ke dalam program MATLAB sesuai dengan formatnya. Kemudian untuk mendapatkan solusi optimum dari permasalahan digunakan fungsi `bintprog`. Inputnya dapat dilihat pada Lampiran 5 dan outputnya dapat dilihat pada Lampiran 6. Dari output yang dihasilkan, diperoleh nilai x dan fx . Variabel x menyatakan sisi-sisi dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1, di mana $x_i = 0$ atau 1 untuk $i = 1, 2, \dots, 111$. Sedangkan fx menyatakan panjang pohon rentang minimum. Variabel x yang bernilai 1 menyatakan bahwa sisi tersebut terpilih sebagai sisi dalam pohon rentang minimum, sedangkan yang bernilai 0 menyatakan sisi yang tidak terpilih.

Pohon rentang minimum yang diperoleh dengan *binary integer programming* menggunakan bantuan program MATLAB mempunyai panjang 11.312 m, hal ini ditunjukkan dengan nilai $fx = 11.312$. Pohon rentang minimum yang diperoleh dengan *binary integer programming* menggunakan bantuan program MATLAB ternyata sama dengan pohon rentang minimum yang diperoleh dengan algoritma Prim, hal ini ditunjukkan dengan sisi-sisi yang terpilih dalam pohon rentang minimum yang diperoleh sama.

4.2.3. Perbandingan hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 dengan algoritma Prim dan *binary integer programming* menggunakan bantuan program MATLAB

Berdasarkan hasil penelitian, pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan wilayah distribusi Pekalongan 1 yang diperoleh baik dengan algoritma Prim maupun dengan *binary integer programming* menggunakan bantuan program MATLAB ternyata sama. Hal ini ditunjukkan dengan sisi-sisi yang terpilih dalam pohon rentang minimum yang diperoleh sama dan panjang pohon rentang minimum yang diperoleh juga sama yaitu 11.312 m. Sehingga dari kedua cara tersebut menghasilkan nilai optimum yang sama. Dengan demikian dapat menghemat kabel sepanjang 5.862 m dari total panjang kabel terpasang yaitu 17.174 m.

Hasil ini dapat diterapkan dengan asumsi besarnya tegangan di wilayah tersebut seragam. Sebagaimana dalam penelitian ini, telah diasumsikan bahwa besarnya tegangan di wilayah distribusi Pekalongan 1 adalah seragam. Karena wilayah distribusi Pekalongan 1 merupakan wilayah perumahan di mana kebanyakan pelanggannya menggunakan listrik untuk kebutuhan rumah tangga, walaupun ada beberapa pelanggan yang menggunakan listrik untuk kebutuhan industri.

BAB V

PENUTUP

5.1. Simpulan

Dari hasil penelitian dan pembahasan maka diperoleh simpulan sebagai berikut.

1. Hasil pohon rentang minimum dari jaringan distribusi listrik PLN Kota Pekalongan yang diperoleh dengan algoritma Prim adalah pohon rentang minimum seperti gambar pada Lampiran 4 dengan panjang 11.312 m. Hasil pohon rentang minimum yang diperoleh dengan *binary integer programming* menggunakan bantuan program MATLAB ternyata juga mempunyai panjang 11.312 m dengan sisi-sisi yang terpilih sama dengan sisi-sisi yang terpilih dalam pohon rantang minimum yang diperoleh dengan algoritma Prim. Dengan demikian dapat menghemat kabel sepanjang 5.862 m dari total panjang kabel terpasang yaitu 17.174 m.
2. Hasil pohon rentang minimum yang diperoleh baik dengan algoritma Prim maupun dengan *binary integer programming* menggunakan bantuan program MATLAB ternyata sama. Hal ini dapat dilihat dari panjang pohon rentang minimum yang diperoleh sama dan sisi-sisi yang terpilih dalam pohon rentang minimum tersebut juga sama. Sehingga kedua cara pencarian pohon rentang minimum tersebut memberikan hasil yang optimum sama.

5.2. Saran

Dari penelitian ini saran yang dapat diberikan adalah sebagai berikut.

