

Sistem Digital

Penggunaan Sistem Bilangan dan Pengkodean

-3-



Penggunaan Bilangan Biner

- # Bilangan biner digunakan dalam komputer yang biasa tidak terlihat oleh pengguna
- # Namun kemampuan untuk membaca bilangan biner sangat menguntungkan
- # Karena komputer menyimpan baik instruksi maupun data dalam bentuk bilangan biner



Format Data

- ✚ Bilangan biner adalah sistem yang dipilih baik untuk penyimpanan data maupun untuk pemrosesan suatu operasi
- ✚ Manusia menggunakan bahasa citra dan suara dalam berkomunikasi dan menggunakan alfanumerik dan simbol yang mewakili bahasa



Format Data

- + Komunikasi yang dikenal manusia misalnya melalui :
 - + Foto, tabel, diagram
 - + Hitam, putih, warna
 - + Gambar bergerak maupun tidak bergerak
 - + Suara, musik
 - + Tertulis melalui huruf dan angka



Data Karakter Alfanumerik

- ✚ Data yang digunakan dalam komputer disajikan dalam bentuk yang bisa dibaca oleh manusia



Data Karakter Alfanumerik

Tipe Data	Standar
Alphanumeric	Unicode, ASCII, EDCDIC
Image (bitmapped)	<ul style="list-style-type: none">•GIF (graphical image format)•TIF (tagged image file format)•PNG (portable network graphics)
Image (object)	PostScript, JPEG, SWF (macromedia flash)
Outline graphics and fonts	PostScript, TrueType
Sound	WAV, AVI MP3, MIDI, MWA
Page diskription	PDF (adobe portable document format), HTML, XML
Video	Quicktime, MPEG-2, RealVideo, WMV



Data Karakter Alfanumerik

- ✚ 3 macam kode alfanumerik :
 - ✚ Unicode
 - ✚ ASCII (American Standard Code for Information Interchange)
 - ✚ EBCDIC (Extended Binary Coded Decimal Interchange Code), dibuat oleh IBM



Kode ASCII

MSB \ LSB	0	1	2	3	4	5	6	7
0	NULL	DLE	SP	O	@	P		p
1	SOH	DC1	!	1	A	Q	a	w
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACJ	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y



Kode ASCII

MSB \ LSB	0	1	2	3	4	5	6	7
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL



Kode ASCII & EBCDIC

- ✦ Kode untuk masing-masing simbol dinyatakan dalam bilangan desimal, dengan angka yang most-significant digit terdapat diatas dan least significant digit ada di samping kiri
- ✦ Kode standar ASCII adalah kode 7 bit sehingga hanya ada 128 simbol dalam tabel.
- ✦ Kode EBCDIC merupakan kode 8 bit sehingga memiliki 256 simbol



Kode ASCII & EBCDIC

- ✚ Keterbatasan tersebut diatasi dengan dikeluarkannya kode internasional baru yang mempunyai kode 16 bit yaitu UNICODE



Type Data

+ 5 tipe data dasar :

+ **Boolean** : variable / konstanta dengan 2 nilai yaitu true atau false

+ **Char** : tipe data karakter. String adalah array dari karakter

+ **Tipe data terbilang** : tipe data yang dibuat oleh pengguna dimana nilai dimasukkan dalam definisi



Type Data

+ 5 tipe data dasar :

+ **Integer** : semua bilangan baik positif maupun negatif

+ **Real** : bilangan yang mempunyai bilangan desimal atau bilangan yang memiliki kemampuan untuk memproses dan menyimpan lebih besar daripada bilangan integer



Penyajian Data Integer dan Bilangan Floating Point

- ✚ Komputer menyimpan semua data dan instruksi program dalam bentuk biner tanpa ada ketentuan khusus yang dibuat untuk penyimpanan tanda / decimal point yang berhubungan dengan bilangan, kecuali ketika bilangan itu disimpan sebagai string yang tidak bisa digunakan untuk perhitungan.



