

---

# **MODUL SISTEM KOMPUTER**

---



**X TKJ (TEKNIK KOMPUTER DAN JARINGAN)**  
**SMK YABHINKA CILEGON**  
**2020/2021**

## MODUL 1 SISTEM BILANGAN

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Sistem Bilangan  
**Alokasi waktu** : Jam Pelajaran

### A. Tujuan pembelajaran :

- 1 Siswa mengenal jenis bilangan desimal, Biner, Oktal, dan Heksadesimal.
- 2 Siswa mampu membedakan jenis bilangan desimal, biner, oktal, heksadesimal.
- 3 Mampu membedakan banyaknya angka/digit pada masing-masing sistem bilangan.
- 4 Siswa terampil melakukan konversi/pengubahan biner dan decimal
- 5 Siswa terampil melakukan konversi sistem bilangan oktal dan heksadesimal

### B. Materi pembelajaran

*Sistem Digital* adalah suatu sistem yang berfungsi untuk mengukur suatu nilai atau besaran yang bersifat tetap atau tidak teratur dalam bentuk diskrit berupa digit digit atau angka angka. Biasanya sebelum mempelajari lebih dalam tentang sistem digital pertama pasti kita akan mempelajari yang namanya Sistem Bilangan.

#### 1. BASIS atau RADIK

Ada bermacam-macam sistem bilangan. Masing-masing sistem bilangan tersebut dibatasi oleh apa yang dinamakan *Basis atau Radik (Radix) yaitu banyaknya angka atau "Digit" yang digunakan.*

Misalnya sistem bilangan Desimal, mempunyai sepuluh digit yaitu : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Sehingga bilangan desimal adalah bilangan yang mempunyai radik :  $r = 10$ .

Nama dari masing-masing sistem bilangan itu pun berasal dari basis atau radiknya. Misalnya, dinamakan bilangan desimal karena desimal berarti sepuluh atau kelipatan sepuluh. Oleh karena itu dinamakan juga bilangan puluhan. Sistem bilangan yang lain misalnya bilangan Oktal, dinamakan begitu karena radiknya adalah delapan (Oktal : delapan). Bilangan Biner, karena radiknya adalah dua (Bi : mengandung arti dua).

Sistem bilangan Digital memiliki 4 macam yaitu Biner, Oktal, Desimal, HexaDesimal.

##### 1.1. Biner

*Biner* merupakan sebuah sistem bilangan yang berbasis dua, yaitu 0 dan 1. Sistem bilangan biner modern ditemukan oleh **Gottfried Wilhelm Leibniz** pada abad ke-17. Sistem bilangan ini merupakan dasar dari semua sistem bilangan berbasis digital. Dalam penulisan biasanya ditulis seperti berikut  $101001_2$ ,  $1001_2$ ,  $1010_2$ , dll.

##### 1.2. Oktal

*Oktal* merupakan sebuah sistim bilangan yang berbasis delapan dan memiliki 8 simbol yang berbeda (0,1,2,3,4,5,6,7). Dalam penulisan biasanya ditulis seperti berikut 2307<sub>8</sub>, 2355<sub>8</sub>, 102<sub>8</sub>, dll.

### 1.3. Desimal

*Desimal* merupakan sebuah sistim bilangan yang berbasis sepuluh dan memiliki 10 simbol yang berbeda (0,1,2,3,4,5,6,7,8,9). Desimal merupakan sistem bilangan yang biasa digunakan manusia dalam kehidupan sehari-hari.

### 1.4. HexaDesimal

*HexaDesimal* merupakan sebuah sistim bilangan yang berbasis 16 dan memiliki 16 simbol yang berbeda (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). Dalam penulisan biasanya ditulis seperti berikut 2D86<sub>16</sub>, 12DA<sub>16</sub>, FA<sub>16</sub>, dll.

## 2. KONVERSI BILANGAN

Konversi Bilangan digunakan untuk mengubah suatu bilangan dari suatu sistim bilangan menjadi bilangan dalam sistim bilangan yang lain.

### 2.1. Konversi Bilangan Biner

#### a. Biner ke Desimal

Cara mengubah bilangan Biner menjadi bilangan Desimal dengan mengalikan 2<sup>n</sup> dimana n merupakan posisi bilangan yang dimulai dari angka 0 dan dihitung dari belakang.

Contoh : 110001<sub>2</sub> diubah menjadi bilangan Desimal

$$\begin{aligned} 110001_2 &= (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 32 + 16 + 0 + 0 + 0 + 1 \\ &= 49 \end{aligned}$$

Jadi, 11001<sub>2</sub> = 49<sub>10</sub>

#### b. Biner ke Oktal

Cara mengubah bilangan Biner menjadi bilangan Oktal dengan mengambil 3 digit bilangan dari kanan.

**Tabel 1.1** Konversi Oktal ke Biner

Bilangan Oktal	Bilangan Biner
----------------	----------------

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

Contoh :  $11110011001_2$  diubah menjadi bilangan Oktal menjadi

$$\begin{aligned}
 11 \ 110 \ 011 \ 001 &= 11_2 &= 2^1 + 2^0 &= 3_8 \\
 &= 110_2 &= 2^2 + 2^1 &= 6_8 \\
 &= 011_2 &= 2^1 + 2^0 &= 3_8 \\
 &= 001_2 &= 2^0 &= 1_8
 \end{aligned}$$

Jadi,  $11110011001_2 = 3631_8$

### c. Biner ke HexaDesimal

Cara mengubah Biner menjadi bilangan HexaDesimal dengan mengambil 4 digit bilangan dari kanan .

**Tabel 1.2** Konversi Heksadesimal ke Biner

Bilangan Heksadesimal	Bilangan Biner
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0

9	1 0 0 1
10 = A	1 0 1 0
11 = B	1 0 1 1
12 = C	1 1 0 0
13 = D	1 1 0 1
14 = E	1 1 1 0
15 = F	1 1 1 1

Contoh:  $0100111101011100_2$  diubah menjadi bilangan HexaDesimal

$$\begin{aligned}
 0100 \ 1111 \ 0101 \ 1100 &= 0100_2 = 2^2 = 4_{16} \\
 &= 1111_2 = 2^3 + 2^2 + 2^1 + 2^0 = 15 = F_{16} \\
 &= 0101_2 = 2^2 + 2^0 = 5_{16} \\
 &= 1100_2 = 2^3 + 2^2 = 12 = C_{16}
 \end{aligned}$$

Jadi,  $0100111101011100_2 = 4F5C_{16}$

## 2.2. Konversi Bilangan Oktal

### a. Oktal ke Biner

Cara mengubah bilangan Oktal menjadi Biner dengan menjadikan satu persatu angka bilangan Oktal menjadi bilangan Biner dahulu kemudian di satukan. Untuk bilangan Oktal haruslah memiliki 3 digit bilangan Biner sehingga jika hanya menghasilkan kurang dari 3 digit maka didepannya ditambahkan bilangan 0.

Contoh :  $261_8$  diubah menjadi bilangan Biner

$$\begin{aligned}
 261 &= 2_8 = 010_2 \\
 &= 6_8 = 110_2 \\
 &= 1_8 = 001_2
 \end{aligned}$$

Jadi,  $261_8 = 010110001_2 = 10110001_2$  (angka 0 di depan angka 1 paling kiri dihilangkan)

### b. Oktal ke Desimal

Cara mengubah bilangan Oktal menjadi bilangan Desimal dengan mengubah bilangan Oktal tersebut menjadi bilangan Biner terlebih dahulu baru kita ubah menjadi bilangan Desimal.

Contoh :  $261_8$  diubah menjadi bilangan Desimal

**Langkah 1 :** mengubah ke bilangan Biner

$$\begin{aligned}
 261 &= 2_8 = 010_2 \\
 &= 6_8 = 110_2 \\
 &= 1_8 = 001_2
 \end{aligned}$$

$$\text{Jadi, } 261_8 = 010110001_2$$

**Langkah 2 :** mengubah bilangan Biner menjadi Desimal

$$\begin{aligned} 010110001_2 &= (0 \times 2^8) + (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) \\ &\quad + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 0 + 128 + 0 + 32 + 16 + 0 + 0 + 0 + 1 \\ &= 177 \end{aligned}$$

$$\text{Jadi, } 261_8 = 177_{10}$$

### c. Oktal ke HexaDesimal

Cara mengubah bilangan Oktal menjadi bilangan HexaDesimal dengan mengubah bilangan Oktal tersebut menjadi bilangan Biner terlebih dahulu baru kita ubah menjadi bilangan Heksadesimal.  
Contoh :  $261_8$  diubah menjadi bilangan HexaDesimal

**Langkah 1 :** mengubah ke bilangan Biner

$$\begin{aligned} 261 &= 2_8 = 010_2 \\ &= 6_8 = 110_2 \\ &= 1_8 = 001_2 \end{aligned}$$

$$\text{Jadi, } 261_8 = 010110001_2$$

**Langkah 2 :** mengubah bilangan Biner menjadi Heksadesimal

$$\begin{aligned} 010110001_2 &= 0001_2 = 2^0 &= 1_{16} \\ &= 1011_2 = 2^3 + 2^1 + 2^0 &= 11_{16} = B_{16} \\ &= 0000_2 &= 0_{16} \end{aligned}$$

$$\text{Jadi } 261_8 = B1_{16}$$

## 2.3. Konversi Bilangan Desimal

### a. Desimal ke Biner

Cara mengubah bilangan Desimal menjadi Biner yaitu dengan membagi bilangan Desimal dengan angka 2 dan tulis sisanya mulai dari bawah ke atas.

Contoh :  $25_{10}$  diubah menjadi bilangan Biner

$$\begin{array}{rcl} 25 : 2 &= 12 & \text{sisanya } 1 \\ 12 : 2 &= 6 & \text{sisanya } 0 \\ 6 : 2 &= 3 & \text{sisanya } 0 \\ 3 : 2 &= 1 & \text{sisanya } 1 \\ 1 : 2 &= 0 & \text{sisanya } 1 \end{array}$$

Maka ditulis nilai sisa dari bawah ke atas = 11001

$$\text{Jadi } 25_{10} = 11001_2$$

### b. Desimal ke Oktal

Cara mengubah bilangan Desimal menjadi Oktal yaitu dengan membagi bilangan Desimal dengan angka 8 dan tulis sisanya mulai dari bawah ke atas.

Contoh :  $80_{10}$  diubah menjadi bilangan Oktal

$$80 : 8 = 10 \text{ sisa } 0$$

$$10 : 8 = 1 \text{ sisa } 2$$

$$1 : 8 = 0 \text{ sisa } 1$$

Maka ditulis nilai sisa dari bawah ke atas = 120

$$\text{Jadi } 80_{10} = 120_8$$

### c. Desimal ke HexaDesimal

Cara mengubah bilangan Desimal menjadi HexaDesimal yaitu dengan membagi bilangan Desimal dengan angka 16 dan tulis sisanya mulai dari bawah ke atas.

Contoh : 275 diubah menjadi bilangan HexaDesimal

$$275 : 16 = 17 \text{ sisa } 3$$

$$17 : 16 = 1 \text{ sisa } 1$$

$$1 : 16 = 0 \text{ sisa } 1$$

Maka ditulis nilai sisa dari bawah ke atas = 113

$$\text{Jadi } 275 = 113_{16}$$

## 2.4. Konversi Bilangan HexaDesimal

### a. HexaDesimal ke Biner

Cara mengubah bilangan HexaDesimal menjadi Biner dengan menjadikan satu persatu angka bilangan HexaDesimal menjadi bilangan Biner dahulu kemudian di satukan. Untuk bilangan HexaDesimal haruslah memiliki 4 digit bilangan Biner sehingga jika hanya menghasilkan kurang dari 4 digit maka didepannya ditambahkan bilangan 0.

Contoh :  $4DA2_{16}$  diubah menjadi bilangan Biner

$$4DA2 = 4_{16} = 0100_2$$

$$= D_{16} = 1101_2$$

$$= A_{16} = 1010_2$$

$$= 2_{16} = 0010_2$$

$$\text{Jadi } 4DA2_{16} = 0100110110100010_2 = 100110110100010_2$$

### b. HexaDesimal ke Desimal

Cara mengubah bilangan biner menjadi bilangan desimal dengan mengalikan  $16^n$  dimana n merupakan posisi bilangan yang dimulai dari angka 0 dan dihitung dari belakang.

Contoh :  $3C2_{16}$  diubah menjadi bilangan Desimal

$$3C2_{16} = (3 \times 16^2) + (C(12) \times 16^1) + (2 \times 16^0)$$

$$= 768 + 192 + 2$$

$$= 962$$

$$\text{Jadi } 3C2_{16} = 962$$

### c. HexaDesimal ke Oktal

Cara mengubah bilangan HexaDesimal menjadi bilangan Oktal dengan mengubah bilangan HexaDesimal tersebut menjadi bilangan Desimal terlebih dahulu baru kita ubah menjadi bilangan Oktal

Contoh :  $3C2_{16}$  diubah menjadi bilangan Oktal

**Langkah 1:** Mengubah bilangan HexaDesimal menjadi Desimal

$$\begin{aligned} 3C2_{16} &= (3 \times 16^2) + (C(12) \times 16^1) + (2 \times 16^0) \\ &= 768 + 192 + 2 \\ &= 962 \end{aligned}$$

**Langkah 2 :** Mengubah bilangan Desimal menjadi Oktal

$$\begin{aligned} 962 : 8 &= 120 \text{ sisa } 2 \\ 120 : 8 &= 15 \text{ sisa } 0 \\ 15 : 8 &= 1 \text{ sisa } 7 \\ 1 : 8 &= 0 \text{ sisa } 1 \end{aligned}$$

Maka ditulis nilai sisa dari bawah ke atas = 1702

▪ Jadi  $3C2_{16} = 1702_8$

### 3. SISTEM BILANGAN

#### a. Bentuk Bilangan Dalam Code Form

Mengkonversi bilangan yang berharga besar, memerlukan hitungan yang cukup melelahkan. Melalui bilangan dalam Code Form maka pekerjaan konversi bilangan dapat dipermudah dan dipercepat. Di bawah ini adalah Code Form dalam bilangan Desimal, Bilangan Oktal dan bilangan Heksadesimal yang sering dipergunakan.

#### **Binary Code Decimal (BCD)**

Bilangan desimal pada setiap tempat dapat terdiri dari 10 bilangan yang berbeda-beda. Untuk bilangan biner bentuk dari 10 elemen yang berbeda beda memerlukan 4 bit. Sebuah BCD mempunyai 4 bit biner untuk setiap tempat bilangan desimal.

Contoh :

$$Z(10) = 317$$

3    1    7            *Desimal*

0011 0001 0111    *Biner Code Desimal*

Dalam contoh ini BCD terdiri dari 3 kelompok bilangan masing-masing terdiri dari 4 bit, dan jika bilangan desimal tersebut di atas dikonversi ke dalam bilangan biner secara langsung adalah  $317_{(10)} = 100111101_{(2)}$  dan hanya memerlukan 9 bit. Untuk contoh proses sebaliknya dapat dilihat di bawah ini.

Contoh :

*Biner Code Desimal*            0101 0001 0111 0000

*Desimal*                            5        1        7        0

Jadi bentuk BCD di atas adalah bilangan  $Z(10) = 5170$ .



Misalkan bilangan yang ingin dikonversi adalah  $170_{10}$ . dapat dilihat bahwa bilangan biner dari :

$1_{10} \rightarrow 0001_2$

$7_{10} \rightarrow 0111_2$

$0_{10} \rightarrow 0000_2$

Tetapi, berhubung hasil yang diinginkan adalah bilangan BCD, maka basis bilangannya tinggal ditulis sebagai berikut :

$1_{10} \rightarrow 0001_{BCD}$

$7_{10} \rightarrow 0111_{BCD}$

$0_{10} \rightarrow 0000_{BCD}$

maka, nilai BCD dari  $170_{10}$  adalah  $0001\ 0111\ 0000_{BCD}$ .

Setiap simbol dari bilangan desimal dikonversi menjadi 4 bit bilangan BCD.

Contoh lain, misalkan bilangan yang ingin dikonversi adalah  $309_{10}$ .

$3_{10} \rightarrow 0011_{BCD}$

$0_{10} \rightarrow 0000_{BCD}$

$9_{10} \rightarrow 1001_{BCD}$

maka, nilai BCD dari  $309_{10}$  adalah  $0011\ 0000\ 1001_{BCD}$ .

### **Binary Code Hexadecimal (BCH)**

Bilangan heksadesimal dalam setiap tempat dapat terdiri dari 16 bilangan yang berbeda-beda (angka dan huruf). Bentuk biner untuk 16 elemen memerlukan 4 bit. Sebuah BCH mempunyai 4 bit biner untuk setiap tempat bilangan heksadesimal.

Contoh :

$Z_{(16)} = 31AF$

*Bilangan Heksadesimal*     3     1     A     F

*Biner Code Heksadesimal*   0011   0001   1010   1111

Untuk proses sebaliknya, setiap 4 bit dikonversi ke dalam bilangan heksadesimal.

Contoh :

*Biner Code Heksadesimal*   1010 0110 0001 1000

*Bilangan Heksadesimal*     A     6     1     8

Jadi bentuk BCH diatas adalah bilangan  $Z_{(16)} = A618$ .

### **4. ASCII Code**

ASCII (*American Standard Code for Information Interchange*) adalah suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit signifikan paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi

logical communication, Device control, Information separator, Code extention, dan physical communication. Kode ASCII ini banyak dijumpai pada keyboard komputer atau instrumen-instrumen digital. Jumlah kode ASCII adalah 255 kode. Kode ASCII 0..127 merupakan kode ASCII untuk manipulasi teks; sedangkan kode ASCII 128..255 merupakan kode ASCII untuk manipulasi grafik. Kode ASCII sendiri dapat dikelompokkan lagi kedalam beberapa bagian :

Kode yang tidak terlihat simbolnya seperti Kode 10(Line Feed), 13(Carriage Return), 8(Tab), 32(Space)

Kode yang terlihat simbolnya seperti abjad (A..Z), numerik (0..9), karakter khusus (~!@#\$%^&\*()\_+?:'"})

Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik.

Dalam pengkodean kode ASCII memanfaatkan 8 bit. Pada saat ini kode ASCII telah tergantikan oleh kode UNICODE (Universal Code). UNICODE dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya.

Pada papan keyboard, aktifkan numlock (tidak terdapat pada laptop), tekan tombol ALT secara bersamaan dengan kode karakter maka akan dihasilkan karakter tertentu. Misalnya: ALT + 44 maka akan muncul karakter koma (,). Mengetahui kode-kode ASCII sangat bermanfaat misalnya untuk membuat karakter-karakter tertentu yang tidak ada di keyboard.

Fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-karakter angka maupun huruf didalam komputer, sebagai contoh dapat kita lihat pada karakter 1, 2, 3, A, B, C, dan sebagainya.

Tabel berikut berisi karakter-karakter ASCII . Dalam sistem operasi Windows dan MS-DOS, pengguna dapat menggunakan karakter ASCII dengan menekan tombol Alt+[nomor nilai ANSI (desimal)]. Sebagai contoh, tekan kombinasi tombol Alt+87 untuk karakter huruf latin "W" kapital.

Karakter	Nilai Unicode (heksadesimal)	Nilai ANSI ASCII (desimal)	Keterangan
NUL	0	0	Null (tidak tampak)
SOH	1	1	Start of heading (tidak tampak)
STX	2	2	Start of text (tidak tampak)
ETX	3	3	End of text (tidak tampak)
EOT	4	4	End of transmission (tidak tampak)
ENQ	5	5	Enquiry (tidak tampak)
ACK	6	6	Acknowledge (tidak tampak)
BEL	7	7	Bell (tidak tampak)
BS	8	8	Menghapus satu karakter di belakang kursor (Backspace)
HT	9	9	Horizontal tabulation
LF	000A	10	Pergantian baris (Line feed)
VT	000B	11	Tabulasi vertikal
FF	000C	12	Pergantian baris (Form feed)

CR	000D	13	Pergantian baris (carriage return)
SO	000E	14	Shift out (tidak tampak)
SI	000F	15	Shift in (tidak tampak)
DLE	10	16	Data link escape (tidak tampak)
DC1	11	17	Device control 1 (tidak tampak)
DC2	12	18	Device control 2 (tidak tampak)
DC3	13	19	Device control 3 (tidak tampak)
DC4	14	20	Device control 4 (tidak tampak)
NAK	15	21	Negative acknowledge (tidak tampak)
SYN	16	22	Synchronous idle (tidak tampak)
ETB	17	23	End of transmission block (tidak tampak)
CAN	18	24	Cancel (tidak tampak)
EM	19	25	End of medium (tidak tampak)
SUB	001A	26	Substitute (tidak tampak)
ESC	001B	27	Escape (tidak tampak)
FS	001C	28	File separator
GS	001D	29	Group separator
RS	001E	30	Record separator
US	001F	31	Unit separator
SP	20	32	Spasi
!	21	33	Tanda seru (exclamation)
"	22	34	Tanda kutip dua
#	23	35	Tanda pagar (kres)
\$	24	36	Tanda mata uang dolar
%	25	37	Tanda persen
&	26	38	Karakter ampersand (&)
'	27	39	Karakter Apostrof
(	28	40	Tanda kurung buka
)	29	41	Tanda kurung tutup
*	002A	42	Karakter asterisk (bintang)
+	002B	43	Tanda tambah (plus)
,	002C	44	Karakter koma
-	002D	45	Karakter hyphen (strip)
.	002E	46	Tanda titik
/	002F	47	Garis miring ( <i>slash</i> )
0	30	48	Angka nol
1	31	49	Angka satu
2	32	50	Angka dua
3	33	51	Angka tiga
4	34	52	Angka empat
5	35	53	Angka lima

6	36	54	Angka enam
7	37	55	Angka tujuh
8	38	56	Angka delapan
9	39	57	Angka sembilan
:	003A	58	Tanda titik dua
;	003B	59	Tanda titik koma
<	003C	60	Tanda lebih kecil
=	003D	61	Tanda sama dengan
>	003E	62	Tanda lebih besar
?	003F	63	Tanda tanya
@	40	64	A keong (@)
A	41	65	Huruf latin A kapital
B	42	66	Huruf latin B kapital
C	43	67	Huruf latin C kapital
D	44	68	Huruf latin D kapital
E	45	69	Huruf latin E kapital
F	46	70	Huruf latin F kapital
G	47	71	Huruf latin G kapital
H	48	72	Huruf latin H kapital
I	49	73	Huruf latin I kapital
J	004A	74	Huruf latin J kapital
K	004B	75	Huruf latin K kapital
L	004C	76	Huruf latin L kapital
M	004D	77	Huruf latin M kapital
N	004E	78	Huruf latin N kapital
O	004F	79	Huruf latin O kapital
P	50	80	Huruf latin P kapital
Q	51	81	Huruf latin Q kapital
R	52	82	Huruf latin R kapital
S	53	83	Huruf latin S kapital
T	54	84	Huruf latin T kapital
U	55	85	Huruf latin U kapital
V	56	86	Huruf latin V kapital
W	57	87	Huruf latin W kapital
X	58	88	Huruf latin X kapital
Y	59	89	Huruf latin Y kapital
Z	005A	90	Huruf latin Z kapital
[	005B	91	Kurung siku kiri
\	005C	92	Garis miring terbalik ( <i>backslash</i> )
]	005D	93	Kurung sikur kanan
^	005E	94	Tanda pangkat
	005F	95	Garis bawah ( <i>underscore</i> )
`	60	96	Tanda petik satu

a	61	97	Huruf latin a kecil
b	62	98	Huruf latin b kecil
c	63	99	Huruf latin c kecil
d	64	100	Huruf latin d kecil
e	65	101	Huruf latin e kecil
f	66	102	Huruf latin f kecil
g	67	103	Huruf latin g kecil
h	68	104	Huruf latin h kecil
i	69	105	Huruf latin i kecil
j	006A	106	Huruf latin j kecil
k	006B	107	Huruf latin k kecil
l	006C	108	Huruf latin l kecil
m	006D	109	Huruf latin m kecil
n	006E	110	Huruf latin n kecil
o	006F	111	Huruf latin o kecil
p	70	112	Huruf latin p kecil
q	71	113	Huruf latin q kecil
r	72	114	Huruf latin r kecil
s	73	115	Huruf latin s kecil
t	74	116	Huruf latin t kecil
u	75	117	Huruf latin u kecil
v	76	118	Huruf latin v kecil
w	77	119	Huruf latin w kecil
x	78	120	Huruf latin x kecil
y	79	121	Huruf latin y kecil
z	007A	122	Huruf latin z kecil
{	007B	123	Kurung kurawal buka
	007C	124	Garis vertikal (pipa)
}	007D	125	Kurung kurawal tutup
~	007E	126	Karakter gelombang (tilde)
DEL	007F	127	Delete
	80	128	Dicadangkan
	81	129	Dicadangkan
	82	130	Dicadangkan
	83	131	Dicadangkan
IND	84	132	Index
NEL	85	133	Next line
SSA	86	134	Start of selected area
ESA	87	135	End of selected area
	88	136	Character tabulation set
	89	137	Character tabulation with justification
	008A	138	Line tabulation set

PLD	008B	139	Partial line down
PLU	008C	140	Partial line up
	008D	141	Reverse line feed
SS2	008E	142	Single shift two
SS3	008F	143	Single shift three
DCS	90	144	Device control string
PU1	91	145	Private use one
PU2	92	146	Private use two
STS	93	147	Set transmit state
CCH	94	148	Cancel character
MW	95	149	Message waiting
	96	150	Start of guarded area
	97	151	End of guarded area
	98	152	Start of string
	99	153	Dicadangkan
	009A	154	Single character introducer
CSI	009B	155	Control sequence introducer
ST	009C	156	String terminator
OSC	009D	157	Operating system command
PM	009E	158	Privacy message
APC	009F	158	Application program command
	00A0	160	Spasi yang bukan pemisah kata
ı	00A1	161	Tanda seru terbalik
¢	00A2	162	Tanda sen (Cent)
£	00A3	163	Tanda Poundsterling
¤	00A4	164	Tanda mata uang ( <i>Currency</i> )
¥	00A5	165	Tanda Yen
ı ı	00A6	166	Garis tegak putus-putus ( <i>broken bar</i> )
§	00A7	167	Section sign
¨	00A8	168	Diaeresis
©	00A9	169	Tanda hak cipta (Copyright)
ª	00AA	170	Feminine ordinal indicator
«	00AB	171	Left-pointing double angle quotation mark
¬	00AC	172	Not sign
	00AD	173	Tanda strip ( <i>hyphen</i> )
®	00AE	174	Tanda merk terdaftar
—	00AF	175	Macron
°	00B0	176	Tanda derajat
±	00B1	177	Tanda kurang lebih (plus-minus)
²	kodok	178	Tanda kuadrat (pangkat dua)
³	00B3	179	Tanda kubik (pangkat tiga)

'	00B4	180	Acute accent
μ	00B5	181	Micro sign
¶	00B6	182	Pilcrow sign
·	00B7	183	Middle dot

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	sr	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(	100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29	)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[	111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	
011 1101	075	61	3D	=	101 1101	135	93	5D	]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					



## MODUL 2

### Relasi Logik & Fungsi Gerbang Dasar

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Relasi Logik & Fungsi Gerbang Dasar  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan pembelajaran :

- 1 Siswa mengenal jenis bilangan dan simbol gerbang logika
- 2 Siswa mampu memahami fungsi gerbang logika
- 3 Siswa terampil merencanakan rangkaian kombinasi gerbang logika

#### B. Materi pembelajaran

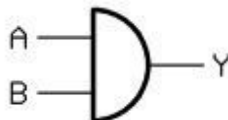
*Elektronika digital adalah sistem elektronik yang menggunakan signal digital.* Signal digital didasarkan pada signal yang bersifat terputus-putus. Biasanya dilambangkan dengan notasi aljabar 1 dan 0. Notasi 1 melambangkan terjadinya hubungan dan notasi 0 melambangkan tidak terjadinya hubungan. Contoh yang paling gampang untuk memahami pengertian ini adalah saklar lampu. Ketika menekan ON berarti terjadi hubungan sehingga dinotasikan 1. Ketika menekan OFF maka akan berlaku sebaliknya.

