BAB III
PEMODELAN OPTIMASI ROUTING OSPF MENGGUNAKAN MODEL MPLS TRAFFIC ENGINEERING

III.1. Analisis Optimasi

Analisis optimasi yang dilakukan pada penelitian ini secara sistematis akan memberikan gambaran awal bagaimana suatu protokol *routing* OSPF bekerja sebelum menerapkan teknologi *traffic engineering* untuk optimasi protokol *routing* OSPF pada jaringan komputer menggunakan model multi protokol label *switching* (MPLS). Tahap selanjutnya adalah membuat rancangan topologi dengan protokol *routing* OSPF dan memodelkan *traffic engineering* pada model jaringan MPLS pada simulator jaringan untuk dilakukan analisis dan penilaian terhadap faktor-faktor terkait optimasi protokol *routing* OSPF pada jaringan internet.

III.1.2. Analisis Masalah

Protokol *routing* OSPF merupakan protokol *routing* yang menggunakan mekanisme *routing link state*, kinerja protokol *routing* OSPF ditentukan pada konsumsi *bandwidth* dan kebutuhan konsumsi memori yang besar. Protokol *routing* OSPF ini akan menentukan dan memilih prioritas jalur utama untuk melewatkan paket data berdasarkan pada jarak terpendek dan berdasarkan pada ukuran kanal *bandwidth* yang paling besar tanpa membandingkan adanya perubahan kondisi jalur lain dengan tingkat kepadatan lalu lintas komunikasi data yang relatif rendah. Semakin tinggi tingkat kepadatan lalu lintas komunikasi data pada waktu yang bersamaan dalam satu jalur, maka akan berakibat semakin tinggi pula resiko terjadinya kegagalan hubungan komunikasi data pada jalur tersebut yang menyebabkan informasi yang dikirimkan saat proses komunikasi data tidak akan sampai ke tujuan karena terjadi penurunan kapasitas *bandwidth* akibat adanya penumpukan data di satu jalur lalu lintas jaringan.

III.2. Analisis Kinerja Protokol Routing OSPF

Protokol *routing* OSPF (*Open Shortest Path First*) merupakan teknologi perutean pada jaringan komputer yang mendistribusikan informasi-informasi jalur pada tabel *routing* di setiap *router*. OSPF dikembangkan menggunakan algoritma dijkstra (*shortest path*) untuk mempelajari berbagai rute dan untuk memilih salah satu jalur menuju rute tujuan. OSPF menggunakan mekanisme *routing link state* dalam menentukan dan mencari jalur komunikasi data. OSPF akan melakukan perhitungan terhadap jalur yang akan dilalui oleh suatu paket data dan akan membentuk suatu tabel *routing* yang berisi informasi-informasi jalur yang akan digunakan dalam proses komunikasi data. Perhitungan yang akan dilakukan oleh protokol OSPF terjadi apabila terdapat perubahan jalur pada topologi jaringan, semakin sering melakukan perhitungan jalur pada informasi tabel *routing* yang dilakukan oleh OSPF memungkinkan terjadinya penurunan trafik data pada suatu jaringan komputer karena konsumsi *bandwidth* yang cukup besar untuk melakukan perhitungan jalur dan mendistribusikan informasi jalur pada tabel routing. Masalah yang akan dihadapi selanjutnya oleh *router* yang menggunakan mekanisme *link state* pada saat melakukan perhitungan jalur untuk membentuk suatu informasi tabel *routing* adalah masalah penggunaan sumber daya memori dan prosesor yang membutuhkan kapasitas lebih besar. Tahapan analisis terhadap kinerja protokol *routing* OSPF diperlukan untuk melakukan penilaian awal terhadap kinerja protokol *routing* OSPF serta untuk mengetahui parameter-parameter apa saja yang dibutuhkan untuk selanjutnya dilakukan proses optimasi.

III.2.1. Analisis Algoritma Dijkstra

Algoritma dijkstra dikembangkan menggunakan strategi *greedy* untuk memecahkan permasalahan jarak terpendek (*shortest path*) dalam sejumlah langkah. Strategi *greedy* pada algoritma dijkstra menyatakan setiap langkah yang dipilih memiliki sisi yang berbobot minimum dan memasukkan ke dalam himpunan solusi. Algoritma dijkstra dinyatakan dalam notasi *pseudo code* berikut:

```
procedure Dijkstra (input m: matriks, a: simpul awal)
{mencari lintasan terpendek dari simpul awal a ke semua simpul
lainnya, masukan : matriks tetangga (m) dari graf berbobot G dan
```

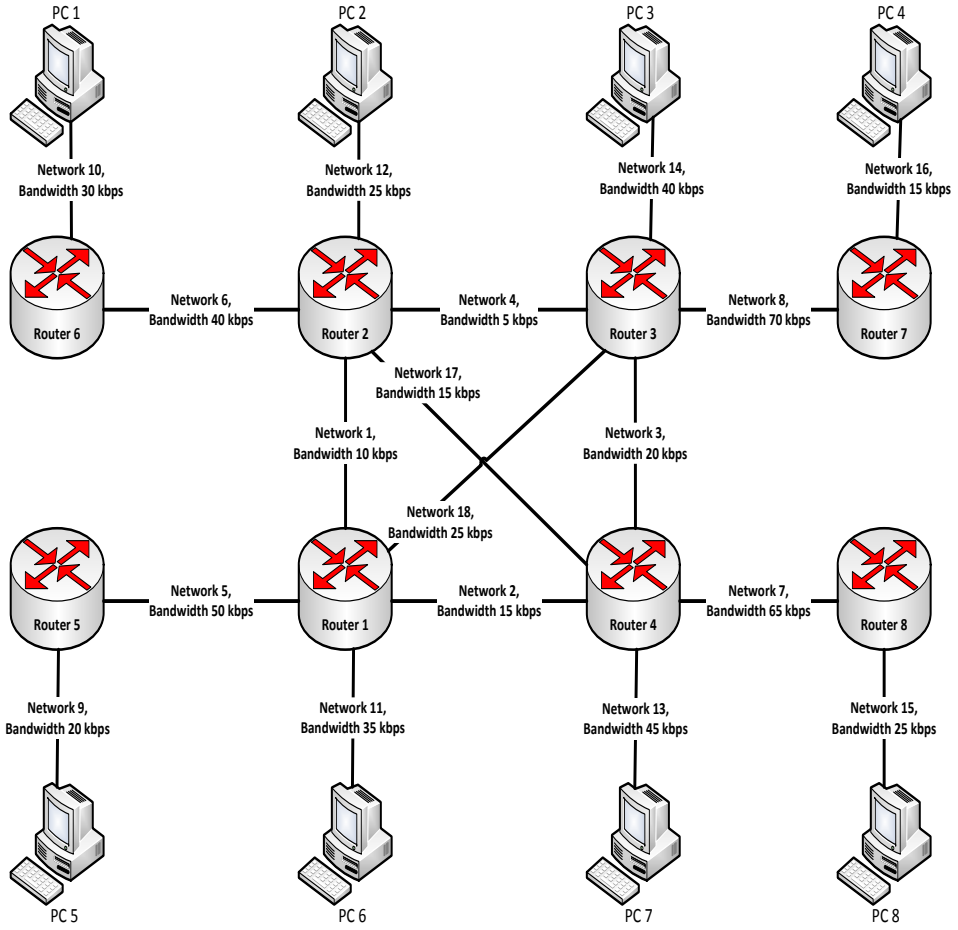
```
simpul awal a,
keluaran : lintasan terpendek dari a ke simpul lainnya}
Deklarasi
s1, s2, ..., sn : integer {array integer}
d1, d2, ..., dn : integer {array integer}
i, j : integer
Algoritma
{iterasi 0 (inisialisasi)}
for i ← 1 to n do
  si ← 0
  di ← mai
endfor
{iterasi 1}
sa ← 1
da ← ∞
{iterasi 2, 3, ..., n-1}
for i ← 2 to n-1 do
j ← sj = 0
sj ← 1
 for si = 0 do
  if dj + mji < di then
   di ← dj + mji
  endif
 endfor
endfor
```

Algoritma dijkstra dapat menyelesaikan beberapa kasus terkait pencarian seperti lintasan terpendek antara dua buah simpul, pencarian lintasan terpendek dari simpul tertentu ke simpul lainnya, serta pencarian lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu. Algoritma dijkstra yang dikembangkan pada protokol *routing* OSPF menerapkan mekanisme *link state* dalam melakukan perhitungan jarak jalur dan pencarian lintasan terpendek dari simpul tertentu ke simpul lainnya berdasarkan kondisi *link* jalur dengan tingkat kanal *bandwidth* yang tinggi untuk membentuk informasi pada tabel *routing* di setiap *router* untuk menangani lalu lintas data pada jaringan komputer.

III.2.2. Analisis Mekanisme Algoritma Dijkstra dalam Jaringan Komputer

Tahapan ini akan dilakukan analisis untuk mengetahui suatu mekanisme algoritma dijkstra yang diterapkan pada protokol *routing* OSPF dalam melakukan pendistribusian paket data dan pemilihan jalur pada suatu jaringan komputer. Proses analisis yang dilakukan pada tahapan ini akan dijelaskan secara teori tahapan-tahapan mulai dari proses iterasi awal sampai iterasi terakhir bagaimana

mekanisme algoritma dijkstra menentukan jalur untuk melewatkan suatu paket data dari pengirim menuju penerima dalam suatu jaringan komputer. Contoh model topologi jaringan yang akan digunakan untuk analisis mekanisme algoritma dijkstra pada jaringan komputer pada penelitian ini menggunakan 8 buah (*node*) *router* yang masing-masing *router* tersebut saling terhubung dengan 18 jalur (*link network*). Contoh dari model topologi jaringan komputer dapat dilihat pada Gambar III-1.



Gambar III-1 Contoh Model Topologi Jaringan

Keterangan untuk alokasi segmentasi alamat IP serta alokasi *bandwidth* yang digunakan pada topologi jaringan yang digunakan untuk analisis mekanisme algoritma dijkstra dari Gambar III-1 di atas, dapat dilihat pada Tabel III-1.

Tabel III-1 Segmentasi Jaringan

Network	IP Address	Bandwidth
Network 1	10.10.1.0/24	Bandwidth 60 kbps

Network	IP Address	Bandwidth
Network 2	10.10.2.0/24	Bandwidth 80 kbps
Network 3	10.10.3.0/24	Bandwidth 70 kbps
Network 4	10.10.4.0/24	Bandwidth 40 kbps
Network 5	10.10.5.0/24	Bandwidth 50 kbps
Network 6	10.10.6.0/24	Bandwidth 40 kbps
Network 7	10.10.7.0/24	Bandwidth 65 kbps
Network 8	10.10.8.0/24	Bandwidth 70 kbps
Network 9	10.10.9.0/24	Bandwidth 20 kbps
Network 10	10.10.10.0/24	Bandwidth 30 kbps
Network 11	10.10.11.0/24	Bandwidth 35 kbps
Network 12	10.10.12.0/24	Bandwidth 25 kbps
Network 13	10.10.13.0/24	Bandwidth 45 kbps
Network 14	10.10.14.0/24	Bandwidth 40 kbps
Network 15	10.10.15.0/24	Bandwidth 25 kbps
Network 16	10.10.16.0/24	Bandwidth 15 kbps
Network 17	10.10.17.0/24	Bandwidth 50 kbps
Network 18	10.10.18.0/24	Bandwidth 85 kbps

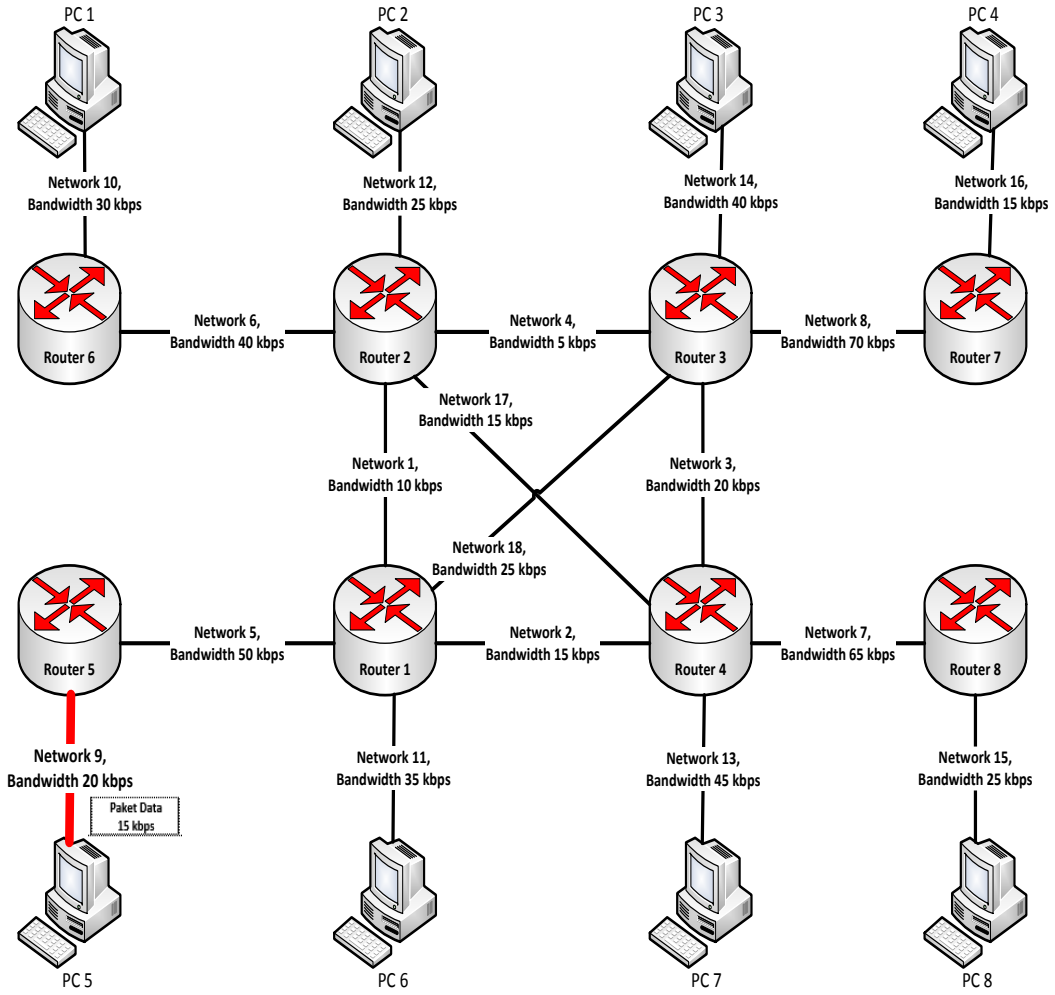
Tahapan selanjutnya akan dibuat beberapa skenario suatu proses komunikasi data yang akan digunakan untuk mengetahui bagaimana mekanisme algoritma dijkstra pada suatu protokol *routing* OSPF ketika melakukan pendistribusian suatu paket data saat proses komunikasi pada jaringan komputer berlangsung.

Skenario pertama akan dilakukan analisis untuk mengetahui bagaimana suatu algoritma dijkstra pada protokol *routing* OSPF melakukan perhitungan jalur untuk membentuk informasi tabel *routing*, serta bagaimana kondisi beban data pada jalur yang di lewati untuk melewatkan paket data ketika ada paket data sebesar 15 kbps yang dikirimkan dari PC 5 yang terdapat pada *network* 9 yang terhubung dengan *router* 5 menuju PC 4 yang terdapat pada *network* 16 yang terhubung dengan *router* 7. Proses perhitungan yang dilakukan pada skenario pertama secara bertahap sebagai berikut:

1. Perhitungan pada Iterasi ke 0

Proses perhitungan pertama yang dilakukan oleh algoritma dijkstra untuk membentuk suatu informasi jalur pada tabel *routing* di iterasi ke 0 yaitu paket data sebesar 15 kbps akan di lewatkan menuju *router* 5 melalui *network* 9 yang memiliki kapasitas maksimum jalur *bandwidth* untuk melewatkan paket sebesar

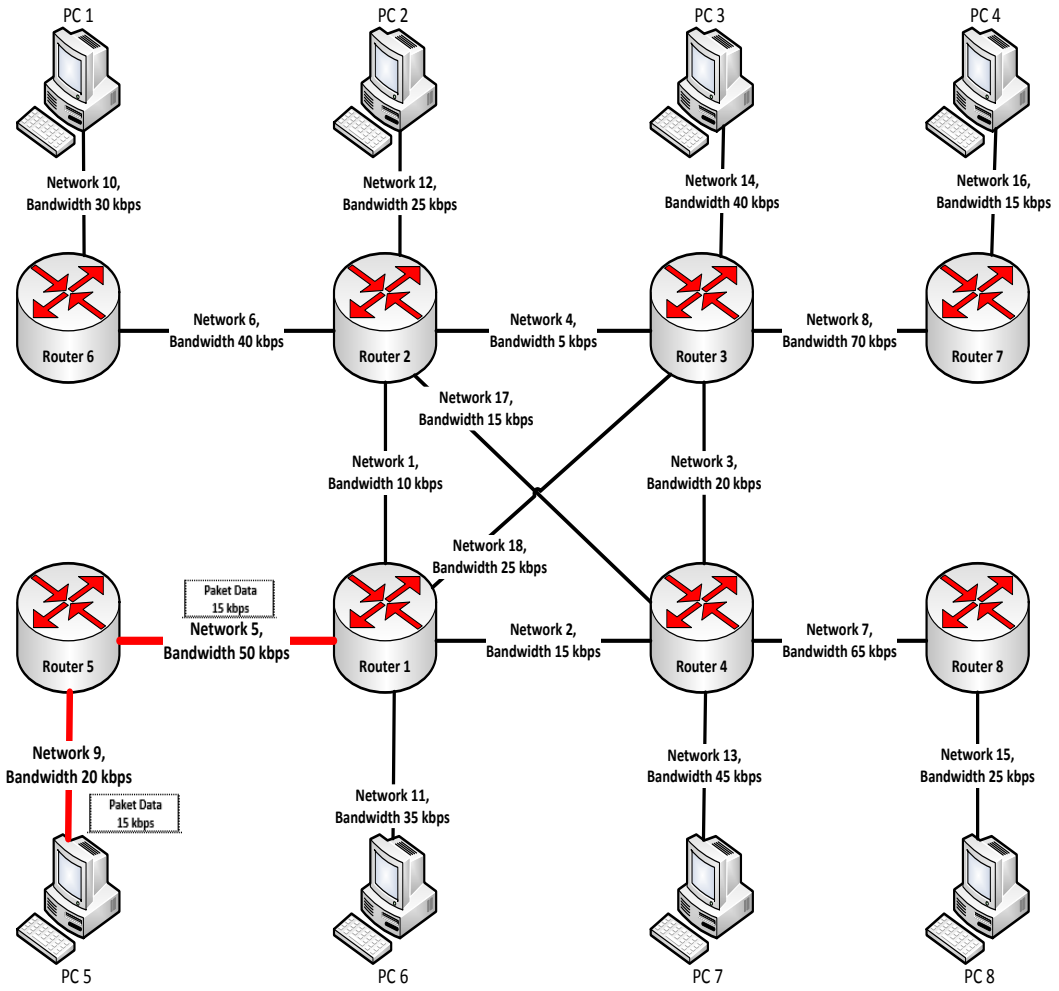
20 kbps. Proses perhitungan pada iterasi ke 0, paket data dengan dengan ukuran 15 kbps akan langsung dialirkan melalui *network* 9 karena ukuran *bandwidth* pada *network* 9 dapat mengalirkan paket data yang berukuran sebesar 15 kbps. Ilustrasi skenario pada iterasi ke 0 dapat dilihat pada Gambar III-2.



