

9. KEGIATAN BELAJAR 9 : TRANSPORT LAYER

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 9 ini siswa diharapkan dapat :

1. Memahami *transport layer* pada Jaringan Komputer
2. Menganalisis *transport layer* pada Jaringan Komputer

b. Uraian Materi

TRANSPORT LAYER

Transort layer atau lapisan transport merupakan lapisan keempat dari model referensi OIS dan jantung dari hirarki protokol secara keseluruhan. Tugas layer ini menyediakan *data trasnport* yang bisa diandalkan dan efektif biayanya dari komputer sumber ke komputer tujuan, yang tidak tergantung pada jaringan fisik atau jaringan-jaringan yang digunakan. Tanpa *transport layer*, seluruh konsep protokol yang menggunakan *layer* tidak akan ada gunanya.

Layer atau lapisan ini yang mengatur koneksi dari satu ujung ke ujung yang lain atau dari komputer pengirim ke komputer penerima dan juga yang membangun koneksi logika antara *host* pengirim dengan penerima dalam jaringan. Layer ini jugalah yang mengatur dan mengimplementasikan layanan transport yang handal antar jaringan yang transparan untuk layer-layer di atasnya (upper layer). Fungsi dari layer ini meliputi *How control, error checking dan recovery*.

1. Layanan Lapisan Transport

Lapisan transport memberikan pelayanan secara transparan dalam hal *error recovery* dan *flow control*. Fungsi dari lapisan keempat ini merupakan mekanisme yang sesuai untuk pertukaran data antara proses dari sistem yang berlainan. Lapisan transport menjafnin bahwa data yang diberikannya pada *session layer* diatasnya dalam keadaan utuh, urut dan tanpa duplikasi serta bebas dari kesalahan. Lapisan ini juga mendukung penggunaan layanan jaringan semaksimal mungkin. Sebagai contoh

entitas pada lapisan session dapat memilihkan layanan yang diinginkannya seperti penentuan kecepatan pertukaran data, kualitas data yang masih dapat diterima, waktu tunda maksimum, masalah prioritas dan keamanan, sehingga lapisan transport ini berlaku sebagai penghubung antara pemakai dengan fasilitas komunikasi yang ada.

Di samping itu lapisan *transport* juga berfungsi untuk memecah data dari lapisan *session* menjadi pesan-pesan dan memastikan pesan-pesan tersebut diterima dengan benar di tujuan. Lapisan ini akan menyediakan koneksi berganda setiap sesinya ke dalam satu koneksi. Lapisan *transport* juga bertugas mencari jalur (routing) yang kosong untuk transmisi data.

Contoh protokol pada lapisan *transport* yang banyak digunakan adalah TP-NBS, UDP, dan TCP. Memberikan *service connection oriented*, bisa diandalkan dan *byte stream service*. Penjelasan untuk *service* tersebut kurang lebih demikian sebelum melakukan pertukaran data, setiap aplikasi yang menggunakan TCP diwajibkan untuk membentuk hubungan (handshake), kemudian dalam proses pertukaran data, TCP mengimplementasikan proses deteksi kesalahan paket dan retransmisi, dan semua proses ini termasuk pengiriman paket data ke tujuan yang dilakukan secara berurutan.

1.1 Kualitas Layanan

Fungsi primer dari lapisan *transport* adalah meningkatkan kualitas pelayanan yang diberikan oleh lapisan jaringan. Jika layanan lapisan jaringan tanpa cacat, maka lapisan *transport* mendapat beban tugas yang mudah. Sebaliknya, jika layanan jaringannya tidak baik, maka lapisan *transport* ini harus menjembatani jarak antara apa yang diinginkan pemakai lapisan *transport* dan apa saja yang diberikan oleh lapisan jaringan.

Meskipun secara sekilahs kualitas layanan tampaknya seperti konsep yang kabur, tetapi kualitas layanan ini dapat ditentukan menurut sejumlah parameter tertentu.

Parameter-parameter itu adalah sebagai berikut (gambar .1).

1. *Connection establishment delay*

Jumlah waktu antara saat permintaan koneksi *transport* dengan diterimanya konfirmasi oleh pengguna oleh layanan transport. Delay meliputi pemrosesan pada *transport entity* di tempat lain. Seperti halnya parameter-parameter yang menjadi ukuran *delay*, semakin pendek *delay*, maka semakin baik layanannya.

2. *Connection establishment failure probability*

Kesempatan koneksi untuk tidak terbentuk di dalam waktu *delay* pembentukan maksimum, misalnya sehubungan dengan kemacetan jaringan atau kurangnya ruang pada tabel dan masalah internal lainnya.