Elektronik digital atau atau rangkaian digital apapun tersusun dari apa yang disebut sebagai gerbang logika. Gerbang logika melakukan operasi logika pada satu atau lebih input dan menghasilkan output yang tunggal. Output yang dihasilkan merupakan hasil dari serangkaian operasi logika berdasarkan prinsip prinsip aljabar boolean. Dalam pengertian elektronik, input dan output ini diwujudkan dan voltase atau arus (tergantung dari tipe elektronik yang digunakan). Setiap gerbang logika membutuhkan daya yang digunakan sebagai sumber dan tempat buangan dari arus untuk memperoleh voltase yang sesuai.

Dasar pembentukan gerbang logika adalah tabel kebenaran (truth table). Ada tiga bentuk dasar dari tabel kebenaran yaitu AND, OR, dan NOT. Berikut penjelasan masing-masing gerbang logika.

#### 1. Gerbang logika AND

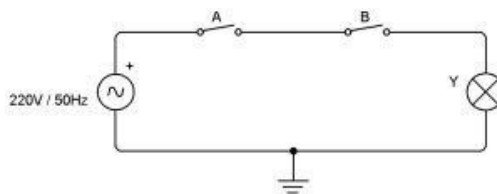
Gerbang AND mempunyai dua atau lebih dari dua sinyal input tetapi hanya satu sinyal output. Dalam gerbang AND, untuk menghasilkan sinyal keluaran berlogika high maka semua sinyal masukan harus bernilai high. Gerbang logika AND pada Datasheet nama lainnya **IC TTL 7408**.



**Gambar 2.1** Simbol Gerbang AND

**Tabel 2.1** Tabel Kebenaran Gerbang AND

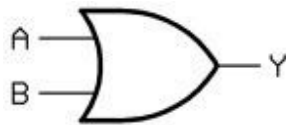
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



**Gambar 2.2** Analogi Elektrikal AND

## 2. Gerbang Logika OR

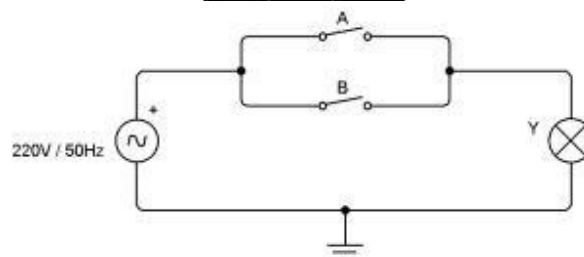
Gerbang OR mempunyai dua atau lebih dari dua sinyal input tetapi hanya satu sinyal output. Dalam gerbang OR, untuk menghasilkan sinyal keluaran berlogika high hanya butuh salah satu saja input berlogika high. Gerbang logika OR pada Datasheet nama lainnya **IC TTL 7432**.



**Gambar 2.3** Simbol Gerbang OR

**Tabel 2.2** Truth Table Gerbang OR

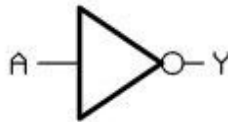
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



**Gambar 2.4** Analogi Elektrikal OR

### 3. Gerbang Logika NOT

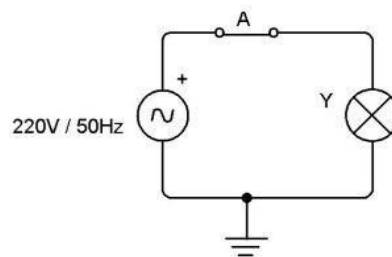
Gerbang NOT hanya mempunyai satu sinyal input dan satu sinyal output. Dalam gerbang NOT, untuk menghasilkan sinyal keluaran berlogika high sinyal masukan justru harus bernilai low. Gerbang logika NOT pada Datasheet nama lainnya **IC TTL 7404**.



**Gambar 2.5** Simbol Gerbang OR

**Tabel 2.3** Truth Table Gerbang NOT

A	Y
0	1
1	0



**Gambar 2.6** Analogi Elektrikal NOT

### 4. Gerbang Logika NAND

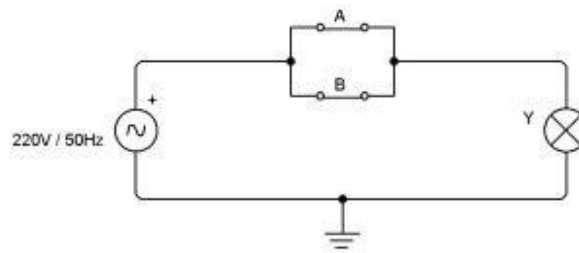
Gerbang NAND mempunyai dua atau lebih dari dua sinyal input tetapi hanya satu sinyal output. Dalam gerbang NAND, apabila salah satu input berlogika low maka output akan berlogika high. Gerbang logika NAND pada Datasheet nama lainnya **IC TTL 7400**.



**Gambar 2.7** Simbol Gerbang NAND

**Tabel 2.4** Truth Table Gerbang NAND

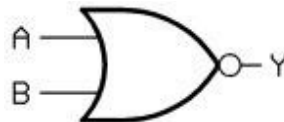
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



**Gambar 2.8** Analogi Elektrikal NAND

## 5. Gerbang Logika NOR

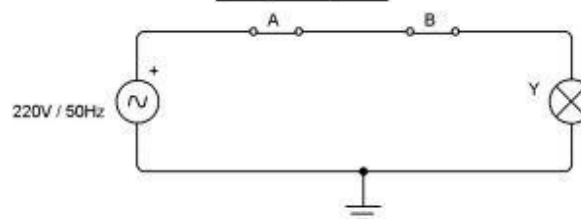
Gerbang NOR mempunyai dua atau lebih dari dua sinyal input tetapi hanya satu sinyal output. Dalam gerbang NOR, untuk menghasilkan sinyal keluaran berlogika high maka semua inputnya harus berlogika low. Gerbang logika NOR pada Datasheet nama lainnya **IC TTL 7402**.



**Gambar 2.9** Simbol Gerbang NOR

**Tabel 2.5** Truth Table Gerbang NOR

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



**Gambar 2.10** Analogi Elektrikal NOR

## 6. Gerbang Logika EX-OR (Exlusive OR)

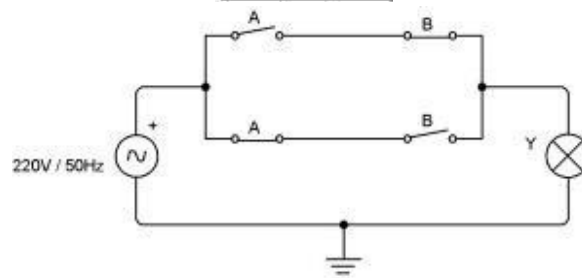
Gerbang Ex-Or mempunyai dua atau lebih dari dua sinyal input tetapi hanya satu sinyal output. Dalam gerbang Ex-Or, untuk menghasilkan sinyal keluaran berlogika high maka semua sinyal masukan harus bernilai berbeda. Gerbang logika Ex-Or pada Datasheet nama lainnya **IC TTL 7486**.



**Gambar 2.11** Simbol Gerbang EX-OR

**Tabel 2.6** Truth Table Gerbang EX-OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



**Gambar 2.12** Analogi Elektrikal EX-OR

## 7. Gerbang Logika EX-NOR

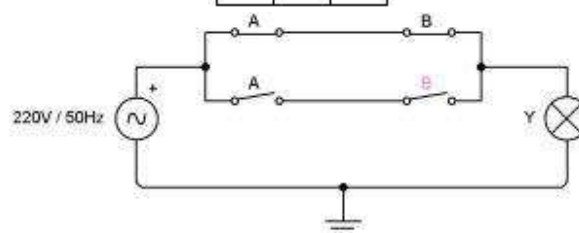
Gerbang Ex-Nor mempunyai dua atau lebih dari dua sinyal input tetapi hanya satu sinyal output. Dalam gerbang Ex-Nor, untuk menghasilkan sinyal keluaran berlogika high maka semua sinyal masukan harus bernilai sama. Gerbang logika Ex-Nor pada Datasheet nama lainnya **IC TTL 74266**.



**Gambar 2.13** Simbol Gerbang EX-NOR

**Tabel 2.7** Truth Table Gerbang EX-NOR

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



**Gambar 2.14** Analogi Elektrikal EX-NOR



## MODUL 3

### Operasi Aritmatik

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Operasi Aritmatik  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan pembelajaran :

- 1 Siswa mengenal jenis operasi aritmatik
- 2 Siswa terampil melakukan percobaan ALU

#### B. Materi Pembelajaran

Operasi Bilangan Biner

#### PENJUMLAHAN BINER

Seperti bilangan desimal, bilangan biner juga dijumlahkan dengan cara yang sama. Pertama-tama yang harus dicermati adalah aturan pasangan digit biner berikut :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ } \square \text{ menyimpan 1}$$

sebagai catatan bahwa jumlah dua yang terakhir adalah :

$$1 + 1 + 1 = 1 \text{ } \square \text{ dengan menyimpan 1}$$

Dengan hanya menggunakan penjumlahan-penjumlahan pada slide sebelumnya, kita dapat melakukan penjumlahan biner seperti ditunjukkan di bawah ini :

1 1111 --> “simpanan 1” ingat kembali aturan di atas

01011011 --> bilangan biner untuk 91

01001110 --> bilangan biner untuk 78

-----+

10101001 --> Jumlah dari  $91 + 78 = 169$

Kita akan menghitung penjumlahan biner yang terdiri dari 5 bilangan:

11101 bilangan <sup>1)</sup>

10110 bilangan <sup>2)</sup>

1100 bilangan <sup>3)</sup>

11011 bilangan <sup>4)</sup>

1001 bilangan <sup>5)</sup>

----- +

Untuk menjumlahkannya, kita hitung berdasarkan aturan yang berlaku, dan untuk lebih mudahnya perhitungan dilakukan bertahap

$$\begin{array}{rcl}
 11101 & \text{bilangan } ^1) & \\
 10110 & \text{bilangan } ^2) & \\
 \hline
 110011 & & \\
 1100 & \text{bilangan } ^3) & \\
 \hline
 111111 & & \\
 11011 & \text{bilangan } ^4) & \\
 \hline
 1011010 & & \\
 1001 & \text{bilangan } ^5) & \\
 \hline
 1100011 & \text{Jumlah Akhir} & 
 \end{array}$$

Apakah benar hasil penjumlahan tersebut?

$$\begin{array}{rcl}
 11101 & \text{bilangan } ^1) & \\
 10110 & \text{bilangan } ^2) & \\
 1100 & \text{bilangan } ^3) & \\
 11011 & \text{bilangan } ^4) & \\
 1001 & \text{bilangan } ^5) & \\
 \hline
 1100011 & \text{Jumlah Akhir} & 
 \end{array}$$

Mari Buktikan dengan merubah biner ke desimal.

$$\begin{array}{rcl}
 11101 & = & 29 \\
 10110 & = & 22 \\
 1100 & = & 12 \\
 11011 & = & 27 \\
 1001 & = & 9 \\
 \hline
 1100011 & = & 99 \text{ Sesuai!}
 \end{array}$$

## PENGURANGAN BINER

Untuk memahami konsep pengurangan biner, kita harus mengingat kembali perhitungan desimal (angka biasa), kita mengurangkan digit desimal dengan digit desimal yang lebih kecil. Jika digit



desimal yang dikurangkan lebih kecil daripada digit desimal yang akan dikurangi, maka terjadi “konsep meminjam”. Digit tersebut akan meminjam 1 dari digit sebelah kirinya.

Bentuk Umum pengurangan sebagai berikut :

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ } \square \text{ meminjam '1' dari digit disebelah kirinya}$$

Contoh :

1111011      desimal 123

101001      desimal 41

----- -

1010010      desimal 82

Pada contoh di atas tidak terjadi “konsep meminjam”.

Perhatikan contoh berikut!

0      kolom ke-3 menjadi ‘0’, sudah dipinjam

111101      desimal 61

10010      desimal 18

----- -

101011      Hasil pengurangan akhir 43

Pada soal yang kedua ini kita pinjam ‘1’ dari kolom 3, karena ada selisih 0-1 pada kolom ke-2  
Lalu bagaimana jika saya tidak dapat meminjam 1 dari kolom berikutnya karena kolom tersebut berupa bilangan ‘0’?

Untuk membahas hal itu mari kita beri bandingkan jika hal ini terjadi pada bilangan desimal. Mari kita hitung desimal 800046 – 397261!

7999

8000<sup>1</sup>46

3972 61

----- -

4027 05

Perhatikan bahwa kita meminjam 1 dari kolom keenam untuk kolom kedua, karena kolom ketiga, keempat dan kolom kelima adalah nol. Setelah meminjam, kolom ketiga, keempat, dan kelima menjadi:  
**10 – 9 = 1**

Hal ini juga berlaku dalam pengurangan biner, kecuali bahwa setelah meminjam kolom nol akan mengandung: **10 – 1 = 1**

Sebagai contoh pengurangan bilangan biner 110001–1010 akan diperoleh hasil sebagai berikut:

$$\begin{array}{r}
 1100^101 \\
 10\ 10 \\
 \hline
 1001\ 11
 \end{array}$$

## PERKALIAN BINER

Metode yang digunakan dalam perkalian biner juga pada dasarnya sama dengan perkalian desimal, akan terjadi pergeseran ke kiri setiap dikalikan 1 bit pengali. Setelah proses perkalian masing-masing bit pengali selesai, dilakukan penjumlahan masing-masing kolom bit hasil.

Contoh :

$$\begin{array}{r}
 1101 \\
 1011 \\
 \hline
 1101 \\
 1101 \\
 0000 \\
 1101 \\
 \hline
 10001111
 \end{array}$$

Perkalian bilangan biner dapat dilakukan seperti pada perkalian bilangan desimal. Sebagai contoh, untuk mengalikan  $1110_2 = 14_{10}$  dengan  $1101_2 = 13_{10}$  langkah-langkah yang harus ditempuh adalah:

Biner	Desimal
1 1 1 0	1 4
1 1 0 1	1 3
-----x	----x
1 1 1 0	4 2
0 0 0 0	1 4
1 1 1 0	
1 1 1 0	
-----+	-----+
1 0 1 1 0 1 1 0	1 8 2

Perkalian juga bisa dilakukan dengan menambahkan bilangan yang dikalikan ke bilangan itu sendiri sebanyak bilangan pengali.

Contoh barusan, hasilnya akan sama dengan jika kita menambahkan  $1112$  ke bilangan itu sendiri sebanyak  $1101$  atau 13 kali.

## PEMBAGIAN BINER

Serupa dengan perkalian, pembagian pada bilangan biner juga menggunakan metode yang sama dengan pembagian desimal. Bit-bit yang dibagi diambil bit per bit dari sebelah kiri. Apabila nilainya lebih dari bit pembagi, maka bagilah bit-bit tersebut, tetapi jika setelah bergeser 1 bit nilainya masih dibawah nilai pembagi maka hasilnya adalah 0.

Contoh :

$$\begin{array}{r}
 \text{Biner} \\
 011 \overline{) 1001} \\
 \underline{- 011} \phantom{0} \\
 0011 \\
 \underline{- 011} \\
 0
 \end{array}
 = 3_{10} \leftarrow \text{hasil bagi}$$

yang dibagi

pembagi

Pembagian pada sistem bilangan biner dapat dilakukan sama seperti contoh pembagian sistem bilangan desimal. Sebagai contoh, untuk membagi 110011 (disebut bilangan yang dibagi) dengan 1001 (disebut pembagi), langkah-langkah berikut yang perlu dilakukan.

$$\begin{array}{r}
 101 \text{ Hasil} \\
 \hline
 1001 \overline{) 110011} \\
 \underline{1001} \phantom{00} \\
 001111 \\
 \underline{1001} \phantom{00} \\
 110 \text{ sisa}
 \end{array}$$

Sehingga hasilnya adalah 101, dan sisa pembagian adalah 110.

Pembagian bisa juga dilakukan dengan cara menjumlahkan secara berulang kali dengan bilangan pembagi dengan bilangan itu sendiri sampai jumlahnya sama dengan bilangan yang dibagi atau setelah sisa pembagian yang diperoleh lebih kecil dari bilangan pembagi.

## MODUL 4

### Operasi Aritmatik

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Operasi Aritmatik  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan Pembelajaran :

- 1 Siswa mengenal rangkaian half adder dan full adder
- 2 Siswa mampu membedakan rangkaian ripple carry adder








#### B. Materi Pembelajaran

#### Gerbang Logika

#### HALF ADDER & FULL ADDER

Gerbang logika atau sering juga disebut gerbang logika Boolean merupakan sebuah sistem pemrosesan dasar yang dapat memproses input-input yang berupa bilangan biner menjadi sebuah output yang berkondisi yang akhirnya digunakan untuk proses selanjutnya. Gerbang logika dapat mengkondisikan input-input yang masuk kemudian menjadikannya sebuah output yang sesuai dengan apa yang ditentukan olehnya. Terdapat tiga gerbang logika dasar, yaitu : gerbang AND, gerbang OR, gerbang NOT. Ketiga gerbang ini menghasilkan empat gerbang berikutnya, yaitu : gerbang NAND, gerbang NOR, gerbang XNOR.

**Tabel 4.1 : Kebenaran Gerbang Logika.**

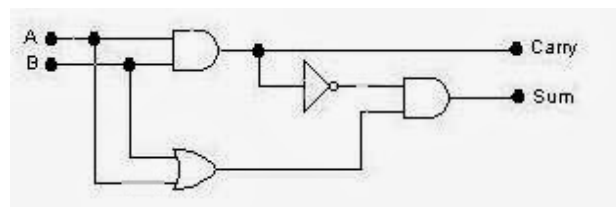
AND (7408)		NAND (7400)		XOR(7486)	
0 0=0	<div><div>A</div><div>B</div><div><math>A \cdot B = C</math></div></div>	0 0=1	<div><div>A</div><div>B</div><div><math>\overline{A \cdot B} = C</math></div></div>	0 0=0	<div><div>A</div><div>B</div><div><math>A \oplus B = C</math></div></div>
0 1=0		0 1=1		0 1=1	
1 0=0		1 0=1		1 0=1	
1 1=1		1 1=0		1 1=0	
OR (7432)		NOR (7402)		XAND	
0 0=0	<div><div>A</div><div>B</div><div><math>A+B = C</math></div></div>	0 0=1	<div><div>A</div><div>B</div><div><math>\overline{A+B} = C</math></div></div>	0 0=1	<div><div>A</div><div>B</div><div><math>\overline{A \oplus B} = C</math></div></div>
0 1=1		0 1=0		0 1=0	
1 0=1		1 0=0		1 0=0	
1 1=1		1 1=0		1 1=1	
		NOT (7404)			
		0 = 1			
		1 = 0			
		<div><div>A</div><div></div><div>C</div></div>			

Rangkaian aritmatika dasar termasuk ke dalam rangkaian kombinasional yaitu suatu rangkaian yang output-nya tidak tergantung pada kondisi output sebelumnya, hanya tergantung pada present state dari input.

### HALF ADDER

Merupakan rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan dari dua buah bilangan binary, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki dua input dan dua buah output, salah satu outputnya dipakai sebagai tempat nilai pindahan dan yang lain sebagai hasil dari penjumlahan.

Rangkaian ini bisa dibangun dengan menggunakan IC 7400 dan IC 7408. Seperti yang terlihat pada gambar dibawah ini, rangkaian half adder merupakan gabungan beberapa gerbang NAND dan satu gerbang AND. Karakter utama sebuah gerbang NAND dalah bahwa ia membalikkan hasil dari sebuah gerbang AND yang karakternya hanya akan menghasilkan nilai satu ketika kedua inputnya bernilai satu, jadi gerbang NAND hanya akan menghasilkan nilai nol ketika semua inputnya bernilai satu.



**Gambar 4.1 : Skema Diagram HALF ADDER**

Ketika salah satu atau lebih input bernilai nol maka keluaran pada gerbang NAND pertama akan bernilai satu. Karenanya kemudian input di gerbang kedua dan ketiga akan bernilai satu dan mendapat input lain yang salah satunya bernilai nol sehingga PASTI gerbang NAND yang masukannya nol tadi menghasilkan nilai satu. Sedangkan gerbang lain akan bernilai nol karena mendapat input satu dan satu maka keluaran di gerbang NAND terakhir akan bernilai satu, karena salah satu inputnya bernilai nol.

Untuk menghitung carry digunakan sebuah gerbang AND yang karakter utamanya adalah bahwa ia hanya akan menghasilkan nilai satu ketika kedua masukannya bernilai satu. Jadi carry satu hanya akan dihasilkan dari penjumlahan dua digit bilangan biner sama-sama bernilai satu, yang dalam penjumlahan utamanya akan menghasilkan nilai nol.