Gambar III-2 Proses Perhitungan pada iterasi ke 0

2. Perhitungan pada Iterasi ke 1

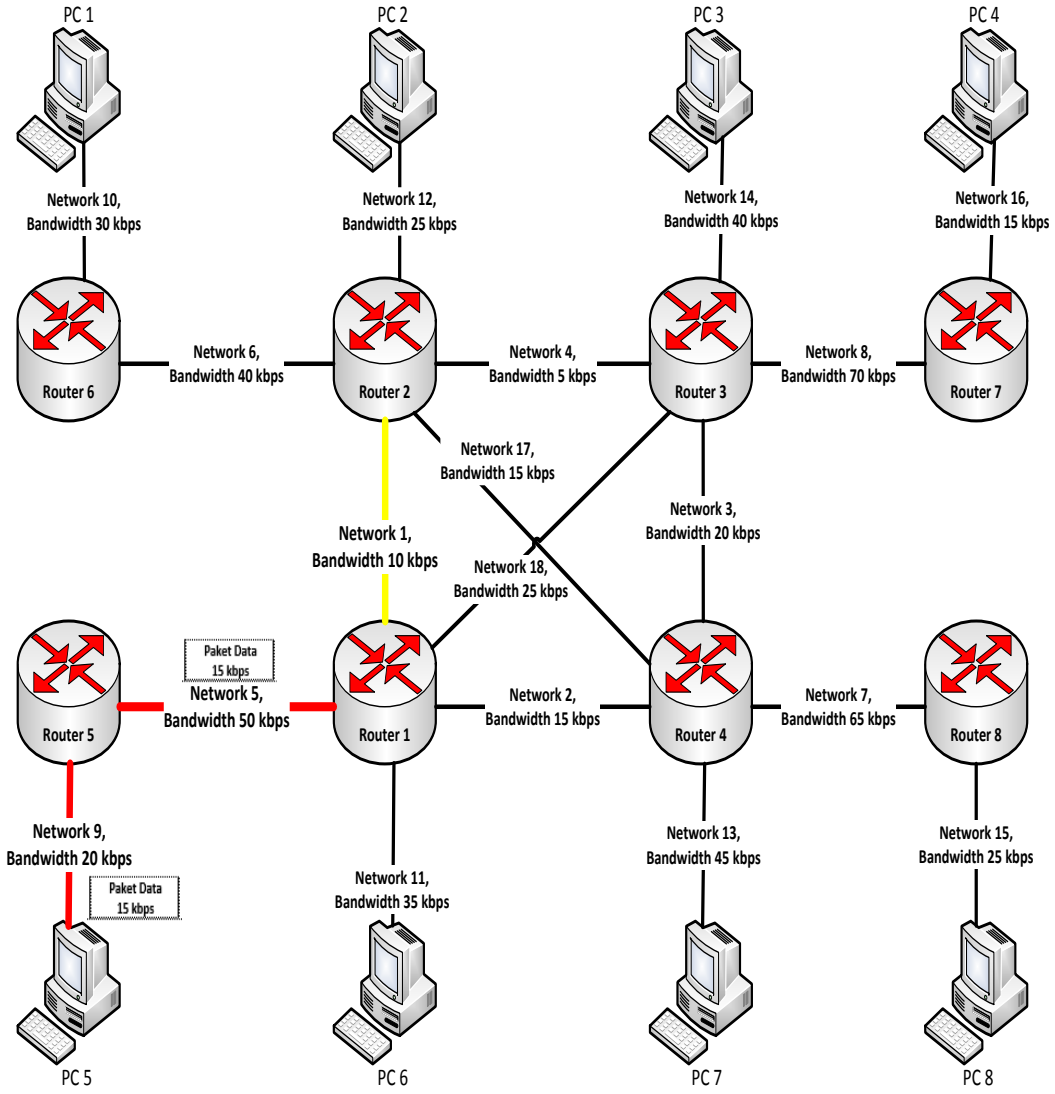
Proses perhitungan pada iterasi ke 1 selanjutnya paket data sebesar 15 kbps yang dikirimkan menuju *router* 5 akan diteruskan menuju *router* 1 melalui *network* 5 yang memiliki kapasitas jalur *bandwidth* dengan ukuran 50 kbps untuk melewatkan paket data, ilustrasi skenario pada iterasi ke 1 dapat dilihat pada Gambar III-3.



Gambar III-3 Proses Perhitungan pada iterasi ke 1

3. Perhitungan pada Iterasi ke 2

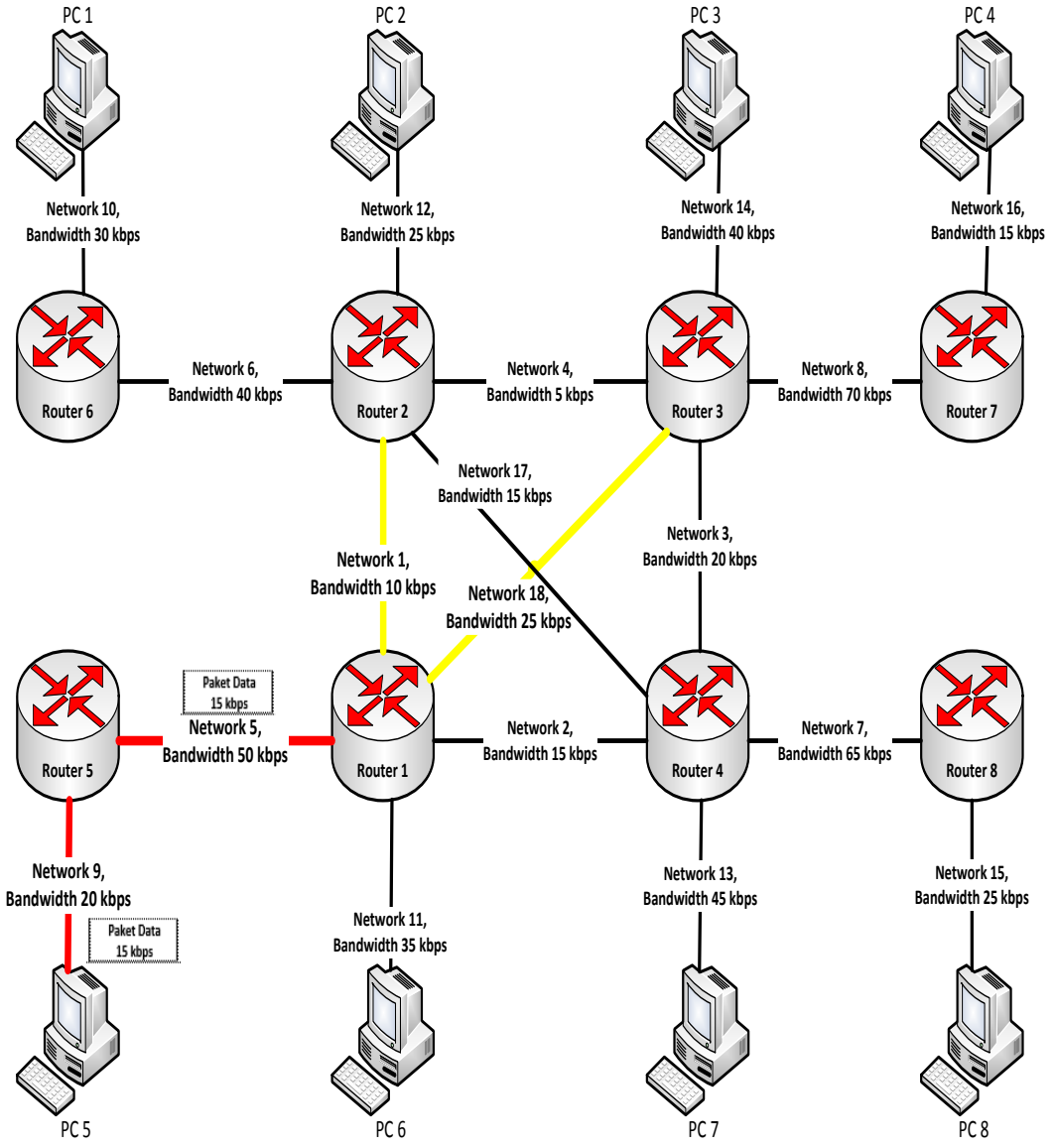
Selanjutnya pada proses perhitungan di iterasi ke 2 paket data dengan ukuran sebesar 15 kbps tidak akan langsung dikirimkan menuju rute selanjutnya sebelum diperoleh suatu informasi jalur data yang akan digunakan untuk mengalirkan data tersebut. *Router* 1 akan terlebih dahulu melakukan proses *broadcast information* berupa pengecekan jalur di *network* 1 pada *level data link* di lapisan OSI yang terhubung dengan *router* 2 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 2 dapat dilihat pada Gambar III-4, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-4 Proses Perhitungan pada iterasi ke 2

4. Perhitungan pada Iterasi ke 3

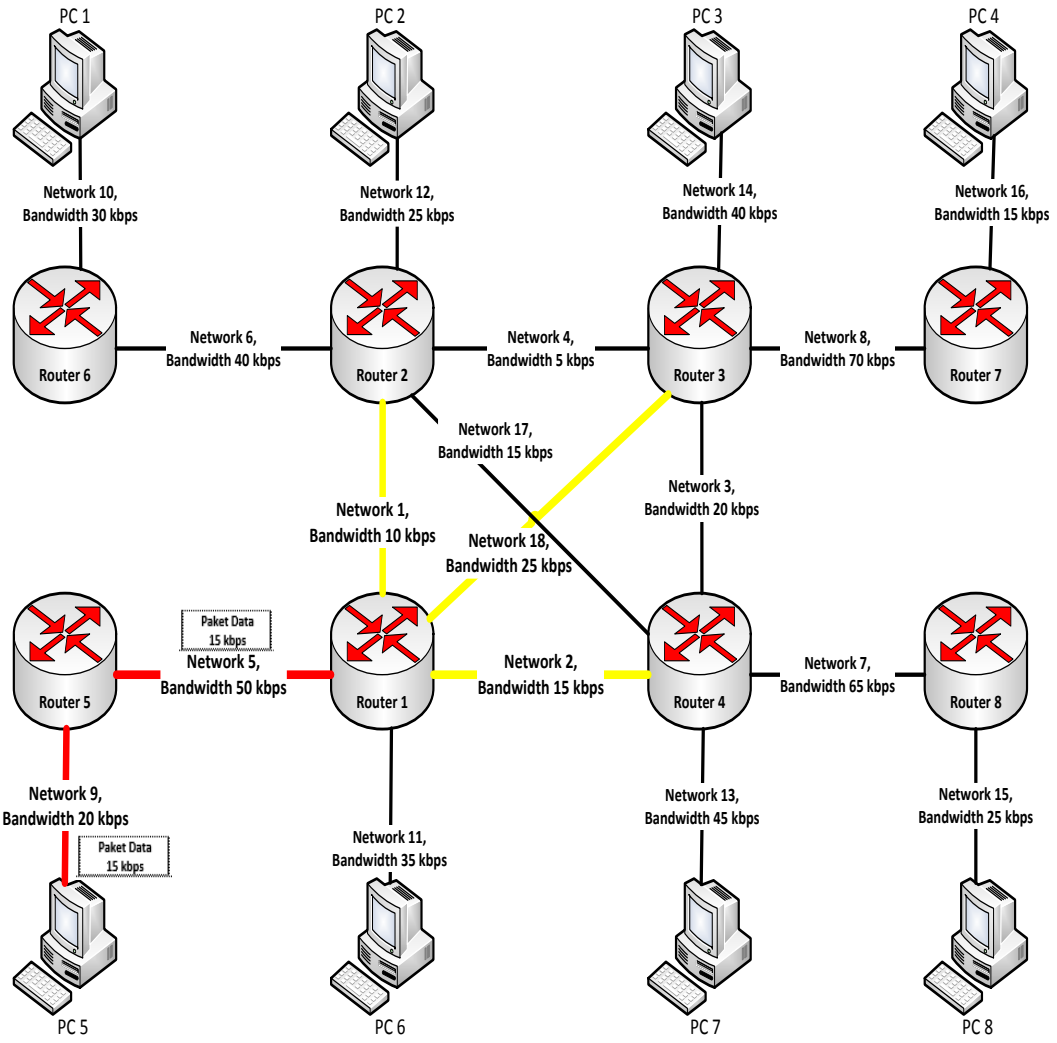
Pada proses perhitungan di iterasi ke 3 sama seperti pada proses perhitungan di iterasi ke 2, *router* 1 akan melakukan *broadcast information* berupa pengecekan jalur di *network* 18 pada *level data link* di lapisan OSI yang terhubung dengan *router* 3 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 3 dapat dilihat pada Gambar III-5, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-5 Proses Perhitungan pada iterasi ke 3

5. Perhitungan pada Iterasi ke 4

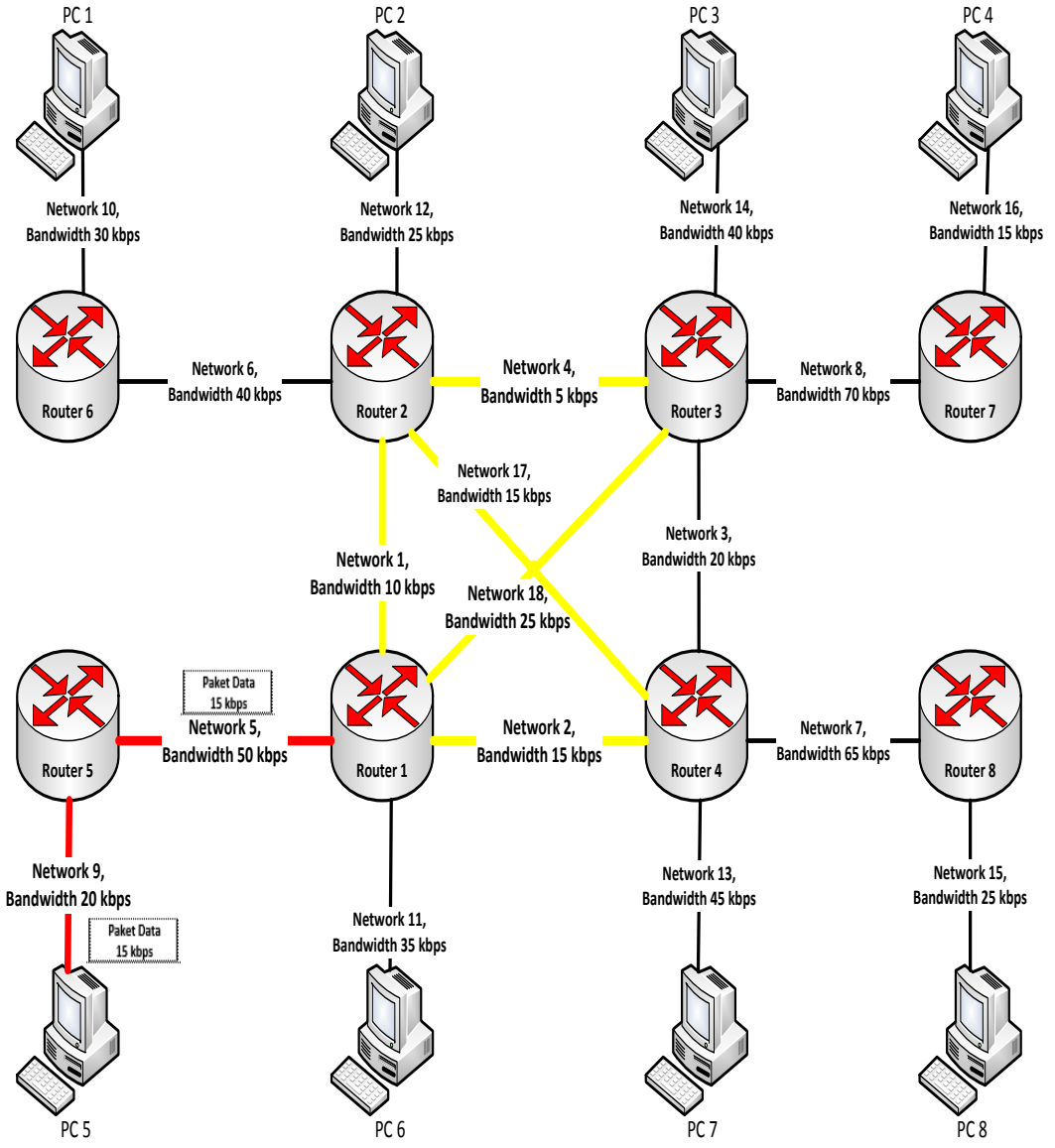
Proses perhitungan pada iterasi ke 4 sama seperti pada proses iterasi ke 2 dan iterasi ke 3, *router* 1 akan melakukan *broadcast information* berupa pengecekan jalur di *network* 2 pada *level data link* di lapisan OSI yang terhubung dengan *router* 4 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 4 dapat dilihat pada Gambar III-6, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-6 Proses Perhitungan pada iterasi ke 4

6. Perhitungan pada Iterasi ke 5

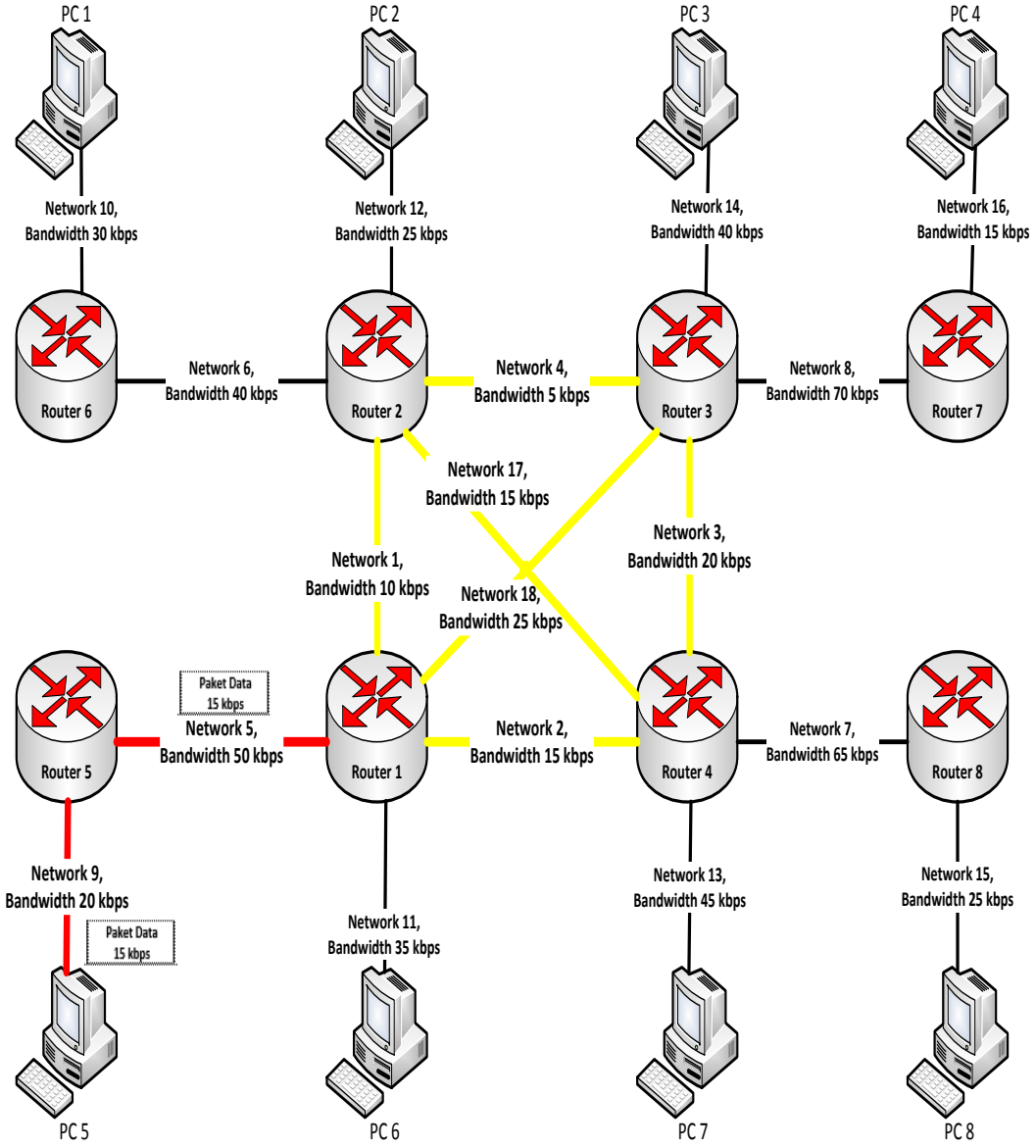
Pada proses perhitungan di iterasi ke 5, setelah *router* 1 melakukan *broadcast information* ke semua *link* yang terhubung dengan *router* 2, *router* 3, dan *router* 4. Proses selanjutnya pada *router* 2 melakukan proses *broadcast information* di *network* 4 yang terhubung dengan *router* 3 dan *network* 17 yang terhubung dengan *router* 4 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 5 dapat dilihat pada Gambar III-7, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-7 Proses Perhitungan pada iterasi ke 5

