

*Kuliah Minggu VI*

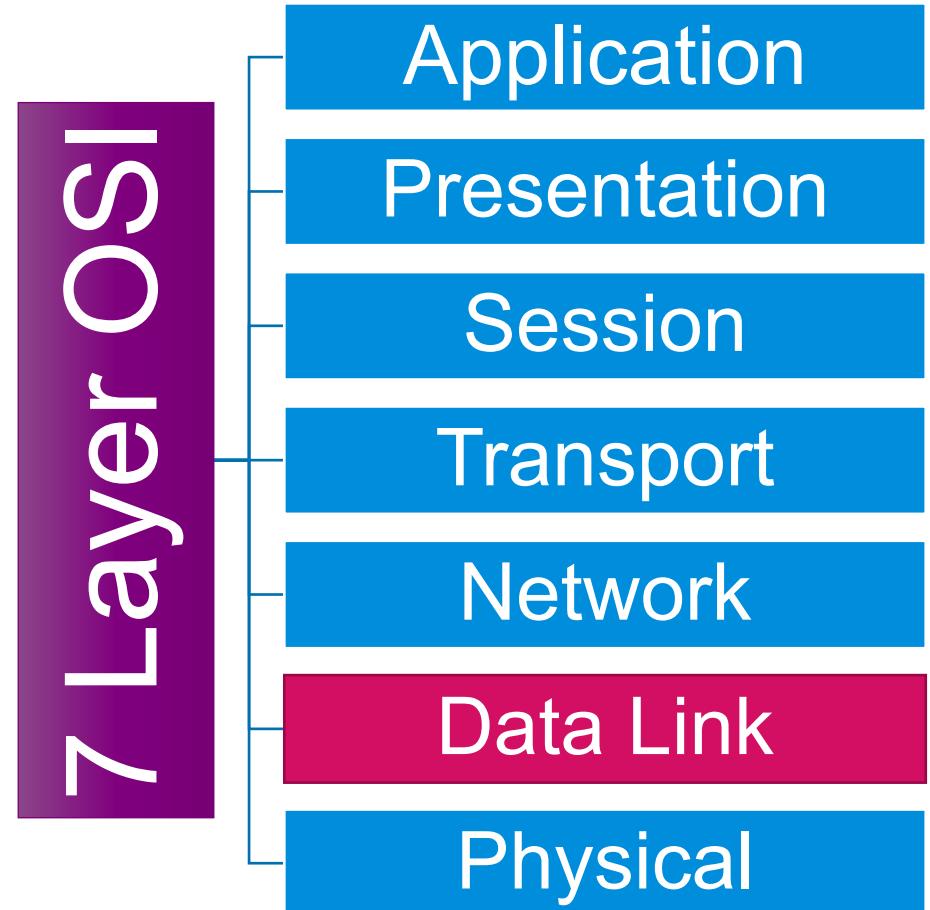
# DATA LINK LAYER: ERROR CONTROLLING



I Putu Arya Dharmaadi, ST, MT

*Program Studi Teknologi Informasi Fakultas Teknik  
Universitas Udayana*

# PENDAHULUAN



# PENANGANAN ERROR

---

- Media fisik sebagai saluran pengantar variasi sinyal/gelombang terkadang mengalami gangguan sehingga membuat bit-bit data yang dikirimkan menjadi berubah (error)

# KARAKTERISTIK MEDIA KOMUNIKASI

---

- Media transmisi kabel seperti fiber optik memiliki rasio error yang sangat kecil
- Sebaliknya, media transmisi wireless memiliki rasio error yang lebih tinggi
- Dibutuhkan model penanganan yang berbeda

# STRATEGI PENANGANAN ERROR

---

- Menambahkan informasi tambahan (bisa ditaruh pada header/trailer) pada frame yang dikirimkan
- Bagi penerima, informasi tersebut bisa menjadi dasar untuk menentukan apakah frame yang dikirimkan mengalami error

