



Error Detection And Correction

Ridho Sholehurrohman, M.Mat

Outline

- ❑ Introduction
- ❑ Block Coding
- ❑ Linear Block Codes
- ❑ Cyclic Codes
- ❑ Checksum

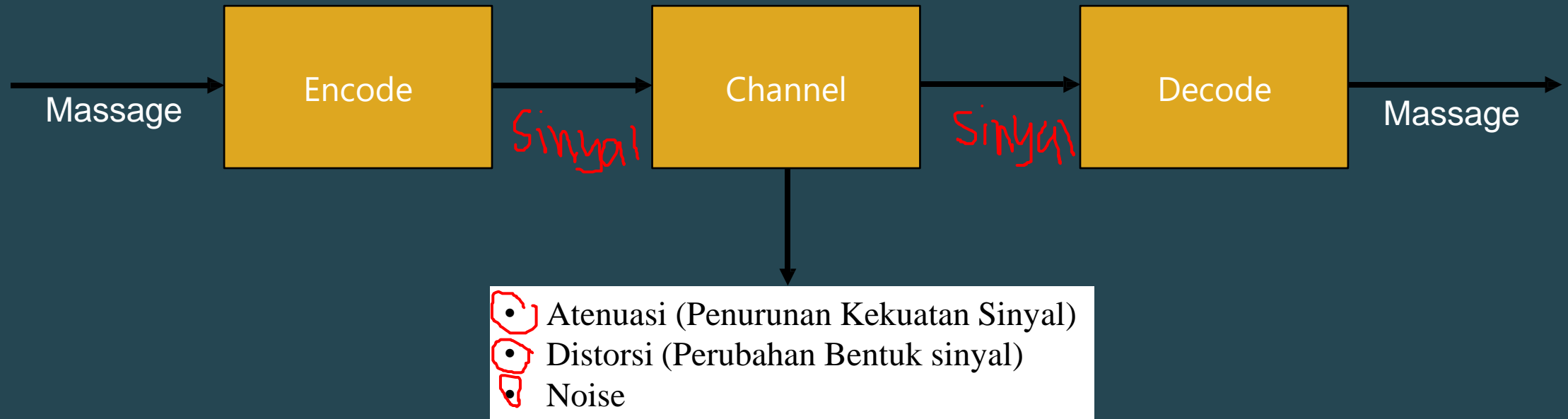




Introduction

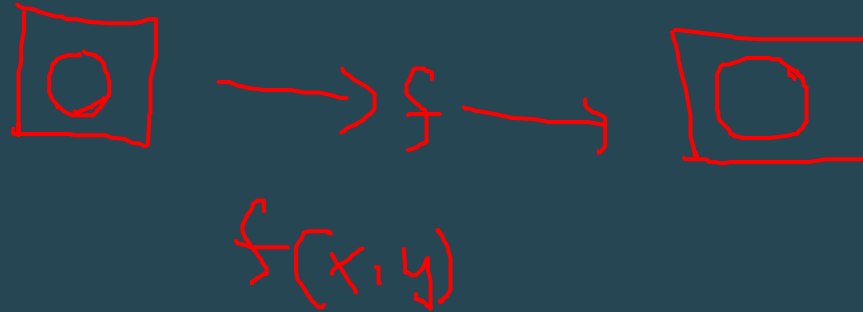
Ridho Sholehurrohman, M.Mat

Komunikasi Data



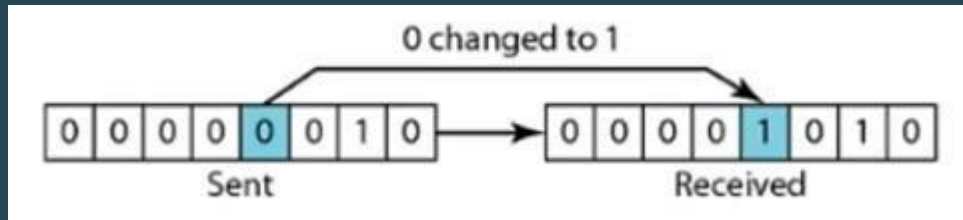
Mengapa Harus Ada Deteksi dan Koreksi Error

- Data dapat mengalami Kerusakan selama transmisi
- Beberapa aplikasi membutuhkan deteksi dan koreksi Error