**Tabel 4.2 : Kebenaran HALF ADDER**

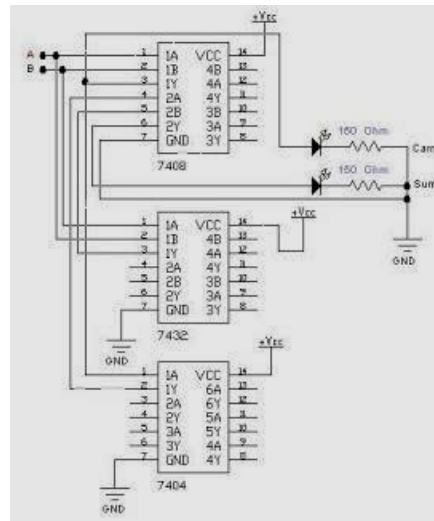
INPUT		OUTPUT	
A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Ket :

1 = Benar

0 = Salah

Jika setiap elemen yang dihubungkan salah satu ada yang Salah/(0) maka pernyataan pada percobaan Rangkaian Half Adder ini menunjukkan Salah/(0).



**Gambar 4.2 : Skema Pengkabelan HALF ADDER**

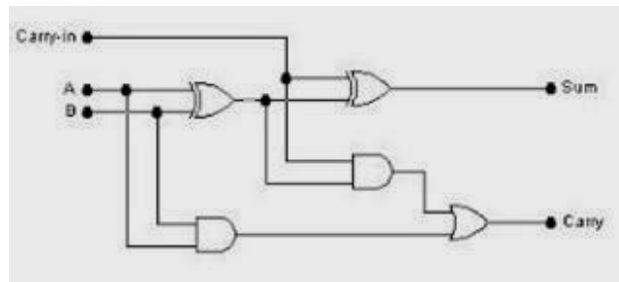
### KESIMPULAN :

*Half Adder* adalah suatu rangkaian penjumlahan sistem bilangan biner yang paling sederhana. Rangkaian ini hanya dapat digunakan untuk operasi penjumlahan data bilangan biner sampai 1bit saja. Rangkaian Half Adder memiliki 2 terminal input untuk 2 variabel bilangan biner dan 2 terminal output, yaitu SUMMARY OUT (SUM) dan CARRY OUT (CARRY).

### FULL ADDER

Merupakan rangkaian elektronik yang bekerja melakukan perhitungan penjumlahan sepenuhnya dari dua buah bilangan binary, yang masing-masing terdiri dari satu bit. Rangkaian ini memiliki tiga input dan dua buah output, salah satu input merupakan nilai dari pindahan penjumlahan, kemudian sama seperti pada half adder salah satu outputnya dipakai sebagai tempat nilai pindahan dan yang lain sebagai hasil dari penjumlahan.

Rangkaian ini dibuat dengan gabungan dua buah half adder dan sebuah gerbang OR. Logika utama rangkaian gerbang full adder adalah bahwa ketika menjumlahkan dua bilangan biner maka ada sebuah carry yang juga mempengaruhi hasil dari penjumlahan tersebut, karenanya rangkaian ini bisa melakukan penjumlahan secara sepenuhnya.



**Gambar 4.3 : Skema Diagram FULL ADDER**

Ketika dua masukan menghasilkan nilai satu pada half adder atau paruh dari full adder pertama, hasilnya akan kembali dijumlahkan dengan carry yang ada. Jika carry bernilai satu maka ia akan menghasilkan keluaran akhir bernilai nol, namun menghasilkan carry out yang bernilai satu, dan jika carry in bernilai nol maka ia akan menghasilkan keluaran akhir satu dengan carry out bernilai nol.

Lain halnya ketika kedua masukan pada paruh full adder pertama menghasilkan nilai nol karena inputnya sama-sama satu, maka carry out untuk paruh pertama half adder adalah satu, penjumlahan paruh pertama yang menghasilkan nol akan kembali dijumlahkan dengan carry in yang ada, yang jika bernilai satu maka hasil penjumlahannya adalah satu dan memiliki carry out satu dari penjumlahan input pertama.

Untuk menghitung carry out pada full adder digunakan sebuah gerbang OR yang menghubungkan penghitung carry out dari half adder pertama dan kedua. Maksudnya bahwa entah paruh pertama atau kedua yang menghasilkan carry out maka akan dianggap sebagai carry out, dan dianggap satu meski kedua gerbang AND yang digunakan untuk menghitung carry out sama-sama bernilai satu.

**Tabel 4.3 : Kebenaran FULL ADDER**

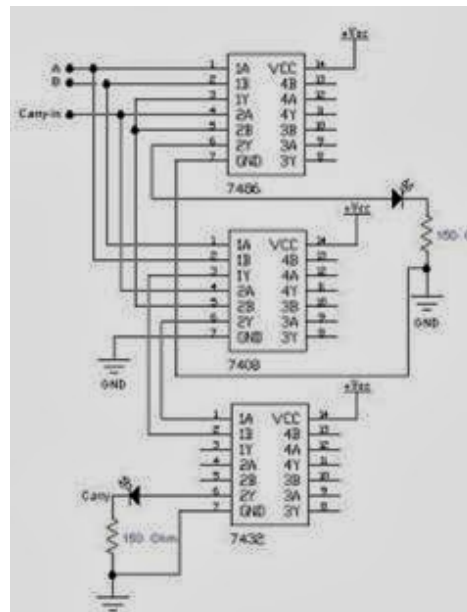
INPUT			OUTPUT	
A	B	CARRY IN	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Ket :

1 = Benar

0 = Salah

Jika setiap elemen yang dihubungkan salah satu ada yang Benar/(1) maka pernyataan pada percobaan Rangkaian Full Adder ini menunjukkan pernyataan Benar/(1)



**Gambar 4.4 : Skema Pengkabelan FULL ADDER**

#### KESIMPULAN :

*Full Adder* dapat digunakan untuk menjumlahkan bilangan-bilangan biner yang lebih dari 1bit. Penjumlahan bilangan-bilangan biner sama halnya dengan penjumlahan bilangan decimal dimana hasil penjumlahan tersebut terbagi menjadi 2 bagian, yaitu SUMMARY (SUM) dan CARRY, apabila hasil penjumlahan pada suatu tingkat atau kolom melebihi nilai maksimumnya maka output CARRY akan berada pada keadaan logika 1.

#### Pengertian dan Cara Kerja Arithmetic Logical Unit (ALU)

##### Pengenalan ALU

Arithmetic Logical Unit (ALU), adalah salah satu bagian/komponen dalam sistem di dalam sistem komputer yang berfungsi melakukan operasi/perhitungan aritmatika dan logika (Contoh operasi aritmatika adalah operasi penjumlahan dan pengurangan, sedangkan contoh operasi logika adalah logika AND dan OR. ALU bekerja bersama-sama memori, di mana hasil dari perhitungan di dalam ALU di simpan ke dalam memori

Perhitungan dalam ALU menggunakan kode biner, yang merepresentasikan instruksi yang akan dieksekusi (opcode) dan data yang diolah (operand). ALU biasanya menggunakan sistem bilangan biner *two's complement*. ALU mendapat data dari register. Kemudian data tersebut diproses dan hasilnya akan disimpan dalam register tersendiri yaitu ALU output register, sebelum disimpan dalam memori.

Pada saat sekarang ini sebuah chip/IC dapat mempunyai beberapa ALU sekaligus yang memungkinkan untuk melakukan kalkulasi secara paralel. Salah satu chip ALU yang sederhana



(terdiri dari 1 buah ALU) adalah IC 74LS382/HC382ALU (TTL). IC ini terdiri dari 20 kaki dan beroperasi dengan 4×2 pin data input (pinA dan pinB) dengan 4 pin keluaran (pinF).

### Kegunaan / Fungsi ALU

Arithmetic Logical Unit (ALU), fungsi unit ini adalah untuk melakukan suatu proses data yang berbentuk angka dan logika, seperti data matematika dan statistika. ALU terdiri dari register-register untuk menyimpan informasi. Tugas utama dari ALU adalah melakukan perhitungan aritmatika (matematika) yang terjadi sesuai dengan instruksi program. Rangkaian pada ALU (Arithmetic and Logic Unit) yang digunakan untuk menjumlahkan bilangan dinamakan dengan Adder. Adder digunakan untuk memproses operasi aritmetika, Adder juga disebut rangkaian kombinasional aritmatika.

### Berikut contoh sederhana dari rangkaian ALU dan cara kerjanya

Rangkaian utama yang digunakan untuk melakukan perhitungan ALU adalah Adder. Adder adalah rangkaian ALU (Arithmetic and Logic Unit) yang digunakan untuk menjumlahkan bilangan dinamakan dengan Adder. Karena Adder digunakan untuk memproses operasi aritmetika, maka Adder juga sering disebut rangkaian kombinasional aritmetika.

Ada 3 jenis adder:

1. Rangkaian Adder dengan menjumlahkan dua bit disebut Half Adder.
2. Rangkaian Adder dengan menjumlahkan tiga bit disebut Full Adder.
3. Rangkaian Adder dengan menjumlahkan banyak bit disebut Paralel Adder.

#### 1. HALF ADDER

Rangkaian Half Adder merupakan dasar penjumlahan bilangan Biner yang terdiri dari satu bit, oleh karena itu dinamai Penjumlah Tak Lengkap.

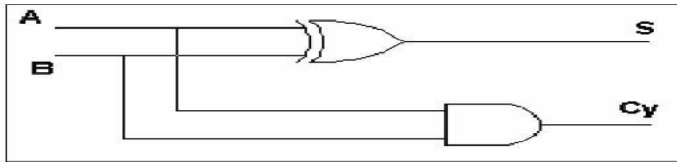
- a. jika  $A = 0$  dan  $B = 0$  dijumlahkan, hasilnya  $S$  (Sum) = 0.
  - b. jika  $A = 0$  dan  $B = 1$  dijumlahkan, hasilnya  $S$  (Sum) = 1.
  - c. jika  $A = 1$  dan  $B = 1$  dijumlahkan, hasilnya  $S$  (Sum) = 0
- jika  $A = 1$  dan  $B = 1$  dijumlahkan, hasilnya  $S$  (Sum) = 0. dengan nilai pindahan  $cy$  (Carry Out) = 1

Dengan demikian, half adder memiliki 2 masukan (A dan B) dan dua keluaran (S dan Cy).

A	B	S	Cy
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Dari tabel diatas, terlihat bahwa nilai logika dari Sum sama dengan nilai logika dari gerbang XOR, sedangkan nilai logika Cy sama dengan nilai dari gerbang logika AND.

Dari table tersebut, dapat dibuat rangkaian half adder seperti pada gambar berikut:



## 2. FULL ADDER

Sebuah Full Adder menjumlahkan dua bilangan yang telah dikonversikan menjadi bilangan-bilangan biner. Masing-masing bit pada posisi yang sama saling dijumlahkan. Full Adder sebagai penjumlah pada bit-bit selain yang terendah. Full Adder menjumlahkan dua bit input ditambah dengan nilai Carry-Out dari penjumlahan bit sebelumnya. Output dari Full Adder adalah hasil penjumlahan (Sum) dan bit kelebihannya (carry-out).

Full adder mengolah penjumlahan untuk 3 bit bilangan atau lebih (bit tidak terbatas), olehkarena itu dinamakan rangkaian penjumlah lengkap. Perhatikan tabel kebenaran dari Full adder berikut :

A	B	C	S	Cy
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

Dari tabel diatas dapat dibuat persamaan boolean sebagai berikut :

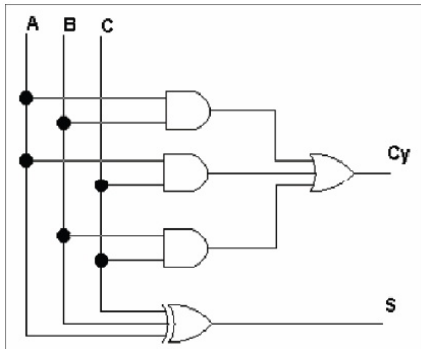
$$S = A B C + A B \bar{C} + A \bar{B} C + A \bar{B} \bar{C}$$

$$S = A \oplus B \oplus C$$

$$Cy = A B C + A B \bar{C} + A \bar{B} C + A \bar{B} \bar{C}$$

Dengan menggunakan peta karnaugh, Cy dapat disederhanakan menjadi :

$$Cy = AB + AC + BC$$



### 3. PARALEL ADDER

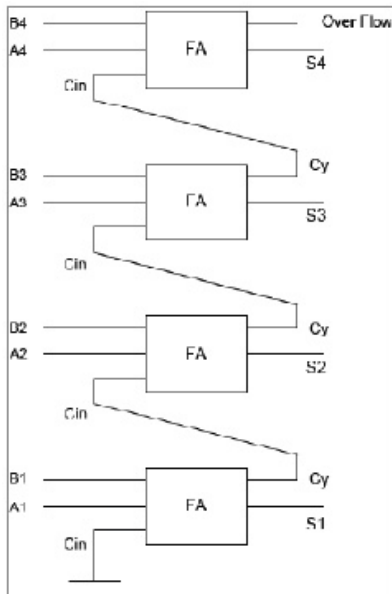
Rangkaian Parallel Adder adalah rangkaian penjumlah dari dua bilangan yang telah dikonversikan ke dalam bentuk biner. Anggap ada dua buah register A dan B, masing-masing register terdiri dari 4 bit biner : A3A2A1A0 dan B3B2B1B0.

Rangkaian Parallel Adder terdiri dari Sebuah Half Adder (HA) pada Least Significant Bit (LSB) dari masing-masing input dan beberapa Full Adder pada bit-bit berikutnya. Prinsip kerja dari Parallel Adder adalah sebagai berikut : penjumlahan dilakukan mulai dari LSB-nya. Jika hasil penjumlahan adalah bilangan desimal “2” atau lebih, maka bit kelebihanya disimpan pada Cout, sedangkan bit di bawahnya akan dikeluarkan pada Σ. Begitu seterusnya menuju ke Most Significant Bit (MSB)nya.

Tugas lain dari ALU adalah melakukan keputusan dari operasi sesuai dengan instruksi program yaitu operasi logika (logical operation). Operasi logika meliputi perbandingan dua buah elemen logika dengan menggunakan operator logika.

Parallel Adder adalah rangkaian Full Adder yang disusun secara parallel dan berfungsi untuk menjumlah bilangan biner berapapun bitnya, tergantung jumlah Full Adder yang diparalelkan. Gambar berikut menunjukkan Parallel Adder yang terdiri dari 4 buah Full Adder yang tersusun parallel sehingga membentuk sebuah penjumlah 4 bit.

Contoh:



### Tugas Arithmetic Logical Unit (ALU)

Bertugas membentuk fungsi – fungsi pengolahan data komputer.

ALU sering disebut mesin bahasa (machine language) karena bagian ini mengerjakan instruksi – instruksi bahasa mesin yang diberikan padanya.

ALU terdiri dari dua bagian, yaitu unit aritmetika dan unit logika boolean, yang masing – masing memiliki spesifikasi dan tugas tersendiri.

Fungsi-fungsi yang didefinisikan pada ALU adalah Add(penjumlahan), Addu (penjumlahan tidak bertanda), Sub(pengurangan), Subu (pengurangan tidak bertanda), and, or, xor, sll (shift left logical), srl (shift right logical), sra (shift right arithmetic), dan lain-lain.

Arithmetic Logical Unit (ALU) merupakan unit penalaran secara logic. ALU ini merupakan Sirkuit CPU berkecepatan tinggi yang bertugas menghitung dan membandingkan. Angka-angka dikirim dari memori ke ALU untuk dikalkulasi dan kemudian dikirim kembali ke memori. Jika CPU diasumsikan sebagai otaknya komputer, maka ada suatu alat lain di dalam CPU tersebut yang kenal dengan nama Arithmetic Logical Unit (ALU), ALU inilah yang berfikir untuk menjalankan perintah yang diberikan kepada CPU tersebut.

ALU sendiri merupakan suatu kesatuan alat yang terdiri dari berbagai komponen perangkat elektronika termasuk di dalamnya sekelompok transistor, yang dikenal dengan nama logic gate, dimana logic gate ini berfungsi untuk melaksanakan perintah dasar matematika dan operasi logika. Kumpulan susunan dari logic gate inilah yang dapat melakukan perintah perhitungan matematika yang lebih komplrit seperti perintah “add” untuk menambahkan bilangan, atau “devide” atau pembagian dari suatu bilangan. Selain perintah matematika yang lebih komplrit, kumpulan dari logic gate ini juga mampu untuk melaksanakan perintah yang berhubungan dengan logika, seperti hasil perbandingan dua buah bilangan.

Instruksi yang dapat dilaksanakan oleh ALU disebut dengan *instruction set*. Perintah yang ada pada masing-masing CPU belum tentu sama, terutama CPU yang dibuat oleh pembuat yang berbeda,

katakanlah misalnya perintah yang dilaksanakan oleh CPU buatan Intel belum tentu sama dengan CPU yang dibuat oleh Sun atau perusahaan pembuat mikroprosesor lainnya. Jika perintah yang dijalankan oleh suatu CPU dengan CPU lainnya adalah sama, maka pada level inilah suatu sistem dikatakan compatible. Sehingga sebuah program atau perangkat lunak atau software yang dibuat berdasarkan perintah yang ada pada Intel tidak akan bisa dijalankan untuk semua jenis prosesor, kecuali untuk prosesor yang compatible dengannya.

Seperti halnya dalam bahasa yang digunakan oleh manusia, instruction set ini juga memiliki aturan bahasa yang bisa saja berbeda satu dengan lainnya. Bandingkanlah beda struktur bahasa Inggris dengan Indonesia, atau dengan bahasa lainnya, begitu juga dengan instruction set yang ada pada mesin, tergantung dimana lingkungan instruction set itu digunakan.

## MODUL 5

### Rangkaian Multiplexer, Decoder, Flip-Flop dan Counter

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Rangkaian Multiplexer, Decoder, Flip-Flop dan Counter  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan Pembelajaran :

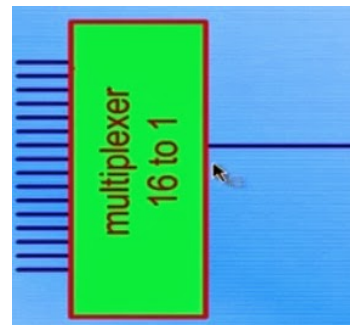
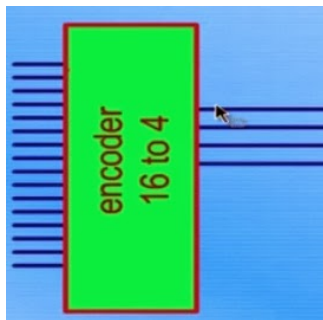
- 1 Siswa mengenal rangkaian multiplexer dan decoder
- 2 Siswa mampu membedakan rangkaian flipflop
- 3 Siswa mengenal rangkaian counter
- 4 Siswa mengenal rangkaian shift register

#### B. Materi Pembelajaran

### Multiplexer dan Demultiplexer

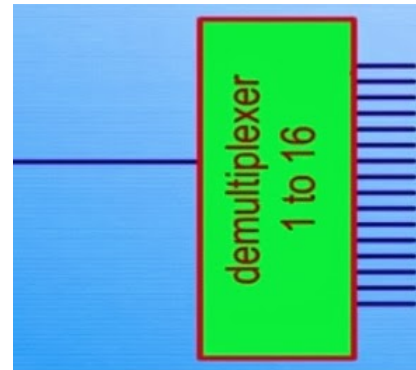
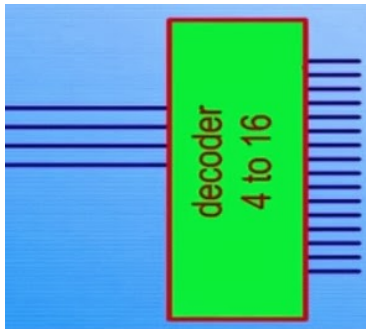
#### 1. Multiplexer

Multiplexer adalah suatu rangkaian yang digunakan untuk memperkecil jumlah output/keluaran. dengan multiplexer beberapa input/masukan dapat diperkecil output nya menjadi satu keluaran. dengan adanya multiplexer maka penggunaan kabel dalam penyaluran informasi dapat ditekan seminimal mungkin, jadi lebih hemat dalam segi biaya. perbedaan multiplexer dengan decoder adalah pada output/keluaran nya. jika pada decoder output/keluaran yang dihasilkan ada beberapa kabel, kalau pada multiplexer output/keluaran nya cukup satu kabel saja.



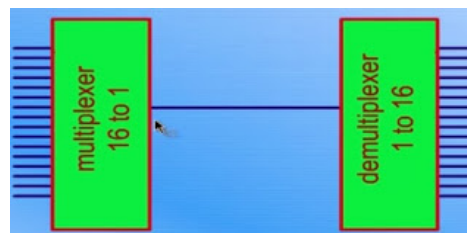
#### 2. Demultiplexer

Demultiplexer atau sering disingkat demux adalah rangkaian rangkain yang digunakan untuk memperbanyak jumlah output/keluaran. pada rangkaian demultiplexer input yang masuk diperbanya output nya menjadi beberapa output. demultiplexer adalah kebalikan dari multiplexer. perbedaan demultiplexer dengan decoder yaitu pada jumlah input nya. kalau pada decoder jumlah input nya terdiri dari beberapa kabel, sedangkan pada demultiplexer input nya hanya satu kabel.

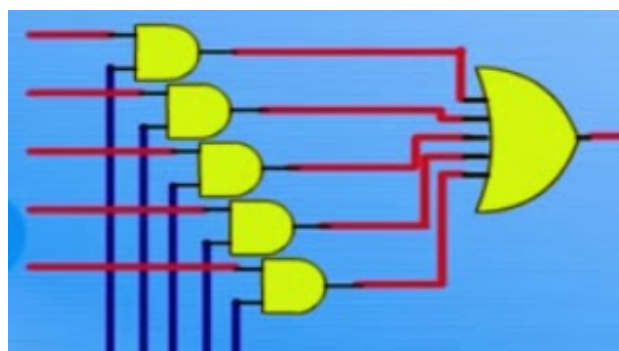


### Cara kerja

Pada rangkaian multiplexer dan demultiplexer terjadi pergantian pengiriman data pada kabel. dari satu kabel dipakai beberapa input, meskipun dengan satu kabel, karena pergantian yang begitu cepat sehingga tidak terasa kalau sedang bergantian dengan pengguna yang lain. pada gambar dibawah, misalkan dari input 1 masuk data, maka akan dikeluarkan di output 1, kemudian gantian dibawah, misalkan dari input 1 masuk data, maka akan dikeluarkan di output 1, kemudian gantian input 2 yang mengirim data, maka data akan dikirim ke output 2, begitu juga seterusnya.

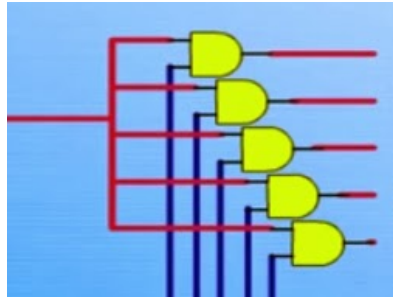


Pada rangkaian multiplexer menggunakan gerbang AND, ada beberapa gerbang AND yang disusun, dan kemudian dihubungkan menjadi satu dengan gerbang OR. pada gerbang AND input yang pertama berfungsi sebagai pengirim data, sedangkan input kedua berfungsi untuk saklar agar bisa bergantian dengan gerbang AND yang lain dalam pengiriman data. pada gambar, jika output gerbang AND yang ke bawah bernilai 1, maka data pada input gerbang AND terkirim.



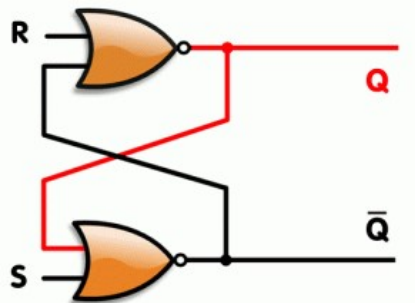
Pada rangkaian demultiplexer, gerbang yang digunakan adalah gerbang AND, output dari multiplexer di cabangkan ke salah satu input-input dari gerbang AND. sedangkan input gerbang yang

satu nya berfungsi sebagai saklar untuk penerima data yang masuk yang kemudian dikeluarkan ke masing-masing output.



**FLIP FLOP**

**Flip-Flop** atau **latch** merupakan [sirkuit](#) elektronik yang memiliki dua arus stabil dan dapat digunakan untuk menyimpan informasi.



Sebuah flip-flop merupakan multivibrator-dwistabil

Sirkuit dapat dibuat untuk mengubah arus dengan sinyal yang dimasukkan pada satu atau lebih input kontrol dan akan memiliki satu atau dua output. Ini merupakan elemen penyimpanan dasar pada *Logika Sekuensial*.

