ALGORITMA BOOLEAN

IF1220 Matematika Diskrit

Narendra Dharma Wistara Marpaung

NIM 13524044

Semester 2 Tahun 2025

Contents

Referensi

1. Rinaldi Munir. (2025). Homepage Rinaldi Munir. https://informatika.stei.itb.ac.id/ rinaldi.munir/Matdis/matdis.htm
   **Salindia bahan pelajaran terdapat di tautan tersebut.*

2. Kenneth H. Rosen. (2019). *Discrete mathematics and its applications (Eight Edition)*. McGraw-Hill.

1 Pendahuluan

Aljabar Boolean ditemukan oleh George Boole pada tahun 1854. Boole melihat bahwa himpunan dan logika proposisional mempunyai sifat-sifat yang serupa. Dalam buku *The Laws of Thought*, Boole memaparkan aturan-aturan dasar logika yang nantinya membentuk struktur matematika yang disebut **Aljabar Boolean**.

Definisi 1.1 – Aljabar Boolean

Misalkan B adalah himpunan yang didefinisikan pada dua operator biner, $+$ dan $\cdot$, serta satu opereator uner, $'$. Misal 0 dan 1 adalah dua elemen yang berbeda dalam 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) $a \cdot b = b \cdot a$
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
   (a) $a + a' = 1$
   (b) $a \cdot a' = 0$

Elemen-elemen dari himpunan B tidak didefinisikan nilainya dan bebas ditentukan anggota-anggotanya, sehingga terdapat banyak sekali aljabar Boolean.

Aljabar himpunan dan aljabar logika proposisi juga merupakan aljabar Boolean karena memenuhi aksioma di atas. Dengan kata lain, aljabar himpunan dan aljabar logika proposisi adalah subset dari aljabar Boolean.

Catatan: Penulisan $x \cdot y$ dapat pula ditulis dalam bentuk xy. Hal ini berlaku juga dengan variabel lainnya (dapat pula lebih dari dua variabel, misal xyz atau $pqrs$)

2 Aljabar Boolean 2–Nilai

Aljabar Boolean 2–nilai merupakan aljabar yang paling populer dikarenakan aplikasinya yang luas. Aljabar Boolean 2–nilai memiliki:

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

ii) operator biner $+$ dan $\cdot$, serta operator uner $'$

iii) kaidah untuk operator biner dan uner sebagai berikut:

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

Table 1: Operator $+$

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

Table 2: Operator $\cdot$

a	a'
0	1
1	0

Table 3: Operator $'$

iv) Keempat aksioma aljabar Boolean terpenuhi.

3 Ekspresi Boolean

Ekspresi Boolean dibentuk dari elemen-elemen B dan/atau peubah-peubah (variabel) yang dapat dikombinasikan satu sama lain dengan operator $+$, $\cdot$, dan $'$. Contoh: $0, 1, a, b, a + b, a \cdot b, a' \cdot (b + c)$, dan lain-lain

4 Hukum–Hukum Aljabar Boolean

Terdapat beberapa hukum-hukum yang berlaku dalam aljabar Boolean. Hukum-hukum ini dapat digunakan untuk menyederhanakan ekspresi Boolean. Hukum-hukum tersebut adalah sebagai berikut:

1. Hukum Identitas

   i) $a + 0 = 0$

   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) $a \cdot a' = 0$
4. Hukum Dominansi
   i) $a \cdot 0 = 0$
   ii) $a + 1 = 1$
5. Hukum Involusi
   i) $(a')' = a$
6. Hukum Penyerapan
   i) $a + a \cdot b = a$
   ii) $a \cdot (a + b) = a$
7. Hukum Komutatif
   i) $a + b = b + a$
   ii) $a \cdot b = b \cdot a$
8. Hukum Asosiatif
   i) $a + (b + c) = (a + b) + c$
   ii) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
9. Hukum Distributif
   i) $a + (b \cdot c) = (a + b) \cdot (a + c)$
   ii) $aa \cdot (b + c) = (a \cdot b) + (a \cdot c)$
10. Hukum De Morgan
   i) $(a + b)' = a' \cdot b'$
   ii) $(a \cdot b)' = a' + b'$
11. Hukum 0/1
   i) $0' = 1$
   ii) $1' = 0$

