# DIKTAT

# SISTEM DIGITAL

**Di Susun Oleh:**

**Yulianingsih**

**Fitriana  Destiawati**

**UNIVERSITAS INDRAPRASTA PGRI**

**JAKARTA**

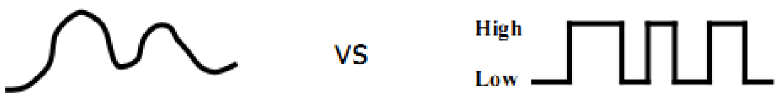
**2013**

**DAFTAR ISI**

## BAB 1. SISTEM DIGITAL

### A. Teori Sistem Digital

Sistem analog maupun digital memproses sinyal-sinyal bervariasi dengan waktu yang memiliki nilai kontinyu/diskrit seperti yang dapat dilihat pada gambar 1 dibawah ini.




Gambar 1 : sinyal analog vs sinyal diskrit

Beberapa keuntungan system digital daripada system analog adalah :

1. Kemampuan memproduksi sinyal dengan baik dan akurat
2. Mempunyai reliabilitas yang lebih baik (noise lebih rendah akibat imunitas yang lebih baik)
3. Mudah didisain, tidak memerlukan kemampuan matematika khusus untuk memvisualisasikan sifat-sifat rangkaian digital yang sederhana.
4. Fleksibilitas dan fungsionalitas yang lebih baik
5. Kemampuan pemrograman yang lebih mudah
6. Lebih cepat
7. Ekonomis jika dilihat dari segi biaya

Sistem digital menggunakan kombinasi-kombinasi biner benar & salah untuk menyerupai cara ketika menyelesaika masalah sehingga disebut juga logika-logika kombinasional. Dengan system digital dapat digunakan langkah-langkah berpikir logis atau keputusan-keputusan masa lalu (memori) untuk menyelesaikan masalah sehingga biasa disebut logika-logika sekuensial (terurut).

Logika digital dapat direpresentasikan dengan beberapa cara yaitu :

- Tabel kebenaran, menyediakan suatu daftar setiap kombinasi yang mungkin dari masukan-masukan biner pada sebuah rangkaian digital dan keluaran yang terkait
- Ekspresi-ekspresi Boolean mengekspresikan logika pada sebuah format fungsional
- Diagram gerbang logika
- Diagram penempatan bagian
- *High Level Description Language.*

## B. Teori Sistem Bilangan

### 1. Bilangan Biner

Bilangan biner merupakan bilangan berbasis 2. Bilangan yang termasuk kedalam bilangan biner hanya 0 dan 1. Contoh: $1011_2$ , $110.11_2$

### 2. Bilangan Desimal

Bilangan oktal merupakan bilangan berbasis 10. Bilangan yang termasuk kedalam bilangan oktal adalah 0, 1, 2, 3, 4, 5, 6, 7,8 dan 9. Contoh : $19_{10}$ , $12.25_{10}$

### 3. Bilangan Oktal

Bilangan desimal merupakan bilangan berbasis 8. Bilangan yang termasuk kedalam bilangan desimal adalah 0, 1, 2, 3, 4, 5, 6, 7. Contoh : $176_8$ , $127.75_8$

### 4. Bilangan Heksadesimal

Bilangan heksadesimal merupakan bilangan berbasis 16. Bilangan yang termasuk kedalam bilangan heksadesimal adalah 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F. Contoh : $4A3_{16}$ , $45C.9D_{16}$

**LATIHAN 1**

1. Sebutkan masing-masing jenis bilangan dibawah ini:
   a. $109_{10}$
   b. $201_{8}$
   c. $679B_{16}$
   d. $10111_{2}$
2. Menurut anda penulisan bilangan dibawah ini benar atau salah? Jelaskan pendapat anda!
   a. $101111_{10}$
   b. $765_{2}$
   c. $698A_{16}$
   d. $328_{8}$
   e. $110111_{16}$
   f. $4AB39_{10}$

# BAB 2. KONVERSI BILANGAN

## A. Konversi Bilangan Biner ke Bilangan Desimal

Konversi bilangan biner ke bilangan desimal dengan cara mengalikan digit bilangan biner dengan 2 pangkat. Kemudian hasil perkalian masing-masing digitnya dijumlahkan.

Contoh 1 : Konversikan bilangan biner $10011_2$ ke dalam bentuk desimal

1 0 0 1 1

$$\begin{array}{r} 1\times2^4=16 \\ 0\times2^3=\ 0 \\ 0\times2^2=\ 0 \\ 1\times2^1=2 \\ \underline{1\times2^0=1} \\ 18 \end{array} +$$

Pangkat yang paling kecil diberikan untuk digit yang paling belakang, sedangkan pangkat yang paling besar diberikan untuk digit yang paling depan.

Dari perhitungan disamping maka $10011_2 = 18_{10}$

Contoh 2 : Konversikan bilangan biner $10011.01_2$ ke dalam bentuk desimal

Untuk menyelesaikan soal tersebut, pisahkan digit-digit di depan koma dan digit-digit dibelakang koma. Untuk digit di depan koma, selesaikan dengan menggunakan perkalian dengan 2 berpangkat positif. Sedangkan untuk digit dibelakang koma, selesaikan menggunakan dengan 2 berpangkat negatif.

10011 → 01 → Dimana :

$$\begin{array}{r} 1\times2^4=16 \\ 0\times2^3=\ 0 \\ 0\times2^2=\ 0 \\ 1\times2^1=2 \\ \underline{1\times2^0=1} \\ 18 \end{array} + \qquad \begin{array}{r} 0\times2^{-1}=\quad 0 \\ \underline{1\times2^{-2}=0{,}25} \\ 0{,}25 \end{array} +$$

$$2^{-1} = \frac{1}{2^1} = \frac{1}{2} = 0{,}5$$

$$2^{-2} = \frac{1}{2^2} = \frac{1}{4} = 0{,}25$$

Maka, 100112 = 18 dan 012 = 0.25 → 10011.012 = 18.2510

## B. Konversi Bilangan Biner ke Bilangan Oktal

Konversi bilangan biner ke bilangan oktal dengan cara mengelompokkan 3 digit bilangan biner mulai dari digit paling belakang (LSB = *Least Significant Bit*) sampai digit yang paling depan (MSB = *Most Significant Bit*).

Contoh : Konversikan bilangan biner 1110011012 ke dalam bentuk octal.

Jika dikelompokkan, maka didapatkan pengelompokkan sebagai berikut

$$\begin{array}{ccc} \underbrace{111} & \underbrace{001} & \underbrace{101} \\ \downarrow & \downarrow & \downarrow \\ a & b & c \\ \downarrow & \downarrow & \downarrow \\ 7 & 1 & 5 \end{array}$$

Dari masing-masing kelompok digit biner, dikonversikan ke bentuk desimal

a : 101 =

$$\begin{array}{r} 1 \times 2^2 = 4 \\ 0 \times 2^1 = 0 \\ \underline{1 \times 2^0 = 1} \; + \\ 5 \end{array}$$

b : 001 =

$$\begin{array}{r} 0 \times 2^2 = 0 \\ 0 \times 2^1 = 0 \\ \underline{1 \times 2^0 = 1} \; + \\ 1 \end{array}$$

c : 111 =

$$\begin{array}{r} 1 \times 2^2 = 4 \\ 1 \times 2^1 = 1 \\ \underline{1 \times 2^0 = 1} \; + \\ 7 \end{array}$$

Maka, 1110011012 = 7158

## C. Konversi Bilangan Biner ke Bilangan Heksadesimal

Konversi bilangan biner ke bilangan heksadesimal dengan cara mengkelompokkan 4 digit biner, mulai dari LSB sampai dengan MSB.

Contoh : Konversikan bilangan biner 100110110012 ke bentuk heksadesimal

Jika dikelompokan, maka didapatkan pengelompokan sebagai berikut :

$$\begin{array}{ccc} \underbrace{100} & \underbrace{1101} & \underbrace{1001} \rightarrow \\ \downarrow & \downarrow & \downarrow \\ a & b & c \end{array}$$

Jika dilihat pada kelompok 3 hanya terdapat 3 digit biner, maka untuk melengkapi menjadi 4 digit biner, maka pada kelompok 3 ditambahkan 1 digit dipaling depan dengan digit 0.

Maka, bentuk pengelompokkannya adalah 0100 1101 1001.

$$\begin{array}{ccc} \underbrace{0100} & \underbrace{1101} & \underbrace{1001} \rightarrow \\ \downarrow & \downarrow & \downarrow \\ a & b & c \\ \downarrow & \downarrow & \downarrow \\ 4 & D & 9 \end{array}$$

Sama halnya dengan mengkonversi dari biner ke octal, dari pengelompokkan digit-digit biner tersebut diubah terlebih dahulu kedalam bentuk desimal

| a : 1001 = | b : 1101 = | c : 0100 = |
|---|---|---|
| $1\times2^3=8$ | $1\times2^3=8$ | $0\times2^3=0$ |
| $0\times2^2=0$ | $1\times2^2=4$ | $1\times2^2=4$ |
| $0\times2^1=0$ | $0\times2^1=0$ | $0\times2^1=0$ |
| $1\times2^0=1$ + | $1\times2^0=1$ + | $0\times2^0=0$ + |
| 9 | 13 | 4 |

Pada bilangan heksadesimal, 13 = D. Maka, $10011011001_2 = 4D9_{16}$

## D. Konversi Bilangan Desimal ke Bilangan Biner

Mengubah sebuah bilangan desimal kedalam bentuk bilangan biner yaitu dengan cara membagi 2 bilangan desimal dengan menggunakan operator mod dimana yang ditulis adalah sisa dari pembagiannya.

Contoh 1 : konversikan bilangan desimal $8_{10}$ kedalam bentuk biner.

$$\frac{10}{2} = 5\ sisa\ 0$$

$$\frac{5}{2} = 2\ sisa\ 1$$

$$\frac{2}{2} = 1\ sisa\ 0$$

Sisa pembagian ditulis dari bawah keatas, maka $10_{10}$ = $1010_2$

Contoh 2 : konversikan bilangan 10.2510 kedalam bentuk biner

Untuk menkonversikan bilangan desimal tersebut ada 2 langkah. Langkah pertama adalah mengkonversikan angka 10 dengan membaginya dengan 2 sedangkan langkah kedua adalah mengkonversikan angka 0.25 dengan cara mengalikan dengan 2.

