

## BAB II TINJAUAN PUSTAKA

### 2.1. Sistem Bilangan

Sistem bilangan (*Number System*) adalah suatu cara untuk mewakili besaran dari suatu item fisik. Sistem bilangan terdiri dari beberapa bilangan antara lain bilangan biner, bilangan oktal, bilangan desimal dan bilangan heksadesimal.

#### 2.1.1. Bilangan Biner

Bilangan biner hanya mempunyai dua simbol angka yang diberi tanda dengan simbol “0” dan “1”. Bilangan biner merupakan sistem bilangan yang mempunyai radix 2. Radix adalah jumlah angka atau simbol yang dipergunakan pada suatu sistem bilangan. Didalam sistem bilangan biner istilah yang disebut BIT ( Binary Digit). Binary digit adalah jumlah angka biner yang dipakai pada suatu bilangan biner.

Contoh: 101 mempunyai 3 BIT

1101 mempunyai 4 BIT

#### 1. Konversi Biner ke Desimal

**Tabel 2.1** konversi Biner ke Desimal

Pangkat	Desimal	Biner
$2^0$	1	1
$2^1$	2	10
$2^2$	4	100
$2^3$	8	1000
$2^4$	16	10000
$2^5$	32	100000
$2^6$	64	1000000
$2^7$	128	10000000
Dst		

(Tokhem, Roger L, 1990)

$$\begin{aligned}\text{Contoh: } 1101 &= (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 8 + 4 + 0 + 1 \\ &= 13_{(10)}\end{aligned}$$

## 2. Konversi Biner ke Oktal

Pengelompokan setiap tiga digit bilangan biner mulai dari LSB hingga MSB. Setiap kelompok akan menandakan nilai oktal dari bilangan tersebut.

Contoh:

Ubahlah bilangan biner 1101 ke bilangan oktal

$$\underline{001\ 101} = 15_{(8)}$$

1     5

## 3. Konversi Biner ke Heksadesimal

Pengelompokan setiap empat digit bilangan biner mulai dari LSB hingga MSB. Setiap kelompok akan menandakan nilai heksadesimal dari bilangan heksadesimal.

Contoh:

Ubahlah bilangan biner 1101 0010 1111 kedalam heksadesimal.

$$\underline{1101\ 0010\ 1111} = D2F_{(16)}$$

D    2    F

### 2.1.2. Bilangan Oktal

Sistem bilangan oktal menggunakan 8 macam simbol bilangan berbasis 8 digit angka, yaitu 0,1,2,3,4,5,6, dan 7. Bilangan oktal juga berhubungan dengan bilangan dasar 8.

**Tabel 2.2** ekuivalen bilangan biner dan oktal untuk bilangan desimal 0 sampai 17

No.	Desimal	Biner	Oktal
1.	0	000	0
2.	1	001	1
3.	2	010	2
4.	3	011	3
5.	4	100	4
6.	5	101	5
7.	6	110	6
8.	7	111	7
9.	8	001 000	10
10.	9	001 001	11
11.	10	001 010	12
12.	11	001 011	13
13.	12	001 100	14

14.	13	001 101	15
15.	14	001 110	16
16.	15	001 111	17
17.	16	010 000	20
18.	17	010 001	21

(Tokhem, Roger L, 1990)

### 1. Konversi Oktal ke Biner

Setiap pembagi digit bilangan oktal dapat direpresentasikan ke dalam tiga digit bilangan biner. Setiap digit bilangan oktal diubah secara terpisah.

Contoh: ubahlah bilangan  $3527_{(8)}$  kedalam bilangan biner

3    5    2    7  
011 101 010 111

Jadi  $3527_{(10)} = 11101010111_{(2)}$

### 2. Konversi Oktal ke Desimal

Ubahlah bilangan  $1161_{(8)} = 625_{(10)}$

$$\begin{aligned} 1161_{(8)} &= (1 \times 8^3) + (1 \times 8^2) + (6 \times 8^1) + (1 \times 1^0) \\ &= 512 + 64 + 48 + 1 \\ &= 625_{(10)} \end{aligned}$$

### 3. Konversi Oktal ke Heksadesimal

Teknik mengonversi bilangan oktal ke heksadesimal adalah dengan mengubah bilangan oktal menjadi biner kemudian mengubah binernya menjadi heksadesimal.

Contoh ubahlah bilangan  $725_{(8)} = \quad_{(16)}$

Oktal    7    2    5  
Biner    1 | 11 01 | 0 101  
Hexa    1 | D | 5

Jadi  $725_{(8)} = 1D5_{(16)}$

### 2.1.3. Bilangan Desimal

Bilangan desimal adalah angka yang terdiri dari 0,1,2,3,4,5,6,7,8,9. Dimana sistem bilangan desimal jumlah angka (simbol) yang digunakan adalah 10 (radix 10).

#### 1. Konversi Desimal ke Biner

Konversi bilangan desimal dilakukan dengan secara berulang-ulang suatu bilangan desimal dengan 2. Sisa setiap pembagian merupakan bit yang didapat.

Contoh: konversi  $625_{(10)}$  ke biner

$625 / 2 = 312$	sisa	1
$312 / 2 = 156$		0
$156 / 2 = 78$		0
$78 / 2 = 39$		0
$39 / 2 = 19$		1
$19 / 2 = 9$		1
$9 / 2 = 4$		1
$4 / 2 = 1$		0
$2 / 2 = 1$		0
$1 / 2 = 0$		1 (MSB)

Jadi  $625_{(10)} = 1001110001_{(2)}$

#### 2. Konversi Desimal ke Oktal

Contoh konversi  $625_{(10)}$  ke Oktal

$625/8 = 78$	sisa	1(LSB)
$78/8 = 9$		6
$9/8 = 1$		1
$1/8 = 0$		1 (MSB)

Jadi  $625_{(10)} = 116_{(8)}$

#### 3. Konversi Desimal ke Heksadesimal

Konversi ini dilakukan dengan membagi secara berulang-ulang suatu bilangan desimal dengan 16. Sisa setiap pembagian merupakan digit heksadesimal yang didapat.