Contoh 4.1

Buktikan bahwa untuk sembarang elemen a dan b dari aljabar Boolean, maka kesamaan berikut:

1. $a + a' \cdot b = a + b$
2. $a \cdot (a' + b) = a \cdot b$

Penyelesaian:

1. Soal 1

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

2. Soal 2

$$
\begin{aligned}
a \cdot (a' + b) &= a \cdot a' + a \cdot b && \text{Hukum Distributif} \\
&= 0 + a \cdot b && \text{Hukum Komplemen} \\
&= a \cdot b && \text{Hukum Identitas}
\end{aligned}
$$

5 Fungsi Boolean

Contoh 5.1 – Contoh Fungsi Boolean

- $f(x) = x$
- $f(x, y) = x' \cdot y + x \cdot y' + y'$
- $f(x, y, z) = x' \cdot y' \cdot z'$

Setiap peubah (variabel) di dalam fungsi Boolean—termasuk dalam bentuk komplemennya—disebut lateral. Misal: fungsi $h(x, y, z) = x \cdot y \cdot z'$ terdiri dari 3 buah literal, yaitu x, y, z'.

6 Bentuk Kanonik

Bentuk kanonik merupakan ekspresi boolean yang menspesifikan suatu fungsi dapat disajikan dalam dua bentuk berbeda, yaitu sebagai penjumlahan dari hasil kali dan perkalian dari hasil jumlah. Misalnya:

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

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

adalah dua buah fungsi yang sama.

Suku (*term*) di dalam ekspresi boolean yang mengandung literal yang lengkap dalam bentuk lengkap dalam bentuk hasil kali disebut ***Minterm***. Sedangkan, suku (*term*) di dalam ekspresi boolean yang mengandung literal yang lengkap dalam bentuk hasil jumlah disebut ***Maxterm***.

Contoh 6.1

$f(x, y, z) = x' \cdot y' \cdot z + x \cdot y' \cdot z' + x \cdot y \cdot z$ memiliki 3 buah minterm, yaitu $x'y'z$, $xy'z'$, dan xyz.

$g(x, y, z) = (x+y+z)\cdot(x+y'+z)\cdot(x+y'+z')\cdot(x'+y+z')\cdot(x'+y'+z)$ memiliki 5 buah maxterm, yaitu $(x+y+z)$, $(x+y'+z')$, $(x+y'+z')$, (x', y, z'), (x', y', z).

Misal, fungsi dengan variabel x, y, z, maka $x'y$ dan $y'z'$ bukan minterm karena literal tidak lengkap. Namum, xyz' merupakan minterm karena literalnya lengkap. $x + z$ bukan maxterm karena literal tidak lengkap. Sedangkan, $x' + y + z'$ merupakan maxterm karena literalnya lengkap. Namun, $xy' + y' + z$ bukan maxterm.

Jadi, bentuk kanonik adalah ekspresi boolean yang dinyatakan sebagai penjumlahan dari satu atau lebih minterm atau perkalian dari satu atau lebih maxterm. Ada dua macam bentuk kanonik, yaitu penjumlahan dari hasil perkalian (*sum-of-product* atau SOP) dan perkalian dari hasil jumlah (*product-of-sum* atau POS). Fungsi f(x, y, z) pada **Contoh 6.1** dikatakan dalam bentuk SOP dan fungsi g(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.
- Untuk maxterm, setiap peubah yang bernilai 0 dinyatakan tanpa komplemen, sedangkan peubah yang bernilai 1 dinyatakan dalam bentuk komplemen.

Berikut tabel kebenaran untuk dua peubah dan tiga peubah.

x	y	Suku Minterm	Lambang	Suku Maxterm	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

Table 4: Tabel Minterm dan Maxterm untuk dua peubah

x	y	z	Suku Minterm	Lambang	Suku Maxterm	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	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

Table 5: Tabel Minterm dan Maxterm untuk tiga peubah

Jika diberikan sebuah tabel kebenaran, kita dapat membentuk fungsi boolean dalam bentuk kanonik dengan cara mengambil minterm dari setiap fungsi yang bernilai 1 untuk SOP dan mengambil maxterm dari setiap fungsi yang bernilai 0.

