

TRANSPORT LAYER



NAMA : 1. HANDOKO DWI HARTONO (13110105)
2. IMANUEL HASIOLAN SIMATUP (13110479)
3. YOSUA MINANDO (181106930)

KELAS : 3KA35

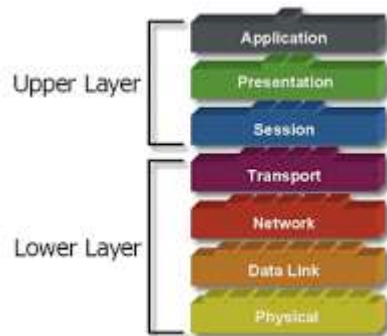
KELOMPOK : 4

KEPADA : IBU MISSA LAMSANI

()
Paraf

7 LAPISAN OSI

OsI Layer adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan International Organization for Standardization (ISO) di Eropa pada tahun 1977. OSI merupakan singkatan dari Open System Interconnection. Model-OSI terbagi menjadi 7 layer, beberapa diantaranya ada yang termasuk upper layer dan lower layer.



Tujuan OSI Layer, yaitu :

1. Penyimpanan data
2. Untuk menjelaskan secara logika tentang jaringan
3. Sebagai patokan untuk pengembangan dimasa depan
4. Manajemen sumber dan proses
5. Koordinasi berbagai kegiatan

• Model OSI Layer

7. Application Layer

Berfungsi sebagai penghubung utama antara aplikasi yang berjalan pada komputer. TCP/IP application layer ini adalah Aplikasi. Application Layer memiliki 8 protokol, yaitu :

- a) DHCP (Dynamic Host Configuration Protocol) => Protokol untuk distribusi IP pada jaringan dengan jumlah IP yang terbatas secara otomatis.
- b) DNS (Domain Name Server) => Database nama domain mesin dan nomor IP.
- c) FTP (File Transfer Protocol) => Protokol untuk transfer file.
- d) HTTP (Hyper Text Transfer Protocol) => Protokol untuk transfer file html dan web.
- e) MIME (Multipurpose Internet Mail Extention) => Protokol untuk mengirimkan file binary dalam bentuk teks.
- f) NNTP (Network News Transfer Protocol) => Protokol untuk menerima dan mengirim newsgroup.
- g) POP (Post Office Protocol) => Protokol untuk mengambil mail dari server.
- h) SMB (Server Message Block) => Protokol untuk transfer berbagai server file DOS & Windows.

6. Presentation Layer

Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan. TCP/IP Presentation Layer ini sama dengan Layer ke-7. Presentation Layer memiliki 4 protokol, yaitu :

- a) SMTP (Simple Mail Transfer Protocol) => Protokol untuk pertukaran mail.
- b) SNMP (Simple Network Management Protocol) => Protokol untuk manajemen jaringan.
- c) Telnet => Protokol untuk akses dari jarak jauh.
- d) TFTP (Trivial FTP) => Protokol untuk transfer file

5. Session Layer

Berfungsi untuk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan. Selain itu, di level ini juga dilakukan resolusi nama. TCP/IP Session Layer ini sama dengan Layer ke-7 dan ke-6. Session Layer mempunyai 3 protokol :

1. NETBIOS (Network Basic Input Output System) => Bios jaringan standar.
2. RPC (Remote Procedure Call) => Prosedur pemanggilan jarak jauh.
3. SOCKET => Input Output untuk network jenis BSD_UNIX.

4. Transport Layer

Berfungsi 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. TCP/IP Transport Layer adalah Transport. Transport Layer mempunyai 2 protokol :

- a) TCP (Transmission Control Protocol) => Connection oriented, berorientasi dengan membutuhkan koneksi jaringan.
- b) UDP (User Datagram Protocol) => Connectionless, berorientasi dengan tidak memerlukan koneksi jaringan.

3. Network Layer

Berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket, dan kemudian melakukan routing melalui *internetworking* dengan menggunakan router dan switch. TCP/IP Network Layer adalah Internet. Network Layer memiliki 4 protokol :

1. IP (Internet Protocol) => Protokol untuk menetapkan routing.
2. RIP (Routing Information Protocol) => Protokol untuk memilih routing.
3. ARP (Address Resolution Protocol) => Protokol untuk mendapatkan informasi hardware dari nomor IP.
4. RARP (Reverse ARP) => Protokol untuk mendapatkan informasi nomor IP dari hardware.

