

7. KEGIATAN BELAJAR 7 : DATA LINK LAYER

a. Tujuan Pembelajaran

Setelah mengikuti kegiatan belajar 7 ini siswa diharapkan dapat :

- 1) Memahami data link layer pada Jaringan Komputer
- 2) Menganalisis data link layer pada Jaringan Komputer

b. Uraian Materi

DATA LINK LAYER

Lapisan data link (data link layer) merupakan lapisan kedua dari standar OSI. Tugas utama data *link layer* adalah sebagai fasilitas transmisi *raw data* dan mentransformasi data tersebut ke saluran yang bebas dari kesalahan transmisi. Sebelum diteruskan ke *network layer*, *data link layer* melaksanakan tugas ini dengan memungkinkan pengirim memecah-mecah data input menjadi sejumlah data *frame* (biasanya berjumlah ratusan atau ribuan *byte*).

Kemudian *data link layer* mentransmisikan *frame* tersebut secara berurutan dan memproses *acknowledgement frame* yang dikirim kembali oleh penerima. Karena *physical layer* menerima dan mengirim aliran bit tanpa mengindah arti atau arsitektur *frame*, maka tergantung pada *data link layer*-lah untuk membuat dan mengenali batas-batas *frame* itu. Hal ini bisa dilakukan dengan cara membubuhkan bit khusus ke awal dan akhir *frame*.

1 .Masalah-masalah Rancangan Data Link Layer

Bila secara insidental pola-pola bit ini bisa ditemui pada data, maka diperlukan perhatian khusus untuk menyakinkan bahwa pola tersebut tidak secara salah dianggap sebagai batas-batas *frame*.

Terjadinya *noise* pada saluran data merusak *frame*. Dalam hal ini, perangkat lunak *data link layer* pada mesin sumber dapat mengirim kembali *frame* yang rusak tersebut. Akan tetapi transmisi *frame* yang sama secara berulang-ulang bisa menimbulkan duplikasi *frame*. Duplikat *frame* perlu dikirim apabila *acknowledgement frame* dari penerima

yang dikembalikan ke pengirim telah hilang. Tergantung pada *layer* inilah untuk mengatasi masalah-masalah yang disebabkan rusaknya, hilangnya dan duplikasi *frame*. *Data link layer* menyediakan beberapa kelas layanan bagi *network layer*. Kelas layanan ini dapat dibedakan dalam hal kualitas dan harganya.

Masalah-masalah lainnya yang timbul pada data *link layer* (dan juga sebagian besar layer –layer di atasnya) adalah mengusahakan kelancaran proses pengiriman data dari pengirim yang cepat ke penerima yang lambat. Mekanisme pengaturan lalu-lintas data yang harus memungkinkan pengirim mengetahui jumlah ruang *buffer* yang dimiliki penerima pada suatu saat tertentu. Seringkali pengaturan aliran dan penanganan error ini dilakukan secara terintegrasi.

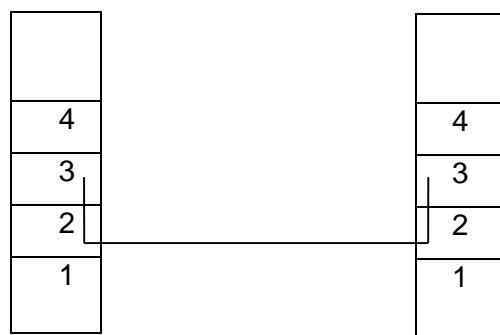
Saluran yang dapat mengirim data pada kedua arahnya juga bisa menimbulkan masalah. Sehingga dengan demikian perlu dijadikan bahan pertimbangan bagi software data *link layer*. Masalah yang dapat timbul disini adalah bahwa *frame-frame acknowledgement* yang mengalir dari A ke B bersaing saling mendahului dengan aliran dari B ke A.

Lapisan data link ini terdiri dari dua *sub layer*, yaitu *Logical Link Control (LLC)* Dan *Medium Access Control (MAC)*. Fungsi dari LLC adalah melakukan pemeriksaan kesalahan dan menangani transmisi *frame*. Setiap *frame* merupakan sebuah paket data dan nomor urut yang digunakan untuk memastikan pengiriman dan sebuah *checksum* untuk melacak data yang hilang. Sedangkan fungsi dari MAC adalah berurusan dengan mengambil dan melepaskan data dari dan ke kabel, menentukan protokol untuk akses ke kabel yang di-share di dalam sebuah LAN.

Sesuai dengan keterangan yang telah disebutkan di atas, bahwa *data link layer* memiliki beberapa fungsi spesifik, diantaranya penyediaan *interface* layanan yang baik bagi *network layer* (mengenai *network layer* akan dibahas pada bab selanjutnya), penentuan cara pengelompokan bit dari *physical layer* ke dalam *frame*, hal hal yang berkaitan dengan error transmisi dan pengaturan aliran frame sehingga receiver yang lambat tidak akan terbanjiri oleh pengirim yang cepat.

