

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

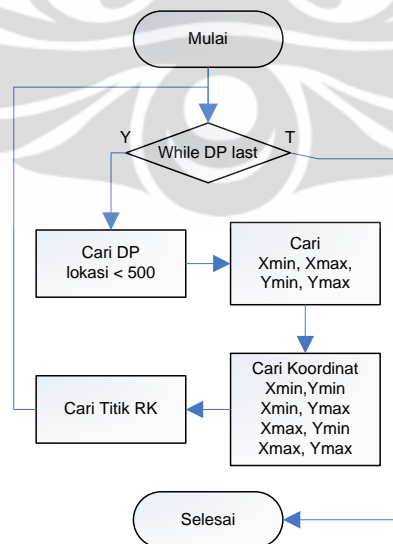
Ket :
A dan B
adalah titik
yang langsung
terhubung
pada suatu sisi
jalan

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

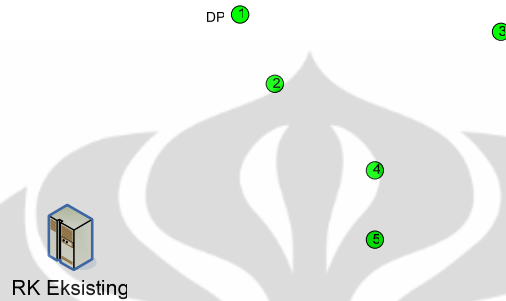
Untuk menentukan titik RK ini maka DP – DP yang berdekatan dan berjarak radius sekitar 500 meter dari RK, dapat dikelompokkan menjadi 1 grup. Spesifikasi teknis VDSL2 sendiri dapat menghantarkan *Bit rate* dalam kecepatan sekitar 50 mbps dalam 750 meter, namun demi menjaga kualitas layanan maka untuk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

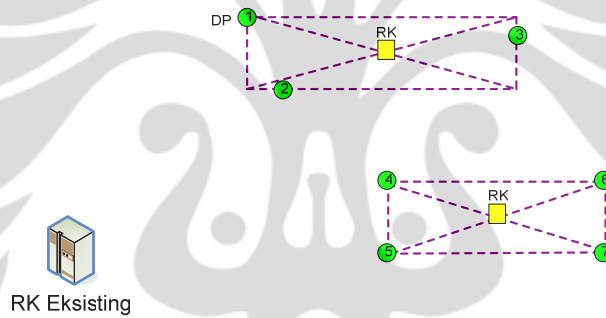


Gambar 3.1 Diagram Alir Penentuan Titik RK

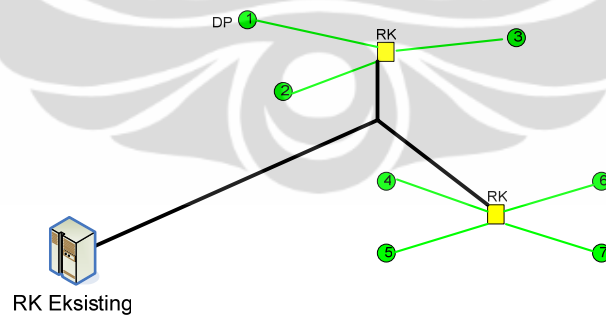
Dalam kondisi sebenarnya, maka semakin baik apabila anggota dari grup DP adalah sebanyak mungkin DP. Apabila satu buah DP secara lokasi terpisah jauh dari DP lainnya, maka tidak ada cara lain selain membangun RK yang terdedikasi kepada DP tersebut.



