MAGISTER MANAJEMEN TEKNOLOGI

Sistem Informasi dan Teknologi

Chapter 5
REPRESENTASI DATA

REPRESENTASI DATA

- Unit Informasi Dasar dalam sistem komputer- 1 byte atau 8 bit.
- Word size (ukuran word) – merupakan ukuran register operasionalnya.
- Contoh :

1. Komputer 16-bit mempunyai register 16-bit – satu word terdiri dari 2 byte
2. Komputer 32-bit mempunyai register 32-bit – satu word terdiri dari 4 byte.

REPRESENTASI KARAKTER

Representasi karakter yang paling dikenal adalah :

1. Pada PC dan minikomputer adalah American Standard Code for Information Interchange (ASCII) : satu byte satu karakter.2.
2. Sedangkan pada mainframe IBM menggunakan Extended Binary Coded Decimal Interchange Code (EBCDIC).

REPRESENTASI FLOATING POINT

Representasi *Floating-Point* terdiri dari empat bagian:

- Sign (S)
- Mantissa atau koefisien (M)
- Radix atau base eksponen (R)
- Eksponen (E)

FORMAT FLOATING-POINT (IEEE) Ada 2 :

- Single Precision (presisi tunggal) – 32 bit terdiri dari : 1 bit sign, 8 bit eksponen, dan 23 bit mantissa.
- Double Precision (presisi ganda) – 64 bit terdiri dari: 1 bit sign, 11 bit eksponen, dan 52 bit mantissa.

REPRESENTASI DATA NUMERIK

Representasi data numerik yang biasa digunakan untuk bilangan bulat dan pecahan (integer dan fraction):

- Unsigned-binary numbers (bil. Positif dan Nol)
- Binary-Coded Decimals (BCD)
- Signed-magnitude Integers
- Ones-Complement Integers
- Twos-Complement Integers
- Excess-n
- Fraction (bilangan pecahan)

REPRESENTASI FIXED POINT

Radiks point/binary point tetap dan diasumsikan akan berada di sebelah kanan dari digit yang paling kanan.

1. Representasi Sign-Magnitude/Nilai tanda

- Untuk merepresentasikan bilangan integer negatif dan positif. Dengan menggunakan MSB sebagai bit tanda →0 = positif, 1 = negatif
- Contoh :
- Sign-Magnitude +9 dalam 8 bit = 00001001
- Sign-Magnitude –4 dalam 4 bit = 1100
- Magnitude dari bilangan positif dan negatif sama yang membedakan hanya MSB saja

2. Representasi Komplemen-1

Untuk mendapat komplemen-1 maka bilangan 0 menjadi 1 dan 1 menjadi 0.

REPRESENTASI FIXED POINT

3. Representasi Komplemen-2

Langkah-langkah Pengubahan bilangan desimal bertanda ke bilangan komplemen (8-bit)

- Tentukan bit tanda/MSB → 0 = positif, 1 = negatif.
- Ubah desimal ke biner (7-bit)
- Ubah ke kompl-1 (setiap 0 diubah ke 1 dan
- setiap 1 diubah ke 0)
- Ubah ke komplemen-2 (tambahkan +1 ke komplemen-1 untuk mendapat bil. komplemen-2)
- Gabung menjadi satu yaitu MSB sebagai tanda bit dan 7-bit sebagai besarannya

Langkah-langkah Pengubahan bil. kompl-2 (8-bit) ke bil. Desimal bertanda :

- Tentukan bit tanda/MSB
- Ubah 7-bit kompl-2 tersebut ke kompl-1
- Ditambah +1 ke kompl-1
- Ubah biner ke desimal

REPRESENTASI FLOATING-POINT

- Menyatakan suatu bilangan yang sangat besar/sangat kecil dengan menggeser titik desimal secara dinamis ke tempat yang sesuai dan menggunakan eksponen 10 untuk menjaga titik desimal itu.
- Sehingga range bilangan yang sangat besar dan sangat kecil untuk direpresentasikan hanya dengan beberapa digit saja.
- Dinyatakan dengan notasi → *a = (m,e)* ,

dimana :

