

FORWARD ERROR CONTROL



BEC & FEC

- **Error-Correcting Codes** dinyatakan sebagai Forward Error Correction untuk mengindikasikan bahwa receiver sedang mengoreksi error.
- Contohnya : pada komunikasi **broadcast** digunakan **transmisi simplex**.

BEC & FEC

- Metode transmisi ulang dinyatakan sebagai **Backward Error Correction (BEC)** karena receiver memberi *informasi balik* ke transmitter yang kemudian *mentransmisi ulang data yang error*.

BEC & FEC

- **BEC** menyebabkan *delay pengiriman paket yang cukup besar* tergantung dari berapa kali paket tersebut harus dikirim
- Untuk sistem transmisi jarak jauh dimana delay propagasi sangat besar (satuan detik, menit atau jam) **BEC tidak bisa menjadi pilihan**

BEC & FEC

- ❑ Khusus *aplikasi multimedia*, dimana ketepatan waktu kedatangan lebih utama dibandingkan dengan ‘kebenaran’ data, ***BEC menyebabkan delay yang lewat batas toleransi waktu***

Forward Error Control (FEC)

- ❑ Solusi → **Forward Error Correction (FEC)** untuk memecahkan masalah ini
- ❑ **FEC** berprinsip dasar : penerima mampu *membetulkan sendiri kesalahan data* yang sudah diterima, karena selain menerima data juga menerima bit-bit redundansi yang diperlukan

Jenis – Jenis FEC

- Metoda **FEC** yang umum dikenal :
 - **Block Parity**
 - **Hamming Code**
 - **Turbo Code, RS Code, BCH Code**

Block Parity

- Sederhana, menggunakan perhitungan **pariti dasar**
- Menggunakan *pariti baris dan kolom* guna koreksi kesalahan
- Hanya mampu **mengkoreksi** kesalahan 1 bit, mampu **mendeteksi** kesalahan lebih dari 1 bit

Block Parity

- ❑ Efisiensi tergantung dari ukuran baris dan kolom yang digunakan, semakin banyak baris dan kolom akan semakin banyak bit pariti

Contoh : Block Parity

Parity Genap

					Bit Pariti ↓					
1	1	0	1	1	1	1	0	1	1	✓
1	0	1	1	1	1	0	0	1	0	X
1	1	0	1	1	1	1	0	1	1	✓
0	0	1	1	0	0	0	1	1	0	✓
1	0	0	0	1	1	0	1	0	1	✓ ← Bit Pariti
					✓	✓	X	✓	✓	✓

SEND : 1000 1 1110 → RECEIVE : 1010 1 1010

Hamming Code

Tetapan Hamming Code :

$$(m + r + 1) \leq 2^r$$

Dimana : m = panjang bit pesan

r = jumlah bit hamming (check bit)

Contoh : Hamming Code

7 bit ASCII Karakter A = $41_{\text{Hex}} = 1000001_{\text{Bin}}$

$$(m + r + 1) \leq 2^r$$

$$8 + r \leq 2^r$$



$$r = 4 \text{ (check bits) ???}$$

Code word = 7 data bits + 4 (check bits) = 11 bits.

Contoh : Hamming Code

Code word untuk Karakter A :

11	10	9	8	7	6	5	4	3	2	1
1	0	0	x	0	0	0	x	1	x	x

(Checkbit ditempatkan pada posisi persamaan deret biner polinomial ($2^{n-1} \dots\dots 2^4 + 2^3 + 2^2 + 2^1 + 2^0$))

Contoh : Hamming Code (Sender)

Bit 1 (pada pesan) berada pada posisi ke : **3** (0011_{bin}) dan posisi ke : **11** (1011_{bin}). Lakukan proses XOR untuk menentukan posisi Bit 1 (check bit)

$$\begin{array}{rcl} 3 & \rightarrow & 0011 \\ 11 & \rightarrow & \underline{1011} \\ & & \mathbf{1000} \rightarrow 8 \text{ (check bit)} \end{array}$$

Code word untuk Karakter ASCII A menjadi :

$$\mathbf{10010000100} \rightarrow \text{SEND}$$

Contoh : Hamming Code (Receiver)

Code word untuk Karakter ASCII A menjadi yang diterima :

10010000100 → RECEIVE

Pada sisi penerima bit 1 ditemukan pada posisi 11 (1011_{bin}), posisi 8 (1000_{bin}) dan posisi 3 (0011_{bin}).

Contoh : Hamming Code (Receiver)

Kemudian lakukan proses XOR pada posisi – posisi tersebut :

$$\begin{array}{r} 11 \quad \rightarrow 1011 \\ 8 \quad \rightarrow \underline{1000} \quad \text{xor} \\ \quad \quad 0011 \\ 3 \quad \rightarrow \underline{0011} \quad \text{xor} \\ \quad \quad 0000 \rightarrow 0 \end{array}$$

Nilai 0 artinya tidak terdapat error bit pada saat pengiriman data. Jika tidak sama dengan 0 maka diartikan bahwa terdapat posisi bit error pada pesan yang dikirimkan.

Latihan Soal : Hamming Code

1. Karakter ASCII (7bit) M = $4D_{\text{HEX}}$
2. Karakter ASCII (7bit) _ = $5F_{\text{HEX}}$
3. Karakter ASCII (7bit) ~ = $7E_{\text{HEX}}$