



ALJABAR BOOLEAN

IK-130
LOGIKA INFORMATIKA

Ani Anisyah, M.T.

OVERVIEW MATERI

- Definisi
- Aljabar Boolean Dua Nilai
- Ekspresi Boolean
- Prinsip Dualitas
- Hukum Aljabar Boolean
- Fungsi Boolean
- Bentuk Kanonik
- Konversi Antar Bentuk Kanonik
- Rangkaian Digital
- Penyederhanaan Fungsi Boolean
- Peta Karnaugh

PENGANTAR

- Aljabar Boolean ditemukan oleh George Boole, pada tahun 1854
- Boole menyatakan bahwa himpunan dan logika proposisi mempunyai sifat-sifat yang serupa (perhatikan kemiripan hukum-hukum aljabar logika dan hukum-hukum aljabar himpunan).
- Dalam buku *The Laws of Thought*, Boole memaparkan aturan-aturan dasar logika.
- Aturan dasar logika ini membentuk struktur matematika yang disebut **aljabar Boolean**.
- Penggunaan Aljabar Boolean:
 1. Perancangan rangkaian sirkuit
 2. Dasar teknologi komputer digital (operasi pada komputer menggunakan operasi bit, 0 dan 1)
 3. Perancangan pensaklaran
 4. Rangkaian digital
 5. Rangkaian IC (integrated circuit)

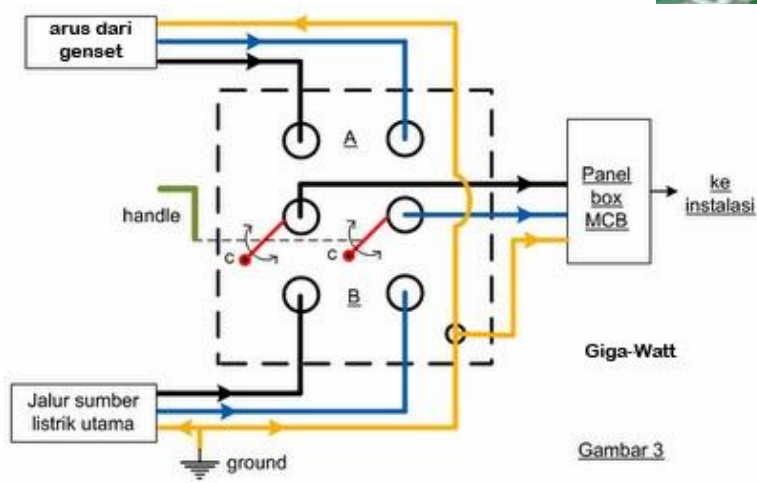
Penggunaan Aljabar Boolean



Papan score digital



IC (integrated circuit)



Jaringan/Rangkaian saklar

Pengantar Tipe Data Boolean

- Menggunakan tipe data Boolean → untuk data yang bertipe logika
- Tipe Boolean mempunyai dua buah nilai, yaitu true dan false
- Operasi pada tipe data Boolean :
 1. AND
 2. OR
 3. XOR
 4. NOT
- Operasi bit bersesuaian dengan operasi bit
 - Sebuah bit mempunyai nilai 1 atau 0
 - $1 \rightarrow \text{true}$, $0 \rightarrow \text{false}$



~ 0
 $1 \wedge 0$
 $0 \vee 0$
 $1 \oplus 0$

KEMIRIPAN HUKUM LOGIKA DAN HUKUM HIMPUNAN

Disebut juga **hukum-hukum aljabar proposisi**.

1. Hukum identitas: - $p \vee \mathbf{F} \Leftrightarrow p$ - $p \wedge \mathbf{T} \Leftrightarrow p$	2. Hukum <i>null</i> /dominasi: - $p \wedge \mathbf{F} \Leftrightarrow \mathbf{F}$ - $p \vee \mathbf{T} \Leftrightarrow \mathbf{T}$
3. Hukum negasi: - $p \vee \sim p \Leftrightarrow \mathbf{T}$ - $p \wedge \sim p \Leftrightarrow \mathbf{F}$	4. Hukum idempoten: - $p \vee p \Leftrightarrow p$ - $p \wedge p \Leftrightarrow p$
5. Hukum involusi (negasi ganda): - $\sim(\sim p) \Leftrightarrow p$	6. Hukum penyerapan (absorpsi): - $p \vee (p \wedge q) \Leftrightarrow p$ - $p \wedge (p \vee q) \Leftrightarrow p$
7. Hukum komutatif: - $p \vee q \Leftrightarrow q \vee p$ - $p \wedge q \Leftrightarrow q \wedge p$	8. Hukum asosiatif: - $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$ - $p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$
9. Hukum distributif: - $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ - $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	10. Hukum De Morgan: - $\sim(p \wedge q) \Leftrightarrow \sim p \vee \sim q$ - $\sim(p \vee q) \Leftrightarrow \sim p \wedge \sim q$

1. Hukum identitas: - $A \cup \emptyset = A$ - $A \cap U = A$	2. Hukum <i>null</i> /dominasi: - $A \cap \emptyset = \emptyset$ - $A \cup U = U$
3. Hukum komplemen: - $A \cup \bar{A} = U$ - $A \cap \bar{A} = \emptyset$	4. Hukum idempoten: - $A \cup A = A$ - $A \cap A = A$
5. Hukum involusi: - $\overline{(\bar{A})} = A$	6. Hukum penyerapan (absorpsi): - $A \cup (A \cap B) = A$ - $A \cap (A \cup B) = A$
7. Hukum komutatif: - $A \cup B = B \cup A$ - $A \cap B = B \cap A$	8. Hukum asosiatif: - $A \cup (B \cup C) = (A \cup B) \cup C$ - $A \cap (B \cap C) = (A \cap B) \cap C$
9. Hukum distributif: - $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ - $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	10. Hukum De Morgan: - $\overline{A \cap B} = \bar{A} \cup \bar{B}$ - $\overline{A \cup B} = \bar{A} \cap \bar{B}$
11. Hukum 0/1 (hukum komplemen 2) - $\bar{\emptyset} = U$ - $\bar{U} = \emptyset$	

