

**OPTIMASI DESAIN JARINGAN VDSL2  
DENGAN METODE HEURISTIK  
ALGORITMA PENCARIAN SCATTER,  
POHON STEINER DAN DIJKSTRA  
PADA JARINGAN TEMBAGA PT. TELKOM**

**TESIS**

Oleh

**ANDRI KURNIA RIYADI**  
**0606003133**



**MANAJEMEN TELKOMUNIKASI  
PROGRAM STUDI TEKNIK ELEKTRO  
PROGRAM PASCA SARJANA BIDANG ILMU TEKNIK  
UNIVERSITAS INDONESIA  
GENAP 2007/2008**

**OPTIMASI DESAIN JARINGAN VDSL2  
DENGAN METODE HEURISTIK  
ALGORITMA PENCARIAN SCATTER,  
POHON STEINER DAN DIJKSTRA  
PADA JARINGAN TEMBAGA PT. TELKOM**

**TESIS**

Oleh

**ANDRI KURNIA RIYADI**  
**0606003133**



**TESIS INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN  
PERSYARATAN MENJADI MAGISTER TEKNIK**

**MANAJEMEN TELKOMUNIKASI  
PROGRAM STUDI TEKNIK ELEKTRO  
PROGRAM PASCA SARJANA BIDANG ILMU TEKNIK  
UNIVERSITAS INDONESIA  
GENAP 2007/2008**

## **PERNYATAAN KEASLIAN TESIS**

Saya menyatakan dengan sesungguhnya bahwa tesis dengan judul :

**OPTIMASI DESAIN JARINGAN VDSL2  
DENGAN METODE HEURISTIK  
ALGORITMA PENCARIAN SCATTER,  
POHON STEINER DAN DIJKSTRA  
PADA JARINGAN TEMBAGA PT. TELKOM**

yang dibuat untuk melengkapi sebagian prasyarat magister teknik pada Program Pendidikan Pasca Sarjana program kekhususan Manajemen Telekomunikasi jurusan Teknik Elektro Fakultas Teknik Universitas Indonesia, bukan merupakan tiruan atau duplikasi dari tesis yang sudah dipublikasikan atau pernah dipakai untuk mendapatkan gelar kemagisteran di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, Juni 2008

**Andri Kurnia Riyadi**  
**NPM : 0606003133**

# PENGESAHAN

Tesis dengan judul :

**OPTIMASI DESAIN JARINGAN VDSL2  
DENGAN METODE HEURISTIK  
ALGORITMA PENCARIAN SCATTER,  
POHON STEINER DAN DJIKSTRA  
PADA JARINGAN TEMBAGA PT. TELKOM**

dibuat untuk melengkapi sebagian persyaratan magister teknik pada Program Pendidikan Pasca Sarjana Jurusan Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian tesis.

Depok, Juni 2008  
Dosen Pembimbing

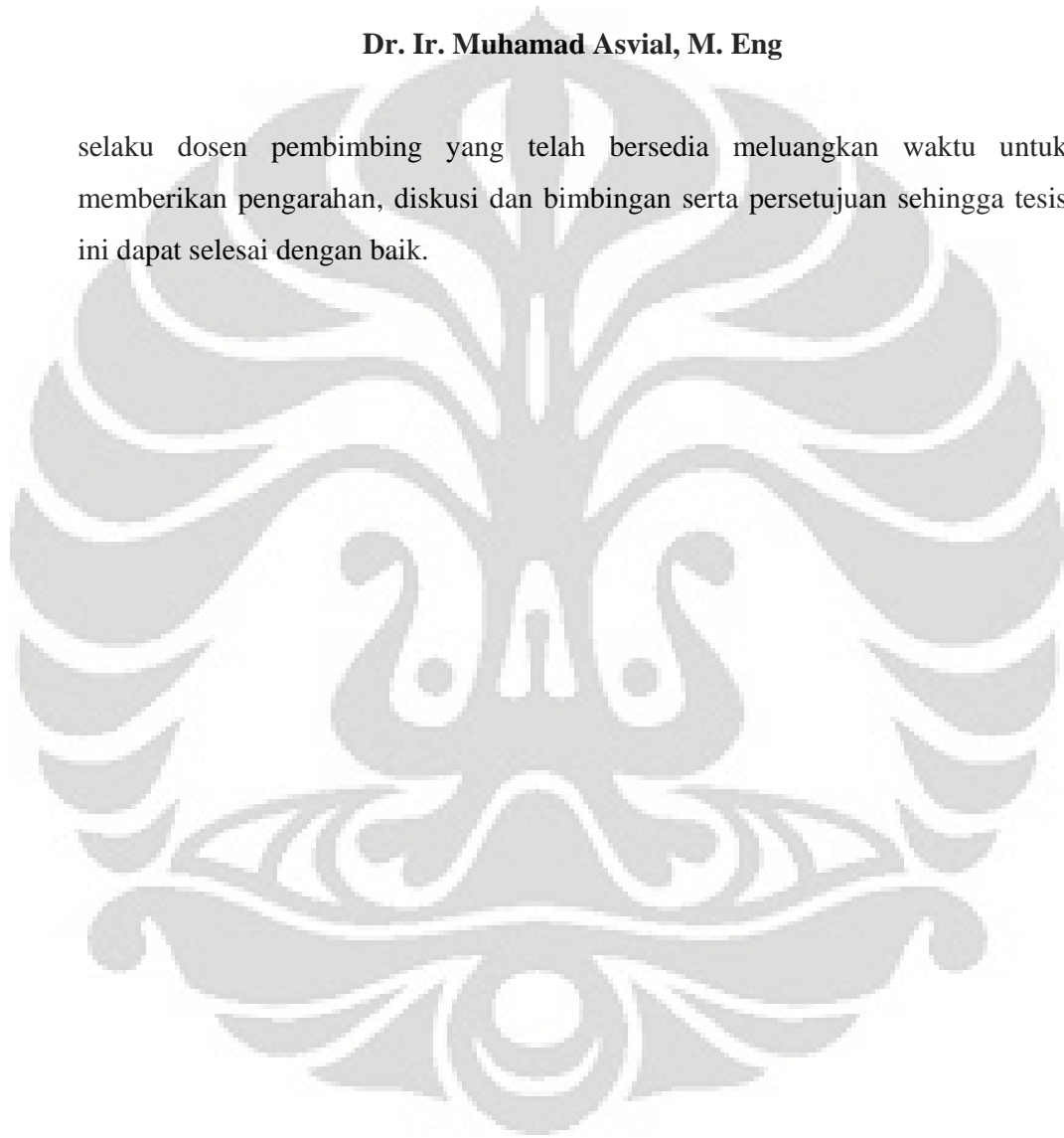
**Dr. Ir. Muhamad Asvial, M. Eng**  
**NIP : 132 094 574**

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

**Dr. Ir. Muhamad Asvial, M. Eng**

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan pengarahan, diskusi dan bimbingan serta persetujuan sehingga tesis ini dapat selesai dengan baik.



Andri Kurnia Riyadi

NPM : 0606003133

Departemen Teknik Elektro

Dosen Pembimbing

Dr. Ir. Muhamad Asvial, M. Eng

**OPTIMASI DESAIN JARINGAN VDSL2 DENGAN METODE  
HEURISTIK ALGORITMA PENCARIAN SCATTER,  
POHON STEINER DAN DIJKSTRA  
PADA JARINGAN TEMBAGA PT. TELKOM**

**ABSTRAK**

Jaringan tembaga POTS PT. Telkom selama ini digunakan dalam menyediakan layanan Telepon, Fax, dan Internet. VDSL2 dapat memberikan suatu solusi jaringan dimana jaringan tembaga POTS yang ada dapat memberikan keleluasaan menggelar layanan yang membutuhkan *Bit Rate* tinggi. Untuk penggelaran jaringan VDSL2, dipilih lokasi rumah – rumah penduduk yang telah tercatat dengan layanan kabel tembaga. Serta lokasi tersebut juga digolongkan daerah pemukiman berskala menengah ke atas. Untuk penelitian ini dipilih daerah pemukiman dibawah STO Slipi.

Desain Jaringan yang optimal harus mempertimbangkan faktor – faktor seperti : kualitas layanan, ketersediaan alat produksi, tidak ruwet serta masuk akal (dapat diwujudkan). Dukungan komputasi algoritma dapat berperan menggantikan analisa manusia, sehingga bisa didapatkan hasil yang lebih cepat dalam memproses parameter – parameter jaringan yang banyak.

Tesis ini ditujukan untuk menciptakan desain jaringan yang mendekati optimal menggunakan algoritma heuristik Pencarian Scatter, Pohon *Steiner* dan Dijkstra. Hasil yang ingin didapatkan adalah antara lain : desain jaringan yang dapat diterapkan pada RA dan RAV, jumlah RK baru, dan letak RK baru.

Pencarian Scatter digunakan untuk membuat pengelompokan pada DP, prinsip pengelompokan ini menggunakan analogi PRKB (Problem Rute Kendaraan Berkapasitas). Setelah pengelompokan terbentuk, kemudian dilanjutkan dengan Pohon *Steiner* untuk membentuk skema cabang pohon. Untuk menyempurnakan jalur kabel agar menyusuri tepi jalan, digunakan algoritma Dijkstra.

**Kata kunci :**

**POTS, Kualitas Layanan, PRKB, Pencarian Scatter, Pohon Steiner, Dijkstra dan VDSL2**

Andri Kurnia Riyadi

NPM : 0606003133

Departemen Teknik Elektro

Dosen Pembimbing

Dr. Ir. Muhamad Asvial, M. Eng

**OPTIMIZING VDSL2 NETWORK DESIGN USING  
HEURISTIC METHOD SCATTER SEARCH,  
STEINER TREE AND DJIKSTRA ALGORITHM  
ON PT. TELKOM COPPER NETWORK**

**ABSTRACT**

PT. Telkom's POTS copper network up to now is commonly used for Telephony, Fax and Internet. VDSL2 is used as network solution where existing POTS copper network can give breathing room for higher Bit Rate services. To deploy VDSL2 , houses location which were already had copper network coverage should be chosen. These location is also classified as middle up dwelling neighbourhood. For this Thesis , dwelling neighbourhood under Central Office Slipi is chosen.

An Optimal Network Design must have QoS, availability of production equipment simplicity and make sense (deployable) as factors of consideration. Computer Algorithm support can be used to replace human analysis , resulting must lesser time in processing so many network parameter.

This Thesis is meant for creating an almost optimal network design using heuristic algorithm Scatter Search, Steiner Tree and Djikstra. Expected results are : network design which can be implemented on RA and RAV, number of new RK and the location of the new RK.

Scatter Search is used to create clusters of customer, the principle of forming these cluster is an analogy of CVRP (Capacitated Vehicle Routing Problem). After the clusters has been established , then followed by Steiner Tree to form tree branch scheme. To perfect the cable course so it surfs the side of the road, Djikstra algorithm is used.

**Keywords :**

**POTS, QoS, CVRP, Scatter Search, Steiner Tree, Djikstra and VDSL2**

## DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN TESIS	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
DAFTAR SINGKATAN	xiv
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 IDENTIFIKASI MASALAH	4
1.3 BATASAN DAN RUMUSAN MASALAH	5
1.4 TUJUAN	6
1.5 METODE PENELITIAN	6
1.6 KERANGKA PENULISAN	7
BAB II VDSL2 DAN ALGORITMA HEURISTIK	9
2.1 KONSEP VDSL2	9
2.2 SKEMA PENGGELARAN PON	12
2.2.1 Konfigurasi PON	12
2.3 ALGORITMA PENCARIAN SCATTER	13
2.4 ALGORITMA POHON STEINER	15
2.5 ALGORITMA DIJKSTRA	15
BAB III PERENCANAAN APLIKASI DESAIN JARINGAN	18
3.1 PEMETAAN TITIK DP DAN TITIK JALAN	18



3.1.1	Simbol Matematis Jaringan Akses Untuk DP dan Titik Jalan	18
3.1.2	Mencatat Koordinat Titik Jalan dan Titik DP pada File Excel	20
3.2	PENCARIAN TITIK RK BARU	21
3.3	PENCARIAN SCATTER	23
3.4	POHON STEINER	25
3.4.1	Pencarian Sudut Antara Tiga Titik	26
3.4.2	Menentukan Titik Steiner	27
3.4.3	Penentuan Solusi Steiner	28
3.4.3.1	Matrik Steiner Atas	29
3.4.3.2	Matrik Steiner Atas	29
3.4.3.3	Matrik Steiner Bawah	30
3.4.4	Update Matrik Steiner	31
BAB IV ANALISA JARINGAN VDSL2 HASIL DESAIN APLIKASI		35
4.1	HASIL DESAIN APLIKASI	35
4.2	TOPOLOGI JARINGAN GEPON	37
4.2.1	Alat Produksi GEPON	37
4.2.2	Penggelaran GEPON Pada STO Slipi	39
4.3	PEMENUHAN KEBUTUHAN ANTARLUKA PON	40
4.3.1	Pembagi Pasif	41
4.3.2	Pemetaan Inti Optik Pada Antarmuka PON	41
4.3.2.1	Jatah Inti Optik Untuk RA	42

4.3.2.2 Jatah Inti Optik Untuk RAV	43
4.3.2.3 Jumlah Pembagi Pasif Pada Antarmuka PON	44
4.4 EFEKTIVITAS PEMAKAIAN ALAT PRODUKSI	46
4.5 PEMAKAIAN ALAT PRODUKSI LAMA	47
4.6 BIAYA PENGGELARAN GEAPON	47
BAB V KESIMPULAN	48
DAFTAR ACUAN	49
LAMPIRAN	50
RIWAYAT PENULIS	155



## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b>	Kinerja VDSL2 [7] 10
<b>Gambar 2.2</b>	Konfigurasi Jaringan PSTN dan VDSL2 11
<b>Gambar 2.3</b>	Contoh Skema Kabel Primer dan Sekunder RAV Slipi 11
<b>Gambar 2.4</b>	Modus Aplikasi FTTC 12
<b>Gambar 2.5</b>	Topologi PON 13
<b>Gambar 2.6</b>	Konfigurasi PON 13
<b>Gambar 2.7</b>	Contoh Pengelompokkan pada Algoritma Pencarian Scatter 14
<b>Gambar 2.8</b>	Algoritma Penambahan Titik Steiner 15
<b>Gambar 2.9</b>	Graph Pada Djikstra 16
<b>Gambar 3.1</b>	Diagram Alir Penentuan Titik RK 21
<b>Gambar 3.2</b>	Letak RK yang sudah ada sebelumnya dan DP 22
<b>Gambar 3.3</b>	Menentukan Titik RK Baru 22
<b>Gambar 3.4</b>	Kabel Primer dan Kabel Sekunder 22
<b>Gambar 3.5</b>	Diagram Alir dari Algoritma Pencarian Scatter 24
<b>Gambar 3.6</b>	Gambar Pengelompokkan Titik Sesuai Sudut Polar RK 25
<b>Gambar 3.7</b>	Diagram Alir Algoritma Pohon Steiner 26
<b>Gambar 3.8</b>	Menghitung Sudut pada 3 titik 27
<b>Gambar 3.9</b>	Menentukan titik steiner "d" 27
<b>Gambar 3.10</b>	Solusi Steiner pada 3 titik 28
<b>Gambar 3.11</b>	Solusi Steiner pada badan utama aplikasi 28
<b>Gambar 3.12</b>	Diagram Alir Matrik Steiner Atas 29
<b>Gambar 3.13</b>	Diagram Alir Matrik Steiner Tengah 30
<b>Gambar 3.14</b>	Diagram Alir Matrik Steiner Bawah 31
<b>Gambar 3.15</b>	Pohon Steiner pada Jaringan 33
<b>Gambar 3.16</b>	Kabel Primer dan Kabel Sekunder Pada Jaringan 34
<b>Gambar 4.1</b>	Topologi Jaringan GEAPON 37
<b>Gambar 4.2</b>	Penggelaran GEAPON + VDSL2 pada STO Slipi 40
<b>Gambar 4.3</b>	Pembagi Pasif 41

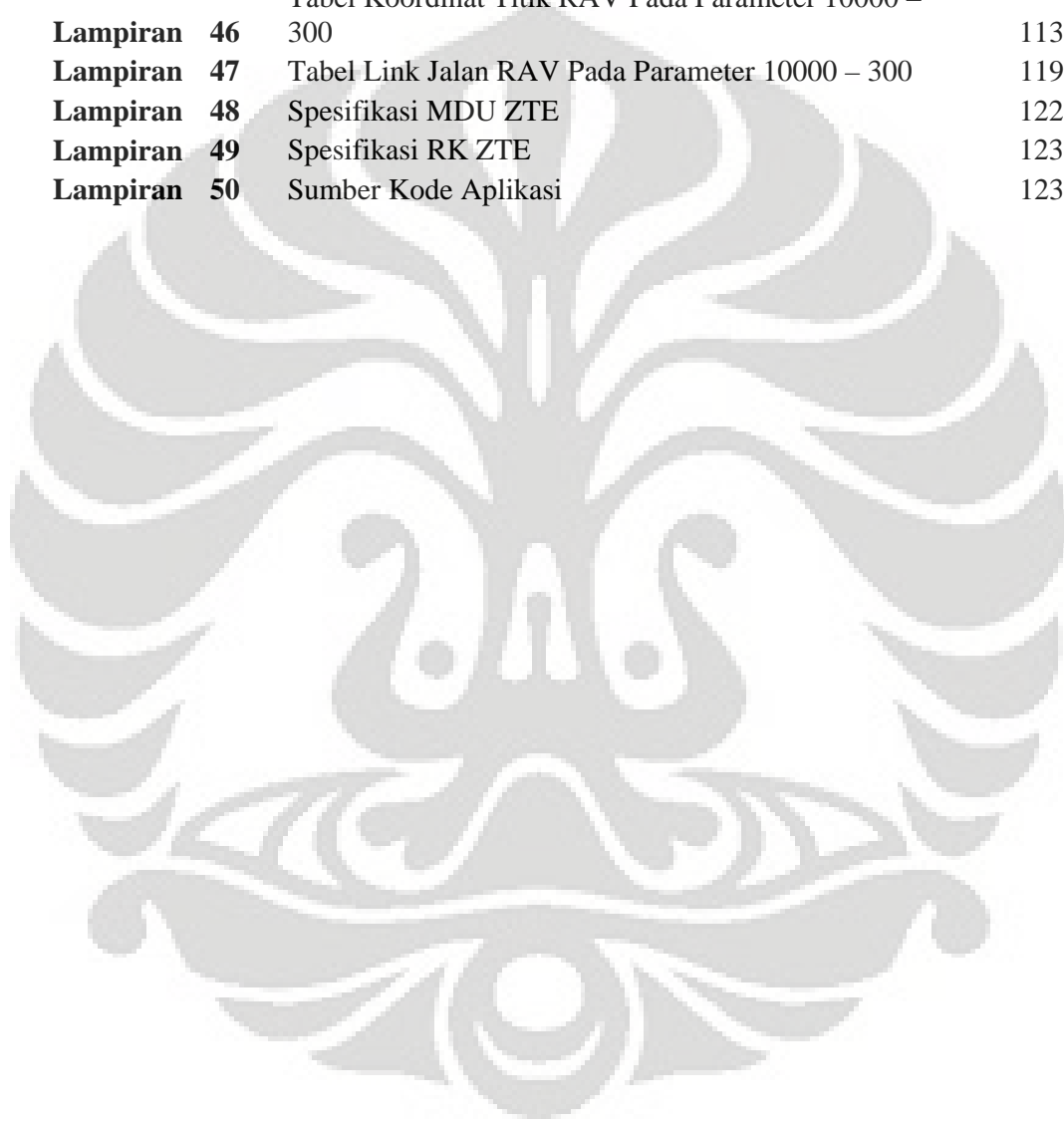
## DAFTAR TABEL

			Halaman
<b>Tabel</b>	<b>2.1</b>	Kebutuhan Arus Hilir dan Mudik Layanan Multimedia [6]	9
<b>Tabel</b>	<b>2.2</b>	Pencarian Rute Dijkstra	16
<b>Tabel</b>	<b>3.1</b>	Penulisan Koordinat DP dan Titik Jalan	20
<b>Tabel</b>	<b>3.2</b>	Penulisan Ruas Jalan	20
<b>Tabel</b>	<b>3.3</b>	Matrik Steiner	31
<b>Tabel</b>	<b>3.4</b>	Solusi Steiner Pada Matrik Steiner	32
<b>Tabel</b>	<b>4.1</b>	Hasil Kinerja Aplikasi	35
<b>Tabel</b>	<b>4.2</b>	Paket Alat Produksi NEC Untuk GEPON	38
<b>Tabel</b>	<b>4.3</b>	Kebutuhan Bit Rate RA dan RAV	40
<b>Tabel</b>	<b>4.4</b>	Pemetaan Inti Optik Untuk Catuan RA	42
<b>Tabel</b>	<b>4.5</b>	Pemetaan Inti Optik Untuk Catuan RAV	43
<b>Tabel</b>	<b>4.6</b>	Antarmuka PON Pada OLT	44
<b>Tabel</b>	<b>5.1</b>	Efektivitas Bit Rate RA	47
<b>Tabel</b>	<b>5.2</b>	Efektivitas Bit Rate RAV	47

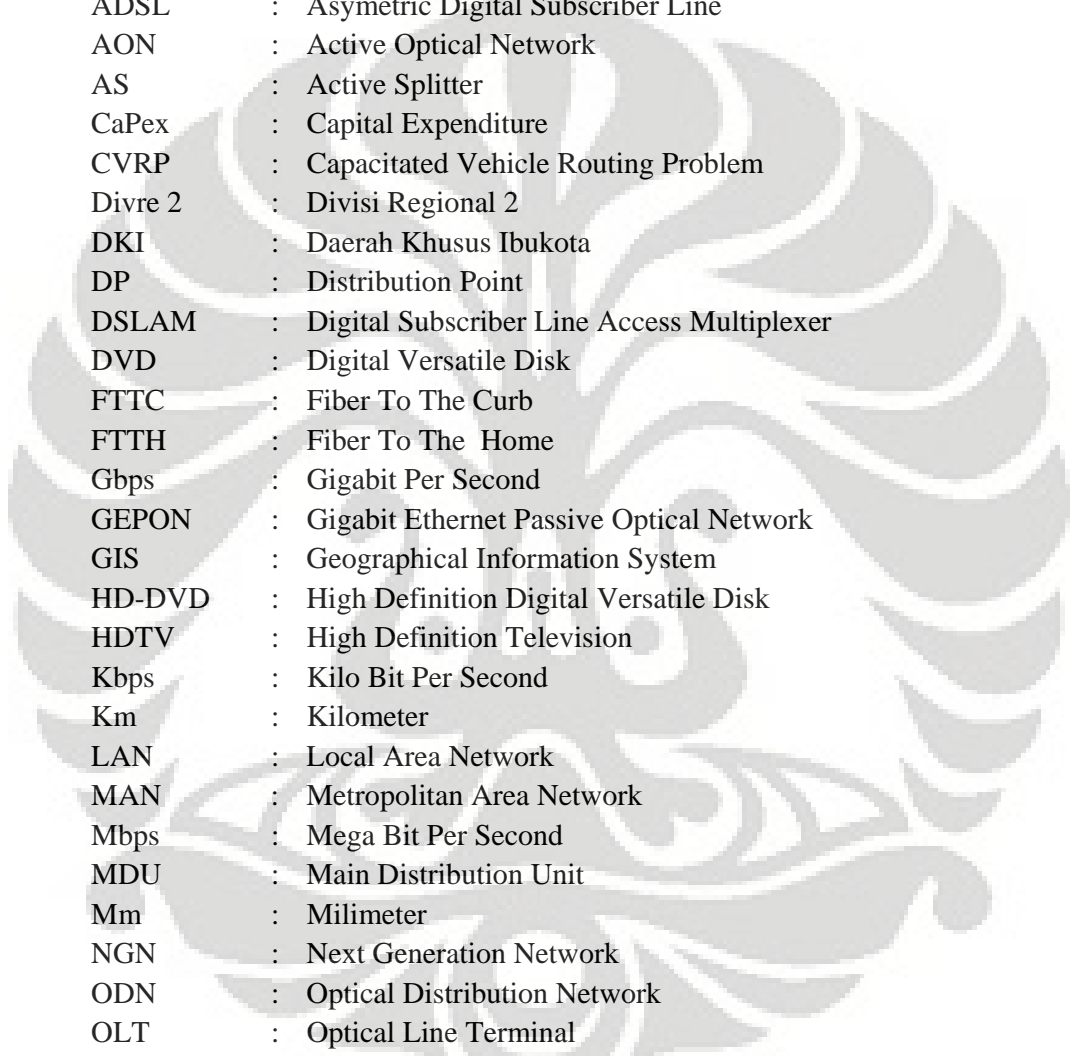
## DAFTAR LAMPIRAN

		Halaman
<b>Lampiran 1</b>	Gambar RX Sebagai RK Model	50
<b>Lampiran 2</b>	Tabel Koordinat Titik Jalan Untuk RX	50
<b>Lampiran 3</b>	Tabel Link Jalan Untuk RX	53
<b>Lampiran 4</b>	Tabel Koordinat DP Untuk RX	56
<b>Lampiran 5</b>	Tabel Kebutuhan Bit Rate DP Untuk RX	57
<b>Lampiran 6</b>	Tabel Link Menyeberang Jalan Untuk RX	60
<b>Lampiran 7</b>	Tabel Link Keluar Sentral Untuk RX	61
<b>Lampiran 8</b>	Gambar RA (RK pada STO SLIPI)	62
<b>Lampiran 9</b>	Tabel Koordinat Titik Jalan RA	63
<b>Lampiran 10</b>	Tabel Link Jalan Untuk RA	66
<b>Lampiran 11</b>	Tabel Koordinat Titik DP Untuk RA	69
<b>Lampiran 12</b>	Tabel Kebutuhan Bit Rae DP Untuk RA	71
<b>Lampiran 13</b>	Tabel Link Menyeberang Jalan Untuk RA	72
<b>Lampiran 14</b>	Tabel Link Keluar Sentral Untuk RA	73
<b>Lampiran 15</b>	Gambar RAV (RK pada STO SLIPI)	74
<b>Lampiran 16</b>	Tabel Koordinat Titik Jalan Untuk RAV	75
<b>Lampiran 17</b>	Tabel Link Jalan Untuk RAV	78
<b>Lampiran 18</b>	Tabel Koordinat Titik DP Untuk RAV	81
<b>Lampiran 19</b>	Tabel Kebutuhan Bit Rate Untuk RAV	82
<b>Lampiran 20</b>	Tabel Link Menyeberang Jalan Untuk RAV	83
<b>Lampiran 21</b>	Tabel Link Keluar Sentral Untuk RAV	85
<b>Lampiran 22</b>	Gambar Logik RX Pada Parameter 10000 - 500	85
<b>Lampiran 23</b>	Gambar Jaringan RX Pada Parameter 10000 - 500	86
<b>Lampiran 24</b>	Gambar Logik RX Pada Parameter 10000 - 400	86
<b>Lampiran 25</b>	Gambar Jaringan RX Pada Parameter 10000 - 400	87
<b>Lampiran 26</b>	Gambar Logik RX Pada Parameter 10000 - 300	87
<b>Lampiran 27</b>	Gambar Jaringan RX Pada Parameter 1000 - 300	88
<b>Lampiran 28</b>	Gambar Logik RA Pada Parameter 10000 - 500	88
<b>Lampiran 29</b>	Gambar Jaringan RA Pada Parameter 10000 - 500	89
<b>Lampiran 30</b>	Gambar Logik RA Pada Pameter 10000 - 200	89
<b>Lampiran 31</b>	Gambar Jaringan RA Pada Pameter 10000 -200	90
<b>Lampiran 32</b>	Gambar Logik RA Pada Parameter 10000 - 100	90
<b>Lampiran 33</b>	Gambar Jaringan RA Pada Parameter 10000 - 100	91
<b>Lampiran 34</b>	Tabel Report Jaringan RA Pada Parameter 10000 - 200	91
<b>Lampiran 35</b>	Tabel Kebutuhan Catuan RA	94
<b>Lampiran 36</b>	Tabel Koordinat Semua Titik Pada RA	95
<b>Lampiran 37</b>	Tabel Link Jalan RA Parameter 10000 - 200	103
<b>Lampiran 38</b>	Gambar Logik RAV Pada Parameter 10000 - 500	108
<b>Lampiran 39</b>	Gambar Jaringan RAV Pada Parameter 10000 - 500	108

<b>Lampiran 40</b>	Gambar Logik RAV Pada Parameter 10000 – 400	109
<b>Lampiran 41</b>	Gambar Jaringan RAV Pada Parameter 10000 – 400	109
<b>Lampiran 42</b>	Gambar Logik RAV Pada Parameter 10000 – 300	110
<b>Lampiran 43</b>	Gambar Jaringan RAV Pada Parameter 10000 – 300	110
<b>Lampiran 44</b>	Tabel Report RAV Pada Parameter 10000 – 300	111
<b>Lampiran 45</b>	Tabel Kebutuhan Catu RAV Pada Parameter 10000 – 300	113
<b>Lampiran 46</b>	Tabel Koordinat Titik RAV Pada Parameter 10000 – 300	113
<b>Lampiran 47</b>	Tabel Link Jalan RAV Pada Parameter 10000 – 300	119
<b>Lampiran 48</b>	Spesifikasi MDU ZTE	122
<b>Lampiran 49</b>	Spesifikasi RK ZTE	123
<b>Lampiran 50</b>	Sumber Kode Aplikasi	123

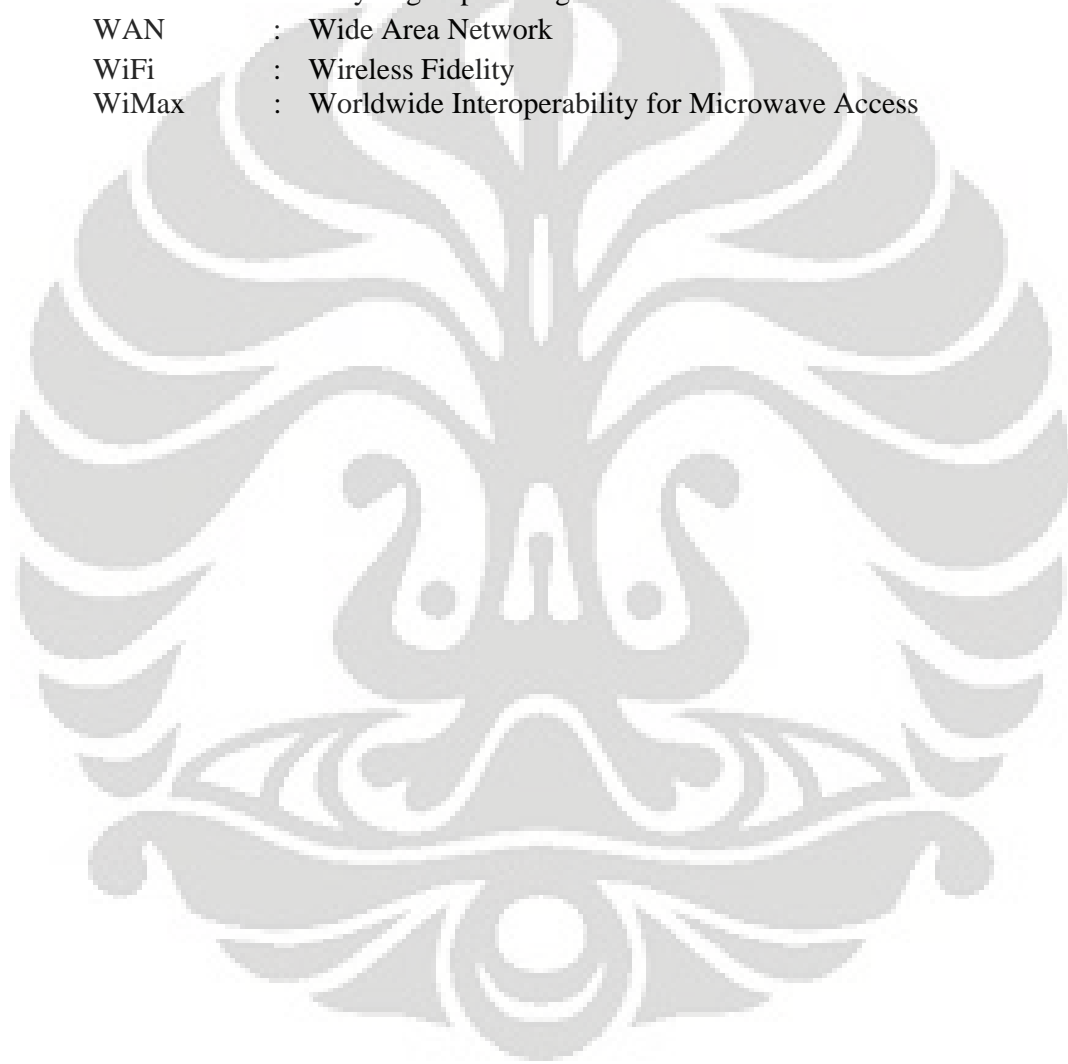


## DAFTAR SINGKATAN



2.5G	: Second and a Half Generation
2G	: Second Generation
3G	: Third Generation
4G	: Fourth Generation
ADSL	: Asymmetric Digital Subscriber Line
AON	: Active Optical Network
AS	: Active Splitter
CaPex	: Capital Expenditure
CVRP	: Capacitated Vehicle Routing Problem
Divre 2	: Divisi Regional 2
DKI	: Daerah Khusus Ibukota
DP	: Distribution Point
DSLAM	: Digital Subscriber Line Access Multiplexer
DVD	: Digital Versatile Disk
FTTC	: Fiber To The Curb
FTTH	: Fiber To The Home
Gbps	: Gigabit Per Second
GEPON	: Gigabit Ethernet Passive Optical Network
GIS	: Geographical Information System
HD-DVD	: High Definition Digital Versatile Disk
HDTV	: High Definition Television
Kbps	: Kilo Bit Per Second
Km	: Kilometer
LAN	: Local Area Network
MAN	: Metropolitan Area Network
Mbps	: Mega Bit Per Second
MDU	: Main Distribution Unit
Mm	: Milimeter
NGN	: Next Generation Network
ODN	: Optical Distribution Network
OLT	: Optical Line Terminal
ONT	: Optical Network Termination
ONU	: Optical Network Unit
PBS	: Problem Berkeliling Sales
POTS	: Plain Old Telephony Service
PRKB	: Problem Rute Kendaraan Berkapasitas
PS	: Passive Splitter
PSTN	: Public Switched Telephony Network
PT	: Perseroan Terbatas

QoS	: Quality Of Service
RA	: Rumah Kabel A
R AV	: Rumah Kabel AV
RK	: Rumah Kabel
STO	: Sentral Telepon Otomat
TDM	: Time Division Multiplexing
TKO	: Titik Konversi Optik
TSP	: Travelling Salesman Problem
VDSL2	: Very High Speed Digital Subscriber Line 2
WAN	: Wide Area Network
WiFi	: Wireless Fidelity
WiMax	: Worldwide Interoperability for Microwave Access





# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Sesuai dengan perkembangan teknologi, jaringan telekomunikasi akan mempunyai tahapan – tahapan dalam proses evolusinya. Misalkan di jaringan Seluler GSM kita kenal beberapa tahapan evolusi seperti : 2G, 2,5G, 3G, 4G , di jaringan komputer ada evolusi LAN, WAN, MAN, WiFi, WiMax , dan di jaringan PSTN ada : TDM dan NGN. Evolusi jaringan telekomunikasi akan berjalan dengan kecepatan yang berbeda di beberapa negara. Di Indonesia sendiri, evolusi jaringan telekomunikasi bisa di bilang tertinggal dari beberapa negara tetangga.

Dengan tersebarnya jaringan tembaga PSTN Telkom di seluruh Indonesia, merupakan aset yang berharga namun dirasa kurang diminati karena hanya memberikan fasilitas suara saja (telepon). Untuk beberapa tempat lain, *bundling* suara dan internet dengan tambahan modem ADSL memang cukup menjanjikan. Tapi sampai kapan ? Fitur jaringan Seluler sudah sedemikian maju dengan ragam jenis konten yang disediakan. Apabila kita bandingkan dengan situasi di luar negeri, PSTN sudah menjelma menjadi jaringan NGN yang serba bisa dimana selain suara dan Internet, juga ada layanan HDTV, Game Online, dan *Fixed Mobile Convergencenya*.

Kondisi ADSL yang ada sekarang pun sebenarnya tidak sesuai dengan standar yang seharusnya. ADSL selayaknya mampu untuk memberikan *Bit Rate* sampai dengan 8 Mbps untuk arus hilirnya. Hal ini disebabkan beberapa faktor yaitu : letak DSLAM yang berjarak kurang lebih 5 km dari pelanggan dan penampang kabel yang hanya 0.6 mm. Akhirnya kualitas layanan yang didapatkan pelanggan hanya 384 Kbps.

Untuk ujung akhir tujuan dari jaringan PSTN , adalah FTTH tergelar sampai ke rumah – rumah. Namun investasi jaringan seperti ini sangat mahal dan belum bisa di terapkan di Indonesia. Sementara ini kita bisa memberdayakan jaringan tembaga yang ada, yang ditambahkan layanan VDSL2 untuk menghantarkan layanan yang setara dengan layanan NGN sampai ke rumah – rumah.

Ada dua arus besar dalam menghantarkan layanan NGN sampai ke rumah – rumah. Yang pertama adalah arus Jaringan Inti, Jaringan Inti adalah segala jenis Elemen Jaringan yang dimana setiap Elemen Jaringan ini ambil bagian dalam menyediakan dan memberikan konten layanan – layanan kepada pelanggan. Jaringan Inti terdiri atas : Managemen NGN , Kontrol NGN dan Inti NGN. Sedangkan untuk menghantarkan layanan NGN sampai ke rumah - rumah , ada arus Akses NGN.

Dalam merencanakan layanan VDSL2 harus memperhitungkan banyak sekali parameter. Diantaranya efektifitas desain jaringan dikaitkan dengan keterhubungan semua titik , kemampuan jaringan menyediakan kecepatan transmisi ke pelanggan, letak jarak MDU dari modem pelanggan dan lain – lain.

Titik rumah pelanggan yang bisa berjumlah ratusan dalam satu cakupan STO dengan letak yang bervariasi akan menyulitkan untuk menentukan desain jaringan yang paling optimal. Kemungkinan kombinasi antara titik dan link yang dapat di gelar bisa berjumlah ribuan kemungkinan. Secara konsep kertas seorang desainer dapat menghitung dan merencanakan jaringannya, namun manusia mempunyai keterbatasan dalam kecepatan dan ketepatan hitungan.

Atau dengan kata lain bisa dikatakan sebagai berikut : apabila suatu desain jaringan dilakukan oleh manusia , kemungkinan untuk mendapatkan desain yang tidak efektif akan semakin besar. Diperlukan bantuan komputer yang dilengkapi algoritma untuk menyelesaikan masalah kombinasional ini untuk mencapai solusi yang mendekati optimal. Kenapa mendekati optimal dan bukan paling optimal, karena problem yang dihadapi adalah kombinasi yang begitu besar dan begitu banyak kemungkinan. Iterasi yang dibutuhkan untuk menghasilkan solusi yang paling optimal mungkin hanya bisa dicari setelah jutaan iterasi apabila problem kombinasional yang dihadapi terdiri dari ribuan titik. Oleh sebab itu mendekati optimal adalah harga tawar yang paling baik.

Heuristik secara bahasa artinya : Suatu fungsi yang menurunkan fungsi yang lain dimana fungsi tersebut dapat menjawab persoalan dengan memberikan solusi. Apabila dikaitkan dengan algoritma komputer, berarti suatu aplikasi yang mempunyai seperangkat cara untuk menghasilkan solusi awal , lalu kemudian aplikasi tersebut akan menurunkan solusi berikutnya dengan prinsip dan metode –

metode optimasi khusus yang terkandung dalam aplikasi tersebut. Solusi akhir adalah solusi terbaik yang pernah dicapai oleh aplikasi tersebut dengan membandingkan solusi awal sampai akhir.

Pencarian *Scatter* adalah metode heuristik yang dikembangkan oleh Fred Glover pada tahun 1977 [1] untuk memecahkan masalah - masalah optimasi diskrit. Solusi referensi berfungsi untuk memproduksi calon solusi yang baru yang berbeda dengan solusi referensi awalnya. Metode ini didasarkan atas prinsip untuk menangkap informasi relevan yang pada solusi baru yang dibutuhkan untuk diintegrasikan menjadi solusi referensi yang baru. Hal ini dilakukan untuk menyusun solusi dari bagian yang berbeda - beda untuk mencapai solusi terakhir yang mendekati optimal.

Problem Berkeliling Sales (PBS) [2] adalah bentuk problem kombinasional dimana seorang salesman di harapkan berkeliling kota untuk mengantar paket ke sejumlah pelanggan dalam waktu dan jarak yang paling efisien. Kapasitas angkut dalam kendaraan yang dipakai akan membuat salesman harus kembali ke depot, sehingga rute yang ditempuh merupakan rute melingkar dimulai dan diakhiri dari dan ke depot. Hal ini disebut juga Problem Rute Kendaraan Berkapasitas (PRKB).

Dengan asumsi problem PRKB di atas, problem dalam dunia telekomunikasi dapat di analogikan kepada konsep yang sama. Apabila kita ambil contoh pelanggan dari VDSL2 membutuhkan transmisi *bit rate* A mbps dengan kemampuan fiber optik kabel sekunder B mbps maka rute melingkar (ring) dari STO ke pelanggan berjumlah  $M = (B/A)$ . Namun apabila dibuat jarak kabel sebagai salah satu parameter optimasi maka hal di atas belum tentu optimal. Untuk mendapatkan solusi yang mendekati optimal maka Pencarian Scatter dapat digunakan.

Setelah M lingkaran terbentuk dari hasil optimasi Pencarian *Scatter*, kita akan dihadapkan kepada masalah bagaimana merubah rute ke pelanggan menjadi berbentuk diagram cabang. Untuk mengoptimalkan Desain Jaringan Fiber kita, selanjutnya digunakan metode Pohon *Steiner* yang juga merupakan sebuah algoritma heuristik untuk menentukan arah pencabangan kabel optik yang kita desain.

Optimasi Desain jaringan VDSL2 dengan menggunakan metode – metode heuristik diharapkan dapat membuat efisiensi dalam skema jaringan, manajemen alat produksi, rute yang efektif , *duct* dan RK yang ada.

Dengan demikian, bentuk desain jaringan akses dapat dirancang oleh komputer dengan menggunakan bantuan algoritma heuristik. Dimana informasi letak jalan dan DP adalah inputan , dan kemudian proses algoritma heuristik membentuk desain jaringan VDSL2 yang mendekati optimal.

Dalam penelitian sebelum – sebelumnya pernah dicoba oleh Thomas Bucsics pada tahun 2002 [11] dimana dipakai cara heuristik untuk merancang jaringan fiber yang handal sampai ke rumah – rumah. Bucsics menekankan pada penggunaan beberapa metode heuristik untuk penghitungan komputasionalnya. Metode – metode yang digunakan adalah *Minimum Spanning Tree*, *Pohon Steiner*, *Local Search* dan *Simulated Annealing*. Thomas Bucsics langsung memfokuskan pada pencabangan yang efektif , lebih ke arah *Pohon Steiner* dan tidak menggunakan PRKB. Untuk riset yang asalnya dari Indonesia , sampai saat ini penulis hanya bisa mengumpulkan 2 abstrak tesis sebagai berikut :

- I Nyoman Sutapa, I Gede Agus Widyadana, Christine, “STUDI TENTANG *TRAVELLING SALESMAN* DAN *VEHICLE ROUTING PROBLEM* DENGAN *TIME WINDOWS*”
- Ira Prasetyaningrum, “Penyelesaian Kombinasi *Vehicle Routing Problem* dan *Container Loading Problem* Menggunakan Algoritma Genetika”

Dimana pada kedua penelitian tersebut lebih menggunakan prinsip PRKB yang diselesaikan dengan algoritma genetika.

