

8. KEGIATAN BELAJAR 8 : NETWORK LAYER

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 8 ini siswa diharapkan dapat :

- 1) Memahami Lapisan jaringan (*Network layer*)
- 2) Menganalisis lapisan jaringan (*Network layer*)

b. Uraian Materi

NETWORK LAYER

Lapisan jaringan (*Network layer*) merupakan lapisan ketiga dari standar OSI yang berfungsi untuk menangani masalah jaringan komunikasi secara rinci. Pada lapisan ini, data yang berupa pesan-pesan (*message*) akan dibagi-bagi dalam bentuk paket-paket data yang dilengkapi dengan header-header tertentu pada setiap paket data tersebut.

Network layer ini berfungsi untuk mengambil paket dari sumber dan mengirimkannya ke tujuan. Supaya sampai ditujuan perlu banyak dibuat *hop* pada *router-router* perantara di sepanjang lintasannya. Fungsi *layer* ini sangat kontras dengan fungsi *data link layer*, yang memiliki tujuan lebih sederhana cukup memindahkan *frame* dari ujung kabel yang satu ke ujung yang lainnya. Jadi *network layer* ini merupakan *layer* terbawah yang berkaitan dengan transmisi *end to end*.

Dalam melaksanakan tugasnya, *network layer* harus mengetahui topologi *subnet* komunikasi yaitu *router* secara keseluruhan dan memilih lintasan yang cocok. Pemilihan *router* ini harus hati-hati agar saluran komunikasi dan *router* tidak kelebihan beban, sementara yang lainnya berada dalam keadaan *idle*.

Selain itu bila sumber dan tujuan berada di jaringan yang berbeda, *network layer* bertugas mengatasi terjadinya perbedaan ini dan menyelesaikan masalah – masalah yang merupakan akibat dari adanya perbedaan ini .

Secara umum , lapisan jaringan menyediakan beberapa layanan antara lain : pengendalian operasi subnet , pemilihan rute , pengendalian kemacetan dan *internetworking* .

1. Pengendalian Operasi Subnet

Pada dasarnya , ada dua buah pandangan dalam mengelola *subnet* , yang satu menggunakan koneksi , sedangkan yang lain tidak menggunakan koneksi. Dalam konteks operasi internal *subnet* , suatu koneksi biasanya disebut rangkaian *virtual* , baik secara internal maupun eksternal , sedangkan yang tidak menggunakan koneksi dinamakan *datagram* baik secara internal maupun eksternal.

Rangkaian virtual biasanya digunakan dalam subnet yang layanan utamanya adalah *connection oriented* . didalam rangkaian virtual pemilihan rute baru bagi setiap paket atau sel yang dikirimkan dihindarkan . ketika koneksi telah terbentuk , sebuah rute dari komputer sumber ke komputer tujuan dipilih sebagai bagian dari pembentukan koneksi dan akan selalu diingat .

Sebaliknya , pada *subnet* diagram tidak terdapat rute yang bekerja sebelumnya , walaupun layanannya *connection oriented* . setiap paket yang dikirimkan dirutekan secara *independen* dengan paket sebelumnya . paket berikutnya dapat mempunyai rute yang berbeda . selain *subnet* datagram harus bekerja lebih banyak , *subnet* juga umumnya harus kuat dan lebih mudah dapat menyesuaikan dengan kemacetan dibanding dengan *subnet* rangkaian virtual .

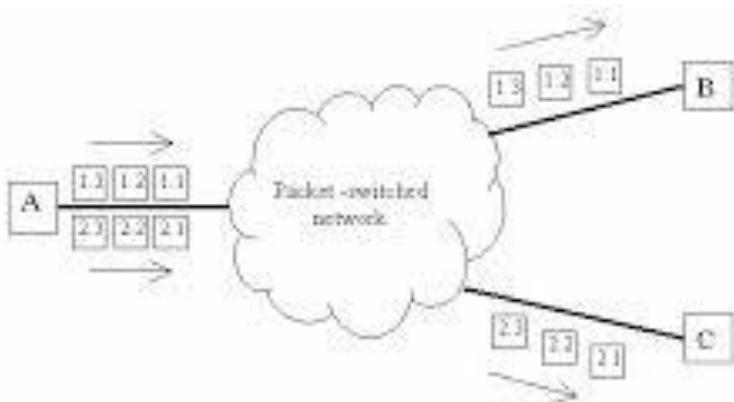
Bila paket yang mengalir melalui rangkaian virtual tertentu selalu mengambil rute yang sama melalui *subnet*, *router* harus mengingat kemana harus meneruskan paket bagi setiap rangkaian virtual terbuka yang melaluiinya. Pada saat koneksi jaringan bentuk, nomor rangkaian virtual yang belum dipakai pada komputer itu dipilih sebagai identifikasi koneksi. Karena rangkaian virtual dapat dimulai dari salah satu ujungnya,

timbul suatu masalah bila pembentukan panggilan berpropogiasi ke dua arah sekaligus sepanjang rantai *router*.

Perlu dicatat, setiap proses diharuskan menandai kapan proses itu melintasi rangkaian virtual, sehingga rangkaian virtual bisa dibersihkan dari tabel *router*. Begitu banyak kegunaan rangkaian internal virtual bagi *subnet*. Terdapat kemungkinan lain, yaitu dengan menggunakan datagram secara internal. Setiap program harus berisi alamat penuh tempat tujuan. Pada jaringan yang besar, alamat ini cukup panjang. Ketika sebuah paket datang, *router* mencari saluran keluar yang akan dipakai dan mengirimkan paket melalui saluran tersebut. Pembentukan dan pelepasan koneksi jaringan atau *transport layer* tidak memerlukan kerja khusus pada router.

1.1 Rangkaian Virtual Eksternal dan Internal

Rangkaian virtual pada dasarnya adalah suatu hubungan seara logika yang dibentuk untuk menyambungkan dua stasiun. Paket dilabelkan dengan nomor sirkuit maya dan nomor urut. Paket dikirimkan dan datang secara berurutan. Gambar 5.1 berikut ini menjelaskan keterangan tersebut.

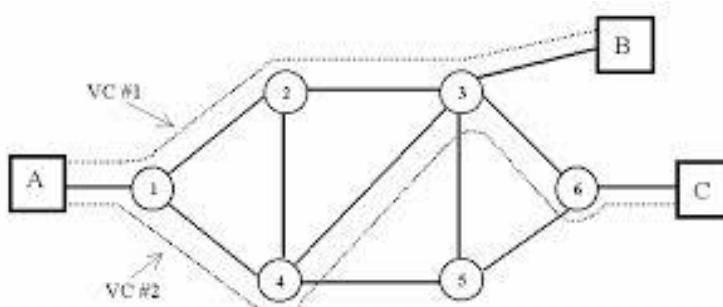


Gambar 1 Rangkaian Virtual Eksternal

Stasiun A mengirimkan 6 paket. Jalur antara logika disebut setara A dan B secara logika disebut sebagai jalur 1, sedangkan jalur antara A

dan C disebut sebagai jalur 2. Paket pertama yang akan dikirimkan lewat jalur 1 dilabelkan sebagai paket 1.1, sedangkan paket ke-2 yang dilewaskan jalur yang sama dilabelkan sebagai paket 1.2 dan paket terakhir yang dilewaskan jalur 1 disebut sebagai paket 1.3. sedangkan paket yang pertama yang dikirimkan lewat jalur 2 disebut sebagai paket 2.1, paket kedua sebagai paket 2.3. dari gambar 5.1 tersebut kiranya jelas bahwa paket yang dikirimkan diberi label jalur yang harus dilewatinya dan paket tersebut akan tiba di stasiun yang dituju dengan urutan pengiriman.