Contoh 6.2

Tinjau dalam 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 dalam lambang minterm, $f(x, y, z) = m_1 + m_4 + m_7 = \Sigma(1, 4, 7)$

- **POS**

  Kombinasi nilai-nilai peubah yang menghasilkan nilai fungsi sama dengan 0 adalah 000, 010, 011, 101, 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 lambang maxterm, $f(x, y, z) = M_0 + M_2 + M_3 + M_5 + M_6 = \Pi(0, 2, 3, 5, 6)$

Contoh 6.3

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

Penyelesaian:

- **SOP**

$$
\begin{aligned}
f(x,y,z) &= x + y'z \\
&= x(y+y')(z+z') + y'z \\
&= (xy + xy')(z+z') + (x+x')y'z \\
&= xy(z+z') + xy'(z+z') + xy'z + x'y'z \\
&= xyz + xyz' + xy'z + xy'z' + xy'z + x'y'z \\
&= xyz + xyz' + xy'z + xy'z' + x'y'z \\
f(x,y,z) &= m_7 + m_6 + m_5 + m_4 + m_1 = \Sigma(1,4,5,6,7)
\end{aligned}
$$

- **POS**

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

7 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 fungsi 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

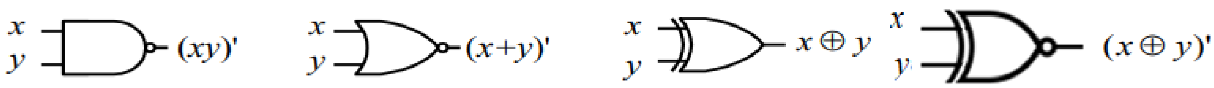
$$\begin{aligned}
f(x,y,z) = (f'(x,y,z))' &= (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' \\
&= (x'y'z')' \cdot (x'yz')' \cdot (x'yz)' \\
&= (x+y+z)(x+y'+z)(x+y'+z') \\
&= M_0 M_2 M_3 = \Pi(0,2,3)
\end{aligned}$$

Kesimpulannya $m_j' = M_j$ dan $M_j' = m_j$ untuk semua j.

8 Rangkaian Logika

Fungsi Boolean dapat juga direpresentasikan dalam bentuk rangkaian logika. Ada tiga gerbang logika dasar, yaitu gerbang logika AND, OR, dan NOT (gambar berturut-turut dari kiri ke kanan).

Adapula gerbang logika turunan yang merupakabn kombinasi dari gerbang logika dasar di atas, yaitu NAND, NOR, XOR, NXOR (gambar berturut-turut dari kiri ke kanan).



9 Penyederhanaan Fungsi Boolean

Menyederhanakan fungsi Boolean artinya mencari bentuk fungsi lain yang ekivalen tetapi dengan jumlah literal atau operasi yang lebih sedikit. Dipandang dari segi aplikasi aljabar Boolean, fungsi Boolean yang lebih sederhana berarti gerbang logika yang lebih sedikit. Tiga metode yang dapat digunakan untuk menyederhanakan fungsi Boolean adalah:

1. Secara aljabar, menggunakan hukum aljabar Boolean
2. metode Peta Karnaugh
3. Metode Quine-McCluskey (tabulasi)

9.1 Peta Karnaugh

Peta Karnaugh (atau K-Map) merupakan metode grafis untuk menyederhanakan fungsi Boolean. Metode ini ditemukan oleh Maurice Karnaugh pada tahun 1953. Peta Karnaugh adalah sebuah diagram/peta yang terbentuk dari kotak-kotak yang bersisian Tiap kotak merepresentasikan sebuah minterm. Tiap kotak dikatakan bertetanggaan jika minterm-minterm yang merepresentasikan berbeda hanya 1 buah literal. Cara mengisi peta karnaugh adalah