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

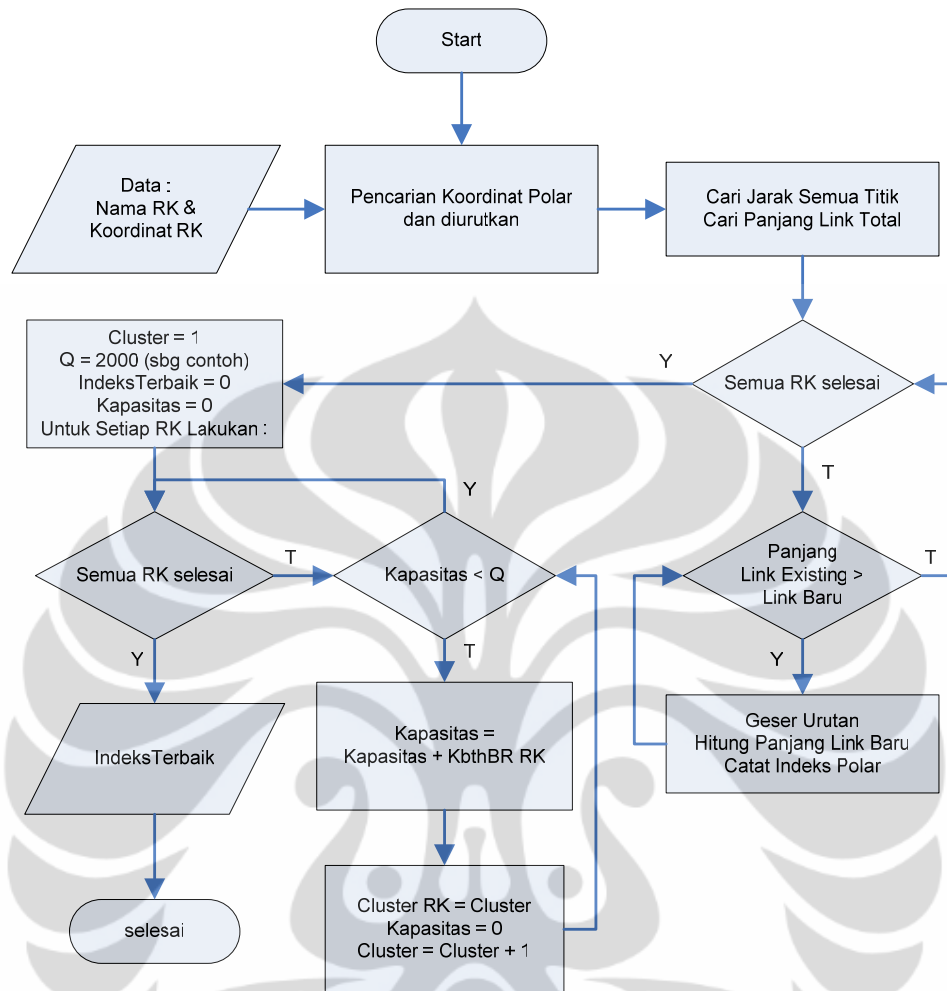
Apabila pengelompokkan RK baru – DP ini sudah terdefinisi, maka hubungan RK baru – DP ini tinggal dicari jalurnya berdasarkan hubungan titik ke titik. Untuk hal tersebut maka diperlukan algoritma djikstra untuk menyelesaikannya.

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.