3. Parameter throughput

Mengukur jumlah *byte* data pengguna yang ditransfer perdetik. *Byte* ini diukur dalam interval waktu tertentu. Throughput diukur secara terpisah pada masing-masing arah lalu lintasnya.

4. Transit delay

Mengukur waktu antara saat pesan yang dikirim oleh pengguna *transport* pada komputer sumber dan saat pesan diterima oleh pengguna *transport* pada komputer tujuan. Seperti halnya throughput.

5. *Residual error ratio*

Mengukur jumlah pesan yang hilang atau rusak sebagai bagian dari total pesan yang dirimkan. Pada teorinya, kelajuan error residu harus sama dengan nol, karena telah merupakan tugas *transport layer* untuk menyembunyikan seluruh *error network layer*. Sedangkan dalam prakteknya, rasio ini bisa berupa nilai kecil yang tertentu.

6. *Transfer failure probability*

Mengukur sejauh mana layanan lapisan *transport* berfungsi sesuai dengan apa yang diharapkan. Ketika koneksi *transport* ditetapkan, harus disepakati dahulu tingkat keluarannya, *delay* transitnya dan angka kesalahan residunya. Probabilitas kegagala transfer datan memberikan tenggang waktu sehingga ketetapan yang telah disepakati ini tidak akan terpenuhi selama periode observasi.

7. *Connection release delay*

Jumlah waktu yang terbuang antara waktu pelepasan awal koneksi oleh komputer sumber dan terjadinya pelepasan pada komputer penerima.

8. *Connection release failure probability* (probabilitas kegagalan pelepasan koneksi) bagian dari usaha pelepasan koneksi yang telah ditetapkan.

9. Parameter *protection*

Menyediakan cara bagi pengguna transport untuk menspesifikasikan pemakaian *transport layer* dalam menyediakan proteksi terhadap pihak ketiga yang tidak berhak, yang berusaha untuk membaca atau memodifikasi data yang hendak ditransmisikan.

10. Parameter *priority*

Menyediakan cara pengguna transport untuk mengindikasikan bahwa beberapa konesinya lebih penting dibanding dengan koneksi lainnya. Pada saat terjadi kemacetan, parameter ini menentukan bahwa koneksi yang berprioritas tinggi dilayani lebih dahulu dari koneksi yang berprioritas rendah.

11. Parameter *resilience*

Memberikan probabilitas *transport layer* itu sendiri yang secara spontan mengakhiri koneksi sehubungan dengan adanya masalah internal kemacetan.

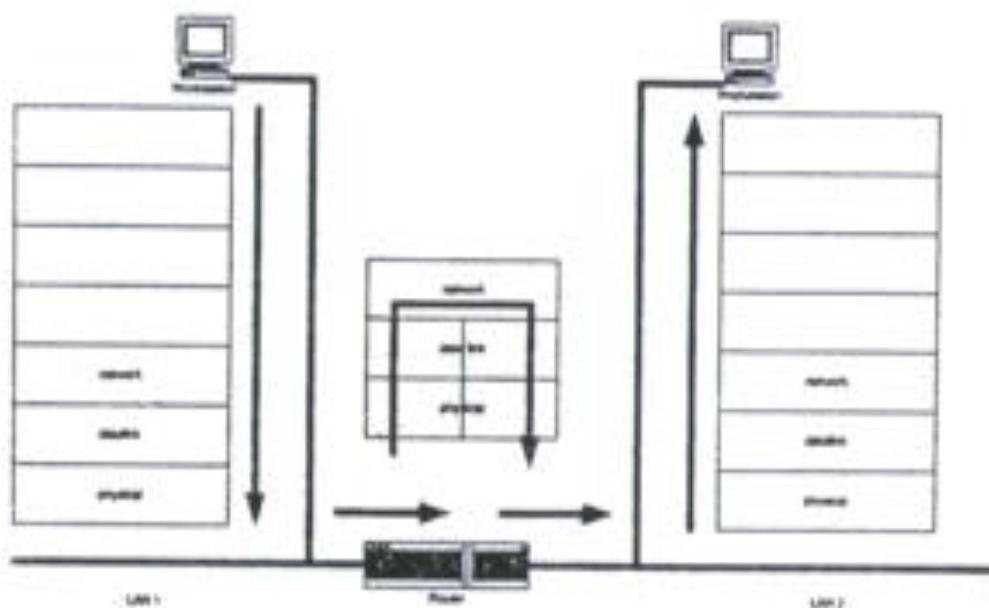
Connection establishment delay
Connection establishment failure probability
Throughput
Translit delay
Residual error rate
Protection
Priority
Resile ice

Gambar 1 .parameter-parameter Kualidis Layanan Transport Layer

1.2 Router

Router adalah perangkat antara yang dapat digunakan untuk menghubungkan dua jaringan lokal yang mempunyai protokol sama pada lapisan jaringan OSI sedangkan protokol pada lapisan fisika dan data ZinL berbeda (Gambar6.2). Router merupakan perangkat pencari jalan yang handal pada situasiinter-koneksi yang kompleks.

