



## Topic 2.2: Number System

CSGE601020 - Dasar-Dasar Pemrograman 1

Lintang Matahari Hasani, S.Kom., M.Kom. | Dr.Eng. Lia Sadita, S.Kom., M.Eng.

# Acknowledgement

This slide is an adapted version of 'Number System' slides used in DDP1 Course (2020/2021) by **Hafizh Rafizal Adnan, M.Kom.**

Several materials are reused from 'Sistem Bilangan pada Komputer' slides used in Dasar-Dasar Pemrograman 1 dengan Python (CSGE601020/4 SKS) Course (<https://ocw.ui.ac.id/course/view.php?id=142>) by **Fariz Darari, Ph.D.** and 'Sistem Bilangan dalam Komputer' slides by **Adila Krisnadhi, Ph.D.**

Some of the design assets used in these slides were provided by ManyPixels under a nonexclusive, worldwide copyright license to download, copy, modify, distribute, perform, and use the assets provided from ManyPixels for free, including for commercial purposes, without permission from or attributing the creator or ManyPixels.

Copyright 2020 MANYPIXELS PTE LTD



# In this session, you will learn ...

Intro to Data Representation: Images, Audio, & Text

Number System: Binary, Hexadecimal, & Octal

How to Convert Into Binary, Hexadecimal, & Octal

Signed Binary

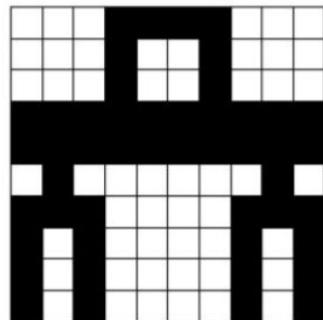
Two's Complement Binary



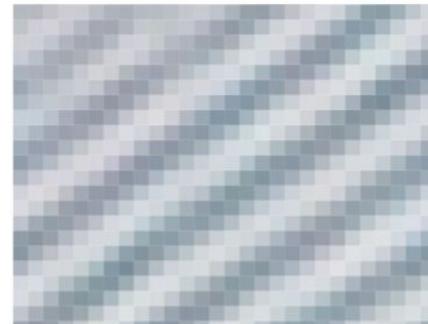
# Image Representation (1)

Representasi citra digital: **matriks** berisi kumpulan nilai warna (pixel)

- Tiap pixel direpresentasikan sebagai suatu **bilangan biner**



0	0	0	1	1	1	1	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	1	1	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1

