Teknologi Informasi

Bab 9

Aljabar Boolean

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.**

Aljabar Boolean

- *Aljabar boole* adalah suatu teknik matematika yang dipakai untuk menyelesaikan masalah-masalah logika.
- Aljabar boole mendasari operasi-operasi aritmatika yang dilakukan oleh komputer dan juga bermanfaat untuk menganalisis dan mendesain rangkaian yang menjadi dasar bagi pembentukkan komputer sendiri

01100
10110
11110

Operasi Logika dan Gerbang Logika

Operasi *Aljabar Boolean*

- Operasi Invers/NOT
- Operasi AND
- Operasi OR

01100
10110
11110

Operasi *Aljabar Boolean* – Operasi Invers

- Yaitu operasi logika yang mengubah logika 1 menjadi 0 atau sebaliknya.
- Jika suatu variabel x, maka invers x (dibaca : bukan x, x-invers, x-not, x-bar)
- $\bar{A}$ = A’ = A-*invers*

A	$\bar{A}$
0	1
1	0

Operasi *Aljabar Boolean* – Operasi AND

- Operasi AND antara 2 (dua) variabel A dan B ditulis A . B (*dibaca*: A and B)
- A . B bernilai 1, hanya jika A dan B bernilai 1
- Tabel kebenaran A . B

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

Operasi *Aljabar Boolean* – Operasi OR

- Operasi OR antara 2 (dua) variabel A dan B ditulis A + B (dibaca: A or B)
- A + B bernilai 0, hanya jika A dan B bernilai 0
- Tabel kebenaran A + B

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Gerbang Logika

- Gerbang logika adalah piranti dua-keadaan, yaitu mempunyai keluaran dua keadaan,
  - Keluaran dengan nol volt yang menyatakan logika 0 (atau rendah)
  - keluaran dengan tegangan tetap yang menyatakan logika 1 (atau tinggi).
- Gerbang logika dapat mempunyai beberapa masukan yang masing-masing mempunyai salah satu dari dua keadaan logika, yaitu 0 atau 1.

Gerbang Logika (cont.)

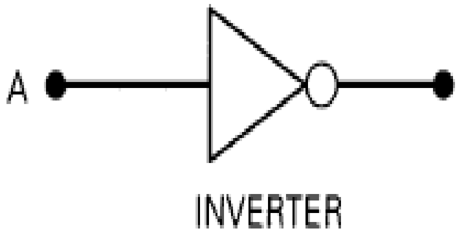
- Gerbang logika dapat digunakan untuk melakukan fungsi-fungsi khusus, misalnya NOT, AND, OR, NAND, NOR, EX-OR (XOR) atau EX-NOR (XNOR).
- Komputer digital pada dasarnya tersusun dari rangkaian gerbang-gerbang logika yang sudah diintegrasikan (IC)
- Bagian-bagian yang membentuk IC terdiri dari transistor-transistor, dioda-dioda dan komponen zat padat lainnya.

Gerbang Logika (cont.)

- NOT
- AND
- OR
- NAND (Not AND)
- NOR (Not OR)
- XOR (Eksklusif OR)
- XNOR (Not XOR)

Gerbang Logika - NOT

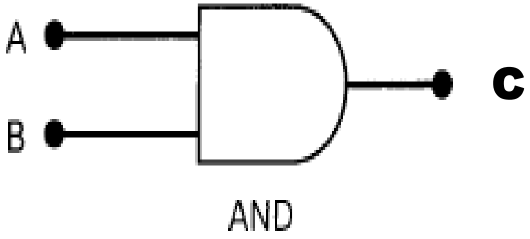
- Gerbang NOT merupakan gerbang satu-masukan yang berfungsi sebagai pembalik (*inverter*). Jika masukannya tinggi, maka keluarannya rendah, dan sebaliknya.