Router dapat melakukan segmentasi lalu lintas secara selektif. Dalam suatu *internetworking* dimana terdapat banyak protokol, router dapat memilih jenis protokol yang harus digunakan jalur yang dilaluinya.



Gambar .2 Router

1.2.1 Alasan Penggunaan Router

Dengan digunakannya protokol lapisan jaringan, router memiliki beberapa kelebihan dari *bridge* yang perlu dipertimbangkan:

Transport Layer

- Bridge hanya menggunakan subset dari keseluruhan topologi (spanning tree), sedangkan router dapat menggunakan jalur terbaik yang memang secara fisik berada alamat sumber dan tujuan.
- Mudah disesuaikan bila ada perubahan topologi keseluruhan jaringan.
- Besar keseluruhan jaringan tidak terbatas.
- Bridge menolak paket yang terlalu besar untuk diteruskan, sedangkan *router* tidak.
- Kesibukan suatu sub jaringan tidak mempengaruhi keseluruhan jaringan.

1.2.2 Protokol Router

Protokol-protokol router yang banyak digunakan antara lain :

1. TCP/IP routing protocol
2. SNA routing protocol
3. OSI routing protocol
4. XNS routing protocol
5. IPX routing protocol
6. Apple Talk routing protocol

1.2.3. Alasan Penggunaan Router

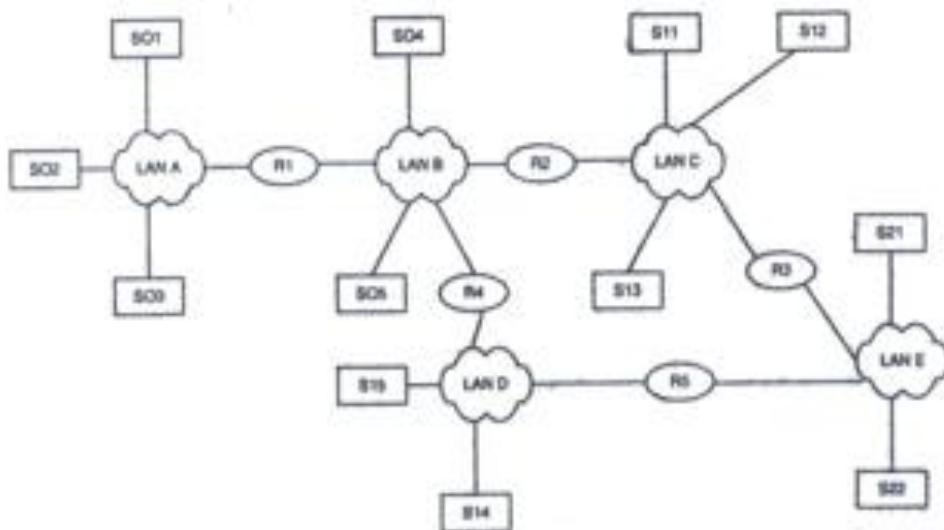
Dengan digunakannya protokol lapisan jaringan, *router* memiliki beberapa kelebihan dari *bridge* yang perlu dipertimbangkan :

- *Bridge* hanya menggunakan subset dari keseluruhan topologi (spanning tree), sedangkan *router* dapat menggunakan jalur terbaik yang memang secara fisik berada antara alamat sumber dan tujuan.
- Mudah disesuaikan bila ada perubahan topologi keseluruhan jaringan.
- Besar keseluruhan jaringan tidak terbatas.
- *Bridge* menolak paket yang terlalu besar untuk diteruskan, sedangkan *router* tidak.

- Kesibukan suatu sub jaringan tidak mempengaruhi keseluruhan jaringan.

