



Pengertian Kesalahan

Ketika melakukan pentransmisi data seringkali kita menjumpai data yang tidak sesuai dengan yang kita harapkan (salah sasaran). Hal ini disebabkan karena adanya gangguan dalam suatu saluran transmisi. Istilah **error** atau kesalahan memang mungkin terjadi pada suatu transmisi data. Kesalahan (galat) adalah hal yang terjadi apabila suatu hal tidak bertindak semestinya, entah salah sasaran, kehilangan satu bit, atau juga berubah datanya.

Tentunya ketika terjadi kesalahan maka kesalahan tersebut harus terdeteksi oleh sistem komunikasi data untuk kemudian dikoreksi supaya kembali menjadi data yang benar. Sistem komunikasi seringkali membuat kesalahan, memiliki data rate yang berbeda, dan terdapat delay (tundaan) yang terjadi ketika suatu bit dikirimkan dengan saat bit diterima. Keterbatasan ini mempengaruhi sekali bagi efisiensi pemindahan data.

Data Link Control

Pengiriman data melalui link komunikasi data yang terlaksana dengan penambahan kontrol layer dalam tiap perangkat keras komunikasi dinyatakan sebagai **data link control** atau **data link protocol**. Data link adalah medium transmisi antara stasiun-stasiun ketika suatu prosedur data link digunakan. Ketika menggunakan data link protocol ada beberapa hal yang harus diperhatikan, yaitu:

a. Frame synchronization

Data dikirimkan dalam blok-blok yang disebut frame. Awal dan akhir tiap frame harus dapat diidentifikasi.

b. Memakai variasi dari konfigurasi line

c. Flow control

Stasiun pengirim harus tidak mengirim frame-frame pada kecepatan yang lebih cepat daripada kecepatan penerimaan data pada stasiun penerima.

d. Error control

Bit-bit error yang dihasilkan oleh sistem transmisi harus diperbaiki

e. Kontrol dan data terletak pada link yang sama

Sinyal-sinyal kontrol tidak diharapkan mempunyai jalur komunikasi yang terpisah. Karena itu, receiver harus mampu membedakan kontrol informasi dari data yang sedang ditransmisi.

f. Addressing (pengalamatan)

Pada jalur komunikasi yang multipoint (banyak jalur), identitas dari dua stasiun (baik stasiun pengirim atau penerima) harus mampu membedakan kontrol informasi dari data yang sedang transmisi.

g. Manajemen link

Permulaan, pemeliharaan, dan penghentian dari pertukaran data memerlukan koordinasi dan kerjasama di antara stasiun pengirim dengan stasiun penerima. Diperlukan prosedur untuk manajemen pertukaran data.

Konfigurasi-konfigurasi Line

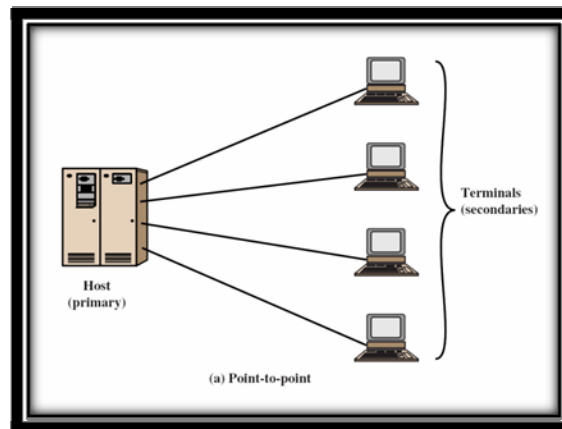
Ada tiga karakteristik yang membedakan berbagai konfigurasi data link, yaitu:

a. Topology

Menyatakan pengaturan fisik dari stasiun pada suatu link. Ada dua konfigurasi topologi yaitu:

1. Point to point

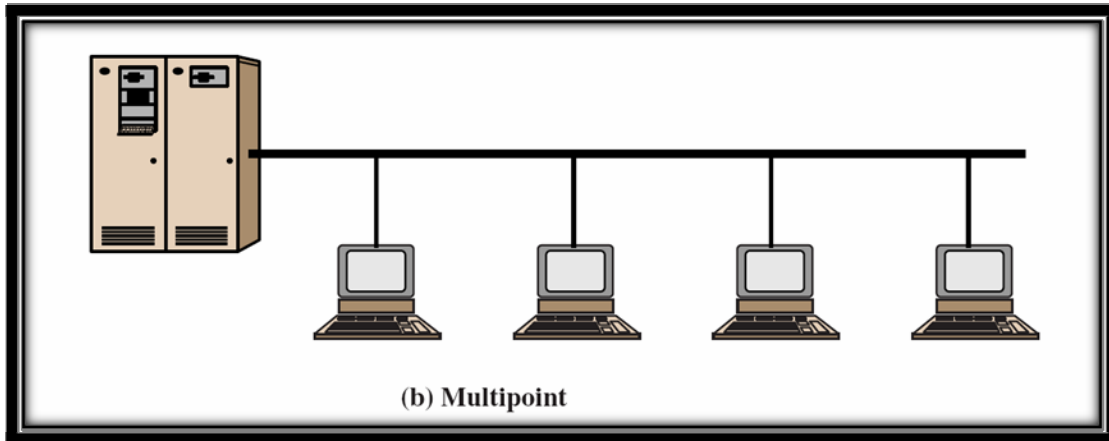
Jika dalam suatu transmisi hanya ada satu stasiun pengirim dan satu stasiun penerima.