1. Dalam skripsi ini, pencarian pohon rentang minimum dilakukan dengan bantuan program MATLAB. Namun demikian metode yang digunakan masih semi manual, di mana output yang dihasilkan tidak langsung berupa gambar pohon rentang minimum. Untuk itu diharapkan pada penelitian-penelitian selanjutnya yang mengkaji tentang pencarian pohon rentang minimum dapat membuat program yang secara langsung memberikan output berupa gambar pohon rentang minimumnya dan menggunakan *software* komputer yang lain sehingga dapat menambah pengetahuan tentang *software-software* komputer yang dapat digunakan untuk mencari pohon rentang minimum.
2. Dalam penelitian ini diasumsikan bahwa besarnya tegangan kabel adalah seragam, padahal dalam kenyataannya besarnya tegangan tidak selalu seragam. Untuk itu diharapkan pada penelitian-penelitian selanjutnya yang mengkaji tentang jaringan distribusi listrik dapat mempertimbangkan besarnya tegangan listrik, karena pendistribusian listrik untuk wilayah perumahan dan wilayah industri menggunakan tegangan yang berbeda.

DAFTAR PUSTAKA

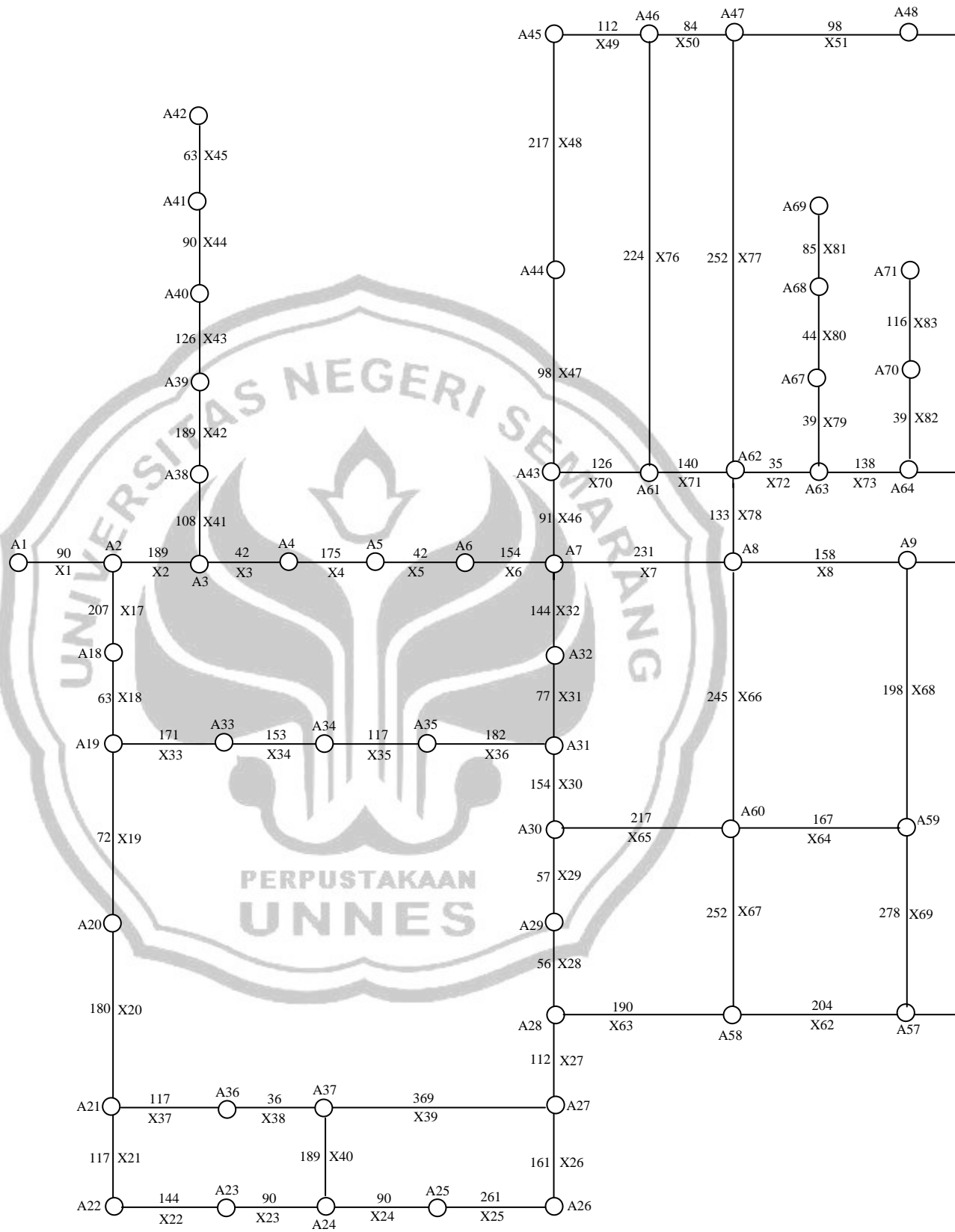
- Arhami, M dan Desiani, A. 2005. *Pemrograman Matlab*. Yogyakarta: Andi
- Dimyati, T dan Dimyati, A. 1999. *Operation Research Model-Model Pengambilan Keputusan*. Bandung: PT. Sinar Baru Algesindo
- Hillier, F.S. et al. Tanpa Tahun. *Pengantar Riset Operasi Edisi Kelima Jilid 1*. Terjemahan oleh Gunawan, E dan Mulia, A.W. 1990. Jakarta: Erlangga
- Mulyono, S. 2002. *Riset Operasi*. Jakarta: Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia
- Munir, R. 2001. *Matematika Diskrit*. Bandung: Informatika
- Santosa, B. 2008. *Matlab untuk Statistika dan Teknik Optimasi, Aplikasi untuk Rekayasa dan Bisnis*. Yogyakarta: Graha Ilmu
- Siang, J.J. 2002. *Matematika Diskrit dan Aplikasinya Pada Ilmu Komputer*. Yogyakarta: Andi
- Siswanto. 2007. *Operation Research Jilid 1*. Jakarta: Erlangga
- Sutarno, H dkk. 2005. *Matematika Diskrit*. Malang: Universitas Negeri Malang
- Suyitno, H. 1999. *Pengantar Program Linear*. Semarang: IKIP Semarang

