

# 2

## GERBANG dan ALJABAR BOOLE

Konsep dasar aljabar Boole (Boolean Algebra) telah diletakkan oleh seorang matematisi Inggris George Boole, pada tahun 1854. Konsep dasar itu membutuhkan waktu yang cukup lama untuk disadari kegunaannya, baik dalam bidang matematika maupun dalam bidang teknik.

Pada tahun 1938 Claude Shannon, seorang ahli komunikasi, memanfaatkan dan menyempurnakan konsep Boole tersebut. Sekarang ini, aljabar Boole memegang peranan yang sangat penting, tidak saja dalam logika, tetapi juga di bidang lain seperti teori peluang/kemungkinan, teori informasi/komunikasi, teori himpunan dan lain-lain. Teori ini juga dipakai dalam merancang komputer elektronik dengan menerjemahkannya ke dalam rangkaian saklar (switching circuits) yang pada dasarnya adalah logika, tertutup atau terbuka, mengalirkan arus listrik atau tidak.

### 2.1 Gerbang Dasar dan Tabel Kebenaran

Harga peubah (variabel) logika, pada dasarnya hanya dua, yaitu benar (true) atau salah (false). Dalam persamaan logika, umumnya simbol 1 dipakai untuk menyatakan benar dan simbol 0 dipakai untuk untuk menyatakan salah. Dengan memakai simbol ini, maka keadaan suatu logika hanya mempunyai dua kemungkinan, 1 dan 0. Kalau tidak 1, maka keadaan itu harus 0 dan kalau tidak 0 maka keadaan itu harus 1.

Operasi yang paling mendasar dalam logika adalah penyangkalan dengan kata-kata "*tidak*" (NOT). Jadi, "*benar*" adalah "*tidak salah*" dan "*salah*" adalah "*tidak benar*". Operasi ini dikenal secara umum dengan nama "inversion" yang disimbolkan dengan garis di atas peubah yang disangkal ataupun tanda petik (') di kanan-atas peubah itu. Dengan notasi ini, maka logika penyangkalan dapat dituliskan sebagai :

$$\bar{1} = 0 \text{ dan } \bar{0} = 1 \text{ atau: } 1' = 0 \text{ dan } 0' = 1$$

Gerbang elektronik yang berfungsi menidakkan ini disebut gerbang NOT dan sering juga disebut "inverter". Bila masukan gerbang NOT dinamakan A dan keluarannya dinamakan Z, maka hubungan masukan dan keluaran itu dituliskan sebagai:

$$Z = \bar{A} \text{ atau } Z = A' \quad (2.1)$$

Karena masukan A hanya dapat berkeadaan 0 atau 1, maka Z juga hanya dapat berkeadaan 1 atau 0. Keadaan keluaran Z untuk setiap keadaan masukannya dapat ditunjukkan dalam bentuk tabel yang disebut "tabel kebenaran" (truth table), yang sering juga disebut tabel kombinasi (combination table), sebagai berikut:

Tabel Kebenaran **NOT**

A	$Z = \bar{A}$
0	1
1	0

Dari pers. (2.1) di atas dapat dilihat, yang juga ditunjukkan dalam tabel kebenaran di atas, bahwa fungsi Z berkeadaan 1 bila A berkeadaan 0. Perhatikan juga bahwa fungsi dinyatakan untuk keadaan 1 dan peubah yang berkeadaan 0 di-NOT-kan (dikomplemenkan) untuk membuat  $Z = 1$ . Hal ini berlaku secara umum dalam aljabar Boole dan untuk peubah yang aktif untuk tegangan 0 Volt (rendah) sering diberi nama dengan garis komplemen di atasnya. Bentuk keluaran suatu rangkaian logika dalam bentuk fungsi Boole dapat diperoleh dengan mudah dari tabel kebenaran rangkaian logika yang bersangkutan. Tetapi fungsi yang dihasilkan dari tabel kebenaran umumnya belumlah dalam bentuk yang sederhana, yang membutuhkan gerbang yang paling sedikit, dan masih perlu disederhanakan. Penyederhanaan ini akan dibahas dalam bab-bab berikutnya.

Dua operasi yang paling mendasar lainnya dalam aljabar logika adalah operasi "DAN" (AND) dan operasi "ATAU" (OR). Gerbang elektronik yang merealisasikan logika ini masing-masing diberi nama gerbang "AND" dan gerbang "OR". Perlu ditegaskan kembali bahwa untuk logika positif yang dipakai seterusnya dalam buku ini, 1 diartikan benar dan 0 diartikan salah dan secara elektroniknya, 1 diartikan sebagai tegangan tinggi (paling umum adalah +5 Volt) dan 0 diartikan sebagai tegangan rendah (0 Volt). Tegangan elektronik 0 - 5 Volt ini dikenal sebagai level TTL, singkatan dari Transistor-Transistor Logic.