Definisi

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 $(B, +, \cdot, ', 0, 1)$**
- disebut aljabar Boolean jika untuk setiap **$a, b, c \in B$** berlaku aksioma-aksioma atau **postulat Huntington** berikut:

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

Untuk setiap $a \in B$ terdapat elemen unik $a' \in B$ sehingga

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

Catatan:

- Operator $+$ \rightarrow Penjumlahan
- Operator \cdot \rightarrow Perkalian
- Operator $'$ \rightarrow Komplemen
- 0 = elemen terkecil (zero)
- 1 = elemen terbesar (unit)

Definisi

- Berhubung elemen-elemen B tidak didefinisikan nilainya (kita bebas menentukan anggota-anggota B), maka terdapat banyak sekali aljabar boolean.
- Aljabar himpunan dan aljabar logika proposisi \rightarrow termasuk aljabar Boolean
- Aljabar himpunan dan aljabar proposisi adalah himpunan bagian (subset) dari aljabar boolean

Definisi

Pada aljabar proposisi misalnya:

- B berisi semua proposisi dengan n peubah.
- dua elemen unik berbeda dari B adalah **T** dan **F**,
- operator biner: \vee dan \wedge , operator uner: \sim
- semua aksioma pada definisi di atas dipenuhi

Dengan kata lain $\langle B, \vee, \wedge, \sim, \mathbf{F}, \mathbf{T} \rangle$ adalah aljabar Boole

Definisi

Untuk mempunyai sebuah aljabar Boolean, yang harus diperlihatkan:

1. elemen-elemen himpunan B ,
2. kaidah/aturan operasi untuk dua operator biner dan operator uner,
3. himpunan B , bersama-sama dengan dua operator tersebut, memenuhi keempat **postulat Huntington**

Perbedaan antara aljabar booelan dan aljabar biasa

1. Aljabar booelan tidak memiliki kebalikan perkalian (multiple inverse) dan kebalikan penjumlahan; sehingga **tidak ada operasi pembagian dan pengurangan dalam aljabar Boolean.**
2. **Operator komplemen hanya tersedia pada aljabar Boolean**
3. Aljabar biasa memperlakukan himpunan bilangan riil, sedangkan aljabar Boolean memperlakukan himpunan B yang sampai sekarang belum didefinisikan, tetapi biasanya menggunakan **aljabar booelan dua nilai (0 dan 1)**

Aljabar Booelan Dua Nilai

- Aljabar booelan dua-nilai didefinisikan pada himpunan B dengan dua buah elemen 0 dan 1
- Pada aljabar 2-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

- Keempat postulat Huntington di atas dipenuhi

Pengecekan Aljabar Booelan Dua Nilai

- Identitas: jelas berlaku karena dari tabel dapat kita lihat bahwa:

$$(i) \quad 0 + 1 = 1 + 0 = 1$$

$$(ii) \quad 1 \cdot 0 = 0 \cdot 1 = 0$$

- Komutatif: jelas berlaku dengan melihat simetri tabel operator biner

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

1. Identitas

$$(i) \quad a + 0 = a$$

$$(ii) \quad a \cdot 1 = a$$

2. Komutatif

$$(i) \quad a + b = b + a$$

$$(ii) \quad a \cdot b = b \cdot a$$

3. Distributif

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

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

4. Komplemen

Untuk setiap $a \in B$ terdapat elemen unik $a' \in B$ sehingga

$$(i) \quad a + a' = 1$$

$$(ii) \quad a \cdot a' = 0$$

Pengecekan Aljabar Booelan Dua Nilai

- Distributif: (i) $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$ dapat ditunjukkan benar dari tabel operator biner 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

Pengecekan Aljabar Booelan Dua Nilai

- Hukum distributif (ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$ dapat ditunjukkan benar dengan membuat tabel kebenaran dengan cara yang sama seperti (i)
- Komplemen:
 - (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 keempat postulat Huntington dipenuhi, maka terbukti bahwa $B = \{0,1\}$ bersama-sama dengan operator biner $+$ dan \cdot operator komplemen $'$ merupakan aljabar Boolean

Ekspresi Boolean

Ekspresi Boolean dibentuk dari elemen-elemen B dan/atau peubah-peubah yang dapat dikombinasikan satu sama lain dengan operator $+$, \cdot , dan $'$.

Contoh 1:

0

1

a

b

$a + b$

$a \cdot b$

$a' \cdot (b + c)$

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

Contoh : Mengevaluasi Ekspresi Boolean Dua- Nilai

- 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 ekuivalen (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 : Mengevaluasi Ekspresi Boolean Dua- Nilai

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

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

Contoh :

Pelihatkan bahwa $a + a'b = a + b$

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

Catatan

- Dalam penulisan ekspresi Boolean selain tanda kurung (), operator ' (komplemen) mempunyai prioritas lebih tinggi daripada operator + dan ·
- Sebagai contoh:
 - $a + b \cdot c \rightarrow a + (b \cdot c)$, bukan $(a+b) \cdot c$
 - $a \cdot b' \rightarrow a \cdot (b')$, bukan $(a \cdot b)'$
- Untuk menyederhanakan penulisan, kita boleh tidak menuliskan notasi · pada operasi perkalian. Contoh :

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

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

$$(iii) \quad a \cdot 0, \text{ bukan } a0$$

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
 - \cdot 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.

$$(i) \quad (a \cdot 1)(0 + a') = 0 \quad \text{dualnya} \quad (a + 0) + (1 \cdot a') = 1$$

$$(ii) \quad a(a' + b) = ab \quad \text{dualnya} \quad a + a'b = a + b$$

Hukum-hukum Aljabar Boolean

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

Hukum aljabar boolelan dengan hukum aljabar himpunan dan hukum logika

- \cup dengan $+$ atau \vee dengan $+$
- \cap dengan \cdot atau \wedge dengan \cdot
- U dengan 1 atau T dengan 1
- \emptyset dengan 0 atau F dengan 0

Contoh

Contoh 2: Buktikan bahwa untuk sembarang elemen a dan b dari aljabar Boolean maka kesamaan berikut:

$$a + a'b = a + b \quad \text{dan} \quad a(a' + b) = ab$$

adalah benar.

