

Aljabar Boolean

Definisi Aljabar Boolean

Misalkan terdapat

- Dua operator biner: $+$ dan \cdot
- Sebuah operator uner: $'$.
- B : himpunan yang didefinisikan pada operator $+$, \cdot , dan $'$
- 0 dan 1 adalah dua elemen yang berbeda dari B .

Tupel

$$(B, +, \cdot, ')$$

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

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*¹: (i) $a + a' = 1$
(ii) $a \cdot a' = 0$

Untuk mempunyai sebuah aljabar Boolean, harus diperlihatkan:

1. Elemen-elemen himpunan B ,
2. Kaidah operasi untuk operator biner dan operator uner,
3. Memenuhi postulat Huntington.

Aljabar Boolean Dua-Nilai

Aljabar Boolean dua-nilai:

- $B = \{0, 1\}$
- operator biner, $+$ dan \cdot
- operator uner, $'$
- 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

Cek apakah memenuhi postulat Huntington:

1. *Closure* : jelas berlaku
2. Identitas: jelas berlaku karena dari tabel dapat kita lihat bahwa:
 - (i) $0 + 1 = 1 + 0 = 1$
 - (ii) $1 \cdot 0 = 0 \cdot 1 = 0$
3. Komutatif: jelas berlaku dengan melihat simetri tabel operator biner.

4. Distributif: (i) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ dapat ditunjukkan benar dari tabel operator biner di atas dengan membentuk tabel kebenaran:

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

(ii) Hukum distributif $a + (b \cdot c) = (a + b) \cdot (a + c)$ dapat ditunjukkan benar dengan membuat tabel kebenaran dengan cara yang sama seperti (i).

5. Komplemen: jelas berlaku karena Tabel 7.3 memperlihatkan bahwa:

(i) $a + a' = 1$, karena $0 + 0' = 0 + 1 = 1$ dan $1 + 1' = 1 + 0 = 1$

(ii) $a \cdot a = 0$, karena $0 \cdot 0' = 0 \cdot 1 = 0$ dan $1 \cdot 1' = 1 \cdot 0 = 0$

Karena kelima postulat Huntington dipenuhi, maka terbukti bahwa $B = \{0, 1\}$ bersama-sama dengan operator biner $+$ dan \cdot operator komplemen $'$ merupakan aljabar Boolean.

Ekspresi Boolean

- Misalkan $(B, +, \cdot, ')$ adalah sebuah aljabar Boolean. Suatu ekspresi Boolean dalam $(B, +, \cdot, ')$ 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

1

a

b

$a + b$

$a \cdot b$

$a' \cdot (b + c)$

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

Mengevaluasi Ekspresi Boolean

- Contoh: $a' \cdot (b + c)$

jika $a = 0$, $b = 1$, dan $c = 0$, maka hasil evaluasi ekspresi:

$$0' \cdot (1 + 0) = 1 \cdot 1 = 1$$

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

Contoh:

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

Contoh. Perhatikan bahwa $a + a'b = a + b$.

Penyelesaian:

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

- Perjanjian: tanda titik (\cdot) dapat dihilangkan dari penulisan ekspresi Boolean, kecuali jika ada penekanan:

(i) $a(b + c) = ab + ac$

(ii) $a + bc = (a + b)(a + c)$

(iii) $a \cdot 0$, bukan $a0$

Prinsip Dualitas

- Misalkan S adalah kesamaan (*identity*) di dalam aljabar Boolean yang melibatkan operator $+$, \cdot , dan komplemen, maka jika pernyataan S^* diperoleh dengan cara mengganti

- dengan $+$

- $+$ dengan \cdot

- 0 dengan 1

- 1 dengan 0

dan membiarkan operator komplemen tetap apa adanya, maka kesamaan S^* juga benar. S^* disebut sebagai *dual* dari S .

Contoh.

(i) $(a \cdot 1)(0 + a') = 0$ dualnya $(a + 0) + (1 \cdot a') = 1$

(ii) $a(a' + b) = ab$ dualnya $a + a'b = a + b$

Hukum-hukum Aljabar Boolean

<p>1. Hukum identitas:</p> <p>(i) $a + 0 = a$</p> <p>(ii) $a \cdot 1 = a$</p>	<p>2. Hukum idempoten:</p> <p>(i) $a + a = a$</p> <p>(ii) $a \cdot a = a$</p>
<p>3. Hukum komplemen:</p> <p>(i) $a + a' = 1$</p> <p>(ii) $aa' = 0$</p>	<p>4. Hukum dominansi:</p> <p>(i) $a \cdot 0 = 0$</p> <p>(ii) $a + 1 = 1$</p>
<p>5. Hukum involusi:</p> <p>(i) $(a')' = a$</p>	<p>6. Hukum penyerapan:</p> <p>(i) $a + ab = a$</p> <p>(ii) $a(a + b) = a$</p>
<p>7. Hukum komutatif:</p> <p>(i) $a + b = b + a$</p> <p>(ii) $ab = ba$</p>	<p>8. Hukum asosiatif:</p> <p>(i) $a + (b + c) = (a + b) + c$</p> <p>(ii) $a(b c) = (a b) c$</p>
<p>9. Hukum distributif:</p> <p>(i) $a + (b c) = (a + b)(a + c)$</p> <p>(ii) $a(b + c) = ab + ac$</p>	<p>10. Hukum De Morgan:</p> <p>(i) $(a + b)' = a'b'$</p> <p>(ii) $(ab)' = a' + b'$</p>
<p>11. Hukum 0/1</p> <p>(i) $0' = 1$</p> <p>(ii) $1' = 0$</p>	

Contoh 7.3. Buktikan (i) $a + a'b = a + b$ dan (ii) $a(a' + b) = ab$

Penyelesaian:

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

(ii) adalah dual dari (i)

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.