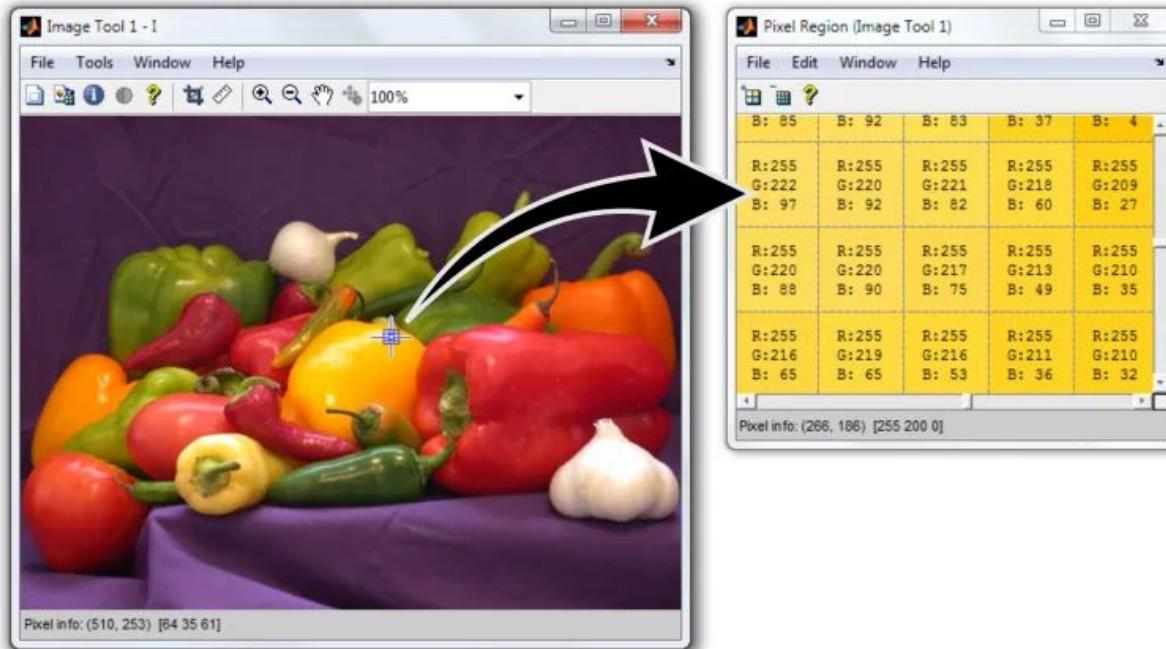


Contoh representasi citra dalam **Binary Images**

0: white  
1: black

*"one bit per pixel"*

# Image Representation (2)

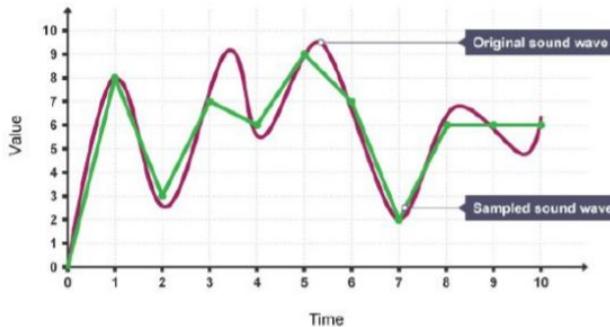
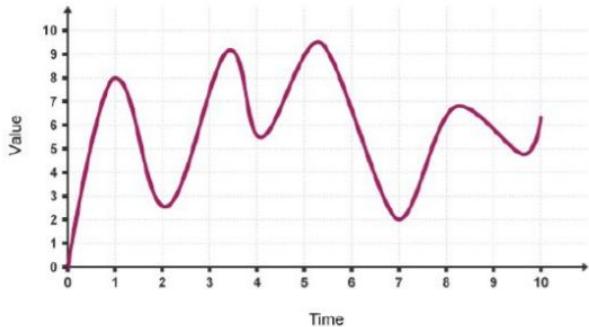


Contoh representasi citra dalam **24-bit RGB**

*Setiap pixel memiliki Red, Green, Blue values*

(255, 255, 255): brightest white  
(0, 0, 0): black

# Representasi Data Audio



Data audio digital: hasil **sampling** dari gelombang suara (yang bersifat analog)

# Character Representation

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	:	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	10110	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	10111	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[					
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135	I					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	-					

- Huruf/karakter direpresentasikan (encoding) sebagai **bilangan (biner)**.  
 → Contoh: ASCII dan UTF-8 encodings.

**ASCII Table.** One of text encoding/decoding standard  
<https://commons.wikimedia.org/wiki/File:ASCII-Table.svg>

American Standard Code for Information Interchange

## Triggering Question 1

Translate this binary code into text

1000100

1001001

1001111

Write your answer in the **comment section**



1000000	100	@
1000001	101	A
1000010	102	B
1000011	103	C
1000100	104	D
1000101	105	E
1000110	106	F
1000111	107	G
1001000	110	H
1001001	111	I
1001010	112	J
1001011	113	K
1001100	114	L
1001101	115	M
1001110	116	N
1001111	117	O
1010000	120	P
1010001	121	Q
1010010	122	R
1010011	123	S
1010100	124	T
1010101	125	U
1010110	126	V
1010111	127	W
1011000	130	X
1011001	131	Y
1011010	132	Z

# Sistem Bilangan Komputer

**11111100101<sub>2</sub>**

Binary (Base 2)

**3745<sub>8</sub>**

Octal (Base 8)

**2021<sub>10</sub>**

Decimal (Base 10)

**7E5<sub>16</sub>**

Hexadecimal (Base 16)

