BAB 4
PETA KARNAUGH
(KARNAUGH MAPS)

4.1. Peta Karnaugh

Selain dengan teorema boole, salah satu cara untuk memanipulasi dan menyederhanakan fungsi boole adalah dengan teknik peta karnaugh.

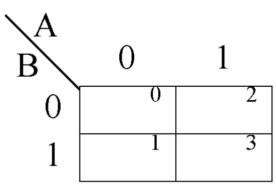
Peta karnaugh merupakan sekumpulan kotak-kotak yang diberi nama sedemikian rupa berdasarkan nama variabelnya dan diletakkan sedemikian rupa pula sehingga dapat mengeliminasi beberapa tabel jika kotak itu digabung. Jumlah kotak tergantung banyaknya variabel input. Jika ada sebanyak n input maka ada 2^n kombinasi input, maka sebanyak itu pula kotak yang dibutuhkan.

Dalam peta karnaugh dikenal istilah tetangga dekat. Yang dimaksud dengan tetangga dekat adalah kotak-kotak yang memiliki satu atau lebih variabel yang sama atau kotak-kotak yang terletak dalam satu atau lebih bidang yang sama.

Yang dimaksud dengan bidang adalah sekumpulan kotak yang sudah diberi nama berdasarkan variabel inputnya.

<u>Peta Karnaugh untuk 2 Variabel (A, B)</u>

Untuk 2 variabel input akan ada sebanyak 2^2 = 4 kombinasi input
- Maka banyaknya kotak yang dibutuhkan adalah 4 kotak.
- Keempat kotak itu diatur sebagai berikut :



Penggabungan kotak-kotak untuk 2 variabel (A, B)

- Jika ada 2 kotak yang ditandai 1 bertetangga dekat dapat digabung, akan menyatakan 1 variabel tunggal.
- Untuk 1 kotak yang ditandai 1 dan tidak memiliki tetangga dekat, akan menyatakan 2 variabel.

Contoh :

$y = \overline{A}\,B + A\,\overline{B}$

B \ A	0	1
0	^{0}	1 ^{2}
1	1 ^{1}	^{3}

Menyatakan 1 tetangga

$y = \overline{A}\,B + A\,B$

B \ A	0	1
0		
1	1	1

Menyatakan 2 tetangga sehingga dapat disederhanakan menjadi
$y = B$

$y = \overline{A}\,\overline{B} + A\,\overline{B} + A\,B$

B \ A	0	1
0	1	1
1		1

Menyatakan 2 buah 2 tetangga sehingga dapat disederhanakan menjadi
$y = A + \overline{B}$

LATIHAN - 1

1. Tentukan fungsi boole yang paling sederhana dari peta karnaugh berikut ini:

   a)

B \ A	0	1
0	1	
1	1	

   b)

B \ A	0	1
0	1	
1		1

<u>Peta Karnaugh untuk 3 Variabel (A, B, C)</u>

Untuk 3 variabel input akan ada sebanyak $2^3 = 8$ kombinasi input
- Maka banyaknya kotak yang dibutuhkan adalah 8 kotak.
- Kedelapan kotak itu diatur (ada 2 cara) sebagai berikut :

C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

BC \ A	0	1
00	0	4
01	1	5
11	3	7
10	2	6

Penggabungan kotak-kotak untuk 3 variabel (A, B, C)
- 4 kotak yang bertetangga dekat dapat digabung dan menyatakan 1 variabel tunggal.
- 2 kotak yang bertetangga dekat dapat digabung dan menyatakan 2 variabel.
- 1 kotak yang tidak bertetangga dekat akan menyatakan 3 variabel

Contoh :

$y = A\,B\,\overline{C} + \overline{A}\,B\,C + A\,B\,C + A\,\overline{B}\,C$

C \ AB	00	01	11	10
0			1	
1		1	1	1

Menyatakan 3 buah 2 tetangga sehingga dapat disederhanakan menjadi

$y = AB + BC + AC$

$y = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + A\,B\,C + A\,\overline{B}\,C$