2. Data Link Layer

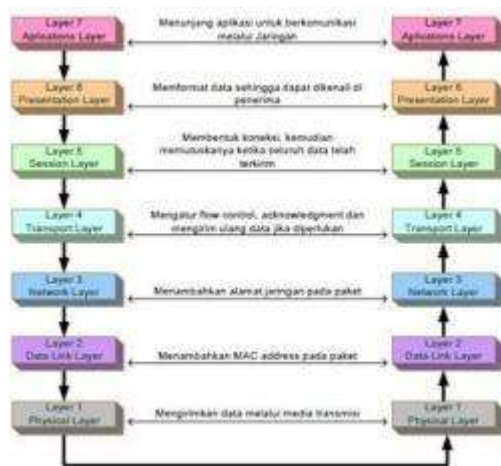
Berfungsi sebagai transmisi data dengan memungkinkan pengirim memecah-mecah data input menjadi sebuah frame yang jumlahnya bisa ratusan maupun ribuan. TCP/IP Data Link Layer adalah Network Interface. Data Link Layer terbagi menjadi dua, yaitu :

- 1) Data Link LLC, nama protokolnya adalah PPP (Point to Point Protocol) yang kegunaannya protokol untuk point ke point.
- 2) Data Link MAC, nama protokolnya adalah SLIP (Serial Line Internet Protocol) yang kegunaannya protokol dengan menggunakan sambungan serial.

1. Physical Layer

Berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan (seperti halnya Ethernet atau Token Ring), topologi jaringan dan pengabelan. TCP/IP Physical Layer sama dengan Data Link. Protokol untuk Physical Layer : Ethernet, FDDI, ISDN, dan ATM

• Cara Kerja Model OSI Layer



Pembentukan paket dimulai dari layer teratas model OSI. Application layer mengirimkan data ke presentation layer, di presentation layer data ditambahkan header atau tailer kemudian dikirim ke layer dibawahnya, pada layer dibawahnya pun demikian, data ditambahkan header atau tailer kemudian dikirimkan ke layer dibawahnya lagi, terus demikian sampai ke physical layer.

Di physical layer data dikirimkan melalui media transmisi ke host tujuan. Di host tujuan paket data mengalir dengan arah sebaliknya, dari

layer paling bawah ke layer paling atas. Protokol pada physical layer di host tujuan mengambil paket data dari media transmisi kemudian mengirimkannya ke data link layer, data link layer memeriksa data-link layer header yang ditambahkan host pengirim pada paket, jika host bukan yang dituju oleh paket tersebut maka paket itu akan di buang, tetapi jika host adalah yang dituju oleh paket tersebut maka paket akan dikirimkan ke network layer, proses ini terus berlanjut sampai ke application layer di host tujuan. Proses pengiriman paket dari layer ke layer ini disebut dengan “peer-layer communication”.

• LAYER TRANSPORT

Data dari masing-masing aplikasi yang dikemas, diangkut, dan dikirim ke server daemon yang sesuai atau aplikasi pada perangkat tujuan. Proses yang dijelaskan dalam lapisan OSI Transport menerima data dari lapisan aplikasi dan mempersiapkannya untuk diserahkan pada layer Network. Lapisan Transport bertanggung jawab untuk transfer end-to-end keseluruhan data aplikasi. Lapisan Transport meliputi fungsi-fungsi ini:

- Segmentasi dan Reassembly
- Multiplexing
- Establishing a Session
- Reliable Delivery

- Same Order Delivery
- Flow Control

1) Segmentasi dan Reassembly

Dengan menggunakan OSI model, berbagai macam jenis aplikasi yang berbeda dapat dikirimkan pada jenis transport yang sama. Transport yang terkirim berupa segmen per segmen. Sehingga data dikirim berdasarkan *first-come first served*. Lapisan Transport menyediakan untuk segmentasi data dan kontrol yang diperlukan untuk memasang kembali potongan-potongan ke dalam berbagai aliran komunikasi. tanggung jawab utamanya untuk mencapai hal ini adalah:

- 1) Pelacakan komunikasi individual antara aplikasi pada sumber dan host tujuan,
- 2) Mengelompokkan data dan mengelola masing-masing bagian,
- 3) Pemasangan kembali segmen ke aliran aplikasi data,
- 4) Mengidentifikasi berbagai aplikasi.

a) Pelacakan komunikasi Individu

Setiap host mungkin memiliki beberapa aplikasi yang berkomunikasi melalui jaringan. Masing-masing aplikasi akan berkomunikasi dengan satu atau lebih aplikasi pada host remote. Ini adalah tanggung jawab lapisan Transport untuk mempertahankan beberapa aliran komunikasi antara aplikasi.

b) Mengelompokkan Data

Setiap aplikasi membuat sebuah data stream untuk dikirim ke sebuah aplikasi remote, data ini harus siap untuk dikirim di media dalam potongan terkelola. Protokol Lapisan Transport memberikan layanan segmentasi data dari lapisan aplikasi. Layanan ini termasuk *enkapsulasi*.

c) Pemasangan Segmen kembali

Diperlukan pada setiap bagian data. Setiap bagian dari data aplikasi membutuhkan header yang akan ditambahkan pada lapisan Transport untuk menunjukkan ke mana komunikasi itu terkait. Bagian data yang diterima dari host sumber dapat diarahkan ke aplikasi yang sesuai. Tetapi sebelumnya segmen (potongan-potongan data) harus kembali ke dalam aliran data lengkap yang berguna untuk lapisan aplikasi. Protokol pada layer Transport memakai informasi header untuk memasang kembali potongan data ke dalam stream untuk diteruskan ke lapisan aplikasi.

d) Mengidentifikasi Aplikasi

Dalam rangka melewatkan data stream ke aplikasi yang tepat, lapisan Transport harus mengidentifikasi aplikasi target. Untuk mencapai hal ini, lapisan Transport menugaskan suatu aplikasi identifier. Protokol TCP / IP menyebutnya identifier nomor port. Setiap proses software yang perlu mengakses jaringan diberi nomor port unik dalam host tersebut. Nomor port yang digunakan di header layer Transport untuk menunjukkan mana aplikasi yang tepat untuk potongan kecil data yang dikirimkan. Lapisan Transport

adalah penghubung antara lapisan aplikasi dan lapisan bawah yang bertanggung jawab untuk transmisi jaringan. Lapisan ini menerima data dari percakapan yang berbeda dan meneruskannya ke lapisan bawah sebagai potongan-potongan dikelola yang dapat akhirnya multiplexing atas media. Aplikasi tidak perlu mengetahui rincian operasional jaringan yang digunakan. Aplikasi menghasilkan data yang dikirim dari satu aplikasi ke yang lain, tanpa memperhatikan jenis host tujuan, jenis media di mana data harus melakukan perjalanan, jalan yang diambil oleh data, kemacetan di link, atau ukuran jaringan.

Selain itu, lapisan bawah tidak menyadari bahwa ada beberapa aplikasi mengirim data pada jaringan. Tanggung jawab mereka adalah untuk mengirimkan data ke perangkat yang sesuai. Lapisan Transport kemudian macam potongan-potongan sebelum mengirimkannya ke aplikasi yang sesuai.

e) **Persyaratan Variasikan Data**

Karena aplikasi yang berbeda memiliki kebutuhan yang berbeda, ada beberapa protokol lapisan Transport. Untuk beberapa aplikasi, segmen harus tiba di urutan sangat spesifik dalam rangka untuk diproses berhasil. Dalam beberapa kasus, semua data yang harus diterima untuk semua itu untuk digunakan. Dalam kasus lain, aplikasi dapat mentolerir kehilangan beberapa data selama transmisi melalui jaringan.

Aplikasi, seperti database, halaman web, dan e-mail, memerlukan bahwa semua data yang dikirim sampai di tempat tujuan dalam kondisi aslinya, agar data yang akan berguna. Setiap data yang hilang bisa menyebabkan komunikasi korup yang tidak lengkap atau tidak terbaca. Oleh karena itu, aplikasi ini dirancang untuk menggunakan protokol lapisan Transport yang mengimplementasikan kehandalan. Overhead jaringan tambahan dianggap diperlukan untuk aplikasi ini. Aplikasi lain yang lebih toleran terhadap hilangnya sejumlah kecil data. Misalnya, jika satu atau dua segmen dari aliran video gagal tiba, itu hanya akan menciptakan gangguan sesaat dalam sungai. Ini mungkin muncul sebagai distorsi pada gambar, tetapi bahkan mungkin tidak terlihat oleh pengguna.