Langkah 1 :

$$\frac{10}{2} = 5\ sisa\ 0$$
$$\frac{5}{2} = 2\ sisa\ 1$$
$$\frac{2}{2} = 1\ sisa\ 0$$

Sisa pembagian ditulis dari bawah ke atas, maka $10_{10} = 1010_2$

Langkah 2 :

$$0.25 \times 2 = 0.5 \rightarrow 0$$
$$0.5 \times 2 = 1 \rightarrow 1$$

Ditulis dari atas ke bawah, maka $0.25_{10} = 01_2$

Dari langkah 1 dan langkah 2 didapatkan hasil $10_{10} = 1010_2$ dan $0.25_{10} = 01_2$ , maka $10.25_{10} = 1010.01_2$

## E. Konversi Bilangan Desimal ke Bilangan Oktal

Untuk mengubah sebuah bilangan desimal kedalam bentuk bilangan oktal, hampir sama seperti mengkonversikan kedalam bentuk biner yaitu dengan cara membagi 8 bilangan desimal dengan menggunakan operator mod dimana yang ditulis adalah sisa dari pembagiannya.

Contoh 1 : konversikan bilangan desimal $544_{10}$ kedalam bentuk oktal.

$$\frac{544}{8} = 68\ sisa\ 0$$
$$\frac{68}{8} = 8\ sisa\ 4$$
$$\frac{8}{8} = 1\ sisa\ 0$$

Sisa pembagian ditulis dari bawah keatas, maka $544_{10} = 1040_8$

Contoh 2 : konversikan bilangan 544.2510 kedalam bentuk oktal

Untuk menkonversikan bilangan desimal tersebut ada 2 langkah. Langkah pertama adalah mengkonversikan angka 544 dengan membaginya dengan 8 sedangkan langkah kedua adalah mengkonversikan angka 0.25 dengan cara mengalikan dengan 8.

Langkah 1 :

$$\frac{544}{8} = 68\ sisa\ 0$$

$$\frac{68}{8} = 8\ sisa\ 4$$

$$\frac{8}{8} = 1\ sisa\ 0$$

Sisa pembagian ditulis dari bawah ke atas, maka 54410 = 10408

Langkah 2 :

$$0.25 \times 8 = 1 \rightarrow 1$$

Ditulis dari atas ke bawah, maka 0.2510 = 18

Dari langkah 1 dan langkah 2 didapatkan hasil 1010 = 10408 dan 0.2510 = 18 , maka 10.2510 = 1040.18

## F. Konversi Bilangan Desimal ke Bilangan Heksadesimal

Untuk mengubah sebuah bilangan desimal kedalam bentuk bilangan heksadesimal, hampir sama seperti mengkonversikan kedalam bentuk biner dan oktal yaitu dengan cara membagi 16 bilangan desimal dengan menggunakan operator mod dimana yang ditulis adalah sisa dari pembagiannya.

Contoh 1 : konversikan bilangan desimal $4256_{10}$ ke bentuk heksadesimal.

$$\frac{4256}{16} = 266\ sisa\ 0$$

$$\frac{266}{16} = 16\ sisa\ 10 \rightarrow A$$

$$\frac{16}{16} = 1\ sisa\ 0$$

Sisa pembagian ditulis dari bawah keatas, maka $4256_{10}$ = $10A0_{16}$

Contoh 2 : konversikan bilangan 4256.2510 kedalam bentuk heksadesimal

Untuk menkonversikan bilangan desimal tersebut ada 2 langkah. Langkah pertama adalah mengkonversikan angka 4256 dengan membaginya dengan 16 sedangkan langkah kedua adalah mengkonversikan angka 0.25 dengan cara mengalikan dengan 16.

Langkah 1 :

$$\frac{4256}{16} = 266\ sisa\ 0$$

$$\frac{266}{16} = 16\ sisa\ 10 \rightarrow A$$

$$\frac{16}{16} = 1\ sisa\ 0$$

Sisa pembagian ditulis dari bawah ke atas, maka 425610 = 10A016

Langkah 2 :

$$0.25 \times 16 = 4 \rightarrow 4$$

Ditulis dari atas ke bawah, maka 0.2510 = 416

Dari langkah 1 dan langkah 2 didapatkan hasil 425610 = 10A016 dan 0.2510 = 416 , maka 10.2510 = 10A0.416

## G. Konversi Bilangan Oktal ke Bilangan Desimal

Konversi bilangan oktal ke bilangan desimal dengan cara mengalikan digit bilangan oktal dengan 8 pangkat. Kemudian hasil perkalian masing-masing digitnya dijumlahkan.

Contoh 1 : Konversikan bilangan oktal $45_8$ ke dalam bentuk desimal

$$
\begin{array}{l}
4\ 5 \\
\rightarrow 4\times8^1 = 32 \\
\rightarrow \underline{5\times8^0 = \ \ 5} \ + \\
\qquad\qquad\ \ 37
\end{array}
$$

Pangkat yang paling kecil diberikan untuk digit yang paling belakang, sedangkan pangkat yang paling besar diberikan untuk digit yang paling depan.

Dari perhitungan disamping maka $45_8 = 37_{10}$

Contoh 2 : Konversikan bilangan oktal 45.678 ke dalam bentuk desimal

Untuk menyelesaikan soal tersebut, pisahkan digit-digit di depan koma dan digit-digit dibelakang koma. Untuk digit di depan koma, selesaikan dengan menggunakan perkalian dengan 8 berpangkat positif. Sedangkan untuk digit dibelakang koma, selesaikan menggunakan dengan 8 berpangkat negatif.

45 →

$$\begin{array}{r} 4\times8^1=32 \\ 5\times8^0=\ \ 5 \\ \hline 37 \end{array} +$$

0.67 →

$$\begin{array}{r} 6\times8^{-1}=\quad 0.75 \\ 7\times8^{-2}=0.109375 \\ \hline 0.859375 \end{array} +$$

Dimana :

$$8^{-1} = \frac{1}{8^1} = \frac{1}{8} = 0.125$$

$$8^{-2} = \frac{1}{8^2} = \frac{1}{64} = 0.015625$$

Maka, $45_8 = 37$ dan $0.67_8 = 0.859375$ → $45.67_8 = 37.859375_{10}$

## H. Konversi Bilangan Oktal ke Bilangan Biner

Konversi bilangan oktal ke bilangan biner dengan cara menjadikan 1 digit bilangan oktal menjadi 3 digit bilangan biner mulai dari digit paling belakang (LSB = *Least Significant Bit*) sampai digit yang paling depan (MSB = *Most Significant Bit*).

Contoh : Konversikan bilangan oktal $327_8$ ke dalam bentuk biner.

Jika dikelompokkan, maka didapatkan pengelompokkan sebagai berikut :

| 3 | 2 | 7 |
|---|---|---|
| ↓ | ↓ | ↓ |
| a | b | c |
| ↓ | ↓ | ↓ |
| 011 | 010 | 111 |

Dari masing-masing kelompok digit oktal, dikonversikan ke bentuk desimal

a →3 =

$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 3 = 11 \rightarrow 011$$

b → 2 =

$$\frac{2}{2} = 1\ sisa\ 0$$
$$maka\ 2 = 10 \rightarrow 010$$

c → 7 =

$$\frac{7}{2} = 3\ sisa\ 1$$
$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 7 = 111$$

Maka, $327_8 = 011010111_2$ → $11010111_2$

## I. Konversi Bilangan Oktal ke Bilangan Heksadesimal

Mengubah sebuah bilangan oktal kedalam bentuk heksadesimal dengan cara mengubahnya terlebih dahulu kedalam bentuk biner kemudian dari bentuk biner diubah kembali kedalam bentuk heksa desimal.

Contoh : konversikan bilangan oktal 3278 kedalam bentuk heksadesimal

Langkah 1 : konversikan dahulu bilangan oktal 3278 ke bentuk biner

| 3 | 2 | 7 |
|---|---|---|
| ↓ | ↓ | ↓ |
| a | b | c |
| ↓ | ↓ | ↓ |
| 011 | 010 | 111 |

a → 3 =

$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 3 = 11 \rightarrow 011$$

b → 2 =

$$\frac{2}{2} = 1\ sisa\ 0$$
$$maka\ 2 = 10 \rightarrow 010$$

c → 7 =

$$\frac{7}{2} = 3\ sisa\ 1$$
$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 7 = 111$$

Maka, $327_8 = 011010111_2 \rightarrow 11010111_2$

Langkah 2 : konversikan kembali bilangan biner $11010111_2$ ke bentuk heksadesimal

| 1101 | 0111 |
|---|---|
| ↓ | ↓ |
| a | b |
| ↓ | ↓ |
| 13(D) | 7 |

a → 1101 =

$$\begin{aligned} 1 \times 2^3 &= 8 \\ 1 \times 2^2 &= 4 \\ 0 \times 2^1 &= 0 \\ 1 \times 2^0 &= 1 \quad + \\ \hline &13 \end{aligned}$$

b → 0111 =

$$\begin{aligned} 0 \times 2^3 &= 0 \\ 1 \times 2^2 &= 4 \\ 1 \times 2^1 &= 2 \\ 1 \times 2^0 &= 1 \quad + \\ \hline &7 \end{aligned}$$

Maka, $327_8 = D7_{16}$

## J. Konversi Bilangan Heksadesimal ke Bilangan Desimal

Konversi bilangan heksadesimal ke bilangan desimal dengan cara mengalikan digit bilangan oktal dengan 16 pangkat. Kemudian hasil perkalian masing-masing digitnya dijumlahkan.

Contoh 1 : Konversikan bilangan heksadesimal $4A_{16}$ ke bentuk desimal

4 A

$$\begin{array}{r} 4\times16^{1}=64 \\ \underline{10\times16^{0}=10} \; + \\ 74 \end{array}$$

Pangkat yang paling kecil diberikan untuk digit yang paling belakang, sedangkan pangkat yang paling besar diberikan untuk digit yang paling depan. A bernilai 10

Dari perhitungan disamping maka $4A_{16} = 74_{10}$

Contoh 2 : Konversikan bilangan heksadesimal $45.10_{16}$ ke bentuk desimal

Untuk menyelesaikan soal tersebut, pisahkan digit-digit di depan koma dan digit-digit dibelakang koma. Untuk digit di depan koma, selesaikan dengan menggunakan perkalian dengan 16 berpangkat positif. Sedangkan untuk digit dibelakang koma, selesaikan menggunakan dengan 16 berpangkat negatif.

4A →

$$\begin{array}{r} 4\times16^{1}=64 \\ \underline{10\times16^{0}=10} \; + \\ 74 \end{array}$$

0.10 →

$$\begin{array}{r} 1\times16^{-1}= 0.0625 \\ \underline{0\times16^{-2}= \quad 0} \; + \\ 0.0625 \end{array}$$

Dimana :

$$8^{-1} = \frac{1}{16^{1}} = \frac{1}{8} = 0.0625$$

$$8^{-2} = \frac{1}{16^{2}} = \frac{1}{64} = 0.00390625$$

Maka, $4A_{16} = 74$ dan $0.10_{16} = 0.0625$ → $45.10_{16} = 74.0625_{10}$

## K. Konversi Bilangan Heksadesimal ke Bilangan Biner

Konversi bilangan heksadesimal ke bilangan biner dengan cara menjadikan 1 digit bilangan heksadesimal menjadi 4 digit bilangan biner mulai dari digit paling belakang (LSB = *Least Significant Bit*) sampai digit yang paling depan (MSB = *Most Significant Bit*).

