



A. Sistem Bilangan

Sistem Bilangan adalah kumpulan simbol khusus yang digunakan dalam membangun sebuah bilangan. Ada beberapa sistem bilangan yang dipergunakan dalam sistem bilangan digital, yang paling umum digunakan adalah *desimal*, *biner*, *oktal* dan *heksadesimal*.

Sistem bilangan yang umum dipakai adalah bilangan desimal, karena kemudahannya dan sudah dipergunakan dalam kehidupan sehari-hari.

Penulisan basis sistem bilangan biasanya diakhir bilangan berupa angka yang diperkecil / *subscrip* :

Contoh :

200_{10} (Desimal atau basis 10).

1010_2 (Biner atau basis 2).

262_8 (Oktal atau basis 8).

$2AB_{16}$ (Hexadesimal atau basis 16).

1. Sistem Bilangan *Desimal*

Sistem bilangan desimal adalah bilangan yang menggunakan sepuluh simbol yaitu 0 s/d 9. Sistem bilangan desimal biasanya disebut sistem bilangan radiks/basis 10.

2. Sistem Bilangan *Biner*

Sistem ini menggunakan dua simbol yaitu 0 dan 1. Disebut juga sistem bilangan radiks/basis 2. Biner merupakan bilangan dasar yang digunakan dalam sistem komputer digital.

Penulisan bilangan biner dalam komputer biasanya dikelompokkan per 4 bilangan, misalnya : 1010 0001.

Contoh : $0010_2 = 0010 = 2_{10}$
 $1010_2 = 1010 = 10_{10}$

3. Sistem Bilangan *Oktal*

Sistem ini menggunakan tujuh simbol yaitu 0 s/d 7. Disebut juga sistem bilangan radiks/basis 8.



Contoh : $2_8 = 2_{10}$
 $10_8 = 8_{10}$

4. Sistem Bilangan Hexadesimal

Sistem ini menggunakan **enam belas** simbol khusus, yaitu 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Disebut juga sistem bilangan radiks/basis 16 dan merupakan satu-satunya sistem bilangan yang menggunakan huruf.

Huruf-huruf A, B, C, D, E, F berturut-turut nilainya adalah : 10, 11, 12, 13, 14, 15.

Contoh : $8_{16} = 8_{10}$
 $A_{16} = 10_{10}$
 $1A_{16} = 26_{10}$



Macam-macam Sistem Bilangan

Sistem	Radiks/ Basis	Himpunan/Elemen Digit	Contoh
Desimal	$r = 10$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	255_{10}
Biner	$r = 2$	$\{0, 1\}$	11111111_2
Oktal	$r = 8$	$\{0, 1, 2, 3, 4, 5, 6, 7\}$	377_8
Heksadesimal	$r = 16$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$	FF_{16}

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



B. Konversi Bilangan

Bilangan Desimal

Bilangan desimal ini merupakan sistem bilangan dengan basis 10 artinya bilangan yang memiliki bobot ke pangkat 10 (semisal 10^0 , 10^1 , 10^2 , 10^3 , ..., 10^n).

Didalam bilangan desimal tempat penulisan angka yang paling kiri adalah angka yang memiliki bobot ke pangkat paling besar, sehingga bobot angka ke pangkat makin ke kanan akan memiliki angka ke pangkat semakin kecil.

Angka yang memiliki bobot ke pangkat paling kecil disebut *least significant digit* (LSD), sedangkan bobot ke pangkat paling besar disebut *most significant digit* (MSD).

	ribuan	ratusan	puluhan	satuan
10^n	10^3	10^2	10^1	10^0
...	1000	100	10	1



$$\begin{array}{rcl}
 2 & 5 & 4_{(10)} \\
 \swarrow & \searrow & \searrow \\
 \rightarrow 10^0 & = 4 \times 10^0 & = 4 \times 1 = 4 \\
 \rightarrow 10^1 & = 5 \times 10^1 & = 5 \times 10 = 50 \\
 \rightarrow 10^2 & = 2 \times 10^2 & = 2 \times 100 = 200 \\
 & & \hline
 & & 254_{10}
 \end{array}$$

LSD

MSD



Bilangan Biner

Bilangan *Biner* ini merupakan sistem bilangan dengan basis 2 artinya bilangan yang memiliki bobot ke pangkat 2 (semisal 2^0 , 2^1 , 2^2 , 2^3 , ..., 2^n). Bilangan yang terdapat pada bilangan biner adalah 0 dan 1.

Angka yang memiliki bobot ke pangkat paling kecil disebut *least significant bit* (LSB), sedangkan bobot ke pangkat paling besar disebut *most significant bit* (MSB).