# Bilangan Desimal dan Contoh Ekspansinya

- Bilangan **desimal** memiliki basis (base) atau **radix 10**.
- Bilangan terdiri dari serangkaian digit, masing-masing dengan sebuah bobot (weight)

5	3	7	.	2	1	4	digit
100	10	1		1/10	1/100	1/1000	bobot

Misal:

**537.214**

- **Bobot: pangkat dari basis.**

5	3	7	.	2	1	4	digit
$10^2$	$10^1$	$10^0$		$10^{-1}$	$10^{-2}$	$10^{-3}$	bobot

- Nilai bilangannya diperoleh dengan cara:  
kalikan tiap digit dengan bobotnya, lalu jumlahkan:

$$(5 \times 10^2) + (3 \times 10^1) + (7 \times 10^0) + (2 \times 10^{-1}) + (1 \times 10^{-2}) \\ + (4 \times 10^{-3}) = 537.214$$

# Bentuk Umum Representasi Berbasis Posisi

- Bilangan  $D$  dalam basis/radix  $R$  ditulis dalam bentuk serangkaian digit, yakni:

$$D = d_{m-1}d_{m-2} \dots d_1d_0.d_{-1}d_{-2} \dots d_{-n}$$

di mana setiap  $d_{m-1}, \dots, d_0, d_{-1}, \dots, d_{-n}$  adalah digit yang diijinkan untuk radix  $R$ .

- Contoh: untuk desimal, radix  $R = 10$ . Digit yang diizinkan: 0,1,...,9.
- Contoh: untuk representasi biner, radix  $R = 2$ , dan digit yang diizinkan: 0,1.

Nilai  $D$  jika dinyatakan dalam desimal adalah:

$$D = \sum_{i=-n}^{m-1} d_i R^i$$

# Bilangan Biner (1)

Representasi **biner menggunakan radix 2**, sehingga bobot yang dipakai adalah pangkat dari 2.

1	1	0	1	.	0	1	binary digits atau <b>bits</b>
$2^3$	$2^2$	$2^1$	$2^0$		$2^{-1}$	$2^{-2}$	bobot dalam basis 2

Secara umum, representasi biner berbentuk:

$$b_{m-1}b_{m-2}\dots b_1b_0.b_{-1}b_{-2}\dots b_{-n}$$

- setiap  $b_{m-1}, \dots, b_0, b_{-1}, \dots, b_{-n}$  bernilai 0 atau 1
- bit  $b_{m-1}$  (yang bobotnya terbesar) disebut **most significant bit (MSB)**
- bit  $b_{-n}$  (yang bobotnya terkecil) disebut **least significant bit (LSB)**

Misal:

**1101.01**

- Posisi digit menandakan pangkat dari 2
- Digit yang dipakai hanya 0 dan 1

# Bilangan Biner (2)

Bilangan biner digunakan untuk **mewakili data** pada komputer

Bilangan biner adalah bilangan yang memiliki **basis dua**:

**0 dan 1**

- Sebagai catatan, bilangan yang biasa kita pakai adalah desimal, yang berbasis sepuluh: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Biner digunakan karena mudah **murah** dan **mudah** diimplementasikan secara **low level** (misal: 0/1, false/true, low voltage/high voltage, etc )
- **Perintah dalam bahasa mesin** dikodekan dalam format biner (binary)

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

# Biner ke Desimal (1)

Nilai desimal dari bilangan biner **1101.01** adalah:

$$\begin{aligned}(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) \\+ (1 \times 2^{-2}) \\= 8 + 4 + 0 + 1 + 0 + 0.25 \\= 13.25\end{aligned}$$