C \ AB	00	01	11	10
0				
1	1	1	1	1

Menyatakan 4 tetangga sehingga dapat disederhanakan menjadi

$y = C$

$y = \overline{A}\,B\,\overline{C} + \overline{A}\,B\,C + A\,B\,\overline{C} + A\,B\,C$

C \ AB	00	01	11	10
0		1	1	
1		1	1	

Menyatakan 4 tetangga sehingga dapat disederhanakan menjadi

$y = B$

LATIHAN - 2

1. Tentukan fungsi boole yang paling sederhana dari peta karnaugh berikut ini:

a)

C \ AB	00	01	11	10
0	1		1	1
1	1		1	1

b)

C \ AB	00	01	11	10
0	1		1	1
1		1	1	

c)

C \ AB	00	01	11	10
0	1			
1	1	1		1

d)

C \ AB	00	01	11	10
0			1	1
1		1	1	1

Peta Karnaugh untuk 4 Variabel (A, B, C, D)

Untuk 4 variabel input akan ada sebanyak $2^4 = 16$ kombinasi input

- Maka banyaknya kotak yang dibutuhkan adalah 16 kotak.
- Keenambelas kotak itu diatur sebagai berikut :

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Penggabungan kotak-kotak untuk 4 variabel (A, B, C, D)

- 8 kotak yang bertetangga dekat dapat digabung dan menyatakan 1 variabel tunggal.
- 4 kotak yang bertetangga dekat dapat digabung dan menyatakan 2 variabel tunggal.
- 2 kotak yang bertetangga dekat dapat digabung dan menyatakan 3 variabel.
- 1 kotak yang tidak bertetangga dekat akan menyatakan 4 variabel

Contoh :

$y = A\,B\,C\,D + A\,B\,C\,\overline{D} + A\,\overline{B}\,C\,D + A\,\overline{B}\,C\,\overline{D}$

CD \ AB	00	01	11	10
00				
01				
11			1	1
10			1	1

Menyatakan 4 tetangga sehingga dapat disederhanakan menjadi
$y = A\,C$

$y = \overline{A}\ \overline{B}\ \overline{C}\ \overline{D} + A\ \overline{B}\ \overline{C}\ \overline{D} + \overline{A}\ \overline{B}\ C\ \overline{D} + A\ \overline{B}\ C\ \overline{D}$

CD \ AB	00	01	11	10
00	1			1
01				
11				
10	1			1

Menyatakan 4 tetangga sehingga dapat disederhanakan menjadi
$y = \overline{B}\ \overline{D}$

$y = \overline{A}\ B + A\ B\ \overline{C} + A\ B\ C\ D + A\ B\ C\ \overline{D}$

CD \ AB	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10		1	1	

Menyatakan 8 tetangga sehingga dapat disederhanakan menjadi
$y = B$

LATIHAN - 3

1. Tentukan fungsi boole yang paling sederhana dari peta karnaugh berikut ini:

   a) b)

CD \ AB	00	01	11	10
00	1			1
01		1		
11				1
10	1			1

Peta Karnaugh untuk 5 Variabel (A, B, C, D, E)

Untuk 5 variabel input akan ada sebanyak 2^5 = 32 kombinasi input

- Maka banyaknya kotak yang dibutuhkan adalah 32 kotak.
- Ketigapuluh dua kotak itu diatur sebagai berikut :

DE \ ABC	000	001	011	010	100	101	111	110	ABC / DE
00	0	4	12	8	16	20	28	24	00
01	1	5	13	9	17	21	29	25	01
11	3	7	15	11	19	23	31	27	11
10	2	6	14	10	18	22	30	26	10

Penggabungan kotak-kotak untuk 5 variabel (A, B, C, D, E)