Secara internal rangkaian maya ini bisa digambarkan sebagai suatu jalur yang sudah disusun untuk berhubungan antara satu stasiun dengan stasiun yang lain. Semua paket dengan asal dan tujuan yang sama akan melewati jalur yang sama sehingga akan sampai ke stasiun yang dituju sesuai dengan urutan pada saat pengiriman (FIFO). Gambar 5.2 berikut menjelaskan tentang sirkuitnya internal.



Gambar .2 Rangkaian Virtual Internal

Gambar 5.2 diatas menunjukkan adanya jalur yang harus dilewati apabila suatu paket ingin dikirimkan dari A menuju B, yaitu sirkuit maya 1 atau rangkaian virtual 1 (*Virtual Sircuit 1*) yang disingkat VC #1. Sirkuit ini dibentuk dengan rute melewati node 1-2-3. Sedangkan untuk mengirimkan paket A menuju C dibentuk sirkuit maya VC #2, yaitu rute yang melewati node 1-4-3-6.

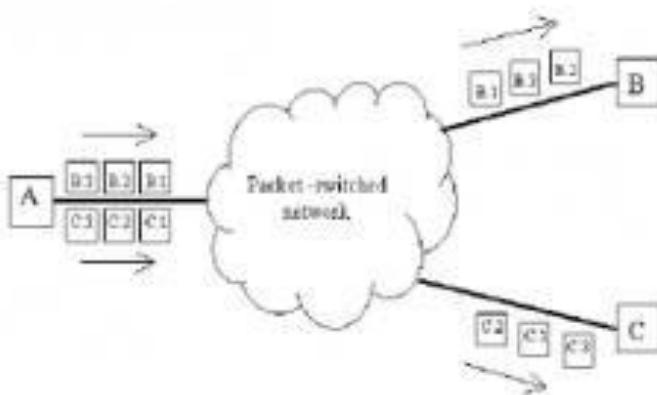
1.2 Datagram Eksternal dan Internal

Dalam bentuk datagram, setiap paket dikirimkan secara *independen*. Setiap paket diberi label alamat tujuan. Berbeda dengan sirkuit maya, datagram memungkinkan paket yang diterima berbeda urutan

dengan urutan saat paket tersebut dikirim. Gambar 5.3 berikut ini akan membantu memperjelas ilustrasi.

Jaringan mempunyai satu stasiun sumber A dan stasiun tujuan yakni B dan C. Paket yang akan dikirimkan ke stasiun B dan ditambah nomor paket sehingga menjadi misalnya B.1, B.37, dan sebagainya. Demikian juga paket yang ditujukan ke stasiun C diberi label yang serupa, misalnya paket C.5, C.17, dan sebagainya.

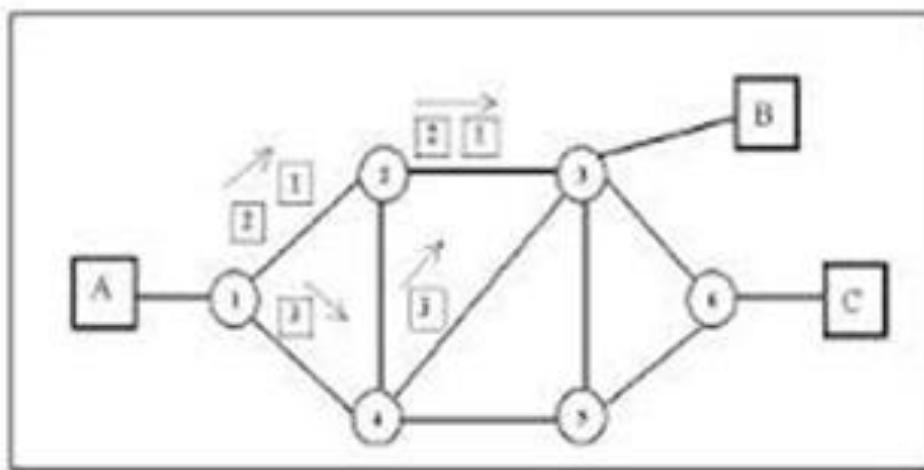
Dari Gambar 5.3, stasiun A mengirimkan enam buah paket. Tiga paket ditujukan ke alamat B. Urutan pengiriman untuk paket B adalah pake B.1, paket B.2 dan paket B.3. sedangkan tiga paket yang dikirimkan ke C masing-masing secara unit adalah paket C.1, paket C.1 dan paket C.3. paket-paket tersebut sampai di B dengan urutan kedatangan B.2, paket B.3 dan terakhir paket B.1 sedangkan di stasiun C, paket-paket tersebut diterima dengan urutan C.3, kemudian paket C.1 dan terakhir paket C.2.



Gambar .3.1 Datagram Eksternal

Ketidak urutan ini lebih disebabkan karena paket dengan alamat tujuan yang sama tidak harus melewati jalur yang sama. Setiap paket bersifat independen terhadap sebuah jalur. Artinya sebuah paket sangat mungkin untuk melewati jalur yang lebih panjang dibanding paket yang lain, sehingga waktu yang dibutuhkan untuk sampai ke alamat tujuan berbeda tergantung rute yang ditempuhnya.

Secara internal datagram dapat digambarkan pada gambar 4 sebagai berikut:



Gambar 3 *Datagram Internal*

1.3 Perbandingan Subnet Rangkaian Virtual dan Datagram

Terdapat perbedaan pendapat dalam penggunaan rangkaian virtual dan datagram. Rangkaian virtual mengijinkan paket berisi nomor dan rangkaian, bukannya alamat penuh tujuan. Bila paketnya pendek, maka alamat penuh tujuan di dalam paket menyebabkan *overhead* yang besar.

Penggunaan rangkaian virtual memerlukan fase pembentukan, yang memakan waktu dan memerlukan sumber daya. Akan tetapi pelaksanaan apa yang harus dikerjakan pada paket dalam rangkaian subnet rangkaian virtual cukup mudah yaitu *router* cukup memakai nomor untuk mengetahui tujuan paket. Pada *subnet* datagram, untuk menentukan kemana paket pergi diperlukan prosedur yang rumit.

Ketika koneksi terbentuk, rangkaian virtual memiliki kelebihan dalam menghindari kemacetan yang terjadi dalam *subnet*, karena sumber daya dapat dipesan sebelumnya. Pada *subnet* datagram cara menghindari kemacetannya lebih rumit. Rangkaian virtual juga memiliki masalah yang kritis. Bila sebuah *router* tabrakan dan kemudian kehilangan memorinya, walaupun *router* kembali pada detik kemudian, maka semua rangkaian virtual yang melaluinya harus dibatalkan. Sebaliknya, bila router gdatagram

mati, hanya para pengguna yang paketnya telah berada dalam antrian pada *router* akan mengalami tabrakan, atau bahkan tidak sama sekali, tergantung apakah paket-paket itu telah diberi *acknowledgement* atau tidak.

Perbandingan secara lengkap antara subnet rangkaian virtual dan datagram dapat dilihat pada Tabel .1 dibawah ini.

Tabel 5.1 *perbandingan antara subnet rangkaian virtual dan datagram*

Persoalan	Datagram Subnet	Rangkaian Virtual Subnet
Pembentukan rangkaian	Tidak diperlukan	Diperlukan
Pengalamatan	Setiap paket berisi sumber penuh dan alamat tujuan	Setiap paket berisi nomor rangkaian virtual yang pendek
Informasi keadaan	Subnet tidak mempunyai informasi keadaan	Setiap rangkaian virtual memerlukan ruang label subnet
Routing	Setiap paket dirutekan secara <i>independen</i>	Rute yang dipilih ketika rangkaian virtual dibentuk seluruh paket mengikuti rute ini
Efek kegagalan niter	Tidak ada, kecuali bagi paket yang hilang pada saat tabrakan	Seluruh rangkaian virtual yang dilewatkan melalui ruter yang gagal dihentikan
Kontrol kemacetan	Sulit	Cukup mudah bila buffer yang cukup dapat dialokasikan terlebih dahulu untuk masing-masing rangkaian virtual

2 .Pemilihan Route (Routing)

salah satu fungsi dari *network layer* adalah mencari rute untuk jalur transmisi paket data dari komputer sumber ke komputer tujuan. Dalam sebagian besar subnet, paket-paket data akan memerlukan banyak lompatan dalam melakukan perjalanan. Algoritma yang memilih rute dan struktur data yang digunakan jaringan merupakan masalah utama rancangan *network layer*.