## 2 STRATEGI PENANGANAN ERROR

---

- ***Error-Correcting Codes*** → penerima bisa mendekripsi sekaligus memperbaiki kesalahan data yang terjadi
- ***Error-Detecting Codes*** → penerima hanya bisa mendekripsi kesalahan saja

# PENANGANAN ERROR TERGANTUNG PADA KARAKTERISTIK MEDIA

---

- ***Error-correcting codes*** → umumnya diterapkan pada media wireless yang sering terjadi error
- ***Error-detecting codes*** → umumnya diterapkan pada media kabel yang jarang error

# TEKNIK PENANGANAN ERROR SECARA UMUM

---

- Menambahkan bit-bit redundan pada data yang dikirimkan
- Sehingga, sebuah frame terdiri dari:
  - $m$  bit data (*message bits*)
  - $r$  bit data (*redundant/check bits*)
- $n$  bit codeword =  $m + r$

# CODE RATE

---

- Perbandingan antara bit pesan dengan keseluruhan bit  $\rightarrow m / n$
- Umumnya, pada media transmisi yang sering error, code rate bernilai  $1/2$
- Sebaliknya, pada media yang bagus, code rate mendekati  $1$

# ERROR-CORRECTING CODES

---

- Hamming codes.
- Binary convolutional codes.
- Reed-Solomon codes.
- Low-Density Parity Check codes.

# PENGIRIMAN HAMMING CODES

---

- Setiap bit pada codeword diberikan nomor urut, dari 1 sampai  $n$
- Bit yang berada pada nomor pangkat 2 (misalnya 1, 2, 4, 8, 16, dll) disebut check bit atau parity bit
- Bit-bit sisanya merupakan message bit

# PENGIRIMAN HAMMING CODES (2)

---

- Sebagai contoh, 8 bit data 11000100 akan dikirim
- Maka, bit-bit data tersebut diletakkan pada posisi yang menjadi message bit
- Hasilnya sebagai berikut:

1	2	3	4	5	6	7	8	9	10	11	12
P1	P2	1	P4	1	0	0	P8	0	1	0	0

# PENGIRIMAN HAMMING CODES (3)

- Kemudian, parity bit diisi dengan aturan sebagai berikut.

Nomor bit codeword	Parity Bit				
	1	2	4	8	16
1					
2					
3	x	x			
4					
5	x		x		
6		x	x		
7	x	x	x		

# PENGIRIMAN HAMMING CODES (4)

Nomor bit codeword	Parity Bit				
	1	2	4	8	16
8					
9	x			x	
10		x		x	
11	x	x		x	
12			x	x	
13	x		x	x	
14		x	x	x	

# CARA BACA TABEL PARITY BITS

---

- Bit codeword 3 akan dikoreksi oleh parity bit 1 dan 2
- Bit codeword 5 akan dikoreksi oleh parity bit 1 dan 4
- Bit codeword 6 akan dikoreksi oleh parity bit 2 dan 4
- Dan seterusnya ...

## CARA BACA TABEL PARITY BITS (2)

---

- Parity bit 1 akan mengoreksi bit 3, 5, 7, 9, 11
- Sehingga nilai parity bit 1 adalah:

$$P1 = \text{sum of bits } (3,5,7,9,11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$

Beginu pula parity bit yang lain . . .

## CARA BACA TABEL PARITY BITS (3)

---

$$P_2 = \text{sum of bits } (3, 6, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$P_4 = \text{sum of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$P_8 = \text{sum of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

- Jika nilai parity bit sudah didapatkan, maka tinggal dimasukkan ke data yang akan dikirim

# PENGIRIMAN HAMMING CODES (5)

---

- Hasil akhir data 8 bit beserta parity bit yang akan dikirim adalah sebagai berikut:

1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	1	1	0	0	1	0	1	0	0

# PENERIMA HAMMING CODES

---

- Ketika data diterima, penerima akan melakukan pengecekan terlebih dahulu
- Misalnya, data yang dikirimkan terjadi error pada bit ke-5, akan dicek apakah penerima bisa mendeteksi bahwa bit ke-5 terjadi error