Secara umum, jika bilangan biner B berbentuk

$$b_{m-1}b_{m-2}\dots b_1b_0.b_{-1}b_{-2}\dots b_{-n}$$

maka, nilai desimalnya:

$$B = \sum_{i=-n}^{m-1} b_i \times 2^i$$

## Biner ke Desimal (2)

$$100101_2 = [(1) \times 2^5] + [(0) \times 2^4] + [(0) \times 2^3] + [(1) \times 2^2] + [(0) \times 2^1] + [(1) \times 2^0]$$

$$100101_2 = [1 \times 32] + [0 \times 16] + [0 \times 8] + [1 \times 4] + [0 \times 2] + [1 \times 1]$$

$$100101_2 = 37_{10}$$

# Pangkat dari 2

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024$$

$$2^{11} = 2048$$

$$2^{12} = 4096$$

$$2^{13} = 8192$$

$$2^{14} = 16384$$

$$2^{15} = 32768$$

$$2^{-1} = 0.5$$

$$2^{-2} = 0.25$$

$$2^{-3} = 0.125$$

$$2^{-4} = 0.0625$$

$$2^{-5} = 0.03125$$

$$2^{-6} = 0.015625$$

## Mari Mencoba!: Biner ke Desimal

Ubah bilangan-bilangan dalam representasi biner berikut menjadi representasi desimal.

$$\begin{aligned}11011_2 &= ([1 \times 2^4] + [1 \times 2^3] + [0 \times 2^2] + [1 \times 2^1] + [1 \times 2^0])_{10} \\&= ([1 \times 16] + [1 \times 8] + [0 \times 4] + [1 \times 2] + [1 \times 1])_{10} \\&= 27_{10}\end{aligned}$$

$$101101_2 = \dots\dots\dots_{10}$$

$$1011.011_2 = \dots\dots\dots_{10}$$

# Desimal Ke Biner (1)

Bagi **bagian bulat desimal** (di depan dot/koma) dengan 2 berulang-ulang **hingga hasil bagi atau quotient-nya 0**. Lalu kumpulkan sisa bagi (remainder) mulai dari hasil bagi yang terakhir.

Untuk **bagian pecahan** (di belakang dot/koma), kalikan bagian pecahan tersebut dengan 2 **sampai jadi 0 di belakang koma**. Lalu kumpulkan bagian bulat dari hasil perkalian tersebut dari depan.

Contoh: desimal 162.375

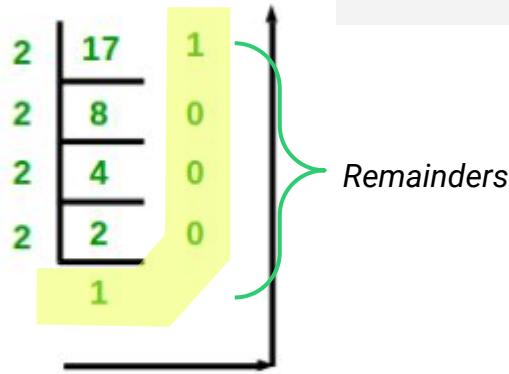
$$\begin{array}{rcl}
 162/2 = 81 \text{ rem } 0 \rightsquigarrow b_0 = 0 & \uparrow & \\
 81/2 = 40 \text{ rem } 1 \rightsquigarrow b_1 = 1 & \uparrow & \\
 40/2 = 20 \text{ rem } 0 \rightsquigarrow b_2 = 0 & \uparrow & \\
 20/2 = 10 \text{ rem } 0 \rightsquigarrow b_3 = 0 & \uparrow & \\
 10/2 = 5 \text{ rem } 0 \rightsquigarrow b_4 = 0 & \uparrow & \\
 5/2 = 2 \text{ rem } 1 \rightsquigarrow b_5 = 1 & \uparrow & \\
 2/2 = 1 \text{ rem } 0 \rightsquigarrow b_6 = 0 & \uparrow & \\
 1/2 = 0 \text{ rem } 1 \rightsquigarrow b_7 = 1 & \uparrow &
 \end{array}$$

$$\begin{array}{rcl}
 0.375 \times 2 = 0.75 \rightsquigarrow b_{-1} = 0 & \downarrow & \\
 0.75 \times 2 = 1.5 \rightsquigarrow b_{-2} = 1 & \downarrow & \\
 0.5 \times 2 = 1.0 \rightsquigarrow b_{-3} = 1 & \downarrow &
 \end{array}$$