Dua protokol lapisan yang paling umum Transport TCP / IP protocol suite adalah Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP). Kedua protokol mengelola komunikasi beberapa aplikasi. Perbedaan antara kedua adalah fungsi khusus yang setiap protokol mengimplementasikan.

2) **Multiplexing**

Memungkinkan ada banyak aplikasi atau service yang berjalan pada setiap host dalam jaringan. Segmentasi data, sesuai dengan protokol lapisan Transport, menyediakan sarana untuk mengirim dan menerima data ketika menjalankan beberapa aplikasi secara bersamaan pada komputer. Tanpa segmentasi, hanya satu aplikasi, video streaming misalnya, akan mampu menerima data. Anda tidak bisa menerima e-mail, chatting di instant messenger, atau halaman tampilan web sementara juga melihat video. Beberapa aplikasi mengirimkan data dalam jumlah besar - dalam

beberapa kasus, banyak.gigabyte . Hal ini akan tidak praktis untuk mengirim semua data ini dalam satu potongan besar. Tidak ada lalu lintas jaringan lainnya bisa dikirim saat ini data sedang dikirim.Sepotong besar data bisa memakan waktu menit atau bahkan berjam-jam untuk mengirim.

Selain itu, jika ada kesalahan, seluruh data file harus hilang atau kembali. Jaringan perangkat tidak akan memiliki memori buffer yang cukup besar untuk menyimpan data ini banyak ketika sedang dikirim atau diterima. Batasan tersebut bervariasi tergantung pada teknologi jaringan dan sarana fisik tertentu yang digunakan. Masing-masing aplikasi atau jasa diberikan sebuah alamat yang dikenal sebagai port sehingga lapisan Transport dapat menentukan dengan yang aplikasi atau layanan data tersebut diidentifikasi.

3) Establishing a Session

Lapisan Transport dapat memberikan orientasi koneksi dengan menciptakan sesi antara aplikasi. Koneksi ini menyiapkan aplikasi untuk berkomunikasi dengan satu sama lain sebelum data apapun ditransmisikan. Dalam sesi ini, data untuk komunikasi antara dua aplikasi dapat dikelola erat. Konsepnya, sebuah perangkat untuk melakukan komunikasi dengan perangkat lainnya, perangkat yang dituju harus menerima koneksi terlebih dahulu sebelum mengirimkan atau menerima data. Proses yang dilakukan sebelum pengiriman data :

- ✓ Pengirim (sender) mengirimkan sinyal Synchronize terlebih dulu ke tujuan,
- ✓ Penerima (receiver) mengirimkan balasan dengan sinyal Negotiate Connection,
- ✓ Penerima mengirimkan Synchronize ulang, apa benar pengirim akan mengirimkan data,
- ✓ Pengirim membalas dengan sinyal Acknowledge dimana artinya sudah siap untuk mengirimkan data,
- ✓ Connection establish,
- ✓ Kemudian segmen dikirim



Gambar 2.6 Proses pembentukan koneksi

4) **Reliable Pengiriman**

Dinamakan data yang reliabel artinya paket data datang sesuai dengan urutan pada saat dikirimkan. Protokol akan gagal apabila terjadi paket yang hilang, rusak, terjadi duplikasi, atau menerima paket data dengan urutan yang berbeda. Untuk memastikan data yang terkirim, si penerima harus mengirimkan acknowledge untuk setiap data yang diterima pada segmen.

5) **Same Order Delivery**

Karena jaringan mungkin menyediakan beberapa rute yang dapat memiliki waktu transmisi yang berbeda, data dapat tiba di urutan yang salah. Dengan penomoran dan urutan segmen, lapisan Transport dapat memastikan bahwa segmen dipasang kembali ke dalam urutan yang tepat.