7. Perhitungan pada Iterasi ke 6

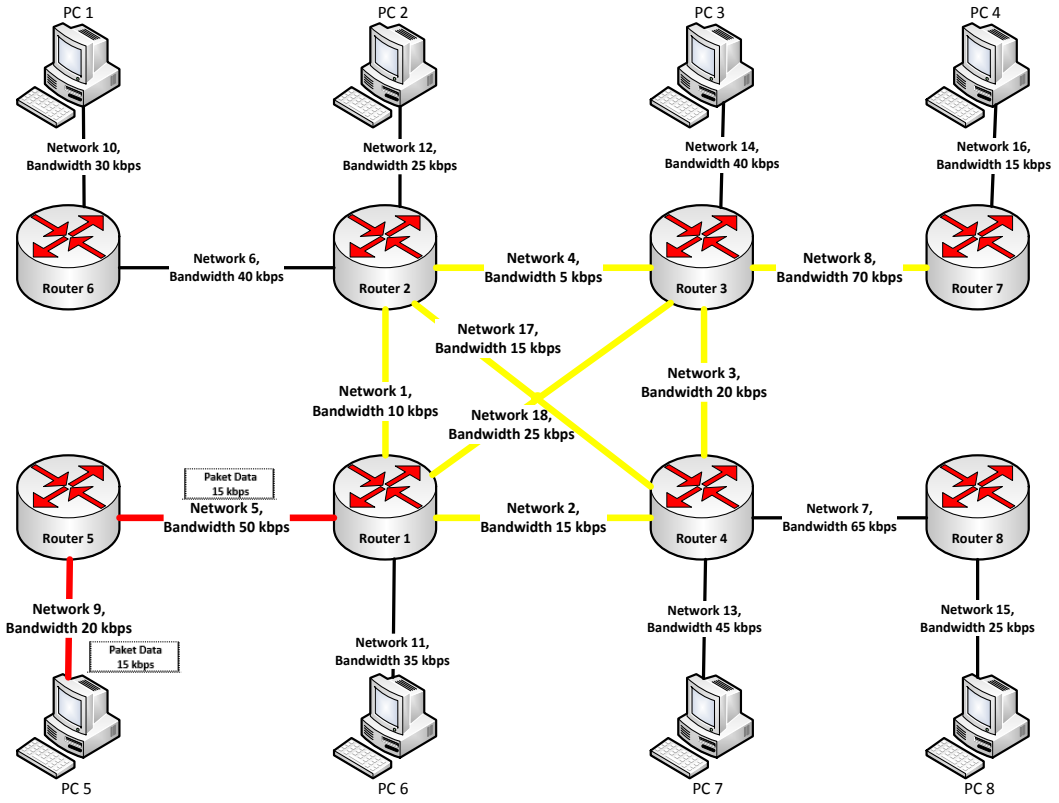
Pada proses perhitungan yang dilakukan di iterasi ke 6 sama seperti pada proses di iterasi ke 5, *router* 4 melakukan proses *broadcast information* pada *network* 17 yang terhubung dengan *router* 2 dan pada *network* 3 yang terhubung dengan *router* 3 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi dari skenario tersebut dapat dilihat pada Gambar III-8, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-8 Proses Perhitungan pada iterasi ke 6

8. Perhitungan pada Iterasi ke 7

Proses pada iterasi ke 7 selanjutnya proses *broadcast information* akan dilakukan oleh *router* 3 pada jalur *network* 8 yang terhubung dengan *router* 7 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 5 dapat dilihat pada Gambar III-9, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-9 Proses Perhitungan pada Iterasi ke 7

9. Perhitungan pada Iterasi ke 8

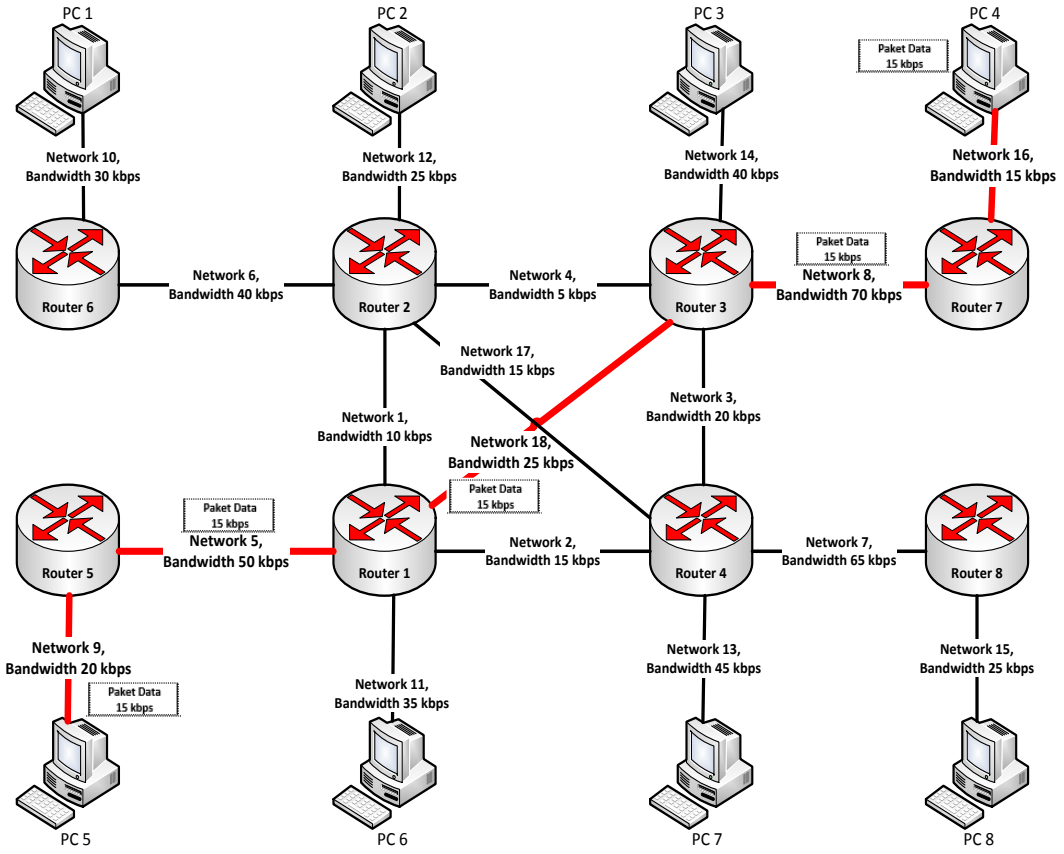
Pada proses perhitungan di iterasi ke 8 setelah dilakukan proses *broadcast information*, selanjutnya akan dilakukan perhitungan untuk menentukan jumlah jarak (hop) dari seluruh jalur yang dimungkinkan untuk dilewati oleh paket data yang dikirim dari PC 5 menuju tujuan terakhir. Proses perhitungan ini dilakukan oleh algoritma dijkstra untuk menentukan rute mana yang akan dilalui oleh paket data dengan ketentuan jumlah jarak paling pendek untuk dilalui dan memiliki kualitas jalur *bandwidth* paling baik. Proses perhitungan ditabulasikan pada Tabel III-2 di bawah ini.

Tabel III-2 Perhitungan Jumlah Jarak dan Bandwidth

Jalur	Jumlah Jarak (Hop)	Jumlah *Bandwidth*
PC 5 → router 5 → router 1 → router 2 → router 3 → router 7 → PC 4	6	170 kbps
PC 5 → router 5 → router 1 → router 3 → router 7 → PC 4	5	180 kbps

Jalur	Jumlah Jarak (Hop)	Jumlah *Bandwidth*
PC 5 → router 5 → router 1 → router 4 → router 3 → router → PC 4	6	190 kbps

Penentuan jalur akhir yang digunakan untuk melewatkan paket data dari PC 5 menuju PC 4 pada skenario pertama yang dilakukan oleh algoritma dijkstra pada protokol *routing* OSPF dalam melakukan perhitungan untuk menentukan jalur dan membentuk informasi berupa tabel *routing* berdasarkan mekanisme *link state* dapat dilihat pada Gambar III-10.



Gambar III-10 Hasil akhir perhitungan jalur yang digunakan

Hasil perhitungan akhir berdasarkan mekanisme teori yang dilakukan oleh algoritma dijkstra pada suatu protokol *routing* OSPF untuk pemilihan jalur terpendek dari PC 5 menuju PC 4 pada skenario analisis pertama dalam memilih

jalur untuk melewatkan paket data di setiap *router* ditabulasikan pada Tabel III- di bawah ini.

Tabel III-9 Perhitungan jalur pada routing OSPF

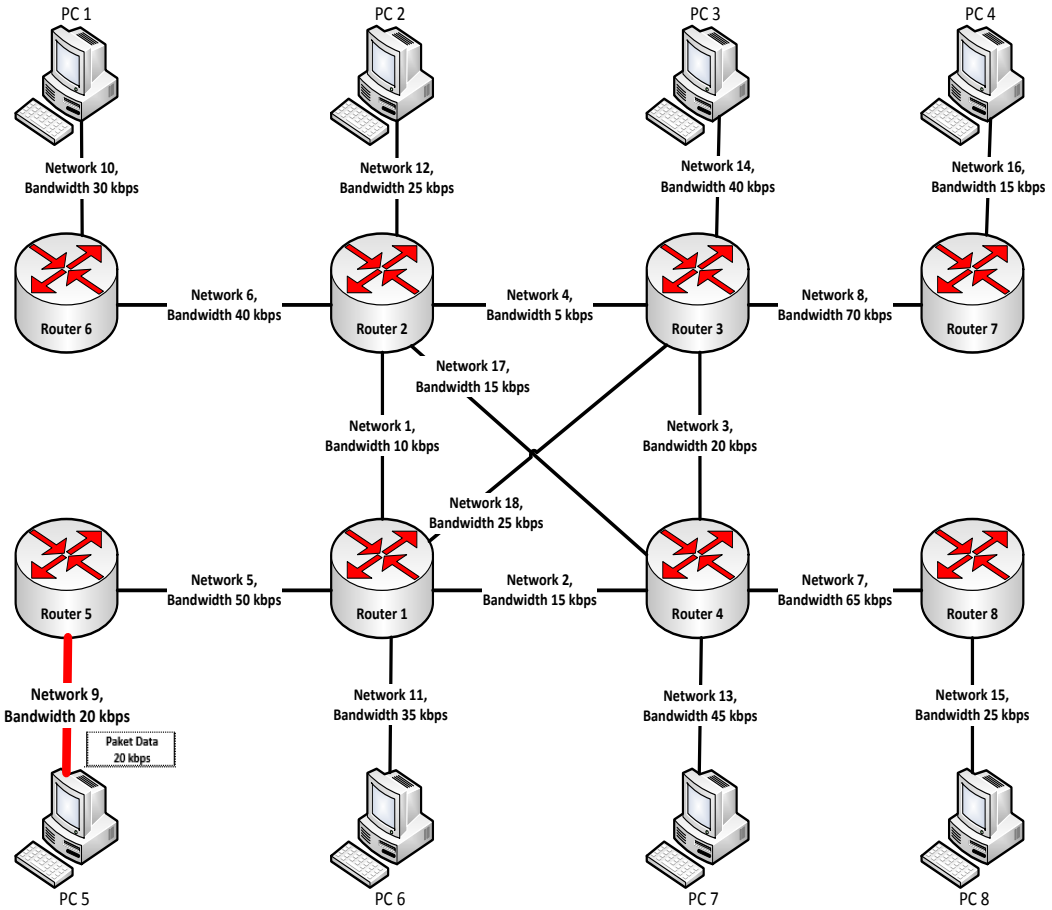
Sumber	Tujuan	*Link* yang digunakan	Jumlah jalur (*hop*)
PC 5	*Router* 5	*network* 9, *bandwidth* 20 kbps	1
PC 5 → *Router* 5	*Router* 1	*network* 9, *bandwidth* 20 kbps *network* 5, *bandwidth* 50 kbps	2
PC 5 → *Router* 5 → *Router* 1	*Router* 2	*network* 9, *bandwidth* 20 kbps *network* 5, *bandwidth* 50 kbps *network* 1, *bandwidth* 60 kbps	3
PC 5 → *Router* 5 → *Router* 1	*Router* 3	*network* 9, *bandwidth* 20 kbps *network* 5, *bandwidth* 50 kbps *network* 18, *bandwidth* 85 kbps	3
PC 5 → *Router* 5 → *Router* 1	*Router* 4	*network* 9, *bandwidth* 20 kbps *network* 5, *bandwidth* 50 kbps *network* 2, *bandwidth* 80 kbps	3
PC 5 → *Router* 5 → *Router* 1 → *Router* 3	*Router* 7	*network* 9, *bandwidth* 20 kbps *network* 5, *bandwidth* 50 kbps *network* 18, *bandwidth* 85 kbps *network* 8, *bandwidth* 70 kbps	4
PC 5 → *Router* 5 → *Router* 1 → *Router* 3 → *Router* 7	PC 4	*network* 9, *bandwidth* 20 kbps *network* 5, *bandwidth* 50 kbps *network* 18, *bandwidth* 85 kbps *network* 8, *bandwidth* 70 kbps *network* 16, *bandwidth* 15 kbps	5

Pemilihan dan penetapan jalur yang dipilih berdasarkan mekanisme teori algoritma dijkstra untuk melewatkan paket data dari PC 5 menuju PC 4 akan terpilih jalur melalui rute awal PC 5 menuju router 5 menuju router 1 menuju router 3 menuju router 7 dan menuju tujuan PC 4 dengan jumlah jarak (hop) sebanyak 5 hop dan jumlah *bandwidth* dari semua jalur yang akan dilalui sebesar 180 kbps. Rute tersebut dipilih oleh algoritma dijkstra pada protokol *routing* OSPF untuk melewatkan paket data karena memiliki jumlah jarak (*hop*) dari PC pengirim menuju PC penerima paling pendek, hal ini karena pada proses penentuannya algoritma dijkstra menggunakan mekanisme *link state* berdasarkan pada jalur terpendek dan tingkat *bandwidth* terbesar yang diperoleh oleh *router* pada saat melakukan proses *broadcast information* di setiap jalur.

Selanjutnya pada tahapan analisis ini akan dibuat juga skenario kedua untuk melihat titik kritis serta kondisi beban jalur lalu lintas komunikasi data pada saat terjadi pengiriman suatu paket data dari sumber pengirim yang sama menuju tujuan penerima yang pula pada waktu proses komunikasi data dilakukan secara bersamaan. Skenario yang kedua akan dilakukan analisis dengan melakukan pengiriman paket data sebesar 20 kbps dari PC 5 menuju PC 4, proses perhitungan yang dilakukan pada skenario kedua secara bertahap sebagai berikut:

1. Perhitungan pada Iterasi ke 0 skenario ke 2

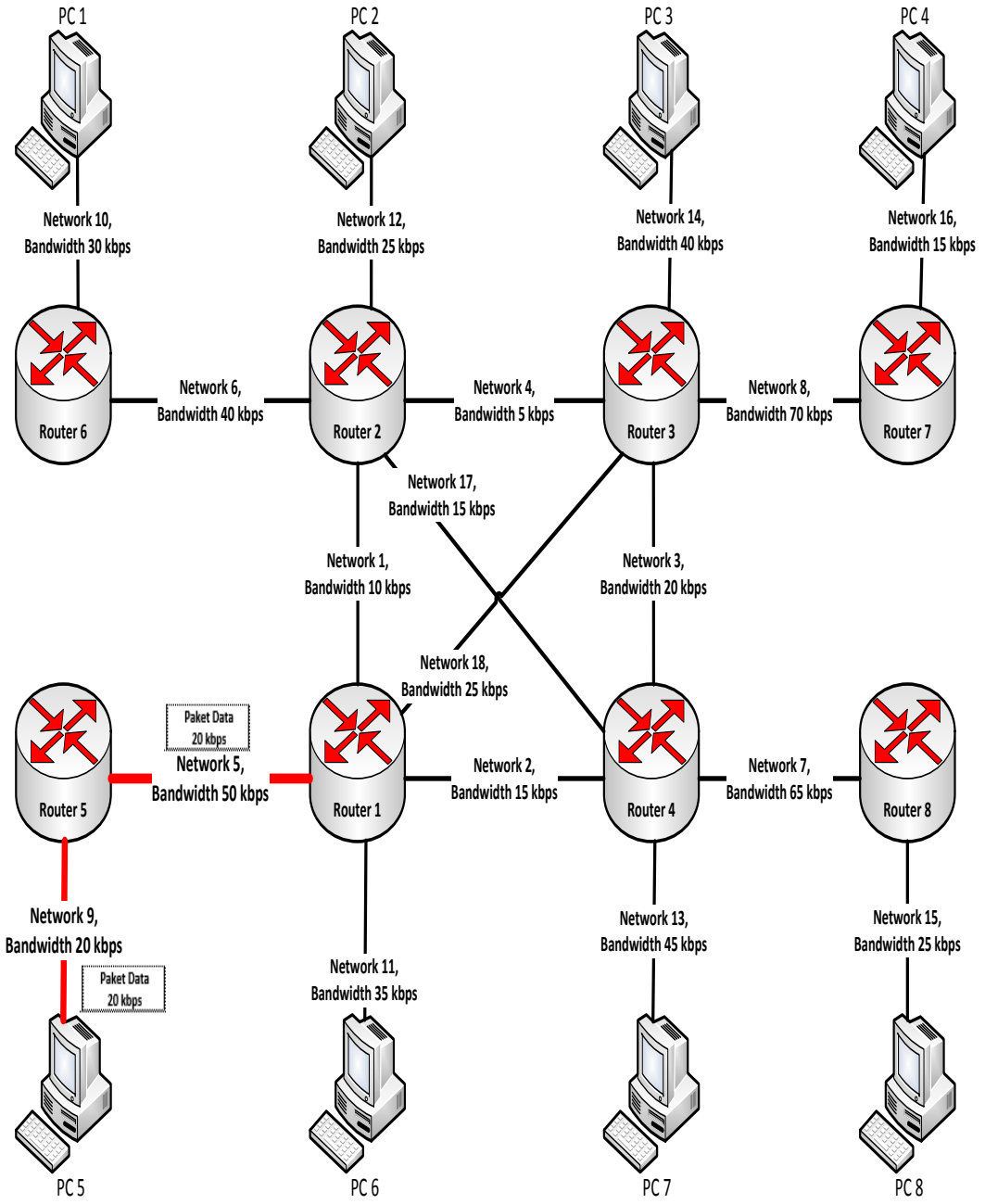
Perhitungan yang dilakukan oleh algoritma dijkstra untuk membentuk suatu informasi pada tabel *routing* pada iterasi ke 0 adalah paket data pertama kalinya akan diteruskan menuju *router* 1 dan melalui jalur *link network* 9 dengan ukuran *bandwidth* sebesar 20 kbps untuk melewatkan paket data sebesar 20 kbps, ilustrasi skenario untuk perhitungan pada iterasi ke 0 dapat dilihat pada Gambar III-11.