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

Contoh. Contoh-contoh fungsi Boolean yang lain:

1. $f(x) = x$

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

3. $f(x, y) = x' y'$

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

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

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

Contoh: Fungsi $h(x, y, z) = xyz'$ pada contoh di atas terdiri dari 3 buah literal, yaitu x , y , dan z' .

Contoh. Diketahui fungsi Boolean $f(x, y, z) = xy z'$, nyatakan h dalam tabel kebenaran.

Penyelesaian:

x	y	z	$f(x, y, z) = xy z'$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Komplemen Fungsi

1. Cara pertama: menggunakan hukum De Morgan
Hukum De Morgan untuk dua buah peubah, x_1 dan x_2 , adalah

Contoh. Misalkan $f(x, y, z) = x(y'z' + yz)$, maka

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

2. Cara kedua: menggunakan prinsip dualitas.

Tentukan dual dari ekspresi Boolean yang merepresentasikan f , lalu komplementkan setiap literal di dalam dual tersebut.

Contoh. Misalkan $f(x, y, z) = x(y'z' + yz)$, maka

dual dari f : $x + (y' + z')(y + z)$

komplementkan tiap literalnya: $x' + (y + z)(y' + z') = f'$

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

Bentuk Kanonik

- 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)

Contoh: 1. $f(x, y, z) = x'y'z + xy'z' + xyz \rightarrow \text{SOP}$

Setiap suku (*term*) disebut *minterm*

$$2. g(x, y, z) = (x + y + z)(x + y' + z)(x + y' + z') \\ (x' + y + z')(x' + y' + z) \rightarrow \text{POS}$$

Setiap suku (*term*) disebut *maxterm*

- Setiap *minterm*/*maxterm* mengandung literal lengkap

		<i>Minterm</i>		<i>Maxterm</i>	
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	xy	m_3	$x' + y'$	M_3

			<i>Minterm</i>		<i>Maxterm</i>	
<i>x</i>	<i>y</i>	<i>z</i>	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

Contoh 7.10. Nyatakan tabel kebenaran di bawah ini dalam bentuk kanonik SOP dan POS.

Tabel 7.10

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:

(a) 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*),

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

(b) POS

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

Penyelesaian:

(a) SOP

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

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

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

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

Rinaldi Munir / P3151166 / Dit

(b) POS

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

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

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

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

$$\text{atau } f(x, y, z) = M_0 M_2 M_3 = \prod(0, 2, 3)$$

Konversi Antar Bentuk Kanonik

Misalkan

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

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:

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

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

Kesimpulan: $m_i' = M_i$

Contoh. Nyatakan

$$f(x, y, z) = \prod (0, 2, 4, 5) \text{ dan}$$

$$g(w, x, y, z) = \Sigma(1, 2, 5, 6, 10, 15)$$

dalam bentuk SOP.

Penyelesaian:

$$f(x, y, z) = \Sigma (1, 3, 6, 7)$$

$$g(w, x, y, z) = \prod (0, 3, 4, 7, 8, 9, 11, 12, 13, 14)$$

Contoh. Carilah bentuk kanonik SOP dan POS dari $f(x, y, z) = y' + xy + x'yz'$

Penyelesaian:

(a) SOP

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

$$\text{atau } f(x, y, z) = m_0 + m_1 + m_2 + m_4 + m_5 + m_6 + m_7$$

(b) POS

$$f(x, y, z) = M_3 = x + y' + z'$$

Bentuk Baku

- Tidak harus mengandung literal yang lengkap.
- Contohnya,

$$f(x, y, z) = y' + xy + x'yz \text{ (bentuk baku SOP)}$$

$$f(x, y, z) = x(y' + z)(x' + y + z') \text{ (bentuk baku POS)}$$

Aplikasi Aljabar Boolean

1. Jaringan Pensaklaran (*Switching Network*)

Saklar: objek yang mempunyai dua buah keadaan: buka dan tutup.

Tiga bentuk gerbang paling sederhana:

1. $a \text{ ————— } \diagup x \text{ ————— } b$

Output b hanya ada jika dan hanya jika x dibuka $\Rightarrow x$

2. $a \text{ ————— } \diagup x \text{ ————— } \diagdown y \text{ ————— } b$

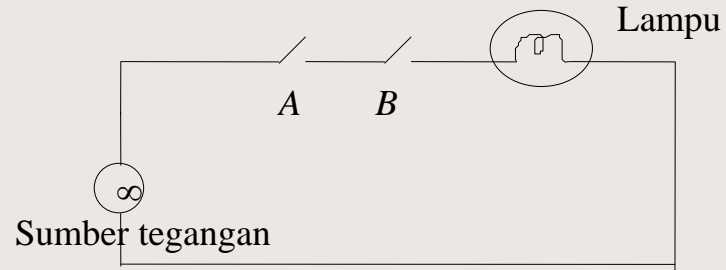
Output b hanya ada jika dan hanya jika x dan y dibuka $\Rightarrow xy$