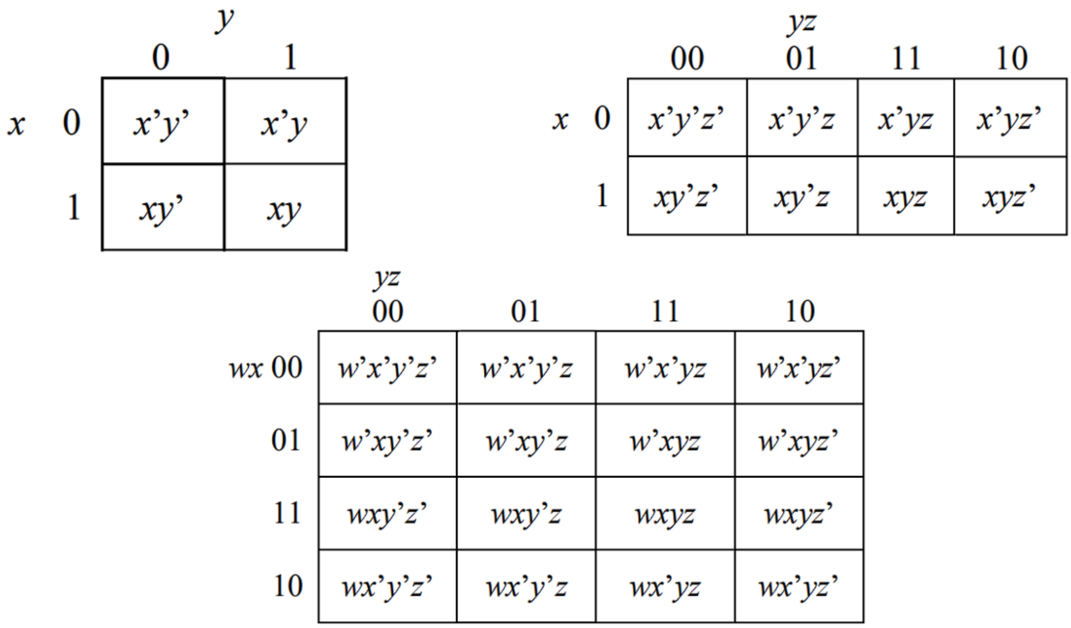


Figure 1: Peta Karnaugh untuk 2, 3, dan 4 peubah

kotak yang menyatakan minterm diisi angka 1 (apabila terdapat lebih dari 1 variabel dalam minterm, maka cari irisannya), sisanya diisi angka 0.

Contoh 9.1

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

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

$xz' + y$

$x \rightarrow$ semua kotak pada baris ke-2

$z' \rightarrow$ semua kotak pada kolom ke-1 dan ke-4

$xz' \rightarrow$ irisan dari keduanya $y \rightarrow$ semua kotak pada kolom ke-3 dan ke-4

Penggunaan Peta Karnaugh dalam penyederhanaan fungsi Boolean dilakukan dengan cara menggabungkan kotak-kotak yang bernilai 1 dan saling bersisian. Kelompok kotak yang bernilai 1 dapat membentuk pasangan(dua), kuad(empat), atau oktet(delapan). Untuk menyederhanakan fungsi Boolean, lihat kelompok kotak yang bernilai 1, lalu cari kesamaannya. Pada kesamaan tersebut, coret literal yang berbeda. Literal yang tersisa adalah fungsi Boolean yang lebih sederhana.

Contoh 9.2

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

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

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

Penyelesaian:

Pada isi tabel yang di bold memiliki nilai 1, maka kita dapat mengelompokkan kotak-kotak tersebut menjadi oktet (delapan). Kita coret literal yang berbeda, sehingga kita dapat literal yang sama hanyalah w (sama-sama memiliki nilai 1). Sehingga, fungsi Boolean yang lebih sederhana adalah $f(w,x,y,z) = w$.

Penyederhanaan juga dapat dilakukan dengan penggulungan, yaitu menautkan sisi kanan peta karnaugh dengan sisi kirinya dan sisi atas dengan sisi bawahnya layaknya digulung (2 gulungan). Sehingga, kolom paling kiri dengan kolom paling kanan dan kolom paling atas dengan kolom paling bawah bersentuhan dan dapat dikelompokkan.

Catatan: *Terdapat kasus unik, jika masing-masing ujung (corner) bernilai 1, maka masing-masing ujung dari kotak juga dapat bersentuhan dan dikelompokkan. (Lihat* ***Contoh minimisasi 6 dan 10*** *halaman 27 dan 30, salindia Aljabar Boolean Bagian 2 pada Referensi 1).*

Untuk pemahaman yang lebih jelas terkait penyederhanaan Aljabar Boolean, silakan pelajari dan lihat kembali salindia Aljabar Boolean Bagian 2 pada Referensi 1.

