

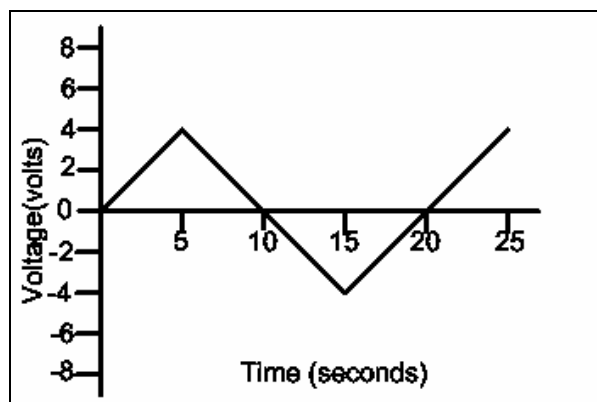
2. Dasar dari Komputer, Sistem Bilangan, dan Gerbang logika

2.1. Data

Komputer yang dipakai saat ini adalah sebuah pemroses data. Fungsinya sangat sederhana : Untuk memproses data, kemudian hasil prosesnya diselesaikan secara elektronik didalam CPU (Central Processing Unit) dan komponen lainnya yang menyusun sebuah komputer personal. Tampaknya sederhana, tetapi apa sebenarnya data?, dan bagaimana data diproses secara elektronik didalam komputer personal?.

2.1.1. Analog

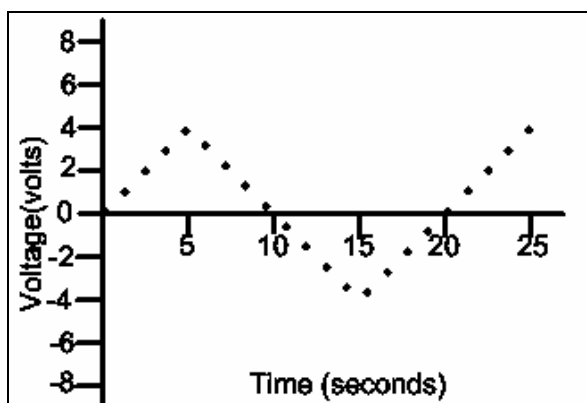
Suatu sinyal yang dikirimkan dari suatu pemancar (transmitter) ke penerima (receiver) untuk berkomunikasi, adalah data. Data-data yang bisa dijumpai sehari-hari memiliki banyak bentuk, antara lain: suara, huruf, angka, dan karakter lain (tulisan tangan atau dicetak), foto, gambar, film dan lain sebagainya. Suatu sistem yang dapat memproses nilai yang kontinyu berbanding terhadap waktu dinamakan sistem analog. Pada sistem analog, nilainya biasa diwakili oleh tegangan, arus dan kecepatan. Berikut ini adalah gambar grafik nilai tegangan analog terhadap waktu.



Gambar 2.1.

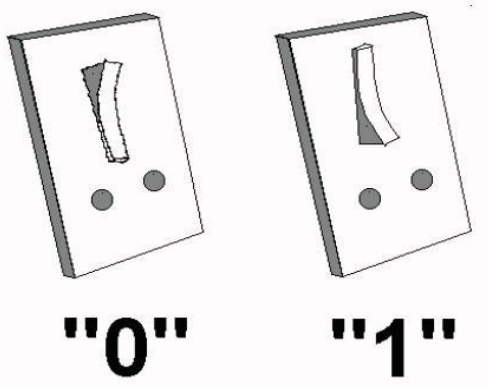
2.1.2. Digital

Sistem yang memproses nilai diskrit (langkah demi langkah) dinamakan digital. Pada sistem digital untuk menunjukkan suatu nilai digunakan simbol yang dinamakan digit. Sinyal pada gambar 2.1. diatas dapat “didigitalkan” dengan menggunakan ADC (*Analog to Digital Converter*). ADC mengubah sinyal kontinyu menjadi sinyal diskrit dengan menyamplingnya tiap detik (tiap satuan waktu). Perhatikan gambar 2.2. berikut:



Gambar 2.2.

Komputer adalah sebuah perangkat elektronik. Data yang dapat diolah adalah data yang direpresentasikan oleh sinyal listrik. Sinyal yang digunakan bisa dianalogikan dengan saklar listrik, yaitu tombol off (mati) atau on (hidup). Jika saklar pada kondisi off, maka komputer membaca sebagai data 0, jika saklar dalam kondisi hidup, maka komputer membaca sebagai angka 1. Perhatikan gambar 2.3. berikut :



Gambar 2.3.