2.1 Algoritma Routing

Algoritma routing adalah bagian algoritma dari perangkat lunak *network layer* yang bertanggung jawab untuk menentukan jalur mana yang menjadi jalur transmisi paket. Jika subnet tersebut menggunakan datagram secara internal, keputusan ini harus selalu dibuat setiap kali paket data datang. Tetapi, jika *subnet* tersebut menggunakan rangkaian virtual secara internal , keputusan *routing* ini hanya akan dibuat pada waktu penetapan rangkaian virtual yang terbaru. Sesudah itu, paket data tinggal mengikuti rute yang telah ditetapkan sebelumnya.

Setiap algoritma *routing* memiliki sifat-sifat seperti kebenaran, kesederhanaan, kekokohan, kestabilan,kewajaran, dan optimalitas. Algoritma *routing* harus dapat menyesuaikan diri atau bertahan terhadap perubahan-perubahan dalam topologi dan lalu lintas data.

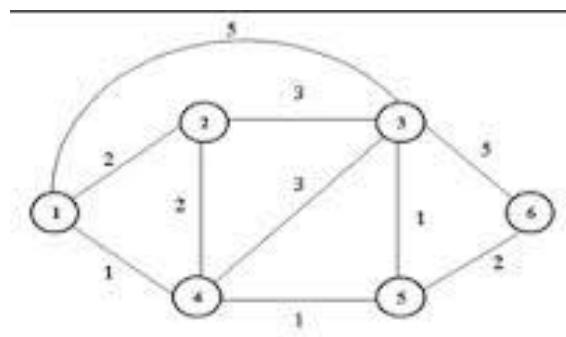
Untuk mencari rute dengan biaya minimum, dapat digunakan dua metode yaitu metode *forward search algorithm* dan *backward search algorithm*.

2.1.1.Forward Search Algorithm

Forward Search Algorithm digunakan untuk menentukan jarak terpendek dari node awal yang ditentukan ke setiap node yang ada. Algoritma diungkapkan dalam stage. Dengan k buah jalur terpendek node k terhadap node sumber ditentukan. Node-node ini ada dalam himpunan N. Pada stage ke (k+1), node yang tidak ada dalam M yg mempunyai jarak terpendek terhadap sumber ditambahkan ke M. Sebagai sebuah node yang di tembarukkan dalam M, maka jalur dari sumber menjadi terdefinisi (Gambar 5.4). Algoritma ini memiliki 3 tahapan:

1. Tetapkan $M=\{S\}$. Untuk setiap node $n \in N-S$, tetapkan $C_1(n)=1(S,n)$.
2. Cari $W \in M$ sehingga $C_1(W)$ minimum dan tambahkan ke M . Kemudian $C_1(n) = \min[C_1(n), C_1(W) + 1(W,n)]$ untuk tiap node $n \in N-M$. Apabila pernyataan terakhir bernilai minimum, jalur dari S ke n sebagai jalur S ke W menotong *link* dari W ke n .
3. Ulang langkah 2 sampai $M=N$. Keterangan:
 - N = himpunan *node* dalam jaringan
 - S = *node* sumber
 - M = himpunan node yang dihasilkan oleh algoritma
 - $C_1(I,J) = \text{link cost}$ dari *node I* sampai *node J*, biaya bernilai $>$ jika node tidak secara langsung terhubung.
 - $C_1(n) = \text{biaya dari jalur biaya terkecil dari } S \text{ ke } n \text{ yang dihasilkan pada saat algoritma dikerjakan.}$

Untuk memperjelas dari penggunaan *forward search algorithm*, perhatikan Gambar.4 yang menjelaskan rute jaringan yang menghubungkan 6 titik (*node*).



Gambar .4 Rute Jaringan Pada 6 Titik

Dengan menggunakan $S=1$ dan berdasarkan gambar diatas, diperoleh hasil dari *forward search algorithm* yang tertuang pada Gambar 5.5.

Iterasi	M	$C_1(2)$	Path	$C_1(3)$	Path	$C_1(4)$	Path	$C_1(5)$	Path	$C_1(6)$	Path
1	{1}	2	1-2	5	1-4	1	1-4	∞	--	∞	--
2	{1,4}	2	1-2	4	1-4	1	1-4	2	1-4-5	∞	--
3	{1,2,4}	2	1-2	4	1-4	1	1-4	2	1-4-5	∞	--
4	{1,2,4,5}	2	1-2	3	1-4	1	1-4	2	1-4-5	4	1-4-5-6
5	{1,2,3,4,5}	2	1-2	3	1-4	1	1-4	2	1-4-5	4	1-4-5-6
6	{1,2,3,4,5,6}	2	1-2	3	1-4	1	1-4	2	1-4-5	4	1-4-5-6

Gambar .5 Hasil Forward Searc Algoritm**2.1.1.Backward Search Algorithm**

Digunakan untuk menentukan jalur biaya terkecil yang diberikan node tujuan dari semua node yang ada (Gambar 5.6). Algoritma ini juga diproses tiap *Stage*. Pada tiap *stage*, algoritma menunjuk masing-masing *node*.

Definisi yang digunakan:

- N = Himpunan node yang terdapat pada jaringan
- D = node tujuan
- $l(i,j)$ = seperti keterangan diatas
- $C_2(n)$ = biaya dari jalur biaya terkecil dari n ke D yang dihasilkan saat algoritma dikerjakan.

Algoritma ini juga terdiri dari 3 tahapan:

1. Tetapkan $C_2(D)=0$. Untuk tiap node $n \in N-D$, tetapkan $C_2(n)=\infty$.
2. Untuk tiap node $n \in N-D$, tetapkan $C_2(n)=\min_{W \in N} [C_2(N), C_2(W) + l(n,W)]$. Apabila pada pernyataan terakhir bernilai minimum, maka jalur dari n ke D saat ini merupakan link dari n ke W dan menggantikan jalur dari W ke D .
3. Ulangi langkah ke-2 sampai tidak ad cost yang berubah.

Gambar 5.6 berikut ini adalah hasil pengolahan Gambar 5.4 dengan $D=1$.

Iterasi	$C_2(2)$	Path	$C_2(3)$	Path	$C_2(4)$	Path	$C_2(5)$	Path	$C_2(6)$	Path
1	∞	--	∞	--	∞	--	∞	--	∞	--
2	2	2-1	5	3-1	1	4-1	2	5-4-1	4	6-5-4-1
3	2	2-1	3	3-5-4-1	1	4-1	2	5-4-1	4	6-5-4-1
4	2	2-1	3	3-5-4-1	1	4-1	2	5-4-1	4	6-5-4-1

Gambar .6 Hasil Backward Search Algoritm

2.2 Strategi Routing

Dalam mencari rute bagi paket yang dikirim dari komputer sumber ke komputer tujuan ada beberapa strategi yang dipakai. Strategi itu meliputi fixed routing, flooding, random routing, dan adaptive routing.

2.2.1 Fixed Routing

Merupakan cara routing yang paling sederhana. Dalam hal ini rute bersifat tetap atau paling tidak, rute hanya diubah apabila topologi jaringan berubah. Gambar 5.7 berikut (mengacu dari Gambar 5.4) memperlihatkan bagaimana sebuah rute yang tetap dikonfigurasikan.

