Penyederhanaan Fungsi Logika [Sistem Digital]

Eka Maulana, ST, MT, MEng.
Universitas Brawijaya

Mengapa perlu Penyederhanaan?

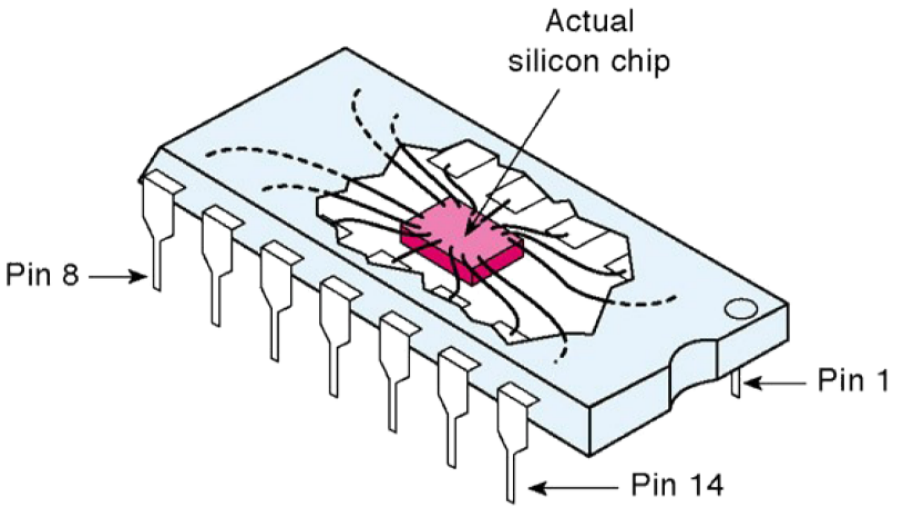
SEDERHANA → Cheaper → Smaller → **Faster**

Diperlukan MANIPULASI ALJABAR BOOLE

Metode:

Penyederhanaan Fungai Boole
Bentuk Standard dan Kanonik
Representasi dengan Tabel Kebenaran
Karnaugh Map (K-MAP)

Makin Ringkas Makin Optimal

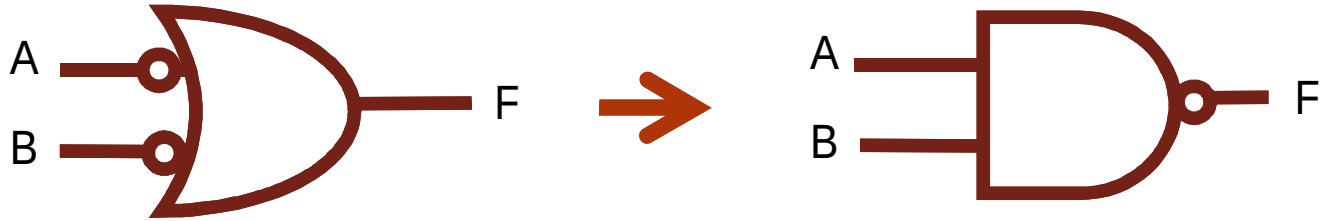


Berdasarkan Jumlah rangkaian Logika dalam satu chip

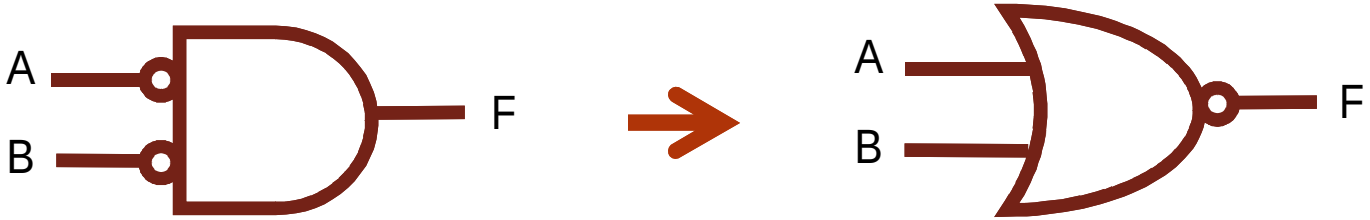
SSI, kurang dari 12
MSI, 12 - 99
LSI, 100 - 9.999
VLSI, 10.000 - 99.999
ULSI, 100.000 - 999.999
GSI, 1.000.000 atau lebih

Penting dalam Penyederhanaan!!!