## 1.2 IDENTIFIKASI MASALAH

Sebelum perencanaan jaringan tembaga dibuat perlu dikumpulkan data – data yang berisi koordinat titik – titik dan jarak kabel antara Rumah – DP – RK yang sudah ada. Data tentang koordinat titik dan jalan akan menjadi pertimbangan untuk menentukan letak dari RK yang baru. Untuk letak DP dan STO tidak dirubah untuk meminimalkan perubahan letak elemen jaringan akses. Untuk

masalah optimasi letak RK baru, perlu mempertimbangkan kedekatan letak DP – DP , dimana nantinya letak DP – DP tersebut dapat dijadikan acuan untuk menentukan letak RK yang baru.

Ketika algoritma Pohon *Steiner* diterapkan dan menghasilkan bentuk pencabangan yang efektif , maka perlu adanya pemetaan dari jalan– jalan yang ada di dalam kota agar pencabangan dari fiber optik (kabel primer) yang dibuat dapat mengikuti bentuk dan alur jalan. Untuk itu digunakan algoritma Dijkstra untuk mencari alur jalan dari titik ke titik.

### 1.3 BATASAN DAN RUMUSAN MASALAH

Ruang lingkup atau batasan dari perencanaan ini hanya menyangkut hal-hal berikut :

- Analisa hanya dilakukan pada STO Telkom Slipi Tandem yang ada di Jakarta dengan mengambil contoh 2 RK dan DP - DP yang dinaungi oleh Kedua RK tersebut.
- Hasil rancangan Jaringan baru, tidak mempermasalahkan Inti Jaringan NGN, Inti Jaringan dianggap telah siap untuk memberikan layanan dengan standar kualitas konten layanan yang prima .
- Dianggap bahwa 10 pelanggan dari suatu DP yang ada sekarang menginginkan jaringan VDSL2 tergelar sampai kerumah.
- Tidak melakukan perhitungan analisa kelayakan investasi, penelitian hanya mempertimbangkan bagaimana menggunakan alat produksi yang efektif ketika menggelar jaringan VDSL2.

Dari batasan-batasan masalah yang dikemukakan diatas, maka didapat rumusan masalah sebagai berikut :

- Rumah, gedung dan bangunan sebagai bidang yang harus dihindari karena merupakan halangan. Nantinya jaringan yang kita desain tidak boleh menabrak bidang bangunan atau bidang jalan, namun harus mengikuti dan menelusuri bidang bangunan dan jalan tersebut.

- Diperlukan beberapa tahapan proses dari Algoritma Heuristik untuk mendapatkan desain yang ideal dan mendekati optimal.
- Pengolahan data dilakukan dalam bentuk Matriks , baik data koordinat maupun data parameter yang diolah dalam aplikasi. Diperlukan beberapa bentuk Matriks yang disesuaikan dengan kebutuhan masing – masing algoritma dan fungsi – fungsi dalam algoritma tersebut.

#### 1.4 TUJUAN

Adapun tujuan dari tesis ini adalah :

- Membuat aplikasi dengan MatLab yang dapat mendesain jaringan VDSL2 yang siap diimplementasikan pada area layanan STO Slipi khususnya pada daerah pemukiman Rumah Kabel RA dan RAV.
- Desain yang mendekati optimal diharapkan bisa memberikan laporan estimasi hitungan, seberapa panjang kabel fiber optik yang dibutuhkan untuk menggantikan kabel primer tembaga yang diganti pada area yang akan digelar jaringan VDSL2.
- Desain yang optimal diharapkan bisa memberikan gambaran alternatif rencana kabel *duct* dan RK, berapa persen *duct* dan RK lama akan digunakan untuk menunjang jaringan VDSL2 yang baru. Sehingga apabila jaringan akan digelar, penggunaan alat produksi seperti RK baru dan sistem *duct* baru dapat seefisien mungkin.
- Dapat menghitung jumlah pencabangan fiber optik menggunakan pembagi pasif pada Antarmuka GEPON yang digunakan.
- Hasil akhir berupa gambar desain jaringan yang merupakan hasil desain ulang jaringan kabel tembaga menjadi jaringan VDSL2 pada daerah pemukiman Rumah Kabel RA dan RAV. Desain jaringan ini dapat memberikan referensi kepada PT. Telkom untuk memutuskan ya / tidaknya beralih menuju jaringan VDSL2 sebagai pengganti jaringan ADSL yang ada.

## 1.5 METODE PENELITIAN

Penelitian bersifat *eksploratif*, dimana koordinat elemen jaringan akses yang di dapat dari peta jaringan kabel Divre 2 PT. Telkom digunakan sebagai sumber penelitian. Adapun koordinat yang diperlukan adalah : Koordinat DP, Koordinat titik jalan, Koordinat RK, dan link ruas kabel.

1. Studi kepustakaan seputar buku, *white paper*, laporan, serta jurnal-jurnal penelitian dan kajian sebelumnya yang berhubungan dengan :
  - Teori Pencarian *Scatter* Heuristik untuk kasus PRKB.
  - Teori Pohon *Steiner* Heuristik untuk kasus desain pengkabelan jaringan.
  - Teori Dijkstra untuk hubungan titik ke titik.
  - Teori vektor dan sudut pada matematika.
2. Analisis persoalan, yaitu rangkaian kegiatan untuk mengidentifikasi skema jaringan hasil sintesis dari Algoritma dan menguji (*verifikasi*) desain jaringan , apakah desain tersebut dapat diwujudkan.
3. Sebagai hasil akhir dari perencanaan ini adalah dibuatnya laporan tertulis dari keseluruhan proses perencanaan.

## 1.6 KERANGKA PENULISAN

### BAB I. Pendahuluan

Berisi latar belakang, identifikasi masalah, batasan dan rumusan masalah, tujuan dan metode penelitian pada tulisan ini.

### BAB II Konsep dan Teori Pendukung

Konsep VDSL2, PON, Teori Pencarian *Scatter*, Teori Pohon *Steiner*

### BAB III Metodologi Penelitian

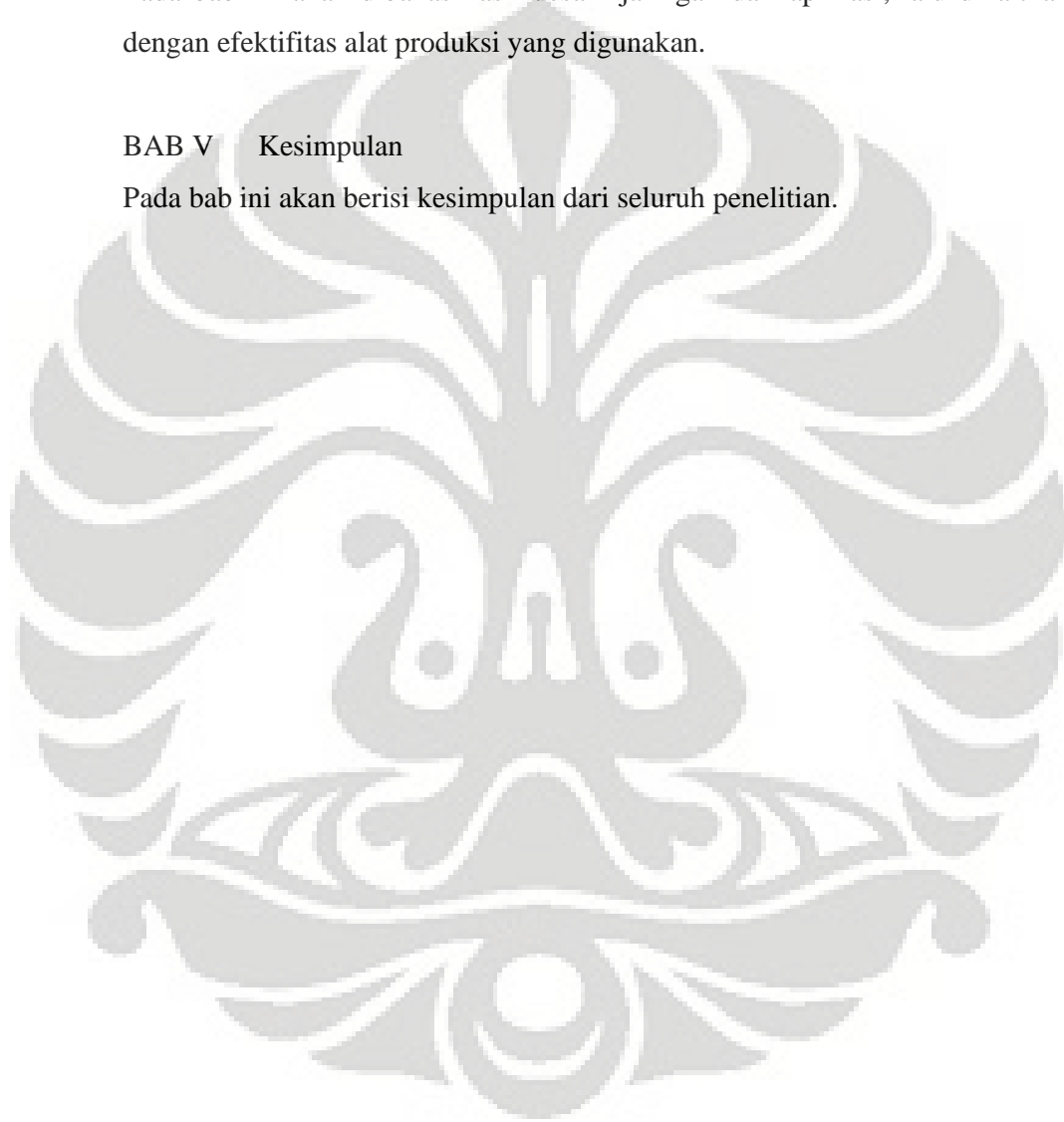
Pada bab ini akan dijelaskan metode pendekatan yang akan digunakan dalam pengolahan data dan desain cara kerja algoritma yang dipakai untuk menghasilkan desain jaringan.

#### BAB IV Analisa Hasil

Pada bab ini akan dibahas hasil desain jaringan dari aplikasi, lalu dikaitkan dengan efektifitas alat produksi yang digunakan.

#### BAB V Kesimpulan

Pada bab ini akan berisi kesimpulan dari seluruh penelitian.





## BAB II

### VDSL2 DAN ALGORITMA HEURISTIK

#### 2.1 KONSEP VDSL2

NGN akan mempunyai layanan konten yang bervariasi dan mengandalkan transmisi *Bit Rate* yang tinggi dalam prakteknya. Semua layanan akan berbasis data dan tidak ada lagi perbedaan layanan, semua di treatment berbasis data. Oleh sebab itu suara tidak terlalu signifikan karena hanya akan mengenerate kebutuhan transmisi *Bit Rate* yang kecil dibandingkan layanan yang lain. Adapun kebutuhan *Bit Rate* masing – masing rumah kurang lebih sebagai berikut :

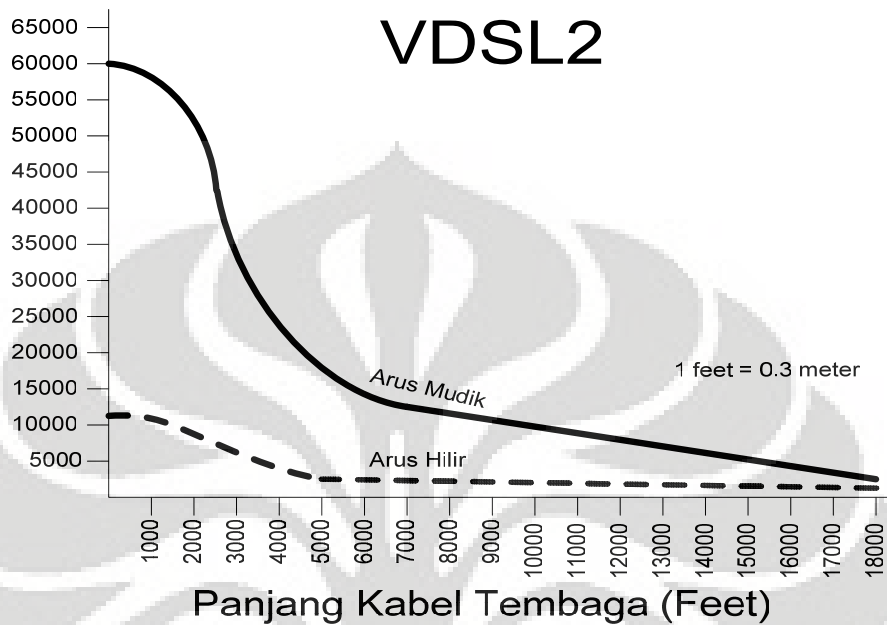
Tabel 2.1 Kebutuhan Arus Hilir dan Mudik Layanan Multimedia [6]

Service	Downstream Bandwidth Required	Upstream Bandwidth Required
HDTV (compressed) (2-3 sets per household)	6-8 Mbps per TV channel	0.5 Mbps per TV channel
High-speed Internet (upstream and downstream)	10-30 Mbps	1 to 5 Mbps
IP voice telephony	0.5 Mbps	0.5 Mbps
Video Conferencing	2 to 5 Mbps	2 to 5 Mbps
On-line Gaming	1 to 3 Mbps	1 to 3 Mbps
<b>Total Bandwidth Needed</b>	<b>19.5 to 46.5 Mbps</b>	<b>5 to 14 Mbps</b>

Pada tabel di atas, kebutuhan arus hilir dan mudik adalah sekitar 46.5 Mbps / 19.5 Mbps untuk 1 rumah. Dari semua layanan yang paling signifikan membutuhkan *Bit Rate* tinggi adalah layanan Internet. Sementara layanan HDTV (High Definition TV) mungkin adalah layanan yang paling di tunggu – tunggu karena dapat menghadirkan kualitas TV dan Video pada layar lebar sebesar 42 inchi dengan kualitas gambar dan suara setara DVD generasi HD-DVD dan BLU-RAY. Tentu saja kebutuhan arus hilir dan mudik ini harus dihantarkan dengan jaringan akses NGN VDSL2 yang mendukung.

Kemampuan VDSL2 dalam memenuhi kebutuhan *Bit Rate* dari sektor perumahan di tunjukkan pada gambar di bawah ini. Tentu saja VDSL2 tetap mempunyai kelemahan, *Bit Rate* yang bisa dihantarkan berbanding lurus dengan jarak kabel tembaga dari RK ke modem rumah. Semakin jauh jarak tembaga VDSL2 antara modem rumah dan RK , maka semakin kecil *Bit Rate* yang bisa

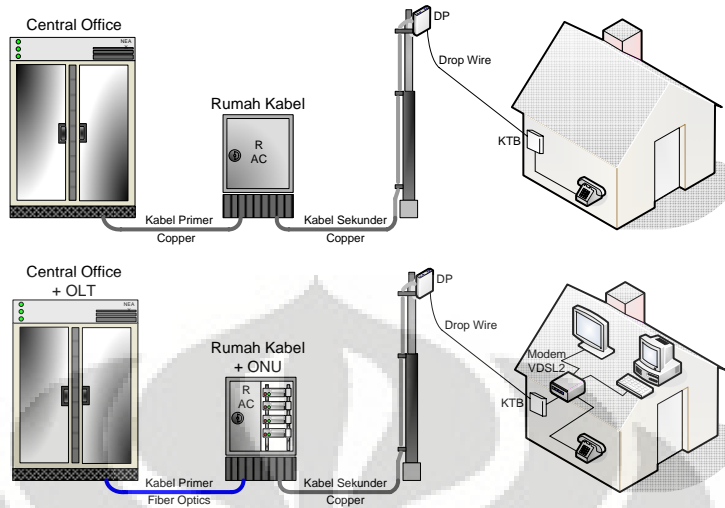
dihantarkan. Kebalikannya, semakin dekat jarak tembaga antara modem rumah dan RK, maka makin cepat transmisi *Bit Rate* yang bisa dihantarkan.



Gambar 2.1 Kinerja VDSL2 [7]

Jumlah pelanggan PSTN di Indonesia mencapai kurang lebih 9 juta. Jumlah ini tersebar di seluruh Indonesia dengan 312 kode area PSTN yang ada. Jumlah paling besar pelanggan PSTN adalah di wilayah Propinsi DKI Jakarta yang mencapai kurang lebih 2 juta sambungan telepon. Dengan tingkat kompleksitas jaringan dan pengkabelan yang ada pada Propinsi DKI Jakarta, desain yang nanti dibuat dapat dijadikan barometer tolak ukur untuk membuat desain jaringan VDSL2 pada propinsi – propinsi lainnya.

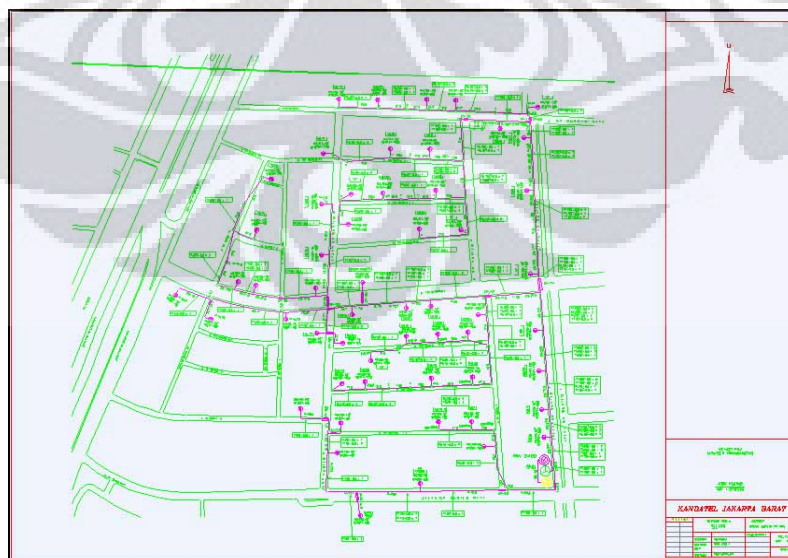
Konfigurasi elemen PSTN yang harus dioptimasi, yaitu pada Kabel Primer antara Sentral dan RK. Pada sebelumnya Kabel Primer ini merupakan jenis tembaga, namun untuk NGN kabel primer ini harus diganti dengan kabel fiber optik. Kemampuan dan kapasitas fiber optik akan disesuaikan dengan kebutuhan yang ada di lapangan.



Gambar 2.2 Konfigurasi Jaringan PSTN dan VDSL2

Desain jaringan akses dapat bervariasi tergantung kondisi lapangan, namun jarak kabel tembaga antara RK dan modem VDSL2 di rumah tidak lebih dari 750 meter. Hal ini penting untuk menjaga kualitas layanan NGN yang menjanjikan arus hilir / mudik sebesar 46.5 / 14 Mbps.

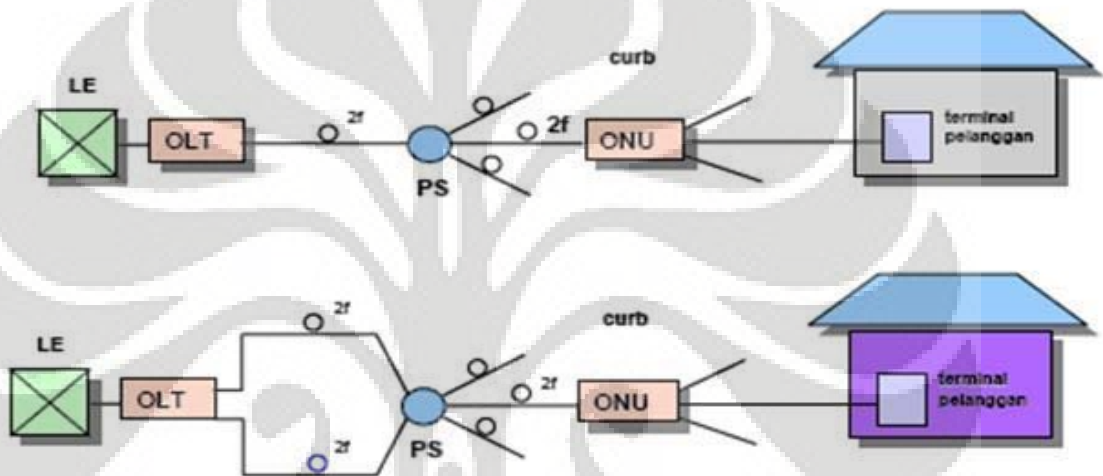
Contoh Kondisi lapangan pada R AV daerah Slipi adalah seperti tergambar di bawah ini. Jarak Udara dari DP ke rumah – rumah maksimal tidak lebih dari 50 meter, ini membuat jarak antara DP ke R AV tidak boleh lebih dari 700 meter. Berarti akan semakin banyak RK yang letaknya akan mendekati daerah rumah – rumah.



Gambar 2.3 Contoh Skema Kabel Primer dan Sekunder RAV Slipi

## 2.2 SKEMA PENGGELARAN PON

Untuk desain jaringan VDSL2 nantinya akan menggunakan skema FTTC (Fiber To The Curb) dimana jaringan primer adalah jaringan kabel *optik* dan jaringan sekunder adalah kabel tembaga. Titik Konversi Optik (TKO) terletak di suatu tempat di luar bangunan, baik di dalam kabinet (RK), di atas tiang maupun lubang manusia. Terminal pelanggan dihubungkan dengan TKO melalui kabel tembaga hingga beberapa ratus meter.

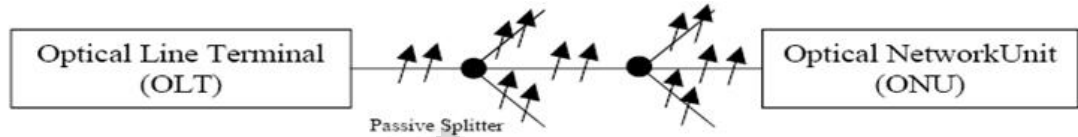


Gambar 2.4 Modus Aplikasi FTTC

### 2.2.1 Konfigurasi PON

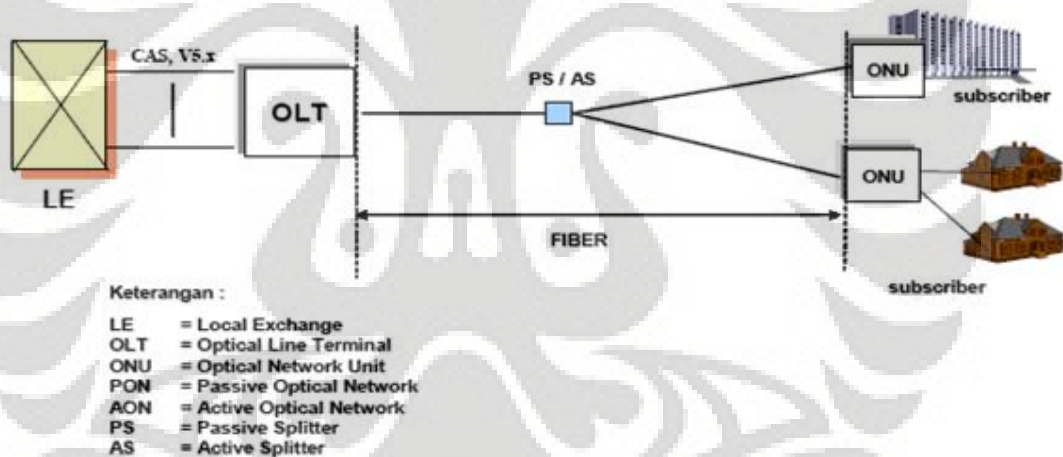
PON adalah bentuk jaringan yang lebih spesifik dari FTTC, dimana PON mengandung perangkat optik pasif dalam jaringan distribusi optik. Perangkat optik pasif yang dipakai adalah konektor, pembagi pasif dan kabel optik itu sendiri. Dengan pembagi pasif kabel optik dapat dipecah menjadi beberapa kabel optik lagi, dengan kualitas informasi yang sama tanpa adanya fungsi addressing dan filtering. Secara analogi PON menganggap ONU adalah pelanggannya, dimana setiap ONU mendapatkan jalur khusus yang didedikasikan kepada ONU tersebut. Hal ini sama dengan kabel tembaga dimana setiap pelanggan PSTN diberikan jalur khusus yang didedikasikan untuk melayani kebutuhan pelanggan PSTN tersebut. Dalam PON terdapat tiga komponen utama yaitu Optical Line

Terminal (OLT), Optical Distribution Network (ODN) dan Optical Network Unit (ONU). Berikut diberikan gambar tentang topologi PON.



Gambar 2.5 Topologi PON

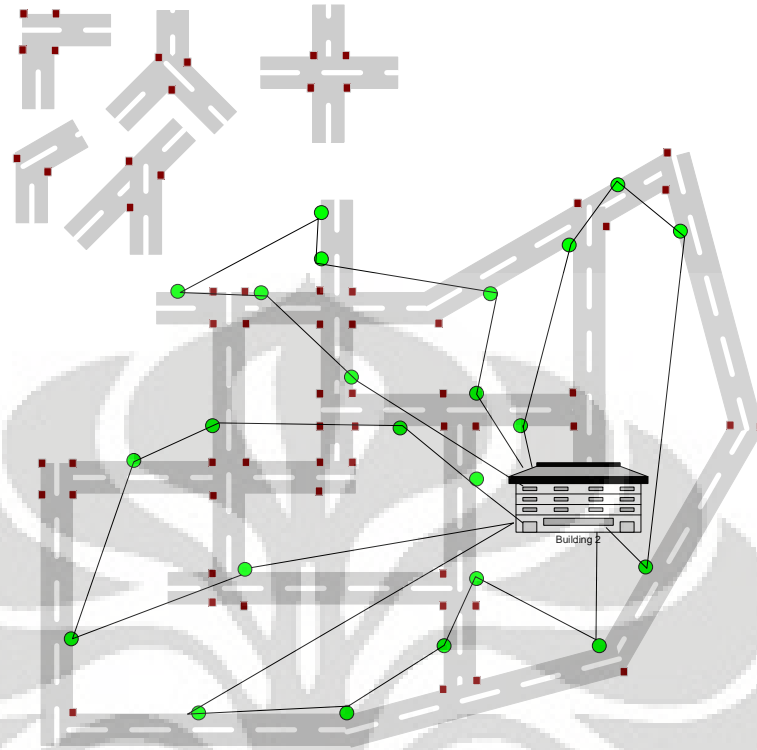
OLT berfungsi untuk melakukan konversi dari sinyal listrik menjadi sinyal optik dan sebaliknya. Dalam sebuah OLT bisa terdiri atas beberapa ODN. Sedangkan ONU berfungsi untuk mengubah sinyal optik menjadi sinyal listrik dan sebaliknya dari sinyal listrik menjadi sinyal optik. Di dalam perencanaan jaringan dengan PON yang utama adalah skenario penggelaran ONU.



Gambar 2.6 Konfigurasi PON

### 2.3 ALGORITMA PENCARIAN SCATTER

Pencarian *Scatter* bekerja berdasarkan 5 tahapan , tahapan – tahapan bekerja berdasarkan metode kombinasi untuk menghasilkan solusi yang berbeda sehingga didapatkan solusi akhir yang mendekati optimal.



Gambar 2.7 Contoh Pengelompokan pada Algoritma Pencarian *Scatter*

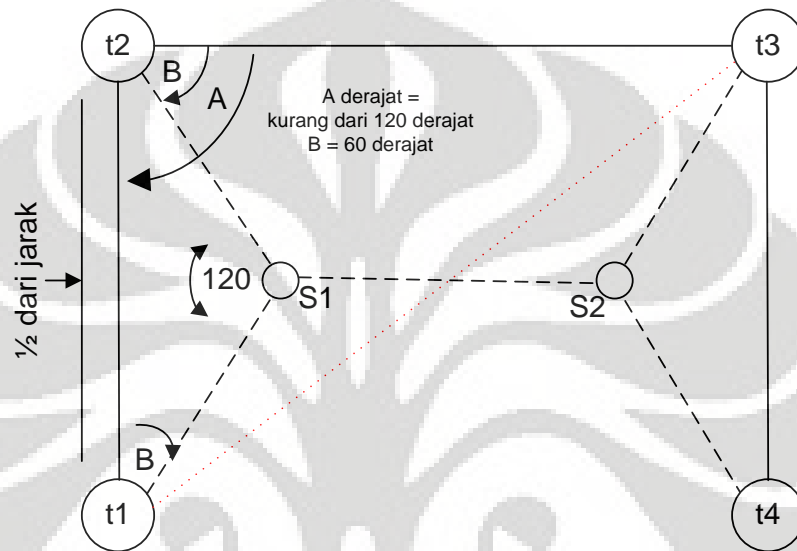
Adapun tahapan – tahapan tersebut adalah :

1. Metode Pembangkitan Tersebar : Dimana di bangkitkan solusi – solusi yang berbeda dari solusi akar yang dibangkitkan secara acak.
2. Metode Perbaikan : Merubah solusi menjadi satu atau lebih solusi yang lebih baik kualitasnya.
3. Metode Pembaharuan Solusi Referensi : Menciptakan dan memantau sederetan solusi – solusi referensi yang terpilih berdasarkan kualitasnya.
4. Metode Pemilihan Inti : Membangkitkan inti dari solusi referensi di atas sebagai dasar untuk menciptakan solusi kombinasional.
5. Metode Kombinasi Solusi : Menggunakan kombinasi pembebanan struktur untuk merubah kombinasi solusi pada proses sebelumnya untuk menciptakan satu atau lebih solusi kombinasi.

Penggunaan algoritma Pencarian Scatter walaupun berfungsi mencari dan menciptakan solusi pengelompokan , pengelompokkan pada penelitian ini menggunakan kedekatan secara sudut polar.

## 2.4 ALGORITMA POHON STEINER

Pohon *Steiner* berfungsi untuk mencari pencabangan yang paling efektif terhadap titik – titik yang tersebar , dimana diciptakan suatu Titik *Steiner* fiktif sebagai titik pencabangannya tersebut. Dengan metode penempatan titik *Steiner* Heuristik, *Steiner* point fiktif dibuat pada sudut 120 derajat dari titik – titik yang lain.



Gambar 2.8 Algoritma Penambahan Titik *Steiner*

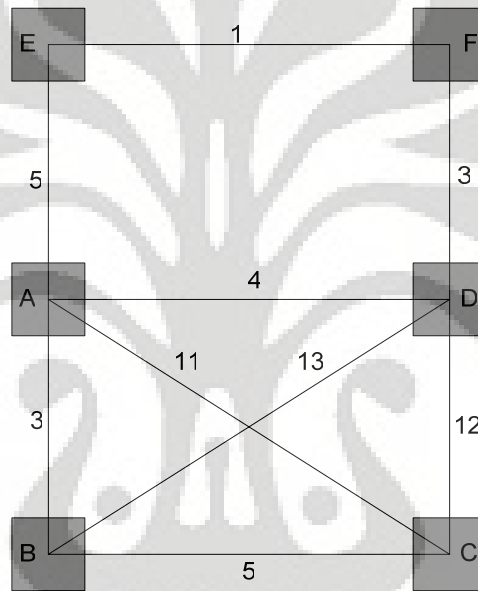
Cara kerja dan menciptakan Titik *Steiner* adalah sebagai berikut : S1 dan S2 merupakan Titik *Steiner* fiktif yang dibuat berdasarkan acuan sudut A dan sudut C. Koneksi sederetan garis antara t1-s1, s1-t2, s1-s2, s2-t3, dan s2-t4 adalah merupakan Pohon *Steiner* paling mendekati efektif.

Pohon *Steiner* akan menciptakan percabangan yang efektif secara jarak antara beberapa titik yang tersebar. Apabila kita lihat seperti gambar di atas, maka percabangan tersebut akan dapat digunakan baik dalam link tanpa kabel maupun link kabel.

## 2.5 ALGORITMA DIJKSTRA

Algoritma Dijkstra adalah algoritma untuk menemukan jarak terpendek dari suatu titik ke titik yang lainnya pada suatu *graph* yang berbobot, dimana jarak antar titik adalah bobot dari tiap ruas pada *graph* tersebut. Algoritma dijkstra mencari jarak terpendek untuk tiap titik dari suatu *graph* yang berbobot. Algoritma dijkstra mencari jarak terpendek dari titik asal ke titik terdekatnya,

kemudian ke titik kedua, dan seterusnya. Secara umum, sebelum dilakukan i iterasi, algoritma sudah mengidentifikasi jarak terdekat dari i-1 titik terdekatnya. Selama seluruh ruas berbobot tertentu yang (positif), maka titik terdekat berikutnya dari titik asal dapat ditemukan selama titik berdekatan dengan titik  $T_i$ . Kumpulan titik yang berdekatan dengan titik di  $T_i$  dapat dikatakan sebagai “titik samping”. Titik inilah yang merupakan kandidat dari algoritma dijkstra untuk memilih titik berikutnya dari titik asal. Pada Gambar dibawah ini diberikan suatu contoh graph, dimana titik awal adalah A.



Gambar 2.9 : Graph Pada Dijkstra

Dengan pencarian rute terpendek algoritma dijkstra adalah sebagai berikut :

Tabel 2.2 : Pencarian Rute Dijkstra

Titik	A	B	C	D	E	F
Status	1	0	0	0	0	0
Bobot	~	3	11	4	5	~
Sebelumnya	A	A	A	A	A	---

Titik	A	B	C	D	E	F
Status	1	1	0	0	0	0
Bobot	~	3	8	4	5	~
Sebelumnya	A	A	D	A	A	---



Titik	A	B	C	D	E	F
Status	1	1	1	1	1	1
Bobot	~	3	8	4	5	6
Sebelumnya	A	A	B	A	A	E

Pencarian Dijkstra akan berhenti ketika Status pada setiap titik adalah 1, sedangkan apabila belum 1 maka akan terus dilakukan. A sebagai titik awal akan mencari titik berikutnya yaitu B dikarenakan bobotnya yang paling kecil. Maka Status B adalah 1 dan baris sebelumnya menjadi A. Otomatis pada kolom C, Bobot menjadi 8 karena A dari C lebih mudah dicapai melalui B. Dan baris sebelumnya diganti menjadi B. Begitu seterusnya sampai semua Status bernilai 1.

Kaitannya dengan algoritma sebelumnya adalah, untuk mencari jarak terpendek dan jalur terbaik dari titik – ke titik . Titik – titik yang dihubungkan adalah Sentral – RK , RK – DP, dan Steiner – RK.

# BAB III

## PERENCANAAN APLIKASI DESAIN JARINGAN

### 3.1 PEMETAAN TITIK DP , RK DAN TITIK JALAN

DP (Distribution Point) adalah kotak pembagi yang tergantung di atas tiang telepon untuk membagi kabel sekunder menjadi drop wire ke rumah – rumah. Titik DP dipetakan sebagai titik yang statis dan tidak berubah. Titik RK yang sudah ada sebelumnya merupakan koordinat pusat, sedangkan RK – RK baru hasil desain aplikasi adalah titik yang dipetakan ke dalam matrik *Euclidean* pada Pencarian Scatter.

Untuk koordinat titik adalah dengan satuan meter dengan ukuran sebenarnya yang ada pada peta, hal ini diperlukan untuk mengetahui secara persis kebutuhan panjang kabel dan kebutuhan alat produksi lainnya. Untuk RK yang sudah ada sebelumnya, akan dipusatkan sebagai titik depot distribusi , seperti konsep PRKB dimana barang yang di distribusikan pada muatan mobil berupa *Bit Rate*. Titik depot tidak perlu ada di tengah, namun tetap menjadi pusat distribusi dari PRKB.

Setiap titik jalan pada peta merupakan titik yang dicatat selayaknya titik DP, sedangkan ruas jalan adalah garis yang digambar berdasarkan letak ke dua titik sudut yang mendefinisikannya. Untuk menggambarkan peta dalam aplikasi, titik – titik jalan merupakan inputan yang akan diproses .

#### 3.1.1 Simbol Matematis Jaringan Akses Untuk DP dan Titik Jalan

Apabila kita anggap  $G$  adalah *Graph* dengan komunitas seperti ini :

$$G = (V, A) \dots\dots\dots (3.1)$$

dimana

$$V = \{v_0, v_1, \dots, v_n\} \dots\dots\dots (3.2)$$

dan  $A$  adalah :

$$A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\} \dots\dots\dots (3.3)$$

Dimana  $V$  adalah semua *vertex* pada gambar, dan  $A$  adalah kumpulan semua garis yang menghubungkan antara *vertex*.

Vertex  $v_o$  adalah depot, dimana sejumlah armada  $m$  yang identik sanggup membawa muatan  $Q$  kepada kumpulan titik – titik *vertex* tersebut (kumpulan titik – titik sekeluarga disebut pengelompokan). Titik selain depot atau didefinisikan sebagai

$$V' = V \setminus \{v_o\} \dots\dots\dots (3.4)$$

adalah lokasi pelanggan yang harus di capai.

$$C = (c_{ij}) \dots\dots\dots (3.5)$$

didefinisikan sebagai matrik cost dimana antara titik  $i$  dan  $j$  adalah jarak yang harus ditempuh. Untuk karakteristik *cost* ini adalah

$$c_{ij} \leq c_{ik} + c_{kj} \dots\dots\dots (3.6)$$

dan

$$c_{ij} = c_{ji} \dots\dots\dots (3.7)$$

yang artinya adalah  $i$  dan  $j$  bukan vektor. Karakteristik *cost* ini berlaku untuk setiap *vertex*

$$(v_i, v_j) \in A \dots\dots\dots (3.8)$$

Kumpulan *vertex*

$$E = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\} \dots\dots\dots (3.9)$$

adalah simetris untuk setiap  $i$  dan  $j$ . Berarti untuk setiap elemen pada matrik *Euclidean* adalah sama antara baris  $i$  kolom  $j$  dan baris  $j$  kolom  $i$ . Berarti juga matrik *Euclidean* berbentuk simetris. Apabila definisi tentang titik PRKB di atas kita analogikan dengan masalah kita maka didapatkan relevansi sebagai berikut :

$$V = \{v_o, v_1, \dots, v_n\} \dots\dots\dots (3.10)$$

adalah banyak nya titik – titik RK baru pada suatu cakupan RK yang sudah ada sebelumnya

$$A = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\} \dots\dots\dots (3.11)$$

adalah garis-garis maya yang menghubungkan titik – titik RK baru tersebut satu sama lainnya.  $v_o$  sebagai depot adalah RK yang sudah ada sebelumnya .  $Q$  adalah kemampuan transmisi kabel fiber .

$$C = (c_{ij}) \dots\dots\dots (3.12)$$

adalah matriks yang berisi jarak udara antara semua titik RK baru.

### 3.1.2 Mencatat Koordinat Titik Jalan dan Titik DP pada File Excel

Untuk Titik DP dan Titik Jalan di satukan sebagai 1 kategori pada aplikasi sehingga seragam. Untuk Ruas Jalan di catat pada file Excel sebagai urutan Titik – Titik Jalan. Untuk lebih jelasnya dapat dilihat pada tabel sebagaimana berikut :

Tabel 3.1 Penulisan Koordinat DP dan Titik Jalan

X	Y	Type	ID
75	831	1	1
75	780	1	2
-408	828	1	3
-408	774	1	4
-462	828	1	5
-462	774	1	6
...	...	...	...
...	...	...	...
-169	150	2	1
-399	157	2	2
-230	401	2	3
-411	463	2	4
-88	833	2	5
-5	404	2	6

Ket :  
type 1  
menyatakan  
bahwa dia  
adalah  
sebagai titik  
jalan

Ket :  
type 2  
menyatakan  
bahwa dia  
adalah  
sebagai DP

Pada tabel di atas, identitas baris adalah sekaligus identitas titik. Dimana Titik Jalan, DP, RK, Sentral dan Steiner adalah juga dianggap sebagai titik. Yang membedakan adalah Typenya, apabila Typenya 1 adalah titik jalan, 2 adalah DP, 3 adalah RK, 4 adalah Steiner dan 5 adalah Sentral. Untuk RK baru dan Steiner merupakan titik tambahan yang merupakan hasil dari proses algoritma.

Tabel 3.2 Penulisan Ruas Jalan

A	B
1	3
2	4
3	5
5	6
6	9
4	7
9	10

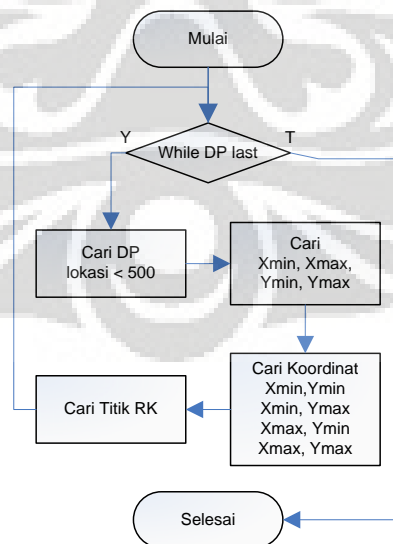
Ket :  
A dan B  
adalah titik  
yang langsung  
terhubung  
pada suatu sisi  
jalan

Pada tabel di atas, yang dimaksud A dan B adalah ruas AB, dimana A dan B adalah titik yang ditunjukkan pada baris dalam Tabel Koordinat Titik. Pada perkembangannya nanti di dalam aplikasi, isi dari tabel ini akan bertambah dengan adanya tambahan link jalan menuju DP, link jalan menuju RK, link menyeberang jalan, dan link keluar sentral. Hal ini penting agar algoritma Dijkstra dapat mengkoneksikan titik – titik tersebut terhadap Sentral.

### 3.2 PENCARIAN TITIK RK BARU

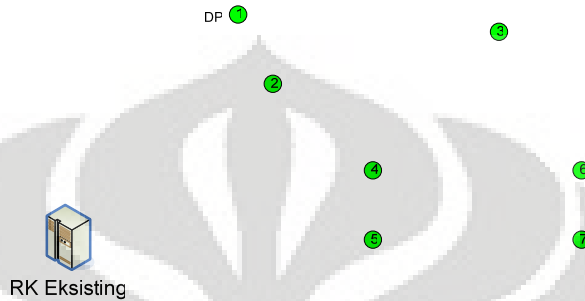
Untuk menentukan titik RK ini maka DP – DP yang berdekatan dan berjarak radius sekitar 500 meter dari RK, dapat dikelompokkan menjadi 1 grup. Spesifikasi teknis VDSL2 sendiri dapat menghantarkan *Bit rate* dalam kecepatan sekitar 50 mbps dalam 750 meter, namun demi menjaga kualitas layanan maka untuk perhitungan maka untuk selanjutnya kita menggunakan angka 500 meter.