Penyelesaian:

(i) $a + a'b = (a + ab) + a'b$	(Hukum Penyerapan)
$= a + (ab + a'b)$	(Hukum Asosiatif)
$= a + (a + a')b$	(Hukum Distributif)
$= a + 1 \cdot b$	(Hukum Komplemen)
$= a + b$	(Hukum Identitas)

Contoh

Contoh 2: Buktikan bahwa untuk sembarang elemen a dan b dari aljabar Boolean maka kesamaan berikut:

$$a + a'b = a + b \quad \text{dan} \quad a(a' + b) = ab$$

adalah benar.

Penyelesaian:

(i) $a + a'b = (a + ab) + a'b$	(Hukum Penyerapan)
$= a + (ab + a'b)$	(Hukum Asosiatif)
$= a + (a + a')b$	(Hukum Distributif)
$= a + 1 \cdot b$	(Hukum Komplemen)
$= a + b$	(Hukum Identitas)

(ii) $a(a' + b) = a a' + ab$	(Hukum Distributif)
$= 0 + ab$	(Hukum Komplemen)
$= ab$	(Hukum Identitas)

atau (ii) adalah dual dari (i)

Tugas

- Pelajari tentang gerbang logika
 - Simbol
 - Operasi
 - Tabel kebenaran
- Carilah tentang aplikasi aljabar Boolean
- Buat pembahasan mengenai materi tersebut pada PPT
- Tuliskan referensi pada PPT
- Tugas dikumpulkan minggu depan di SPOT (deadline, 18 Okt 2021 09:00)

Fungsi Boolean

- Fungsi Boolean = fungsi biner, pemetaan dari $f: B^n \rightarrow B$
- Contoh-contoh fungsi Boolean:

$$f(x) = x$$

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

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

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

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

- Setiap peubah di dalam fungsi Boolean, termasuk dalam bentuk komplemennya, disebut **literal**.
- Fungsi $h(x, y, z) = xyz'$ terdiri dari 3 buah literal, yaitu x , y , dan z' .
- Jika diberikan $x = 1$, $y = 1$, $z = 0$, maka nilai fungsinya:

$$h(1, 1, 0) = 1 \cdot 1 \cdot 0' = (1 \cdot 1) \cdot 1 = 1 \cdot 1 = 1$$

Fungsi Boolean dengan Tabel Kebenaran

- Untuk fungsi dengan n peubah, kombinasi dari nilai peubahnya sebanyak 2^n
- Cara membuat kombinasi tabel kebenaran:
 - Untuk peubah pertama, isi 4 baris pertama diisi dengan 0 dan 4 baris terakhir diisi dengan 1
 - Untuk peubah kedua, isi 2 baris pertama dengan 0 dan 2 baris berikutnya dengan 1, 2 baris berikutnya diisi dengan 0 dan 2 baris terakhir diisi dengan 1
 - Untuk peubah ketiga, isi kolom ketiga secara selang-seling dengan 0 dan 1 mulai baris pertama sampai baris terakhir

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

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

Fungsi Boolean dengan Tabel Kebenaran

- Fungsi Boolean yang ekspresinya berbeda dapat menyatakan fungsi yang sama

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

Dan

$$g(x, y, z) = x'z + xy'$$

x	y	z	z'	$x'y'z + x'yz + xy'$	$x'z + xy'$
0	0	0	1	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	0	0	0

Penjumlahan dan Perkalian Dua Fungsi

- Misalkan f dan g adalah dua buah fungsi Boolean dengan n peubah, maka penjumlahan $f + g$ didefinisikan sebagai:

$$(f + g)(x_1 + x_2 + x_3 + \dots + x_n) = f(x_1 + x_2 + x_3 + \dots + x_n) + g(x_1 + x_2 + x_3 + \dots + x_n)$$

- Sedangkan perkalian $(f \cdot g)$ didefinisikan sebagai:

$$(f \cdot g)(x_1 + x_2 + x_3 + \dots + x_n) = f(x_1 + x_2 + x_3 + \dots + x_n) \cdot g(x_1 + x_2 + x_3 + \dots + x_n)$$

- Contoh:

$$f(x, y) = xy' + y \text{ dan } g(x, y) = x' + y' \text{ maka } h(x, y) = f + g = xy' + y + x' + y'$$

Yang bila disederhanakan lebih lanjut

$$h(x, y) = f + g = xy' + y + x' + y' = xy' + x' + (y + y') = xy' + x' + 1$$

dan

$$i(x, y) = f \cdot g = (xy' + y)(x' + y')$$

Komplemen Fungsi Boolean

- Fungsi komplemen dari suatu fungsi $f = f'$
- Fungsi komplemen dapat dicari dengan dua cara berikut:

1. Cara Pertama : Menggunakan hukum De Morgan

- Dua peubah x_1 dan x_2 adalah

(i) $(x_1 + x_2)' = x_1' x_2'$ dan dualnya

(ii) $(x_1 \cdot x_2)' = x_1' + x_2'$

- Tiga peubah x_1, x_2, x_3 adalah

$$\begin{aligned}
 (i) \quad (x_1 + x_2 + x_3)' &= (x_1 + y)' \quad \text{Dimana } y = x_2 + x_3 \\
 &= x_1' y' \\
 &= x_1' (x_2 + x_3)' \\
 &= x_1' x_2' x_3'
 \end{aligned}$$

Dan dualnya adalah

$$(x_1 x_2 x_3)' = x_1' + x_2' + x_3'$$

Komplemen Fungsi Boolean

- Fungsi komplemen dari suatu fungsi $f = f'$
- Fungsi komplemen dapat dicari dengan dua cara berikut:

2. Cara Kedua : Menggunakan prinsip dualitas

Tentukan **dual** dari ekspresi Boolean yang direpresentasikan f , lalu **komplemenkan** setiap literal di dalam dual tersebut. Bentuk akhir yang diperoleh menyatakan fungsi komplemen

Contoh : Komplemen Fungsi Boolean

- Misalkan $f(x, y, z) = x(y'z' + yz)$ maka fungsi komplemennya adalah

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

Contoh : Komplemen Fungsi Boolean

- Misalkan $f(x, y, z) = x(y'z' + yz)$ maka **dual dari ekspresi** Booleannya adalah

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

Komplemenkan tiap literal dari dual di atas menjadi

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

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

Bentuk Kanonik

- Ekspresi Boolean yang menspesifikasikan suatu fungsi dapat disajikan dalam dua bentuk berbeda.
- Pertama, sebagai **penjumlahan dari hasil kali** dan kedua sebagai **perkalian dari hasil jumlah**.