Hukum De Morgan Commonly used…

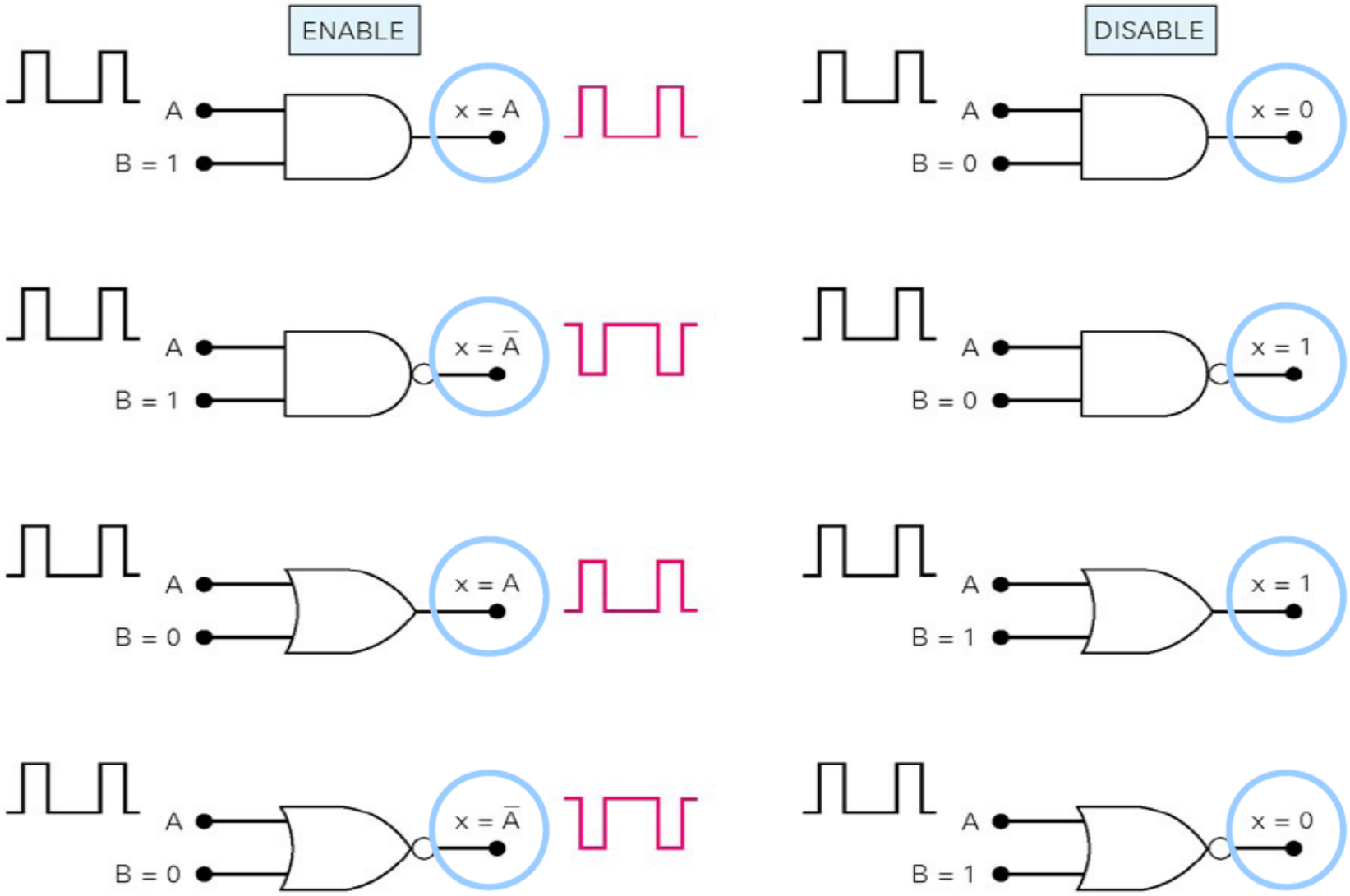


$$\overline{A}+\overline{B} = \overline{A.B}$$

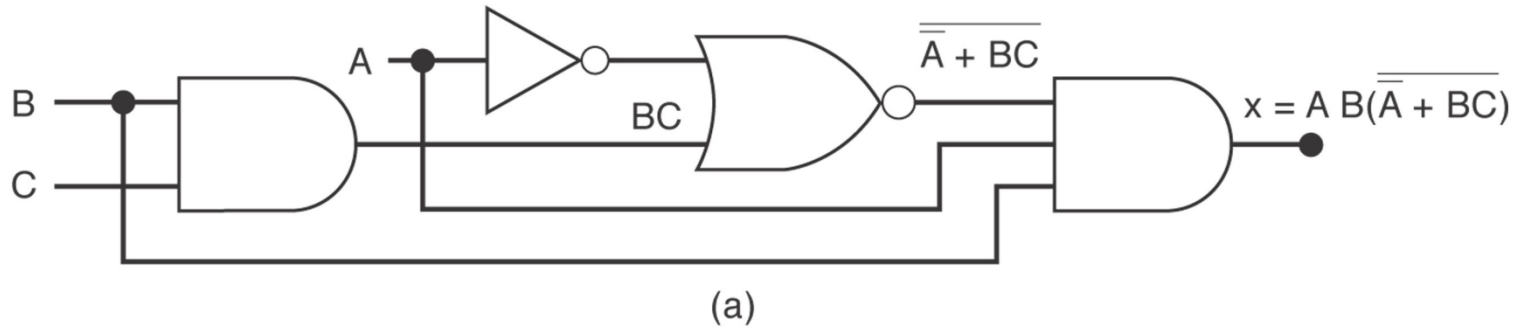


$$\overline{A}.\overline{B} = \overline{A+B}$$

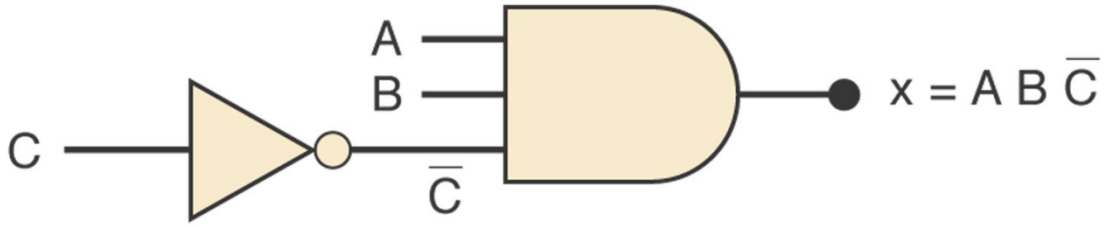
Gerbang Logika dapat meneruskan sinyal input?



Mana yang kita pilih?