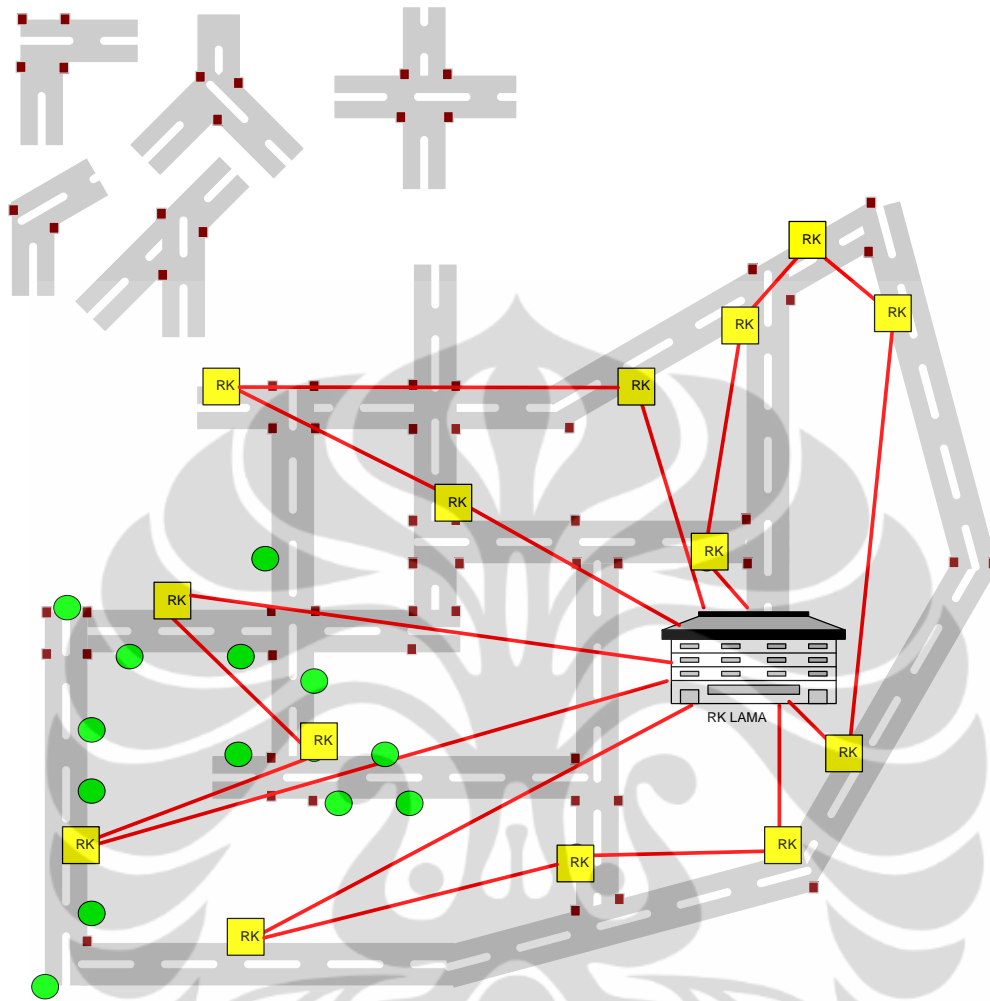
Hasil dari Pencarian Scatter adalah Indeks Terbaik, dimana Indeks Terbaik akan berisi kumpulan titik – titik RK baru yang dikelompokkan satu sama lain berdasarkan kedekatan sudut polar dari titik – titik RK. Indeks Terbaik juga dapat menggambarkan suatu bentuk grafik seperti kelopak bunga dimana pusat 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.