3.
$$\begin{array}{c} a \text{ ————— } \diagup x \\ b \text{ ————— } \diagdown y \end{array} \Bigg| c$$

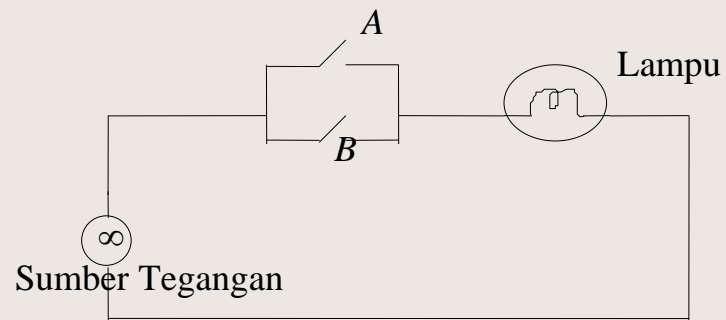
Output c hanya ada jika dan hanya jika x atau y dibuka $\Rightarrow x + y$

Contoh rangkaian pensaklaran pada rangkaian listrik:

1. Saklar dalam hubungan SERI: logika AND



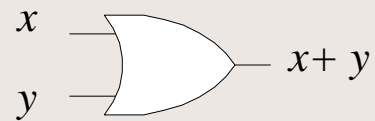
2. Saklar dalam hubungan PARALEL: logika OR



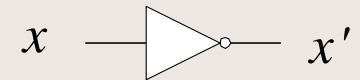
2. Rangkaian Logika



Gerbang AND



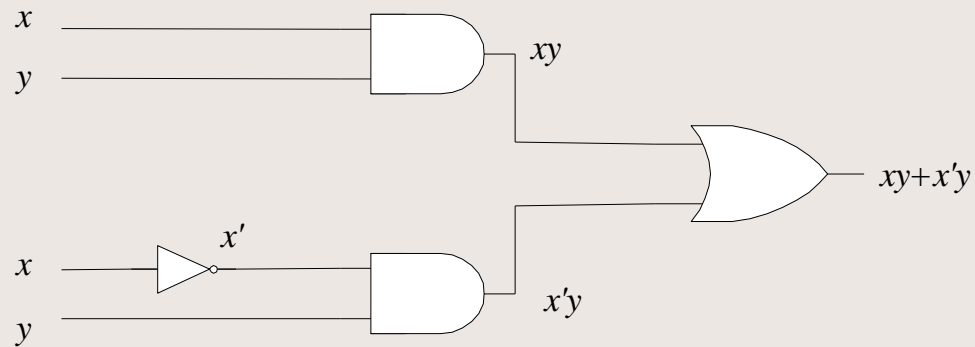
Gerbang OR



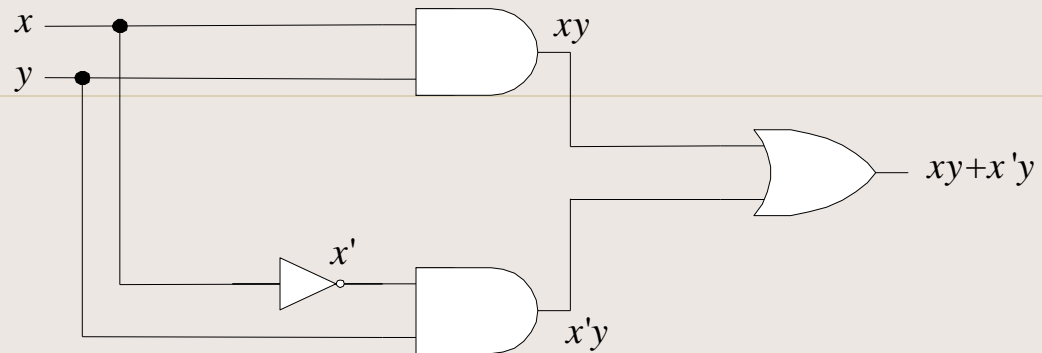
Gerbang NOT (*inverter*)