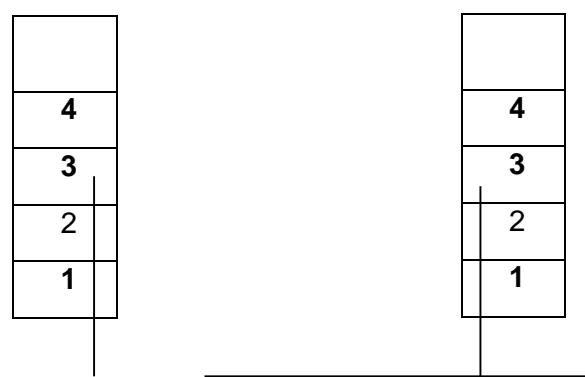
1.1 Layanan yang Disediakan Bagi Network Layer

Fungsi dari data link layer adalah menyediakan layanan bagi network layer. Layanannya yang penting adalah pemindahan data dari network layer di komputer sumber ke network layer di komputer yang dituju. Tugas data link layer adalah mentrasmisikan bit-bit ke komputer yang dituju, sehingga bit-bit tersebut dapat diserahkan ke network layer(Gambar 1)



Gambar 1 komunikasi Virtual

Transmisi aktual yang mengikuti lintasan akan lebih mudah lagi jika dianggap sebagai proses dua kata link layer yang berkomunikasi menggunakan data link protokol



(Gambar .2).

Data link layer dapat dirancang sehingga mampu bermacam-macam layanan. Layanan aktual yang ditawarkan suatu sistem akan berbeda dengan layanan sistem yang lainnya. Tiga layanan yang disediakan adalah sebagai berikut:

1. Layanan unacknowledged connectionless
2. Layanan acknowledged connectionless
3. Layanan acknowledged connection-oriented

Setiap layanan yang diberikan data link layer akan dibahas satu persatu.

1.1.1 Layanan Unacknowledged Connectionless

Layanan jenis ini mempunyai arti dimana komputer sumber mengirimkan sejumlah *frame* ke komputer lain yang dituju dengan tidak memberikan *acknowledgment* bagi diterimanya *frame-frame* tersebut. Tidak ada koneksi yang dibuat baik sebelum atau sesudah dikirimkannya *frame*. bila sebuah frame hilang sehubungan dengan adanya noise, maka tidak ada usaha untuk memperbaiki masalah tersebut di data link layer. Jenis layanan ini cocok bila laju error sangat rendah, sehingga recovery bisa dilakukan oleh layer yang lebih tinggi sebagian besar LAN menggunakan layanan unacknowledgement connectionless pada data link layer

1.1.2 Layanan Acknowledged Connectionless

Pada layanan jenis ini berkaitan dengan masalah rehabilitas. Layanan ini juga tidak menggunakan koneksi, akan tetapi setiap frame dikirim secara independen dan secara acknowledged. Dalam hal ini, si pengirim akan mengetahui apakah frame yang dikirim ke komputer tujuan telah diterima dengan baik atau tidak. Bila ternyata belum tiba pada interval waktu yang telah ditentukan maka frame akan dikirimkan kembali layanan ini akan berguna untuk saluran unreliable, seperti sistem tanpa kabel.

1.1.3 Layanan Acknowledged Connection-oriented

Layanan jenis ini merupakan layanan yang paling canggih dari semua layanan yang disediakan data link layer bagi network layer. Dengan

layanan ini, komputer sumber dan komputer tujuan membuat koneksi sebelum memindahkan datanya. Setiap frame yang dikirim tentu saja diterima. Selain itu, layanan ini menjamin bahwa setiap frame yang diterima benar-benar hanya sekali dan semua frame diterima dalam urutan yang benar. Sebaliknya dengan layanan connectionsless, mungkin saja hilangnya acknowledgement akan menyebabkan sebuah frame perlu dikirimkan beberapa kali juga. Sedangkan layanan connections-oriented menyediakan proses-proses network layer dengan aliran bit yang bisa diandalkan.

Pada saat layanan connection oriented dipakai, pemindahan data mengalami tiga fase. Pada fase pertama koneksi ditentukan dengan membuat kedua komputer menginisiasi variabel-variabel dan counter-counter yang diperlukan untuk mengawasi frame yang mana telah diterima dan yang mana yang belum. Dalam fase kedua, satu frame atau lebih mulai ditransmisikan. Pada fase ketiga, koneksi dilepaskan, pembebasan variabel, buffer dan resource yang lain yang dipakai untuk menjaga berlangsungnya koneksi.