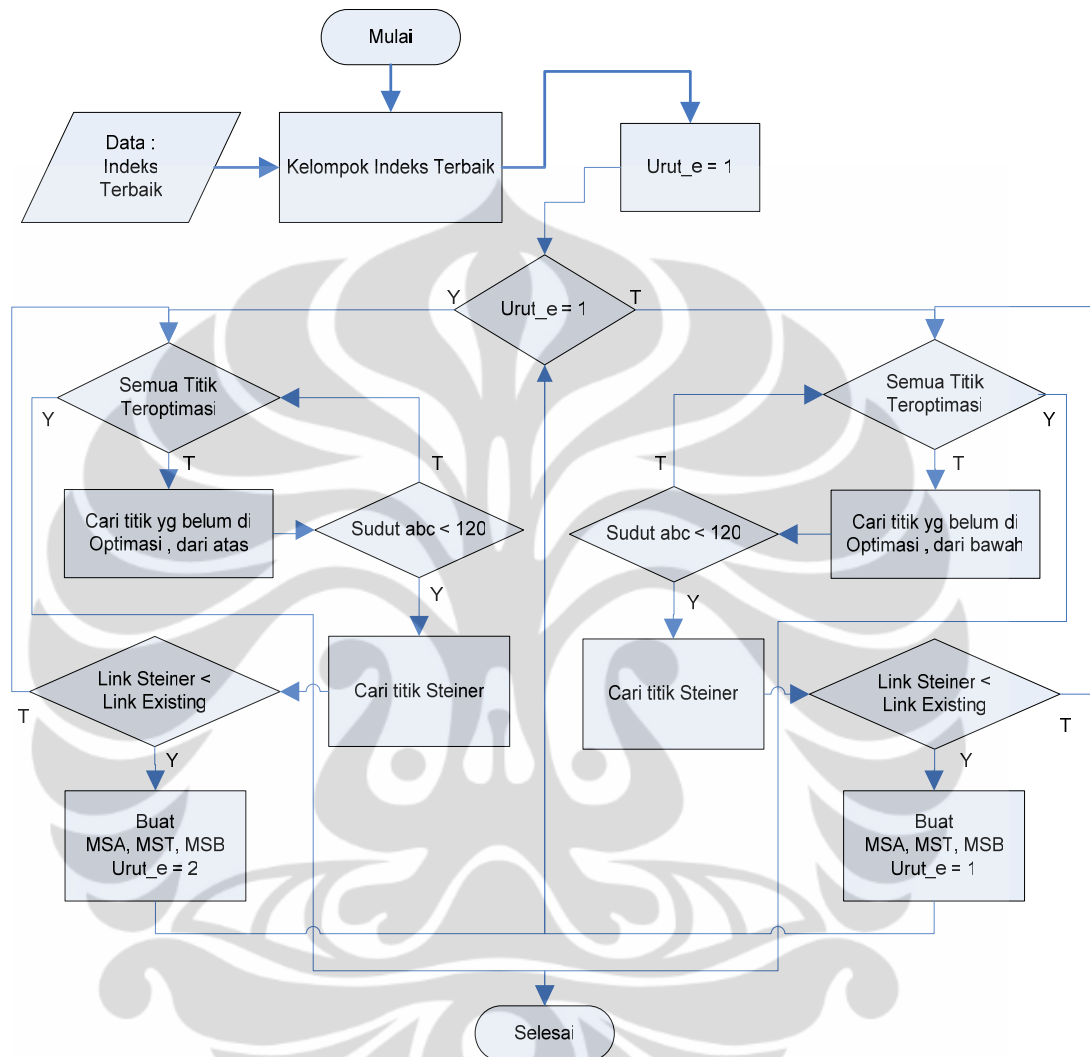


Gambar 3.6 Gambar Pengelompokan Titik Sesuai Sudut Polar RK

3.4 POHON STEINER

Algoritma Pohon Steiner berfungsi untuk mencari pencabangan yang efektif berdasarkan pengelompokan 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

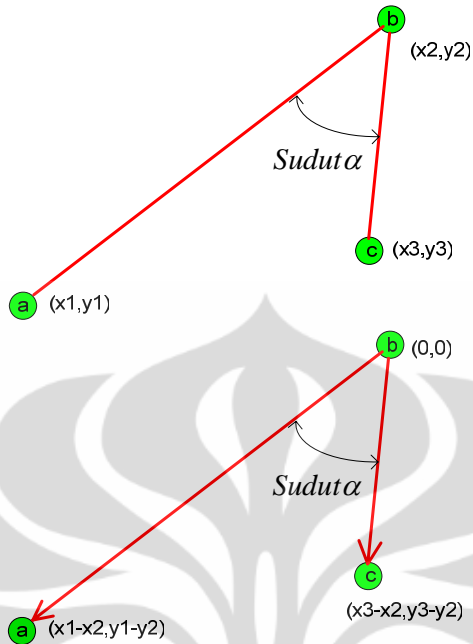
panjang link yang lebih pendek, maka solusi itu diterima, apabila tidak maka aplikasi berlanjut mencari 3 titik lainnya untuk dicoba dicari solusi steinernya.



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 :



Gambar 3.8 Menghitung sudut pada 3 titik

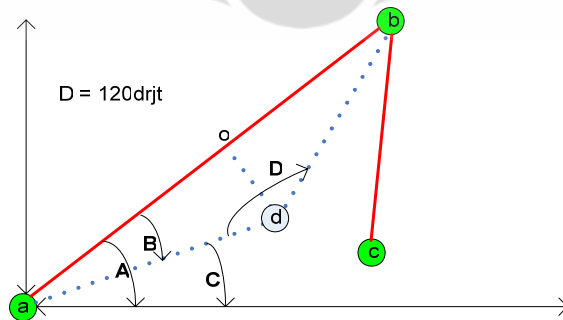
Cara menghitung sudut yang mengapit dua vektor :

$$\cos \alpha = \frac{a \cdot c}{|a| |c|} \dots \dots \dots (3.13)$$

$$\cos \alpha = \frac{((x1-x2)(x3-x2)) + ((y1-y2)(y3-y2))}{\sqrt{((x1-x2)^2 + (y1-y2)^2)} \sqrt{((x3-x2)^2 + (y3-y2)^2)}} \dots \dots \dots (3.14)$$