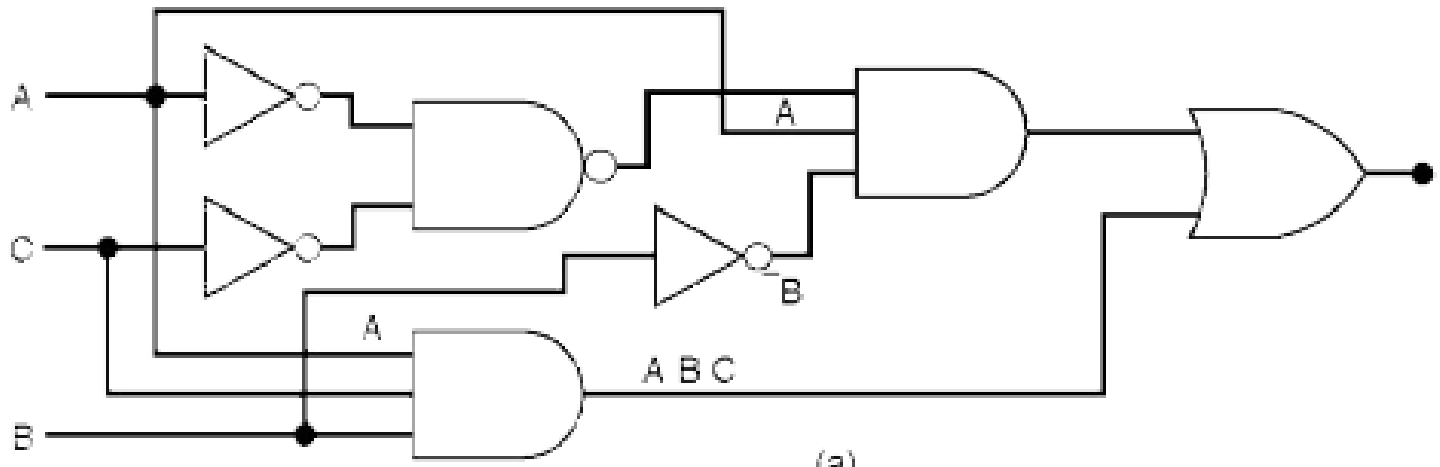
(a)



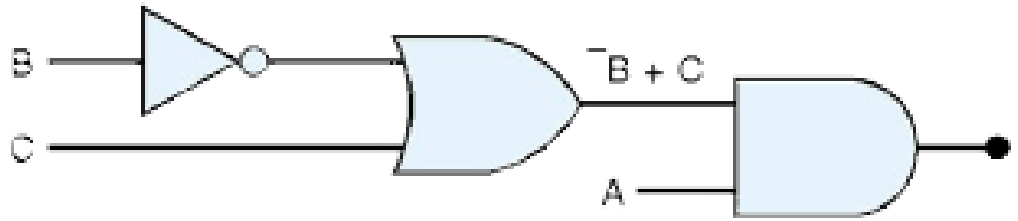
(b)

Mari kita sederhanakan…[manual]

Sederhanakan rangkaian a ke b



(a)



(b)

Mari kita sederhanakan…[manual]

Esensi Penyederhanaan?

Penyederhanaan berarti meminimalisir jumlah **operasi** dan **variabel** dalam ekspresi aljabar logika.

Sederhanakan ekspresi logika berikut:

a. F = ABC + ABC’ + AB’C

b. F = A’C(A’BD)’ + A’BC’D’ + AB’C

c. F = (A’+B).(A+B+D)D’

d. F = AB’C+A’BD+C’D’

e. F = A’BC + A’BC’ + AC

f. F = A’B’C’ + A’BC’ + ABC’

Bentuk Standard dan Kanonik

Bentuk Standard

SOP (Sum of Product) → *Term-term* **AND** di **OR** kan
contoh: AB'C + A'BC'

POS (Product of Sum) → *Term-term* **OR** di **AND** kan
contoh: (A+B'+C).(A'+B+C')

Bentuk Kanonik

Minterm→ product term in which all the variables appear exactly once, either complemented or uncomplemented

Maxterm → sum term in which all the variables appear exactly once, either complemented or uncomplemented

Minterm (m)

- Represents exactly one combination in the truth table.
- Denoted by m_j, where j is the decimal equivalent of the minterm's corresponding binary combination (b_j).
- A variable in m_j is complemented if its value in b_j is 0, otherwise is uncomplemented.
- Example: Assume 3 variables (A,B,C), and j=3. Then, b_j = 011 and its corresponding minterm is denoted by m_j = A'BC

Maxterm (M)

- Represents exactly one combination in the truth table.
- Denoted by M_j, where j is the decimal equivalent of the maxterm's corresponding binary combination (b_j).
- A variable in M_j is complemented if its value in b_j is 1, otherwise is uncomplemented.
- Example: Assume 3 variables (A,B,C), and j=3. Then, b_j = 011 and its corresponding maxterm is denoted by M_j = A+B'+C'

Notasi Tabel Kebenaran Minterm & Maxterm

- Minterms dan Maxterms mudah direpresentasikan menggunakan tabel kebenaran.
- Contoh:

  Asumsikan 3 variabel x,y,z

x	y	z		Minterm	Maxterm
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		$xyz = m_7$	$x'+y'+z' = M_7$

Contoh Penulisan m dan M

Tabel kebenaran untuk $f_1(a,b,c)$

- The canonical sum-of-products form for f_1 is
  $f_1(a,b,c) = m_1 + m_2 + m_4 + m_6$
  $= a'b'c + a'bc' + ab'c' + abc'$
- The canonical product-of-sums form for f_1 is
  $f_1(a,b,c) = M_0 \cdot M_3 \cdot M_5 \cdot M_7$
  $= (a+b+c)\cdot(a+b'+c')\cdot$
  $(a'+b+c')\cdot(a'+b'+c').$

Observe that: $m_j = M_j'$

a	b	c		f_1
0	0	0		0
0	0	1		1
0	1	0		1
0	1	1		0
1	0	0		1
1	0	1		0
1	1	0		1
1	1	1		0

Shorthand: $\sum$ and $\prod$

- $f_1(a,b,c) = \sum m(1,2,4,6)$, where $\sum$ indicates that this is a sum-of-products form, and m(1,2,4,6) indicates that the minterms to be included are m_1, m_2, m_4, and m_6.
- $f_1(a,b,c) = \prod M(0,3,5,7)$, where $\prod$ indicates that this is a product-of-sums form, and M(0,3,5,7) indicates that the maxterms to be included are M_0, M_3, M_5, and M_7.
- Since $m_j = M_j'$ for any *j*,

  $\sum m(1,2,4,6) = \prod M(0,3,5,7) = f_1(a,b,c)$

Konversi diantara bentuk Kanonik ($\sum$ dan $\prod$)

- Replace $\sum$ with $\prod$ (or *vice versa*) and replace those *j*'s that appeared in the original form with those that do not.
- Example:

$f_1(a,b,c) = a'b'c + a'bc' + ab'c' + abc'$

$= m_1 + m_2 + m_4 + m_6$

$= \sum(1,2,4,6)$

$= \prod(0,3,5,7)$

$= (a+b+c)\bullet(a+b'+c')\bullet(a'+b+c')\bullet(a'+b'+c')$

Perancangan dgn Tabel Kebenaran

Diketahui output x seperti dalam tabel, tentukan fungsi logika dari x.