Untuk suatu gerbang OR dengan 2 masukan, katakanlah A dan B, keluarannya akan benar (= 1) bila salah satu masukan A "atau" B adalah benar dan keluaran itu akan salah (= 0) bila kedua masukan A dan B secara bersama-sama salah. Untuk gerbang AND dengan dua masukan A dan B, keluarannya akan benar hanya bila kedua masukannya A "DAN" B adalah benar dan salah bila salah satu masukan itu salah. Keterangan ini ditunjukkan lebih jelas oleh

tabel kebenaran pada Gambar 2.1.

masukan		keluaran
A	B	Z= A+B
0	0	0
0	1	1
1	0	1
1	1	1

(a)

masukan		keluaran
A	B	Z= A.B
0	0	0
0	1	0
1	0	0
1	1	1

(b)

Gambar 2.1. Tabel-tabel kebenaran gerbang OR dan AND

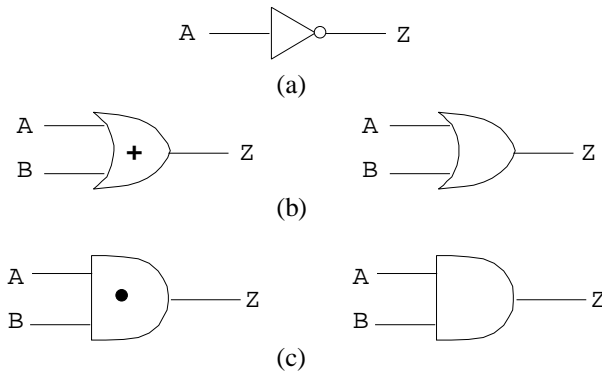
(a) Gerbang OR:  $Z = A + B$       (b) Gerbang AND:  $Z = A.B = AB$

Dalam aljabar Boole, operasi yang dilakukan oleh gerbang OR disimbolkan dengan operator "+" dan dibaca OR atau "ATAU" dan operasi AND disimbolkan dengan operator "." dan dibaca AND atau "DAN". Tanda operator "." sering dihilangkan saja dengan catatan bahwa tanpa ada operator lain diartikan sebagai operasi AND. Seperti ditunjukkan dalam Gambar 2.1, operasi OR dan AND untuk dua peubah masukan dituliskan sebagai berikut :

OR :  $Z = A + B$  (2.2)

AND :  $Z = A.B = AB$  (2.3)

Simbol yang umum dipakai dalam penyajian rangkaian logika untuk gerbang OR dan AND, juga NOT, ditunjukkan pada Gambar 2.2.



Gambar 2.2. Simbol-simbol gerbang dasar NOT, OR dan AND

(a) NOT:  $Z = \bar{A}$ , (b) OR:  $Z = A+B$ , (c) AND:  $Z = A \cdot B$

Dalam praktek, terutama dalam hubungan pernyataan fungsi Boole dan penyederhanaannya, operator OR sering dibaca "tambah" dan operator AND sering dibaca "kali". Karena kebiasaan ini, sering orang menganggap bahwa peubah logika (Boole) adalah peubah biner. Perlu ditegaskan bahwa peubah logika bukanlah peubah biner. Kalau peubah biner mempunyai harga yang padanya dapat dilakukan operasi aritmatika, maka peubah logika hanyalah simbol dan tidak mempunyai harga yang dapat ditambah-kurangkan atau dikali-bagikan. Tabel kebenaran OR pada Gambar 2.1 menunjukkan hal ini. Dalam logika,  $1 + 1 = 1$  sedangkan dalam biner,  $1 + 1 = 10$ . Selain itu, dalam logika tidak ada pengurangan dan pembagian.

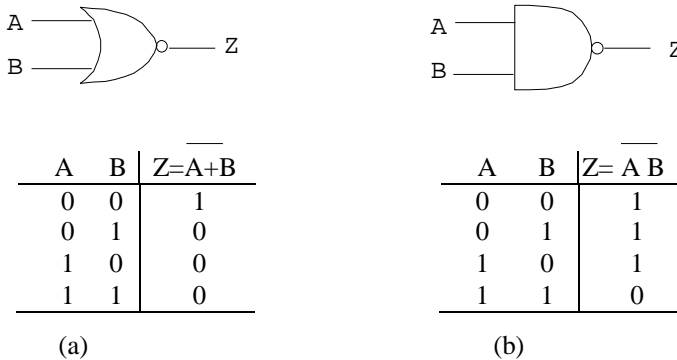
Pernyataan untuk gerbang OR dan AND dengan 2 masukan di atas dapat dikembangkan untuk semua jumlah masukan; keluaran OR adalah 1 (benar) bila salah satu masukannya 1 dan hanyalah 0 (salah) bila semua masukannya 0; keluaran AND adalah 0 (salah) bila salah satu masukannya 0 dan hanyalah benar bila semua masukannya 1. Dalam pernyataan ini tersirat suatu dualitas antara OR dan AND, yaitu pernyataan untuk OR adalah lawan/ kebalikan daripada pernyataan untuk AND. Bila pernyataan untuk OR dipakai untuk AND, artinya menggantikan AND pada tempat OR, maka keadaan 1 (benar) harus digantikan dengan 0 (salah) dan keadaan 0 (salah) digantikan dengan 1 (benar), jadi keadaannya dikomplemenkan. Keadaan serupa berlaku bila AND pada pernyataan AND digantikan dengan OR.

## 2.2 Gerbang Tambahan

Di samping gerbang-gerbang elektronik NOT, OR, dan AND, dibuat juga gerbang elektronik lain yang sangat mempermudah perencanaan beberapa bentuk rangkaian logika. Gerbang tersebut adalah gerbang-gerbang NOR, NAND, Exclusive-OR (EXOR), Exclusive-NOR (EXNOR) atau Equivalence.