Gambar III-11 Proses Perhitungan pada iterasi ke 0 skenario ke 2

2. Perhitungan pada Iterasi ke 1 skenario ke 2

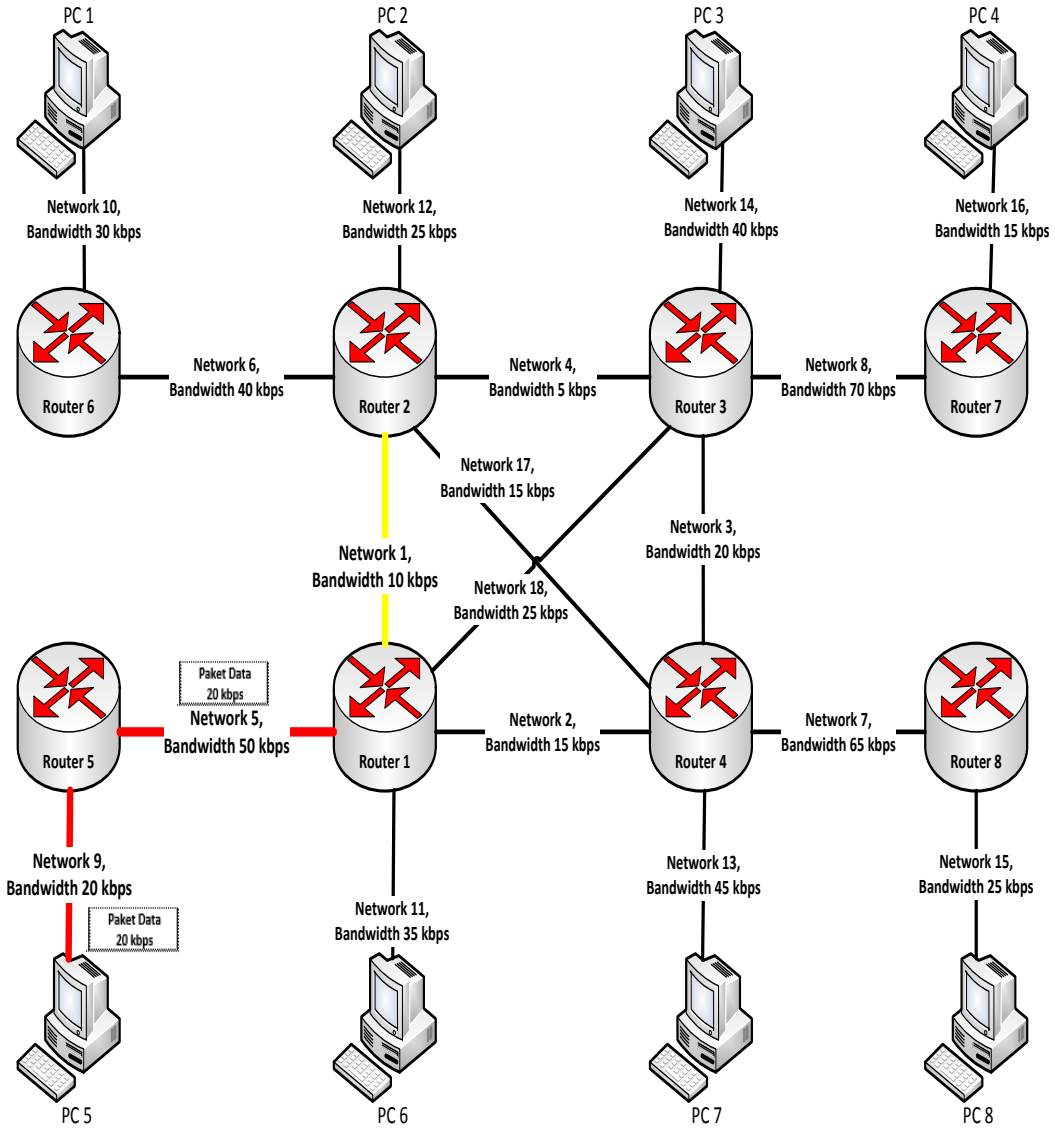
Proses perhitungan pada iterasi ke 1, proses yang dilakukan selanjutnya adalah *router* 5 akan langsung meneruskan paket data menuju *router* 1, ilustrasi skenario pada iterasi ke 1 pada skenario ke 2 dapat dilihat pada Gambar III-12.



Gambar III-12 Proses Perhitungan pada iterasi ke 1 skenario ke 2

3. Perhitungan pada Iterasi ke 2 skenario ke 2

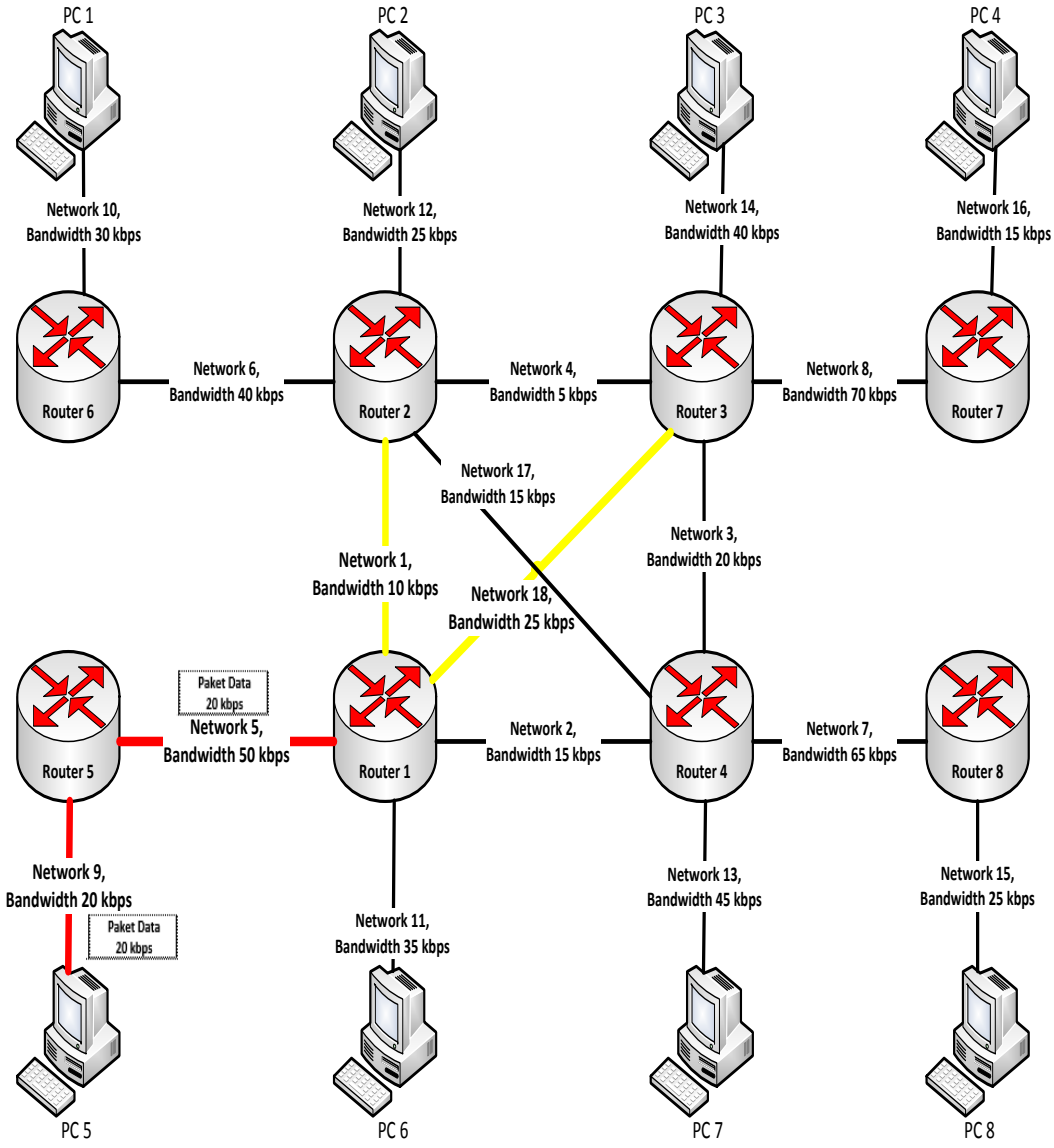
Proses perhitungan pada iterasi ke 2, paket data akan menunggu terlebih dahulu sampai didapatkan informasi tabel *routing* yang berisi jalur yang akan digunakan untuk proses komunikasi. *Router* 1 melakukan *broadcast information* berupa pengecekan jalur di *level data link* pada lapisan OSI menuju *router* 2 yang dilakukan untuk mengumpulkan informasi berupa kondisi jalur dengan kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 2 dapat dilihat pada Gambar III-13, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-13 Proses Perhitungan pada iterasi ke 2 skenario ke 2

4. Perhitungan pada Iterasi ke 3 skenario ke 2

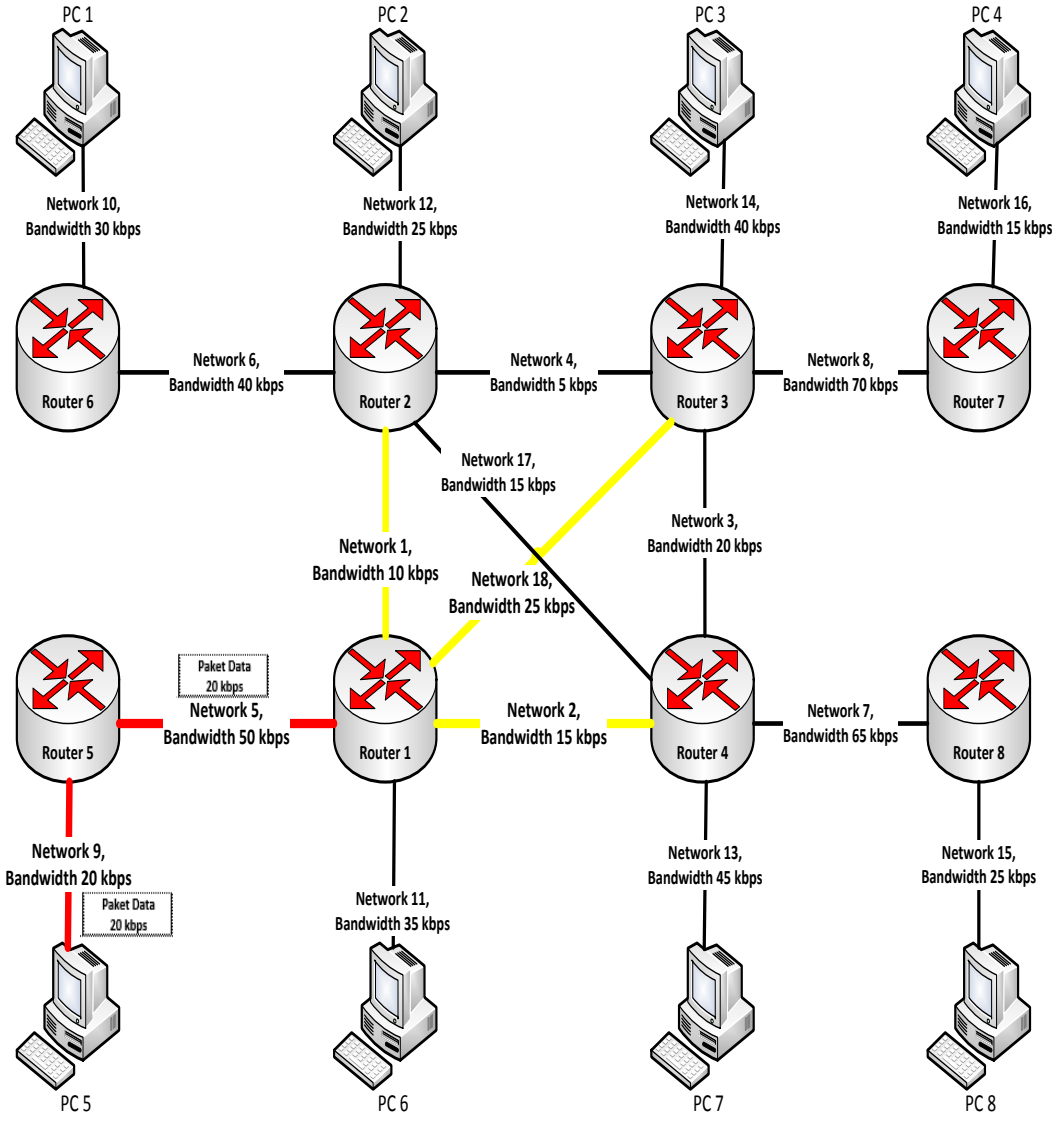
Perhitungan pada iterasi ke 3, setelah *router* 1 melakukan *broadcast information* menuju *router* 2. *Router* 1 melakukan *broadcast information* berupa pengecekan jalur di *level data link* pada lapisan OSI menuju *router* 3 yang dilakukan untuk mengumpulkan informasi berupa kondisi jalur dengan kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 3 dapat dilihat pada Gambar III-14, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-14 Proses Perhitungan pada iterasi ke 3 skenario ke 2

5. Perhitungan pada Iterasi ke 4 skenario ke 2

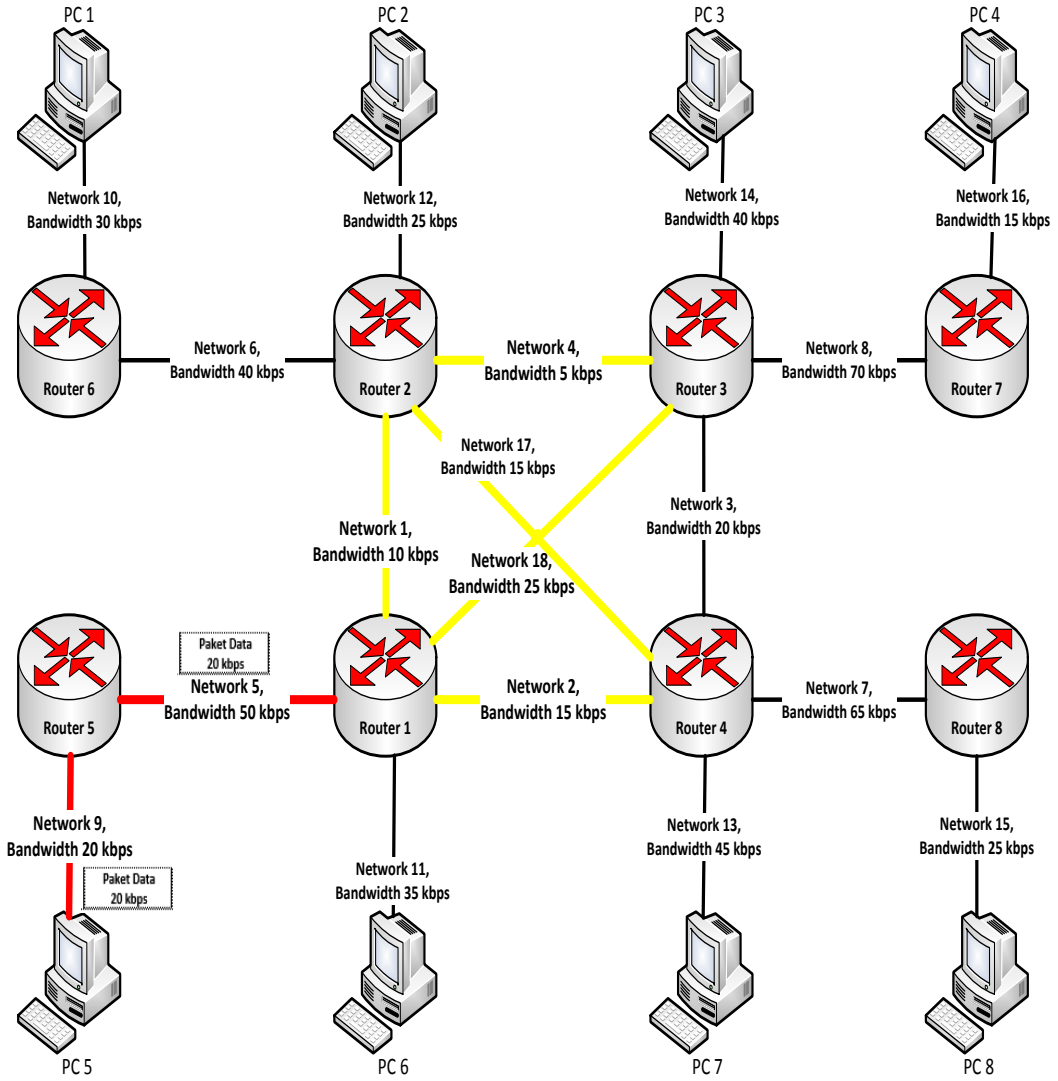
Perhitungan pada iterasi ke 3, setelah *router* 1 melakukan *broadcast information* menuju *router* 2 dan *router* 3. *Router* 1 selanjutnya melakukan *broadcast information* berupa pengecekan jalur di *level data link* pada lapisan OSI menuju *router* 4 yang dilakukan untuk mengumpulkan informasi berupa kondisi jalur dengan kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 4 dapat dilihat pada Gambar III-15, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-15 Proses Perhitungan pada iterasi ke 4 skenario ke 2