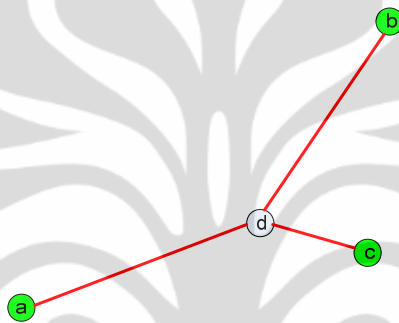
3.4.2 Menentukan Titik Steiner

Titik Steiner akan menjadi usulan solusi bagi algoritma pohon steiner, apabila solusi steiner lebih kecil membutuhkan panjang link maka akan dijadikan solusi yang valid dan diterima.



Gambar 3.9 Menentukan titik steiner "d"

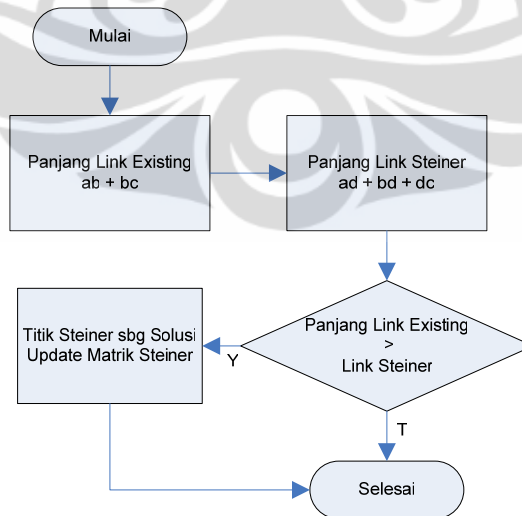
Untuk menentukan apakah diantara 3 titik tersebut dapat dilakukan proses selanjutnya yaitu pencarian titik steiner, cos sudut adalah indikasinya. Apabila cos sudut < 120 derajat maka akan dicari letak titik bantu steiner. Pertama kali titik a akan menjadi pusat sumbu, dan link ab dicari koordinat polarnya.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

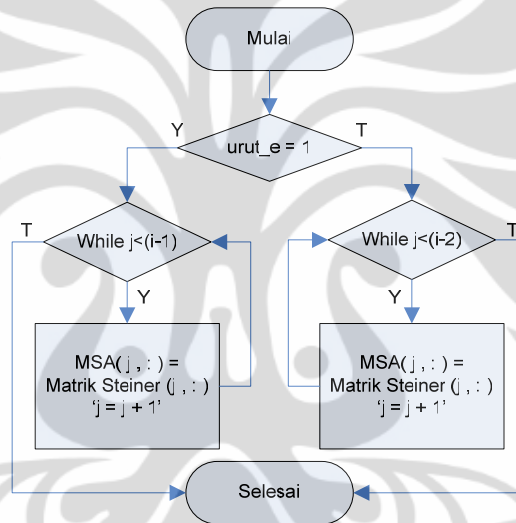


Gambar 3.11 Solusi Steiner pada badan utama aplikasi

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 :



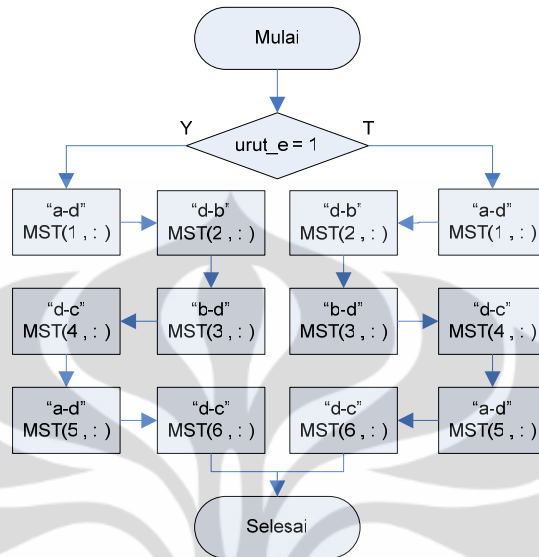
Gambar 3.12 Diagram Alir Matrik Steiner Atas

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.