Contoh 3:

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

dan

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

adalah dua buah fungsi yang sama.

Bentuk Kanonik

Minterm: suku (*term*) di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk **hasil kali**

Maxterm: suku (*term*) di dalam ekspresi boolean mengandung literal yang lengkap dalam bentuk **hasil jumlah**.

Contoh 4:

$$f(x, y, z) = x'y'z + xy'z' + xyz \rightarrow 3 \text{ buah minterm: } x'y'z, xy'z', xyz$$

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

$\rightarrow 5 \text{ buah maxterm: } (x + y + z), (x + y' + z), (x + y' + z'),$
 $(x' + y + z'), \text{ dan } (x' + y' + z)$

Bentuk Kanonik

Cara membentuk *minterm* dan *maxterm*:

Untuk *minterm*, **setiap peubah yang bernilai 0 dinyatakan dalam bentuk komplemen**, sedangkan peubah yang bernilai 1 dinyatakan tanpa komplemen.

Sebaliknya, untuk *maxterm*, **setiap peubah yang bernilai 0 dinyatakan tanpa komplemen**, sedangkan **peubah yang bernilai 1 dinyatakan dalam bentuk komplemen**.

Bentuk Kanonik

Cara membentuk *minterm* dan *maxterm* dari tabel kebenaran untuk dua peubah:

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

Bentuk Kanonik

Cara membentuk *minterm* dan *maxterm* dari tabel kebenaran untuk tiga peubah:

			<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

Bentuk Kanonik

Jika diberikan sebuah tabel kebenaran, kita dapat membentuk fungsi Boolean dalam bentuk kanonik (SOP atau POS) dari tabel tersebut dengan cara:

- mengambil *minterm* dari setiap **nilai fungsi yang bernilai 1 (untuk SOP)**

atau

- mengambil *maxterm* dari setiap **nilai fungsi yang bernilai 0 (untuk POS).**

Contoh 5: Tinjau 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 (dengan menggunakan lambang *minterm*),

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

Contoh 5: Tinjau 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

			<i>Minterm</i>		<i>Maxterm</i>	
x	y	z	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	$x'yz'$	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

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 (dengan menggunakan lambang *minterm*),

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

- 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

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,

$$f(x, y, z) = M_0 M_2 M_3 M_5 M_6 = \Pi(0, 2, 3, 5, 6)$$

- 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

			Minterm		Maxterm	
x	y	z	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

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,

$$f(x, y, z) = M_0 M_2 M_3 M_5 M_6 = \Pi(0, 2, 3, 5, 6)$$

Contoh 6: Nyatakan fungsi Boolean $f(x, y, z) = x + y'z$ dalam bentuk kanonik SOP dan POS.

Penyelesaian:

(a) SOP

Lengkapi terlebih dahulu literal untuk setiap suku agar jumlahnya sama.

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

$$= xy + xy'$$

$$= xy(z + z') + xy'(z + z')$$

$$= xyz + xyz' + xy'z + xy'z'$$

dan

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

			<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

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

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

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

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

(b) POS

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

Lengkapi terlebih dahulu literal pada setiap suku agar jumlahnya sama:

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

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

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

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

Contoh 7: Nyatakan fungsi Boolean $f(x, y, z) = xy + x'z$ dalam bentuk kanonik POS.

Penyelesaian:

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

Lengkapi literal untuk setiap suku agar jumlahnya sama:

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

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

atau $f(x, y, z) = M_0 M_2 M_4 M_5 = \prod (0, 2, 4, 5)$

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 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:

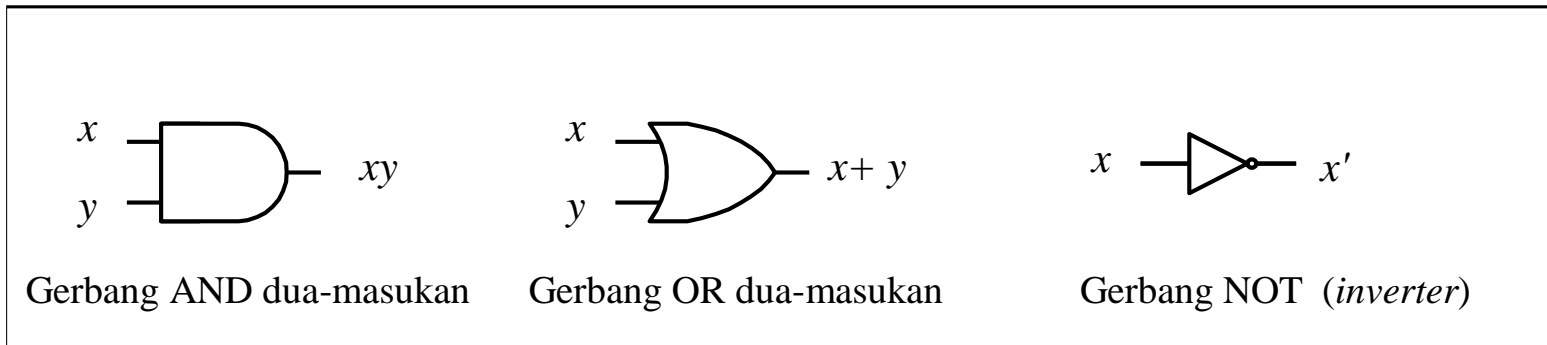
$$\begin{aligned} f(x, y, z) &= (f'(x, y, z))' = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' \\ &= (x'y'z')' (x'y z')' (x'y z)' \\ &= (x + y + z) (x + y' + z) (x + y' + z') \\ &= M_0 M_2 M_3 = \Pi (0, 2, 3) \end{aligned}$$

