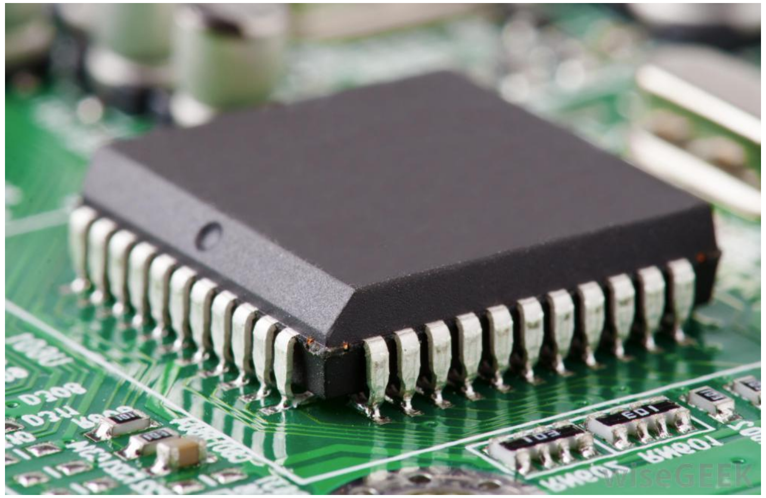
ALJABAR BOOLEAN

Egi Safitri, S.Mat., M.Si

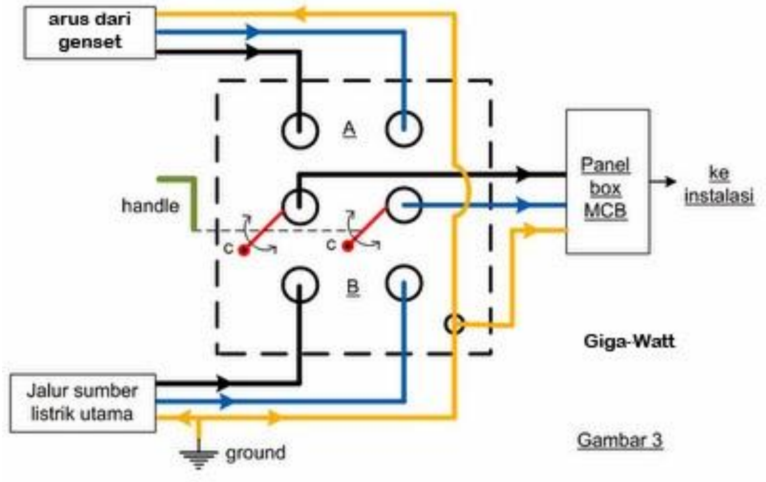
PENGANTAR

- Aljabar Boolean ditemukan oleh George Boole, pada tahun 1854.
- Boole melihat bahwa himpunan dan logika proposisi mempunyai sifat-sifat yang serupa (perhatikan kemiripan hukum-hukum aljabar logika dan hukum-hukum aljabar himpunan).
- Dalam buku *The Laws of Thought*, Boole memaparkan aturan-aturan dasar logika.
- Aturan dasar logika ini membentuk struktur matematika yang disebut **aljabar Boolean.**
- Aplikasi: perancangan rangkaian pensaklaran, rangkaian digital, dan rangkaian *IC* (*integrated circuit*) komputer

Peraga digital



Integarted Circuit (IC)



Jaringan saklar

DEFINISI ALJABAR BOOLEAN

DEFINISI. Misalkan B adalah himpunan yang didefinisikan pada dua operator biner, + dan ·, dan sebuah operator uner, ’. Misalkan 0 dan 1 adalah dua elemen yang berbeda dari B. Maka, tupel

$$<B, +, \cdot, ', 0, 1>$$

disebut **aljabar Boolean** jika untuk setiap $a, b, c \in B$ berlaku aksioma berikut:

1. Identitas
   - (i) $a + 0 = a$
   - (ii) $a \cdot 1 = a$

2. Komutatif
   - (i) $a + b = b + a$
   - (ii) $a \cdot b = b . a$

3. Distributif
   - (i) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
   - (ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$

4. Komplemen

   Untuk setiap $a \in B$ terdapat elemen unik $a' \in B$ sehingga
   - (i) $a + a' = 1$
   - (ii) $a \cdot a' = 0$

Tupel adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen atau nilai, di mana elemen-elemen tersebut bisa memiliki tipe data yang berbeda. Tupel mirip dengan daftar (list), tetapi perbedaan utamanya adalah bahwa tupel bersifat tidak dapat diubah setelah pembuatannya. Artinya, setelah tupel dibuat, Anda tidak bisa menambah, menghapus, atau mengubah nilai-nilai di dalamnya.