Penyajian Data Integer dan Bilangan Floating Point

- + Bilangan integer
 - + Bilangan integer bertanda
 - + Bilangan integer tidak bertanda



Bilangan Biner Tak Bertanda

- ✚ Batasan integer yang dapat disimpan suatu komputer ditentukan oleh jumlah bit.
8 bit menyimpan integer tak bertanda 0-255,
16 bit menyimpan integer tak bertanda 0-65535



Bilangan Biner Tak Bertanda

- 4 lokasi penyimpanan sebesar 1 byte digunakan untuk menyediakan batasan 32 bit

Memori Location	1 byte
M	Bit 31-Bit 24
M+1	Bit 23- Bit 16
M+2	Bits 15- Bit 8
M+3	Bit 7- Bit 0
M+4	Next data item



Binary Coded Decimal (BCD)

- ✦ Setiap angka pada bilangan desimal disajikan sebagai bilangan biner
- ✦ Memerlukan 4 bit per-angka, 8 bit menyimpan 2 angka BCD
- ✦ 4 bit dapat menyimpan 16 nilai yang berbeda yaitu 0-F (hexadecimal), tetapi dengan BCD nilai A-F tidak digunakan



Binary Coded Decimal (BCD)

Banyaknya bit	Range BCD		Range Binary	
4	0-9	1 digit	0-15	1+ digit
8	0-99	2 digit	0-255	2+ digit
12	0-999	3 digit	0-4095	3+ digit
16	0-9999	4 digit	0-65535	4+ digit
20	0-99999	5 digit	0-1 million	6 digit
24	0-999999	6 digit	0-16 million	7+ digit



Binary Coded Decimal (BCD)

✚ Contoh : Bilangan Biner 2954

2 9 5 4

0010 1001 0101 0100 = 0010100101010100

✚ Contoh : Bilangan BCD 101001110010111

0101 0011 1001 0111

5 3 9 7 = 5397



Perkalian BCD

Contoh 76 x 7

76	→	0111	0110	_{bcd}	
x 7	→		0111	_{bcd}	
42	→		101010	_{bin}	→
49	→	110001	_{bin}	→	+
4 ¹ 32	→				0100 1001
13	←	adjust	Convert	→	0100 1101 0010
532	→	carry	13 back	→	0001 0011
			to bcd		0101 0011 0010
					= 5 3 2
					_{bcd}



Bilangan Integer Bertanda

- ✚ Integer tidak bertanda dapat dikonversikan langsung ke bilangan biner dan diproses tanpa diperlakukan khusus
- ✚ Penjumlahan bilangan yang bertanda memperumit masalah karena tidak ada cara langsung untuk menyatakan tanda dalam bentuk biner.
- ✚ Salah satunya menggunakan ***2's complement***



Penyajian Sign dan Magnitude

- ✚ Penyajian Sign dan Magnitude menggunakan tanda + dan – pada suatu nilai
- ✚ Komputer hanya mengenal 0 dan 1, sehingga memilih bit tertentu untuk menyatakan tanda, misalnya bit paling kiri dan ditentukan jika 0 menyatakan + dan jika 1 menyatakan –



Penyajian Sign dan Magnitude

- ✚ Misalkan 32 bit untuk penyimpanan dan manipulasi suatu bilangan
 - ✚ 1 bit untuk tanda
 - ✚ 31 bit untuk nilai bilangan itu
 - ✚ Bit paling kiri digunakan sebagai tanda dimana jika 0 menyatakan + dan jika 1 menyatakan –



Nine's Decimal dan One's Binary Complement

- ✚ Komputer menggunakan metode yang berbeda untuk menyajikan bilangan integer bertanda (complement)
- ✚ Suatu bilangan sudah dinyatakan pada bilangan itu sendiri sehingga tanda tidak perlu ditangani secara khusus dan penggunaan complement konsisten untuk semua bilangan tanda berbeda.