**162.375<sub>10</sub> = 10100010.011<sub>2</sub>**

# Desimal Ke Biner (2)

Decimal number : 17



*Hint:* Use successive division by 2

Binary number: 10001

$$\begin{array}{r}
 2 \overline{)47} \\
 2 \overline{)23} \quad 1 \\
 2 \overline{)11} \quad 1 \\
 2 \overline{)5} \quad 1 \\
 2 \overline{)2} \quad 1 \\
 2 \overline{)1} \quad 0 \\
 0 \quad 1
 \end{array}$$

Remainder

$$(47)_{10} = (101111)_2$$

## Triggering Question 2

---

**Ubah bilangan-bilangan desimal berikut ke dalam representasi biner.**

$$17_{10} = \dots\dots\dots_2$$

$$17.125_{10} = \dots\dots\dots_2$$

$$0.375_{10} = \dots\dots\dots_2$$

# Hexadecimal

Notasi biner paaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaanjang :(

- Heksadesimal (Radix 16)
- bisa dipakai untuk **menyederhanakan notasi biner.**
  - ◆ Contohnya digunakan untuk mendeskripsikan **IPv6** addresses atau

IPv6: FE80:0000:0000:0000:0202:B3FF:FE1E:8329

- ◆ **24-bit colors**

#f5f92a

Digit yang dipakai:

**0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F** (boleh huruf kecil).

# Menyingkat biner dengan Hexadecimal

Desimal	Biner	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

- Konversi **biner ke hex**: ambil 4 bit dari dot/koma, konversi sesuai tabel, tambahkan 0 di ujung jika perlu.
- Konversi **hex ke biner**: konversi tiap digit dengan 4 bit sesuai tabel.

	Biner	Hex
	10110100	B4
	101.101	5.A
	1001100001.00110101	261.35
	1010011100	?
	10.10111	?

# Hex pada Python/Java

## Unicode character values

Contoh: "\u03B1" adalah karakter α

## Hexadecimal integer biasanya ditulis dengan awalan 0x

Contoh: 0xFACE21



# Base-8 Octal

Representasi octal menggunakan **8 digits**

- Radix = 8
- Digits:

**0,1,2,3,4,5,6,7**

Banyak digunakan di awal sejarah komputer, sekarang masih muncul misalnya di **file access permissions untuk sistem Unix**.

Filename	Permissions	Octal	Description
private	-----	000	Nobody can read or write this file.
readable	-r-----	400	The owner of the file (jjm in this case) can read it.
runable	-r-x----	500	The owner of the file can read and execute it.
writable	-rw-----	600	The owner of the file can read and write the file.

# Konversi Biner ke Octal

- Konversi **biner ke octal**: ambil 3 bit dari dot/koma, konversi sesuai tabel, tambahkan 0 diujung jika perlu.
- Konversi **octal ke biner**: konversi tiap digit dengan 3 bit sesuai tabel.

Desimal	Biner	Octal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
55	110111	67
?	?	765
?	?	7.432

### Triggering Question 3

---

**Konversi bilangan-bilangan berikut.**

$$\mathbf{9FB}_{16} = (\dots\dots\dots)_2$$

$$\mathbf{4.C8A}_{16} = (\dots\dots\dots)_8$$

$$\mathbf{330}_{10} = (\dots\dots\dots)_8$$

# Unsigned Binary Number

Jumlah bit menentukan **rentang** bilangan yang dapat direpresentasikan.

Jika hanya mempertimbangkan bilangan positif:

**2 bit** min:  $0_{10}$ , max:  $3_{10}$

**4 bit** min:  $0_{10}$ , max:  $15_{10}$

**8 bit** min:  $0_{10}$ , max:  $255_{10}$

**16 bit** min:  $0_{10}$ , max:  $W^{16}-1 = 65535_{10}$

**n bit** min:  $0_{10}$ , max:  $2^n - 1_{10}$

# Signed Binary Number

Untuk mengakomodasi **bilangan negatif**, n bit tidak merepresentasikan 0 s.d.  $2^{n-1}$   
Tapi, n bit merepresentasikan  $-(2^{n-1})$  s.d.  $+(2^{n-1}-1)$

**Unsigned**

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

**Signed**

1	0	0	1
sign			

# Representasi Bilangan Negatif

## **Signed-and-magnitude:**

Tambahkan sign pada bilangan biner

## **One's complement:**

Invert semua bilangan biner (0 menjadi 1 dan sebaliknya)

## **Two's complement**

- a. Konversi ke 1s complement
- b. Tambahkan hasil konversi ke 1

# Two's Complements

Digunakan di banyak komputer modern, karena lebih mudah diimplementasikan di hardware.