Sebuah komputer personal terdiri dari saklar-saklar yang banyak jumlahnya (menggunakan komponen elektronik berupa transistor). Jumlah dari transistor yang digunakan bisa sampai jutaan, sehingga dapat memproses data dari jutaan angka 0 dan 1.

2.1.2.1. Bits

Setiap angka 0 dan 1 biasa disebut **Bit**. Bit adalah singkatan dari **Binary Digit**. Kata *Binary* diambil dari nama *Binary Number System* (Sistem Bilangan Biner). Tabel 2.1. berikut menunjukkan tentang bit :

0	1 bit
1	1 bit
0110	4 bit
10011101	8 bit

Tabel 2.1.

2.1.2.2. Sistem Bilangan Biner

Sistem bilangan biner disusun dari angka-angka, sama seperti sistem bilangan desimal (sistem bilangan 10) yang sering digunakan saat ini. Tetapi untuk desimal menggunakan angka 0 sampai 9, sistem bilangan biner hanya menggunakan angka 0 dan 1.

Berikut adalah tabel contoh sistem bilangan biner.

Sistem Desimal	Sistem Biner
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Tabel 2.2.

Penjelasan lebih detail tentang bilangan biner dapat dibaca pada modul 2.2. tentang Sistem Bilangan.

2.1.2.3. Bytes

Pengolahan data yang paling sering digunakan adalah pengolah kata (*word processing*), yang akan digunakan sebagai contoh. Ketika melakukan suatu pengolahan kata, komputer bekerja dengan *keyboard*. Ada 101 tombol yang mewakili karakter alphabet A, B, C, dst. Selain itu juga akan ditemui karakter angka 0 sampai dengan 9, dan karakter-karakter lain yang diperlukan, antara lain : ,.-;():_?!"#*%&.

Seluruh karakter yang ada pada keyboard harus didigitalkan. Karakter-karakter tersebut diwakili oleh angka-angka 0 dan 1. Bit yang digunakan adalah 8 bit biner. 8 bit biner dinamakan *Byte*.

8 bit = 1 bytes, sistem inilah yang digunakan. Jika menggunakan 8 bit biner, berapa kombinasi angka yang dapat diwakili?.

Untuk sistem bilangan biner, banyaknya kombinasi dihitung dengan $2^n \leq m$. n adalah jumlah bit, m adalah kombinasi yang dapat diwakili.

Sehingga pada 8 bit biner, dapat mewakili $2^8 = 256$ kombinasi maksimal.

Karakter	Bit	Byte		Karakter	Bit	Byte
A	01000001	65		¼	10111100	188
B	01000010	66		.	00101110	46
C	01000011	67		:	00111010	58
a	01100001	97		\$	00100100	36
b	01100010	98		\	01011100	92

Tabel 2.3.

Ketika mengetik kata “digital” simbol yang digunakan adalah 6 huruf, saat komputer mengolahnya, 6 huruf tersebut didigitalkan menjadi 6 bytes, yang kemudian “diletakkan” pada RAM komputer saat mengetik, dan akan “diletakkan” pada harddisk, jika disimpan.

Tabel berikut menunjukkan perbandingan ukuran unit data

Unit	Definisi	Bytes	Bits	Contoh
Bit (b)	Binary Digit, 0 dan 1	1	1	On/Off, buka/tutup
Byte (B)	8 bits	1	8	Kode ASCII
Kilobyte (KB)	1.024 bytes	1000	8000	Ukuran email biasa = 2 KB 10 halaman dokumen = 10 KB
Megabyte (MB)	1.024 kilobytes 1.048.576 bytes	1 juta	8 juta	Floppy disks = 1,44 MB CDROM = 650 MB
Gigabyte (GB)	1.024 megabytes 1.073.741.824 bytes	1 milyar	8 milyar	Hard drive = 40 GB
Terrabyte (TB)	1.024 gigabytes	1 trilyun	8 trilyun	Data yang dapat ditransmit (secara teori) pada <i>fiber optic</i> selama 1 detik.

Tabel 2.4.

Standard yang digunakan sebagai digitalisasi alphanumerik adalah ASCII.

2.1.2.4. ASCII

ASCII singkatan dari *American Standard Code for Information Interchange*. Standard yang digunakan pada industri untuk mengkodekan huruf, angka, dan karakter-karakter lain pada 256 kode (8 bit biner) yang bisa ditampung.