Flip-flop dan latch merupakan bangunan penting dalam sistem elektronik digital yang digunakan pada komputer, komunikasi dan tipe lain dari sistem.

Flip-flop dan latch digunakan sebagai elemen penyimpanan data, seperti penyimpan data yang dapat digunakan untuk menyimpan memori, seperti sirkuit yang dijelaskan pada logika sekuensial.

Flip-flops juga dapat digunakan untuk menghitung detak, dan untuk mengsinkronisasikan input signal waktu variable untuk beberapa signal waktu yang direferensi.

Flip-flop dan *latch* digunakan sebagai elemen penyimpanan data. Penyimpanan data ini digunakan untuk menyimpan *state* (keadaan) pada ilmu komputer, dan sirkuit ini merupakan logika sekuensial.

Sirkuit juga dapat digunakan untuk menghitung bunyi teratur dan sinkronisasi sinyal.

Rangkaian Flip flop adalah rangkaian digital yang digunakan untuk menyimpan satu bit secara semi permanen sampai ada suatu perintah untuk menghapus atau mengganti isi dari bit yang disimpan.

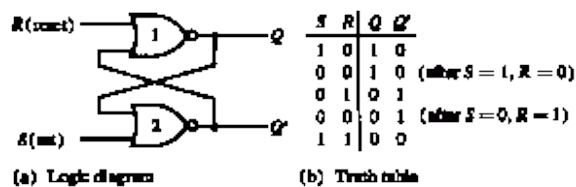
Prinsip dasar dari flip-flop adalah suatu komponen elektronika dasar seperti transistor, resistor dan dioda yang di rangkai menjadi suatu gerbang logika yang dapat bekerja secara sekuensial. Nama lain dari flip-flop adalah multivibrator bistabil.



Multivibrator adalah suatu rangkaian regeneratif dengan dua buah piranti aktif, yang dirancang sedemikian sehingga salah satu piranti bersifat menghantar pada saat piranti lain terpancung.

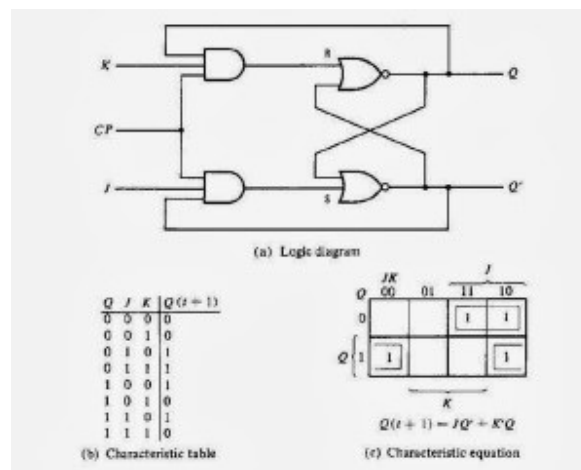
Macam-macam flip-flop :

### 1.) RS flip-flop



Flip-flop RS atau SR (Set-Reset) merupakan dasar dari flip-flop jenis lain. Flip-flop ini mempunyai 2 masukan: satu disebut S (SET) yang dipakai untuk menyetel (membuat keluaran flip-flop berkeadaan 1) dan yang lain disebut R (RESET) yang dipakai untuk me-reset (membuat keluaran berkeadaan 0). Flipflop RS dapat dibentuk dari dua gerbang NOR atau dua gerbang NAND.

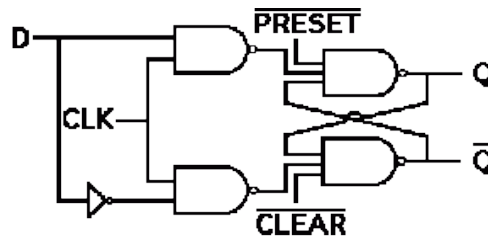
### 2.) JK flip-flop



Flip-flop JK yang diberi nama berdasarkan nama masukannya, yaitu J dan K. Flip-flop ini mengatasi kelemahan flip-flop RS, yang tidak mengizinkan pemberian masukan R=S= 1, dengan meng-AND-kan masukan dari luar dengan keluaran.

Flip-flop JK ini bekerja tak serempak. Untuk memperoleh flip-flop JK yang dapat bekerja serempak dengan rangkaian lain perlu ditambahkan kelengkapan untuk penabuhan (clocking). Ini dapat dilakukan dengan meng-AND-kan pulsa CP (clock Pulse) dengan masukan K dan J seperti yang ditunjukkan pada Gambar 6.8. Perlu dicatat bahwa untuk flip-flop yang peka terhadap perubahan pulsa negatif, pada masukan CP diberikan lingkaran kecil seperti pada NOR dan NAND.

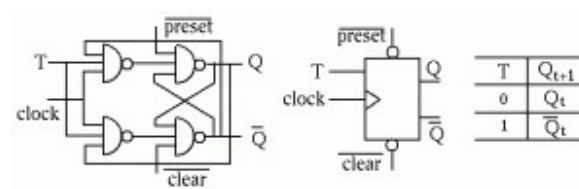
### 3.) D flip-flop



Nama flip-flop ini berasal dari Delay. Flip-flop ini mempunyai hanya satu masukan, yaitu D. Jenis flip-flop ini sangat banyak dipakai sebagai sel memori dalam komputer. Pada umumnya flip-flop ini dilengkapi masukan penabuh.

Dapat dilihat bahwa sebenarnya flip-flop D berfungsi seperti apa yang dilakukan oleh flip-flop JK bila masukan masukan K dihubungkan dengan komplemen masukan J.

#### 4.) T flip-flop

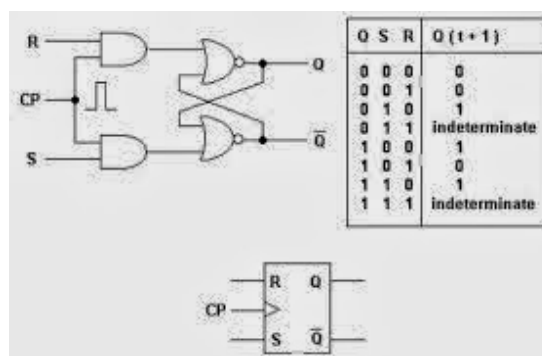


Nama flip-flop T diambil dari sifatnya yang selalu berubah keadaan setiap ada sinyal pemicu (trigger) pada masukannya. Input T merupakan satu-satunya masukan yang ada pada flip-flop jenis ini sedangkan keluarannya tetap dua, seperti semua flip-flop pada umumnya.

Kalau keadaan keluaran flip-flop 0, maka setelah adanya sinyal pemicu keadaan-berikut menjadi 1 dan bila keadaannya 1, maka setelah adanya pemicuan keadaannya berubah menjadi 0. Karena sifat ini sering juga flip-flop ini disebut sebagai flip-flop toggle (berasal dari skalar toggle/pasak).

Flip-flop T dapat disusun dari satu flip-flop RS dan dua gerbang AND.

#### 5.) CRS flip-flop



RS Flip-flop adalah clocked RS-FF yang dilengkapi dengan sebuah terminal pulsa clock. Pulsa clock ini berfungsi mengatur keadaan Set dan Reset. Bila pulsa clock berlogik 0, maka perubahan logik pada input R dan S tidak akan mengakibatkan perubahan pada output Q dan Qnot.

Akan tetapi apabila pulsa clock berlogik 1, maka perubahan pada input R dan S dapat mengakibatkan perubahan pada output Q dan Q not.

### Prinsip kerja Flip-flop:

1. Jika clock bernilai rendah (0) maka flip-flop J-K master akan tidak aktif, tetapi karena input clock flip-flop J-K slave merupakan komplemen dari clock flip-flop master maka flip-flop slave menjadi aktif, dan outputnya mengikuti output flip-flop J-K master.
2. Jika clock bernilai tinggi (1), flip-flop master aktif sehingga outputnya tergantung pada input J dan K, pada sisi lain flip-flop slave menjadi tidak aktif karena clock pemicunya bernilai rendah (0).
3. Pada saat sinyal detak berada pada tingkat tinggi, master-nya yang aktif dan slave-nya tidak aktif.
4. Pada saat sinyal detak berada pada tingkat rendah, master-nya yang tidak aktif dan slave-nya yang aktif.
5. Jika input J diberikan bersama-sama dengan tepi naik pulsa pemicu, flip-flop *master* akan bekerja terlebih dahulu memantapkan inputnya selama munculnya tepi naik sampai *clock* bernilai rendah (0).
6. Setelah *clock* bernilai rendah (0), flip-flop *master* akan tidak aktif dan flip-flop *slave* bekerja menransfer keadaan output flip-flop *master* ke output flip-flop *slave* yang merupakan output flip-flop secara keseluruhan.

## SHIFT REGISTER

Register adalah sekelompok flip-flop yang dapat dipakai untuk menyimpan dan untuk mengolah informasi dalam bentuk linier. Ada 2 jenis utama Register yaitu :

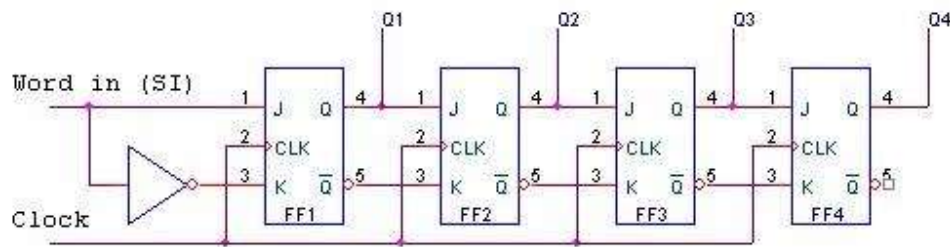
1. Storage Register (register penyimpan)
2. Shift Register (register geser)

Register penyimpan (Storage Register) digunakan apabila kita hendak menyimpan informasi untuk sementara, sebelum informasi itu dibawa ke tempat lain. Banyaknya kata/bit yang dapat disimpan, tergantung dari banyaknya flip-flop dalam register. Satu flip-flop dapat menyimpan satu bit. Bila kita hendak menyimpan informasi 4 bit maka kita butuhkan 4 flip-flop. Contoh: Register yang mengingat bilangan duaan (biner): 1101 terbaca pada keluaran Q.

Shift Register adalah suatu register dimana informasi dapat bergeser (digeserkan). Dalam register geser flip-flop saling dikoneksi, sehingga isinya dapat digeserkan dari satu flip-flop ke flip-flop yang lain, ke kiri atau ke kanan atas perintah denyut lonceng (Clock). Dalam alat ukur digit, register dipakai untuk mengingat data yang sedang ditampilkan.

Ada 4 Shift Register yaitu:

### 1. SISO (Serial Input Serial Output)



Gambar Register SISO yang menggunakan JK FF

### Prinsip kerja:

Informasi/data dimasukan melalui word in dan akan dikeluarkan jika ada denyut lonceng berlalu dari 1 ke 0. Karena jalan keluarnya flip-flop satu dihubungkan kepada jalan masuk flip-flop berikutnya, maka informasi didalam register akan digrser ke kanan selama tebing dari denyut lonceng (Clock).

Tabel Kebenaran (Misal masuknya 1101)

Clock ke Word in Q1 Q2 Q3 Q4

0	0	0	0	0
1	1	1	0	0
2	0	0	1	0
3	1	1	0	1
4	1	1	0	1

Register geser SISO ada dua macam yaitu:

- Shift Right Register (SRR)/Register geser kanan
- Shift Left Register (SLR)/Register geser kiri
- Shift Control Register dapat berfungsi sebagai SSR maupun SLR

Rangkaian Shift control adalah sebagi berikut:

Rangkaian ini untuk mengaktifkan geser kanan/kiri yang ditentukan oleh SC. Jika SC=1, maka akan mengaktifkan SLR. Jika SC=0, maka akan mengaktifkan SRR.

Keterangan:

Jika SC=0, maka input geser kanan akan aktif. Keluaran NAND diumpamakan ke input DFF1 dan setelah denyut lonceng berlaku (saat tebing depan), maka informasi diteruskan ke output Q1. Dan output Q1 terhubung langsung ke output DFF2 berikutnya sehingga dengan proses ini terjadi pergeseran ke kanan.

TABEL KEBENARAN (jika input 1101)

Clock ke Input Q1 Q2 Q3 Q4

0	0	0	0	0
1	1	1	0	0
2	1	1	1	0
3	0	0	1	1
4	1	1	0	1

Informasi digit digeser kekanan setiap ada perubahan pulsa clock tebing atas. Geser kanan berfungsi sebagai operasi aritmatika yaitu pembagi dua untuk tiap-tiap flip-flop.

Jika SC = 1, maka akan mengaktifkan input geser kiri. Output NAND masuk ke input D-FF4 dan setelah diberi pulsa clock informasi dikeluarkan melalui Q4 dan keluaran Q4 dihubungkan ke input D-FF3, keluaran D-FF3 dimasukan ke D-FF berikutnya, sehingga dengan demikian terjadi pergeseran informasi bit ke arah kiri.

TABEL KEBENARAN (jika input 1101)

Clock ke Input Q1 Q2 Q3 Q4

0 0 0 0 0

1 1 0 0 1

2 1 0 0 1

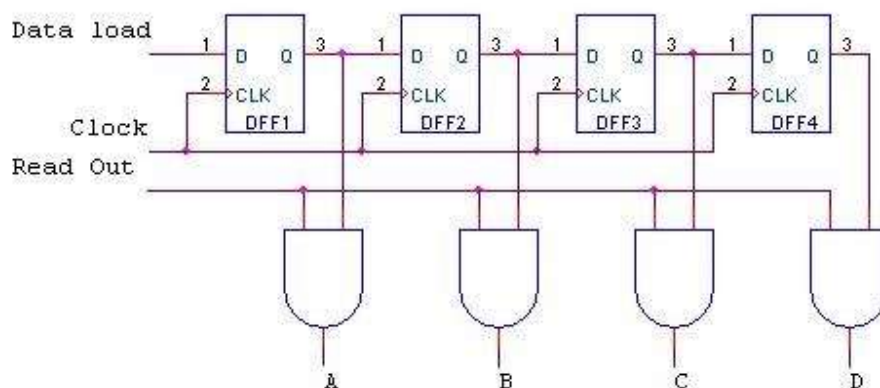
3 0 0 1 1 0

4 1 1 1 0 1

Register geser kiri berfungsi sebagai operasi aritmatika yaitu sebagai pengali dua untuk tiap-tiap flip-flop.

## 2. Register Geser SIPO

Adalah register geser dengan masukan data secara serial dan keluaran data secara paralel.



Gambar rangkaian SIPO menggunakan D-FF

### Cara kerja:

Masukan-masukan data secara deret akan dikeluarkan oleh D-FF setelah masukan denyut lonceng dari 0 ke 1. Keluaran data/informasi serial akan dapat dibaca secara paralel setelah diberikan satu komando (Read Out). Bila di jalan masuk Read Out diberi logik 0, maka semua keluaran AND adalah 0 dan bila Read Out diberi logik 1, maka pintu-pintu AND menghubungkan langsung sinyal-sinyal yang ada di Q masing-masing flip-flop.

Contoh: Bila masukan data 1101

TABEL KEBENARANNYA:

Read Out Clock Input Q1 Q2 Q3 Q4 A B C D

0 0 0 0 0 0 0 0 0 0

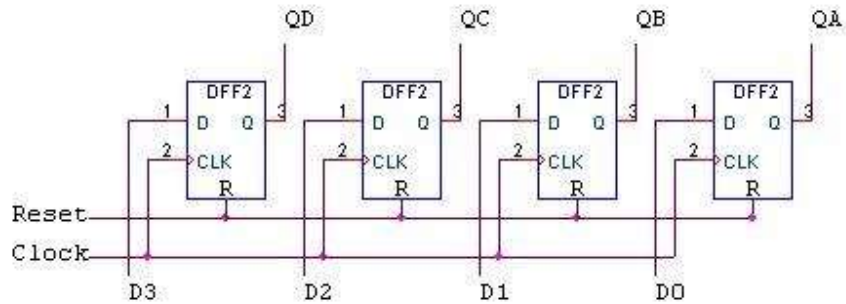
0 1 1 1 0 0 0 0 0 0

0 2 1 1 1 0 0 0 0 0

0 3 0 0 1 1 0 0 0 0  
0 4 1 1 0 1 1 0 0 0  
1 1 0 1 1 1 0 1 1

### 3. Register Geser PIPO

Adalah register geser dengan masukan data secara jajar/paralel dan keluaran jajar/paralel.



Gambar rangkaian PIPO menggunakan D-FF

#### Cara kerja:

Sebelum dimasuki data rangkaian direset dulu agar keluaran Q semuanya 0. Setelah itu data dimasukkan secara paralel pada input D-FF dan data akan diloloskan keluar secara paralel setelah flip-flop mendapat pulsa clock dari 0 ke 1.

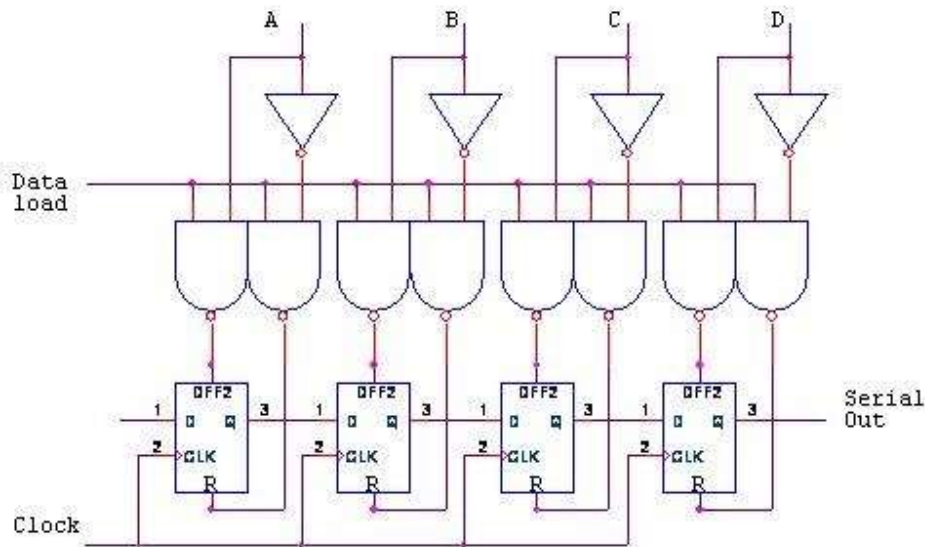
Contoh:

TABEL KEBENARAN:

Clock	D1	D2	D3	D4	QD	QC	QB	QA
0	1	1	0	1	0	0	0	0
1	1	1	0	1	1	1	0	1
2	1	0	0	1	1	0	0	1
3	0	0	0	1	0	0	0	1

### 4. Register geser PISO

Adalah register geser dengan masukan data secara paralel dan dikeluarkan secara deret/serial.



Gambar rangkaian register PISO menggunakan D-FF

Rangkaian diatas merupakan register geser dengan panjang kata 4 bit. Semua jalan masuk clock dihubungkan jajar. Data-data yang ada di A, B, C, D dimasukkan ke flip-flop secara serempak, apabila dijalan masuk Data Load diberi logik 1.

### Cara Kerja:

Mula-mula jalan masuk Data Load = 0, maka semua pintu NAND mengeluarkan 1, sehingga jalan masuk set dan reset semuanya 1 berarti bahwa jalan masuk set dan reset tidak berpengaruh.

Jika Data Load = 1, maka semua input paralel akan dilewatkan oleh NAND. Misal jalan masuk A=1, maka pintu NAND 1 mengeluarkan 0 adapun pintu NAND 2 mengeluarkan 1. Dengan demikian flip-flop diset sehingga menjadi Q=1. Karena flip-flop yang lainpun dihubungkan dengan cara yang sama, maka mereka juga mengoper informasi pada saat Data Load diberi logik 1. Setelah informasi berada didalam register, Data Load diberi logik 0. Informasi akan dapat dikeluarkan dari register dengan cara memasukkan denyut lonceng, denyut-demi denyut keluar deret/seri. Untuk keperluan ini jalan masuk D dihubungkan kepada keluaran Q.

Ada juga register yang dapat digunakan sebagai Shift register SISO maupun PIPO dengan bantuan suatu control sbb:

Input Control = 0, berfungsi sebagai register geser SISO

Input Control = 1, berfungsi sebagai register geser PIPO

Data IC Preset Reset

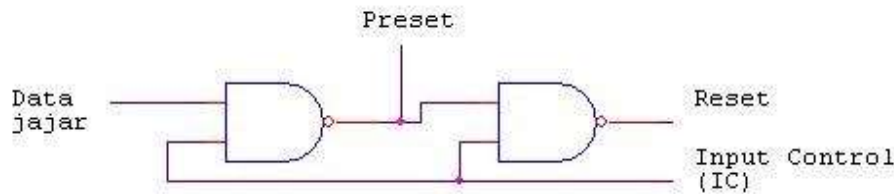
0 1 1 0

1 1 0 1

0 0 1 1

1 0 1 1

Rangkaian kontrol diatas dapat disimbolkan sbb:



Catatan:

Jika  $IC=0$ , maka input yang dimasukkan ke D0, D1, D2, D3 tidak mempengaruhi keadaan output QA, QB, QC, QD tetapi yang mempengaruhinya adalah data yang dimasukkan ke input D-FF secara serial, maka pada kondisi ini rangkaian akan bekerja sebagai register geser SISO.

Jika  $IC=1$ , maka input yang dimasukkan ke gate D seri tidak akan mempengaruhi output, tetapi output dipengaruhi oleh data paralel (D0, D1, D2, D3). Input dimasukkan secara serempak dan keluaran ditunjukkan secara serempak begitu pulsa clock berguling dari 1 ke 0, maka pada kondisi ini rangkaian akan bekerja sebagai register geser PIPO.

## COUNTER

*Counters* (pencacah) adalah alat/rangkaian digital yang berfungsi menghitung/mencacah banyaknya pulsa *clock* atau juga berfungsi sebagai pembagi frekuensi, pembangkit kode biner, *Gray*. Ada 2 jenis pencacah yaitu:

1. Pencacah sinkron (*synchronous counters*) atau pencacah jajar.
2. Pencacah tak sinkron (*asynchronous counters*) yang kadang-kadang disebut juga pencacah deret (*series counters*) atau pencacah kerut (*ripple counters*).

Karakteristik penting daripada pencacah adalah:

1. Kerjanya sinkron atau tak sinkron.
2. Mencacah maju atau mundur sampai beberapa banyak ia dapat mencacah (modulo pencacah).
3. Dapat berjalan terus (*free running*) atautkah dapat berhenti sendiri (*self stopping*)

Langkah-Langkah dalam merancang pencacah adalah menentukan:

1. Karakteristik pencacah (tersebut diatas).
2. Jenis flip-flop yang diperlukan/digunakan (D-FF, JK FF atau RS-FF).
3. Prasyarat perubahan logikanya (dari flip-flop yang digunakan).

### a) Pencacah Tak Sinkron

Dianamai pencacah tak sinkron (*asynkronuous counters*) atau *ripple through counters*, sebab flip-flop nya bergulingan secara tak serempak tetapi secara berurutan. Hal ini disebabkan karena hanya flip-flop yang paling ujung saja yang dikendalikan oleh sinyal clock untuk flip-flop lainnya diambilkan dari masing-masing flip-flop sebelumnya. Banyaknya denyut yang dimasukkan diterjemahkan oleh flip-flop kedalam bentuk biner. Itulah sebabnya pencacah tak sinkron disebut juga



pencacah biner. Pada pencacah tak sinkron penundaan adalah sama dengan penundaan-penundaan flip-flop dijumlahkan.

Ada dua macam pencacah yaitu pencacah sinkron dan asinkron. Pencacah sinkron terdiri dari 4 macam yaitu:

- 1) Pencacah maju sinkron yang berjalan terus (*Free Running*).
- 2) Pencacah maju sinkron yang dapat berhenti sendiri (*Self Stopping*).
- 3) Pencacah mundur sinkron.
- 4) Pencacah maju dan mundur sinkron (*Up-down Counter*).

Pencacah tak sinkron terdiri dari 4 macam yaitu:

- 1) Pencacah maju taksinkron yang berjalan terus (*Free Running*).
- 2) Pencacah maju taksinkron yang dapat berhenti sendiri (*Self Stopping*).
- 3) Pencacah mundur tak sinkron.
- 4) Pencacah maju dan mundur tak sinkron (*Up-down Counter*).