Gambar 5.1 Konfigurasi point to point

2. Multipoint

Jika dalam suatu transmisi ada lebih dari dua stasiun. Dipakai dalam suatu komputer dan suatu rangkaian terminal.



Gambar 5.2 Konfigurasi multipoint

b. Duplexity

Menyatakan arah dan timing dari aliran sinyal. Jenis-jenisnya adalah sebagai berikut:

1. Simplex transmission
2. Half duplex link
3. Full duplex link.

Untuk transmisi yang menggunakan sinyal digital dapat memakai full duplex dan half duplex. Untuk yang menggunakan sinyal analog penentuan duplexity tergantung pada frekuensi bila stasiun pengirim (transmisi) dan penerimaan pada frekuensi yang sama.

c. Line Discipline

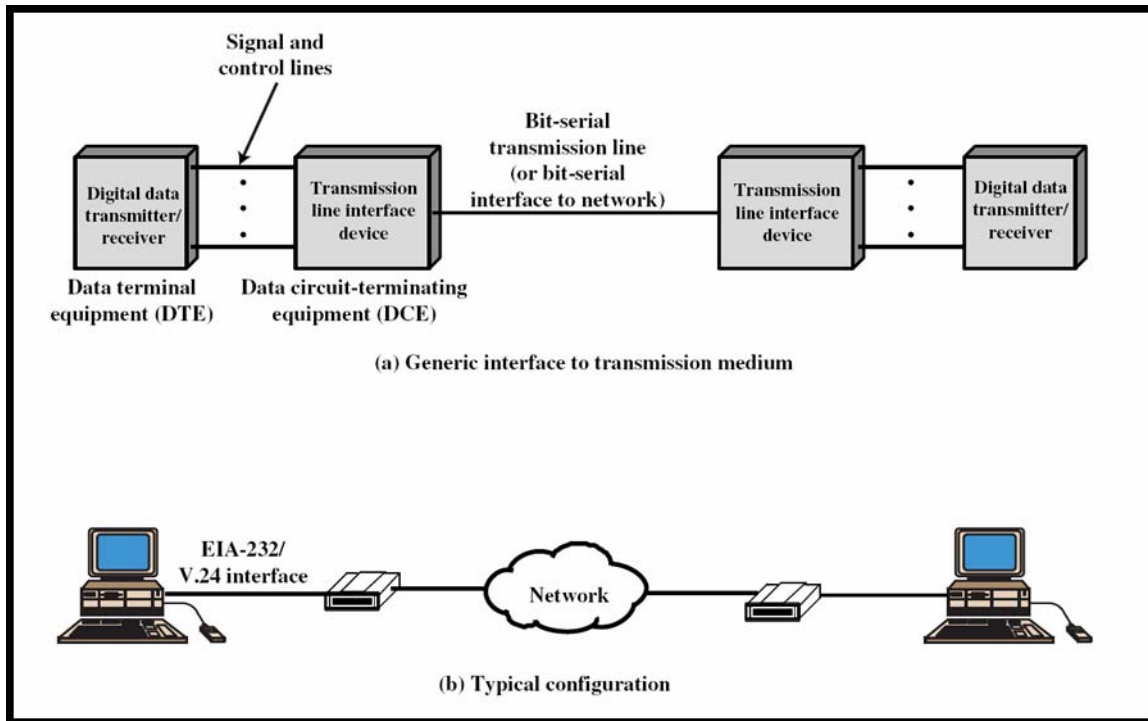
Beberapa tata tertib yang diperlukan dalam penggunaan link transmisi. Pada mode half duplex hanya satu stasiun yang dapat mentransmisi pada satu waktu. Baik mode hal atau full duplex, suatu stasiun hanya mentransmisi jika mengetahui bahwa receiver telah siap untuk menerima.