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

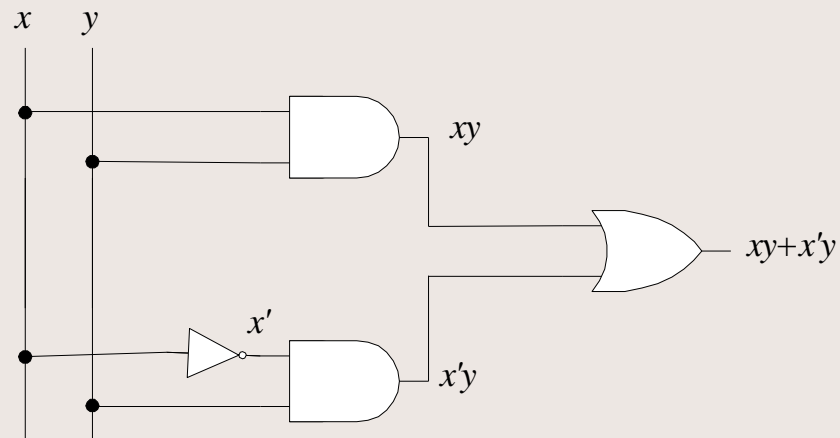
Jawab: (a) Cara pertama



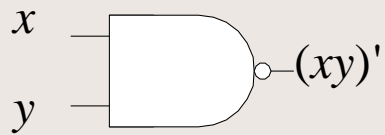
(b) Cara kedua



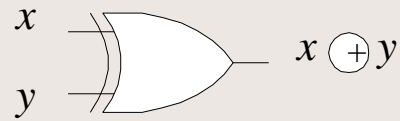
(c) Cara ketiga



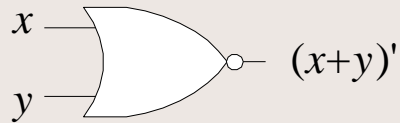
Gerbang turunan



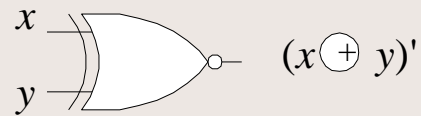
Gerbang NAND



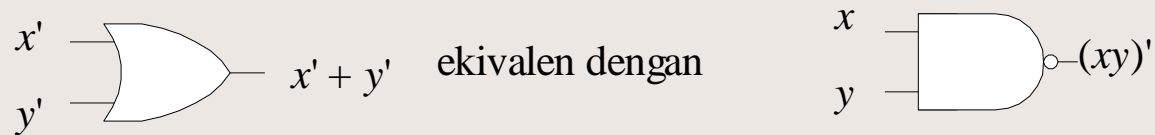
Gerbang XOR



Gerbang NOR



Gerbang XNOR



Penyederhanaan Fungsi Boolean

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

disederhanakan menjadi

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

Penyederhanaan fungsi Boolean dapat dilakukan dengan 3 cara:

1. Secara aljabar
2. Menggunakan Peta Karnaugh
3. Menggunakan metode Quine Mc Cluskey (metode Tabulasi)

1. Penyederhanaan Secara Aljabar

Contoh:

$$\begin{aligned} 1. f(x, y) &= x + x'y \\ &= (x + x')(x + y) \\ &= 1 \cdot (x + y) \\ &= x + y \end{aligned}$$

$$\begin{aligned} 2. f(x, y, z) &= x'y'z + x'yz + xy' \\ &= x'z(y' + y) + xy' \\ &= x'z + xz' \end{aligned}$$

$$3. f(x, y, z) = xy + x'z + yz = xy + x'z + yz(x + x')$$

$$= xy + x'z + xyz + x'yz$$

$$= xy(1 + z) + x'z(1 + y) = xy + x'z$$

Rinaldi Munir/IF2151, Mat. Diskrit

2. Peta Karnaugh

a. *Peta Karnaugh dengan dua peubah*

m_0	m_1
m_2	m_3

		y	
		0	1
x	0	$x'y'$	$x'y$
	1	xy'	xy

b. *Peta dengan tiga peubah*

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

		yz			
		00	01	11	10
x	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	1	$xy'z'$	$xy'z$	xyz	xyz'

Contoh. Diberikan tabel kebenaran, gambarkan Peta Karnaugh.

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

		yz			
		00	01	11	10
x	0	0	0	0	1
	1	0	0	1	1

b. Peta dengan empat peubah

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

		yz			
		00	01	11	10
wx	00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
	01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
	11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
	10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

Contoh. Diberikan tabel kebenaran, gambarkan Peta Karnaugh.

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

		yz			
		00	01	11	10
wx	00	0	1	0	1
	01	0	0	1	1
	11	0	0	0	1
	10	0	0	0	0

Teknik Minimisasi Fungsi Boolean dengan Peta Karnaugh

1. *Pasangan*: dua buah 1 yang bertetangga

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	0	0	1	1
	10	0	0	0	0

Sebelum disederhanakan: $f(w, x, y, z) = wxyz + wxyz'$

Hasil Penyederhanaan: $f(w, x, y, z) = wxy$

Bukti secara aljabar:

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

$$= wxy(z + z')$$

$$= wxy(1)$$

$$= wxy$$

Rinaldi Munir/IF2151 Mat. Diskrit

2. *Kuad*: empat buah 1 yang bertetangga

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	0	0

Sebelum disederhanakan: $f(w, x, y, z) = wxy'z' + wxy'z + wxyz + wxyz'$

Hasil penyederhanaan: $f(w, x, y, z) = wx$

Bukti secara aljabar:

$$\begin{aligned}f(w, x, y, z) &= wxy' + wxy \\&= wx(z' + z) \\&= wx(1) \\&= wx\end{aligned}$$

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	0	0

Contoh lain:

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	0	0
	10	1	1	0	0

Sebelum disederhanakan: $f(w, x, y, z) = wxy'z' + wxy'z + wx'y'z' + wx'y'z$

Hasil penyederhanaan: $f(w, x, y, z) = wy'$

3. *Oktet*: delapan buah 1 yang bertetangga

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	1	1	1	1

Sebelum disederhanakan: $f(a, b, c, d) = wxy'z' + wxy'z + wxyz + wxyz' + wx'y'z' + wx'y'z + wx'yz + wx'yz'$