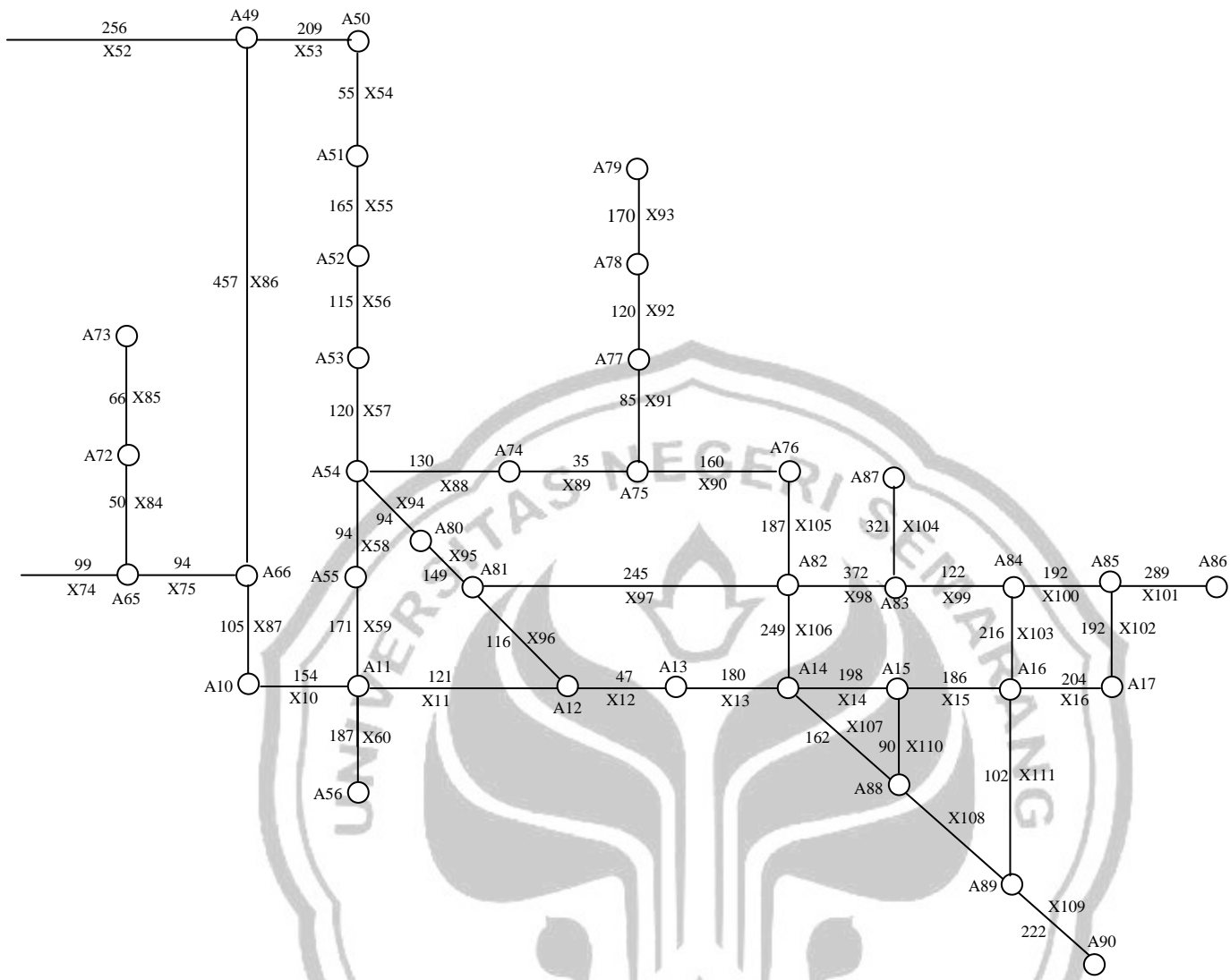
Data Hasil Penelitian

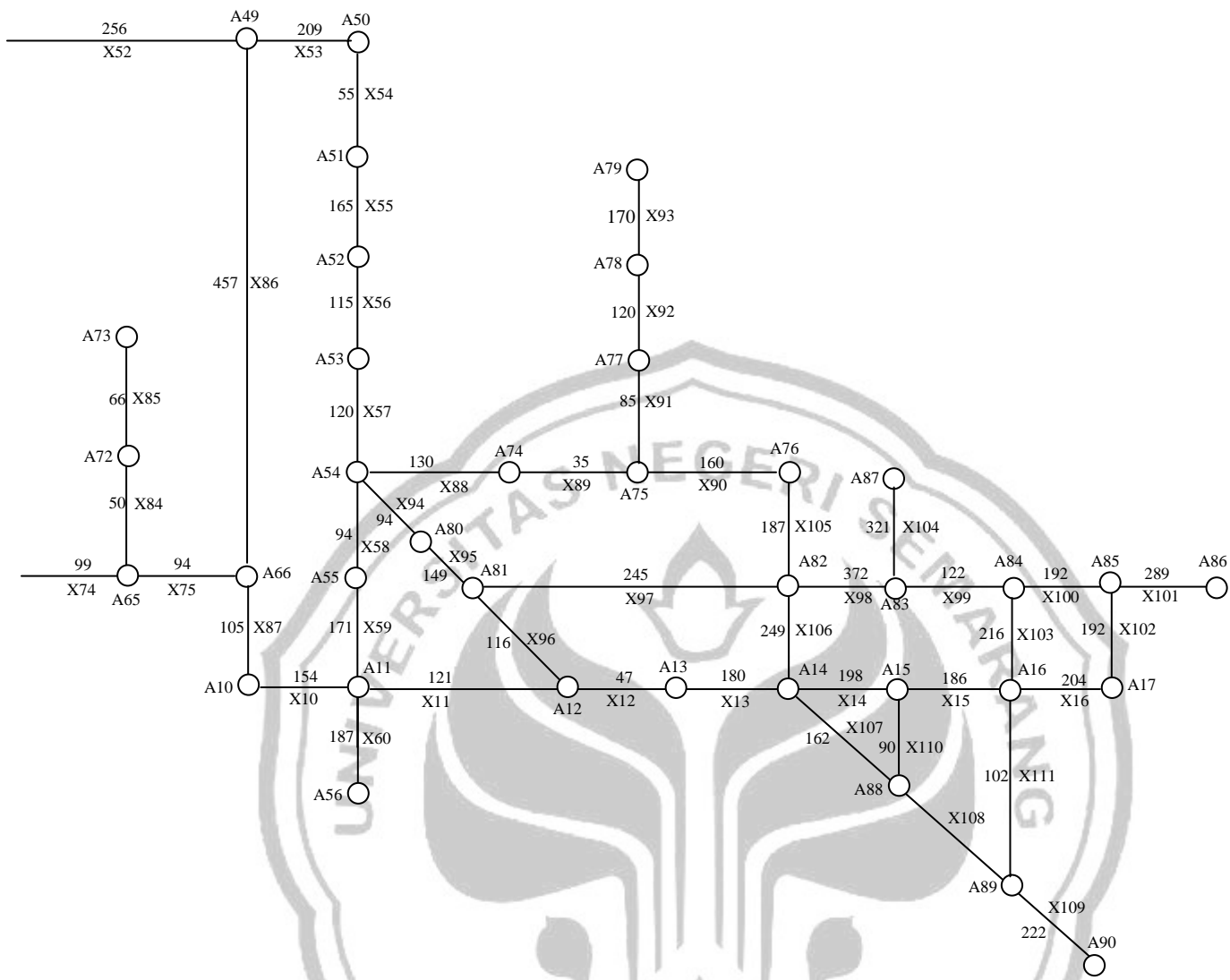
No.	Nama Jalan	Titik	Sisi	Jarak (m)
1.	Jl. Urip Sumoharjo	A1-A2	X1	90
		A2-A3	X2	189
		A3-A4	X3	42
		A4-A5	X4	175
		A5-A6	X5	42
		A6-A7	X6	154
2.	Jl. KHM Mansyur	A7-A8	X7	231
		A8-A9	X8	158
		A9-A10	X9	270
		A10-11	X10	154
3.	Jl. Pemuda	A11-A12	X11	121
4.	Jl. Imam Bonjol	A12-A13	X12	47
		A13-A14	X13	180
5.	Jl. Progo	A14-A15	X14	198
		A15-A16	X15	186
		A16-A17	X16	204
6.	Jl. Karya Bakti	A2-A18	X17	207
		A18-A19	X18	63
		A19-A20	X19	72
		A20-A21	X20	180
		A21-A22	X21	117
7.	Jl. Dharma Bakti	A22-A23	X22	144
