BAB. II

SISTIM BILANGAN

Suatu sistim biner bekerja berdasarkan hukum-hukum Aljabar Penyambungan.

Aljabar Penyambungan adalah sebenarnya sama dengan Aljabar Boole.

Aljabar Penyambungan adalah Aljabar Boole dua elemen yaitu "0" dan "1" dengan tiga operasi yaitu Operasi AND (perkalian Boole), operasi OR (penjumlahan Boole) dan operasi NOT (komplemen).

Pada Aljabar Penyambungan, besaran yang abstrak dari Aljabar Boole diterjemahkan ke dalam besaran-besaran yang dapat diwujudkan secara praktis seperti kontak-kontak dan gerbang elektronika.

Sistim biner atau dual, sistim yang ditemukan pada abad 17 oleh Leibnitz hanya menggunakan dua angka yaitu nol dan satu.

Pada suatu bilangan desimal tiap-tiap angka merepresentasi - kelipatan dari pangkat sepuluh.

Karena itu bilangan desimal disebut bilangan yang berdasar - kan atau berradiks sepuluh.

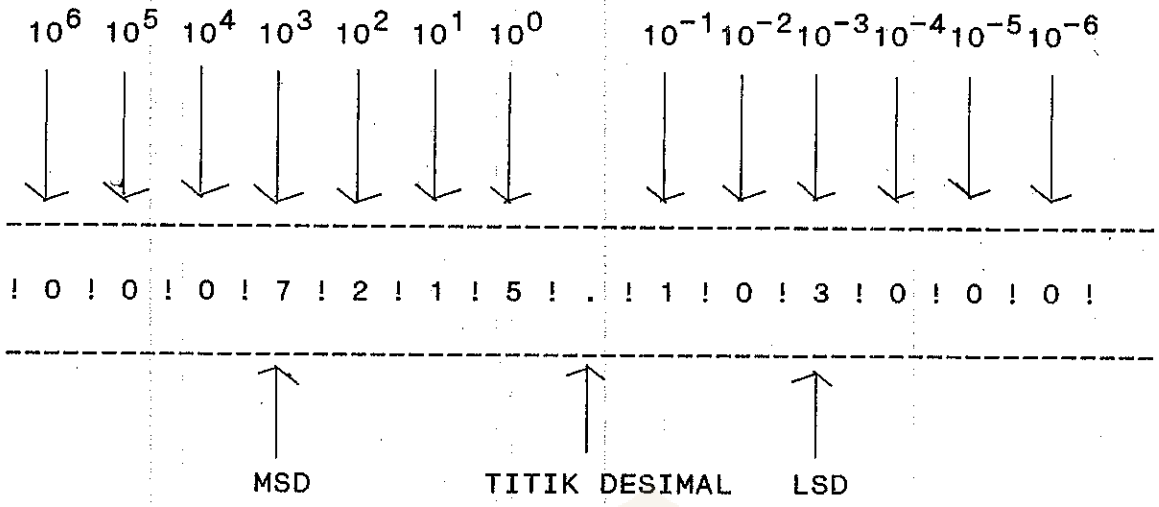
Kesepuluh simbol adalah 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, dengan menggunakan simbol-simbol ini memungkinkan untuk menyatakan setiap kwantitas.

Sistim desimal adalah suatu sistim nilai posisional (*Positional Value System*), didalam sistim ini nilai suatu digit bergantung pada posisinya.

Misalnya :

$$7215,103 = (7*10^{3})+(2*10^{2})+(1*10^{1})+(5*10^{0})+(1*10^{-1}) +(0*10^{-2})+(3*10^{-3})$$

Hal ini bisa ditunjukkan pada gambar :



7 merupakan digit paling berbobot diantara digit yang lain disebut juga digit bobot terbesar atau *Most Significant Digit (MSD)*

Digit 3 bobotnya paling kecil disebut digit bobot terkecil- atau *Least Significant Digit (LSD).*