$a = m \times r^{e}$

r = radiks

m = mantissa

e = eksponen

Contoh : Tunjukkan bilangan-bilangan berikut ini dalam notasi floating point.

a. $(45.382)10 \rightarrow 0.45382 \times 10^{2} = (0.45382,2)$

b. $(-21,35)8 \rightarrow -2135,0 \times 8^{-2} = (-2135.0,-2)$

KODE BINER

1. Kode Biner yg berbobot →BCD (Binary coded Decimal)

- Kode BCD 8421 artinya MSB = Most Significant Bit mempunyai bobot 8, sedang LSB = Least Significant Bit mempunyai bobot 1.
- Konversi BCD ke sistem bilangan basis yang lain :

  BCD ke basis X → ubah BCD ke Desimal kemudian ubah Desimal ke basis X.

- Contoh :

a. 000101011 . 00100101BCD = …… $_2$

b. 0 0010 1001 . 0010 0101 = 29,2510 =11101, 01

   0 2 9 , 2 5

KODE BINER

2. Kode Biner yang tidak berbobot

a. Kode Excess-3 → kode yang tiga angka lebih besar dari BCD 8421.

Contoh : 6210 = …….xs3

Caranya :

Tambah desimal 3 di setiap digit desimalnya

Ubah desimal tersebut ke BCD

6 2

3 3 +

9 5 → 1001 0101(xs3)

Sistem Bilangan

- Secara umum dalam sistim mikroprosesor sistim bilangan yang digunakan ada empat jenis yaitu:
  1. Sistim Bilangan Desimal
  2. Sistim Bilangan Biner
  3. Sistim Bilangan Heksadesimal
  4. Sistim Bilangan Oktal
- Keempat sistim bilangan ini satu sama lain dibedakan oleh sebuah nilai yang disebut dengan BASIS.
  - ✓ Sistim bilangan desimal menggunakan basis 10,
  - ✓ Biner menggunakan basis 2
  - ✓ Heksa-desimal menggunakan basis 16
  - ✓ Oktal menggunakan basis 8.

Bilangan Desimal

- Bilangan desimal adalah bilangan berbasis sepuluh.
- Dalam desimal dikenal sepuluh simbol bilangan yaitu; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Nilai sebuah angka ditentukan oleh posisi angka tersebut. Dalam sistim desimal dikenal nilai posisi:

  $10^0 = 1$ = satuan

  $10^1 = 10$ = puluhan

  $l0^2 = 100$ = ratusan

  $l0^3 = 1000$ = ribuan

  $10^4 = 10000$ = puluhan ribu

  $l0^5 = 100000$ = ratusan ribu

  dan seterusnya berdasarkan nilai basis dan pangkat .

Contoh:

$1011 = 1 \text{ X } 10^3 + 0 \text{ x } 10^2 + 1\text{x } 10^1 + 1\text{x } 10^0$

$= 1000 + 0 + 10 + 1$ (dibaca seribu sebelas)

Bilangan Biner

- Bilangan biner adalah bilangan berbasis dua.
- Dalam biner dikenal dua simbol bilangan yaitu; 0, 1.
- Nilai sebuah angka ditentukan oleh posisi angka tersebut.
- Dalam sistim biner dikenal nilai posisi :

$2^0 = 1$ = satuan
$2^1 = 2$ = duaan
$2^2 = 4$ = empatan
$2^3 = 8$ = delapanan
$2^4 = 16$ = enam-belasan
$2^5 = 32$ = tiga-puluh-duaan
$2^6 = 64$ = enam-puluh-empatan
$2^7 = 128$ = seratus-dua-puluh-delapanan
dan seterusnya berdasarkan nilai basis dan pang kat

Contoh:

$1011_2 = 1x2^3+0x2^2+1x2^1+1x2^0$
$= 8+0+2+1$
$= 11$
jadi nilai bilangan $1011_2 = 11_{10}$