Macam-macam penggunaan pencacah:

- 1) Penggunaan pencacah dalam teknologi industri. Dalam hal ini pencacah dioperasikan untuk menghitung obyek (barang produksi) dengan tujuan untuk mencapai kecepatan dan kecermatan penghitungan.
- 2) Digunakan sebagai pembagi frekuensi.
- 3) Untuk mengukur besarnya frekuensi.
- 4) Untuk mengukur waktu interval antara dua pulsa.
- 5) Untuk mengukur jarak.
- 6) Untuk mengukur kecepatan.
- 7) Penggunaan dalam digital komputer.
- 8) Untuk mengubah sinyal analog menjadi digital (*Analog to Digital Converters/ADC*) maupun untuk mengubah sinyal digital ke analog (*Digital to Analog Converter/DAC*).

- Pencacah maju tak sinkron

Dasar dari pencacah ini adalah JK-FF yang dioperasikan sebagai T-FF (JK-FF dalam kondisi *toggle*) yaitu dimana kedua input J dan K diberi nilai logika “1”. Dan dalam keadaan demikian JK-FF akan berfungsi sebagai pembagi dua. Atau dengan kata lain, frekuensi output JK-FF tersebut sama dengan setengah frekuensi clock yang diberikan.

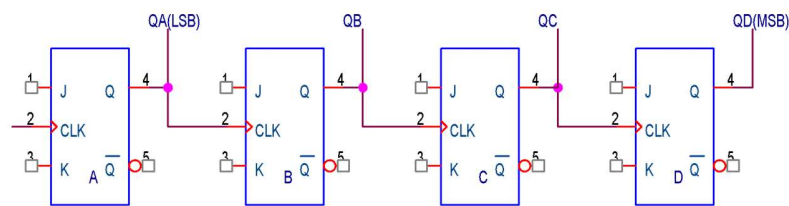
Rumus frekuensi output flip-flop dalam kondisi ini adalah:

$$F_{\text{output}} = 1/2^n \times F_{\text{in}}$$

$$= \frac{\text{Frekuensi input pulsa clock}}{2^n}$$

(n = banyaknya toggle flip-flop yang dipakai)

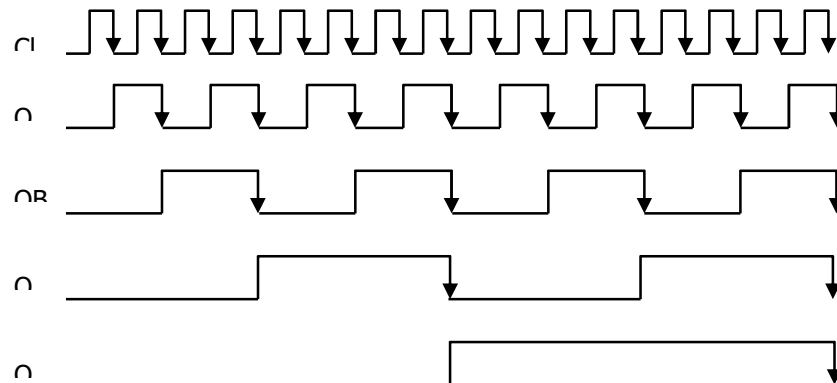
Rangkaian berikut merupakan pencacah maju tak sinkron yang menggunakan 4 buah JK-FF:



Cara kerja rangkaian diatas adalah sebagai berikut:

- Output flip-flop yang pertama (QA) akan berguling (menjadi 0 atau 1) setiap pulsa clock pada sisi negatif/*trailing edge* atau dari kondisi 1 ke 0.
- Output flip-flop yang lainnya akan berguling bila dan hanya bila output flip-flop sebelumnya berganti kondisi dari 1 ke 0 (sisi negatif/*trailing edge*) juga.

Diagram waktu/timing diagram rangkaian tersebut adalah sebagai berikut:



Dari diagram waktu diatas dapat dilihat dengan jelas bahwa QA berguling setiap kali pulsa clock pada sisi negatifnya. QB berguling setiap kali sisi negatif dari QA. QC berguling setiap kali sisi negatif dari QB dan QD bergulingan setiap kali sisi negatif dari QC. Dan karena masing-masing flip-flop berfungsi sebagai pembagi dua, maka frekuensi masing-masing outpunya adalah:

$$QA = \frac{1}{2} \text{ frekuensi sinyal clock.}$$

$$QB = \frac{1}{2} \text{ frekuensi QA} = \frac{1}{4} \text{ frekuensi sinyal clock.}$$

$QC = \frac{1}{2}$  frekuensi  $QB = \frac{1}{8}$  frekuensi sinyal clock.

$QD = \frac{1}{2}$  frekuensi  $QC = \frac{1}{16}$  frekuensi sinyal clock.

Dengan demikian didapat suatu pembagi  $2^n = 16$  ( $n$  = banyaknya flip-flop), yaitu dengan melihat frekuensi output flip-flop terakhir.

Dari diagram waktu diatas dapat dibuat tabel kebenaran sebagai berikut:

Clock	QD MSB	QC	QB	QA LSB	Desimal
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15

Pecacah diatas dapat mencacah dari bilangan biner 0000 sampai dengan 1111 (dari 0 sampai 15 desimal). Pencacah tersebut merupakan pencacah 16 modulus (modulo 16 counters).

#### 1) Pencacah mundur tak sinkron

Dari pencacah maju dapat kita buat menjadi pencacah mundur dengan cara yang dibaca bukan keluaran Q melainkan keluaran **Qnot** atau dengan cara output **Qnot** sebagai masukan clock pada flip-flop berikutnya. Gambar rangkaiannya adalah sebagai berikut:

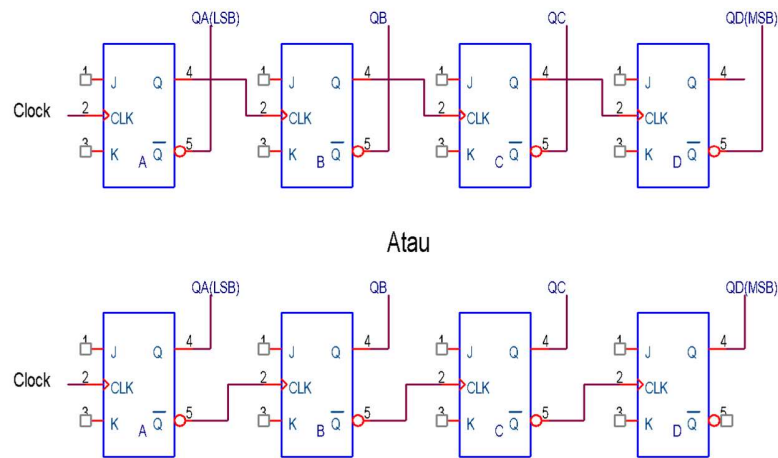
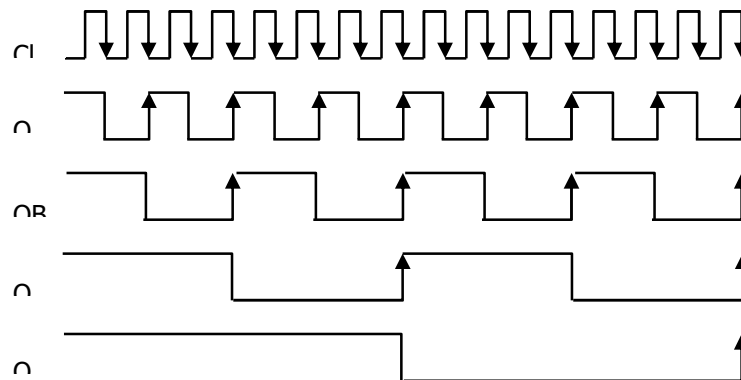


Diagram waktu/timing diagram dari rangkaian tersebut adalah sebagai berikut:



Selanjutnya dari diagram waktu tersebut dapat dibuat tabel kebenaran seperti berikut:

Clock	QD	QC	QB	QA	Desimal
0	1	1	1	1	15
1	1	1	1	0	14
2	1	1	0	1	13
3	1	1	0	0	12
4	1	0	1	1	11
5	1	0	1	0	10
6	1	0	0	1	9
7	1	0	0	0	8
8	0	1	1	1	7
9	0	1	1	0	6
10	0	1	0	1	5
11	0	1	0	0	4
12	0	0	1	1	3

13	0	0	1	0	2
14	0	0	0	1	1
15	0	0	0	0	0
16	1	1	1	1	15

Pecacah diatas dapat mencacah mundur dari bilangan biner 1111 sampai dengan 0000 (atau 15 s/d 0 dasan).

## **MODUL 6**

### **Pengantar Organisasi dan Arsitektur Komputer**

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Pengantar Organisasi dan Arsitektur Komputer  
**Alokasi waktu** : Jam Pelajaran

#### **A. Tujuan Pembelajaran :**

- 1 Siswa mengenal Pengertian dan perbedaan organisasi dan arsitektur komputer
- 2 Siswa mengenal Struktur dan fungsi utama
- 3 Siswa mengenal Konsep dasar operasi computer dan Struktur mesin Von Neumann
- 4 Siswa mampu mengenal Sejarah perkembangan teknologi sistem komputer dari generasi ke generasi

#### **B. Materi Pembelajaran**

Komputer merupakan perangkat elektronik yang sudah tidak asing lagi di kalangan anak-anak hingga orang dewasa, tentunya dengan tingkat pemahaman dan penggunaan yang berbeda-beda. Arsitektur Von Neumann menggambarkan komputer dengan empat bagian utama: Unit Aritmatika dan Logis (ALU), unit kontrol, memori, dan alat masukan dan hasil (secara kolektif dinamakan I/O). Bagian ini dihubungkan oleh berkas kawat, “bus”.

##### **1. Organisasi Komputer**

Organisasi Komputer adalah bagian yang terkait erat dengan unit-unit operasional dan interkoneksi antar komponen penyusun sistem komputer dalam merealisasikan aspek arsitekturalnya. Contoh aspek organisasional adalah teknologi hardware, perangkat antarmuka, teknologi memori, sistem memori, dan sinyal–sinyal kontrol.

##### **2. Arsitektur Komputer**

Arsitektur Komputer lebih cenderung pada kajian atribut–atribut sistem komputer yang terkait dengan seorang programmer. Contohnya, set instruksi, aritmetika yang digunakan, teknik pengalamatan, mekanisme I/O.

##### **3. CPU (Central Processing Unit)**

Unit Pengolah Pusat atau CPU (*Central processing Unit*) berperan untuk memproses perintah yang diberikan oleh pengguna komputer, mengelolanya bersama data-data yang ada di komputer. Unit atau peranti pemrosesan juga akan berkomunikasi dengan peranti input , output dan storage untuk melaksanakan instruksi yang saling terkait.

Dalam arsitektur von Neumann yang asli, ia menjelaskan sebuah Unit Aritmatika dan Logika, dan sebuah Unit Kontrol. Dalam komputer-komputer modern, kedua unit ini terletak dalam satu sirkuit terpadu (IC – *Integrated Circuit*), yang biasanya disebut CPU (*Central Processing Unit*).

**Unit Aritmatika dan Logika**, atau *Arithmetic Logic Unit* (ALU), adalah alat yang melakukan pelaksanaan dasar seperti pelaksanaan aritmatika (tambahan, pengurangan, dan semacamnya), pelaksanaan logis (AND, OR, NOT), dan pelaksanaan perbandingan (misalnya, membandingkan isi sebanyak dua slot untuk kesetaraan). Pada unit inilah dilakukan “kerja” yang sebenarnya.

**Unit kontrol** menyimpan perintah saat ini yang dilakukan oleh komputer, memerintahkan ALU untuk melaksanakan dan mendapatkan kembali informasi (dari memori) yang diperlukan untuk melaksanakan perintah itu, dan memindahkan kembali hasil ke lokasi memori yang sesuai. Unit ini berfungsi mengontrol pembacaan instruksi program komputer.

#### 4. Memori

Memori adalah sebuah *array* yang besar dari *word* atau *byte*, yang ukurannya mencapai ratusan, ribuan, atau bahkan jutaan. Setiap *word* atau *byte* mempunyai alamat tersendiri. *Main memory* berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. *Main-memory* termasuk tempat penyimpanan data yang sementara (*volatile*), artinya data dapat hilang begitu sistem dimatikan. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen memori seperti: menjaga track dari memori yang sedang digunakan dan siapa yang menggunakannya; memilih program yang akan di-load ke memori; dan mengalokasikan dan mendealokasikan *memoryspace* sesuai kebutuhan. *Main memory* dapat dibayangkan sebagai kumpulan kotak-kotak yang masing dapat menyimpan suatu penggal informasi baik berupa data maupun instruksi. Umumnya 1 *byte memory* terdiri dari 8 bit dan tiap bit diwakili oleh 1 atau 0. Kombinasi bit dalam 1 byte tersebut membentuk suatu kode yang mewakili isi dari lokasi memori. Kode yang digunakan untuk mewakilinya tergantung dari komputer yang digunakan, dapat membentuk sistem kode BCD (*Binary-Coded Decimal*), sistem kode SBCDIC (*Standard Binary Coded Decimal Interchange Code*), sistem kode EBCDIC (*Extended Binary Coded Decimal Interchange Code*) atau sistem kode ASCII (*American Standard Code for Information Interchange*).

Memori dikelompokkan menjadi 2 yaitu:

##### **RAM (*Random Access Memory*)**

RAM (*Random Access Memory*) adalah memori yang dapat dibaca atau ditulisi. Data dalam sebuah RAM bersifat *volatile*, artinya data akan terhapus bila catu daya dihilangkan. Karena sifat RAM yang *volatile* ini, maka program komputer tidak tersimpan di RAM. RAM hanya digunakan untuk menyimpai data sementara, yang tiadak begitu vital saat aliran daya terputus.

Struktur dari RAM, dibagi menjadi:

1. *Input Storage*, digunakan untuk menampung input yang dimasukkan lewat alat input;
2. *Program Storage*, digunakan untuk menyimpan semua instruksi program yang akan diproses;
3. *Working Storage*, digunakan untuk menyimpan data yang akan diolah dan hasil dari pengolahan;

4. *Output Storage*, digunakan untuk menampung hasil akhir dari pengolahan data yang akan ditampilkan ke alat output.

### ROM (*Read Only Memory*)

Memori ini hanya dapat dibaca saja, programmer tidak bisa mengisi sesuatu ke dalam ROM. Isi ROM sudah diisi oleh pabrik pembuatnya berupa sistem operasi yang terdiri dari program-program pokok yang diperlukan oleh sistem komputer, seperti program untuk mengatur penampilan karakter, pengisian tombol kunci dan *bootstrap program*.

*Bootstrap* program diperlukan pada waktu pertama kali sistem komputer diaktifkan, yang proses ini disebut dengan istilah *booting*, yang terdiri dari:

- *Cold booting*, yaitu proses mengaktifkan sistem komputer pertama kali untuk mengambil *bootstrap* program dari keadaan listrik komputer mati.
- *Warm booting*, yaitu proses pengulangan pengambilan *bootstrap* program dalam keadaan komputer masih hidup.

Instruksi yang tersimpan di ROM disebut dengan *microinstruction* atau *microcode* atau disebut juga *firmware*. Isi dari ROM tidak boleh hilang atau rusak, karena dapat menyebabkan sistem komputer tidak berfungsi. ROM bersifat *non volatile*, artinya isinya tidak hilang bila listrik komputer dimatikan.

Jenis-jenis ROM:

- 1) **PROM** (*Programmable Read Only Memory*), yaitu ROM yang dapat diprogram sekali saja dan tidak dapat diubah kembali
- 2) **EPROM** (*Erasable Programmable Read Only Memory*), yaitu ROM yang dapat dihapus dengan sinar ultra violet serta dapat diprogram kembali berulang-ulang
- 3) **EEPROM** (*Electrically Erasable Programmable Read Only Memory*), yaitu ROM yang dapat dihapus secara elektronik dan dapat diprogram kembali.

### 5. I/O Port

Alat-alat input/output tidak dilekatkan langsung dengan bus tetapi melalui suatu I/O port atau *I/O interface*. Alat-alat input/output dapat berkomunikasi dengan CPU dengan cara mengirimkan informasi yang akan dikomunikasikan lewat bus. Informasi yang dikirim dari alat input/output (*peripheral device*) ke *main memory* atau ke register di CPU diletakkan di I/O port dan dikirimkan lewat data bus. Demikian juga bila informasi dari *main memory* akan dikirimkan ke *peripheral device* juga melalui data bus dan diterima di I/O port. Cara ini disebut juga dengan *program-controlled I/O*. Cara ini banyak diterapkan pada alat I/O yang hanya dapat menangani satu karakter atau 1 byte atau 1 word saja tiap saat misalnya keyboard.

### 6. Instruksi

Perintah yang dibicarakan di atas bukan perintah seperti bahasa manusiawi. Komputer hanya mempunyai perintah sederhana dalam jumlah terbatas yang dirumuskan dengan baik. Perintah biasa yang dipahami kebanyakan komputer ialah “menyalin isi sel 123, dan tempat tiruan di sel 456”,



“menambahkan isi sel 666 ke sel 042, dan tempat akibat di sel 013”, dan “jika isi sel 999 adalah 0, perintah berikutnya anda di sel 345”.

Instruksi diwakili dalam komputer sebagai nomor – kode untuk “menyalin” mungkin menjadi 001, misalnya. Suatu himpunan perintah khusus yang didukung oleh komputer tertentu diketahui sebagai bahasa mesin komputer. Dalam praktiknya, orang biasanya tidak menulis perintah untuk komputer secara langsung di bahasa mesin tetapi memakai bahasa pemrograman ”tingkat tinggi” yang kemudian diterjemahkan ke dalam bahasa mesin secara otomatis oleh program komputer khusus (interpreter dan kompiler). Beberapa bahasa pemrograman berhubungan erat dengan bahasa mesin, seperti *assembler* (bahasa tingkat rendah); di sisi lain, bahasa seperti prolog didasarkan pada prinsip abstrak yang jauh dari detail pelaksanaan sebenarnya oleh mesin (bahasa tingkat tinggi).

## 7. Pengalamatan

Pengalamatan adalah bagaimana cara menunjuk dan mengalami suatu lokasi memori pada sebuah alamat di mana *operand* akan diambil. Mode pengalamatan diterapkan pada set instruksi, dimana pada umumnya instruksi terdiri dari *opcode* (kode operasi) dan alamat. Setiap mode pengalamatan memberikan fleksibilitas khusus yang sangat penting. Mode pengalamatan ini meliputi *direct addressing*, *indirect addressing*, dan *immediate addressing*.

### a) *Direct Addressing*

Dalam mode pengalamatan *direct addressing*, harga yang akan dipakai diambil langsung dalam alamat memori lain. Contohnya: MOV A,30h. Dalam instruksi ini akan dibaca data dari RAM internal dengan alamat 30h dan kemudian disimpan dalam akumulator. Mode pengalamatan ini cukup cepat, meskipun harga yang didapat tidak langsung seperti immediate, namun cukup cepat karena disimpan dalam RAM internal. Demikian pula akan lebih mudah menggunakan mode ini daripada mode immediate karena harga yang didapat bisa dari lokasi memori yang mungkin variabel.

Kelebihan dan kekurangan dari *Direct Addressing* antara lain : Kelebihannya adalah *Field* alamat berisi alamat efektif sebuah *operand*. Kelemahannya adalah keterbatasan *field* alamat karena panjang *field* alamat biasanya lebih kecil dibandingkan panjang *word*.

### b) *Indirect Addressing*

Mode pengalamatan *indirect addressing* sangat berguna karena dapat memberikan fleksibilitas tinggi dalam mengalami suatu harga. *Mode ini pula satu-satunya cara untuk mengakses 128 byte lebih dari RAM internal pada keluarga 8052*. Contoh: MOV A,@R0. Dalam instruksi tersebut, 89C51 akan mengambil harga yang berada pada alamat memori yang ditunjukkan oleh isi dari R0 dan kemudian mengisikannya ke akumulator. Mode pengalamatan *indirect addressing* selalu merujuk pada RAM internal dan tidak pernah merujuk pada SFR. Karena itu, menggunakan mode ini untuk mengalami alamat lebih dari 7Fh hanya digunakan untuk keluarga 8052 yang memiliki 256 byte spasi RAM internal.

Kelebihan dan kekurangan dari *Indirect Addressing* antara lain : Kelebihannya adalah ruang bagi alamat menjadi besar sehingga semakin banyak alamat yang dapat referensi. Kekurangannya adalah diperlukan referensi memori ganda dalam satu *fetch* sehingga memperlambat preoses operasi.

### c) *Immediate Addressing*

Mode pengalamatan *immediate addressing* sangat umum dipakai karena harga yang akan disimpan dalam memori langsung mengikuti kode operasi dalam memori. Dengan kata lain, tidak diperlukan pengambilan harga dari alamat lain untuk disimpan. Contohnya: MOV A,#20h. Dalam instruksi tersebut, akumulator akan diisi dengan harga yang langsung mengikutinya, dalam hal ini 20h. Mode ini sangatlah cepat karena harga yang dipakai langsung tersedia.

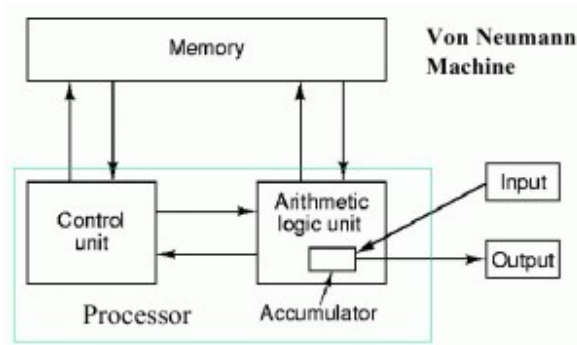
Kelebihan dari *Immediate Addressing* antara lain :

- Tidak adanya referensi memori selain dari instruksi yang diperlukan untuk memperoleh operand.
- Menghemat siklus instruksi sehingga proses keseluruhan akan cepat.

Kekurangan dari *Immediate Addressing* antara lain :

- Ukuran bilangan dibatasi oleh ukuran *field* alamat.

## 8. Mesin Von Neumann



Mesin Von Neumann adalah arsitektur yang diciptakan oleh John von Neumann (1903-1957). Arsitektur ini digunakan oleh hampir semua komputer saat ini.

Arsitektur Von Neumann menggambarkan komputer dengan empat bagian utama: Unit Aritmatika dan Logis (ALU), unit kontrol, memori, dan alat masukan dan hasil (secara kolektif dinamakan I/O). Bagian ini dihubungkan oleh berkas kawat, “bus”

### Cara kerja dari mesin Von Neumann :

#### a) Pemrosesan dari mesin Von Neumann

Unit Pemroses Pusat atau CPU ( central processing unit) berperan untuk memproses arahan, melaksanakan pengiraan dan menguruskan informasi melalui system komputer. Unit atau peranti pemrosesan juga akan berkomunikasi dengan peranti input , output dan storan bagi melaksanakan arahan-arahan berkaitan.

Arsitektur von Neumann menjelaskan sebuah Unit Aritmatika dan Logika, dan sebuah Unit Kontrol. Dalam komputer-komputer modern, kedua unit ini terletak dalam satu sirkuit terpadu (IC – Integrated Circuit), yang biasanya disebut CPU (Central Processing Unit).

Unit Aritmatika dan Logika, atau Arithmetic Logic Unit (ALU), adalah alat yang melakukan pelaksanaan dasar seperti pelaksanaan aritmatika (tambahan, pengurangan, dan semacamnya),

pelaksanaan logis (AND, OR, NOT), dan pelaksanaan perbandingan (misalnya, membandingkan isi sebanyak dua slot untuk kesetaraan).

Unit kontrol menyimpan perintah sekarang yang dilakukan oleh komputer, memerintahkan ALU untuk melaksanakan dan mendapat kembali informasi (dari memori) yang diperlukan untuk melaksanakan perintah itu, dan memindahkan kembali hasil ke lokasi memori yang sesuai. Sekali yang terjadi, unit kontrol pergi ke perintah berikutnya (biasanya ditempatkan di slot berikutnya, kecuali kalau perintah itu adalah perintah lompatan yang memberitahukan kepada komputer bahwa perintah berikutnya ditempatkan di lokasi lain).

#### **b) Input dan hasil dari mesin Von Neumann**

I/O membolehkan komputer mendapatkan informasi dari dunia luar, dan menaruh hasil kerjanya di sana, dapat berbentuk fisik (hardcopy) atau non fisik (softcopy). Yang dimiliki oleh semua alat masukan biasa ialah bahwa mereka meng-encode (mengubah) informasi dari suatu macam ke dalam data yang bisa diolah lebih lanjut oleh sistem komputer digital. Dalam pengertian diatas sistem komputer digital adalah contoh sistem pengolah data.