Contoh : 1101_2

1	1	0	1	(2)		
↓	↓	↓	↓		2^0	$= 1 \times 2^0$
↓	↓	↓	↓		2^1	$= 0 \times 2^1$
↓	↓	↓	↓		2^2	$= 1 \times 2^2$
↓	↓	↓	↓		2^3	$= 1 \times 2^3$
						$= 1 \times 1 = 1$
						$= 0 \times 2 = 0$
						$= 1 \times 4 = 4$
						$= 1 \times 8 = 8$
						<hr/>
						13_{10}

Diagram showing bit weights and calculations. A red arrow points from the text "LSB" to the first bit (1). A blue arrow points from the text "MSB" to the last bit (1).



Bilangan Oktal

Bilangan *Oktal* ini merupakan sistem bilangan dengan basis 8 artinya bilangan yang memiliki bobot ke pangkat 8 (semisal 8^0 , 8^1 , 8^2 , 8^3 , ..., 8^n). Bilangan yang terdapat pada bilangan oktal adalah 0, 1, 2, 3, 4, 5, 6, dan 7.

Contoh : 572_8

5	7	2	(8)		
↓	↓	↓		8^0	$= 2 \times 8^0$
↓	↓	↓		8^1	$= 7 \times 8^1$
↓	↓	↓		8^2	$= 5 \times 8^2$
					$= 2 \times 1 = 2$
					$= 7 \times 8 = 56$
					$= 5 \times 36 = 320$
					<hr/>
					378_{10}

Diagram showing bit weights and calculations. A red arrow points from the text "LSD" to the first digit (2). A blue arrow points from the text "MSD" to the last digit (5).



Bilangan Heksadesimal

Bilangan *Heksadesimal* ini merupakan sistem bilangan dengan basis 16 artinya bilangan yang memiliki bobot kepangkatan 16 (semisal 16^0 , 16^1 , 16^2 , 16^3 , ... , 16^n). Bilangan yang terdapat pada bilangan biner adalah 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E dan F.

Contoh : $1A2_{16}$

$$\begin{array}{rcl}
 1A2_{(16)} & & \\
 \swarrow & \searrow & \swarrow \\
 16^0 & = 2 \times 16^0 & = 2 \times 1 = 2 \\
 16^1 & = 10 \times 16^1 & = 10 \times 16 = 160 \\
 16^2 & = 1 \times 16^2 & = 1 \times 256 = 256 \\
 & & \hline
 & & 418_{10}
 \end{array}$$

LSD

MSD



Konversi Bilangan *Desimal* ke bilangan lain.

Aturan umum :

Bagilah bilangan dengan basisnya, kemudian sisa hasil bagi diurutkan mulai dari yang terakhir.

Konversi bilangan *Desimal* ke *Biner*

Gunakan pembagian dengan 2 secara suksesif sampai sisanya = 0. Sisa-sisa pembagian membentuk jawaban, yaitu sisa yang pertama akan menjadi *least significant bit* (LSB) dan sisa yang terakhir menjadi *most significant bit* (MSB).

Contoh: Konversi 83_{10} ke biner:

$$\begin{array}{rcl}
 83 : 2 & = & 41 \text{ sisa } 1 \\
 41 : 2 & = & 20 \text{ sisa } 1 \\
 20 : 2 & = & 10 \text{ sisa } 0 \\
 10 : 2 & = & 5 \text{ sisa } 0 \\
 5 : 2 & = & 2 \text{ sisa } 1 \\
 2 : 2 & = & 1 \text{ sisa } 0 \\
 1 : 2 & = & 0 \text{ sisa } 1
 \end{array}$$

LSB

MSB

Maka $83_{10} = 1010011_2$

1 0 1 0 0 1 1



Konversi bilangan *Desimal* ke *Oktal*

Gunakan pembagian dengan 8 secara suksesif sampai sisanya = 0. Sisa-sisa pembagian membentuk jawaban, yaitu sisa yang pertama akan menjadi *least significant bit* (LSB) dan sisa yang terakhir menjadi *most significant bit* (MSB).

Contoh: Konversi 1059_{10} ke Oktal :

$$\begin{array}{l} 1059 : 8 = 132 \text{ sisa } 3 \\ 132 : 8 = 16 \text{ sisa } 4 \\ 16 : 8 = 2 \text{ sisa } 0 \\ 2 : 8 = 0 \text{ sisa } 2 \end{array}$$

2 0 4 3

LSB

Maka $1059_{10} = 2043_8$

MSB



Konversi bilangan *Desimal* ke *Heksadesimal*

Gunakan pembagian dengan 16 secara suksesif sampai sisanya = 0. Sisa-sisa pembagian membentuk jawaban, yaitu sisa yang pertama akan menjadi *least significant bit* (LSB) dan sisa yang terakhir menjadi *most significant bit* (MSB).