Flow Control

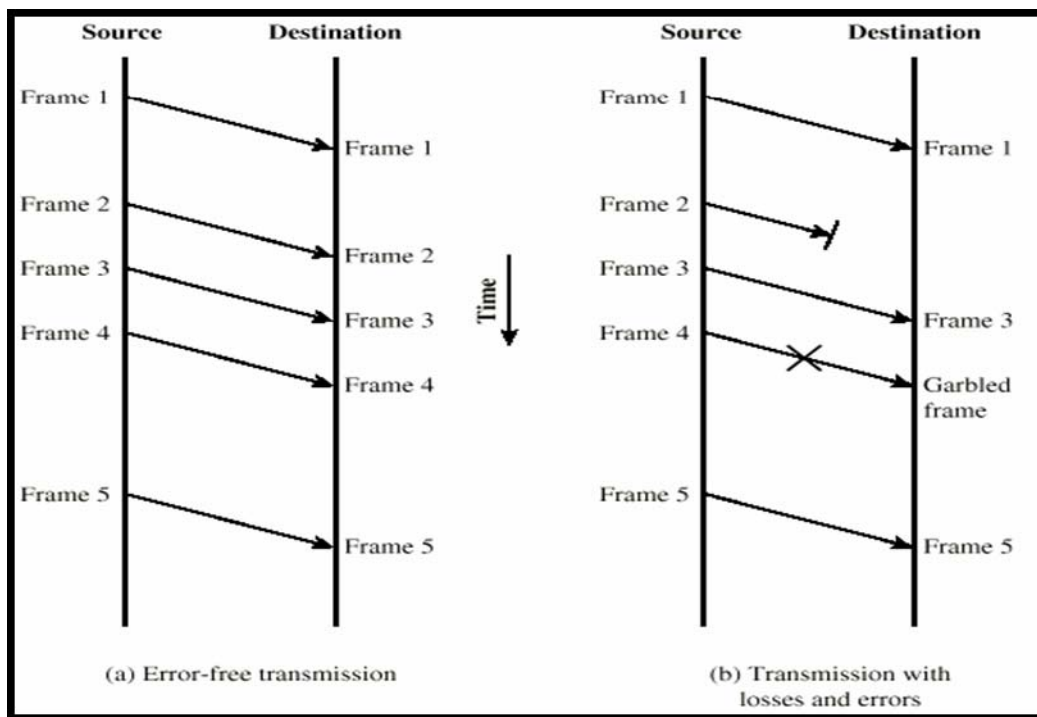
Flow control adalah suatu teknik untuk memastikan/meyakinkan bahwa suatu stasiun transmisi tidak menumpuk data pada suatu stasiun penerima. Tanpa flow control, buffer (memori penyangga) dari receiver akan penuh sementara masih

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

banyak data lama yang akan diproses. Ketika data diterima, harus dilaksanakan sejumlah proses sebelum buffer dapat dikosongkan dan siap menerima banyak data.



Gambar 5.3 Antarmuka komunikasi data

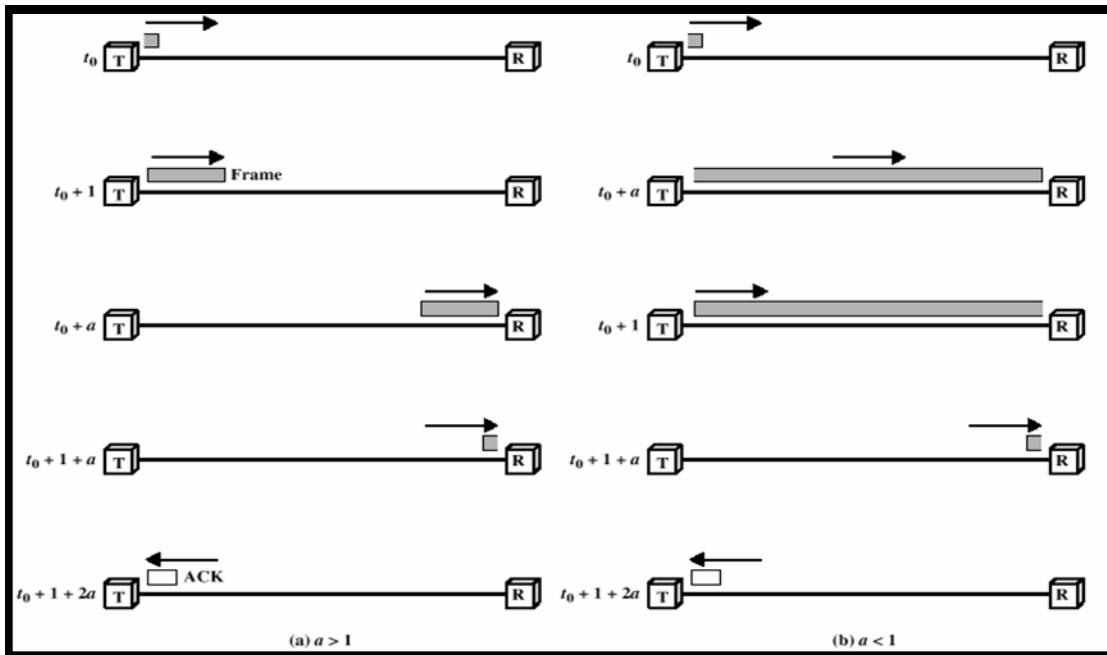


Gambar 5.4 Model transmisi frame

Ada beberapa bentuk dari flow control antara lain:

a. Stop and wait flow control

Cara kerjanya suatu sumber mengirimkan frame. Setelah diterima, penerima memberi isyarat untuk menerima frame lainnya dengan mengirim acknowledgement ke frame yang baru diterima. Pengirim atau sumber harus menunggu sampai menerima acknowledgement sebelum mengirim frame berikutnya. Penerima kemudian dapat menghentikan aliran data dengan tidak memberi acknowledgement. Frame yang dikirimkan tidak akan menjadi masalah jika ukuran datanya tidak terlalu besar. Jika data yang dikirim besar, maka secara otomatis jumlah framenya akan bertambah sehingga menyebabkan stop and wait control menjadi tidak efisien.