A	$\bar{A}$
0	1
1	0

Gerbang Logika - AND

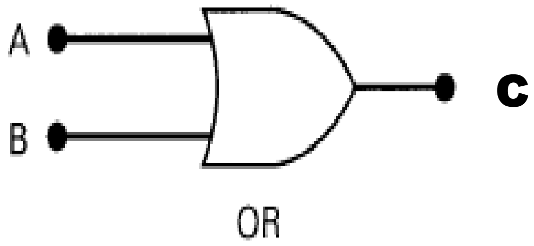
- Gerbang AND digunakan untuk menghasilkan logika 1 jika semua masukan mempunyai logika 1, jika tidak maka akan dihasilkan logika 0.



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Gerbang Logika - OR

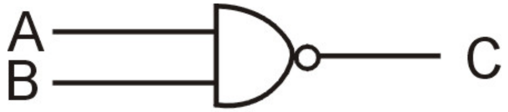
- Gerbang OR akan memberikan keluaran 1 jika salah satu dari masukannya pada keadaan 1. Jika diinginkan keluaran bernilai 0, maka semua masukan harus dalam keadaan 0.



A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Gerbang Logika - NAND

- Kata NAND merupakan kependekan dari NOT-AND, yang merupakan ingkaran dari gerbang AND.
- Gerbang NAND akan mempunyai keluaran 0 bila semua masukan pada logika 1.
- Sebaliknya, jika ada sebuah logika 0 pada sembarang masukan pada gerbang NAND, maka keluarannya akan bernilai 1.



$$C = \overline{A \,.\, B}$$

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Gerbang Logika – X-NOR

- Kata X-NOR merupakan kependekan dari NOT-XOR, yang merupakan ingkaran dari gerbang XOR.
- Gerbang X-NOR akan memberikan keluaran 1 jika masukan-masukannya mempunyai keadaan yang sama.

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

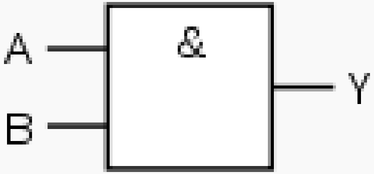
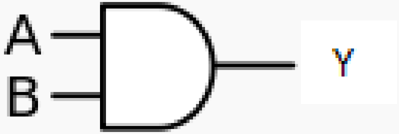
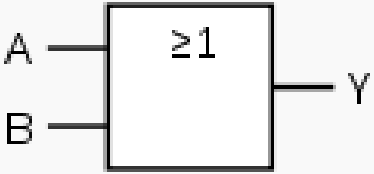
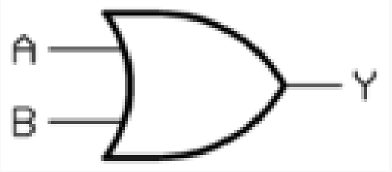
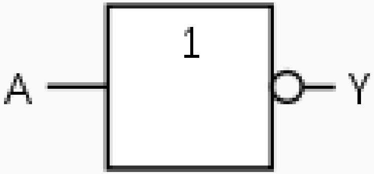
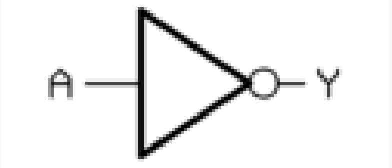
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

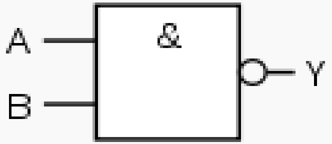
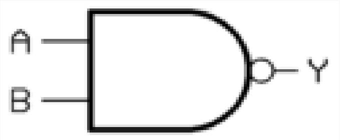
Notasi Boole

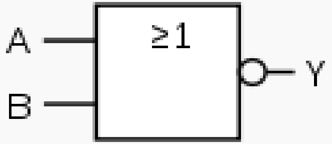
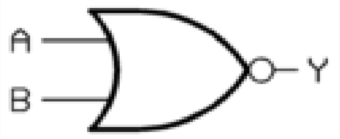
- Keluaran dari satu atau kombinasi beberapa buah gerbang dapat dinyatakan dalam suatu ungkapan logika yang disebut *ungkapan Boole*.
- Teknik ini memanfaatkan aljabar Boole dengan notasi-notasi khusus dan aturan-aturan yang berlaku untuk elemen-elemen logika termasuk gerbang logika.