Hasil penyederhanaan: $f(w, x, y, z) = w$

Bukti secara aljabar:

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

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	1	1	1	1

Contoh 5.12. Andaikan suatu tabel kebenaran telah diterjemahkan ke dalam Peta Karnaugh. Sederhanakan fungsi Boolean yang bersesuaian sesederhana mungkin.

		yz			
		00	01	11	10
wx	00	0	1	1	1
	01	0	0	0	1
	11	1	1	0	1
	10	1	1	0	1

Jawab: (lihat Peta Karnaugh) $f(w, x, y, z) = wy' + yz' + w'x'z$

Contoh 5.13. Minimisasi fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	1	0	0
	11	1	1	1	1
	10	1	1	1	1

Jawab: (lihat Peta Karnaugh) $f(w, x, y, z) = w + xy'z$

Jika penyelesaian Contoh 5.13 adalah seperti di bawah ini:

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	1	0	0
	11	1	1	1	1
	10	1	1	1	1

maka fungsi Boolean hasil penyederhanaan adalah

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

(jumlah literal = 5)

yang ternyata masih belum sederhana dibandingkan $f(w, x, y, z) = w + xy'z$ (jumlah literal = 4).

Contoh 5.14. (Penggulungan/*rolling*) Sederhanakan fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	1	0	0	1
	11	1	0	0	1
	10	0	0	0	0

Jawab: $f(w, x, y, z) = xy'z' + xyz' \implies$ belum sederhana

Penyelesaian yang lebih minimal:

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	1	0	0	1
	11	1	0	0	1
	10	0	0	0	0

$$f(w, x, y, z) = xz' \implies \text{lebih sederhana}$$

Contoh 5.11. Sederhanakan fungsi Boolean $f(x, y, z) = x'yz + xy'z' + xyz + xyz'$.

Jawab:

Peta Karnaugh untuk fungsi tersebut adalah:

		yz			
		00	01	11	10
x	0			1	
	1	1		1	1

Hasil penyederhanaan: $f(x, y, z) = yz + xz'$

Contoh 5.15: (Kelompok berlebihan) Sederhanakan fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	1	0	0
	11	0	1	1	0
	10	0	0	1	0

Jawab: $f(w, x, y, z) = xy'z + wxz + wyz \rightarrow$ masih belum sederhana.

Penyelesaian yang lebih minimal:

		yz			
		00	01	11	10
wx	00	0	0	0	0
	01	0	1	0	0
	11	0	1	1	0
	10	0	0	1	0

$$f(w, x, y, z) = xy'z + wyz \implies \text{lebih sederhana}$$

Contoh 5.16. Sederhanakan fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

		<i>cd</i>			
		00	01	11	10
<i>ab</i>	00	0	0	0	0
	01	0	0	1	0
	11	1	1	1	1
	10	0	1	1	1

Jawab: (lihat Peta Karnaugh di atas) $f(a, b, c, d) = ab + ad + ac + bcd$

Contoh 5.17. Minimisasi fungsi Boolean $f(x, y, z) = x'z + x'y + xy'z + yz$

Jawab:

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

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

$$yz = yz(x + x') = xyz + x'yz$$

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

Peta Karnaugh untuk fungsi tersebut adalah:

		yz			
		00	01	11	10
x	0	0	1	1	1
	1	0	1	1	0

Hasil penyederhanaan: $f(x, y, z) = z + x'yz'$

Peta Karnaugh untuk lima peubah

	000	001	011	010	110	111	101	100
00	m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
01	m_8	m_9	m_{11}	m_{10}	m_{14}	m_{15}	m_{13}	m_{12}
11	m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
10	m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

↑
Garis pencerminan

Contoh 5.21. (Contoh penggunaan Peta 5 peubah) Carilah fungsi sederhana dari $f(v, w, x, y, z) = \Sigma (0, 2, 4, 6, 9, 11, 13, 15, 17, 21, 25, 27, 29, 31)$

Jawab:

Peta Karnaugh dari fungsi tersebut adalah:

		xyz							
		00	00	01	01	11	11	10	10
		0	1	1	0	0	1	1	0
vw	00	1			1	1			1
	01		1	1			1	1	
	11		1	1			1	1	
	10		1					1	

Jadi $f(v, w, x, y, z) = wz + v'w'z' + vy'z$

Kondisi *Don't care*

Tabel 5.16

<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>	desimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	<i>don't care</i>
1	0	1	1	<i>don't care</i>
1	1	0	0	<i>don't care</i>
1	1	0	1	<i>don't care</i>
1	1	1	0	<i>don't care</i>
1	1	1	1	<i>don't care</i>

Contoh 5.25. Diberikan Tabel 5.17. Minimisasi fungsi f sederhana mungkin.