Tabel ASCII dibagi menjadi 3 seksi:

- a. Kode sistem tak tercetak (*Non Printable System Codes*) antara 0 – 31.
- b. ASCII lebih rendah (*Lower ASCII*), antara 32 – 137. Diambil dari kode sebelum ASCII digunakan, yaitu sistem American ADP, sistem yang bekerja pada 7 bit biner.
- c. ASCII lebih tinggi (*Higher ASCII*), antara 128 – 255. Bagian ini dapat diprogram, sehingga dapat mengubah-ubah karakter.

2.1.2.5. Program Code

Telah disebutkan diatas tentang data yang digunakan pada komputer. Tetapi begitu banyak data yang ada pada komputer personal. Tipe data dasar dapat dikelompokkan menjadi 2 :

- a. Program Code, dimana data digunakan untuk menjalankan fungsi komputer.
- b. Data User, seperti teks, gambar dan suara.

Suatu komputer harus memiliki instruksi-instruksi agar dapat berfungsi sebagaimana fungsinya. Hal ini akan dijelaskan lebih detail pada modul 3. CPU didesain untuk mengenali instruksi-instruksi ini, yang kemudian diproses bersama-sama data user.

Program Code adalah kumpulan instruksi-instruksi, dieksekusi satu persatu, ketika program dijalankan. Saat meng-klik mouse, atau mengetikkan sesuatu pada *keyboard*, instruksi-instruksi dikirimkan dari *software* (perangkat lunak) ke CPU.

2.1.2.6. Files

Program Code dan Data User disimpan sebagai *file* pada media penyimpanan. Tipe *file* dapat dikenali dari ekstensi *file* tersebut. Berikut adalah contohnya :

	Contoh nama <i>file</i>
<i>Program Code</i>	Start.exe, win.com, help.dll, vmm32.vxd
Data User	Letter.doc, house.bmp, index.htm

Tabel 2.5.

Tabel 2.5. diatas menunjukkan tentang penamaan suatu *file*. Ekstensi suatu *file* menentukan bagaimana PC menanganinya.

2.2. Sistem Bilangan

2.2.1. Desimal

Sebelum mempelajari tentang bilangan biner, ada baiknya mengetahui tentang sistem bilangan yang umum dipakai, yaitu desimal (bilangan basis 10). Perhatikan tabel 1.6. berikut:

Base ^{Exponent}	$10^2 = 100$ $10^1 = 10$ $10^0 = 1$
Jumlah simbol (radiks)	10
Simbol	0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Tabel 2.6.

Untuk menghitung suatu basis bilangan, harus dimulai dari nilai yang terkecil (yang paling kanan). Pada basis 10, maka kalikan nilai paling kanan dengan 10^0 ditambah dengan nilai dikirinya yang dikalikan dengan 10^1 , dst. Untuk bilangan dibelaang koma, gunakan faktor pengali 10^{-1} , 10^{-2} , dst.

Contoh :

$$1243 = (1 \times 10^3) + (2 \times 10^2) + (4 \times 10^1) + (3 \times 10^0)$$

$$= 1000 + 200 + 40 + 3$$

$$752,91 = (7 \times 10^2) + (5 \times 10^1) + (2 \times 10^0) + (9 \times 10^{-1}) + (1 \times 10^{-2})$$

$$= 700 + 50 + 2 + 0,9 + 0,01$$

2.2.2. Biner

Untuk bilangan biner (bilangan basis 2), perhatikan tabel 2.7. berikut :

Base ^{Exponent}	$2^5 = 32$	$2^2 = 4$
	$2^4 = 16$	$2^1 = 2$
	$2^3 = 8$	$2^0 = 1$
Jumlah simbol (radiks)	2	
Simbol	0, 1	

Tabel 2.7.

Untuk bilangan biner, kalikan bilangan paling kanan terus ke kiri dengan 2^0 , 2^1 , 2^2 , dst.

Contoh :

$$10110_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) \\ = (16 + 0 + 4 + 2 + 0) = 22$$

Dari contoh diatas, menunjukkan bahwa bilangan biner 10110 sama dengan bilangan desimal 22.

Dari dua sistem bilangan diatas, dapat dibuat rumus umum untuk mendapatkan nilai desimal dari radiks bilangan tertentu :

$$(N)_r = [(d_0 \times r^0) + (d_1 \times r^1) + (d_2 \times r^2) + \dots + (d_n \times r^n)]_{10}$$

dimana; N = Nilai

r = Radiks

d_0, d_1, d_2 = digit dari yang terkecil (paling kanan) untuk d_0

Untuk mengkonversi bilangan desimal kebiner ada dua cara, perhatikan contoh berikut :

Cara I :

168_{10} kurangkan dengan pangkat terbesar dari 2 yang mendekati 168_{10} yaitu 128 (2^7).