- Berhubung elemen-elemen B tidak didefinisikan nilainya (kita bebas menentukan anggota-anggota B), maka terdapat banyak sekali aljabar boolean.

- Untuk mempunyai sebuah aljabar Boolean, orang harus memperlihatkan:
  1. elemen-elemen himpunan B,
  2. kaidah/aturan operasi untuk dua operator biner dan operator uner,
  3. himpunan B, bersama-sama dengan dua operator tersebut, memenuhi keempat aksioma di atas

- Aljabar himpunan dan aljabar logika proposisi juga merupakan aljabar Boolean karena memenuhi empat aksioma di atas.

- Dengan kata lain, aljabar himpunan dan aljabar proposisi adalah himpunan bagian (*subset*) dari aljabar Boolean.

- Pada aljabar proposisi misalnya:

    - B berisi semua proposisi dengan n peubah.

    - dua elemen unik berbeda dari B adalah **T** dan **F**,

    - operator biner: $\vee$ dan $\wedge$, operator uner: $\sim$

    - semua aksioma pada definisi di atas dipenuhi

Dengan kata lain $<B, \vee, \wedge, \sim, \mathbf{F}, \mathbf{T}>$ adalah aljabar Booelan

ALJABAR BOOLEAN 2-NILAI

- Merupakan aljabar Boolean yang paling popular, karena aplikasinya luas.
- Pada aljabar 2-nilai:

(i) $B = \{0, 1\}$,

(ii) operator biner: + dan ·, operator uner: ’

(iii) Kaidah untuk operator biner dan operator uner:

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

a	a'
0	1
1	0

(iv) Keempat aksioma di atas dipenuhi

EKSPRESI BOOLEAN

- Ekspresi Boolean dibentuk dari elemen-elemen *B* dan/atau peubah-peubah yang dapat dikombinasikan satu sama lain dengan operator +, ·, dan ’.

- **Contoh 1:**

  0

  1

  a

  b

  $a + b$

  $a \cdot b$

  $a' \cdot (b + c)$

  $a \cdot b' + a \cdot b \cdot c' + b'$, dan sebagainya

HUKUM-HUKUM ALJABAR BOOLEAN

1. Hukum identitas: (i) $a + 0 = a$ (ii) $a \cdot 1 = a$	2. Hukum idempoten: (i) $a + a = a$ (ii) $a \cdot a = a$
3. Hukum komplemen: (i) $a + a' = 1$ (ii) $aa' = 0$	4. Hukum dominansi: (i) $a \cdot 0 = 0$ (ii) $a + 1 = 1$
5. Hukum involusi: (i) $(a')' = a$	6. Hukum penyerapan: (i) $a + ab = a$ (ii) $a(a + b) = a$

HUKUM-HUKUM ALJABAR BOOLEAN

7. Hukum komutatif: (i) $a + b = b + a$ (ii) $ab = ba$	8. Hukum asosiatif: (i) $a + (b + c) = (a + b) + c$ (ii) $a\ (b\ c) = (a\ b)\ c$
9. Hukum distributif: (i) $a + (b\ c) = (a + b)\ (a + c)$ (ii) $a\ (b + c) = a\ b + a\ c$	10. Hukum De Morgan: (i) $(a + b)' = a'b'$ (ii) $(ab)' = a' + b'$
11. Hukum 0/1 (i) $0' = 1$ (ii) $1' = 0$	

Contoh 2: Buktikan bahwa untuk sembarang elemen a dan b dari aljabar Boolean maka kesamaaan berikut:

$$a + a'b = a + b \quad \text{dan} \quad a(a' + b) = ab$$

adalah benar.