Contoh: konversi  $625_{(10)}$  ke Heksadesimal.

$$625/16 = 39 \text{ sisa } 1 \text{ (LSB)}$$

$$39/16 = 2 \quad 7$$

$$2/16 = 0 \quad 2 \text{ (MSB)}$$

$$\text{Jadi } 625_{(10)} = 271_{(16)}$$

#### 2.1.4. Bilangan Heksadesimal

Sistem bilangan heksadesimal mempunyai radix 16 sesuai dengan jumlah angka yang terkandung didalamnya. Jadi, didalamnya bilangan heksadesimal digunakan 16 simbol. Simbol yang dipergunakan dalam bilangan heksadesimal: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Huruf-huruf A sampai F dalam bilangan heksadesimal dipakai untuk menggantikan bilangan 10 sampai 15.

**Tabel 2.3** ekuivalen biner dan heksadesimal ke bilangan desimal 0 sampai 17

No.	Desimal	Biner	Heksadesimal
1.	0	0000	0
2.	1	0001	1
3.	2	0010	2
4.	3	0011	3
5.	4	0100	4
6.	5	0101	5
7.	6	0110	6
8.	7	0111	7
9.	8	1000	8
10.	9	1001	9
11.	10	1010	A
12.	11	1011	B
13.	12	1100	C
14.	13	1101	D
15.	14	1110	E
16.	15	1111	F
17.	16	10000	10
18.	17	10001	11

(Tokhem, Roger L, 1990)

### 1. Konversi Heksadesimal ke Biner

Satu digit heksadesimal dikonversi ke 4 bit biner

$$271_{(16)} = 1001110001_{(2)}$$

$$\begin{array}{ccc} 2 & 7 & 1 \\ 10 & 11 & 0001 \end{array}$$

$$10 \quad 11 \quad 0001$$

### 2. Konversi Heksadesimal ke Desimal

$$271_{(16)} = 625_{(10)}$$

$$271_{(16)} = (2 \times 16^2) + (7 \times 16^1) + (1 \times 16^0)$$

$$= 512 + 112 + 1$$

$$= 625_{(10)}$$

### 3. Konversi Heksadesimal ke Oktal

Konversi bilangan heksadesimal ke oktal ini dengan cara mengkonversi bilangan heksadesimal ke biner terlebih dulu kemudian dikonversi lagi ke oktal.

Contoh soalnya dibawah ini:

$$1D5_{(16)} = \dots_{(8)}$$

Heksa	1	D	5			
Biner	000	1	11	01	0	101
Oktal	0	7	2	5		

$$\text{Jadi } 1D5_{(16)} = 725_{(8)}$$

## 2.2. Operasi Bilangan

Pada operasi sistem bilangan ini terdapat 4 operasi bilangan yaitu operasi bilangan biner, operasi bilangan oktal, operasi bilangan desimal dan operasi bilangan heksadesimal.

### 2.2.1. Operasi Bilangan Biner

Operasi bilangan biner adalah operasi-operasi perhitungan yang terjadi dalam bilangan biner. Operasi bilangan Biner ada 4 yaitu penjumlahan, pengurangan, perkalian, pembagian.

## 1. Operasi penjumlahan

**Tabel 2.4** Operasi Bilangan Biner

$0 + 0 = 0$
$1 + 0 = 1$
$0 + 1 = 1$
$1 + 1 = 0$ bawa 1 dan pidah keselanjutnya

(Mismail, Budiono, 1998)

Contohnya 1010

$$\begin{array}{r} 1110 + \\ \hline 11000 \end{array}$$

## 2. Operasi pengurangan

Jika angka yang dikurangi lebih kecil dari angka pengurang maka angka 1 akan dipinjam dari kolom berikutnya dan begitu seterusnya.

**Tabel 2.5** Operasi Pengurangan biner

$0 - 0 = 0$
$1 - 0 = 1$
$1 - 1 = 0$
$0 - 1 = 1$ pinjam 1 dari kolom berikutnya

(Mismail, Budiono, 1998)

$$\begin{array}{r} \text{Contohnya} \quad 11101 \\ \quad \quad \quad \underline{1011} \quad - \\ \quad \quad \quad 10010 \end{array}$$

## 3. Operasi Perkalian

**Tabel 2.6** Operasi Perkalian biner

$0 \times 0 = 0$
$1 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 1 = 1$

(Mismail, Budiono, 1998)

$$\begin{array}{r}
 \text{Contohnya} \quad 1110 \\
 \quad \quad \quad \underline{1100} \quad \times \\
 \quad \quad \quad 0000 \\
 \quad \quad \quad 0000 \\
 \quad \quad \quad 1110 \\
 \quad \quad \underline{1110} \quad + \\
 \quad \quad 10101000
 \end{array}$$

#### 4. Operasi Pembagian

**Tabel 2.7** Operasi Pembagi

0 : 0 = 1
0 : 1 = 0
1 : 0 = 1
1 : 1 = 1

(Mismail, Budiono, 1998)

$$\begin{array}{r}
 \text{Contohnya} \quad 101 \overline{)1111101} = 11001 \\
 \quad \quad \quad \underline{101} \quad - \\
 \quad \quad \quad 101 \\
 \quad \quad \quad \underline{101} \quad - \\
 \quad \quad \quad 0101 \\
 \quad \quad \quad \underline{101} \quad - \\
 \quad \quad \quad 0
 \end{array}$$

#### 2.2.2. Operasi Bilangan Oktal

Operasi bilangan oktal adalah operasi-operasi perhitungan yang terjadi dalam bilangan oktal. Operasi bilangan oktal ada 4 yaitu penjumlahan, pengurangan, perkalian, pembagian.

##### 1. Operasi Penjumlahan

Pertambahan bilangan oktal dapat dilakukan seperti halnya bilangan desimal, dengan langkah-langkah sebagai berikut:

- Tambahkan masing-masing kolom secara desimal
- Ubah hasil dari desimal ke oktal
- Tuliskan hasil dari digit paling kanan dari hasil oktal
- Jika hasil pertambahan tiap-tiap kolom terdiri dari 2 digit, maka digit paling kiri merupakan membawa untuk pertambahan pada kolom selanjutnya.

$$\begin{array}{r}
 \text{Contohnya} \quad 2 \ 5_8 \\
 \quad \quad \quad \underline{1 \ 2 \ 7_8} \quad + \\
 \quad \quad \quad 1 \ 5 \ 4_8
 \end{array}$$

## 2. Operasi Pengurangan

Pengurangan bilangan oktal dapat dilakukan seperti halnya bilangan desimal, dengan langkah-langkah sebagai berikut:

- a. Konversikan bilangan yang akan dikurangkan ke bentuk desimal
- b. Kurangkan setiap bilangan secara desimal
- c. Jika b bilangan yang akan dikurangkan nilainya lebih kecil dari bilangan pengurang, maka pinjamlah dari sebelah kirinya dan konversikan ke bentuk desimal.