Bilangan Heksa Desimal

- Bilangan heksa-desimal adalah bilangan berbasis enambelas.
- Dalam heksa-desimal dikenal enambelas simbol bilangan yaitu; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,A, 8, C, D, E, F. Dimana A = 10; B = 11; C = 12; D = 13; E= 14; don F = 15.
- Nilaisebuah angka ditentukan oleh posisi angka tersebut.
- Dalam sistim Heksa-desimal dikenal nilai posisi :

  $16^0 = 1$ = satuan

  $16^1 = 16$ = enam-belasan

  $16^2 = 256$ = dua-ratus-lima-puluh-enaman

  $16^3 = 4096$ = empat-ribu-sembilan-puluh-enaman

  dan seterusnya berdasarkan nilai basis dan pangkat

Contoh :

$1011_{16} = 1 \times 16^3 + 0 \times 16^2 + 1x\ 16^1 + 1x\ 16^0$

$= 4096 + 0 + 16 + 1$

$= 4113$

Jadi nilai bilangan $1011_{16} = 4113_{10}$

Bilangan Oktal

- Bilangan oktal adalah bilangan berbasis delapan.
- Dalam oktal dikenal delapan simbol bilangan yaitu; 0, 1, 2, 3, 4, 5, 6, 7.
- Nilai sebuah angka ditentukan oleh posisi angka tersebut.
- Dalam sistim Oktal dikenal nilai posisi :

  $8^0 = 1$ = satuan

  $8^1 = 8$ = delapanan

  $8^2 = 611$ =enam-puluh-empatan

  $8^3 = 512$ =lima-ratus-dua-belasan

  dan seterusnya berdasarkan nilai basis dan pangkat

Contoh:

$1011_8 = 1 \text{ X } 8^3 + 0 \text{ x } 8^2 + 1\text{x } 8^1 + 1\text{x } 8^0$

$= 512 + 0 + 8 + 1$

$= 521$

Jadi nilai bilangan $1011_8 = 521_{10}$

Konversi Bilangan

- Sebuah bilangan dapat dinyatakan dalam empat penyajian angka atau simbol berbeda.
- Untuk mendapatkan nilai suatu bilangan atau padanan suatu bilangan dalam satu basis ke basis lainnya digunakan cara konversi bilangan.
- Ada dua teknik konversi yaitu :
  1. Teknik bagi
  2. Teknik kurang

Konversi Bilangan Desimal ke Biner

- Contoh: 44_{10}=……$_2$?
- Dengan teknik bagi dua:

44 ÷ 2 = 22	sisa: 0	LSB
22 ÷ 2 = 11	sisa: 0	
11 ÷ 2 = 5	sisa: 1	
5 ÷ 2 = 2	sisa: 1	
2 ÷ 2 = 1	sisa: 0	
1 ÷ 2 = 0	sisa: 1	MSB

Jadi $44_{10} = 101100_2$

Konversi Bilangan Desimal ke Biner

- Contoh: 44_{10}=……$_2$?
- Dengan teknik pengurangan:

44 - 128 = K	bit: 0	MSB
44 - 64 = K	bit: 0	
44 - 32 = 12	bit: 1	
12 - 16 = K	bit: 0	
12 - 8 = 4	bit: 1	
4 - 4 = 0	bit: 1	
0 - 2 = 0	bit: 0	
0 - 1 = 0	bit: 0	LSB

Jadi $44_{10} = 00101100_2$

Catatan:

Jika bilangan yang dikurangkan nilainya lebih kecil dari bilangan pengurang maka nilai bit sama dengan 0 (nol).

Jika bilangan yang dikurangkan nilainya lebih besar dari bilangan pengurang maka nilai bit sama dengan 1(satu).

Konversi Bilangan Desimal ke Heksa-Desimal

- Contoh: 44_{10} = ……$_{16}$?
- Dengan teknik bagi 16

  44 ÷ 16 = 2 sisa : 12

  $12_{10} = C_{16}$

  Jadi $44_{10} = 2C_{16}$

Konversi Bilangan Desimal ke Oktal

- Contoh: 44_{10} = ……..$_8$?
- Dengan teknik bagi 8

  44 ÷ 8 = 5 sisa : 4

  4_{10} =4_8 LSB

  Jadi 44_{10} = 54_8

Konversi Bilangan Biner ke Heksa-Desimal dan Oktal

- Konversi bilangan Biner ke Heksa-Desimal menggunakan satuan 4 bit
- Sedangkan konversi bilangan Biner ke Oktal menggunakan satuan 3 bit.