Contoh : Konversikan bilangan oktal 32716 ke dalam bentuk biner.

Jika dikelompokkan, maka didapatkan pengelompokkan sebagai berikut

$$\begin{array}{ccc} 3 & 2 & 7 \\ \downarrow & \downarrow & \downarrow \\ a & b & c \\ \downarrow & \downarrow & \downarrow \\ 0011 & 0010 & 0111 \end{array}$$

Dari masing-masing kelompok digit heksadesimal, dikonversikan ke bentuk desimal

Maka, $327_{16} = 001100100111_2 \rightarrow 1100100111_2$

a → 3 =

$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 3 = 11 \rightarrow 0011$$

b → 2 =

$$\frac{2}{2} = 1\ sisa\ 0$$
$$maka\ 2 = 10 \rightarrow 0010$$

c → 7 =

$$\frac{7}{2} = 3\ sisa\ 1$$
$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 7 = 0111$$

**L. Konversi Bilangan Heksadesimal ke Bilangan Oktal**

Mengubah sebuah bilangan heksadesimal kedalam bentuk oktal dengan cara mengubahnya terlebih dahulu kedalam bentuk biner kemudian dari bentuk biner diubah kembali kedalam bentuk oktal.

Contoh : konversikan bilangan oktal $327_{16}$ kedalam bentuk heksadesimal

Langkah 1 : konversikan dahulu bilangan hksadesimal $327_{16}$ ke bentuk biner

$$\begin{array}{ccc} 3 & 2 & 7 \\ \downarrow & \downarrow & \downarrow \\ a & b & c \\ \downarrow & \downarrow & \downarrow \\ 0011 & 0010 & 0111 \end{array}$$

Maka, $327_8$ = = $001100100111_2 \rightarrow 1100100111_2$

a → 3 =

$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 3 = 11 \rightarrow 0011$$

b → 2 =

$$\frac{2}{2} = 1\ sisa\ 0$$
$$maka\ 2 = 10 \rightarrow 0010$$

c → 7 =

$$\frac{7}{2} = 3\ sisa\ 1$$
$$\frac{3}{2} = 1\ sisa\ 1$$
$$maka\ 7 = 0111$$

Langkah 2 : konversikan kembali bilangan biner = $1100100111_2$ ke bentuk oktal

$$\underbrace{1}_{\substack{\downarrow\\ a}}\ \underbrace{100}_{\substack{\downarrow\\ b}}\ \underbrace{100}_{\substack{\downarrow\\ c}}\ \underbrace{111}_{\substack{\downarrow\\ d}} \rightarrow \underbrace{001}_{\substack{\downarrow\\ a\\ \downarrow\\ 1}}\ \underbrace{100}_{\substack{\downarrow\\ b\\ \downarrow\\ 4}}\ \underbrace{100}_{\substack{\downarrow\\ c\\ \downarrow\\ 4}}\ \underbrace{111}_{\substack{\downarrow\\ d\\ \downarrow\\ 7}}$$

Maka, $327_{16} = 1447_8$

a → 001 = b & c → 100 = d → 111 =

$$\begin{array}{r} 0 \times 2^2 = 0 \\ 0 \times 2^1 = 0 \\ \underline{1 \times 2^0 = 1} \\ 1 \end{array} + \qquad \begin{array}{r} 1 \times 2^2 = 4 \\ 0 \times 2^1 = 0 \\ \underline{0 \times 2^0 = 0} \\ 4 \end{array} + \qquad \begin{array}{r} 1 \times 2^2 = 4 \\ 1 \times 2^1 = 2 \\ \underline{1 \times 2^0 = 1} \\ 7 \end{array} +$$

**LATIHAN 2**

1. Konversikan bilangan dibawah ini ke bentuk biner
   a. $23_{10}$
   b. $3AB_{16}$
   c. $675_{8}$
   d. $35.75_{10}$
2. Konversikan bilangan dibawah ini ke bentuk oktal
   a. $679.25_{10}$
   b. $67BD_{16}$
   c. $101110_{2}$
   d. $92_{10}$
3. Konversikan bilangan dibawah ini ke bentuk desimal
   a. $10111.11_{2}$
   b. $1110_{2}$
   c. $64.75_{8}$
   d. $651_{8}$
   e. $7A9_{16}$
   f. $9BF.43_{16}$
4. Konversikan bilangan dibawah ini ke bentuk heksadesimal
   a. $10111001_{2}$
   b. $571_{8}$
   c. $987_{10}$
   d. $710.25_{10}$

# BAB 3. OPERASI ARITMATIKA

## A. Operasi Penjumlahan

### 1. Penjumlahan Biner

Dasar-dasar dari operasi penujmlahan bilangan biner adalah sebagai berikut :

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 → dengan carry of 1, yaitu 1 + 1 = 2, karena digit terbesar binari 1, maka harus dikurangi dengan 2 (basis), jadi 2 – 2 = 0 dengan carry of 1.

Contoh : $101_2 + 11_2 = \ldots$

$$\begin{array}{r} 101 \\ 11 \\ \hline 1000 \end{array} +$$

Catatan :

1+1=2 → dianggap $2_{10}$ → $2_{10} = 10_2$ → 0 ditulis dan 1 disimpan untuk angka didepannya

0+1=1+1(hasil simpan)=2 → $2_{10} = 10_2$ → 0 ditulis dan 1 disimpan untuk angka didepannya

1+1(hasil simpan)=2 → $2_{10} = 10_2$

Maka, $101_2 + 11_2 = 1000_2$

### 2. Penjumlahan Oktal

Langkah-langkah operasi penjumlahan pada bilangan octal adalah sebagai berikut:

- tambahkan masing-masing kolom secara desimal
- rubah dari hasil desimal ke octal
- tuliskan hasil dari digit paling kanan dari hasil octal
- kalau hasil penjumlahan tiap-tiap kolom terdiri dari dua digit, maka digit paling kiri merupakan carry of untuk penjumlahan kolom selanjutnya.

Contoh : $37_8 + 16_8 = \ldots$

$$\begin{array}{r} 37 \\ \underline{16} \\ 55 \end{array} +$$

Catatan :

→ 7+6=13 → dianggap $13_{10}$ → $13_{10} = 15_8$ → 5 ditulis dan 1 disimpan untuk angka didepannya

→ 3+1=4+1(hasil simpan)= 5

Maka, $37_8 + 16_8 = 55_8$

### 3. Penjumlahan Heksadesimal

Penjumlahan bilangan heksadesimal dapat dilakukan secara sama dengan penjumlahan bilangan oktal, dengan langkah-langkah sebagai berikut:

- tambahkan masing-masing kolom secara desimal
- rubah dari hasil desimal ke heksadesimal
- tuliskan hasil dari digit paling kanan dari hasil heksadesimal
- kalau hasil penjumlahan tiap-tiap kolom terdiri dari dua digit, maka digit paling kiri merupakan carry of untuk penjumlahan kolom selanjutnya.

Contoh : $37_{16} + 1A_{16} = \ldots$

$$\begin{array}{r} 37 \\ \underline{1A} \\ 51 \end{array} +$$

Catatan :

→ 7+A(10)=17 → dianggap $17_{10}$ → $17_{10} = 11_{16}$ → 1 ditulis dan 1 disimpan untuk angka didepannya

→ 3+1=4+1(hasil simpan)= 5

Maka, $37_{16} + 1A_{16} = 51_{16}$

## B. Operasi Pengurangan

### 1. Pengurangan Biner

Bilangan biner dikurangkan dengan cara yang sama dengan pengurangan bilangan desimal. Dasar pengurangan untuk masing-masing digit bilangan biner adalah :

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 – 1 = 1 → dengan borrow of 1, (pinjam 1 dari posisi sebelah kirinya).

Pada saat pinjam 1, digit yang meminjam akan bertambah 2 sedangkan digit yang dipinjam akan berkurang 1.

Contoh : $101_2 - 11_2 = \ldots$

$$\begin{array}{r} 101 \\ 11 \\ \hline 10 \end{array} -$$

Catatan :

→ 1-1=0

→ 0-1=tidak bisa → pinjam 1 digit didepan → 0+2=2 → 2-1=1

→ Angka 1 didepan sudah dipinjam dengan angka 0 dibelakangnya→ 1-1=0

Maka, $101_2 + 11_2 = 1000_2$

### 2. Pengurangan Oktal

Pengurangan Oktal dapat dilaukan secara sama dengan pengurangan bilangan desimal. Pada saat pinjam 1, digit yang meminjam akan bertambah 8 sedangkan digit yang dipinjam akan berkurang 1.

Contoh : $35_8 - 17_8 = \dots$

$$\frac{\begin{array}{c}35\\17\end{array}}{16} -$$

Catatan :

5-7= tidak bisa → pinjam 1 → 5+8 =13 → 13-7=6

Angka 3 sudah dipinjam 1 dengan angka dibelakangnya → 3-1=2 → 2-1=1

Maka, $35_8 - 17_8 = 16_8$

### 3. Pengurangan Heksadesimal

Pengurangan heksadesimal dapat dilakukan secara sama dengan pengurangan bilangan desimal. Pada saat pinjam 1, digit yang meminjam akan bertambah 16 sedangkan digit yang dipinjam akan berkurang 1.

Contoh : $37_{16} - 1A_{16} = \dots$

$$\frac{\begin{array}{c}37\\1A\end{array}}{1D} -$$

Catatan :

7-A(10)=tidak bisa → pinjam 1 → 7+16=23 → 23-A(10)=13 → 13=D

Angka 3 sudah dipinjam 1 dengan angka dibelakangnya → 3-1=2 → 2-1=1

Maka, $37_{16} - 1A_{16} = 1D_{16}$

## C. Operasi Perkalian

### 1. Perkalian Biner

Dilakukan sama dengan cara perkalian pada bilangan desimal. Dasar perkalian bilangan biner adalah :

0 x 0 = 0

1 x 0 = 0

0 x 1 = 0

1 x 1 = 1

Contoh : $101_2$ x $11_2$ = …

$$\begin{array}{r} 101 \\ 11 \\ \hline 101 \\ 101\phantom{0} \\ \hline 1111 \end{array}$$

→ penjumlahan biner

Maka, $101_2$ x $11_2$ = $1111_2$

**2. Perkalian Oktal**

Langkah-langkah operasi perkalian pada bilangan oktal atau bilangan berbasis 8 adalah sebagai berikut :

- kalikan masing-masing kolom secara desimal
- rubah dari hasil desimal ke octal
- tuliskan hasil dari digit paling kanan dari hasil octal
- kalau hasil perkalian tiap kolom terdiri dari 2 digit, maka digit paling kiri merupakan carry of untuk ditambahkan pada hasil perkalian kolom selanjutnya.
- Pada perkalian oktal juga terdapat penjumlahan oktal didalamnya.