		CENTRAL ROUTING DIRECTORY					
		To node					
		1	2	3	4	5	6
From Node	1	-	2	4	4	4	4
	2	1	-	3	4	4	4
	3	5	2	-	5	5	5
	4	1	2	5	-	5	5
	5	4	4	3	4	-	6
	6	5	5	5	5	5	-

Gambar .7 Direktori Untuk Fixed Routing

Kemungkinan rute yang bisa dikonfigurasikan, dirunjukkan pada Gambar .8 sebagai berikut:

Node 1 Directory		Node 2 Directory		Node 3 Directory	
Destination	Next node	Destination	Next node	Destination	Next node
2	2	1	1	1	5
3	4	2	3	2	2
4	4	3	4	4	5
5	4	4	4	5	5
6	4	5	4	6	5

Node 4 Directory		Node 5 Directory		Node 6 Directory	
Destination	Next node	Destination	Next node	Destination	Next node
1	1	1	4	1	5
2	2	2	4	2	5
3	5	3	3	3	5
4	5	4	4	4	5
5	5	5	6	5	5

Gambar .8 Direktori Untuk Masing-Masing Node

Tabel pada Gambar 5.8 disusun berdasarkan rute terpendek dengan menggunakan least-cost algorithm. Sebagai misal direktori node 1. Dari node 1 untuk mencapai node 6, maka rute terpendek yang bisa dilewati adalah rute dari node 1,4,5,6. Maka pada tabel direktori node 1 dituliskan destination = 6, dan next node = 4.

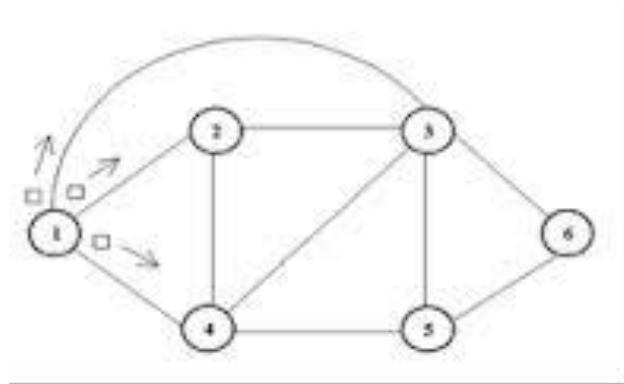
Keuntungan konfigurasi dengan rute tetap semacam ini adalah bahwa konfigurasi menjadi sederhana. Penggunaan sirkuit aya atau datagram tidak dibedakan. Artinya semua paket dari sumber menuju titik tujuan akan melewati rute yang sama . kinerja yang bagus terdapat apabila beban bersifat tetap. Tetapi pada beban yang bersifat dinamis, kinerja menjadi turun. Sistem ini tidak memberi tanggapan apabila terjadi error maupun kemacetan jalur.

2.2.2 Flooding

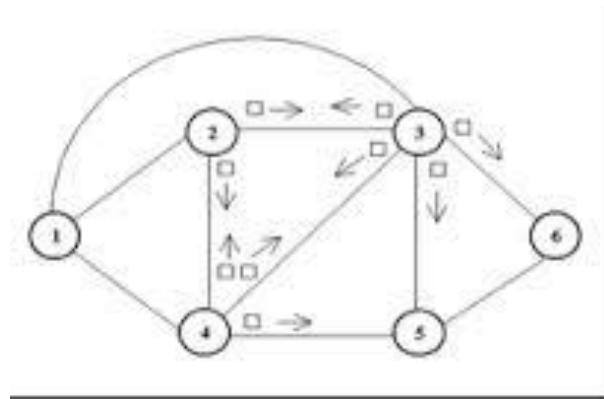
Teknik routing yang lain yang dirasa sederhana adalah flooding. Cara kerja teknisi ini adalah mengirimkan paket dari suatu sumber ke seluruh node tetangganya. Pada tiap ode, setiap paket yang datang akan ditransmisikan kembali ke seluruh link yang dipunyai kecuali link yang dipakai untuk menerima paket tersebut. Mengambil dari contoh yang sama, sebutlah bahwa node 1 akan mengirimkan paketnya ke node 6. Pertama

kali node 1 akan mengirimkan paket ke seluruh tetangganya, yakni ke node 2, node 4 dan node 5 (Gambar 5.9).

Selanjutnya operasi terjadi pada node 2, node 3 dan node 4. Node 2 mengirimkan paket ke tetangganya yaitu node 3 dan node 4. Sedangkan node 3 meneruskan paket ke node 2, node 4, node 5 dan node 6. Node 4 meneruskan paket ke node 2, node 3, node 5. Semua node ini tidak mengirimkan paket ke node 1. Ilustrasi tersebut digambarkan pada Gambar 5.10.



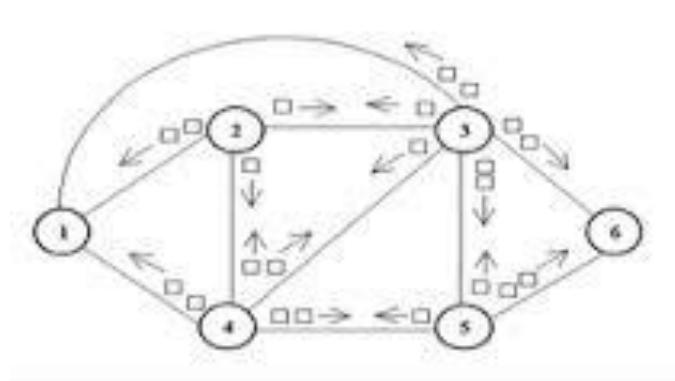
Gambar 5.9 Hop Pertama



Gambar 5.10 Hop Kedua

Pada saat ini jumlah salinan yang diciptakan berjumlah 9 buah. Paket-paket yang sampai ketitik tujuan, yakni node 6, tidak lagi diteruskan. Posisi terakhir node-node yang menerima paket dan harus meneruskan adalah node 2, node 3, node 4, node 5. Dengan cara yang sama masing-masing node tersebut membuat salinan yang memberikan dan ke node

tetangganya. Pada saat ini ainasilkan salinan sebanyak 22 (lihat Gambar 5.11).



Gambar 5.11 Hop Ketiga

Terdapat dua catatan penting dengan penggunaan teknik flooding ini, yaitu:

1. Semua rute yang dimungkinkan akan dicoba. Karena itu teknik ini memiliki keandalan yang tinggi dan cenderung memberi rioritas untuk pengiriman-pengiriman paket tertentu.
2. Karena keseluruhan rute dicoba, maka akan muncul paling tidak satu buah salinan paket dititik tujuan dengan waktu paling minimum. Tetapi hal ini akan menyebabkan naiknya beban lalu lintas yang pada akhirnya menambah delay bagi rute-rute secara keseluruhan.

2.2.3 Random Routing

Prinzip utama dari teknik ini adalah sebuah *node* memiliki hanya satu jalur keluaran untuk menyalurkan paket yang datang kepadanya. Pemilihan terhadap sebuah jalur keluaran bersifat acak. Apabila *link* yang akan dipilih memiliki bobot yang sama, maka bisa dilakukan dengan pendekatan seperti teknik *round robin*.

Routing ini adalah untuk mencari probabilitas untuk tiap-tiap outgoing *link* dan memilih *link* berdasarkan nilai probabilitasnya. Probabilitas bisa dicari berdasarkan data rate, dalam kasus ini didefinisikan sebagai berikut:

$$P_i = \frac{R_i}{\sum_i R_i}$$

Dimana:

P_i = probabilitas pemilihan i

R_j = data rate pada link j

Penjumlahan dilakukan untuk keseluruhan link outgoing. Skema seperti ini memungkinkan distribusi lalu lintas yang baik. Seperti teknik flooding, random routing tidak memerlukan informasi jaringan, karena route akan dipilih dengan cara random.