Mengubah bilangan biner ke two's complement-nya:

- **Ubah semua bit 1 jadi 0 dan semua 0 jadi 1.**
- **Jumlahkan dengan 1 ke hasilnya.**
  
- Jumlah digit yang disediakan harus diperhatikan.
- Buang carry (lebihan dalam operasi) jika ada.

Notes:

- Nilai  $2^n - 1$  dalam sistem biner menjadi  $n$  buah bit 1.  
Contoh:  $2^5 - 1 = 11111_2$
- Untuk bilangan biner  $B$  yang terdiri  $n$  bit,  $(2^n - 1) - B$  adalah **komplemen dari  $B$** . Contoh: komplemen (5-bit) dari  $10001_2$  adalah  $(2^5 - 1) - 10001_2 = 01110_2$ .
- Two's complement dari  $B$  adalah  $1 + \text{komplemen dari } B$ , atau  $(2^n - 1) - B + 1$ . Jadi, two's complement dari  $10001$  adalah  $1 + 01110 = 01111$ .

# Contoh untuk 4 bit

Untuk  $-7_{10}$

Biner dari  $7_{10}$ :

$0111_2$

1's comp. dari  $7_{10}$ :

$1000_2$

2's comp dari  $7_{10}$ :

$$1000_2 + 0001_2 = 1001_2$$

Decimal	Two's compl. bin.	Signed-magnitude bin.
-8	1000	-
-7	1001	<span style="background-color: pink;">1111</span>
-6	1010	<span style="background-color: pink;">1110</span>
-5	1011	<span style="background-color: pink;">1101</span>
-4	1100	<span style="background-color: pink;">1100</span>
-3	1101	<span style="background-color: pink;">1011</span>
-2	1110	<span style="background-color: pink;">1010</span>
-1	1111	<span style="background-color: pink;">1001</span>
0	0000	<span style="background-color: green;">1000 atau 0000</span>
1	0001	<span style="background-color: green;">0001</span>
2	0010	<span style="background-color: green;">0010</span>
3	0011	<span style="background-color: green;">0011</span>
4	0100	<span style="background-color: green;">0100</span>
5	0101	<span style="background-color: green;">0101</span>
6	0110	<span style="background-color: green;">0110</span>
7	0111	<span style="background-color: green;">0111</span>

*Sign*

#### Triggering Question 4

---

**Konversikan bilangan berikut dengan representasi binary 8 bit 2s complement**

$$-21_{10} = (\dots\dots\dots)_2$$

$$-118_{10} = (\dots\dots\dots)_2$$

$$-125_{10} = (\dots\dots\dots)_2$$

# Penjumlahan dengan Two's Complement

- Penjumlahan dilakukan seperti biasa asalkan tidak melebihi kapasitas (tidak overflow)
- Buang carry jika ada.

Contoh untuk penjumlahan untuk 4 bit:

$$\begin{array}{r} & 0011 & \text{(carry)} & 0110 & \text{(carry)} \\ 0010 & (+2) & 0011 & (+3) & 0111 & (+7) \\ 0100 & (+4) & 0011 & (+3) & 0110 & (+6) \\ \hline & + & \hline & + & \hline & + \\ 0110 & (+6) & 0110 & (+6) & 1101 & (-5) \text{ invalid!} \\ & & & & & \text{overflow } ^\wedge \\ 1100 & \text{(carry)} & & & & \\ 1110 & (-2) & & & & \\ 1100 & (-4) & & & & \\ \hline & + & & & & \\ 1010 & (-6) & & & & \end{array}$$

# Pengurangan dengan Two's Complement

- Pengurangan dilakukan dengan **menjumlahkan** bilangan yang dikurangi (minuend) **dengan two's complement dari bilangan pengurang** (subtrahend)
- Buang carry jika ada.