1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	1	0	0	0	1	0	1	0	0

# PENERIMA HAMMING CODES (2)

- Proses pengecekan error, akan dibandingkan hasil perhitungan dengan nilai parity bit yang diterima

Parity Bit	Bit yang Dikoreksi oleh Parity Bit	Hasil Penjumlahan Bit yang Dikoreksi	Nilai Parity Bit	Status
P1	3,5,7,9,11	$1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1$	0	FALSE
P2	3,6,7,10,11	$1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$	0	TRUE
P4	5,6,7,12	$0 \oplus 0 \oplus 0 \oplus 0 = 0$	1	FALSE
P8	9,10,11,12	$0 \oplus 1 \oplus 0 \oplus 0 = 1$	1	TRUE

# PENERIMA HAMMING CODES (3)

---

- Berdasarkan perhitungan, ada 2 bit parity yang error (false), yaitu P1 dan P4, maka bit yang error adalah  $1 + 4 = 5$
- Jadi, untuk mengoreksi data, tinggal membalikkan (flip) nilai dari bit ke 5, yaitu dari nilai nol ke satu
- Sehingga data kembali benar

# ERROR-DETECTING CODES

---

- Parity.
- Checksums.
- Cyclic Redundancy Checks (CRCs).

# PARITY

---

- Melengkapi 1 bit dibelakang data untuk membuat data menjadi genap
- Contoh data awal yang akan dikirim: 1011010
  - Maka data yang dikirim menjadi: 10110100
- Contoh data awal yang akan dikirim: 1011110
  - Maka data yang dikirim menjadi: 10111101

# PENGECEKAN PARITY

---

- Apabila data yang diterima tidak genap, maka dipastikan data telah terjadi error
- Namun, pengirim tidak bisa mengetahui bit mana yang error karena metode ini hanyalah error-detecting, bukan error-correcting
- Jadi, solusinya adalah meminta pengirim mengirim ulang data

# CHECKSUM

---

- Menggunakan operasi perhitungan tertentu terhadap keseluruhan data, kemudian meletakkan hasilnya pada bagian belakang data
- Contoh paling umum: operasi penjumlahan
- Contoh data: 7, 11, 12, 0, 6
- Data yang dikirimkan menjadi: 7, 11, 12, 0, 6, 36

# PENGECEKAN CHECKSUM

---

- Apabila data bagian checksum (data yang paling belakang) tidak sama nilainya dengan perhitungannya, maka dipastikan telah terjadi error
- Misalnya data yang diterima: 7, 11, 12, 0, 6, 37
- $7 + 11 + 12 + 0 + 6 = 36 \rightarrow$  tidak cocok dengan checksumnya, maka data ini error