Pada perkembangannya kedepan nilai 500 dapat diberikan sebagai parameter kedekatan jarak antara titik RK baru dan DP. Nilai ini yang dapat diubah – ubah untuk mengantisipasi bentuk jaringan yang berbeda – beda. Untuk bentuk jaringan dimana banyak jalan panjang namun buntu akan memerlukan jarak antara RK dan DP dalam radius yang dekat, sedangkan apabila bentuk jaringan terkotak – kotak nilai 500 ini akan berlaku efektif karena titik DP dapat dari 2 arah. Untuk diagram alir penentuan grup DP – DP adalah sebagai berikut :

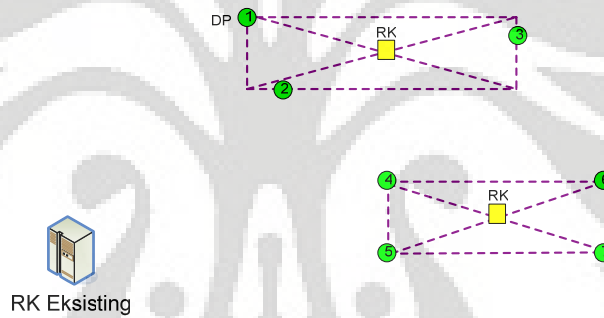


Gambar 3.1 Diagram Alir Penentuan Titik RK

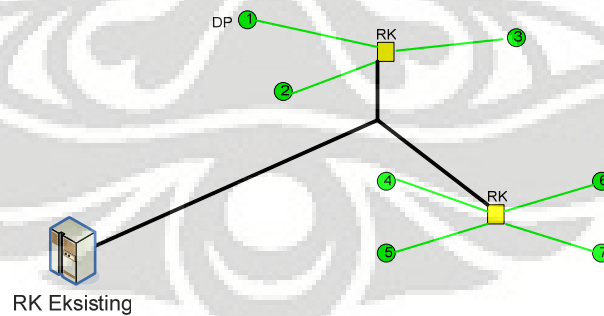
Dalam kondisi sebenarnya, maka semakin baik apabila anggota dari grup DP adalah sebanyak mungkin DP. Apabila satu buah DP secara lokasi terpisah jauh dari DP lainnya, maka tidak ada cara lain selain membangun RK yang terdedikasi kepada DP tersebut.



Gambar 3.2 Letak RK yang sudah ada sebelumnya dan DP



Gambar 3.3 Menentukan Titik RK Baru



Gambar 3.4 Kabel Primer dan Kabel Sekunder

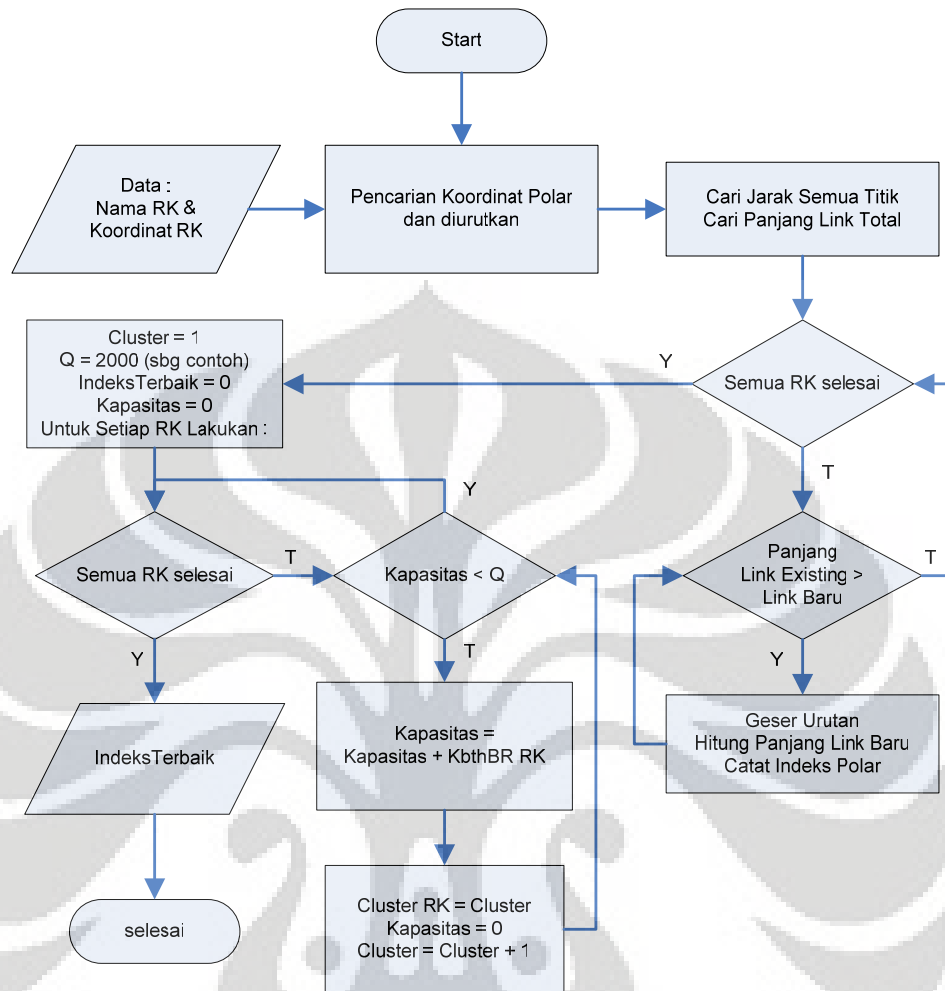
Apabila pengelompokkan RK baru – DP ini sudah terdefinisi, maka hubungan RK baru – DP ini tinggal dicari jalurnya berdasarkan hubungan titik ke titik. Untuk hal tersebut maka diperlukan algoritma djikstra untuk menyelesaikannya.

### 3.3 PENCARIAN SCATTER

Algoritma Pencarian Scatter berfungsi untuk mencari pengelompokan titik RK baru berdasarkan kapasitas Q dari PON (Kabel Primer) yang akan di gelar. Titik – Titik RK baru akan dikelompokkan berdasarkan sudut koordinat polar yang dimilikinya relative terhadap RK yang sudah ada sebelumnya, hal ini nanti berguna untuk mendesain pohon steiner yang dikerjakan algoritma Pohon Steiner. Diagram dari Algoritma Scatter dipaparkan pada Gambar 3.5.

Hasil dari Pencarian Scatter adalah Indeks Terbaik, dimana Indeks Terbaik akan berisi kumpulan titik – titik RK baru yang dikelompokkan satu sama lain berdasarkan kedekatan sudut polar dari titik – titik RK. Indeks Terbaik juga dapat menggambarkan suatu bentuk grafik seperti kelopak bunga dimana pusat dari kelopak bunga tersebut adalah RK Yang sudah ada sebelumnya. Indeks Terbaik digunakan sebagai inputan untuk algoritma Pencarian Scatter , dimana bentuk kelopak bunga akan di jadikan bentuk pohon steiner yang membuat link lebih efektif secara jarak.

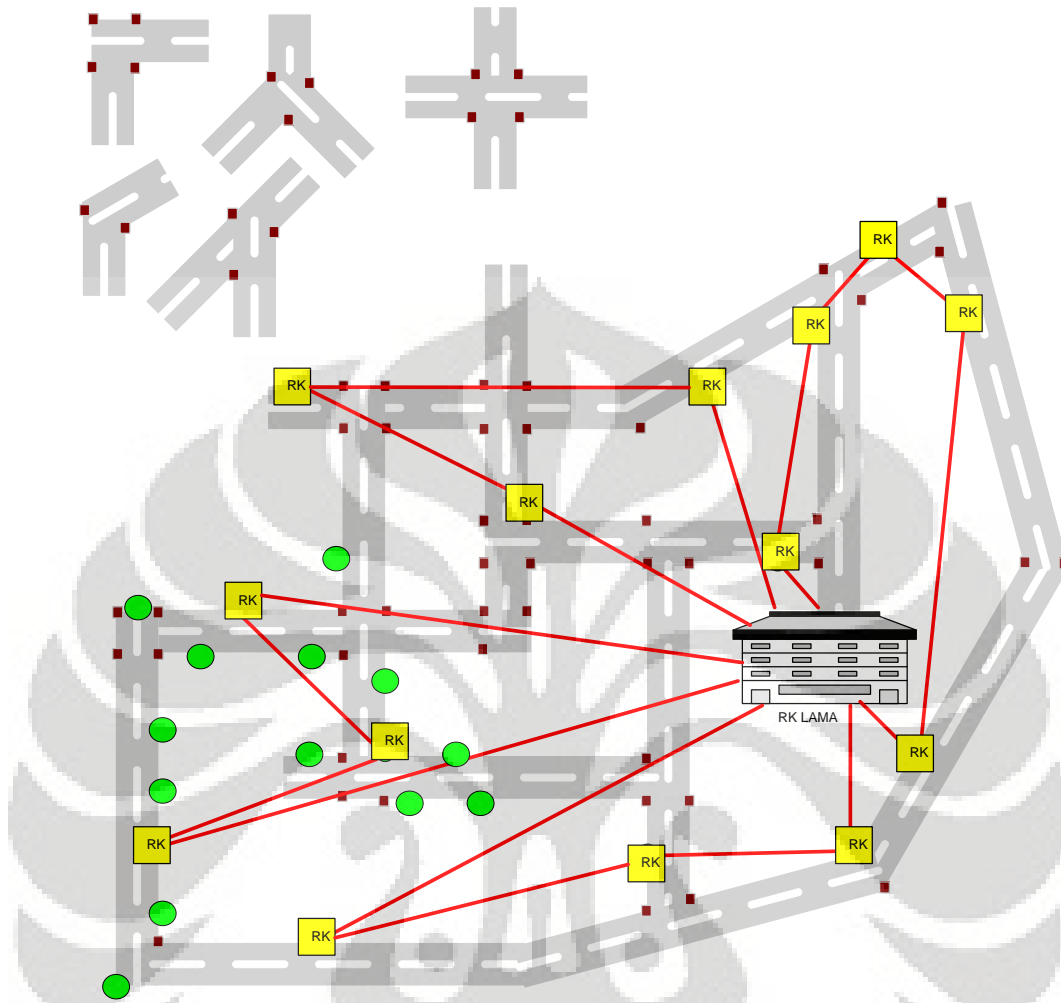
Berbeda dengan yang dicetuskan oleh Fred Glover, maka penulis menggunakan pendekatan koordinat polar karena lebih mudah daripada menggunakan koordinat kartesian. Dengan pendekatan koordinat polar , titik – titik RK baru akan diputar untuk mencari bentuk kelopak bunga yang baik. RK satu dengan lainnya dikelompokkan berdasarkan kedekatan sudutnya, hal ini disebut mencari solusi. lalu kelompok – kelompok tersebut diputar sehingga berganti anggota untuk mendapatkan solusi baru. Pilihan akhir dijatuhkan kepada solusi dimana sederetan RK yang telah menjadi beberapa kelompok tersebut mempunyai panjang total link yang paling pendek, apabila dibandingkan dengan solusi lain yang pernah dicari. Perbaikan solusi ini dilakukan untuk menuju pencabangan pohon steiner yang efektif sehingga bentuk jaringan yang didesain menjadi tidak ruwet dan otomatis penggunaan alat produksi akan menjadi lebih efektif.



Gambar 3.5 Diagram Alir dari Algoritma Pencarian Scatter

Hasil akhir dari Algoritma Pencarian Scatter adalah Indeks Terbaik, yang akan berisi jalur titik – titik RK yang sudah dikelompokkan. RK – RK baru yang berkelompok ini akan mengikuti kaidah Pencarian Scatter dimana, suatu kelompok memulai perjalanan dari titik depot (RK yang sudah ada sebelumnya) menuju titik – titik transit (RK baru) dan berakhir kembali di depot. Untuk lebih jelasnya diilustrasikan pada Gambar 3.6 pada suatu model jaringan.



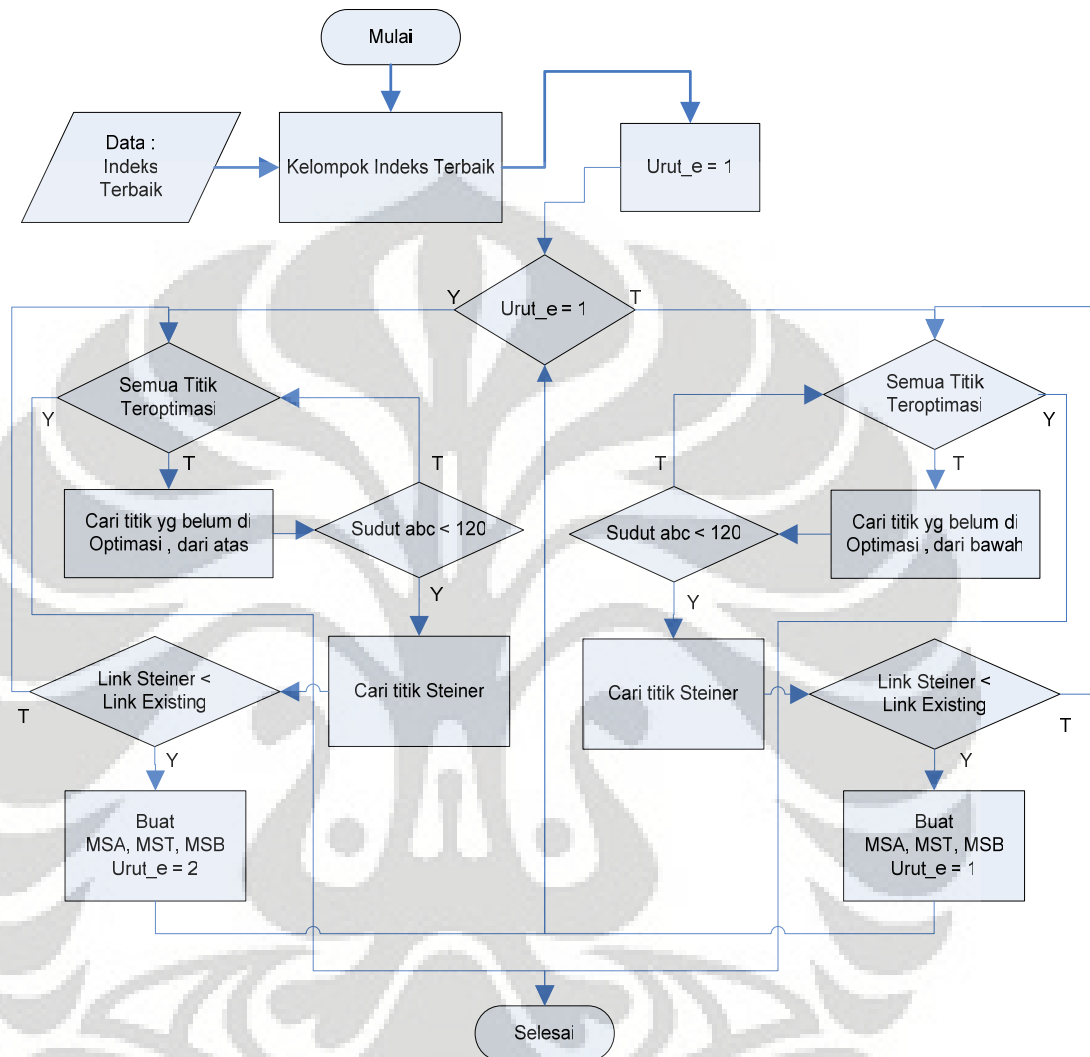


Gambar 3.6 Gambar Pengelompokkan Titik Sesuai Sudut Polar RK

### 3.4 POHON STEINER

Algoritma Pohon Steiner berfungsi untuk mencari pencabangan yang efektif berdasarkan pengelompokkan pada Algoritma Pencarian Scatter. Proses algoritma ini adalah serangkaian optimasi dimana pencarian sudut antara 3 titik dan pencarian koordinat titik steiner tambahan, seperti ditunjukkan pada Gambar 3.7. Pertama kali 3 titik yang akan dianalisa dihitung terlebih dahulu besar sudutnya, apabila sudut  $< 120$  derajat maka dicari titik steiner, apabila solusi steiner tersebut adalah yang lebih baik, maka solusi steiner dianggap sebagai solusi yang valid. Untuk menentukan apakah solusi steiner adalah solusi yang valid atau bukan dilakukan penghitungan panjang link sebelum dan sesudah adanya titik steiner pada 3 titik yang terlibat. Apabila solusi steiner mempunyai

panjang link yang lebih pendek, maka solusi itu diterima, apabila tidak maka aplikasi berlanjut mencari 3 titik lainnya untuk dicoba dicari solusi steinernya.



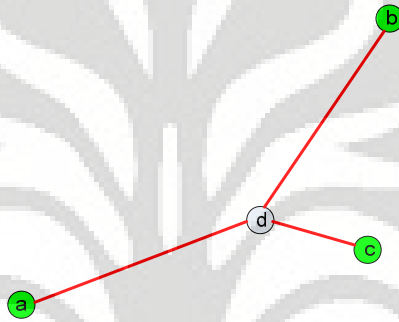
Gambar 3.7 Diagram Alir Algoritma Pohon Steiner

### 3.4.1 Pencarian Sudut Antara Tiga Titik

Mencari sudut 3 titik dimaksud sebagai kriteria awal untuk dilakukannya optimasi terhadap 3 titik., Apabila sudut  $< 120$  derajat maka optimasi akan dipertimbangkan untuk dilakukan. Pendekatan yang dilakukan adalah pendekatan sudut yang mengapit antara 2 vektor. Untuk itu titik a dan c dinormalkan terhadap titik b, sehingga b seakan – akan menjadi titik pusat dengan koordinat (0,0). Selanjutnya dilakukan penghitungan sudut yang mengapit 2 vektor. Untuk lebih jelasnya dapat mengacu pada gambar berikut :



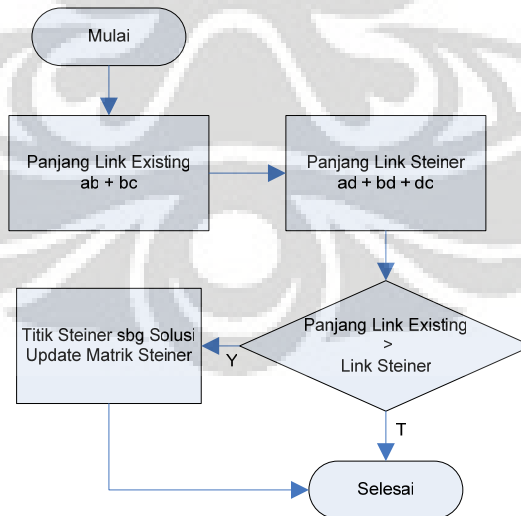
Untuk menentukan apakah diantara 3 titik tersebut dapat dilakukan proses selanjutnya yaitu pencarian titik steiner, cos sudut adalah indikasinya. Apabila cos sudut  $< 120$  derajat maka akan dicari letak titik bantu steiner. Pertama kali titik a akan menjadi pusat sumbu, dan link ab dicari koordinat polarnya. Sudut – Sudut yang diketahui  $adb = 120^\circ, ado = 60^\circ, oad = 30^\circ$ , sedangkan karakteristik polar ruas garis ab adalah  $R_{ab}, \theta$ . Maka titik D berada adalah setengah dari  $R_{ab}$  dan sudutnya adalah  $\theta - 30^\circ$ . Apabila diterima sebagai solusi maka link antara 3 titik abc menjadi sebagai berikut :



Gambar 3.10 Solusi Steiner pada 3 titik

### 3.4.3 Penentuan Solusi Steiner

Kembali ke jalur aplikasi, maka implikasi titik steiner pada badan utama aplikasi adalah sebagai berikut:

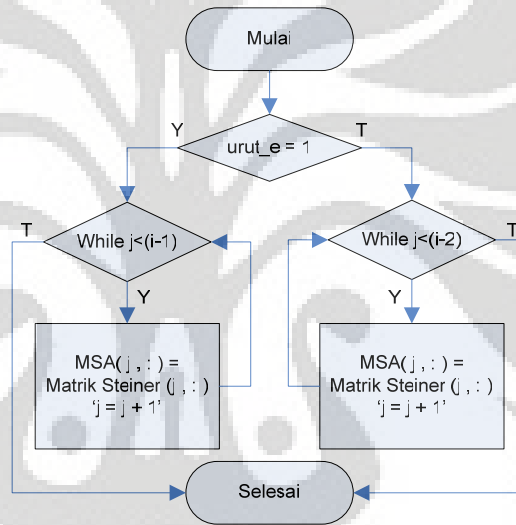


Gambar 3.11 Solusi Steiner pada badan utama aplikasi

Solusi *Steiner* akan diterima apabila total panjang link ( $ad + bd + dc$ ) adalah lebih kecil daripada total panjang link ( $ab + bc$ ). Sedangkan solusi *Steiner* tidak akan diterima apabila total panjang link ( $ad + bd + dc$ ) adalah lebih besar daripada total panjang link ( $ab + bc$ ). Hal ini dapat saja terjadi apabila titik  $c$  adalah cukup dekat ke titik  $b$ .

### 3.4.3.1 Matrik Steiner Atas

Matrik Steiner Atas adalah setiap baris di atas baris yang sedang dioptimasi. Matrik Steiner Atas ini akan tidak ada apabila indikator penghitung adalah 1. Untuk diagram alirnya adalah sebagai berikut :



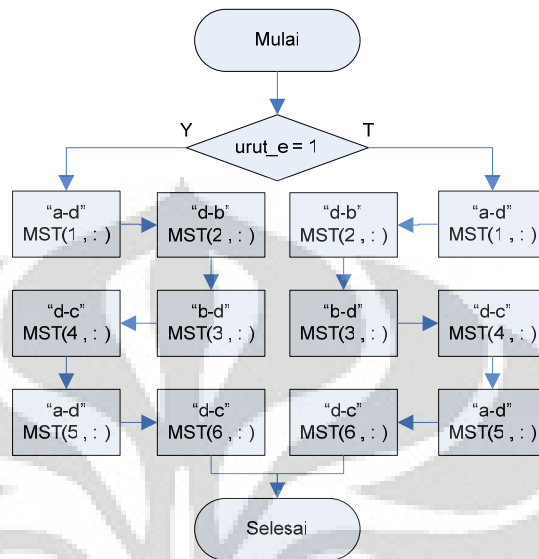
Gambar 3.12 Diagram Alir Matrik Steiner Atas

Dalam Matrik Steiner, posisi penghitung ( $i$ ) adalah pada suatu baris yang menunjuk pada baris yang mengandung informasi  $ab$ . Baris ini adalah baris yang akan berusaha dioptimasi, baris – baris sebelumnya sudah pernah dilakukan optimasi sehingga dianggap optimal. Setiap baris diatas baris  $ab$  ini adalah baris yang tidak berubah secara isi dan disimpan untuk nantinya digabung kembali dengan Matrik Steiner Tengah dan Matrik Steiner Bawah.

### 3.4.3.2 Matrik Steiner Tengah

Matrik Steiner Tengah akan berisi 6 baris baru yang merupakan perluasan dari 2 baris yang sedang dioptimasi. Sebagaimana telah disebutkan di atas, bahwa

posisi penghitung (i) sedang berada pada baris yang mengandung informasi ab, atau baris yang sedang dioptimasi.

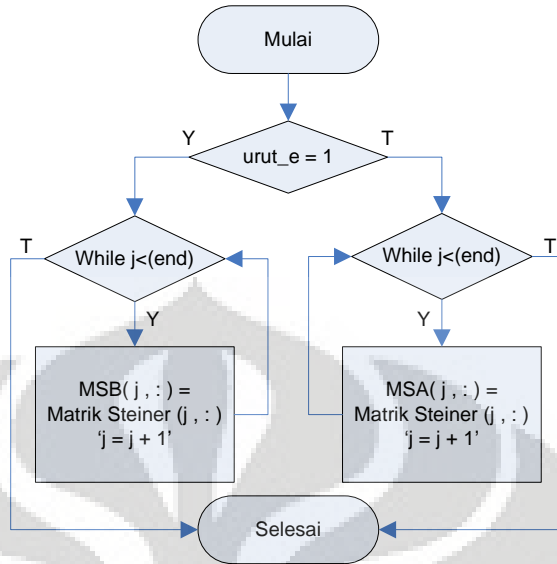


Gambar 3.13 Diagram Alir Matrik Steiner Tengah

6 baris tersebut mendeskripsikan kehadiran titik steiner sebagai solusi dan implikasinya pada penambahan link dan hadirnya sudut – sudut baru pada link – link tersebut. Yaitu bermula dari hanya ab dan bc (2 baris) , sekarang menjadi ad,db, ad, dc, bd dan dc. Sangat mungkin dari 6 baris ini akan dioptimasi ulang pada saat posisi penghitung (i) ada di baris yang baru ini.

### 3.4.3.3 Matrik Steiner Bawah

Matrik Steiner Bawah adalah setiap baris dibawah 2 baris yang sedang dioptimasi. Matrik Steiner Bawah tidak dicari apabila penghitung pada nilai terakhir , atau letak cursor sedang menunjuk baris terakhir dari loop. Diagram alir dari Matrik Steiner Bawah ada pada Gambar 3.14. Ketika posisi penghitung (i) ada pada baris yang mengandung informasi ab, maka tugas dari Matrik Steiner bawah adalah mengumpulkan semua baris yang dimana berawal dari posisi penghitung (i+2) sampai dengan akhir baris pada Matrik Steiner. Serangkaian Matrik Steiner Atas, Matrik Steiner Tengah dan Matrik Steiner Bawah adalah proses optimasi yang saling melengkapi untuk mencari solusi *Steiner* .



Gambar 3.14 Diagram Alir Matrik Steiner Bawah

### 3.4.4 Update Matrik Steiner

Apabila hadirnya link steiner sebagai akibat dari munculnya titik steiner menjadi solusi bagi link antara 3 titik maka akan diberikan update pada Matrik Steiner menyangkut link tersebut. Rancangan Matrik Steiner dan insert solusi steinernya adalah sebagai berikut :

Tabel 3.3 Matrik Steiner

No Ruas	Titik a	Titik b	Xa	Ya	Xb	Yb	Sudut abc	Optimasi
1	45	35	0	0	4274	94	33	0
2	35	22	4274	94	3282	702	33	0
3	35	22	4274	94	3282	702	52	0
4	22	34	3282	702	4313	1081	52	0

Pada Tabel 3.3 di atas, titik 45 dan 35 melambangkan adanya ruas antara titik 45 dan 35. Koordinat titik 45 adalah  $X_a, Y_a$  dan koordinat titik 35 adalah  $X_b, Y_b$ . Sudut yang dibentuk antara titik 45 – 35 – 22 adalah analogi dari sudut abc yang ada prinsip mencari solusi steiner yaitu 33 derajat.

Sekarang ini posisi penghitung (i) ada pada baris pertama, berarti segitiga 45 – 35 – 22 harus di optimasi. Setelah dilakukan penghitungan solusi *Steiner* maka dapat aplikasi menyimpulkan bahwa segitiga ini tidak perlu dilakukan solusi *Steiner* karena tidak membawa hasil yang lebih baik. Lalu kolom optimasi diisi

oleh angka 1 untuk menandakan tidak perlunya ada proses optimasi (sudah optimal). Posisi penghitung (i) akan melanjutkan ke baris 3 dan 4.

Pada baris ini aplikasi mengatakan bahwa solusi *steiner* dapat dilaksanakan. 2 baris (baris 3 dan 4) akan diperbanyak menjadi 6 baris dengan lahirnya titik baru yaitu titik 46, dapat dilihat pada Tabel 3.4. Baris 1 dan 2 adalah baris yang termasuk Matrik Steiner Atas dan Baris 3 dan 4 menjelma menjadi 6 baris yang disebut Matrik Steiner Tengah, sedangkan Matrik Steiner Bawah tidak ada.

Tabel 3.4 Solusi Steiner Pada Matrik Steiner

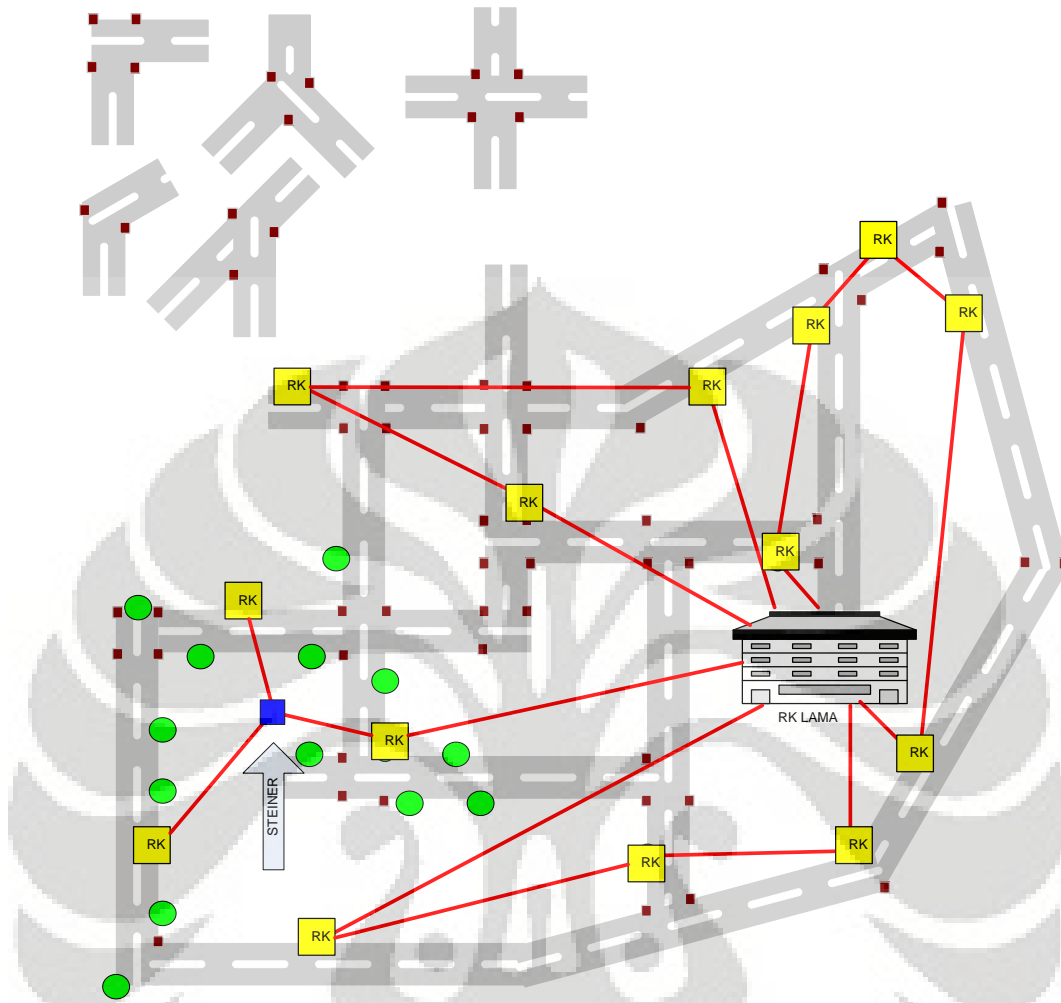
No Ruas	Titik a	Titik b	Xa	Ya	Xb	Yb	Sudut abc	Optimasi
1	45	35	0	0	4274	94	63	1
2	35	46	4274	94	3954	684	63	1
0	35	46	4274	94	3954	684	120	1
0	46	22	3954	684	3282	702	120	1
0	22	46	3282	702	3954	684	131	1
0	46	34	3954	684	4313	1081	131	1
0	35	46	4274	94	3954	684	109	0
0	46	34	3954	684	4313	1081	109	0

Selanjutnya aplikasi akan mengecek ulang setiap titik yang semula terhubung dengan titik 22 pada Matrik Steiner Atas dan Matrik Steiner Bawah. Apabila ditemukan maka titik tersebut akan digantikan dengan titik 46 dan dihitung ulang sudutnya.

Hasil dari Matrik Steiner Tengah adalah penghitungan sudut ulang yang berada didalam Matrik Steiner Tengah. Setiap sudut yang sudah mencapai 120 derajat atau lebih pada kolom optimasi akan diberi nilai 1. Berarti untuk penghitungan kedepannya tidak perlu dioptimasi lagi. Sedangkan yang belum mencapai 120 derajat akan melalui proses optimasi.

Proses optimasi akan berlanjut sesuai posisi penghitung (i), apabila semula posisi penghitung (i) pada baris ke-3 sekarang karena semuanya telah teroptimasi maka posisi penghitung (i) akan berlanjut pada baris ke-7. Yaitu satu – satunya baris yang masih mungkin dioptimasi. Proses akan berlanjut terus sampai dengan semua nilai pada kolom optimasi bernilai 1.



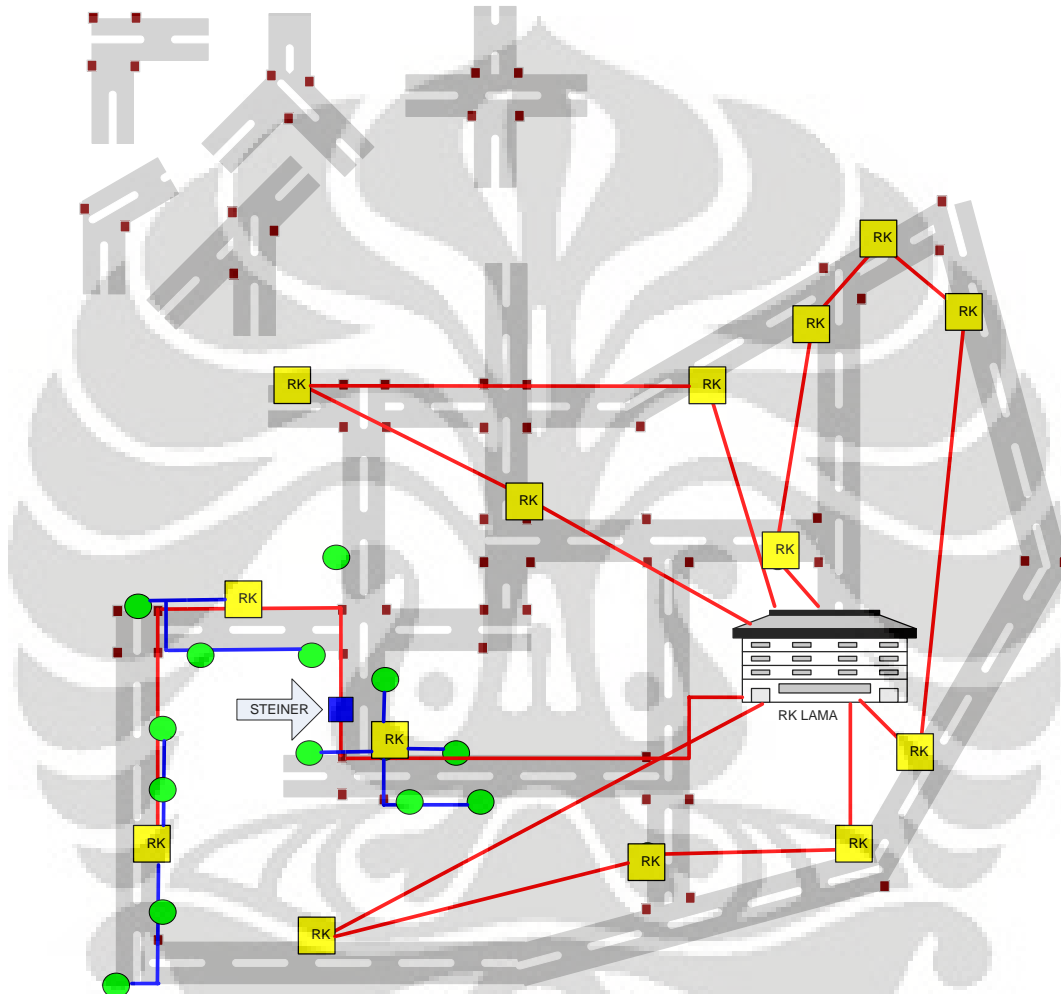


Gambar 3.15 Pohon Steiner pada jaringan

Pada gambar di atas, diberikan gambaran bentuk hasil dari diagram pohon steiner dimana solusi didasarkan pada proses – proses yang sudah dijelaskan sebelumnya. Bentuk ini bukan merupakan desain jaringan yang akan diimplementasikan, namun hanyalah skema logik dari keterhubungan antara titik RK, *Steiner* dan Sentral.

Sebagaimana telah disebutkan pada Bab 3.1.2 , koordinat RK yang baru dan titik steiner dicatat sebagai titik. Maka setelah koordinat RK baru dan titik steiner diketahui, maka tugas selanjutnya adalah mencari jalur dari RK baru ke DP dan dari RK yang sudah ada sebelumnya ke RK baru atau ke Titik Steiner. Hal ini diselesaikan dengan menggunakan algoritma Dijkstra. Desain dari Dijkstra telah disebutkan pada Bab 2.5.

Inputan untuk Dijkstra sudah lengkap dimana diketahui koordinat titik – titik yang akan dihubungkan yaitu titik Sentral , RK , *Steiner* dan DP. Tugas dari algoritma Dijkstra adalah bekerja secara tersegmentasi dimana ruas antara Sentral ke RK dan RK ke DP dihubungkan dengan menggantinya sebagai hubungan rute dari titik ke titik.



Gambar 3.16 Kabel Primer dan Kabel Sekunder Pada Jaringan

Pada Gambar 3.16 di atas, Dijkstra menghubungkan titik Sentral dengan titik RK menggunakan garis berwarna merah. Dan titik RK ke DP dengan garis berwarna biru.

# BAB IV

## ANALISA JARINGAN VDSL2

### HASIL DESAIN APLIKASI

#### 4.1 HASIL DESAIN APLIKASI

Pada peta yang ada, baik yang merupakan model maupun diagram sebenarnya pada RA dan RAV, akan diberikan parameter awal Q (Max Kapasitas Primer) standar 10000 (10 Gbps) dan Radius Grup RK – DP standar 500 meter. Selain itu, dari setiap DP yang ada pada RK kita asumsikan 10 pelanggan anggota DP ingin berlangganan VDSL2 dengan kecepatan Bit Rate 45 Mbps (untuk mempermudah hitungan dari kebutuhan sebenarnya 46.5 Mbps). Hasil lengkapnya didapatkan sebagai berikut :

Tabel 4.1 : Hasil Kinerja Aplikasi

Peta RK	DP	PON Primer	Radius RK - DP	Wkt Lari Logik	Wkt Lari Network	RK	Rasio RK - DP	Rata2 Jarak tbg	DP OK	% DP OK
RX	44	10000	500	7.85	11.47	7	6.29	511.09	38	86.36%
RX	44	10000	400	7.52	11.85	9	4.89	458.51	38	86.36%
RX	44	10000	300	4.50	12.50	13	3.38	332.30	44	100.00%
RA	73	10000	500	5.21	23.36	7	10.43	659.48	46	63.01%
RA	73	10000	200	16.38	30.21	26	2.81	446.62	62	84.93%
RA	73	10000	100	14.01	34.15	49	1.49	326.43	70	95.89%
RAV	55	10000	500	4.10	15.49	9	6.11	553.32	40	72.73%
RAV	55	10000	400	5.44	16.52	14	3.93	500.73	42	76.36%
RAV	55	10000	300	6.20	18.14	17	3.24	424.67	46	83.64%

Catatan : RX adalah model jaringan

Kolom Peta RK adalah menandakan daerah catuan RK yang sedang dianalisa, kolom DP memberikan informasi jumlah DP pada catuan RK tersebut. Waktu Lari Logik dan Waktu Lari Network adalah waktu (detik) yang dibutuhkan aplikasi untuk mencari solusi keterhubungan dan solusi yang diterapkan pada peta. Kolom RK adalah jumlah RK baru yang dibutuhkan untuk mencatu seluruh DP. Rata – rata Jarak Tembaga adalah rata – rata jarak kabel tembaga yang diperlukan untuk menghubungkan antara RK ke DP. Kolom OK adalah jumlah DP yang tercatu kabel tembaga dengan jarak maksimal 700 meter.

Dari hasil di atas yang didasarkan pada hasil file Report.xls, didapatkan beberapa skema pilihan untuk menggelar jaringan VDSL2. Untuk memilih skema yang tepat harus didasarkan alat produksi yang disediakan vendor. Alat produksi akan berkisar pada bentuk PON yang tersedia dan perbandingan Pembagi Pasif yang ada.

Secara kualitatif Aplikasi dapat mendesain jaringan relatif lebih baik apabila peta daerah yang harus dicatu mempunyai bentuk jalan berbentuk kotak – kotak, hal ini didasarkan pengamatan pada setiap gambar hasil desain aplikasi. Peta kota yang berbentuk kotak – kotak dapat dengan mudah mendapatkan catuan VDSL2 yang efektif dengan menset parameter PON Primer pada nilai 10000 dan Radius RK DP 500 meter. Hal ini dapat dilihat pada Lampiran 22 sampai dengan 33, Lampiran 38 sampai dengan 43. Gambar desain jaringan pada RA relatif lebih sulit dicatu daripada jaringan RAV. Hal ini dikarenakan RA mempunyai banyak jalan buntu yang menyulitkan DP untuk dijangkau secara efisien.

Untuk RA , berdasarkan file Report.xls, kita akan mengambil solusi dengan parameter PON Primer adalah 10000 dan Radius RK DP 200. Hal ini diputuskan berdasarkan pertimbangan :

- Catuan RK baru mempunyai radius 400 meter (200 x 2)
- 84.93% DP pada area RA tercatu dengan baik
- 62 DP dari 73 DP dapat menggelar layanan VDSL2 dengan rata – rata jarak kabel tembaga sebesar 446.62 meter.

Untuk RAV , berdasarkan file Report.xls, kita akan mengambil solusi dengan parameter PON Primer adalah 10000 dan Radius RK DP 300. Hal ini diputuskan berdasarkan pertimbangan :

- Catuan RK baru mempunyai radius 600 meter (300 x 2)
- 83.64% DP pada area RAV tercatu dengan baik
- 46 DP dari 55 DP dapat menggelar layanan VDSL2 dengan rata – rata jarak kabel tembaga sebesar 424.67 meter.

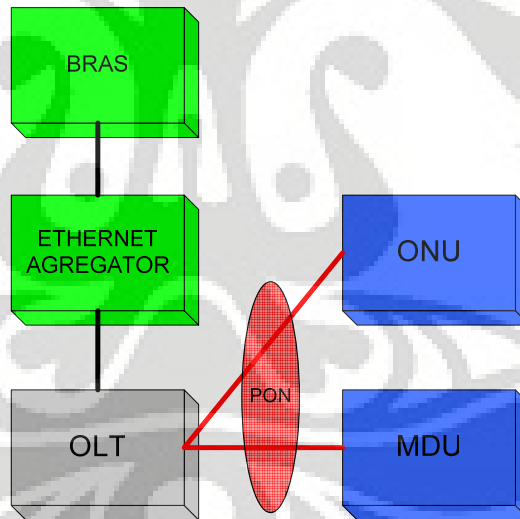
Untuk menentukan seberapa panjang kabel fiber optik maka mengacu kembali ke file Report.xls pada Lampiran 34 dan Lampiran 44. Pada lampiran tersebut dijumlahkan kolom Distance dengan mengacu pada Connection ID yang

bukan DP-RK. Maka didapatkan kebutuhan fiber optik untuk RA adalah sepanjang 33,579 km dan RAV sepanjang 15,06 km.

Untuk menentukan seberapa panjang kabel tembaga maka mengacu kembali ke file Report.xls pada Lampiran 34 dan Lampiran 44. Pada lampiran tersebut dijumlahkan kolom Distance dengan mengacu pada Connection ID DP-RK. Maka didapatkan kebutuhan kabel tembaga untuk RA adalah sepanjang 33,579 km dan RAV sepanjang 15,06 km.

#### 4.2 TOPOLOGI JARINGAN GEPON

Untuk kabel primer akan digunakan teknologi GEPON (Gigabit Ethernet PON) sebagai teknologi penghantar sampai ke ONT (Optical Network Termination). ONT yang dimaksud adalah RK yang dilengkapi modul VDSL2 MDU. Spesifikasi GEPON dapat dibagi menjadi 1 : 64 dengan 2 langkah splitting. Letak, konfigurasi dan pembagian splitter dapat dicari pada konfigurasi logik dari masing – masing RK yang sudah disimulasikan di atas.