Keluaran gerbang NOR adalah komplemen dari keluaran OR, dan dari kenyataan itulah disebut NOR yang merupakan singkatan dari NOT OR. Jadi, gerbang NOR merupakan gerbang OR yang di keluarannya diberi gerbang NOT pada keluarannya. NAND, yang merupakan singkatan daripada NOT AND, juga dapat dipandang sebagai gabungan antara AND dan NOT, yaitu gerbang AND dengan NOT pada keluarannya. Jadi, walaupun NOT-nya ditempelkan didepan nama gerbang-gerbang NOR dan NAND, sebenarnya NOT itu ditempelkan di bagian keluaran gerbang OR dan AND. Simbol yang dipakai untuk menyatakan

NOR adalah lambang OR yang ditambahkan lingkaran kecil pada keluarannya, dan lambang untuk NAND adalah lambang AND dengan lingkaran kecil di keluarannya. Lambang-lambang gerbang NOR dan NAND ditunjukkan pada Gambar 2.3 yang juga menunjukkan tabel kebenaran masing-masing gerbang.



Gambar 2.3. Tabel kebenaran dan simbol gerbang-gerbang NOR (a) dan NAND (b).

Untuk masukan A dan B, persamaan keluaran daripada gerbang-gerbang NOR dan NAND adalah :

$$\text{NOR} : Z = \overline{A + B} \qquad \text{NAND} : Z = \overline{A \cdot B}$$

Perhatikan bahwa keluaran NOR benar-benar merupakan komplement daripada keluaran OR dan keluaran NAND merupakan komplement daripada AND.

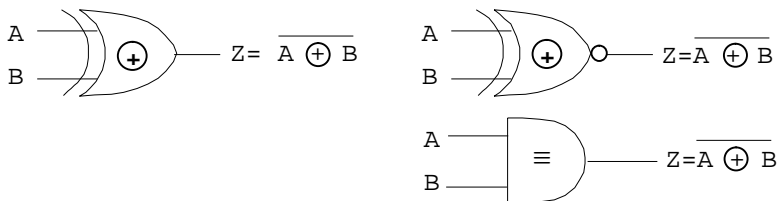
Gerbang-gerbang OR dan NOR sebenarnya adalah gerbang-gerbang inclusive-OR dan inclusive-NOR, walaupun kata inklusivenya tidak disebutkan dengan tegas. Kalau keluaran (inclusive) OR berlogika 1 asal salah satu masukannya berlogika 1, maka keluaran exclusive-OR (EXOR) hanya akan berlogika 1 bila kedua masukannya tidak sama. Keluaran exclusive-NOR (EXNOR), disebut juga Equivalence, hanya akan berlogika 1 bila kedua masukannya sama. Dalam Gambar 2.4 ditunjukkan lambang dan tabel kebenaran beserta persamaan gerbang EXOR dan EXNOR. Operasi EXOR ditunjukkan dengan + dan operasi EXNOR ditunjukkan dengan tanda "≡".

Dari tabel kebenaran dalam Gambar 2.4 dapat dilihat bahwa gerbang EXOR dan EXNOR dapat juga dinyatakan sebagai berikut:

EXOR :  $Z = A \oplus B = \bar{A} B + A \bar{B}$

EXNOR:  $Z = \overline{A \oplus B} = \bar{A} \bar{B} + A B$

Dari kesamaan ini dapat dilihat bahwa EXOR dan EXNOR dapat dibentuk dengan menggunakan AND dan OR ditambah NOT.



A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

(a)

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

(b)

Gambar 2.4. Tabel kebenaran dan Simbol gerbang-gerbang EXOR (a) dan EXNOR (b).

### 2.3 Teorema dan Hukum Dasar Aljabar Boole

Seperti telah diterangkan di bagian depan, setiap peubah Boole hanya dapat berkeadaan satu dari dua keadaan, 0 atau 1. Jadi, kalau satu peubah di-OR-kan dengan 0 maka hasilnya akan tidak berubah sedangkan bila satu peubah di-OR-kan dengan 1, maka apapun keadaan peubah itu sebelumnya akan menjadi 1. Tetapi, bila satu peubah di-AND-kan dengan 1, maka hasilnya tidak akan berubah sedangkan bila di-AND-kan dengan 0, apapun keadaan peubah itu sebelumnya akan berubah menjadi 0. Ini dapat disimpulkan dalam bentuk *teorema dasar*:

$$\begin{array}{ll}
 X + 0 = X & X \cdot 0 = 0 \\
 X + 1 = 1 & X \cdot 1 = X
 \end{array}
 \tag{2.4}$$