Nine's Decimal dan One's Binary Complement

- ✚ bentuk penyajian complement :
 - ✚ Radix complement, basis yang digunakan dalam operasi complement dikurangi (diminished) dengan 1 dari radix / basisnya
 - ✚ Penyajian dari diminished radix complement untuk basis 10 menggunakan nilai 9 dari basisnya dan biner menggunakan 1



Nine's Decimal

- + 3 angka dalam sistem decimal dimanipulasi dengan membagi dua bilangan tersebut pada angka 500
- + 0-499 sebagai bilangan positif
- + Bilangan yang dimulai dengan angka 5, 6, 7, 8 atau 9 dianggap sebagai bilangan negatif
- + Contoh nine's complement -467 adalah : $(999-467) = 532$ menunjukkan angka negatif



Nine's Decimal

Number	Negative		Positive	
Representation method	Complement		Bilangan itu sendiri	
Range number decimal	-499	-000	+0	499
Calculation	999 number –		None	
Contoh Representation	500	999	0	499



One's Complement

- ✚ Jangkauan suatu bilangan biner harus dibagi dua tepat di tengah-tengah
- ✚ Bilangan yang dimulai 0 adalah positif
- ✚ Bilangan yang dimulai 1 adalah negatif
- ✚ One's complement (invers) dilakukan dengan mengubah 0 menjadi 1 dan sebaliknya



One's Complement

✚ One's complement untuk bilangan biner 8 bit :

Number	Negative		Positive	
Representation method	Complement		Bilangan itu sendiri	
Range number decimal	-127_{10}	-0_{10}	$+0_{10}$	127_{10}
Calculation	Inversi		None	
Contoh Representation	10000000	11111111	00000000	01111111



One's Complement

- ✚ Penjumlahan 2 bilangan tanpa tanda mengikuti aturan end-around carry :

$$\begin{array}{r} 45 = 00101101 \\ 58 = 00111010 \quad + \\ \hline 103 = 01100111 \end{array}$$



One's Complement

- ✚ Penjumlahan 2 bilangan dengan tanda, dilakukan inversi terhadap bilangan yang bertanda :

$$45 = 0000000000101101$$

$$\underline{-58 = 1111111111000101} \quad +$$

$$-13 = 111111111110010$$

- ✚ -13 di konversi ulang menjadi 00001101



Ten's Complement

- ✚ Ten's complement mengatasi kekurangan pada nine's complement dimana adanya 2 buah komplement dalam skalanya, sehingga dengan menggeser skala negatif 1 langkah ke kanan, kita dapat membuat sistem komplement yang hanya mempunyai 1 buah 0. dan menggunakan radix sebagai basis untuk operasi komplement.



Ten's Complement

- ✦ Tetapi sistemnya lebih sulit untuk mencari komplemen dalam 1 bilangan.
- ✦ Contoh ten's complement dari 247 = $(1000 - 247) = 753$ (menunjukkan bilangan negatif)
- ✦ Tens's complement = nine's complement + 1



Two's Complement

Two's complement dari bilangan 8 bit

Number	Negative		Positive	
Representation method	Complement		Bilangan itu sendiri	
Range number decimal	-128_{10}	-1_{10}	$+0_{10}$	127_{10}
Calculation	Inversi		None	
Contoh Representation	10000000	11111111	00000000	01111111



Two's Complement

- ✚ Bilangan positif adalah bilangan itu sendiri dan diawali dengan angka 0
- ✚ Bilangan negatif kecil dinyatakan dengan bilangan yang mempunyai angka 1 banyak, seperti -2 pada two's complement = 1111110 sedangkan -128 yang merupakan bilangan negatif terbesar pada two's complement dinyatakan 10000000
- ✚ Penyajian bilangan negatif pada one's complement dan two's complement hanya beda 1