1.2 Framming

Untuk melayani network layer, data link layer harus menggunakan layanan yang disediakan oleh physical layer. Apa yang dilakukan phisical layer adalah menerima aliran raw bit dan berusaha mengirimkannya ke tujuan. Aliran bit ini tidak dijamin bebas dari error. Jumlah bit yang diterima mungkin bisa lebih sedikit, sama atau lebih banyak dari jumlah bit yang di transmisikan dan juga bit-bit itu memiliki nilai yang berbeda-beda. Bila diperlukan, data link layer bertanggung jawab untuk mendekripsi dan mengoreksi error.

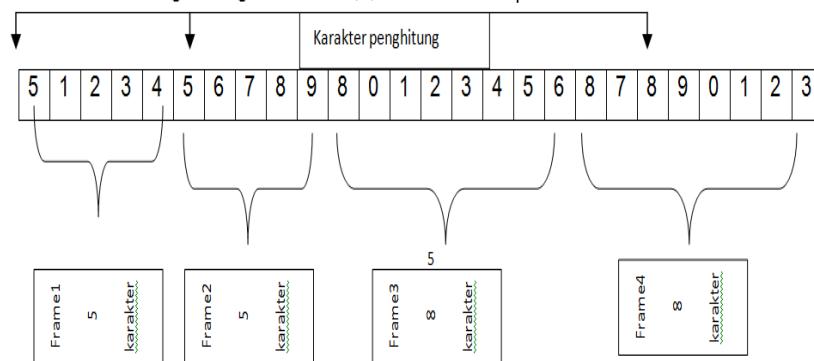
Pendekatan yang umum dipakai adalah data link layer memecahkan aliran bit menjadi frame-frame dan menghitung checksum setiap frame-nya. Memecahkan-mecahkan aliran bit menjadi frame-frame lebih sulit dibanding dengan apa yang di bayangkan.

Untuk memecah-mecahkan aliran bit ini, digunakan metode-metode khusus. Ada empat buah metode yang dipakai dalam pemecahan bit menjadi frame, yaitu:

1. Karakter penghitung.
2. Pemberian karakter awal dan akhir, dengan pengisian karakter.
3. Pemberian flag awal dan akhir, dengan pengisian bit.
4. Pelanggaran pengkodean physical layer.

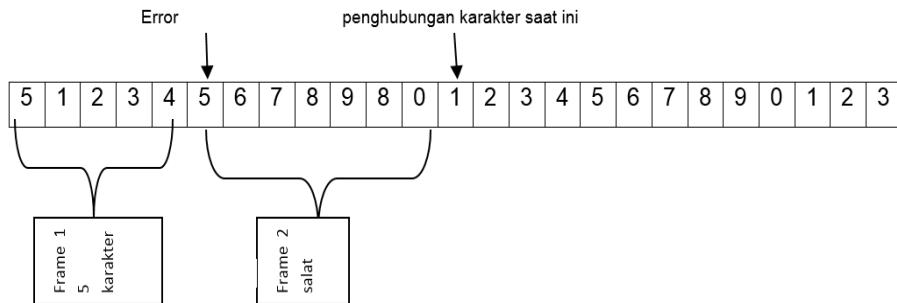
1.2.1. Karakter penghitung

Metode ini menggunakan sebuah field pada header untuk menspesifikasikan jumlah karakter di dalam frame. ketika data link layer pada komputer yang dituju melihat karakter penghitung, maka data link layer akan mengetahui jumlah karakter yang mengikutinya dan kemudian juga akan mengetahui posisi ujung frame-nya. Teknik ini bisa dilihat pada Gambar 4.3 di bawah ini, dimana ada empat buah frame yang masing-masing berukuran 5,5,8 dan 8 karakter.



Gambar 3 sebuah aliran karakter tanpa error

Masalah yang akan timbul pada aliran karakter ini apabila terjadi error transmisi. Misalnya, bila hitungan karakter 5 pada frame kedua menjadi 7 (Gambar 4.4), maka tempat yang dituju tidak sinkron dan tidak akan dapat mengetahui awal frame berikutnya. Oleh karena permasalahan ini, metode hitungan karakter sudah jarang dilakukan.



Gambar 4 sebuah aliran karakter dengan sebuah

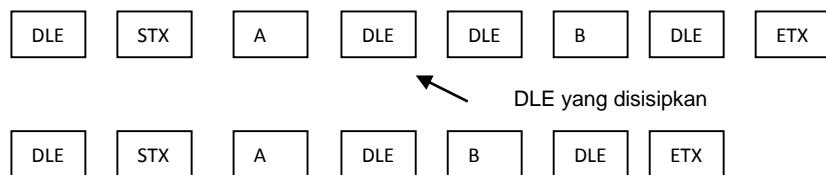
1.2.2. Pemberian karakter awal dan akhir

Metode yang kedua ini mengatasi masalah resinkronisasi setelah terjadi suatu error dengan membuat masing-masing frame diawali dengan deretan karakter DLE, STX, ASCII, dan ETX.

