

# Karnaugh MAP (K-Map)

## *Pokok Bahasan :*

1. K-map 2 variabel
2. K-map 3 variabel
3. K-map 4 variabel
4. Penyederhanaan rangkaian dengan k-map

## *Tujuan Instruksional Khusus :*

1. Mahasiswa dapat menerangkan dan memahami cara membuat k-map 2, 3, 4 variabel.
2. Mahasiswa dapat menerangkan dan memahami cara peng-cover-an minterm dalam sebuah k-map..
3. Mahasiswa dapat menyederhanakan persamaan logika melalui metode k-map.

## Karnaugh Map (K-Map)

- Suatu peralatan grafis yang digunakan untuk menyederhanakan persamaan logika atau mengkonversikan sebuah tabel kebenaran menjadi sebuah rangkaian logika.
- Salah satu metode yang paling mudah untuk penyederhanaan Rangkaian Logika.

# Karnaugh Map 2 Variabel : ( A dan B )

Tabel Kebenaran

Map Value	A	B	Y
0	0	0	$A'B'$
1	0	1	$A'B$
2	1	0	$AB'$
3	1	1	$AB$

Model I

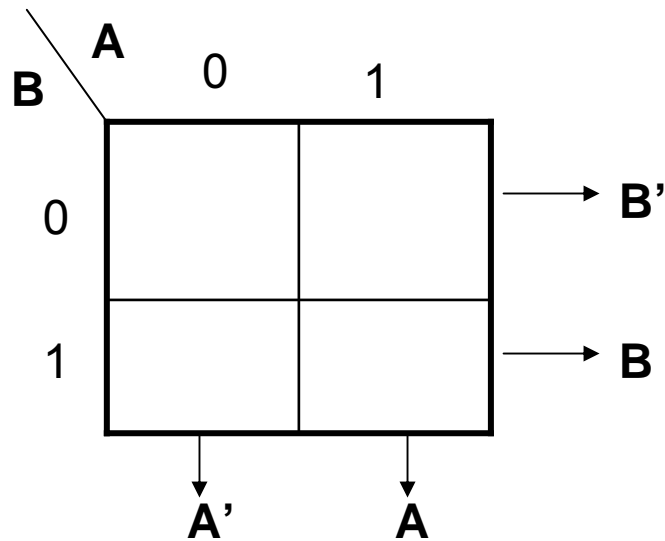
		B	
		0	1
A	0	$A'B'$ 0	$A'B$ 1
	1	$AB'$ 2	$AB$ 3

Map Value

Model II

		A	
		0	1
B	0	$A'B'$ 0	$AB'$ 2
	1	$A'B$ 1	$AB$ 3

## Desain Pemetaan K- Map 2 Variabel



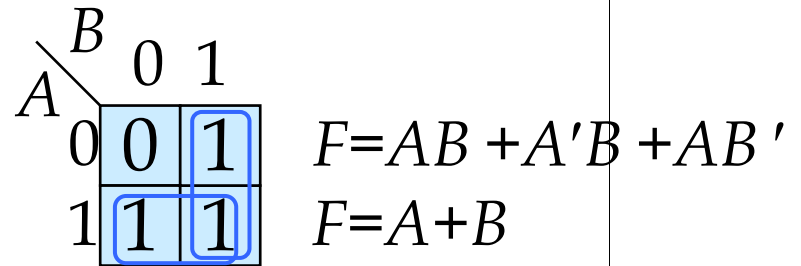
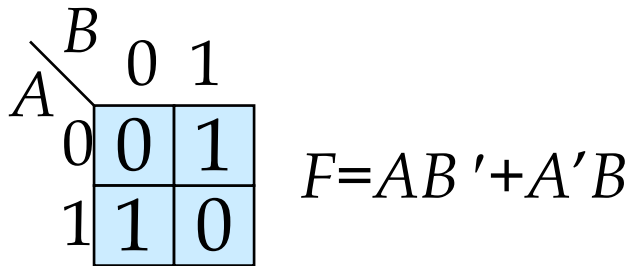
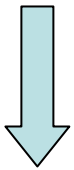
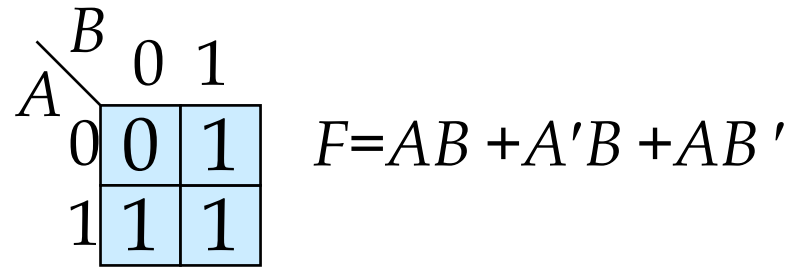
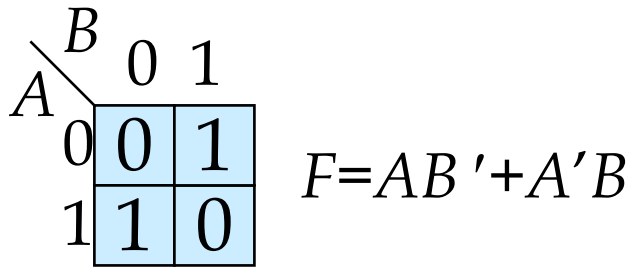
# Karnaugh Map 2 Variabel : dengan minterm-mintermnya

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

$$F = \Sigma(m_0, m_1) = x'y + x'y'$$

		y	
		0	1
x	0	1	1
	1	0	0

x	y	F
0	0	1
0	1	1
1	0	0
1	1	0



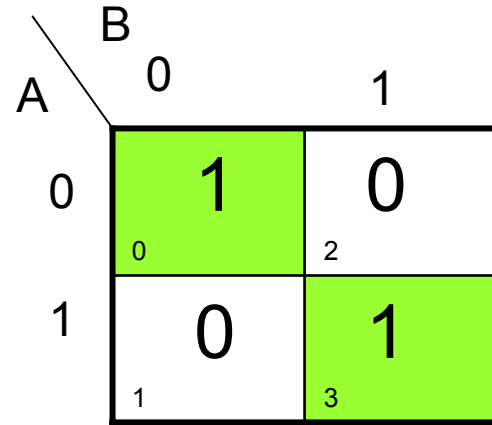
# Contoh : 1

Tabel Kebenaran

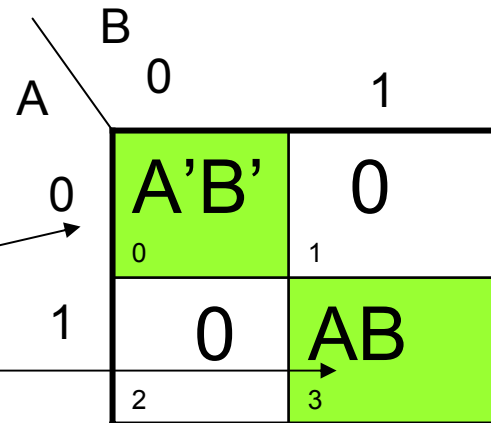
Map Value	A	B	Y
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

$A'B'$

$AB$



||



Jadi  $Y = A'B' + AB$

## Contoh : 2

Tabel Kebenaran

Map Value	A	B	Y
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	0

$A'B'$

$A'B$

Jadi  $Y = A'$

		B	
		0	1
A	0	1 0	1 1
	1	0 2	0 3

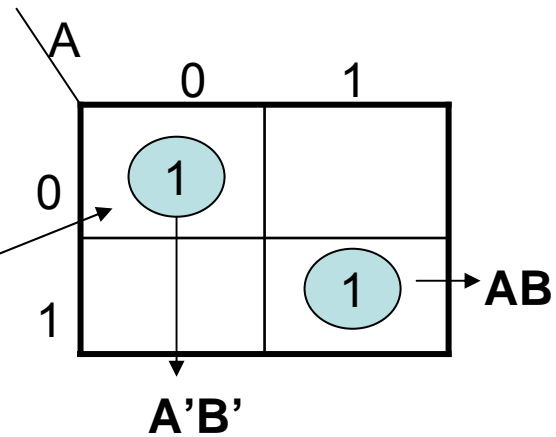
||

		B	
		0	1
A	0	$A'B'$ 0	$A'B$ 1
	1	0 2	0 3

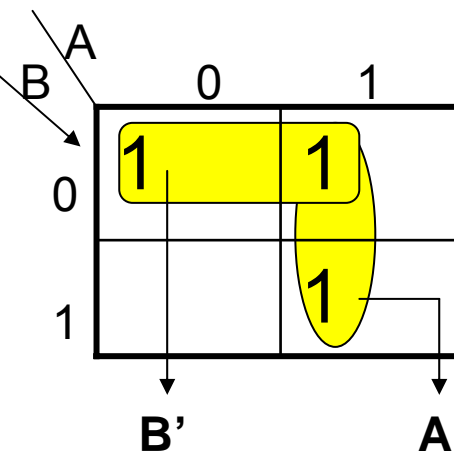


## Catatan untuk K-Map 2 Variabel

- 0 kotak terlingkupi = "0" (Low)
- 1 kotak terlingkupi = 2 variabel output
- 2 kotak terlingkupi = 1 variabel output
- 4 kotak terlingkupi = "1" (High)
- Melingkupinya harus posisi "Horisontal" atau "vertikal", yang dilingkupi digit "1" dan jumlah digit "1" yang dilingkupi  $2^n$  (1, 2, 4, 8, 16, ...)