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

Kesimpulan: $m_j' = M_j$

Rangkaian Logika

- Fungsi Boolean dapat juga direpresentasikan dalam bentuk rangkaian logika.
- Ada tiga gerbang logika dasar: gerbang AND, gerbang OR, dan gerbang NOT

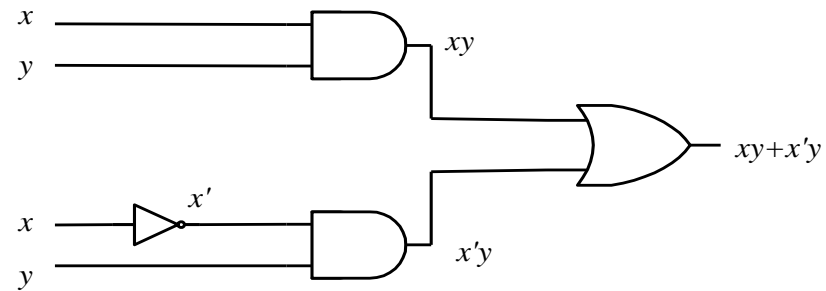


Rangkaian Logika

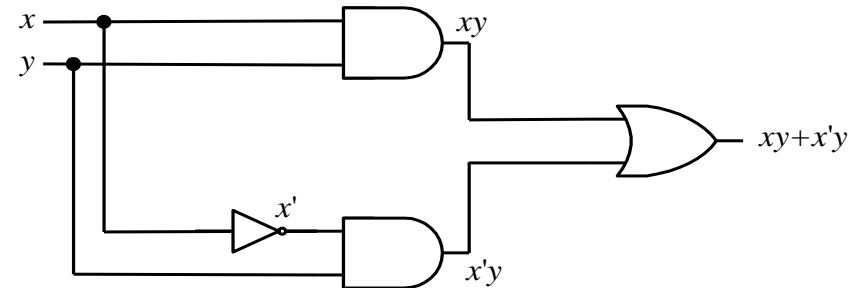
Contoh 8: Nyatakan fungsi $f(x, y, z) = xy + x'y$ ke dalam rangkaian logika.

Penyelesaian: Ada beberapa cara penggambaran

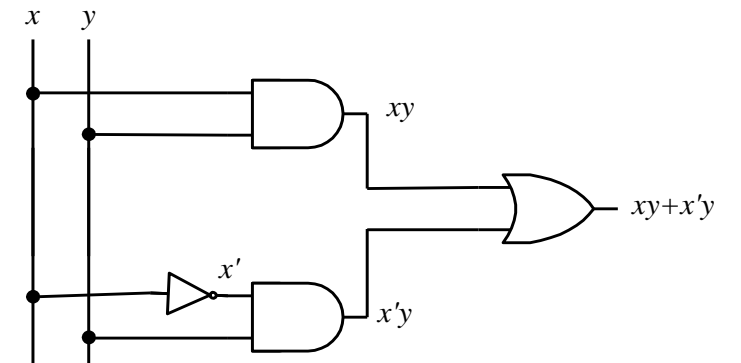
Cara pertama:



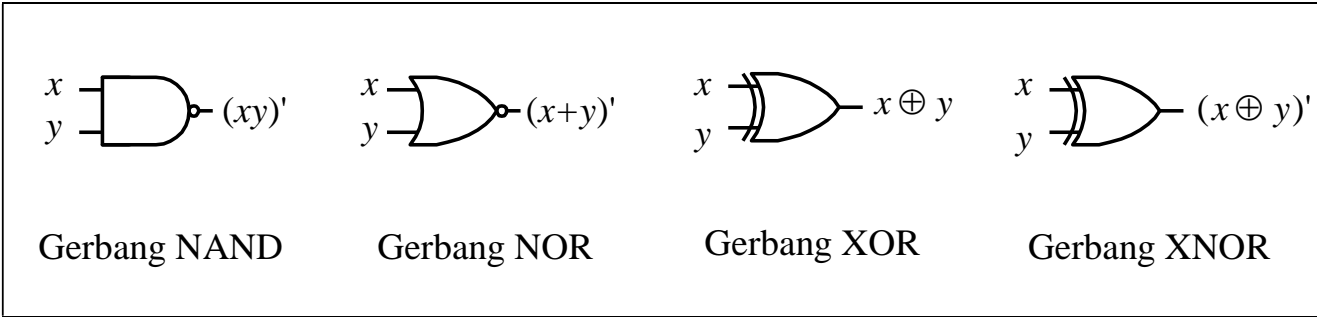
Cara kedua:



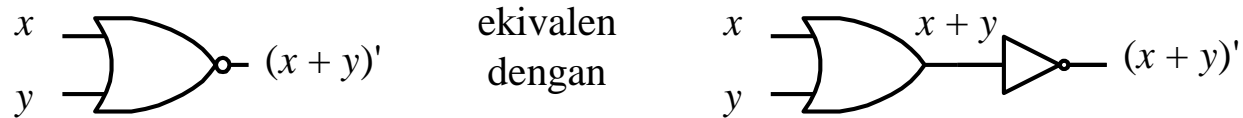
Cara ketiga:



Rangkaian Logika



Keempat gerbang di atas merupakan kombinasi dari gerbang-gerbang dasar, misalnya gerbang NOR disusun oleh kombinasi gerbang OR dan gerbang NOT:



Selain itu, dengan menggunakan hukum De Morgan, kita juga dapat membuat gerbang logika yang ekivalen dengan gerbang NOR dan NAND di atas:



Penyederhanaan Fungsi Boolean

Menyederhanakan fungsi Boolean artinya **mencari bentuk fungsi lain yang ekuivalen tetapi dengan jumlah literal atau operasi yang lebih sedikit.**

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

Dipandang dari segi aplikasi aljabar Boolean, fungsi Boolean yang lebih sederhana berarti rangkaian logikanya juga lebih sederhana (menggunakan jumlah gerbang logika lebih sedikit).

Penyederhanaan Fungsi Boolean

Tiga metode yang dapat digunakan untuk menyederhanakan fungsi Boolean:

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

Yang dibahas hanyalah **Metode Peta Karnaugh**

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 (berbentuk bujursangkar) yang bersisian.
- Tiap kotak **merepresentasikan sebuah *minterm***.
- Tiap kotak dikatakan bertetangga jika *minterm-minterm* yang merepresentasikannya **berbeda hanya 1 buah literal**.