Penyelesaian:

(i) $a + a'b = (a + ab) + a'b$ (Hukum Penyerapan)

$= a + (ab + a'b)$ (Hukum Asosiatif)

$= a + (a + a')b$ (Hukum Distributif)

$= a + 1 \cdot b$ (Hukum Komplemen)

$= a + b$ (Hukum Identitas)

(ii) $a(a' + b) = a\, a' + ab$ (Hukum Distributif)

$= 0 + ab$ (Hukum Komplemen)

$= ab$ (Hukum Identitas)

FUNGSI BOOLEAN

- Contoh-contoh fungsi Boolean:

  $f(x) = x$

  $f(x, y) = x'y + xy' + y'$

  $f(x, y) = x'\, y'$

  $f(x, y) = (x + y)'$

  $f(x, y, z) = xyz'$

- Setiap peubah di dalam fungsi Boolean, termasuk dalam bentuk komplemennya, disebut **literal**.

- Fungsi $h(x, y, z) = xyz'$ terdiri dari 3 buah literal, yaitu x, y, dan z'.

- Jika diberikan $x = 1$, $y = 1$, $z = 0$, maka nilai fungsinya:

  $h(1, 1, 0) = 1 \cdot 1 \cdot 0' = (1 \cdot 1) \cdot 1 = 1 \cdot 1 = 1$

BENTUK KANONIK

- Ekspresi Boolean yang menspesifikasikan suatu fungsi dapat disajikan dalam dua bentuk berbeda.

- Pertama, sebagai **penjumlahan dari hasil kali** dan kedua sebagai **perkalian dari hasil jumlah**.

- **Contoh 3:**

$$f(x, y, z) = x'y'z + xy'z' + xyz$$

dan

$$g(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$$

adalah dua buah fungsi yang sama.

- *Minterm*: suku (*term*) di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk hasil kali

- *Maxterm*: suku (*term*) di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk hasil jumlah.

- **Contoh 4:**

  $f(x, y, z) = x'y'z + xy'z' + xyz$ → 3 buah *minterm*: $x'y'z$, $xy'z'$, xyz

  $g(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$

  → 5 buah *maxterm*: $(x + y + z)$, $(x + y' + z)$, $(x + y' + z')$, $(x' + y + z')$, dan $(x' + y' + z)$

- Misalkan peubah (*variable*) fungsi Boolean adalah x, y, dan z

  Maka:

  $x'y$ → bukan *minterm* karena literal tidak lengkap

  $y'z'$ → bukan minterm karena literal tidak lengkap

  $xy'z$, xyz', $x'y'z$ → *minterm* karena literal lengkap

  $(x + z)$ → bukan maxterm karena literal tidak lengkap

  $(x' + y + z')$ → *maxterm* karena literal lengkap

  $(xy' + y' + z)$ → bukan *maxterm*

- Ekspresi Boolean yang dinyatakan sebagai penjumlahan dari satu atau lebih *minterm* atau perkalian dari satu atau lebih *maxterm* disebut dalam **bentuk kanonik**.

- Jadi, ada dua macam bentuk kanonik:
    1. Penjumlahan dari hasil kali (*sum-of-product* atau SOP)
    2. Perkalian dari hasil jumlah (*product-of-sum* atau POS)

- Fungsi $f(x, y, z) = x'y'z + xy'z' + xyz$ dikatakan dalam bentuk SOP

- Fungsi $g(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$ dikatakan dalam bentuk POS

Cara membentuk *minterm* dan *maxterm*:

- Untuk *minterm*, setiap peubah yang bernilai 0 dinyatakan dalam bentuk komplemen, sedangkan peubah yang bernilai 1 dinyatakan tanpa komplemen.

- Sebaliknya, untuk *maxterm*, setiap peubah yang bernilai 0 dinyatakan tanpa komplemen, sedangkan peubah yang bernilai 1 dinyatakan dalam bentuk komplemen.

- Cara membentuk *minterm* dan *maxterm* dari tabel kebenaran untuk dua peubah:

		Minterm		*Maxterm*	
x	y	Suku	Lambang	Suku	Lambang
0	0	$x'y'$	m_0	$x + y$	M_0
0	1	$x'y$	m_1	$x + y'$	M_1
1	0	xy'	m_2	$x' + y$	M_2
1	1	$x\,y$	m_3	$x' + y'$	M_3

- Cara membentuk *minterm* dan *maxterm* dari tabel kebenaran untuk tiga peubah:

			Minterm		*Maxterm*	
x	y	z	Suku	Lambang	Suku	Lambang
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'y\,z'$	m_2	$x + y'+z$	M_2
0	1	1	$x'y\,z$	m_3	$x + y'+z'$	M_3
1	0	0	$x\,y'z'$	m_4	$x'+ y + z$	M_4
1	0	1	$x\,y'z$	m_5	$x'+ y + z'$	M_5
1	1	0	$x\,y\,z'$	m_6	$x'+ y'+ z$	M_6
1	1	1	$x\,y\,z$	m_7	$x'+ y'+ z'$	M_7

- Jika diberikan sebuah tabel kebenaran, kita dapat membentuk fungsi Boolean dalam bentuk kanonik (SOP atau POS) dari tabel tersebut dengan cara:

  - mengambil *minterm* dari setiap nilai fungsi yang bernilai 1 (untuk SOP)

    atau

  - mengambil *maxterm* dari setiap nilai fungsi yang bernilai 0 (untuk POS).

Contoh 5: Tinjau fungsi Boolean yang dinyatakan oleh Tabel di bawah ini. Nyatakan fungsi tersebut dalam bentuk kanonik SOP dan POS

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Penyelesaian:

- **SOP**

  Kombinasi nilai-nilai peubah yang menghasilkan nilai fungsi sama dengan 1 adalah 001, 100, dan 111, maka fungsi Booleannya dalam bentuk kanonik SOP adalah

$$f(x, y, z) = x'y'z + xy'z' + xyz$$

atau (dengan menggunakan lambang *minterm*),

$$f(x, y, z) = m_1 + m_4 + m_7 = \sum (1, 4, 7)$$

- **POS**

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Kombinasi nilai-nilai peubah yang menghasilkan nilai fungsi sama dengan 0 adalah 000, 010, 011, 101, dan 110, maka fungsi Booleannya dalam bentuk kanonik POS adalah

$$f(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$$

atau dalam bentuk lain,

$$f(x, y, z) = M_0 M_2 M_3 M_5 M_6 = \prod(0, 2, 3, 5, 6)$$

Contoh 6: Nyatakan fungsi Boolean $f(x, y, z) = x + y'z$ dalam bentuk kanonik SOP dan POS.

Penyelesaian:

(a) SOP

Lengkapi terlebih dahulu literal untuk setiap suku agar jumlahnya sama.

$$
\begin{aligned}
x &= x(y + y') \\
&= xy + xy' \\
&= xy\,(z + z') + xy'(z + z') \\
&= xyz + xyz' + xy'z + xy'z'
\end{aligned}
$$

dan

$$y'z = y'z\,(x + x') = \mathrm{xy'z} + \mathrm{x'y'z}$$

$$
\begin{aligned}
\text{Jadi } f(x, y, z) &= x + y'z \\
&= xyz + xyz' + xy'z + xy'z' + xy'z + x'y'z \\
&= x'y'z + xy'z' + xy'z + xyz' + xyz
\end{aligned}
$$

$$\text{atau } f(x, y, z) = m_1 + m_4 + m_5 + m_6 + m_7 = \Sigma\,(1,4,5,6,7)$$

(b) POS

$$f(x, y, z) = x + y'z$$
$$= (x + y')(x + z)$$

Lengkapi terlebih dahulu literal pada setiap suku agar jumlahnya sama:

$$x + y' = x + y' + zz'$$
$$= (x + y' + z)(x + y' + z')$$

$$x + z = x + z + yy'$$
$$= (x + y + z)(x + y' + z)$$

Jadi, $f(x, y, z) = (x + y' + z)(x + y' + z')(x + y + z)(x + y' + z)$
$$= (x + y + z)(x + y' + z)(x + y' + z')$$

atau $f(x, y, z) = M_0M_2M_3 = \prod(0, 2, 3)$

Contoh 7: Nyatakan fungsi Boolean $f(x, y, z) = xy + x'z$ dalam bentuk kanonik POS.

Penyelesaian:

$$
\begin{aligned}
f(x, y, z) &= xy + x'z \\
&= (xy + x')(xy + z) \\
&= (x + x')(y + x')(x + z)(y + z) \\
&= (x' + y)(x + z)(y + z)
\end{aligned}
$$