- 16 kotak yang bertetangga dekat dapat digabung dan menyatakan 1 variabel tunggal.
- 8 kotak yang bertetangga dekat dapat digabung dan menyatakan 2 variabel tunggal.
- 4 kotak yang bertetangga dekat dapat digabung dan menyatakan 3 variabel tunggal.
- 2 kotak yang bertetangga dekat dapat digabung dan menyatakan 4 variabel.
- 1 kotak yang tidak bertetangga dekat akan menyatakan 5 variabel

Contoh :

$$y = \overline{A}\,B\,\overline{D} + A\,B\,\overline{D} + \overline{A}\,\overline{B}\,D + A\,\overline{B}\,D$$

DE \ ABC	000	001	011	010	100	101	111	110	ABC / DE
00			1	1			1	1	00
01			1	1			1	1	01
11	1	1			1	1			11
10	1	1			1	1			10

Menyatakan 2 buah 8 tetangga sehingga dapat disederhanakan menjadi

$$y = B\,\overline{D} + \overline{B}\,D$$

$y = \bar{A}\ B\ C + \bar{A}\ B\ \bar{C} + A\ B\ \bar{D} + A\ B\ D$

DE \ ABC	000	001	011	010	100	101	111	110
00			1	1			1	1
01			1	1			1	1
11			1	1			1	1
10			1	1			1	1

Menyatakan 16 tetangga sehingga dapat disederhanakan menjadi
y = B

LATIHAN - 4

1. Tentukan fungsi boole yang paling sederhana dari peta karnaugh berikut ini:

a)

DE \ ABC	000	001	011	010	100	101	111	110
00		1	1	1		1	1	1
01		1	1	1		1	1	1
11		1	1			1	1	
10		1	1			1	1	

b)

DE \ ABC	000	001	011	010	100	101	111	110
00	1				1			
01	1				1			
11				1				1
10		1	1			1	1	

c)

DE \ ABC	000	001	011	010
00	1			1
01				
11				
10	1	1	1	1

100	101	111	110	ABC / DE
1			1	00
				01
				11
1	1	1	1	10

Peta Karnaugh untuk 6 Variabel (A, B, C, D, E, F)

Untuk 6 variabel input akan ada sebanyak 2^6 = 64 kombinasi input

- Maka banyaknya kotak yang dibutuhkan adalah 64 kotak.
- Keenampuluh empat kotak itu diatur sebagai berikut :

EF \ ABCD	0000	0001	0011	0010
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

0100	0101	0111	0110	ABCD / EF
16	20	28	24	00
17	21	29	25	01
19	23	31	27	11
18	22	30	26	10

00	32	36	44	40
01	33	37	45	41
11	35	39	47	43
10	34	38	46	42
EF / ABCD	1000	1001	1011	1010

48	52	60	56	00
49	53	61	57	01
51	55	63	59	11
50	54	62	58	10
1100	1101	1111	1110	EF / ABCD

Penggabungan kotak-kotak untuk 6 variabel (A, B, C, D, E, F)

- 32 kotak yang bertetangga dekat dapat digabung dan menyatakan 1 variabel tunggal.
- 16 kotak yang bertetangga dekat dapat digabung dan menyatakan 2 variabel tunggal.
- 8 kotak yang bertetangga dekat dapat digabung dan menyatakan 3 variabel tunggal.
- 4 kotak yang bertetangga dekat dapat digabung dan menyatakan 4 variabel tunggal.