6) **Flow Control**

Pada waktu pengiriman data sedang berjalan, kepadatan jalur bisa terjadi (congestion). Alasan terjadinya congestion antara lain: komputer berkecepatan tinggi mengirimkan data lebih cepat dari pada jaringannya, apabila beberapa komputer mengirimkan data ke tujuan yang sama secara simultan. Untuk mengatasi hal tersebut setiap perangkat dilengkapi dengan yang namanya kontrol aliran (flow control). Dimana apabila ada pengirim yang mengirimkan data terlalu banyak, maka dari pihak penerima akan mengirimkan pesan ke pengirim bahwa jangan mengirim data lagi, karena data yang sebelumnya sedang di proses. Dan apabila telah selesai diproses, si penerima akan mengirimkan pesan ke pengirim untuk melanjutkan pengiriman data. Ilustrasi flow control.

• **TCP dan UDP**

Dua protokol lapisan yang paling umum Transport TCP / IP protocol suite adalah Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP). Kedua protokol mengelola komunikasi beberapa aplikasi, perbedaan antara keduanya adalah fungsi khusus yang setiap protokol mengimplementasikan.

1. **User Datagram Protocol (UDP)**

UDP adalah protokol sederhana, connectionless, dijelaskan dalam RFC 768. Ini memiliki keuntungan dari memberikan untuk rendah overhead pengiriman data. Potongan-potongan komunikasi dalam UDP disebut datagrams. Datagrams ini dikirim sebagai "usaha terbaik" oleh protokol lapisan Transport. Contoh aplikasi yang menggunakan UDP adalah:

- ❖ Domain Name System (DNS)
- ❖ Video Streaming Voice over IP (VoIP)

2. **Transmission Control Protocol (TCP)**

TCP merupakan protokol berorientasi koneksi. TCP menimbulkan overhead tambahan untuk mendapatkan fungsi. Fungsi tambahan yang ditetapkan oleh TCP adalah pengiriman urutan yang sama, pengiriman yang handal, dan flow control. Setiap segmen TCP memiliki 20 byte overhead di header encapsulating data lapisan aplikasi, sedangkan setiap segmen UDP hanya memiliki 8 byte overhead. Lihat angka untuk perbandingan. Aplikasi yang menggunakan TCP adalah:

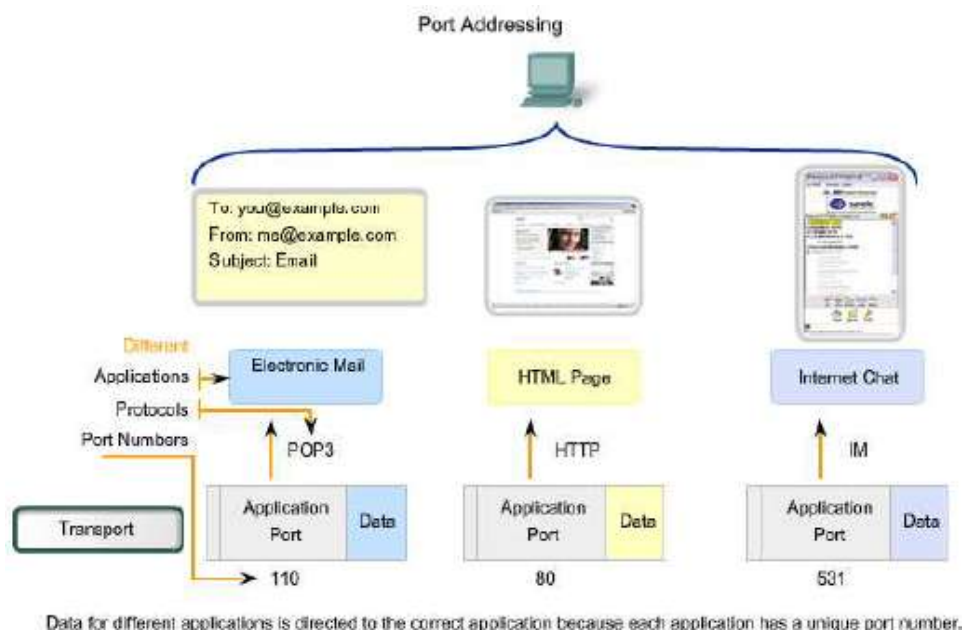
- ❖ Browser web
- ❖ E-mail
- ❖ Transfer file

• Identifikasi Komunikasi

TCP dan layanan berbasis UDP melacak berbagai aplikasi yang berkomunikasi. Untuk membedakan segmen dan datagram untuk setiap aplikasi, baik TCP dan UDP memiliki header field yang secara unik dapat mengidentifikasi aplikasi ini. Dalam header dari masing-masing segmen atau datagram, ada sumber dan port tujuan. Nomor port sumber adalah port untuk komunikasi dengan aplikasi terkait. Nomor port tujuan adalah nomor untuk komunikasi terkait dengan aplikasi tujuan pada host remote. Identifikasi unik adalah nomor port.