1.2.4.Cara kerja Router dan Contoh Kasus

Suatu sistem dengan *router* menggunakan *internet protocol* (IP) pada setiap *router* dan *host* masing-masing jaringan. Protocol ini digunakan untuk membungkus data yang dikirimkan. Contoh proses pengiriman data yang dibungkus dengan protocol IP dapat dilihat pada gambar 3



Gambar 3 contoh pemanfaatan router

Ada empat jenis hubungan stasiun relatif terhadap *router* , yaitu :

1. Stasiun terhubung pada jaringan yang tersambung langsung pada *router*. Pada gambar diatas, semua stasiun berawalan SO berada pada jaringan yang tersambung langsung dengan router R1.
2. Stasiun berada pada jaringan yang tersambung *router* yang tersambung langsung dengan *router* aktif. *Router* tersambung ini disebut *router* tetangga. Pada contoh diatas, misalnya router aktif adalah R1, maka R2 dan R4 merupakan router tetangga, dan semua stasiun berawalan S1 disebut berjarak satu lompatan (one hop) dari R1.
3. Stasiun berada pada jaringan yang tersambung pada router yang bukan tetangga *router* aktif. Stasiun tujuan yang memenuhi keadaan ini disebut berjarak banyak lompatan (multiple hop) dari router aktif. Pada contoh, bila router aktif adalah R1, semua stasiun yang berawalan S2, masuk pada kategori ini. Statiun berawalan S2 berjarak dia lompatan dari R1.
4. *Router* tidak mengenali alamat yang tertera pada protokol IP datagram.

Bila suatu router mendapat datagram untuk diteruskan pada stasiun, maka proses pengiriman datagram akan dilakukan sesuai letak stasiun relatif terhadap router, yang dapat dijelaskan sebagai berikut.

1. Pada kondisi hubungan stasiun relatif router yang pertama, datagram dibungkus protokol sesuai LAN tujuan dengan alamat stasiun tujuan.
2. Pada kondisi hubungan stasiun relatif router yang kedua dan ketiga, datagram juga dibungkus protokol LAN berikutnya, tetapi dengan alamat router berikutnya.
3. Pada kondisi hubungan stasiun relatif router yang keempat, pesan kesalahan dikirimkan kepada stasiun pengirim data.

Proses pengiriman data dari stasiun SOI pada LAN A ke beberapa stasiun lainnya dapat dijelaskan sebagai berikut.

1. Modul pengelola IP pada stasiun SOI membangun datagram dengan alamat jaringan global dan mengenali bahwa alamat tersebut terletak pada LAN lain. Bentuk bungkusan datagram ini dapat dilihat pada gamar 6.4

IP	DATA
----	------

Gambar 4 bentuk bungkusan datagram dengan alamat jaringan global

Datagram IP dikirimkan kepada router R1 dengan dibungkus protokol jaringan yang sesuai dengan LAN A. Lambang HA (R1) maksudnya adalah header protokol untuk LAN A yang berisi alamat R1 dan T-A adalah trailler. Perlu diperhatikan bahwa protokol ini adalah protokol lapisan dibawah jaringan (lapisan ketiga). Bentuk bungkusnya dapat dilihat pada gambar 5

H-A(R1)	IP	DATA	T-A
---------	----	------	-----

Gambar 5 bungkusan datagram IP dikirim ke router R1

2. Router R1 menerima kiriman dan membuka kembali bungkusan protokol LAN A tadi, sehingga diperoleh datagram asli.
3. Berdasarkan protokol IP paket tersebut, router R1 harus menentukan apa yang harus dilakukan. Bila stasiun tujuan tersambung langsung, misalnya SO4 pada LAN B, maka bentuk bungkusan yang dikirim kepada LAN B dapat dilihat gambar 6

H-B(SO4)	IP	DATA	T-P
----------	----	------	-----

Gambar 6 bentuk bungkusan yang dikirim ke LAN B

4. Bila stasiun tujuan berjarak satu lompatan, misalnya stasiun S11 pada LAN C, maka bentuk bungkusan yang dikirim pada LAN B ditujukan untuk router yang sesuai, seperti pada gambar 7

H-B(R2)	IP	DATA	T-P
---------	----	------	-----

Gambar 7. stasiun tujuan berjarak satu lompatan

5. Bila stasiun tujuan berjarak satu lompatan, misalnya stasiun S21 pada LAN E, maka R1 harus menentukan, lewat *router* mana jarak yang ditempuh lebih dekat yaitu dalam satu lompatan. Pada contoh

diatas, jarak untuk menjangkau stasiun tujuan bila melewati R2 maupun R4 adalah sama.

6. Sebelum data sebenarnya disalurkan pada jaringan tujuan, router dapat saja membagi ukuran paket untuk menyesuaikan dengan batasan ukuran paket pada jaringan tujuan. Masing-masing bagian ini menjadi satu datagram IP terpisah, kemudian data ini diantrikan untuk di salurkan.