#### **Kelebihan dan kekurangan dari sistem Von Neumann**

Kekurangan dari mesin von neumann diantara nya adalah :

1. *bus* tunggalnya itu sendiri. Sehingga instruksi untuk mengakses program dan data harus dijalankan secara sekuensial dan tidak bisa dilakukan *overlapping* untuk menjalankan dua instruksi yang berurutan.
2. Selain itu *bandwidth* program harus sama dengan *bandwidth* data. Jika memori data adalah 8 bits maka program juga harus 8 bits. Satu instruksi biasanya terdiri dari *opcode* (instruksinya sendiri) dan diikuti dengan *operand* (alamat atau data). Karena memori program terbatas hanya 8 bits, maka instruksi yang panjang harus dilakukan dengan 2 atau 3 bytes. Misalnya byte pertama adalah *opcode* dan byte berikutnya adalah *operand*.
3. Secara umum prosesor Von Neumann membutuhkan jumlah *clock* CPI (*Clock per Instruction*) yang relatif lebih banyak dan walhasil eksekusi instruksi dapat menjadi relatif lebih lama.

Kelebihan dari mesin Von Neumann adalah :

1. Dengan arsitektur Von Neuman prosesor tidak perlu membedakan program dan data. Prosesor tipe ini tidak memerlukan *control bus* tambahan berupa pin I/O khusus untuk membedakan program dan data. Karena kemudahan ini, tidak terlalu sulit bagi prosesor yang berarsitektur Von Neumann untuk menambahkan peripheral eksternal seperti A/D converter, LCD, EEPROM dan devais I/O lainnya.
2. Biasanya devais eksternal ini sudah ada di dalam satu chips, sehingga prosesor seperti ini sering disebut dengan nama mikrokontroler (*microcontroller*).
3. fleksibilitas pengalamatan program dan data. Biasanya program selalu ada di ROM dan data selalu ada di RAM. Arsitektur Von Neumann memungkinkan prosesor untuk menjalankan program yang ada didalam memori data (RAM). Misalnya pada saat *power on*, dibuat program inisialisasi yang mengisi *byte* di dalam RAM.

4. Program ini adalah instruksi untuk mengisi accumulator A dengan data yang ada di alamat 4000 (ROM). Instruksi tersebut singkat hanya perlu satu baris saja. Pada prinsipnya, kode biner yang ada di RAM bisa berupa program dan bisa juga berupa data

## MODUL 7

### Media Penyimpanan Data Eksternal

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Media Penyimpanan Data Eksternal  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan Pembelajaran :

- 1 Siswa mengenal jenis magnetic disk dan teknologi RAID
- 2 Siswa mampu membedakan optical disk dan pita magnetik
- 3 Siswa mampu memahami Hirarki dan karakteristik sistem memori (inboard memory, outboard storage, off-line storage)

#### B. Materi Pembelajaran

### MEMORI INTERNAL

#### Karakteristik Sistem Memori (secara umum)

- |                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1. Lokasi</b> <ul style="list-style-type: none"><li><input type="checkbox"/> CPU</li><li><input type="checkbox"/> Internal (main)</li><li><input type="checkbox"/> External (secondary)</li></ul>                                                                | <b>5. Kinerja</b> <ul style="list-style-type: none"><li><input type="checkbox"/> Access time</li><li><input type="checkbox"/> Cycle time</li><li><input type="checkbox"/> Transfer rate</li></ul> |
| <b>2. Kapasitas</b> <ul style="list-style-type: none"><li><input type="checkbox"/> Ukuran word</li><li><input type="checkbox"/> Banyaknya word</li></ul>                                                                                                            | <b>6. Tipe Fisik</b> <ul style="list-style-type: none"><li><input type="checkbox"/> Semikonduktor</li><li><input type="checkbox"/> Permukaan magnetik</li></ul>                                   |
| <b>3. Satuan Transfer</b> <ul style="list-style-type: none"><li><input type="checkbox"/> Word</li><li><input type="checkbox"/> Block</li></ul>                                                                                                                      | <b>7. Karakteristik Fisik</b> <ul style="list-style-type: none"><li><input type="checkbox"/> Volatile/nonvolatile</li><li><input type="checkbox"/> Erasable/nonerasable</li></ul>                 |
| <b>4. Metode Akses</b> <ul style="list-style-type: none"><li><input type="checkbox"/> Sequential access</li><li><input type="checkbox"/> Direct access</li><li><input type="checkbox"/> Random access</li><li><input type="checkbox"/> Associative access</li></ul> | <b>8. Organisasi</b>                                                                                                                                                                              |

Catatan:

Bagi pengguna dua karakteristik penting memori adalah

- Kapasitas,
- Kinerja.

#### Lokasi Memori

Ada tiga lokasi keberadaan memori di dalam sistem komputer, yaitu:

- Memori lokal
  - Memori ini *built-in* berada dalam CPU (mikroprosesor),
  - Memori ini diperlukan untuk semua kegiatan CPU,

- Memori ini disebut **register**.
- Memori internal
  - Berada di luar CPU tetapi bersifat internal terhadap sistem komputer,
  - Diperlukan oleh CPU untuk proses eksekusi (operasi) program, sehingga dapat diakses secara langsung oleh prosesor (CPU) tanpa modul perantara,
  - Memori internal sering juga disebut sebagai **memori primer** atau **memori utama**.
  - Memori internal biasanya menggunakan media RAM
- Memori eksternal
  - Bersifat eksternal terhadap sistem komputer dan tentu saja berada di luar CPU,
  - Diperlukan untuk menyimpan data atau instruksi secara permanen.
  - Tidak diperlukan di dalam proses eksekusi sehingga tidak dapat diakses secara langsung oleh prosesor (CPU). Untuk akses memori eksternal ini oleh CPU harus melalui pengontrol/modul I/O.
  - Memori eksternal sering juga disebut sebagai **memori sekunder**.
  - Memori ini terdiri atas perangkat storage peripheral seperti : disk, pita magnetik, dll.

### Kapasitas Memori

- Kapasitas register (memori lokal) dinyatakan dalam **bit**.
- Kapasitas memori internal biasanya dinyatakan dalam bentuk **byte** (1 byte = 8 bit) atau **word**. Panjang word umum adalah 8, 16, dan 32 bit.
- Kapasitas memori eksternal biasanya dinyatakan dalam **byte**.

### Satuan Transfer (Unit of Transfer)

- Satuan transfer sama dengan jumlah saluran data yang masuk ke dan keluar dari modul memori.
- Bagi memori internal (memori utama), satuan transfer merupakan jumlah bit yang dibaca atau yang dituliskan ke dalam memori pada suatu saat.
- Bagi memori eksternal, data ditransfer dalam jumlah yang jauh lebih besar dari word, dalam hal ini dikenal sebagai **block**.

### Word

Ukuran word biasanya sama dengan jumlah bit yang digunakan untuk representasi bilangan dan panjang instruksi, kecuali CRAY-1 dan VAX.

- CRAY-1 memiliki panjang word 64 bit, memakai representasi integer 24 bit.
- VAX memiliki panjang instruksi yang beragam, ukuran wordnya adalah 32 bit.

### Addressable Units

Pada sejumlah sistem, addressable unit adalah word. Hubungan antara panjang suatu alamat (A) dengan jumlah addressable unit (N) adalah

$$2^A = N$$

### Metode Akses Memori

Terdapat empat jenis pengaksesan satuan data, sbb.:

- *Sequential Access*

- *Direct Access*
- *Random Access*
- *Associative Access*
- ❖ **Sequential Access**
  - Memori diorganisasikan menjadi unit-unit data, yang disebut **record**.
  - Akses dibuat dalam bentuk urutan linier yang spesifik.
  - Informasi pengalamatan dipakai untuk memisahkan record-record dan untuk membantu proses pencarian.
  - Mekanisme baca/tulis digunakan secara bersama (shared read/write mechanism), dengan cara berjalan menuju lokasi yang diinginkan untuk mengeluarkan record.
  - Waktu access record sangat bervariasi.
  - Contoh sequential access adalah akses pada pita magnetik.
- ❖ **Direct Access**
  - Seperti sequential access, direct access juga menggunakan shared read/write mechanism, tetapi setiap blok dan record memiliki alamat yang unik berdasarkan lokasi fisik.
  - Akses dilakukan secara langsung terhadap kisaran umum (general vicinity) untuk mencapai lokasi akhir.
  - Waktu aksesnya bervariasi.
  - Contoh direct access adalah akses pada **disk**.
- ❖ **Random Access**
  - Setiap lokasi dapat dipilih secara random dan diakses serta dialamati secara langsung.
  - Waktu untuk mengakses lokasi tertentu tidak tergantung pada urutan akses sebelumnya dan bersifat konstan.
  - Contoh random access adalah sistem memori utama.
- ❖ **Associative Access**
  - Setiap word dapat dicari berdasarkan pada isinya dan bukan berdasarkan alamatnya.
  - Seperti pada RAM, setiap lokasi memiliki mekanisme pengalamatannya sendiri.
  - Waktu pencariannya tidak bergantung secara konstan terhadap lokasi atau pola access sebelumnya.
  - Contoh associative access adalah memori cache.

### Kinerja Memori

Ada tiga buah parameter untuk kinerja sistem memori, yaitu :

- Waktu Akses (*Access Time*)
  - Bagi RAM, waktu akses adalah waktu yang dibutuhkan untuk melakukan operasi baca atau tulis.
  - Bagi non RAM, waktu akses adalah waktu yang dibutuhkan untuk melakukan mekanisme baca tulis pada lokasi tertentu.
- Waktu Siklus (*Cycle Time*)

- Waktu siklus adalah waktu akses ditambah dengan waktu transien hingga sinyal hilang dari saluran sinyal atau untuk menghasilkan kembali data bila data ini dibaca secara destruktif.
- Laju Pemindahan (*Transfer Rate*)
  - Transfer rate adalah kecepatan pemindahan data ke unit memori atau ditransfer dari unit memori.
  - Bagi RAM, transfer rate sama dengan  $1/(\text{waktu siklus})$ .
  - Bagi non-RAM, berlaku persamaan sbb.:

$$T_N = T_A + \frac{N}{R}$$

$T_N$  = Waktu rata-rata untuk membaca atau menulis sejumlah N bit.

$T_A$  = Waktu akses rata-rata

N = Jumlah bit

R = Kecepatan transfer, dalam bit per detik (bps)

### Tipe Fisik Memori

Ada beberapa tipe fisik memori, yaitu :

- **Memori semikonduktor**, memori ini memakai teknologi LSI atau VLSI (very large scale integration). Memori ini banyak digunakan untuk memori internal misalnya RAM.
- **Memori permukaan magnetik**, memori ini banyak digunakan untuk memori eksternal yaitu untuk disk atau pita magnetik.
- **Memori Optik**, memori ini banyak digunakan untuk memori eksternal yaitu untuk disk optic (CD, DVD, LD).

### Karakteristik Fisik

Ada dua kriteria yang mencerminkan karakteristik fisik memori, yaitu:

- **Volatile dan Non-volatile**
  - Pada memori volatile, informasi akan rusak secara alami atau hilang bila daya listriknya dimatikan.
  - Pada memori non-volatile, sekali informasi direkam akan tetap berada di sana tanpa mengalami kerusakan sebelum dilakukan perubahan. Pada memori ini daya listrik tidak diperlukan untuk mempertahankan informasi tersebut.  
Memori permukaan magnetik adalah non volatile.  
Memori semikonduktor dapat berupa volatile atau non volatile.
- **Erasable dan Non-erasable**
  - Erasable artinya isi memori dapat dihapus dan diganti dengan informasi lain.
  - Memori semikonduktor yang tidak terhapuskan dan non volatile adalah ROM.

### Organisasi

Yang dimaksud dengan organisasi adalah pengaturan bit dalam menyusun word secara fisik.



### **Hirarki Memori**

Tiga pertanyaan dalam rancangan memori, yaitu :

Berapa banyak?                      Berapa cepat?                      Berapa mahal?

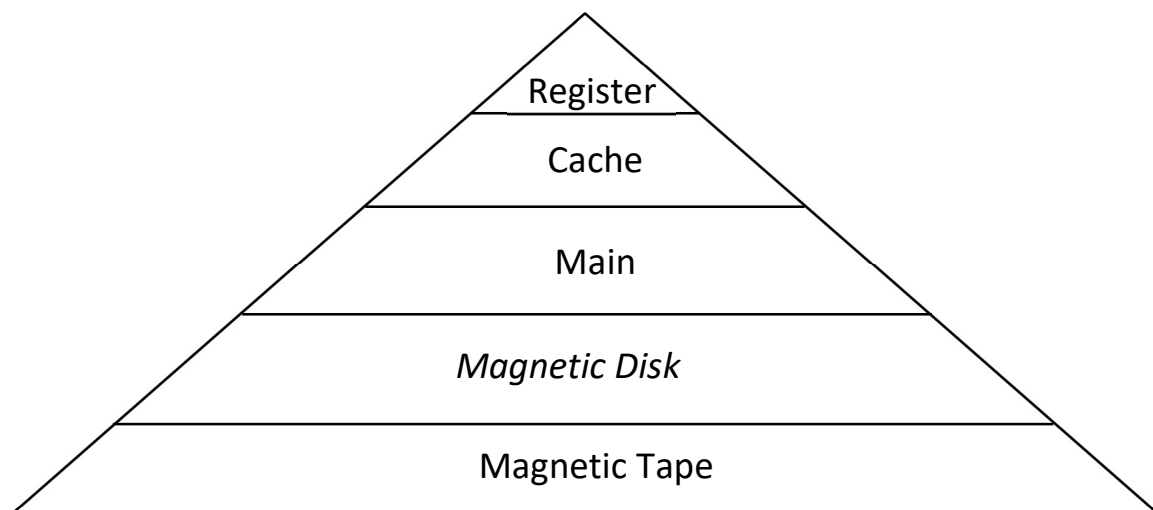
Kapasitas                                  Waktu access                      Harga

Setiap spektrum teknologi mempunyai hubungan sbb.:

- Semakin kecil waktu access, semakin besar harga per bit.
- Semakin besar kapasitas, semakin kecil harga per bit.
- Semakin besar kapasitas, semakin besar waktu access.

Untuk mendapatkan kinerja terbaik, memori harus mampu mengikuti CPU. Artinya apabila CPU sedang mengeksekusi instruksi, kita tidak perlu menghentikan CPU untuk menunggu datangnya instruksi atau operand.

Untuk memperoleh kinerja yang optimal, perlu kombinasi teknologi komponen memori. Dari kombinasi ini dapat disusun hirarki memori sbb.:



Semakin menurun hirarki, maka hal-hal di bawah ini akan terjadi :

- a) Penurunan harga per bit
- b) Peningkatan kapasitas
- c) Peningkatan waktu akses
- d) Penurunan frekuensi akses memori oleh CPU.

Kunci keberhasilan organisasi adalah penurunan frekuensi akses memori oleh CPU. Bila memori dapat diorganisasikan dengan penurunan harga per bit melalui peningkatan waktu akses, dan bila data dan instruksi dapat didistribusikan melalui memori ini dengan penurunan frekuensi akses memori oleh CPU, maka pola ini akan mengurangi biaya secara keseluruhan dengan tingkatan kinerja tertentu.

**Register** adalah jenis memori yang tercepat, terkecil, dan termahal yang merupakan memori internal bagi prosesor.

**Cache** adalah perangkat untuk pergerakan data antara memori utama dan register prosesor untuk meningkatkan kinerja.

**Memori utama** merupakan sistem internal memory dari sebuah komputer. Setiap lokasi di dalam memori utama memiliki alamat yang unik.

Ketiga bentuk memori di atas bersifat volatile dan memakai teknologi semikonduktor.

**Magnetic disk** dan **Magnetic tape** adalah external memory dan bersifat non-volatile.

### Memori Semikonduktor

Ada beberapa memori semikonduktor, yaitu :

1. RAM : RAM statik (SRAM) dan RAM dinamik (DRAM).
2. ROM : ROM, Programmable ROM (PROM), Erasable PROM (EPROM), Electrically EPROM (EEPROM), Flash Memory.

### Random Access Memory (RAM)

- Baca dan tulis data dari dan ke memori dapat dilakukan dengan mudah dan cepat.
- Bersifat volatile
- Perlu catu daya listrik.

### RAM Dinamik (DRAM)

Disusun oleh sel-sel yang menyimpan data sebagai muatan listrik pada kapasitor. Ada dan tidak ada muatan listrik pada kapasitor dinyatakan sebagai bilangan biner 1 dan 0. Perlu pengisian muatan listrik secara periodik untuk memelihara penyimpanan data.

### RAM Statik (SRAM)

Disusun oleh deretan flip-flop. Baik SRAM maupun DRAM adalah volatile. Sel memori DRAM lebih sederhana dibanding SRAM, karena itu lebih kecil. DRAM lebih rapat (sel lebih kecil = lebih banyak sel per satuan luas) dan lebih murah. DRAM memerlukan rangkaian pengosong muatan. DRAM cenderung lebih baik bila digunakan untuk kebutuhan memori yang lebih besar. DRAM lebih lambat.

### Read Only Memory (ROM)

- Menyimpan data secara permanen
- Hanya bisa dibaca

Dua masalah ROM

- Langkah penyisipan data memerlukan biaya tetap yang tinggi.
- Tidak boleh terjadi kesalahan (error).

### Programmabel ROM (PROM)

Bersifat non volatile dan hanya bisa ditulisi sekali saja. Proses penulisan dibentuk secara elektris. Diperlukan peralatan khusus untuk proses penulisan atau “pemrograman”.

## Erased PROM (EPROM)

Dibaca secara optis dan ditulisi secara elektris. Sebelum operasi write, seluruh sel penyimpanan harus dihapus menggunakan radiasi sinar ultra-violet terhadap keping paket. Proses penghapusan dapat dilakukan secara berulang, setiap penghapusan memerlukan waktu 20 menit. Untuk daya tampung data yang sama EPROM lebih mahal dari PROM.

## Electrically EPROM (EEPROM)

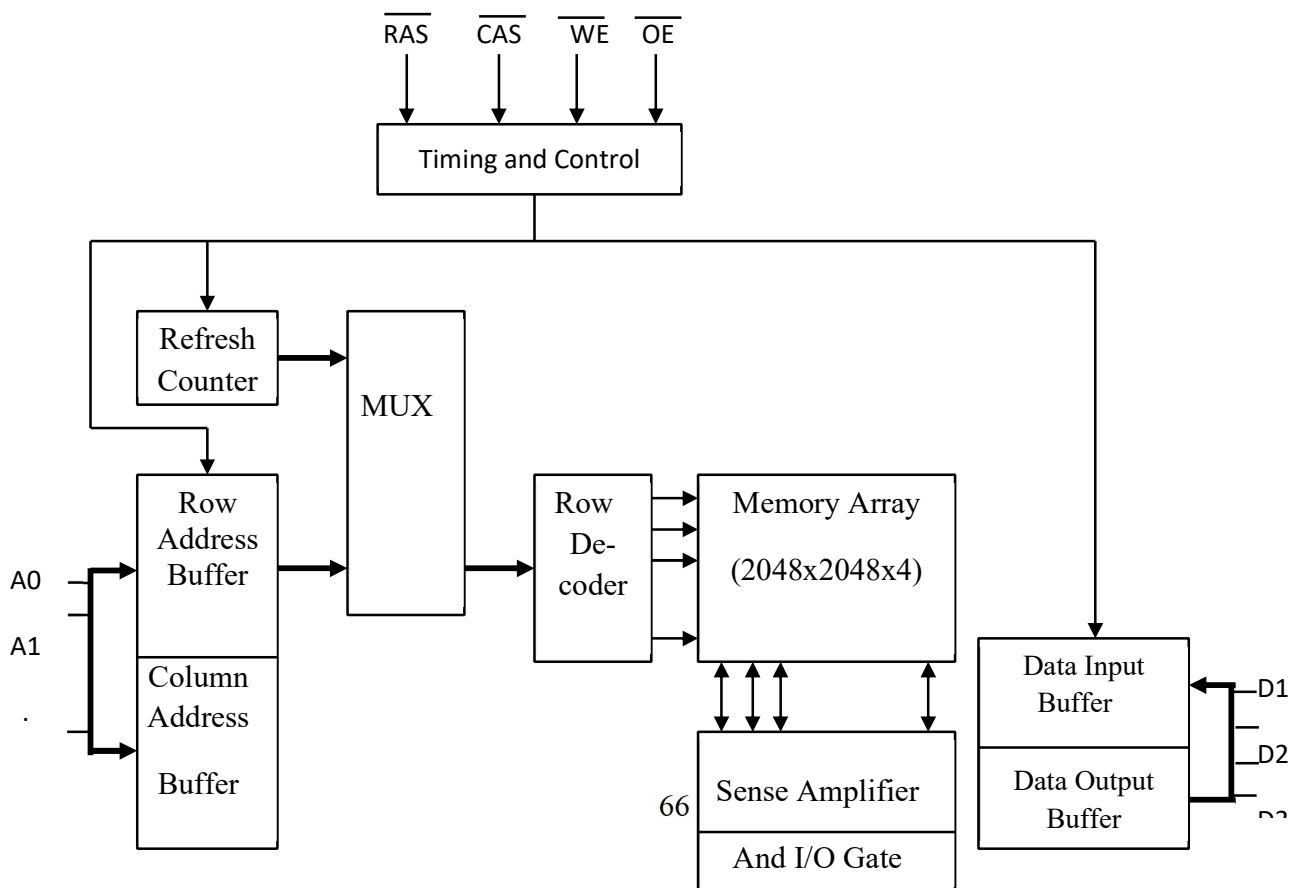
Dapat ditulis kapan saja tanpa menghapus isi sebelumnya. Operasi write memerlukan waktu lebih lama dibanding operasi read. Gabungan sifat kelebihan non-volatilitas dan fleksibilitas untuk update dengan menggunakan bus control, alamat dan saluran data. EEPROM lebih mahal dibanding EPROM.

Sel memori memiliki sifat tertentu sbb :

- Memiliki dua keadaan stabil untuk representasi bilangan biner 1 atau 0.
- Memiliki kemampuan untuk ditulisi
- Memiliki kemampuan untuk dibaca.

### *Organisasi Logik Keping (Chip Logic) Memori*

Organisasi DRAM 16 Mbit secara umum. Array memori diorganisasikan sebagai empat buah kuadrat 2048 terhadap 2048 elemen. Elemen-elemen array dihubungkan dengan saluran horizontal (baris) dan vertikal (kolom). Setiap saluran horizontal terhubung ke terminal Data-in/Sense masing-masing sel pada kolomnya.



## **MEMORI EKSTERNAL**

### **Magnetik Disk**

- Disk merupakan sebuah piringan bundar yang terbuat dari logam atau plastik yang dilapisi dengan bahan yang dapat dimagnetisasi.
- Data direkam di atasnya dan dapat dibaca dari disk dengan menggunakan kumparan pengkonduksi (*conducting coil*) yang dinamakan *head*
- Pada operasi penulisan, arus listrik pada head memagnetisasi disk.
- Pada operasi pembacaan, medan magnet pada disk yang bergerak di bawah *head* menghasilkan arus listrik pada *head*.
- Selama operasi pembacaan dan penulisan, *head* bersifat stasioner sedangkan piringan bergerak-gerak di bawahnya.

### **Organisasi Data dan Pemformatan**

- Organisasi data pada piringan berbentuk sejumlah cincin-cincin yang konsentris yang disebut *track*.
- Masing-masing *track* lebarnya sama dengan lebar *head*.
- Track yang berdekatan dipisahkan oleh *gap*
- *Gap* bertujuan untuk mencegah/mengurangi error akibat melesetnya *head* atau interferensi medan magnet.
- Kerapatan (*density*), dalam bit per inci linear, pada *track* sebelah dalam lebih tinggi (lebih rapat) dibanding *track* sebelah luarnya.
- Data disimpan pada daerah berukuran blok yang dikenal sebagai *sector*.
- Biasanya terdapat antara 10 hingga 100 *sector per track*.
- Sector-sector yang berdekatan dipisahkan oleh *gap-gap intra-track* atau *inter-record*.
- Layout data disk meliputi:
  - *Track*
  - *Inter-track Gaps*
  - Kerapatan (*density*)
  - *Sector*
- Salah satu contoh pemformatan disk adalah format track disk Winchester (Seagate ST506)

- Setiap track berisi 30 sector yang panjangnya tetap
- Masing-masing track berisi 600 byte
- Setiap sector menampung 512 byte data ditambah informasi kontrol yang berguna bagi disk controller.

