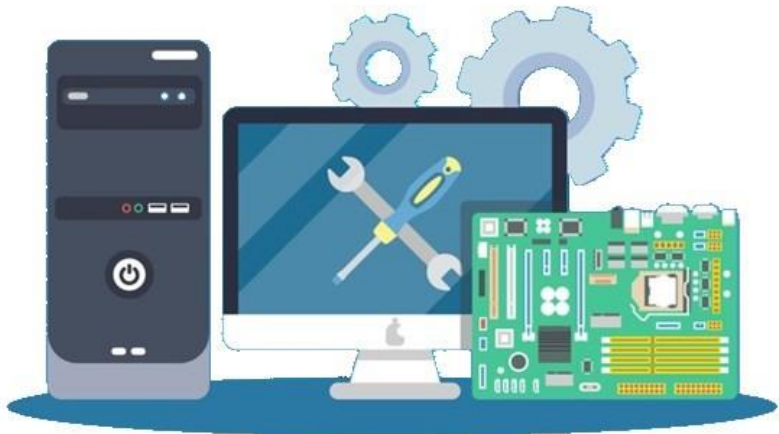




**PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

MATA KULIAH
**ORGANISASI DAN
ARSITEKTUR KOMPUTER**



COMPUTER ARITHMETIC

TIM Organisasi dan Arsitektur Komputer
2020

Capaian Pembelajaran

- Mahasiswa mampu memberikan contoh system bilangan decimal, biner, heksadesimal dan menyelesaikan konversi antara bilangan decimal, biner atau heksadesimal

Latar Belakang

- Operasi aritmatika dilakukan oleh **Arithmetic Logic and Unit (ALU)**
- ALU adalah bagian dari computer yang melakukan kerja perhitungan dan logika – logika operasi pada data
- Semua komponen pada computer didasarkan pada penggunaan sirkuit dengan logika digital pada perangkat yang akan direpresentasikan oleh **binary digit (bit)** dan simple operasi logika **boolean**

Content

Number System

- Decimal
- Binary
- Octal
- Hexadecimal

Conversion

- Decimal to Binary
- Binary to Hexadecimal
- Hexadecimal to Binary

Number System

Decimal

Binary

Octal

Hexadecimal

Pengantar – Sistem Bilangan Desimal

- Dalam kehidupan sehari – hari, kita biasa menggunakan bilangan dengan digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Bilangan tersebut disebut dengan desimal
- Desimal adalah sistem bilangan yang memiliki basis atau radix 10
- Artinya, tiap digit pada bilangan tersebut akan dikalikan dengan 10^n dimana n adalah pangkat yang disesuaikan dengan posisi digit bilangan
- Contoh:

$$83_{(10)} \quad (8 \times 10^1) + (3 \times 10^0)$$

$$4758_{(10)} \quad (4 \times 10^3) + (7 \times 10^2) + (5 \times 10^1) + (8 \times 10^0)$$

Biner

Jika pada system bilangan **decimal** memiliki 10 digit yaitu 0 sampai 9, maka pada sistem bilangan **biner** hanya memiliki 2 digit yaitu 1 dan 0

Tiap digit pada system bilangan biner memiliki nilai berdasarkan posisinya

Contoh:

| | |
|--------------|---|
| $10_{(2)}$ | $(1 \times 2^1) + (0 \times 2^0)$ |
| $11_{(2)}$ | $(1 \times 2^1) + (1 \times 2^0)$ |
| $1010_{(2)}$ | $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$ |

Oktal

Jika pada system bilangan **decimal** memiliki 10 digit yaitu 0 sampai 9, maka pada sistem bilangan **oktal** memiliki 8 digit yaitu 0 hingga 7

Tiap digit pada system bilangan oktal memiliki nilai berdasarkan posisinya

Contoh:

| | |
|----------------------------|---|
| 10 ₍₈₎ | (1 x 8 ¹) + (0 x 8 ⁰) |
| 132 ₍₈₎ | (1 x 8 ²) + (3 x 8 ¹) + (2 x 8 ⁰) |
| 1024 ₍₈₎ | (1 x 8 ³) + (0 x 8 ²) + (2 x 8 ¹) + (4 x 8 ⁰) |

Hexadecimal

- Teks dan bilangan dapat dikodekan dalam komputer dengan pola dari bilangan biner
- Hexadecimal merupakan *shortcut* untuk merepresentasikan bilangan biner
- Hexadecimal digunakan untuk menampilkan nilai dari alamat memory dalam pemrograman komputer

Mengapa Menggunakan Hexadecimal?

- Hexadecimal lebih kompak dibandingkan notasi biner
 - Hexadecimal adalah base 16, direpresentasikan dengan 4 digit
- Secara keseluruhan, komputer menggunakan kelipatan dari 4 digit biner atau lebih. Karenanya, beberapa 4 kelipatan tersebut adalah satu digit hexadecimal
 - contoh: $1111_{(2)}$ sama dengan $F_{(16)}$,
4 digit pada bilangan biner dikonversi menjadi 1 digit pada hexadecimal
- Sangat mudah untuk melakukan konversi antara biner dan heksadesimal

Representasi Hexadecimal

- **Hexadecimal** adalah sistem bilangan 16 simbol, yaitu **0-9** dan **A-F** dapat direpresentasikan dalam bentuk 4 digit biner

| | |
|----------|----------|
| 0000 = 0 | 1000 = 8 |
| 0001 = 1 | 1001 = 9 |
| 0010 = 2 | 1010 = A |
| 0011 = 3 | 1011 = B |
| 0100 = 4 | 1100 = C |
| 0101 = 5 | 1101 = D |
| 0110 = 6 | 1110 = E |
| 0111 = 7 | 1111 = F |