6. Perhitungan pada Iterasi ke 5 skenario ke 2

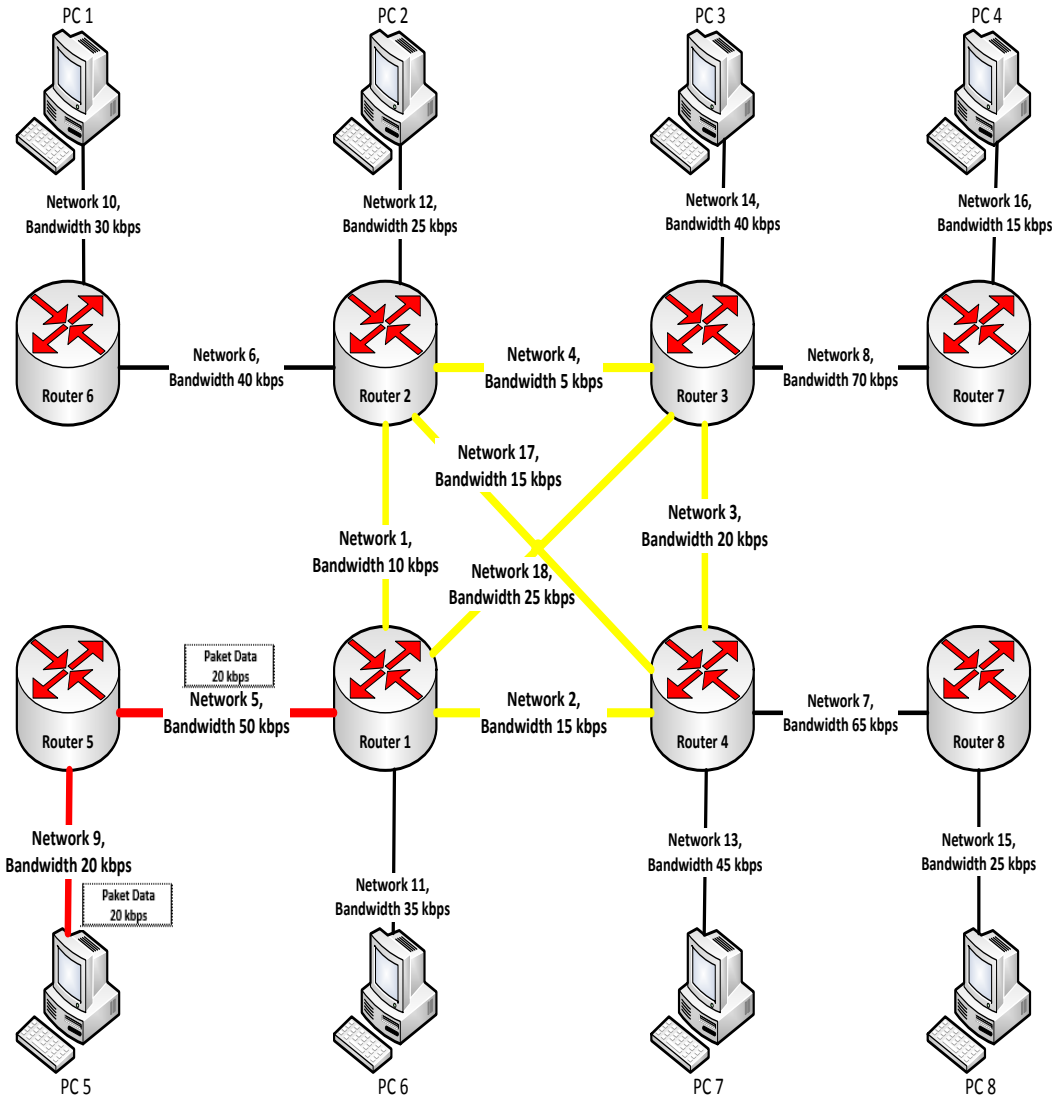
Pada proses perhitungan di iterasi ke 5, setelah *router* 1 melakukan *broadcast information* ke semua *link* yang terhubung dengan *router* 2, *router* 3, dan *router* 4. Proses selanjutnya pada *router* 2 melakukan proses *broadcast information* di *network* 4 yang terhubung dengan *router* 3 dan *network* 17 yang terhubung dengan *router* 4 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 5 dapat dilihat pada Gambar III-16, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-16 Proses Perhitungan pada iterasi ke 5 skenario ke 2

7. Perhitungan pada Iterasi ke 6 skenario ke 2

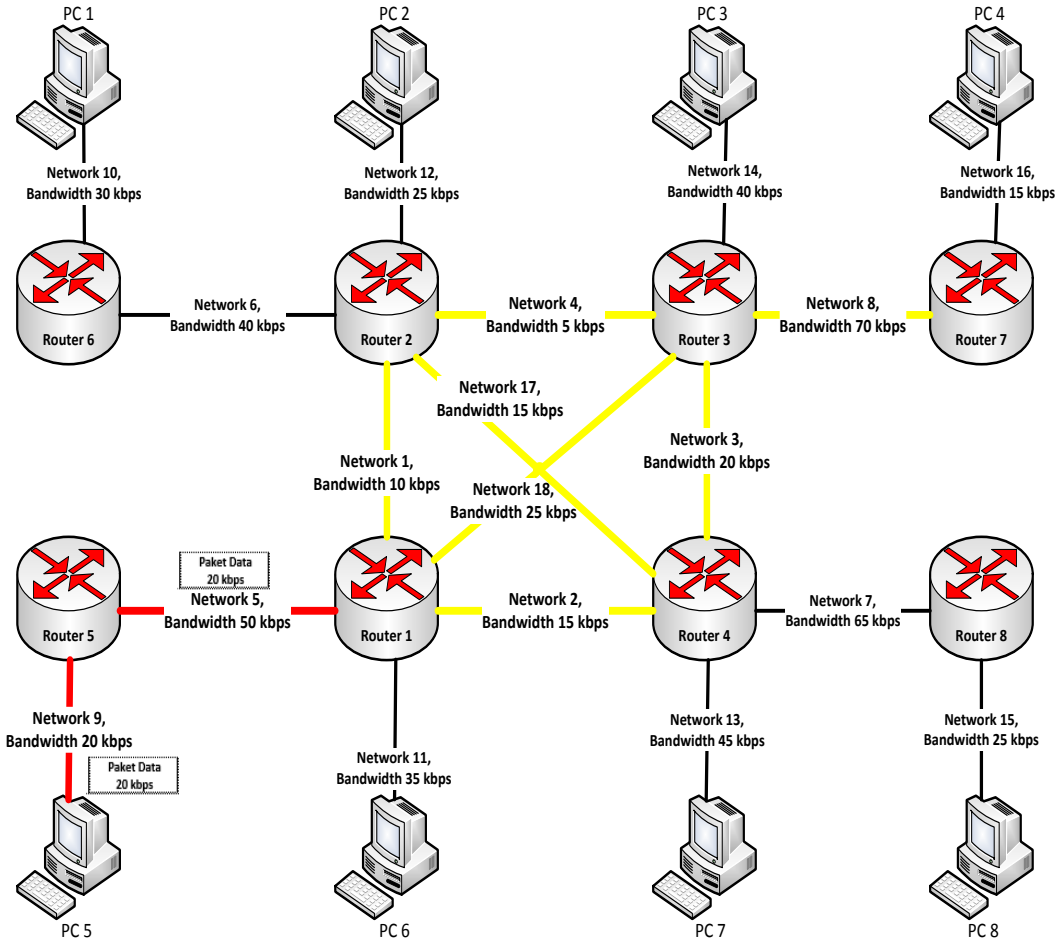
Pada proses perhitungan yang dilakukan di iterasi ke 6 sama seperti pada proses di iterasi ke 5, *router* 4 melakukan proses *broadcast information* pada *network* 17 yang terhubung dengan *router* 2 dan pada *network* 3 yang terhubung dengan *router* 3 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi dari skenario tersebut dapat dilihat pada Gambar III-17, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-17 Proses Perhitungan pada iterasi ke 6 skenario ke 2

8. Perhitungan pada Iterasi ke 7 skenario ke 2

Proses pada iterasi ke 7 selanjutnya proses *broadcast information* akan dilakukan oleh *router* 3 pada jalur *network* 8 yang terhubung dengan *router* 7 untuk mengumpulkan informasi berupa kondisi jalur, informasi ukuran serta kualitas *bandwidth* terbaik. Ilustrasi skenario pada iterasi ke 5 dapat dilihat pada Gambar III-18, proses *broadcast information* di tandai dengan pewarnaan pada jalur dengan warna kuning.



Gambar III-18 Proses Perhitungan pada iterasi ke 7 skenario ke 2

9. Perhitungan pada Iterasi ke 8 skenario ke 2

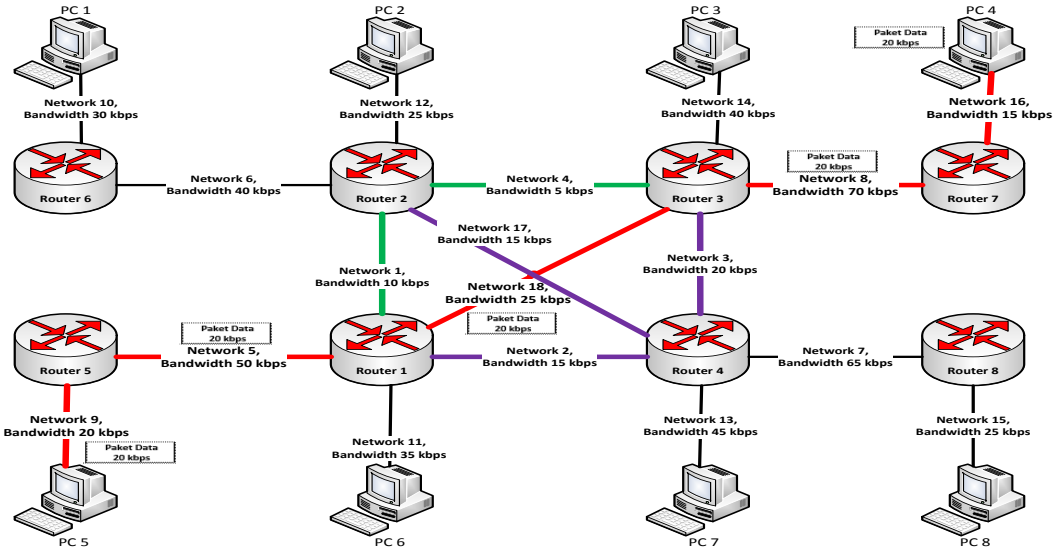
Pada proses perhitungan di iterasi ke 8 skenario ke 2, setelah dilakukan proses *broadcast information*, selanjutnya akan dilakukan perhitungan untuk menentukan jumlah jarak (hop) dari seluruh jalur yang dimungkinkan untuk dilewati oleh paket data yang dikirim dari PC 5 menuju tujuan terakhir. Proses

perhitungan ini dilakukan oleh algoritma dijkstra untuk menentukan rute mana yang akan dilalui oleh paket data dengan ketentuan jumlah jarak paling pendek untuk dilalui dan memiliki kualitas jalur *bandwidth* paling baik. Penggunaan *bandwidth* dan kondisi jalur ditabulasikan pada Tabel III-3.

Tabel III-3 Penggunaan Bandwidth dan Kondisi Jalur

Jalur	Jumlah Jarak (Hop)	Jumlah *Bandwidth*	Kondisi Jalur
PC 5 → router 5 → router 1 → router 2 → router 3 → router 7 → PC 4	6	170 kbps	Gagal
PC 5 → router 5 → router 1 → router 3 → router 7 → PC 4	5	180 kbps	Penuh
PC 5 → router 5 → router 1 → router 4 → router 3 → router 7 → PC 4	6	190 kbps	Kosong

Penggunaan jalur akhir yang digunakan pada skenario 2 dapat dilihat pada Gambar III-19. Jalur yang diwarnai dengan warna merah merupakan jalur utama yang digunakan ketika proses komunikasi data berlangsung, jalur yang diwarnai dengan warna ungu merupakan jalur kosong yang dapat dilalui, namun diabaikan oleh protokol *routing* OSPF karena jumlah jarak (*hop*) menuju tujuan terlalu jauh, terakhir jalur yang diwarnai dengan warna hijau adalah jalur yang tidak dapat digunakan karena terjadi kegagalan ketika melakukan sambungan jalur karena *bandwidth* yang dibutuhkan tidak cukup untuk melewatkan paket data.



Gambar III-19 Penentuan Jalur yang digunakan pada skenario ke 2

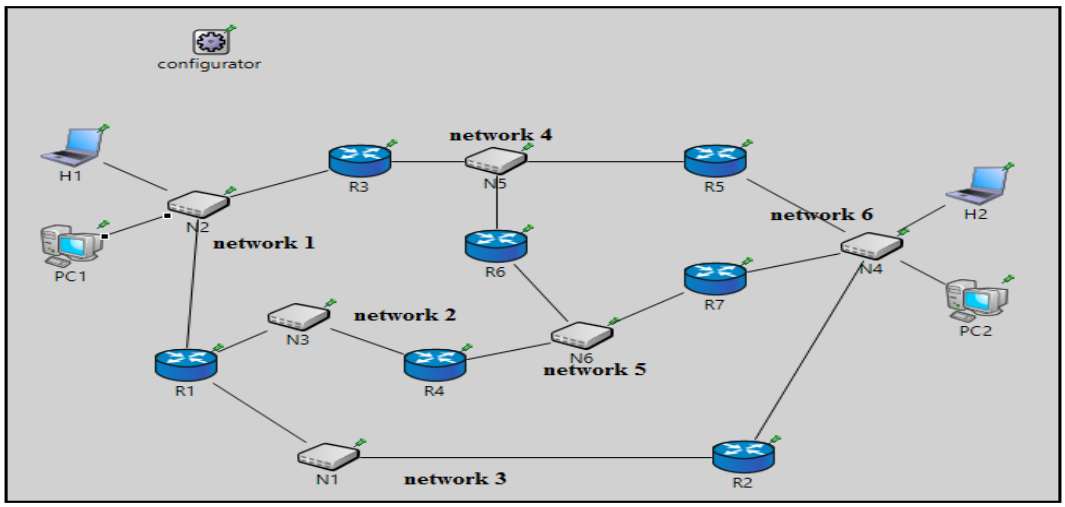
Dari hasil analisis pada skenario kedua penentuan akhir jalur yang dapat dilalui ketika terjadi proses komunikasi data dalam waktu yang bersamaan, protokol *routing* OSPF akan tetap memilih jalur utama melalui rute PC 5, *router* 5, *router* 1, *router* 3, *router* 7 kemudian data disampaikan menuju PC 4. Proses komunikasi pada skenario pertama kondisi jalur tidak akan terjadi penumpukan data, karena hanya terjadi satu proses pengiriman data dari PC 5 menuju PC 4. Ketika pada waktu yang bersamaan terjadi pengiriman data dari sumber menuju tujuan yang sama, maka akan terjadi antrian di *router* 1 yang terjadi karena paket data yang dikirimkan sebesar 20 kbps yang dikirimkan pada skenario kedua akan menunggu terlebih dahulu proses pengiriman yang pertama selesai, hal ini terjadi karena pada skenario pertama *router* 1 sudah menentukan jalur utama untuk mengirimkan paket data melalui *link* jalur *network* 18. Paket akan mengalami antrian karena pada *network* 18 tidak dapat melewatkan paket lebih dari ukuran *bandwidth* pada jalur yaitu sebesar 25 kbps pada waktu yang bersamaan. Skenario yang pertama sedang dilakukan juga pengiriman paket data sebesar 15 kbps, oleh karena itu paket data pada skenario kedua akan di antrikan terlebih dahulu, hal ini dapat mengakibatkan tumpukan antrian yang berakibat pada tingginya waktu pengiriman data (*delay ratio*) dan perubahan trafik pada jalur komunikasi. Kondisi jalur pada Tabel III-3 terdapat keterangan kondisi jalur gagal, hal tersebut disebabkan karena kapasitas maksimum *bandwidth* pada jalur yang menghubungkan *router* 1 dengan *router* 2 berukuran 15 kbps dan tidak dapat melewatkan paket data yang memiliki ukuran sebesar 20 kbps.

III.2.3. Analisis Mekanisme Protokol OSPF saat Proses Komunikasi Data

Proses analisis terhadap mekanisme protokol OSPF saat proses komunikasi data dilakukan untuk mengetahui bagaimana mekanisme dari suatu protokol OSPF bekerja serta untuk mengetahui parameter-parameter apa saja yang terlibat ketika protokol OSPF bekerja, sehingga dapat diketahui hasil dari proses analisis untuk digunakan sebagai parameter acuan pada saat dilakukan tahapan optimasi kinerja dari protokol OSPF. Protokol *routing* OSPF menggunakan mekanisme *link state,* mekanisme *link state* yang diterapkan pada protokol *routing* OSPF akan

mengabaikan informasi jalur (*link network*) yang memiliki kapasitas ukuran *bandwidth* yang relatif lebih kecil, sedangkan terdapat jalur (*link network*) lainnya yang dapat digunakan untuk melewatkan paket data dengan tingkat kepadatan jalur yang tidak terlalu padat walaupun kapasitas ukuran *bandwidth* lebih kecil. Mekanisme *link state* yang digunakan pada OSPF sangat baik digunakan dalam menentukan jalur terpendek menuju tujuan, akan tetapi kurang efisien diterapkan pada jaringan dengan topologi multi jalur karena akan membutuhkan banyak sekali konsumsi *bandwidth* yang terpakai yang dapat menyebabkan penurunan kapasitas *bandwidth* serta perubahan beban data pada suatu jalur, dan juga membutuhkan sumber daya memori pada *router* saat terjadi perhitungan jalur ketika ada perubahan kondisi jalur data pada suatu jaringan komputer.

Model topologi jaringan yang digunakan pada tahap analisis mekanisme protokol *routing* OSPF akan menggunakan contoh model topologi dari jaringan di lingkungan *inherent* sekolah SMKN 1 Cimahi. Analisis akan dilakukan menggunakan *tools* pemodelan dan simulasi jaringan *omnet++*. Hasil dari analisis mekanisme protokol *routing* OSPF yang dilakukan berupa informasi penggunaan *bandwidth*, nilai *throughput*, informasi *packet delay*, *packet delivery ratio*, dan informasi jalur *routing* yang terbentuk ketika protokol *routing* OSPF diterapkan pada suatu jaringan komputer. Contoh model topologi jaringan *inherent* yang digunakan untuk analisis mekanisme protokol *routing* OSPF dapat dilihat pada Gambar III-20 di bawah ini.



Gambar III-20 Model Topologi *Inherent*

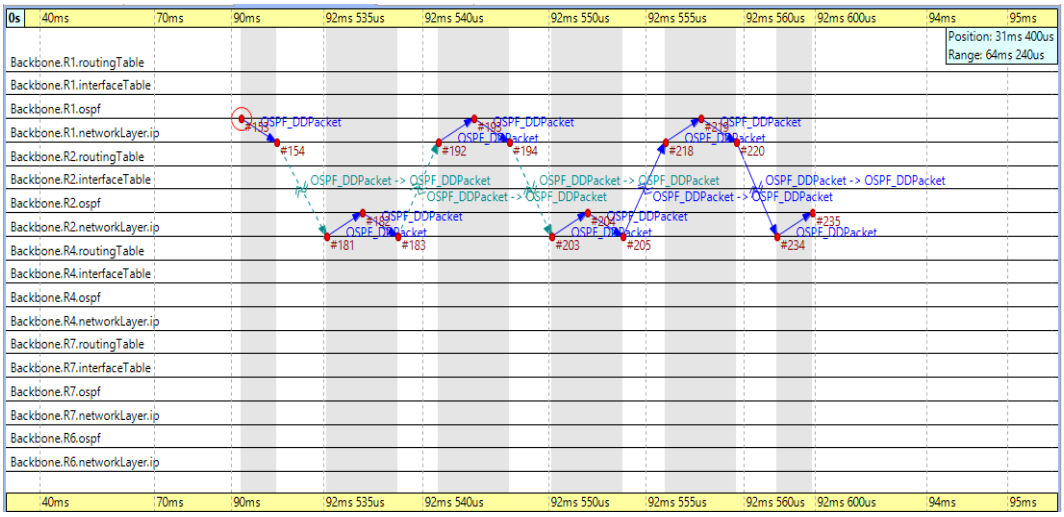
Keterangan untuk alokasi segmentasi alamat IP dari masing-masing *network* pada model topologi jaringan *inherent* di atas dapat dilihat pada Tabel III-4.