Contoh: Konversi 179_{10} ke Heksadesimal :

$$\begin{array}{l} 179 : 16 = 11 \text{ sisa } 6 \\ 11 : 16 = 0 \text{ sisa } 11 \end{array}$$

11 dalam bilangan
heksadesimal = B

B 6₁₆

LSB

Maka $179_{10} = B6_{16}$

MSB



Konversi Bilangan *Biner* ke bilangan lain.

Konversi bilangan *Biner* ke *Oktal*

Aturan umum :

Lakukan pengelompokkan menjadi 3 digit bilangan *biner*, dimulai dari digit terakhir atau posisi LSB sampai dengan MSB. kemudian konversikan ke basis 8.

Contoh: konversikan 10110011_2 ke bilangan oktal

Jawab : 10 110 011



2 6 3

Jadi $10110011_2 = 263_8$



Konversi bilangan *Biner* ke *Heksadesimal*

Aturan umum :

Lakukan pengelompokkan menjadi 4 digit bilangan biner, dimulai dari digit terakhir atau posisi LSB sampai dengan MSB. kemudian konversikan ke basis 16.

Contoh: konversikan 10110011_2 ke bilangan heksadesimal

Jawab : 1011 0011



B 3

Jadi $10110011_2 = B3_{16}$



Konversi Bilangan Oktal ke bilangan lain.

Konversi bilangan Oktal ke Biner

Aturan umum :

Konversi setiap digit bilangan oktal ke bilangan biner 3 digit, kemudian digabungkan.

Contoh Konversikan 263_8 ke bilangan biner.

Jawab: 2 6 3
 ↓ ↓ ↓
 010 110 011

Jadi $263_8 = 010110011_2$

Karena 0 didepan tidak ada artinya kita bisa menuliskan

$263_8 = 10110011_2$

17



Konversi bilangan Oktal ke Heksadesimal

Aturan umum :

Rubahlah bilangan oktal ke biner terlebih dahulu, kemudian biner tersebut dirubah ke heksadesimal..

Contoh Konversikan 345_8 ke bilangan Heksadesimal.

Jawab :

- ubah oktal ke biner

3 4 5
 ↓ ↓ ↓
 011 100 101

Jadi $345_8 = 11100101_2$

- Ubah biner ke heksadesimal

1110 0101
 ↓ ↓
 E 5

Jadi $345_8 = E5_{16}$

18



Konversi Bilangan *Heksa* ke bilangan lain. Konversi bilangan *Heksadesimal* ke *Biner*

Aturan umum :

Sebaliknya untuk mengkonversi bilangan hexadesimal ke biner yang harus dilakukan adalah terjemahkan setiap digit bilangan hexadesimal ke 4 digit bilangan biner.

Contoh Konversikan $B3_{16}$ ke bilangan biner.

Jawab : B 3
 ↓ ↓
 1011 0011

Jadi $B3_{16} = 10110011_2$



Konversi bilangan *Heksadesimal* ke *Oktal*

Aturan umum :

Rubahlah atau konverikan bilangan *heksa* ke biner terlebih dahulu, kemudian biner tersebut dirubah ke *oktal*.

Contoh Konversikan $F7_{16}$ ke bilangan Oktal.

Jawab :
- ubah heksa ke biner

F 7
 ↓ ↓
1111 0111

Jadi $F7_{16} = 11\ 110\ 111_2$

- Ubah biner ke Oktal

11 110 111
 ↓ ↓ ↓
 3 6 7

Jadi $F7_{16} = 367_8$



Tabel 6.1 Biner-Oktal-Desimal-Hexadesimal

Biner	Oktal	Desimal	Hexadesimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

C. Aritmatika Biner

Suatu bilangan selain dapat dikonversikan dari satu bentuk bilangan ke bilangan yang lainnya, namun bisa juga dilakukan bentuk perhitungan yaitu melakukan perhitungan biner dalam penjumlahan dan pengurangan.

Penjumlahan Biner

Aturan dasar penjumlahan:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0, \text{ simpan } 1$$

Contoh : 11001

$$\begin{array}{r} 11011 \\ + \\ \hline 110100 \end{array}$$

Pengurangan Biner

Aturan dasar pengurangan:

$$0 - 0 = 0$$

$$1 - 1 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1, \text{ pinjam } 1$$

Contoh : 1111

$$\begin{array}{r} 0101 \\ - \\ \hline 1010 \end{array}$$



D. Organisasi Data

Komputer secara umum bekerja dengan beberapa jumlah bit khusus. Kumpulan yang Umum adalah *bit* tunggal, kelompok 4 bit (disebut *nibbles*), kelompok 8 bit (disebut *byte*), kelompok 16 bit (disebut *word*), dan lain-lain

a. Bits

- "Unit" paling kecil dari data pada komputer biner adalah satu bit tunggal.
- satu bit tunggal mampu merepresentasikan hanya dua nilai yang berbeda (secara tipikal nol atau satu).
- Anda bisa merepresentasikan dua item data apapun yang berbeda dengan satu bit tunggal. Contoh meliputi nol atau satu, benar atau salah, on atau off, pria atau wanita. Anda tidak dibatasi untuk merepresentasikan jenis data biner (yaitu, objek yang hanya mempunyai dua nilai yang berbeda).