2.2.4 Adaptive Routing

Strategi routing yang dibahas di depan, tidak mempunyai reaksi terhadap perubahan kondisi yang terjadi didalam suatu jaringan. Untuk itu pendekatan dengan atrategi adaptif mempunyai kemampuan yang lebih dibandingkan dengan beberapa hal di atas.

Dua hal yang penting yang menguntungkan adalah:

- Strategi routing adaptif dapat meningkatkan kinerja seperti apa yang diinginkan user.
- Strategi adaptif dapat membantu kendali lalu lintas.

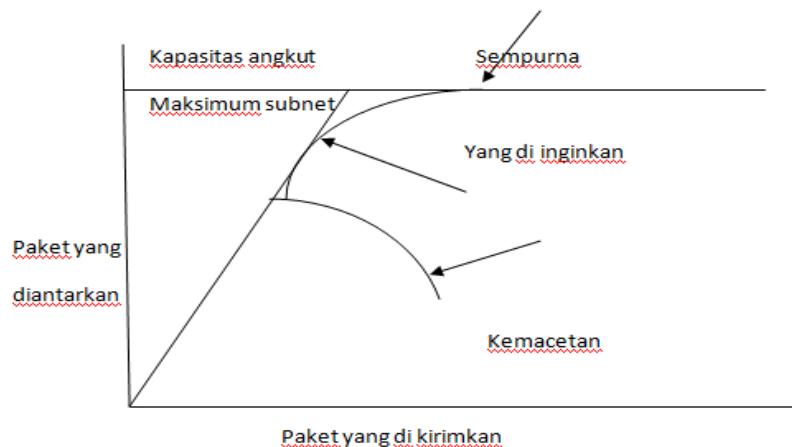
Akan tetapi, strategi ini dapat menimbulkan beberapa akibat, misalnya:

- Proses pengambilan keputusan untuk menetapkan rute menjadi sangat rumit akibatnya beban pemrosesan pada jaringan meningkat.
- Pada kebnyakan kasus, strategi adaptif tergantung pada informasi status yang dikumpulkan pada satu tempat tetapi digunakan ditempat lain. Akibatnya beban lalu lintas meningkat.
- Strategi adaptif bisa memunculkan masalah seperti kemacetan apabila reaksi yang terjadi terlalu cepat, atau menjadi tidak relevan apabila reaksi sangat lambat.

3. kemacetan

Bila terlalu banyak paket yang berada di dalam *subnet*, maka unjuk kerja jaringan akan mengalami penurunan (Gambar 5.12). situasi seperti ini disebut keacetan (*congestion*). Bila jumlah paket yang mengalir ke dalam

subnet dari host masih berada dalam daya tampungnya, paket-paket tersebut seluruhnya akan dihantarkan. Jumlah paket yang dihantarkan proporsional dengan jumlah paket yang dikirimkan. Akan tetapi dengan semakin meningkatnya lalu lintas, *router* tidak mampu lagi menangani paket yang datang dan *router* akan memulai kahilangan paket.



Gambar .12 Kemacetan

Kemacetan bisa disebabkan oleh beberapa faktor. Bila semuanya terjadi dengan tiba-tiba , aliran paket yang datang pada tiga atau empat saluran input dan semuanya memerlukan saluran output yang sama, maka antrian mulai membesar. Bila tidak terdapat memori yang cukup untuk menampung seluruh antrian, maka paket akan hilang.

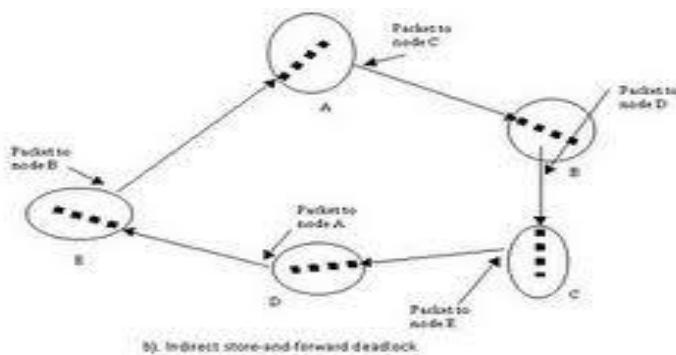
Proses yang lambat juga dapat menimbulkan kemacetan. Bila CPU router lambat dalam melakukan tugas yang diperlukan, antrian akan menjadi semakin panjang. Permasalahan yang serius yang diakibatkan efek congestion adalah deadlock, yaitu suatu kondisi dimana sekelompok node tidak bisa meneruskan pengiriman paket karena tidak ada buffer yang tersedia. Trknik deadlock avoidance digunakan untuk merancang jaringan sehingga deadlock tidak terjadi.

Bentuk deadlock yang paling sederhana adalah direct store-and-forward deadlock. Pada Gambar 5.13 memperlihatkan situasi bagaimana antara node A dan node B berinteraksi di mana kedua buffer penuh dan deadlock terjadi.



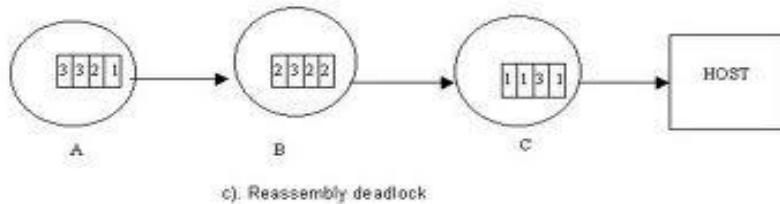
Gambar .13 Direct store and forward deadlock

Bentuk deadlock kedua adalah indirect store-and-forward deadlock (Gambar 5.14). hal ini terjadi tidak pada sebuah link tunggal seperti bentuk deadlock di atas. ^pada tiap node, antrian yang ditujukan untuk node terdekatnya bersifat searah dan J menjadi penuh.



Gambar .14 Indirect stote and forward deadlock

Bentuk deadlock yang ketiga adalah reassembly deadlock. Situasi ini digambarkan pada Gambar 5.15 dimana node C memiliki 4 paket terdiri dari paket 1 tiga buah dan sebuah paket 3. Seluruh buffer penuh dan tidak lagi menerima paket baru.



Gambar .15 Reassembly deadlock

4. Internetworking

Ketika dua atau lebih jaringan bergabung dalam sebuah aplikasi, biasanya kita sebut ragam kerja antar sistem seperti ini sebagai sebuah internetworking. Penggunaan istilah internet-work (atau juga sebuah intrnet0 mengacu pada perpaduan jaringan, misalnya LAN-MAN-WAN, yang digunakan. Masing-masing jaringan (LAN atau WAN) yang terlibat dalam internet work disebut sebagai subnetwork atau subnet.

Piranti yang digunakan untuk menghubungkan antara dua jaringan, meminjam istilah ISO, disebut sebagai Intermediate System (IS) atau sebuah Internetworking Unit (IWU). Selanjutnya apabila fungsi utama dari sebuah intermmmediate system adalah melakukan routing, maka piranti yang dimaksud disebut sebagai router, sedangkan apabila tugas piranti adalah menghubungkan antara dua tipe jaringan disebut sebagai gateway.

Selain menggunakan gateway dan router, piranti yang juga digunakan untuk perantara antar segmen jaringan yang berhubungan adalah repeater dan bridge.

4.1 Repeater

Repeater pada dasarnya merupakan alat yang sederhana yang berfungsi untuk memperbaiki dan memperkuat sinyal yang melewatkinya. Dua sub jaringan yang dihubungkan oleh perangkat ini memiliki protokol yang sama untuk semua lapisan.

