

# **RANGKAIAN LOGIKA**

## **TE 2214**

---

---

**Materi Kuliah ke-2**

**SISTEM BILANGAN**

### **Sistem Bilangan**

---

1. **Bilangan Desimal**
2. **Bilangan Biner**
3. **Desimal → ke Biner**
4. **Aritmatika Biner**
5. **Komplemen 1 dan 2**
6. **Sign Bit**
7. **Operasi aritmatik dengan sign bilangan**
8. **Bilangan Hexadesimal**
9. **Bilangan Oktal**
10. **Binary Code Decimal (BCD)**
11. **Digital Code & Parity**

## Aritmatika Biner

---

1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian

## Penjumlahan bilangan biner

---

$$0 + 0 = 0 \quad \text{Hasil } 0 \text{ Simpanan } 0$$

$$0 + 1 = 1 \quad \text{Hasil } 1 \text{ Simpanan } 0$$

$$1 + 0 = 1 \quad \text{Hasil } 1 \text{ Simpanan } 0$$

$$1 + 1 = 10 \quad \text{Hasil } 0 \text{ Simpanan } 1$$

$$\begin{array}{r} 100 \\ + 10 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 110 \\ + 11 \\ \hline 1001 \end{array}$$

## Pengurangan bilangan biner

---

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

$$10 - 1 = 1 \quad 0 - 1 \text{ dengan pinjaman } 1$$

1 0 0
1 0
-----
1 0

1 1 0
1 1
----- +
0 1 1

## Perkalian bilangan biner

---

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

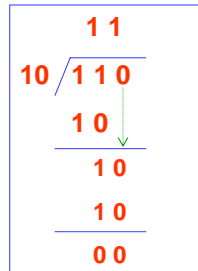
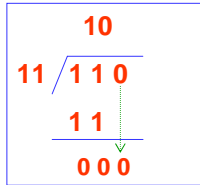
1 0 0
1 0
----- x
0 0 0 1 0 0
----- +
1 0 0 0

1 1 0
1 1
----- x
1 1 0
1 1 0
----- +
1 0 0 1 0

## Pembagian bilangan biner

---

Caranya hampir sama dengan bilangan desimal



## Complement 1 dan complement 2

---

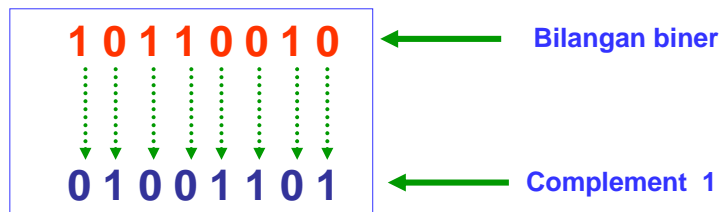
komplemen 1 dan komplemen 2 dalam bilangan biner merupakan hal yang penting untuk membuat bilangan negatif. Ada dua metode dalam membuat bilangan negatif yaitu :

a. Dengan Komplemen 1

b. Dengan Komplemen 2

**Dengan komplemen 1**

Yaitu dengan merubah setiap bit biner  $0 \rightarrow 1$  atau dari  $1 \rightarrow 0$



## Complimen 2

---

**Complement 2 = Complement 1 + 1**

1 0 1 1 0 0 1 0	←	Bilangan biner
0 1 0 0 1 1 0 1	←	Complement 1
+ 1	←	Tambah 1
0 1 0 0 1 1 1 0	←	Complement 2

## Signed Numbers

---

Sistem digital harus mampu menangani kedua bilangan positif dan bilangan negatif. Sign bilangan biner ditentukan oleh sign dan magnitude

Sign menentukan tanda positif dan negatif sedangkan magnitude menentukan nilai dari bilangan. Ada tiga bentuk sign integer yang dapat direpresentasikan :

1. Sign-magnitude
2. Complement 1
3. Complement 2

Yang paling penting adalah complement 2 sedangkan Sign-Magnitude yang paling sering digunakan

Yang bukan integer dan angka yang sangat besar atau bilangan yang kecil diexpresikan dengan Floating-point format.