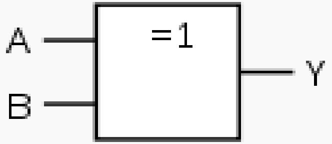
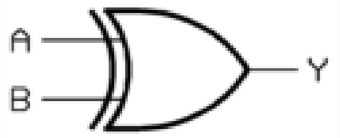
Notasi Boole

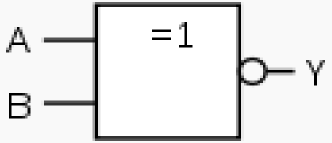
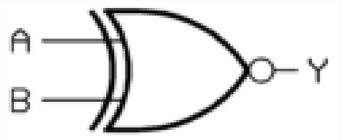
Fungsi	Notasi Boole
NOT	$\overline{A}$
AND	$A \,.\, B = AB$
OR	$A + B$
NAND	$\overline{A \,.\, B}$
NOR	$\overline{A + B}$
EX-OR	$A \oplus B$
EX-NOR	$\overline{A \oplus B}$

NAMA GERBANG	SIMBOL/ LAMBANG DALAM RANGKAIAN		FUNGSI/ KARAKTERISTIK	TABEL KEBENARAN
	SIMBOL IEC	SIMBOL AMERIKA		
ANDGATE (GERBANG AND)			Gerbang AND terdiri dari dua input atau lebih. Jika salah satu input = 0 maka output akan = 0 **Y = A.B**	A B Y 0 0 0 0 1 0 1 0 0 1 1 1
OR GATE (GERBANG OR)			Gerbang OR terdiri dari dua input atau lebih. Jika salah satu input = 1 maka output akan = 1 **Y = A + B**	A B Y 0 0 0 0 1 1 1 0 1 1 1 1
NOT GATE (GERBANG NOT)			Gerbang NOT hanya memiliki satu input. Output merupakan kebalikan dari input **Y = $\bar{A}$**	A Y 0 1 1 0

<table>
<tr>
<td>NAND GATE (GERBANG NAND)</td>
<td>
</td>
<td>
</td>
<td>Gerbang NAND terdiri dari dua input atau lebih. Jika salah satu input = 0 maka output akan = 1

$\mathbf{Y = \overline{A.B}}$</td>
<td><table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
</tr>
<tr>
<td>NOR GATE (GERBANG NOR)</td>
<td>
</td>
<td>
</td>
<td>Gerbang NOR terdiri dari dua input atau lebih. Jika salah satu input = 0 maka output akan = 0

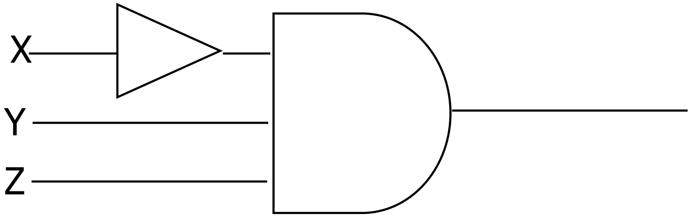
$\mathbf{Y = \overline{A + B}}$</td>
<td><table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
</tr>
<tr>
<td>X-OR GATE (GERBANG X-OR)</td>
<td>
</td>
<td>
</td>
<td>Gerbang X-OR hanya terdiri dari dua input. Jika input sama maka output akan = 0

$\mathbf{Y = A \oplus B}$</td>
<td><table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td>
</tr>
<tr>
<td>X-NOR GATE (GERBANG X-NOR)</td>
<td>
</td>
<td>
</td>
<td>Gerbang X-NOR hanya terdiri dari dua input. Jika input sama maka output akan = 1

