

# Teknologi Informasi



# Bab 10

## Representasi Data



## Representasi Data

- Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan.
- Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep.



## Representasi Data (Cont.)

- Dalam **sistem komputer**, **data** yang diolah, diproses dan disimpan menggunakan bilangan biner (*binary*).
- Bilangan biner (bit) adalah unit data terkecil dalam komputasi, diwakili oleh digit 0 dan 1.
- Komputer hanya mengenal data berupa bit, digit 0 dan 1.

01100  
10110  
11110

## Pertanyaan

**Kenapa bilangan **biner** dipilih untuk mekanisme representasi data komputer ?**



## Jawaban

- Komputer secara elektronika hanya mampu membaca dua kondisi sinyal
  1. Ada sinyal (ada tegangan / ada arus listrik)
  2. Tidak ada sinyal (tidak ada tegangan / tidak ada arus listrik)
- Dua kondisi tersebut yang digunakan untuk merepresentasikan bilangan dan kode-kode biner
  - Level tinggi (ada tegangan) sebagai representasi bilangan 1
  - Level rendah (tidak ada arus) sebagai representasi bilangan 0

# Pertanyaan

**Kenapa bisa begitu?**



## Jawab

- Sirkuit dalam prosesor komputer terdiri dari miliaran transistor.
- Transistor adalah saklar kecil yang diaktifkan oleh sinyal elektronik yang diterimanya.
- Angka 1 dan 0 yang digunakan dalam biner mencerminkan keadaan on dan off dari sebuah transistor.





## Jawab (Cont.)

- Program komputer adalah serangkaian instruksi yang setiap instruksi diterjemahkan ke dalam kode mesin - kode biner sederhana yang mengaktifkan CPU.
- *Programmer* menulis kode pada komputer dan kode ini dikonversi oleh penerjemah menjadi instruksi biner yang dapat dijalankan oleh prosesor.
- Semua perangkat lunak, musik, dokumen, dan informasi lain apa pun yang diproses oleh komputer, juga disimpan menggunakan biner.



## Sistem Bilangan

- System bilangan (number system) adalah suatu cara untuk mewakili besaran dari suatu item fisik.
- Sistem bilangan yang banyak dipergunakan oleh manusia adalah system bilangan desimal, yaitu sistem bilangan yang menggunakan 10 macam symbol untuk mewakili suatu besaran.
- Sistem ini banyak digunakan karena manusia mempunyai sepuluh jari untuk dapat membantu perhitungan.



## Sistem Bilangan (cont.)

- Lain halnya dengan komputer, logika di komputer diwakili oleh dua keadaan, yaitu **on** (ada arus) dan **off** (tidak ada arus).
- Konsep inilah yang dipakai dalam sistem bilangan binary yang mempunyai dua macam nilai untuk mewakili suatu besaran nilai, yaitu **1 (on)** dan **0 (off)**
- Selain system bilangan biner, komputer juga menggunakan system bilangan octal dan hexadesimal.



## Sistem Bilangan – DECIMAL (DEC)

- Merupakan Radiks-10, Basis bilangan ini terdiri dari rentang angka 0 s.d 9. yaitu 0,1,2,3,4,5,6,7,8,9.
- Bilangan yang digunakan manusia secara luas dalam mengukur suatu besaran nilai.
- Contoh:
  - $(23)_{10} = 23_{10} = \text{DEC}23$
  - $(907)_{10} = 907_{10} = \text{DEC}907$



## Sistem Bilangan – BINARY (BIN)

- Digunakan dalam komunikasi data dan merupakan operasi dasar sebuah komputer
- Merupakan Radiks-2, basis bilangan ini terdiri dari angka 0 dan 1 dan tiap satu variable mewakili 1 bit
- Representai bilangan biner n-digit  $\rightarrow \mathbf{B_n = b_{n-1} \dots b_1 b_0}$
- Contoh:
  - $(101)_2 = 101_2 \quad \rightarrow \text{panjang data} = 3 \text{ bit}$
  - $(0010)_2 = 0010_2 \quad \rightarrow \text{panjang data} = 4 \text{ bit}$



## Sistem Bilangan – OCTADECIMAL (OCT)

- Biasanya digunakan dalam pengalamatan di memori utama komputer.
- Merupakan Radiks-8, basis bilangan ini terdiri dari rentang angka 0 s.d 7
- Contoh:
  - $(23)_8 = 16_{10}$
  - $(10)_8 = 8_{10}$



## Sistem Bilangan – HEXADECIMAL (HEX)

- Biasanya digunakan dalam pengalamatan di memori dan urusan pengkodean warna.
- Basis bilangan ini terdiri dari 16 deret (Radiks-16) yang terdiri dari angka **0-9** dan huruf **a-f**, dengan urutan dari nilai terkecil sampai terbesar adalah 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
- Contoh:
  - $(a2)_{16} = A2_{16} = 0xA2 = 0xa2$