$$Y = AB + A'B'$$



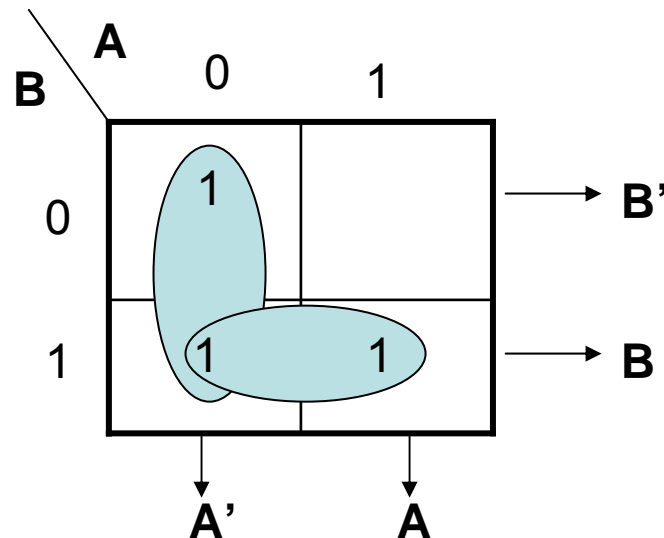
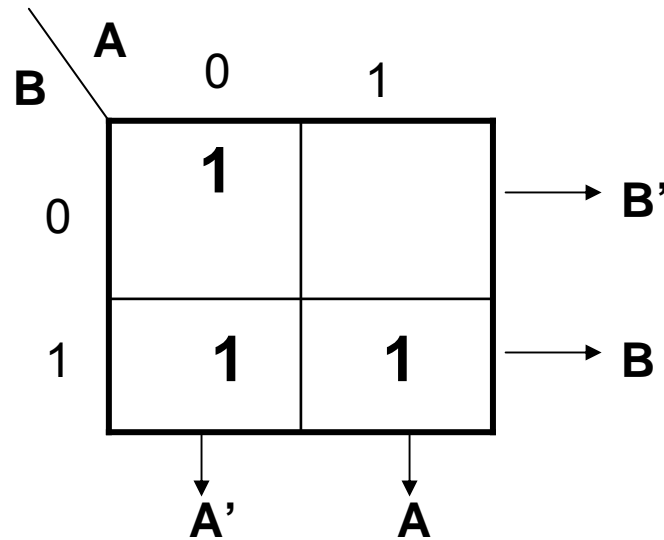
$$Y = B' + A$$

### Contoh 3:

Dari Tabel Kebenaran dibawah, tulis persamaan logikanya dengan menggunakan K-map :

Map Value	A	B	Y
0	0	0	1
1	0	1	1
2	1	0	0
3	1	1	1

Jadi  $Y = A' + B$

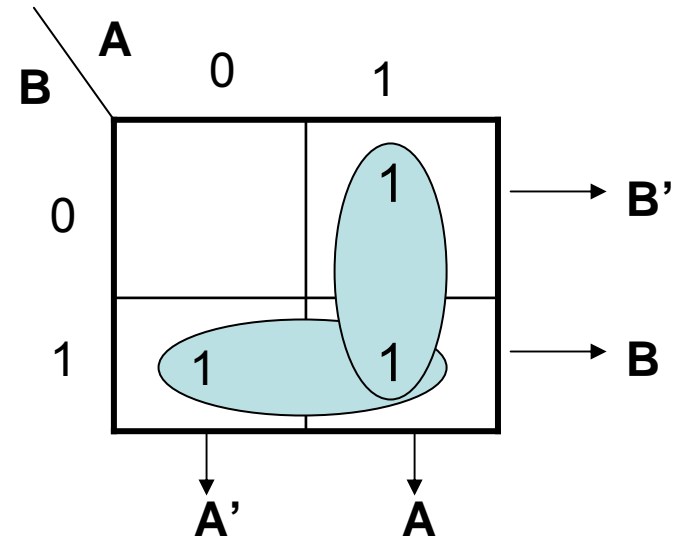
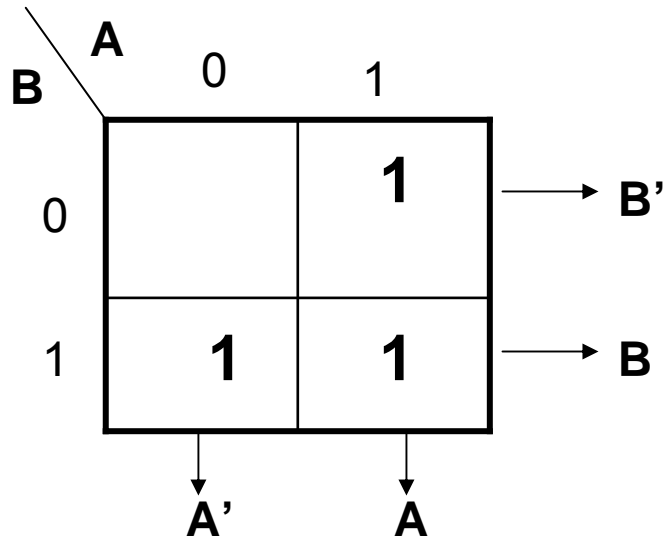


Contoh 4 :

Sederhanakan persamaan logika :

$$Y = A + AB' + A'B$$

Menggunakan K- map :

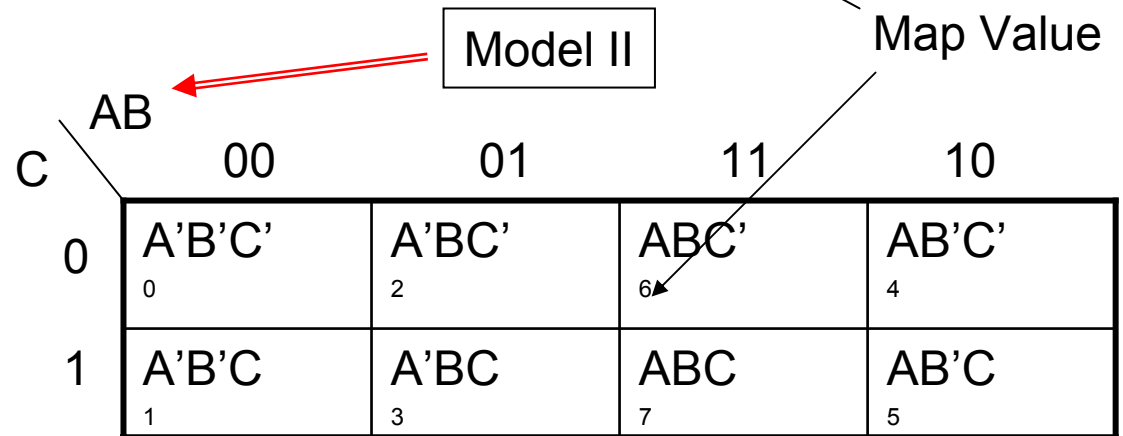
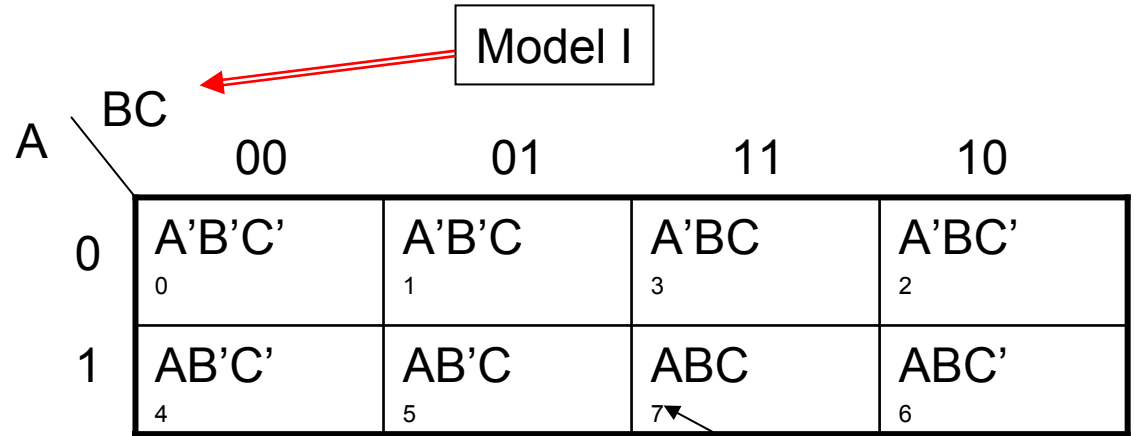