Contoh : $37_8$ x $16_8$ = …

$$\begin{array}{r} 37 \\ 16 \\ \hline 272 \\ 37\phantom{0} \\ \hline 662 \end{array}$$

Catatan :

7x6=42 → dianggap $42_{10}$ → $42_{10}$ = $52_8$ → 2 ditulis dan 5 disimpan untuk angka didepannya

3x6=18+5(hasil simpan)=23 → dianggap $23_{10}$ → $23_{10}$ = $27_8$

Maka, $37_8$ x $16_8$ = $662_8$

**3. Perkalian Heksadesimal**

Langkah-langkah operasi perkalian pada bilangan heksadesimal adalah sebagai berikut :

- kalikan masing-masing kolom secara desimal
- rubah dari hasil desimal ke octal
- tuliskan hasil dari digit paling kanan dari hasil octal

- kalau hasil perkalian tiap kolol terdiri dari 2 digit, maka digit paling kiri merupakan carry of untuk ditambahkan pada hasil perkalian kolom selanjutnya.
- Pada perkalian heksadesimal juga terdapat operasi penjumlahan heksadesimal

Contoh : $37_{16}$ x $1A_{16}$ = …

$$\begin{array}{r} 37 \\ \underline{1A} \times \\ 226 \\ \underline{37\phantom{0}} + \\ 596 \end{array}$$

Catatan :

7xA(10)=70 → dianggap $70_{10}$ → $70_{10} = 46_{16}$ → 6 ditulis dan 4 disimpan untuk angka didepannya

3xA(10)=30+4(hasil simpan)=34 → dianggap $34_{10}$ → $34_{10} = 22_{16}$

Maka, $37_{16}$ x $1A_{16} = 596_{16}$

## D. Operasi Pembagian

### 1. Pembagian Biner

Pembagian bilangan biner dilakukan juga dengan cara yang sama dengan bilangan desimal. Pembagian biner 0 tidak mempunyai arti, sehingga dasar pemagian biner adalah 0 : 1 = 0 dan 1 : 1 = 1

Contoh : $1111101_2$ : $101_2$ = …

```
101 / 1111101 \ 11001
      101 -
      -------
        101
        101 -
        --------
          0101
            101 -
            -------
              0
```

maka, $1111101_2 : 101_2 = 11001_2$

## 2. Pembagian Oktal

Pembagian bilangan oktal dilakukan juga dengan cara yang sama dengan bilangan desimal.

Contoh : $250_8 : 14_8 = \ldots$

14 / 250 \ 16

14 - ← $14_8 \times 1_8 = 14_8$

110

110 - ← $14_8 \times 6_8 = 4_8 \times 6_8 = 30_8$

0 $\qquad 1_8 \times 6_8 = 6_8$ +

$110_8$

maka, $250_8 : 14_8 = 16_8$

## 3. Pembagian Heksadesimal

Pembagian bilangan heksadesimal dilakukan juga dengan cara yang sama dengan bilangan desimal.

Contoh : $1214_{16} : 1B_{16} = \ldots$

1B / 1214 \ AC

10E - ← $1B_{16} \times A_{16} = 27_{10} \times 10_{10} = 270_{10} = 10E_{16}$

144

144- ← $1B_{16} \times C_{16} = 27_{10} \times 12_{10} = 324_{10} = 144_{16}$

0

Maka, $1214_{16} : 1B_{16} = AC_{16}$

**LATIHAN 3**

1. Hhitunglah operasi penjumlahan dibawah ini :
   a. $267_8 + 345_8 =$
   b. $AC7_{16} + 35D_{16} =$
   c. $11101_2 + 110_2 =$
2. Hitunglah operasi pengurangan dibawah ini :
   a. $65A_{16} – 34B_{16} =$
   b. $1000_2 – 111_2 =$
   c. $327_8 – 77_8 =$
3. Hitunglah operasi perkalian dibawah ini :
   a. $798_{16}$ x $23_{16} =$
   b. $57_8$ x $71_8 =$
   c. $10011_2$ x $101_2 =$
4. Hitunglah operasi pembagian dibawah ini :
   a. $10011_2 : 10_2 =$
   b. $657_{16} : 12_{16} =$
   c. $523_8 : 4_8 =$

# BAB 4. BINARY CODED DECIMAL

Binary coded decimal atau BCD merupakan  suatu sistem bilangan yang menggunakan kode biner 4 bit untuk merepresentasikan bilangan desimal 0 sampai 9. Bilangan yang lebih besar dari bilangan ini dinyatakan dengan 2 atau lebih kelompok bilangan biner 4 bit.

Nibble adalah string dari 4 bit. Bilangan BCD (Binary-coded-desimal) mengungkapkan setiap digit desimal sebagai sebuah nibble. Pada penjumlahan bilangan BCD yang hasilnya lebih besar dari 9 ( 1001 ) maka harus ditambahkan 6 atau 0110.

| Desimal | Sandi 8421 | Sandi 2421 |
|---|---|---|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0010 |
| 3 | 0011 | 0011 |
| 4 | 0100 | 0100 |
| 5 | 0101 | 1011 |
| 6 | 0110 | 1100 |
| 7 | 0111 | 1101 |
| 8 | 1000 | 1110 |
| 9 | 1001 | 1111 |

Pada umumnya yang digunakan untuk standarisasi bilangan BCD adalah menggunakan sandu 8421.

Contoh : tentukan bilangan BCD dari $873_{10}$

$$\left.\begin{array}{l} 8 \rightarrow 1000 \\ 7 \rightarrow 0111 \\ 3 \rightarrow 0011 \end{array}\right\} 873_{10} \text{ bilangan BCD nya adalah } 1000\ 0111\ 0011$$

Pada penjumlahan bilangan BCD yang hasilnya lebih besar dari 9 ( 1001 ) maka harus ditambahkan 6 atau 0110.

Contoh : $65_{10} + 17_{10} =$

Langkah 1 : ubah masing-masing angka ke bentuk bilangan BCD

| | |
|---|---|
| $65_{10} \rightarrow 6 \rightarrow 0110$ | $17_{10} \rightarrow 1 \rightarrow 0001$ |
| $5 \rightarrow 0101$ | $7 \rightarrow 0111$ |
| $65_{10} \rightarrow 0110\ 0101$ | $17_{10} \rightarrow 0001\ 0111$ |

Langkah 2 : jumlahkan bilangan BCD tersebut. Apabila hasil penjumlahan masing-masing nibble melebihi 9 maka ditambahkan kembali dengan 6 atau 0110

$$
\begin{array}{r l}
0110\ 0101 & \\
\underline{0001\ 0111} & + \\
0111\ 1100 & \rightarrow \text{lebih dar 9 atau 1001} \\
\underline{0110} & + \quad \rightarrow \text{ditambah 6 atau 0110} \\
1000\ 0010 &
\end{array}
$$

Maka, $65_{10} + 17_{10} = 83_{10}$ atau dalam bentuk BCD nya adalah 0110 0101 + 0001 0111 = 1000 0010

Latihan 4

1. Ubahlah bilangan desimal dibawah ini ke bentuk BCD sandi 8421
   a. 98710
   b. 651210
   c. 9810
   d. 44510
   e. 31910
2. Hitunglah bingan dibwah ini menggunakan bentuk BCD 8421
   a. 6510 +3410
   b. 54910 + 7610
   c. 6610 + 1310

## BAB 5. BILANGAN BINER BERTANDA

Di dalam matematika, bilangan negatif biasanya dinyatakan dengan cara menambahkan tanda minus (−) di depan bilangan tersebut. Namun di dalam komputer, bilangan hanya dapat dinyatakan sebagai kode biner 0 dan 1 tanpa ada simbol yang lainnya, sehingga diperlukan suatu cara untuk mengkodekan tanda minus.

Beberapa metode yang dapat digunakan untuk menyatakan bilangan bertanda di sistem bilangan biner adalah sign-and-magnitude, komplemen satu dan komplemen dua. Komputer modern pada umumnya menggunakan metode komplemen dua, namun metode lain juga digunakan pada situasi tertentu. Sign-and-magnitude adalah cara yang banyak dipakai untuk merepresentasikan significand di dalam bilangan floating point.

Dalam metode sign-and-magnitude untuk menyatakan tanda bilangan (positif atau negatif), dapat digunakan salah satu bit yang ada untuk menyatakan tanda tersebut. Bit tersebut (biasanya bit yang pertama atau most significant bit) diset bernilai 0 untuk bilangan positif, dan 1 untuk bilangan negatif.

Bit-bit yang lain menyatakan magnitude atau nilai mutlak dari bilangan. Jadi di dalam satu byte (8-bit), satu bit digunakan sebagai tanda, dan 7 bit sisanya sebagai magnitude yang nilainya bisa berisi mulai dari 0000000 (0) sampai 1111111 (127). Cara ini dapat digunakan untuk merepresentasikan bilangan dari $-127_{10}$ sampai $+127_{10}$.

Contoh 1:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| $-45_{10} \rightarrow$ | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

Tanda 32 + 8 + 4 + 1 = 45

Angka 1 paling depan menyatakan tanda yaitu negatif (-) dan 7 digit dibelakangnya menunjukan bilangannya yaitu 45 maka, $-45_{10} \rightarrow 10101101_{2}$.

Contoh 2:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 01110011→ | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

Tanda 64 + 32 + 16 + 2 + 1 = 115

Angka 0 paling depan menyatakan tanda yaitu positif sedangakan 7 digit dibelakangnya menyatakan bilangannya yaitu 115 maka, $01110011_2$ → $+115_{10}$

Latihan 5 :

1. Ubahlah bilangan dibawah ini kedalam bentuk bilangan biner bertanda 8 bit!
   a. $+35_{10}$
   b. $-121_{10}$
   c. $-74_{10}$
   d. $35_{10}$
   e. $109_{10}$
2. Perhatikan bilangan biner bertanda dibawah ini :
   a. $10011011_2$
   b. $11100101_2$
   c. $01110010_2$
   d. $01010101_2$
   e. $11001100_2$

   Ubahlah bilangan-bilangan tersebut kedalam bentuk desimal!

# BAB 6. BILANGAN KOMPLEMEN

Metode pengurangan binary biasa dilakukan oleh manusia, untuk komputer biasanya menggunakan metode komplemen (complement) yaitu komplemen baris min – 1 ( Radix minus one complement ) dan komplemen baris ( Radix ). Komplemen pada dasarnya merubah bentuk pengurangan menjadi bentuk pertambahan.

Dalam sistem biner disebut komplemen 1 dan komplemen 2. Dalam sistem oktal yaitu komplemen 7 dan komplemen 8. Dalam sistem desimal, ada 2 macam komplemen yaitu komplemen 9 dan komplemen 10. Sedangkan dalam sistem heksadesimal disebut komplemen 15 dan komplemen 16.