Contoh:  $154_8$

$$\begin{array}{r} 127_8 \\ \underline{\quad} \\ 25_8 \end{array} -$$

## 3. Operasi Perkalian

Perkalian bilangan oktal dapat dilakukan dengan cara berikut :

- a. Kalikan setiap bilangan secara desimal lalu konversikan hasilnya ke oktal
- b. Jika hasil perkalian setiap bilangan bernilai 2 digit, maka digit yang paling kiri merupakan membawa untuk penjumlahan bilangan berikutnya.

Contoh:  $16_8$

$$\begin{array}{r} 14_8 \times \\ \underline{\quad} \\ 70 \\ 16 + \\ \underline{\quad} \\ 250_8 \end{array}$$

### 2.2.3. Operasi Bilangan Desimal

Operasi bilangan desimal adalah operasi-operasi perhitungan yang terjadi dalam bilangan desimal. Operasi bilangan desimal ada 4 yaitu penjumlahan, pengurangan, perkalian, pembagian. Operasi bilangan desimal ini sama seperti operasi bilangan biasa.

### 2.2.4. Operasi Bilangan Heksadesimal

Operasi bilangan heksadesimal adalah operasi-operasi perhitungan yang terjadi dalam bilangan heksadesimal. Operasi bilangan heksadesimal ada 4 yaitu penjumlahan, pengurangan, perkalian, pembagian.

### 1. Operasi Penjumlahan

Pertambahan bilangan desimal dapat dilakukan dengan cara berikut :

- Tambahkan masing – masing bilangan secara desimal
- Konversikan hasilnya ke heksadesimal
- Jika hasil pertambahan terdiri dari 2 digit maka digit paling kiri merupakan membawa untuk pertambahan bilangan berikutnya.

$$\begin{array}{r} \text{Contohnya } \quad \text{B A D}_{(16)} \\ \quad \quad \quad \text{4 3 1}_{(16)} \quad + \\ \hline \quad \quad \quad \text{F D E}_{(16)} \end{array}$$

### 2. Operasi Pengurangan

Pengurangan bilangan Heksadesimal dapat dilakukan dengan cara berikut :

- Koversikan bilangan yang akan dikurang ke desimal
- Kurangkan setiap bilangan secara desimal
- Jika Bilangan yang akan dikurang lebih kecil dari bilangan pengurang maka pinjam dari sebelah kirinya dan konvesikan pula ke desimal.

$$\begin{array}{r} \text{Contoh: } \text{1 2 E 1}_{(16)} \\ \quad \quad \quad \text{6 2 7}_{(16)} \quad - \\ \hline \quad \quad \quad \text{C B A}_{(16)} \end{array}$$

### 3. Operasi Perkalian

Perkalian bilangan heksadesimal dapat dilakukan dengan cara berikut :

- Kalikan setiap bilangan secara desimal
- Konversikan hasilnya ke heksadesimal
- Jika hasil perkalian setiap bilangan bernilai 2 digit, maka digit yang paling kiri merupakan membawa untuk penjumlahan bilangan berikutnya.

$$\begin{array}{r} \text{Contoh: } \text{A C}_{(16)} \\ \quad \quad \quad \text{1 B}_{(16)} \quad \mathbf{x} \\ \hline \quad \quad \quad \text{7 6 4} \\ \quad \quad \quad \text{A C} \quad + \\ \hline \text{1 2 2 4} \end{array}$$

### 2.3. Komplemen Biner

Komplemen pada dasarnya merubah bentuk pengurangan menjadi bentuk pertambahan.

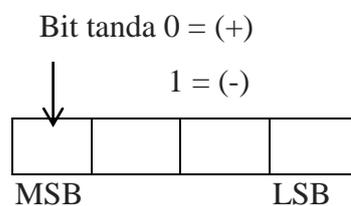
#### 2.3.1. Komplemen Satu

Sistem yang dikenal dengan nama komplemen satu (*ones' complement*) juga dapat digunakan untuk merepresentasikan bilangan negatif. Bentuk komplemen satu untuk bilangan biner negatif diperoleh dengan cara membalik seluruh bit dari bilangan biner positifnya. Bit yang bernilai 0 dibalik menjadi 1, dan bit yang bernilai 1 dibalik menjadi 0. Berikut adalah contoh dari komplemen 1.

Pengurangan Desimal $\begin{array}{r} 10 \\ - 6 \\ \hline 4 \end{array}$	Pengurangan biner $\begin{array}{r} 1010 \\ - 0110 \\ \hline 100 \end{array}$	dan penambah $\longrightarrow$	pengurangan tetnik khusus $\begin{array}{r} 1010 \\ + 1001 \\ \hline 10011 \\ \rightarrow + 1 \\ \hline 100 \end{array}$
	Komplemen 1 Muatan pembuatan akhir		

#### 2.3.2. Komplemen Dua

Di dalam metode komplemen dua, bilangan negatif direpresentasikan dengan cara menambahkan satu pada bentuk komplemen satu dari suatu bilangan positif. Suatu bit tanda 0 berarti angka positif sedangkan bit tanda 1 berarti angka negatif.



**Gambar 2.1** MSB dari register 4 bit berupa bit tanda  
(Tokhem, Roger L, 1990)

MSB merupakan bit tanda dari angka. Pada gambar 2.1 suatu bit tanda 0 berarti angka positif sedangkan bit tanda 1 berarti angka negatif.

**Tabel 2.8** Representasi komplementen 2 dari angka positif dan negatif

(Desimal) Ditandai	Representasi Komplementen 2
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
0	0000
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

(Tokhem, Roger L, 1990)

Tabel 2.8 memperlihatkan representasi komplementen 2 untuk semua angka positif dan negatif 4 bit dari +7 sampai -8. MSB pada gambar 2.1 dari angka komplementen 2 positif adalah 0. Semua angka negatif (-1 sampai -8) mulai dengan 1. Representasi komplementen 2 dari angka negatif diperoleh dengan mula-mula mengambil komplementen 1 dari angka tersebut kemudian menambahkan 1.