Peta Karnaugh dengan dua peubah

m_0	m_1
m_2	m_3

Penyajian 1

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

Penyajian 2

	y'	y
x'	$x'y'$	$x'y$
x	xy'	xy

Penyajian 3

Peta Karnaugh 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'

Peta Karnaugh dengan empat peubah

				yz			
				00	01	11	10
m_0	m_1	m_3	m_2	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
m_4	m_5	m_7	m_6	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
m_{12}	m_{13}	m_{15}	m_{14}	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
m_8	m_9	m_{11}	m_{10}	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

Pengisian peta Karnaugh dari tabel kebenaran

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	1
1	1	0	0
1	1	1	1

Tinjau hanya nilai fungsi yang memberikan 1. Fungsi Boolean yang merepresentasikan tabel kebenaran adalah $f(x, y, z) = x'y'z + xy'z' + xy'z + xyz$.

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

Teknik Minimisasi Fungsi Boolean dengan Peta Karnaugh

- 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),
 - oktet (delapan).

Pasangan

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

Bukti secara aljabar:

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

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

Sesudah disederhanakan: $f(w, x, y, z) = wxy$

Kuad (1)

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

Bukti secara aljabar (kuad = 2 buah pasangan):

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

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

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

Kuad (2)

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

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

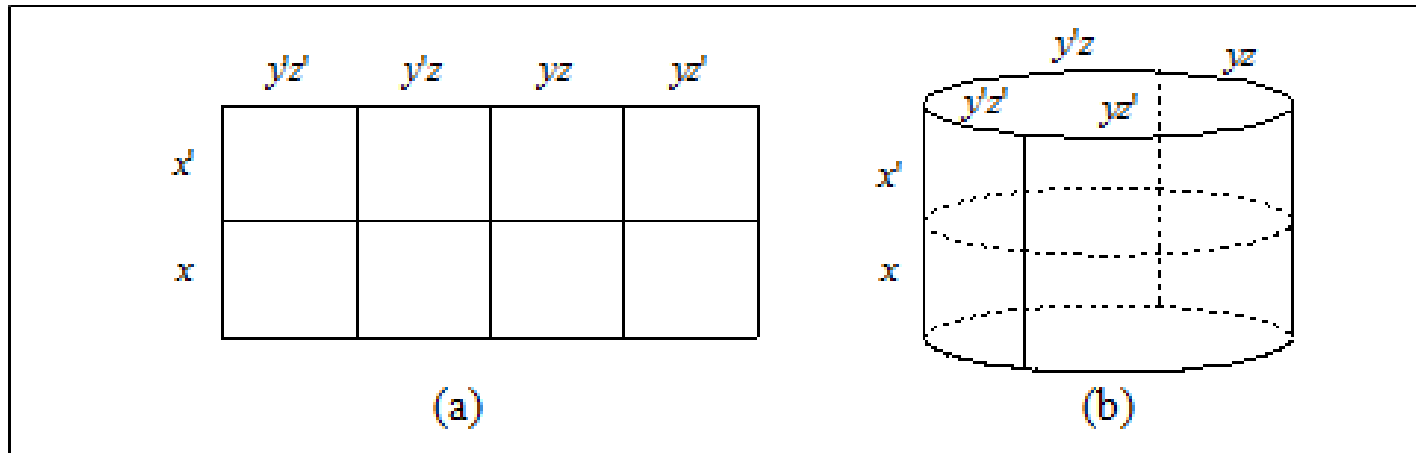
Oktet

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

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

Sesudah: $f(w, x, y, z) = w$

Penggulungan (1)



Gambar (a) Peta Karnaugh "normal" dengan 3 peubah
 (b) Peta Karnaugh dengan sisi kiri dan sisi kanan ditautkan (seperti digulung).

Penggulungan (2)

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

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

Sebelum: $f(x, y, z) = x'yz + xy'z' + xyz + xyz'$

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

Tips menyederhanakan dengan Peta Karnaugh

- Kelompokkan 1 yang bertetangga sebanyak mungkin
- Dimulai dengan mencari oktet sebanyak-banyaknya terlebih dahulu, kemudian kuad, dan terakhir pasangan.

Contoh minimisasi 1:

		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

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

Contoh minimisasi 2:

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

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

Contoh minimisasi 3:

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

Hasil penyederhanaan: $f(w, x, y, z) = wx + wz + wy + xyz$

Contoh minimisasi 4:

Tentukan bentuk sederhana dari fungsi Boolean yang merepresentasikan tabel kebenaran berikut dalam bentuk baku SOP dan bentuk baku POS.

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

Penyelesaian:

(a) Bentuk baku SOP: kelompokkan 1

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

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

(b) Bentuk baku POS: kelompokkan 0

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

Fungsi minimasi: $f(x, y, z) = (x' + z')(x + z)$

Contoh minimisasi 5:

Minimisasi fungsi Boolean $f(x, y, z) = \Sigma (0, 2, 4, 5, 6)$

Penyelesaian:

Peta Karnaugh untuk fungsi tersebut adalah:

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

			Minterm		Maxterm	
x	y	z	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

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

Contoh minimisasi 6:

Minimisasi $f(w, x, y, z) = w'x'y' + x'yz' + w'xyz' + wx'y'$

Penyelesaian:

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

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

Contoh minimisasi 8:

Sederhanakan fungsi $f(w,x,y,z) = (w + x')(w + x + y)(w' + x' + y')(w' + x + y + z')$. Hasil penyederhanaan dalam bentuk baku SOP dan POS.