A	*B*	*C*	*x*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Langkah-langkah solusi dgn Tabel Kebenaran

#1 Tulis bentuk AND pada output = 1.

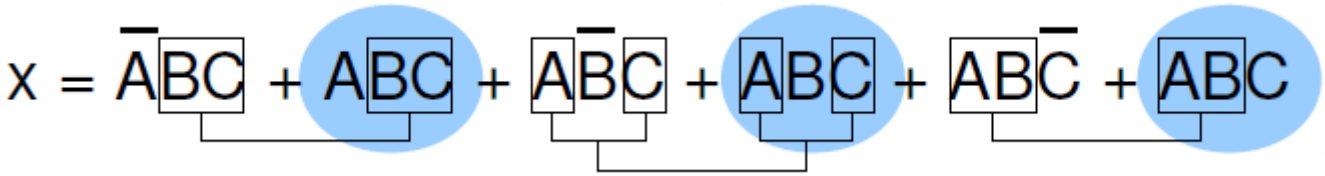
A	B	C	x	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \bar{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\bar{B}C$
1	1	0	1	$\rightarrow AB\bar{C}$
1	1	1	1	$\rightarrow ABC$

Langkah-langkah solusi

#2 Tulis ekspresi SOP nya

$$x = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

#3 Sederhanakan Ekspresi outputnya

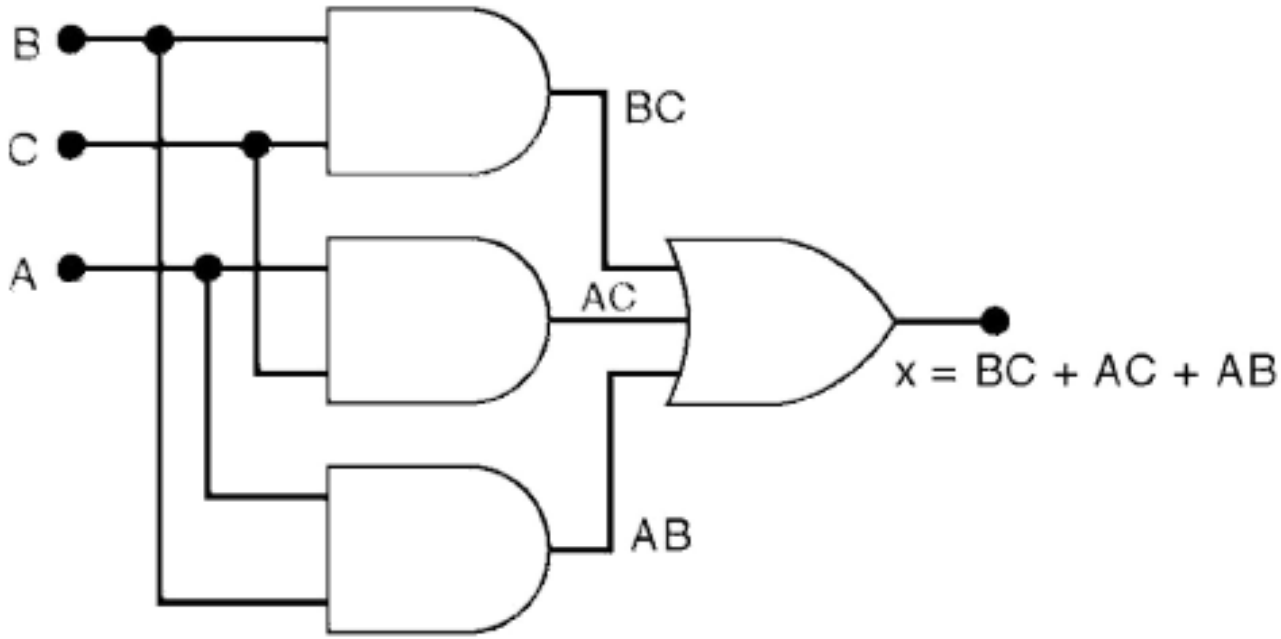


$$x = BC(\bar{A} + A) + AC(\bar{B} + B) + AB(\bar{C} + C)$$

$$x = BC + AC + AB$$

Langkah-langkah solusi

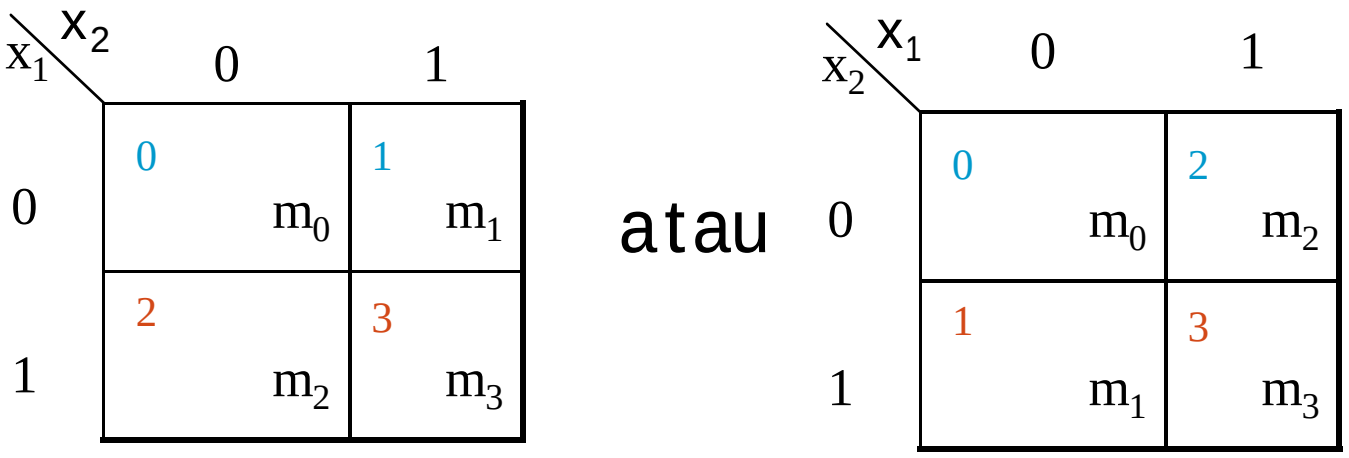
#4 Merancang Rangkaian Gerbang logikanya



Peta Karnaugh (K-Map)

Metode grafik untuk menyederhanakan ekspresi logika atau tabel kebenaran

Dapat digunakan dengan banyak variabel masukan, tetapi dalam praktiknya terbatas pada 5-6 variabel saja



Metode K-Map

1. Nilainilai tabel kebenaran diletakkan pada Kmap.
2. Kotakkotak Kmap yang berdekatan secara horisontal dan vertikal **hanya berbeda 1 variabel.**
3. Pola dari atas ke bawah atau kiri ke kanan *harus* berbentuk A B, AB, AB, AB