Masalah yang akan terjadi pada metode ini adalah ketika data biner ditransmisikan. Karakter-karakter DLE, STX, DLE dan ETX yang terdapat pada data akan mudah sekali mengganggu framing. Salah satu cara untuk mengatasi masalah ini adalah dengan membuat data link layer, yaitu pengirim menyisipkan sebuah karakter DLE ASCII tepat



sebelum karakter DLE pada data. Teknik ini disebut character stuffing (pengisian karakter) dan cara pengisiannya dapat dilihat pada gambar 4.5

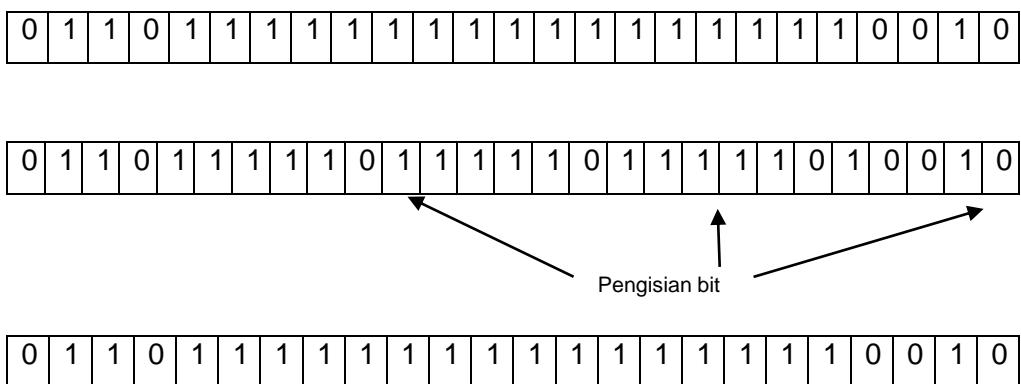


Gambar .5 pengisian karakter

1.2.2.1. Pemberian flag awal dan akhir

Teknik baru memungkinkan frame data berisi sejumlah bit dan mengijinkan kode karakter dengan sejumlah bit per karakter. Setiap frame diawali dan diakhiri oleh pola bit khusus, 01111110, yang disebut byte flag. Kapanpun data link layer pada pengirim menemukan lima buah flag yang berurutan pada data, maka data link layer secara otomatis mengisikan sebuah bit 0 ke aliran bit keluar. Pengisian bit ini analog dengan pengisian karakter, dimana sebuah DLE diisikan ke aliran karakter keluar sebelum DLE pada data (Gambar 4.6)

Ketika penerima melihat lima buah bit 1 masuk yang berurutan, yang diikuti oleh sebuah bit 0, maka penerima secara otomatis menghapus bit 0 tersebut. Bila data pengguna berisi pola flag, 01111110, maka flag ini akan ditransmisikan kembali sebagai 0111111010 tapi akan disimpan di memori penerima sebagai 01111110.



Gambar 6 pengisian bit

1.2.2.2. pelanggaran pengkodean physical layer

Metode yang terakhir hanya bisa digunakan bagi jaringan yang encoding pada medium fisiknya mengandung pengulangan. Misalnya, sebagian LAN melakukan encode bit 1 data dengan menggunakan 2 bit fisik. Umumnya, bit 1 merupakan pasangan tinggi rendah dan bit 0 adalah pasangan rendah-tinggi. Kombinasi

pasangan tinggi-tinggi dan rendah-rendah tidak digunakan bagi data.

1.3. Paket data

Pada saat data akan ditransmisikan, maka data akan dibagi menjadi paket yang kecil-kecil. Hal ini dilakukan karena:

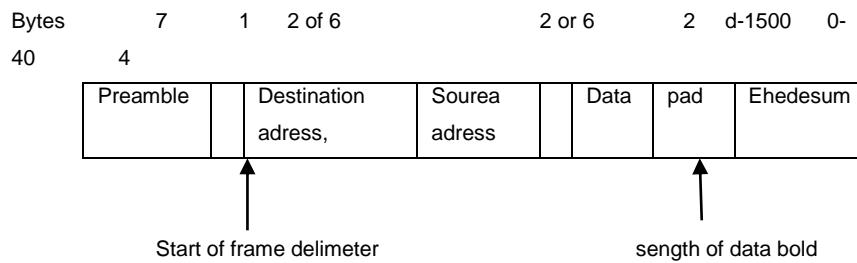
1. Jaringan tertentu hanya dapat menerima paket dengan panjang tertentu (misal ARPANET hanya mampu mengirim paket sebanyak 8063).
2. Jenis flow control tertentu akan efisien jika berita yang dikirim dibagi dalam paket-paket yang kecil (misal pada select repeat ARQ bila terjadi kerusakan data pada saat transmisi, maka transmitter hanya perlu mengirim kembali paket data tersebut)
3. Agar pengiriman jaringan tidak didominasi oleh user tertentu, karena dengan paket data maka setiap user dapat di batasi jumlah paket data yang akan dikirimkan sehingga dapat bergantian dengan user lainnya dalam memanfaatkan jaringan tersebut.
4. Paket data yang kecil hanya memerlukan buffer yang kecil pada bagian receiver.