Kalau suatu peubah di-OR-kan dengan dirinya sendiri, maka hasilnya akan 0 bila keadaan variabel itu adalah 0 dan hasilnya akan 1 bila keadaan variabel itu adalah 1. Jadi, peng-OR-an satu variabel dengan dirinya sendiri

menghasilkan keadaan yang sama dengan keadaan variabel itu. Keadaan serupa berlaku untuk operasi AND. Ini disebut ***hukum idempoten***:

$$X + X = X \qquad X.X = X \qquad (2.5)$$

Sesuai dengan logika, maka kalau tidak benar disangkal (di-NOT-kan), hasilnya menjadi benar dan kalau tidak-salah di-NOT-kan, hasilnya menjadi salah. Dengan kata lain, penidakan/penyangkalan (komplementasi) dua kali akan menghasilkan keadaan aslinya. Ini dikenal dengan nama ***hukum involusi*** yang dituliskan sebagai:

$$\overline{\overline{X}} = X \qquad (2.6)$$

Hasil dari keadaan ***benar ATAU tidak benar*** pasti selalu ***benar*** dan keadaan ***salah ATAU tidak salah*** juga akan selalu ***benar*** (terpenuhi). Tetapi keadaan ***salah DAN tidak salah*** dan ***benar DAN tidak benar*** akan selalu ***salah***. Jadi, dalam aljabar Boole dapat dinyatakan dengan hukum komplemen sebagai berikut:

$$X + \overline{X} = 1 \quad (\text{selalu benar}) \qquad (2.7)$$

$$X . \overline{X} = 0 \quad (\text{selalu salah})$$

Untuk fungsi-fungsi Boole dengan dua peubah atau lebih, dikenal juga hukum-hukum kumulatif, assosiatif dan distributif yang berlaku dalam aljabar biasa, yaitu:

***Hukum Kumulatif*** :  $XY = YX$  (I) (2.8)

$$X + Y = Y + X \qquad (II)$$

***Hukum Assosiatif***:  $(X Y) Z = X (Y Z) = XYZ$  (I) (2.9)

$$(X+Y) + Z = X + (Y+Z) = X + Y + Z \qquad (II)$$

***Hukum Distributif***:  $X (Y + Z) = X Y + X Z$  (I) (2.10)

$$X + Y Z = (X + Y)(X + Z) \qquad (II)$$

Hukum yang terakhir ini, yang tidak ada dalam hukum distributif aljabar biasa, dapat dibuktikan sebagai berikut:

$$\begin{aligned} (X+Y)(X+Z) &= XX + XZ + YX + YZ && (\text{distributif I}) \\ &= X + XZ + XY + YZ && (\text{idempoten}) \\ &= X.1 + XZ + XY + YZ \\ &= X(1+Z+Y) + YZ && (\text{substitusi } p= Z+Y \\ &= X + YZ && \text{dan } 1 + p = 1 ) \end{aligned}$$

Di samping dengan cara seperti di atas, keadaan itu juga dapat dibuktikan dengan mudah dengan membuat tabel kebenaran. Perlu ditegaskan disini bahwa dua fungsi adalah sama bila kedua fungsi itu berlogika sama untuk semua kombinasi masukan yang mungkin. Untuk pembuktian pers. (2.10) di atas, karena ada 3 peubah, maka ada 8 ( $= 2^3$ ) kemungkinan kombinasi masukan. Harus dapat ditunjukkan bahwa untuk setiap kombinasi masukan X, Y dan Z, keadaan  $f_1 = X + YZ$  adalah sama dengan keadaan  $f_2 = (X+Y)(X+Z)$ . Ini ditunjukkan dalam Gambar 2.6. Kadang-kadang, pembuktian kesamaan dua fungsi lebih mudah dengan tabel kebenaran daripada pembuktian dengan memakai hukum-hukum dasar, tentunya terbatas pada fungsi dengan peubah yang sedikit.

X	Y	Z	YZ	$f_1$	X+Y	X+Z	$f_2$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

$f_1 = X + YZ$   
 $f_2 = (X+Y)(X+Z)$   
 $f_1 = f_2$

Gambar 2.6. Contoh pembuktian kesamaan dengan memakai tabel kebenaran.

Satu hal yang perlu diperhatikan dalam pembentukan tabel kebenaran seperti dalam Gambar 2.6 adalah penyusunan kombinasi masukan secara berurut, mulai dari setara biner terkecil sampai yang terbesar. Ini merupakan suatu cara standar penyusunan tabel kebenaran dan perlu untuk menghindari kemungkinan adanya kombinasi yang terlupakan. Perlu juga diperhatikan bahwa walaupun keadaan peubah X, Y dan Z yang 0 dan 1 bukanlah biner, kombinasinya itu dapat diartikan dalam harga biner.

Dalam Tabel 2.1 dirangkum Hukum Dasar Aljabar Boole yang dapat digunakan dalam menyederhanakan fungsi-fungsi Boole seperti yang akan dibahas dalam sub-bab berikut ini.

## 2.4 Penyederhanaan Fungsi Boole Secara Aljabar

Ditinjau dari segi rangkaian logika, semua sistem digital dapat dibedakan atas dua jenis:



- rangkaian kombinasi (combinational circuit) dan
- rangkaian berurut (sequential circuit).

Dalam rangkaian kombinasi, keluaran rangkaian pada setiap saat hanya ditentukan oleh masukannya pada saat itu. Pada rangkaian berurut, selain ditentukan oleh masukan saat itu, keluaran juga ditentukan oleh keadaan keluaran sebelumnya. Jadi, rangkaian berurut mempunyai kemampuan untuk mengingat keadaan keluarannya pada saat sebelumnya dan karena itu rangkaian berurut digunakan sebagai alat penyimpan/pengingat (storage/memory) dalam sistem digital.