Repeater juga berfungsi untuk memperbesar batasan panjang satu segmen. Berikut ini adalah contoh beberapa jenis sistem yang menggunakan repeater untuk memperbaiki dan memperkuat sinyal transmisi data:

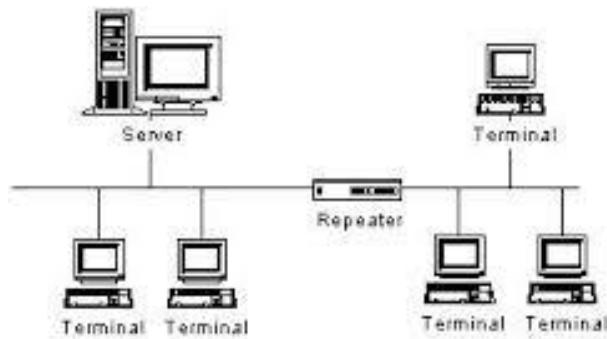
1. Sistem *baseband* bertopologi *bus*

media transmisi yang paling populer pada sistem baseband bertopologi bus (Gambar 5.16) adalah kabel koaxial, yang juga digunakan sebagai standart IEEE 802.3. batasan maksimum jumlah repeater yang dapat digunakan untuk satu segmen adalah empat. Berikut ini adalah tabel jenis kabel koaxial dengan spesifikasinya (Tabel 5.2).

Tabel .2 Tabel Jenis Coaxial Dan Spesifikasinya

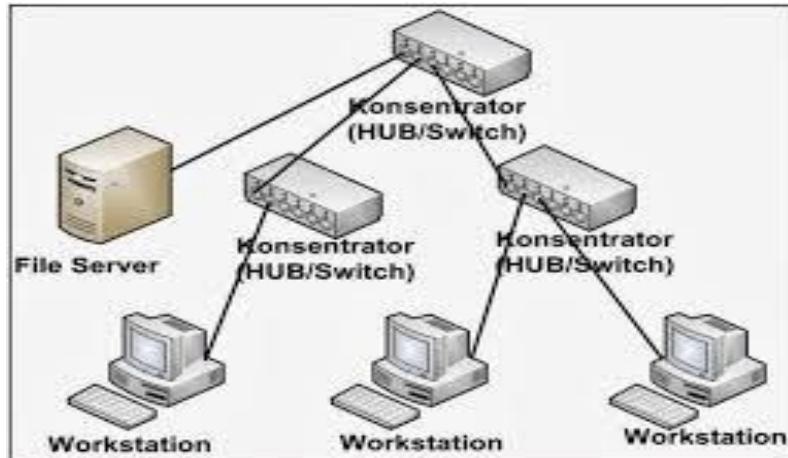
Sifat	10 Base 5	10 Base 2
Kecepatan	10 Mbps	10 Mbps
Panjang segmen maksimum	500 m	200 m
Bentang jaringan maksimum	2500 m	1000 m
Simpul per segmen	100	30
Selang antar simpul	2.5 m	0.5 m
Diameter kabel	0.4 in	0.25 in

Pada gambar .16 menunjukkan *repeater* pada topologi bus.

**Gambar .16** Repeater pada topologi bus

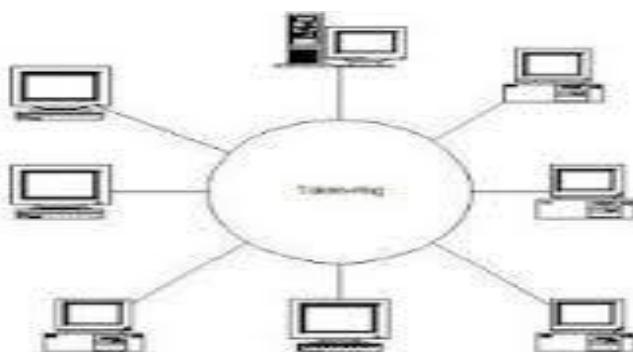
2. Sistem baseband bertopologi star

sistem yang dimaksud disini adalah sistem yang secara fisik bertopologi star, tetapi secara logika bertopologi bus, seperti yang terlihat pada gambar 5.17. susunan jaringan yang tergambar menggunakan kabel *twisted pair*, dengan protokol Ethernet. Kabel jenis ini disebut 10 Base T, yang berkecepatan 10 Mbps dan memiliki panjang segmen maksimum 100 m.



Gambar .17 Repeater pada topologi star

3. Sistem baseband bertopologi ring pada sistem standart IEEE 802.5, repeater digunakan pada setiap simpul (*node*) jaringan seperti yg ditunjukkan pada Gambar 5.18.

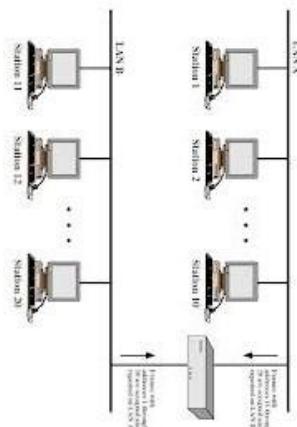


Gambar .18 Repeater pada topologi ring

.4.2 Bridge

Bridge adalah jenis perangkat antara yang menghubungkan dua jaringan yang protokol lapisan fisiknya berbeda. Hal ini berarti komunikasi terjadi pada level MAC (lapisan data link bagian bawah) yang serupa. Sebagai contoh adalah *bridge* untuk menghubungkan IEEE 802.3 (Ethernet) dengan IEEE 802.4 (Token Bus). Untuk lebih jelasnya perhatikan Gambar 5.19.

Bridge memiliki sifat yang tidak mengubah isi maupun bentuk *frame* yang diterimanya, disamping itu *bridge* memiliki *buffer* yang cukup untuk menghadapi ketidak sesuaian kecepatan pengiriman dan penerimaan data.



Gambar .19 Bridge

4.2.1 Alasan Penggunaan Bridge

Beberapa alasan mengapa bridge digunakan untuk menghubungkan beberapa LAN adalah sebagai berikut:

❖ Ketentuan LAN

Hal ini berkaitan erat dengan:

- Jumlah maksimum stasiun
- Panjang maksimum stasiun
- Bentang jaringan (*Network span*)

❖ Kehandalan dan keamanan lalu lintas data

Bridge dapat menyaring lalu lintas data antar dua segmen jaringan

❖ Unjuk kerja

Semakin besar LAN (jumlah stasiun maupun jarak), unjuk kerja semakin menurun.

❖ Keterpisahan geografis

Bila dua sistem pada tempat yang berjauhan disambungkan, penggunaan *bridge* dengan saluran komunikasi jarak jauh (misalnya radio atau gelombang mikro) jauh lebih masuk akal dibandingkan langsung dua sistem tadi dengan kabel *coaxial* misalnya.

4.2.2 Penggolongan Bridge

Dari sudut kelengkapan fungsi, perangkat ini dapat digolongkan menjadi tiga macam, yaitu:

1. *Bridge* sederhana

Bila suatu simpul jaringan mengirimkan data ke simpul jaringan lain, maka *bridge* sederhana akan menyebarkan data tersebut kesemua jaringan.

Bridge sederhana, memiliki urutan kerja sebagai berikut:

- Baca semua paket data yang datang dari suatu jaringan.
- sebarkan ke semua simpul jaringan yang lain.

2. Bridge belajar

jenis ini memiliki kemampuan memilih paket mana yang ditunjukkan pada segmen lain jaringan, dan meneruskan paket tersebut pada jaringan yang sesuai tersebut. Hal ini dimungkinkan karena protokol lapisan MAC memang terdapat field alamat tujuan paket. Kini *bridge* sederhana juga telah dilengkapi dengan kemampuan belajar tersebut.