## Tabel Bilangan

Desimal	Biner (8 bit)	Oktal	Heksadesimal
0	0000 0000	000	00
1	0000 0001	001	01
2	0000 0010	002	02
3	0000 0011	003	03
4	0000 0100	004	04
5	0000 0101	005	05
6	0000 0110	006	06
7	0000 0111	007	07
8	0000 1000	010	08
9	0000 1001	011	09
10	0000 1010	012	0A
11	0000 1011	013	0B
12	0000 1100	014	0C
13	0000 1101	015	0D
14	0000 1110	016	0E
15	0000 1111	017	0F
16	0001 0000	020	10





## Tipe Data

- Disetiap bahasa pemrograman, disediakan berbagai jenis tipe data.
- Penentuan tipe data yang tepat (sesuai dengan karakteristik data yang akan diolah) akan menjadikan sebuah program dapat dieksekusi secara efektif.



## Tipe Data (cont.)

- **Data Numerik:** mempresentasikan Integer dan pecahan Fixed-point, real floating-point dan decimal berkode biner.
- **Data Logikal:** Digunakan oleh operasi logika dan untuk menentukan atau memeriksa kondisi seperti yang dibutuhkan untuk instruksi kondisi bercabang.
- **Data Bit-Tunggal:** Untuk Operasi SHIFT, CLEAR, dan TEST.
- **Data Alfnumerik:** data yang tidak hanya dikodekan dengan bilangan tetapi juga dengan huruf dari alpabet dan karakter khusus lainnya



## Jenis Tipe Data

Secara garis besar terdapat 2 jenis tipe data

- Tipe Data *Primitive*

Tipe data yang mampu menyimpan satu nilai tiap satu variable dan merupakan tipe data dasar yang sering dipakai oleh program

- Tipe Data *Non-primitive*

tipe data yang secara default tidak terdefinisi oleh suatu bahasa pemrograman dan didefinisikan oleh programmer itu sendiri



## Tipe Data *Primitive*

Adalah tipe data yang mampu menyimpan satu nilai tiap satu variable. Tipe Data Primitive terdiri dari 2 jenis, yaitu

### ■ Numeric

Tipe data numeric digunakan pada variabel atau konstanta untuk menyimpan nilai dalam bentuk angka

### ■ Non-Numeric

Tipe data yang digunakan pada variabel atau konstanta untuk menyimpan nilai dalam bentuk selain angka

## Tipe Data *Primitive* - Numeric

- **Integer**, tipe data numeric yang merepresentasikan bilangan bulat

Tipe Data	Ukuran	Rentang Nilai
Byte	1 Byte	0 ~ 255
Short Int	2 Byte	-127 ~ 127
Integer	4 Byte	-32768 ~ 32768
Unsigned Integer	4 Byte	0 ~ 65535
Long Integer	8 Byte	-2147483648 ~ 2147483648

- **Floating Point**, tipe data numeric yang merepresentasikan bilangan pecahan

Tipe Data	Ukuran	Rentang Nilai
Float (5 point)	4 Byte	$3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
Double (15 point)	8 Byte	$1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$

## Tipe Data *Primitive* – *Non-Numeric*

- ***Char***, tipe data yang dapat menyimpan suatu karakter. Karakter dapat terdiri dari suatu angka, huruf, tanda baca atau bahkan karakter khusus. Dibutuhkan 1 byte (8 bit) ruang di dalam memori agar dapat menyimpan sebuah karakter.
- ***Boolean***, merupakan tipe data logika, yang berisi dua kemungkinan nilai: TRUE (benar, direpresentasikan dengan nilai 1) atau FALSE (salah, direpresentasikan dengan nilai 0).



## Tipe Data *Non-Primitive*

Adalah tipe data yang mampu menyimpan banyak nilai pada tiap satu variable (**composite**) dan biasanya didefinisikan sendiri oleh pemrogram. Tipe Data *Non-Primitive* terdiri dari beberapa jenis, yaitu

1. **String**, tipe data yang dapat menampung banyak karakter (angka, huruf, symbol) dan penulisan karakter string dimulai dan diakhiri dengan symbol *double-quote* ("""). Contoh **"ini string"**