4. Bentuk **SOP bisa didapatkan dengan melakukan** operasi **OR pada semua *term (AND) dari kotak yang* bernilai 1**

Setiap kotak di baris **paling atas dianggap** berdekatan dengan kotakkotak pada baris **paling bawah, samping juga**

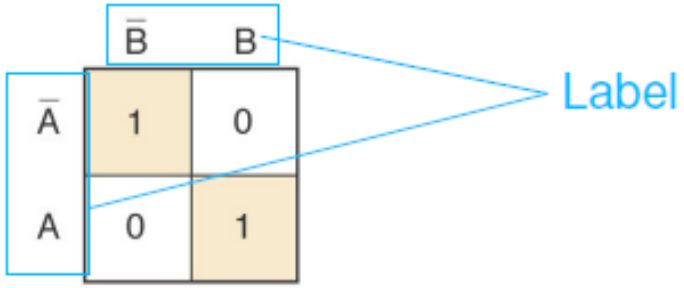
Contoh penggunaan K-MAP

2 variabel

A	B	X
0	0	1 → $\bar{A}\bar{B}$
0	1	0
1	0	0
1	1	1 → AB

$$x = \bar{A}\bar{B} + AB$$

(a)

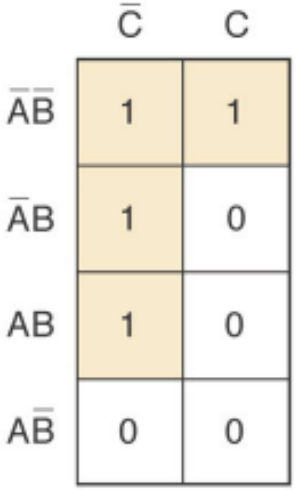


3 variabel

A	B	C	X
0	0	0	1 → $\bar{A}\bar{B}\bar{C}$
0	0	1	1 → $\bar{A}\bar{B}C$
0	1	0	1 → $\bar{A}B\bar{C}$
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1 → $AB\bar{C}$
1	1	1	0

$$X = \bar{A}B\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$$

(b)



Contoh penggunaan K-MAP

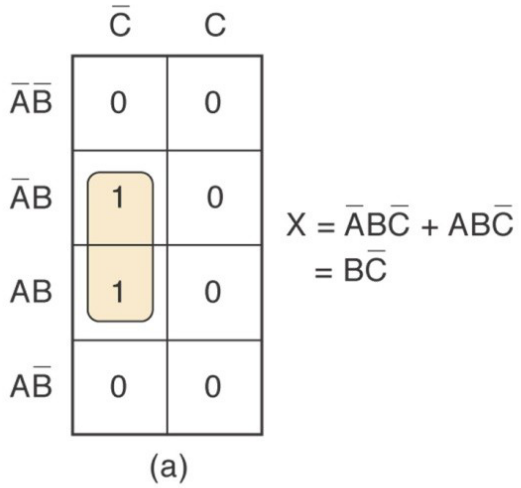
4 variabel

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\rightarrow \bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
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	1	$\rightarrow AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	$\rightarrow ABCD$

$$X = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$$

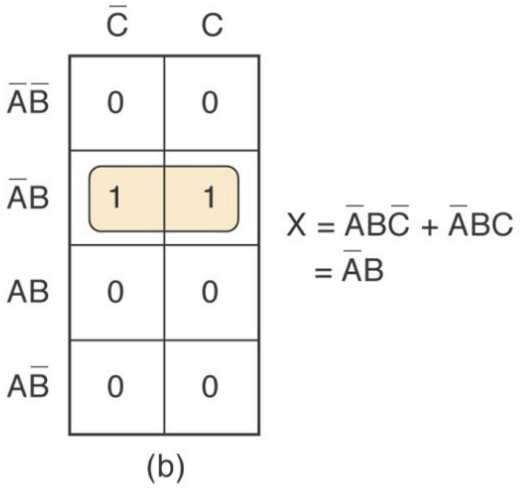
	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