Akan tetapi dalam melakukan pemotongan data menjadi paket data menjadi paket data, harus memperhatikan bahwa data tidak boleh dipotong terlalu kecil, karena :

1. Setiap data memerlukan bit overhead (tiap paket harus disertai dengan SYN, ADDRESS, CONTROL FIELD, CRC, FLAG, dan lain-lain). Pengiriman paketakan efisien jika bagian data lebih besar dari bit overhead.
2. Bila paket data terlalu kecil maka waktu pemrosesan yang di perlukan untuk pemrosesan sebuah paket yang besar.

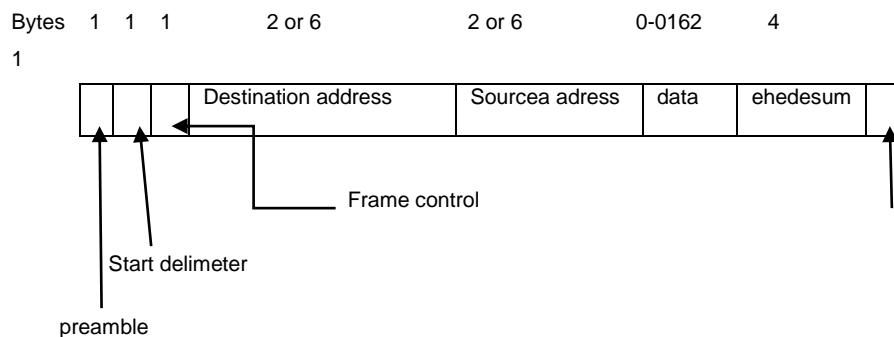
Berikut ini beberapa contoh paket (frame) data dari standart IEEE 802 yang secara umum telah banyak digunakan sebagai standart paket data.

1. Format frame data standart IEEE 802.3 (Gambar 4.7).



Gambar 7 format frame IEEE 802.3 (gambar 4.7)

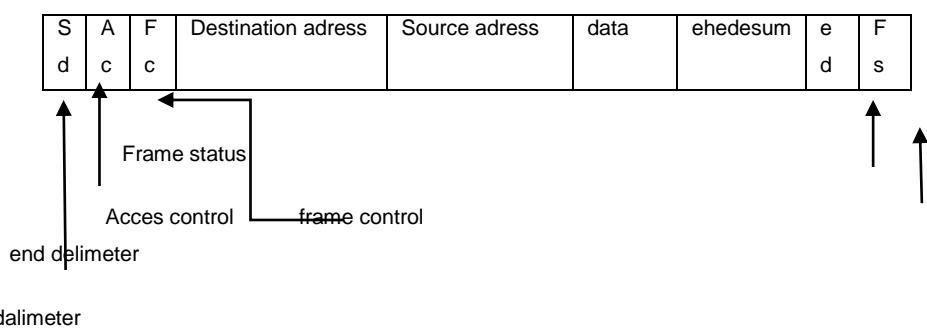
2. Format frame data standart IEEE 802.4 (Gambar 4.8)



Gambar 8 format frame IEEE 802.4

3. Format frame data standart IEEE 802.5 (gambar 4.9).

Bytes 1 1 1 2 or 6 no limit 4
1 1



Gambar .9 format frame IEEE 802.5

1.4. Sistem penyandian bit data

Penyandian adalah proses penggambaran dari satu set simbol menjadi set simbol yang lain. Sandi yang digunakan tergantung dari interface atau hubungan yang dikehendaki. Sistem penyandian yang utama dan banyak dipakai adalah:

1. American standart code for information interchange (ASCII)
2. Sandi baudot code (CCITT Alfabet no 2/Telex code)
3. Sandi 4 atau 8
4. Binary code decimal
5. EBCDIC

Pada umumnya sistem sandi ASCII merupakan standart umum yang sering dipergunakan. Pemilihan sandi tergantung pada kecepatan dan kehandalah yang di miliki.

1.4.1. American standart code for information tnterchange (ASCII)

hampir sama dengan CCITT alfabeth no 5, ASCII merupakan sandi 7 bit, sehingga terdapat 2 pangkat 7 yang berarti ada 128 macam simbol yang dapat disandikan dengan sistem sandi ini, sedangkan bit ke 8 merupakan bit paritas. Sandi ini dapat dikatakan yang paling banyak dipakai sebagai standart pensiyanan pada peralatan komunikasi data. Untuk transmisi asinkron tiap karakter disandikan dalam 10 atau 11 bit yang terdiri dari 1 bit awal, 7 bit data, 1 bit paritas, 1 atau 2 bit akhir.