Berikut adalah contoh persoalan pengurangan bertanda yang menggunakan angka komplementen 2.

$$\begin{array}{r}
 \begin{array}{l}
 (+7) \quad \text{Bentuk komplementen} \quad 0111 \\
 \text{kedua} \\
 - (+3) = 0011 \xrightarrow{\quad} + 1101 \\
 \hline
 +4
 \end{array} \\
 \begin{array}{l}
 \swarrow 10100 \text{ (komplementen kedua)} \\
 \text{Discard (Dibuang)}
 \end{array}
 \end{array}$$

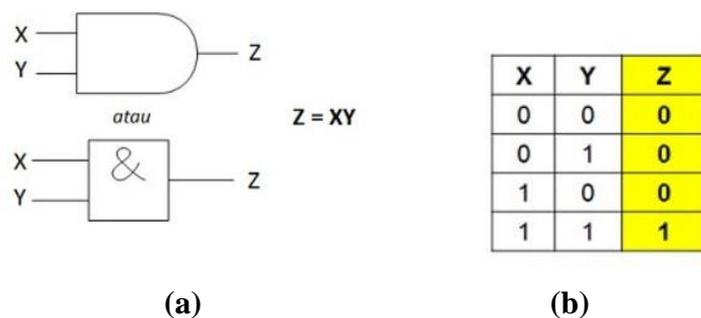
## 2.4 Gerbang Logika

Gerbang Logika atau dalam bahasa Inggris disebut dengan *Logic Gate* adalah dasar pembentuk Sistem Elektronika Digital yang berfungsi untuk mengubah satu atau beberapa input (masukan) menjadi sebuah sinyal output (keluaran) logis. Gerbang logika beroperasi berdasarkan sistem bilangan biner yaitu bilangan yang hanya memiliki 2 kode simbol yaitu 0 dan 1 dengan menggunakan Teori Aljabar Boolean.

Gerbang-gerbang dasar logika merupakan elemen rangkaian digital dan rangkaian digital merupakan kesatuan dari gerbang-gerbang logika dasar yang membentuk fungsi pemroses sinyal digital. Gerbang dasar logika terdiri dari 3 gerbang utama yaitu Gerbang AND, Gerbang OR, dan Gerbang NOT. Gerbang lainnya seperti Gerbang NAND, Gerbang NOR, Gerbang EX-OR dan Gerbang EX-NOR merupakan kombinasi dari 3 gerbang logika utama tersebut.

### 2.4.1 Gerbang AND

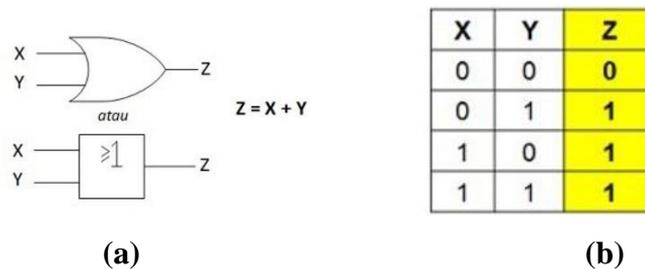
Gerbang AND memerlukan 2 atau lebih masukan untuk menghasilkan keluaran. Gerbang AND akan menghasilkan keluaran logika 1 jika semua masukan bernilai Logika 1 dan akan menghasilkan keluaran logika 0 jika salah satu dari masukan bernilai logika 0. Simbol yang menandakan operasi gerbang logika AND adalah tanda titik (“.”) atau tidak memakai tanda sama sekali. Contohnya  $Z = X.Y$  atau  $Z = XY$ . Simbol dan tabel kebenaran gerbang AND.



**Gambar 2.2** a. Simbol Gerbang AND  
b. Tabel Kebenaran Gerbang AND  
(Tokhem, Roger L, 1990)

### 2.4.2. Gerbang OR

Gerbang OR memerlukan 2 atau lebih masukan untuk menghasilkan hanya 1 keluaran. Gerbang OR akan menghasilkan keluaran 1 jika salah satu dari masukan (Input) bernilai Logika 1 dan jika ingin menghasilkan keluaran logika 0, maka semua masukan harus bernilai logika 0. Simbol yang menandakan operasi logika OR adalah tanda *plus* (“+”). Contohnya :  $Z = X + Y$ . Simbol dan tabel kebenaran gerbang OR



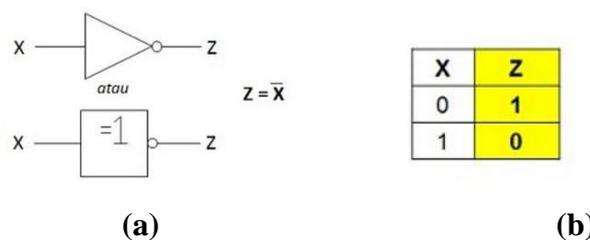
**Gambar 2.3** a. Simbol Gerbang OR

b. Tabel Kebenaran Gerbang OR

(Tokhem, Roger L, 1990)

### 2.4.3. Gerbang NOT

Gerbang NOT hanya memerlukan sebuah masukan untuk menghasilkan hanya 1 keluaran. Gerbang NOT disebut juga dengan *Inverter* (Pembalik) karena menghasilkan keluaran yang berlawanan (kebalikan) dengan masukan atau inputnya. Jika ingin mendapatkan keluaran dengan nilai logika 0 maka input atau masukannya harus bernilai logika 1. Gerbang NOT biasanya dilambangkan dengan simbol *minus* (“-“) di atas variabel inputnya. Simbol dan Tabel Kebenaran Gerbang NOT (NOT Gate)



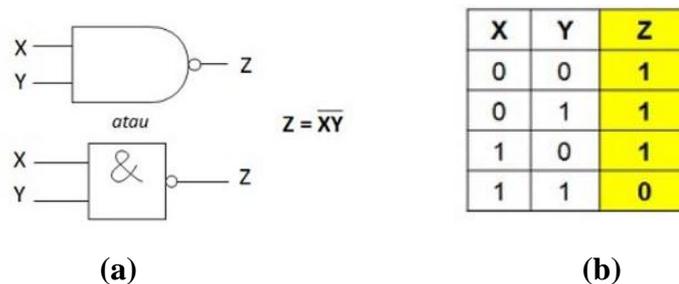
**Gambar 2.4** a. Gambar Simbol Gerbang NOT

b. Tabel Kebenaran Gerbang NOT

(Tokhem, Roger L, 1990)

#### 2.4.4. Gerbang NAND

Arti NAND adalah NOT AND atau BUKAN AND, Gerbang NAND merupakan kombinasi dari Gerbang AND dan Gerbang NOT yang menghasilkan kebalikan dari Keluaran (Output) Gerbang AND. Gerbang NAND akan menghasilkan Keluaran Logika 0 apabila semua Masukan (Input) pada Logika 1 dan jika terdapat sebuah Input yang bernilai Logika 0 maka akan menghasilkan Keluaran (Output) Logika 1. Simbol dan Tabel Kebenaran Gerbang NAND.