Gambar 4.1 Topologi Jaringan GEPON

##### 4.2.1 Alat Produksi GEPON

Topologi standar dari jaringan GEPON terdapat pada komponen Antarmuka PON pada sentral. 1 Antarmuka PON dapat menghantarkan 12 inti yang setara dengan 12 Gbps. 12 Gbps dapat mencatu RK – RK yang membutuhkan, apabila masih kurang maka pengadaan sepaket Antarmuka PON

dapat ditambah. Pada Gambar 4.1 di atas diberikan konfigurasi GEAPON untuk vendor NEC.

Untuk kasus bisnis ini, NEC memberikan bentuk konfigurasi dan spesifikasi seperti tabel 4.2. Dalam konfigurasi ini, maka belum tentu semua alat produksi dipakai secara efektif, penggunaan maksimal spesifikasi yang ditawarkan NEC akan otomatis membuat CapEx pengadaan VDSL2 menjadi lebih murah.

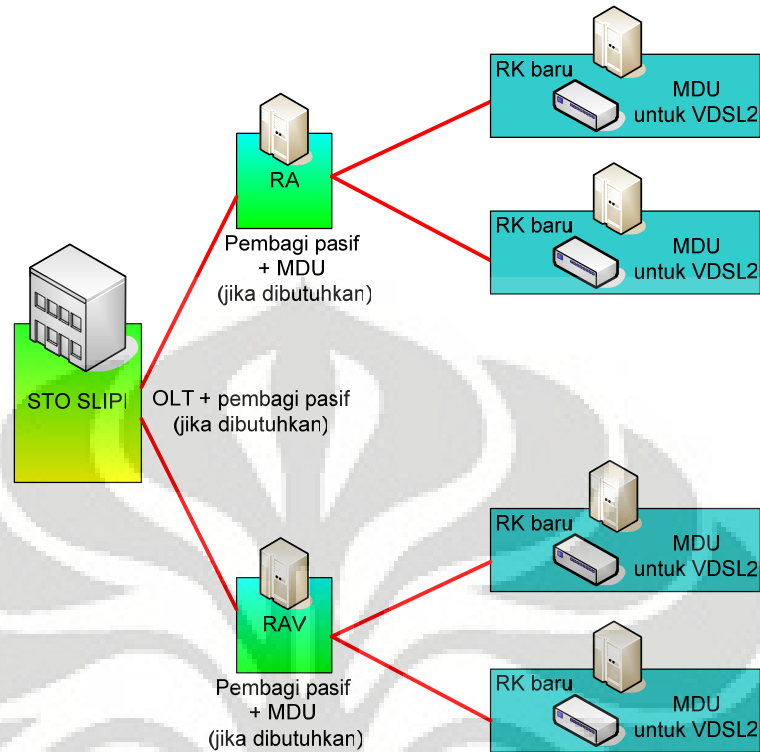
Tabel 4.2 : Paket Alat Produksi NEC Untuk GEAPON

Description	Part #	Unit	Order Quantities & Distribution	Total Qty	Unit Price (IDR)	Total Price (IDR)
<b>GEAPON</b>			Site 1			
<b>BBS 1000+</b>						
<b>BBS 1000+ Main(R3.x cortina based)</b>						
BBS 1000+ 1U chassis Assembly	2424243100	Ea	1	1	4,038,060.00	<b>4,038,060.00</b>
BBS 1000+ GSM2 Board Assembly for cortina solution	2424440305	Ea	1	1	25,528,035.00	<b>25,528,035.00</b>
BBS 1000+ LTM6 Board Assembly	2424277605	Ea	2	2	29,693,040.00	<b>59,386,080.00</b>
BBS 1000 + DC Power Assembly	2424243200	Ea	2	2	4,437,030.00	<b>8,874,060.00</b>
BBS 1000+ FAN Tray	2424243500	Ea	1	1	1,481,025.00	<b>1,481,025.00</b>
DC Power Cable	3648177600	Ea	2	2	531,960.00	<b>1,063,920.00</b>
BBS 1000+ DC Installation Accessory	4070026700	Ea	1	1	175,305.00	<b>175,305.00</b>
Installation Accessory	4070026700	Ea	1	1	175,305.00	<b>175,305.00</b>
<b>Transceiver for BBS1000+ and BBS 4000</b>						
GEAPON Optical transceiver, 1490nmTx/1310nmRx, SC, SFP with Spring Latch, 20Km	IXX0852200	Ea	12	12	10,802,415.00	<b>129,628,980.00</b>
<b>Pigtail</b>						
Single-core FC-SC,SMF, 10M	LXX0007700	Ea	12	12	253,890.00	<b>3,046,680.00</b>
<b>Netman 4000 OMC-A (for GEAPON)</b>						
<b>Netman 4000 OMC-A for 1 to 10k subscribers (Windows-based)</b>						
<b>Server Software</b>						
Netman 4000 OMC-A Software management license per GEAPON subscriber (Minimum 400 subscribers)	5298095306	Ea	864	<b>864</b>	302,250.00	<b>261,144,000.00</b>
Netman 4000 OMC-A software installation media(CD)	2410010300	Ea	1	<b>1</b>	302,250.00	<b>302,250.00</b>
<b>Database</b>						

Sybase ASE DB for Windows (no more than 2 CPUs), 32 bit, English version	NXX0398000	Ea	1	1	18,135,000.00	<b>18,135,000.00</b>
<b>Client Software</b>						
Netman 4000 OMC-A Client Software for GEAPON	NXX0382200	Ea	1	1	30,225,000.00	<b>30,225,000.00</b>
<b>Server Hardware</b>						
Dell OptiPlex 755MT/2.0G Duo Core/1G Memory/80G HD/19" LCD/16xDVD/WinXP Pro	QXX0258900	Ea	1	1	16,998,540.00	<b>16,998,540.00</b>
<b>Client Hardware</b>						
Dell OptiPlex 755MT/2.0G Duo Core/1G Memory/80G HD/19" LCD/16xDVD/WinXP Pro	QXX0258900	Ea	1	1	16,998,540.00	<b>16,998,540.00</b>
<b>Local Material</b>						
Rectifier for 12 A		Ea	1	1	15,112,500.00	<b>15,112,500.00</b>
Rack 19"		Ea	1	1	1,450,800.00	<b>1,450,800.00</b>
Grounding		Lot	1	1	6,045,000.00	<b>6,045,000.00</b>
Optical Distribution Frame (ODF) 12 core		Ea	2	2	2,430,090.00	<b>4,860,180.00</b>
Installation Material		Lot	1	1	108,326,400.00	<b>108,326,400.00</b>
<b>TOTAL</b>						<b>712,995,660.00</b>

#### 4.2.2 Penggelaran GEAPON Pada STO Slipi

Berdasarkan Tabel 4.2 di atas, maka terminasi dari fiber optik tidak akan menggunakan ONU melainkan menggunakan modul MDU untuk VDSL2. Modul MDU ini akan merubah sinyal optik menjadi informasi yang disalurkan melalui tembaga kerumah – rumah. 1 modul MDU memerlukan 1 input inti optik, dan dapat mendistribusikannya menjadi 64 output VDSL2. Untuk transmisi tembaga kerumah – rumah tidak ada lagi pengadaan kabel tembaga baru, karena kabel tembaga existing dapat digunakan kembali. Sedangkan pengadaan modem VDSL2 dibebankan kepelanggan sehingga tidak dihitung sebagai salah satu alat produksi yang harus disediakan. Untuk harga dari MDU ini belum ada vendor yang dapat memberikan harganya. Pada Gambar 4.3 diberikan skema penggelaran GEAPON + VDSL2 pada STO Slipi.



Gambar 4.2 Penggelaran GEAPON + VDSL2 pada STO Slipi

#### 4.3 PEMENUHAN KEBUTUHAN ANTARMUKA PON

Pada Bab 4.1 telah dilakukan sistesis dari aplikasi untuk memberikan alternatif beberapa solusi untuk RA dan RAV, dari hasil tersebut semua DP dicatu sebesar 450 Mbps. Sentral STO Slipi adalah letak dari perangkat OLT maka jumlah kebutuhan *bit rate* dari pelanggan yang harus disiapkan di STO Slipi adalah :

Tabel 4.3 Kebutuhan *Bit rate* RA dan RAV

Jumlah DP	<i>bit rate</i>	Total
128	450	57600

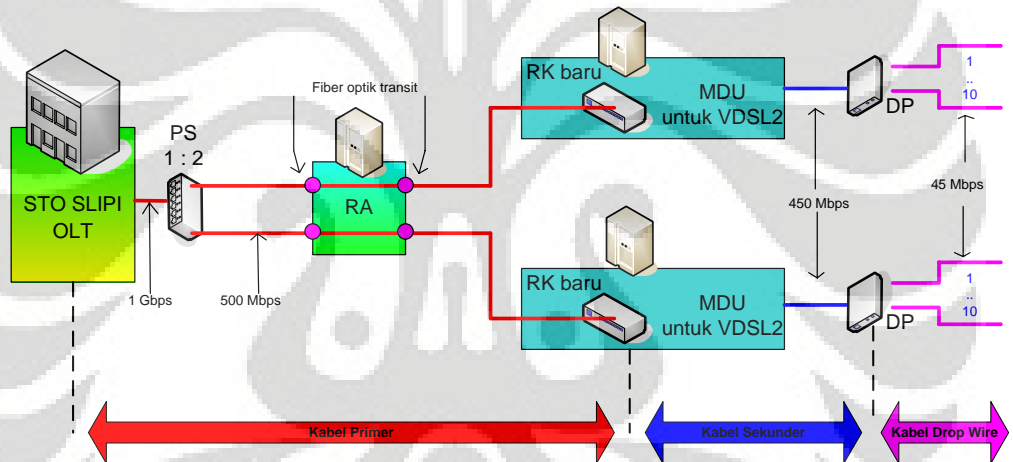
Menunjuk model kasus bisnis NEC di atas, maka 1 Antarmuka PON terdiri dari 12 inti optik yang dapat digunakan untuk melayani MDU VDSL2. 12 inti setara dengan 12000 Mbps atau apabila kebutuhan *Bit rate* RA dan RAV adalah 57600 Mbps maka dibutuhkan 5 Antarmuka PON (perkiraan awal). MDU



yang digunakan menerima 1 catuan GEPON dan mendistribusikannya menjadi 64 port VDSL2 ke arah pelanggan.

#### 4.3.1 Pembagi Pasif

Dengan catuan per DP membutuhkan *Bit rate* sebesar 450 Mbps, 1 inti jaringan optik adalah 1 Gbps, dan 1 Modul MDU membutuhkan 1 inputan GEPON, maka untuk variasi pembagi pasif yang mungkin hanyalah 1 inti dibagi 2. Dengan demikian maka Kabel Primer fiber optik berawal dari OLT STO Slipi sampai dengan MDU di RK baru. Sedangkan Kabel Sekunder tembaga berawal dari MDU sampai ke rumah – rumah pelanggan. Sedangkan RA (RK yang sudah ada sebelumnya) berfungsi sebagai tempat transit dari fiber optik yang menuju MDU. Untuk lebih jelasnya dapat mengacu kepada Gambar 4.4 dibawah ini.



Gambar 4.3 Pembagi Pasif

#### 4.3.2 Pemetaan Inti Optik Pada Antarmuka PON

Sebagaimana telah disebutkan pada Bab 4.3 , untuk mencatu RA dan RAV diperlukan 5 Antarmuka PON. Dari 5 Antarmuka PON tersebut dapat mendistribusikan 60 inti optik yang dapat diteruskan untuk mencatu RK yang ada. Inti optik dari 5 Antarmuka PON diurutkan dari mulai 1 sampai dengan 60. Langkah selanjutnya adalah menentukan inti optik mana yang mencatu suatu RK.

#### 4.3.2.1 Jatah Inti Optik Untuk RA

Untuk pembagian ke inti optik ke jaringan RK, maka harus dilihat kembali peta logical hasil sistesis aplikasi. Pengelompokkan RK baru berdasarkan catuan primer yang keluar dari RA. Setelah pengelompokkan RK baru telah selesai, kemudian dipetakan kebutuhan RK berdasarkan kebutuhan *Bit ratenya* setelah dikurangi DP yang tidak layak diberikan catuan VDSL2. Dengan hasil Tabel 4.4 dibawah ini :

Tabel 4.4 Pemetaan Inti optik Untuk Catuan RA

RK ID	Bit rate	Kurang	Bit rate Akhir	Jumlah MDU	Inti ID
1	1350	450	900	1	1
2	1800	0	1800	2	2-3
6	3150	900	2250	3	4-5-6
7	900	450	450	1	7
15	900	450	450	1	7
3	900	450	450	1	8
23	900	0	900	1	9
14	1800	0	1800	2	10-11
12	1800	450	1350	2	12-13
25	900	0	900	1	14
11	1800	0	1800	2	15-16
10	1350	0	1350	2	17-18
5	900	0	900	1	19
13	450	0	450	1	20
4	450	0	450	1	20
16	450	0	450	1	21
8	450	0	450	1	21
19	450	0	450	1	22
17	1350	900	450	1	22
20	1350	0	1350	2	23-24
18	1800	450	1350	2	25-26
21	2250	450	1800	2	27-28
9	900	0	900	1	29
22	1800	0	1800	2	30-31
26	1350	0	1350	2	32-33
24	1350	0	1350	2	34-35

Penjelasan untuk tabel di atas adalah sebagai berikut :

- RK ID adalah identitas RK , dimana warna selang seling menggambarkan pengelompokkan RK berdasarkan diagram Pohon Steiner pada skema gambar logik.
- *Bit Rate* adalah jumlah kebutuhan total *Bit Rate* yang digunakan untuk mencatu DP – DP dibawahnya.
- *Bit Rate* Akhir adalah jumlah kebutuhan total *Bit Rate* yang digunakan untuk mencatu DP – DP dibawahnya setelah dikurangi DP yang tidak layak diberikan catuan VDSL2.
- Jumlah MDU adalah kebutuhan modul perangkat MDU yang diperlukan ditempatkan didalam RK.
- Inti ID adalah identitas fiber optik yang berasal dari OLT menuju ke MDU, sedangkan pemetaannya berdasarkan kemampuan transmisi GEAPON (1 Gbps) dalam memenuhi kebutuhan *Bit Rate* RK.

#### 4.3.2.2 Jatah Inti Optik Untuk RAV

Untuk pembagian ke inti optik ke jaringan RK, maka harus dilihat kembali peta logical hasil sistesis aplikasi. Pengelompokkan RK baru berdasarkan catuan primer yang keluar dari RAV. Setelah pengelompokkan RK baru telah selesai, kemudian dipetakan kebutuhan RK berdasarkan kebutuhan *bit ratenya* setelah dikurangi DP yang tidak layak diberikan catuan VDSL2. Dengan hasil Tabel 4.5 dibawah ini :

Tabel 4.5 Pemetaan Inti optik Untuk Catuan RAV

RK ID	<i>Bit rate</i>	Kurang	<i>Bit rate</i> Akhir	Jumlah MDU	Inti ID
13	1800	0	1800	2	36-37
14	1800	450	1350	2	38-39
15	900	0	900	1	40
16	900	0	900	1	41
7	900	0	900	1	42
8	2250	450	1800	2	43-44
9	2250	0	2250	3	45-46-47
11	2250	900	1350	2	48-49
12	1800	450	1350	2	50-51
1	1350	0	1350	2	52-53
2	1800	900	900	1	54

3	1350	450	900	1	55
5	1800	450	1350	2	56-57
6	1800	0	1800	2	58-59
10	900	0	900	1	60
17	450	0	450	1	61
4	450	0	450	1	61

#### 4.3.2.3 Jumlah Pembagi Pasif Pada Antarmuka PON

Dari hasil sebelumnya, dari perkiraan 5 Antarmuka PON yang dibutuhkan untuk mencatu RA dan RAV (1 Antarmuka PON = 12 inti optik) ternyata berubah menjadi 6 Antarmuka PON. Hal ini dapat dilihat pada RK Terakhir pada daerah catuan RAV dimana mendapatkan bagian inti optik nomor 61. Selanjutnya pemetaan inti optik pada Antarmuka PON adalah sebagaimana tercantum pada Tabel 4.6 dibawah ini :

Tabel 4.6 Antarmuka PON Pada OLT

Antarmuka PON	inti ID	Menuju	Pembagi Pasif 1 : 2
1	1	RA	-
1	2	RA	-
1	3	RA	-
1	4	RA	-
1	5	RA	-
1	6	RA	-
1	7	RA	1
1	8	RA	-
1	9	RA	-
1	10	RA	-
1	11	RA	-
1	12	RA	-
2	13	RA	-
2	14	RA	-
2	15	RA	-
2	16	RA	-
2	17	RA	-
2	18	RA	-
2	19	RA	-
2	20	RA	1
2	21	RA	1
2	22	RA	1
2	23	RA	-

2	24	RA	-
3	25	RA	-
3	26	RA	-
3	27	RA	-
3	28	RA	-
3	29	RA	-
3	30	RA	-
3	31	RA	-
3	32	RA	-
3	33	RA	-
3	34	RA	-
3	35	RA	-
3	36	RAV	-
4	37	RAV	-
4	38	RAV	-
4	39	RAV	-
4	40	RAV	-
4	41	RAV	-
4	42	RAV	-
4	43	RAV	-
4	44	RAV	-
4	45	RAV	-
4	46	RAV	-
4	47	RAV	-
5	48	RAV	-
5	49	RAV	-
5	50	RAV	-
5	51	RAV	-
5	52	RAV	-
5	53	RAV	-
5	54	RAV	-
5	55	RAV	-
5	56	RAV	-
5	57	RAV	-
5	58	RAV	-
5	59	RAV	-
5	60	RAV	-
6	61	RAV	1
6	62	idle	-
6	63	idle	-
6	64	idle	-
6	65	idle	-
6	66	idle	-

6	67	idle	-
6	68	idle	-
6	69	idle	-
6	70	idle	-
6	71	idle	-
6	72	idle	-

Penjelasan Tabel adalah sebagai berikut :

- Antarmuka PON memberikan informasi kebutuhan Antarmuka PON yang dibutuhkan untuk mencatu RA dan RAV.
- Inti ID adalah identitas dari inti optik yang dibutuhkan untuk mencatu daerah RK.
- Pembagi Pasif adalah antarmuka pembagi pasif yang ditempatkan pada OLT untuk membagi inti optik menjadi 2 kabel optik.

#### 4.4 EFEKTIVITAS PEMAKAIAN ALAT PRODUKSI

Kebutuhan Antarmuka PON dapat dicari dengan membagi total kebutuhan *Bit rate* dengan output maksimal dari Antarmuka PON (12 Gbps). Hal ini cukup efektif karena hasil pemetaan inti optik pada Antarmuka PON untuk mencatu RA dan RAV hampir tepat. Dari 60 total inti yang tersedia, ternyata dibutuhkan 61 inti untuk mencatu RA dan RAV sehingga dibutuhkan 6 Antarmuka PON.

Dengan demikian , hasil pemetaan inti optik di Antarmuka PON semakin menguatkan pernyataan pada Bab 4.1. Dimana dengan menghasilkan catuan DP VDSL2 > 80% akan menyebabkan pemetaan inti optik di Antarmuka PON cukup efektif.

Dengan hasil di atas, maka efektifitas pemakaian *Bit rate* di daerah catuan RK adalah total kebutuhan *Bit rate* RK dibagi kemampuan maksimal catuan inti optik. Hal ini tercantum pada Tabel 4.7 dan 4.8. Semakin efektif pemakaian *Bit rate* maka semakin efektif alat produksi inti optik yang digunakan. Sedangkan sisa *Bit rate* dari inti optik yang belum dipakai merupakan potensial *Bit rate* yang dapat digunakan apabila ada penambahan pelanggan. Dengan kata lain jumlah RK berbanding lurus dengan inti optik berbanding lurus dengan jumlah Antarmuka PON yang dibutuhkan.

Tabel 4.7 Efektivitas *Bit rate* RA

Total Kebutuhan <i>Bit rate</i> Catuan RA =	27900
Total Kemampuan Catuan Inti Optik =	35000
Persentase Pemakaian <i>Bit rate</i> =	79.71%

Tabel 4.8 Efektivitas *Bit rate* RAV

Total Kebutuhan <i>Bit rate</i> Catuan RAV =	20700
Total Kemampuan Catuan Inti optik =	25000
Persentase Pemakaian <i>Bit rate</i> =	82.80%

#### 4.5 PEMAKAIAN ALAT PRODUKSI LAMA

Alat produksi lama yang dimaksud adalah *duct*, RK, DP, tiang dan kabel tembaga. Dalam penggelaran jaringan VDSL2 dari STO Slipi menuju RA dan RAV, jalur *duct* dari STO Slipi menuju RA dan RAV akan dipakai kembali. Begitu pula dengan struktur bangunan RA dan RAV akan dipakai kembali sebagai salah satu titik kontrol kabel primer. Sedangkan DP, tiang dan kabel tembaga yang lama juga akan dipakai seluruhnya karena memang tidak mengalami perubahan.

#### 4.6 BIAYA PENGGELARAN GEAPON

Untuk mencatu RA dan RAV dengan GEAPON dibutuhkan 6 Antarmuka PON yang membutuhkan biaya pengadaan sebesar Rp. 4,278 Milyar.

## **BAB V**

### **KESIMPULAN**

Dari analisa pada Bab 4 didapatkan hasil sebagaimana berikut :

1. Algoritma Heuristik Pencarian Scatter , Pohon Steiner dan Dijkstra dapat digunakan untuk mendesain jaringan VDSL2 pada daerah catuan RA dan RAV. Untuk RA , 84,93% DP tercatu jaringan VDSL2 dengan baik sedangkan dengan rata – rata panjang kabel tembaga antara RK – DP adalah 446,62 meter. Sedangkan RAV , 83,64% DP tercatu jaringan VDSL2 dengan baik dengan rata – rata panjang kabel tembaga antara RK – DP adalah 424,67 meter.
2. Untuk mencatu RA dan RAV dengan jaringan VDSL2 diperlukan total kabel primer (fiber optik) sepanjang 33,579 km dan 15,06 km , total kabel sekunder tembaga sepanjang 32,603 km dan 23,536 km. Jumlah unit bangunan RK yang baru adalah 26 untuk RA dan 17 untuk RAV.
3. Untuk *Duct* dari STO Slipi menuju RA dan RAV dipakai 100%. RK lama yaitu RA dan RAV dipakai 100%. Kabel sekunder tembaga yang sudah ada dapat dipakai kembali untuk menghubungkan RK baru dan DP. DP dan tiang telepon lama juga digunakan kembali tanpa ada yang dirubah letak dan posisinya.
4. Untuk mencatu RA dan RAV dengan GEPON + VDSL2 diperlukan 6 Antarmuka PON dimana terdapat 5 Pembagi Pasif yang ditempatkan pada OLT. 1 Pembagi Pasif pada Antarmuka PON No.1 , 3 Pembagi Pasif pada Antarmuka PON No.2 dan 1 Pembagi Pasif pada Antarmuka PON No.6.
5. Untuk RA dan RAV terdapat sisa *Bit Rate* yang dapat dialokasikan kepada pelanggan baru sebanyak 7100 Mbps dan 4300 Mbps. Sehingga apabila ada pelanggan baru yang meminta layanan multimedia berbasis jaringan VDSL2 tidak perlu ada pengadaan jalur fiber optik baru. Cukup mencari DP dan RK terdekat dengan lokasi rumah kemudian dilakukan catuan kabel tembaga dari RK dan DP tersebut.



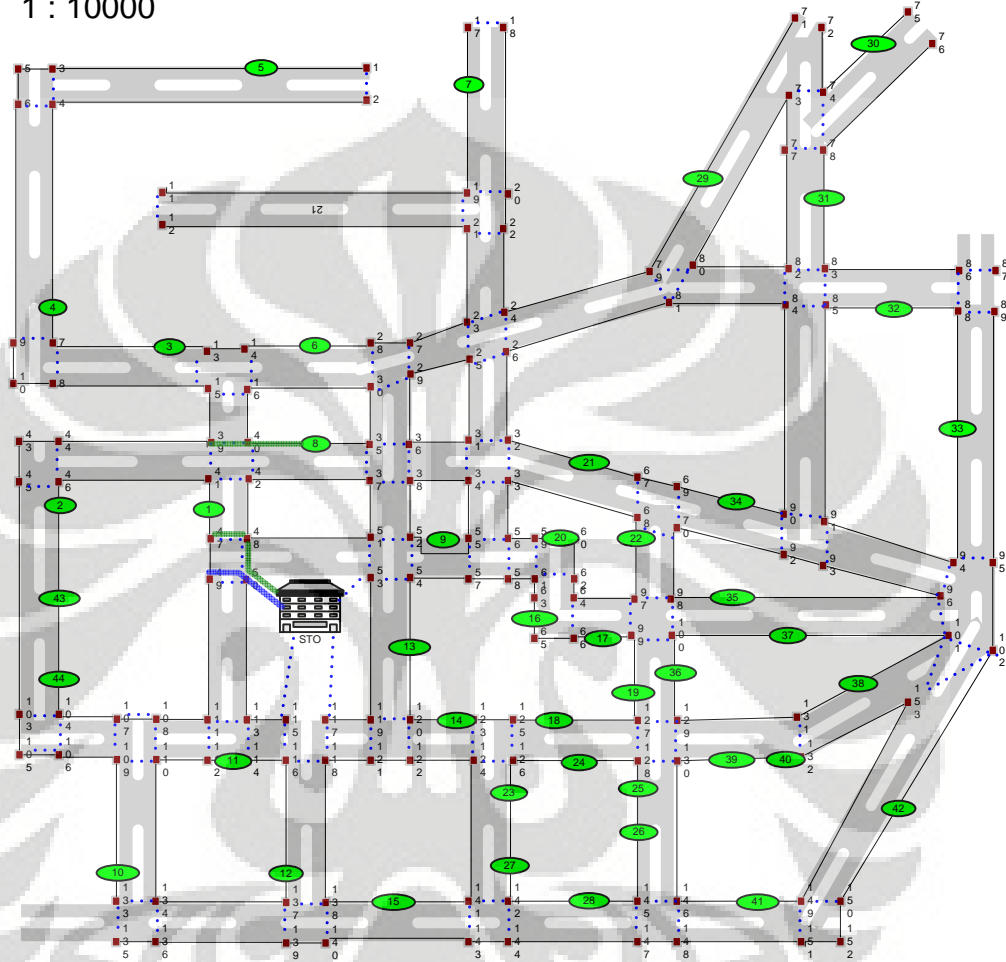
## DAFTAR ACUAN

- [1] Sri Kusumadewi , Hari Purnomo, *Penyelesaian Masalah Optimasi dengan Teknik – teknik Heuristik* (Yogyakarta : Penerbit Graha Ilmu , 2005).
- [2] Cesar Rego , Pedro Leao, “A Scatter Search Tutorial for Graph-Based Permutation Problems”, *University of Mississippi*, 2001, hal 7 – 18.
- [3] Derek R. Dreyer, Michael L. Overton, “Two Heuristics for the Euclidean Steiner Tree Problem” , 30 September 2002, hal 2 – 6.
- [4] Hsin-Hsiung Huang, De-Jing Huang , Tsai-Ming Hsieh, “Rectilinear Steiner Tree Construction with Obstacles, Institute of Electronic Engineering”, *Chung Yuan University, Taiwan*, 2003, hal 2.
- [5] Universidad De Malaga, “The VRP Web”, *Universidad De Malaga*, Maret 2007.
- [6] Dean Grumlose , “VDSL for MxU’s: The Fast Track to Interactive BroadbandVDSL, using existing copper, can give MDU owners years of breathing room”, *Campus Connection, Ikanos Communications*, Juli 2005, hal 2.
- [7] Piyush Sevalia, “Swimming Upstream: The Case for Higher Speeds”, *Ikanos Communications* , 2006, hal 8.
- [8] Telkom Divisi Infratel UNR Jakarta Barat, “RA dan RAV Slipi.wmf”, *Telkom Indonesia*, 2007.
- [9] I Nyoman Sutapa, I Gede Agus Widyadana, Christine. “STUDI TENTANG TRAVELLING SALESMAN DAN VEHICLE ROUTING PROBLEM DENGAN TIME WINDOWS.”, Tesis, Jurusan Teknik Industri, Fakultas Teknologi Industri, Universitas Kristen Petra.
- [10] Ira Prasetyaningrum. “Penyelesaian Kombinasi Vehicle Routing Problem dan Container Loading Problem Menggunakan Algoritma Genetika”, Tugas Akhir urusan Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Surabaya.
- [11] Thomas Bucsics. “Metaheuristik Approaches For Designing Survivable Fiber Optic Networks”, Thesis, Vienna University of Technology .

# LAMPIRAN

Lampiran 1 : Gambar RX Sebagai RK Model

1 : 10000



Lampiran 2 : Tabel Koordinat Titik Jalan Untuk RX

X	Y	type	ID
75	831	1	1
75	780	1	2
-408	828	1	3
-408	774	1	4
-462	828	1	5
-462	774	1	6
-408	405	1	7
-408	342	1	8
-471	405	1	9
-471	342	1	10
-240	639	1	11
-240	588	1	12
-171	393	1	13

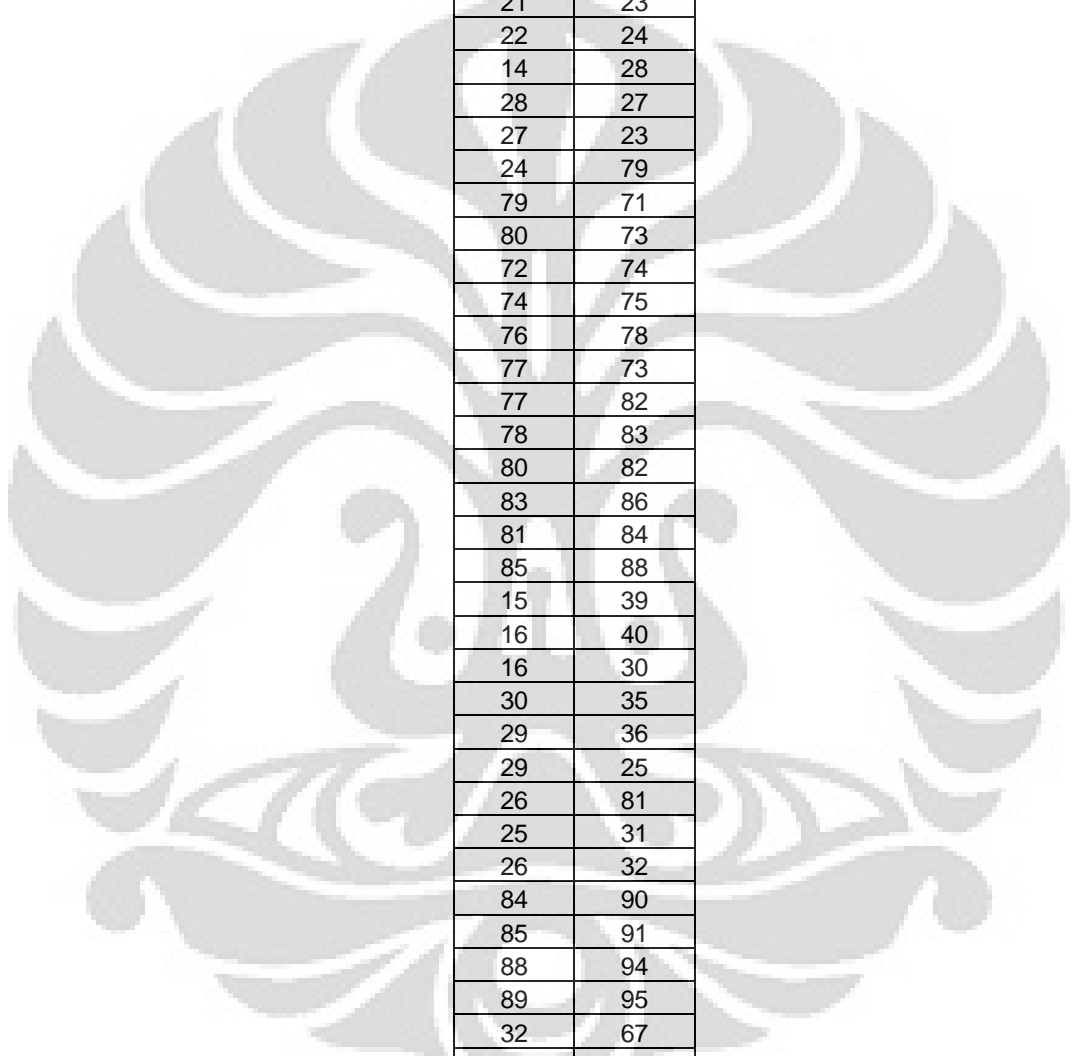
-111	396	1	14
-168	336	1	15
-108	333	1	16
231	894	1	17
285	894	1	18
228	636	1	19
291	636	1	20
228	582	1	21
285	582	1	22
228	438	1	23
285	453	1	24
234	381	1	25
288	393	1	26
141	405	1	27
81	405	1	28
141	357	1	29
81	339	1	30
231	255	1	31
291	255	1	32
291	192	1	33
231	192	1	34
78	249	1	35
138	249	1	36
78	192	1	37
141	192	1	38
-162	252	1	39
-108	252	1	40
-168	195	1	41
-105	195	1	42
-462	255	1	43
-399	255	1	44
-462	192	1	45
-399	162	1	46
-165	102	1	47
-108	102	1	48
-165	42	1	49
-111	39	1	50
81	105	1	51
141	105	1	52
81	42	1	53
141	42	1	54
231	105	1	55
291	105	1	56
231	42	1	57
291	42	1	58
333	105	1	59
396	105	1	60
333	42	1	61
396	42	1	62
333	12	1	63

393	12	1	64
333	-48	1	65
393	-48	1	66
492	198	1	67
492	135	1	68
552	183	1	69
552	120	1	70
735	909	1	71
777	894	1	72
726	789	1	73
780	792	1	74
912	918	1	75
948	867	1	76
720	705	1	77
780	705	1	78
510	516	1	79
585	525	1	80
540	468	1	81
726	522	1	82
783	522	1	83
720	465	1	84
783	465	1	85
1008	519	1	86
1047	519	1	87
999	456	1	88
1044	453	1	89
720	141	1	90
780	129	1	91
720	78	1	92
780	63	1	93
981	66	1	94
1041	66	1	95
960	15	1	96
489	9	1	97
543	9	1	98
483	-42	1	99
546	-42	1	100
975	-44.7	1	101
1041	-67.8	1	102
-461.7	-165	1	103
-400.8	-165	1	104
-459	-228	1	105
-399	-228	1	106
-310.8	-174	1	107
-249	-174	1	108
-309	-234	1	109
-249	-234	1	110
-171	-174	1	111
-171	-237	1	112
-108	-174	1	113

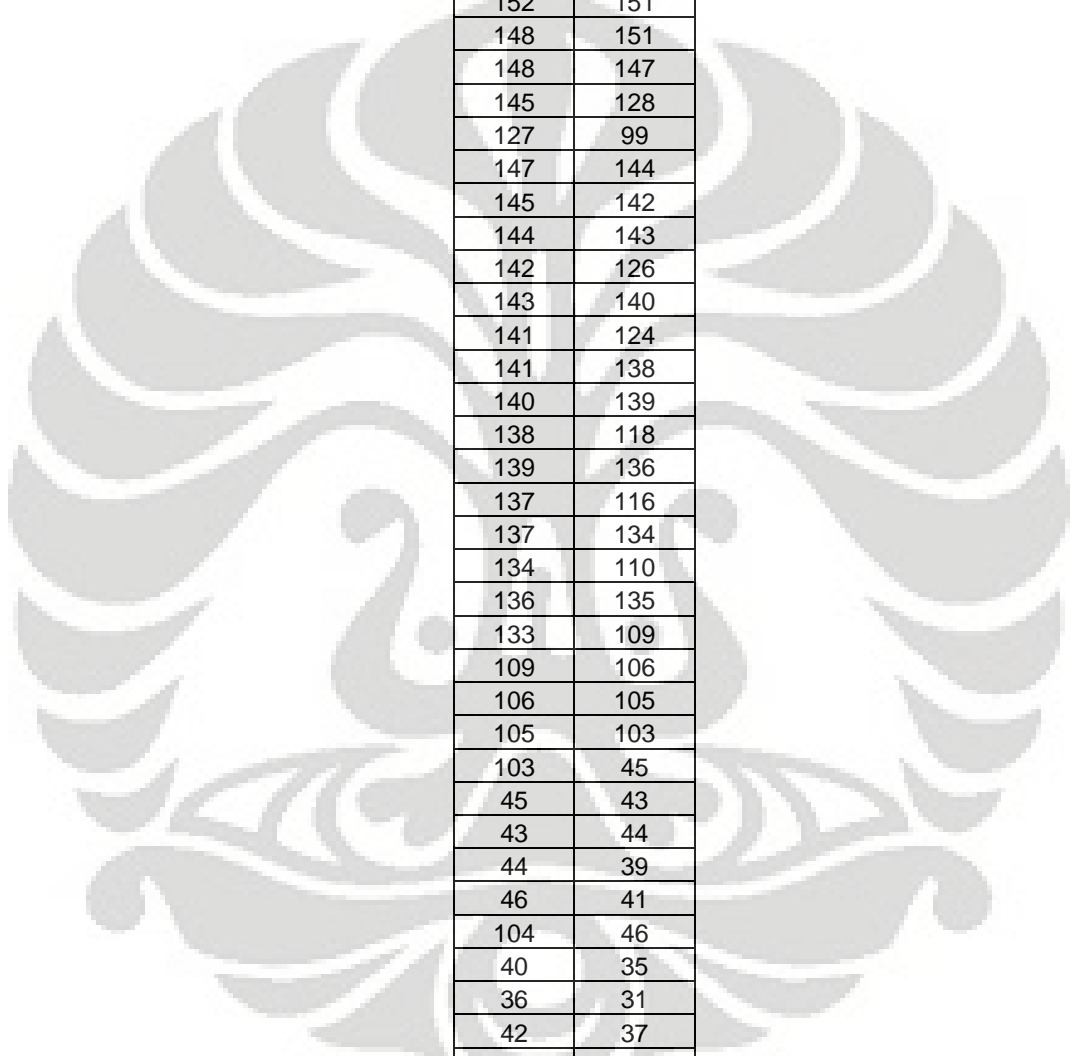
-108	-237	1	114
-49.8	-174	1	115
-49.8	-237	1	116
10.8	-174.9	1	117
10.8	-237	1	118
81	-174	1	119
141	-174	1	120
81	-237	1	121
141	-237	1	122
240	-174	1	123
240	-237	1	124
300	-174	1	125
300	-237	1	126
492	-174	1	127
492	-237	1	128
555	-174	1	129
555	-237	1	130
738	-168	1	131
747	-232.2	1	132
-312	-453	1	133
-249	-453	1	134
-312.3	-516	1	135
-249	-516	1	136
-49.2	-456	1	137
11.7	-457.2	1	138
-48	-519	1	139
9	-519	1	140
231	-453	1	141
291	-453	1	142
231	-516	1	143
291	-516	1	144
492	-453	1	145
552	-453	1	146
492	-516	1	147
552	-516	1	148
747	-453	1	149
807	-453	1	150
747	-516	1	151
807	-516	1	152
912	-147	1	153

**Lampiran 3 : Tabel Link Jalan Untuk RX**

A	B
1	3
2	4
3	5
5	6
6	9
4	7
9	10



7	13
10	8
8	15
13	14
17	19
19	11
18	20
21	12
20	22
21	23
22	24
14	28
28	27
27	23
24	79
79	71
80	73
72	74
74	75
76	78
77	73
77	82
78	83
80	82
83	86
81	84
85	88
15	39
16	40
16	30
30	35
29	36
29	25
26	81
25	31
26	32
84	90
85	91
88	94
89	95
32	67
67	69
69	90
91	94
95	102
96	93
93	92
92	70
33	68
96	98



101	131
100	101
102	150
131	129
132	130
100	129
130	146
149	146
150	152
152	151
148	151
148	147
145	128
127	99
147	144
145	142
144	143
142	126
143	140
141	124
141	138
140	139
138	118
139	136
137	116
137	134
134	110
136	135
133	109
109	106
106	105
105	103
103	45
45	43
43	44
44	39
46	41
104	46
40	35
36	31
42	37
38	34
41	47
42	48
48	51
37	51
38	52
34	55
52	55
33	56

56	59
59	60
60	62
62	64
64	97
97	68
98	70
47	49
49	111
50	113
113	115
117	119
119	53
120	54
54	57
57	58
58	61
61	63
63	65
65	66
66	99
104	107
107	108
108	111
120	123
123	125
125	127
110	112
112	114
114	116
118	121
121	122
122	124
126	128
132	153
149	153

**Lampiran 4 : Tabel Koordinat DP Untuk RX**

xx	yy	type	ID
-169	150	2	1
-399	157	2	2
-230	401	2	3
-411	463	2	4
-88	833	2	5
-5	404	2	6
234	807	2	7
-4	251	2	8



194	104	2	9
-310	-412	2	10
-132	-240	2	11
-50	-413	2	12
142	-63	2	13
215	-177	2	14
117	-457	2	15
336	-19	2	16
418	-49	2	17
366	-177	2	18
488	-134	2	19
375	106	2	20
420	223	2	21
493	106	2	22
296	-288	2	23
404	-242	2	24
496	-282	2	25
497	-349	2	26
297	-401	2	27
420	-455	2	28
596	661	2	29
860	870	2	30
783	631	2	31
891	460	2	32
990	275	2	33
647	163	2	34
642	15	2	35
554	-103	2	36
727	-45	2	37
836	-120	2	38
640	-238	2	39
724	-237	2	40
681	-457	2	41
898	-313	2	42
-400	-13	2	43
-401	-113	2	44