**Tabel 2.1. Rumus-rumus dasar aljabar Boole.**

1. Operasi dengan 0 dan 1	$x + 0 = x$ $x \cdot 0 = 0$ $x + 1 = 1$ $x \cdot 1 = x$
2. Hukum Idempoten	$x + x = x$ $x \cdot x = x$
3. Hukum Involusi	$\overline{\overline{x}} = x$
4. Hukum Komplement	$x + \overline{x} = 1$ $\overline{\overline{x}} = x$ $x \cdot x = 0$
5. Hukum Komutatif	$x + y = y + x$ $x \cdot y = y \cdot x$
6. Hukum Asosiatif	$(x+y)+z = x+(y+z) = x+y+z$ $(xy)z = x(yz) = xyz$
7. Hukum Distributif	$x(y+z) = xy+xz$ $x+yz = (x+y)(x+z)$

Pada umumnya, sedikit-tidaknya di bagian masukan atau keluarannya, rangkaian berurut juga mempergunakan rangkaian kombinasi. Karena itu, penyederhanaan rangkaian kombinasi merupakan hal yang penting dalam setiap perencanaan sistem digital. Dengan penyederhanaan akan diperoleh rangkaian yang akan membutuhkan gerbang yang lebih sedikit dengan jumlah masukan yang lebih sedikit dibandingkan dengan merealisasikan/mengimplementasikan fungsi Boole hasil perencanaan awal.

Penyederhanaan fungsi Boole dapat dilakukan dengan beberapa cara/metoda, antara lain:

- cara aljabar,
- cara pemetaan dan
- cara tabulasi.

Dua cara terakhir akan diuraikan kemudian. Berikut ini akan diberikan beberapa contoh penyederhanaan fungsi Boole sederhana secara aljabar. Rumus-rumus penyederhanaan berikut ini dapat dipandang sebagai rumus dasar yang siap pakai. Dengan memakai hukum-hukum dan teorema dasar di depan dapat diperoleh:

$$XY + X\bar{Y} = X(Y + \bar{Y}) = X \cdot 1 = X \quad (2.11)$$

$$X + X\bar{Y} = X(1 + \bar{Y}) = X \cdot 1 = X \quad (2.12)$$

$$\begin{aligned} (X+Y)(X+\bar{Y}) &= X \cdot X + X(Y+\bar{Y}) + Y \cdot \bar{Y} \\ &= X + X \cdot 1 + 0 = X + X \\ &= X \end{aligned} \quad (2.13)$$

$$\begin{aligned} X(X+Y) &= X + XY = X \cdot 1 + XY = X(1+Y) \\ &= X \end{aligned} \quad (2.14)$$

$$\begin{aligned} (X+Y)\bar{Y} &= X\bar{Y} + Y\bar{Y} \\ &= X\bar{Y} \end{aligned} \quad (2.15)$$

$$\begin{aligned} \overline{XY} + Y &= \overline{(X+Y)(\bar{Y} + Y)} \quad (\text{hukum distributif}) \\ &= X + Y \end{aligned} \quad (2.16)$$

Satu teorema yang sangat penting dalam aljabar Boole adalah teorema *de Morgan* yang menunjukkan dualitas dalam komplementasi operasi OR dan AND. Dalil de Morgan mengubah perkalian (operasi AND) menjadi perjumlahan (operasi OR) dengan komplementasi. Hukum *de Morgan* adalah:

$$\begin{aligned} \overline{X + Y} &= \bar{X} \cdot \bar{Y} \quad (a) \\ \text{dan} \quad \overline{X \cdot Y} &= \bar{X} + \bar{Y} \quad (b) \end{aligned} \quad (2.17)$$

Hukum ini dapat dibuktikan dengan membuat tabel kebenaran untuk masing-masing operasi seperti ditunjukkan dalam Gambar 2.7.

X	Y	$\overline{X+Y}$	$\overline{X.Y}$	$\overline{X.Y}$	$\overline{X+Y}$
0	0	1	1	1	1
0	1	0	0	1	1
1	0	0	0	1	1
1	1	0	0	0	0

Gambar 2.7. Tabel kebenaran pembuktian hukum de Morgan.

Perhatikan bahwa untuk semua kombinasi masukan X dan Y keadaan di kolom 3 tepat sama dengan keadaan di kolom 4 (hukum a) dan keadaan di kolom 5 tepat sama dengan keadaan di kolom 6 (bukti hukum b).

Walaupun ditunjukkan hanya untuk 2 peubah, tetapi hukum de Morgan pers. (2.7) berlaku juga untuk sembarang cacah peubah. Ini dapat dibuktikan dengan mudah dengan metode substitusi, yaitu dengan memberikan satu nama peubah baru untuk suatu bagian pernyataan. Sebagai contoh, untuk tiga peubah dilakukan sebagai berikut :

$$\overline{X + Y + Z} = \overline{X} . \overline{Y + Z} = \overline{X} . \overline{Y} . \overline{Z}$$

$$\overline{X} . \overline{Y} . \overline{Z} = \overline{X + Y} . \overline{Z} = \overline{X + Y + Z}$$

Dengan memakai dalil de Morgan, kita dapat merealisasikan fungsi AND dengan gerbang NOR atau fungsi OR dengan gerbang NAND. Mengenai gerbang NOR dan NAND akan dijelaskan kemudian; tetapi dapat disebutkan sebelumnya bahwa pada dasarnya, semua gerbang dapat direalisasikan dengan menggunakan gerbang NAND dan NOR sehingga biasanya lebih mudah memperoleh gerbang-gerbang ini di pasaran.