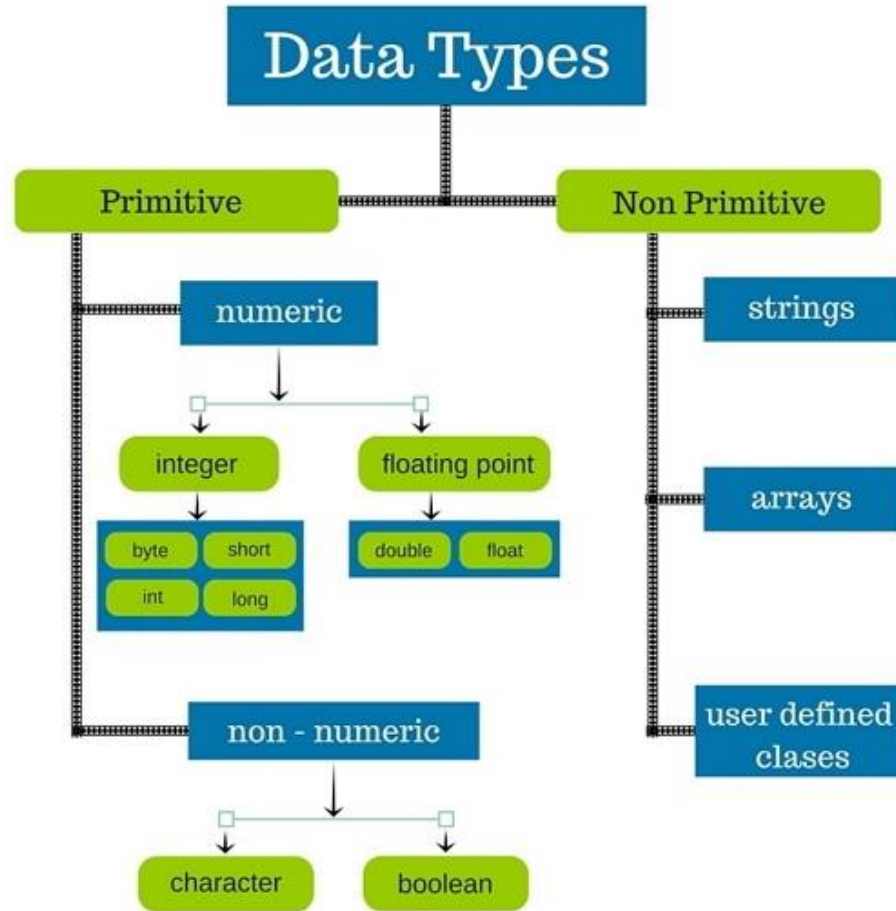
## Tipe Data *Non-Primitive* (cont.)

2. **Array**, tipe data yang dapat menampung banyak nilai. Array ibarat seperti loker, dimana tiap elemen (pintu) dapat menyimpan nilai. Array dapat bersifat homogen (menyimpan nilai dengan tipe data yang sama) ataupun bersifat heterogen, tergantung Bahasa pemrograman. Contoh Array = ["saya", "cinta", "Polinema"]
3. **User define class**, tipe data yang nilainya merupakan hasil dari definisi class yang dilakukan user (pemrogram). Contoh Object class





# Jenis Tipe Data





## Konversi Bilangan

- Konversi bilangan adalah suatu proses dimana satu system bilangan dengan basis tertentu akan dijadikan bilangan dengan basis yang lain.



## Konversi Bilangan - Radiks-r ke Desimal

### Konversi Radiks-r ke Desimal

#### Contoh

$$D_r = \sum_{i=-n}^{n-1} d_i \times r^i$$

- $1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$   
 $= 8 + 4 + 0 + 1 = 13_{10}$
- $572_8 = (5 \times 8^2) + (7 \times 8^1) + (2 \times 8^0)$   
 $= 320 + 56 + 2 = 378_{10}$
- $1CD_{16} = (1 \times 16^2) + (12 \times 16^1) + (13 \times 16^0)$   
 $= 256 + 192 + 13 = 461_{10}$



## Konversi Bilangan – Desimal ke Radiks-r

### Konversi Desimal ke Radiks-r

1. Bagi nilai decimal dengan radiks (jumlah symbol) dan ambil nilai bilangan bulat saja.
2. Bilangan bulat sisa pembagian merupakan kumpulan hasil konversi, dimana sisa yang pertama menjadi **least significant bit (LSB)** dan sisa terakhir menjadi **most significant bit (MSB)**
3. Lakukan proses ke-1 sampai hasil pembagian menjadi 0.