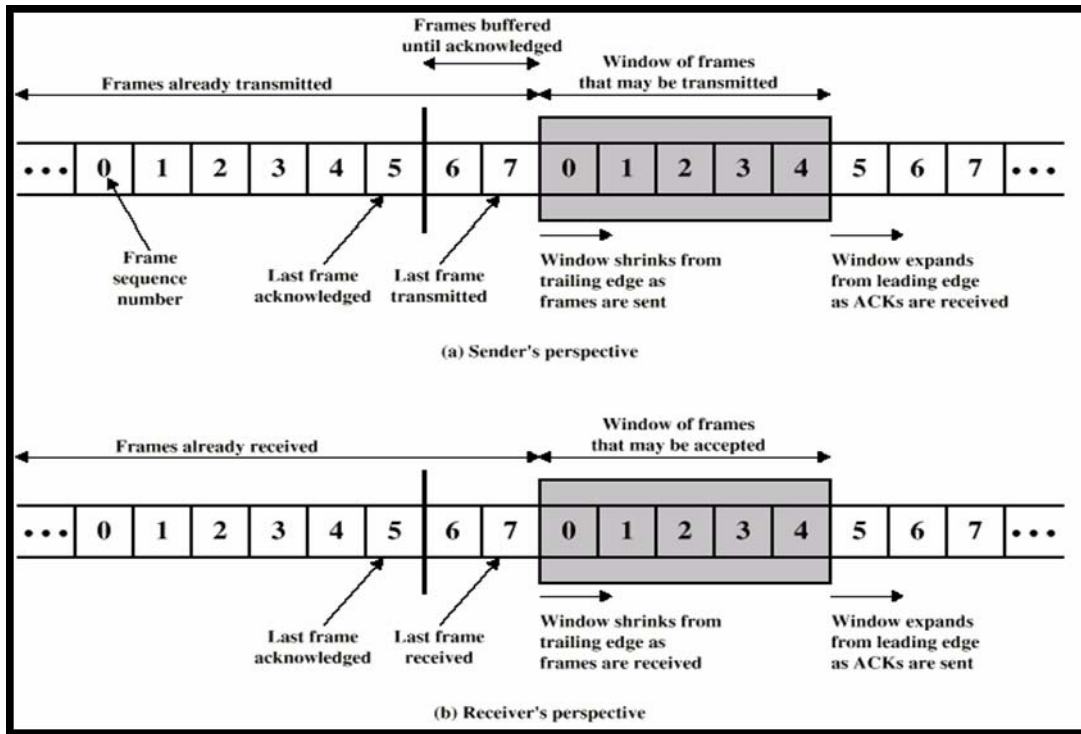


Gambar 5.5 Penggunaan stop and wait control

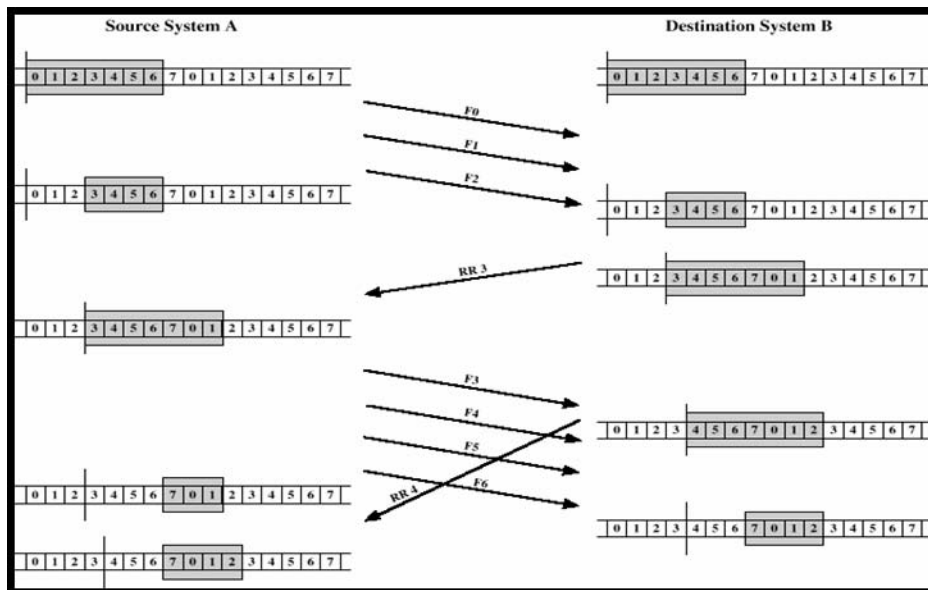
b. Sliding window flow control

Masalah utama yang dimiliki oleh stop and wait control adalah bahwa hanya satu frame yang dapat dikirimkan pada saat yang sama. Dalam keadaan antrian bit yang akan dikirimkan lebih besar daripada panjang frame ($a > 1$) maka diperlukan suatu langkah efisiensi (memperbolehkan pengiriman lebih dari satu frame pada saat yang sama). Dalam langkah ini, ditambahkan juga label pada setiap frame yang telah masuk sebagai penanda sudah sejauh mana frame tersebut diterima. Sliding window flow control ini mengizinkan