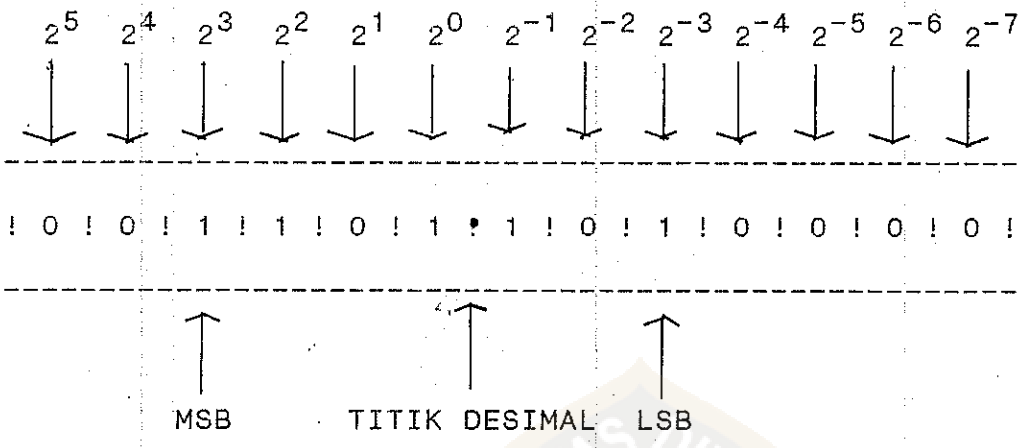
Dengan demikian setiap bilangan sebenarnya merupakan jumlah hasil kali setiap nilai digit dikalikan nilai posisionalnya.

Telah disebutkan di atas bahwa dalam Sistim Biner hanya ada dua simbol atau nilai digit yang mungkin yaitu 0 dan 1.

1 dan 0 yang membentuk bilangan biner disebut "bit" sebagai singkatan dari *"binary digits"*. Bit yang terletak pa - ling kanan disebut bit berbobot rendah atau *least signi - ficant bit (LSB)*, sedangkan bit paling kiri merupakan bit berbobot terbesar atau *most significant bit (MSB).*

Contoh :

$$1101,101 = (1*2^3)+(1*2^2)+(0*2^1)+(1*2^0)+(1*2^{-1})+(1*2^{-2})+(0*2^{-3})$$



Pada gambar di atas ditunjukkan bahwa ada 4 bit ke kiri titik biner, menyatakan bagian bilangan yang bulat dan 3 bit ke kanan titik biner menyatakan bagian Pecahan Posisi Bit yang paling *significant* (MSB) adalah bit paling kiri (bobot terbesar) dan posisi bit yang paling kurang *significant* (LSB) adalah bit yang paling kanan (bobot terkecil).

Contoh :

1. Perubahan dari desimal ke biner

```
120 : 2 = 60 sisa 0 ---------------------------
 60 : 2 = 30 sisa 0 ----------------------     ¦
 30 : 2 = 15 sisa 0 -----------------     ¦    ¦
 15 : 2 =  7 sisa 1 ------------     ¦    ¦    ¦
  7 : 2 =  3 sisa 1 -------     ↓    ↓    ↓    ↓
  3 : 2 =  1 sisa 1 --↓    ↓
  1 : 2 =  0 sisa 1   1    1    1    0    0    0
```

Bilangan biner yang sesuai dengan desimal 120 adalah : 1111000

2. Perubahan biner ke desimal

Tuliskan bilangan desimal dari bilangan biner berikut : 11010101

Desimal	128	64	32	16	8	4	2	1
Bobot dalam pangkat	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bilangan biner	1	1	0	1	0	1	0	1

Bilangan desimal = 128 + 64 + 16 + 4 + 1 = 213

2.1. Penjumlahan Biner

Menjumlah merupakan operasi terpentingdari semua oto - mat penghitung.

Pada penjumlahan bilangan desimal terlebih dahulu dilakukan penjumlahan lajur satuan. Jika hasil penjum-lahannya lebih kecil dari sepuluh, maka hasilnya dituliskan langsung di bawah lajur satuan. Jika hasilnya lebih besar dari sepuluh, maka satuannya dituliskan di bawah lajur satuan. Sedangkan puluhan dialihkan ke lajur puluhan. Angka-angka di lajur puluhan selanjutnya dijumlahkan bersama-sama dengan pindahan yang dialihkan tadi. Jika hasil penjumlahan lajur puluhan ini lebih kecil dari sepuluh, maka hasilnya langsung dituliskan di bawah lajur puluhan, jika hasilnya lebih besar dari sepuluh, maka hasil satuannya dituliskan di lajur puluhan, sedangkan puluhannya dipindahkan ke lajur ratusan dan seterusnya.