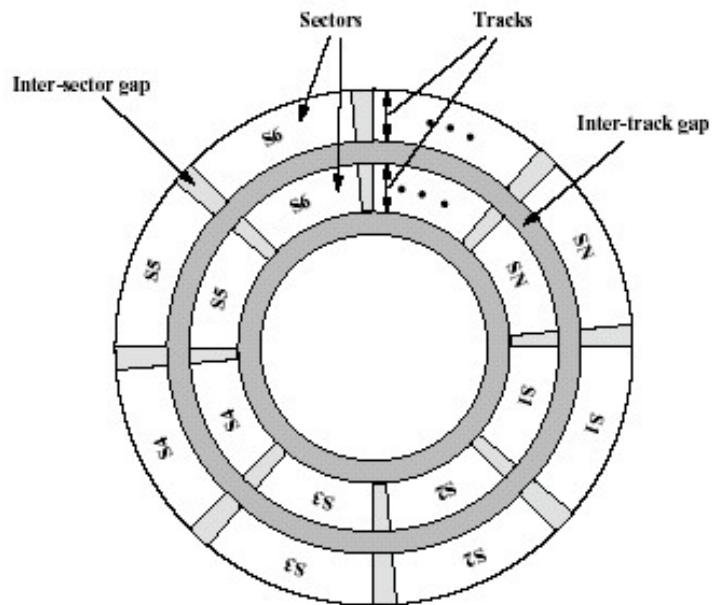


Figure 5.1 Disk Data Layout

### Karakteristik

Ada beberapa karakteristik Sistem Disk:

#### ➤ Gerakan *head*

- *Fixed head disk* □ terdapat sebuah *head* baca/tulis per *track* jadi ada beberapa *head* baca/tulis per *surface*. Semua *head* ditempatkan pada lengan memanjang ke seluruh *track*.
- *Movable head disk* □ hanya terdapat sebuah *head* baca/tulis per *surface*. Lengan dimana *head* ditempatkan dapat memanjang dan memendek untuk menuju ke salah satu *track*.

#### ➤ Portabilitas disk

Disk berada pada sebuah *disk drive* yang terdiri dari lengan, tangkai yang dapat menggerakkan disk, dan perangkat elektronik untuk keperluan input dan output data biner.

- *Non-removable disk* □ secara permanen berada pada disk drive.
- *Removable disk* □ dapat dilepas dan diganti dengan disk lain.

#### ➤ Permukaan yang dimagnetisasi

- *Double-sided* □ kedua sisi permukaannya dimagnetisasi
- *Single-sided* □ hanya satu permukaan yang dimagnetisasi (disk bermuka tunggal)

#### ➤ Banyaknya piringan pada *disk drive*

- *Single platter*
- *Multiple platter*

#### ➤ Mekanisme Head

- *Contact (floppy)* □ terdapat kontak secara fisik antara *head* dengan medium (disk) selama operasi baca/tulis.
- *Fixed Gap* □ ada jarak yang tetap antara *head* dengan disk.
- *Aerodynamic Gap (Winchester)* □ ada kertas timah pelindung yang aerodynamis antara *head* dengan disk sehingga jarak antara *head* dan disk dapat diperpendek.

Tabel karakteristik Sistem Disk:

Gerakan Head Fixed head (one per track) Movable head (one per surface)	Platters Single platter Multiple platter
Portabilitas Disk Nonremovable disk Removable disk	Mekanisme Head Contact (floppy) Fixed Gap Aerodynamic Gap (Winchester)
Sides Single sided Double sided	

### Waktu Akses Disk

- Ketika *disk drive* beroperasi, disk berputar dengan kecepatan tetap.
- Untuk dapat membaca dan menulis, *head* harus berada pada awal *sector* dari *track* yang diinginkan.
- Pemilihan *track* meliputi perpindahan *head* pada sistem *movable head* atau mekanisme elektronis pada *head* untuk sistem *fixed head*.
  - Waktu yang diperlukan untuk menempatkan *head* pada *track* yang diinginkan dikenal sebagai *seek time*.
- Sekali *track* sudah dipilih, sistem akan menunggu sampai *sector* yang bersangkutan berputar agar sesuai dengan *head*.
  - Waktu yang diperlukan oleh *sector* untuk mencapai *head* disebut *rotational latency*
- *Access time*, yaitu waktu yang dibutuhkan untuk berada pada posisi siap membaca atau menulis.
  - Jumlah antara *seek time* dan *rotational latency* sama dengan *Access time*.

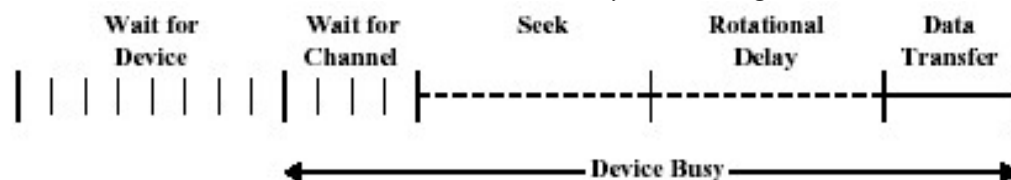


Figure 5.5 Timing of a Disk I/O Transfer

### RAID (Redundancy Array of Independent Disk)

RAID (*Redundancy Array of Independent Disk*) diajukan untuk mendekatkan jurang yang lebar antara kecepatan prosesor dan elektromekanis *disk drive* yang relatif lambat.

Strateginya adalah dengan mengganti disk berkapasitas besar dengan sejumlah disk drive berkapasitas kecil, dan mendistribusikan data sedemikian rupa sehingga memungkinkan akses data dari sejumlah drive secara simultan, yang akan meningkatkan kinerja I/O dan memungkinkan peningkatan kapasitas secara mudah.

RAID mengatasi permasalahan standarisasi bagi rancangan database dengan disk berjumlah banyak.

Pola RAID terdiri dari enam tingkat, nol hingga lima.

Tiga karakteristik umum pada Tingkatan RAID, yaitu:

1. RAID merupakan sekumpulan *disk drive* yang dianggap oleh sistem operasi sebagai sebuah drive logik tunggal.
2. Data didistribusikan ke *drive* fisik *array*
3. Kapasitas redundant disk digunakan untuk menyimpan informasi paritas, yang menjamin recoverability data ketika terjadi kegagalan disk.

RAID Tingkat 0

- RAID tingkat 0 sebenarnya bukan anggota keluarga RAID karena tidak menggunakan redundansi untuk meningkatkan kinerja.
- Bagi RAID tingkat 0, data pengguna dan data sistem didistribusi ke seluruh disk pada array.

RAID Tingkat 1

- RAID tingkat 1 berbeda dengan RAID tingkat 2 sampai 5 dalam cara memperoleh redundansinya.
- Pada RAID lainnya, beberapa bentuk kalkulasi paritas digunakan untuk mendapatkan redundansi.
- Pada RAID tingkat 1, redundansi diperoleh cukup dengan cara menduplikasikan seluruh data.
- Beberapa aspek positif bagi organisasi RAID 1 :
  - *Read request* dapat dilayani oleh salah satu dari kedua disk yang berisi data yang diminta, yang memiliki *seek time* plus *rotational latency* yang minimum.
  - *Write request* memerlukan kedua strip yang berkaitan untuk di-update, namun hal ini dapat dilakukan secara paralel.
  - *Recovery* dari kegagalan cukup sederhana. Bila *drive* mengalami kegagalan, maka data masih dapat diakses dari *drive* kedua.

RAID tingkat 2

- ⇒ RAID tingkat 2 dan 3 menggunakan teknik akses paralel.
- ⇒ Dalam parallel access array, seluruh anggota disk berpartisipasi dalam mengeksekusi setiap request I/O.
- ⇒ Pemutar setiap drive umumnya disinkronisasikan sehingga seluruh head disk selalu berada pada posisi yang sama.

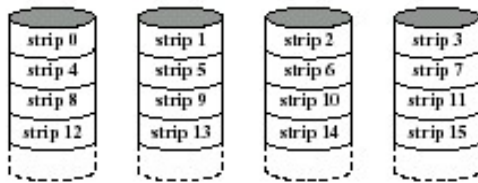
RAID tingkat 3

- ⇒ RAID 3 diorganisasikan dengan cara yang sama dengan RAID 2, bedanya adalah bahwa RAID 3 hanya membutuhkan disk redundan tunggal, tidak tergantung pada berapa besar array disknya.
- ⇒ RAID 3 menggunakan akses paralel dengan data yang didistribusikan dalam bentuk strip-strik kecil.
- ⇒ Di sini kode *error-correcting* tidak dihitung.

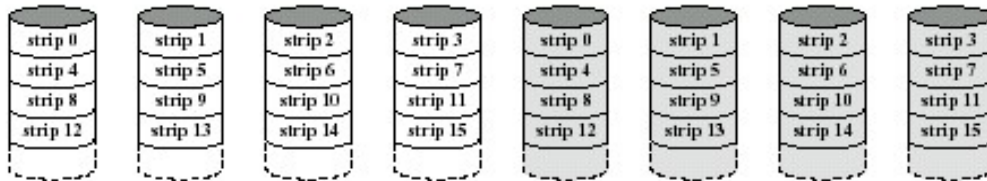
RAID tingkat 4

- ⇒ RAID tingkat 4 dan 5 menggunakan teknik akses yang independen.
- ⇒ Dalam array dengan akses independen, setiap disk anggota beroperasi secara independen, sehingga request I/O dapat dipenuhi secara paralel.
- ⇒ Laju transfer data tinggi

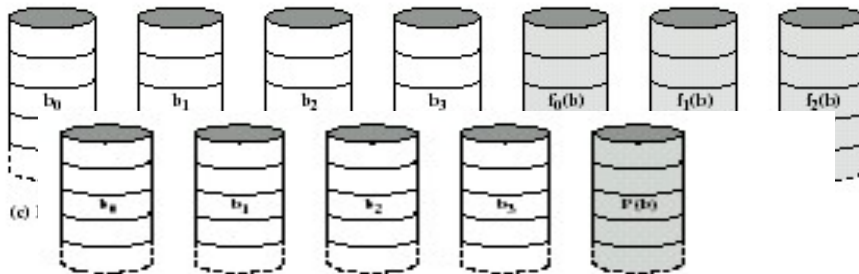
⇒ Juga digunakan striping data



(a) RAID 0 (non-redundant)



(b) RAID 1 (mirrored)



(c)

(d) RAID 5 (bit-interleaved parity)



(e) RAID 4 (block-level parity)



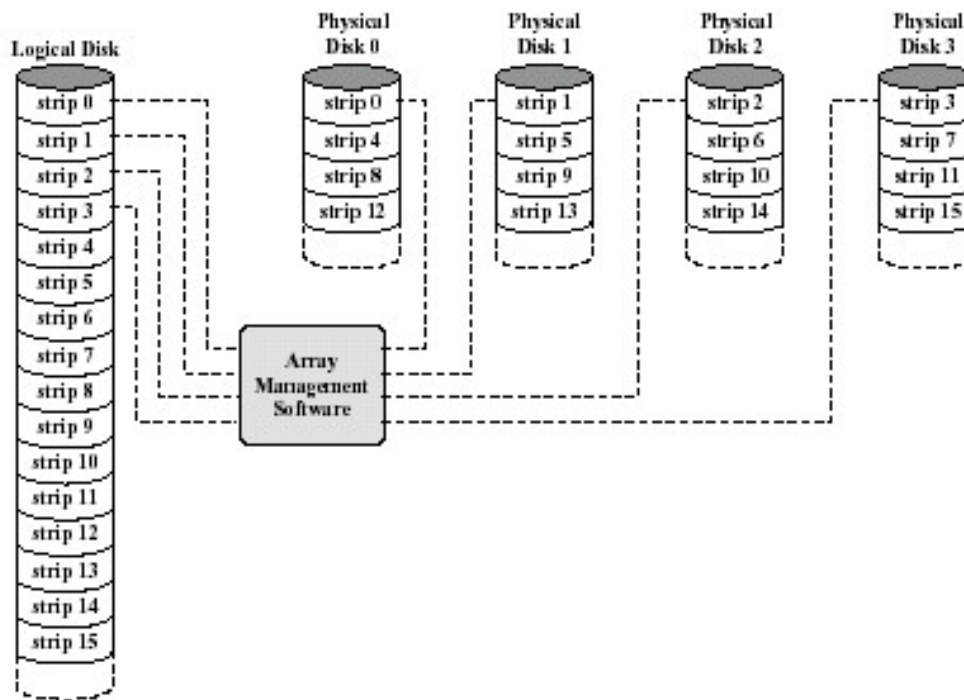
(f) RAID 5 (block-level distributed parity)



(g) RAID 6 (dual redundancy)

Figure 5.6 RAID Levels (page 2 of 2)





**Figure 5.7 Data Mapping for a RAID Level 0 Array**

## Optical Memory

### Produk-produk disk optis

1. CD (Compact Disk) □ suatu disk yang tidak dapat dihapus yang menyimpan informasi audio yang telah didigitasi. Sistem standar menggunakan disk 12 cm yang dapat merekam lebih dari 60 menit waktu putar tanpa henti.
2. CD-ROM (Compact Disk Read Only Memory) □ Disk yang tidak dapat dihapus untuk menyimpan data komputer. Sistem standar menggunakan disk 12 cm yang dapat menampung lebih dari 550 Mbyte.
3. CD-I (Compact Disk Interactive) □ Suatu spesifikasi yang didasarkan pada penggunaan CD-ROM. Spesifikasi ini menjelaskan metode penyediaan audio, video, grafis, teks, dan kode yang dapat dieksekusi mesin pada CD-ROM.
4. DVI (Digital Video Interactive) □ Sebuah teknologi untuk memproduksi representasi informasi video yang didigitasi dan terkompresi. Representasi dapat disimpan pada CD atau media disk lainnya. Sistem yang ada sekarang menggunakan CD dan dapat menyimpan sekitar 20 menit video pada satu disk.
5. WORM (Write One Read Many) □ Sebuah disk yang lebih mudah ditulisi dibandingkan dengan CD-ROM, yang membuatnya secara komersial feasible untuk menyalin sebuah CD. Ukuran yang populer adalah 5,25 inchi yang dapat menampung 200 hingga 800 Mbyte data.
6. Erasable Optical Disk □ Suatu disk yang menggunakan teknologi optik namun dapat dihapus dan ditulisi ulang dengan mudah. Terdapat dua jenis ukuran yang umum dipakai: 3,25 inchi dan 5,25 inchi. Umumnya mempunyai kapasitas 650 Mbyte.

## CD-ROM

- CD-ROM player memiliki perangkat *error-correcting* untuk menjamin bahwa data ditransfer dengan benar dari disk ke komputer.
- Disk terbuat dari resin, seperti polycarbonate, dan dilapisi dengan permukaan yang sangat reflektif, biasanya aluminium.
- Informasi yang direkam secara digital diterbitkan sebagai sekumpulan lubang-lubang mikroskopik pada permukaan yang reflektif.
- Permukaan disk dilindungi dari debu dan gesekan dengan lapisan bening.
- Layout disk yang menggunakan **constant angular velocity** (CAV)
  - ❖ Keuntungan CAV : blok data dapat dialamati secara langsung oleh track dan sector. Untuk memindahkan head ke alamat tertentu hanya memerlukan gerakan head yang pendek dan waktu tunggu yang singkat.
  - ❖ Kerugian CAV : jumlah data yang dapat disimpan pada track yang jauh di luar sama dengan yang dapat disimpan dengan track yang berada dekat titik pusat.
- Kapasitas penyimpanan CD-ROM adalah 774,57 Mbyte.
- Format blok CD-ROM terdiri dari field-field sbb.:
  - ❖ Sync : Field sync mengidentifikasikan awal sebuah blok.
  - ❖ Header : header terdiri dari alamat blok dan byte mode.
    - Mode 0 menandakan suatu field data blanko;
    - mode 1 menandakan penggunaan kode *error-correcting* dan 2048 byte data;
    - mode 2 menandakan 2336 byte data pengguna tanpa kode *error-correcting*.
  - ❖ Data : data adalah data pengguna
  - ❖ Auxiliary : data pengguna tambahan dalam mode 2. Pada mode 1, data ini data pengguna tambahan dalam mode 2. Pada mode 1, data ini merupakan kode *error-correcting*
- Ada pula disk dengan menggunakan layout **kecepatan linear konstant** (CLV)
- Keuntungan CD-ROM:
  - ❖ Kapasitas penyimpanan informasinya jauh lebih besar dibandingkan dengan disk magnetik.
  - ❖ Dapat diperbanyak dengan harga murah
  - ❖ Dapat dipindah-pindah. Sebagian besar disk magnetik tidak dapat dipindah-pindahkan.
- Kekurangan CD-ROM:
  - ❖ CD-ROM hanya dapat dibaca saja (read only) dan tidak dapat di update.
  - ❖ CD-ROM memiliki waktu akses yang lebih lama dibandingkan dengan waktu akses disk drive magnetik.

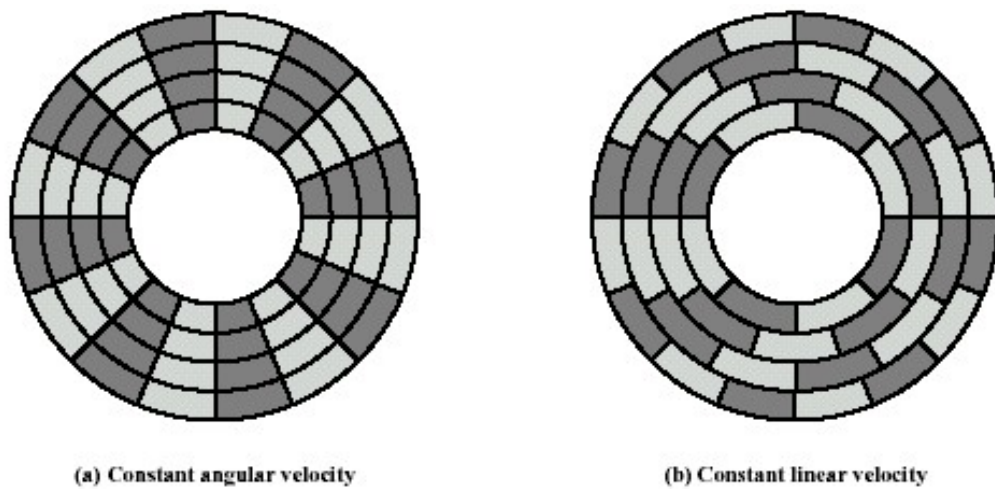


Figure 5.8 Comparison of Disk Layout Methods

### WORM

- WORM adalah Write Once Read Many CD.
- Dapat ditulisi sekali menggunakan sinar laser berintensitas sedang.
- Teknik yang dipakai untuk menyiapkan disk adalah dengan menggunakan laser berdaya tinggi.
- Menggunakan kecepatan angular yang konstan untuk memberikan akses yang lebih cepat.
- Digunakan untuk penyimpanan arsip dokumen dan file dalam ukuran besar.

### Disk Optis yang Dapat Dihapus

- Disk dapat ditulisi berulang-ulang
- Menggunakan teknologi sistem magneto-optis: pada sistem ini, energi sinar laser digunakan secara bersama dengan medan magnet untuk merekam dan menghapus informasi.
- Menggunakan kecepatan angular konstan.
- Keuntungan utama disk optis ini dibandingkan dengan disk magnetis:
  - ❖ Berkapasitas besar : sebuah disk optis 5,25 inchi dapat menampung data sekitar 650 Mbyte.
  - ❖ Portabilitas : Disk optis dapat dipindahkan dari drivenya.
  - ❖ Reliabilitas
  - ❖ Tahan lama

### Pita Magnetik

Sistem pita menggunakan teknik pembacaan dan penulisan yang sama dengan sistem disk.

Media sistem ini adalah pita mylar lentur yang dilapisi dengan oksida magnet.

Pita dan drive pita merupakan analog terhadap sistem tape recorder.

Medium pita berbentuk track-track paralel dalam jumlah sedikit.

Sistem pita magnetik kuno memakai 9 buah track.

Sistem pita magnetik terbaru menggunakan 18 atau 36 track.

Data ditulisi dan dibaca dalam bentuk blok-blok continuous yang disebut *physical records* pada pita.

Blok-blok pada pita dipisahkan dipisahkan oleh gap yang dikenal sebagai *inter-record gaps*.

## **MODUL 8**

### **Karakteristik Memori**

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Karakteristik Memori  
**Alokasi waktu** : Jam Pelajaran

#### **A. Tujuan Pembelajaran :**

- 1 Siswa mengenal Karakteristik pada memori (lokasi, kapasitas, satuan transfer, metode akses, kinerja, tipe fisik dan karakteristik fisik).
- 2 Siswa mampu Menyajikan gagasan untuk merangkai beberapa memori dalam sistem komputer.
- 3 Siswa mampu mengidentifikasi keandalan memori
- 4 Siswa mampu Mengidentifikasi Rangkaian memori RAM - EPROM

#### **B. Materi Pembelajaran**

##### **Karakteristik Media Penyimpanan**

##### **Location**

Lokasi adalah dimana letak memori tersebut, biasanya terdapat dua tempat yaitu didalam (Internal) atau diluar (Eksternal) komputer, untuk yang didalam komputer kadang sering disamakan sebagai main memori, padahal ada bentuk memori yang lain. Contohnya processor memiliki register memori dan ada juga yang disebut chace memori. Untuk yang eksternal memori, yaitu suatu media yang pengaksesannya memerlukan proses I/O seperti disk atau tape.

##### **Capacity**

Kapasitas adalah sebuah karakteristik nyata dari memori. Biasanya kapasitas internal memori dituliskan dalam bytes (1 byte = 8 bits) atau instruksi. Umumnya panjang instruksi adalah 8, 16, dan 32 bits. Sedangkan eksternal memori biasa di tuliskan dalam byte.

##### **Unit of Transfer**

Untuk internal memori, Unit Of Transfer sama dengan jumlah baris data yang keluar masuk ruang memori dalam satu waktu.

##### **Access Method**

Sequential Access : memori diorganisasikan dalam unit-unit data yang disebut record. Pengaksesannya harus dibuat secara linear berurutan. Setiap data diberikan indeks alamat untuk memisahkan setiap record dan agar memudahkan pengaksesan kembali data. Dengan metode akses ini akan membutuhkan waktu yang lama untuk mengakses suatu data karena pencarian data dilakukan secara satu-persatu. Contohnya tape memori.

**Direct Access** : berbeda dengan Sequential Access pada metode ini, setiap blok record diberikan alamat khusus dalam memori. Untuk mengakses suatu data, kita langsung dapat mengetahui tempat data tersebut disimpan melalui alamat khusus. Sehingga proses pembacaan data cepat. Contohnya disk memori.

**Random Access** : setiap data diberikan lokasi pengalamatan khusus dan ditempatkan disembarang tempat pada memori. Sehingga data diakses secara acak dan langsung. Contohnya memori utama dan chace memori.

**Assosiative** : ini adalah salah satu metode Random Access yang dapat membentuk instruksi penempatan data dan pencocokkan data sendiri, dan semua instruksi dijalankan bersamaan. Chace memori dapat menggunakan metode ini.

### Performance

Ada tiga parameter Performance yang digunakan :

- **Access Time** : yaitu waktu yang ditempuh untuk melakukan proses baca tulis data.
- **Memory Cycle Time** : konsep ini adalah pokok yang direpkan pada Random Access, dan terdiri dari Access Time ditambah waktu proses yang dibutuhkan sebelum akses kedua dimulai. Memory Cycle Time berhubungan dengan sistem bus dan tak berhubungan dengan Processor.
- **Transfer Rate** : yaitu kecepatan mentransfer data masuk atau keluar unit memori. Untuk Random Access sama dengan  $1 / (\text{Cycle Time})$ , sedangkan pada Non-Random-Access menggunakan persamaan ini :

$$T_n = T_a + N/R$$

Dimana  $T_n$  = Rata-rata waktu untuk membaca dan menulis N bits.

$T_a$  = Rata-rata waktu akses.

$N$  = Banyak bits.

$R$  = Kecepatan transfer, dalam bits per second (bps).

### Physical Type

Yaitu bahan yang digunakan untuk membuat memori, kebanyakan saat ini digunakan bahan semikonduktor.

### Physical Characteristic

Beberapa karakter fisik memori sangat penting. Pada Volatile Memory informasi akan hilang dengan otomatis ketika power listrik dimatikan. Sedangkan pada Unvolatile Memory, data akan direkam dalam memori tanpa adanya kehilangan data. Contohnya magnetik memori adalah Unvolatile Memory. Sedangkan semikonduktor bisa Volatile ataupun Unvolatile Memory. Nonerasable Memory contohnya termasuk Unvolatile Memory, data dalam memori ini tak dapat dihapus kecuali jika terjadi kerusakan media. Nonerasable Memory dalam semikonduktor dikenal dengan ROM (Read Only Memory).