Tabel III-4 Alokasi Segmentasi Jaringan *Inherent*

Area	Network ID
Network 1	192.168.2.0/24
Network 2	192.168.3.0/24
Network 3	192.168.1.0/24
Network 4	192.168.5.0/24
Network 5	192.168.6.0/24
Network 6	192.168.4.0/24

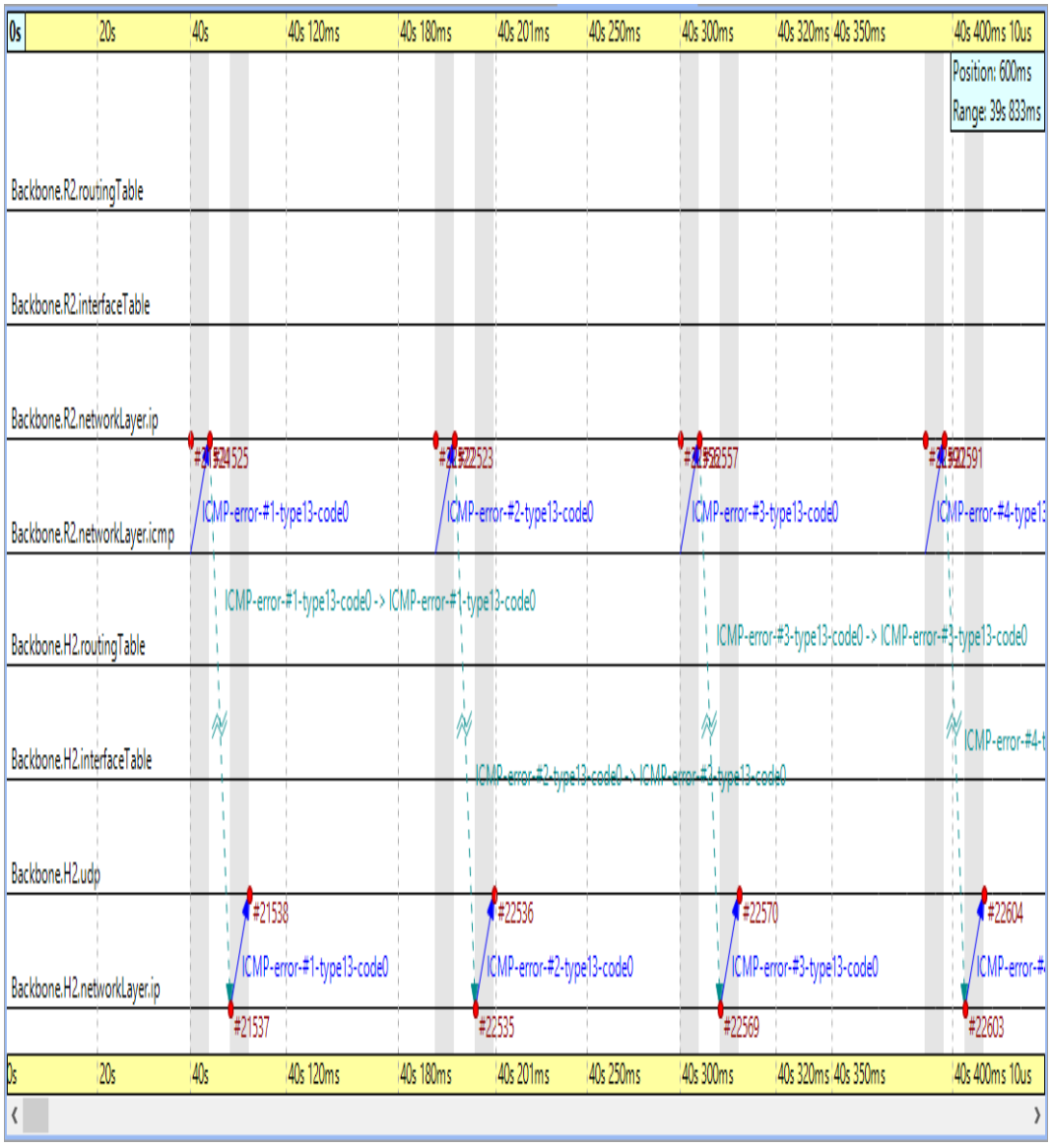
Skenario pertama yang akan diterapkan untuk melakukan analisis mekanisme protokol *routing* OSPF dalam memodelkan dan mensimulasikan topologi jaringan *inherent* pada Gambar III-20 adalah terjadi proses komunikasi data berupa layanan video *teleconference* yang dilakukan oleh PC1 dengan PC2. Skenario kedua adalah dilakukan proses pengiriman berupa paket data udp yang dikirimkan dalam waktu bersamaan dari PC1 menuju PC2 dan dari H1 menuju H2 untuk mengetahui kepadatan lalu lintas data pada jalur yang dilalui oleh suatu paket data serta mengetahui rata-rata konsumsi *bandwidth* yang digunakan.

Hasil untuk analisis dari skenario pertama maupun skenario kedua didapatkan hasil berupa *sequential chart* yang menunjukkan jalur utama yang dipilih oleh protokol *routing* OSPF untuk melewatkan paket data ketika terjadi proses komunikasi data. Hasil dari *sequential chart* dapat dilihat pada Gambar III-21.



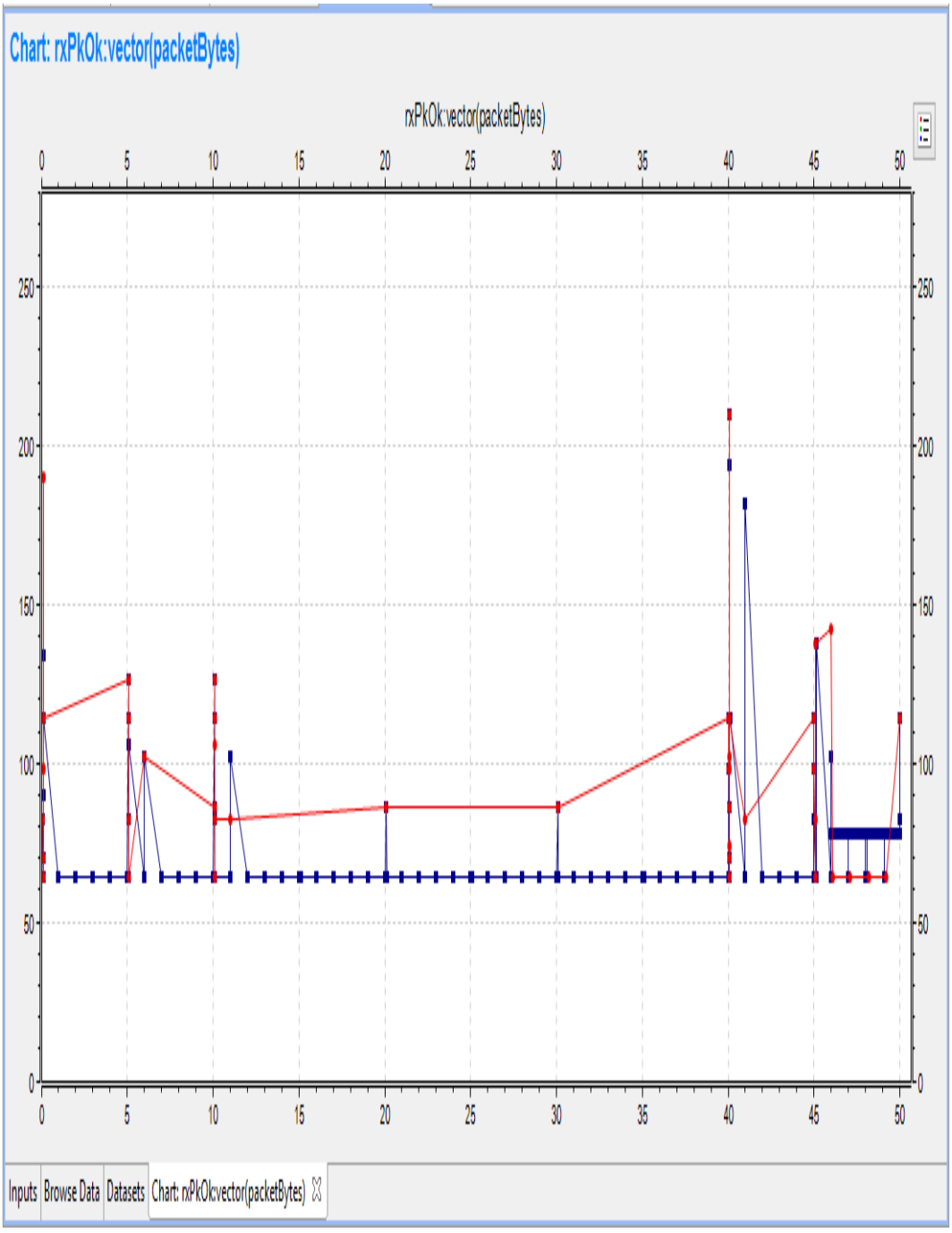
Gambar III-21 *Sequential Chart* Penggunaan Jalur pada OSPF

Hasil yang ditampilkan pada *sequential chart* pada Gambar III-21 diatas menunjukkan pada interval waktu 90 ms terbentuk informasi berupa header paket data OSPF yang digunakan sebagai informasi jalur utama dalam proses komunikasi data berlangsung. Paket data yang dikirimkan akan dilewatkan melalui *router* R1 dan *router* R2. Pemilihan jalur dan rute yang digunakan dalam proses komunikasi data ditentukan oleh mekanisme *link state* protokol *routing* OSPF berdasarkan pada jumlah hop yang terpendek. Hasil yang ditampilkan pada Gambar III-22 menunjukkan *packet loss* yang terjadi pada jalur utama yang dipakai ketika proses komunikasi data berlangsung.



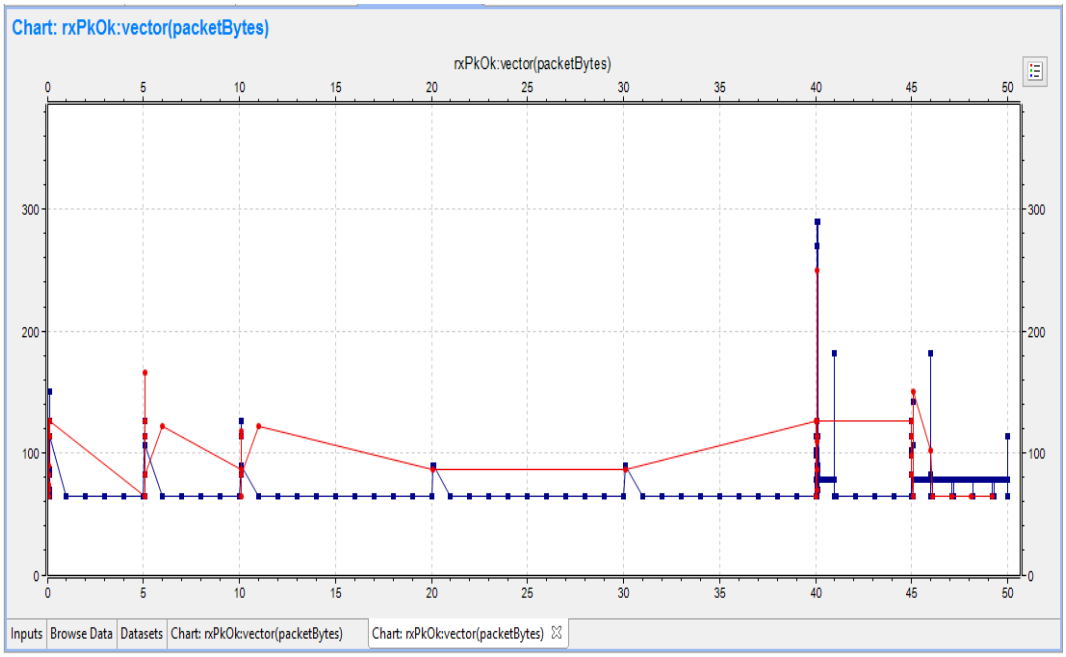
Gambar III-22 *Packet Loss* pada jalur komunikasi yang digunakan

Hasil yang ditampilkan pada Gambar III-23 menunjukkan penggunaan *bandwidth* yang diperlukan oleh *router* R1 saat melewatkan suatu paket data, grafik yang ditandai dengan warna merah merupakan *bandwidth* yang digunakan oleh *router* R1 untuk menerima paket data dan grafik yang ditandai dengan warna biru digunakan untuk melakukan pengiriman paket data.



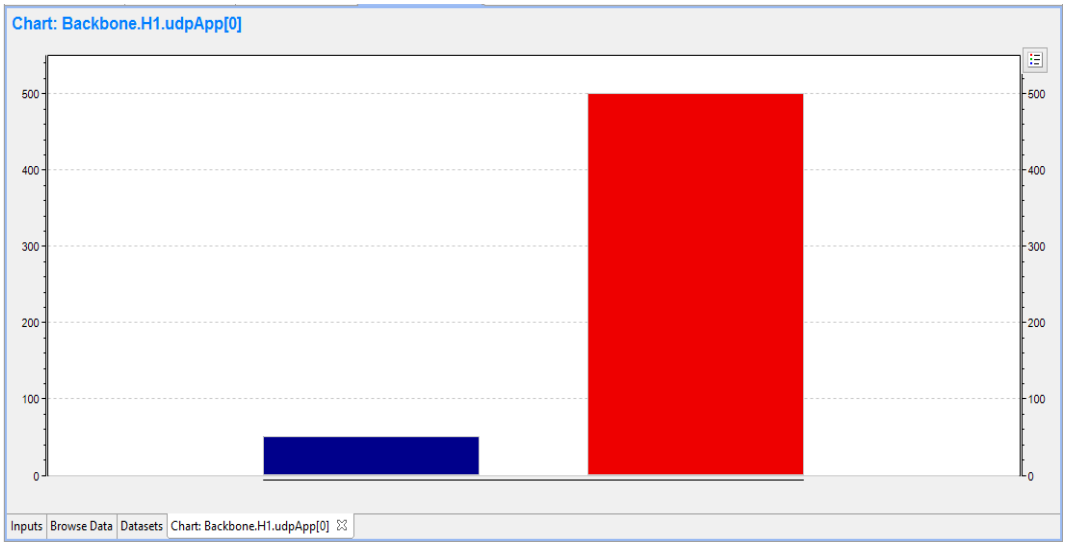
Gambar III-23 *Bandwidth* yang digunakan di *router* R1

Hasil yang ditunjukkan pada Gambar III-24 menunjukkan penggunaan *bandwidth* yang diperlukan oleh *router* R2 saat melewatkan suatu paket data.



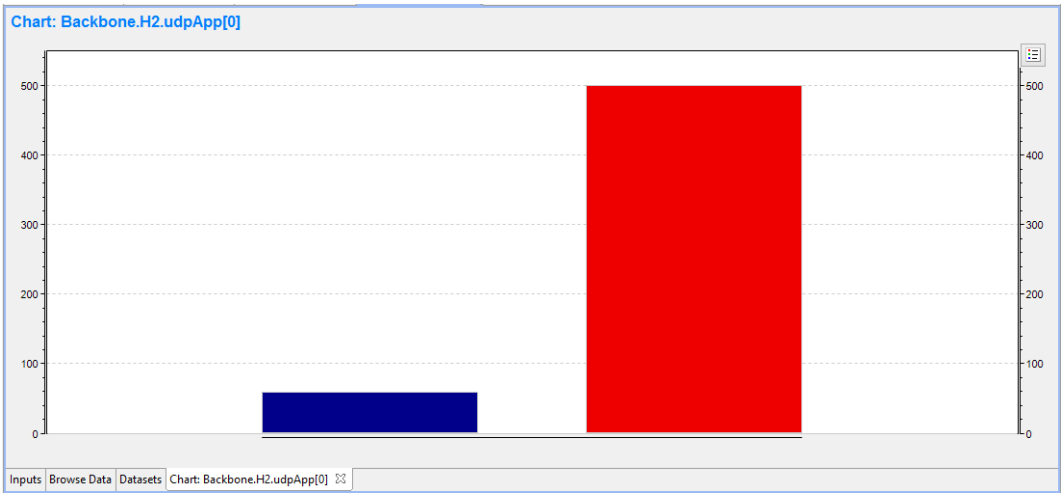
Gambar III-24 *Bandwidth* yang digunakan di *router* R2

Throughput yang dihasilkan pada saat terjadinya proses komunikasi data pada *host* H1 dapat dilihat pada Gambar III-25. Diagram batang yang ditandai dengan warna merah merupakan paket data yang dikirim sebesar 500 Kbps, sedangkan diagram batang yang ditandai dengan warna biru merupakan paket data yang diterima sebesar 50 Kbps saat proses komunikasi data berlangsung.



Gambar III-25 *Throughput* pada *Host* H1

Hasil yang ditunjukkan pada Gambar III-26 merupakan *throughput* yang dihasilkan pada saat terjadinya proses komunikasi data pada *host* H2. Diagram batang yang ditandai dengan warna merah merupakan paket data yang dikirim sebesar 500 Kbps, sedangkan diagram batang yang ditandai dengan warna biru merupakan paket data yang diterima sebesar 53 Kbps saat proses komunikasi data berlangsung.



Gambar III-26 *Throughput* pada *Host* H2

Hasil dari analisis yang dilakukan terhadap mekanisme protokol *routing* OSPF diperoleh beberapa parameter-parameter hasil yang dapat dijadikan untuk acuan untuk dilakukan optimasi terhadap kinerja protokol *routing* OSPF. Optimasi yang dilakukan guna meningkatkan efisiensi pada penggunaan jalur dan konsumsi *bandwidth* ketika terjadi proses komunikasi data berlangsung. Contoh parameter dari hasil proses analisis pada mekanisme protokol *routing* OSPF salah satunya hasil yang ditampilkan pada Gambar III-21. Hasil yang ditampilkan menunjukkan penggunaan jalur komunikasi data dibebankan hanya pada salah satu jalur saja sebagai jalur utama. Kondisi demikian pada jalur dengan intensifitas lalu lintas data yang padat sangat beresiko terjadinya kegagalan pada saat pengiriman paket data apabila jalur utama mengalami putus hubungan, ketika protokol *routing* OSPF pada suatu *router* membentuk kembali jalur untuk rute komunikasi baru diperlukan sumber daya memori dan prosesor yang cukup besar. Kejadian seperti tersebut dapat terjadi sewaktu-waktu tanpa bisa diprediksi oleh admin jaringan, maka dari diperlukan adanya optimasi berupa pembentukan

model multi label protokol label *switching (MPLS)* untuk meningkatkan guna meminimalisir kejadian-kejadian tersebut.