- 2 kotak yang bertetangga dekat dapat digabung dan menyatakan 5 variabel.
- 1 kotak yang tidak bertetangga dekat akan menyatakan 6 variabel

<u>*Contoh :*</u>
Sederhanakan fungsi boole berikut ini :
Y (A,B,C,D,E,F) = Σm(0,9,11,24,25,27,34,35,38,39,43,47,51,55,58,59,62,63)

EF \ ABCD	0000	0001	0011	0010	0100	0101	0111	0110	ABCD / EF
00	0	4	12	8	16	20	28	24	00
01	1	5	13	9	17	21	29	25	01
11	3	7	15	11	19	23	31	27	11
10	2	6	14	10	18	22	30	26	10

EF / ABCD	1000	1001	1011	1010	1100	1101	1111	1110	EF \ ABCD
00	32	36	44	40	48	52	60	56	00
01	33	37	45	41	49	53	61	57	01
11	35	39	47	43	51	55	63	59	11
10	34	38	46	42	50	54	62	58	10

4.2. Kelompok Berlebihan

Jika pengelompokan 1 tidak dilakukan secara hati – hati, maka ada kemungkinan kita membuat kelompok 1 yang tidak perlu. Pengelompokan yang berlebihan (redundan) ini menghasilkan fungsi Boolean dengan *term* yang tidak perlu. Hal ini ditunjukkan pada Contoh 1 berikut ini.

Contoh 1

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

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

Penyelesaian :

Jika pengelompokannya adalah seperti Peta Karnaugh di atas, maka fungsi hasil penyederhanaan adalah

$f(w,x,y,z) = w'xz + wxy + xyz \rightarrow$ masih belum disederhanakan

Kita dapat membuat fungsi yang lebih sederhana lagi dengan memperhatikan bahwa kelompok yang ketida (lingkaran horizontal) merupakan kelompok yang berlebihan karena ia meningkatkan jumlah suku (*term*). Kelompok ketiga ini memuat dua buah 1 yang sebenarnya sudah termasuk ke dalam kelompok lain. Kelompok berlebihan dapat dihilangkan dari Peta Karnaugh menjadi :

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

maka fungsi Boolean hasil penyederhanaan adalah :

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

yang ternyata lebih sederhana dibandingkan dengan jawaban pertama karena mengandung jumlah literal dan operasi biner lebih sedikit.

4.3. Ketidakunikan Fungsi Hasil Penyederhanaan

Metode peta Karnaugh menghasilkan fungsi Boolean yang lebih sederhana. Fungsi yang lebih sederhana mempunyai jumlah literal dan jumlah *term* yang lebih sedikit daripada fungsi asalnya. Namun, hasil penyederhanaan dengan peta Karnaugh tidak selalu unik. Artinya, mungkin terdapat beberapa bentuk fungsi minimasi yang berbeda meskipun jumlah literal dan jumlah *term*-nya sama. Hal ini diberikan pada contoh peta Karnaugh di bawah ini.

Kemungkinan pengelompokan I :

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

Fungsi minimasi :

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

Kemungkinan pengelompokan II :

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

Fungsi minimasi :

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

Kemungkinan pengelompokan III :

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

Fungsi minimasi : $f(w,x,y,z) = w'x'y + w'xy'z + wxz' + wxy + x'yz + wy'z'$

4.4. METODE QUINE- McCLUSKEY

Metode peta Karnaugh hanya cocok digunakan jika fungsi Boolean mempunyai jumlah peubah paling banyak 6 buah. Jika jumlah peubah yang terlibat pada suatu fungsi Boolean lebih dari 6 buah maka penggunaan peta Karnaugh menjadi semakin rumit, sebab ukuran peta bertambah besar. Selain itu, metode peta Karnaugh lebih sulit diprogram dengan komputer karena diperlukan pengamatan visual untuk mengidentifikasi *minterm-minterm* yang akan dikelompokkan. Untuk itu diperlukan metode penyederhanaan yang lain yang dapat diprogram dan dapat digunakan untuk fungsi Boolean dengan sembarang jumlah peubah. Metode alternatif tersebut adalah metode Quine-McCluskey yang dikembangkan oleh W.V. Quine dan E.J. McCluskey pada tahun 1950.