Dalam menyederhanaan fungsi-fungsi Boole secara aljabar, penguasaan sekumpulan rumus dasar akan sangat membantu. Untuk memudahkan pemakaiannya dalam Tabel 2.2 dikumpulkan beberapa rumus tambahan yang melengkapi rumus dasar yang diberikan dalam Tabel 2.1. Semakin banyak kita melakukan penyederhanaan, semakin sering memakai rumus-rumus tersebut, semakin hafal pula kita akan rumus-rumus tersebut.

## 2.5 Penyajian Fungsi Boole

Seperti disebutkan di bagian depan, dalam pembicaraan aljabar Boole pengertian operasi AND sering disebut sebagai perkalian dan operasi OR disebut penjumlahan. Dengan memakai pengertian ini, maka istilah *sukumin* (singkatan dari "suku minimum" yang berasal dari istilah *minterm*, minimum term) dan *sukumax* (singkatan dari "suku maksimum" yang berasal dari istilah *maxterm*, maximum term) dapat dijelaskan lebih mudah. Sukumin dan sukumax juga dikenal dengan nama lain, yaitu "*standard product*" untuk sukumin dan "*standard sum*" untuk sukumax. Ini lebih memudahkan uraian aljabar dan penyajian fungsi-fungsi logika (fungsi Boole).

*Sukumin* adalah perkalian (operasi AND) dari sejumlah literal. Literal disini dimaksudkan sebagai peubah, baik dalam bentuk sebenarnya maupun komplemennya. Dalam satu suku, setiap literal muncul paling banyak satu kali. Ini berarti bahwa bila satu suku mengandung literal A, misalnya, suku tersebut tidak boleh

**Tabel 2.2. Rumus-rumus Tambahan Boole.**

1. Teorema penyederhanaan:	$x y + x \bar{y} = x$ $x + x y = x$ $(x + \bar{y}) y = x y$ $(x+y)(x+\bar{y}) = x$ $x(x+y) = x$ $xy + y = x + y$
2. Hukum de Morgan:	$\overline{x + y + z + \dots} = \bar{x} \bar{y} \bar{z} \dots$ $\overline{x \cdot y \cdot z \cdot \dots} = \bar{x} + \bar{y} + \bar{z} + \dots$
3. Teorema Konsensus:	$xy + yz + xz = xy + xz$ $(x+y)(y+z)(x+z) = (x+y)(x+z)$ $(x+y)(x+z) = xz + xy$
4. Dualitas	$(x + y + z + \dots)^D = x y z$ $(x y z \dots)^D = x + y + z + \dots$ $[f(x_1, x_2, x_3, \dots, x_n, 0, 1, +, \cdot)]^D = f(x_1, x_2, x_3, \dots, x_n, 0, 1, +, \cdot)$

mengandung literal A. Karena untuk n peubah dapat dibentuk 2n macam kombi-

- - - - -  
- - - - -

nasi, maka untuk  $n$  peubah dapat dibentuk sejumlah  $2^n$  sukumin. Setiap sukumin berharga 1 hanya untuk satu kombinasi. Sebagai contoh, untuk dua peubah  $A$  dan  $B$ , sukumin yang dapat dibentuk adalah  $AB$ ,  $\bar{A}B$ ,  $A\bar{B}$  dan  $\bar{A}\bar{B}$ . Sukumin  $AB$  akan berharga 1 hanya untuk  $A = B = 0$  atau  $A = 1$  dan  $B = 1$ ; sukumin  $\bar{A}B = 1$  hanya bila  $A = 0$  dan  $B = 1$ , dan seterusnya. Untuk penyingkatan penulisan, sukumin sering ditulis secara singkat dengan  $m_i$ , dengan  $i$  menunjukkan harga desimal daripada sukumin tersebut. Sebagai contoh, sukumin  $AB$  akan berharga 1 hanya untuk  $AB = 01$ , artinya  $A = 0$  dan  $B = 1$ , dan karena harga desimal daripada biner  $01$  adalah 1 maka sukumin  $AB$  disebut sukumin 1 atau  $m_1$ , sukumin  $\bar{A}B$  disebut  $m_3$ , dan sebagainya.

**Sukumax** adalah penjumlahan (operasi OR) daripada sejumlah literal dengan setiap literal muncul hanya 1 kali, dan setiap sukumax mempunyai harga 0 hanya untuk satu macam kombinasi daripada literal pembentuknya.  $A + B + C$  adalah sukumax yang dapat dibentuk dari 3 peubah  $A$ ,  $B$  dan  $C$  dan berharga 0 hanya bila  $A = 1$ ,  $B = 0$ , dan  $C = 0$ . Untuk penulisan secara singkat, sukumax ditulis dengan  $M_i$ , dengan  $i$  sebagai harga desimal daripada biner yang dibentuk oleh kombinasi AND peubahnya. Sukumax  $(A+B+C)$ , yang akan berharga 0 hanya bila  $A = 0$ ,  $B = 0$  dan  $C = 0$ , yaitu bila  $ABC = 000 = 0$  desimal, dituliskan dengan  $M_0$ .

Perhatikan dalam penentuan sukumin dan sukumax di atas, bahwa untuk sukumin setiap literal yang dalam bentuk komplemen diartikan 0 sedangkan dalam penentuan sukumax setiap literal dalam bentuk komplemen diartikan 1. Ini adalah karena dalam sukumin kita membentuk suku yang berharga 1 sedangkan dalam sukumax kita membentuk suku yang berharga 0.