III.3. Analisis Metode Optimasi

Tahap analisis metode optimasi merupakan tahap yang digunakan untuk melakukan analisis optimasi terhadap protokol *routing* OSPF serta untuk mengetahui parameter-parameter apa saja yang terlibat untuk melakukan proses optimasi terhadap kinerja protokol *routing* OSPF yang akan dilakukan pada penelitian ini diataranya:

III.3.1. Analisis Algoritma Constraint Shortest Path First

Algoritma *constraint shortest path first* (CSPF) pada dasarnya sama seperti algoritma dijkstra secara mekanisme, hanya saja ada perbedaan pada strategi algoritma yang tidak menggantungkan pada tingkat kualitas *bandwidth* saat pemilihan jalur untuk melewatkan suatu paket data. Algoritma ini dikembangkan untuk meningkatkan kinerja dari algoritma dijkstra dalam menyelesaikan masalah pencarian jalur terpendek. Algoritma CSPF secara mekanisme akan melawatkan suatu paket data dengan cara menyebarkan informasi jalur ke semua jalur yang terhubung dengan suatu *link state routing* (LSR), sehingga paket data yang dikirimkan akan dilewatkan menyebar melalui semua jalur. *Pseudo code* untuk algoritma *constraint shortest path first* (CSPF) sebagai berikut:

```
function CSPF (Graph, source):
   for each vertex v in Graph:
      dist[v] := infinity ;
      previous[v] := undefined;
end for

dist[source] := 0;
Q := the set of all nodes in Graph;

while Q is not empty:
   u := vertex in Q with smallest distance in dist[] ;
   remove u from Q ;
   if dist[u] = infinity:
      break;
   end if

   for each neighbor v of u:
      alt := dist[u] + dist_between(u,v);
```

```
      if alt < dist[v]:
        dist[v] := alt;
        previous[v] := u;
        decrease-key v in Q;
      end if
   end for
end while

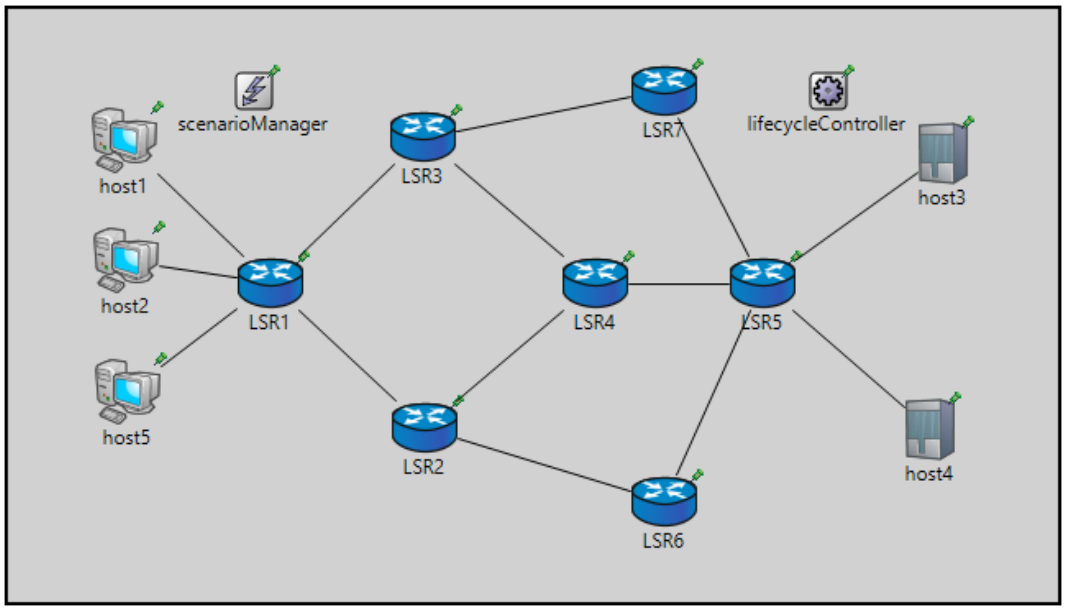
return dist;

endfunction
```

Algoritma CSPF ini tidak akan melewatkan paket data pada satu jalur saja dengan ukuran *bandwidth* terbaik, algoritma ini secara mekanisme akan langsung melakukan *broadcast information* keseluruh jalur yang ada pada setiap *router* yang saling terhubung pada suatu topologi multi jalur. Algoritma ini biasanya di implementasikan untuk memperbaiki kinerja protokol *routing* OSPF pada jaringan multi jalur guna meningkatkan perfoma pengiriman suatu paket data ketika proses komunikasi data berlangsung, sehingga dapat meminimalisir kemungkinan terjadinya penumpukan antrian pada satu jalur komunikasi.

III.3.2. Analisis Rancangan Model MPLS

Tahap analisis yang dilakukan untuk merancang dan membuat model MPLS yang akan digunakan untuk melakukan optimasi protokol *routing* OSPF, mulai dari model konseptual rancangan kebutuhan perangkat yang diperlukan terkait pembuatan jaringan MPLS untuk optimasi OSPF. Hasil yang ditunjukkan pada Gambar III-27 menunjukkan rancangan model topologi jaringan MPLS yang akan dilakukan untuk optimasi protokol *routing* OSPF.



Gambar III-27 Rancangan Topologi dengan Model MPLS

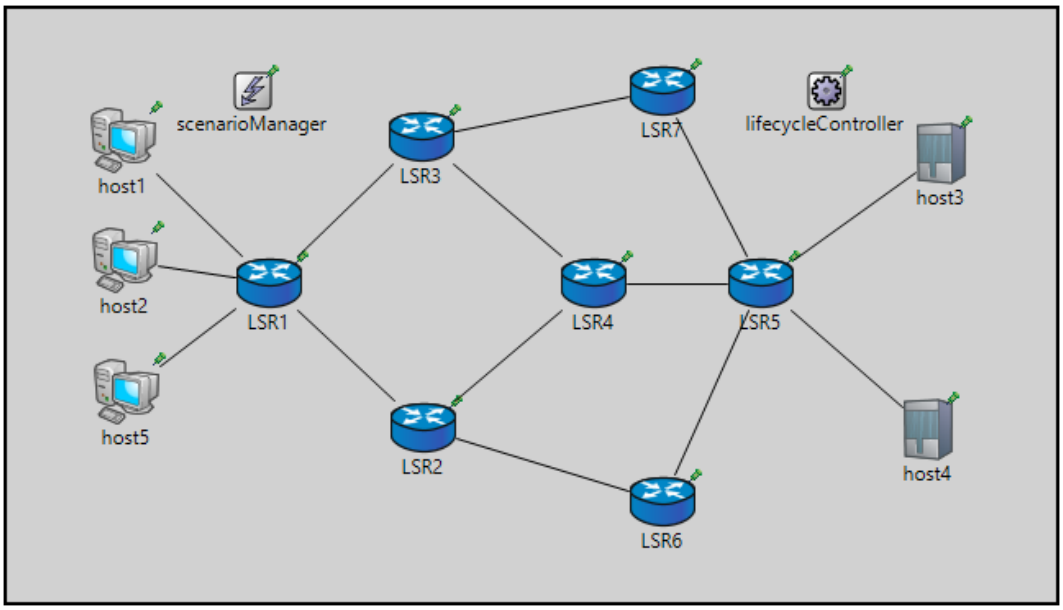
III.3.3. Analisis Mekanisme Traffic Engineering

Tahap analisis dilakukan untuk melakukan penilaian terkait optimasi protokol *routing* OSPF menggunakan model multi protokol label *switching* pada suatu jaringan internet. *Traffic engineering* digunakan untuk optimasi protokol *routing* OSPF agar dapat menangani secara dinamis dan merespon dengan cepat perubahan trafik beban data pada jaringan apabila terjadi suatu perubahan kondisi pada suatu jalur komunikasi data di jaringan komputer. Metode *traffic engineering* dikembangkan menggunakan algoritma *constraint shortest path first* yang merupakan perkembangan dari algoritma dijkstra untuk mengatur aliran trafik data dalam jaringan untuk mengoptimalkan penggunaan *bandwidth* yang berlebihan pada jaringan, untuk memilih rute dan memindahkan trafik dari jalur dengan tingkat kongesti lebih besar ke jalur dengan tingkat kongesti lebih kecil.

III.3.4. Optimasi OSPF dengan Model MPLS

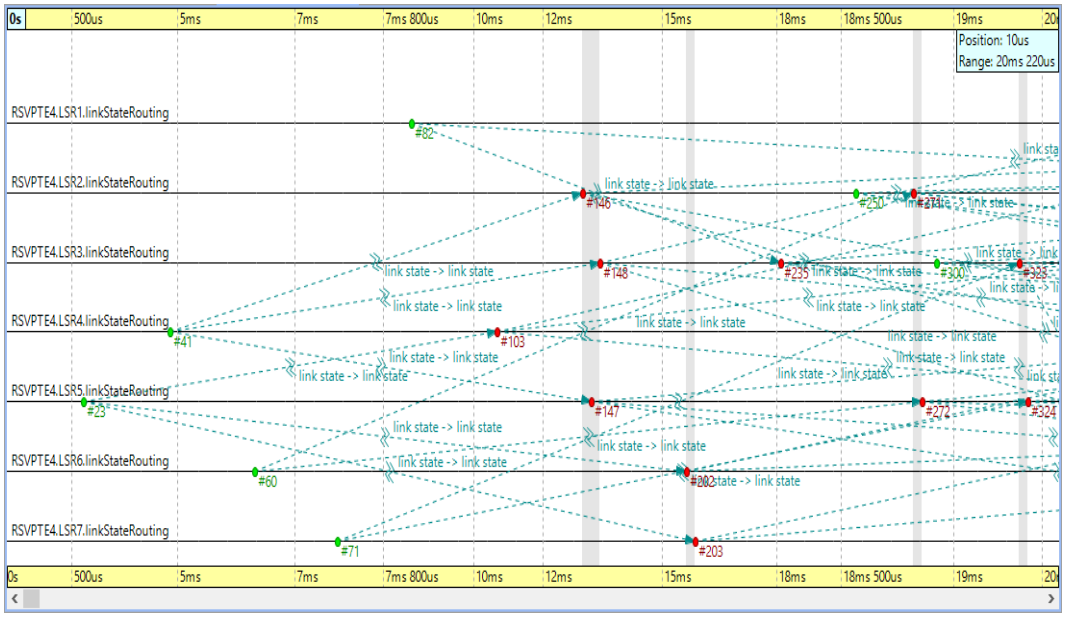
Optimasi yang akan dilakukan untuk meningkatkan kinerja protokol *routing* OSPF menggunakan model MPLS akan terlebih dahulu dibuat suatu skenario yang akan digunakan untuk tahap analisis optimasi menggunakan model MPLS

dengan algoritma CSPF pada suatu jaringan *inherent* yang sudah ada sebelumnya dengan membuat model pada topologi jaringan seperti pada Gambar III-28.



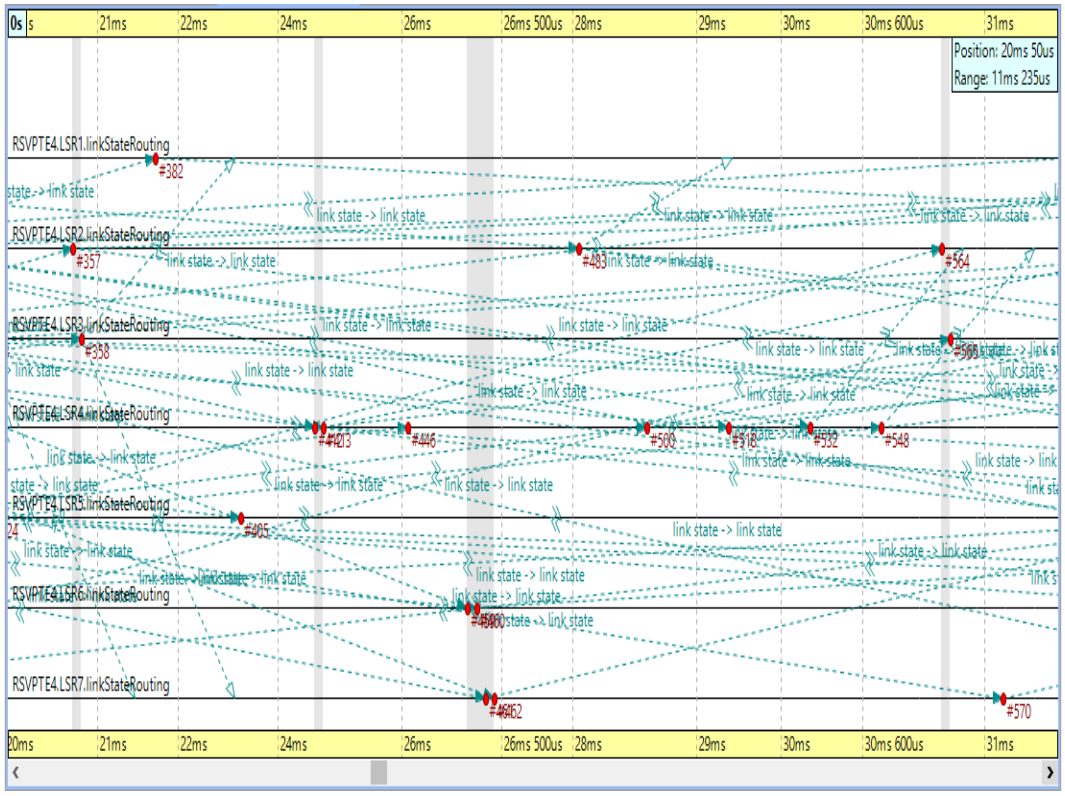
Gambar III-28 Model Jaringan MPLS

Skenario yang akan dibuat untuk melakukan analisis optimasi menggunakan rancangan model topologi MPLS pada Gambar III-28 di atas dilakukan untuk mengetahui dan mengukur tingkat performansi. Skenario pertama akan dilakukan pengukuran terhadap kinerja LSR (*Label Switching Router*) pada MPLS yang digunakan untuk menentukan dan mendistribusikan paket data dari pengirim menuju penerima ketika ada proses komunikasi data, analisis yang digunakan pada skenario pertama dengan cara melakukan *broadcast information* menggunakan paket data udp. Hasil yang ditampilkan pada *sequential chart* dari hasil pengukuran dengan melakukan *broadcast information* yang terjadi dari interval waktu 0 ms sampai 20 ms dapat di lihat pada Gambar III-29.



Gambar III-29 Sequential Chart Broadcast Information LSR

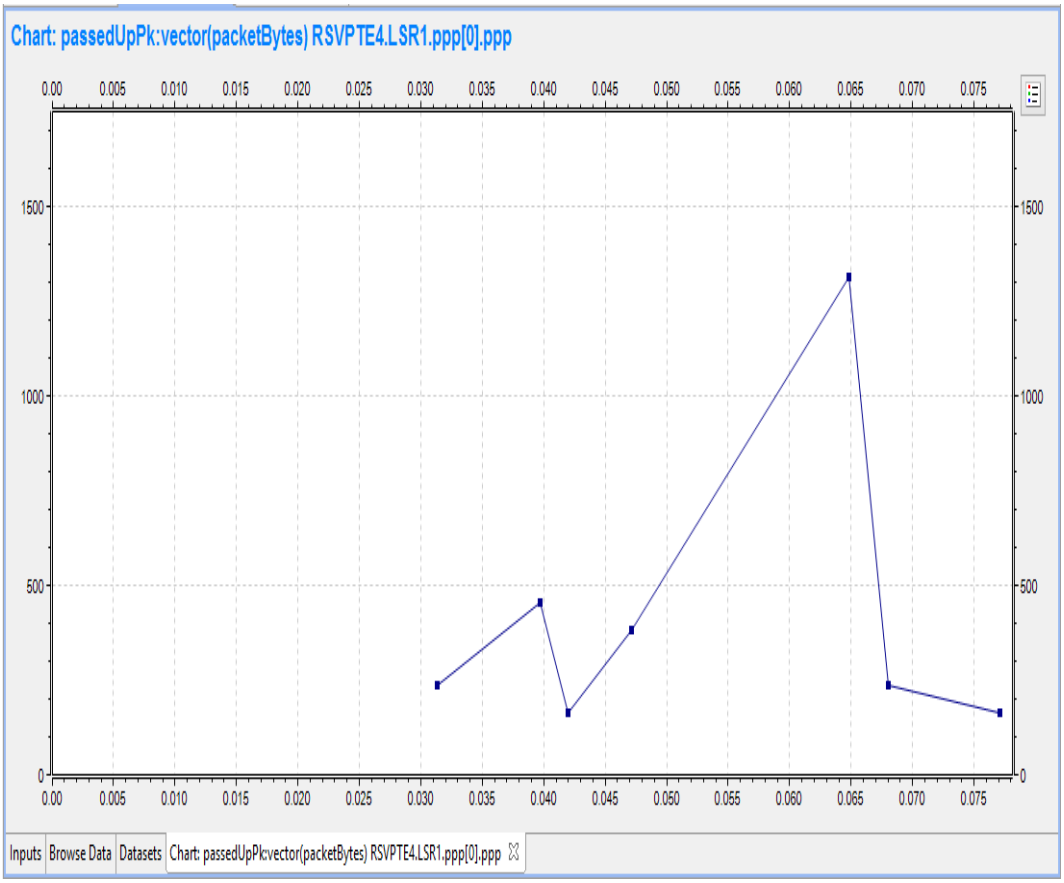
Hasil yang ditampilkan pada interval waktu 20 ms sampai 31 ms dari hasil *broadcast information* yang dilakukan oleh LSR dapat dilihat pada Gambar III-30.