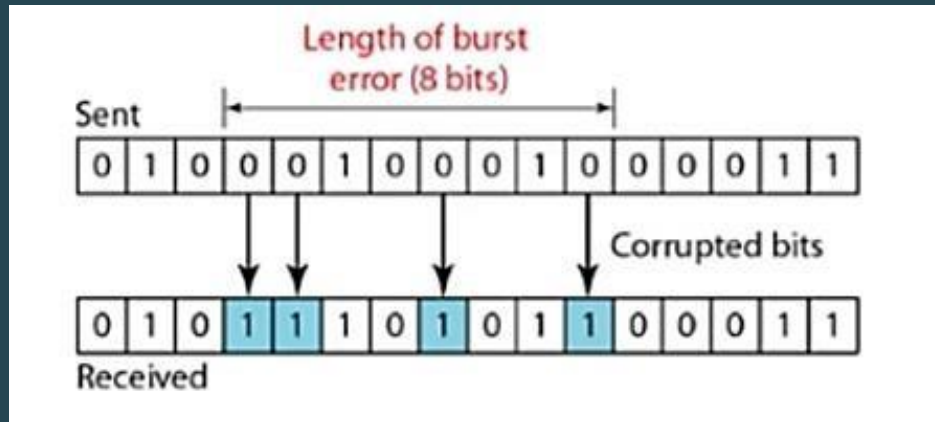
Jenis-Jenis Error

- Pada Error bit tunggal, hanya 1 bit dalam unit data yang berubah



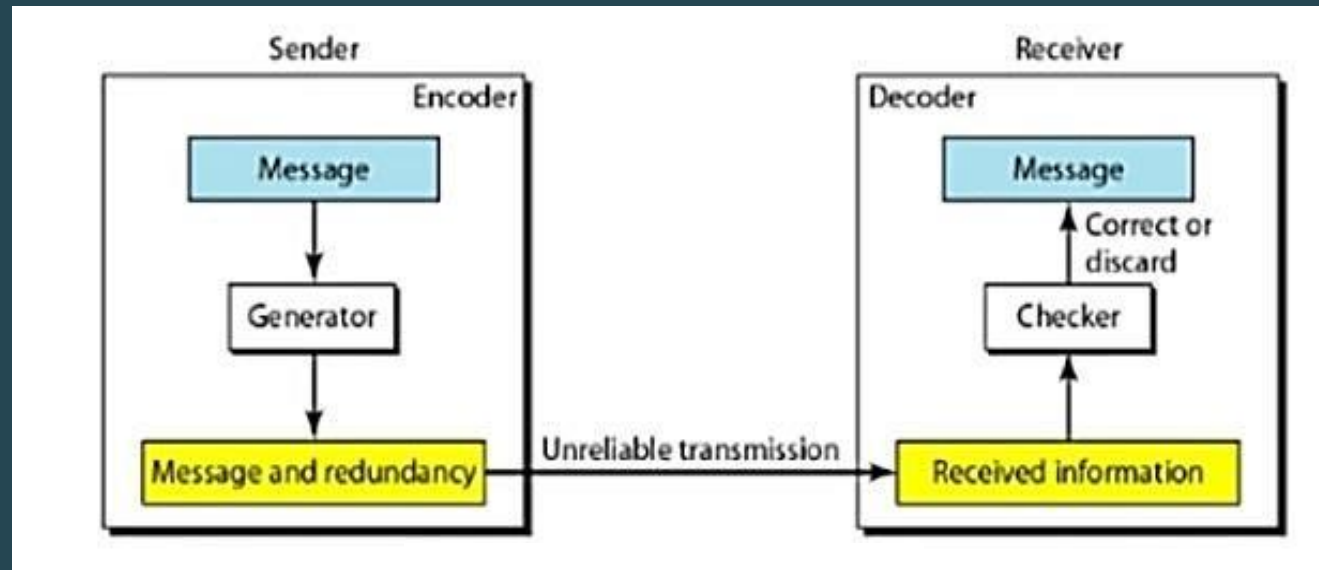
- *Burst Error*, terdapat 2 atau lebih dalam unit data yang error

rangkai



Redundansi

- Bit tambahan digunakan dalam deteksi dan koreksi *error*
- Bit yang ditransmisikan → *Bit data + Bit Redundan*
- Diperoleh dari mekanisme pengkodean (coding)
- Coding : *Blok coding* dan *Convolution coding*



Deteksi Vs Koreksi

- Koreksi *error* lebih sulit dibanding dengan deteksi
- Deteksi *error* → ada tidaknya *error* (ya/tidak)
- Koreksi *error* → jumlah bit *error*, lokasi bit yang *error*
- Koreksi 8 bit data:
 - 1 bit error → 8 kemungkinan
 - 2 bit error → 28 kemungkinan

1 unit → 8

x	x	x	x	x	x	x	x
---	---	---	---	---	---	---	---

$$C_8^2 \rightarrow \frac{8!}{2!6!} = \frac{8 \cdot 7}{2 \cdot 1} = 28$$

100 bit → 100 bit → C_{100}^{100} ?