		A23-A24	X23	90
		A24-A25	X24	90
		A25-A26	X25	261
8.	Jl. Jend. Sudirman	A26-A27	X26	161
		A27-A28	X27	112
		A28-A29	X28	56
		A29-A30	X29	57
		A30-A31	X30	154
		A31-A32	X31	77
		A32-A7	X32	144
9.	Jl. Sikembang	A19-A33	X33	171
		A33-A34	X34	153
		A34-A35	X35	117
		A35-A31	X36	182

10.	Jl. Jaya Bakti	A21-A36	X37	117
11.	Jl. Yudha Bakti	A37-A27	X39	369
12.	Jl. Setia Bakti	A36-A37	X38	36
		A37-A24	X40	189
13.	Jl. Binagriya	A3-A38	X41	108
		A38-A39	X42	189
		A39-A40	X43	126
		A40-A41	X44	90
		A41-A42	X45	63
14.	Jl. Wilis	A7-A43	X46	91
		A43-A44	X47	98
		A44-A45	X48	217
15.	Jl. Sriwijaya	A45-A46	X49	112
		A46-A47	X50	84
		A47-A48	X51	98
		A48-A49	X52	256
		A49-A50	X53	209
16.	Jl. Gajah Mada	A50-A51	X54	55
		A51-A52	X55	165
		A52-A53	X56	115
		A53-A54	X57	120
		A54-A55	X58	94
		A55-A11	X59	171
17.	Jl. Hayam Wuruk	A11-A56	X60	187
18.	Jl. Sulawesi	A56-A57	X61	228
19.	Jl. Irian	A57-A58	X62	204
		A58-A28	X63	190
20.	Jl. Kalimantan	A59-A60	X64	167
		A60-A30	X65	217
21.	Jl. Sumatra	A8-A60	X66	245
		A60-A58	X67	252
22.	Jl. Jawa	A9-A59	X68	198
		A59-A57	X69	278
23.	Jl. Kurinci	A43-A61	X70	126
		A61-A62	X71	140
		A62-A63	X72	85
		A63-A64	X73	138
		A64-A65	X74	99
		A65-A66	X75	94