Keseluruhan proses diatas berlangsung terus sehingga semua data selesai dikirimkan. Kerja pencarian jalan ini pada umumnya dilakukan dengan membuat sebuah table pencarian jalan untuk setiap router dan stasiun, yang berisi informasi mengenai semua kemungkinan jaringan tujuan dan router diantaranya.

2. Protokol-protokol Transport Internet

Internet memiliki dua buah protokol utama dalam *transport layer*, sebuah protokol *connection oriented* dan sebuah protokol *connectionless*. Protokol *connection oriented* tersebut adalah TCP, sedangkan yang *connectionless* adalah UDP. Karena UDP pada dasarnya merupakan IP dengan tambahan header yang pendek, maka yang akan dibahas adalah TCP.

TCP (Transmission Control Protocol) secara spesifik dirancang untuk menyediakan aliran *byte end to end* yang bisa suatu *internetwork* yang tidak bisa diandalkan. Suatu *internetwork* dibedakan dengan jaringan tunggal karena bagian-bagian yang tidak sama dapat memiliki topologi, *bandwidth*, *delay*, ukuran paket dan parameter lain yang sangat beda.

Setiap mesin yang mendukung TCP memiliki *transport entity TCP*, yaitu proses pengguna atau bagian kernel yang mengatur aliran TCP dan *interface* bagi IP layer. Entitas TCP menerima aliran data pengguna dari proses-proses lokal kemudian memecahnya menjadi beberapa

bagian yang masing-masing bagian tidak lebih dari 64KB dan mengirimkan setiap bagian itu sebagai datagram IP yang terpisah.

IP layer tidak menjamin datagram akan dikirimkan dengan benar, maka hal ini tergantung pada TCP dalam memberikan benar, maka hal ini tergantung pada TCP dalam memberikan *timeout* dan mentrasmisi ulang datagram tersebut bila diperlukan.

2.1. Model Layanan TCP

Layanan TCP diperoleh dengan membiarkan pengirim dan penerima membuat end point yang disebut dengan socket. Setiap socket memiliki nomor (alamat) socket yang terdiri atas alamat IP host dan nomor lokal 16 bit bagi host tersebut, yang disebut port. Untuk mendapatkan layanan TCP, koneksi harus dibentuk secara eksplisit diantara socket pada komputer pengirim dan socket pada komputer penerima. Panggilan socket dapat dilihat pada tabel 1 di bawah ini.

Tabel 1 panggilan socket

Primitive	Arti
SOCKET	Membentuk suatu <i>end point</i> komunikasi
BIND	Memasangkan alamat lokal pada socket
LISTEN	Mengumumkan kesediaan menerima koneksi
ACCEPT	Memblok pemanggil sampai usaha koneksi tiba
CONNECT	Aktif mencoba membuat koneksi
SEND	Mengirimkan data melalui koneksi
RECEIVE	Menerima data melalui koneksi
CLOSE	Melepaskan Koneksi

Sebuah socket dapat digunakan untuk sejumlah koneksi pada saat yang bersamaan. Dengan kata lain, dua buah koneksi atau lebih dapat ditempatkan pada socket yang sama. Suatu koneksi diidentifikasi oleh pengenal socket pada kedua sisinya, yaitu socket 1 dan socket 2.

Semua koneksi TCP merupakan *full duplex* dan point to point. Yang dimaksud *full duplex* disini adalah bahwa lalu lintas dapat berjalan dua arah

dalam waktu yang bersamaan. Sedangkan *point to point* artinya bahwa setiap koneksi memiliki dua buah end point.

Koneksi TCP merupakan suatu aliran byte, bukan aliran pesan. Batas-batas pesan tidak memelihara koneksi end to end. Dan layanan terakhir dari TCP adalah *urgent data*. Pada saat pengguna interaktif menekan tombol **DEL** atau **CTRL+C** untuk menghentikan komputasi, aplikasi pengirim menyimpan beberapa informasi kontrol di dalam aliran data dan memberikannya ke TCP bersama-sama dengan *flag URGENT*. Kejadian ini menyebabkan TCP menghentikan pengakumulasian data dan segera mengirimkan seluruh data yang dimilikinya untuk koneksi itu.