Tabel Dasar Konversi Bilangan

Des	Bin	Hex	Okt
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10

Des	Bin	Hex	Okt
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	1 0000	10	20
17	1 0001	11	21

BILANGAN BINER TAK BERTANDA 8 BIT

- Bilangan biner tak bertanda 8 bit dapat menyajikan bilangan sebanyak 256 nilai dari 0 sampai dengan 255.
- Berdasarkan satuan dan proses konversi maka dapat disusun tabel konversi desimal ke biner dan Heksa-Desimal sebagai berikut:

Des	Biner	Hex
0	0000 0000	00
1	0000 0001	01
…	.....	…
…	.....	…
127	0111 1111	7F
128	1000 0000	80
129	1000 0001	81
…	.....	…
…	.....	…
254	1111 1110	FE
255	1111 1111	FF

BILANGAN BINER TAK BERTANDA 8 BIT

- Dari tabel dapat dibuat garis bilangan dengan bilangan terkecil 00000000 =0_{10}= 00_{16} dan bilangan terbesar 11111111 = 255_{10} = FF_{I6}

Penjumlahan dan Pengurangan Biner

- Dalam penjumlahan bilangan biner berlaku kaidah sebagai berikut :

Carry In	B	A	ADC=A+B+Cin	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Penjumlahan dan Pengurangan Biner

- Penjumlahan bilangan biner dimulai dari bit LSB menuju bit MSB.
- Contoh:

Desimal		Biner	Hexa-Desimal
	Carry:	0110 0010	
A = 53		0011 0101	35
B = 25	+	0001 1001	19
78		0100 1110	4D

Penjumlahan dan Pengurangan Biner

- Penjumlahan bilangan biner dimulai dari bit LSB menuju bit MSB.
- Contoh:

Desimal	Biner	Hexa-Desimal
	Carry: 0000 0000	
A = 129	1000 0001	81
+ B = 138	1000 1010	8A
267	1 0000 1011	1 0B

Penjumlahan dan Pengurangan Biner

- Dalam pengurangan bilangan biner berlaku kaidah sebagai berikut :

Borrow In	B	A	SUB=A-B-Bin	Borrow Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

Penjumlahan dan Pengurangan Biner

- Pengurangan bilangan biner dimulai dari bit LSB menuju bit MSB.
- Contoh:

Desimal	Biner	Hexa-Desimal
	Borrow: 0011 0000	
A = 53	0011 0101	35
B = 25	0001 1001	19
28	0001 1100	1C

Penjumlahan dan Pengurangan Biner

- Pengurangan bilangan biner dimulai dari bit LSB menuju bit MSB.
- Contoh:

Desimal	Biner	Hexa-Desimal
	Borrow: 0000 0000	
A = 129	1000 0001	81
B = 128	1000 0000	80
1	0000 0001	1

Pengurangan dengan Metoda Komplemen

- Pengurangan suatu bilangan dapat dilakukan dengan penjumlahan bilangan tersebut dengan komplemen bilangan pengurangnya (A-B) = A+(-B).
- Dalam desimal dikenal istilah komplemen 9 dan komplemen 10.
- Dalam biner dikenal komplemen 1 dan komplemen 2.

Pengurangan dengan Metoda Komplemen

Desimal	
Bilangan	Komplemen 9
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

Biner	
Bilangan	Komplemen 1
0	1
1	0

- Disamping komplemen 9 dalam desimal dikenal komplemen 10 yaitu komplemen 9 + 1.
- Sedangkan dalam biner dikenal komplemen 2 yaitu komplemen 1+1.