24.	Jl. Majapahit	A46-A61	X76	224
25.	Jl. Singasari	A47-A62	X77	252
26.	Jl. Argopuro	A62-A8	X78	133
27.	Jl. Rinjani	A63-A67	X79	39
		A67-A68	X80	44
		A68-A69	X81	85
28.	Jl. Semeru	A64-A70	X82	39
		A70-A71	X83	116
29.	Jl. Merbabu	A65-A72	X84	50
		A72-A73	X85	66
30.	Jl. Slamet	A49-A66	X86	457
		A66-A10	X87	105
31.	Jl. Perintis Kemerdekaan	A54-A74	X88	130
		A74-A75	X89	35
		A75-A76	X90	160
32.	Jl. Angkatan 66	A76-A77	X91	85
		A77-A78	X92	120
		A78-A79	X93	170
33.	Jl. Merdeka	A54-A80	X94	94
		A80-A81	X95	149
		A81-A12	X96	116
34.	Jl. Angkatan 45	A81-A82	X97	245
35.	Jl. Veteran	A82-A83	X98	372
		A83-A84	X99	122
		A84-A85	X100	192
36.	Jl. Tentara Pelajar	A85-A86	X101	289
		A85-A17	X102	192
37.	Jl. Indragiri	A84-A16	X103	216
38.	Jl. Manunggal	A83-A87	X104	321
39.	Jl. Bahagia	A82-A76	X105	187
40.	Jl. Kemakmuran	A82-A14	X106	249
41.	Jl. Diponegoro	A14-A88	X107	162
		A88-A89	X108	192
		A89-A90	X109	222
42.	Jl. Serayu	A15-A88	X110	90
43.	Jl. Barito	A16-A89	X111	102
Jumlah				17,174