Lengkapi literal untuk setiap suku agar jumlahnya sama:

$$x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$$

$$x + z = x + z + yy' = (x + y + z)(x + y' + z)$$

$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$

Jadi, $f(x, y, z) = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z')$

atau $f(x, y, z) = M_0 M_2 M_4 M_5 = \prod (0,2,4,5)$

KONVERSI ANTAR BENTUK KANONIK

Misalkan f adalah fungsi Boolean dalam bentuk SOP dengan tiga peubah:

$$f(x, y, z) = \Sigma\ (1, 4, 5, 6, 7)$$

dan f ’adalah fungsi komplemen dari f,

$$f\,'(x, y, z) = \Sigma\ (0, 2, 3) = m_0 + m_2 + m_3$$

Dengan menggunakan hukum De Morgan, kita dapat memperoleh fungsi f dalam bentuk POS:

$$\begin{aligned} f\,(x, y, z) &= (f\,'(x, y, z))' = (m_0 + m_2 + m_3)' = m_0'\,.\,m_2'\,.\,m_3' \\ &= (x'y'z')'\,(x'y\,z')'\,(x'y\,z)' \\ &= (x + y + z)\,(x + y' + z)\,(x + y' + z') \\ &= M_0\,M_2\,M_3 = \prod\ (0,2,3) \end{aligned}$$

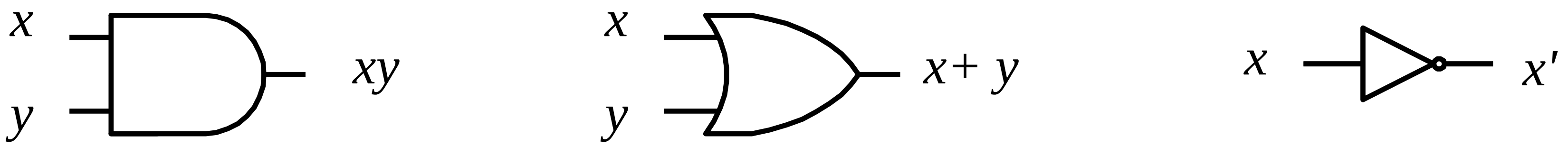
Jadi, $f(x, y, z) = \Sigma\ (1, 4, 5, 6, 7) = \prod\ (0,2,3)$.

<u>**Kesimpulan**</u>: $m_j' = M_j$

RANGKAIAN LOGIKA

- Fungsi Boolean dapat juag direpresentasikan dalam bentuk rangkaian logika.

- Ada tiga gerbang logika dasar: gerbang AND, gerbang OR, dan gerbang NOT

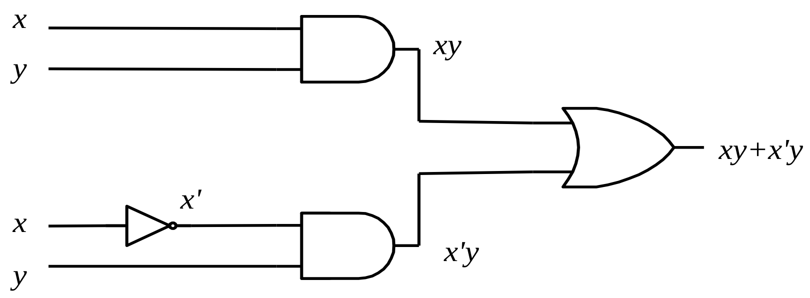


Gerbang AND dua-masukan Gerbang OR dua-masukan Gerbang NOT (*inverter*)

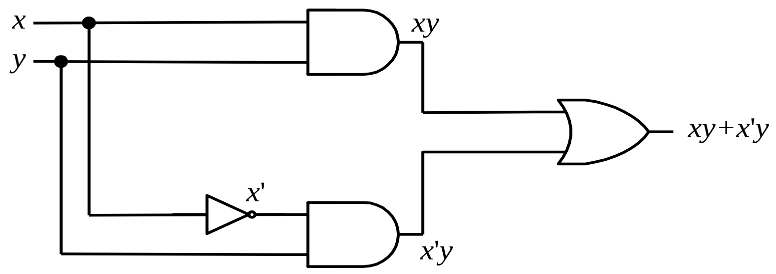
Contoh 8: Nyatakan fungsi $f(x, y, z) = xy + x'y$ ke dalam rangkaian logika.