Sandi ASCII dapat dilihat pada tabel 4.1 berikut ini.

		000	001	010	011	100	101	110	111
1234	octal	0	1	2	3	4	5	6	7
0000	00	NUL	SOH	STX	EXT	EOT	ENG	ACK	BEL
0001	01	BS	HT	LF	VT	FF	CR	SO	SI
0010	02	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
0011	03	CAN	EM	SUB	ESC	FS	GS	RS	VS
0100	04	SP	!	"	#	\$	%	&	'
0101	05	()	*	+	,	-	.	/
0110	06	0	1	2	3	4	5	6	7
0111	07	8	9	:	;	<	=	>	?
1000	10	@	A	B	C	D	E	F	G
1001	11	H	I	J	K	L	M	N	O
1010	12	P	Q	R	S	T	U	V	W
1011	13	X	Y	Z	[/]	^	-
1100	14	'	a	B	C	d	E	f	G
1101	15	H	i	J	K	I	M	n	O
1110	16	P	q	R	S	t	U	v	W
1111	17	X	y	Z	{		}	~	DEL

Tabel 4.1 sandi ASCII

1.4.1.1. karakter data

pada komunikasi data diadakan pertukaran informasi antara si pengirim dan penerima. Informasi yang dipertukarkan terdiri atas dua grup baik untuk ASCII maupun EBCDIC yaitu karakter data dan karakter kendali.

Untuk melakukan encoding dari berita yang dikirimkan, ada terminal yang hanya mengenal huruf besar saja, sehingga set untuk huruf kecil tidak di pakai. Tiap karakter mempunyai kombinasi yang unik, sehingga tidak akan terdapat salah penafsiran.

1.4.1.2. karakter kendali

karakter kendali digunakan untuk mengendalikan transmisi data, bentuk atau format data, hubungan data dan fungsi fisik terminal. Karakter kendali dibedakan atas:

1. *transmission control* berfungsi untuk membentuk paket data yang mudah di kenal dan mengendalikan transmisi data dalam media transmisi.
2. *Format effectors* berfungsi untuk mengendalikan tata letak fisik informasi pada print out atau tampilan layar.
3. *Device control* berfungsi untuk mengendalikan peralatan tambahan pada terminal.
4. *Information separators* berfungsi untuk mengelompokan data secara logis.

Dibawah ini akan diuraikan masing-masing karakter yang telah disebutkan di atas.

4.3.1.2.1 transmission control

bentuk berita tergantung pada sistem komunikasi yang dipilih. Isi atau informasi berita biasanya disebut teks. Kalau berita tersebut di pecah-pecah dalam blok yang lebih kecil.

Blok teks ini kemudian disalurkan melalui saluran transmisi ke tempat tujuannya. Untuk mengenali dan mengetahui apa yang harus dilakukan dengan blok data yang diterima digunakan karakter *transmission control*.

Karakter *transmission control* yang umum dikenal adalah sebagai berikut:

1. SOH (Start of header); digunakan sebagai karakter pertama yang menunjukkan bahwa karakter berikutnya adalah *header*.
2. STK (start of text); digunakan untuk mengakhiri *header* dan menunjukkan awal dari informasi atau teks.
3. ETX (End of text); digunakan untuk mengakhiri teks.

4. EOT (End of Transmission); untuk menyatakan bahwa transmisi dari teks baik satu maupun lebih telah berakhir.
5. ENQ (Enquiry); untuk meminta agar *remote station* mengirimkan tanggapan. Tanggapannya dapat berupa identifikasinya atau status.
6. ACK (Acknowledge); untuk memberikan tanggapan positif ke pengirim dari si penerima.
7. NACK (Negative acknowledge); merupakan tanggapan negatif dari penerima ke pengirim.
8. SYN (Synchonous); digunakan dalam transmisi sinkron untuk menjaga atau memperoleh sinkronisasi antar peralatan terminal.
9. ETB (End of Transmission Block); digunakan untuk menyatakan akhir dari blok data yang ditransmisikan, bila berita dipecah menjadi beberapa blok.
10. DLE (Data Link Escape); mengubah arti karakter berikutnya. Hanya digunakan untuk lebih mengendalikan transmisi data.

Gambar 4.10 dibawah ini merupakan contoh berita yang dikirimkan dalam beberapa blok.

SO	HEAD	ST	TE	BT	ST	TE	BT	ST	TE	ST
H	ER	X	XT	B	X	XT	B	K	XT	K

Gambar 10 *format frame IEEE 802.5*

Dari gambar diatas, kita bisa melihat bahwa HEADER disini dapat berisi informasi mengenai terminal, misalnya alamat, prioritas, tanggal dan sebagainya. Selain itu, tidak semua sistem memerlukan ETB untuk berita yang terdiri dari beberapa blok. Sebagian ada yang menggunakan EXT sehingga dalam teks harus ada informasi yang dapat digunakan untuk merangkai berita.