2.1.1 Protokol TCP

Entitas TCP pengirim dan penerima saling bertukar dalam bentuk segmen. Sebuah segmen *header* berukuran tetap 20 byte yang diikuti oleh nol atau lebih byte-byte data. Sebuah segmen yang terlalu besar bagi jaringan yang disinggahinya dapat dipecah menjadi beberapa segmen oleh router. Masing-masing segmen baru mendapatkan TCP dan header IP-nya sendiri, sehingga fragmentasi yang dilakukan router menambah overload total.

Protokol dasar yang digunakan oleh intitas TCP merupakan protokol jendela geser. Ketika pengirim mentransmisikan sebuah segmen, pengirim juga menghidupkan timer. Pada saat segmen tiba di tempat tujuan, entitas TCP penerima mengirimkan kembali sebuah segmen yang berkaitan dengan nomor *acknowledgement* yang sama dengan nomor urut berikutnya yang akan diterimanya. Bila timer pengirim berhenti sebelum *acknowledgement* diterima, maka pengirim mentransmisikan segmen lagi.

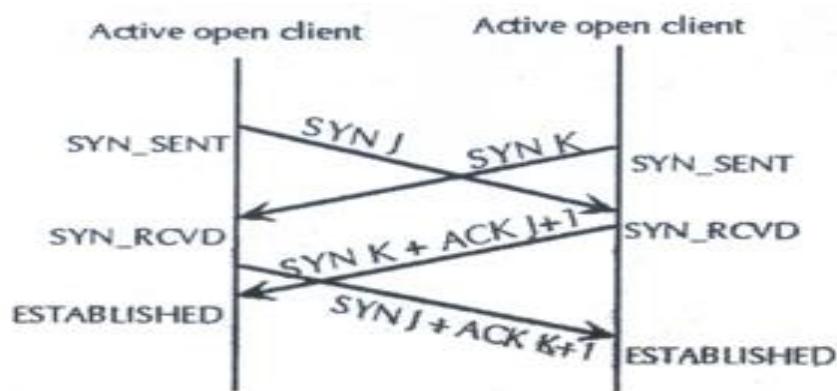
TCP harus disiapkan untuk menghadapi masalah-masalah ini dan mengatasinya dengan cara efisien. Banyak usaha yang telah dilakukan untuk mengoptimasi untuk kerja aliran TCP, termasuk menghadapi masalah jaringan.

2.1.1.1 Manajemen Koneksi TCP

Koneksi dibentuk dalam TCP dengan menggunakan *handshake* tiga arah yang telah dibahas pada bagian sebelumnya. Untuk membentuk sebuah koneksi, satu sisi, misalnya server, secara pasif menunggu koneksi yang masuk dengan mengeksekusi perintah LISTEN dan ACCEPT. Sisi

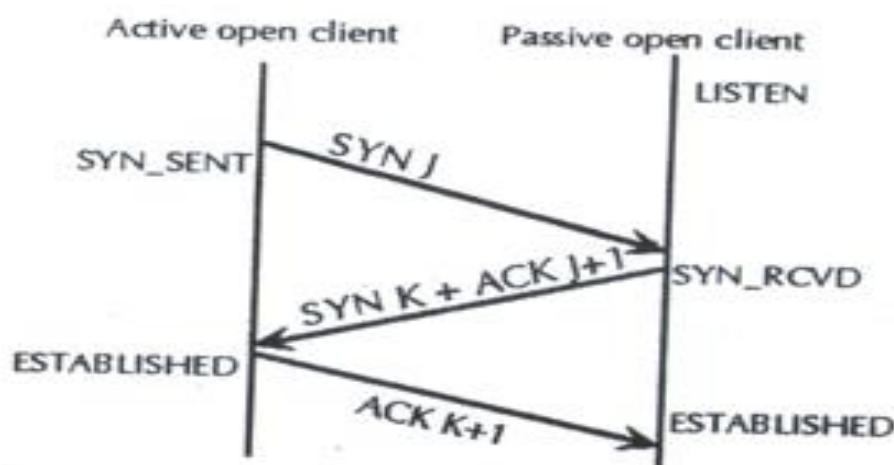
yang lain sebagai client, mengeksekusi perintah CONNECT yang menspesifikasikan alamat IP dan port tempat koneksi akan dibuat, ukuran segmen TCP maksimum yang akan diterima.

Bila beberapa proses mendengarkan port, maka proses tersebut diberi segmen TCP masuk. Proses dapat menerima atau akan dikirim balik. Sejumlah segmen TCP yang dikirimkan pada keadaan normal ditunjukkan oleh gambar 6.9. perlu dicatat bahwa setiap segmen SYN mengkonsumsi 1 byte ruang urut sehingga segmen dapat diberi *acknowledgement* dengan jelas.