Jadi  $Y = A + B$

# Karnaugh Map 3 Variabel : ( A, B dan C )

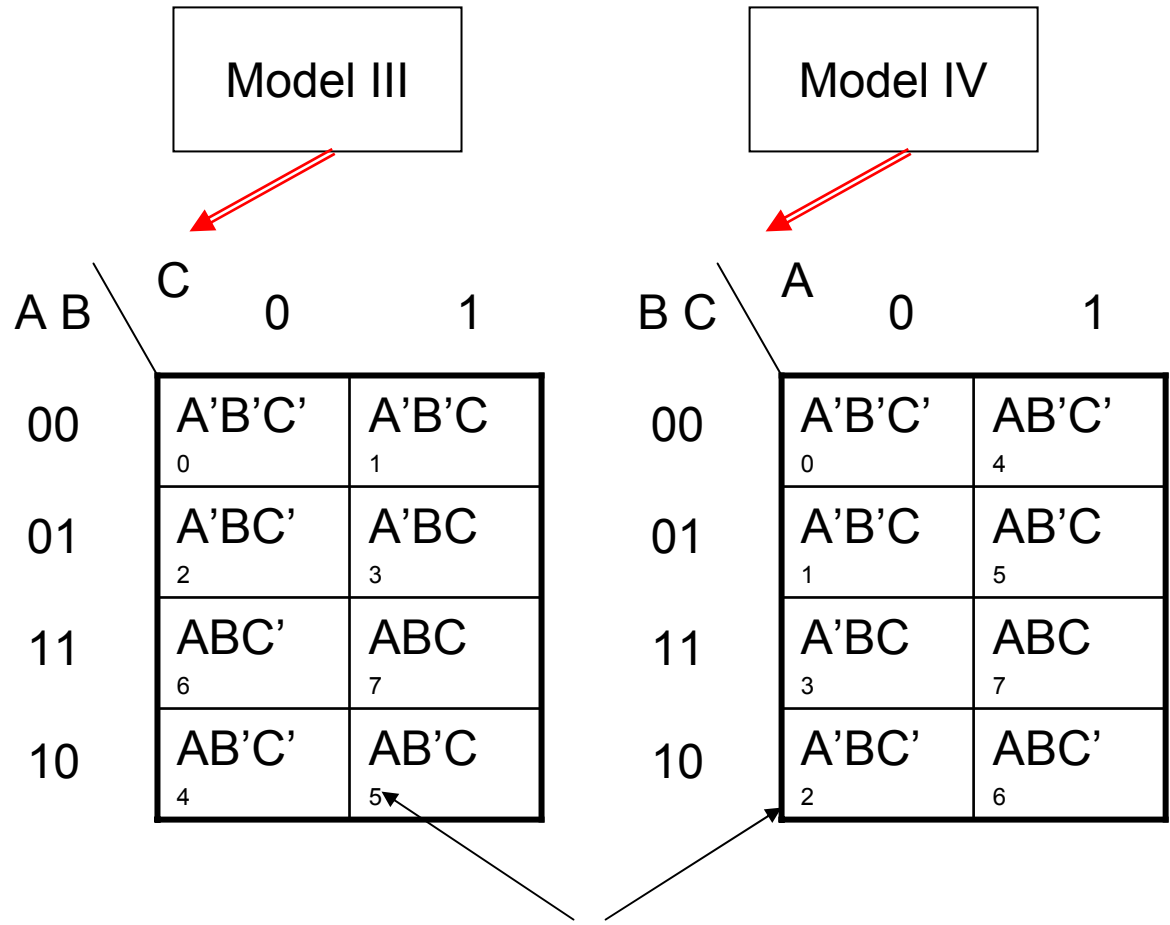
Tabel Kebenaran

Map Value	A	B	C	Y
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

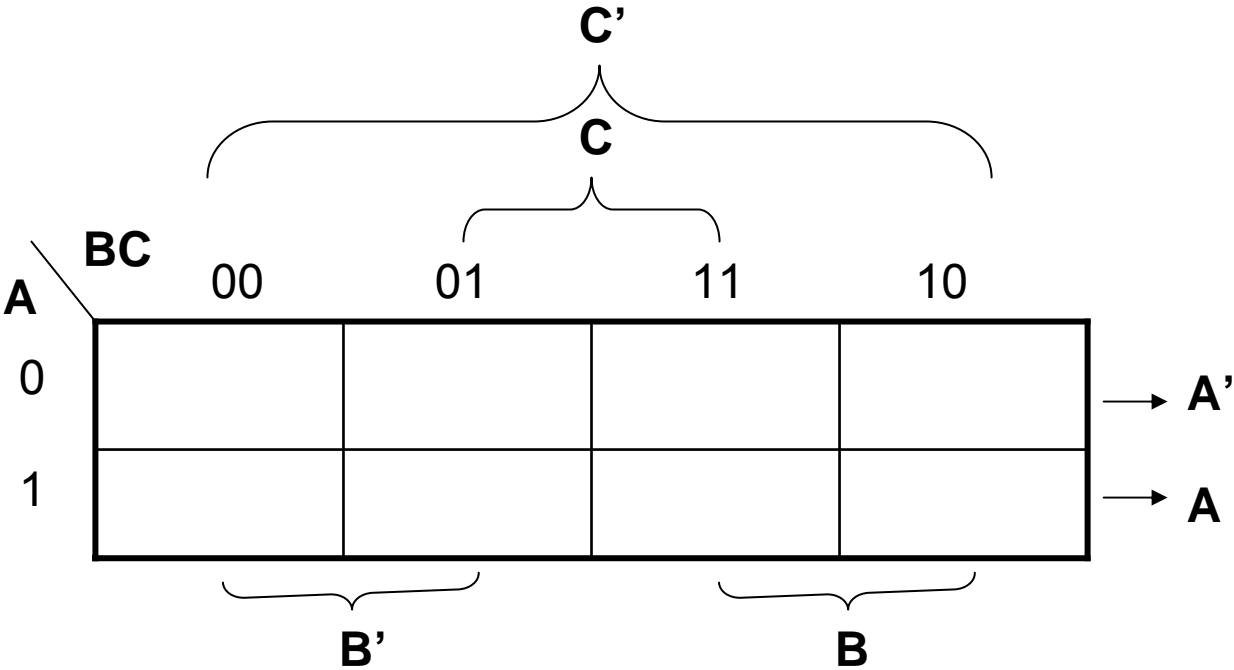


# Tabel Kebenaran

Map Value	A	B	C	Y
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

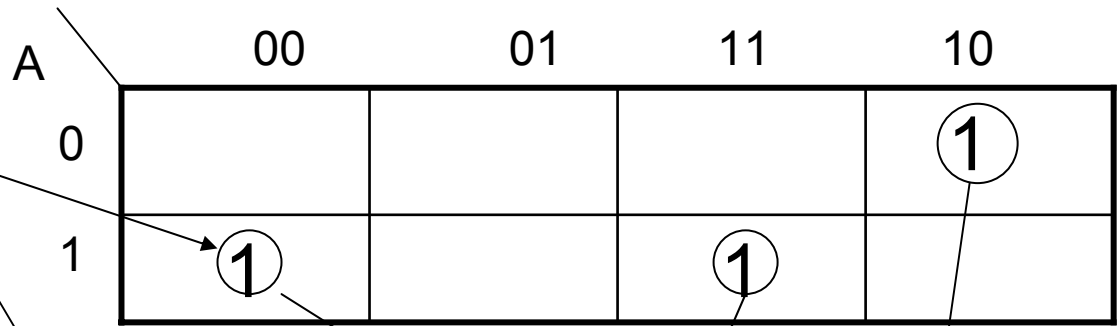


**Desain Pemetaan K- Map 3  
Variabel**

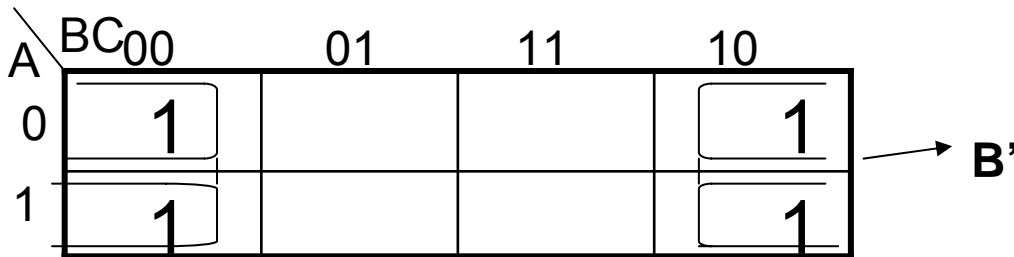
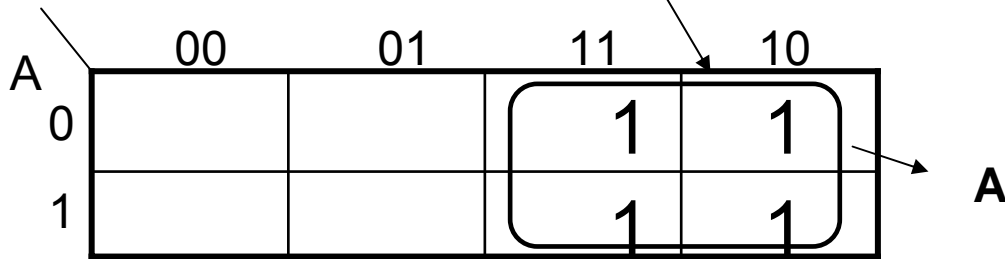


# Catatan untuk K- Map 3 Variabel

- 0 kotak terlingkupi = "0" (Low)
- 1 kotak terlingkupi = 3 variabel output
- 2 kotak terlingkupi = 2 variabel output
- 4 kotak terlingkupi = 1 variabel output
- 8 kotak terlingkupi = "1" (High)
- Melingkupinya harus posisi "Horisontal" atau "vertikal", yang dilingkupi digit "1" dan jumlah digit "1" yang dilingkupi  $2^n$  (1, 2, 4, 8, ...)



$$Y = AB'C' + A'BC + A'BC'$$



# Contoh pengcoveran

		AB		A	
		00	01	11	10
C	0				
	1				
		B			

		ab			
		00	01	11	10
c	0	0	0	1	0
	1	0	1	1	1

$cout = ab + bc + ac$

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

$G(A,B,C) = A$

		ab			
		00	01	11	10
c	0	0	0	1	1
	1	0	0	1	1

$f = a$

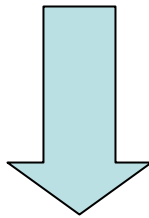
		A			
		1	0	0	1
C	0	0	1	1	
	1	0	1	1	
		B			