Contoh – Representasi Hexadecimal

$$\begin{aligned}
 2C_{(16)} &= (2 \times 16^1) + (C \times 16^0) \\
 &= (2 \times 16^1) + (12 \times 16^0) \\
 &= 32 + 12 = 44_{(10)}
 \end{aligned}$$

$$\begin{aligned}
 2C_{(16)} &= 2 \quad C \\
 &= 0010 \quad 1100 \\
 &= 00101100_{(2)}
 \end{aligned}$$

- Hexadecimal tidak hanya digunakan untuk representasi bilangan integer, namun juga dapat digunakan untuk merepresentasikan deret bilangan biner baik representasi dalam teks, angka atau tipe data lainnya
- Contoh: ASCII

Konversi – Desimal ke Biner

- Konversi dari base 10 ke base 2 dengan melakukan pembagian secara berulang
- Hasil pembagian terdiri dari **Quotient** (Hasil Bagi) dan **Remainder** (Sisa Hasil Bagi)
- Proses pembagian seperti model **modulus**

| Pembagian | Quotient | Remainder |
|----------------|----------|-----------|
| $\frac{11}{2}$ | 5 | 1 |
| $\frac{5}{2}$ | 2 | 1 |
| $\frac{2}{2}$ | 1 | 0 |
| $\frac{1}{2}$ | 0 | 1 |
| Hasil | | 1101 |

Hasil Konversi dibaca dari
bawah ke **atas**

Latihan

- Konversi bilangan desimal berikut ke biner:

- $64_{(10)} = \dots\dots (2)$

- $100_{(10)} = \dots\dots (2)$

- $128_{(10)} = \dots\dots (2)$

- $255_{(10)} = \dots\dots (2)$

Konversi – Biner Ke Desimal

- Tiap digit biner dikalikan dengan 2^n dimana n adalah bilangan pangkat yang disesuaikan posisinya.
- Proses perkalian menggunakan **Least Significant Bit (LSB)**, yaitu komputasi dimulai dari barisan atau deret yang paling kecil atau paling kanan atau paling belakang
- Contoh : $101_{(2)} = \dots_{(10)}$

| | | |
|-------|-------|-------|
| 1 | 0 | 1 |
| 2^2 | 2^1 | 2^0 |
| 4 | 0 | 1 |



$$101_2 = (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$= 4 + 0 + 1 = 5$$

Latihan

- Konversikan bilangan biner berikut ke decimal:
 - $1100_{(2)} = \dots\dots_{(10)}$
 - $0111_{(2)} = \dots\dots_{(10)}$
 - $11110_{(2)} = \dots\dots_{(10)}$
 - $101010_{(2)} = \dots\dots_{(10)}$

Konversi Biner Ke Hexadecimal

- Digit biner dikelompokkan 4 digit dari bit paling belakang (LSB)
- Kemudian tiap kelompok di konversi ke symbol hexadecimal yaitu 0-9 dan A-F
- Contoh: $00101100_2 = \dots_{16}$

| 00101100 | |
|----------|----------|
| 0010 | 1100 |
| 2 | 12 / (C) |

$$00101100_{(2)} = 2C_{(16)}$$

Problem - Konversi Biner Ke Hexadecimal

- Bagaimana jika jumlah digit pada biner ganjil ??? Maka ada bagian kelompok yang jumlahnya **kurang** dari 4 digit ???
- Contoh : **10100**₍₂₎
maka jika dibagi sesuai LSB, pembagiannya adalah **1** dan **0100**
 - **Salah satu bagian ada yang berjumlah 1 digit**
- **Solusi** : Tambahkan **digit 0** pada bagian yang kurang dari 4 digit

| 101000 | |
|--------------|-------------|
| 000 1 | 0100 |
| 1 | 4 |

$$10100_{(2)} = 14_{(16)}$$



Latihan

- Konversikan bilangan biner berikut ke hexadecimal:

- $11111010_{(16)} = \dots_{(2)}$

- $111101000_{(16)} = \dots_{(2)}$

- $110011_{(16)} = \dots_{(2)}$

- $0011001_{(16)} = \dots_{(2)}$

Konversi - Hexadecimal Ke Biner

- Tiap digit hexadecimal akan dikonversikan 4 digit biner.
- Konversi digit hexadecimal ke 4 digit biner dimulai dari bit paling kecil atau paling belakang (LSB)
- Contoh: $2C_{16} = \dots_2$

| 2C | |
|------|--------|
| 2 | C = 12 |
| 0010 | 1100 |

$$2C_{(16)} = 00101100_{(2)}$$

Latihan

- Konversikan bilangan hexadecimal berikut ke biner:
 - $17D_{(16)} = \dots_{(2)}$
 - $A64_{(16)} = \dots_{(2)}$
 - $FF1_{(16)} = \dots_{(2)}$
 - $BCD_{(16)} = \dots_{(2)}$

Referensi

- Mustafa Abd-el Bhar, Hesham El Rewini – Fundamentals of Computer Organization and Architecture 9th edition (2005)
- William Stallings – Computer Organization and Architecture Designing For Performance 9th Edition (2013)



TERIMA KASIH

ANY QUESTIONS?