Penyelesaian:

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

Hasil penyederhanaan

SOP: $f(w, x, y, z) = x'y + wxy' + wy'z'$ (garis penuh)

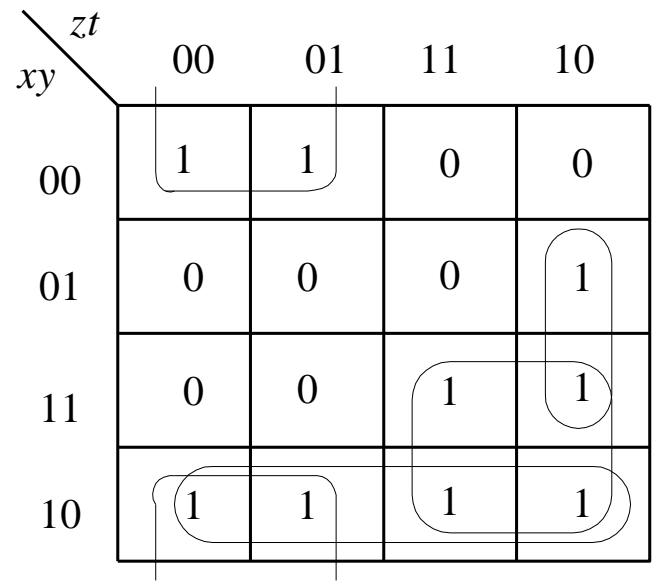
POS: $f(w, x, y, z) = (x' + y')(w + y)(x + y + z')$ (garis putus-putus)

Contoh minimisasi 9:

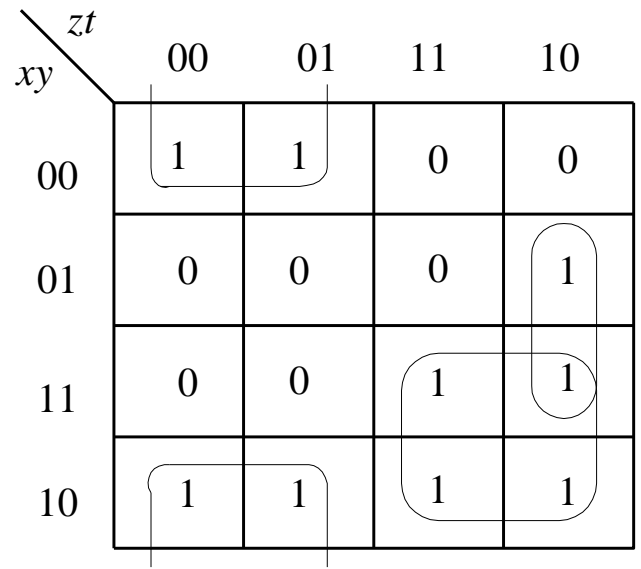
Sederhanakan fungsi $f(x, y, z, t) = xy' + xyz + x'y'z' + x'yz't'$

Penyelesaian:

Pengelompokan yang berlebihan



Pengelompokan yang benar



Fungsi minimasi: $f(x, y, z, t) = y'z' + xz + yzt'$

Contoh minimisasi 10:

Minimasi fungsi yang telah dipetakan ke peta Karnaugh di bawah ini dalam bentuk baku SOP dan bentuk baku POS

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

Penyelesaian:

$$\text{SOP : } f(w, x, y, z) = yz + wz + xz + w'xy'$$

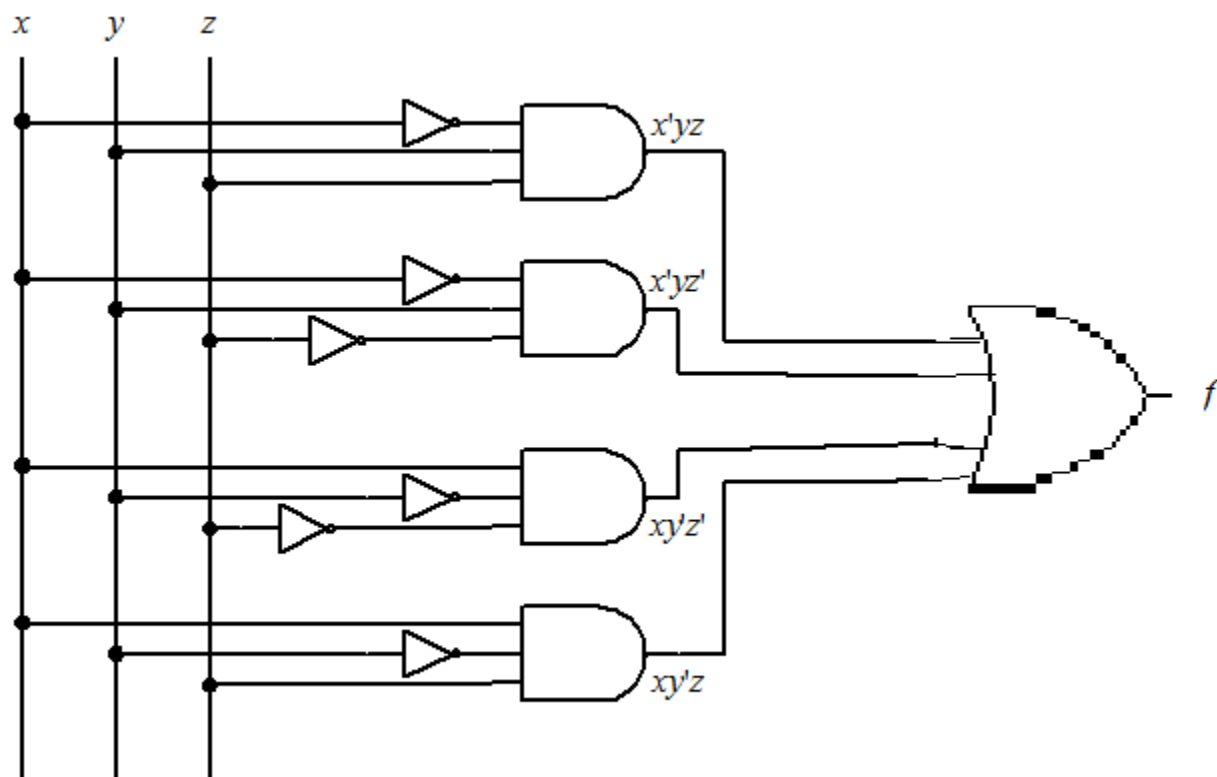
(garis penuh)

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

(garis putus-putus)

Contoh minimisasi 11:

Sederhanakan rangkaian logika berikut:



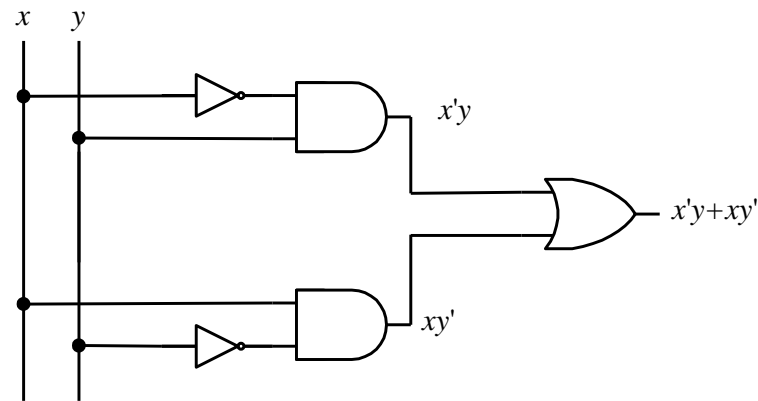
Penyelesaian: Fungsi yang berkoresponden dengan rangkaian logika tsb: $f(x, y, z) = x'yz + x'yz' + xy'z' + xy'z$

$x \backslash yz$	00	01	11	10
0	1	0	1	1
1	1	1	0	0

Fungsi Boolean hasil minimisasi:

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

Rangkaian logika hasil penyederhanaan:



Keadaan don't care

- 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.
- Contoh:
 - peraga digital angka desimal 0 sampai 9.
 - Jumlah bit yang diperlukan untuk merepresentasikan = 4 bit.
 - 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

} don't care

- Dalam menyederhanakan Peta Karnaugh yang mengandung keadaan *don't care*, ada dua hal penting sebagai pegangan.
- Pertama, kita anggap semua nilai *don't care* (X) sama dengan 1 dan kemudian membentuk kelompok sebesar mungkin yang melibatkan angka 1 termasuk tanda X tersebut.
- Kedua, semua nilai X yang tidak termasuk dalam kelompok tersebut kita anggap bernilai 0.
- Dengan cara ini, keadaan-keadaan X telah dimanfaatkan semaksimal mungkin, dan kita boleh melakukannya secara bebas.

Contoh: Sebuah fungsi Boolean, f , dinyatakan dengan tabel berikut. Minimisasi fungsi f sesederhana 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:

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

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

Contoh: Minimisasi fungsi Boolean berikut (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:

$wx \backslash yz$	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:

SOP: $f(w, x, y, z) = yz + w'z$

(kelompok garis penuh)

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

(kelompok garis putus-putus)

Perancangan Rangkaian Logika

1. *Majority gate* merupakan sebuah rangkaian digital yang keluarannya sama dengan 1 jika mayoritas masukannya bernilai 1 (mayoritas = 50% + 1). Keluaran sama dengan 0 jika tidak memenuhi hal tersebut di atas. Dengan bantuan tabel kebenaran, carilah fungsi Boolean yang diimplementasikan dengan *3-input majority gate*. Sederhanakan fungsinya, lalu gambarkan rangkaian logikanya.

Penyelesaian:

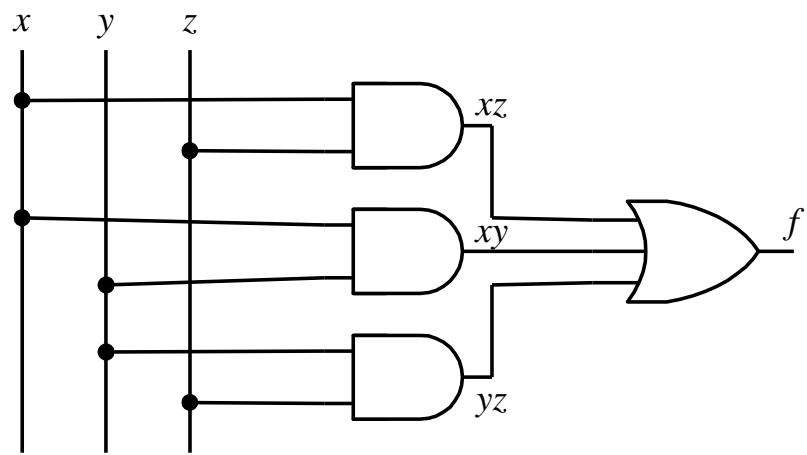
Tabel kebenaran:

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

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

$f(x, y, z) = xz + xy + yz$

Rangkaian logika:



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).

Penyelesaian:

Angka desimal: 0 .. 9 (direpresentasikan dalam 4 bit biner, misalkan $a_0a_1a_2a_3$).

Fungsi $f(a_0, a_1, a_2, a_3)$ bernilai 1 jika representasi desimal dari $a_0a_1a_2a_3$ menyatakan bilangan genap, dan bernilai 0 jika tidak genap.

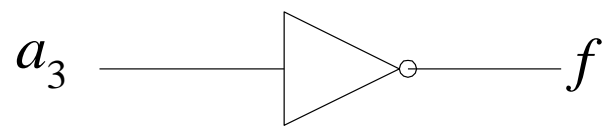
Tabel Kebenaran:

a_0	a_1	a_2	a_3	Desimal	$f(a_0, a_1, a_2, a_3)$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	2	1
0	0	1	1	3	0
0	1	0	0	4	1
0	1	0	1	5	0
0	1	1	0	6	1
0	1	1	1	7	0
1	0	0	0	8	1
1	0	0	1	9	0
1	0	1	0	10	X
1	0	1	1	11	X
1	1	0	0	12	X
1	1	0	1	13	X
1	1	1	0	14	X
1	1	1	1	15	X

a_2a_3	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

$f(a_0, a_1, a_2, a_3) = a_3'$

Rangkaian logika:



REFERENSI

1. Kenneth H. Rosen, *Discrete Mathematics and Application to Computer Science 5th Edition*, McGraw-Hill.
2. **Dr. Ir. Rinaldi Munir, M.T, *Matematika Diskrit (Edisi Keempat)*, Bandung: Informatika , 2013.**
3. Richard Johsonbaugh, *Discrete Mathematics*, Prentice-Hall, 1997