Dengan dalil de Morgan dapat dilihat dengan mudah bahwa:

$$m_i = \bar{M}_i$$

dan

$$M_i = \bar{m}_i$$

Untuk 3 peubah, misalnya  $a$ ,  $b$ , dan  $c$ , sukumin-5 dan sukumax-5 ( $i = 5$ ) dapat ditulis:

$$m_5 = \bar{a} \bar{b} c$$

$$\bar{m}_5 = \bar{a} + b + \bar{c} = M_5$$

Bila suatu fungsi Boole ditulis sebagai perjumlahan daripada sukumin, maka fungsi itu disebut sebagai ekspansi sukumin atau jumlah-perkalian standar (minterm expansion, standard sum-of-products) dan bila ditulis sebagai perkalian daripada sukumax, maka fungsi itu disebut dalam bentuk ekspansi sukumax atau perkalian-jumlah standar (maxterm expansion, standard product-of-sum). Bentuk jumlah perkalian sering ditulis dengan notasi sigma ( $S$ ) dan bentuk perkalian

jumlah ditulis dalam bentuk pi (p) yang sedikit diubah, yaitu:

$$m_0 + m_1 + m_2 + \dots + m_{n-1} = \sum_{i=0}^{n-1} m_i = \sum m(0,1,2,\dots,n-1)$$

dan

$$M_0 M_1 M_2 \dots M_{n-1} = \prod_{i=0}^{n-1} M_i = \prod M(0,1,2,\dots,n-1)$$

Bilangan yang dicantumkan dalam tanda kurung ruas paling kanan rumus di atas hanyalah sukumin/sukumax penyusun fungsi.

**Contoh:**

Perhatikan tabel kebenaran fungsi seperti yang ditunjukkan dalam Gambar 2.8. Dari tabel kebenaran ini diperoleh pernyataan fungsi dalam bentuk jumlah-perkalian (ekspansi sukumin) dan dalam bentuk perkalian-jumlah (ekspansi sukumax) sebagai berikut:

A	B	C	f	
0	0	0	0	
0	0	1	1	$f = \sum m(1,3,4,6)$
0	1	0	0	
0	1	1	1	$= \prod M(0,2,5,7)$
1	0	0	1	
1	0	1	0	
1	1	0	1	
1	1	1	0	

Gambar 2.8. Tabel kebenaran untuk memperoleh fungsi dalam bentuk ekspansi sukumin dan sukumax.

**Ekspansi sukumin :**

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

yang diperoleh dari penjumlahan (peng-OR-an) suku-suku 001 (=1), 011 (=3), 100 (=4), dan 110 (=6) yang membuat  $f = 1$ . Fungsi ini dapat dinyatakan sebagai:

$$\begin{aligned} f &= m_1 + m_3 + m_4 + m_6 \\ &= \sum m(1,3,4,6) \end{aligned}$$



0	0	0	1
0	0	1	x
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	x
1	1	1	1

Gambar 2.9. Tabel kebenaran fungsi tak lengkap.

$$y = \sum m(0,3,7) + \sum d(1,6)$$

Kita perhatikan kemungkinan harga  $x$  yang dapat kita pilih untuk suku abaikan sebagai berikut ini:

### 1. Semua $x$ kita anggap 0.

Untuk pilihan ini, fungsi tersebut akan berbentuk:

$$\begin{aligned} y &= m_0 + m_3 + m_7 \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C = \bar{A}\bar{B}\bar{C} + B\bar{C} \end{aligned}$$

### 2. Semua $x$ kita anggap 1.

$$\begin{aligned} y &= m_0 + m_1 + m_3 + m_6 + m_7 \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + \bar{A}BC + A\bar{B}C \\ &= \bar{A}\bar{B}(\bar{C}+C) + (\bar{A}+A)B\bar{C} + A\bar{C}(\bar{B}+B) = \bar{A}\bar{B} + B\bar{C} + A\bar{C} \end{aligned}$$

### 3. Untuk $m_1$ kita pilih $x=1$ dan untuk $m_6$ kita pilih $x=0$ .

$$\begin{aligned} y &= m_0 + m_1 + m_3 + m_7 \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C = \bar{A}\bar{B} + B\bar{C} \end{aligned}$$

### 4. Untuk $m_1$ kita pilih $x=0$ dan untuk $m_6$ kita pilih $x=1$ .

$$\begin{aligned} y &= m_0 + m_3 + m_6 + m_7 \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C \\ &\quad \bar{\quad} \quad \bar{\quad} \quad \bar{\quad} \quad \bar{\quad} \\ &\quad \bar{\quad} \quad \bar{\quad} \quad \bar{\quad} \quad \bar{\quad} \end{aligned}$$



$$\begin{aligned}
 &= A B C + A B C + A B C + A B C + A B C \\
 &= A B C + B C + A B
 \end{aligned}$$