Pengurangan dengan Metoda Komplemen

- Contoh Pengurangan dengan Komplemen:

Desimal

Konvensional	Komplemen 9	Komplemen 10
67 24 **43**	67 75 + 142 1 **43**	67 76 + 1**43**

Catatan : Pada komplemen 10, Jika carry = 1 hasil positif
Jika carry = 0 hasil negatif

Pengurangan dengan Metoda Komplemen

- Contoh Pengurangan dengan Komplemen:

Biner

Konvensional	Komplemen 1	Komplemen 2
0100 0011 0001 1000 - 0010 1011	0100 0011 1110 0111 + 1 0010 1010 1 0010 1011	0100 0011 1110 1000 + 0010 1011

Bilangan Biner Bertanda 8 Bit

- Dalam operasi aritmetika sering diperlukan juga penyajian bilangan dengan tanda positif dan negatif.
- Bilangan semacam ini disebut bilangan bertanda.
- Untuk menyajikan tanda suatu bilangan biner apakah positif atau negatif digunakan satu bit data yaitu bit MSB atau b7 untuk data 8 bit.
- Jika b7 = 1 menandakan bilangan tersebut negatif (-) sedangkan jika b7 = 0 menunjukkan bilangan tersebut positif (+).

Bilangan Biner Bertanda 8 Bit

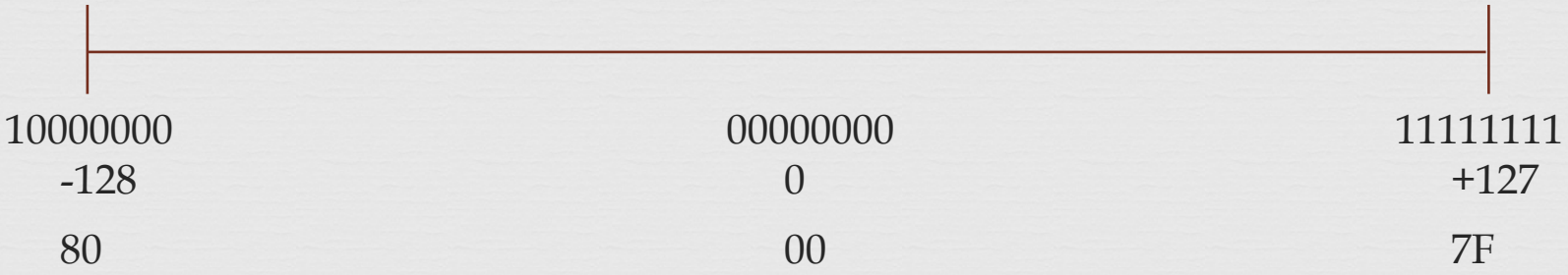
- Tabel Bilangan Biner Bertanda

Desimal Positif	Biner Bertanda
+1	0 000 0001
+2	0 000 0010
+3	0 000 0011
+4	0 000 0100
….	……..
+10	0 000 1010
….	……..
+126	0 111 1110
+127	0 111 1111

Desimal Negatif	Biner Bertanda
-1	1 111 1111
-2	1 111 1110
-3	1 111 1101
-4	1 111 1100
….	……..
-10	1 111 0110
….	……..
-126	1 000 0010
-127	1 000 0001

BILANGAN BINER BERTANDA 8 BIT

- Dari tabel dapat dibuat garis bilangan dengan bilangan terkecil 10000000 = -128_{10}= 80_{16} dan bilangan terbesar 01111111 = $+127_{10}$ = $7F_{I6}$



Bilangan Biner Bertanda 8 Bit

- Penyajian bilangan biner bertanda dengan menggunakan tanda bilangan pada bit b7 belum memenuhi kebutuhan pengolahan data dalam operasi aritmetika.
- Dua contoh berikut sebagai bukti:

Desimal	Biner
+3 + -2 +1	0 000 0011 1 000 0010 1 000 0101 = - 5
Desimal	Biner
+4 + -5 -1	0 000 0100 1 000 0101 1 000 1001 = - 9