Tabel 5.17

a	b	c	d	$f(a, b, c, d)$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	X
1	0	0	1	X
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

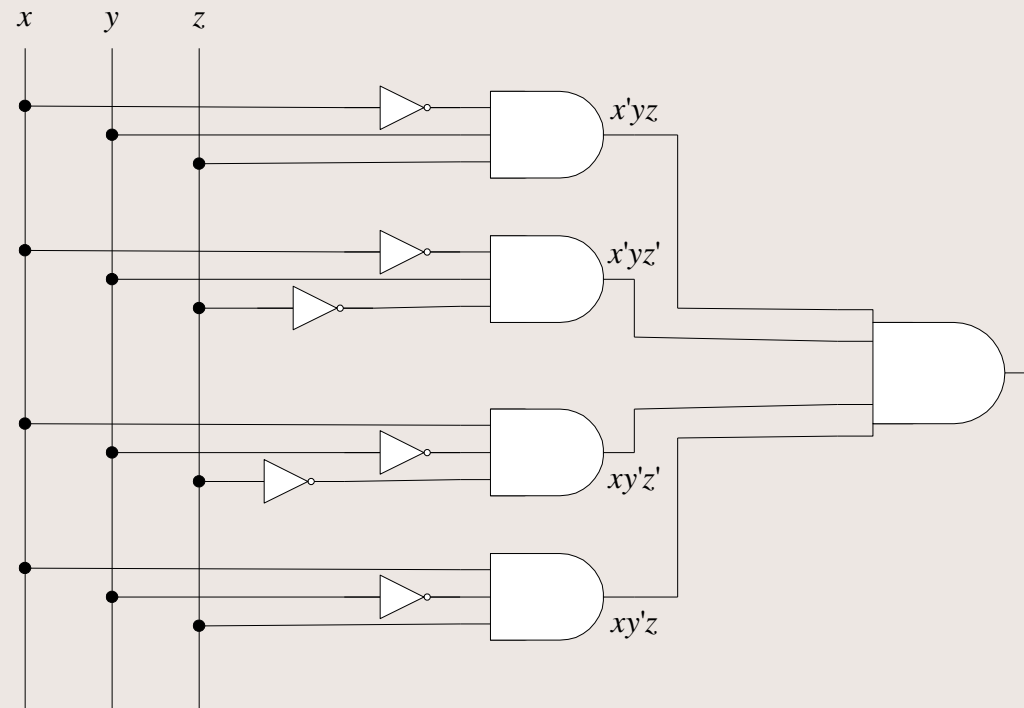
Jawab: Peta Karnaugh dari fungsi tersebut adalah:

	<i>cd</i>			
	00	01	11	10
<i>ab</i> 00	1	0	1	0
01	1	1	1	0
11	X	X	X	X
10	X	0	X	X

Hasil penyederhanaan: $f(a, b, c, d) = bd + c'd' + cd$

Contoh 5.26. Minimisasi fungsi Boolean $f(x, y, z) = x'yz + x'yz' + xy'z' + xy'z$. Gambarkan rangkaian logikanya.

Jawab: Rangkaian logika fungsi $f(x, y, z)$ sebelum diminimisasikan adalah seperti di bawah ini:



Minimisasi dengan Peta Karnaugh adalah sebagai berikut:

		yz			
		00	01	11	10
x	0	0	0	1	1
	1	1	1	0	0

Hasil minimisasi adalah $f(x, y, z) = x'y + xy'$.

Contoh 5.28. Berbagai sistem digital menggunakan kode *binary coded decimal* (BCD). Diberikan Tabel 5.19 untuk konversi BCD ke kode *Excess-3* sebagai berikut:

Tabel 5.19

	Masukan BCD				Keluaran kode <i>Excess-3</i>			
	w	x	y	z	$f_1(w, x, y, z)$	$f_2(w, x, y, z)$	$f_3(w, x, y, z)$	$f_4(w, x, y, z)$
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