$F(A,B,C) = \sum m(0,4,5,7) = AC + B'C'$



		BC				A	B	C	F
		00	01	11	10	0	1	0	1
A	0	0	1	0	1	0	1	0	1
	1	1	1	1	1	1	0	0	1
		1	0	1	1	1	1	0	1
		1	1	0	1	1	1	1	1
		1	1	1	1	1	1	1	1

$$F = AB'C' + AB'C + ABC + ABC' + A'B'C + A'BC'$$



		BC				A	B	C	F
		00	01	11	10	0	1	0	1
A	0	0	1	0	1	0	1	0	1
	1	1	1	1	1	1	0	0	1

$$F = A + B'C + BC'$$

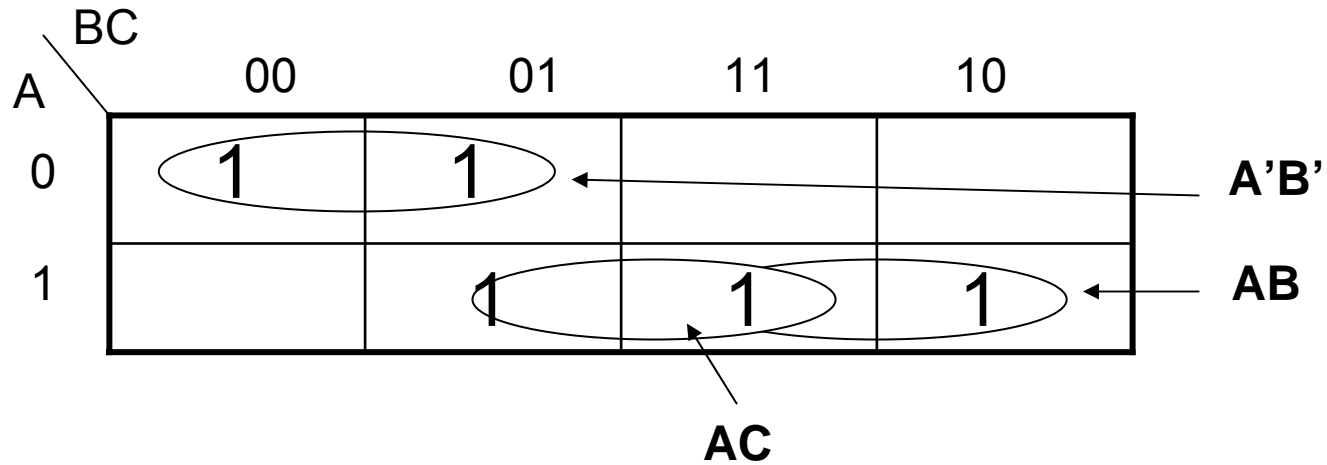
$$F = AB'C' + AB'C + ABC + ABC' + A'B'C + A'BC'$$

Contoh 1 :

Tabel Kebenaran

Map Value	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Diketahui Tabel Kebenaran seperti disamping :  
Cari persamaan logikanya :



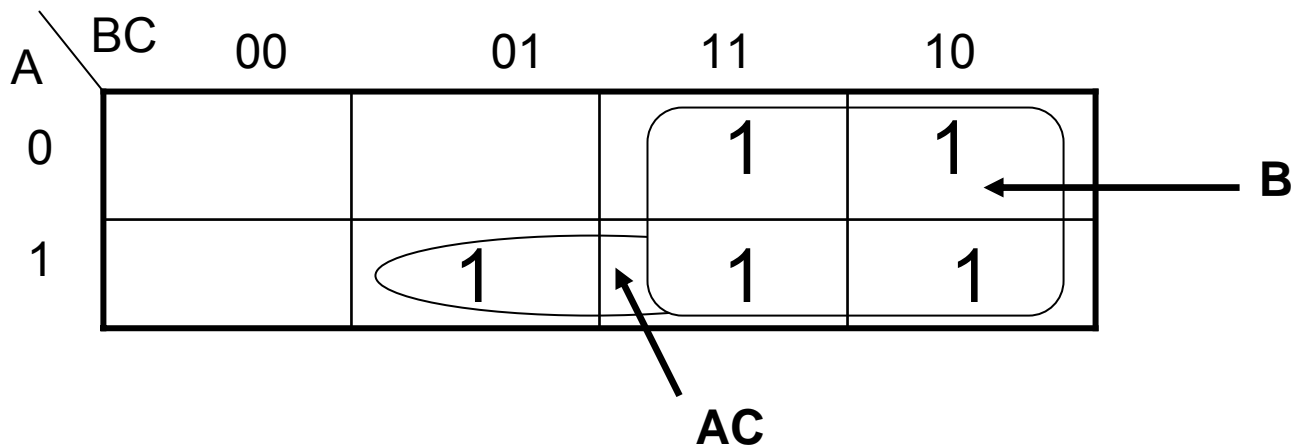
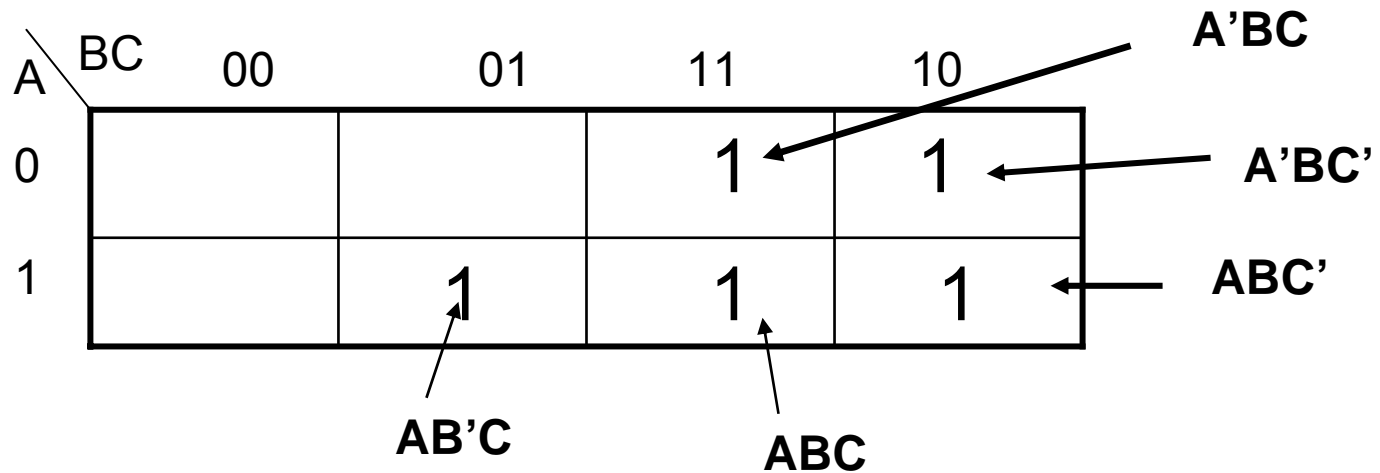
Jadi  $Y = AC + AB + A'B'$

## Contoh 2 :

Diketahui Persamaan Boolean :

$$D = A'BC + A'BC' + ABC' + ABC + AB'C$$

Sederhanakan dengan metode K-map



Jadi  $D = B + AC$

# Karnaugh Map 4 Variabel : ( A, B, C dan D )

Tabel Kebenaran

Map Valu e	A	B	C	D	Y
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

Model 1

		CD			
		00	01	11	10
AB	00	A'B'C'D' 0	A'B'C'D 1	A'B'CD 3	A'B'CD' 2
	01	A'BC'D' 4	A'BC'D 5	A'BCD 7	A'BCD' 6
	11	ABC'D' 12	ABC'D 13	ABCD 15	ABCD' 14
	10	AB'C'D' 8	AB'C'D 9	AB'CD 11	AB'CD' 10

Model 2

		CD			
		00	01	11	10
AB	00	A'B'C'D' 0	A'BC'D' 4	ABC'D' 12	AB'C'D' 8
	01	A'B'C'D 1	A'BC'D 5	ABC'D 13	AB'C'D 9
	11	A'B'CD 3	A'BCD 7	ABCD 15	AB'CD 11
	10	A'B'CD' 2	A'BCD' 6	ABCD' 14	AB'CD' 10