## The Sign Bit

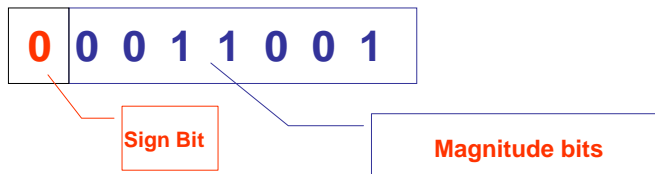
---

Sign Bit ditentukan oleh bit yang paling kiri, yang mana nilainya 0 berarti positif dan 1 adalah bilangan negatif

## Sign-Magnitude form

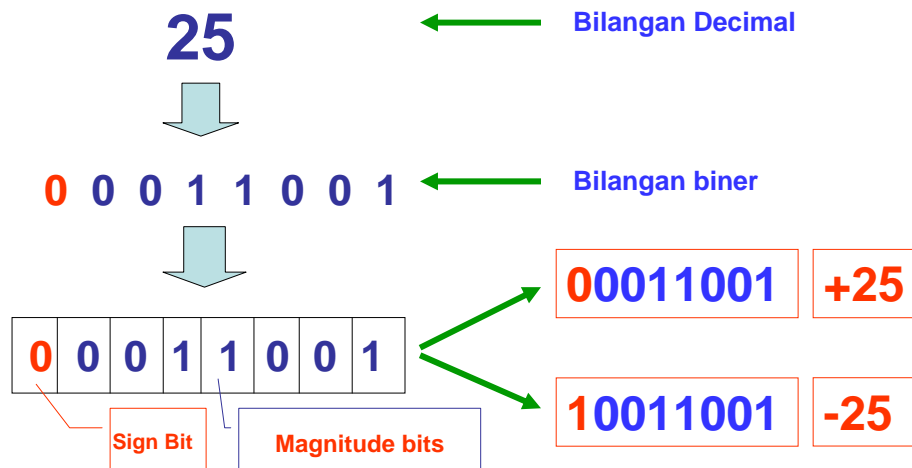
---

Magnitude merupakan nilai dari angka biner yang direpresentasikan dalam 8-bit

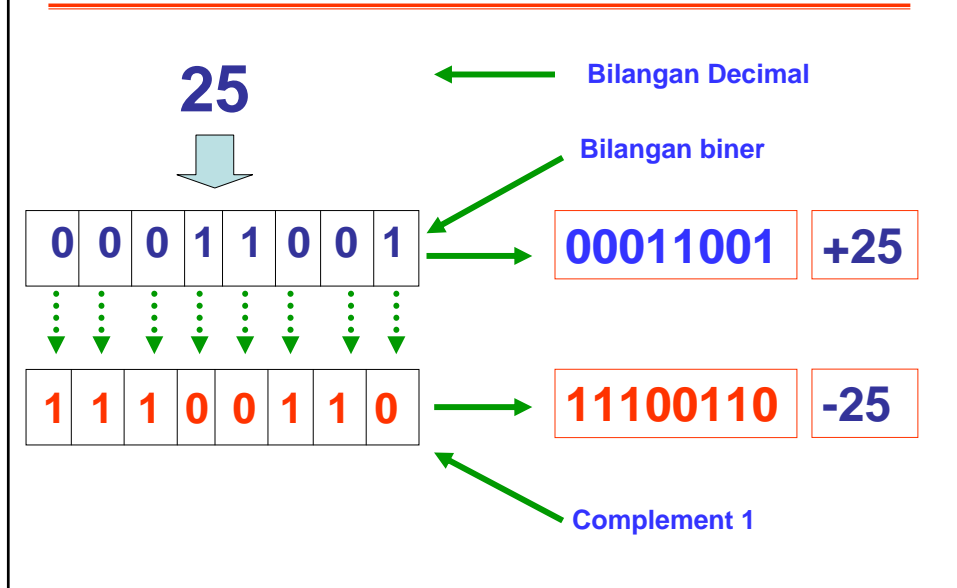


## Sign-Magnitude

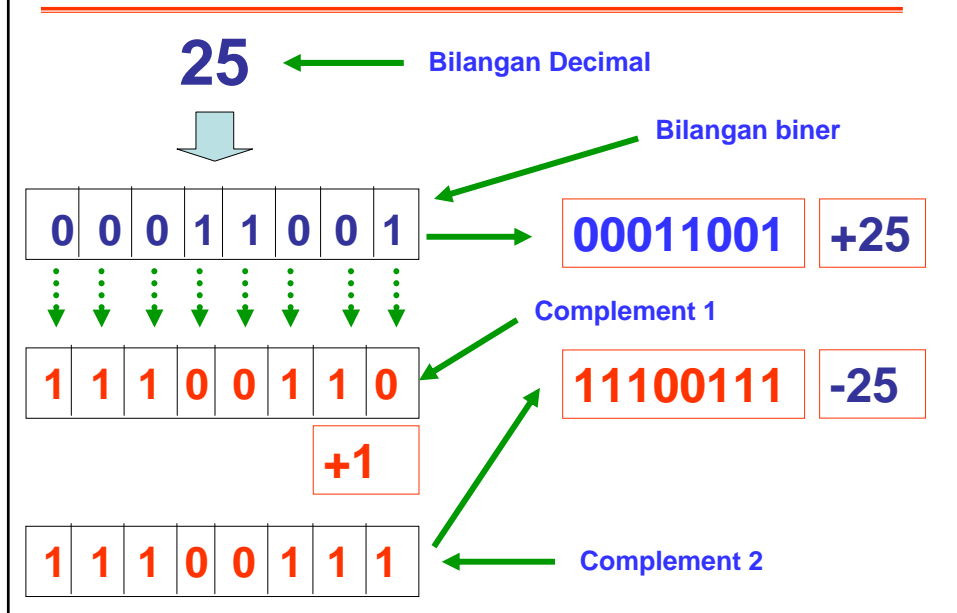
---



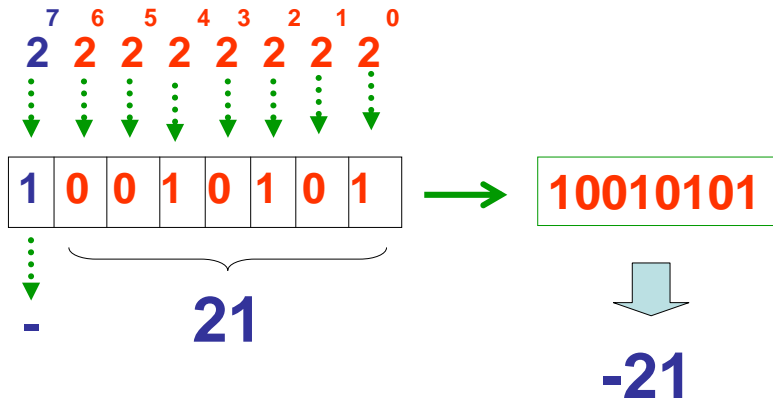
## Complement 1



## Complement 2



## The Decimal Value of Signed Number



## Range of sign Integer Number

8 bit number sebagai ilustrasi dikarenakan 8 bit secara paling umum dalam computer dinamakan **BYTE**. Maka 1 byte dapat direpresentasikan dalam **256** angka yang berbeda, 16 bit didapat **65536** angka yang berbeda dan 32 bit kita nyatakan dengan  **$4295 \times 10^9$**  jumlah angka yang berbeda.

**Formula dari kombinasi n bits maka total kombinasi adalah  $2^n$  unutup complement 2 sign number maka range dari nilai kombinasi n bits adalah :**

**-  $(2^{n-1})$  sampai dengan  $+ (2^{n-1}-1)$**

## Floating-Point Number

---

A Floating point number (bilangan real) terdiri dari dua bagian yaitu bagian **Mantissa** yang merupakan floating point bilangan yang menjelaskan mengenai **magnitude bilangan** dan bagian **exsponent** yang merupakan bagian floating point bilangan yang menjelaskan **angka tempat** dari point desimal / biner yang dipindahkan.