Bilangan Biner Komplement 1

✚ Bilangan biner komplement 1 dapat diperoleh dengan mengganti semua bit 0 menjadi 1, dan semua bit 1 menjadi 0

✚ Contoh :

bilangan biner : 100101

bilangan biner komplement 1 : 011010



Bilangan Biner Komplement 2

✚ Bilangan biner komplement 2 dapat diperoleh dengan menambahkan 1 pada bilangan biner komplement 1

✚ Contoh :

bilangan biner : 100101

bilangan biner komplement 1 : 011010
+1

bilangan biner komplement 2 : 011011

✚ Bilangan biner komplement 2 digunakan untuk pengurangan bilangan biner



Konversi biner ke kode Gray

- # Kode gray biasanya dipakai pada mechanical encoder. Misalnya telegraf.
- # Konversi biner ke kode gray, terdapat beberapa langkah :
 - a. Tulis kebawah bilangan biner
 - b. MSB bilangan biner adalah MSB kode gray
 - c. Jumlahkan (dengan menggunakan modulo 2) bit pertama bilangan biner dengan bit kedua, hasilnya adalah bit kedua kode gray
 - d. Ulangi langkah c untuk bit-bit selanjutnya



Konversi biner ke kode Gray

✚ Contoh bilangan biner 1001001 menjadi kode gray :

biner	gray	keterangan
1001001		
1001001	1	MSB biner = MSB gray
1001001	11	$1 \text{ modulo } 2 \ 0 = 1$
1001001	110	$0 \text{ modulo } 2 \ 0 = 0$
1001001	1101	$0 \text{ modulo } 2 \ 1 = 1$
1001001	11011	$1 \text{ modulo } 2 \ 0 = 1$
1001001	110110	$0 \text{ modulo } 2 \ 0 = 0$
1001001	1101101	$0 \text{ modulo } 2 \ 1 = 1$

jadi kode gray dari bilangan biner 1001001 adalah 1101101



Konversi kode gray ke bilangan biner

- ✚ Terdapat beberapa langkah untuk mengubah kode gray menjadi bilangan biner
 - a. Tulis kebawah bilangan biner
 - b. MSB kode gray adalah MSB bilangan biner
 - c. Jumlahkan (dengan menggunakan modulo 2) bit pertama kode gray dengan bit kedua bilangan biner, hasilnya adalah bit kedua bilangan biner
 - d. Ulangi langkah c untuk bit-bit selanjutnya



Konversi kode gray ke bilangan biner

✚ Contoh kode gray 1101101 menjadi bilangan biner :

gray	biner	keterangan
1101101		
1101101	1	MSB biner = MSB gray
1101101	10	$1 \text{ modulo } 2 \text{ } 1 = 0$
1101101	100	$0 \text{ modulo } 2 \text{ } 0 = 0$
1101101	1001	$0 \text{ modulo } 2 \text{ } 1 = 1$
1101101	10010	$1 \text{ modulo } 2 \text{ } 1 = 0$
1101101	100100	$0 \text{ modulo } 2 \text{ } 0 = 0$
1101101	1001001	$0 \text{ modulo } 2 \text{ } 1 = 1$

jadi bilangan biner dari kode gray 1101101 adalah 1001001



Kode Excess-3

✚ Kode excess-3 didapat dengan menjumlahkan nilai decimal dengan 3, selanjutnya di ubah ke dalam bilangan biner

decimal	biner	excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100



Daftar Pustaka

- ✚ Digital Principles and Applications, Leach-Malvino, McGraw-Hill
- ✚ Sistem Digital konsep dan aplikasi, freddy kurniawan, ST.
- ✚ Elektronika Digital konsep dasar dan aplikasinya, Sumarna, GRAHA ILMU
- ✚ Pengenalan Teknologi Komputer dan Informasi, Janner Simarmata, Andi Offset, Yogyakarta, 2006 (Bab 9,10)



Alhamdulillah....

Thanks!