untuk pengiriman lebih dari satu frame. Receiver juga memiliki sebuah buffer untuk menampung antrian frame yang masuk dengan syarat setiap frame yang masuk diberi nomor. Nomor tersebut nantinya akan digunakan sebagai penanda yang akan diloncati tiap ukuran field (k). frame yang masuk akan dinomori dengan modulo 2^k .



Gambar 5.6 Diagram sliding window



Gambar 5.7 Contoh sliding window

Deteksi Error

Pada subbab sebelumnya dibahas tentang penggunaan flow control, sekarang akan dibahas bagaimana cara mendeteksi kesalahan yang terjadi pada flow control tersebut. Ada dua pendekatan yang bisa digunakan untuk mendeteksi error, yaitu:

a. Forward Error Control

Karakter yang ditransmisikan atau disebut juga frame, berisi informasi tambahan sehingga apabila penerima mengalami kesalahan, penerima tidak hanya bisa mendeteksi kesalahannya saja tetapi juga bisa menjelaskan letak kesalahan tersebut.

b. Feedback (backward) Error Control

Setiap karakter atau frame memiliki informasi yang cukup untuk memperbolehkan penerima mendeteksi bila menemukan kesalahan tetapi tidak lokasi kesalahannya. Feedback error control dibagi menjadi 2 bagian, yaitu:

1. Teknik yang digunakan untuk deteksi kesalahan
2. Kontrol algoritma yang telah disediakan untuk mengontrol transmisi ulang.

Metode Pendeteksian Error

Ada dua metode deteksi kesalahan yang sering digunakan, yaitu:

a. Echo

Metode sederhana dengan sistem interaktif. Operator memasukkan data melalui sebuah terminal dan mengirimkan ke komputer lain, setelah itu komputer akan menampilkan data yang dikirim kembali ke terminal sehingga operator dapat memeriksa apakah data yang dikirimkan benar atau tidak.

b. Error otomatis

Metode dengan tambahan bit pariti (pariti ganjil atau pariti genap). Ada beberapa metode yang bisa digunakan dalam pendeteksian error, antara lain:

1. Vertical redundancy checking
2. Longitudinal redundancy checking
3. Cyclic redundancy checking

Vertical Redundancy Checking

Metode ini lebih umum disebut dengan parity checking karena menggunakan sistem pengecekan paritas dan merupakan sistem untuk mencari kesalahan data yang paling sederhana. Dalam satu byte terdapat satu bit pariti. Bit ini nilainya tergantung kepada ganjil atau genapnya jumlah bit satu dalam satu byte. Pengecekan parity terbagi menjadi dua yaitu:

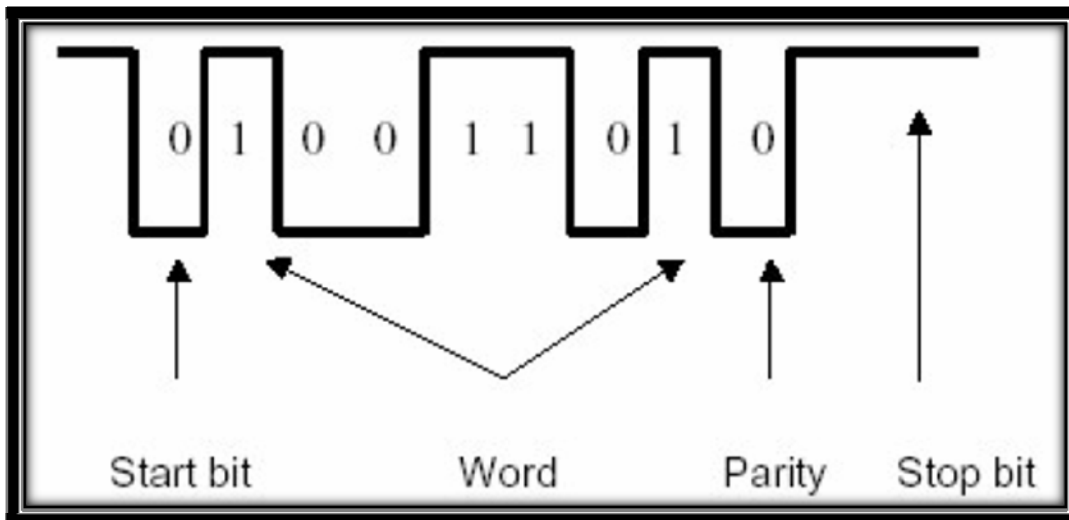
a. Odd parity (pariti ganjil)

Jumlah bit satu dalam satu byte data harus berjumlah ganjil.

b. Even parity (pariti genap)

Jumlah bit satu dalam satu byte data harus berjumlah genap.