A. Komplemen 1 dan Komplemen 2

Komplemen 1 dan komplemen 2 merupakan salah satu bentuk metode untuk menyatakan suatu bilangan bertanda pada sistem bilangan biner. Pada dasarnya dalam sebuah sistem komputer hanya mengenal angka 0 dan 1. Oleh karena itu, untuk menyatakan tanda negatif komputer menggunakan angka 1 dan angka 0 untuk menyatakan tanda positif.

Komplemen 1 dari suatu bilangan biner dilakukan dengan cara mengurangkan semua digit dengan nilai 1 bit / merubah bit ‘0’ menjadi ‘1’ atau bit ‘1’ menjadi ‘0’.

Contoh : hitunglah komplemen 1 dari 10111

Komplemen 1 : 11111-10111
= 01000

Komplemen 2 dari suatu bilangan biner dilakukan dengan cara, hasil komplemen 1 ditambah 1.

Contoh : hitunglah komplemen 2 dari 10111

Komplemen 1 : 01000

Komplemen 2 : 01000

$$\begin{array}{r} \phantom{01000}1\ + \\ \hline 01001 \end{array}$$

maka, komplemen 2 dari 10111 adalah 01001

Untuk menentukan tanda positif atau negatif suatu bilangan biner yaitu dengan cara mengurangi suatu bilangan biner dengan menggunakan komplemen-2 yaitu pengurangnya diubah dahulu ke bentuk komplemen-2 kemudian dijumlahkan dengan bilangan yang dikurangi. Jika ada pindahan (carry) pada bit MSB-nya, maka pindahan tersebut dibaikan dan hasilnya berupa bilangan positif.

Contoh 1: 7 - 5 = …

Langkah 1 : ubah angka 7 dan angka 5 kedalam bentuk biner

7 → 111 dan 5 → 101

Langkah 2 : angka 5 sebagai pengurangnya diubah kebentuk komplemen 2

5 → 101 → Komplemen 1 : 010

Komplemen 2 : 010

$$\begin{array}{r} 1\ + \\ \hline 011 \end{array}$$

Langkah 3 : biner dari 7 dan komplemen 2 dari 5 dijumlahkan, jika terdapat carry atau simpanan maka carry tersebut diabaikan dan hasilnya berupa bilangan positif

$$\begin{array}{r} 111 \\ \underline{011} + \\ 1010 \end{array}$$

Maka, hasilnya adalah 010 = +2

Angka 1 merupakan carry jadi diabaikan

Contoh 2: 5 - 7 = …

Langkah 1 : ubah angka 7 dan angka 5 kedalam bentuk biner

7 → 111 dan 5 → 101

Langkah 2 : angka 7 sebagai pengurangnya diubah kebentuk komplemen 2

7 → 111 → Komplemen 1 : 000

Komplemen 2 : 000

$$\begin{array}{r} \underline{1\ +} \\ 001 \end{array}$$

Langkah 3 : biner dari 5 dan komplemen 2 dari 7 dijumlahkan, jika tidak terdapat carry atau simpanan hasilnya berupa bilangan negatif

$$\begin{array}{r} 101 \\ \underline{001} + \\ 010 \end{array}$$

Maka, hasilnya adalah 010 = -2

Tidak terdapat carry didepan angka 0

B. Komplemen 7 dan Komlemen 8

Komplemen 7 dari suatu bilangan oktal dilakukan dengan cara, mengurangkan angka 7 untuk masing-masing digit dalam bilangan pengurangan.

Contoh : hitunglah komplemen 7 dari 456

Komplemen 7 : 777 – 456 = 321

Komplemen 8 dari suatu bilangan dilakukan dengan cara, hasil komplemen 7 ditambah 1 (cari komplemen 7 dulu lalu ditambah 1).

Contoh : hitunglah komplemen 7 dari 456

Komplemen 7 : 321

$$
\begin{array}{lr}
\text{Komplemen 8 :} & 321 \\
 & 1\ + \\
\hline
 & 322
\end{array}
$$

Maka komplemen 8 dari 456 adalah 322

C. Komplemen 9 dan Komplemen 10

Komplemen 9 dari suatu bilangan desimal delakukan dengan cara mengurangkan angka 9 untuk masing-masing digit dalam bilangan pengurang.

Contoh : hitunglah komplemen 9 dari 678

Komplemen 9 : 999 – 678 = 321

Komplemen 10 dari suatu bilangan dilakukan dengan cara, hasil komplemen 9 ditambah 1 ( cari komplemen 9 lalu ditambah 1 ).

Contoh : hitunglah komplemen 10 dari 678

Komplemen 9 : 321

Komplemen 10 : 321

$$\begin{array}{r} 321 \\ 1\ + \\ \hline 322 \end{array}$$

Maka, komplemen 10 dari 678 adalah 322

D. Komplemen 15 dan Komplemen 16

Komplemen 15 dari suatu bilangan hexadesimal dilakukan dengan cara, mengurangkan angka 15 untuk masing-masing digit dalam bilangan pengurangan. Dalam bilangan heksadesimal angka 15 = F

Contoh : hitunglah komplemen 15 dari CDE

Komplemen 15 : FFF – CDE = 321

Komplemen 16 dari suatu bilangan dilakukan dengan cara, hasil komplemen 15 ditambah 1 (cari komplemen 15 dulu lalu ditambah 1).

Contoh : hitunglah komplemen 10 dari 678

Komplemen 9 : 321

Komplemen 10 : 321

$$\begin{array}{r} 321 \\ 1\ + \\ \hline 322 \end{array}$$

Maka, komplemen 10 dari 678 adalah 322

LATIHAN 6

1. Tentukan komplemen 1 dan komplemen 2 dari bilangan dibawah ini :
   a. $10111_2$
   b. $34_{10}$
   c. $45_8$
   d. $A3_{16}$
2. Tentukan komplemen 7 dan komplemen 8 dari bilangan dibawah ini :
   a. $1101_2$
   b. $556_{10}$
   c. $657_8$
   d. $987_{16}$
3. Tentukan komplemen 9 dan komplemen 10 dari bilangan dibawah ini :
   a. $89A_{16}$
   b. $337_8$
   c. $11011_2$
   d. $654_{10}$
4. Tentukan komplemen 15 dan komplemen 16 dari bilangan dibawah ini :
   a. $677_8$
   b. $101100_2$
   c. $698_{10}$
   d. $AB7_{16}$
5. Selesaikan perhitungan dibawah ini menggunakan cara komplemen 1 dan komplemen 2 ;
   a. $45_{10} - 76_{10}$
   b. $10111_2 - 1111_2$
   c. $67_8 - 17_8$
   d. $3A_{16} - B1_{16}$

## BAB 7. GERBANG LOGIKA

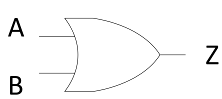
### Aljabar Boole

Diperkenalkan oleh George Boole pada tahun 1854 dipergunakan dalam logika matematika, peluang/kemungkinan, teori komunikasi/informasi, teori himpunan dan lain sebagainya. Dalam ilmu komputer dipergunakan sebagai switching circuits yang dimaksudkan untuk melambang simbol mengalir atau tidaknya arus listrik dengan logika 1 untuk keadaan tertutup atau tersambung dan 0 untuk keadaan terbuka atau mati.

Gerbang dasar aljabar boole terdiri dari:

**1. Gerbang OR ( + )**

Rangkaian logika yang memiliki satu output dan dua atau lebih input dilambangkan dengan



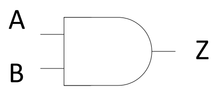

Bentuk persamaan Boole : A + B = Z

Pada gerbang OR output akan memiliki muatan atau bernilai 1 jika salah satu atau kedua inputnya memiliki muatan atau bernilai 1. Dijelaskan dengan logika pada tabel kebenaran sebagai berikut:

| **INPUT** | | **OUTPUT** |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## 2. Gerbang AND ( . )

Rangkaian logika yang memiliki satu output dan dua atau lebih input dilambangkan dengan



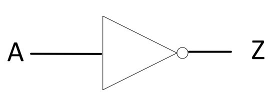

Bentuk persamaan Boole : A . B = Z

Pada gerbang AND output akan memiliki muatan atau bernilai 1 jika kedua inputnya mengandung muatan atau bernilai 1. Dijelaskan dengan logika pada tabel kebenaran sebagai berikut:

| **INPUT** | | **OUTPUT** |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## 3. Gerbang NOT

Rangkaian logika yang memiliki satu output dan satu input yang disebut juga sebagai inverter dilambangkan dengan :



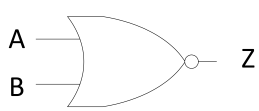

Bentuk persamaan Boole : Z = $\bar{A}$

Pada gerbang NOT output akan memiliki nilai kebalikan dari inputnya. Dijelaskan dengan logika pada tabel kebenaran sebagai berikut:

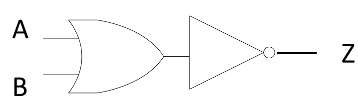
| INPUT | OUTPUT |
|---|---|
| **A** | **Z** |
| 0 | 1 |
| 1 | 0 |

**4. Gerbang NOR**

Rangkaian logika yang memiliki satu output dan dua atau lebih input dan merupakan gabungan dari gerbang NOT OR yang berarti kebalikan dari nilai yang dimiliki gerbang OR. Dilambangkan dengan:




atau

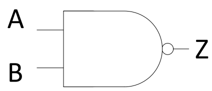



Bentuk persamaan Boole : Z = $\overline{A + B}$

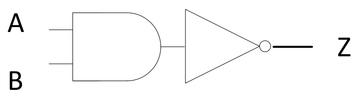
| INPUT | | OUTPUT |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

## 5. Gerbang NAND

Rangkaian logika yang memiliki satu output dan dua atau lebih input dan merupakan gabungan dari gerbang NOT AND yang berarti kebalikan dari nilai yang dimiliki gerbang AND. Dilambangkan dengan:




atau



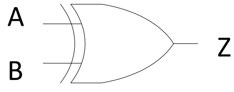

Bentuk persamaan Boole : Z = $\overline{A.B}$

Pada gerbang NAND output tidak akan memiliki muatan atau bernilai 0 jika kedua inputnya mengandung muatan atau bernilai 1. Dijelaskan dengan logika pada tabel kebenaran sebagai berikut:

| **INPUT** | | **OUTPUT** |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## 6. Gerbang EX-OR

Ekslusif OR atau XOR disimbolkan dengan ⊕ merupakan rangkaian logika yang memiliki satu output dan dua atau lebih input. Dilambangkan dengan:




Bentuk persamaan Boole Z = A ⊕ B

Pada gerbang XOR output akan memiliki muatan atau bernilai 1 jika salah satu input memiliki muatan atau bernilai 1. Dijelaskan dengan logika pada tabel kebenaran sebagai berikut:

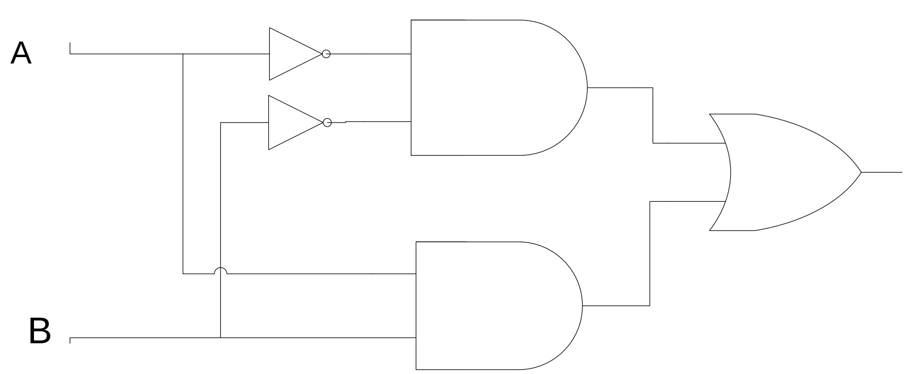
| **INPUT** | | **OUTPUT** |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Contoh 1.