(a) $f_1(w, x, y, z)$

		yz			
		00	01	11	10
wx	00				
	01		1	1	1
	11	X	X	X	X
	10	1	1	X	X

$$f_1(w, x, y, z) = w + xz + xy = w + x(y + z)$$

(b) $f_2(w, x, y, z)$

		yz			
		00	01	11	10
wx	00		1	1	1
	01	1			
	11	X	X	X	X
	10		1	X	X

$$f_2(w, x, y, z) = xy'z' + x'z + x'y = xy'z' + x'(y + z)$$

(c) $f_3(w, x, y, z)$

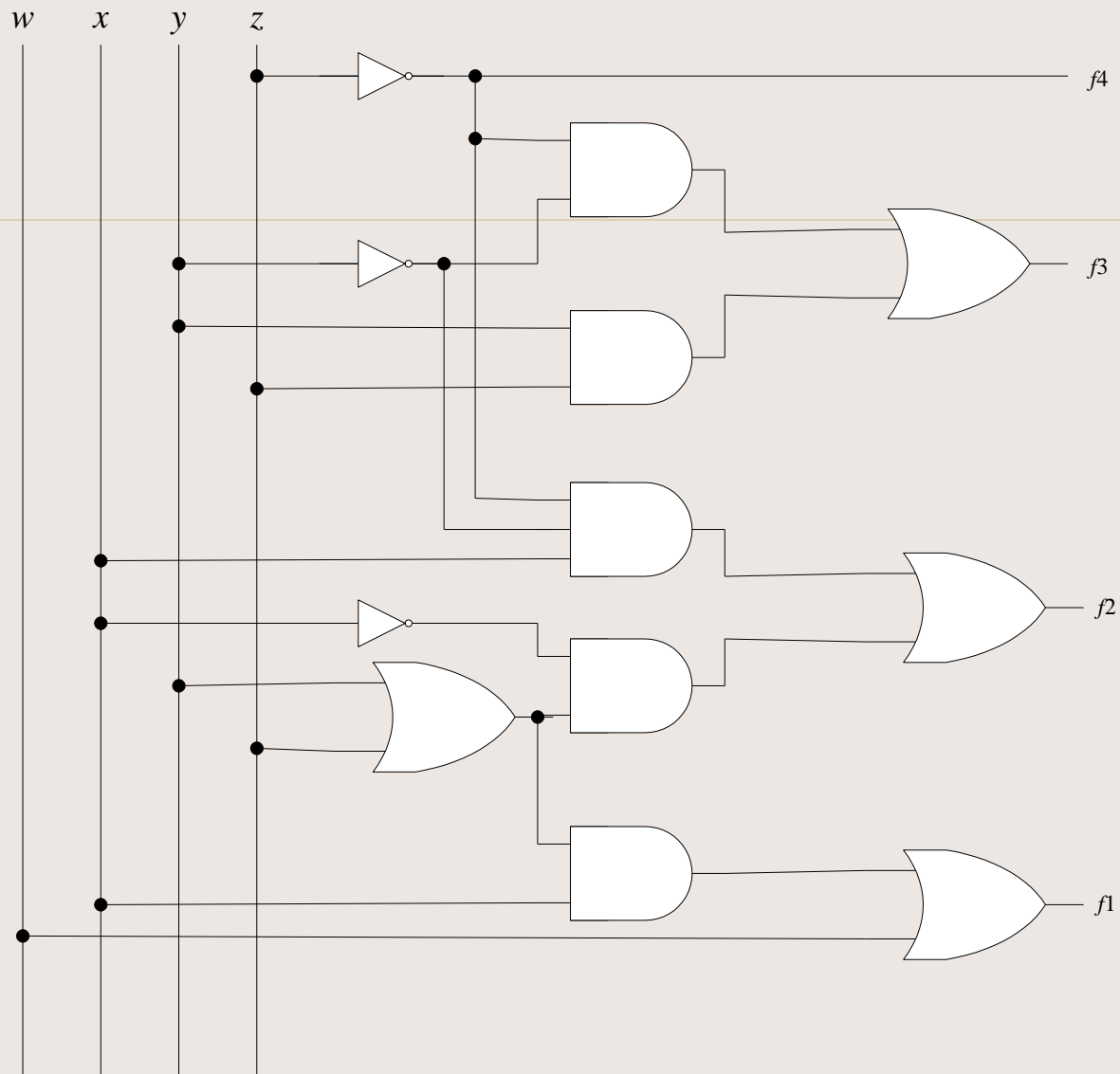
		yz			
		00	01	11	10
wx	00	1		1	
	01	1		1	
	11	X	X	X	X
	10	1		X	X

$$f_3(w, x, y, z) = y'z' + yz$$

(d) $f_4(w, x, y, z)$

		yz			
		00	01	11	10
wx	00	1			1
	01	1			1
	11	X	X	X	X
	10	1		X	X

$$f_4(w, x, y, z) = z'$$



Contoh 7.43

Minimisasi fungsi Boolean berikut (hasil penyederhanaan dalam bentuk baku SOP dan bentuk baku POS):

$$f(w, x, y, z) = \Sigma (1, 3, 7, 11, 15)$$

dengan kondisi *don't care* adalah $d(w, x, y, z) = \Sigma (0, 2, 5)$

Penyelesaian:

Peta Karnaugh dari fungsi tersebut adalah:

wx \ yz	00	01	11	10
	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

Hasil penyederhanaan dalam bentuk SOP

$$f(w, x, y, z) = yz + w'z \quad (\text{SOP}) \quad (\text{garis penuh})$$

dan bentuk baku POS adalah

$$f(w, x, y, z) = z (w' + y) \quad (\text{POS}) \quad (\text{garis putus2})$$

Metode Quine-McCluskey

- Metode Peat Karnaugh tidak mangkus untuk jumlah peubah > 6 (ukuran peta semakin besar).
- Metode peta Karnaugh lebih sulit diprogram dengan komputer karena diperlukan pengamatan visual untuk mengidentifikasi *minterm-minterm* yang akan dikelompokkan.
- Metode alternatif adalah metode Quine-McCluskey . Metode ini mudah diprogram.

Contoh 7.46

Sederhanakan fungsi Boolean $f(w, x, y, z) = \Sigma (0, 1, 2, 8, 10, 11, 14, 15)$.

