Pertemuan XI

DETEKSI & KOREKSI KESALAHAN

13.1. Deteksi Kesalahan

Pengiriman informasi yg menggunakan sinyal digital atau analog selalu mengalami perubahan yg dialami oleh informasi tsb. Perubahan tsb bisa itu disebabkan oleh :

- media pengirimannnya itu sendiri
- gangguan thd media tsb
- sinyal informasi itu sendiri yg melemah krn jarak tempuh
- peralatan perantara lain yg digunakan dlm pengiriman informasi

Media pengiriman data sangat dipengaruhi oleh gangguan gejala listrik spt :

- kilat
- pengaruh medan listik motor atau peralatan elektronika lain
- pengaruh media lain yg membawa sinyal listrik yg berdekatan dg-nya

Semua gejala ini disebut derau yg dpt menyebabkan informasi mengalami perubahan atau kesalahan.

Oleh krn itu tdpt usaha utk mencegah, mendeteksi, bahkan memperbaiki kesalahan yg terjadi pd data yg dikirimkan.

Cara mencegah terjadinya kesalahan dilakukan dg memperbaiki peralatan pengiriman & penerimaan, serta media pengiriman datanya. Selain itu, sistem yg dirancang hrs dpt melacak kesalahan & memperbaikinya.

Salah satu deteksi kesalahan dlm komunikasi data adalah ; menggunakan tambahan informasi yg tdk ada kaitannya dg isi informasi yg dikirimkan. Data tambahan inilah yg menunjukkan ada atau tdknya kesalahan pd data yg dikirimkan tadi.

Data tambahan ini disebut dg pariti ; yaitu penambahan 1 atau beberapa *i "non-information carrying bit"*, shg penerima dpt melakukan perhitungan matematis utk memeriksa ke-*valid*-an data yg diterimanya.

13.2. Cara Deteksi Kesalahan

Ada beberapa metode utk mengetahui adanya suatu kesalahan

a. Metode *Echo*
b. Metode deteksi *error* otomatis
c. Framing check

a. Metode *Echo*

Metode yg paling sederhana & digunakan secara interaktif. Operator memasukkan data melalui terminal yg kemudian mengirimkannya ke komputer. Komputer kemudian mengirimkannya kembali ke terminal & ditampilkan ke monitor. Operator dpt melihat apakah data yg dikirimkannya benar.

b. Metode deteksi *error* otomatis

Sistem komputer lebih menghendaki sedikit mungkin melibatkan manusia. Oleh krn itu digunakan sistem bit pariti, yaitu bit tambahan yg digunakan utk mendeteksi kesalahan. Tdpt 2 macam cara penambahan bit pariti:

Pariti ganjil (*Odd parity*)
Bit pariti tambahan, spy banyaknya bit "1" tiap karakter/data, ganjil.

Parity genap (*Even parity*)
Bit pariti tambahan, spy banyaknya bit "1" tiap karakter/data, genap.

Ada 3 macam teknik deteksi kesalahan dg menggunakan bit pariti :

- *Vertical Redudancy Check* (VRC)
- *Longitudinal Redudancy Check* (LRC)
- *Cyclic Redudancy Check* (CRC)

•***Character Parity* (*Vertical Redudancy Check* / VRC)**

Mrpkn metode pemeriksaan kesalahan per-karakter & digunakan pd sistem yg berorientasi karakter, misalnya terminal.

Dg cara ini, setiap karakter yg dikirimkan (terdiri dari 7 bit) diberi tambahan 1 bit pariti yg akan diperiksa oleh penerima utk mengetahui kebenaran karakter yg diterima tsb.

Cara ini hanya dpt melacak kesalahan 1 bit & hanya digunakan utk melacak kesalahan yg terjadi pd pengiriman data berkecepatan menengah.