# Dengan wxyz input

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

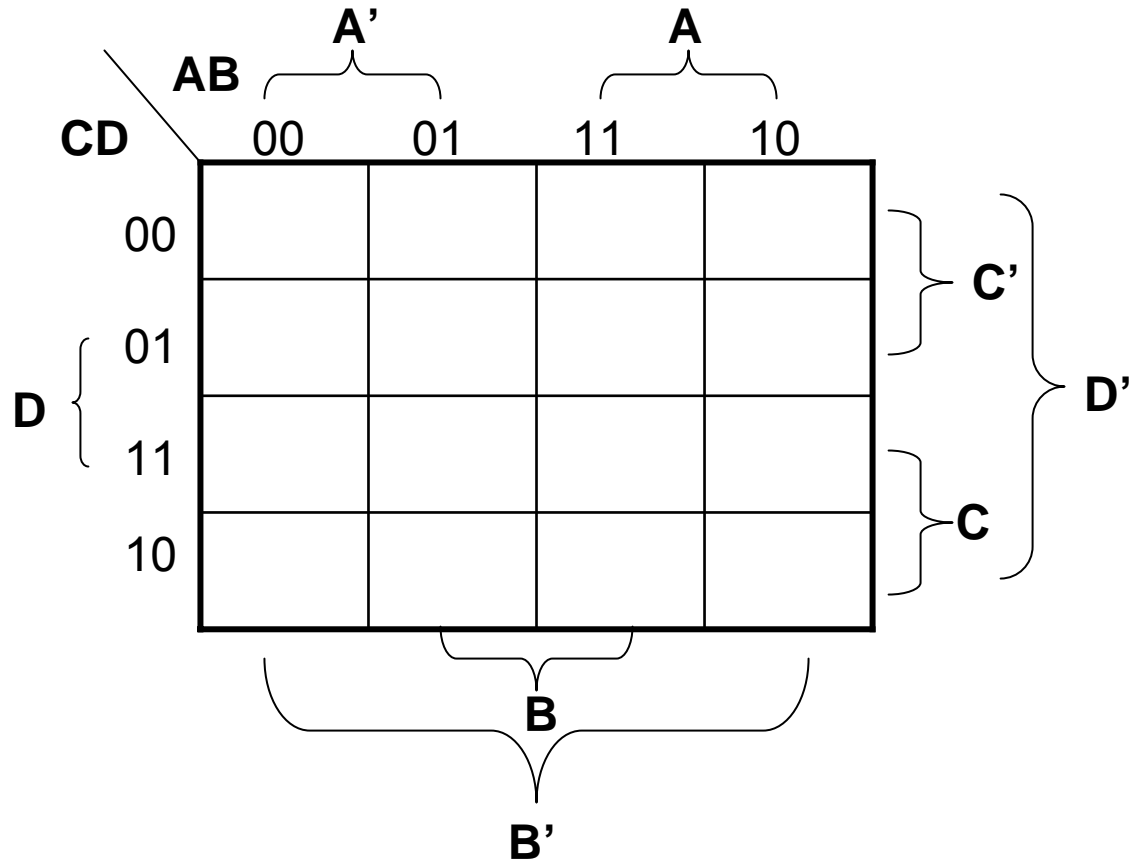
(a)

		$yz$		$y$	
	$wx$	00	01	11	10
	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	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$
	$w$	} $x$			
		} $z$			

(b)

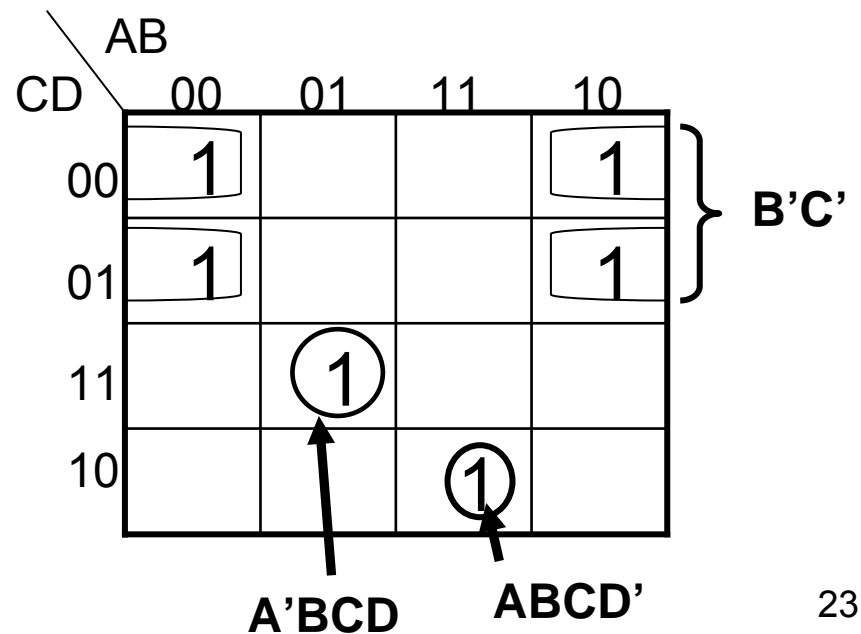
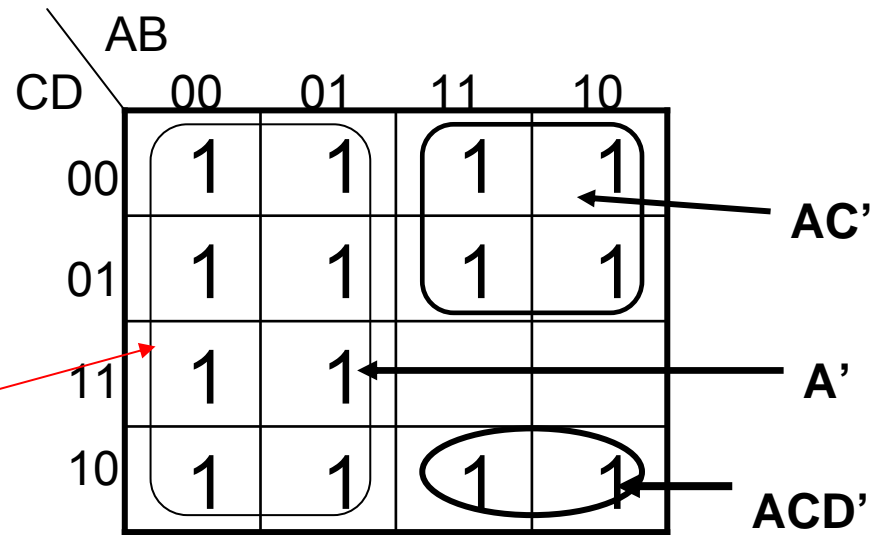
Fig. 3-8 Four-variable Map

# Desain Pemetaan K- Map 4 Variabel



# Catatan untuk K-Map 4 Variabel

- 0 kotak terlingkupi = "0" (Low)
- 1 kotak terlingkupi = 4 variabel output
- 2 kotak terlingkupi = 3 variabel output
- 4 kotak terlingkupi = 2 variabel output
- 8 kotak terlingkupi = 1 variabel output
- 16 kotak terlingkupi = "1" (High)
- Melingkupinya harus posisi "Horizontal" atau "vertikal", yang dilingkupi digit "1" dan jumlah digit "1" yang dilingkupi  $2^n$  ( 1,2, 4, 8, 16, ... )



# Contoh pengcoveran :

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

K-map untuk LT

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

K-map untuk EQ

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

K-map untuk GT

$$LT = A' B' D + A' C + B' C D$$

$$EQ = A' B' C' D' + A' B C' D + A B C D + A B' C D'$$

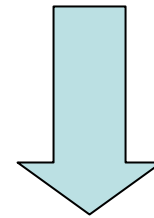
$$GT = B C' D' + A C' + A B D'$$



# Contoh pengcoveran :

		CD			
		00	01	11	10
AB	00	0	0	0	1
	01	1	1	0	1
	11	1	1	1	1
	10	1	0	1	1

$$F = A'BC' + A'CD' + ABC + AB'C'D' + ABC' + AB'C$$



$$F = BC' + CD' + AC + AD'$$

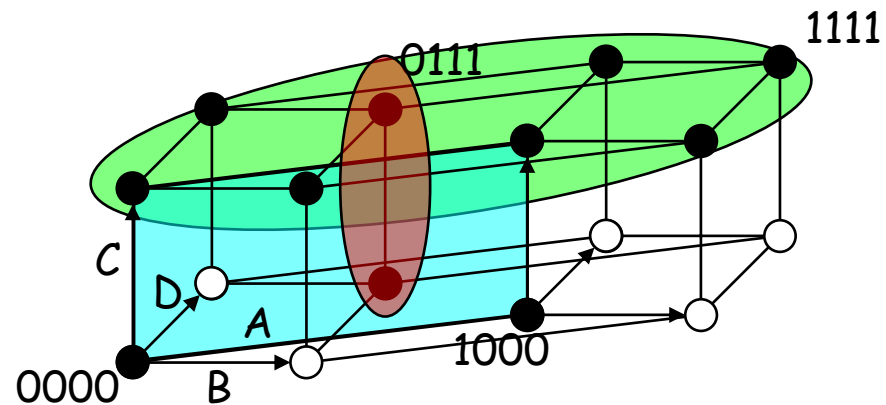
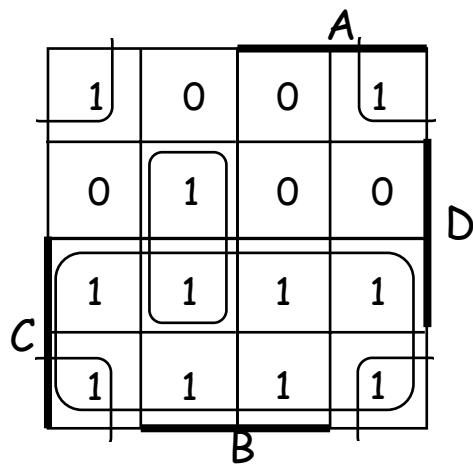
# Contoh 1