Sebenarnya hampir semua sistem komputer mampu menjalankan metode ini, jadi jika di dalam saluran transmisi terjadi suatu gangguan maka jumlah bit yang diterima akan menjadi tidak sesuai. Tetapi metode ini punya kelemahan terutama jika jumlah bit yang rusak jumlahnya genap, maka kerusakan ini menjadi tidak terdefinisi (tergantung dari jenis pengecekannya). Karakter yang mengandung kesalahan 2 atau 5 bit bila hanya dilihat dari sisi genap atau ganjilnya jumlah bit satunya saja maka kesalahannya tidak akan terlihat.



Gambar 5.8 Karakter 'M' dengan pariti genap

Sebagai contoh pada pengiriman teks CAT maka pendeteksian errornya sebagai berikut:

Tabel 5.1 Pengecekan error dengan VRC dan HRC

Data	C	A	T	HRC
b0	1	1	0	0
b1	1	0	0	1
b2	0	0	1	1
b3	0	0	0	0
b4	0	0	1	1
b5	0	0	0	0
b6	1	1	1	1
VRC	0	1	0	1

Penjelasan:

- Ubah teks CAT menjadi kode ASCII (dalam biner).
- Setelah itu buat tabel seperti pada tabel 5.1.
- VRC melakukan pemeriksaan dengan parity ganjil, sedangkan HRC melakukan pemeriksaan dengan parity genap.
- Pada titik pertemuan antara HRC dan VRC ada satu bit yang disebut BCC.
- Setelah melakukan pengecekan pariti maka bagi HRC menjadi 2 bagian (masing- masing 4 bit) setelah itu ubah 4 digit tersebut menjadi bilangan hexadesimal. Setelah itu susunlah dua digit tersebut dari arah bawah ke atas. Itulah yang akan menjadi **HRC Value**. HRC value harus mengandung satu digit bilangan ganjil dan satu digit bilangan genap (urutan tidak dipermasalahkan). Kalau tidak memenuhi persyaratan tadi, maka data dianggap error.

Longitudinal Redundancy Checking

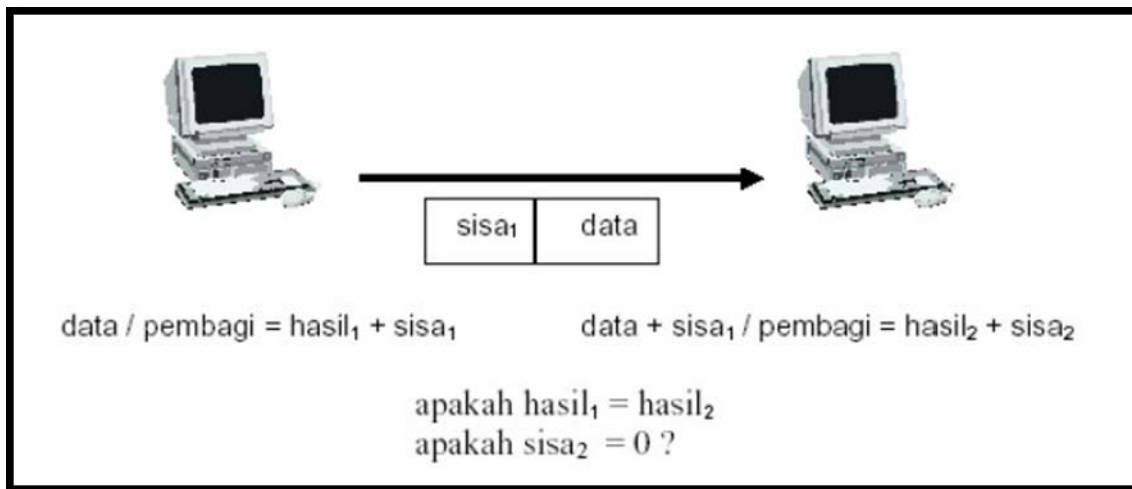
Metode ini sebenarnya digunakan untuk memperbaiki kelemahan yang ada pada VRC. Pada metode LRC ini, data dikirimkan per blok (frame) berisi 8 byte dan setiap frame terdapat satu bit pariti. Fungsi dari bit pariti tersebut adalah sebagai kontrol kesalahan seperti pada parity checking.

Walaupun masih memiliki beberapa kelemahan namun sistem LRC lebih baik dari VRC sebab apabila terjadi kesalahan yang tidak terlihat oleh parity bit maka akan diketahui oleh pariti byte. Dalam pentransmision data, LRC membutuhkan

banyak tambahan bit pada setiap data yang dikirim, misalkan untuk mengirimkan 7 karakter (49 bit) maka diperlukan bit tambahan sebanyak 15 bit sehingga metode LRC ini tidak banyak dipakai walupun bermanfaat.