Langkah-langkah metode Quine-McCluskey untuk menyederhanakan ekspresi Boolean dalam bentuk SOP adalah sebagai berikut :

1. Nyatakan tiap *minterm* dalam *n* peubah menjadi *string* bit yang panjangnya n, yang dalam hal ini peubah komplemen dinyatakan dengan ‘0’, peubah yang bukan komplemen dengan ‘1’.
2. Kelompokkan tiap *minterm* berdasarkan jumlah ‘1’ yang dimilikinya.
3. Kombinasikan *minterm* dalam *n* peubah dengan kelompok lain yang jumlah ‘1’-nya berbeda satu, sehingga diperoleh bentuk prima (*prime-implicant*) yang terdiri dari *n*-1 peubah. Minterm yang dikombinasikan diberi tanda “√”.
4. Kombinasikan *minterm* dalam *n*-1 peubah dengan kelompok lain yang jumlah ‘1’-nya berbeda satu, sehingga diperoleh bentuk prima yang terdiri dari *n*-2 peubah.
5. Teruskan langkah 4 sampai diperoleh bentuk prima yang sesederhana mungkin.
6. Ambil semua bentuk prima yang tidak bertanda “√”. Buatlah tabel baru yang memperlihatkan *minterm* dari ekspresi Boolean semula yang dicakup oleh bentuk prima tersebut (tandai dengan “×”). Setiap *minterm* harus dicakup oleh paling sedikit satu buah bentuk prima.
7. Pilih bentuk prima yang memiliki jumlah literal paling sedikit namun mencakup sebanyak mungkin *minterm* dari ekspresi Boolean semula. Hal ini dapat dilakukan dengan cara berikut :
   a. Tandai kolom–kolom yang mempunyai satu buah tanda “×” dengan tanda “*”, lalu beri tanda “√” di sebelah kiri bentuk prima yang berasosiasi dengan tanda “*” tersebut. Bentuk prima ini telah dipilih untuk fungsi Boolean sederhana.

b. Untuk setiap bentuk prima yang telah ditandai "√", beri tanda *minterm* yang dicakup oleh bentuk prima tersebut dengan tanda "√".

c. Periksa apakah masih ada *minterm* yang belum dicakup oleh bentuk prima terpilih. Jika ada, pilih dari bentuk prima yang tersisa yang mencakup sebanyak mungkin *minterm* tersebut. Beri tanda "√" bentuk prima yang dipilih itu serta *minterm* yang dicakupnya.

d. Ulangi langkah c sampai seluruh *minterm* sudah dicakup oleh semua bentuk prima.

Aslinya, metode Quine McCluskey digunakan untuk menyederhanakan fungsi Boolean yang ekspresinya dalam bentuk SOP, namun metode ini dapat dimodifikasi sehingga juga dapat digunakan untuk ekspresi dalam bentuk POS.

Contoh 1

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

Penyelesaian :

(i). Langkah 1 sampai 5 :

(a)						(b)						(c)				
term	w	x	y	z		*term*	w	x	y	z		term	w	x	y	z
0	0	0	0	0	√	0,1	0	0	0	-		0,2,8,10	-	0	-	0
						0,2	0	0	-	0	√	0,8,2,10	-	0	-	0
1	0	0	0	1	√	0,8	-	0	0	0	√					
2	0	0	1	0	√							10,11,14,15	1	-	1	-
8	1	0	0	0	√	2,10	-	0	1	0	√	10,14,11,15	1	-	1	-
						8,10	1	0	-	0	√					
10	1	0	1	0	√											
						10,11	1	0	1	-	√					
11	1	0	1	1	√	10,14	1	-	1	0	√					
14	1	1	1	0	√											
						11,15	1	-	1	1	√					
15	1	1	1	1	√	14,15	1	1	1	-	√					

(ii). Langkah 6 dan 7 :

		minterm							
	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 1 di atas kurang begitu bagus dalam memberikan ilustrasi metode Quine McCluskey. Contoh 2 di bawah ini dapat memberikan gambaran metode untuk kasus yang lebih umum.