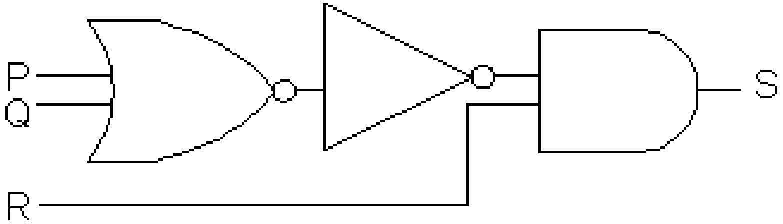
$\mathbf{Y = \overline{A \oplus B}}$</td>
<td><table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td>
</tr>
</table>

Menentukan Fungsi Boole dari Gerbang Logika

Tentukan persamaan boole-nya



$= \bar{X} \cdot Y \cdot Z$



$= \overline{(\overline{P + Q})} \cdot R$

Definisi Aljabar Boolean

Definisi *Aljabar Boolean*

- Misalkan terdapat
  - Dua operator biner: **+** dan ·
  - Sebuah operator uner: **’**
  - *B*: himpunan yang didefinisikan pada operator **+**, ·, dan **’**
  - 0 dan 1 adalah dua elemen yang berbeda dari B.
- Tupel

  (B, **+**, ·, **’**)

  disebut ***Aljabar Boolean*** jika untuk setiap a, b, c $\in$ B berlaku **aksioma-aksioma** atau ***Postulat Huntington*** berikut:

Aksioma / Postulat Huntington

1. ***Closure***: (i) $a + b \in B$
   (ii) $a \cdot b \in B$
2. **Identitas**: (i) $a + 0 = a$
   (ii) $a \cdot 1 = a$
3. **Komutatif**: (i) $a + b = b + a$
   (ii) $a \cdot b = b \cdot a$
4. **Distributif**: (i) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
   (ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$
5. **Komplemen**: (i) $a + a' = 1$
   (ii) $a \cdot a' = 0$

Aksioma / Postulat Huntington (cont.)

- Berhubung elemen-elemen *B* tidak didefinisikan nilainya (kita bebas menentukan anggota-anggota *B*), maka terdapat banyak sekali aljabar boolean.
- Untuk mempunyai sebuah aljabar Boolean, maka 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 aksioma/ Postulat Huntington

Urutan Operasi (*Parentheses*)

- Operasi bilangan biner hanya mengenal AND dan OR
- Jika terjadi operasi AND dan OR bersamaan tanpa ada kurung, maka yang didahulukan adalah AND
- Misal : $x = A.B+C = (A.B)+C$ → A dan B di-**and**-kan dulu, baru di-**or**-kan dengan C
- $A.B+C \neq A.(B+C)$

Aturan Aljabar Boolean

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = A$
5. $A + A = A$
6. $A + \bar{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \bar{A} = 0$
9. $\bar{\bar{A}} = A$
10. $A + AB = A$
11. $A + \bar{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

Aljabar Boolean Dua-Nilai

Aljabar Boolean Dua-Nilai

- $B = \{0, 1\}$
- Operator Biner: **+** dan •
- Operator Uner: ‘
- Kaidah untuk operator biner dan operator uner

A	B	A·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	Ā
0	1
1	0

Aljabar Boolean Dua-Nilai (cont.)

Cek apakah Boolean tersebut memenuhi Postulat Huntington:

1. ***Closure*:** jelas berlaku
2. **Identitas:** jelas berlaku karena dari table dapat diketahui

$$(i)\ \ 1 + 0 = 1$$
$$(ii)\ 1 \cdot 1 = 1$$

3. **Komutatif:** jelas berlaku karena dari table dapat diketahui

$$(i)\ \ 0 + 1 = 1 + 0 = 1$$
$$(ii)\ 1 \cdot 0 = 0 \cdot 1 = 0$$

Aljabar Boolean Dua-Nilai (cont.)

4. ***Distributif***: (i) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ dapat ditunjukkan

a	b	c		b + c	a · (b + c)		a · b	a · c	(a · b) + (a · c)
0	0	0		0	0		0	0	0
0	0	1		1	0		0	0	0
0	1	0		1	0		0	0	0
0	1	1	→	1	0	=	0	0	0
1	0	0		0	0		0	0	0
1	0	1		1	1		0	1	1
1	1	0		1	1		1	0	1
1	1	1		1	1		1	1	1

Aljabar Boolean Dua-Nilai (cont.)