Forward Error Correction Vs Retransmission

Cara Koreksi Error:

- Forward Error Correction

Penerima menentukan data asli berdasarkan redundan bit.

Hanya jika jumlah bit error sedikit.

- Retransmission ✓

Meminta pengirim mengirimkan ulang pesan.

Modular Aritmatic

a. Dua bit yang sama, maka hasilnya 0

$$0 \oplus 0 = 0$$

$$1 \oplus 1 = 1$$

b. Dua bit yang berbeda, maka hasilnya 1

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

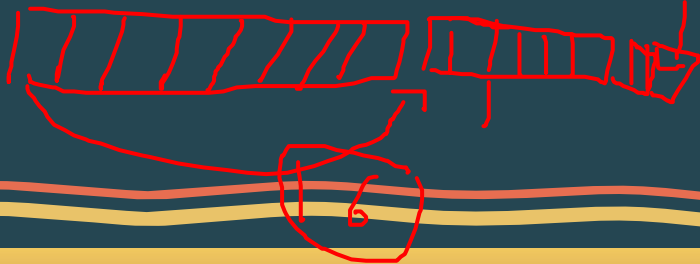
Contoh

$$\begin{array}{r} \oplus \quad \begin{array}{cccc} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 \end{array} \end{array}$$

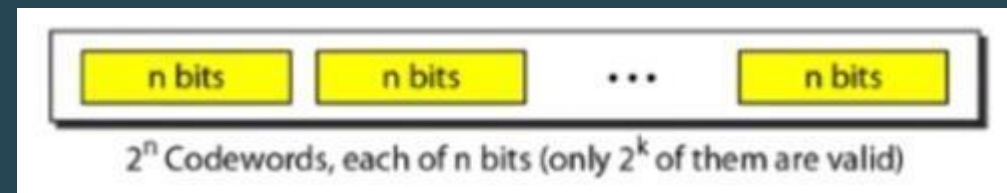
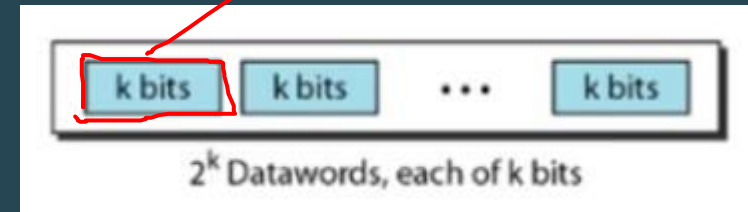
Block Coding

- ❑ Pesan dibagi kedalam beberapa Blok data
- ❑ Blok data dengan k bit disebut dengan *datawords*
- ❑ k bit Blok data + r Redundan = *Codewords*(n)
- ❑ $n = k + r$

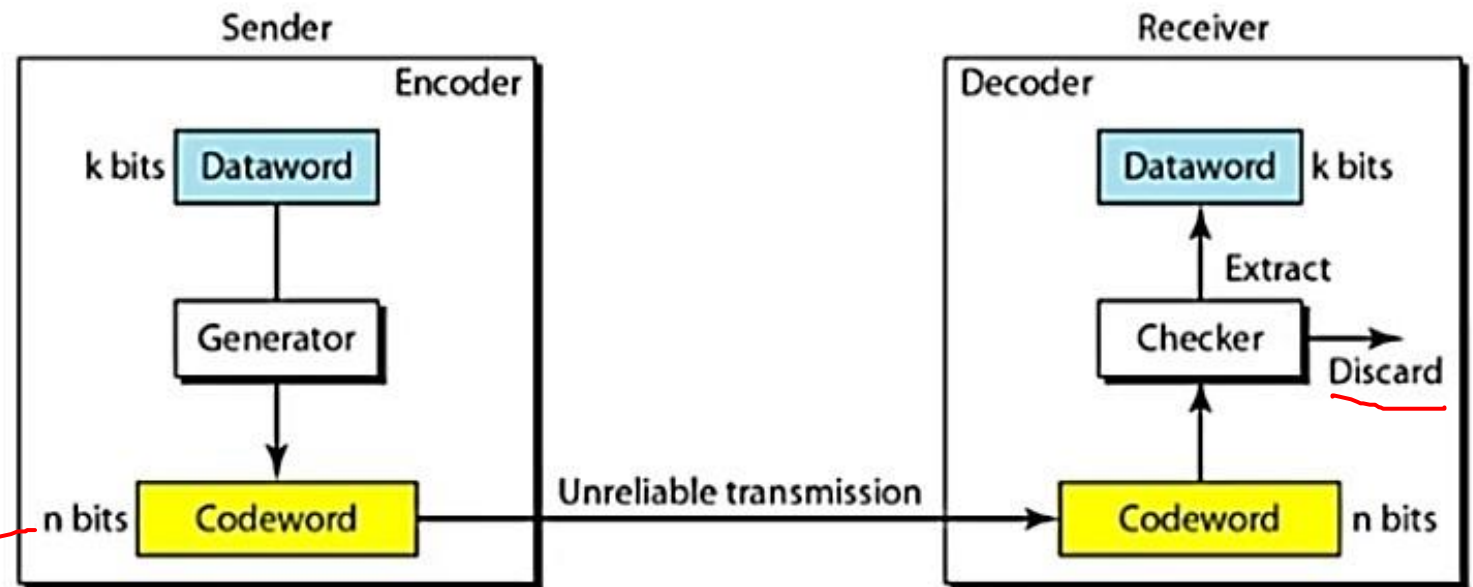
Variabel penting:
datawords, codeword, dan
 d_{min}