<u>*Contoh 2*</u>

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

Penyelesaian :

(i). Langkah 1 sampai 5 :

(a)

term	w	x	y	z	
1	0	0	0	1	√
4	0	1	0	0	√
8	1	0	0	0	√
6	0	1	1	0	√
9	1	0	0	1	√
10	1	0	1	0	√
7	0	1	1	1	√
11	1	0	1	1	√
15	1	1	1	1	√

(b)

term	w	x	y	z	
1,9	-	0	0	1	
4,6	0	1	-	0	
8,9	1	0	0	-	√
8,10	1	0	-	0	√
6,7	0	1	1	-	
9,11	1	0	-	1	√
10,11	1	0	1	-	√
7,15	-	1	1	1	
11,15	1	-	1	1	

(c)

term	w	x	y	z
8,9,10,11	1	0	-	-
8,10,9,11	1	0	-	-

(ii). Langkah 6 dan 7 :

		minterm								
	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 kadidat ini, kita pilih bentuk prima (7,15) karena bentuk prima ini mencakup *minterm* 7 dan 15 sekaligus.

	Bentuk Prima	1	4	6	7	8	9	10	11	15
		minterm								
√	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 *term* $x'\,y'\,z$
4,6	yang bersesuaian dengan *term* $w'\,x\,z'$
7,15	yang bersesuaian dengan *term* $x\,y\,z$
8,9,10,11	yang bersesuaian dengan *term* $w\,x'$

Dengan demikian, fungsi Boolean hasil penyederhanaan adalah

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

LATIHAN - 5

1. Dari tabel kebenaran berikut ini, tentukan output Y = 1 (SOP) dan Y = 0 (POS), dan gambarkan gerbang logikanya.

Desimal	A	B	Y
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

2. Dengan input 4 variabel, buatlah tabel kebenaran yang memberikan output 1 jika keempat variabel dikonversikan ke desimal merupakan bilangan yang habis dibagi 4. Tuliskan juga persamaan dari output 0 dan gerbang logikanya.

3. Sederhanakan fungsi boole berikut dengan peta karnaugh :

   $Y = \overline{A}\,C + A\,\overline{B}\,D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,C\,\overline{D}$

4. Tentukan bentuk sederhana dari fungsi Boolean yang merepresentasikan tabel kebenaran (Tabel 1) dalam bentuk SOP dan bentuk POS.

Tabel 1

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

5. **Minimisasi fungsi Boolean** $f(x,y,z) = x'z + x'y + xy'z + yz$ **(K-Map)**
6. **Minimisasi fungsi Boolean** $f(x,y,z) = \sum m(0,2,4,5,6)$ **(K-Map)**
7. **Sederhanakan fungsi Boolean yang bersesuaian dengan peta Karnaugh di bawah ini dalam bentuk SOP.**

yz \ *wx*	00	01	11	10
00				
01			1	
11	1	1	1	1
10		1	1	1

8. Minimisasi fungsi Boolean $f(w,x,y,z) = w'x'y' + x'yz' + w'xyz' + wx'y'$
9. **Minimisasi fungsi Boolean $f(w,x,y,z) = \sum(0,1,2,4,5,6,8,9,12,13,14)$ (Quine McCluskey)**
10. Sederhanakan fungsi Boolean berikut : $f(w,x,y,z) = \sum(0,1,2,5,8,9,10)$ ke dalam bentuk POS.
11. Sederhanakan fungsi
    $f(w,x,y,z) = (w + x')(w + x + y)(w' + x' + y')(w' + x + y + z')$
    dengan menggunakan Peta Karnaugh. Hasil penyederhanaan dalam bentuk SOP dan POS.
12. Sederhanakan fungsi $f(x,y,z,t) = xy' + xyz + x'y'z' + x'yzt'$
13. Minimasi fungsi yang telah dipetakan ke Peta Karnaugh di bawah ini dalam bentuk SOP dan bentuk POS.

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