Misalnya ASCII huruf A, kodenya adalah hex 41 :

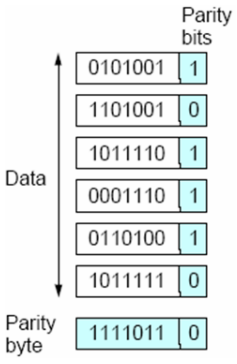
1000001 ASCII 7 bit (tdpt 2 bit 1)

1000001 1 tambahkan 1, jml bit 1 jadi ganjil (*odd parity*)

1000001 0 tambahkan 0, jml bit 1 jadi genap (*even parity*)

Contoh "*even parity* ":
Pengirim:
Data: 1 1 0 0 0 0 1
b1 b2 b3 b4 b5 b6 b7
Ada 3 bit "1" (ganjil), tambahkan bit 1, jml bit "1" jadi genap.
Kirim: Data & Parity = 11000011

Penerima:
Proses (algoritma) even parity:
Hitung jumlah bit 1 => x
Jika x = genap disimpulkan tidak ada error
Jika x = ganjil, terjadi error
Terima: Data & Parity = 11100011
Error?



Gambar 13.1. *Vertical Redudancy Check* dgn *Even Parity*

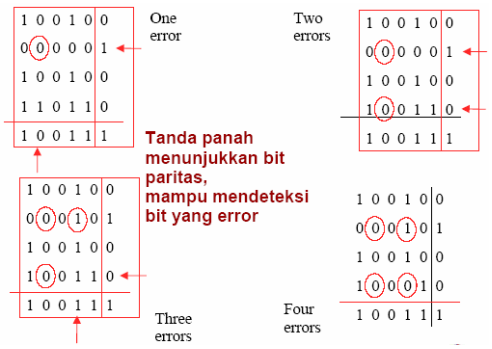
Penerima memeriksa pariti dari karakter yg diterima, bila tdk sesuai dg ketentuan maka akan diketahui adanya kesalahan pd waktu penyaluran data.

VRC mempunyai kekurangan, yaitu bila ada 2 bit yg terganggu maka tdk dpt terlacak, shg dianggap paritinya akan benar.

•***Longitudinal Redudancy Check*** / **LCR**

Utk memperbaiki kinerja VRC, digunakan LRC utk data yg dikirim secara blok. Cara ini mirip dg VRC, hanya saja penambahan bit dilakukan pada akhir setiap blok karakter yg dikirimkan.

Dg cara ini maka kesalahan lebih dari 1 bit juga dpt ditemukan, shg kecepatan pengiriman data dpt dipertinggi.



Gambar 13.2. *Longitudinal Redudancy Check* dgn Even Parity

•***Cyclic Redudancy Check*** / **CRC**

CRC digunakan utk pengiriman data berkecepatan tinggi. CRC disebut sbg pengujian berorientasi bit, krn dasar pemeriksaan kemungkinan kesalahan adalah bit atau karakter & menggunakan rumus matematika khusus.

Dlm metode ini 1 blok informasi dilihat sbg deretan bit yg ditransmisikan. Bit yg akan disalurkan dimasukkan ke dlm register geser siklis yg disebut generator CRC. Operasi matematik dikerjakan atas deretan bit tsb.

Operasi CRC ini didasarkan atas pembagian deretan bit dg sebuah fungsi khusus. Hasil bagi pembagian diabaikan. Sisa disalurkan sbg *Block Check Sequence* (BCS) yaitu akhir dari deretan bit isi register geser.

Berdasarkan pemeriksaan perbandingan hasil perhitungan rumus matematika dr pd saat dikirim & setelah diterima akan dpt ditentukan adanya kesalahan atau tidak. Pd penerima, deretan bit data termasuk BCS juga dimasukkan ke dlm register geser siklis yg disebut penguji CRC. Hasil operasi matematik ini berupa isi register geser yg yg dpt diperkirakan ada tdknya kesalahan transmisi.

c. *Framing Check*

Digunakan pd transmisi asinkron dg adanya bit awal & bit akhir. Dg memeriksa ke-2 bit ini dpt diketahui apakah data diterima dg baik

Transmisi sinkron mempunyai berbagai bentuk bingkai sesuai dg ketentuan yg digunakan