- 128 (2^7) lebih kecil dari 168, maka bilangan paling kiri adalah 1. $168 - 128 = 40$.
- 64 (2^6) lebih besar dari 40, maka bilangan kedua adalah 0.
- 32 (2^5) lebih kecil dari 40, maka bilangan ketiga adalah 1. $40 - 32 = 8$.

- d. $16 (2^4)$ lebih besar dari 8, maka bilangan keempat adalah 0.
- e. $8 (2^3)$ lebih kecil/sama dengan 8, maka bil. kelima adalah 1. $8 - 8 = 0$.
- f. Karena sisa 0, maka seluruh bit di kanan bil. kelima adalah 0.

$$168_{10} = 10101000_2.$$

Cara II :

$$168 / 2 = 84 \quad \text{sisa } 0$$

$$84 / 2 = 42 \quad \text{sisa } 0$$

$$42 / 2 = 21 \quad \text{sisa } 0$$

$$21 / 2 = 10 \quad \text{sisa } 1$$

$$10 / 2 = 5 \quad \text{sisa } 0$$

$$5 / 2 = 2 \quad \text{sisa } 1$$

$$2 / 2 = 1 \quad \text{sisa } 0$$

$$1 / 2 = 0 \quad \text{sisa } 1$$

Bit biner terbesar dimulai dari bawah, sehingga $168_{10} = 10101000_2$

2.2.3. Heksadesimal

Bilangan heksadesimal biasa disebut bilangan basis 16, artinya ada 16 simbol yang mewakili bilangan ini. Tabel 1.8. berikut menunjukkan konversi bilangan heksadesimal :

Desimal	Biner	Heksadesimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Tabel 2.8.

Untuk konversi bilangan biner ke heksadesimal, perhatikan contoh berikut :

$$10110101010010010_2 = 0001\ 0110\ 1010\ 1001\ 0010$$

$$= 1\ 6\ A\ 9\ 2$$

Jadi bil. biner 10110101010010010 sama dengan bil. heksadesimal 16A92.

Penulisan bilangan heksadesimal biasa juga ditambahkan dengan karakter “0x” didepannya. Nilai 2543_{16} sama nilainya dengan $0x2543$.

2.2.4. Oktal

Bilangan oktal disebut bilangan basis 8, artinya ada 8 simbol yang mewakili bilangan ini. Tabel 1.9. berikut menunjukkan konversi bilangan oktal :

Desimal	Biner	Oktal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Tabel 2.9.

Untuk konversi bilangan biner ke oktal, perhatikan contoh berikut :

$$10110101010010010_2 = 010\ 110\ 101\ 010\ 010\ 010$$

$$= 2\ 6\ 5\ 2\ 2\ 2_8$$

Jadi bil. biner 10110101010010010 sama dengan bil. oktal 265222.

Untuk konversi dari oktal ke heksadesimal, ubah terlebih dahulu bilangan oktal yang akan dikonversi menjadi biner. Hal ini berlaku juga untuk konversi dari heksadesimal ke oktal. Perhatikan contoh berikut :

$$725_8 = 111\ 010\ 101_2$$

$$= 0001\ 1101\ 0101$$

$$= 1\ D\ 5_{16}$$

$$FE_{16} = 1111\ 1110_2$$

$$= 011\ 111\ 110$$

$$= 3\ 7\ 6_8$$

2.3. Sandi Biner

2.3.1 Sandi 8421 BCD (*Binary Coded Decimal*)

Sandi 8421 BCD adalah sandi yang mengkonversi bilangan desimal langsung ke bilangan binernya, sehingga jumlah sandi BCD adalah 10, sesuai dengan jumlah simbol pada desimal. Perhatikan tabel 2.10. berikut :

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

Tabel 2.10.

Contoh :

197_{10} sandi BCD-nya adalah : 0001 1001 0111

2.3.2. Sandi 2421

Sandi 2421 hampir sama dengan sandi 8421, terutama untuk bilangan desimal 0 sampai dengan 4. Tetapi sandi berikutnya merupakan pencerminan yang diinversi.

Perhatikan tabel 2.11. berikut :

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

Tabel 2.11.

Perhatikan sandi desimal 5. Sandi tersebut merupakan cermin dari sandi 4 desimal, tetapi logikanya diinversi. Begitu pula pada sandi desimal 6 yang merupakan cermin dari sandi desimal 3 yang diinversi, dst.