Cyclic Redundancy Checking

Sistem ini banyak dipakai dalam komunikasi data karena prosesnya **cukup** sederhana dan tidak membutuhkan banyak tambahan bit sebagai bit pariti. Pada sistem CRC data dikirimkan per frame dan setiap frame terdiri dari deretan bit panjang. Pada akhir blok ditambahkan beberapa control bit untuk menjamin kebenaran data. Control bit dibentuk oleh komputer pengirim berdasarkan perhitungan atas data yang dikirim. Setelah data sampai pada komputer penerima akan dilakukan perhitungan seperti perhitungan di sisi pengirim. Hasil perhitungan yang didapatkan dibandingkan dengan control bit, bila sama berarti data dikirim tanpa mengalami kesalahan.



Gambar 5.9 Metode CRC

Untuk bisa menghitung dengan metode CRC ada baiknya mengetahui terlebih dahulu tentang operasi XOR. Adapun tabel eksklusif XOR adalah sebagai berikut:

Tabel 5.2 Tabel XOR

XOR	0	1
0	0	1
1	1	0

Beberapa istilah dalam CRC:

- a. $T(x)$
Data atau aliran bit.
- b. $P(x)$
Bit konstanta berpola polynomial (contoh: 110011).
- c. CRC
 $= P(x) / T(x)_{\text{Bit } 0} P(x) \setminus$
- d. $T^1(x)$
 $= T(x)_{\text{CRC}}$
- e. CRC^1
 $= P(x) / T^1(x) \setminus$. Apabila nilai CRC^1 menghasilkan 0 maka data yang dikirim bebas error.

Sebagai contoh dilakukan suatu pengiriman pesan sebagai berikut:

$T(x)$: 10110111

$P(x)$: 110011 (5 bit) \rightarrow 1 1 0 0 1 1

$2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$ tertinggi adalah 2^5

CRC:

$110011 / 1011011100000 \setminus \rightarrow$ bit data + bit 0 sebanyak pangkat 2 tertinggi

$$\begin{array}{r}
 110011 \\
 \hline
 111101 \\
 110011 \\
 \hline
 111010 \\
 110011 \\
 \hline
 100100 \\
 110011 \\
 \hline
 101110 \\
 110011 \\
 \hline
 111010 \\
 110011 \\
 \hline
 \end{array}$$

1001 \rightarrow karena jumlahnya harus sama dengan pangkat 2 Tertinggi maka tambahkan 1 digit 0 di awal (01001).

$T^1(x) = 1011011101001$

CRC¹ :

11011 / 1011011101001 \

$$\begin{array}{r} 110011 \\ \hline 111101 \\ 110011 \\ \hline 111010 \\ 110011 \\ \hline 100110 \\ 110011 \\ \hline 101010 \\ 110011 \\ \hline 110011 \\ 110011 \\ \hline \end{array}$$

0 → karena CRC¹ 0 maka data bebas error.

Koreksi Kesalahan

Bila dijumpai kesalahan pada data yang telah diterima, maka perlu adanya error recovery atau pengkoreksian kesalahan agar jangan sampai kesalahan ini menyebabkan dampak yang besar bagi pengiriman datanya. Metode yang digunakan antara lain:

a. Substitusi simbol

Bila ada data yang rusak maka komputer penerima mengganti bagian tersebut dengan karakter lain, seperti karakter SUB yang nerupa tanda tanya terbalik. Jika pemakai menjumpai karakter ini (pada program pengolah kata), maka berarti data yang diterima mengalami kerusakan, selanjutnya perbaikan dilakukan sendiri.

b. Mengirim data koreksi

Data yang dikirimkan harus ditambah dengan kode tertentu dan data duplikat. Bila penerima menjumpai kesalahan pada data yang diterima maka perbaikan dilakukan dengan mengganti bagian yang rusak dengan data duplikat. Cara ini jarang digunakan.

c. Kirim ulang

Cara ini merupakan cara yang paling sederhana, yaitu apabila komputer penerima menemukan kesalahan pada data yang diterima maka selanjutnya meminta komputer pengirim untuk mengulang pengiriman data.

Forward Error Correction

Salah satu cara untuk mengoreksi kesalahan adalah dengan menggunakan metode forward error correction. Ada beberapa notasi yang harus terlebih dahulu dipahami, yaitu:

- a. m : Jumlah bit sample.
- b. n : Jumlah bit hamming.
- c. Bit sample : $\frac{1}{2} * \text{jumlah bit data}$.
- d. Bit hamming : $2^n > m+n+1$.

Adapun langkah perhitungannya adalah sebagai berikut:

- a. Contoh karakter yang akan dikoreksi adalah CAT. Ubah karakter CAT menjadi biner kode ASCII dan tambahkan pariti ganjil (letakkan terbalik dari bit pariti sampai bit pertama).