Gambar 8 Pembentukan Koneksi TCP pada Keadaan Normal.

Pada saat dua buah client mencoba untuk membentuk koneksi secara bersamaan di antara dua buah socket yang sama, maka hasil dari kejadian ini dapat dilihat pada gambar 6.10



Gambar 9 Tabrakan Panggilan

Untuk membentuk dan melepaskan koneksi pada mesin terbatas dengan kondisi 11 keadaan, maka perlu diterapkan langkah-langkah sebagai berikut yang terlihat pada tabel 6.2 berikut.

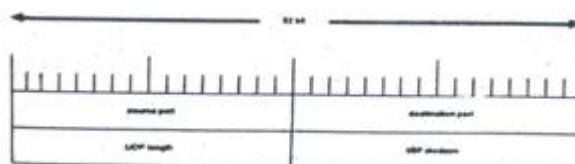
Keadaan	Deskripsi
CLOSED	Tidak terdapat koneksi sedang aktif atau sedang menunggu
LISTEN	Server sedang menunggu panggilan yang datang
SYN RCVD	Permintaan koneksi telah tiba, menunggu ACK
SYN SENT	Aplikasi telah mulai membuka koneksi
ESTABLISHED	Keadaan transfer data normal
FIN WAIT 1	Aplikasi menyatakan bahwa koneksi telah selesai
FIN WAIT 2	Sisi lain setuju untuk melepaskan koneksi
TIMED WAIT	Menunggu seluruh paket untuk mati
CLOSING	Kedua sisi telah mencoba menutup koneksi secara bersamaan
CLOSE WAIT	Sisi lain telah menginisiasi pelepasan koneksi
LAST ACK	Menunggu semua paket untuk mati

Tabel 2 Pembentukan dan Pelepasan Koneksi pada Mesin Terbatas dengan Kondisi 11 Keadaan

2.1.2 Model Layanan UDP

Protokol transport connectionless yaitu UDP (User data Protocol) menyediakan cara bagi aplikasi untuk mengirimkan datagram IP yang dikemas kemudian mengirimkan datagram ini tanpa melakukan pembentukan koneksi.

Segmen UDP terdiri dari header 8 byte yang diikuti oleh data (gambar 6.11). Dua buah party mempunyai fungsi yang sama seperti yang dikejakan pada TCP yaitu mengidentifikasi end point pada komputer sumber dan komputer tujuan. Field length UDP meliputi header 8 byte dan data.



Gambar 10 Header UDP

2.1.3 Rangkuman

Layer transport data, menggunakan protocol seperti UDP, TCP dan/atau SPX(Sequence Packet eXchange, yang satu ini digunakan oleh NetWare, tetapi khususuntuk koneksi berorientasi IPX). Layer transport adalah pusat dari mode-OSI. Layer ini menyediakan transfer yang reliable dan transparan antara kedua titik akhir, layer ini juga menyediakan multiplexing, kendali aliran dan pemeriksaan error serta memperbaikinya. Fungsi transport layer antara lain: Untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada level ini juga membuat sebuah tanda bahwa paket diterima dengan sukses (acknowledgement), dan mentransmisikan ulang terhadap paket-paket yang hilang di tengah jalan.