Gambarkan gerbang logika dari fungsi berikut $\overline{AB}$ + AB sebagai sinyal masukan !

Jawab:

Dari nilai masukan yang diberikan kita memiliki dua suku $\overline{AB}$ dan AB dengan operasi AND dan dilakukan operasi OR terdahap kedua suku tersebut maka kita akan membutuhkan dua gerbang logika AND dan satu gerbang logika OR digambarkan sebagai berikut



# BAB 8. PENYEDERHANAAN FUNGSI BOOLE

## A. Bentuk Kanonik

Ekspresi Boolean yang mengespesifikasikan suatu fungsi dapat disajikan kedalam dua bentuk kanonik berbeda yaitu:

1. Penjumalahan dari hasil kali (sum of product atau SOP)

Memiliki bentuk kanonik $f(A,B,C\ ) = \bar{A}\ \bar{B}$ C + A $\bar{B}\ \bar{C}$ + ABC dimana setiap suku didalam ekspresi mengandung literal yang lengkap baik peubah yang tulis dalam komplemen maupun tidak. Pada SOP peubah tanpa komplemen memiliki nilai 1 dan peubah dengan komplemen memiliki nilai 0.

2. Perkalian dari hasil jumlah (product of sum atau POS)

Memiliki bentuk kanonik $f(A,B,C)$ = (A + B + C) (A + $\bar{B}$ + C) ($\bar{A}$ + $\bar{B}$ + C) dimana setiap suku didalam ekspresi mengandung literal yang lengkap baik peubah yang tulis dalam komplemen maupun tidak. Pada POS peubah tanpa komplemen memiliki nilai 0 dan peubah dengan komplemen memiliki nilai 1.

Lebih jelas dapat dilihat pada tabel berikut

| **ABCD** | Minterm | | Maksterm | |
|---|---|---|---|---|
| | Term | Lambang | Term | Lambang |
| 0000 | $\overline{ABCD}$ | $m_0$ | A + B + C + D | $M_0$ |
| 0001 | $\overline{ABC}D$ | $m_1$ | A + B + C + $\bar{D}$ | $M_1$ |
| 0010 | $\overline{AB}C\bar{D}$ | $m_2$ | A +B + $\bar{C}$ + D | $M_2$ |
| 0011 | $\overline{AB}CD$ | $m_3$ | A + B + $\bar{C}$ + $\bar{D}$ | $M_3$ |
| 0100 | $\bar{A}B\overline{CD}$ | $m_4$ | A + $\bar{B}$ + C + D | $M_4$ |
| 0101 | $\bar{A}B\bar{C}D$ | $m_5$ | A + $\bar{B}$ + C + $\bar{D}$ | $M_5$ |
| 0110 | $\bar{A}BC\bar{D}$ | m6 | A + $\bar{B}$ + $\bar{C}$ + D | $M_6$ |
| 0111 | $\bar{A}BCD$ | $m_7$ | A + $\bar{B}$ + $\bar{C}$ + $\bar{D}$ | $M_7$ |
| 1000 | $A\overline{BCD}$ | $m_8$ | $\bar{A}$ + B + C + D | $M_8$ |

| | | | | |
|---|---|---|---|---|
| 1001 | $A\overline{B}\overline{C}D$ | $m_9$ | $\overline{A} + B + C + \overline{D}$ | $M_9$ |
| 1010 | $A\overline{B}C\overline{D}$ | $m_{10}$ | $\overline{A} + B + \overline{C} + D$ | $M_{10}$ |
| 1011 | $A\overline{B}CD$ | $m_{11}$ | $\overline{A} + B + \overline{C} + \overline{D}$ | $M_{11}$ |
| 1100 | $AB\overline{C}\overline{D}$ | $m_{12}$ | $\overline{A} + \overline{B} + C + D$ | $M_{12}$ |
| 1101 | $AB\overline{C}D$ | $m_{13}$ | $\overline{A} + \overline{B} + C + \overline{D}$ | $M_{13}$ |
| 1110 | $ABC\overline{D}$ | $m_{14}$ | $\overline{A} + \overline{B} + \overline{C} + D$ | $M_{14}$ |
| 1111 | $ABCD$ | $m_{15}$ | $\overline{A} + \overline{B} + \overline{C} + \overline{D}$ | $M_{15}$ |

Contoh 1.

Tuliskan bentuk kanonik SOP dan POS, tabel kebenaran dan gerbang logika dari hasil SOP dari persamaan *f(A,B,C)* = $A + \overline{B}C$

Jawab:

**SOP**

Lengkapi terkebih dahulu literal untuk setiap suku agar jumlahnya sama.

| | | |
|---|---|---|
| A | $= A(B + \overline{B})$ | Dikalikan $(B + \overline{B})$ karena suku 1 kekurangan varibel B |
| | $= AB + A\overline{B}$ | |
| | $= AB(C + \overline{C}) + A\overline{B}(C + \overline{C})$ | Dikalikan $(C + \overline{C})$ karena suku 1 kekurangan varibel C |
| | $= ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C}$ | |
| $\overline{B}C$ | $= \overline{B}C(A + \overline{A})$ | Dikalikan $(A + \overline{A})$ karena suku 2 kekurangan varibel A |
| | $= A\overline{B}C + \overline{A}\,\overline{B}C$ | |

Jadi *f(A,B,C)* = $A + \overline{B}C$

$= ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C + \overline{A}\,\overline{B}C$

$= \overline{A}\,\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$ suku yang bernilai sama dihilangkan

$= m_1 + m_4 + m_5 + m_6 + m_7 = \sum(1,4,5,6,7)$

**POS**

Perhatikan berdasarkan contoh soal persamaan fungsi diberikan dalam bentuk SOP untuk itu harus dilkukan perubahan kedalam bentuk POS !

*f(A.B.C)* = A + $\bar{B}$C ubah terlebih dahulu kedalam format POS

= (A + $\bar{B}$) (A + C) Hukum Distributif

Lengkapi literal untuk setiap suku agar jumlahnya sama

| | | |
|---|---|---|
| (A + $\bar{B}$) | = (A + $\bar{B}$) + (C$\bar{C}$) | Dijumlahkan dengan (C$\bar{C}$ ) karena suku 1 kekurangan C |
| | = (A + $\bar{B}$ + C ) (A + $\bar{B}$ + $\bar{C}$) | |
| (A + C) | = (A + C) + B$\bar{B}$ | Dijumlahkan dengan (B$\bar{B}$) karena suku 2 kekurangan B |
| | = (A + B + C) (A + $\bar{B}$ + C) | |

Jadi *f*(A,B,C) = (A + $\bar{B}$) (A + C)

= (A + $\bar{B}$ + C ) (A + $\bar{B}$ + $\bar{C}$) (A + B + C) (A + $\bar{B}$ + C)

suku yang bernilai sama (A + $\bar{B}$ + C) hilangkan hingga menjadi

$M_0$,$M_2$,$M_3$ = π (0,2,3)

Jika diperhatikan bahwa apabila SOP menghasilkan ∑(1,4,5,6,7) makan POS merupakan sisanya π (0,2,3)

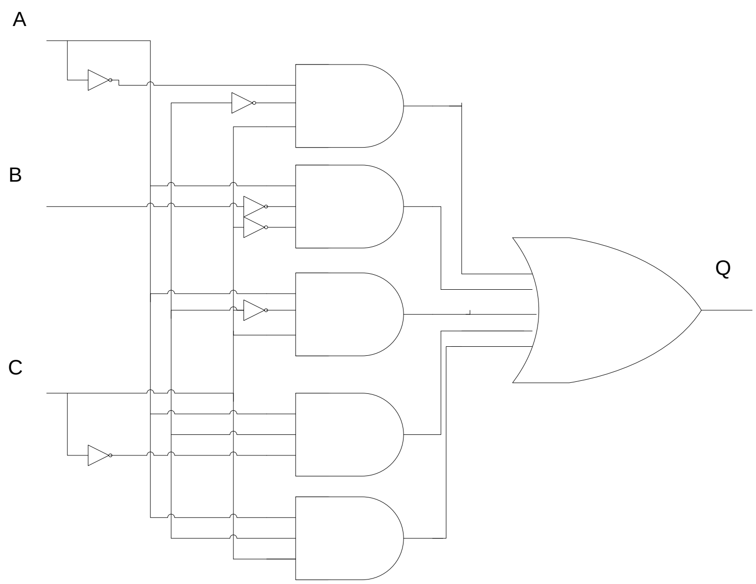
**Tabel Kebenaran**

| SOP | |
|---|---|
| ABC | *f* |
| 000 | 0 |
| 001 | 1 |
| 010 | 0 |
| 011 | 0 |
| 100 | 1 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

Kolom fungsi diberikan nilai 1 untuk semua kombinasi hasil SOP dan 0 untuk hasil POS
Kolom nilai variabel ABC diisikan dengan kombinasi kemungkinan munculnya variabel. Jumlah kemungkinan didapat dari $2^n$ dimana n merupakan variabel. $2^3$ = 8 baris kemungkinan.