**Lampiran 5 : Tabel Kebutuhan Bit Rate DP Untuk RX**

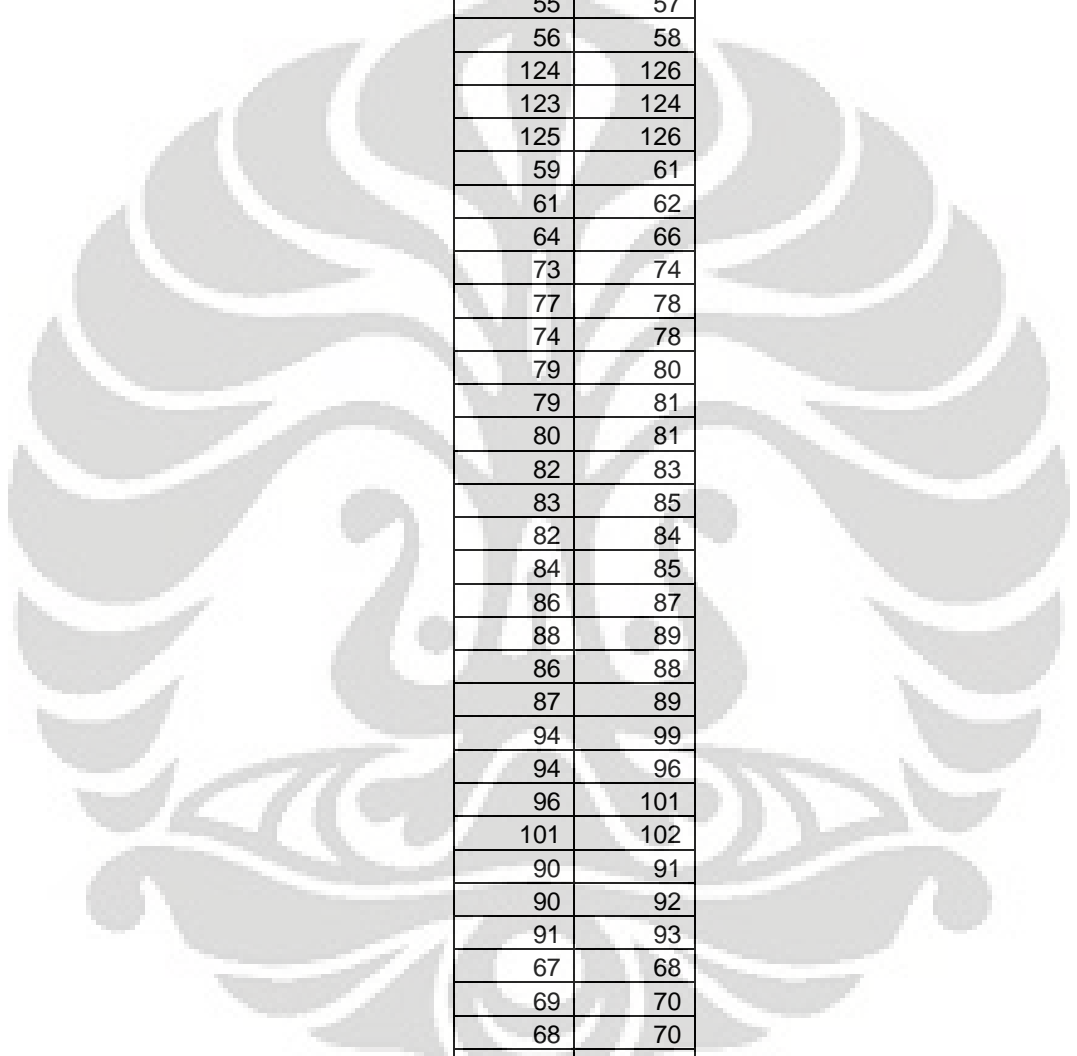
DP	Rate
1	450
2	450
3	450
4	450
5	450
6	450
7	450
8	450

9	450
10	450
11	450
12	450
13	450
14	450
15	450
16	450
17	450
18	450
19	450
20	450
21	450
22	450
23	450
24	450
25	450
26	450
27	450
28	450
29	450
30	450
31	450
32	450
33	450
34	450
35	450
36	450
37	450
38	450
39	450
40	450
41	450
42	450
43	450
44	450

**Lampiran 6 : Tabel Link Menyeberang Jalan Untuk RX**

A	B
1	2
3	4
4	6
7	9
7	8
11	12
17	18
19	20
19	21
21	22
13	15

15	16
14	16
44	46
45	46
40	42
39	41
39	40
41	42
47	48
48	50
49	50
103	104
104	106
105	106
107	108
109	110
107	109
108	110
133	134
133	135
134	136
111	112
113	114
111	113
154	115
154	117
115	116
117	118
116	118
154	53
137	138
137	139
138	140
119	120
121	119
120	122
141	143
141	142
142	144
28	30
27	29
30	29
35	36
35	37
36	38
37	38
51	52
52	53
52	54
53	54

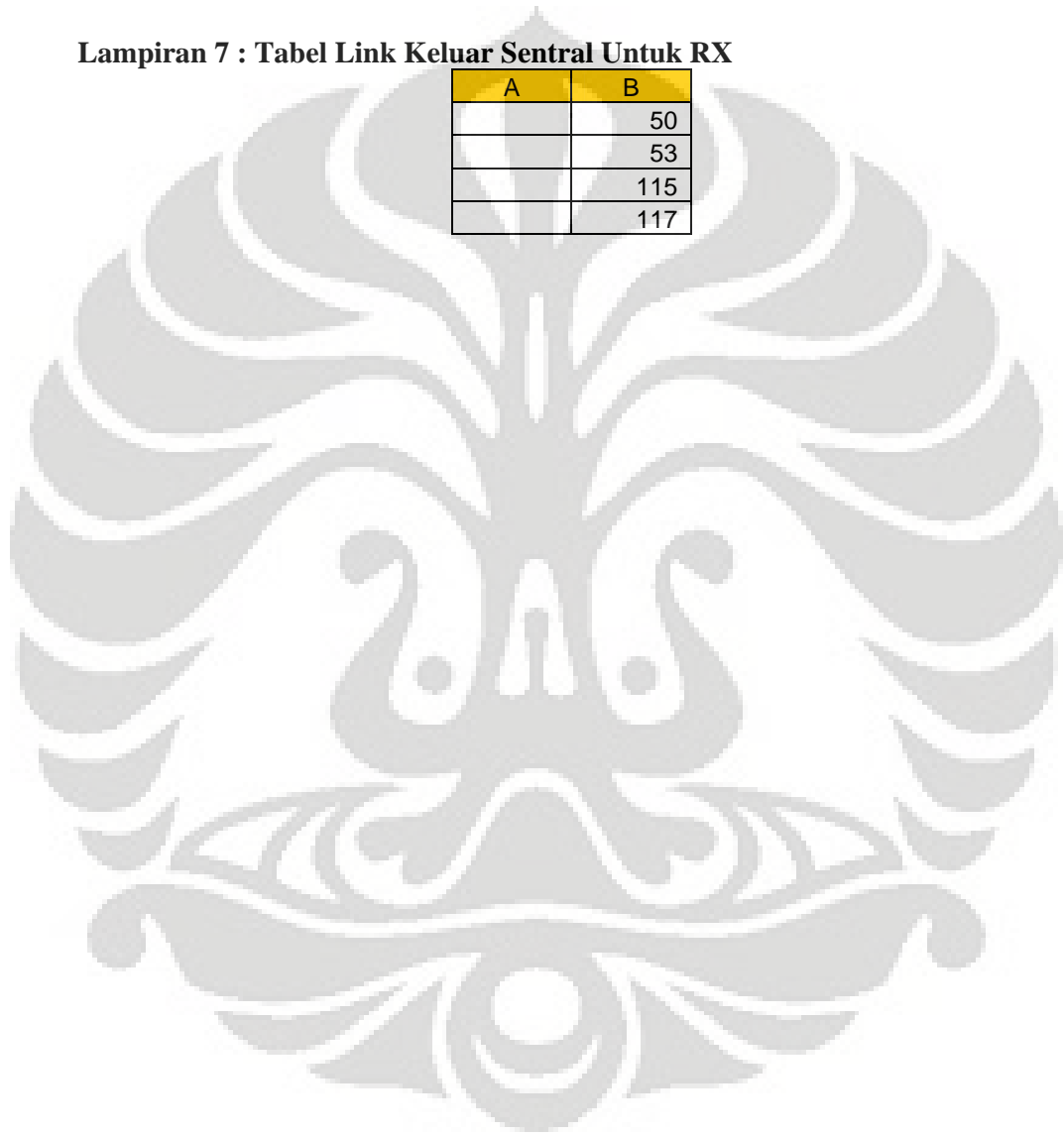


23	25
24	26
23	24
25	26
31	32
31	34
32	33
34	33
55	56
55	57
56	58
124	126
123	124
125	126
59	61
61	62
64	66
73	74
77	78
74	78
79	80
79	81
80	81
82	83
83	85
82	84
84	85
86	87
88	89
86	88
87	89
94	99
94	96
96	101
101	102
90	91
90	92
91	93
67	68
69	70
68	70
97	98
97	99
98	100
99	100
127	129
127	128
129	130
128	130
145	146

145	147
146	148
131	132
149	151
149	150
101	153
102	153

**Lampiran 7 : Tabel Link Keluar Sentral Untuk RX**

A	B
	50
	53
	115
	117





**Lampiran 9 : Tabel Koordinat Titik Jalan RA**

X	Y	type	ID
-1480	519	1	1
-1410	521	1	2
-1310	549	1	3
-1235	550	1	4
-1160	526	1	5
-1480	464	1	6
-1385	465	1	7
-1295	489	1	8
-1160	466	1	9
-1135	290	1	10
-1490	290	1	11
-1490	230	1	12
-1125	230	1	13
-1475	124	1	14
-1110	124	1	15
-1475	64	1	16
-1110	66	1	17
-1050	64	1	18
-1050	120	1	19
-1100	470	1	20
-1040	546	1	21
-1035	484	1	22
-1075	294	1	23
-1065	230	1	24
-970	536	1	25
-970	476	1	26
-1010	20	1	27
-945	20	1	28
-1010	-40	1	29
-945	-40	1	30
-915	16	1	31
-865	14	1	32
-920	-40	1	33
-860	-60	1	34
-920	-170	1	35
-860	-160	1	36
-920	-230	1	37
-860	-236	1	38
-920	-300	1	39
-860	-300	1	40
-1015	-170	1	41
-1035	-230	1	42
-1095	-126	1	43
-1105	-184	1	44
-1230	-124	1	45
-1220	-186	1	46
-1295	-150	1	47

-1295	-216	1	48
-1350	-140	1	49
-1440	-140	1	50
-1440	-196	1	51
-1355	-200	1	52
-935	526	1	53
-935	466	1	54
-875	520	1	55
-875	460	1	56
-875	186	1	57
-935	190	1	58
-830	520	1	59
-775	524	1	60
-830	460	1	61
-770	460	1	62
-835	154	1	63
-775	160	1	64
-390	520	1	65
-395	464	1	66
-720	244	1	67
-660	239	1	68
-765	-235	1	69
-705	-234	1	70
-770	-294	1	71
-705	-294	1	72
-620	-236	1	73
-605	-294	1	74
-620	154	1	75
-560	154	1	76
-560	-220	1	77
-550	-280	1	78
-380	644	1	79
-285	634	1	80
-100	804	1	81
-75	754	1	82
15	806	1	83
-10	756	1	84
-290	586	1	85
-295	530	1	86
-225	586	1	87
-195	534	1	88
-135	676	1	89
-115	620	1	90
-75	674	1	91
-15	670	1	92
-35	620	1	93
10	584	1	94
45	626	1	95
75	536	1	96
105	586	1	97

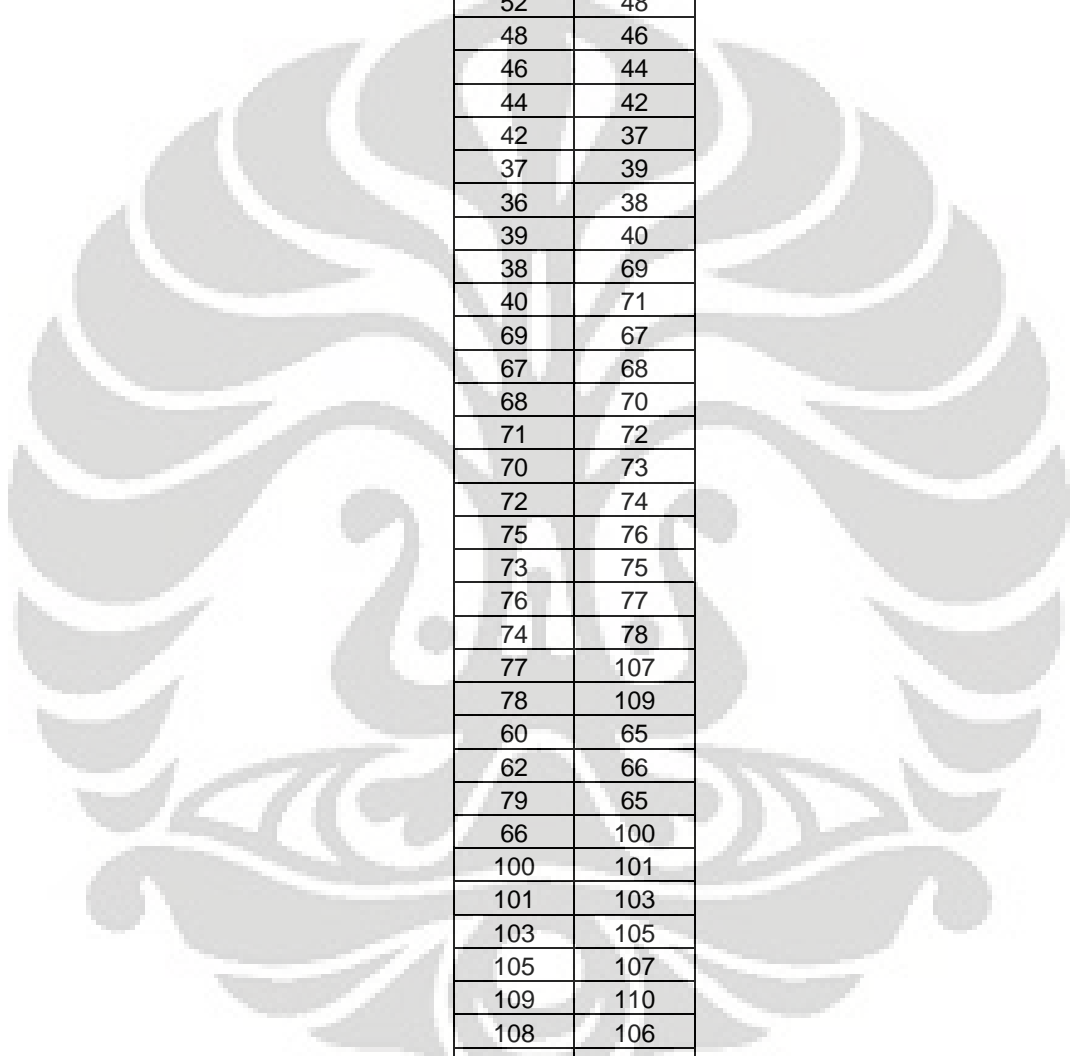


70	264	1	98
-320	266	1	99
-415	270	1	100
-420	206	1	101
-325	200	1	102
-430	94	1	103
-335	96	1	104
-435	40	1	105
-340	36	1	106
-455	-194	1	107
-355	-164	1	108
-465	-254	1	109
-360	-224	1	110
-265	-164	1	111
-270	-216	1	112
-200	-160	1	113
-200	-220	1	114
-260	-40	1	115
-200	-40	1	116
140	496	1	117
205	516	1	118
10	204	1	119
70	200	1	120
140	264	1	121
140	200	1	122
200	254	1	123
200	194	1	124
145	94	1	125
200	96	1	126
145	34	1	127
200	34	1	128
140	-164	1	129
200	-164	1	130
155	-220	1	131
200	-220	1	132
265	564	1	133
325	594	1	134
270	-166	1	135
270	-224	1	136
325	-164	1	137
330	-224	1	138
10	264	1	139
-1200	480	1	140
-1180	126	1	141
-1265	126	1	142
-880	326	1	143
-835	266	1	144
-690	-80	1	145
-495	466	1	146
-560	-54	1	147

-255	534	1	148
-160	95	1	149
-175	35	1	150
25	94	1	151
80	686	1	152
145	646	1	153
195	614	1	154

**Lampiran 10 : Tabel Link Jalan Untuk RA**

A	B
1	2
6	7
2	3
3	4
4	5
7	8
8	9
10	11
13	12
14	15
16	17
17	18
18	19
19	24
24	23
23	20
21	5
20	22
21	25
22	27
29	30
27	29
26	28
28	31
31	32
30	33
25	53
26	54
53	55
54	58
58	57
57	56
55	59
59	60
56	61
61	63
63	64
64	62
32	34
33	35



34	36
50	51
50	49
49	47
47	45
45	43
43	41
41	35
51	52
52	48
48	46
46	44
44	42
42	37
37	39
36	38
39	40
38	69
40	71
69	67
67	68
68	70
71	72
70	73
72	74
75	76
73	75
76	77
74	78
77	107
78	109
60	65
62	66
79	65
66	100
100	101
101	103
103	105
105	107
109	110
108	106
104	102
99	86
80	85
79	81
81	83
80	82
82	84
83	134
84	133

133	135
134	137
137	138
136	138
135	130
136	132
132	131
131	114
113	129
127	129
128	130
126	124
123	118
117	121
122	125
85	87
89	91
97	118
91	92
95	97
86	88
88	90
90	93
93	94
96	117
139	94
96	98
99	139
102	119
98	121
119	120
120	122
104	125
106	127
9	10
13	15
87	89
123	124
126	128
108	111
111	115
110	112
112	114
113	116
92	95
140	9
140	8
141	142
141	15
142	14

143	56
143	57
144	63
144	61
145	70
145	68
146	66
146	62
147	76
147	77
148	86
148	88
149	104
149	151
150	106
150	127
151	149
151	125
152	84
152	153
153	154
154	133

**Lampiran 11 : Tabel Koordinat Titik DP Untuk RA**

X	Y	type	ID
-1440	466	2	1
-1340	475	2	2
-1235	490	2	3
-1440	296	2	4
-1324	298	2	5
-1220	298	2	6
-1424	124	2	7
-1320	126	2	8
-1216	124	2	9
-1144	124	2	10
-1084	450	2	11
-1076	400	2	12
-1020	310	2	13
-1004	140	2	14
-872	420	2	15
-874	360	2	16
-874	300	2	17
-872	200	2	18
-832	250	2	19
-834	170	2	20
-704	460	2	21
-466	460	2	22
-1396	-200	2	23
-1256	-200	2	24
-1136	-184	2	25

-1080	-202	2	26
-1024	-224	2	27
-976	-226	2	28
-856	-40	2	29
-855	-190	2	30
-856	-90	2	31
-764	-220	2	32
-684	-55	2	33
-659	220	2	34
-559	-130	2	35
-559	10	2	36
-616	140	2	37
-496	-190	2	38
-444	-80	2	39
-420	120	2	40
-424	230	2	41
-404	390	2	42
-84	740	2	43
36	710	2	44
116	660	2	45
160	630	2	46
224	590	2	47
-216	530	2	48
-14	606	2	49
-1440	-140	2	50
184	530	2	51
10	420	2	52
10	340	2	53
200	400	2	54
-164	260	2	55
-216	200	2	56
-74	200	2	57
-16	200	2	58
196	270	2	59
-224	34	2	60
-120	96	2	61
-124	38	2	62
-20	96	2	63
-24	38	2	64
120	98	2	65
124	36	2	66
-116	-160	2	67
4	-160	2	68
136	-220	2	69
264	290	2	70
264	200	2	71
260	-90	2	72
260	-160	2	73

**Lampiran 12 : Tabel Kebutuhan Bit Rate DP Untuk RA**

DP	Rate
1	450
2	450
3	450
4	450
5	450
6	450
7	450
8	450
9	450
10	450
11	450
12	450
13	450
14	450
15	450
16	450
17	450
18	450
19	450
20	450
21	450
22	450
23	450
24	450
25	450
26	450
27	450
28	450
29	450
30	450
31	450
32	450
33	450
34	450
35	450
36	450
37	450
38	450
39	450
40	450
41	450
42	450
43	450
44	450
45	450
46	450
47	450
48	450

49	450
50	450
51	450
52	450
53	450
54	450
55	450
56	450
57	450
58	450
59	450
60	450
61	450
62	450
63	450
64	450
65	450
66	450
67	450
68	450
69	450
70	450
71	450
72	450
73	450

**Lampiran 13 : Tabel Link Menyeberang Jalan Untuk RA**

A	B
5	9
9	20
10	23
10	13
13	24
15	19
15	17
21	22
22	26
25	26
27	28
28	30
33	34
35	36
35	37
37	38
38	40
53	54
54	56
55	56
59	60



59	61
61	62
60	62
69	70
69	71
69	70
70	72
73	77
73	74
73	77
77	78
107	108
109	110
103	104
105	106
100	99
101	102
65	86
86	85
79	80
87	88
111	113
111	112
114	113
81	82
83	84
133	134
94	96
94	95
96	97
117	118
139	98
119	139
98	120
121	123
122	124
125	126
125	127
125	126
127	128
129	130
129	131
130	132
135	136
135	137

**Lampiran 14 : Tabel Link Keluar Semtral Untuk RA**

A	B
	115
	116



Lampiran 16 : Tabel Koordinat Titik Jalan Untuk RAV

X	Y	type	ID
-980	1661	1	1
-980	1635	1	2
-900	1660	1	3
-920	1621	1	4
-385	1601	1	5
-655	1655	1	6
-395	1641	1	7
-110	1635	1	8
-75	1589	1	9
-82	1641	1	10
-220	1596	1	11
-965	1431	1	12
-925	1441	1	13
-965	1409	1	14
-925	1415	1	15
-725	1429	1	16
-555	1421	1	17
-375	1415	1	18
-370	1391	1	19
-1265	1409	1	20
-1280	1415	1	21
-1295	1329	1	22
-1315	1334	1	23
-1300	1315	1	24
-1195	1301	1	25
-1195	1281	1	26
-1170	1295	1	27
-1420	1089	1	28
-1395	1081	1	29
-1425	1089	1	30
-1405	1055	1	31
-1510	866	1	32
-1485	850	1	33
-1200	1035	1	34
-1205	1001	1	35
-1180	1029	1	36
-1180	995	1	37
-1360	824	1	38
-1205	795	1	39
-1680	859	1	40
-1640	921	1	41
-1535	795	1	42
-1515	781	1	43
-1210	729	1	44
-1183	810	1	45
-1185	729.5	1	46
-995	809	1	47
-995	741	1	48

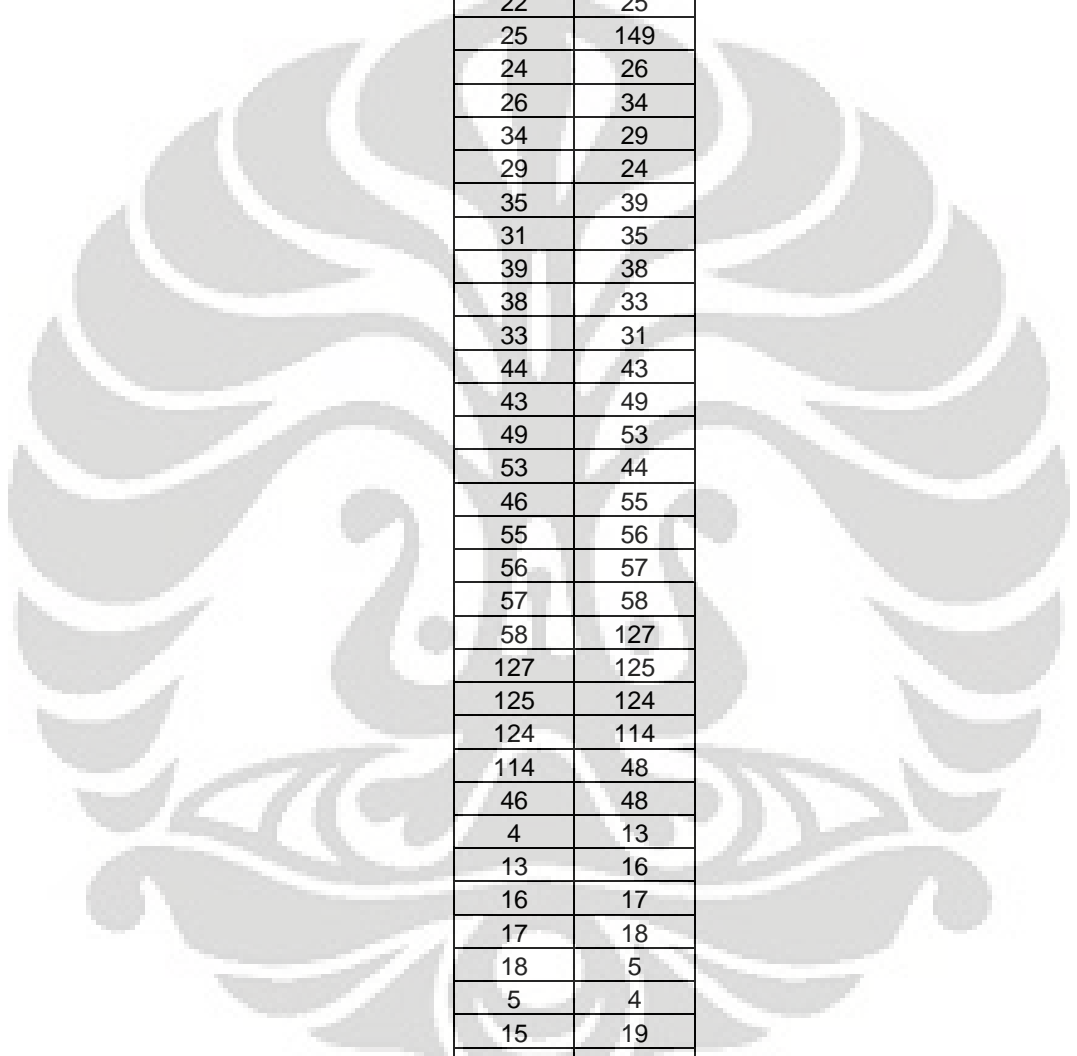
-1573	662	1	49
-1590	661	1	50
-1600	635	1	51
-1580	626	1	52
-1215	595	1	53
-1215	569	1	54
-1195	595	1	55
-1195	569	1	56
-1200	241	1	57
-1015	239	1	58
-1020	201	1	59
-980	235	1	60
-980	201	1	61
-975	169	1	62
-975	149	1	63
-985	-85	1	64
-1025	-85	1	65
-1220	215	1	66
-1025	-145	1	67
-995	-145	1	68
-585	-91	1	69
-215	-85	1	70
45	-151	1	71
40	-79	1	72
35	41	1	73
35	-20	1	74
20	209	1	75
5	369	1	76
-5	555	1	77
-25	809	1	78
-35	881	1	79
-240	795	1	80
-265	795	1	81
-55	1261	1	82
-65	1501	1	83
-295	869	1	84
-340	1595	1	85
-345	1641	1	86
-105	1595	1	87
-360	1249	1	88
-520	1241	1	89
-720	1239	1	90
-935	1235	1	91
-975	1236	1	92
-975	1211	1	93
-935	1206	1	94
-360	1215	1	95
-335	1415	1	96
-320	1215	1	97
-355	1081	1	98

-310	1085	1	99
-345	1055	1	100
-305	1055	1	101
-595	1055	1	102
-940	1035	1	103
-945	1009	1	104
-455	1069	1	105
-955	815	1	106
-845	815	1	107
-340	861	1	108
-955	741	1	109
-835	749	1	110
-575	769	1	111
-415	781	1	112
-335	789	1	113
-995	621	1	114
-965	621	1	115
-960	555	1	116
-660	569	1	117
-665	541	1	118
-555	581	1	119
-420	581	1	120
-255	595	1	121
-255	569	1	122
-960	529	1	123
-1000	529	1	124
-1005	369	1	125
-965	361	1	126
-1005	335	1	127
-965	335	1	128
-800	369	1	129
-500	381	1	130
-315	381	1	131
-245	389	1	132
-245	369	1	133
-515	175	1	134
-355	175	1	135
-235	181	1	136
-215	369	1	137
-205	181	1	138
-225	161	1	139
-200	161	1	140
-185	-85	1	141
-175	-151	1	142
135	-340	1	143
125	-79	1	144
-5	1621	1	145
-1275	1441	1	146
-1170	1419	1	147
-975	1020	1	148

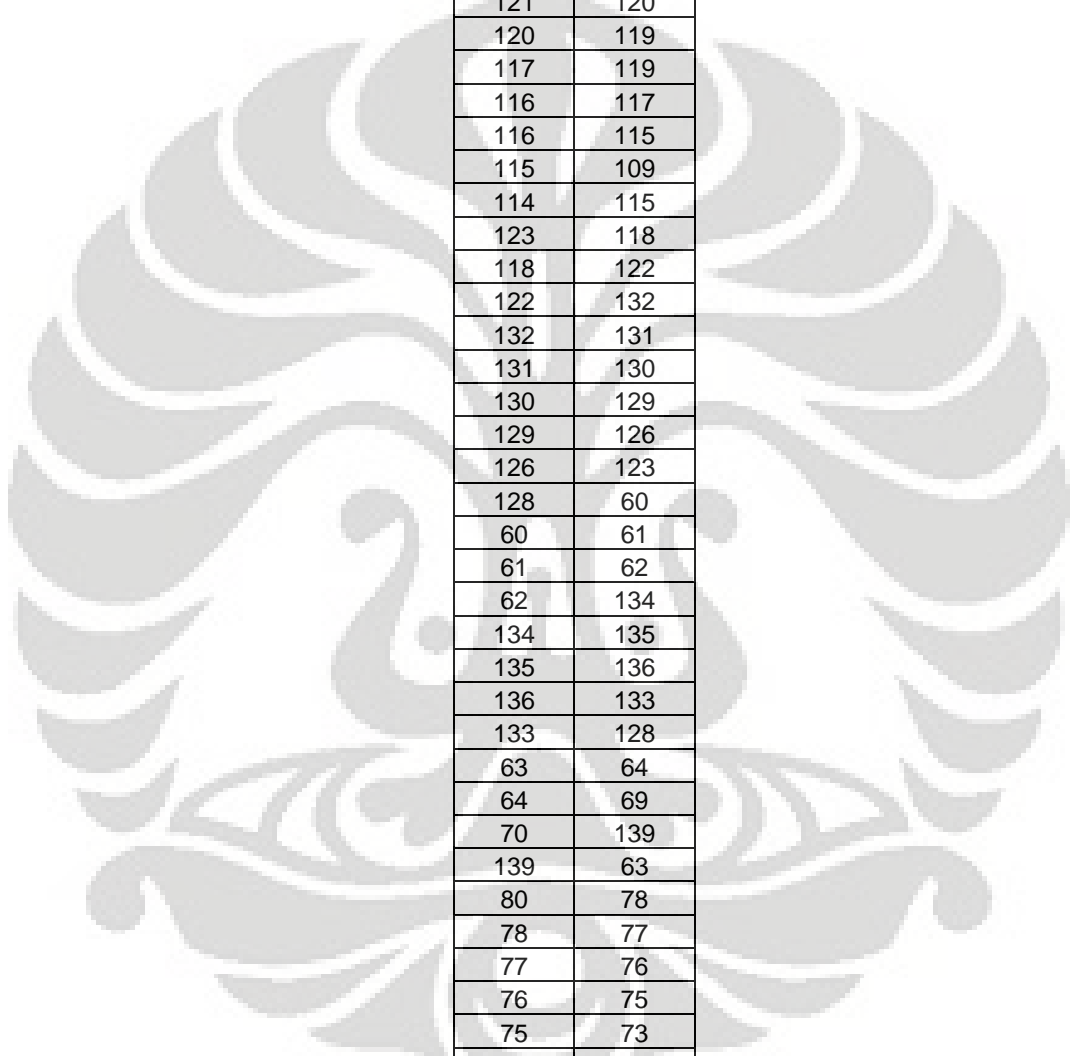
-1195	1420	1	149
-1055	812	1	150

**Lampiran 17 : Tabel Link Jalan Untuk RAV**

A	B
1	3
1	2
3	6
6	7
7	86
86	8
8	10
10	145
2	12
4	13
146	12
14	92
21	23
23	28
28	30
30	32
32	41
41	40
40	42
42	50
50	51
51	52
52	54
54	66
66	59
59	65
65	67
67	68
68	142
142	71
71	143
143	144
144	145
145	10
4	5
5	18
17	18
16	17
13	16
4	13
14	147
147	27
27	36
36	37



37	45
45	150
47	148
148	93
93	92
92	14
14	147
20	149
20	22
22	25
25	149
24	26
26	34
34	29
29	24
35	39
31	35
39	38
38	33
33	31
44	43
43	49
49	53
53	44
46	55
55	56
56	57
57	58
58	127
127	125
125	124
124	114
114	48
46	48
4	13
13	16
16	17
17	18
18	5
5	4
15	19
19	88
88	89
89	90
90	91
91	15
94	95
95	98
98	103
103	94



104	100
108	107
106	107
106	104
109	110
111	112
112	113
113	81
81	121
121	120
120	119
117	119
116	117
116	115
115	109
114	115
123	118
118	122
122	132
132	131
131	130
130	129
129	126
126	123
128	60
60	61
61	62
62	134
134	135
135	136
136	133
133	128
63	64
64	69
70	139
139	63
80	78
78	77
77	76
76	75
75	73
73	74
74	72
72	141
141	140
140	138
137	138
137	80
69	70
110	111



100	108
85	96
96	97
97	99
99	101
85	11
11	87
87	9
9	83
83	82
82	79
79	84
84	101
101	99
99	97
97	96
47	150
146	21
102	105
103	102

**Lampiran 18 : Tabel Koordinat Titik DP Untuk RAV**

X	Y	type	ID
-940	1660	2	1
-760	1655	2	2
-540	1650	2	3
-434	1645	2	4
-186	1595	2	5
-80	1655	2	6
-966	1445	2	7
-686	1430	2	8
-514	1425	2	9
-66	1525	2	10
-54	1275	2	11
-486	1245	2	12
-706	1240	2	13
-934	1245	2	14
-1274	1345	2	15
-1326	1065	2	16
-974	1185	2	17
-934	1125	2	18
-566	1065	2	19
-354	1105	2	20
-1406	830	2	21
-1326	815	2	22
-1026	805	2	23
-980	985	2	24
-814	825	2	25
-34	915	2	26
-1640	835	2	27

-1546	695	2	28
-1166	715	2	29
-880	745	2	30
-634	765	2	31
-520	775	2	32
-1000	635	2	33
-954	585	2	34
-620	575	2	35
-486	585	2	36
-346	590	2	37
-266	705	2	38
-794	535	2	39
-966	375	2	40
-834	365	2	41
-549	380	2	42
-354	385	2	43
-1106	235	2	44
-934	165	2	45
-474	175	2	46
-314	175	2	47
-986	75	2	48
-854	-155	2	49
-540	-95	2	50
34	145	2	51
26	275	2	52
6	455	2	53
-14	635	2	54
-226	105	2	55

**Lampiran 19 : Tabel Kebutuhan Bit Rate Untuk RAV**

DP	Rate
1	450
2	450
3	450
4	450
5	450
6	450
7	450
8	450
9	450
10	450
11	450
12	450
13	450
14	450
15	450
16	450
17	450
18	450

19	450
20	450
21	450
22	450
23	450
24	450
25	450
26	450
27	450
28	450
29	450
30	450
31	450
32	450
33	450
34	450
35	450
36	450
37	450
38	450
39	450
40	450
41	450
42	450
43	450
44	450
45	450
46	450
47	450
48	450
49	450
50	450
51	450
52	450
53	450
54	450
55	450

**Lampiran 20 : Tabel Link Menyeberang Jalan Untuk RAV**

A	B
1	2
2	4
3	4
7	5
5	85
86	85
10	145
146	21
21	20
149	147

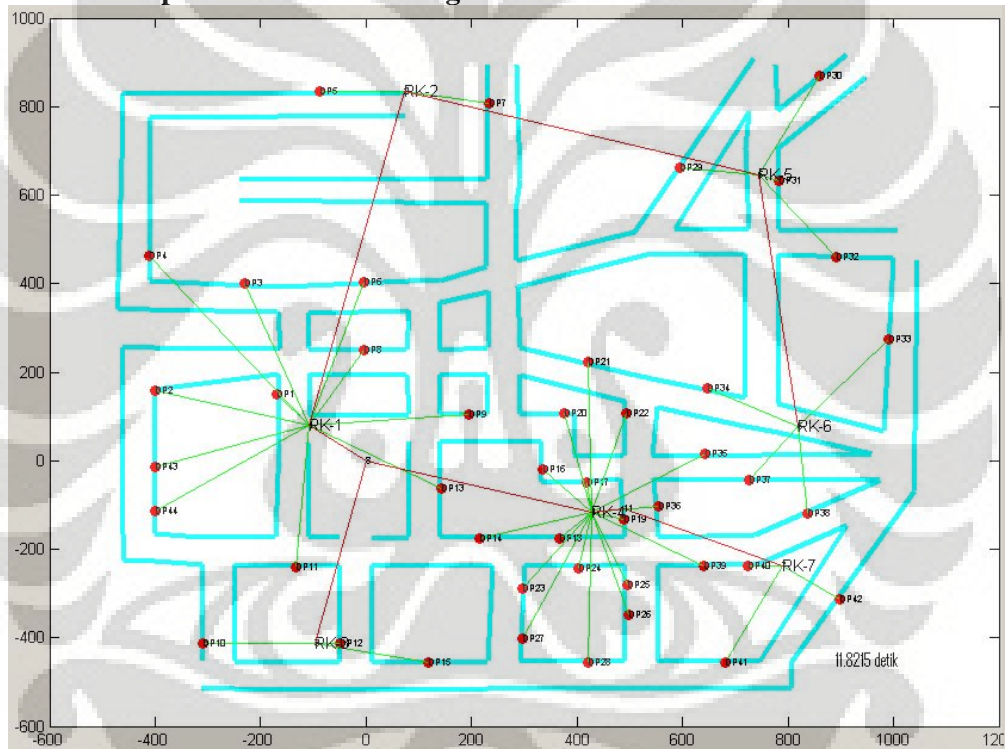
12	13
14	15
12	14
13	15
18	96
18	19
23	22
22	24
25	27
25	26
92	93
93	94
91	94
88	95
95	97
28	29
30	31
29	31
34	36
35	37
148	104
103	104
99	98
98	100
100	101
32	33
42	43
39	45
39	44
44	46
47	48
48	109
107	110
108	84
113	108
79	78
50	49
53	55
114	115
124	123
116	123
117	118
121	122
127	128
125	126
132	133
133	137
57	66
58	60
60	61

62	63
65	67
65	64
64	68
70	141
136	139
136	138
139	140

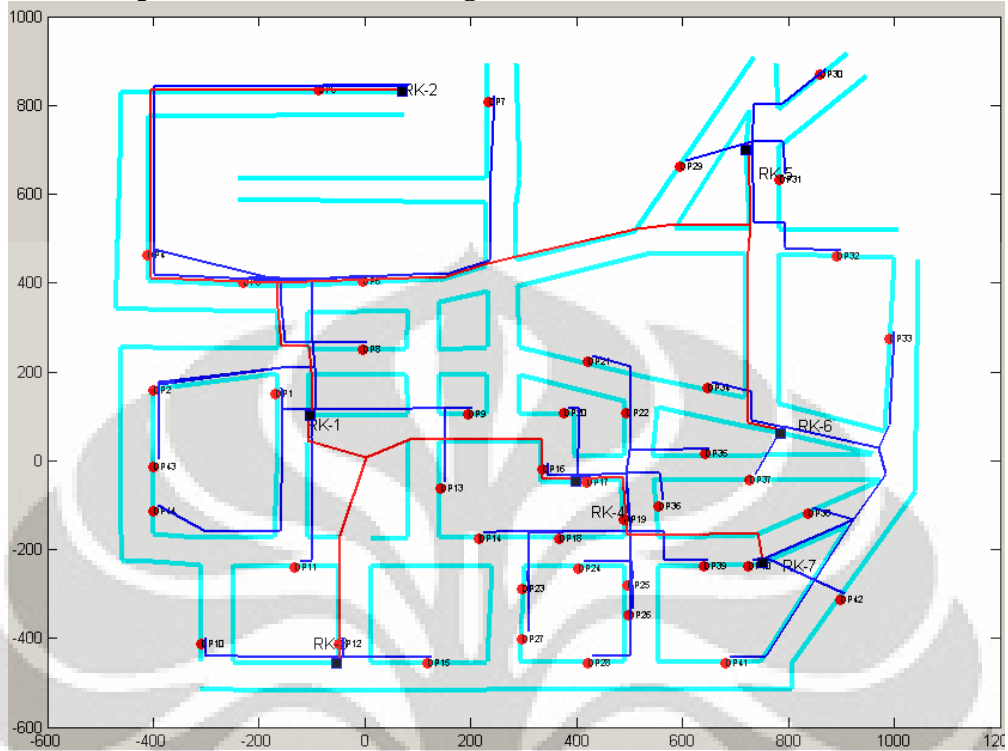
Lampiran 21 : Tabel Link Keluar Sentral Untuk RAV

A	B
	74

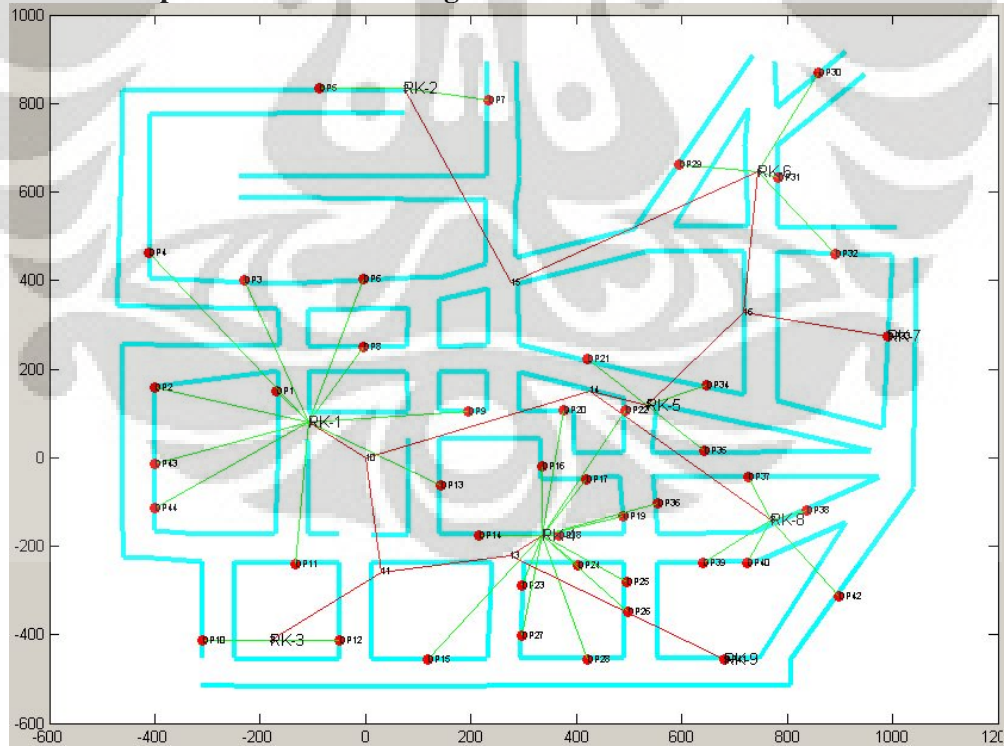
Lampiran 22 : Gambar Logik RX Pada Parameter 10000 – 500



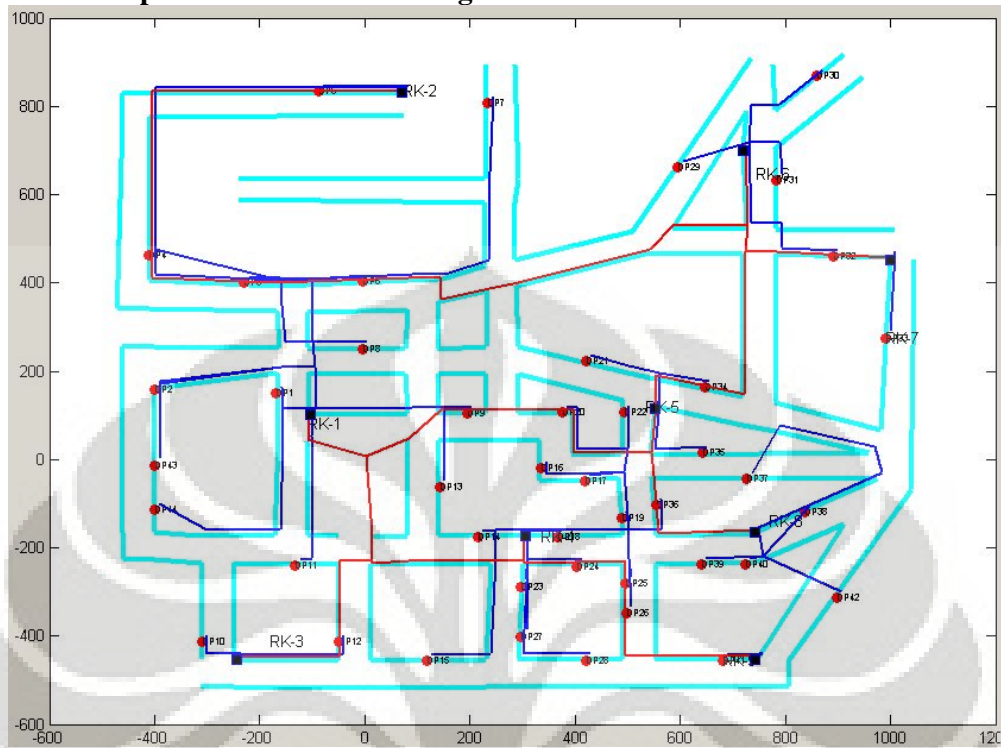
Lampiran 23 : Gambar Jaringan RX Pada Parameter 10000 – 500