Bilangan Biner Bertanda 8 Bit

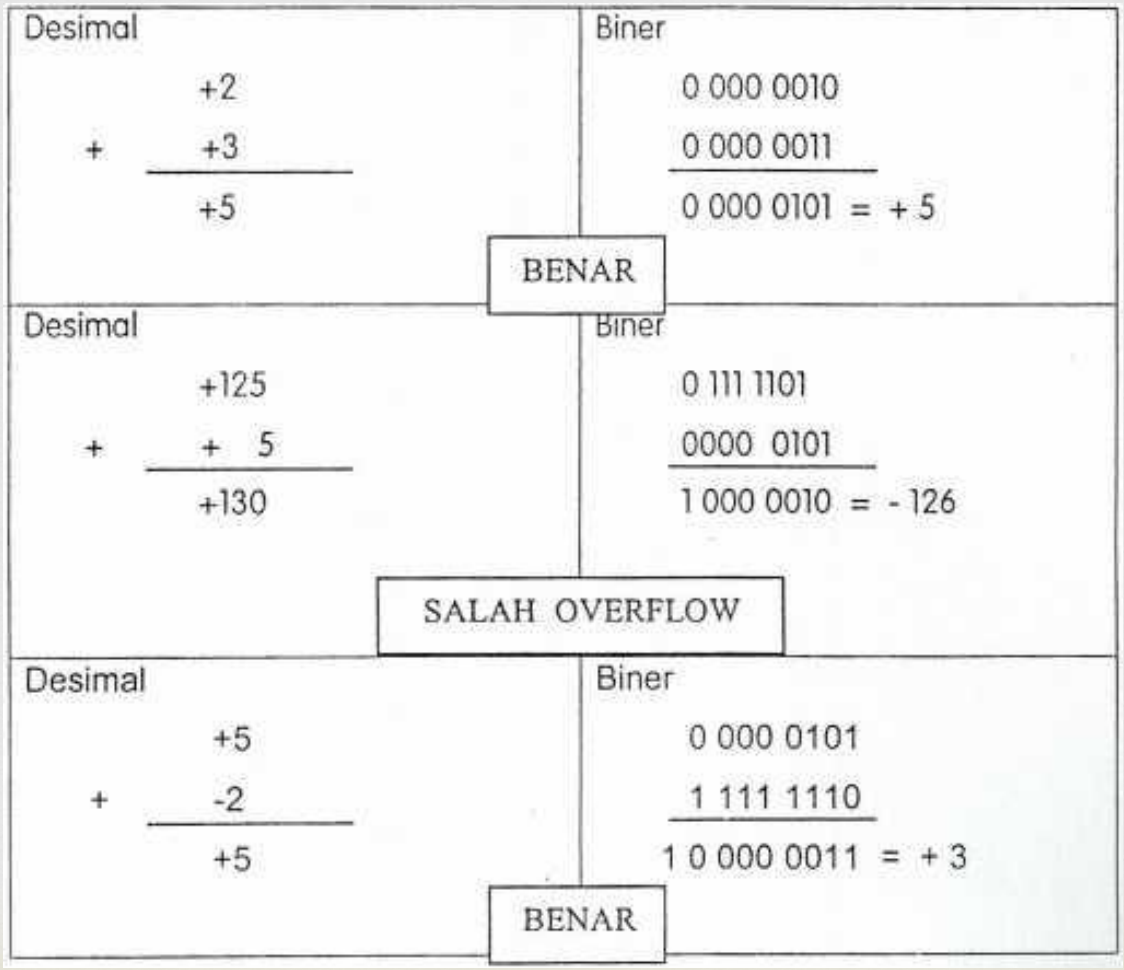
- Dari dua contoh terbukti hasil penjumlahannya salah.
- Untuk itu perlu dicari jalan keluarnya. Karena komputer tidak hanya untuk menyajikan informasi tetapi juga untuk melakukan pengolahan data seperti operasi aritmetika.
- Jalan keluar yang dapat digunakan adalah dengan menggunakan operasi bilangan bertanda komplemen dua.