Bridge belajar memiliki urutan kerja sebagai berikut:

- Baca semua paket data yang datang dari suatu jaringan.
- Pilih dan terima semua paket data yang tidak dialamatkan untuk jaringan pertama tadi.
- Kirimkan (teruskan) paket data yang diterima tadi ke jaringan lain yang terhubung pada *bridge*.

3. Bridge dengan kemampuan pencarian jalan (routing)

jenis ini juga memiliki kemampuan jenis sebelumnya, ditambah dengan kemampuan pencarian jalan.

Pada bridge yang mempunyai fasilitas pencari jalan, terdapat beberapa strategi yang digunakan, antara lain:

- *Fixed routing*. Pada cara ini dibuat sebuah tabel yang berisi semua jalur yang mungkin terjadi antara suatu stasiun pengirim

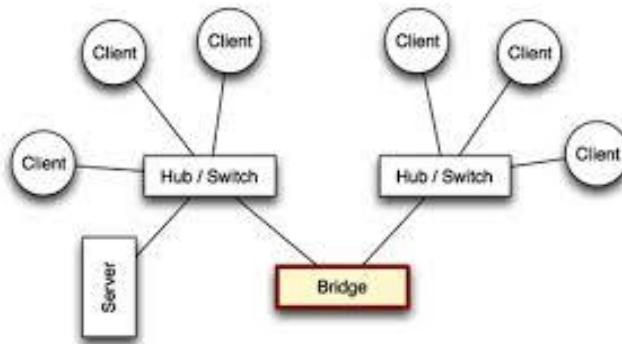
dan penerima. Cara ini relatif mudah untuk sistem jaringan yang sederhana.

- Penggunaan algoritma *spanning tree*, dengan menganggap LAN sebagai simbol (*node*) *graph* dan *bridge* sebagai sis (*edge*) *graph*. Dengan algoritma ini, dibangun bentuk *spanning tree* dari suatu *graph*, yaitu *graph* yang tidak memiliki putaran (*closed loop*).
- *Source routing*. Pada cara ini, setiap stasiun yang akan mengirimkan paket, harus mendefinisikan jalur yang harus ditempuh. Dengan demikian alamat semua stasiun tujuan harus tercatat pada stasiun sumber. Informasi jalur ini dimasukkan ke dalam protokol lapisan MAC.

Dari sudut jangkauan, perangkat ini dapat dikelompokkan menjadi dua bagian,yaitu:

1. *Bridge setempat (Local Bridge)*

Jenis ini tersambung langsung pada dua jaringan yang dihubungkan. Biasanya jenis ini digunakan untuk menghubungkan dua jaringan yang letaknya relatif dekat. Untuk lebih jelasnya perhatikan Gambar 5.20.

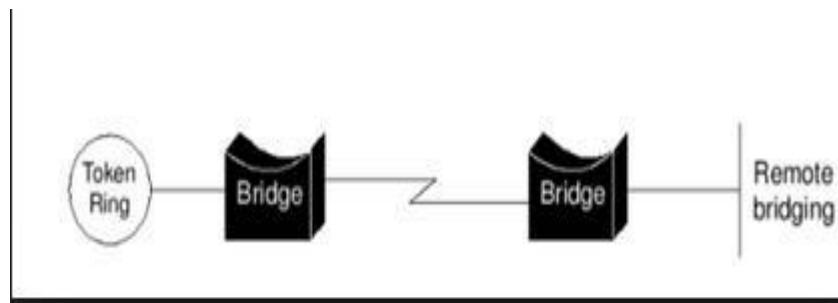


Gambar .29 *Bridge Setempat*

2. *Bridge jarak jauh*

Bridge jenis ini, terdapat pada dua segmen jaringan. Kedua *bridge* jenis ini dihubungkan dengan saluran komunikasi tertentu. Dengan demikian *bridge* jenis ini selalu bekerja berpasangan. Pasangan *bridge* ini umumnya digunakan untuk menghubungkan dua jaringan

yang letak geografisnya berjauhan. Secara logika, fungsi pasangan *bridge* ini sama saja dengan satu *bridge* setempat. Untuk lebih jelasnya Perhatikan Gambar 5.21.



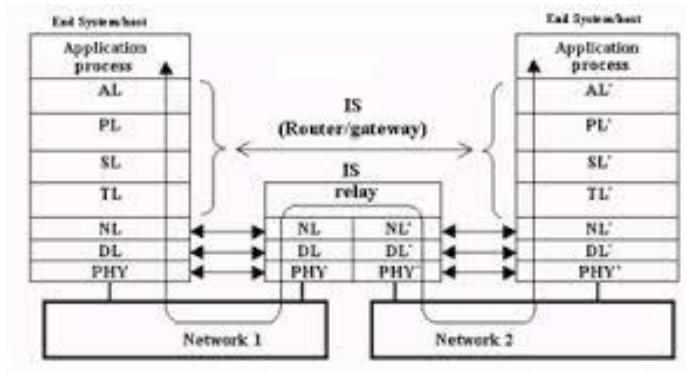
Gambar 5.21 *Bridge jarak jauh*

4.3 Gateway

Gateway digunakan untuk interkoneksi jaringan dimana masing-masing jaringan memiliki arsitektur yang sangat berbeda (Gambar 5.22). jaringan yang dihubungkannya mempunyai protokol yang berbeda mulai dari lapisan hubungan data sampai dengan lapisan aplikasi.

Gateway juga merupakan bentuk khusus dari *router* yang digunakan untuk bertukar informasi dengan router lain yang berlainan cara mengelola informasinya maupun cara pencarian jalannya. Jadi bila satu jaringan yang berisi sekumpulan *router* berjenis sama disebut *autonomus system*, hendak berhubungan dengan jaringan dengan sistem router lain, diperlukan satu buah *gateway* untuk masing-masing jaringan. Dua buah *gateway* ini saling bertukar informasi dengan protokol antar router yang berbeda sistem, yang disebut *ERP (Exterior Router Protocol)*.

Protokol yang dibuat untuk komunikasi *gateway* ini bekerja dalam bentuk permintaan dan tanggapan yang dikirim dalam datagram IP. Contoh permintaan adalah permintaan untuk menjadi *router* tetangga. Permintaan tersebut dapat menjadi dua jenis tanggapan yaitu diterima atau ditolak.



Gambar .22 Gateway

Tiga kegiatan gateway yang berkaitan dengan komunikasi antar sistem adalah :

1. *Neighbour acquisition*

Terjadi ketika dua router bertetangga tetapi berbeda sistem otonomi saling menyetujui untuk saling bertukar informasi pencarian jalan. Prosedur resmi untuk persetujuan ini diperlukan mengingat kemungkinan tidak tersedianya salah satu router untuk berbagi informasi. Prosedur ini dimulai dengan salah satu router mengirimkan permintaan untuk bertetangga. Router lainnya dapat memberi tanggapan menerima atau menolak. Untuk mengakhiri ketanggapan, salah satu router dapat mengirimkan pesan untuk berhenti dan dijawab dengan persetujuan.

2. *Neighbour reachability*

Prosedur ini dilakukan bila hubungan ketanggapan telah ditetapkan, dan digunakan untuk memelihara hubungan. Suatu gateway harus yakin bahwa tetangganya masih ada dan masih berstatus tetangga. Untuk melakukan ini, kedua gateway harus saling bertukar pesan "Hello" dan "I heard you" secara berkala.

3. *Network reachability*

Prosedur ini berkaitan dengan penukaran permintaan dan tanggapan secara berkala. Bila salah satu gateway mengirimkan permintaan poll, yaitu meminta informasi jaringan, maka tetangganya menanggapi dengan pesan update. Tanggapan ini berisi jaringan yang dapat dijangkau oleh gateway pengirim poll,

termasuk jarak masing – masing jaringan tersebut. Dari informasi ini gateway peminta tadi dapat menyusun tabel pencarian jalan.

c.Rangkuman

Lapisan jaringan (*Network layer*) merupakan lapisan ketiga dari standar OSI yang berfungsi untuk menangani masalah jaringan komunikasi secara rinci. Pada lapisan ini, data yang berupa pesan-pesan (*message*) akan dibagi-bagi dalam bentuk paket-paket data yang dilengkapi dengan header-header tertentu pada setiap paket data tersebut.