Looping > penggabungan kotak yang bernilai 1



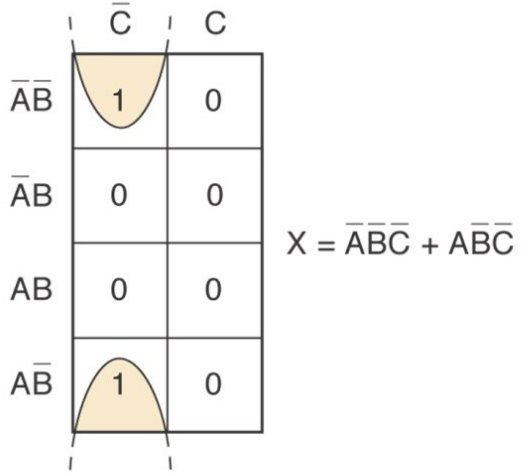
(a)

$X = \bar{A}B\bar{C} + AB\bar{C}$
$= B\bar{C}$



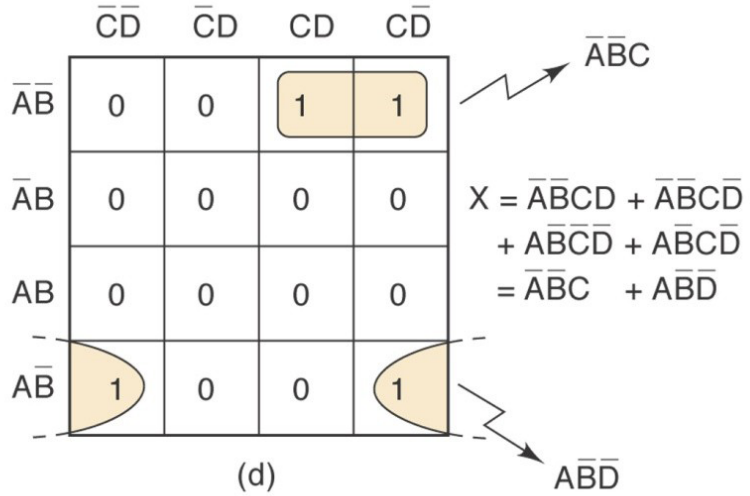
(b)

$X = \bar{A}B\bar{C} + \bar{A}BC$
$= \bar{A}B$



(c)

$X = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} = \bar{B}\bar{C}$



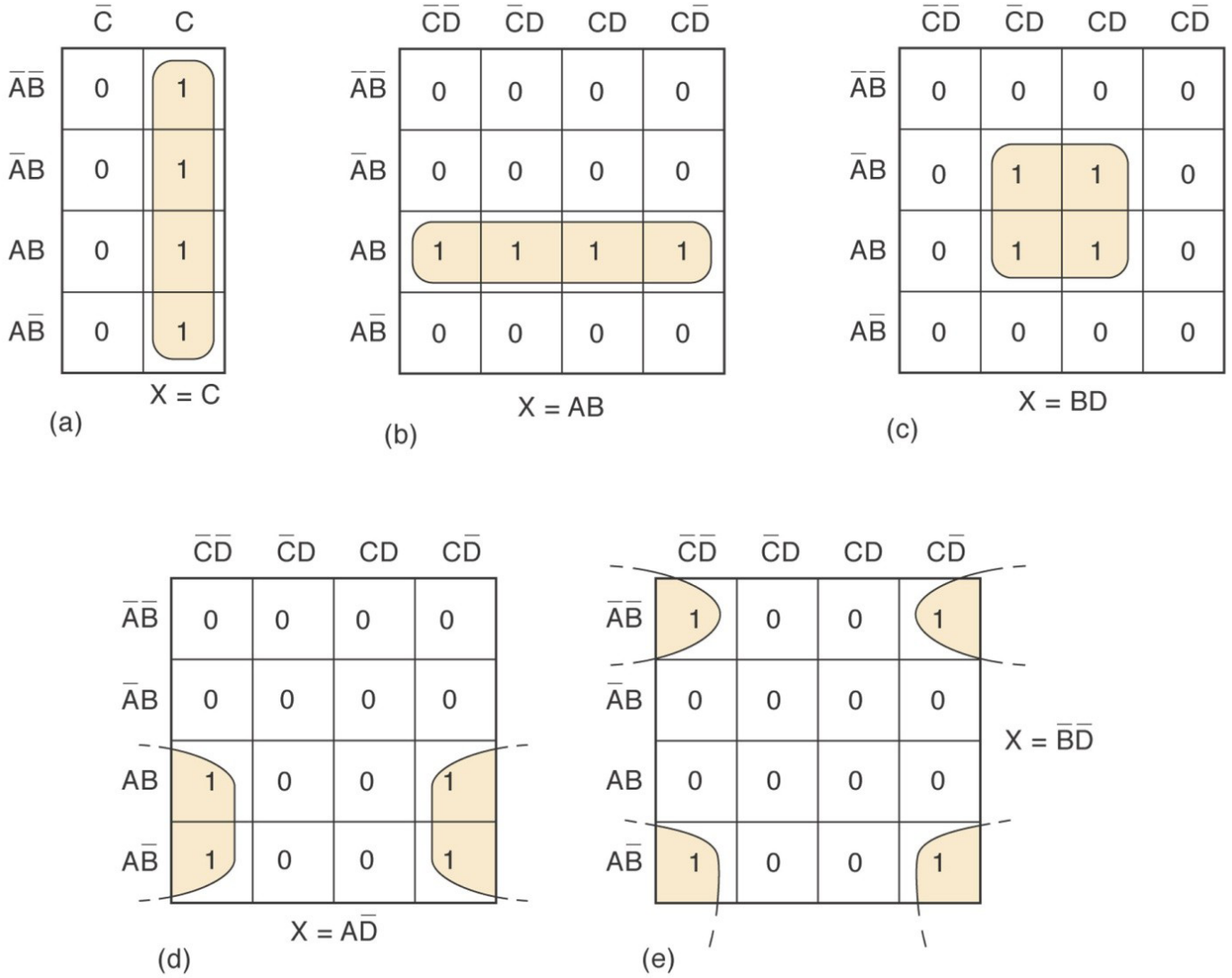
(d)