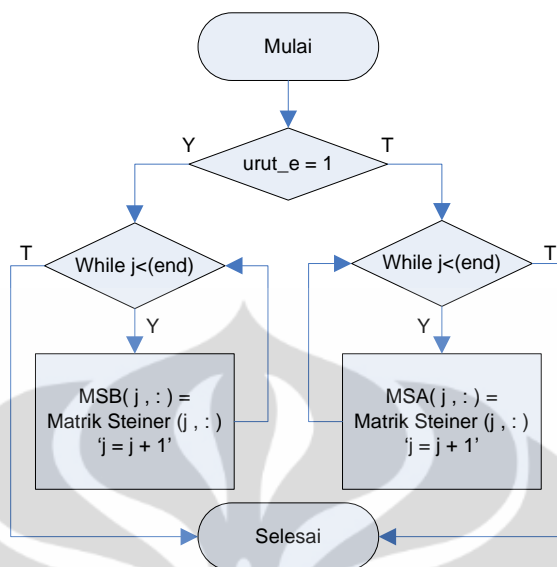


Gambar 3.13 Diagram Alir Matrik Steiner Tengah

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 dibaris 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.

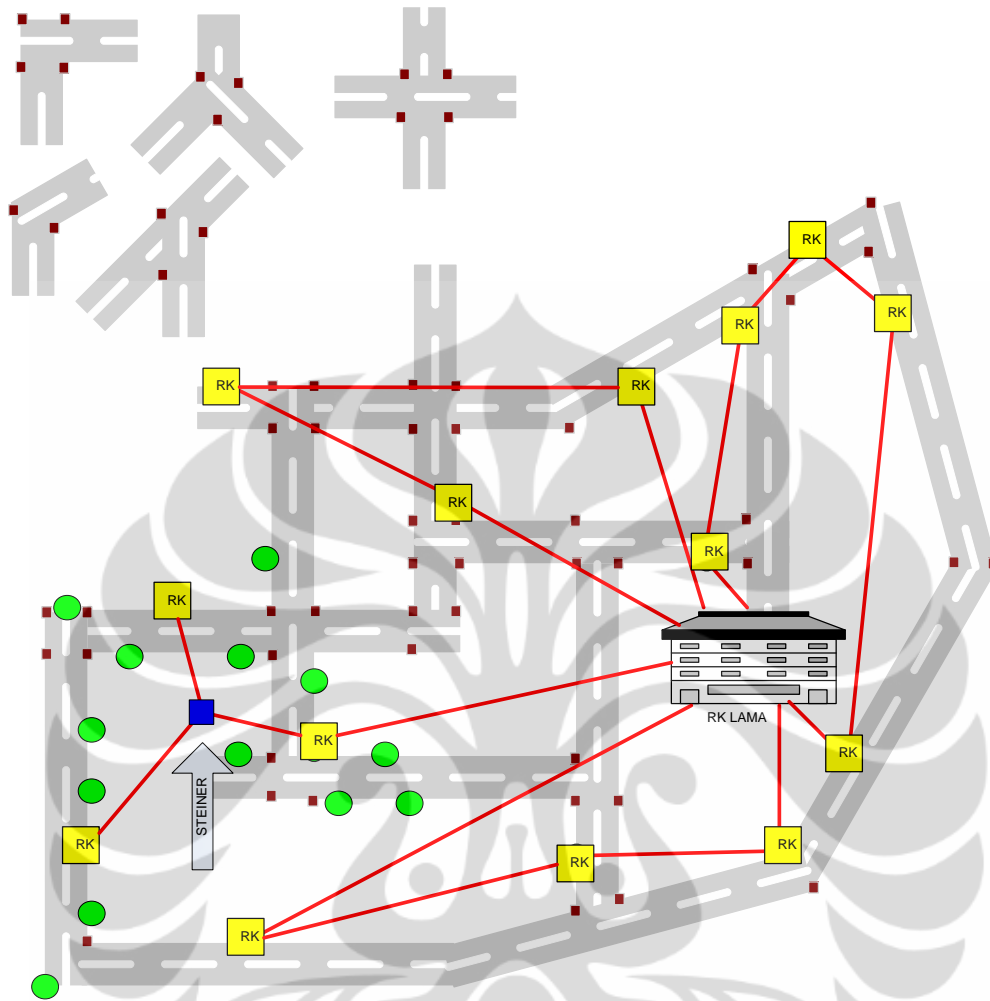
Tabel 3.4 Solusi Steiner Pada Matrik Steiner