## Konversi Bilangan – Desimal ke Biner

- Dilakukan dengan membagi **2** bilangan decimal secara berulang-ulang sampai nilainya 0. Sisa setiap pembagian merupakan bit yang didapat, dimana sisa yang pertama akan menjadi **least significant bit (LSB)** dan sisa yang terakhir menjadi **most significant bit (MSB)**
- Contoh :  $76_{10} = \dots_2$ ?
  - $76/2 = 38 \rightarrow$  sisa **0**
  - $38/2 = 19 \rightarrow$  sisa **0**
  - $19/2 = 9 \rightarrow$  sisa **1**
  - $9/2 = 4 \rightarrow$  sisa **1**
  - $4/2 = 2 \rightarrow$  sisa **0**
  - $2/2 = 1 \rightarrow$  sisa **0**
  - $1/2 = 0 \rightarrow$  sisa **1**
- Hasil  $76_{10} = \mathbf{1001100}_2$

## Konversi Bilangan – Desimal ke Oktal

- Dilakukan dengan membagi **8** bilangan decimal secara berulang-ulang sampai nilainya 0. Sisa setiap pembagian merupakan bit yang didapat, dimana sisa yang pertama akan menjadi **least significant bit (LSB)** dan sisa yang terakhir menjadi **most significant bit (MSB)**

Contoh :  $76_{10} = \dots_8 ?$

■	$76/8$	$= 9$	$\rightarrow$	sisa <b>4</b>
	$9/8$	$= 1$	$\rightarrow$	sisa <b>1</b>
	$1/8$	$= 0$	$\rightarrow$	sisa <b>1</b>

■ Hasil  $76_{10} = \mathbf{114}_8$

## Konversi Bilangan – Desimal ke Heksa

- Dilakukan dengan membagi **16** bilangan decimal secara berulang-ulang sampai nilainya 0. Sisa setiap pembagian merupakan bit yang didapat, dimana sisa yang pertama akan menjadi **least significant bit (LSB)** dan sisa yang terakhir menjadi **most significant bit (MSB)**
- Contoh :  $76_{10} = \dots_{16} ?$ 
  - $76/16 = 4 \rightarrow$  sisa **12**  
 $4/16 = 0 \rightarrow$  sisa **4**
  - Hasil  $76_{10} = \mathbf{4C}_{16}$



## Konversi Bilangan - Biner ke Oktal

1. Bilangan Oktal adl Radiks-7, untuk dapat menjangkau nilai maksimal Oktal (7), bilangan biner butuh 3 bit (3 digit biner)
2. Untuk mengkonversi biner ke octal, bagi panjang bit biner untuk masing-masing bagian adalah 3 digit, dari kanan (LSB) ke kiri (MSB).
3. Untuk digit MSB (paling kiri), apabila panjang digit kurang dari 3, tambahkan 0 di posisi paling kiri.

Contoh 10110110  $\rightarrow$  10 110 110  $\rightarrow$  010 110 110





## Konversi Bilangan - Biner ke Oktal (cont.)

4. Hitung nilai untuk masing-masing bagian bit

**010** **110** **110**



**2**

**6**

**6**

Hasil  $10110110_2 = 266_8$



## Konversi Bilangan – Oktal ke Biner

1. Merupakan kebalikan dari konversi Biner ke Oktal
2. Satu bilangan octal akan diterjemahkan ke dalam 3 digit bilangan biner.
3. Sehingga untuk konversi n-digit bilangan octal, memerlukan 3n-digit bilangan biner.

Contoh:  $164_8 = \dots_2$  ?

1	6	4
↓	↓	↓
001	110	100

$164_8 = \text{001}110\text{100}_2$  (nilai 0 di depan bisa di abaikan)



## Konversi Bilangan - Biner ke Heksa

1. Bilangan Heksa adl Radiks-16, untuk dapat menjangkau nilai maksimal Heksa(16), bilangan biner butuh 4 bit (4 digit biner)
2. Untuk mengkonversi biner ke heksa, bagi panjang bit biner untuk masing-masing bagian adalah 4 digit, dari kanan (LSB) ke kiri (MSB).
3. Untuk digit MSB (paling kiri), apabila panjang digit kurang dari 4, tambahkan 0 di posisi paling kiri.

Contoh  $1011000110 \rightarrow 10\ 1100\ 0110 \rightarrow 0010\ 1100\ 0110$



## Konversi Bilangan - Biner ke Heksa (cont.)

4. Hitung nilai untuk masing-masing bagian bit

0010 1100 0110



2



C



6

Hasil  $1011000110_2 = 2C6_{16}$



## Konversi Bilangan – Heksa ke Biner

1. Merupakan kebalikan dari konversi Biner ke Heksa
2. Satu bilangan heksa akan diterjemahkan ke dalam 4 digit bilangan biner.
3. Sehingga untuk konversi n-digit bilangan heksa, memerlukan 4n-digit bilangan biner.

Contoh:  $1CD_{16} = \dots_2$  ?

1  
↓

C  
↓

D  
↓

0001

1100

1101

$1CD_{16} =$ ~~0001~~ $11001001_2$  (nilai 0 di depan bisa di abaikan)



**Pertanyaan?**