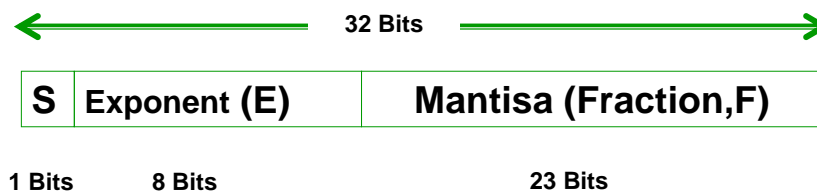
Contoh :

241,506,800 → maka mantisanya adalah **0,2415068** dan exponenya adalah **9** maka floating point bilangan tersebut  **$0,2415068 \times 10^9$**

## Single-precision Floating Point Binari Number

---

Single precision floating point binary number dengan standard format dimana Sign bit (S) yang merupakan bit paling kiri dan exponent (E) adalah 8 bit berikutnya dan bagian mantisa (F) dalam 23 bit berikutnya



### contoh

1011010010001

← Bilangan biner +

$$= 1,011010010001 \times 10^{12}$$

← 32 Bits →

0 10001011 011010010001000000000000

12 00001100

127 01111111

129 10001011

01000101101101001000100000000000

### Single-precision Floating Point Binari Number

$$\text{Number} = (-1)^S (1 + F) (2^{E-127}) \leftarrow \text{Formula}$$

1 10010001 100011100010000000000000

S	E	F
1 Bits	8 Bits	23 Bits

$$E = 10010001 = 145$$

$$\text{Number} = (-1)^1 (1.10001110001) (2^{145-127})$$

$$\text{Number} = (-1)^1 (1.10001110001) (2^{18})$$

$$\text{Number} = -1100011100010000000$$

## TUGAS II

---

Rubah bilangan  
desimal  **$3.248 \times 10^4$**   
kedalam *single  
precision floating-  
point binary*