$X = \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D}$
$+ \bar{A}B\bar{C}D + A\bar{B}C\bar{D}$
$= \bar{A}\bar{B}C + A\bar{B}\bar{D}$

Aturan Looping

Proses *looping 2 kotak bernilai 1 yang berdekatan* dalam K-Map (**pasangan), akan menghilangkan 1 variabel yang muncul dalam bentuk normal dan komplemennya.**

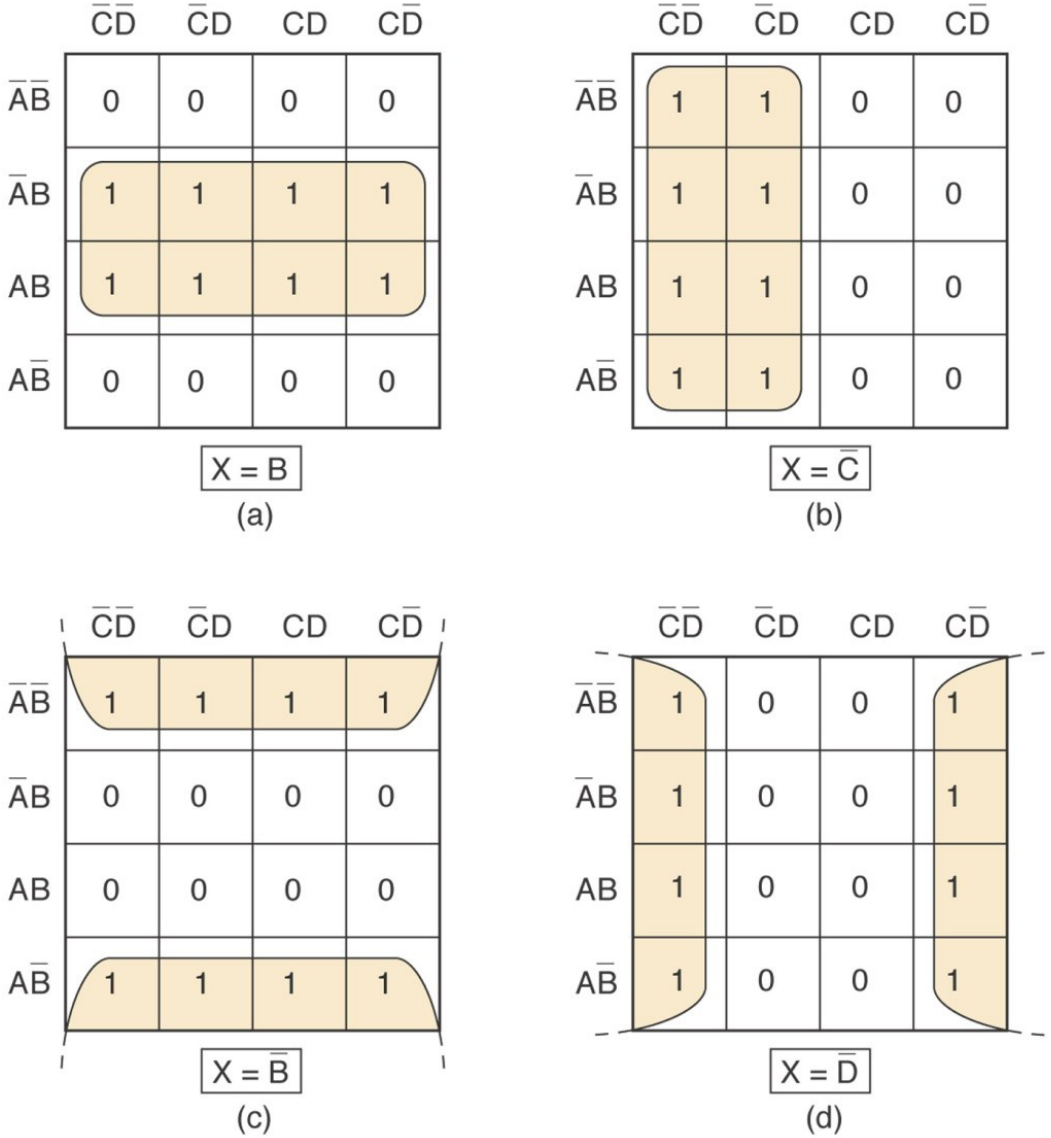
QUAD



Aturan Looping

Proses *looping kotak bernilai 1 berjumlah 4 buah* yang berdekatan dalam KMap (***quad), akan menghilangkan 2 variabel yang muncul dalam*** bentuk **normal dan komplemennya.**

OCTET


$\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$
$\bar{A}\bar{B}$ 0 0 0 0
$\bar{A}B$ 1 1 1 1
AB 1 1 1 1
$A\bar{B}$ 0 0 0 0
$X = B$
(a)
$\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$
$\bar{A}\bar{B}$ 1 1 0 0
$\bar{A}B$ 1 1 0 0
AB 1 1 0 0
$A\bar{B}$ 1 1 0 0
$X = \bar{C}$
(b)
$\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$
$\bar{A}\bar{B}$ 1 1 1 1
$\bar{A}B$ 0 0 0 0
AB 0 0 0 0
$A\bar{B}$ 1 1 1 1
$X = \bar{B}$
(c)
$\bar{C}\bar{D}$ $\bar{C}D$ CD $C\bar{D}$
$\bar{A}\bar{B}$ 1 0 0 1
$\bar{A}B$ 1 0 0 1
AB 1 0 0 1
$A\bar{B}$ 1 0 0 1
$X = \bar{D}$
(d)