Tabel 5.3 Karakter CAT

bit	Pariti	b6	b5	b4	b3	b2	b1	b0
C	0	1	0	0	0	0	1	1
A	1	1	0	0	0	0	0	1
T	0	1	0	1	0	0	0	1

- b. Gabungkan seluruh bit dalam satu deret. Setelah itu hitung jumlah bit dari keseluruhan karakter. Untuk karakter CAT jumlah bitnya adalah 24 bit.
- c. Hitung m dengan rumus yang disediakan:

$$\begin{aligned} M &= \frac{1}{2} * \text{jumlah bit data} \\ &= \frac{1}{2} * 24 \\ &= 12 \text{ bit.} \end{aligned}$$

Maka bit samplanya adalah 12 bit pertama dari karakter CAT yaitu **010000111100**.

- d. Hitung n dengan rumus yang disediakan dimulai dari $n=1$:

1. $2^n > m + n + 1$
 $2^1 > 12 + 1 + 1$ (n tidak terpenuhi. Lanjutkan ke n=2)
2. $2^n > m + n + 1$
 $2^2 > 12 + 2 + 1$ (n tidak terpenuhi. Lanjutkan ke n=3)
3. $2^n > m + n + 1$
 $2^3 > 12 + 3 + 1$ (n tidak terpenuhi. Lanjutkan ke n=4)
4. $2^n > m + n + 1$
 $2^4 > 12 + 4 + 1$ (n tidak terpenuhi. Lanjutkan ke n=5)
5. $2^n > m + n + 1$
 $2^5 > 12 + 5 + 1$ (n terpenuhi. Maka n=5).

Kesimpulan:

1. $n=5 \rightarrow$ bit hamming berjumlah 5 bit (h5 h4 h3 h2 h1).
 2. $m + n = 17$ bit.
- e. Gabungkan bit sample dengan bit hamming secara acak (khusus bit sample).
 Sebagai contoh lihat tabel di bawah ini:

Tabel 5.4 Gabungan bit sample dengan bit hamming

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	1	h5	0	h4	0	0	h3	0	1	h2	1	1	h1	1	0	0

- f. Setelah itu hitung bit hamming. Tentukan letak bit 1 pada deret tersebut.
 Pada deret di atas, letak bit 1 ada pada posisi ke 3, 5, 6, 8, 16. Ubah posisi tersebut menjadi biner 5 bit lalu kenakan operasi XOR terhadap posisi tersebut. Sebagai contoh lihat perhitungan di bawah ini:

Tabel 5.5 Perhitungan XOR posisi bit '1'

Posisi bit '1'	Biner 5 bit				
3	0	0	0	1	1
5	0	0	1	0	1
XOR	0	0	1	1	0
6	0	0	1	1	0
XOR	0	0	0	0	0
8	0	1	0	0	0
XOR	0	1	0	0	0
16	1	0	0	0	0
XOR	1	1	1	0	0
Bit Hamming	h5	h4	h3	h2	h1

- g. Setelah bit hamming diketahui, maka terbentuklah data kirim. Data kirim adalah bit sample digabungkan dengan bit hamming. Jadi data kirim dari contoh kasus ini adalah sebagai berikut:

Tabel 5.6 Data kirim

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	1	1	0	1	0	0	0	0	1	0	1	1	0	1	0	0

- h. Langkah terakhir adalah tentukan lagi letak bit '1' pada deret di atas kecuali untuk bit hamming tidak perlu dihitung. Lalu ubah posisi tersebut menjadi 5 bit biner lalu kenakan operasi XOR. Untuk permulaanya lakukan operasi XOR terhadap bit hamming dengan posisi yang pertama kali dijumpai.

Tabel 5.5 Perhitungan XOR posisi bit '1' setelah ditambah bit hamming

Posisi bit '1'	Biner 5 bit				
Bit hamming	1	1	0	0	0
3	0	0	0	1	1
XOR	1	1	0	1	1
5	0	0	1	0	1
XOR	1	1	1	1	0
6	0	0	1	1	0
XOR	1	1	0	0	0
8	0	1	0	0	0
XOR	1	0	0	0	0
16	1	0	0	0	0
XOR	0	0	0	0	0
Hasil Akhir	0				

- i. Jika hasil akhir sama dengan 0 maka tidak ada kesalahan yang terjadi dan tidak perlu adanya koreksi terhadap data yang dikirim atau diterima. Sedangkan apabila hasil akhir tidak sama dengan 0 maka terjadi kesalahan dan perlu adanya koreksi kesalahan. Karena menggunakan koreksi kesalahan dengan metode forward error correction maka penerima data bisa membetulkan kesalahannya sendiri karena sudah mengetahui letak kesalahan kalau menghitung manual seperti ini memang tidak akan terlihat letaknya, akan tetapi peralatan komunikasi data di sisi penerima akan bisa mendeteksi dan mengkoreksi kesalahannya melalui bit hamming).