Contoh :

```
        11 <----- Pindahan
        7284
        4951
       ------ +
       12235
```

Pada penjumlahan biner aturannya jauh lebih sederhana, sebagai contoh yaitu :

0 + 0 = 0

1 + 0 = 1

0 + 1 = 1

1 + 1 = 0 + 1
        ¦
        ¦---->merupakan pindahan yang dialih-
              kan ke bit biner berikutnya
              (yang lebih tinggi).

Tabel 2.1. Penjumlahan Biner

B	A	Hasil = C	Pindahan = D
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Output akan 1, hanya bila kedua *input* berbeda.

Contoh :

```
1.  25              11001
    17              10001
   ------           1  1  <--- pindahan
    42            ---------- +
                    101010
```

```
2.  250             11111010
    110             00110111
   -----            111111  <--- pindahan
    360          -------------- +
                   100110001
```

Berikut ini adalah aturan untuk penjumlahan lajur 2'an dan lajur-lajur sebelah kiri berikutnya.

1. Jika ketiga bit (dua dari bilangan semula dan pindahan adalah 0, maka hasil lajur tersebut juga 0 dan pindahan ke lajur berikut juga 0.
2. Jika hanya satu bit yang satu, maka hasil menjadi 1 dan pindahan 0.
3. Jika dua bit yang satu, maka hasil menjadi 0 dan pindahan 1.
4. Jika tiga bit yang satu, maka hasil menjadi 1 dan pindahan 1.

2.2. Pengurangan Biner

Seperti pada penjumlahan, pengurangan dalam berbagai sistim bilangan dapat dilakukan sebagaimana dalam bilangan desimal.

Bila digit pengurangan lebih besar daripada digit yang akan dikurangi, maka harus "pinjam" pada digit di sebelah kirinya yang mempunyai bobot lebih besar.

Contoh :

```
  726
  378
------- -
  348
```

Pengurangan digit-digit bilangan biner dapat dilakukan seperti pengurangan pada sistim desimal.

Pengurangan masing-masing digit mempunyai empat kemungkinan ditunjukkan pada Tabel 2.2.

Tabel 2.2. Pengurangan Bilangan Biner

0	-	0	=	0
1	-	0	=	1
1	-	1	=	0
0	-	1	=	1 , pinjam 1

Contoh :

	Biner :	Desimal :
a.	100101	37
	000101	5
	----------	------ -
	100000	32
b.	11000011	195
	00111101	125
	----------	------- -
	1000110	70

2.3. Perkalian Biner

Pada perkalian biner jauh lebih sederhana dibandingkan dengan perkalian bilangan desimal.

Untuk mengalikan dua bilangan desimal berganda ditem - puh prosedur berikut :

1. Multiplikan (bilangan yang dikalikan) terlebih dahulu dikalikan dengan digit satuan multiplikator (pengali), jawabannya dituliskan.
2. Multiplikan selanjutnya dikalikan dengan digit puluhan multiplikator, jawabannya digeser ke kiri satu langkah agar mewujudkan perkalian puluhan, dan jawaban ini ditambahkan dengan hasil dari (1).
3. Multiplikan dikalikan dengan digit ratusan dari multiplikator, jawaban digeser ke kiri dua langkah agar mewujudkan perkalian 100 dan ini ditambahkan dengan hasil dari (2) dan demikian seterusnya.

Untuk perkalian biner hanya terdapat 1 dan 0 pada multiplikator (dan juga pada multiplikan), sehingga prosedur perkalian menjadi hanya langkah-langkah menggeser dan menjumlah.

Contoh :

a.

```
    1001011          75
       1101          13
  ----------       ------
    1001011         225
   0000000          75
  1001011          ------
 1001011           975
-------------
 1111001111
```

b.

```
    10010110         150
        1111          15
  ------------     -------
    10010110         750
   10010110          150
  10010110          -------
 10010110           2250
---------------
 100011001010
```

2.4. Komplemen

Komplemen 2 bilangan biner

Untuk mengubah bilangan biner ke bentuk komplemen 2 dilakukan dengan mengubah masing-masing digit bilangan biner tersebut, digit "0" diubah menjadi "1" dan sebaliknya digit "1" diubah menjadi "0", setelah itu digit yang paling kanan (LSB) ditambah "1".