Contoh :

378_{10} sandi 2421-nya adalah : 0011 1101 1110

2.4. Boolean atau Logika Biner

Logika memberi batasan yang pasti dari suatu keadaan. Sehingga keadaan tersebut tidak dapat berada dalam dua ketentuan sekaligus. Karena itu, dalam logika dikenal aturan-aturan sebagai berikut :

- Suatu keadaan tidak dapat benar dan salah sekaligus.
- Masing-masing adalah hanya benar atau salah (salah satu).
- Suatu keadaan disebut BENAR bila TIDAK SALAH.

Dua keadaan itu dalam aljabar boole ditunjukkan dengan dua konstanta, yaitu logika “1” dan logika “0”.

Misal :

Logika “1”	Logika “0”
Benar	Salah
Hidup	Mati
Siang	Malam

Contoh diatas dapat dituliskan :

Tidak Benar atau $\overline{\text{Benar}}$ = Salah

Tidak Hidup atau $\overline{\text{Hidup}}$ = Mati

Tidak Siang atau $\overline{\text{Siang}}$ = Malam

Tanda garis atas dipakai untuk menunjukkan pertentangan atau lawan dari keadaan itu. Sehingga tanda garis tersebut merupakan pertentangan logika (*Logical Inversion*) yang mempunyai fungsi untuk menyatakan “Tidak” (*Not*).

\bar{A} = Tidak A atau \bar{A} = NOT A

Himpunan adalah kumpulan dari elemen yang setidaknya memiliki sifat yang sama, dan bisa memiliki kelompok yang terbatas atau tidak terbatas jumlahnya. Misalnya himpunan mahasiswa politeknik. Himpunan tersebut tentu saja terdiri dari bermacam-macam kelompok. Jika dapat diambil tiga kelompok :

- Kelompok yang berasal dari luar jawa : J.

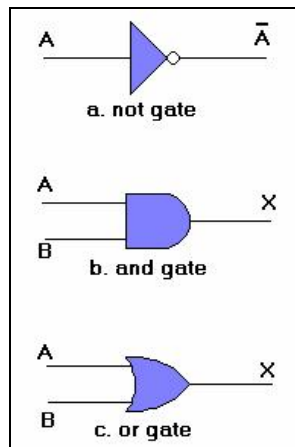
- Kelompok yang sedang kuliah : K.
- Kelompok yang mengerjakan laporan akhir : L.

Sehingga seseorang setidaknya masuk dalam satu kelompok tersebut, bahkan dapat terjadi masuk dalam dua kelompok sekaligus. Misalnya mahasiswa luar jawa yang sedang mengerjakan laporan akhir, berarti masuk kelompok J dan L (J AND L). J AND L dituliskan juga dengan $J \cdot L$.

Gabungan antara mahasiswa luar jawa dan mahasiswa yang mengerjakan laporan akhir memiliki pengertian : mahasiswa luar jawa atau mahasiswa mengerjakan laporan akhir, J atau L (J OR L). J OR L dituliskan juga dengan $J + L$.

Logika Biner (gerbang *Boolean*) adalah rangkaian digital yang menerima satu atau lebih masukan tegangan untuk memperoleh keluaran tertentu sesuai dengan aturan boole yang berlaku.

Jika membicarakan komputer, maka perbedaan tegangan yang digunakan sebagai on/off atau nilai biner 1/0. nilai 1 ekuivalen dengan tegangan +5 volt dan nilai 0 ekuivalen dengan tegangan 0 volt. Perhatikan Gambar 2.4. yang menunjukkan lambang gerbang-gerbang dasar NOT, AND dan OR. Sedangkan Tabel 2.14. menunjukkan tabel kebenaran dari logika gerbang-gerbang dasar yang ada.