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.

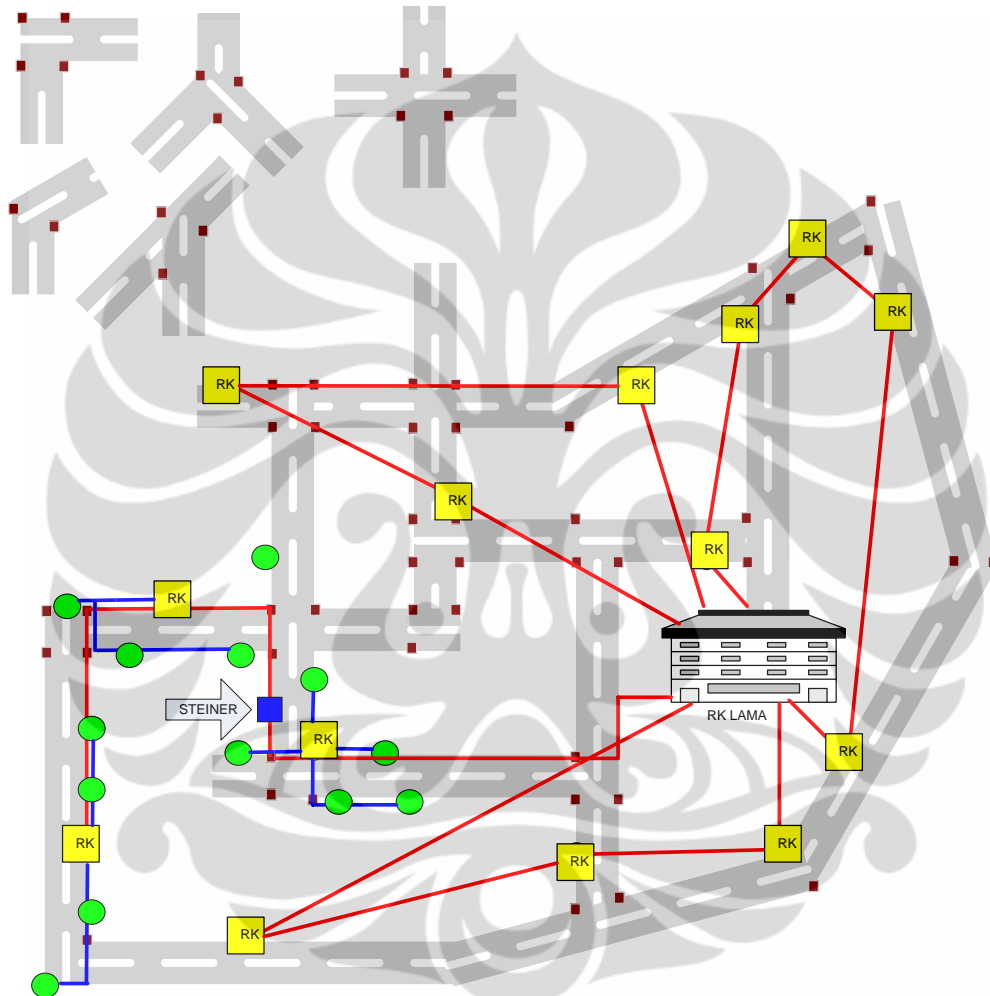


Gambar 3.15 Pohon Steiner pada jaringan

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.