Bilangan Biner Bertanda 8 Bit

- Dengan menggunakan penyajian bilangan biner komplemen 2 didapat hasil operasi aritmatika yang benar.
- Untuk kode 8 bit sebagaimana terlihat pada garis bilangan kemampuan operasinya dibatasi diantara -128 sampai dengan +127.
- Operasi aritmetika diatas atau lebih besar dari +127 dan di bawah atau lebih kecil dari -128 akan mengakibat kesalahan yang disebut dengan Kesalahan Overflow.

Bilangan Biner Bertanda 8 Bit

- Contoh:



Bilangan Biner Bertanda 8 Bit

- Contoh:

Desimal	Biner
-5	1 111 1011
+ -2	1 111 1110
+5	1 1 111 1001 = -7

BENAR

Desimal	Biner
-128	1 000 0000
+ - 1	1 111 1111
-129	1 0 111 1111 = + 127

SALAH OVERFLOW

PENYAJIAN DESIMAL TERKODE BINER (DTB)

- Untuk mengkodekan bilangan desimal dari 0 sampai dengan 9 dalam format biner diperlukan empat angka biner (I nible).
- Empat angka biner membentuk $2^4 = 16$ kemungkinan. Karena angka desimal hanya membutuhkan 10 kode angka maka ada 6 kode yang tidak digunakan dalam penyajian DTB.
- Hal ini akan memungkinkan timbulnya permasalahan dalam operasi aritmetika.

PENYAJIAN DESIMAL TERKODE BINER (DTB)

- Penyajian DTB hanya memerlukan 1 nible, maka untuk data 1 byte dapat memuat 2 angka desimal.
- Contoh:

DTB	**Desimal**
0000 0000	= 00
0010 0000	= 20
1001 1001	= 99

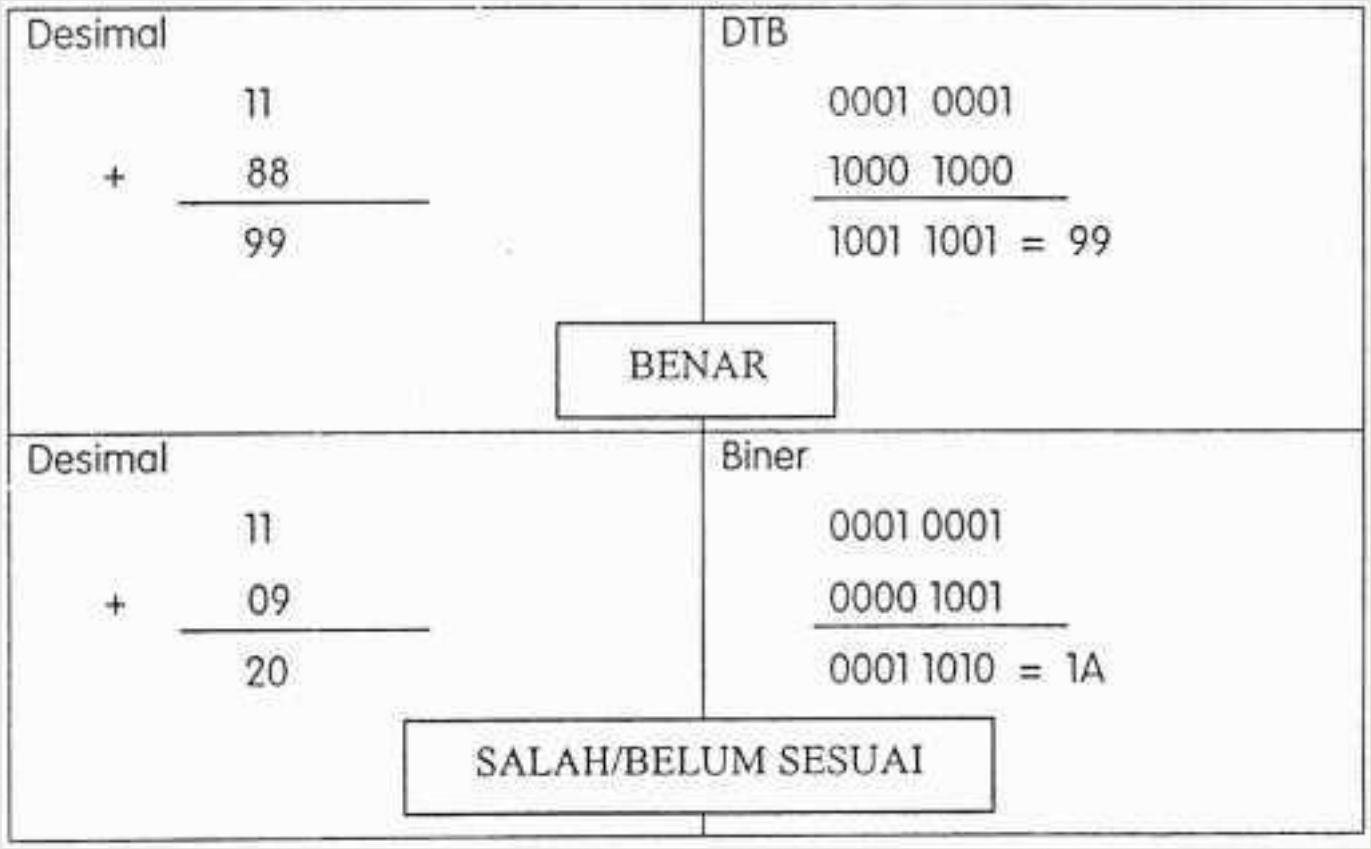
DTB	Desimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	not used
1011	not used
1100	not used
1101	not used
1110	not used
1111	not used