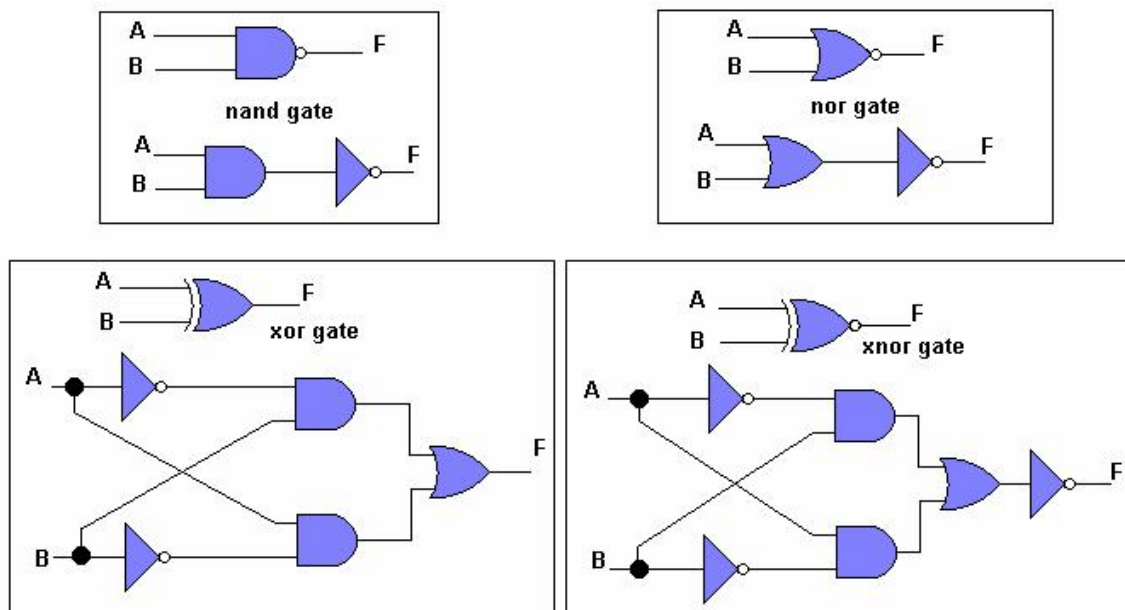
Gambar 2.4.

	Gerbang							
	NOT		AND			OR		
Nilai	A	\bar{A}	A	B	X	A	B	X
	0	1	0	0	0	0	0	0
	1	0	1	0	0	1	0	1
				0	1	0	0	1
			1	1	1	1	1	1

Tabel 2.14.

Gerbang NOT membutuhkan minimal 1 masukan agar dapat berfungsi, sedangkan gerbang lainnya membutuhkan minimal 2 masukan. Dari tabel 1.14. diatas dapat dilihat bahwa gerbang AND hanya akan bernilai 1 pada keluarannya, jika semua masukannya bernilai 1. Sedangkan gerbang OR akan bernilai 1 pada keluarannya, jika salah satu atau semua masukannya bernilai 1. Salah satu contoh komponen penyusun komputer yang menggunakan gerbang adalah memory.

Selain gerbang-gerbang dasar yang telah disebutkan, ada juga gerbang-gerbang kombinasi yang merupakan campuran dari beberapa gerbang dasar. Diantaranya adalah gerbang NAND, NOR, XOR, dan XNOR. Gambar 2.5. berikut menunjukkan tentang lambang-lambang gerbang kombinasi yang ada. Sedangkan tabel 2.15. menunjukkan Tabel kebenaran dari gerbang kombinasi tersebut.



Gambar 2.5.

	Gerbang											
	NAND			NOR			XOR			XNOR		
Nilai	A	B	F	A	B	F	A	B	F	A	B	F
	0	0	1	0	0	1	0	0	0	0	0	1
	0	1	1	0	1	0	0	1	1	0	1	0
	1	0	1	1	0	0	1	0	1	1	0	0
	1	1	0	1	1	0	1	1	0	1	1	1

Tabel 2.15.

Gerbang NAND = NOT AND

$$F = \overline{A \cdot B}$$

Gerbang NOR = NOT OR

$$F = \overline{A + B}$$

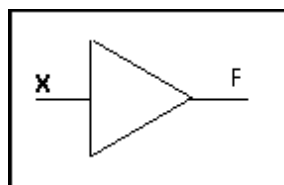
Gerbang XOR = $\overline{A} \cdot B + A \cdot \overline{B}$

$$F = A \oplus B$$

Gerbang XNOR = $\overline{\overline{A} \cdot B + A \cdot \overline{B}}$

$$F = \overline{A \oplus B}$$

Selain gerbang dasar dan gerbang kombinasi diatas, terdapat satu lagi gerbang logika yang berfungsi sebagai penyangga (*Buffer*). Gerbang Buffer tidak mengubah masukan tetapi berfungsi untuk menguatkan sinyal masukan. Selain memperkuat sinyal masukan, *Buffer* juga berfungsi untuk menambah waktu tunda (*time delay*). Gambar 2.6. menunjukkan lambang dari gerbang *Buffer*.



Gambar 2.6.