Penyelesaian:

(i) Langkah 1 sampai 5:

(a)		(b)		(c)	
<i>term</i>	<i>w x y z</i>	<i>term</i>	<i>w x y z</i>	<i>term</i>	<i>w x y z</i>
0	0 0 0 0 \checkmark	0,1	0 0 0 -	0,2,8,10	- 0 - 0
1	0 0 0 1 \checkmark	0,2	0 0 - 0 \checkmark	0,8,2,10	- 0 - 0
2	0 0 1 0 \checkmark	0,8	- 0 0 0 \checkmark		
8	1 0 0 0 \checkmark			10,11,14,15	1 - 1 -
10	1 0 1 0 \checkmark	2,10	- 0 1 0 \checkmark	10,14,11,15	1 - 1 -
11	1 0 1 1 \checkmark	8,10	1 0 - 0 \checkmark		
14	1 1 1 0 \checkmark	10,11	1 0 1 - \checkmark		
15	1 1 1 1 \checkmark	10,14	1 - 1 0 \checkmark		
		11,15	1 - 1 1 \checkmark		
		14,15	1 1 1 - \checkmark		

(i) Langkah 6 dan 7:

		<i>minterm</i>							
Bentuk prima		0	1	2	8	10	11	14	15
✓ 0,1		×	×						
✓ 0,2,8,10		×		×	×	×			
✓ 10,11,14,15						×	×	×	×
			*	*	*		*	*	*
		✓	✓	✓	✓	✓	✓	✓	✓

Bentuk prima yang terpilih adalah:

0,1 yang bersesuaian dengan *term* $w'x'y$
 0, 2, 8, 10 yang bersesuaian dengan *term* $x'z'$
 10, 11, 14, 15 yang bersesuaian dengan *term* wy

Semua bentuk prima di atas sudah mencakup semua *minterm* dari fungsi Boolean semula. Dengan demikian, fungsi Boolean hasil penyederhanaan adalah $f(w, x, y, z) = w'x'y' + x'z' + wy$.

Contoh 7.47

Sederhanakan fungsi Boolean $f(w, x, y, z) = \Sigma (1,4,6,7,8,9,10,11,15)$

Penyelesaian:

(i) Langkah 1 sampai 5:

(a)					(b)					(c)				
<i>term</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>	<i>term</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>	<i>term</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
1	0	0	0	1	√	1,9	-	0	0	1	8,9,10,11	1	0	-
4	0	1	0	0	√	4,6	0	1	-	0	8,10,9,11	1	0	-
8	1	0	0	0	√	8,9	1	0	0	-	√			
						8,10	1	0	-	0	√			
6	0	1	1	0	√									
9	1	0	0	1	√	6,7	0	1	1	-				
10	1	0	1	0	√	9,11	1	0	-	1	√			
						10,11	1	0	1	-	√			
7	0	1	1	1	√									
11	1	0	1	1	√	7,15	-	1	1	1				
						11,15	1	-	1	1				
15	1	1	1	1	√									

(i) Langkah 6 dan 7

		<i>minterm</i>								
Bentuk prima		1	4	6	7	8	9	10	11	15
✓ 1,9		×					×			
✓ 4,6			×	×						
6,7				×	×					
7,15					×					×
11,15									×	×
✓ 8,9,10,11						×	×	×	×	
		*	*			*		*		
		✓	✓	✓		✓	✓	✓	✓	

Sampai tahap ini, masih ada dua *minterm* yang belum tercakup dalam bentuk prima terpilih, yaitu 7 dan 15. Bentuk prima yang tersisa (tidak terpilih) adalah (6,7), (7,15), dan (11, 15). Dari ketiga kandidat ini, kita pilih bentuk prima (7,15) karena bentuk prima ini mencakup *minterm* 7 dan 15 sekaligus.

		<i>minterm</i>								
Bentuk prima		1	4	6	7	8	9	10	11	15
√ 1,9		×					×			
√ 4,6			×	×						
6,7				×	×					
√ 7,15					×					×
11,15									×	×
√ 8,9,10,11						×	×	×	×	
		*	*			*		*		
		√	√	√	√	√	√	√	√	√

Sekarang, semua *minterm* sudah tercakup dalam bentuk prima terpilih. Bentuk prima yang terpilih adalah:

1,9	yang bersesuaian dengan <i>term</i>	$x'y'z$
4,6	yang bersesuaian dengan <i>term</i>	$w'xz'$
7,15	yang bersesuaian dengan <i>term</i>	xyz
8,9,10,11	yang bersesuaian dengan <i>term</i>	wx'

Dengan demikian, fungsi Boolean hasil penyederhanaan adalah $f(w, x, y, z) = x'y'z + w'xz' + xyz + wx'$.

Latihan soal

1. Implementasikan fungsi $f(x, y, z) = \Sigma (0, 6)$ dan hanya dengan gerbang NAND saja.
2. Gunakan Peta Karnaugh untuk merancang rangkaian logika yang dapat menentukan apakah sebuah angka desimal yang direpresentasikan dalam bit biner merupakan bilangan genap atau bukan (yaitu, memberikan nilai 1 jika genap dan 0 jika tidak).

3. Sebuah instruksi dalam sebuah program adalah

```
if   A   >   B   then   writeln (A)   else  
writeln (B) ;
```

Nilai A dan B yang dibandingkan masing-masing panjangnya dua bit (misalkan a_1a_2 dan b_1b_2).

- (a) Buatlah rangkaian logika (yang sudah disederhanakan tentunya) yang menghasilkan keluaran 1 jika $A > B$ atau 0 jika tidak.
- (b) Gambarkan kembali rangkaian logikanya jika hanya menggunakan gerbang *NAND* saja (petunjuk: gunakan hukum de Morgan)

5. Buatlah rangkaian logika yang menerima masukan dua-bit dan menghasilkan keluaran berupa kudrat dari masukan. Sebagai contoh, jika masukannya 11 (3 dalam sistem desimal), maka keluarannya adalah 1001 (9 dalam sistem desimal).