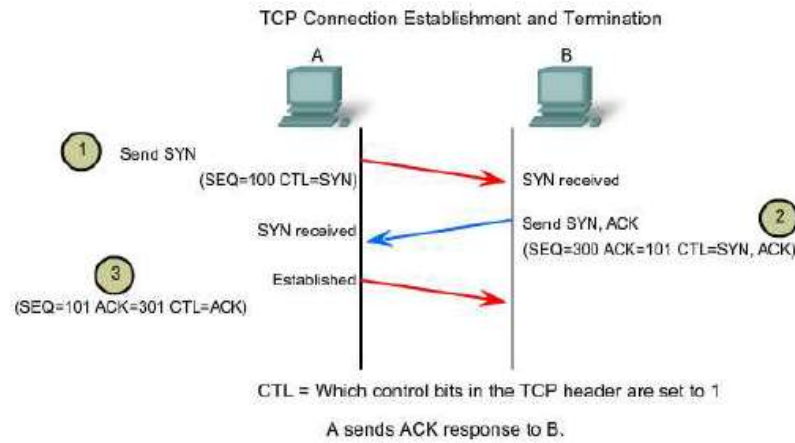
Port Sumber: dinamik (acak)

Port Tujuan : tetap



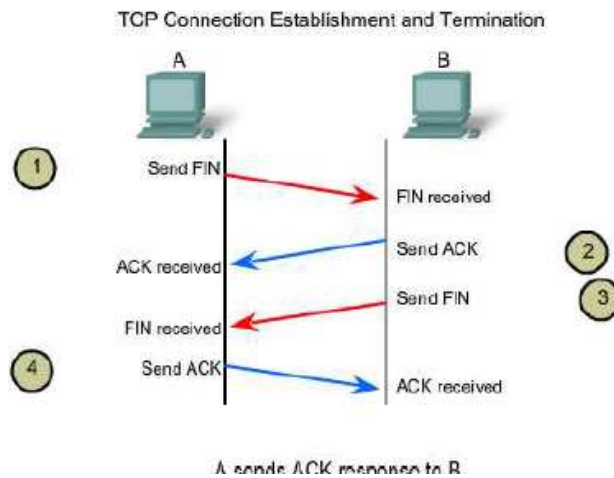
Ada berbagai jenis nomor port:

- 1) **Well Known Ports (Bilangan 0-1023)** - Angka-angka ini dicadangkan untuk layanan dan aplikasi. Mereka umumnya digunakan untuk aplikasi seperti HTTP (web server) POP3/SMTP (email server) dan Telnet. Dengan mendefinisikan port ini dikenal untuk aplikasi server, aplikasi client dapat diprogram untuk meminta koneksi ke port tertentu dan layanan yang terkait.
- 2) **Terdaftar Ports (Registered Ports) (Bilangan 1024-49151)** - Nomor port ini ditugaskan untuk proses user atau aplikasi. Proses ini terutama aplikasi individual bahwa seorang pengguna telah memilih untuk menginstal aplikasi umum yang akan menerima Well Known Port. Bila tidak digunakan untuk sumber daya server, port ini dapat juga digunakan secara dinamis dipilih oleh klien sebagai port sumbernya.



- 1) Klien memulai mengirimkan (**send SYN**) sebuah segmen yang berisi nilai urutan awal (**SEQ**), yang berfungsi sebagai permintaan ke server untuk memulai sesi komunikasi.
- 2) Server merespon dengan segmen yang mengandung nilai pengakuan (**ACK**) sama dengan nilai urutan (**SEQ**) yang diterima ditambah 1, ditambah nilai urutan sinkronisasi nya sendiri. Nilai ini merupakan salah satu lebih besar dari nomor urutan karena ACK selalu yang diharapkan adalah Byte berikutnya atau oktet. Pengakuan (**ACK**) nilai ini memungkinkan klien untuk mengikat respon kembali ke segmen asli yang dikirimkan ke server.
- 3) Memulai merespon klien dengan nilai pengakuan (**ACK**) sama dengan nilai urutan (**SEQ**) yang diterimanya ditambah satu. Ini melengkapi proses pembentukan koneksi.