# CYCLIC REDUNDANCY CHECK

---

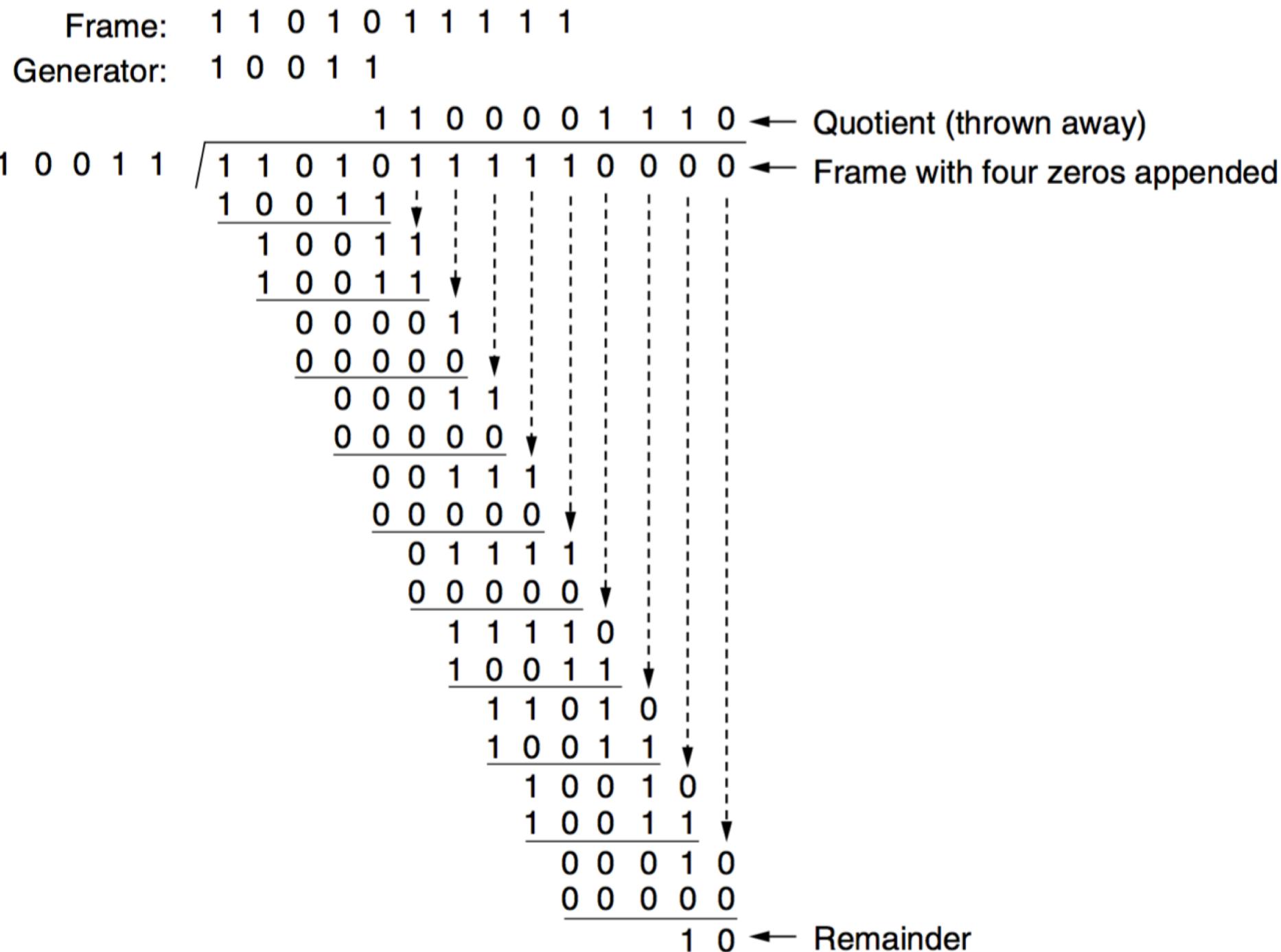
- Merupakan proses pembagian dimana pengirim dan penerima menetapkan dulu nilai pembaginya.
- Misalnya 10011 ditetapkan sebagai **nilai pembagi**
- Kemudian, data yang dikirim: 1101 0111 11

## CYCLIC REDUNDANCY CHECK (2)

---

- Data yang dikirim akan ditambahkan dengan 4 bit nol, karena nilai pembaginya menggunakan bilangan 5 bit
- Setelah proses pembagian bit, sisa hasil pembagian akan ditambahkan pada data yang sudah ditambah dengan 4 bit nol

# PROSES PEMBAGIAN



# CYCLIC REDUNDANCY CHECK (3)

---

- Jadi, data yang dikirim menjadi:  
1101 0111 1100 10
- Untuk pengecekan, maka data yang diterima akan dibagi dengan nilai pembagi yang sama
- Jika hasil pembagian tidak ada sisa, maka data dipastikan bebas error

**NEXT . . .**

## Network Layer



# REFERENSI

---

Tanenbaum,  
Wetherall. 2011.  
*Computer Networks*  
*5th Edition*. Prentice  
Hall