### Organization

Dengan mengorganisasikan berarti mengatur penyusunan instruksi-instruksi setiap bit. Walaupun kenyataannya pengorganisasian tidak selalu digunakan.

### **Hirarki Media Penyimpanan**

Ada banyak media penyimpanan berkas, dan semuanya diklasifikasikan berdasarkan kecepatan baca tulis data, dan waktu yang dibutuhkan untuk mengakses data, yaitu :

#### **Chace Memory**

Memori berkapasitas terbatas, berkecepatan tinggi yang lebih mahal daripada memori utama. Cache memory ini ada diantara Main Memory dan Register pemroses, berfungsi agar pemroses tidak langsung mengacu pada memori utama agar kinerja dapat ditingkatkan. Chace memory terbagi menjadi dua yaitu :

1. Cache Memory yang terdapat pada internal Processor , chace memory jenis ini kecepatan aksesnya sangat tinggi, dan harganya sangat mahal. Hal ini bisa terlihat pada Processor yang berharga mahal seperti P4,P3,AMD-Athlon dll, semakin tinggi kapasitas L1,L2 Chace memori maka semakin mahal dan semakin ceppat Processor.
2. Chace Memory yang terdapat diluar Processor, yaitu berada pada MotherBoard, memori jenis ini kecepatan aksesnya sangat tinggi, meskipun tidak secepat chache memori jenis pertama ( yang ada pada internal Processor). Semakin besar kapasitasnya maka semakin mahal dan cepat. Hal ini bisa kita lihat pada Motherboard dengan beraneka ragam kapasitas chace memory yaitu 256kb, 512kb, 1Mb, 2Mb dll.

#### **Main Memory**

Media penyimpanan yang digunakan ketika data akan dioperasikan adalah Main Memory (Memori Utama). Umumnya mekanisasi pengoperaian data dilakukan di Main Memory. Walaupun Main Memory berukuran bermega-mega byte atau bergiga-giga byte, tetap saja Main Memory terlalu kecil untuk penyimpanan database. Selain itu Main Memory akan kehilangan data ketika listrik dimatikan atau ketika sistem crash (error).

#### **ROM (Read Only Memory)**

ROM ( Read Only memory) yaitu memory yang hanya bisa dibaca saja datanya atau programnya. Pada PC, ROM terdapat pada BIOS ( Basic Input Output System) yang terdapat pada Mother Board yang berfungsi untuk men-setting peripheral yang ada pada system. Contoh: AMIBIOS, AWARD BIOS, dll

ROM untuk BIOS terdapat beragam jenis diantaranya jenis Flash EEPROM BIOS yang memiliki kemampuan untuk dapat diganti programnya dengan software yang disediakan oleh perusahaan pembuat Mother Board, yang umumnya penggantian tersebut untuk peningkatan unjuk kerja dari peripheral yang ada di Mother Board.

#### **RAM(Random Access Memory)**

##### **DRAM**

DRAM (Dynamic RAM) yaitu salah satu tipe RAM yang menyimpan setiap bit data dalam sebuah Capacitor terpisah dalam sebuah IC (Integrated Circuit). Ketika Capacitor – Capacitor jarang di charge, maka dimungkinkan data yang ada pada Capacitor hilang, sehingga Capacitor harus di charge (refresh) secara periodik. DRAM sangat berlawanan dengan SRAM (Static RAM) dan static memori lainnya. Keuntungan SRAM adalah struktur nya yang sederhana. Hanya satu Transistor dan satu Capacitor yang diperlukan per bit. Sehingga dibandingkan dengan SRAM, DRAM lebih padat. Karena DRAM akan kehilangan data ketika power listrik dimatikan maka DRAM termasuk Volatile Memory.

### **SRAM**

SRAM (Static RAM), sama halnya dengan DRAM atalah salah satu tipe dari RAM. Perbedaannya dengan DRAM adalah data yang disimpan dalam SRAM bersifat static, sehingga selama teraliri power listrik, maka data akan tetap utuh, berbeda dengan DRAM yang secara periodik harus di refresh. Selain itu SRAM digunakan untuk Chace Memory (baik itu didalam maupun diluar chip), sedangkan DRAM digunakan untuk Main Memory, sehingga kecepatan SRAM lebih cepat dari DRAM.

RAM (Random Acces Memory) yaitu memori yang memiliki kemampuan untuk dirubah data atau program yang tersimpan didalamnya.

### **Flash Memory**

Flash memory juga dikenal sebagai EEPROM (Electrically Erasable Programmable Read-Only Memory), flash memory berbeda dengan Main Memory dalam penyelamatan data. Flash Memory tak akan kehilangan data walaupun kehilangan power listrik sehingga digolongkan sebagai Unvolatile Memory. Pembacaan data dalam Flash Memory dibutuhkan waktu kurang dari 1 nanosecond (1/1000 microsecond), yang kira-kira secepat pembacaan data dari Main Memory. Bagaimanapun penulisan data ke Flash Memory lebih ruwet, data bisa ditulis sekali yang memakan waktu 4 – 10 microsecond, tapi tidak bisa dioverwrite secara langsung. Untuk mengoverwrite data yang telah ada kita harus menghapus data tersebut lebih dulu, setelah itu data baru dapat dioverwrite. Kekurangan Flash Memory yaitu hanya dapat melakukan penghapusan ulang sekitar 10.000 sampai 1 juta kali. Flash Memory populer digunakan sebagai pengganti Magnetic Disk dalam menampung data yang berukuran tidak besar. Selain itu Flash Memory juga populer sebagai media penyimpanan untuk system computer sederhana, seperti HP, Camera Digital, dan media digital lainnya.

### **Magnetic Disk**

Magnetic Disk memberikan media penyimpanan sekunder dalam jumlah besar. Kapasitas Disk selalu meningkat lebih dari 50% setiap tahunnya, namun kebutuhan media penyimpanan untuk aplikasi besarpun meningkat dengan cepat, dalam beberapa kasus kadang database yang besar membutuhkan ratusan disk untuk menampung data tersebut.

Data yang disimpan dalam disk dalam unit disebut Disk Block. Block Disk tersusun atas cincin yang konsentris yang disebut Track, dalam sebuah piringan atau lebih. Seset Track yang memiliki diameter sama disebut Cylinder. Setiap track terbagi atas lengkungan-lengkungan yang disebut Sector



yang besarnya berdasarkan karakteristik disk dan tak bisa diubah, sedangkan ukuran Block dapat ditentukan ketika diinisialisasikan sebagai multiple sector.

### Physical Characteristics of Disks

- Head Motion
  - Fixed Head Disk : memiliki head untuk setiap track, sehingga computer dapat berpindah dari track ke track lain dengan cepat tanpa perlu memindahkan head. Namun dikarenakan banyaknya head sehingga membuat disk ini harganya sangat mahal.
  - Moveable Head Disk : berbeda dengan Fixed Head Disk, Moveable Head Disk hanya memiliki sebuah head, sehingga head harus bisa berada diatas track manapun. Karena hal ini sehingga lengan head dapat merentang dan kembali ke tujuan tertentu.
- Disk Portability
  - Nonremovable Disk : yaitu disk drive yang permanen, contohnya hardisk dalam sebuah PC adalah Nonremoveable Disk.
  - Removeable Disk : yaitu disk yang dapat dipindahkan atau bisa diganti dengan disk yang lain. Floppy Disk dan ZIP Cartridge adalah contoh Removeable Disk.
- Sides
  - Single Sided : lapisan magnetic berada pada kedua sisi piringan disk.
  - Double Sided : lapisan magnetic berada pada satu sisi piringan.
- Platters
  - Single Platter : magnetic disk dengan satu piringan.
  - Multiple Platter : magnetic disk dengan lebih dari satu piringan, yang tertumpuk secara vertikal. Untuk membaca disk ini diperlukan head dengan banyak lengan, tergantung banyaknya piringan disk, sehingga setiap piringan terdapat lengan head.
- Head Mechanism
  - Contact (Floppy) : mekanisasi head yang ketika proses pembacaan dan penulisan data, head mengalami kontak dengan disk.
  - Fixed Gap : head telah diletakkan dalam tempat yang tetap diatas piringan disk.
  - Aerodynamic Gap (Winchester) : Winchester head dirancang untuk beroperasi lebih dekat ke permukaan disk dari pada yang selazimnya, sehingga membentuk kepadatan data. Headnya biasanya merupakan Aerodinamic Foil yang ringan yang berada diatas permukaan piringan ketika disk berputar. Tekanan udara dari perputaran disk cukup membuat head terangkat ke atas permukaan piringan. Sehingga terbentuklah suatu sistem dimana head berada dekat diatas permukaan piringan, namun tetap tidak melakukan kontak langsung.
- Optical Disk
  - Optical Disk adalah media penyimpanan yang proses pembacaan dan penulisan menggunakan laser. Produk Optical Disk :

- CD (Compact Disk), sebuah nonerasable disk yang menyimpan informasi audio digital. Standar sistemnya menggunakan 12-cm disk dan bisa merekam 60 menit waktu main.
- CD-ROM (CD-Read Only Memory), sebuah nonerasable disk yang digunakan untuk menampung data komputer. Standar sistemnya menggunakan 12-cm disk dan dapat menampung data sebesar 640 megabyte.
- CD-R (CD-Recordable), sama seperti CD-ROM, user hanya bisa menulis data sekali.
- CD-RW (CD-Rewriteable), sama seperti CD-ROM, user dapat menghapus dan menulis ulang data berkali-kali.
- DVD (Digital Video Disk), sebuah teknologi untuk memproduksi digital, yang merekam informasi video. Antara diameter 8 dan 12-cm bisa digunakan dengan kedua sisi disknya, sehingga dapat mendobelkan kapasitasnya menjadi 17 gigabyte. Standar DVD adalah read-only (DVD-ROM)
- DVD-R (DVD-Recordable), sama seperti DVD-ROM, user hanya dapat merekam data satu kali dan hanya pada satu sisi yang dapat digunakan.
- DVD-RW (DVD-Rewriteable), sama seperti DVD-ROM, user hanya dapat merekam data berkali-kali dan hanya satu sisi disk yang dapat digunakan.

### **Magnetic Tapes**

Magnetic Tapes umumnya digunakan untuk membackup data atau mengarcipkan data. Walaupun lebih murah dari Magnetic Disk tetapi proses pengaksesan data lebih lambat, karena menggunakan Sequential Access yang harus mengakses data berurutan dari awal, berbeda dengan Magnetic Disk yang menggunakan Direct Access yang dapat mengakses data dari manapun. Magnetic Tapes memiliki daya tampung data yang besar. Dan dapat di pundahkanke dalam Tape Drive. Tape Jukeboxes dapat menampung data yang sangat besar, seperti misalnya data remote-sensing dari satellit yang dimana bisa mencapai ratusan terabyte (112 byte atau kadang bisa mencapai petabyte (115 byte).

## MODUL 9

### Memori Semikonduktor

**KELAS** : X TKJ  
**GURU MAPEL** : NurRahmat, S.Kom  
**Materi pokok** : Memori Semikonduktor  
**Alokasi waktu** : Jam Pelajaran

#### A. Tujuan Pembelajaran :

- 1 Siswa memahami Pengantar Memori semikonduktor.
- 2 Siswa mampu memahami perbedaan RAM dan ROM
- 3 Siswa mampu mengidentifikasi PROM, EPROM, EEPROM, dan EAPROM
- 4 Siswa mampu mengidentifikasi Alamat dan Data pada memori yang dinyatakan dalam bilangan hexa dan biner

#### B. Materi Pembelajaran

##### Memori Utama Semikonduktor

##### 1. Pengertian

Pada tahun 1970, Fairchild menemukan Memori Utama Semikonduktor dengan ukuran kecil (sebesar 1 sel core memory) dapat menyimpan 256 bits secara Non-Destructive Read. Sehingga lebih cepat dari core memory dan kapasitas meningkat 2 x lipat setiap tahun.

Memori utama semikonduktor sering disebut sebagai inti. Penggunaan keping semikonduktor bagi memori utama hampir universal. Memori utama merupakan media penyimpanan dalam bentuk array yang disusun word atau byte, kapasitas daya simpannya bisa jutaan susunan. Setiap word atau byte mempunyai alamat tersendiri. Data yang disimpan pada memori utama ini bersifat volatile, artinya data yang disimpan bersifat sementara dan dipertahankan oleh sumber-sumber listrik, apabila sumber listrik dimatikan maka datanya akan hilang.

Memori utama digunakan sebagai media penyimpanan data yang berkaitan dengan CPU atau perangkat Input/Output. Elemen dasar suatu memori semikonduktor adalah sel memori. Semua sel memori semikonduktor mempunyai sifat-sifat tertentu : Sel memori memiliki dua keadaan stabil yang dapat digunakan untuk merepresentasikan bilangan biner 1 dan 0. Sel memori mempunyai kemampuan untuk ditulisi untuk menetapkan keadaan. Sel memori mempunyai kemampuan untuk dibaca untuk merasakan keadaan.

##### 2. Peranan dari Memori Utama

Address bus pertama kali mengontak computer yang disebut memori. Yang dimaksud dengan memori disini adalah suatu kelompok chip yang mampu untuk menyimpan instruksi atau data. CPU sendiri dapat melakukan salah satu dari proses berikut terhadap memori tersebut, yaitu membacanya (read) atau menuliskan/menyimpannya (write) ke memori tersebut. Memori ini diistilahkan juga sebagai Memori Utama.

Tipe chip yang cukup banyak dikenal pada memori utama ini DRAM ( Dynamic Random Access Memory ). Kapasitas atau daya tampung dari satu chip ini bermacam-macam, tergantung kapan dan pada komputer apa DRAM tersebut digunakan.

Memori dapat dibayangkan sebagai suatu ruang kerja bagi komputer dan memori juga menentukan terhadap ukuran dan jumlah program yang bias juga jumlah data yang bias diproses. Memori terkadang disebut sebagai primary storage, primary memory, main storage, main memory, internal memory. Ada beberapa macam tipe dari memori komputer, yaitu :

- Random Access Memory (RAM).
- Read Only Memory (ROM).
- CMOS Memory.
- Virtual Memory.

### **Perbedaan dan Fungsi dari RAM dan ROM**

Dalam dunia komputer kita sering mendengar tentang ROM dan RAM. Mungkin dari sebagian kita sudah paham apa perbedaan ROM dengan Ram, tetapi pasti ada juga dari kita yang kurang paham tentang istilah ini. Dalam dunia komputer memang banyak sekali istilah-istilah yang asing menurut kita. Kadang kita yang sudah lama berada lingkungan komputer, maksudnya keseharian kita selalu dengan komputer, ada istilah-istilah yang tidak kita ketahui maksudnya.

Sebagai pengguna komputer, Istilah-istilah program atau sistim kerja komputer kadang tidak begitu kita pikirkan. Kita sebagai pengguna hanya tau bagaimana caranya menggunakan komputer. Mungkin untuk internet, grafis, atau untuk mengelola dokumen di kantor. Tetapi tak ada salahnya juga kita tau istilah-istilah asing pada dunia komputer. Bagi para teknisi komputer, hal-hal ini tentu bukan istilah asing, karena mereka pasti di tuntut untuk mengerti dan mempelajari tentang perangkat komputer dan sistim kerjanya.

Dari segi istilah, ROM dan RAM memiliki pengertian sebagai berikut ini.

- ROM (*Read Only Memory*), Adalah sebuah Ruang atau memory yang berfungsi untuk menyimpan berbagai program yang ada pada komputer tersebut. ROM biasanya menyimpan file-file seperti Musik, Film, Gambar dan file lainnya.
- RAM (*Random Access Memory*), Adalah sebuah Ruang atau memory yang berfungsi sebagai tempat penyimpanan sementara program komputer yang sedang berjalan. ROM biasanya berisi instruksi/program khusus yang bisa digunakan pemakai untuk memanfaatkan komputer secara maksimal.

Dari pengertian diatas, sudah sangat jelas perbedaan antara RAM dan ROM. Secara Singkatnya, ROM adalah ruang yang digunakan untuk menyimpan file yang sudah jadi seperti gambar, musik dan sebagainya. Sedangkan RAM adalah ruang yang digunakan untuk menjalankan aktifitas dari sebuah program yang dibuka pada komputer tersebut.

Di bawah ini terdapat Perbedaan diantara keduanya antara lain:

1. ROM tidak dapat diisi atau ditulisi data sewaktu-waktu seperti RAM. Pengisian atau penulisan data, informasi, ataupun program pada ROM memerlukan proses khusus yang tidak semudah dan se-fleksibel cara penulisan pada RAM. Biasanya, data atau program yang tertulis pada ROM diisi oleh pabrik yang membuatnya. Umumnya ROM digunakan untuk menyimpan

firmware, yaitu perangkat lunak yang berhubungan dengan perangkat keras. Contoh ROM semacam ini adalah ROM BIOS. ROM BIOS berisi program dasar sistem komputer yang berfungsi untuk mengatur dan menyiapkan semua peralatan atau komponen yang ada atau yang terpasang pada komputer saat komputer ‘dinyalakan/dihidupkan’.

2. Informasi/data/program yang tertulis pada ROM (isi ROM) bersifat permanen dan tidak mudah hilang dan tidak mudah berubah walaupun komputer ‘dimatikan’ atau dalam keadaan mati (off). Sedangkan pada RAM, semua isinya (baik berupa data, program atau informasi) akan hilang dengan sendirinya jika komputer ‘dimatikan’ (dalam keadaan off).
3. ROM dapat menyimpan data tanpa membutuhkan daya. Itulah sebabnya data dalam ROM tidak akan hilang walaupun komputer mati. Sedangkan RAM membutuhkan daya agar dapat menyimpan data, jika RAM tidak mendapatkan daya, dengan sendirinya tidak akan dapat menyimpan data. Hal inilah yang menyebabkan data yang terdapat dalam RAM secara otomatis akan hilang bila komputer mati (off).
4. ROM modern sering ditemukan dalam bentuk IC (Integrated Circuit), sama seperti RAM yang wujudnya kebanyakan juga berupa IC. Teks atau kode yang tertulis pada kedua jenis IC ini berbeda. IC ROM biasanya memiliki kode tulisan (teks) 27xxx. Angka 27 menunjukkan kode untuk ROM, sedangkan xxx menunjukkan kapasitas ROM dalam satuan kilo bit.

### **Pengertian EEPROM, PROM, ROM, SRAM**

#### **EEPROM (Electrically Erasable Programmable Read-Only Memory)**

EEPROM merupakan kependekan dari Electrically Erasable Programmable Read-Only Memory. EEPROM adalah tipe khusus dari PROM (Programmable Read-Only Memory) yang bisa dihapus dengan memakai perintah elektrik. Seperti juga tipe PROM lainnya, EEPROM dapat menyimpan isi datanya, bahkan saat listrik sudah dimatikan.

EEPROM sangat mirip dengan flash memory yang disebut juga flash EEPROM. Perbedaan mendasar antara flash memory dan EEPROM adalah penulisan dan penghapusan EEPROM dilakukan dilakukan pada data sebesar satu byte, sedangkan pada flash memory penghapusan dan penulisan data ini dilakukan pada data sebesar satu block. Oleh karena itu flash memory lebih cepat.

Dengan ROM biasa, penggantian BIOS hanya dapat dilakukan dengan mengganti chip. Sedangkan pada EEPROM program akan memberikan instruksi kepada pengendali chip supaya memberikan perintah elektronis untuk kemudian mendownload kode BIOS baru untuk diidkan kepada chip. Hal ini berarti perusahaan dapat dengan mudah mendistribusikan BIOS baru atau update, misalnya dengan menggunakan disket. Hal ini disebut juga flash BIOS.(dna)

#### **PROM (Programmable Read Only Memory)**

PROM (*Programmable Read Only Memory*) merupakan sebuah chip memory yang hanya dapat diisi data satu kali saja. Sekali saja program dimasukkan ke dalam sebuah PROM, maka program tersebut akan berada pada PROM seterusnya. Berbeda halnya dengan RAM, pada PROM data akan tetap ada walaupun komputer dimatikan.

Perbedaan mendasar antara PROM dan ROM (*Read Only Memory*) adalah bahwa PROM diproduksi sebagai memory kosong, sedangkan ROM telah diprogram pada waktu diproduksi. Untuk menuliskan data pada chip PROM, dibutuhkan 'PROM Programmer' atau 'PROM Burner'. EPROM (*Erasable Programmable Read Only Memory*) adalah jenis khusus PROM yang dapat dihapus dengan bantuan sinar ultra violet. Setelah dihapus, EPROM dapat diprogram lagi. EEPROM hampir sama dengan EPROM, hanya saja untuk menghapus datanya memerlukan arus listrik. (dna)

## Pengertian dan Fungsi ROM

### Pengertian ROM

ROM kependekan dari Read Only Memory, yaitu perangkat keras pada komputer berupa chip memori semikonduktor yang isinya hanya dapat dibaca. ROM tidak dapat digolongkan sebagai RAM, walaupun keduanya memiliki kesamaan yaitu dapat diakses secara acak (random). ROM berbeda dengan RAM.

Perbedaan diantara keduanya antara lain:

- ROM tidak dapat diisi atau ditulisi data sewaktu-waktu seperti RAM. Pengisian atau penulisan data, informasi, ataupun program pada ROM memerlukan proses khusus yang tidak semudah dan se-fleksibel cara penulisan pada RAM. Biasanya, data atau program yang tertulis pada ROM diisi oleh pabrik yang membuatnya. Umumnya ROM digunakan untuk menyimpan firmware, yaitu perangkat lunak yang berhubungan dengan perangkat keras. Contoh ROM semacam ini adalah ROM BIOS. ROM BIOS berisi program dasar sistem komputer yang berfungsi untuk mengatur dan menyiapkan semua peralatan atau komponen yang ada atau yang terpasang pada komputer saat komputer 'dinyalakan/dihidupkan'.
- Informasi/data/program yang tertulis pada ROM (isi ROM) bersifat permanen dan tidak mudah hilang dan tidak mudah berubah walaupun komputer 'dimatikan' atau dalam keadaan mati (off). Sedangkan pada RAM, semua isinya (baik berupa data, program atau informasi) akan hilang dengan sendirinya jika komputer 'dimatikan' (dalam keadaan off).
- ROM dapat menyimpan data tanpa membutuhkan daya. Itulah sebabnya data dalam ROM tidak akan hilang walaupun komputer mati. Sedangkan RAM membutuhkan daya agar dapat menyimpan data, jika RAM tidak mendapatkan daya, dengan sendirinya tidak akan dapat menyimpan data. Hal inilah yang menyebabkan data yang terdapat dalam RAM secara otomatis akan hilang bila komputer mati (off).
- ROM modern sering ditemukan dalam bentuk IC (Integrated Circuit), sama seperti RAM yang wujudnya kebanyakan juga berupa IC. Teks atau kode yang tertulis pada kedua jenis IC ini berbeda. IC ROM biasanya memiliki kode tulisan (teks) 27xxx. Angka 27 menunjukkan kode untuk ROM, sedangkan xxx menunjukkan kapasitas ROM dalam satuan kilo bit.

### Fungsi ROM

Seperti telah diungkapkan sebelumnya bahwa umumnya ROM digunakan untuk menyimpan firmware. Pada perangkat komputer, sering ditemukan untuk menyimpan BIOS. Pada saat sebuah komputer dinyalakan, BIOS tersebut dapat langsung dieksekusi dengan cepat, tanpa harus menunggu

untuk menyalakan perangkat media penyimpan lebih dahulu seperti yang umum terjadi pada alat penyimpan lain selain ROM.

Umumnya, pada media simpan lain, jika dieksekusi untuk dibaca isi atau datanya, media simpan tersebut harus dinyalakan lebih dahulu sebelum dibaca, yang tentu saja membutuhkan waktu agak lama. Hal seperti ini tidak terjadi pada ROM.

Pada komputer (PC) modern, BIOS disimpan dalam chip ROM yang dapat ditulisi ulang secara elektrik yang dikenal dengan nama Flash ROM. Itulah sebabnya istilah flash BIOS lebih populer daripada ROM BIOS.

### **Pengertian SRAM**

SRAM (Statik RAM) adalah jenis memory yang tidak perlu penyegaran oleh CPU agar data yang terdapat didalamnya tetap tersimpan dengan baik. RAM jenis ini memiliki kecepatan lebih tinggi dari pada DRAM.