Network layer ini berfungsi untuk mengambil paket dari sumber dan mengirimkannya ke tujuan. Supaya sampai ditujuan perlu banyak dibuat *hop* pada *router-router* perantara di sepanjang lintasannya. Fungsi *layer* ini sangat kontras dengan fungsi *data link layer*, yang memiliki tujuan lebih sederhana cukup memindahkan *frame* dari ujung kabel yang satu ke ujung yang lainnya. Jadi *network layer* ini merupakan *layer* terbawah yang berkaitan dengan transmisi *end to end*.

d.Tugas : Membuat Ringkasan Materi *network layer*

Sebelum mengerjakan tugas, buatlah kelompok terdiri atas 2-3 orang. Dalam kegiatan ini peserta didik akan membuat ringkasan materi *network layer*

Kemudian secara bergantian masing-masing kelompok mempresentasikan hasilnya didepan kelas.

- 1.1 Bacalah uraian materi diatas dengan teliti dan cermat.
- 1.2 Buatlah ringkasan materi untuk *network layer* menggunakan software pengolah presentasi. Topik yang di tulis meliputi 1) *network layer* ,2) fungsi *network layer* 3) implementasi dari *network layer* 1.3. Presentasikan hasil ringkasan di depan kelas.

e.Tes Formatif

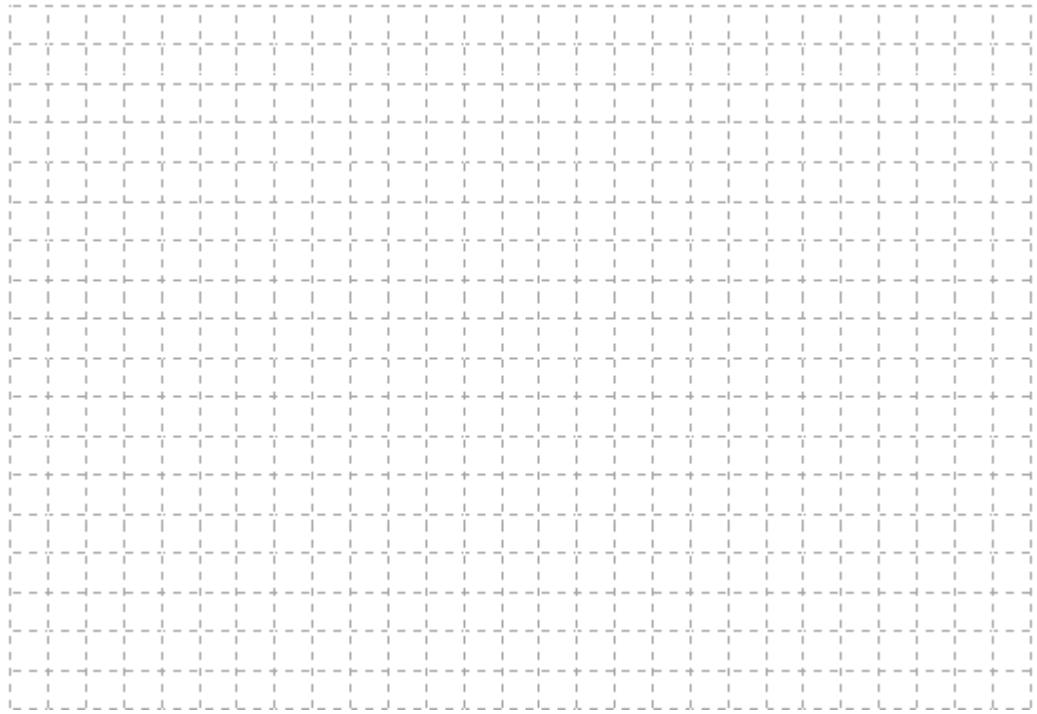
1. Jelaskan lapisan *network layer* dari OSI !
2. Apakah fungsi dari lapisan *network layer* !
3. Jelaskan prinsip kerja dari l *network layer* !
4. Apa yang dimaksud *network layer*! jelaskan !
5. Berikan contoh aplikasi OSI dari *network layer* !

c. Lembar Jawaban Test Formatif

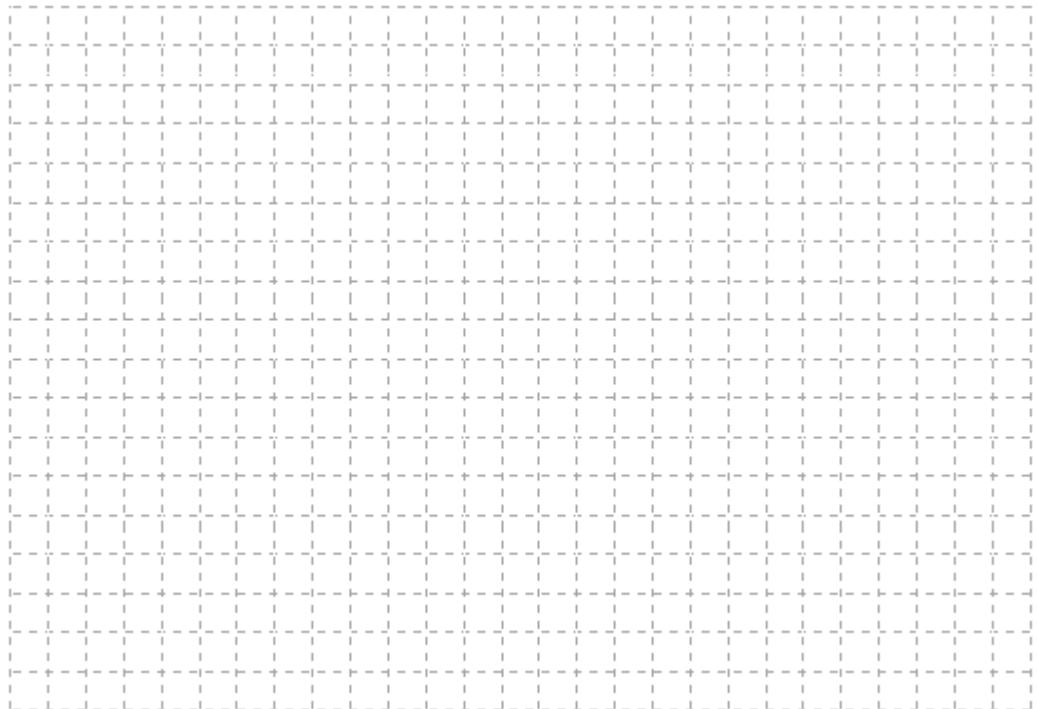
- **Test Essay (LJ.01).**

A large rectangular grid consisting of 20 columns and 25 rows of dashed lines, intended for students to write their answers to the essay questions.

- **Test Essay (LJ.02).**

A large rectangular grid consisting of 20 columns and 25 rows of dashed lines, intended for students to practice handwriting or short essays.

- **Test Essay (LJ.03).**

A large rectangular grid consisting of 20 columns and 25 rows of dashed lines, intended for students to practice handwriting or short essays.

- **Test Essay (LJ.04).**

A large rectangular dashed grid area designed for handwritten responses. It consists of 20 columns and 10 rows of dashed lines.

- **Test Essay (LJ.05).**

A large rectangular dashed grid area designed for handwritten responses. It consists of 20 columns and 10 rows of dashed lines.**e.Lembar Jawaban Siswa**