Data adalah apa yang anda ingin definisikan, maka.

- Jika anda menggunakan *bit* untuk merepresentasikan suatu nilai *boolean* (benar/salah) maka *bit* itu (oleh definisi anda) merepresentasikan benar atau salah.
- Agar *bit* mempunyai maksud/arti yang benar, anda harus konsisten. Maka, jika anda sedang menggunakan *bit* untuk merepresentasikan benar atau salah di dalam program anda, anda tidak boleh menggunakan nilai benar/salah yang disimpan dalam bit tsb untuk merepresentasikan merah atau biru.



b. Nibbles.

- *nibble* adalah satu koleksi empat *bit*. Ia bukan merupakan jenis data yang menarik kecuali dua item: bilangan BCD (*binary coded decimal*) dan bilangan berbasis enam belas.
- Ia menggunakan empat *bit* untuk merepresentasikan satu BCD tunggal atau digit hexadecimal. Dengan suatu *nibble*, kita bisa merepresentasikan sampai dengan 16 nilai berbeda.
- Dalam kasus bilangan berbasis enambelas, nilai dapat berupa 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F direpresentasikan dengan empat bit. BCD menggunakan sepuluh angka berbeda (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

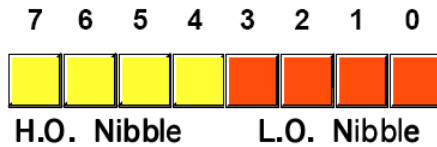


c. Byte.

- Struktur data terpenting yang digunakan oleh *mikroprosesor* 80x86 adalah *byte*. Sebuah *byte* terdiri dari delapan *bit* dan adalah *datum addressable* paling kecil (data item) pada *mikroprosesor* 80x86.
- Memori Utama dan alamat I/O pada 80x86 adalah semua alamat *byte*. Artinya bahwa item paling kecil yang mungkin diakses secara individu oleh satu program 80x86 adalah nilai delapan-bit.
- Bit dalam satu *byte* secara normal dinomori dari nol sampai tujuh menggunakan konvensi di dalam gambar 1.
- Bit 0 adalah urutan bit terendah atau bit paling tidak berarti (signifikan), bit 7 adalah urutan bit paling berarti (signifikan) dari byte. Kita akan mengacu pada penomoran semua bit lain.

Gambar 1. Penomoran bits dalam 1 byte



Gambar 2. Dalam satu *byte* berisi dua *nibble***d. Word.**

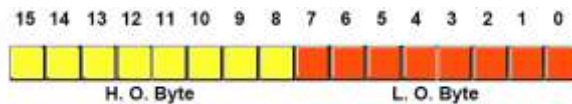
- Sebuah *word* adalah kelompok 16 *bit*. Kita akan menomori *bit* dalam *word* mulai dari nol sampai dengan lima belas. Penomoran *bit* muncul di gambar 3.

Gambar 3. Nomor *bit* dalam *word*

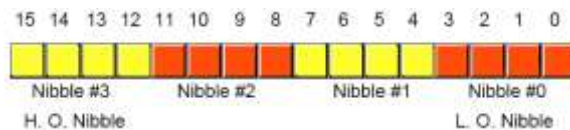
- Seperti *byte*, bit 0 adalah urutan *bit* terendah dan bit 15 adalah urutan *bit* tertinggi.



- Perhatikan bahwa satu *word* berisi persis dua *byte*. Bit 0 sampai 7 membentuk urutan *byte* terendah, bit 8 hingga 15 membentuk urutan *byte* tertinggi (gambar 4).

Gambar 4. Dua *Byte* dalam *word*

- Secara alami, satu *word* mungkin saja dipecah ke dalam empat *nibble* seperti diperlihatkan di dalam gambar 5.

Gambar 5. Nibble dalam sebuah *word*

- Nibble nol adalah nibble urutan terendah dalam word dan nibble tiga adalah nibble urutan tertinggi dari word. Dua nibble lain adalah “nibble satu” atau “nibble dua”.
- Dengan 16 bit, anda bisa merepresentasikan 2¹⁶ (65,536) nilai yang berbeda. Ini bisa menjadi nilai dalam jangkauan 0..65,535 (atau, sebagai kasus biasanya, -32,768..+32,767) atau jenis data lain apapun tanpa lebih dari 65,536 nilai



TERIMA KASIH

Topik Perkuliahan