Penjumlahan DTB

- Penjumlahan DTB menggunakan kaidah yang sama dengan kaidah penjumlahan biner.
- Hasil penjumlahan DTB dalam tiap kode lebih kecil dari 10 adalah benar, sedangkan hasil penjumlahan lebih besar dari 9 masih perlu dikoreksi.

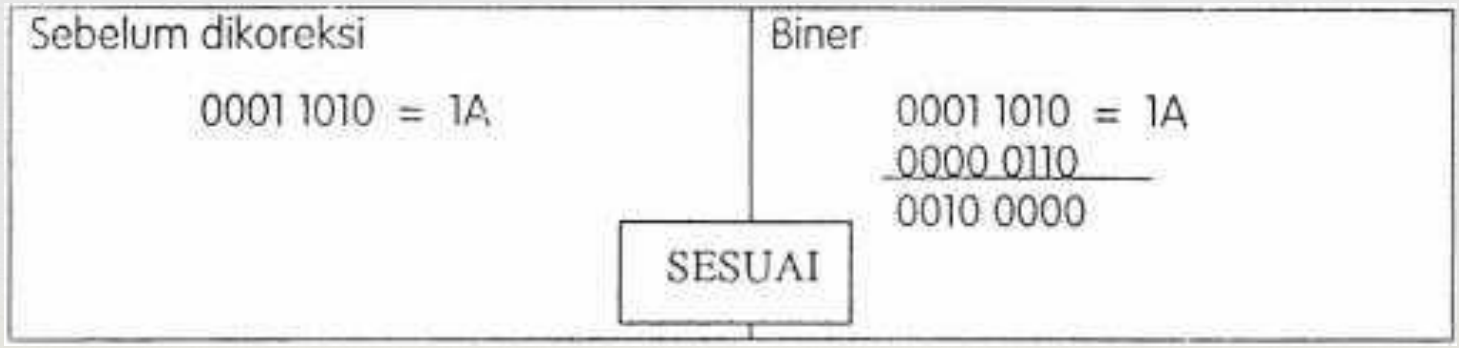
Penjumlahan DTB

- Contoh:



Penjumlahan DTB

- Kesalahan semacam contoh yang kedua dikoreksi dengan perintah Decimal Addjust for Addition = DAA dengan cara :
  - ✓ Jika bit b3,b2,bl,b0 > 9 atau ada carry dari b3 ke b4 nibble rendah ditambahkan 0110
  - ✓ Jika b7,b6,b5,b4,b3,b2,b1,b0 > 9F atau ada carry dari b7 keluar maka ditambahkan dengan 0110 0000.



END