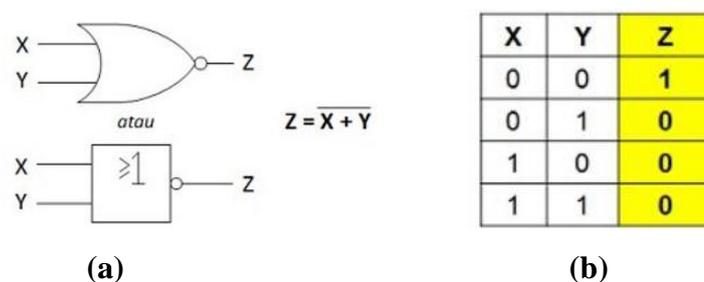
**Gambar 2.5** a. Simbol Gerbang NAND

b. Tabel Kebenaran Gerbang NAND

(Tokhem, Roger L, 1990)

#### 2.4.5. Gerbang NOR

Arti NOR adalah NOT OR atau bukan OR, Gerbang NOR merupakan kombinasi dari gerbang OR dan gerbang NOT yang menghasilkan kebalikan dari keluaran Gerbang OR. Gerbang NOR akan menghasilkan keluaran logika 0 jika salah satu dari masukan bernilai logika 1 dan jika ingin mendapatkan keluaran logika 1, maka semua masukan harus bernilai logika 0. Simbol dan Tabel Kebenaran Gerbang NOR (NOR Gate).



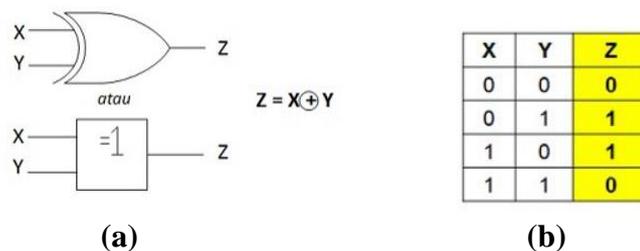
**Gambar 2.6** a. Simbol Gerbang NOR

b. Tabel Kebenaran Gerbang NOR

(Tokhem, Roger L, 1990)

### 2.4.6. Gerbang X-OR

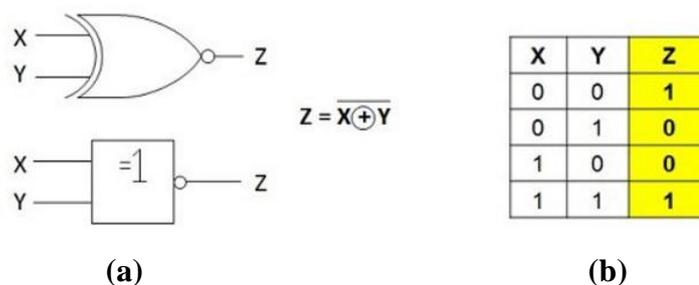
X-OR adalah singkatan dari *Exclusive OR* yang terdiri dari 2 masukan dan 1 keluaran logika. Gerbang X-OR akan menghasilkan keluaran logika 1 jika semua masukan-masukannya mempunyai nilai logika yang berbeda. Jika nilai logika inputnya sama, maka akan memberikan hasil keluaran logika 0. Simbol dan Tabel Kebenaran Gerbang X-OR (X-OR Gate).



(a) (b)  
**Gambar 2.7** a. Simbol Gerbang X-OR  
 b. Tabel Kebenaran X-OR  
 (Tokhem, Roger L.,1990)

### 2.4.7. Gerbang X-NOR

Gerbang X-NOR juga terdiri dari 2 masukan dan 1 keluaran. X-NOR adalah singkatan dari *Exclusive NOR* dan merupakan kombinasi dari gerbang X-OR dan gerbang NOT. Gerbang X-NOR akan menghasilkan keluaran logika 1. Jika semua masukan atau inputnya bernilai logika yang sama dan akan mnghasilkan keluaran logika 0. Jika semua masukan atau inputnya bernilai logika yang berbeda. Hal ini merupakan kebalikan dari gerbang X-OR (Exclusive OR). Simbol dan Tabel Kebenaran Gerbang X-NOR (X-NOR Gate)



(a) (b)  
**Gambar 2.8** a. Simbol EX-NOR  
 b. Tabel Kebenaran EX-NOR  
 (Tokhem, Roger L, 1990)

## 2.5. Counter

Rangkaian counter adalah rangkaian elektronika yang berfungsi untuk melakukan penghitungan angka secara berurutan baik itu perhitungan maju ataupun perhitungan mundur. Yang dimaksud dengan perhitungan maju adalah di mana rangkaian akan menghitung mulai dari angka yang kecil menuju angka yang lebih besar. Sedangkan perhitungan mundur adalah sebaliknya. Perhitungan bisa mencapai jumlah yang tidak terbatas tergantung perancangan rangkaian ataupun tuntutan kebutuhan.

*Counter* merupakan salah satu rangkaian elektronika digital yang menggunakan urutan logika digital dan dipicu oleh pulsa atau *clock* (rangkaiannya sekuensial). *Counter* biasanya mencacah atau menghitung dalam biner dan dapat dibuat untuk berhenti atau berulang ke hitungan awal setiap saat. Pada *counter* yang berulang, jumlah kondisi biner yang berbeda menunjukkan *modulus* (MOD) *counter*. Sebagai contoh, *counter* yang mencacah dari 0-1-2-3-4-5-6-7 secara berulang disebut juga *modulus*8 atau MOD-8. Rangkaian dasar *counter* adalah beberapa flip-flop yang jumlahnya bergantung pada *modulus* yang diperlukan. Secara umum, *counter* terbagi menjadi 2 jenis, yaitu *asynchronous counter* (*ripple counter*) dan *synchronous counter*.

### 2.5.1. Asynchronous Counter (Ripple Counter)

*Asynchronous counter* (*ripple counter*) merupakan *counter* yang masukan *clock* pemicunya tidak terhubung ke setiap flip-flop secara langsung. *Clock* pemicunya harus merambat melalui setiap flip-flop untuk mencapai masukan flip-flop yang berikutnya.