<u>Penyelesaian</u>: Ada beberapa cara penggambaran

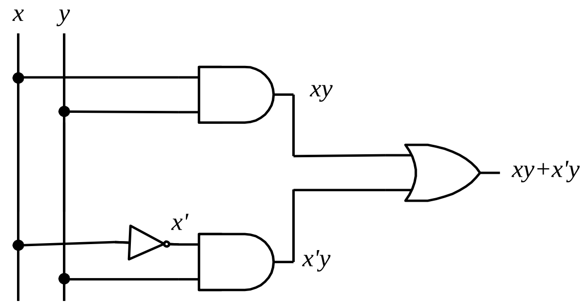
Cara pertama:



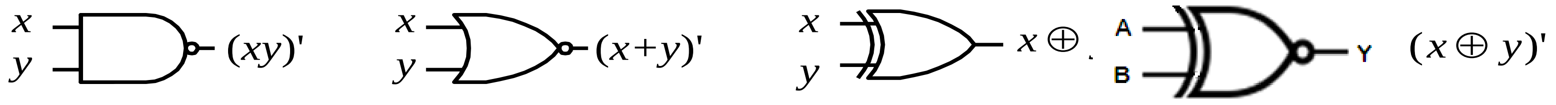
Cara kedua:



Cara ketiga:



- Gerbang logika turunan: NAND, NOR, XOR, dan XNOR

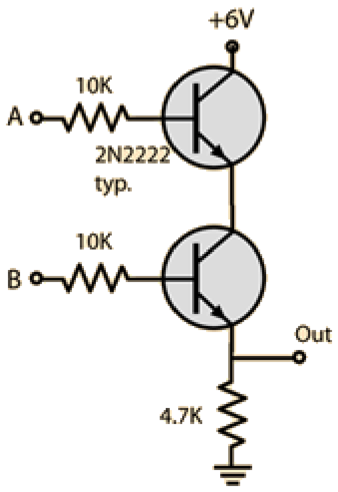


Gerbang NAND Gerbang NOR Gerbang XOR Gerbang XNOR

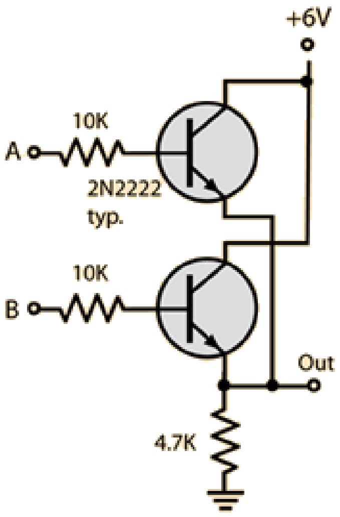
Keempat gerbang di atas merupakan kombinasi dari gerbang-gerbang dasar, misalnya gerbang NOR disusun oleh kombinasi gerbang OR dan gerbang NOT:

Selain itu, dengan menggunakan hukum De Morgan, kita juga dapat membuat gerbang logika yang ekivalen dengan gerbang NOR dan NAND di atas:

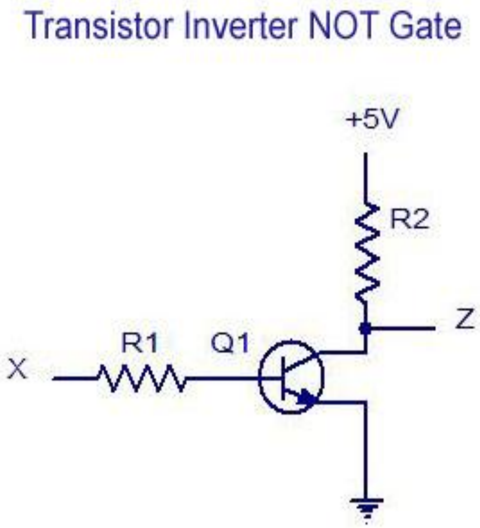
Transistor untuk gerbang logika



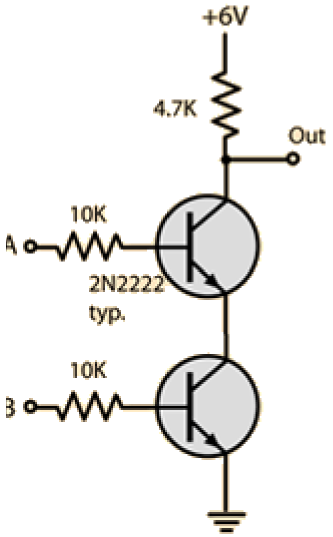
AND



OR



NOT



NAND

Sumber gambar: http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/trangate.html#c3

TERIMA KASIH

Sains Data – Matematika Diskrit