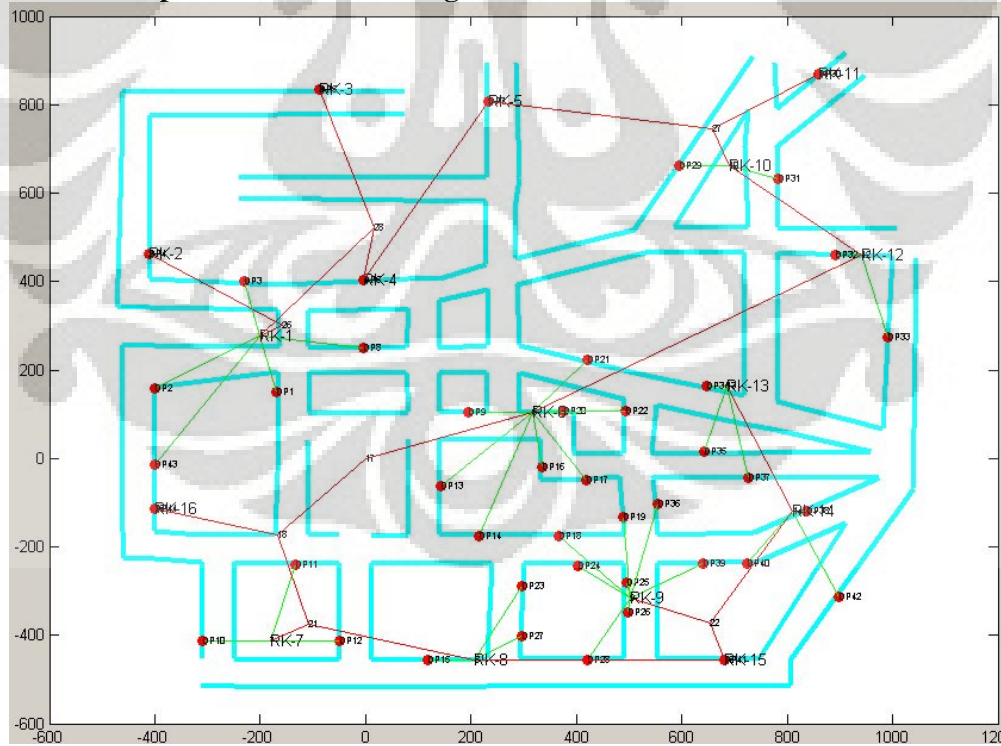
Lampiran 24 : Gambar Logik RX Pada Parameter 10000 - 400



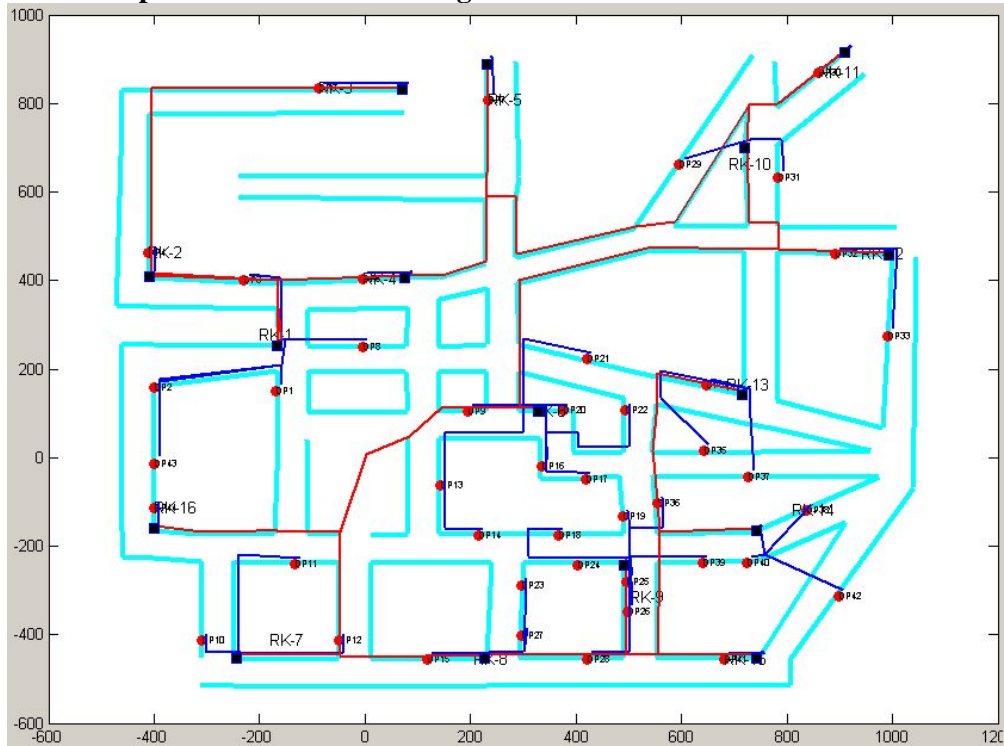
Lampiran 25 : Gambar Jaringan RX Pada Parameter 10000 - 400



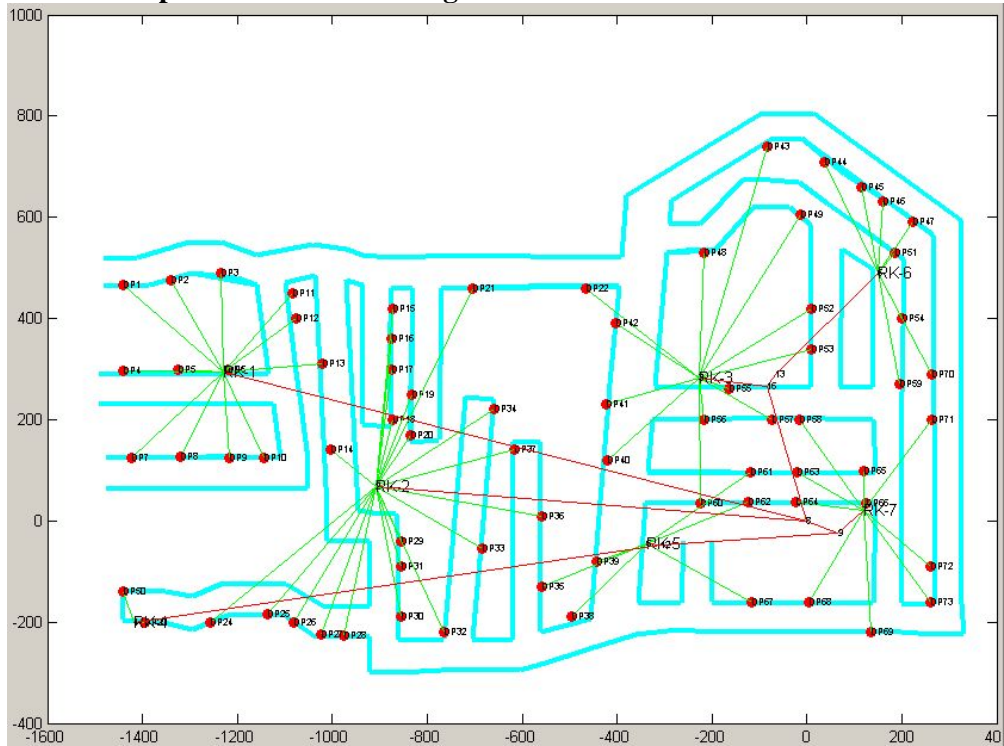
Lampiran 26 : Gambar Logik RX Pada Parameter 10000 - 300



Lampiran 27 : Gambar Jaringan RX Pada Parameter 10000 - 300

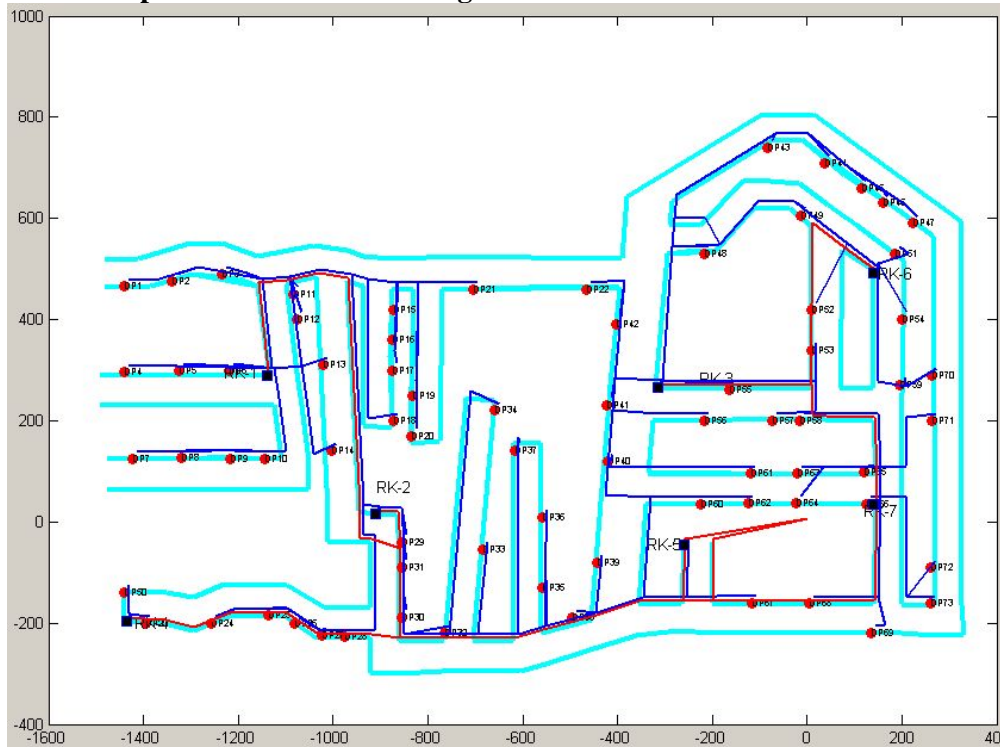


Lampiran 28 : Gambar Logik RA Pada Parameter 10000 - 500

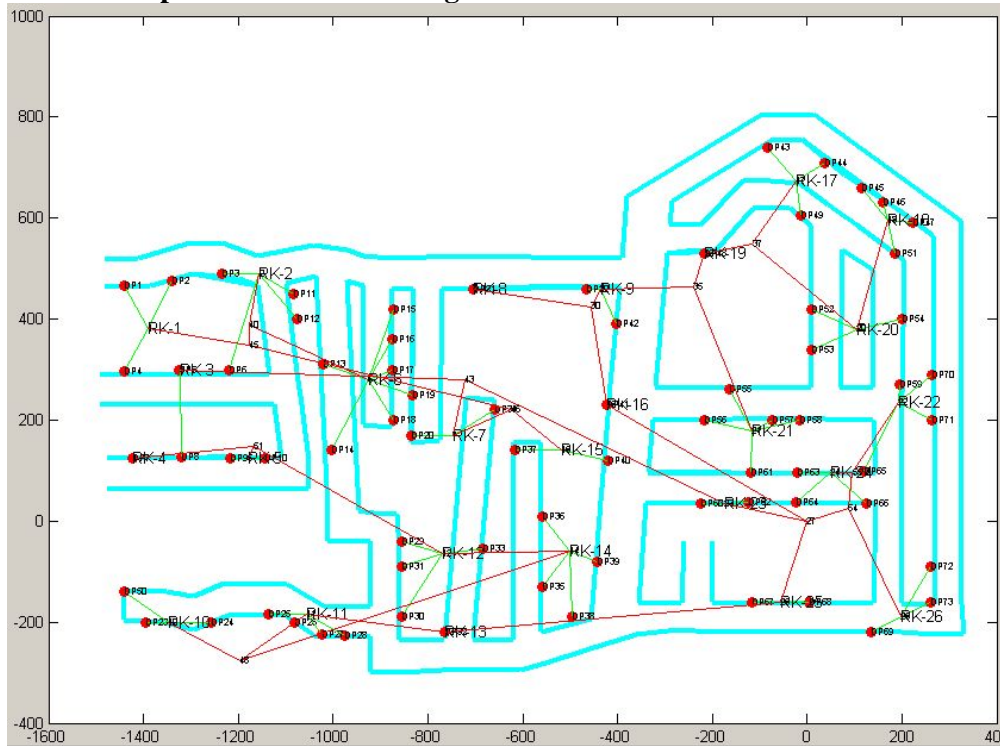




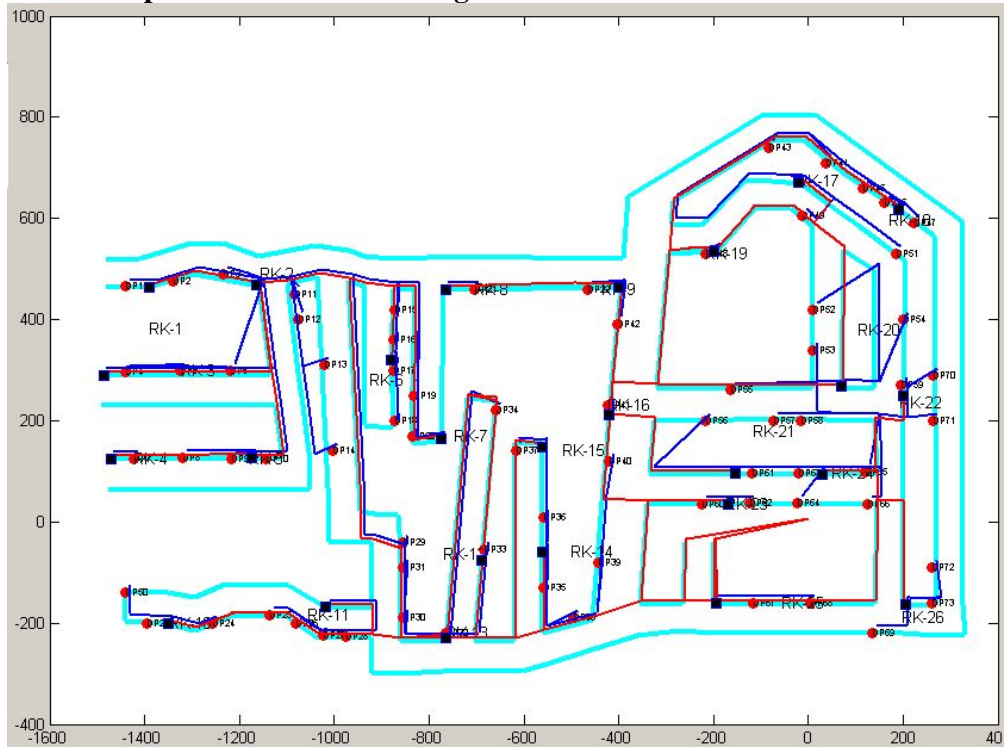
Lampiran 29 : Gambar Jaringan RA Pada Parameter 10000 – 500



Lampiran 30 : Gambar Logik RA Pada Parameter 10000 – 200



Lampiran 31: Gambar Jaringan RA Pada Parameter 10000 – 200



Lampiran 32 : Gambar Logik RA Pada Parameter 10000 – 100



**Lampiran 33 : Gambar Jaringan RA Pada Parameter 10000 – 100**



**Lampiran 34 : Tabel Report Jaringan RA Pada Parameter 10000 – 200**

A	B	Connection-ID	Distance	Path
155	228	DP-RK	378.9611362	155,-7,-,228
156	228	DP-RK	377.0367523	156,-7,-,228
157	229	DP-RK	397.9373836	157,-,140,-,9,-,229
158	228	DP-RK	968.0007916	158,-,10,-,9,-,8,-,7,-,228
159	230	DP-RK	441.6083309	159,-,11,-,230
160	229	DP-RK	453.7879564	160,-,9,-,229
161	231	DP-RK	372.6656532	161,-,14,-,231
162	230	DP-RK	867.1041378	162,-,141,-,15,-,13,-,10,-,230
163	232	DP-RK	133.4084617	163,-,141,-,232
164	232	DP-RK	379.376206	164,-,15,-,232
165	229	DP-RK	428.4449961	165,-,20,-,9,-,229
166	229	DP-RK	444.8825805	166,-,20,-,9,-,229
167	233	DP-RK	821.1635066	167,-,23,-,20,-,22,-,26,-,54,-,56,-,143,-,233
168	233	DP-RK	999.3204918	168,-,19,-,24,-,23,-,20,-,22,-,26,-,54,-,56,-,143,-,233
169	233	DP-RK	369.9171089	169,-,56,-,143,-,233
170	233	DP-RK	134.0015876	170,-,143,-,233
171	233	DP-RK	128.8520514	171,-,143,-,233
172	233	DP-RK	337.7685249	172,-,57,-,233

173	233	DP-RK	539.6218852	173,-,144,-,61,-,56,-,143,-,233
174	234	DP-RK	293.7490658	174,-,63,-,234
175	235	DP-RK	303.9501388	175,-,62,-,235
176	236	DP-RK	301.0738218	176,-,66,-,236
177	237	DP-RK	316.6344335	177,-,52,-,237
178	237	DP-RK	334.1137585	178,-,48,-,237
179	238	DP-RK	689.7074311	179,-,44,-,42,-,37,-,35,-,41,-,238
180	238	DP-RK	615.2358284	180,-,42,-,37,-,35,-,41,-,238
181	238	DP-RK	606.972032	181,-,42,-,37,-,35,-,41,-,238
182	238	DP-RK	507.7889646	182,-,37,-,35,-,41,-,238
183	239	DP-RK	718.9381416	183,-,34,-,36,-,38,-,69,-,70,-,239
184	239	DP-RK	545.6590604	184,-,38,-,69,-,70,-,239
185	239	DP-RK	633.200763	185,-,36,-,38,-,69,-,70,-,239
186	240	DP-RK	289.0508711	186,-,69,-,240
187	239	DP-RK	143.4085704	187,-,145,-,239
188	234	DP-RK	2139.042426	188,-,67,-,69,-,38,-,36,-,34,-,33,-,30,-,28,-,26,-,54,-,56,-,61,-,63,-,234
189	241	DP-RK	180.9988179	189,-,147,-,241
190	241	DP-RK	171.2709235	190,-,147,-,241
191	242	DP-RK	294.4475942	191,-,75,-,242
192	241	DP-RK	364.8170851	192,-,77,-,241
193	241	DP-RK	485.309465	193,-,107,-,77,-,241
194	242	DP-RK	1007.815445	194,-,105,-,107,-,77,-,76,-,242
195	243	DP-RK	239.205114	195,-,101,-,243
196	236	DP-RK	319.9452073	196,-,66,-,236
197	244	DP-RK	821.6315359	197,-,82,-,80,-,85,-,87,-,89,-,91,-,244
198	244	DP-RK	901.749921	198,-,84,-,82,-,80,-,85,-,87,-,89,-,91,-,244
199	245	DP-RK	163.0406374	199,-,153,-,245
200	245	DP-RK	151.1425324	200,-,154,-,245
201	245	DP-RK	151.3837811	201,-,154,-,245
202	246	DP-RK	274.0831245	202,-,88,-,246
203	244	DP-RK	395.0039254	203,-,94,-,95,-,92,-,244
204	237	DP-RK	433.1077356	204,-,51,-,52,-,237
205	245	DP-RK	1275.6153	205,-,97,-,95,-,92,-,91,-,89,-,87,-,85,-,80,-,82,-,84,-,152,-,153,-,245
206	247	DP-RK	551.1521001	206,-,117,-,121,-,247
207	247	DP-RK	323.7530659	207,-,139,-,98,-,247
208	247	DP-RK	316.432987	208,-,121,-,247
209	248	DP-RK	814.4428652	209,-,139,-,119,-,120,-,122,-,125,-,151,-,248

210	248	DP-RK	470.1419971	210,-,104,-,149,-,248
211	248	DP-RK	673.8045431	211,-,120,-,122,-,125,-,151,-,248
212	248	DP-RK	629.3982999	212,-,120,-,122,-,125,-,151,-,248
213	249	DP-RK	217.6191251	213,-,123,-,249
214	250	DP-RK	180.7612705	214,-,150,-,250
215	248	DP-RK	176.3344384	215,-,149,-,248
216	250	DP-RK	183.613452	216,-,150,-,250
217	251	DP-RK	180.0573471	217,-,151,-,251
218	251	DP-RK	200.279804	218,-,151,-,251
219	249	DP-RK	339.0880069	219,-,122,-,124,-,249
220	251	DP-RK	378.3302924	220,-,127,-,125,-,151,-,251
221	252	DP-RK	275.9289828	221,-,113,-,252
222	252	DP-RK	370.4006241	222,-,113,-,252
223	253	DP-RK	290.4099931	223,-,132,-,130,-,253
224	249	DP-RK	250.9909653	224,-,123,-,249
225	249	DP-RK	256.3043313	225,-,124,-,249
226	253	DP-RK	253.7342471	226,-,135,-,253
227	253	DP-RK	227.4884293	227,-,135,-,253
				245,-,153,-,152,-,84,-,82,-,80,-,85,-,86,-,99,-,139,-,98,-,247
245	247	RK-RK	1547.339739	
247	259	RK-Steiner	745.4499916	247,-,98,-,96,-,94,-,93,-,259
259	244	Steiner-RK	524.7134426	259,-,93,-,94,-,95,-,92,-,244
259	246	Steiner-RK	444.6657086	259,-,90,-,88,-,246
246	262	RK-Steiner	357.0935847	246,-,88,-,148,-,262
236	262	RK-Steiner	910.374498	236,-,66,-,100,-,99,-,86,-,262
				262,-,86,-,99,-,100,-,101,-,102,-,104,-,149,-,248
262	248	Steiner-RK	1086.057909	
236	257	RK-Steiner	383.4366262	236,-,66,-,257
243	257	RK-Steiner	566.1016989	243,-,100,-,66,-,257
257	235	Steiner-RK	571.8424751	257,-,146,-,62,-,235
				243,-,101,-,103,-,105,-,106,-,108,-,111,-,115,-,254
243	254	RK-Sentral	1116.261247	
229	267	RK-Steiner	478.1317823	229,-,9,-,267
				267,-,9,-,20,-,22,-,26,-,54,-,56,-,143,-,233
267	233	Steiner-RK	807.5909655	
272	228	Steiner-RK	765.8494102	272,-,9,-,8,-,7,-,228
		Steiner-Steiner		
267	272		572.7503806	267,-,9,-,272
				234,-,63,-,61,-,56,-,54,-,26,-,28,-,30,-,33,-,34,-,36,-,38,-,69,-,67,-,273
234	273	RK-Steiner	2216.105947	
273	242	Steiner-RK	1332.837549	273,-,68,-,70,-,73,-,75,-,242

273	272	Steiner-Steiner	2084.881129	273,-,67,-,69,-,38,-,36,-,34,-,33,-,30,-,28,-,26,-,22,-,20,-,9,-,272
234	270	RK-Steiner	2204.507113	234,-,63,-,61,-,56,-,54,-,26,-,28,-,30,-,33,-,34,-,36,-,38,-,69,-,67,-,270
250	270	RK-Steiner	1471.908169	250,-,106,-,108,-,107,-,77,-,73,-,70,-,68,-,270
270	230	Steiner-RK	2349.772681	270,-,67,-,69,-,38,-,36,-,34,-,33,-,30,-,28,-,26,-,22,-,20,-,9,-,10,-,230
250	254	RK-Sentral	935.141213	250,-,106,-,108,-,111,-,115,-,254
232	232	RK-RK	0	232
232	231	RK-RK	399.6689985	232,-,141,-,142,-,231
232	239	RK-RK	1939.253591	232,-,15,-,13,-,10,-,9,-,20,-,22,-,26,-,28,-,30,-,33,-,34,-,36,-,38,-,69,-,70,-,239
239	241	RK-RK	609.6773999	239,-,70,-,73,-,77,-,241
241	275	RK-Steiner	1140.392304	241,-,77,-,73,-,70,-,69,-,38,-,37,-,42,-,44,-,46,-,275
238	275	RK-Steiner	895.2992852	238,-,41,-,35,-,37,-,42,-,44,-,46,-,275
275	237	Steiner-RK	440.0222529	275,-,48,-,237
238	240	RK-RK	636.1161902	238,-,41,-,35,-,37,-,38,-,240
240	252	RK-RK	889.3732297	240,-,69,-,70,-,73,-,77,-,107,-,108,-,111,-,113,-,252
252	254	RK-Sentral	427.7838771	252,-,116,-,254
254	281	Sentral-Steiner	1058.913337	254,-,116,-,113,-,129,-,127,-,281
281	253	Steiner-RK	530.2473707	281,-,127,-,128,-,130,-,253
251	282	RK-Steiner	380.0375018	251,-,151,-,125,-,282
282	249	Steiner-RK	388.8625741	282,-,122,-,124,-,249
282	281	Steiner-Steiner	371.2519171	282,-,125,-,127,-,281

**Lampiran 35 : Tabel Kebutuhan Catuan RA**

No	Kbth RK
1	1350
2	1800
3	900
4	450
5	900
6	3150
7	900
8	450
9	900

10	1350
11	1800
12	1800
13	450
14	1800
15	900
16	450
17	1350
18	1800
19	450
20	1350
21	2250
22	1800
23	900
24	1350
25	900
26	1350

**Lampiran 36 : Tabel Koordinat Semua Titik Pada RA**

Index	X	Y	Type
1	-1480	519	1
2	-1410	521	1
3	-1310	549	1
4	-1235	550	1
5	-1160	526	1
6	-1480	464	1
7	-1385	465	1
8	-1295	489	1
9	-1160	466	1
10	-1135	290	1
11	-1490	290	1
12	-1490	230	1
13	-1125	230	1
14	-1475	124	1
15	-1110	124	1
16	-1475	64	1
17	-1110	66	1
18	-1050	64	1
19	-1050	120	1
20	-1100	470	1
21	-1040	546	1
22	-1035	484	1

23	-1075	294	1
24	-1065	230	1
25	-970	536	1
26	-970	476	1
27	-1010	20	1
28	-945	20	1
29	-1010	-40	1
30	-945	-40	1
31	-915	16	1
32	-865	14	1
33	-920	-40	1
34	-860	-60	1
35	-920	-170	1
36	-860	-160	1
37	-920	-230	1
38	-860	-236	1
39	-920	-300	1
40	-860	-300	1
41	-1015	-170	1
42	-1035	-230	1
43	-1095	-126	1
44	-1105	-184	1
45	-1230	-124	1
46	-1220	-186	1
47	-1295	-150	1
48	-1295	-216	1
49	-1350	-140	1
50	-1440	-140	1
51	-1440	-196	1
52	-1355	-200	1
53	-935	526	1
54	-935	466	1
55	-875	520	1
56	-875	460	1
57	-875	186	1
58	-935	190	1
59	-830	520	1
60	-775	524	1
61	-830	460	1
62	-770	460	1
63	-835	154	1
64	-775	160	1
65	-390	520	1



66	-395	464	1
67	-720	244	1
68	-660	239	1
69	-765	-235	1
70	-705	-234	1
71	-770	-294	1
72	-705	-294	1
73	-620	-236	1
74	-605	-294	1
75	-620	154	1
76	-560	154	1
77	-560	-220	1
78	-550	-280	1
79	-380	644	1
80	-285	634	1
81	-100	804	1
82	-75	754	1
83	15	806	1
84	-10	756	1
85	-290	586	1
86	-295	530	1
87	-225	586	1
88	-195	534	1
89	-135	676	1
90	-115	620	1
91	-75	674	1
92	-15	670	1
93	-35	620	1
94	10	584	1
95	45	626	1
96	75	536	1
97	105	586	1
98	70	264	1
99	-320	266	1
100	-415	270	1
101	-420	206	1
102	-325	200	1
103	-430	94	1
104	-335	96	1
105	-435	40	1
106	-340	36	1
107	-455	-194	1
108	-355	-164	1

109	-465	-254	1
110	-360	-224	1
111	-265	-164	1
112	-270	-216	1
113	-200	-160	1
114	-200	-220	1
115	-260	-40	1
116	-200	-40	1
117	140	496	1
118	205	516	1
119	10	204	1
120	70	200	1
121	140	264	1
122	140	200	1
123	200	254	1
124	200	194	1
125	145	94	1
126	200	96	1
127	145	34	1
128	200	34	1
129	140	-164	1
130	200	-164	1
131	155	-220	1
132	200	-220	1
133	265	564	1
134	325	594	1
135	270	-166	1
136	270	-224	1
137	325	-164	1
138	330	-224	1
139	10	264	1
140	-1200	480	1
141	-1180	126	1
142	-1265	126	1
143	-880	326	1
144	-835	266	1
145	-690	-80	1
146	-495	466	1
147	-560	-54	1
148	-255	534	1
149	-160	95	1
150	-175	35	1
151	25	94	1

152	80	686	1
153	145	646	1
154	195	614	1
155	-1440	466	2
156	-1340	475	2
157	-1235	490	2
158	-1440	296	2
159	-1324	298	2
160	-1220	298	2
161	-1424	124	2
162	-1320	126	2
163	-1216	124	2
164	-1144	124	2
165	-1084	450	2
166	-1076	400	2
167	-1020	310	2
168	-1004	140	2
169	-872	420	2
170	-874	360	2
171	-874	300	2
172	-872	200	2
173	-832	250	2
174	-834	170	2
175	-704	460	2
176	-466	460	2
177	-1396	-200	2
178	-1256	-200	2
179	-1136	-184	2
180	-1080	-202	2
181	-1024	-224	2
182	-976	-226	2
183	-856	-40	2
184	-855	-190	2
185	-856	-90	2
186	-764	-220	2
187	-684	-55	2
188	-659	220	2
189	-559	-130	2
190	-559	10	2
191	-616	140	2
192	-496	-190	2
193	-444	-80	2
194	-420	120	2

195	-424	230	2
196	-404	390	2
197	-84	740	2
198	36	710	2
199	116	660	2
200	160	630	2
201	224	590	2
202	-216	530	2
203	-14	606	2
204	-1440	-140	2
205	184	530	2
206	10	420	2
207	10	340	2
208	200	400	2
209	-164	260	2
210	-216	200	2
211	-74	200	2
212	-16	200	2
213	196	270	2
214	-224	34	2
215	-120	96	2
216	-124	38	2
217	-20	96	2
218	-24	38	2
219	120	98	2
220	124	36	2
221	-116	-160	2
222	4	-160	2
223	136	-220	2
224	264	290	2
225	264	200	2
226	260	-90	2
227	260	-160	2
228	-	-	-
228	1389.999723	464.9473713	3
229	-	-	-
229	1164.719292	467.6517521	3
230	-1485	290	3
231	-1470	124	3
232	-1175.00204	125.8572011	3
233	-	-	-
233	879.8215423	321.0031857	3
234	-	-	-
234	774.9166782	164.9993057	3

	-		
235	765.0011897	460.109065	3
	-		
236	399.9997156	463.9466697	3
	-	-	
237	1350.168825	201.2883133	3
	-	-	
238	-1019.38108	167.5904063	3
	-	-	
239	689.5318462	75.02196504	3
	-	-	
240	764.5323306	230.0219195	3
241	-560	-59	3
242	-560	149	3
	-		
243	419.6105617	210.9848107	3
	-		
244	19.98892579	670.3325951	3
245	190.788643	616.6952685	3
	-		
246	199.9960048	533.8001598	3
247	70.09189624	268.9991554	3
248	-155.000073	94.97297337	3
249	200	249	3
	-		
250	170.0000244	34.98437508	3
251	30	94	3
	-		
252	-195.000346	160.0588195	3
	-		
253	204.9979604	164.1427989	3
254	0	0	5
	-		
255	499.9988103	465.890935	4
256	10	579	4
	-		
257	490.0009997	465.90002	4
258	-110	620	4
	-		
259	118.4055197	616.3390663	4
	-		
260	199.9960048	533.8001598	4
	-		
261	118.4055197	616.3390663	4
262	-250	534	4
	-		
263	118.4055197	616.3390663	4

264	-	118.4055197	616.3390663	4
265	-	1075.703169	298.9503084	4
266	-	765.0011897	460.109065	4
267	-	1164.928977	466.8397517	4
268	-	664.9827288	239.4152274	4
269	-	-1140	290	4
270	-	715.0172712	243.5847726	4
271	-	-1140	290	4
272	-	1135.703169	294.9503084	4
273	-	660.4681538	234.021965	4
274	-	1269.999773	125.9523831	4
275	-	1215.000756	185.9130566	4
276	-	-1185	126	4
277	-	-1185	126	4
278	-	-1175.00204	125.8572011	4
279	-	144.8737776	29.00159346	4
280	-	144.7644129	98.99444679	4
281	-	140.0000244	34.01562492	4
282	-	140	94	4

**Lampiran 37 : Tabel Link Jalan RA Parameter 1000 – 200**

Index	Ruas A	Ruas B	Index	Ruas A	Ruas B	Index	Ruas A	Ruas B
1	1	2	201	174	63	401	275	46
2	6	7	202	174	61	402	275	48
3	2	3	203	175	62	403	276	141
4	3	4	204	175	64	404	276	15
5	4	5	205	176	146	405	277	141
6	7	8	206	176	66	406	277	15
7	8	9	207	177	52	407	278	141
8	10	11	208	177	51	408	278	15
9	13	12	209	178	46	409	279	127
10	14	15	210	178	48	410	279	106
11	16	17	211	179	44	411	280	125
12	17	18	212	179	46	412	280	122
13	18	19	213	180	44	413	281	127

14	19	24	214	180	42	414	281	106
15	24	23	215	181	42	415	282	125
16	23	20	216	181	44	416	282	122
17	21	5	217	182	37	417	5	9
18	20	22	218	182	42	418	9	20
19	21	25	219	183	34	419	10	23
20	22	27	220	183	32	420	10	13
21	29	30	221	184	36	421	13	24
22	27	29	222	184	38	422	15	19
23	26	28	223	185	34	423	15	17
24	28	31	224	185	36	424	21	22
25	31	32	225	186	69	425	22	26
26	30	33	226	186	38	426	25	26
27	25	53	227	187	145	427	27	28
28	26	54	228	187	70	428	28	30
29	53	55	229	188	68	429	33	34
30	54	58	230	188	67	430	35	36
31	58	57	231	189	147	431	35	37
32	57	56	232	189	77	432	37	38
33	55	59	233	190	147	433	38	40
34	59	60	234	190	76	434	53	54
35	56	61	235	191	75	435	54	56
36	61	63	236	191	76	436	55	56
37	63	64	237	192	107	437	59	60
38	64	62	238	192	77	438	59	61
39	32	34	239	193	107	439	61	62
40	33	35	240	193	105	440	60	62
41	34	36	241	194	103	441	69	70
42	50	51	242	194	105	442	69	71
43	50	49	243	195	101	443	69	70
44	49	47	244	195	100	444	70	72
45	47	45	245	196	66	445	73	77
46	45	43	246	196	62	446	73	74
47	43	41	247	197	82	447	73	77
48	41	35	248	197	84	448	77	78
49	51	52	249	198	152	449	107	108
50	52	48	250	198	84	450	109	110
51	48	46	251	199	153	451	103	104
52	46	44	252	199	152	452	105	106
53	44	42	253	200	153	453	100	99
54	42	37	254	200	154	454	101	102
55	37	39	255	201	154	455	65	86
56	36	38	256	201	133	456	86	85

57	39	40	257	202	88	457	79	80
58	38	69	258	202	86	458	87	88
59	40	71	259	203	93	459	111	113
60	69	67	260	203	94	460	111	112
61	67	68	261	204	50	461	114	113
62	68	70	262	204	51	462	81	82
63	71	72	263	205	118	463	83	84
64	70	73	264	205	97	464	133	134
65	72	74	265	206	96	465	94	96
66	75	76	266	206	117	466	94	95
67	73	75	267	207	139	467	96	97
68	76	77	268	207	94	468	117	118
69	74	78	269	208	117	469	139	98
70	77	107	270	208	121	470	119	139
71	78	109	271	209	99	471	98	120
72	60	65	272	209	139	472	121	123
73	62	66	273	210	102	473	122	124
74	79	65	274	210	104	474	125	126
75	66	100	275	211	119	475	125	127
76	100	101	276	211	120	476	125	126
77	101	103	277	212	119	477	127	128
78	103	105	278	212	120	478	129	130
79	105	107	279	213	123	479	129	131
80	109	110	280	213	124	480	130	132
81	108	106	281	214	150	481	135	136
82	104	102	282	214	106	482	135	137
83	99	86	283	215	149	483	254	115
84	80	85	284	215	151	484	254	116
85	79	81	285	216	150			
86	81	83	286	216	106			
87	80	82	287	217	151			
88	82	84	288	217	149			
89	83	134	289	218	151			
90	84	133	290	218	149			
91	133	135	291	219	125			
92	134	137	292	219	122			
93	137	138	293	220	127			
94	136	138	294	220	106			
95	135	130	295	221	113			
96	136	132	296	221	116			
97	132	131	297	222	129			
98	131	114	298	222	113			
99	113	129	299	223	131			

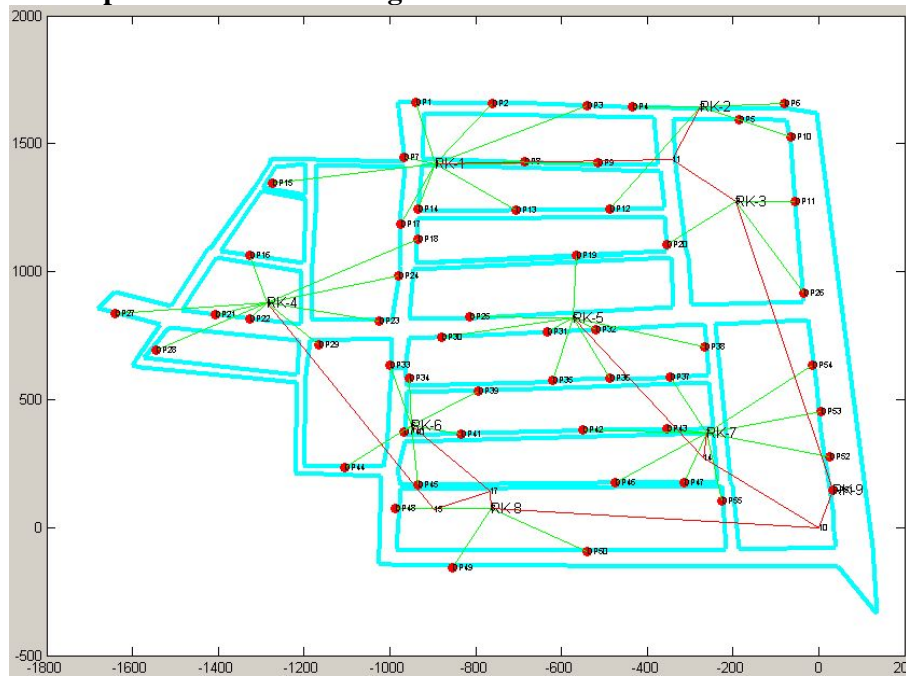


100	127	129	300	223	132
101	128	130	301	224	123
102	126	124	302	224	124
103	123	118	303	225	124
104	117	121	304	225	123
105	122	125	305	226	135
106	85	87	306	226	130
107	89	91	307	227	135
108	97	118	308	227	130
109	91	92	309	228	7
110	95	97	310	228	6
111	86	88	311	229	9
112	88	90	312	229	8
113	90	93	313	230	11
114	93	94	314	230	10
115	96	117	315	231	14
116	139	94	316	231	142
117	96	98	317	232	141
118	99	139	318	232	15
119	102	119	319	233	143
120	98	121	320	233	57
121	119	120	321	234	64
122	120	122	322	234	63
123	104	125	323	235	62
124	106	127	324	235	64
125	9	10	325	236	66
126	13	15	326	236	62
127	87	89	327	237	52
128	123	124	328	237	48
129	126	128	329	238	41
130	108	111	330	238	43
131	111	115	331	239	145
132	110	112	332	239	70
133	112	114	333	240	69
134	113	116	334	240	38
135	92	95	335	241	147
136	140	9	336	241	77
137	140	8	337	242	76
138	141	142	338	242	75
139	141	15	339	243	101
140	142	14	340	243	100
141	143	56	341	244	92
142	143	57	342	244	91

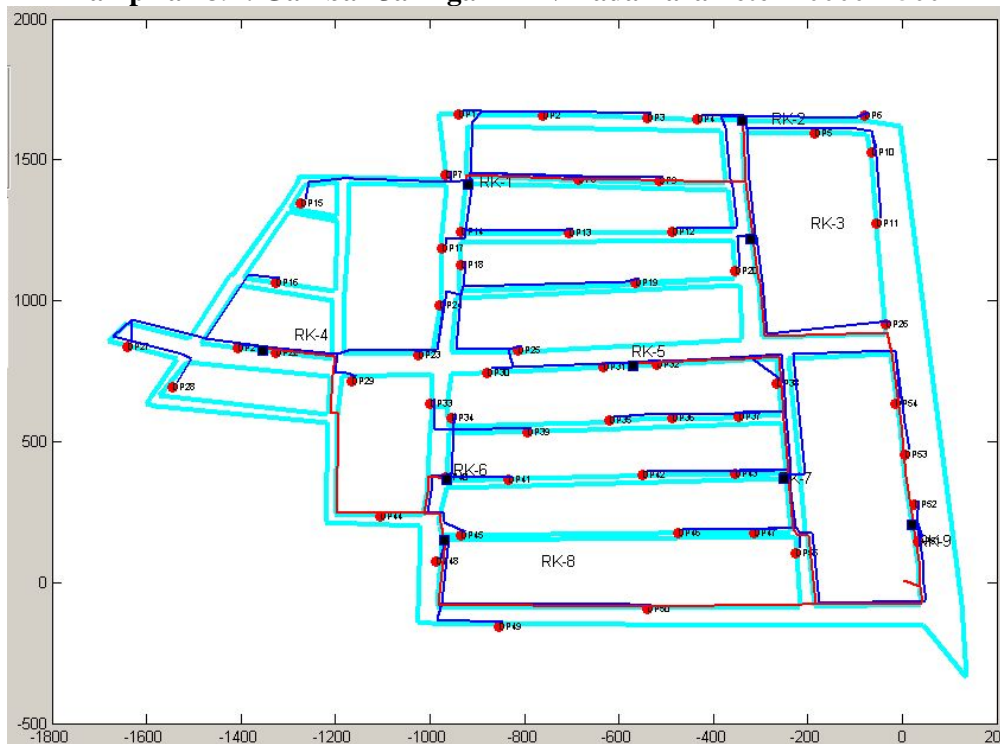
143	144	63	343	245	154
144	144	61	344	245	153
145	145	70	345	246	88
146	145	68	346	246	86
147	146	66	347	247	98
148	146	62	348	247	121
149	147	76	349	248	149
150	147	77	350	248	151
151	148	86	351	249	123
152	148	88	352	249	124
153	149	104	353	250	150
154	149	151	354	250	106
155	150	106	355	251	151
156	150	127	356	251	149
157	151	149	357	252	113
158	151	125	358	252	116
159	152	84	359	253	130
160	152	153	360	253	135
161	153	154	361	255	146
162	154	133	362	255	66
163	155	6	363	256	94
164	155	7	364	256	93
165	156	7	365	257	146
166	156	8	366	257	66
167	157	140	367	258	90
168	157	8	368	258	93
169	158	11	369	259	90
170	158	10	370	259	93
171	159	11	371	260	88
172	159	10	372	260	86
173	160	10	373	261	90
174	160	9	374	261	93
175	161	14	375	262	148
176	161	142	376	262	86
177	162	142	377	263	90
178	162	141	378	263	93
179	163	141	379	264	90
180	163	142	380	264	93
181	164	15	381	265	23
182	164	14	382	265	24
183	165	20	383	266	62
184	165	22	384	266	64
185	166	20	385	267	9