1.5. Format Effectors

Didalam format effectors, simbol-simbol yang digunakan adalah sebagai berikut.

1. BS (*Back Space*). *Back Space* dapat menyebabkan print head atau kursor mundur satu posisi.
2. HT (*Horizontal Tabulation*). *Horizontal Tabulation* digunakan untuk maju ke posisi yang telah di tentukan.
3. LF (*Line Feed*). *Line Feed* digunakan untuk maju satu baris (spasi)
4. VT (*Vertical Tabulation*). *Vertical Tabulation* digunakan untuk maju beberapa baris (spasi)
5. FF (*Form Feed*). *Form Feed* digunakan untuk maju satu halaman (halaman baru)
6. CR (*Carriage Return*). *Carriage Return* digunakan untuk *print head* atau kursor menuju ke awal baris.

1.5.1.. Device Control

Device Control/1 sampai devive control 4 digunakan untuk mengendalikan fungsi fisik terminal misalnya menghidupkan, mematikan, dan lainnya tergantung dari si perancang.

1.5.2..Informasi Separator

Secara umum informasi disusun berdasarkan hirarkhi sebagai berikut :

1. Unit separator (US). Unit separator digunakan untuk memisahkan tiap-tiap unit informasi.
2. Record Separator (RS). Record separator digunakan untuk memisahkan tiap *record* yang terdiri atas beberapa unit informasi.
3. Group Separator (GS). Group separator digunakan untuk memisahkan beberapa *record* yang membentuk suatu group.
4. File Separator (FS). File separator digunakan untuk memisahkan beberapa grup yang membentuk suatu file.

1.5.3.Sandi Baudot Code (CCITT Alf abet No 2/Telex Code)

Nama sistem sandi ini berasal dari nama ahli teknik pos dari Perancis yang bekerja di bidang telepon sekitar tahun 1874. Sistem sandi

ini terdiri atas 5 bit, sehingga terdapat 2 pangkal 5 sama dengan 32 macam karakter yang dapat disandikan. Tetapi 32 macam karakter tersebut tidak cukup untuk mewakili semua karakter alfanumerik, sehingga kode ini dibagi menjadi dua bagian, yaitu karakter huruf (*letter character*) dengan kode biner 11111 dan karakter bentuk (*figure character*) dengan kode biner 11011.

Jadi untuk menyandikan huruf harus diawali dengan kode 11111 dan untuk menyandikan karakter bentuk harus diawali dengan kode 11011. Pada transmisi sinkron, satu sandi transmisi satu karakter terdiri dari 1 bit awal, 5 bit kode huruf/bentuk 5 bit data dan 1 bit akhir.

Adapun bentuk sandi baudot Code dapat dilihat pada tabel 4.2 berikut ini.

Binary	Letter Character	Figure Character
00000	Blank	Blank
00001	E	3
00010	Line feed	Line feed
00011	A	-
00100	SPACE	SPACE
00101	S	'
00110	I	8
00111	U	7
01000	<	<
01001	D	+
01010	R	4
01011	J	BELL
01100	N	,
01101	F	%
01110	C	:
01111	K	(
10000	T	5
10001	Z	+
10010	L)
10011	"	2

Binary	Letter Character	Figure Character
D10100	H	L
10101	Y	6
10110	P	0
10111	Q	1
11000	O	9
11001	B	?
11010	G	\$
11011		
11100	M	.
11101	X	/
11110	V	=
11111		

1.5.5. Sandi 4 atau 8

Sistem sandi 4 dan 8 ini merupakan sistem sandi yang standar dari IBM dengan kombinasi yang diperbolehkan adalah 4 “1” dan 4 buah “0”, sehingga hanya 70 karakter yang dapat diberi sandi. Pada transmisi sinkron, sebuah karakter dibutuhkan 10 bit, yang terdiri dari 1 bit awal, 8 bit data dan 1 bit akhir.

1.5.6. BCD (Binary Code Decimal)

BCD merupakan sistem sandi dengan 6 bit, sehingga kombinasi yang dapat digunakan sebagai sandi banyaknya adalah 2 pangkat 6 sama dengan 64 kombinasi. Pada transmisi sinkron sebuah karakter dibutuhkan 9 bit, yang terdiri dari 1 bit awal, 6 bit data, 1 bit paritas dan 1 bit akhir.

1.5.7. EBCDIC

EBCDIC merupakan sistem sandi 8 bit, sehingga dapat untuk menyandikan sebanyak 2 pangkat 8 sama dengan 256 karakter.

Pada transmisi asinkron, sebuah karakter dibutuhkan 11 bit, yang terdiri dari 1 bit awal, 8 bit data, 1 bit paritas dan 1 bit akhir. Sistem sandi ini juga merupakan sistem sandi yang umum digunakan disamping sistem ASCII.