4. (ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$ dapat ditunjukkan seperti poin (i)
5. ***Komplemen:*** jelas berlaku

$$(i)\ \ a + a' = 1 \ \ \rightarrow \ 1 + 0 = 1$$

$$(ii)\ a \ \cdot a' \ = 0 \ \ \rightarrow \ 1 \ \cdot \ 0 = 0$$

Karena kelima postulat Huntington dipenuhi, maka terbukti bahwa B = {0, 1} bersama dengan operator biner (+, ·) dan uner (ʹ) merupakan aljabar Boolean.

Ekspresi Boolean

Ekspresi Boolean

Misalkan (B, +, ·, ’) adalah sebuah aljabar Boolean. Suatu ekspresi Boolean dalam (B, +, ·, ’) adalah:

i. Setiap elemen di dalam B,
ii. Setiap peubah,
iii. Jika e_1 dan e_2 adalah ekspresi Boolean, maka $e_1 + e_2$, $e_1 \cdot e_2$, e_1' adalah ekspresi Boolean

Contoh:

0	$a + b$
1	$a \cdot b$
a	$a' \cdot (b + c)$
b	$a \cdot b' + a \cdot b$, dan sebagainya…

Evaluasi Ekspresi Boolean

- Contoh: $a' \cdot (b + c)$
  jika $a = 0, b = 1, c = 0$, maka hasil evaluasi ekspresi: $0' \cdot (1 + 0)$

- Dua ekspresi Boolean dikatakan **ekivalen** (dilambangkan dengan “=“) jika keduanya mempunyai nilai yang sama untuk setiap pemberian nilai-nilai kepada n peubah.
  Contoh

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

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) $a(a + b) = a$
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) $a\,(b + c) = a\,b + a$	10.Hukum De Morgan: (i) $(a + b)' = a'b'$ (ii) $(ab)' = a' + b'$
11. Hukum 0/1 (i) $0' = 1$ (ii) $1' = 0$	

Contoh

Buktikan (i) $a + a'b = a + b$ dan (ii) $a(a' + b) = ab$

Penyelesaian:

(i)

$$\begin{aligned} a + a'b &= (a + ab) + a'b && \text{(Penyerapan)} \\ &= a + (ab + a'b) && \text{(Asosiatif)} \\ &= a + (a + a')b && \text{(Distributif)} \\ &= a + 1 \bullet b && \text{(Komplemen)} \\ &= a + b && \text{(Identitas)} \end{aligned}$$

(ii)

$$\begin{aligned} a(\mathrm{a}' + \mathrm{b}) &= (\mathrm{a} \bullet \mathrm{a}') + (\mathrm{a} \bullet \mathrm{b}) && \text{(Distributif)} \\ &= 0 + (a \bullet \mathrm{b}) && \text{(Identitas)} \\ &= \mathrm{ab} \end{aligned}$$

Fungsi Boolean

Fungsi Boolean

- **Fungsi Boolean** (disebut juga fungsi biner) adalah pemetaan dari B^n ke B melalui ekspresi Boolean, kita menuliskannya sebagai

$$f : B^n \rightarrow B$$

yang dalam hal ini B^n adalah himpunan yang beranggotakan pasangan terurut ganda-n (*ordered n-tuple*) di dalam daerah asal B.

- Setiap ekspresi Boolean tidak lain merupakan fungsi Boolean.

Fungsi Boolean

- Misalkan sebuah fungsi Boolean adalah

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

Fungsi f memetakan nilai-nilai pasangan terurut ganda-3 (x, y, z) ke himpunan {0, 1}.

Contohnya, (1, 0, 1) yang berarti $x = 1$, $y = 0$, dan $z = 1$

sehingga $f(1, 0, 1) = 1 \cdot 0 \cdot 1 + 1' \cdot 0 + 0' \cdot 1 = 0 + 0 + 1 = 1$.

Pertanyaan?