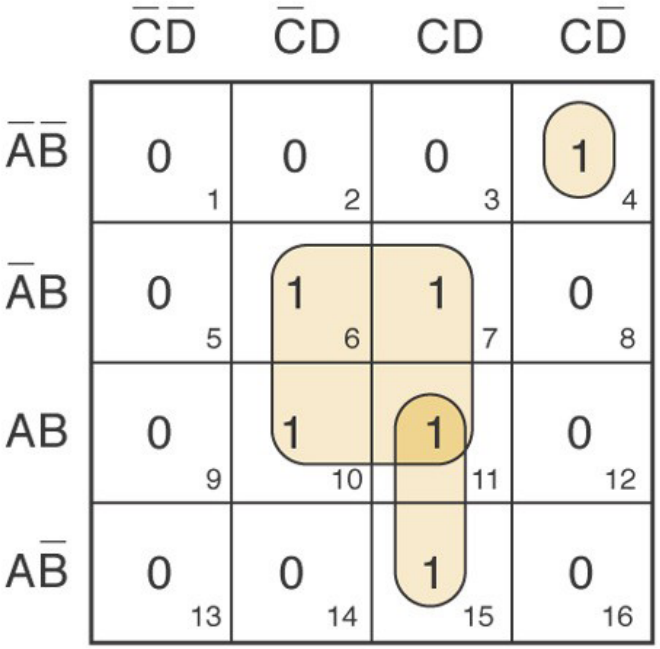
Penyederhanaan K-Maps

1. Buat KMap dan letakkan nilainilai 1 dan 0 pada kotak-kotak sesuai dengan tabel kebenaran.
2. Cari kotak bernilai 1 yang tidak berdekatan dengan kotak bernilai 1 lainnya, dan lakukan proses *looping (isolated 1.)*
3. Cari kotak bernilai 1 yang berdekatan dengan hanya 1 kotak bernilai 1 lainnya (pasangan) dan lakukan proses *looping.*
4. Cari kotakkotak bernilai 1 yang dapat dilakukan proses *looping*
5. *octet, walaupun sudah dilakukan proses looping padanya.*
6. Cari kotak bernilai 1 yang dapat dilakukan proses *looping quad.*
7. Cari kotakkotak bernilai 1 yang belum dilakukan proses *looping.*
8. Bentuk operasi OR untuk semua *term yang dihasilkan dari* setiap proses *looping. (SOP)*

Minimalkan penggunaan jumlah *loop.*

Penggunaan K-MAP

$X(A,B,C,D)=m(2, 5, 7, 11, 13,15)$



(a)

$$X = \underbrace{\bar{A}\bar{B}C\bar{D}}_{\text{loop 4}} + \underbrace{ACD}_{\text{loop 11, 15}} + \underbrace{BD}_{\text{loop 6, 7, 10, 11}}$$

Don’t Care (x)

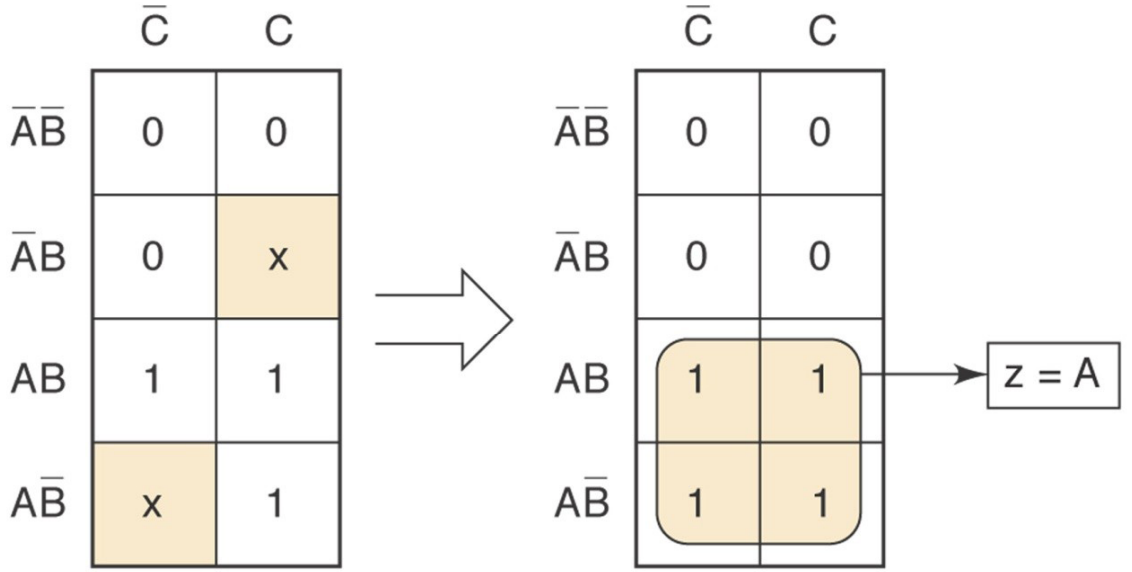
Pada beberapa rancangan rangkaian logika, terdapat kondisi masukan yang nilai keluarannya tidak ditentukan.

Tidak peduli dengan nilai keluaran dari beberapa masukan tersebut, TINGGI atau RENDAH.

Contoh Don’t Care

A	B	C	z	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	x	"don't
1	0	0	x	care"
1	0	1	1	
1	1	0	1	
1	1	1	1	

(a)



(b) (c)

Soal Latihan

Kerjakan di kertas terpisah

Contoh Don’t Care

1. Sederhanakan fungsi rangkaian Logika berikut:
   a. F= ABC+AB’C+A
   b. G=A’B’C’+A’BC+ABC+AB’C’+AB’C
   c. H=(C+D)’+A’CD’+ABC+A’B’CD+ACD’
2. Sederhanakan dan gambar rangkaian dari dengan K-Maps:
   a. f(ABC)=m(1,2, 3,5,7)
   b. g(ABCD)=m(0,1,6,7,8,9, 14,15)
   c. h(ABCD)=M(1,5,6,9,13,14)
   d.i(ABCD)=m(2,3,7,9,13,14) , d(6,10)