- $F(A,B,C,D) =$

$$\Sigma m(0,2,3,5,6,7,8,10,11,14,15)$$

$$F =$$

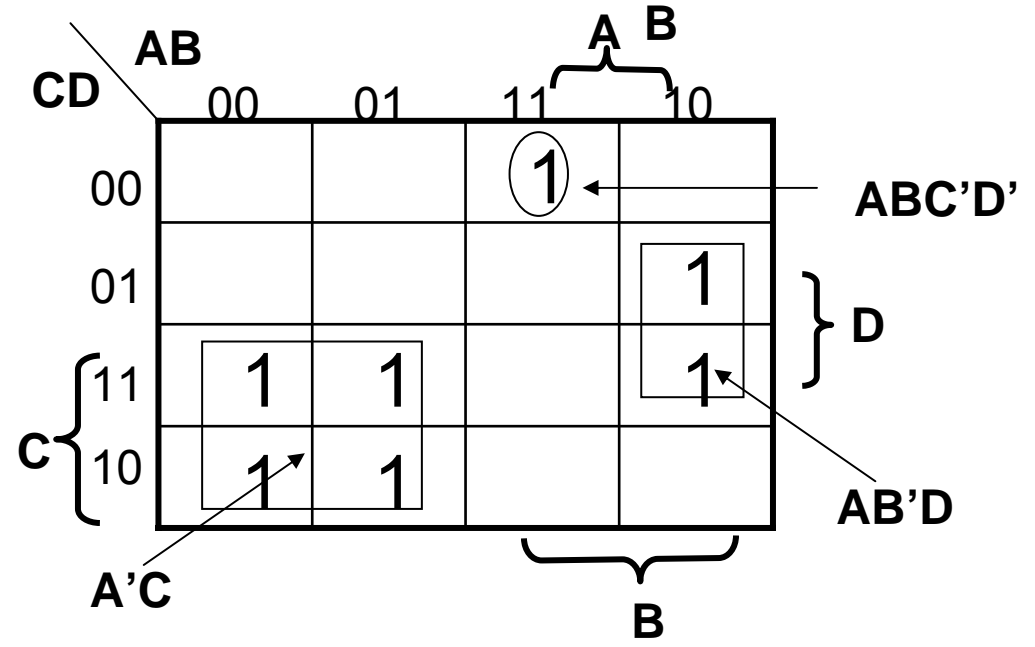
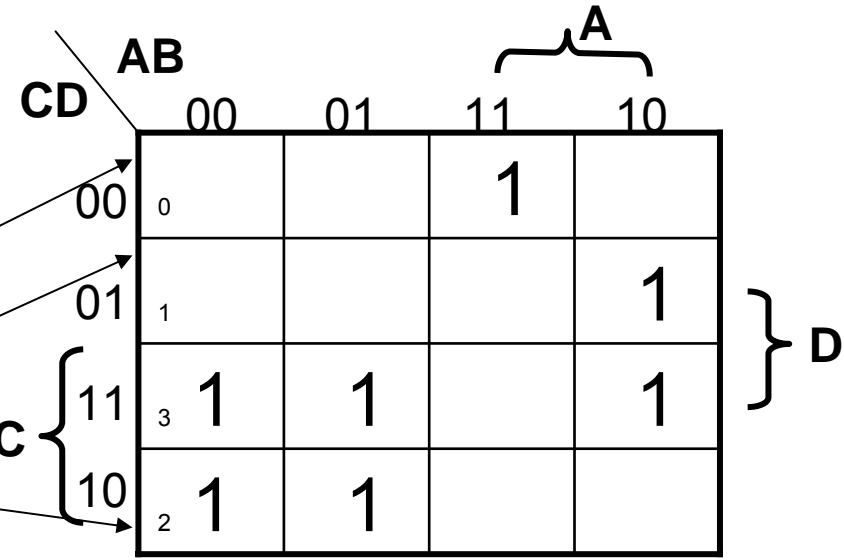
$$C + A'BD + B'D'$$



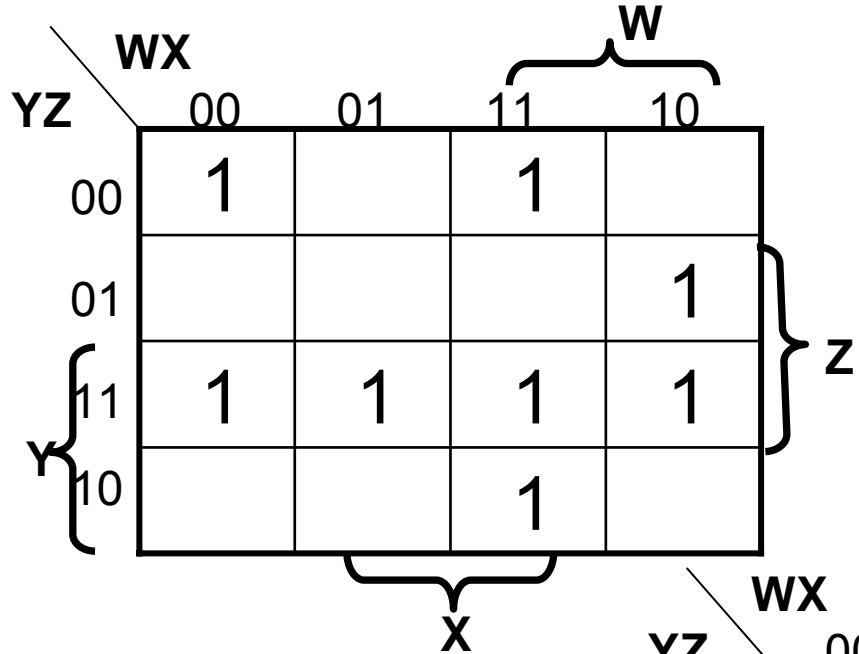
Kalau digambarkan dengan system coordinate

**Contoh 2 : Diketahui Tabel Kebenaran , cari persamaan logikanya.**

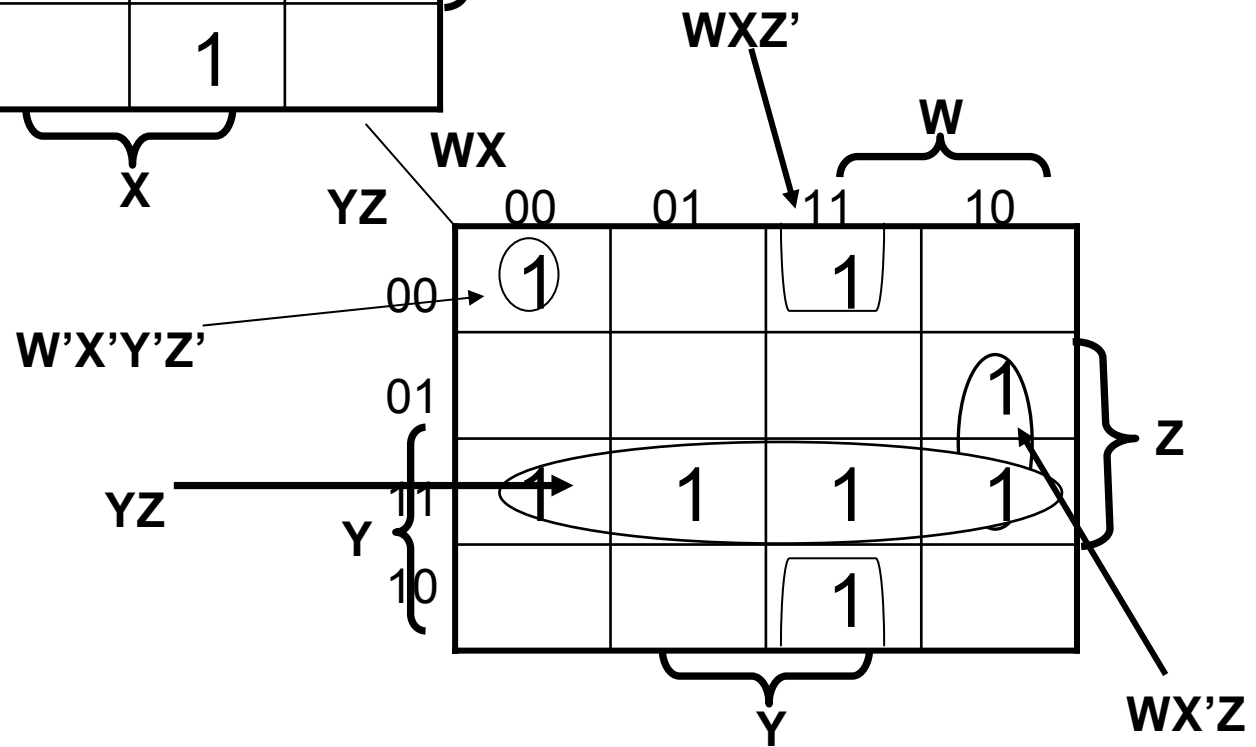
Map Value	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0



**Jadi  $Y = A'C + AB'D + ABC'D'$**

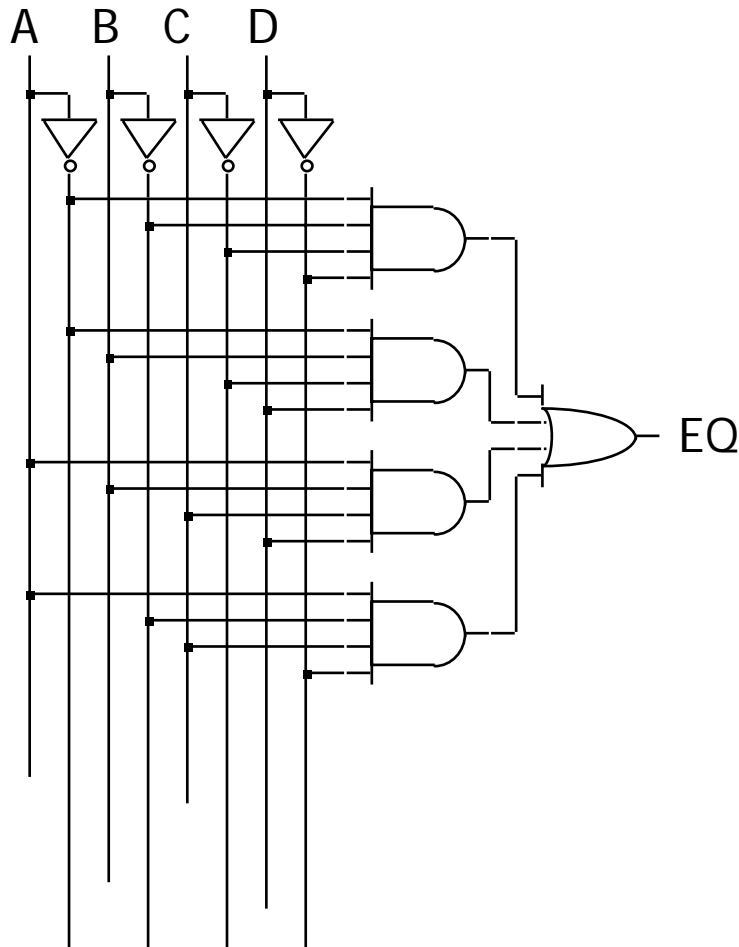


**Contoh 3 : Lingkariilah dan Tulis Persamaan Logiknya.**



**Jadi  $M = W'X'Y'Z' + WXZ' + WXX'Z + YZ$**

# Physical Implementasi



- Step 1: Truth table
- Step 2: K-map
- Step 3: Minimized sum-of-products
- Step 4: Implementasi dengan gates