**Gerbang Logika**

SOP = $\bar{A}\ \bar{B}C$ + A$\bar{B}\bar{C}$ + A$\bar{B}$C + AB$\bar{C}$ + ABC




## B. Penyederhaan Fungsi Bolean secara alajabar

Jumlah literal dalam sebuah fungsi aljabar Boolean dapat diminumkan dengan dua metode yaitu:

- Penyederhanaan fungsi Boolean secara aljabar
- Penyederhaan Fungsi Bolean Dengan Peta Karnaugh (K-Map)

### 1. Penyederhanaan fungsi Boolean secara hukum aljabar

**Hukum-Hukum dan Teori Aljabar Boole**

| | | |
|---|---|---|
| Hukum Identitas: | A + 0 = A | A x 1 = A |
| Hukum Idempoten: | A + A = A | A x A = A |
| Hukum Komplemen | A + $\bar{A}$ = 1 | A x $\bar{A}$ = 0 |
| Hukum Dominansi | A + 1 = 1 | A x 0 = 0 |
| Hukum Involusi: | $\bar{\bar{A}}$ = A | |

| | | |
|---|---|---|
| Hukum Komutatif: | A + B + C = C + B + A | A x B x C = C x B x A |
| Hukum Assosiatif: | (A + B) + C | A + (B + C) |
| | (A x B) x C | A x (B x C) |
| Hukum Distributif: | A(B + C) = AB + AC | A + (B x C) = (A+B) (A+C) |

## 2. Penyederhaan Fungsi Bolean Dengan Peta Karnaugh (K-Map)

Ditemukan oleh Maurice Kaurnaugh tahun 1953 dengan metode grafis. Yaitu dengan mengelompokan pasangan angka 1 yang saling berdekatan. Dua kotak berdekatan (pair), empat kotak (Quad) atau empat kotak berdekatan (oktet). Untuk dapat melakukan minimisasi peta Karnaugh sebaiknya pemahaman mengenai pengisian nilai-nilai SOP dan POS pada Karnaugh dipahami terlebih dahulu.

Apabila pengelompokan dimungkinkan untuk keadaan 2 dan 4 berdampingan

**Metode Penempatan SOP dan POS pada peta Karnaugh**

**Peta Karnaugh untuk 2 variabel** (Jumlah kotak $2^n$, n = variabel maka $2^2$ = 4)

| | | 0 | 1 | B |
|---|---|---|---|---|
| | 0 | $\bar{A}\bar{B}$ | $\bar{A}$B | |
| A | 1 | A$\bar{B}$ | AB | |

Contoh : 2

$f$(A,B,C) = A$\bar{B}$ + AB

A$\bar{B}$ dalam biner 10 penempatan pada karnaugh baris A = 1 , kolom B = 0

AB dalam biner 11 penempatan pada karnaugh baris A = 1 , kolom B = 1

Menjadi sebagai berikut:

| | | 0 | 1 | B |
|---|---|---|---|---|
| | 0 | 0 | 0 | |
| A | 1 | 1 | 1 | |

Baris 2 kolom 1 dan baris 2 kolom 2 diisikan dengan 1 karena fungsi merupakan bentuk SOP. Untuk mendapatkan fungsi POS didapat dari nilai 0 yaitu : $\overline{AB}$ + $\bar{A}$B

POS $= \overline{AB} + \bar{A}B$

$= \overline{\overline{AB} + \bar{A}B}$

$= (A + B) \cdot (A + \bar{B})$

**Peta Karnaugh untuk 3 variabel** (Jumlah kotak $2^n$ , n = variabel maka $2^3 = 8$)

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $\overline{ABC}$ | $\overline{AB}C$ | $\bar{A}BC$ | $\bar{A}B\bar{C}$ |
| 1 | $A\overline{BC}$ | $A\bar{B}C$ | ABC | $AB\bar{C}$ |

Contoh : 3

$f$ (A,B,C) = $\bar{A}B\bar{C}$ + $AB\bar{C}$ + ABC

$\bar{A}B\bar{C}$ dalam bentuk biner 010 penempatan pada A= 0 , BC = 10

$AB\bar{C}$ dalam bentuk biner 110 penempatan pada A = 1,  BC = 10

ABC dalam bentuk biner 111 penempatan pada A = 1 , BC = 11

Pengisian pada peta Kaurnaugh menjadi sebagai berikut:

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |

Baris 1 kolom 1, baris 2 kolom 3 dan baris 2 kolom 4 diisikan dengan 1 karena fungsi merupakan bentuk SOP. Untuk mendapatkan fungsi POS didapat dari nilai 0 pada Karnaugh yaitu : $\overline{ABC}$ + $\overline{AB}$C + $\bar{A}$BC + A$\overline{BC}$ + A$\bar{B}$C

POS = $\overline{ABC}$ + $\overline{AB}$C + $\bar{A}$BC + A$\overline{BC}$ + A$\bar{B}$C

= $\overline{\overline{ABC} + \overline{AB}C + \bar{A}BC + A\overline{BC} + A\bar{B}C}$

= (A+B+C) (A+B+$\bar{C}$ ) + ($\bar{A}$ + B + C) ($\bar{A}$ + B + $\bar{C}$

**Peta Karnaugh untuk 4 variabel** (Jumlah kotak $2^n$ , n = variabel maka $2^4$ = 16)

| | | CD | | | |
|---|---|---|---|---|---|
| | | 00 | 01 | 11 | 11 |
| AB | 00 | $\overline{ABCD}$ | $\overline{ABC}$D | $\overline{AB}$CD | $\overline{AB}$C$\bar{D}$ |
| | 01 | $\bar{A}$B$\overline{CD}$ | $\bar{A}$B$\bar{C}$D | $\bar{A}$BCD | $\bar{A}$BC$\bar{D}$ |
| | 11 | AB$\overline{CD}$ | AB$\bar{C}$D | ABCD | ABC$\bar{D}$ |
| | 10 | A$\overline{BCD}$ | A$\overline{BC}$D | A$\bar{B}$CD | A$\bar{B}$C$\bar{D}$ |

Pengisian pada peta Karnaugh untuk 4 variabel mengikuti aturan yang berlaku pada variabel sebelumnya.

**Penggulungan**

Teknik ini digunakan melakukan pengelompokan ketika ditemukan letak posisi angka 1 berada pada sisi yang berseberangan (sisi kiri dengan kanan dan sisi atas dengan bawah). Kotak yangberada pada sisi berseberangan dianggap bertetangga juga dengan cara menautkan atau melakukan penggulungan.

Semakin banyak terbentuknya kelompok angka satu maka semakin banyak suku (term) yang terbentuk. Semakin banyak suku maka persamaan semakin tidak sederhana.

Bila ditemukan sebuah kelompok dapat terbentuk dengan 2 atau 4 pasangan dalam satu peta maka pilihlah pasangan dengan jumlah terbesar yaitu 4.

**Minimisasi fungsi Boolean dengan peta Karnaugh**

Minimisasi dilakukan dengan cara melingkari kelompok angka 1 berdekatan yang berpasangan 2, 4 atau 8 pasang. Kemudian lakukan peninjauan terhadap baris dan kolom pada masing-masing pasangan tersebut untuk mengambil posisi nilai yang sama atau menghilangkan nilai yang tidak sama.

Contoh 4.

Sederhanakan fungsi berikut ini *f(A,B,C)* = $\overline{ABCD}$ + $\bar{A}$ B$\overline{CD}$ + $ABC\bar{D}$ + A$\bar{B}$C$\bar{D}$ dengan metode :

a. K-Map
b. Hukum Aljabar

Jawab:

a. K-Map

Perhatikan nilai kesamaan terhadap AB dan CD untuk masing-masing kelompok

*f(A,B,C)* = $\overline{ABCD}$ + $\bar{A}$ B$\overline{CD}$ + $ABC\bar{D}$ + A$\bar{B}$C$\bar{D}$ = $\overline{ACD}$ + AC$\bar{D}$

| AB \ CD | **00** | 01 | 11 | **10** |
|---|---|---|---|---|
| **0**0 | 1 | 0 | 0 | 0 |
| **0**1 | 1 | 0 | 0 | 0 |
| **1**1 | 0 | 0 | 0 | 1 |
| **1**0 | 0 | 0 | 0 | 1 |

Ling I = $\overline{ACD}$

Ling II = AC$\bar{D}$

Lingkaran I = ditarik terhadap CD kelompok angka sama pada posisi $\overline{CD}$

= ditarik terhadap AB kelompok angka sama pada posisi $\bar{A}$

Maka lingkaran I bernilai $\overline{ACD}$

Lingkaran II = ditarik terhadap CD kelompok angka sama pada posisi C$\bar{D}$

= ditarik terhadap AB kelompok angka sama pada posisi A

Maka lingkaran I bernilai AC$\bar{D}$

b. Dengan Hukum Aljabar

*f(A,B,C)* = $\overline{ABCD}$ + $\bar{A}$ B$\overline{CD}$ + $ABC\bar{D}$ + A$\bar{B}$C$\bar{D}$

= $\overline{ACD}$ (B+$\bar{B}$) + AC$\bar{D}$ (B+ $B$) Hukum distributif
= $\overline{ACD}$ . 1 + AC$\bar{D}$ . 1 Hukum komplemen
= $\overline{ACD}$ + AC$\bar{D}$

*Contoh* 5.

Sederhanakan fungsi berikut *f* (A,B,C) = A$\bar{C}$ + $\bar{B}$C + AB$\bar{C}$ dengan metode:

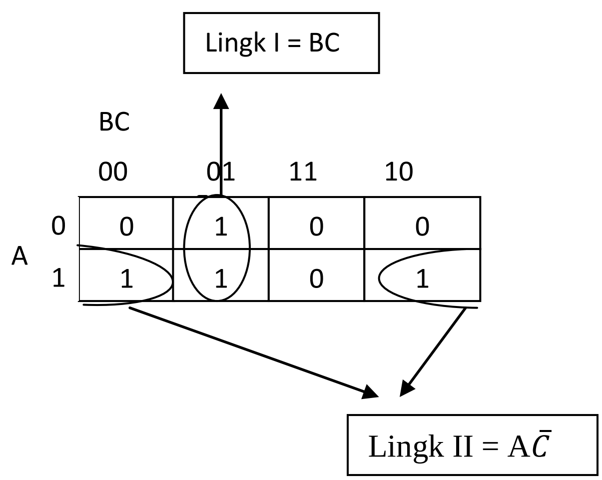
a. K-Map
b. Hukum Aljabar

Jawab:

a. K-Map

Perhatikan nilai kesamaan terhadap A dan BC untuk masing-masing kelompok

*f* (A,B,C) = A$\bar{C}$ + $\bar{B}$C + AB$\bar{C}$ = A$\bar{C}$ + $\bar{B}$C




Lingkaran I = ditarik terhadap BC kelompok angka sama pada posisi $\bar{B}$C

= ditarik terhadap A kelompok angka tidak memiliki kesamaan

Maka lingkaran I bernilai $\bar{B}$C

Lingkaran II = ditarik terhadap BC kelompok angka sama pada posisi $\bar{C}$

= ditarik terhadap A kelompok angka sama pada posisi A

Maka lingkaran II bernilai A$\bar{C}$

b. Hukum Aljabar

$f$ (A,B,C) = A$\bar{C}$ + $\bar{B}$C + AB$\bar{C}$

= A$\bar{C}$ . 1 + $\bar{B}$C + AB$\bar{C}$ Hukum identitas

= A$\bar{C}$ (1 + B) + $\bar{B}$C Hukum distributif

= A$\bar{C}$ . 1 + $\bar{B}$C Hukum dominansi

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

**Keadaan Don't Care**

Keadaan don't care berlaku pada kombinasi BCD dimana kombinasi variabel hanya memenuhi 0000 sampai 1001 sementara untuk 1010 sampai 1111 tidak mungkin terjadi pada operasi normalnya.