Sangat disarankan untuk mengelompokkan nilai 1 yang bertetanggaan sebanyak mungkin, mulai dari mencari oktet, lalu kuad, dan terakhir pasangan.

Peta Karnaugh ini juga dapat digunakan untuk merepresentasikan maxterm (atau POS). Namun, untuk penyederhanaannya, kita mengelompokkan kotak-kotak yang bernilai 0.

9.2 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}

Figure 2: Peta Karnaugh untuk 5 Peubah

Dalam Peta Karnaugh lima peubah, dua kotak dianggap bertetanggaan jika secara fisik bersebalahan atau merupakan pencerminan terhadap garis ganda

Contoh 9.3

Carilah fungsi sederhana dari

$$f(v, w, x, y, z) = \Sigma(0, 2, 4, 6, 9, 11, 13, 15, 17, 21, 25, 27, 29, 31)$$

Peta karnaugh dari fungsi tersebut adalah sebagai berikut:

vw/xyz	000	001	011	010	110	111	101	100
00	**1**	0	0	**1**	**1**	0	0	**1**
01	0	**1**	**1**	0	0	**1**	**1**	0
11	0	**1**	**1**	0	0	**1**	**1**	0
10	0	**1**	0	0	0	0	**1**	0

$f(v, w, x, y, z)$

**Garis ganda pada tabel adalah garis pencerminan*

Penyederhanaan dari fungsi tersebut adalah $f(v, w, x, z) = wz + v'w'z' + vy'z$

Keadaan *Don't Care* adalah kondisi nilai peubah yang tidak diperhitungkan oleh fungsinya. Artinya nilai 1 atau 0 dari peubah *don't care* tidak berpengaruh pada hasil fungsi tersebut. Misalnya, peraga digital angka desimal 0 sampai 9, sehingga jumlah angka bit yang diperlukan untuk merepresentasikan angka 0 sampai 9 (membutuhkan 4 bit). Sehingga, bit-bit untuk angka 10–15 tidak terpakai.

w	x	y	z	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	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

Desimal yang ditandai oleh X adalah keadaan *don't care*

Dalam menyederhanakan Peta Karnaugh yang mengandung keadaan *don't care*, ada hal yang perlu diperhatikan sebagai berikut:

1. Anggap semua nilai *don't care* (tanda X) dapat dibuat kelompok bersama angka 1.
2. Buatlah kelompok sebesar mungkin yang melibatkan angka 1 dan tanda X tersebut.
3. Semua nilai X yang tidak termasuk dalam kelompok tersebut dianggap bernilai 0.

Dengan mengikuti cara ini, keadaan X telah dimanfaatkan semaksimal mungkin. Hal ini dapat dilakukan secara bebas.

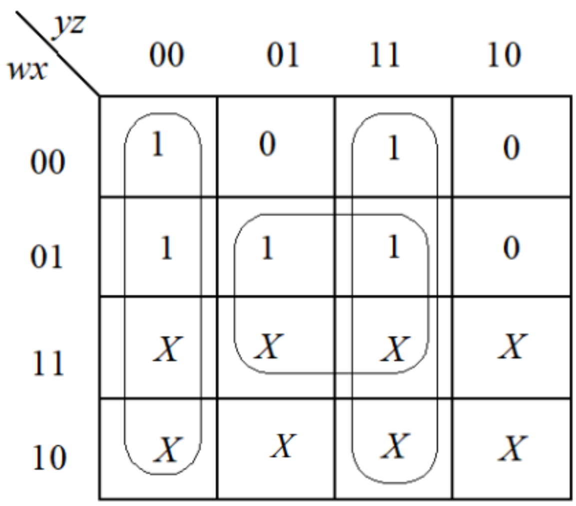
Contoh 9.4

Sebuah fungsi Boolean, f dinyatakan dalam tabel berikut.

Minimisasi fungsi f sederhana mungkin.

w	x	y	z	f(w,x,y,z)
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

Penyelesaian:



Hasil penyederhanaan dari fungsi tersebut adalah $f(w, x, y, z) = xz + y'z' + yz$