Untuk Pemutusan Sambungan antar Host



Dalam segmen header TCP, ada enam bidang 1-bit yang mengandung informasi kontrol yang digunakan untuk mengelola proses TCP. Bidang-bidang itu adalah:

URG - Urgent pointer field yang signifikan

ACK - Pengakuan lapangan signifikan

PSH - Push fungsi

RST - Reset koneksi

SYN - Menyinkronkan urutan nomor

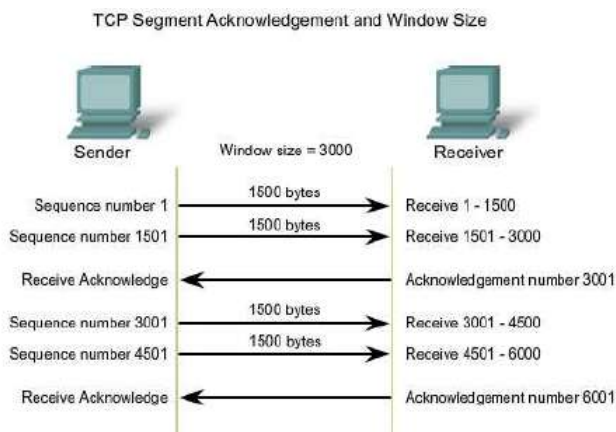
FIN - Tidak ada data lebih dari pengirim

Bidang ini disebut sebagai flag, karena nilai salah satu bidang ini hanya 1 bit dan, karenanya, hanya memiliki dua nilai: 1 atau 0. Ketika nilai bit set ke 1, ini menunjukkan apa kontrol informasi yang terkandung dalam segmen ini.

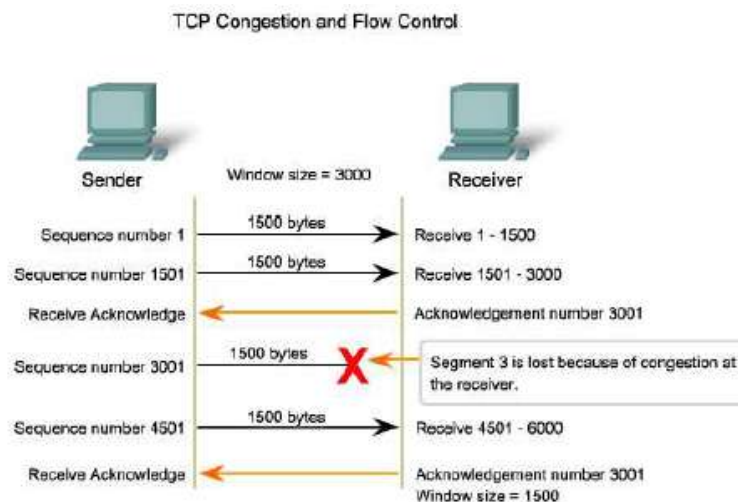
Flow Control pada TCP

TCP menyediakan mekanisme untuk kontrol aliran. Flow control membantu keandalan transmisi TCP dengan menyesuaikan tingkat efektif aliran data antara dua layanan di sesi. Ketika sumber diberitahu bahwa jumlah tertentu dari data dalam segmen diterima, dapat melanjutkan pengiriman lebih banyak data untuk sesi berikutnya. Window Size pada header TCP menentukan jumlah data yang dapat ditransmisikan sebelum pengakuan (ACK) harus diterima. Window Size awal ditentukan selama sesi startup melalui jabat tangan tiga arah (**three-way handshake**). mekanisme umpan balik TCP dimaksudkan untuk menyesuaikan antar data yang akan dikirim kembali dengan tingkat efektif transmisi data

yang bisa dikirimkan ke tujuan tanpa ada data yang hilang. Upaya ini dilakukan TCP untuk mengelola proses transmisi sehingga semua data akan diterima dan transmisi ulang akan diminimalkan.



The window size determines the number of bytes sent before an acknowledgment is expected. The acknowledgement number is the number of the next expected byte.



If segments are lost because of congestion, the Receiver will acknowledge the last received sequential segment and reply with a reduced window size.