Contoh pengurangan untuk 4 bit:

$$\begin{array}{r}
 & 0000 \quad (\text{carry}) \\
 0010 \quad (+2) & \Rightarrow 0010 \quad (+2) \\
 0100 \quad (+4) & \Rightarrow 1100 \quad (-4) \quad \boxed{\text{two's complement dari } 4_{10}} \\
 \hline - & \hline + \\
 & 1110 \quad (-2)
 \end{array}$$

$$\begin{array}{r}
 & 1100 \quad (\text{carry}) \\
 1110 \quad (-2) & \Rightarrow 1110 \quad (-2) \\
 1100 \quad (-4) & \Rightarrow 0100 \quad (+4) \\
 \hline - & \hline + \\
 & 0010 \quad (+2)
 \end{array}$$

# Konversi Biner Two's Complement ke Desimal

- Sama caranya dengan konversi biner unsigned untuk semua bit kecuali sign bit.
- Untuk sign bit, bobotnya dinegatifkan sebelum dikalikan dengan bit-nya.

Contoh untuk 3 bit:

- Unsigned:

$$1100_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 12_{10}$$

$$0100_2 = (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 4_{10}$$

- Signed (two's complement):

$$1100_2 = (1 \times -(2^3)) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = -4_{10}$$

$$0100_2 = (0 \times -(2^3)) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 4_{10}$$

# Sign Extension

Kadang-kadang kita perlu “memperpanjang” representasi biner, misalnya dari 4 bit jadi 8 bit.

Agar tidak kehilangan informasi sign bit, maka tambahan bit (padding) dilakukan di sebelah kiri sign bit dengan menduplikasi sign bit.

Nilai bilangan tidak berubah.

- 4 bit: 0101      6 bit: 000101  
Nilainya sama-sama  $5_{10}$ .
- 4 bit: 1100      6 bit: 111100  
Nilainya sama-sama  $-4_{10}$ .  
(Coba cek dengan menghitung two's complement dari 4 untuk panjang 6 bit.)

# Bits, Bytes, Words

Bit: satu digit 0 atau 1

**Byte: 8 bit**

**Word: standar satuan penyimpanan data di komputer**

- Komputer 32-bit, 1 word = 32 bit = 4 byte
- Komputer 64-bit, 1 word = 64 bit = 8 byte

## About

Your PC is monitored and protected.

[See details in Windows Security](#)

## Device specifications

Device name	DESKTOP-ME6KFVR
Processor	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
Installed RAM	8,00 GB (7,88 GB usable)
Device ID	DCA7B48D-4372-42D2-AEC9-85E53AD9E91B
Product ID	00327-35822-53512-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	Pen support

# Aproksimasi Desimal ke Biner

$$2^{10} = 1024 \approx 10^3 = 1000 \text{ (kilo)}$$

$$2^{20} = 1,048,576 \approx 10^6 = 1,000,000 \text{ (mega)}$$

$$2^{30} = 1,073,741,824 \approx 10^9 = 1,000,000,000 \text{ (giga)}$$

$$2^{40} = 1,099,511,627,776 \approx 10^{12} = 1,000,000,000,000 \text{ (tera)}$$

$$2^{50} = 1,125,899,906,842,624 \approx 10^{15} = 1,000,000,000,000,000 \text{ (peta)}$$

$$2^{60} \approx 10^{18} \text{ (exa)}$$

$$2^{70} \approx 10^{21} \text{ (zetta)}$$

⋮

# Binary, octal, and hexadecimal in Python

```
# bin to dec  
print(int(0b101))  
  
# dec to bin  
print(bin(5))  
  
# oct to dec  
print(int(0o101))
```

```
# dec to oct  
print(oct(65))  
  
# hex to dec  
print(int(0x101))  
  
# dec to hex  
print(hex(257))
```

# Review Questions

**Explain what binary number is? What does base-2 number mean?**

**Mention number systems that are used in computer system?**

**Why binary number is used in most of data representation?**





# Q&A Session