$3 \text{ bit}, 2 \text{ bit} = 2^k = 2^2 = 4$



Error Deteksi



Contoh: Misalkan $k = 2$ dan $n = 3$. Table 13.1 menampilkan daftar datawords dan codewords.

Misalkan pengirim mengkodekan **dataword 01** menjadi **011** dan mengirimkannya ke penerima.

Maka : $\rightarrow 000$

Datawords	Codewords
00	000
01	011
10	101
11	110

Tabel 13.1 Kode Untuk Deteksi Error
Sumber : (Forouzan, 2007)

→ 1. **Penerima menerima 011**. Codeword valid. Penerima mengekstrak dataword 01 dari codeword tersebut.

2. Codeword rusak pada saat transmisi, dan **111** diterima. Ini bukan codeword yang valid dan akan dibuang.

3. Codeword rusak pada saat transmisi, and **000** diterima. Ini adalah codeword yang valid. Perima mengekstrak dataword 00 (data salah). **2 Bit yang rusak** membuat error tidak bisa dideteksi.

① bit deteksi

$0011 \rightarrow 0001$

$0101 \rightarrow 010 \rightarrow 01$

$2B / 3B$

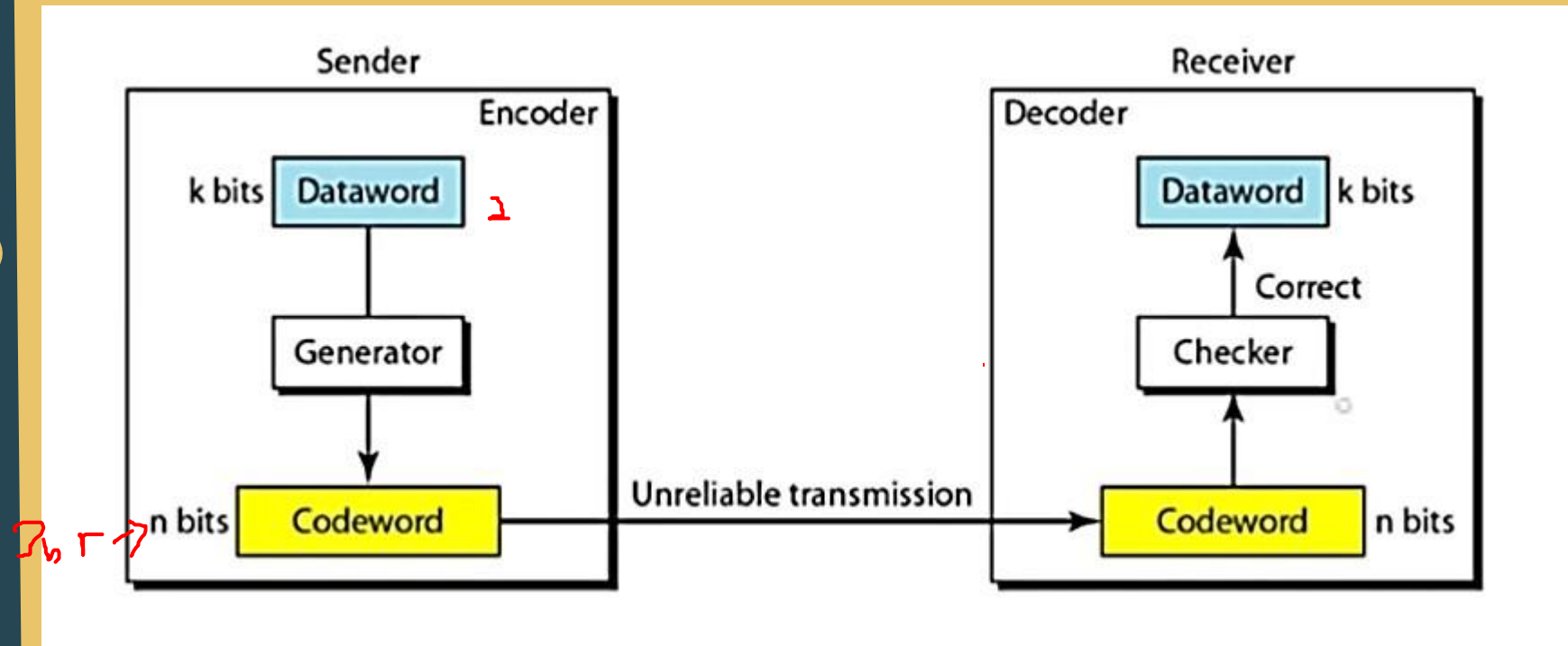
$\rightarrow 3B / 4B$

$000 = 0001$
 $010 = 0101$
 $101 = 1011$
 $110 = 1101$

Deteksion Error

Blok Kode hanya bisa mendeteksi jenis error yang sesuai peruntukannya

Error Correction



Contoh

Tambahkan lebih banyak bit redundan untuk untuk melihat apakah penerima mampu memperbaiki sebuah error tanpa tahu data sebenarnya yang dikirimkan. Ditambahkan 3 bit redundan ke 2-bit dataword dan menghasilkan 5-bit codewords. Table 13.2 menunjukkan datawords dan codewords.

Misalkan dataword adalah 01. Pengirim membuat codeword 01011. Codeword rusak pada saat transmisi sehingga diterima

→ 01001. ↔ 01011

Dataword	Codeword
00	00000
✓ 01	01011
10	10101
11	11110

Tabel 13.2 Kode Untuk koreksi Error
Sumber : (Forouzan, 2007)

Bagaimana menentukan dataword yang benar ?

Contoh

Penerima **tidak menemukan codeword** pada **tabel** maka terjadi error. Penerima mengasumsikan bahwa hanya **ada 1 bit yang error**. Strategi menentukan dataword yang benar :