Representasi angka-angka 1010 sampai dengan 1111 pada K-Map dilakukan dengan memberikan tanda silang yang diartikan bahwa dapat dipergunakan untuk nilai 0 atau 1.

Contoh 6.

Sederhanakan persamaan Boole $f$(A,B,C,D) = ∑m(2,3,7,8,9) dengan keadaan don't care $f$(A,B,C,D) = ∑m(10,11,12,13,14,15)

| | | CD | | | | |
|---|---|---|---|---|---|---|
| | | 00 | 01 | 11 | 11 | |
| AB | 00 | 0 | 0 | x | 1 | → $\bar{B}$CD |
| | 01 | 0 | 0 | x | 0 | |
| | 11 | 1 | 1 | x | x | |
| | 10 | 1 | 1 | x | x | → A |

Bentuk Sum Of Product dari peta tersebut adalah A + $\bar{B}$CD

| AB \ CD | 00 | 01 | 11 | 11 |
|---|---|---|---|---|
| 00 | 0 | 0 | x | 1 |
| 01 | 0 | 0 | x | 0 |
| 11 | 1 | 1 | x | x |
| 10 | 1 | 1 | x | x |

$AC$

BCD

Bentuk Product Of Sum $= \overline{AC} + \text{BCD}$

$= \overline{\overline{AC} + \text{BCD}}$

$= (\text{A} + \text{C})\ (\bar{B} + \bar{C} + \bar{D})$

**Latihan !**

1. Tuliskan bentuk SOP dan POS dari fungsi berikut:
   a. Π(0,2,4,5)
   b. $f$(A,B,C) = $\bar{C}$ + AB + $\bar{A}$B$\bar{C}$
2. Nyatakan persamaan fungsi-fungsi boole dibawah ini kedalam rangkaian gerbang logika:
   a. AB + BC + $\bar{C}$D
   b. $\overline{AB}$ . (A+C)
   c. AB + $\bar{A}$ B
3. Sederhanakan persamaan Boole berikut ini dengan menggunakan Hukum aljabar Boolean
   a. AB + $\bar{A}$C + BC
   b. AB$\overline{CD}$ + AB$\bar{C}$D + ABC$\bar{D}$ + AB$\bar{C}$D + A$\overline{BCD}$ + A$\overline{BC}$D + A$\bar{B}$CD + A$\bar{B}$C$\bar{D}$
4. Sederhanakan persamaan Boole berikut ini dengan menggunakan peta Karnaugh
   a. $f$(A,B,C) = ABC + AB$\bar{C}$ + A$\bar{B}$C + $\bar{A}$B$\bar{C}$
   b. $f$(A,B,C,D) = $\bar{A}$B + BD + A$\bar{B}$C + BC$\bar{D}$
5. Sederhanakan maksterm dan minterm berikut dengan menggunakan peta Karnaugh
   a. π M = (1,3,5,7,9,11,13,15)
   b. ∑m = (0,4,6,8,9,10,11,15)

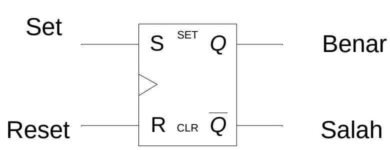
# BAB 9. FLIP-FLOP

Flip-flop merupakan rangkaian yang dapat memiliki output dengan dua keadaan berlainan yang stabil pada saat yang sama. Rangkaian ini umum digunakan pada elemen memori, counter, register dan sebagainya. Flip-flop dikelompokan atas beberapa jenis RS, JK, D dan T.

## A. SR Flip-flop

Merupakan dasar dari flip-flop jenis lain dengan dua output yang saling berlawanan yaitu Q dan $\bar{Q}$ dan dua buah input yaitu R (reset) dan S (set).

Digunakan pada rangkaian digital komputer dengan menggunakan sinyal logik 1 atau 0. Flip-flop ini mempunyai dua masukan yaitu:




Simbol SR flip-flop

Cara kerja:

Apabila muatan keluaran Q sekarang berada pada keadaan 0 untuk membuatnya menjadi satu maka harus diberikan pada set (Set). Dan untuk mengembalikan nilai Q kembali menjadi 0 maka dilakukan trigger pada reset (R)

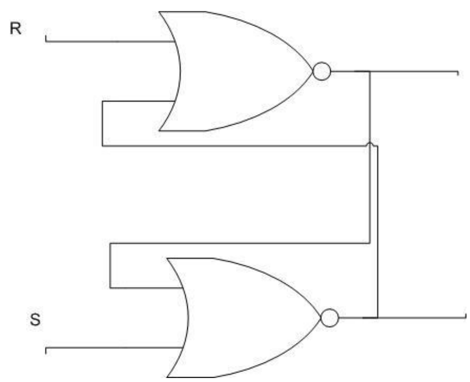
Apabila output Q = 0 maka untuk menjadinya menjadi 1 harus diberikan trigger pada S

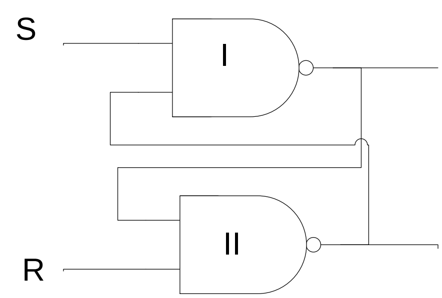
Tabel kebenaran SR flip-flop dapat dilihat sebagai berikut:

| INPUT | OUTPUT | |
|---|---|---|
| S | R | Q |
| 0 | 0 | Tidak berubah (keadaan terakhir) |
| 0 | 1 | 0 (reset) |
| 1 | 0 | 1 (set) |
| 1 | 1 | Tidak diperkenankan |

Untuk keadaan input SR = 11 maka keadaan output tidak diperbolehkan (dihindarkan) karena kedua output yaitu Q dan $\bar{Q}$ pada keadaan sama hal ini tidak sesuai dengan fungsi SR fli-flop sebagai mana mestinya.

SR flip-flop dapat dibangun dengan dua buah gerbang NOR atau NAND yang mengandung dua input dan dua output




Output dari kedua gerbang diatas disesuaikan dengan tabel kebenaran yang dimiliki oleh masing-masing gerbang yang telah dibahas pada bab sebelumnya.

Tabel masukan :

| S | R | Q | Q+ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | Terlarang |
| 1 | 1 | 1 | Terlarang |

Peta Karnaugh

| Q \ SR | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | x | 1 |
| 1 | 1 | | x | 1 |

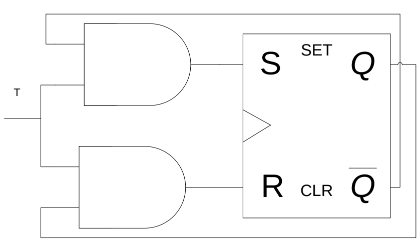
Persamaan pada SR flip-flop $Q^+ = S + RQ$ dimana $SR = 0$

Flip-flop akan berubah pada output jika ada perubahan pada input dengan menggunakan clock gerbang logika sebagai sinyal penabuh untuk menyerempakan flip-flop.

## B. T Flip-flop

Memiliki sifat yang selalu berubah keadaanya setiap masukan mendapat sinyal pemicu (trigger) dengan sifat ini flip-flop T sering disebutsebagaifli-flop Togle. Memilki satu bagian masukan dengan dua keluaran.

Dapat disusun dari satu flip-flop RS dan dua gerbang AND




Tabel kebenaran

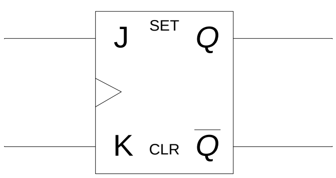
| **T** | **Q** | **Q+** |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

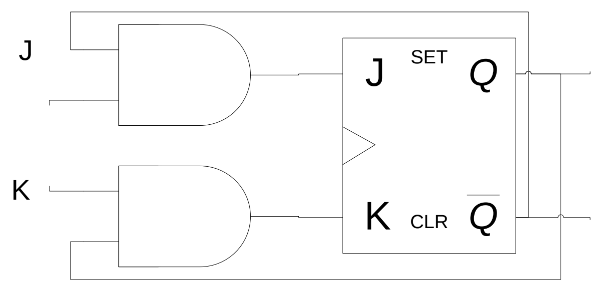
Perubahan pulsa dari 0 ke 1 disebut sebagai naik atau pulsa positif dan perubahan dari 1 menuju 0 disebut sebagai pulsa turun atau pulsa negatif.

Persamaan pada T flip-flop Q+ = $\bar{T}$Q + T$\bar{Q}$

**C. JK Flip-flop**

Digunakan untuk memperbaiki keadaan yang tidak diperkenankan pada SR flip-flop yang tidak mengizinkan pemberian masukan S dan R dengan 1. Flip-flop ini mempunyai dua input J dan K berfungsi sama denga input pada SR di filp-flop SR yang membedakan bahwa J dan K jika memiliki input = 1 maka akan membuat JK flip-flop berfungsi sebagai flip-flop T




Simbol JK flip-flop

Tabel kebenaran:

| J | K | Q | Q+ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Peta Karnaugh

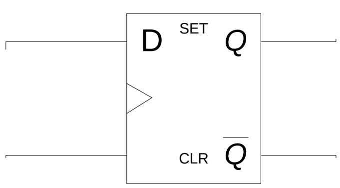
JK

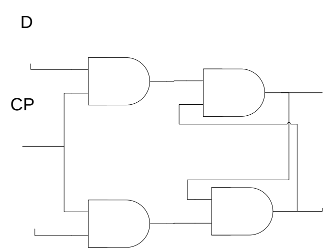
| | | 00 | 01 | 11 | 10 |
|---|---|---|---|---|---|
| Q | 0 | | | 1 | 1 |
| | 1 | 1 | | x | 1 |

Persamaan pada JK flip-flop: $Q^{+} = J\overline{Q} + Q\overline{K}$

**D. D Flip-flop**

Berasal dari kata delay yang mempunyai satu masukan dan banyak dipakai sebagai sel memori pada komputer dilengkapi dengan trigger pada masukan.




Tabel kebenaran

| D | Q | Q+ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# DAFTAR PUSTAKA

1. John G. Proakis/Dimitris, *Digital Signal processing, G. Manolakis second edition*, Prentice Hall 2001
2. Ganti Depari, Teori dan Aplikasi Teknik Digital, Nuansa Aulia, 2011
3. Pernantin Tarigan, Dasar Teknik Digital, Graha Ilmu 2012
4. Rinaldi Munir, Matematika Diskrit Edisi 3, Informatika, 2005