### 2.5.2. Synchronous Counter

*Synchronous counter* merupakan *counter* yang masukan *clock* pemicunya dihubungkan ke setiap flip-flop, sehingga setiap flip-flop akan beroperasi dengan masukan transisi *clock* yang sama. Pada *synchronous counter* harus dipastikan bahwa setiap flip-flop menahan transisi luarannya sampai tiba gilirannya. (Eddy, Purwoko, 2010)

## 2.6. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* yang menyertakan *middleware (virtual machine)* dan sejumlah aplikasi utama. Android merupakan modifikasi dari kernel *linux*. Pada awalnya sistem operasi ini dikembangkan oleh sebuah perusahaan bernama Android Inc. Dari sinilah awal mula nama Android muncul. Android Inc. adalah sebuah perusahaan *start-up* kecil yang berlokasi di Palo Alto, California, Amerika Serikat yang didirikan oleh Andy Rubin bersama Rixh Miner, Nick Sears, dan Chris White. Pada bulan Juli 2005, perusahaan tersebut diakuisisi oleh Google dan para pendirinya bergabung ke Google, Andy Rubin sendiri kemudian diangkat menjadi Wakil Presiden divisi *Mobile* dari Google.

Tujuan pembuatan sistem operasi ini adalah untuk menyediakan *platform* yang terbuka, yang memudahkan orang mengakses internet menggunakan telepon seluler. Android juga dirancang untuk memudahkan pengembang aplikasi membuat aplikasi dengan batasan yang minim sehingga kreativitas pengembang menjadi lebih berkembang.

Pada bulan Oktober 2008, perangkat pertama yang menggunakan Android diluncurkan di Amerika Serikat, yaitu T-Mobile G1. Perangkat ini aslinya diproduksi oleh HTC dan merupakan satu-satunya perangkat android komersial yang pernah menggunakan Android versi 1.0 dan 1.1 (Andry, 2011).

Berikut adalah tingkatan android sejak dimulainya pengembangan pada tahun 2008 sampai pada tahun 2015(Murya, 2014): Android versi 1.1, Android versi 1.5( Cupcake), Android versi 1.6 (Donut), Android versi 2.0/2.1 (Enchair), Android versi 2.2 (Frozen Yoghut), Andriod versi 2.3 (Gingerbread), Android versi 3.1 (Honeycomb), Android versi 4.0 (ICS: Ice Cream Sandwich), Android versi 4.1-4.3(Jelly Bean), Android versi 4.4 (Kitkat), Android versi 5.0/5.1(Lollipop) dan Android versi 6.0 (Mashmallow).



**Gambar 2.9** *Icon* Android  
(developer.android.com)

### 2.6.1. Android versi 1.1

Pada tanggal 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan gmail, dan pemberitahuan email.

### 2.6.2. Android versi 1.5 (Cupcake)

Pada pertengahan bulan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 yang disebut Cupcake. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Android A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, *keyboard* pada layar yang dapat disesuaikan dengan sistem.

### **2.6.3. Android versi 1.6 (Donut)**

Android versi 1.6 yang disebut Donut dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indicator dan control applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; CDMA/EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech engine (tidak tersedia pada semua ponsel; pengadaan VWGA).

### **2.6.4. Android versi 2.0/2.1 (Eclair)**

Pada tanggal 3 Desember 2009 android kembali meluncurkan versi 2.0/2.1 atau Eclair. Perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI (User Interface) dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3.2 MP, digital Zoom, dan Bluetooth 2.1.

### **2.6.5. Android versi 2.2 (Froyo: Frozen Yoghurt)**

Nama Froyo diambil dari kependekan dari Frozen Yoghurt yaitu yoghurt yang telah mengalami proses pendinginan sehingga terlihat seperti es krim. Versi ini dirilis pada bulan Mei 2010 dengan update memperbaiki segi kecepatan dan pengadopsian Javascript dari browser Google Chrome dengan fitur-fiturnya.

### **2.6.6. Android versi 2.3 (Gingerbread)**

Nama Gingerbread diambil dari jahe atau cookie dengan rasa khas jahe yang biasanya berbentuk boneka mirip manusia. Versi ini dirilis pada bulan Desember 2010. Smartphone pertama yang memakai versi android ini adalah Nexus S yang dikeluarkan oleh produsen Samsung. Android 2.3 Gingerbread ini merupakan OS Android yang paling lama berkuasa bahkan sampai sekarangpun beberapa vendor masih mengeluarkan ponsel dengan versi ini.

Adapun perbaikan di versi Gingerbread ini adalah tambahan fitur dukungan untuk SIP internet calling, kemampuan nirkabel NFC, dukungan untuk dual

kamera, dukungan untuk sensor giroskop dan sensor lainnya, fitur download manager, sejumlah tweak untuk penggunaan di Tablet, dan lainnya.

#### **2.6.7. Android versi 3.1 ( Honeycomb)**

Nama Honeycomb diambil dari nama sereal manis yang terbuat dari jagung dengan rasa madu yang berbentuk sarang lebah, makanan ini yang populer sejak tahun 1965. Untuk versi ini merupakan versi yang ditujukan untuk gadget Tablet. Android Honeycomb rilis pada Februari 2011, kemudian upgrade ke versi 3.1 dan 3.2.

#### **2.6.8. Android versi 4.0 (ICS: Ice Cream Sandwich)**

Android Ice Cream Sandwich atau biasa dikenal dengan nama Android 4.0 ICS adalah versi terbaru Android yang sangat mendukung baik untuk smartphone, tablet, dan lainnya. Android ini rilis pada 19 October 2011. Yang baru dalam android ini adalah perubahan interface dari android sebelumnya, antara lain pengoptimalan multitasking, variasi layar beranda yang bisa disesuaikan dan interaktivitas mendalam serta cara baru yang ampuh untuk berkomunikasi dan berbagi konten. Nama Ice Cream Sandwich diambil dari makanan dimana lapisan es krim yang biasanya berupa vanila yang terjepit antara dua cookies coklat, dan biasanya berbentuk persegi panjang.