c. Tes Formatif

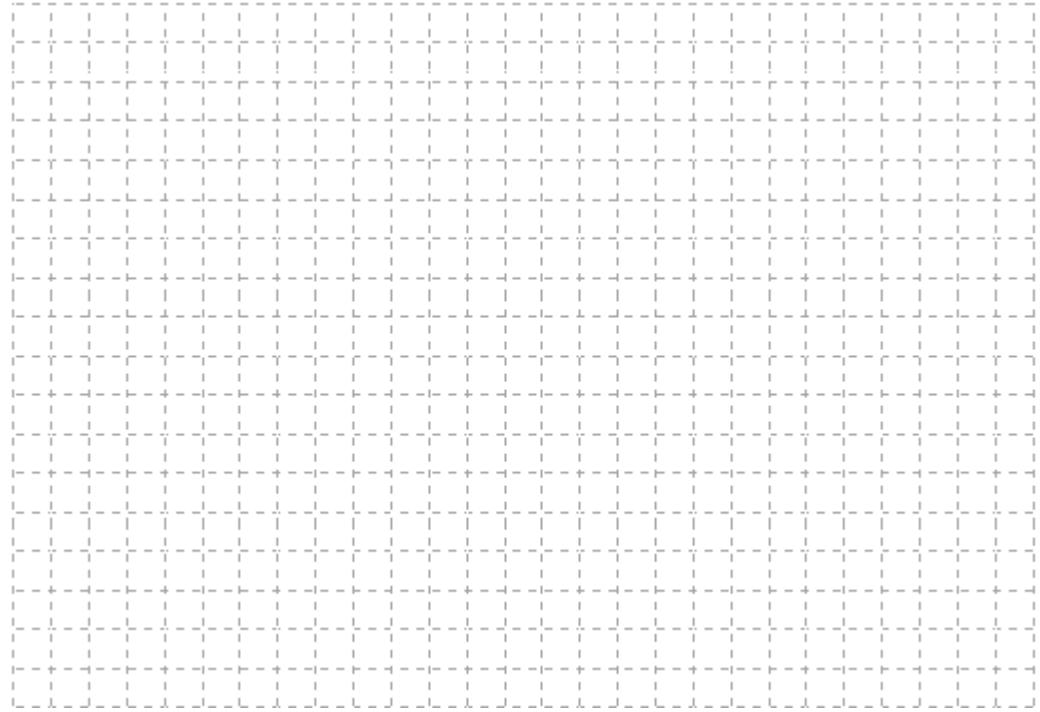
1. Sebutkan dan jelaskan kualitas layanan transport layer
2. Apa yang disebut dengan router ? jelaskan penggunaan router pada jaringan komputer !
3. Sebutkan dan jelaskan protokol-protokol yang ada lapisan transport !
4. Apa yang dimaksud dengan :
 - a. Listen
 - b. CLOSING
 - c. TIME WAIT
5. Sebutkan dan jelaskan model layanan UDP !

a. Lembar Jawaban Test Formatif

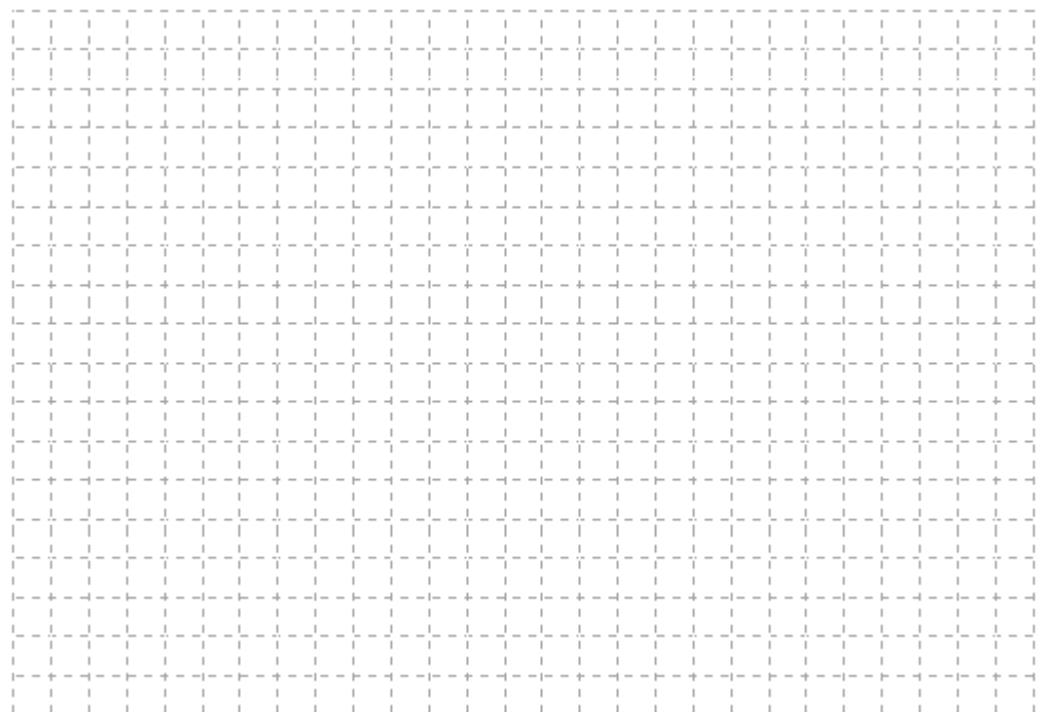
- Test Essay (LJ.01).



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

A large rectangular grid consisting of 20 columns and 25 rows of dashed lines, intended for handwritten text.

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

A large rectangular grid consisting of 20 columns and 25 rows of dashed lines, intended for handwritten text.

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

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

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

A second large rectangular grid consisting of 20 columns and 25 rows of dashed lines, located below the first one, intended for students to practice handwriting or write their answers.**b. Lembar kerja siswa**