186	166	22	386	267	8
187	167	23	387	268	68
188	167	24	388	268	67
189	168	19	389	269	10
190	168	18	390	269	9
191	169	56	391	270	67
192	169	57	392	270	68
193	170	143	393	271	10
194	170	56	394	271	9
195	171	143	395	272	10
196	171	57	396	272	9
197	172	57	397	273	68
198	172	58	398	273	67
199	173	144	399	274	142
200	173	63	400	274	141

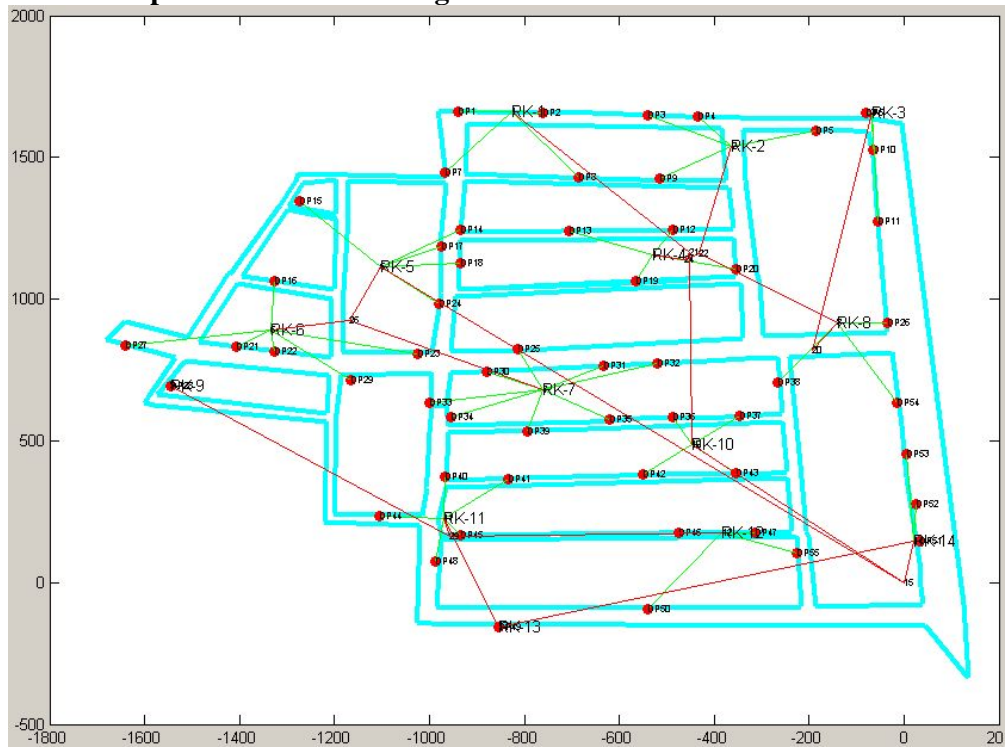
Lampiran 38 : Gambar Logik RAV Pada Parameter 10000 – 500



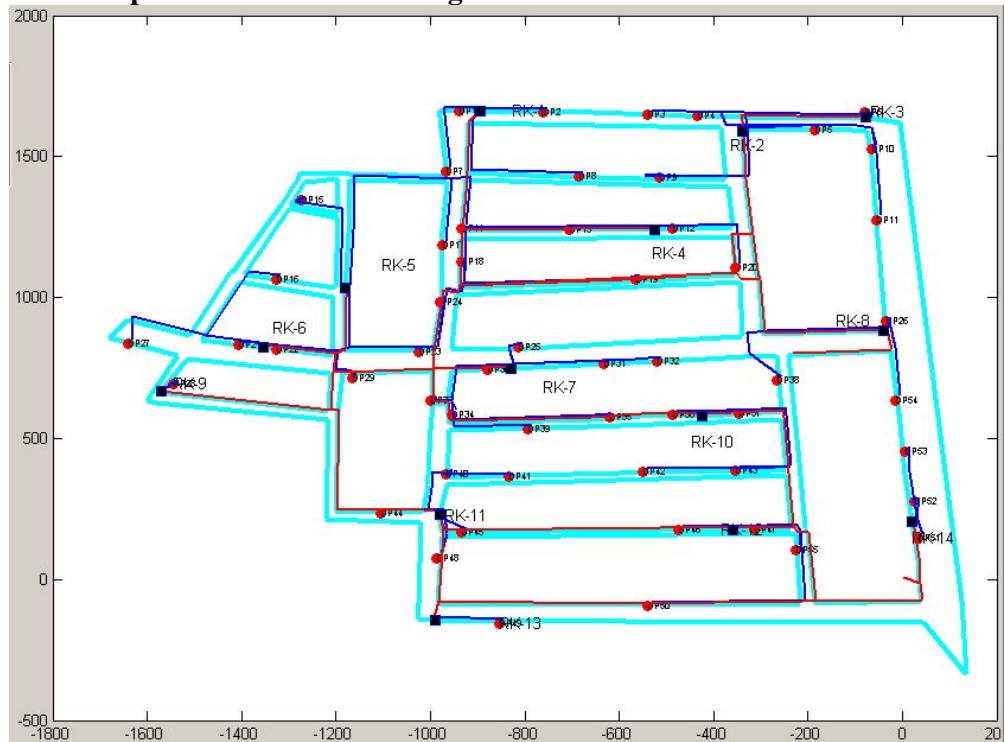
Lampiran 39 : Gambar Jaringan RAV Pada Parameter 10000 – 500



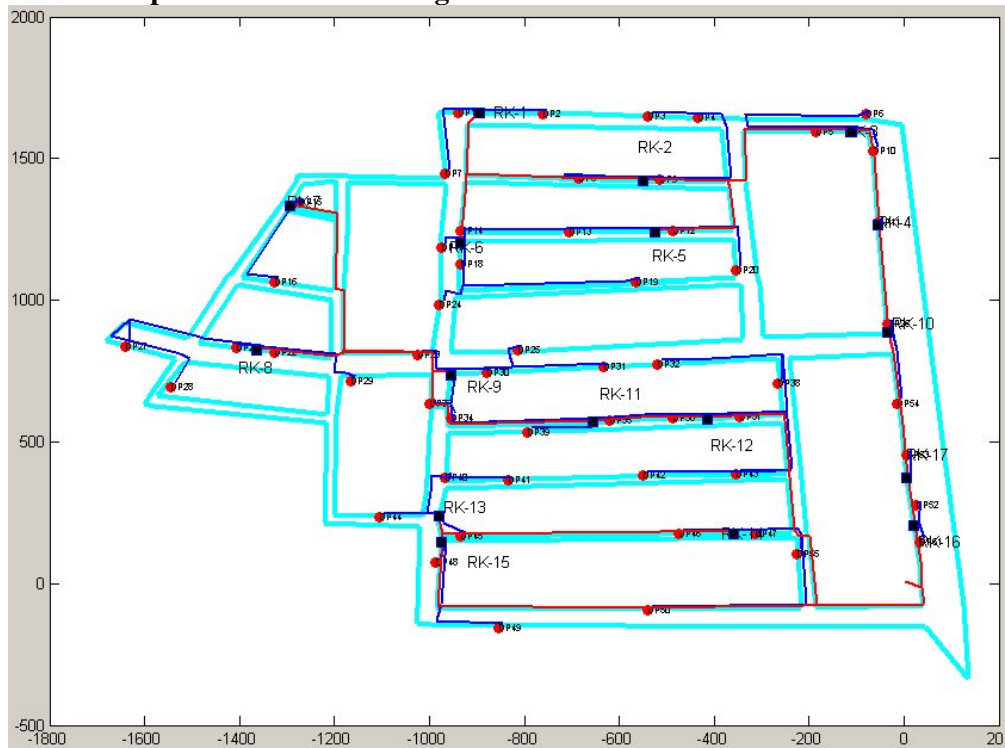
Lampiran 40 : Gambar Logik RAV Pada Parameter 10000 – 400



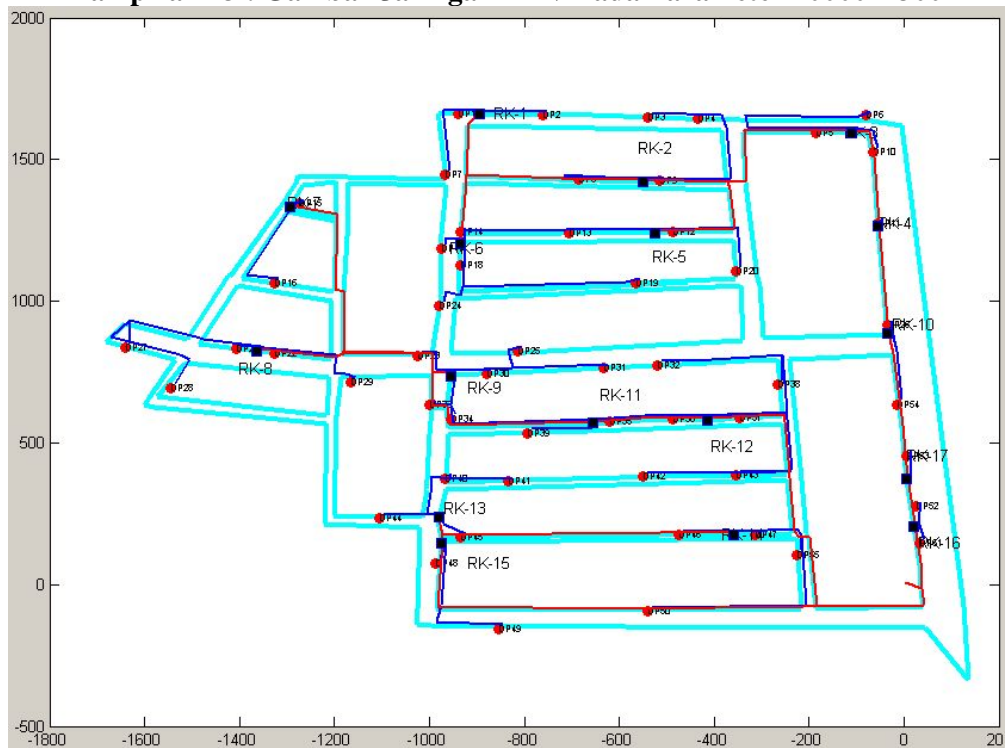
Lampiran 41 : Gambar Jaringan RAV Pada Parameter 10000 – 400



Lampiran 42 : Gambar Logik RAV Pada Parameter 10000 – 300



Lampiran 43 : Gambar Jaringan RAV Pada Parameter 10000 – 300



**Lampiran 44 : Tabel Report RAV Pada Parameter 10000 – 300**

A	B	Connection-ID	Distance	Path
151	206	DP-RK	356.3848205	151,-3,-,206
152	206	DP-RK	407.587919	152,-3,-,206
153	207	DP-RK	804.1487461	153,-7,-5,-,18,-,17,-,207
154	207	DP-RK	750.3216835	154,-7,-5,-,18,-,17,-,207
155	208	DP-RK	226.8836634	155,-87,-,208
156	208	DP-RK	846.2461808	156,-8,-86,-,85,-,11,-,87,-,208
157	206	DP-RK	493.5593577	157,-2,-,1,-,206
158	207	DP-RK	406.4478627	158,-16,-,207
159	207	DP-RK	337.9343405	159,-17,-,207
160	208	DP-RK	366.4075233	160,-9,-,208
161	209	DP-RK	207.3575017	161,-82,-,209
162	210	DP-RK	201.7547554	162,-89,-,210
163	210	DP-RK	321.3046899	163,-89,-,210
164	211	DP-RK	219.9739442	164,-91,-94,-,211
165	212	DP-RK	335.4963816	165,-22,-,212
166	212	DP-RK	611.9051433	166,-29,-,24,-,22,-,212
167	211	DP-RK	235.894991	167,-93,-94,-,211
168	211	DP-RK	226.846243	168,-94,-,211
169	210	DP-RK	1155.912216	169,-102,-,103,-,94,-,91,-,90,-,89,-,210
170	210	DP-RK	373.3594134	170,-95,-,88,-,210
171	213	DP-RK	315.9444718	171,-38,-,213
172	213	DP-RK	313.6252498	172,-38,-,213
173	214	DP-RK	375.9134984	173,-47,-48,-,109,-,214
174	211	DP-RK	322.8829545	174,-148,-,104,-,103,-,211
175	214	DP-RK	367.6347585	175,-107,-,110,-,109,-,214
176	215	DP-RK	238.9024902	176,-79,-,215
177	213	DP-RK	549.8917331	177,-41,-,32,-,33,-,213
178	213	DP-RK	808.7541538	178,-43,-42,-,40,-,41,-,32,-,33,-,213
179	213	DP-RK	559.3723324	179,-46,-44,-,39,-,38,-,213
180	214	DP-RK	208.4966036	180,-109,-,214
181	216	DP-RK	920.2802947	181,-110,-,109,-,115,-,116,-,117,-,216
182	216	DP-RK	928.5743497	182,-112,-,113,-,81,-,121,-,120,-,119,-,216
183	214	DP-RK	252.6788566	183,-114,-,115,-,214
184	214	DP-RK	230.6784424	184,-115,-,214
185	216	DP-RK	178.2728375	185,-117,-,216
186	216	DP-RK	236.0871635	186,-119,-,216
187	217	DP-RK	197.3840941	187,-120,-,217
188	217	DP-RK	606.7759323	188,-81,-,121,-,120,-,217

189	216	DP-RK	274.9805301	189,-,118,-,117,-,216
190	218	DP-RK	452.4146409	190,-,126,-,125,-,127,-,58,-,60,-,218
191	218	DP-RK	533.183663	191,-,126,-,125,-,127,-,58,-,60,-,218
192	217	DP-RK	781.6062602	192,-,131,-,132,-,122,-,121,-,120,-,217
193	217	DP-RK	594.2951854	193,-,132,-,122,-,121,-,120,-,217
194	218	DP-RK	357.065032	194,-,58,-,60,-,218
195	218	DP-RK	307.9535746	195,-,61,-,218
196	219	DP-RK	217.8997766	196,-,135,-,219
197	219	DP-RK	158.5095122	197,-,135,-,219
198	220	DP-RK	311.4393412	198,-,63,-,220
199	220	DP-RK	530.9545766	199,-,68,-,64,-,220
200	219	DP-RK	778.6101388	200,-,70,-,139,-,136,-,219
201	221	DP-RK	288.1168859	201,-,75,-,221
202	221	DP-RK	289.3542987	202,-,75,-,221
203	222	DP-RK	299.4844175	203,-,76,-,222
204	215	DP-RK	423.9185047	204,-,78,-,79,-,215
205	219	DP-RK	259.3088129	205,-,139,-,136,-,219
223	222	Sentral-RK	605.6115779	223,-,74,-,73,-,75,-,222
222	215	RK-RK	795.9633175	222,-,76,-,77,-,78,-,79,-,215
215	209	RK-RK	643.7584752	215,-,79,-,82,-,209
209	208	RK-RK	584.4057221	209,-,83,-,9,-,208
208	207	RK-RK	1038.195909	208,-,87,-,11,-,85,-,96,-,18,-,17,-,207
207	224	RK-Steiner	467.2645103	207,-,16,-,224
224	210	Steiner-RK	946.4762907	224,-,16,-,17,-,18,-,19,-,88,-,210
211	224	RK-Steiner	662.4507299	211,-,94,-,91,-,15,-,13,-,224
224	206	Steiner-RK	721.659004	224,-,13,-,4,-,3,-,206
224	224	Steiner-Steiner	0	224
223	217	Sentral-RK	1420.746362	223,-,74,-,72,-,141,-,140,-,139,-,136,-,133,-,132,-,122,-,121,-,120,-,217
217	216	RK-RK	310.7112451	217,-,119,-,216
216	227	RK-Steiner	965.0449992	216,-,117,-,116,-,115,-,114,-,48,-,47,-,150,-,227
227	212	Steiner-RK	982.4529314	227,-,45,-,37,-,36,-,34,-,26,-,25,-,22,-,212
213	228	RK-Steiner	543.1631375	213,-,38,-,39,-,45,-,228
228	214	Steiner-RK	515.2139145	228,-,150,-,47,-,48,-,109,-,214
228	227	Steiner-Steiner	295.0421131	228,-,150,-,227



219	218	RK-RK	903.9907964	219,-,135,-,134,-,62,-,61,-,218
218	220	RK-RK	371.2931979	218,-,61,-,62,-,63,-,220
220	229	RK-Steiner	1587.624683	220,-,64,-,69,-,70,-,141,-,72,-,74,-,229
223	229	Sentral-Steiner	322.8946998	223,-,74,-,229
229	221	Steiner-RK	376.7790903	229,-,73,-,221

**Lampiran 45 : Tabel Kebutuhan Catu RAV Pada Parameter 10000 – 300**

RK	Kbth RK
1	1350
2	1800
3	1350
4	450
5	1800
6	1800
7	900
8	2250
9	2250
10	900
11	2250
12	1800
13	1800
14	1800
15	900
16	900
17	450

**Lampiran 46 : Tabel Koordinat Titik RAV Pada Parameter 10000 – 300**

Index	X	Y	Type
1	-980	1661	1
2	-980	1635	1
3	-900	1660	1
4	-920	1621	1
5	-385	1601	1
6	-655	1655	1
7	-395	1641	1
8	-110	1635	1
9	-75	1589	1
10	-82	1641	1
11	-220	1596	1
12	-965	1431	1
13	-925	1441	1
14	-965	1409	1

15	-925	1415	1
16	-725	1429	1
17	-555	1421	1
18	-375	1415	1
19	-370	1391	1
20	-1265	1409	1
21	-1280	1415	1
22	-1295	1329	1
23	-1315	1334	1
24	-1300	1315	1
25	-1195	1301	1
26	-1195	1281	1
27	-1170	1295	1
28	-1420	1089	1
29	-1395	1081	1
30	-1425	1089	1
31	-1405	1055	1
32	-1510	866	1
33	-1485	850	1
34	-1200	1035	1
35	-1205	1001	1
36	-1180	1029	1
37	-1180	995	1
38	-1360	824	1
39	-1205	795	1
40	-1680	859	1
41	-1640	921	1
42	-1535	795	1
43	-1515	781	1
44	-1210	729	1
45	-1183	810	1
46	-1185	729.5	1
47	-995	809	1
48	-995	741	1
49	-1573	662	1
50	-1590	661	1
51	-1600	635	1
52	-1580	626	1
53	-1215	595	1
54	-1215	569	1
55	-1195	595	1
56	-1195	569	1
57	-1200	241	1

58	-1015	239	1
59	-1020	201	1
60	-980	235	1
61	-980	201	1
62	-975	169	1
63	-975	149	1
64	-985	-85	1
65	-1025	-85	1
66	-1220	215	1
67	-1025	-145	1
68	-995	-145	1
69	-585	-91	1
70	-215	-85	1
71	45	-151	1
72	40	-79	1
73	35	41	1
74	35	-20	1
75	20	209	1
76	5	369	1
77	-5	555	1
78	-25	809	1
79	-35	881	1
80	-240	795	1
81	-265	795	1
82	-55	1261	1
83	-65	1501	1
84	-295	869	1
85	-340	1595	1
86	-345	1641	1
87	-105	1595	1
88	-360	1249	1
89	-520	1241	1
90	-720	1239	1
91	-935	1235	1
92	-975	1236	1
93	-975	1211	1
94	-935	1206	1
95	-360	1215	1
96	-335	1415	1
97	-320	1215	1
98	-355	1081	1
99	-310	1085	1
100	-345	1055	1

101	-305	1055	1
102	-595	1055	1
103	-940	1035	1
104	-945	1009	1
105	-455	1069	1
106	-955	815	1
107	-845	815	1
108	-340	861	1
109	-955	741	1
110	-835	749	1
111	-575	769	1
112	-415	781	1
113	-335	789	1
114	-995	621	1
115	-965	621	1
116	-960	555	1
117	-660	569	1
118	-665	541	1
119	-555	581	1
120	-420	581	1
121	-255	595	1
122	-255	569	1
123	-960	529	1
124	-1000	529	1
125	-1005	369	1
126	-965	361	1
127	-1005	335	1
128	-965	335	1
129	-800	369	1
130	-500	381	1
131	-315	381	1
132	-245	389	1
133	-245	369	1
134	-515	175	1
135	-355	175	1
136	-235	181	1
137	-215	369	1
138	-205	181	1
139	-225	161	1
140	-200	161	1
141	-185	-85	1
142	-175	-151	1
143	135	-340	1

144	125	-79	1
145	-5	1621	1
146	-1275	1441	1
147	-1170	1419	1
148	-975	1020	1
149	-1195	1420	1
150	-1055	812	1
151	-940	1660	2
152	-760	1655	2
153	-540	1650	2
154	-434	1645	2
155	-186	1595	2
156	-80	1655	2
157	-966	1445	2
158	-686	1430	2
159	-514	1425	2
160	-66	1525	2
161	-54	1275	2
162	-486	1245	2
163	-706	1240	2
164	-934	1245	2
165	-1274	1345	2
166	-1326	1065	2
167	-974	1185	2
168	-934	1125	2
169	-566	1065	2
170	-354	1105	2
171	-1406	830	2
172	-1326	815	2
173	-1026	805	2
174	-980	985	2
175	-814	825	2
176	-34	915	2
177	-1640	835	2
178	-1546	695	2
179	-1166	715	2
180	-880	745	2
181	-634	765	2
182	-520	775	2
183	-1000	635	2
184	-954	585	2
185	-620	575	2
186	-486	585	2

187	-346	590	2
188	-266	705	2
189	-794	535	2
190	-966	375	2
191	-834	365	2
192	-549	380	2
193	-354	385	2
194	-1106	235	2
195	-934	165	2
196	-474	175	2
197	-314	175	2
198	-986	75	2
199	-854	-155	2
200	-540	-95	2
201	34	145	2
202	26	275	2
203	6	455	2
204	-14	635	2
205	-226	105	2
206	-895.0010409	1659.89798	3
207	-550.0027755	1420.833426	3
208	-109.999811	1595.043477	3
209	-55.20815272	1265.995665	3
210	-524.99975	1240.950002	3
211	-935.1461364	1201.002136	3
212	-1293.244383	1333.681646	3
213	-1364.895228	825.0182074	3
214	-955.4152274	736.0172712	3
215	-35.26279417	885.9930891	3
216	-655.0323366	569.567733	3
217	-415.0179016	581.4227235	3
218	-979.2582977	239.9446818	3
219	-360	175	3
220	-975.2134804	144.0045595	3
221	20.44465969	204.0198115	3
222	4.731570465	373.9927894	3
223	0	0	5
224	-720.0055272	1428.764966	4
225	-720.0055272	1428.764966	4
226	-720.0055272	1428.764966	4
227	-1178.00061	810.0781155	4
228	-1182.91893	814.9993427	4
229	34.55534031	45.98018853	4

**Lampiran 47 : Tabel Link Jalan RAV Pada Parameter 10000 – 300**

Index	Ruas A	Ruas B	Index	Ruas A	Ruas B	Index	Ruas A	Ruas B
1	1	3	131	80	78	261	199	68
2	1	2	132	78	77	262	199	67
3	3	6	133	77	76	263	200	69
4	6	7	134	76	75	264	200	70
5	7	86	135	75	73	265	201	75
6	86	8	136	73	74	266	201	73
7	8	10	137	74	72	267	202	75
8	10	145	138	72	141	268	202	76
9	2	12	139	141	140	269	203	76
10	4	13	140	140	138	270	203	77
11	146	12	141	137	138	271	204	77
12	14	92	142	137	80	272	204	78
13	21	23	143	69	70	273	205	139
14	23	28	144	110	111	274	205	70
15	28	30	145	100	108	275	206	3
16	30	32	146	85	96	276	206	1
17	32	41	147	96	97	277	207	17
18	41	40	148	97	99	278	207	16
19	40	42	149	99	101	279	208	87
20	42	50	150	85	11	280	208	9
21	50	51	151	11	87	281	209	82
22	51	52	152	87	9	282	209	83
23	52	54	153	9	83	283	210	89
24	54	66	154	83	82	284	210	88
25	66	59	155	82	79	285	211	94
26	59	65	156	79	84	286	211	103
27	65	67	157	84	101	287	212	22
28	67	68	158	101	99	288	212	20
29	68	142	159	99	97	289	213	38
30	142	71	160	97	96	290	213	33
31	71	143	161	47	150	291	214	109
32	143	144	162	146	21	292	214	115
33	144	145	163	102	105	293	215	79
34	145	10	164	103	102	294	215	84
35	4	5	165	151	3	295	216	117
36	5	18	166	151	1	296	216	119
37	17	18	167	152	6	297	217	120
38	16	17	168	152	3	298	217	119
39	13	16	169	153	6	299	218	60
40	4	13	170	153	7	300	218	61

41	14	147
42	147	27
43	27	36
44	36	37
45	37	45
46	45	150
47	47	148
48	148	93
49	93	92
50	92	14
51	14	147
52	20	149
53	20	22
54	22	25
55	25	149
56	24	26
57	26	34
58	34	29
59	29	24
60	35	39
61	31	35
62	39	38
63	38	33
64	33	31
65	44	43
66	43	49
67	49	53
68	53	44
69	46	55
70	55	56
71	56	57
72	57	58
73	58	127
74	127	125
75	125	124
76	124	114
77	114	48
78	46	48
79	4	13
80	13	16
81	16	17
82	17	18
83	18	5

171	154	7
172	154	86
173	155	11
174	155	87
175	156	10
176	156	8
177	157	12
178	157	2
179	158	16
180	158	13
181	159	17
182	159	16
183	160	83
184	160	9
185	161	82
186	161	83
187	162	89
188	162	88
189	163	90
190	163	89
191	164	91
192	164	15
193	165	22
194	165	20
195	166	29
196	166	34
197	167	93
198	167	92
199	168	94
200	168	103
201	169	102
202	169	105
203	170	98
204	170	95
205	171	38
206	171	33
207	172	38
208	172	39
209	173	150
210	173	47
211	174	148
212	174	47
213	175	107

301	219	135
302	219	136
303	220	63
304	220	64
305	221	75
306	221	73
307	222	76
308	222	75
309	224	16
310	224	13
311	225	16
312	225	13
313	226	16
314	226	13
315	227	45
316	227	150
317	228	45
318	228	150
319	229	73
320	229	74
321	1	2
322	2	4
323	3	4
324	7	5
325	5	85
326	86	85
327	10	145
328	146	21
329	21	20
330	149	147
331	12	13
332	14	15
333	12	14
334	13	15
335	18	96
336	18	19
337	23	22
338	22	24
339	25	27
340	25	26
341	92	93
342	93	94
343	91	94



84	5	4
85	15	19
86	19	88
87	88	89
88	89	90
89	90	91
90	91	15
91	94	95
92	95	98
93	98	103
94	103	94
95	104	100
96	108	107
97	106	107
98	106	104
99	109	110
100	111	112
101	112	113
102	113	81
103	81	121
104	121	120
105	120	119
106	117	119
107	116	117
108	116	115
109	115	109
110	114	115
111	123	118
112	118	122
113	122	132
114	132	131
115	131	130
116	130	129
117	129	126
118	126	123
119	128	60
120	60	61
121	61	62
122	62	134
123	134	135
124	135	136
125	136	133
126	133	128

214	175	106
215	176	79
216	176	84
217	177	40
218	177	41
219	178	49
220	178	43
221	179	46
222	179	55
223	180	110
224	180	109
225	181	111
226	181	110
227	182	111
228	182	112
229	183	114
230	183	124
231	184	116
232	184	115
233	185	117
234	185	119
235	186	120
236	186	119
237	187	120
238	187	121
239	188	81
240	188	113
241	189	118
242	189	123
243	190	126
244	190	123
245	191	129
246	191	126
247	192	130
248	192	131
249	193	131
250	193	132
251	194	58
252	194	57
253	195	62
254	195	61
255	196	134
256	196	135

344	88	95
345	95	97
346	28	29
347	30	31
348	29	31
349	34	36
350	35	37
351	148	104
352	103	104
353	99	98
354	98	100
355	100	101
356	32	33
357	42	43
358	39	45
359	39	44
360	44	46
361	47	48
362	48	109
363	107	110
364	108	84
365	113	108
366	79	78
367	50	49
368	53	55
369	114	115
370	124	123
371	116	123
372	117	118
373	121	122
374	127	128
375	125	126
376	132	133
377	133	137
378	57	66
379	58	60
380	60	61
381	62	63
382	65	67
383	65	64
384	64	68
385	70	141
386	136	139

127	63	64	257	197	135	387	136	138
128	64	69	258	197	136	388	139	140
129	70	139	259	198	63	389	223	74
130	139	63	260	198	64			

### Lampiran 48 : Spesifikasi MDU ZTE

#### ZXDSL 9806H[MDU]



#### Network Interface

- 1 GPON/1 EPON/1 GE port
- 2.488Gbit/s downstream with 1490 nm wavelength
- 1.244Gbit/s upstream with 1310 nm wavelength

#### Subscriber Interface

- 96 ADSL/ADSL2+/SHDSL or 64 VDSL2 interfaces

#### Physical specification

Dimensions: 482.6mm(W) ×88.9mm(H) ×240mm(D)

## Lampiran 49 : Spesifikasi RK ZTE

Photograp	Model	Physical parameters	Interface	Capability indices
	Standard Rack	Open rack Dimensions: 2000mm(H)*600mm(W)*600mm(D) Operating voltage -57V~40V Weight(max): 220kg Ambient temperature: 0°C~40°C Ambient humidity: 20%~80%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	1E6POTS, 424ADSL+528POTS(front access) 1056POTS,528ADSL-528POTS(front access) Uplink: GE, FE, E1. Supports 256 VLANs or 4K 50K BHCA 8.8G/12.8G/48G switching capacity for Data service
	ONU100	Indoor Dimensions: 421mm(H)*218mm(W)*456mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 18kg Ambient temperature: 0°C~40°C Ambient humidity: 20%~80%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	96POTS, 48ADSL+48POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8G/12.8G/48G switching capacity for Data service
	OUT30	Outdoor Dimensions: 1992mm(H)*1350mm(W)*710mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 600kg Ambient temperature: -35°C~55°C Ambient humidity: 5%~95%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	1088POTS, 544ADSL+544POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8.8G/12.8G/48G switching capacity for Data service
	OUT30E	Outdoor Dimensions: 2073mm(H)*1400mm(W)*650mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 600kg Ambient temperature: -35°C~55°C Ambient humidity: 5%~95%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	1088POTS, 544ADSL+544POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8.8G/12.8G/48G switching capacity for Data service
	OUT30F	Outdoor Dimensions: 1900mm(H)*1450mm(W)*600mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 750kg Ambient temperature: -35°C~55°C Ambient humidity: 5%~95%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	2048POTS, 1024ADSL+1024POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8.8G/12.8G/48G switching capacity for Data service
	OUT40	Outdoor Dimensions: 1450mm(H)*1500mm(W)*500mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 520kg Ambient temperature: -35°C~55°C Ambient humidity: 5%~95%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	704POTS, 352ADSL+352POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8.8G/12.8G/48G switching capacity for Data service
	OUT50	Outdoor/Indoor Dimensions: 1100mm(H)*700mm(W)*400mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 140kg Ambient temperature: -35°C~50°C Ambient humidity: 5%~95%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	192POTS, 96ADSL+96POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8G/12.8G/48G switching capacity for Data service
	OUT60	Outdoor/Front access Dimensions: 1600mm(H)*1594mm(W)*780mm(D) Operating voltage -57V~40V or 220VAC±30% Weight(max): 620kg Ambient temperature: -35°C~55°C Ambient humidity: 5%~95%	POTS, ADSL/ADSL2+, VDSL, S/HDSL, FE, BRI/PRI, DDN; Build-in MSTP/EPON/GPON	1056POTS, 528ADSL+528POTS, Uplink: GE, FE, E1 Supports 256 VLANs or 4K 50K BHCA 8.8G/12.8G/48G switching capacity for Data service

## Lampiran 50 : Sumber Kode Aplikasi

Fungsi gui utama :

```
function varargout = gui_utama(varargin)
% GUI_UTAMA M-file for gui_utama.fig
% GUI_UTAMA, by itself, creates a new GUI_UTAMA or raises the
existing
% singleton*.
%
```

```

%      H = GUI_UTAMA returns the handle to a new GUI_UTAMA or the
handle to
%      the existing singleton*.
%
%      GUI_UTAMA('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in GUI_UTAMA.M with the given input
arguments.
%
%      GUI_UTAMA('Property','Value',...) creates a new GUI_UTAMA
or raises the
%      existing singleton*. Starting from the left, property
value pairs are
%      applied to the GUI before gui_utama_OpeningFunction gets
called. An
%      unrecognized property name or invalid value makes property
application
%      stop. All inputs are passed to gui_utama_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help gui_utama

% Last Modified by GUIDE v2.5 21-May-2008 06:39:34

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @gui_utama_OpeningFcn, ...
                  'gui_OutputFcn',  @gui_utama_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gui_utama is made visible.
function gui_utama_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to gui_utama (see VARARGIN)

% Choose default command line output for gui_utama
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_utama wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_utama_OutputFcn(hObject, eventdata,
handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in petadasar.
function petadasar_Callback(hObject, eventdata, handles)
% hObject handle to petadasar (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of petadasar

% --- Executes on button press in dp.
function dp_Callback(hObject, eventdata, handles)
% hObject handle to dp (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of dp

% --- Executes on button press in gl.
function gl_Callback(hObject, eventdata, handles)
% hObject handle to gl (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of gl

```

```

% --- Executes on button press in hubunganlogik.
function hubunganlogik_Callback(hObject, eventdata, handles)
% hObject    handle to hubunganlogik (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of hubunganlogik

% --- Executes on button press in primer.
function primer_Callback(hObject, eventdata, handles)
% hObject    handle to primer (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of primer

% --- Executes on button press in sekunder.
function sekunder_Callback(hObject, eventdata, handles)
% hObject    handle to sekunder (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of sekunder

% --- Executes on button press in jalankan.
function jalankan_Callback(hObject, eventdata, handles)
% hObject    handle to jalankan (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global LinkJalan KoordinatTitikJalan KoordinatDP KebutuhanDPRate
LinkSentral Nyebrang
tic
Q=str2num(get(handles.Q, 'string'));
RadiusGrup=str2num(get(handles.RadiusGrup, 'string'));
axes(handles.axes1)
if get(handles.petadasar, 'value')==1
gambarpetadasar(handles, LinkJalan, KoordinatTitikJalan)
%pause;
end
if get(handles.dp, 'value')==1
axes(handles.axes1)
    for i=1:size(KoordinatDP,1)

plot(KoordinatDP(:,1),KoordinatDP(:,2), 'ro', 'markerfacecolor', 'r')
        hold on
    end
end
end

```

```

if get(handles.hubunganlogik,'value')==1
hublogik = 1
else
hublogik = 0
end

[KoordinatRK, NamaRK, KoordinatTitikJalan,
LinkJalan]=ScatterSteiner(handles, hublogik, RadiusGrup,
KoordinatDP,KebutuhanDPRate,Q, KoordinatTitikJalan,
LinkJalan,LinkSentral,Nyebrang);
%[KoordinatTitikJalan,LinkJalan, KoordinatRK, NamaRK]=...
%KoordinatRK_dipeta(KoordinatTitikJalan,LinkJalan,KoordinatRK,Nama
RK, MatrikKoneksi);

T=toc;

set(handles.waktu,'string',num2str(T))

% --- Executes on button press in bersihkan.
function bersihkan_Callback(hObject, eventdata, handles)
% hObject    handle to bersihkan (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1)
cla

% --- Executes on button press in tutup.
function tutup_Callback(hObject, eventdata, handles)
% hObject    handle to tutup (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

function Q_Callback(hObject, eventdata, handles)
% hObject    handle to Q (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Q as text
%        str2double(get(hObject,'String')) returns contents of Q
as a double

% --- Executes during object creation, after setting all
properties.
function Q_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Q (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundCol
or'));
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit2 as text
%       str2double(get(hObject, 'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundCol
or'));
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit3 as text
%       str2double(get(hObject, 'String')) returns contents of
edit3 as a double

```



```

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundCol
or'));
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit4 as text
%         str2double(get(hObject, 'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundCol
or'));
end

```

```

% --- Executes on button press in filedata.
function filedata_Callback(hObject, eventdata, handles)
% hObject    handle to filedata (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global LinkJalan KoordinatTitikJalan KoordinatDP KebutuhanDPRate
LinkSentral Nyebrang
[file,path]=uigetfile('*.xls');

if file==0
    return
end
h = waitbar(0,'Mohon Tunggu...');
[numk,txtk]=xlsread([path file],'Koordinat');
waitbar(1/7,h)
[numl,txtl]=xlsread([path file],'LinkJalan');
waitbar(2/7,h)
[numdp,txtdp]=xlsread([path file],'KoordinatDP');
waitbar(3/7,h)

[numdprate,txtdprate]=xlsread([path file],'KebutuhanDPRate');
waitbar(4/7,h)

[numlinkDP,txtlinkDP]=xlsread([path file],'LinkDP');
waitbar(5/7,h)

[numlinksentral,txtlinksentral]=xlsread([path
file],'KeluarSentral');
waitbar(6/7,h)

[numlinknyebrang,txtlinknyebrang]=xlsread([path
file],'NyeberangJalan');
waitbar(7/7,h)

close(h)
LinkJalan=numl;
KoordinatTitikJalan=numk;
KoordinatDP=numdp;
KebutuhanDPRate=numdprate;
LinkDP = numlinkDP;
LinkSentral= numlinksentral;
Nyebrang= numlinknyebrang;

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of
edit5 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol
or'));
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol
or'));
end

function waktu_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to waktu (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of waktu as text
%           str2double(get(hObject,'String')) returns contents of
waktu as a double

% --- Executes during object creation, after setting all
properties.
function waktu_CreateFcn(hObject, eventdata, handles)
% hObject      handle to waktu (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundCol
or'));
end

% --- Executes on button press in br.
function br_Callback(hObject, eventdata, handles)
% hObject      handle to br (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of br

function RadiusGrup_Callback(hObject, eventdata, handles)
% hObject      handle to RadiusGrup (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RadiusGrup as
text
%           str2double(get(hObject,'String')) returns contents of
RadiusGrup as a double

```

```

% --- Executes during object creation, after setting all
properties.
function RadiusGrup_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RadiusGrup (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

### Fungsi Scatter Steiner :

```

function [KoordinatRK, NamaRK, KoordinatTitikJalan,
LinkJalan]=ScatterSteiner(handles, hublogik, RGrup,
KoordinatDP, KebutuhanDPRate, Q, KoordinatTitikJalan,
LinkJalan, LinkSentral, Nyebrang);

% ScatterSteiner(handles,X,Y,Q)
global MatrikSteiner
global N
global titiksteinerpertama

%Diatas adalah DP, dibawah mempersiapkan RK untuk di steiner
[KoordinatRK, NamaRK, MatrikTeman ] = CariTitikRK(hublogik, RGrup,
KoordinatDP, KebutuhanDPRate, KoordinatTitikJalan, LinkJalan);

KoordinatSwitch = [0 0];
KoordinatRK = [KoordinatRK; KoordinatSwitch];
sizeRK = size(NamaRK);
sizeRK = sizeRK(1,1);
BitRateSwitch = [(sizeRK+1) 0];
NamaRK = [NamaRK; BitRateSwitch];
%Persiapak RK selesai

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%JANGAN DILAKUKAN
%Memasukkan Koordinat DP pada Koordinat Titik Jalab
% KoordinatTitikJalan = [KoordinatTitikJalan; KoordinatDP];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if Q<max(NamaRK(:,2))
    warndlg(['Maksimum Kapasitas PON Primer Kurang, Minimum = '
num2str(max(NamaRK(:,2)))]);
    return
end
%Q adalah kapasitas PON untuk kabel primer RK
Q = Q;
PanjangJalurTerbaik = 999999;
IndeksTerbaik = 0;

```

```

%cari indeks
N = size(KoordinatRK,1);
Indeks = 1:1:N;

%Menghitung jarak antar kota, langsung di tampilkan di solusi awal
D = zeros(N,N);
for i=1:(N-1)
    for j=i+1:N,
        D(i,j)=D(i,j)+sqrt((KoordinatRK(Indeks(i),1)-
KoordinatRK(Indeks(j),1))^2 ...
+ (KoordinatRK(Indeks(i),2)-
KoordinatRK(Indeks(j),2))^2);
        D(j,i)=D(i,j);
    end;
end;

%Mencari Koordinat Polar setiap titik
KoordinatPolar = 0;
for i=1:N-1
    [Theta, Rho] = cart2pol(KoordinatRK(i,1),KoordinatRK(i,2));
    KoordinatPolar(i,1)=Theta;
    KoordinatPolar(i,2)=Rho;
end;

%Menggabung Matrik ThetaRho dgn Matrik BitRate
for i=1:N-1
    MatrikUmum(i,1)=KoordinatPolar(i,1);
    MatrikUmum(i,2)=KoordinatPolar(i,2);
    MatrikUmum(i,3)=NamaRK(i,1);
    MatrikUmum(i,4)=NamaRK(i,2);
end

%Mencari Klusterisasi, hasilnya SortMatrikUmum
SortMatrikUmum = sortrows(MatrikUmum)
SortMatrikUmum(N,1)=0;
SortMatrikUmum(N,2)=0;
SortMatrikUmum(N,3)=N;
SortMatrikUmum(N,4)=0;

%Disini harusnya adalah function, ada fungsi shifted
%Proses shifted (Heuristic), LOOP BESAR
%Terpaksa N dikopi dulu
TotalElemen = N;
%Dikopi di awal sebagai acuan, dgn loop dibawah baru di obok2
SortMatrikUmumKopi= SortMatrikUmum;
%Mari kita catat SOLusi
KumpulanIndeks = zeros(1,60);
CounterIndeks = 1;

for CounterLoopBesar = 1:N
    if CounterLoopBesar ==1
        %SortMatrikUmum = SortMatrikUmum;
    end
    MatrikSementara = 0;
    if CounterLoopBesar >1 %Ambil 1 geser 39, taruh 1 di 39 dan
digeser terus menerus selama 39x
        %BaruMatrikDitukar
        MatrikSementara(1,1)=SortMatrikUmumKopi(1,1);
    end
end

```

```

MatrikSementara(1,2)=SortMatrikUmumKopi(1,2);
MatrikSementara(1,3)=SortMatrikUmumKopi(1,3);
MatrikSementara(1,4)=SortMatrikUmumKopi(1,4);
for CounterJ=1:N-2

SortMatrikUmumKopi(CounterJ,1)=SortMatrikUmumKopi(CounterJ+1,1);

SortMatrikUmumKopi(CounterJ,2)=SortMatrikUmumKopi(CounterJ+1,2);