#### **2.6.9. Android versi 4.1-4.3(Jelly Bean)**

Android 4.1 Jelly Bean diumumkan pada 27 Juni 2012 pada konferensi Google I/O yang secara resmi dikenalkan ke publik sekitar Oktober 2012. Versi ini adalah yang tercepat dan terhalus dari semua versi Android. Fitur baru yang terpat di versi ini adalah meningkatkan kemudahan dan keindahan tampilan dari Ice Cream Sandwich dan memperkenalkan pengalaman pencarian Google yang baru di Android. Android 4.2 Jelly Bean juga menawarkan peningkatan kecepatan dan kemudahan Android 4.1 serta mencakup semua fitur baru seperti Photo Sphere dan desain baru aplikasi kamera, keyboard Gesture Typing, Google

Now dan lainnya. Untuk nama Jelly Bean pasti semua tahu kan, ya nama ini diambil dari sejenis permen yang juga populer disebut kacang jeli.

Kelebihan Android Jelly Bean antara lain:

1. Notifikasi, perbedaan cukup menarik dari android jelly bean. Di android jelly bean pada notification bar terdapat fitur reply untuk pesan, dan fitur call untuk memanggil penelepon saat ada panggilan tak terjawab.
2. Perintah suara, fitur perintah suara atau google assistant ini bisa membuat pesan, browsing internet menggunakan perintah suara. Dalam android versi sebelumnya browsing menggunakan suara memang telah ada, namun uniknya di android jelly bean kita juga bisa mendapatkan jawaban berupa suara.
3. Face unlock, kegunaan face unlock adalah untuk keamanan android anda, saat akan membuka kunci telepon android akan mendeteksi wajah anda dengan cara mengedipkan mata.
4. Google now, fitur ini dapat memberikan informasi kepada pengguna berupa kartu informasi, informasi yang diberikan misalnya berita terkini, cuaca, lokasi, dll
5. Screen capture, berguna untuk memotret layar android anda, sehingga percakapan sms, settingan android, dll dapat diabadikan dan di bagikan ke sosial media dalam bentuk foto.
6. Say it offline, fitur ini dapat membuat suara anda menjadi sebuah teks tanpa harus terkoneksi ke internet.
7. Smart widget, ketika ingin menambahkan widget ke layar utama dengan adanya smart widget ukuran dan bentuk widget akan secara otomatis dapat menyesuaikan dengan ruang yang tersedia. selain itu widget yang telah terpasang akan otomatis bergeser untuk meberikan ruang jika akan menambahkan widget baru.
8. Project butter, responsifitas android kini semakin ditingkatkan pada versi jelly bean, walaupun sebenarnya pada versi ice cream sandwich sudah sangat baik. Project butter selain membuat jelly bean terlihat lebih mempesona juga adanya fitur tritple buffer, anti flickr, dll

9. Full chrome, Kini dengan jelly bean anda akan merasakan berselancar di internet seperti layaknya menggunakan komputer. Tampilan google chrome dalam android sama seperti di komputer.
10. Hemat baterai, seperti yang kita ketahui android OS sangat boros baterai terlebih jika terus-terusan digunakan untuk internet baterai akan cepat habis dan panas. Android jelly bean kini lebih dapat menghemat daya.

#### **2.6.10. Android versi 4.4 (Kitkat)**

Awalnya android versi ini di isukan bernama Key Lime Pie. Namun pada tanggal oktober 2013 google merilis kitkat sebagai generasi android berikutnya. Android versi ini memiliki banyak fitur & semakin memanjakan para pengguna android. Diantaranya : Immersive mode, Akses kontak langsung dari aplikasi telepon, google now launcher, dan pastinya memiliki interface UI yang baru.

#### **2.6.11. Android versi 5.0/5.1 (Lollipop)**

Android lollipop adalah versi stabil dengan antara 5.0 dan 5.1. diresmikan pada tanggal 24 Juli 2014. Salah satu perubahan yang paling menonjol adalah *user interface* yang didesain ulang dan dibangun dengan yang dalam bahasa desain disebut sebagai material desain. Perubahan lain termasuk perbaikan pemberitahuan yang dapat diakses dari *lockscreen* dan ditampilkan pada banner dibagian bagian atas screen. Google juga membuat perubahan internal untuk *platform*, dengan android *runtime* (ART) secara resmi menggantikan Dalvik untuk meningkatkan kinerja aplikasi, dan dengan perubahan yang ditunjukkan untuk meningkatna dan mengoptimalkan penggunaan baterai, yang dikenal secara internal sebagai *Project Volta*.

#### **2.6.12. Android versi 6.0 (Marshmallow)**

Android 6.0 Marshmallow adalah lanjutan pengembangan dari android lollipop. OS ini diperkenalkan pada bulan Mei 2015 di Google I/O dengan codename Android 'M' dan resmi dirilis pada bulan Oktobel 2015. Marshmallow berfokus untuk meningkatkan pengalaman dari pengguna Lollipop dengan

memperkenalkan arsitektur app permission yang baru, API baru untuk menyempurnakan asisten kontekstual dalam Google Search, lalu sebuah sistem (Doze) manajemen daya baru yang bisa mengurangi aktivitas pada background jika perangkat sedang tidak digunakan, kemudian dukungan asli untuk pengenalan sidik jari dan konektor USB tipe- C. Ditambah juga dengan kemampuan untuk migrasi data ke kartu micro SD dan menggunakannya sebagai penyimpan utama.