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

K-map untuk EQ

# Poin-poin penggunaan K-map

- Buat persamaan ke bentuk SOP (melalui tabel kebenaran).
- Minterm-mintermnya masukkan ke k-map (sesuaikan jumlah kotak atau variabel input).
- Lingkari (pengcoveran) yang benar.
- Tulis persamaan logika hasil pengcoveran.

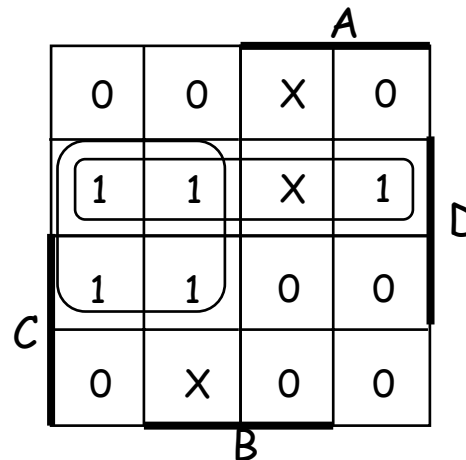
# Don't Care

- Kondisi don't care merupakan kondisi dimana ada beberapa kombinasi variable input yang tidak selalu dapat dinyatakan nilai outputnya.
- Keadaan dimana nilai outputnya tersebut bisa berlogic '1' atau berlogic '0' yang disimbulkan dengan "X" atau "d".
- Kegunaan dari kondisi don't care pada penyederhanaan fungsi dapat dinyatakan pada fakta bahwa dapat diset dengan logic '1' atau logic '0', berdasar kegunaannya untuk format kelompok logic '1' yang lebih besar.

# Karnaugh maps: don't cares (cont'd)

- $f(A,B,C,D) = \sum m(1,3,5,7,9) + d(6,12,13)$ 
  - $f = A'D + B'C'D$       tanpa don't cares
  - $f = A'D + C'D$       dengan don't cares

A	B	C	D	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	X
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	X
1	1	0	1	X
1	1	1	0	0
1	1	1	1	0





# Pengcoveran dengan Don't Cares

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	1	0	0
	01	x	x	x	1
	11	1	1	1	x
	10	x	0	1	1



$$F = A'C'D + B + AC$$

# Bentuk ilustrasi pengkoveran

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

6 prime implicants:

$A'B'D$ ,  $BC'$ ,  $AC$ ,  $A'C'D$ ,  $AB$ ,  $B'CD$

essential

minimum cover: 3 essential implicants

minimum cover:  $AC + BC' + A'B'D$

5 prime implicants:

$BD$ ,  $ABC'$ ,  $ACD$ ,  $A'BC$ ,  $A'C'D$

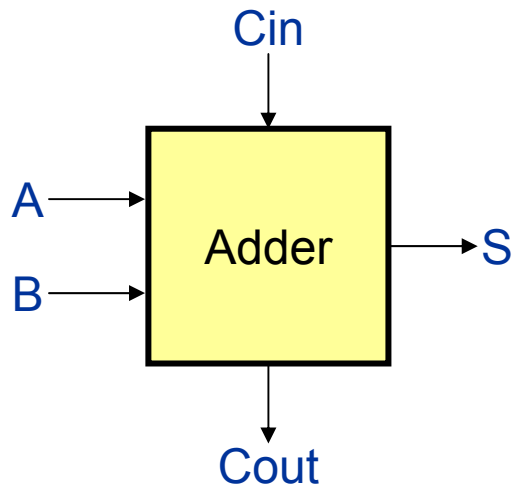
essential

minimum cover: 4 essential implicants

minimum cover:  $ABC' + ACD + A'BC + A'C'D$

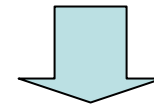
		A		
	0	0	1	0
	1	1	1	0
C	0	1	1	1
	0	1	0	0
		B		

# Aplikasi K-map Pada Rangkaian Full Adder



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Metode Aljabar Boole



$$S = A'B'Cin + A'BCin' + A'BCin + ABCin$$

$$Cout = A'BCin + A B'Cin + ABCin' + ABCin$$

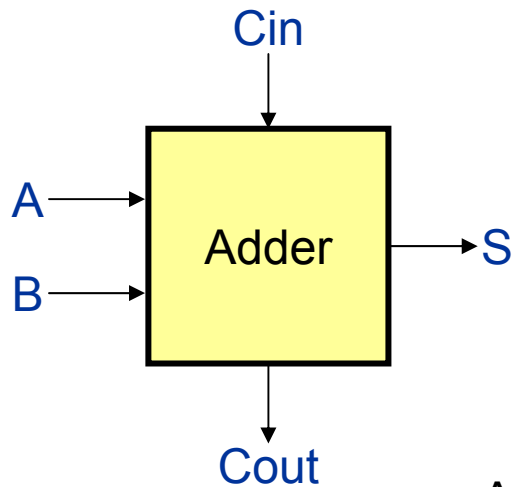
$$= A'BCin + ABCin + AB'Cin + ABCin + ABCin' + ABCin$$

$$= (A' + A)BCin + (B' + B)ACin + (Cin' + Cin)AB$$

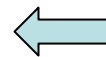
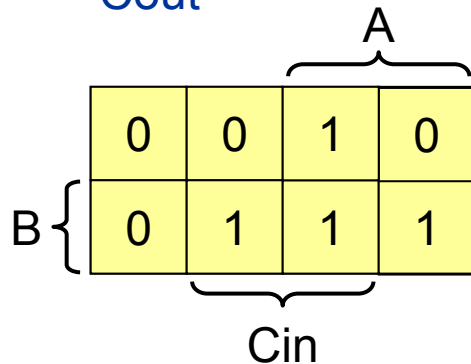
$$= 1 \cdot BCin + 1 \cdot ACin + 1 \cdot AB$$

$$= BCin + ACin + AB$$

# Aplikasi K-map Pada Rangkaian Full Adder



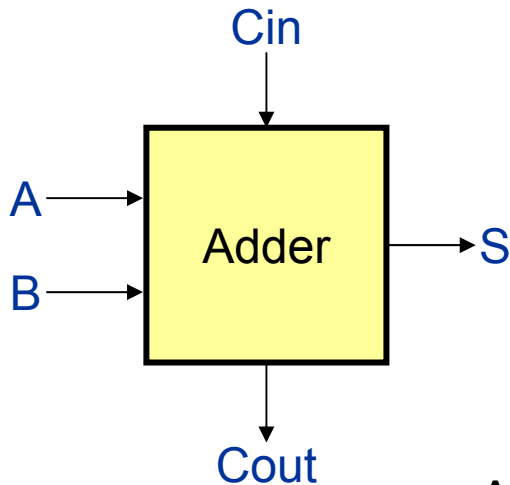
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



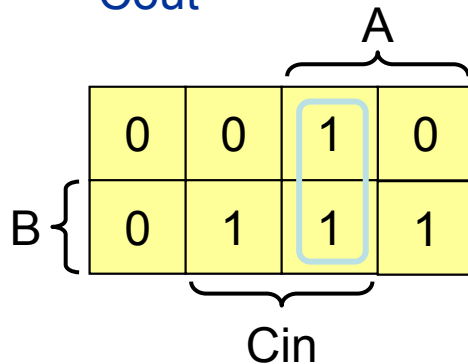
Pengisian digit 1 ke K-map

Karnaugh Map for Cout

# Aplikasi K-map Pada Rangkaian Full Adder



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

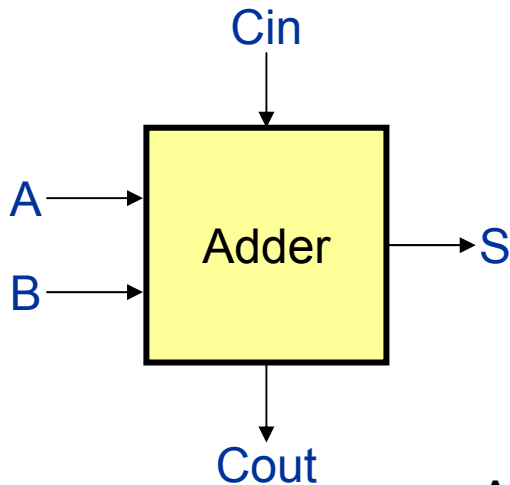


← Pengcoveran pertama.