1.6. Penanganan Kesalahan Transmisi

Dalam komunikasi data, baik melalui komputer atau peralatan komunikasi lainnya, kesalahan pengiriman dan penerimaan data tak dapat terhindarkan. Kesalahan ini mungkin saja disebabkan oleh *cross talk*, gema, panas, noise atau derau dan masih banyak gangguan lainnya. Dalam usaha menghindari gangguan ini dapat dilakukan dengan menggunakan tiga cara, yaitu metode *echo*, metode paritas dan *frame check*.

1.6.1 Metode Echo

Metode *Echo* merupakan metode yang paling sederhana dimana pengguna komputer dapat melihat proses pertukaran data tersebut melalui layar monitor. Dengan tampilanya semua data

g. Rangkuman

Befungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi format yang disebut sebagai frame. Selain itu, pada level ini terjadi koreksi kesalahan, flow control, pengalamanan perangkat keras (seperti halnya Media Access Control Address (MAC Address)), dan menentukan bagaimana perangkat-perangkat jaringan seperti hub, bridge, repeater, dan switch layer 2 beroperasi.

Data *link layer* adalah sebagai fasilitas transmisi *raw data* dan mentransformasi data tersebut ke saluran yang bebas dari kesalahan transmisi. Sebelum diteruskan ke *network layer*, data *link layer* melaksanakan tugas ini dengan memungkinkan pengirim memecah-mecah

data input menjadi sejumlah data *frame* (biasanya berjumlah ratusan atau ribuan byte).

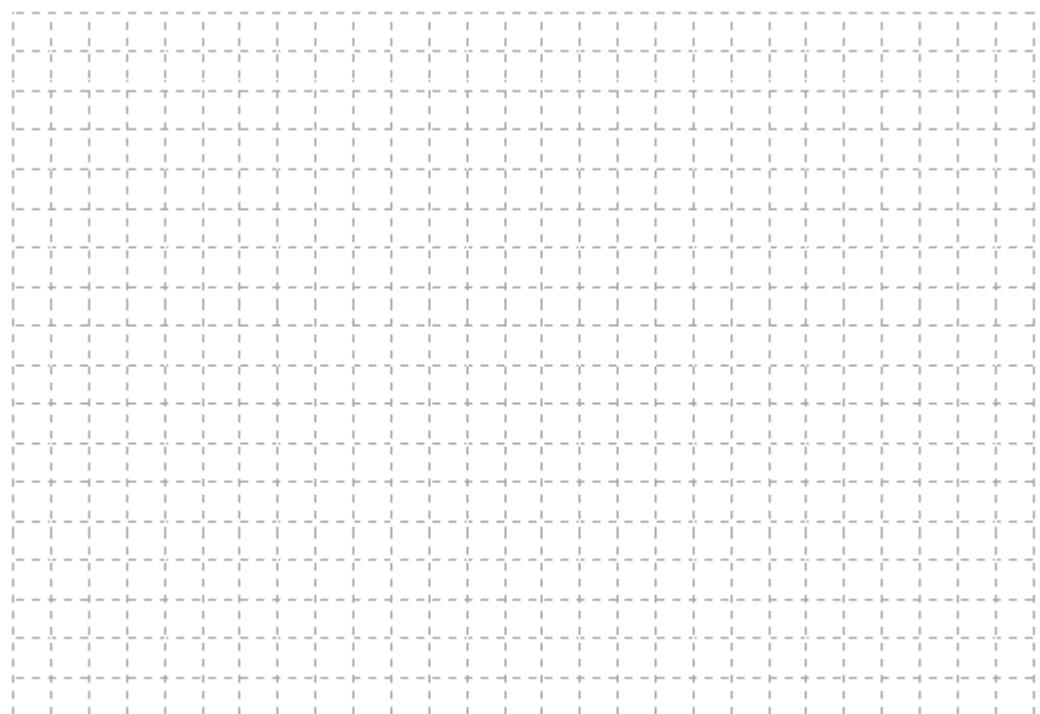
Kemudian *data link layer* mentransmisikan *frame* tersebut secara berurutan dan memproses *acknowledgement frame* yang dikirim kembali oleh penerima. Karena *physical layer* menerima dan mengirim aliran bit tanpa mengindah arti atau arsitektur *frame*, maka tergantung pada *data link layer*-lah untuk membuat dan kengenali batas-batas *frame* itu. Hal ini bisa dilakukan dengan cara membubuhkan bit khusus ke awal dan akhir *frame*.

h. 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 !

i. Lembar Jawaban Test Formatif

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

A large grid for writing answers, consisting of approximately 20 columns and 20 rows of small squares.

- Test Essay (LJ.02).



- Test Essay (LJ.03).



- Test Essay (LJ.04).



- Test Essay (LJ.05).



j. Lembar kerja siswa