Gambar III-30 Broadcast Information pada interval 20 ms sampai 31 ms

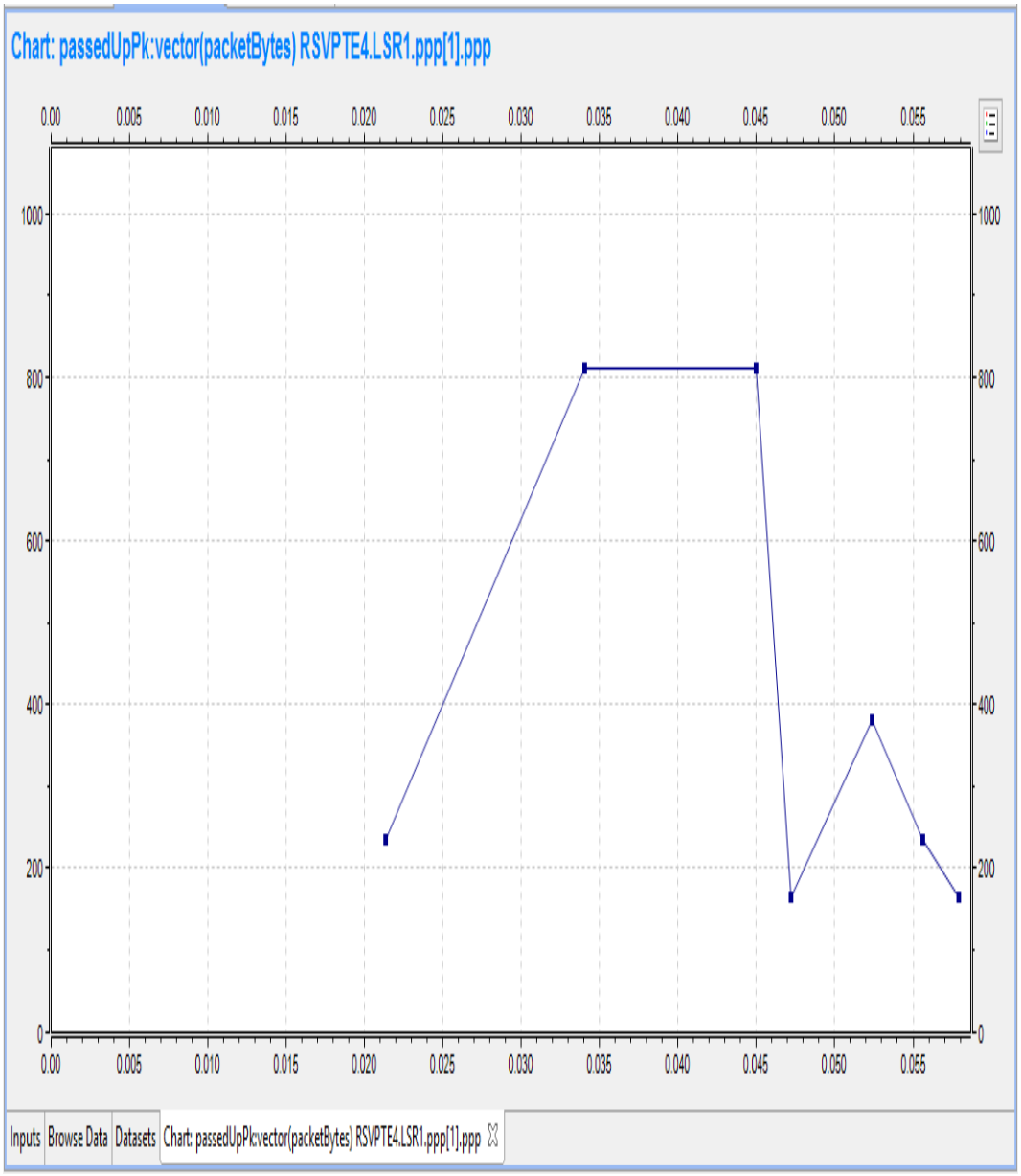
Hasil analisis optimasi protokol *routing* OSPF pada model MPLS dengan melakukan proses *broadcast information* yang disajikan pada Gambar III-29 dan pada Gambar III-30 didapatkan parameter berupa informasi distribusi jalur yang digunakan untuk melewatkan suatu paket data ketika terjadi proses komunikasi data akan dilewatkan keseluruh jalur yang terhubung dengan setiap LSR, kondisi demikian dapat tercipta suatu keseimbangan beban (*load balancing*) data pada jalur komunikasi data. Jalur komunikasi yang digunakan tidak lagi tergantung pada satu jalur komunikasi dengan tingkat jalur terpendek dan tingkat ukuran *bandwidth* besar, tetapi memungkinkan semua jalur yang terhubung akan dilalui untuk melewatkan suatu paket data.

Skenario selanjutnya akan dilakukan suatu proses pengiriman paket data udp untuk melakukan pengukuran tingkat kepadatan data dari suatu jalur pada setiap LSR. Hasil pengukuran yang dilakukan untuk mengetahui tingkat kepadatan pada LSR 1 yang terjadi di *network interface* ppp 0 dapat dilihat pada Gambar III-31.



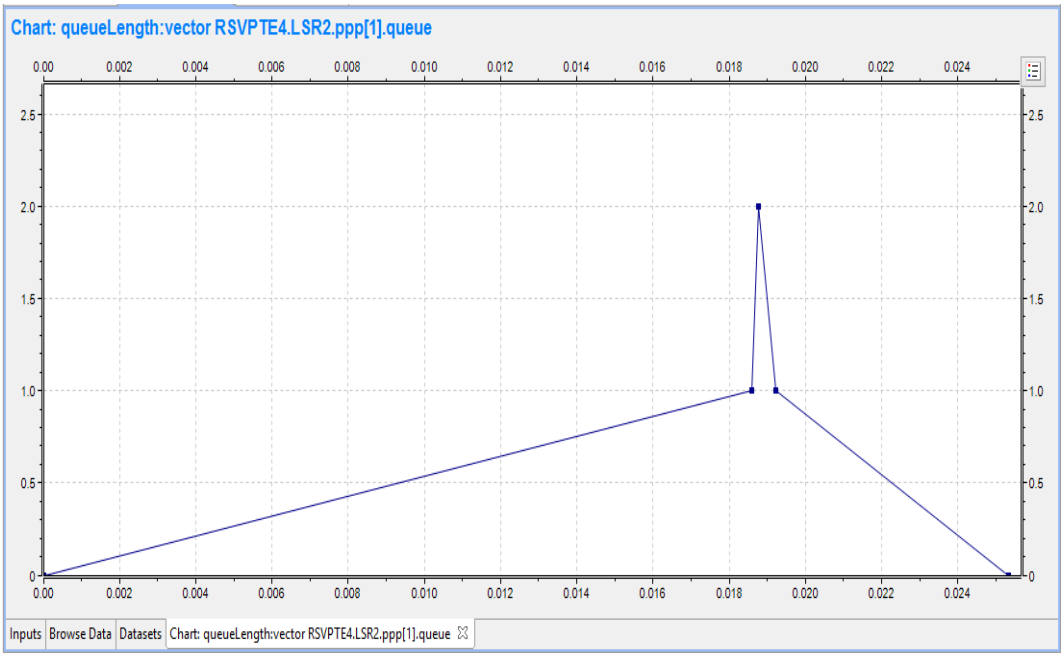
Gambar III-31 Grafik kepadatan jalur pada LSR 1 interface ppp 0

Hasil pengukuran yang dilakukan untuk mengetahui tingkat kepadatan pada LSR 1 yang terjadi di *network interface* ppp 0 dapat dilihat pada Gambar III-32.



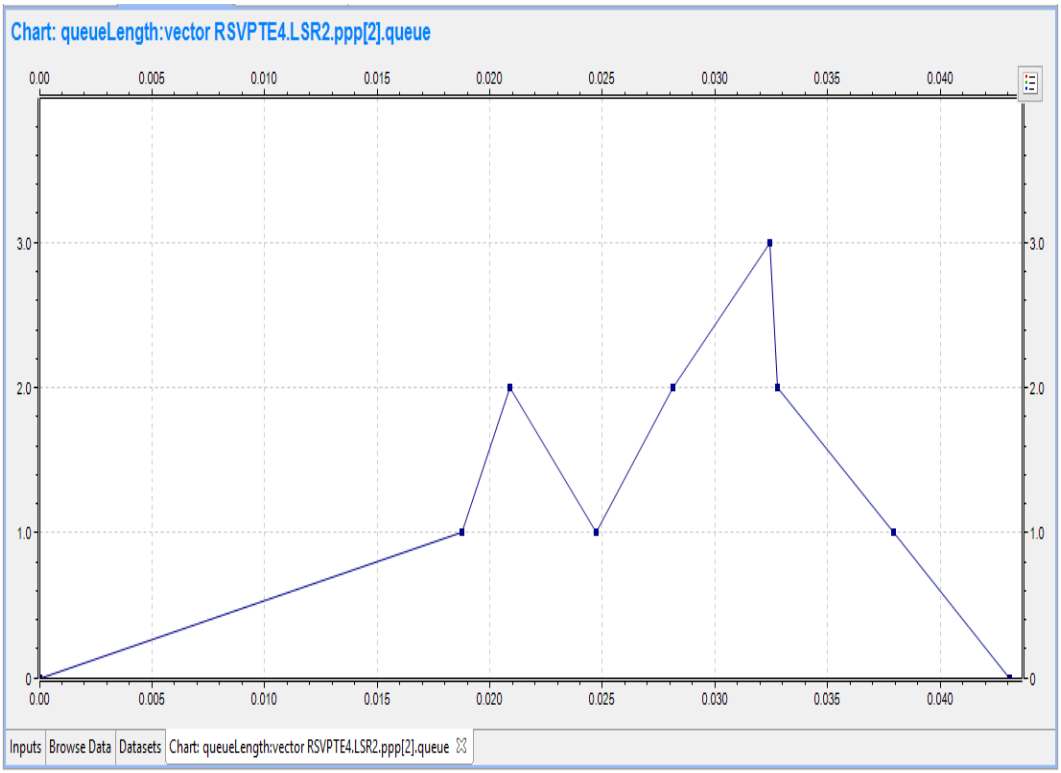
Gambar III-32 Grafik kepadatan jalur pada LSR 1 interface ppp 1

Skenario selanjutnya akan dilakukan pengukuran pada LSR 2 untuk mengetahui panjangnya antrian data yang terjadi di LSR 2 pada network interface ppp 1, hasil dari pengukuran dapat dilihat pada Gambar III-33.



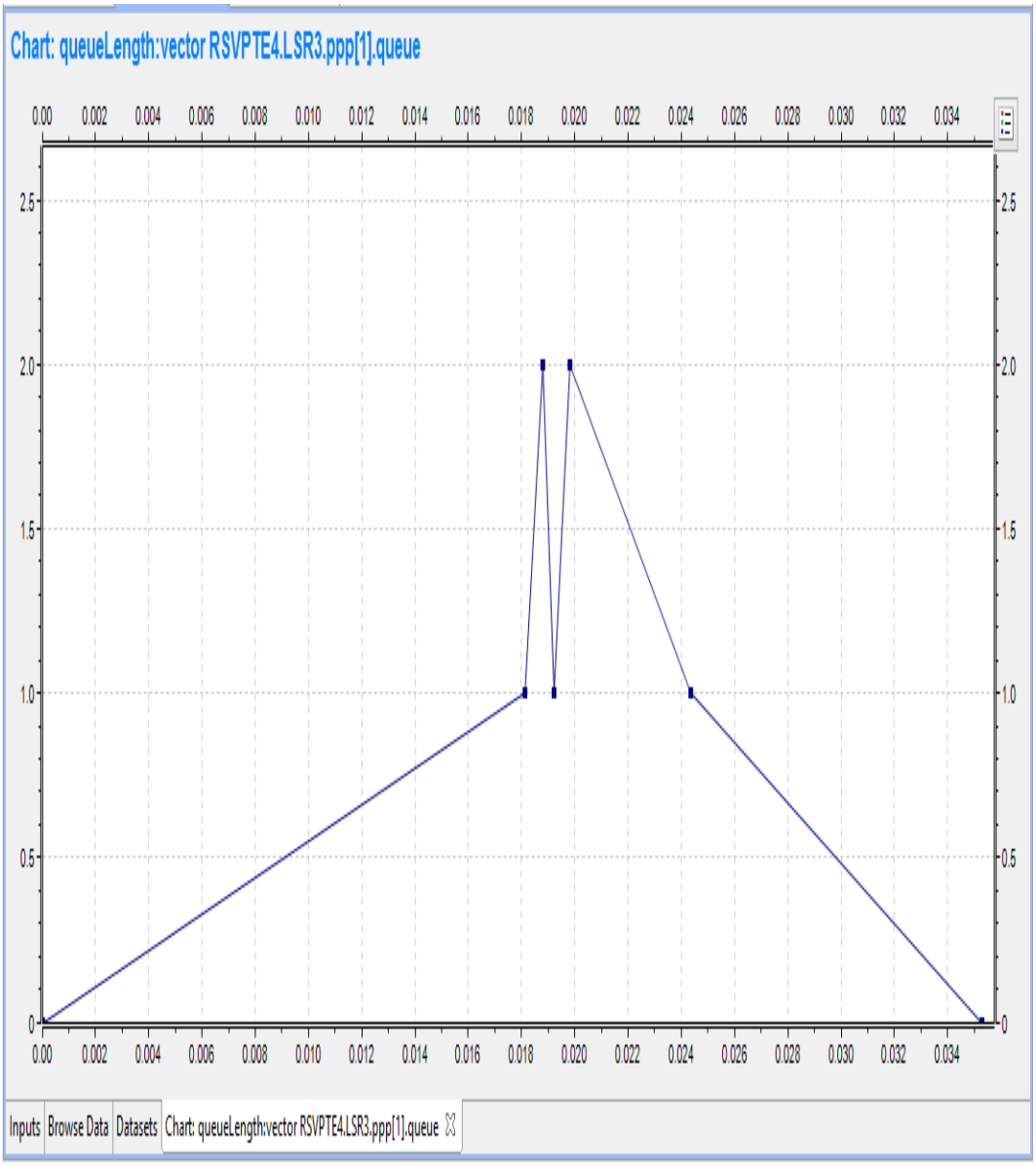
Gambar III-33 Grafik panjang antrian data pada LSR 2 interface ppp 1

Hasil dari pengukuran pada LSR 2 untuk mengetahui panjangnya antrian data yang terjadi di LSR 2 pada network interface ppp 2, hasil dari pengukuran dapat dilihat pada Gambar III-34.



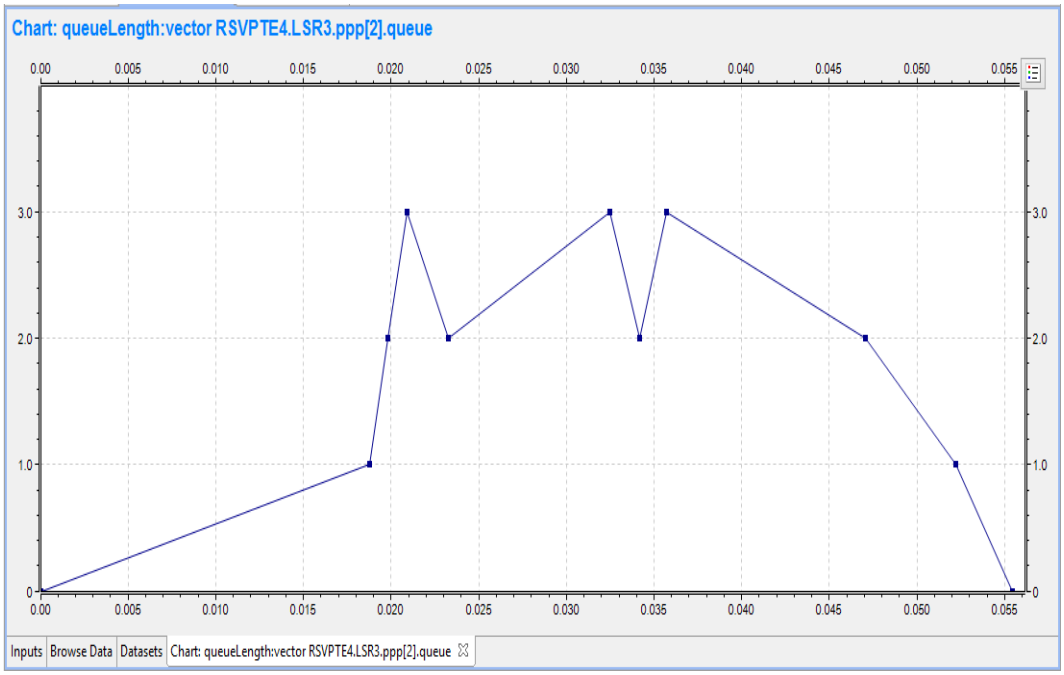
Gambar III-34 Grafik panjang antrian data pada LSR 2 interface ppp 2

Hasil dari pengukuran pada LSR 3 untuk mengetahui panjangnya antrian data yang terjadi di LSR 3 pada network interface ppp 1, hasil dari pengukuran dapat dilihat pada Gambar III-35.



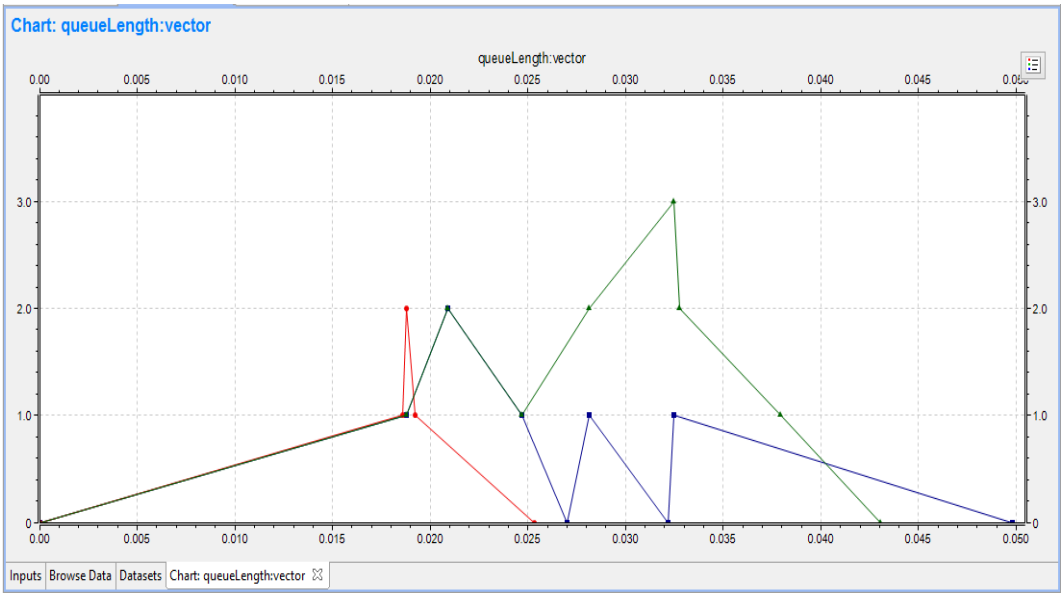
Gambar III-35 Grafik antrian data pada LSR 3 interface ppp 1

Hasil dari pengukuran pada LSR 3 untuk mengetahui panjangnya antrian data yang terjadi di LSR 3 pada network interface ppp 2, hasil dari pengukuran dapat dilihat pada Gambar III-36.



Gambar III-36 Grafik antrian data pada LSR 3 interface ppp 2

Hasil yang ditampilkan pada Gambar III-37 menunjukkan rata-rata panjangnya antrian data yang terjadi di setiap LSR yang terlibat dalam pendistribusian paket data ketika terjadi komunikasi data.



Gambar III-37 Grafik rata-rata antrian pada setiap LSR

Hasil dari analisis yang dilakukan untuk melakukan optimasi protokol *routing* OSPF menggunakan model MPLS yang dilakukan pada pembahasan analisis metode optimasi di sini dapat diketahui penambahan efektifitas mulai dari

penggunaan jalur komunikasi data khususnya ketika pendistribusian jalur setelah dilakukan optimasi menggunakan model MPLS, jalur komunikasi yang digunakan tidak lagi bergantung pada satu jalur saja. Tingkat antrian data yang terjadi pada jalur akan diminimalisir oleh LSR yang diimplementasikan pada model MPLS. Hasil dari pengukuran terhadap setiap LSR yang di analisis sebelumnya menunjukkan grafik penurunan pada antrian data, hal ini terjadi karena setiap LSR akan melakukan pembagian paket data menjadi beberapa *header* paket yang disesuaikan dengan besaran kapasitas kanal *bandwidth* yang dapat mengalirkan suatu paket data di setiap jalur yang terhubung dengan LSR tersebut. Metode optimasi protokol *routing* OSPF menggunakan model MPLS ini sangat dianjurkan untuk membangun infrastruktur jaringan komputer multi jalur, karena dengan menerapkan model MPLS pada jaringan multi jalur akan tercipta suatu keseimbangan beban data (*load balancing*) pada suatu jalur komunikasi data dan meminimalisir terjadinya perubahan kondisi beban data yang dapat berakibat pada kegagalan pada saat proses komunikasi data berlangsung.