Contoh :

Biner		Komplemen 2
(0) 1001	=	(1) 0110
		1
		---------- +
		(1) 0111
(1) 101,01	=	(0) 010,10
		1
		---------- +
		(0) 010,11

Selanjutnya untuk melakukan pengurangan (menjumlah dengan komplemen 2), bila tanda modulus menjadi "0" berarti hasilnya sudah benar, karena merupakan bilangan positif, tetapi bila tanda modulus menjadi "1", berarti masih berbentuk komplemen (bilangan negatif). Bila ada nilai pindahan (*carry*) baik "1" maupun "0" diabaikan.

Contoh :

```
        1001  ( 9 )    -    0101  ( 5 )    =
   ( 0 ) 1001
   ( 1 ) 1011  ( komplemen dari 5 )
  -------------- +
   1 ( 0 ) 0100
   \----------> pindahan ( carry ) diabaikan, se -
                hingga hasilnya   :  0100
```

Komplemen 1 Bilangan Biner

Seperti pada komplemen 2, untuk mengubah bilangan biner kebentuk komplemen 1, yaitu dengan mengubah digit "0" menjadi "1" dan sebaliknya digit "1" diubah menjadi digit "0". Dalam hal ini tidak perlu ditambah-kan digit "1" pada LSB.

Contoh :

```
   ( 0 )  1101101       =       ( 1 )  0010010
   ( 1 )   10010,101    =       ( 0 )   01101,010
```

Komplemen dari suatu bilangan biner diperoleh dengan jalan membalik tiap-tiap bit dari bilangan.
Jadi komplemen dari pengurangan (*substrahend*) dijumlahkan dengan yang dikurangi (*minued*), dan hasil pindahan dari jumlah bit yang paling besar bobotnya (MSB) dicatat. Jika hasilnya 1, berarti bahwa hasil pengurangan adalah bilangan positif.
Untuk memperoleh hasil akhir selanjutnya harus dilaku-kan apa yang disebut pindahan memutar ke ujung (*end*

around carry) dan untuk memperoleh hasil akhir bit yang paling kecil bobotnya (LSB) harus ditambahkan dengan pindahan tersebut.

Jika bit pindahan 0, maka dua hal dapat disimpulkan :

1. Hasil pengurangan adalah bilangan negatif.
2. Hasil pengurangan adalah komplemen dari jawaban akhir dan karenanya untuk memperoleh hasil yang benar, jawaban akhir harus dikomplemenkan.

Contoh :

1. 11001 (25) - 1010 (10)

```
   11001  ( 25 )
    0101  ( komplemen dari 10 )
   ---------
  ┌─(1)1110
  │   ┌┘
  │   │   1
  │   │  ------
  │   ↓
  │  + 1111        ( pindahan memutar dari ujung ke
  │     ↑
  └─────┘           ujung )
```

Jawaban = 15

2. 0110 (6) - 1101 (13)

```
   0110  ( 6 )
   0010  ( komplemen 13 )
   ------- +
   0 1000
   ↓
   0111  ( komplemen )
```

Jawaban = - 7

2.5. Sandi BCD (*Binary - Coded - Decimal*)

Pada perhitungan biasa kebanyakkan orang menggunakan bilangan desimal. Perhitungan bilangan biner hanya digunakan dalam mesin komputer atau peralatan logika lainnya. Sehingga untuk menghubungkan antara perhitungan biasa oleh manusia dengan perhitungan oleh mesin logika, perlu menjadikan bilangan desimal kebilangan yang diwujudkan oleh mesin logika tersebut.

Sistim Sandi BCD (*Binary - Coded - Decimal*) menggunakan kode biner 4 bit untuk mempresentasikan bilangan desimal (0, 1, ... 9), dimana nilai tiap-tiap bit sudah tertentu berdasarkan kode yang di-pakai.

Sandi 8421 BCD

Maksud sandi 8421 BCD adalah bahwa tiap kelompok empat bit bilangan biner yang mengganti bilangan desimal mempunyai urutan bobot bilangan : 8, 4, 2, dan 1 (mu-lai dari MSB sampai LSB).

Tabel 2.3.

Desimal	BCD *Codes*							
	8 - 4 - 2 - 1				XS - 3			
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Contoh :

1. Buatlah sandi 8421 BCD dari 296

   296 = 0010 1001 0110

   = (1010010110) 8421 BCD

2. Dari sandi 8421 BCD : 111 0111 0011 ubah-lah menjadi bilangan desimal biasa :

   111 0111 0011 = 111 0111 0011

   LSB ----- ------ ------

   = (7 7 3)