## 2.7. Android Studio



**Gambar 2.10** Logo Android Studio  
([developer.android.com](http://developer.android.com))

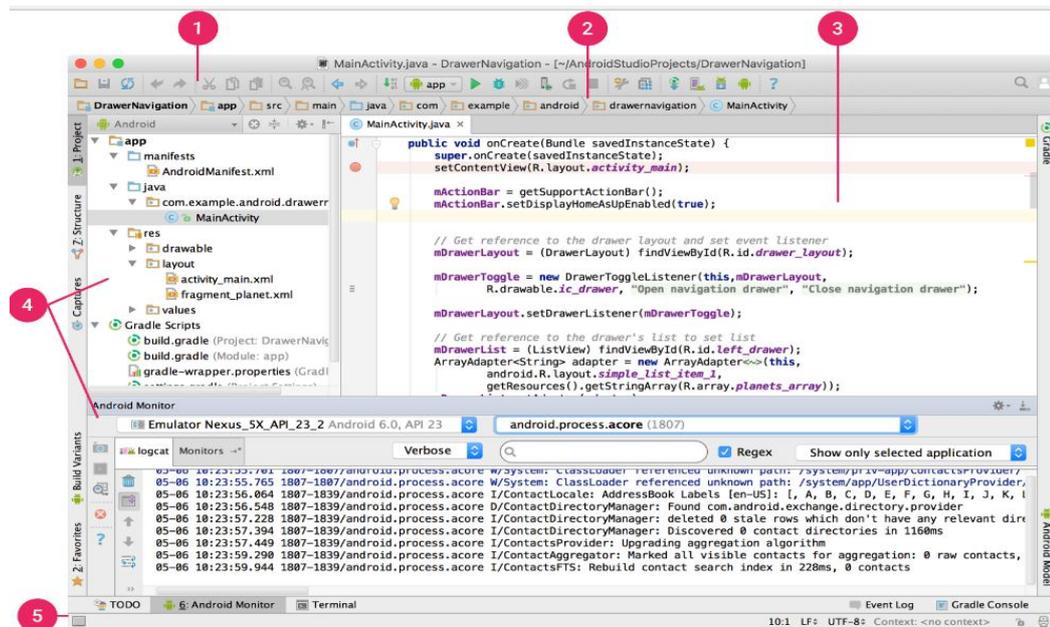
Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

1. Sistem pembuatan berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh

6. Alat penguji dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

### 2.7.1. Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa bidang logika yang diidentifikasi dalam gambar dibawah ini.



**Gambar 2.11** Jendela utama Android Studio.  
(developer.android.com)

1. Alat memungkinkan untuk melakukan berbagai jenis tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. Navigasi akan membantu menavigasi di antara proyek dan file yang dibuka untuk pengeditan. Di sini tampilan struktur yang terlihat tampak lebih ringkas daripada jendela Project.

3. Jendela editor merupakan tempat membuat dan mengubah kode. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat file tata letak, editor akan menampilkan Layout Editor.
4. Jendela alat memberi akses ke tugas-tugas spesifik seperti pengelolaan proyek, penelusuran, kontrol versi, dan banyak lagi. Anda bisa meluaskan dan juga menciutkannya.
5. Bila status menampilkan status proyek dan IDE itu sendiri, serta setiap peringatan atau pesan.

Jadi bisa menata jendela utama untuk memberi ruang layar yang lebih luas dengan menyembunyikan atau memindahkan bila alat dan jendela alat. juga bisa menggunakan pintasan *keyboard* untuk mengakses sebagian besar fitur IDE.

Dapat menelusuri seluruh kode sumber, basis data, tindakan, elemen antarmuka pengguna, dan seterusnya setiap saat dengan menekan tombol Shift dua kali, atau mengeklik kaca pembesar di sudut kanan atas dari jendela Android Studio. Ini akan sangat berguna misalnya saat mencoba menemukan tindakan IDE tertentu yang terlupakan cara memicunya.([developer.android.com](http://developer.android.com))

## 2.8. JAVA



**Gambar 2.12** Logo Java  
( [Java.com](http://Java.com))

Java adalah bahasa pemrograman yang dapat dijalankan diberbagai computer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James

Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan diliris tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya di kompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, “Tulis sekali, jalankan dimanapun”. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.(java.com)

## 2.9. Extensible Markup Language

*Extensible Markup Language* (XML) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari *HyperText Markup Language* (HTML) yang merupakan bahasa standar untuk melacak Internet.

XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang *view* database, tetapi dengan suatu cara yang standar. XML memiliki tiga tipe file antara lain sebagai berikut:

1. XML merupakan standar format dari struktur berkas (*file*) yang ada.

2. XSL merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD merupakan standar yang mendefinisikan struktur database dalam XML.(wikipedia)

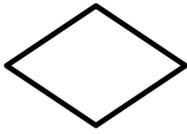
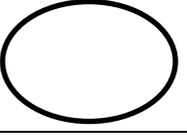
### 2.10. Flowchart

*Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung.

*Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

**Tabel 2.9** Simbol Flowchart

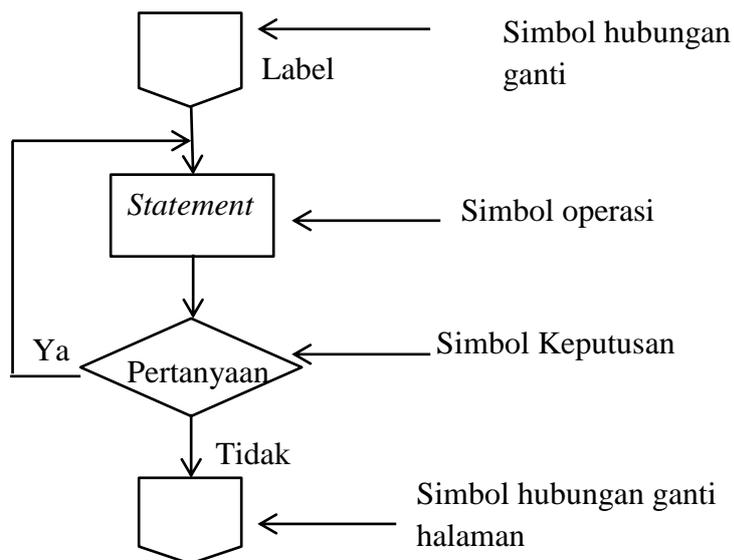
No	Simbol	Nama	Fungsi
1.		Flow Direction Symbol	Arah aliran program
2.		Terminator symbol	Permulaan atau akhir program.
3.		Preparation	Proses inialisasi/ pembentukan harga awal
4.		Processing Symbol	Proses perhitungan atau proses pengolahan data.
5.		Input- Output Symbol	Proses input atau output data, parameter, informasi.

6.		Predefined Process (Sub Program)	Permulaan sub program/ proses menjalankan sub program.
7.		Decision Symbol	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
8.		On Page Connector Symbol	Penghubung bagian- bagian flowchart yang berada pada satu halaman.
9.		Off Page Connector Symbol	Penghubung bagian- bagian flowchart yang berada pada halaman berbeda.

(Lee, Samuel C, 1991)

### 2.10.1. Contoh Penggunaan Berbagai Simbol Flowchart

Penggunaan *flowchart* berhubungan dengan penggunaan diagram blok dalam perancangan aplikasi. Perancangan aplikasi membagi masalahnya secara garis besar ke dalam *flowchart* yang semakin terinci, sampai masing-masing blok menyatakan suatu fungsi.



**Gambar 2.13** Contoh *Flowchart*  
(Lee, Samuel C, 1991)