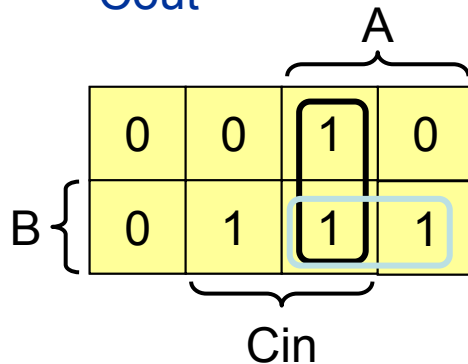
$$\text{Cout} = ACin$$

Karnaugh Map untuk Cout

# Aplikasi K-map Pada Rangkaian Full Adder



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

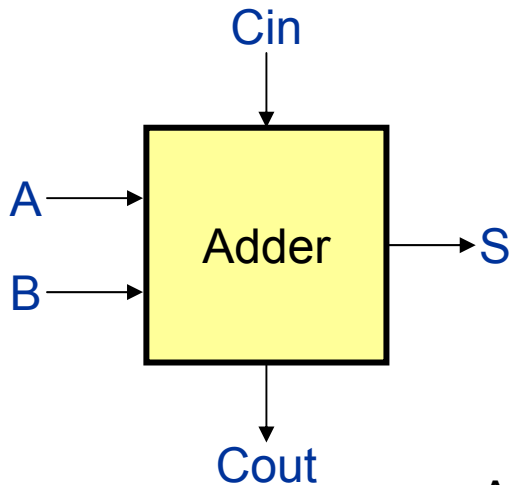


← Pengcoveran kedua.

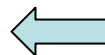
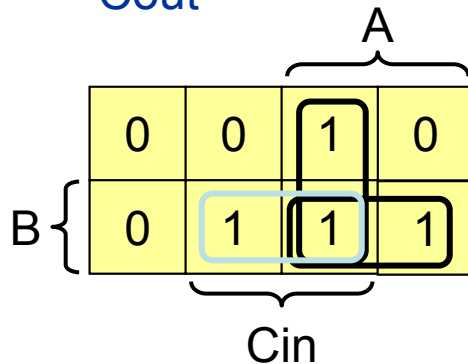
$$Cout = A_{cin} + AB$$

Karnaugh Map for Cout

# Aplikasi K-map Pada Rangkaian Full Adder



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

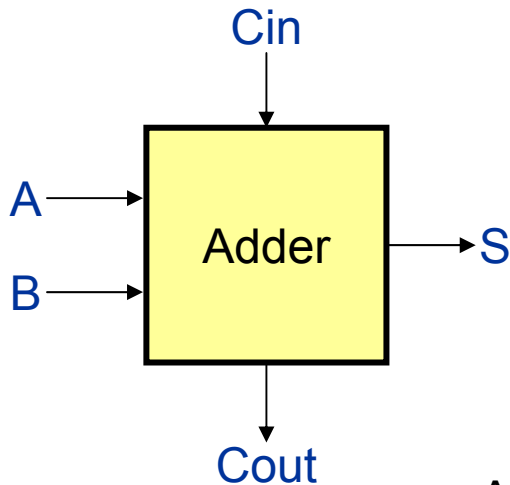


Pengcoveran ketiga (seluruhnya)

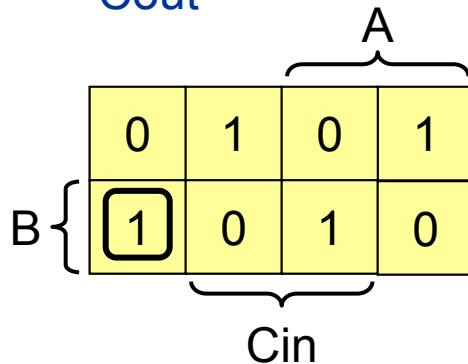
$$Cout = ACin + AB + BCin$$

Karnaugh Map untuk Cout

# Aplikasi K-map Pada Rangkaian Full Adder



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

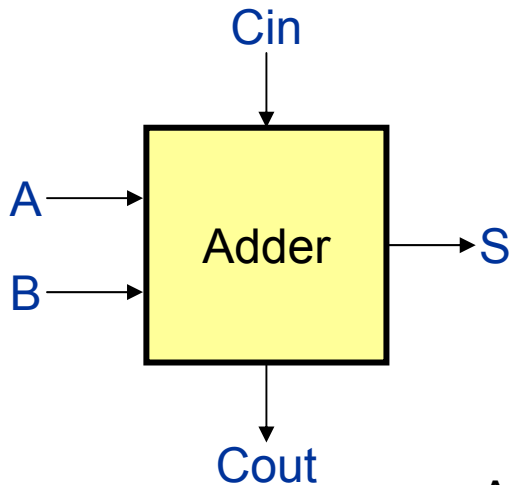


$$S = A'BCin'$$

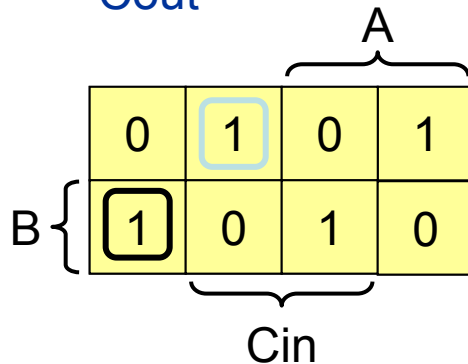
Karnaugh Map untuk S



# Aplikasi K-map Pada Rangkaian Full Adder



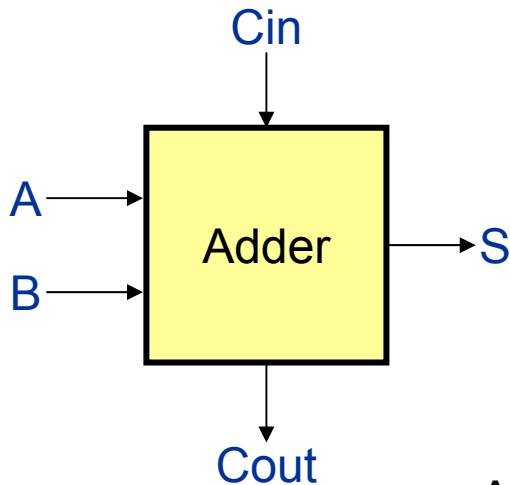
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



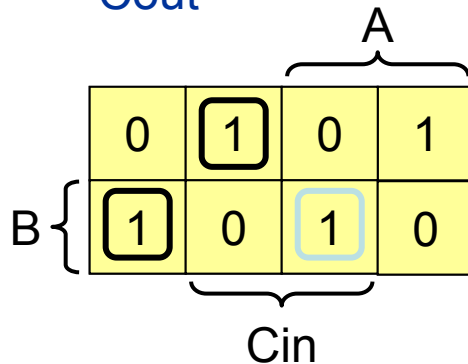
$$S = A'BCin' + A'B'Cin$$

Karnaugh Map untuk S

# Aplikasi K-map Pada Rangkaian Full Adder



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

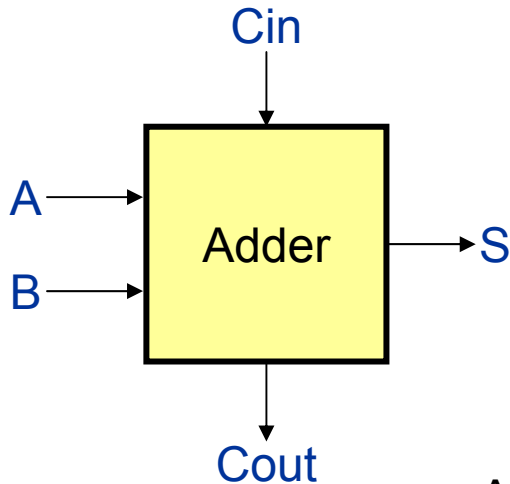


$$S = A'BCin' + A'B'Cin + ABCin$$

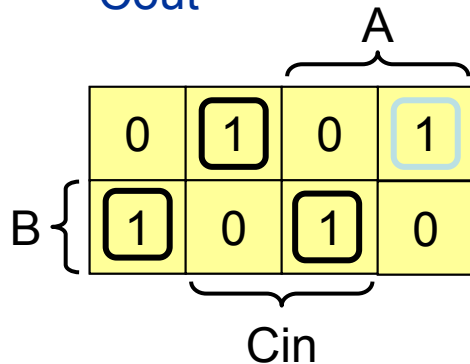
Karnaugh Map untuk S

# Aplikasi K-map Pada Rangkaian Full Adder

Coba anda gambar rangkaian diagramnya ?



A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Karnaugh untuk S

$$S = A'BCin' + A'B'Cin + ABCin + AB'Cin'$$

Tidak bisa direduksi

## Latihan Soal 1:

Gambarlah K-map untuk setiap ekspresi logika dibawah serta sederhanakan dengan pengcoveran yang benar :

1.  $AB + B'C + A'B'$
2.  $AC + AC'B + BC + B'C'$
3.  $XY + X'Z + Y'Z'$
4.  $XY + YZ + XZ + X'Y'$

## Latihan Soal 2 :

Gambarlah K-map untuk setiap ekspresi logika dibawah serta sederhanakan dengan pengcoveran yang benar :

1.  $A(BC' + C) + B(A + A'C)$
2.  $(AC + AC'B) \cdot (BC + B'C')$
3.  $Z(XY + X'Z) \cdot Y'Z'(X + Z)$

Catatan : cari minterm-mintermnya dulu (rubah kebentuk SOP)