1. **Bandingkan codeword** yang diterima dengan codeword pertama pada tabel (01001 - 00000). Penerima memutuskan bukan codeword pertama karena terdapat **2 bit yang berbeda**.
2. Dengan cara yang sama, codeword ketiga atau keempat juga bukan.
3. **Codeword kedua** adalah yang benar karena hanya ini yang **berbeda 1 bit**. Maka, penerima mengganti 01001 menjadi 01011 dan berdasarkan tabel mengekstrak **dataword 01**.

Hamming Distance

- **Hamming distance** antar dua data word = **jumlah bit yang berbeda.**

1. Hamming distance $d(000, 011)$ adalah 2 karena

$$000 \oplus 011 = 011 \text{ (2 bit 1)}$$

2. Hamming distance $d(10101, 11110)$ adalah 3 karena

$$10101 \oplus 11110 = 01011 \text{ (3 bit 1)}$$

Minimum Hamming Distance →

- Minimum Hamming Distance adalah jarak terkecil dari semua pasangan word data yang mungkin pada suatu kumpulan word.
- Menentukan MHD untuk Tabel 13.2

0101
= 3

$$\begin{array}{lll}
 d(00000, 01011) = 3 & d(00000, 10101) = 3 & d(00000, 11110) = 4 \\
 d(01011, 10101) = 4 & d(01011, 11110) = 3 & d(10101, 11110) = 3
 \end{array}$$

Maka, $d_{\min} = 3$

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

2b / 5b

① 2 bit Error
② 1 bit Error

00000
+ 11110

11110 ⇒ d = 4

- Untuk mendeteksi s bit error, maka MHD sebuah block code adalah $d_{\min} = s + 1$
- Untuk koreksi t bit error, maka MHD sebuah block code adalah $d_{\min} = 2t + 1$

$$\begin{aligned}
 d_{\min} &= s + 1 \\
 3 &= s + 1 \\
 s &= 2
 \end{aligned}$$

$$\begin{aligned}
 3 &= 2t + 1 \\
 2 &= 2t \rightarrow t = 1
 \end{aligned}$$

A decorative vertical band on the left side of the slide. It features a repeating geometric pattern of interlocking hexagons. The hexagons are colored in a dark navy blue and a burnt orange-red, separated by thin white lines. The entire pattern is set against a solid mustard yellow background that covers the rest of the slide.

Terimakasih