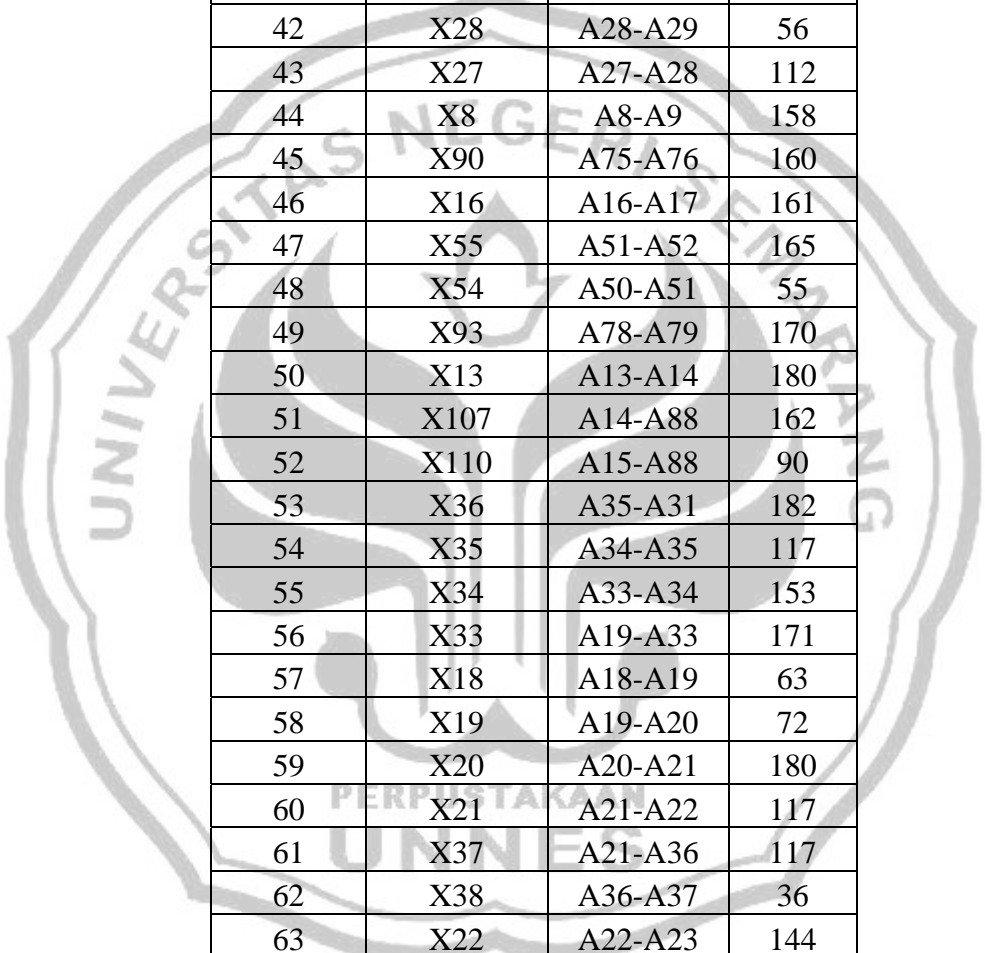






Hasil Iterasi Pencarian Pohon Rentang Minimum dari Jaringan Distribusi Listrik
PLN Kota Pekalongan Wilayah Distribusi Pekalongan 1
dengan Algoritma Prim

Iterasi ke-	Sisi yang terpilih	Titik yang terkait	Bobot
1	X1	A1-A2	90
2	X2	A2-A3	189
3	X3	A3-A4	42
4	X41	A3-A38	108
5	X4	A4-A5	175
6	X5	A5-A6	42
7	X6	A6-A7	154
8	X46	A7-A43	91
9	X47	A43-A44	98
10	X70	A43-A61	126
11	X71	A61-A62	140
12	X72	A62-A63	85
13	X79	A63-A67	39
14	X80	A67-A68	44
15	X81	A68-A69	85
16	X78	A62-A8	135
17	X73	A63-A64	138
18	X82	A64-A70	39
19	X74	A64-A65	99
20	X84	A65-A72	50
21	X85	A72-A73	66
22	X75	A65-A66	94
23	X87	A66-A10	105
24	X83	A70-A71	116
25	X32	A32-A7	144
26	X31	A31-A32	77
27	X10	A10-A11	154
28	X11	A11-A12	121
29	X12	A12-A13	47
30	X96	A81-A12	116
31	X95	A80-A81	148
32	X94	A54-A80	94
33	X58	A54-A55	94



34	X57	A53-A54	120
35	X56	A52-A53	115
36	X88	A54-A74	130
37	X89	A74-A75	35
38	X91	A76-A77	85
39	X92	A77-A78	120
40	X30	A30-A31	154
41	X29	A29-A30	57
42	X28	A28-A29	56
43	X27	A27-A28	112
44	X8	A8-A9	158
45	X90	A75-A76	160
46	X16	A16-A17	161
47	X55	A51-A52	165
48	X54	A50-A51	55
49	X93	A78-A79	170
50	X13	A13-A14	180
51	X107	A14-A88	162
52	X110	A15-A88	90
53	X36	A35-A31	182
54	X35	A34-A35	117
55	X34	A33-A34	153
56	X33	A19-A33	171
57	X18	A18-A19	63
58	X19	A19-A20	72
59	X20	A20-A21	180
60	X21	A21-A22	117
61	X37	A21-A36	117
62	X38	A36-A37	36
63	X22	A22-A23	144
64	X23	A23-A24	90
65	X24	A24-A25	90
66	X15	A15-A16	186
67	X111	A16-A89	101
68	X60	A11-A56	187
69	X105	A82-A76	187
70	X42	A38-A39	189
71	X43	A39-A40	126
72	X44	A40-A41	90

73	X45	A41-A42	63
74	X63	A58-A28	190
75	X68	A9-A59	198
76	X64	A59-A60	167
77	X16	A16-A17	204
78	X102	A85-A17	192
79	X100	A84-A85	192
80	X99	A83-A84	122
81	X62	A57-A58	204
82	X53	A49-A50	209
83	X48	A44-A45	217
84	X49	A45-A46	112
85	X50	A46-A47	84
86	X51	A47-A48	98
87	X109	A89-A90	222
88	X101	A85-A86	289
89	X104	A83-A87	321
Jumlah			11.312

Lampiran 4