SortMatrikUmumKopi(CounterJ,3)=SortMatrikUmumKopi(CounterJ+1,3);

SortMatrikUmumKopi(CounterJ,4)=SortMatrikUmumKopi(CounterJ+1,4);
end
SortMatrikUmumKopi(N-1,1)=MatrikSementara(1,1);
SortMatrikUmumKopi(N-1,2)=MatrikSementara(1,2);
SortMatrikUmumKopi(N-1,3)=MatrikSementara(1,3);
SortMatrikUmumKopi(N-1,4)=MatrikSementara(1,4);
%kopi Kembali SortMatrikUmum
%SortMatrikUmum = 0;
SortMatrikUmum = SortMatrikUmumKopi;
end;

%Dibawah ini mulai baganProgram
%Kabel yang akan di deploy adalah Q Mbps.
%
%Membuat Clusterisasi

i = 1; j = 2; kaps = 0; IsiCluster = 0;
SortMatrikUmumKaps = 0;
while kaps <= Q
    IsiCluster(1,1) = 0; IsiCluster(1,2) = 0; IsiCluster(1,3)
= TotalElemen; IsiCluster(1,4) = 0;
    kaps = kaps + SortMatrikUmum(i,4);

    if (kaps <= Q) && ((i ~= TotalElemen))
        IsiCluster(j,1) = SortMatrikUmum(i,1);
        IsiCluster(j,2) = SortMatrikUmum(i,2);
        IsiCluster(j,3) = SortMatrikUmum(i,3);
        IsiCluster(j,4) = SortMatrikUmum(i,4);
        j = j + 1;
        i = i + 1;
        IsiCluster(j,1) = 0; IsiCluster(j,2) = 0;
IsiCluster(j,3) = TotalElemen; IsiCluster(j,4) = 0;
    end
    if (kaps >= Q) && (i ~= TotalElemen)
        j = 2; kaps = 0;
        if SortMatrikUmumKaps == 0
            SortMatrikUmumKaps = IsiCluster;
        else
            SortMatrikUmumKaps = [SortMatrikUmumKaps;
IsiCluster];
        end
        IsiCluster = 0;
    end
end
if (i == TotalElemen)
    if SortMatrikUmumKaps == 0
        SortMatrikUmumKaps = IsiCluster;
    else

```

```

        SortMatrikUmumKaps = [SortMatrikUmumKaps;
IsiCluster];
        end
        break
    end
end
SortMatrikUmum = SortMatrikUmumKaps;

%Extract Indeks
A = size(SortMatrikUmum);
Indeks=1;
A= A(1,1);
for i=1:A
    Indeks(i,1)=SortMatrikUmum(i,3);
end;
Indeks=Indeks';

%Menggambar semua titik dan dihubungkan dengan garis
%matris Koordinat adalah x,y, Indeks adalah urutan titiknya
Koordinat = KoordinatRK;
KbthBR = NamaRK;

%Membuat analisa jarak
Counter = size(Indeks);
Counter = Counter(1,2);
PanjangJalur = 0;
for i=1:Counter-1
    PanjangJalur = PanjangJalur +
D(Indeks(1,i),Indeks(1,i+1));
end;

%Peringat siapa yg terbaik
PanjangJalurLoop = PanjangJalur;
IndeksLoop = Indeks;

if PanjangJalurTerbaik > PanjangJalurLoop
    PanjangJalurTerbaik = PanjangJalurLoop;
    IndeksTerbaik = IndeksLoop;
    SortMatrikUmumTerbaik = SortMatrikUmum;
end;

%Catat Solusi
A = size(IndeksLoop);
A = A(1,2);
for l = 1:A
    KumpulanIndeks(CounterIndeks,l) = IndeksLoop(1,l);
end
KumpulanIndeks(CounterIndeks,60)=PanjangJalurLoop;
CounterIndeks = CounterIndeks+1;

%Akhir Loop Besar
%pause;
end

IndeksTerbaik;

```



```

%Flower Petal Terbaik selesai

%..... Proses selanjutnya mengerti sudut, Gunakan Indeks
Terbaik
UkuranIndeks = size(IndeksTerbaik);
UkuranIndeks =UkuranIndeks(1,2);
SortMatrikUmumTerbaik(1,6)=NaN;
SortMatrikUmumTerbaik(UkuranIndeks,6)=NaN;
for i = 2 : UkuranIndeks-1
    TitikPertama = IndeksTerbaik(1,i-1);
    TitikKedua   = IndeksTerbaik(1,i);
    TitikKetiga  = IndeksTerbaik(1,i+1);
    x1=Koordinat(TitikPertama,1); y1=Koordinat(TitikPertama,2);
    x2=Koordinat(TitikKedua,1); y2=Koordinat(TitikKedua,2);
    x3=Koordinat(TitikKetiga,1); y3=Koordinat(TitikKetiga,2);
    %SekarangMencariCos Alpha
    Atas = ((x1-x2)*(x3-x2))+((y1-y2)*(y3-y2));
    Bawah = sqrt((x1-x2)^2+(y1-y2)^2)*sqrt((x3-x2)^2+(y3-y2)^2);
    Alpha = acosd(Atas/Bawah);
    SortMatrikUmumTerbaik(i,6)=Alpha;
end

%----- BATAS SORT Matrik UMUM-----
%_____BIG
FOR _____
UkuranIndeksTerbaik = size(IndeksTerbaik);
UkuranIndeksTerbaik = UkuranIndeksTerbaik(1,2);
j = 1
for i = 1: UkuranIndeksTerbaik
    if IndeksTerbaik(1,i) == N
        MatrikN(j,1)= i;
        j = j + 1;
    end
end
UkuranMatrikN = size(MatrikN);
UkuranMatrikN = UkuranMatrikN(1,1);
%Start BIG FOR
MSUbuf=[];
for LoopCluster = 1:2:UkuranMatrikN
    BatasAtasCluster = MatrikN(LoopCluster,1);
    BatasBawahCluster = MatrikN(LoopCluster+1,1);
    IndeksCluster = 0;
    jj = 1;
    for IsiCluster = BatasAtasCluster : BatasBawahCluster
        IndeksCluster(1,jj) = IndeksTerbaik(1,IsiCluster);
        jj = jj + 1;
    end
    %..... BIG FOR CONTINUES
    %Buat Matrik Steiner = [Row TtkA TtkB Xa Ya Xb Yb Derajat
Status]
    jumlahanggota = size(IndeksCluster);
    jumlahanggota = jumlahanggota(1,2);
    banyakruassudut = (jumlahanggota-2);
    MatrikSteiner = 0;
    MatrikSteiner = zeros(2*banyakruassudut,9);
    for i = 1 : banyakruassudut
        MatrikSteiner(2*i-1,1) = 2*i-1;
        MatrikSteiner(2*i-1,2) = IndeksCluster(1,(1+i)-1);
    end
end

```

```

        MatrikSteiner(2*i-1,3) = IndeksCluster(1,(1+i));
        MatrikSteiner(2*i-1,4) = Koordinat(IndeksCluster(1,(1+i)-
1),1);
        MatrikSteiner(2*i-1,5) = Koordinat(IndeksCluster(1,(1+i)-
1),2);
        MatrikSteiner(2*i-1,6) =
Koordinat(IndeksCluster(1,(1+i)),1);
        MatrikSteiner(2*i-1,7) =
Koordinat(IndeksCluster(1,(1+i)),2);
        MatrikSteiner(2*i-1,8) = 0;
        MatrikSteiner(2*i-1,9) = 0;
        MatrikSteiner(2*i,1) = 2*i;
        MatrikSteiner(2*i,2) = IndeksCluster(1,(1+i));
        MatrikSteiner(2*i,3) = IndeksCluster(1,(1+i)+1);
        MatrikSteiner(2*i,4) = Koordinat(IndeksCluster(1,(1+i)),1)
        MatrikSteiner(2*i,5) = Koordinat(IndeksCluster(1,(1+i)),2)
        MatrikSteiner(2*i,6) =
Koordinat(IndeksCluster(1,(1+i)+1),1)
        MatrikSteiner(2*i,7) =
Koordinat(IndeksCluster(1,(1+i)+1),2)
        MatrikSteiner(2*i,8) = 0;
        MatrikSteiner(2*i,9) = 0;
        MatrikSteiner(2*i-
1,8)=hitungsudut3titik(MatrikSteiner(2*i-1,4),MatrikSteiner(2*i-
1,5),MatrikSteiner(2*i-1,6),MatrikSteiner(2*i-
1,7),MatrikSteiner(2*i,6),MatrikSteiner(2*i,7));
        MatrikSteiner(2*i,8) = MatrikSteiner(2*i-1,8);
    end;
    %Mencari Sudut NaN
    UkuranMS = size(MatrikSteiner);
    UkuranMS = UkuranMS(1,1);
    for i = 1: UkuranMS
        if isnan(MatrikSteiner(i,8)) == 1
            MatrikSteiner(i,8) = 0;
        end
    end
    %-----MatrikSteinerAwalSelesai

    %-----Optimizing Steiner
    urut_e=1;
    quitoptimizing = 0;
    titiksteiner=size(KoordinatRK);
    titiksteiner = titiksteiner(1,1)+1;
    titiksteinerpertama = titiksteiner;
    while quitoptimizing ~= 1
        clear MSA; clear MST; clear MSB;
        if urut_e == 1
            %pause
            ukuranloop = size(MatrikSteiner);
            ukuranloop = ukuranloop(1,1);
            for i=1:2:ukuranloop
                if (MatrikSteiner(i,9) == 1)
                    i = i + 2;
                end
                if (MatrikSteiner(i,8) >= 120) &&
(MatrikSteiner(i,9) == 0) && (ukuranloop ~= 4)
                    MatrikSteiner(i,9) = 1;
                    MatrikSteiner(i+1,9) = 1;
                    urut_e = 2;
                end
            end
        end
    end
end

```

```

        break
    end
    if (MatrikSteiner(i,8)) < 120 &&
(MatrikSteiner(i,9) == 0)
        [Xd,Yd,PjgKabelSteiner] = carititiksteiner(...
            MatrikSteiner(i,4),MatrikSteiner(i,5),...
            MatrikSteiner(i,6),MatrikSteiner(i,7),...
MatrikSteiner(i+1,6),MatrikSteiner(i+1,7));
        PjgKabelExisting = sqrt((MatrikSteiner(i,4)-
MatrikSteiner(i,6))^2+(MatrikSteiner(i,5)-
MatrikSteiner(i,7))^2)+...
            sqrt((MatrikSteiner(i,6)-
MatrikSteiner(i+1,6))^2+(MatrikSteiner(i,7)-
MatrikSteiner(i+1,7))^2);

        %Apabila i==1 & Ukuran Loop =4
        %KASUS KHUSUS
        if (PjgKabelSteiner > PjgKabelExisting) &&
(i==1) && (ukuranloop == 4)
            [MSB] =
MatrikSteinerBawah(i,MatrikSteiner,urut_e,ukuranloop);
            MatrikSteiner = MSB;
            MatrikSteiner(i,9) = 1;
            MatrikSteiner(i+1,9) = 1;
            urut_e = 2;
            break
        end

        if PjgKabelSteiner > PjgKabelExisting
            MatrikSteiner(i,9) = 1;
            MatrikSteiner(i+1,9) = 1;
            urut_e = 2;
            break
        end

        if PjgKabelSteiner < PjgKabelExisting
            % Apabila i == 1 & ukuran MatrikSteiner=2
(9-1-9)

            %KASUS KHUSUS
            if (i == 1) && (ukuranloop == 2)
                MatrikSteiner(1,9)=1;
                MatrikSteiner(2,9)=1;
                break
            end

            if i == 1
                a=0;b=0;c=0;
                %MatrikSteinerAtas =
MatrikSteinerTengah
                a =
MatrikSteiner(i,2);b=MatrikSteiner(i,3);c=MatrikSteiner(i+1,3);
                [MST KoordinatRK
NamaRK]=MatrikSteinerTengah(a,b,c,titiksteiner,Xd,Yd,urut_e,
KoordinatRK, NamaRK);

                %Merubah dulu sisa dari b dengan titik
steiner pada

```

```

%sisas MatrikSteiner sebelum di proses
kebih lanjut
for ii = (i+2) : ukuranloop
    if MatrikSteiner(ii,2) == b
        MatrikSteiner(ii,2) =
titiksteiner;
            MatrikSteiner(ii,4) = Xd;
            MatrikSteiner(ii,5) = Yd;
        end
        if MatrikSteiner(ii,3) == b
            MatrikSteiner(ii,3) =
titiksteiner;
                MatrikSteiner(ii,6) = Xd;
                MatrikSteiner(ii,7) = Yd;
            end
        end
        %MembetulkanSudut lagi
        for ii = (i+2):2:ukuranloop
MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));
MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
            if MatrikSteiner(ii,8)>=120
                MatrikSteiner(ii,9)=1;
                MatrikSteiner(ii+1,9)=1;
            elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
                MatrikSteiner(ii,9)=0;
                MatrikSteiner(ii+1,9)=0;
            elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)
                MatrikSteiner(ii,9)=1;
                MatrikSteiner(ii+1,9)=1;
            else
            end
        end
        %MatrikSteinerBawah
        [MSB] =
MatrikSteinerBawah(i,MatrikSteiner,urut_e,ukuranloop);
        %rapikan parameter kontrol
        urut_e = 2;
        titiksteiner = titiksteiner + 1;
        MatrikSteiner = [MST; MSB];
        break
    end

    if (i ~= 1) && (i ~= ukuranloop-1)
        a=0;b=0;c=0;
        %MembuatMatrikTengah
        (a,b,c,titiksteiner,Xd,Yd,urut)
        a =
MatrikSteiner(i,2);b=MatrikSteiner(i,3);c=MatrikSteiner(i+1,3);
        [MST KoordinatRK
NamaRK]=MatrikSteinerTengah(a,b,c,titiksteiner,Xd,Yd,urut_e,
KoordinatRK, NamaRK);
        %Merubah dulu sisa dari b dengan titik
        steiner pada

```

```

%sisanya MatrikSteiner sebelum di proses
kebijah lanjut
for ii = 1 : (i-1) %Before
    if MatrikSteiner(ii,2) == b
        MatrikSteiner(ii,2) =
titiksteiner;
            MatrikSteiner(ii,4) = Xd;
            MatrikSteiner(ii,5) = Yd;
        end
    if MatrikSteiner(ii,3) == b
        MatrikSteiner(ii,3) =
titiksteiner;
            MatrikSteiner(ii,6) = Xd;
            MatrikSteiner(ii,7) = Yd;
        end
    end
    %MembetulkanSudut lagi
    for ii = 1:2:(i-1)
        MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));
        MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
        if MatrikSteiner(ii,8)>=120
            MatrikSteiner(ii,9)=1;
            MatrikSteiner(ii+1,9)=1;
        elseif (MatrikSteiner(ii,8)<120)
            MatrikSteiner(ii,9)=0;
            MatrikSteiner(ii+1,9)=0;
        elseif (MatrikSteiner(ii,8)<120)
            MatrikSteiner(ii,9)=1;
            MatrikSteiner(ii+1,9)=1;
        else
            end
        end
    end
    for ii =(i+2): ukuranloop %after
        if MatrikSteiner(ii,2) == b
            MatrikSteiner(ii,2) =
titiksteiner;
                MatrikSteiner(ii,4) = Xd;
                MatrikSteiner(ii,5) = Yd;
            end
        if MatrikSteiner(ii,3) == b
            MatrikSteiner(ii,3) =
titiksteiner;
                MatrikSteiner(ii,6) = Xd;
                MatrikSteiner(ii,7) = Yd;
            end
        end
        %MembetulkanSudut lagi
        for ii = (i+2):2:ukuranloop
            MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));

```

```

MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
    if MatrikSteiner(ii,8)>=120
        MatrikSteiner(ii,9)=1;
        MatrikSteiner(ii+1,9)=1;
    elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
        MatrikSteiner(ii,9)=0;
        MatrikSteiner(ii+1,9)=0;
    elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)
        MatrikSteiner(ii,9)=1;
        MatrikSteiner(ii+1,9)=1;
    else
    end
end
%MembuatMatrikAtas
[MSA] =
MatrikSteinerAtas(i,MatrikSteiner,urut_e);
%MembuatMatrikBawah
[MSB] =
MatrikSteinerBawah(i,MatrikSteiner,urut_e,ukuranloop);
%rapikan parameter kontrol
urut_e = 2;
titiksteiner = titiksteiner + 1;
MatrikSteiner = [MSA; MST; MSB];
break
else
%MembuatMatrikTengah
(a,b,c,titiksteiner,Xd,Yd,urut)
a =
MatrikSteiner(i,2);b=MatrikSteiner(i,3);c=MatrikSteiner(i+1,3);
[MST KoordinatRK
NamaRK]=MatrikSteinerTengah(a,b,c,titiksteiner,Xd,Yd,urut_e,
KoordinatRK, NamaRK);
%Merubah dulu sisa dari b dengan titik
steiner pada
%sisia MatrikSteiner sebelum di proses
kebih lanjut
for ii = 1 : (i-1)
    if MatrikSteiner(ii,2) == b
        MatrikSteiner(ii,2) =
titiksteiner;
        MatrikSteiner(ii,4) = Xd;
        MatrikSteiner(ii,5) = Yd;
    end
    if MatrikSteiner(ii,3) == b
        MatrikSteiner(ii,3) =
titiksteiner;
        MatrikSteiner(ii,6) = Xd;
        MatrikSteiner(ii,7) = Yd;
    end
end
%Mengecek Sudut Lagi
for ii = 1:2:ukuranloop-2
MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));

```

```

MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
                                if MatrikSteiner(ii,8)>=120
                                    MatrikSteiner(ii,9)=1;
                                    MatrikSteiner(ii+1,9)=1;
                                elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
                                    MatrikSteiner(ii,9)=0;
                                    MatrikSteiner(ii+1,9)=0;
                                elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)
                                    MatrikSteiner(ii,9)=1;
                                    MatrikSteiner(ii+1,9)=1;
                                else
                                    end
                                end
                                end
                                %MembuatMatrikAtas
                                [MSA] =
MatrikSteinerAtas(i,MatrikSteiner,urut_e);
                                %MembuatMatrikBawah, tdk perlu
dilakukan

                                %rapikan parameter kontrol
                                urut_e = 2;
                                titiksteiner = titiksteiner + 1;
                                MatrikSteiner = [MSA; MST];
                                break
                                end
                                end
                                end
                                end
                                %pause;
end
% -----
if urut_e == 2
    %pause
    ukuranloop = size(MatrikSteiner);
    ukuranloop = ukuranloop(1,1);
    for i=ukuranloop:-2:0
        if (MatrikSteiner(i,9) == 1)
            i = i - 2;
            if i == 0
                break
            end
        end
        end
        if (MatrikSteiner(i,8) >= 120) &&
(MatrikSteiner(i,9) == 0)
            MatrikSteiner(i,9) = 1;
            MatrikSteiner(i-1,9) = 1;
            urut_e = 1;
            break
        end
        if (MatrikSteiner(i,8) < 120) &&
(MatrikSteiner(i,9) == 0)
            [Xd,Yd,PjgKabelSteiner] = carititiksteiner(...
                MatrikSteiner(i,6),MatrikSteiner(i,7),...
                MatrikSteiner(i-1,6),MatrikSteiner(i-
1,7),...

```

```

MatrikSteiner(i-1,4),MatrikSteiner(i-
1,5));
PjgKabelExisting = sqrt((MatrikSteiner(i-1,4)-
MatrikSteiner(i-1,6))^2+(MatrikSteiner(i-1,5)-MatrikSteiner(i-
1,7))^2)+...
sqrt((MatrikSteiner(i-1,6)-
MatrikSteiner(i,6))^2+(MatrikSteiner(i-1,7)-
MatrikSteiner(i,7))^2);
if PjgKabelSteiner > PjgKabelExisting
MatrikSteiner(i,9) = 1;
MatrikSteiner(i-1,9) = 1;
urut_e = 1;
break
end
if PjgKabelSteiner < PjgKabelExisting
if i == ukuranloop
%MatrikSteinerTengah =
MatrikSteinerBawah
a =
MatrikSteiner(i,3);b=MatrikSteiner(i,2);c=MatrikSteiner(i-1,2);
[MST KoordinatRK
NamaRK]=MatrikSteinerTengah(a,b,c,titiksteiner,Xd,Yd,urut_e,
KoordinatRK, NamaRK);
%Merubah dulu sisa dari b dengan titik
steiner pada
%sisanya MatrikSteiner sebelum di proses
kebih lanjut
for ii = 1 : (i-2)
if MatrikSteiner(ii,2) == b
MatrikSteiner(ii,2) =
titiksteiner;
MatrikSteiner(ii,4) = Xd;
MatrikSteiner(ii,5) = Yd;
end
if MatrikSteiner(ii,3) == b
MatrikSteiner(ii,3) =
titiksteiner;
MatrikSteiner(ii,6) = Xd;
MatrikSteiner(ii,7) = Yd;
end
end
%Mengecek Sudut Lagi
for ii = 1:2:ukuranloop-2
MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSt
einer(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(
ii+1,6),MatrikSteiner(ii+1,7));
MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
if (MatrikSteiner(ii,8)>=120) &&
(MatrikSteiner(ii,9) == 0)
MatrikSteiner(ii,9)=1;
MatrikSteiner(ii+1,9)=1;
elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
MatrikSteiner(ii,9)=0;
MatrikSteiner(ii+1,9)=0;
elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)

```



```

MatrikSteiner(ii,9)=1;
MatrikSteiner(ii+1,9)=1;
else
end
end
end
%MatrikSteinerAtas
[MSA] =
MatrikSteinerAtas(i,MatrikSteiner,urut_e);
%rapikan parameter kontrol
titiksteiner = titiksteiner + 1;
MatrikSteiner = [MSA; MST];
urut_e = 1;
break
end
if (i ~= ukuranloop) && (i ~= 2)
%MembuatMatrikTengah
(a,b,c,titiksteiner,Xd,Yd,urut)
a =
MatrikSteiner(i,3);b=MatrikSteiner(i,2);c=MatrikSteiner(i-1,2);
[MST KoordinatRK
NamaRK]=MatrikSteinerTengah(a,b,c,titiksteiner,Xd,Yd,urut_e,
KoordinatRK, NamaRK);
%Merubah dulu sisa dari b dengan titik
steiner pada
%sisanya MatrikSteiner sebelum di proses
kebih lanjut
for ii = 1 : (i-2)%Before
if MatrikSteiner(ii,2) == b
MatrikSteiner(ii,2) =
titiksteiner;
MatrikSteiner(ii,4) = Xd;
MatrikSteiner(ii,5) = Yd;
end
if MatrikSteiner(ii,3) == b
MatrikSteiner(ii,3) =
titiksteiner;
MatrikSteiner(ii,6) = Xd;
MatrikSteiner(ii,7) = Yd;
end
end
%Mengecek Sudut Lagi
for ii = 1:2:(i-2)
MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));
MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
if MatrikSteiner(ii,8)>=120
MatrikSteiner(ii,9)=1;
MatrikSteiner(ii+1,9)=1;
elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
MatrikSteiner(ii,9)=0;
MatrikSteiner(ii+1,9)=0;
elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)
MatrikSteiner(ii,9)=1;
MatrikSteiner(ii+1,9)=1;

```

```

else
end
end
for ii = (i+1) : ukuranloop%After
if MatrikSteiner(ii,2) == b
MatrikSteiner(ii,2) =
titiksteiner;
MatrikSteiner(ii,4) = Xd;
MatrikSteiner(ii,5) = Yd;
end
if MatrikSteiner(ii,3) == b
MatrikSteiner(ii,3) =
titiksteiner;
MatrikSteiner(ii,6) = Xd;
MatrikSteiner(ii,7) = Yd;
end
end
%Mengecek Sudut Lagi
for ii =(i+1):2:ukuranloop
MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));
MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
if MatrikSteiner(ii,8)>=120
MatrikSteiner(ii,9)=1;
MatrikSteiner(ii+1,9)=1;
elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
MatrikSteiner(ii,9)=0;
MatrikSteiner(ii+1,9)=0;
elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)
MatrikSteiner(ii,9)=1;
MatrikSteiner(ii+1,9)=1;
else
end
end
%MembuatMatrikAtas
[MSA] =
MatrikSteinerAtas(i,MatrikSteiner,urut_e);
%MembuatMatrikBawah
[MSB] =
MatrikSteinerBawah(i,MatrikSteiner,urut_e,ukuranloop);
%rapikan parameter kontrol
urut_e = 1;
titiksteiner = titiksteiner + 1;
MatrikSteiner = [MSA; MST; MSB];
break
else
a=0;b=0;c=0;
%MembuatMatrikAtas Tdk perlu
%MembuatMatrikTengah
(a,b,c,titiksteiner,Xd,Yd,urut)
a =
MatrikSteiner(i,3);b=MatrikSteiner(i,2);c=MatrikSteiner(i-1,2);

```

```

[MST KoordinatRK
NamaRK]=MatrikSteinerTengah(a,b,c,titiksteiner,Xd,Yd,urut_e,
KoordinatRK, NamaRK);
%Merubah dulu sisa dari b dengan titik
steiner pada
%sisanya MatrikSteiner sebelum di proses
kebih lanjut
for ii = (i+1):ukuranloop
    if MatrikSteiner(ii,2) == b
        MatrikSteiner(ii,2) =
titiksteiner;
        MatrikSteiner(ii,4) = Xd;
        MatrikSteiner(ii,5) = Yd;
    end
    if MatrikSteiner(ii,3) == b
        MatrikSteiner(ii,3) =
titiksteiner;
        MatrikSteiner(ii,6) = Xd;
        MatrikSteiner(ii,7) = Yd;
    end
end
%Mengecek Sudut Lagi
for ii = (i+1):2:ukuranloop
MatrikSteiner(ii,8)=hitungsudut3titik(MatrikSteiner(ii,4),MatrikSteiner(ii,5),MatrikSteiner(ii,6),MatrikSteiner(ii,7),MatrikSteiner(ii+1,6),MatrikSteiner(ii+1,7));
MatrikSteiner(ii+1,8)=MatrikSteiner(ii,8);
    if MatrikSteiner(ii,8)>=120
        MatrikSteiner(ii,9)=1;
        MatrikSteiner(ii+1,9)=1;
    elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==0)
        MatrikSteiner(ii,9)=0;
        MatrikSteiner(ii+1,9)=0;
    elseif (MatrikSteiner(ii,8)<120)
&& (MatrikSteiner(ii,9)==1)
        MatrikSteiner(ii,9)=1;
        MatrikSteiner(ii+1,9)=1;
    else
    end
end
%MembuatMatrikBawah
[MSB] =
MatrikSteinerBawah(i,MatrikSteiner,urut_e,ukuranloop);
%rapikan parameter kontrol
urut_e = 1;
titiksteiner = titiksteiner + 1;
MatrikSteiner = [MST; MSB];
break
end
end
end
end
end
%serangkaian re CHECK ing
%-----
%Mendeteksi sdt optimal yg belum di patenkan "1"

```

```

UkuranMS = size(MatrikSteiner);
UkuranMS = UkuranMS(1,1);
for k = 1:UkuranMS
    if (MatrikSteiner(k,8) >= 120)
        MatrikSteiner(k,9)= 1;
    end
end

%Mententukan apakah masih berlanjut loopnya
%-----
UkuranMS = size(MatrikSteiner);
UkuranMS = UkuranMS(1,1);
Total = 0;
for k = 1:UkuranMS
    Total = Total + MatrikSteiner(k,9);
end
if Total == UkuranMS
    quitoptimizing = 1;
else
    quitoptimizing = 0;
end

end

%Menggambar Ruas Total
[MSU MSUX] =
BuatMSU(handles,MatrikSteiner,N,titiksteinerpertama,
IndeksTerbaik);
MSUbuf=[MSUbuf;MSUX(:,1:2)];
%_____BIG FOR _____
test = 0;

end

% KoordinatRK = [
% -108.5 80.5
% 73 833
% -96.5 -412
% 428.5 -116
% 743.5 646
% 818.5 77.5
% 789.5 -237
% 0 0
% ];
% NamaRK = [
% 1 4950
% 2 900
% 3 1350
% 4 7650
% 5 1800
% 6 1800
% 7 1350
% 8 0
% ];

```

```

%Sekarang Mencari titik terdekat ke RK
A = size>NamaRK); A = A(1,1);
B = size>KoordinatTitikJalan); B = B(1,1);
TitikPilihan = 0; Jarak = 0;
TitikPilihanTerbaik = 0; JarakTerbaik = 0;
for i = 1 : A
    for j = 1 : B
        TitikPilihan = j;
        Jarak = sqrt((KoordinatTitikJalan(j,1)-KoordinatRK(i,1))^2
+ (KoordinatTitikJalan(j,2)-KoordinatRK(i,2))^2);
        if TitikPilihanTerbaik == 0
            TitikPilihanTerbaik = j; JarakTerbaik = Jarak;
            NamaRK(i,3) = j; NamaRK(i,4) = JarakTerbaik;
            KoordinatRK(i,3) = KoordinatTitikJalan(j,1);
            KoordinatRK(i,4) = KoordinatTitikJalan(j,2);
            KoordinatRK(i,5) = JarakTerbaik;
        elseif TitikPilihanTerbaik > 0
            if JarakTerbaik > Jarak
                TitikPilihanTerbaik = j; JarakTerbaik = Jarak;
                NamaRK(i,3) = j; NamaRK(i,4) = JarakTerbaik;
                KoordinatRK(i,3) = KoordinatTitikJalan(j,1);
                KoordinatRK(i,4) = KoordinatTitikJalan(j,2);
                KoordinatRK(i,5) = JarakTerbaik;
            else
                end
            else
                end
        end
        TitikPilihanTerbaik = 0;
        Jarak = 0;
    end
end

%Sekarang cari 2 teman yang mengapit si titik terdekat ke RK
B = size(LinkJalan);
B = B(1,1);
for i = 1 : A %Counter A untuk size>NamaRK
    mencariteman = NamaRK(i,3);
    NamaRK(i,5) = 0;
    NamaRK(i,6) = 0;
    for j = 1 : B
        if LinkJalan(j,1) == mencariteman
            if NamaRK(i,5) == 0
                NamaRK(i,5) = LinkJalan(j,2);
            end
            if NamaRK(i,5) > 0
                NamaRK(i,6) = LinkJalan(j,2);
            end
        elseif LinkJalan(j,2) == mencariteman
            if NamaRK(i,5) == 0
                NamaRK(i,5) = LinkJalan(j,1);
            end
            if NamaRK(i,5) > 0
                NamaRK(i,6) = LinkJalan(j,1);
            end
        else
            end
        end
    end
end
end

```

```

%Mencari Koordinat RK sesuai peta
C = size(KoordinatTitikJalan);
C = C(1,1) + 1; %(Counter utk Koordinat)
D = size(LinkJalan);
D = D(1,1) + 1; %(Counter utk link jalan)
for i = 1 : A %Counter A untuk size NamaRK
    KoordinatRKnya = [KoordinatRK(i,1) KoordinatRK(i,2)];
    KoordinatReff = [KoordinatRK(i,3) KoordinatRK(i,4)];
    % try
    KoordinatKiri = [KoordinatTitikJalan>NamaRK(i,5),1
KoordinatTitikJalan>NamaRK(i,5),2)];
    % catch
    %     save datasementara
    %     return
    % end
    KoordinatKanan = [KoordinatTitikJalan>NamaRK(i,6),1
KoordinatTitikJalan>NamaRK(i,6),2)];
    KoordinatRKnya = KoordinatRKnya - KoordinatReff;
    KoordinatKiri = KoordinatKiri - KoordinatReff;
    KoordinatKanan = KoordinatKanan - KoordinatReff;
    [THEkiri,RHOkiri] =
cart2pol(KoordinatKiri(1,1),KoordinatKiri(1,2));
    [THEkanan,RHOkanan] =
cart2pol(KoordinatKanan(1,1),KoordinatKanan(1,2));
    [XRKkiri,YRKkiri] = pol2cart(THEkiri,5);
    [XRKkanan,YRKkanan] = pol2cart(THEkanan,5);
    JarakKiri = sqrt((XRKkiri - KoordinatRKnya(1,1))^2 + (YRKkiri
- KoordinatRKnya(1,2))^2);
    JarakKanan = sqrt((XRKkanan - KoordinatRKnya(1,1))^2 +
(YRKkanan - KoordinatRKnya(1,2))^2);
    if JarakKiri < JarakKanan
        XRKkiri = XRKkiri+KoordinatReff(1,1);
        YRKkiri = YRKkiri+KoordinatReff(1,2);
        KoordinatRK(i,6) = XRKkiri; KoordinatRK(i,7) = YRKkiri;
        KoordinatRK(i,8) = sqrt((XRKkiri-
KoordinatRK(i,1))^2+(YRKkiri-KoordinatRK(i,2))^2);
        %     LinkJalan(D,1) = C; LinkJalan(D,2) =
NamaRK(i,3);
        D = D + 1;
        %     LinkJalan(D,1) = C; LinkJalan(D,2) =
NamaRK(i,5);
        D = D + 1;
        C = C + 1;
    elseif JarakKiri > JarakKanan
        XRKkanan = XRKkanan+KoordinatReff(1,1);
        YRKkanan = YRKkanan+KoordinatReff(1,2);
        KoordinatRK(i,6) = XRKkanan; KoordinatRK(i,7) = YRKkanan;
        KoordinatRK(i,8) = sqrt((XRKkanan-
KoordinatRK(i,1))^2+(YRKkanan-KoordinatRK(i,2))^2);
        %     LinkJalan(D,1) = C; LinkJalan(D,2) =
NamaRK(i,3);
        D = D + 1;
        %     LinkJalan(D,1) = C; LinkJalan(D,2) =
NamaRK(i,6);
        D = D + 1;
        C = C + 1;
    elseif JarakKiri == JarakKanan
        XRKkiri = XRKkiri+KoordinatReff(1,1);

```

```

        YRKKkiri = YRKKkiri+KoordinatReff(1,2);
        KoordinatRK(i,6) = XRKKkiri; KoordinatRK(i,7) = YRKKkiri;
        KoordinatRK(i,8) = sqrt((XRKKkiri-
KoordinatRK(i,1))^2+(YRKKkiri-KoordinatRK(i,2))^2);
        %           LinkJalan(D,1) = C; LinkJalan(D,2) =
NamaRK(i,3);
        D = D + 1;
        %           LinkJalan(D,1) = C; LinkJalan(D,2) =
NamaRK(i,5);
        D = D + 1;
        C = C + 1;
    else
    end
    %
end

[x,y]=find(and(KoordinatRK(:,1)==0,KoordinatRK(:,2)==0));
KoordinatRK(x,6:8)=0;

B=size(KoordinatTitikJalan,1);
KoordinatTitikJalan=[KoordinatTitikJalan;KoordinatDP];
% for i=1:size(KoordinatDP)
%
% end
A=size(KoordinatDP,1);
for i=1:A
    MatrikTeman(i,1)=MatrikTeman(i,1)+B;
end
A=max(MatrikTeman);

for i=1:size(MatrikTeman,1)
    MatrikTeman(i,3)= MatrikTeman(i,3)+A(1,1);
end

status=1;
for i=1:size(KoordinatRK,1)
    if status==1
        if and(KoordinatRK(i,1)~=0,KoordinatRK(i,2)~=0)==1
            KoordinatTitikJalanX=[KoordinatRK(i,6)
KoordinatRK(i,7) 3 i];

KoordinatTitikJalan=[KoordinatTitikJalan;KoordinatTitikJalanX];
            status=1;
        elseif and(KoordinatRK(i,1)==0,KoordinatRK(i,2)==0)==1
            %Sentral
            KoordinatTitikJalanX=[0 0 5 i];

KoordinatTitikJalan=[KoordinatTitikJalan;KoordinatTitikJalanX];
            status=0;
        end
    elseif status==0
        %Steiner
        KoordinatTitikJalanX=[KoordinatRK(i,6) KoordinatRK(i,7) 4
i];

KoordinatTitikJalan=[KoordinatTitikJalan;KoordinatTitikJalanX];
    end
end

```

```

end

KoordinatTitikJalan(:,4)=(1:size(KoordinatTitikJalan,1))';

LinkJalanDP=TambahLinkJalan(KoordinatTitikJalan,1,2,LinkJalan);

LinkJalanRK=TambahLinkJalan(KoordinatTitikJalan,1,3,LinkJalan);

LinkJalanSteiner=TambahLinkJalan(KoordinatTitikJalan,1,4,LinkJalan
);

for i=1:size(LinkSentral,1)
    [v,pos]=find(KoordinatTitikJalan(:,3)==5);
    LinkSentralBaru(i,:)=[KoordinatTitikJalan(v,4)
LinkSentral(i,1)];
end

LinkJalanBaru=[LinkJalan;LinkJalanDP;LinkJalanRK;LinkJalanSteiner;
Nyebrang;LinkSentralBaru];
MatrikKoneksi=[MatrikTeman(:,1) MatrikTeman(:,3)];

for i=1:size(MSUbuf,1)
    Koord=KoordinatRK(MSUbuf(i,1),6:7);
    [x,y]=find(and(KoordinatTitikJalan(:,1)==Koord(1,1),...
    KoordinatTitikJalan(:,2)==Koord(1,2)));
    if size(x,1) > 1
        x = x(1,1);
    end
    MSUXA(i,1)=KoordinatTitikJalan(x,4);

    Koord=KoordinatRK(MSUbuf(i,2),6:7);
    [x,y]=find(and(KoordinatTitikJalan(:,1)==Koord(1,1),...
    KoordinatTitikJalan(:,2)==Koord(1,2)));
    if size(x,1) > 1
        x = x(1,1);
    end
    MSUXA(i,2)=KoordinatTitikJalan(x,4);
end

v=find(KoordinatTitikJalan(:,3)==3);
for i=1:size(v,1)
    [x,y]=find(KoordinatTitikJalan(:,3)==5);
    MatrixSentralRK(i,1)=KoordinatTitikJalan(x,4);
end
MatrixSentralRK=[MatrixSentralRK v];

% MatrikKoneksiBaru=[MatrikKoneksi;MSUXA;MatrixSentralRK];
MatrikKoneksiBaru=[MatrikKoneksi;MSUXA];
% for i=1:3
MatrikKoneksiBaru(1:size(MatrikKoneksi,1),3)=1;
MatrikKoneksiBaru(size(MatrikKoneksi,1)+1:size(MatrikKoneksi,1)+si
ze(MSUXA,1),3)=2;
%
MatrikKoneksiBaru(size(MatrikKoneksi,1)+size(MSUXA,1)+1:end,3)=2;

```



```

% end
LinkJalanBaru=(1:size(LinkJalanBaru,1))' LinkJalanBaru;
KoordinatTitikJalanBaru(:,1)=(1:size(KoordinatTitikJalan,1))';
KoordinatTitikJalanBaru=[KoordinatTitikJalanBaru
KoordinatTitikJalan];
for i=1:size(MatrikKoneksiBaru,1)
    [dist,path] =
dijkstra(KoordinatTitikJalanBaru,LinkJalanBaru,MatrikKoneksiBaru(i
,1),MatrikKoneksiBaru(i,2),handles);
    axes(handles.axes1)
    if MatrikKoneksiBaru(i,3)==2
        warna='r-';tebal=2;geserx=3;gesery=7;
    else
        warna='b-';tebal=2;geserx=10; gesery=14;
    end
    if get(handles.gl,'value')==1
        for n = 2:length(path)
            plot(KoordinatTitikJalanBaru(path(n-
1:n),2)+geserx,gesery+KoordinatTitikJalanBaru(path(n-
1:n),3),warna,'linewidth',tebal);
        end
    end
    if get(handles.br,'value')==1
        Path='';
        for k=1:size(path,2)
            if k~=1
                Path=[Path ',-', num2str(path(1,k))];
            else
                Path=[num2str(path(1,k))];
            end
        end
        % MatrixReport
        % for i=1:size(MatrikKoneksiBaru,1)
        MatrikReport(i,1)={MatrikKoneksiBaru(i,1)};
        MatrikReport(i,2)={MatrikKoneksiBaru(i,2)};
        %
[x,y]=find(MatrikKoneksiBaru(i,1)==KoordinatTitikJalanBaru(:,4));
% KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,1),4)
    if KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,1),4)==3
        A='RK';
    elseif
KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,1),4)==2
        A='DP';
    elseif
KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,1),4)==4
        A='Steiner';
    elseif
KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,1),4)==5
        A='Sentral';
    end
    if KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,2),4)==3
        B='RK';
    elseif
KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,2),4)==2
        B='DP';
    elseif
KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,2),4)==4
        B='Steiner';

```

```

        elseif
KoordinatTitikJalanBaru(MatrikKoneksiBaru(i,2),4)==5
            B='Sentral';
        end
        MatrikReport(i,3)=[A '-' B];
        MatrikReport(i,4)={dist};
        MatrikReport(i,5)={Path};
    end
end
if get(handles.br,'value')==1
    MatrikReport=[{'A'} {'B'} {'Connection-ID'} {'Distance'}
{'Path'};
    MatrikReport];
[s,m]=xlswrite('Report.xls',MatrikReport,'Report');
%ID-RK dan Catuan Primer
[s,m]=xlswrite('Report.xls',NamaRK,'CatPrimer');
for i=1:size(KoordinatTitikJalanBaru,1)
    for j=1:size(KoordinatTitikJalanBaru,2)
        ReportKoordinat(i,j)={KoordinatTitikJalanBaru(i,j)};
    end
end
for i=1:size(LinkJalanBaru,1)
    for j=1:size(LinkJalanBaru,2)
        ReportLinkJalan(i,j)={LinkJalanBaru(i,j)};
    end
end
ReportKoordinat=[{'Index'} {'X'} {'Y'} {'Type'} {''};...
    ReportKoordinat];
ReportLinkJalan=[{'Index'} {'Ruas A'} {'Ruas B'} ;...
    ReportLinkJalan];

[s,m]=xlswrite('Report.xls',ReportKoordinat(:,1:4),'KoordinatBaru'
);
[s,m]=xlswrite('Report.xls',ReportLinkJalan,'LinkJalan');
end

if get(handles.gl,'value')==1
    [x,y]=find(KoordinatTitikJalanBaru(:,4)==3);
    for i=1:size(x,1)

plot(KoordinatTitikJalanBaru(x(i,1),2),KoordinatTitikJalanBaru(x(i
,1),3),'s','MarkerFaceColor','k')
        end
%     [x,y]=find(KoordinatTitikJalanBaru(:,4)==4);
%     for i=1:size(x,1)
%
plot(KoordinatTitikJalanBaru(x(i,1),2),KoordinatTitikJalanBaru(x(i
,1),3),'s','MarkerFaceColor','y')
%     end

end
% s=gca;
% h=figure(1);
%
% saveas()

return

```

## RIWAYAT PENULIS



**ANDRI KURNIA RIYADI** lahir di Surakarta 8 September 1978. Menyelesaikan sekolah di SDN Dukuh 01 Pagi Jakarta Timur tahun 1991, SMPN 24 Jakarta Timur tahun 1994, SMUN 14 Jakarta Timur tahun 1997, S1 Teknik Telekomunikasi STTTelkom Bandung tahun 2003 dan Manajemen Telekomunikasi Universitas Indonesia tahun 2008. Riwayat pekerjaan yang pernah ditekuni : Staff Entry PT. Surveyor Indonesia tahun 2003 – 2004, Staff Administrasi Kredit Bank BNI 46 Kantor Cabang Utama Mayestik tahun 2004 – 2005, dan mulai 2005 sampai dengan sekarang aktif sebagai Officer Ordering & Provisioning Telkom Divisi Carrier Interconnection Service (Telkom CIS).