Dari hasil-hasil di atas dapat dilihat bahwa penyelesaian paling sederhana adalah dengan pilihan ke 3, yaitu dengan  $m_1 = 1$  dan  $m_6 = 0$ . Kesederhanaan suatu rangkaian logika pada umumnya diukur dari jumlah gerbang yang dibutuhkan dan jumlah terminal masukan paling sedikit. Yang lebih gampang adalah hanya menghitung jumlah masukan yang dibutuhkan tanpa memperdulikan gerbang apa yang dipakai dan berapa jumlah masukan setiap gerbang. Sebagai contoh, kalau dihitung, jumlah masukan untuk pilihan 1 di atas adalah  $3 + 2 + 2 = 7$  (1 AND dua masukan, 1 AND tiga masukan dan 1 OR dua masukan). Untuk pilihan 2 dibutuhkan  $2 + 2 + 2 + 2 + 2 = 10$  masukan (3 AND dua masukan dan 2 OR dua masukan), untuk pilihan 3 dibutuhkan  $2 + 2 + 2 = 6$  masukan (2 AND dua masukan dan 1 OR dua masukan), dan untuk pilihan 4 dibutuhkan  $2 + 2 + 3 + 3 = 10$  masukan bila memakai 2 OR dua masukan, 1 AND 3 masukan dan 1 OR tiga masukan atau  $3 + 2 + 2 + 2 + 2 = 11$  masukan bila memakai 2 AND dengan 2 masukan, 2 OR dengan 2 masukan dan 1 AND dengan 3 masukan. (Periksa dengan menggambar rangkaianannya !).

Sekarang perhatikan kembali pada Gambar 2.9, dan andaikan  $x$  dipilih seperti pada pilihan ke 2 di atas. Bila fungsi tersebut diekspansikan ke sukumax, maka akan diperoleh :

$$\begin{aligned}
 y &= M_2 M_4 M_5 \\
 &= (A + \bar{B} + C) (\bar{A} + B + C) (\bar{A} + B + \bar{C}) \\
 &= (A B + A C + \bar{A} \bar{B} + \bar{B} C + \bar{A} C + B C + C) (\bar{A} + B + \bar{C}) \\
 &= \{(A B + \bar{A} \bar{B} + (A + \bar{B} + \bar{A} + B + 1) C\} (\bar{A} + B + \bar{C}) \\
 &= (A B + \bar{A} \bar{B} + C) (\bar{A} + B + \bar{C}) \\
 &= A B + A B \bar{C} + \bar{A} \bar{B} + \bar{A} \bar{B} \bar{C} + \bar{A} C + B C \\
 &= A B (1 + \bar{C}) + \bar{A} \bar{B} (1 + \bar{C}) + \bar{A} C + B C \\
 &= A B + \bar{A} \bar{B} + \bar{A} C + B C \\
 &= A B + (\bar{B} C + \bar{A} C + \bar{A} \bar{B}) \\
 &= A B + B C + \bar{A} \bar{B} \quad (\text{lihat rumus teorema konsensus pertama di depan}).
 \end{aligned}$$

Dapat dilihat disini bahwa fungsi yang diperoleh dengan ekspansi ke sukumax tepat sama dengan yang diperoleh dengan ekspansi ke sukumin sebelumnya. Sebenarnya, hasil terakhir ini dapat diperoleh dalam bentuk lain yang sedikit berbeda dengan harga yang sama (sama jumlah masukannya). Ini akan lebih jelas kalau kita menyederhanakannya dengan cara peta yang akan diuraikan dalam bab berikutnya.

## 2.7 Soal Latihan

1. Buktikanlah Rumus-rumus penyederhanaan dan teorema konsensus Tabel 2.2.
2. Suatu sistem dengan 3 peubah masukan membutuhkan hubungan logika seperti yang ditunjukkan pada tabel kebenaran Tabel S2.1.
  - a. Tentukanlah pernyataan logika fungsi keluaran  $f$  dalam bentuk sukumin dan dalam bentuk sukumax
  - b. Tentukanlah realisasi fungsi  $f$  yang paling murah
  - c. Gambarkanlah rangkaian logikanya dalam bentuk OR-AND (OR diikuti AND) dan AND-OR (AND diikuti OR).

Tabel S2.1.

p	q	r	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

3. Sederhanakanlah pernyataan Boole berikut:
  - a.  $ABC(ABC + ABC + ABC)$
  - b.  $AB + AB + AC + BC$

$$c. A(A+B+C)(\bar{A}+\bar{B}+\bar{C})(A+B+C)$$

$$d. (\bar{A}+\bar{B}+\bar{C})(A+\bar{B}+C)(A+B+\bar{C})(A+B+C)$$

4. Sederhanakanlah ke dalam bentuk ekspansi sukumin dan ekspansi sukumax dan gambarkan rangkaian untuk fungsi berikut:

$$a. x = (\bar{A}\bar{B} + \bar{C} + DE)(\bar{A}\bar{B} + \bar{C})$$

$$b. y = (CD + \bar{A} + \bar{B})(CD + \bar{A} + B)$$

5. Gambarkan rangkaian untuk soal No. 3 diatas dengan hanya menggunakan gerbang-gerbang:

- NAND sembarang cacah masukan
- NOR sembarang cacah masukan
- NAND 2 masukan
- NOR 2 masukan

6. Sederhanakanlah secara aljabar Boole fungsi

$$f(a,b,c) = \sum m(1,3,4,5) + \sum d(6,7)$$

$$f(a,b,c) = \prod M(0,2) + \prod d(6,7)$$

dengan  $d_6$  dan  $d_7$  adalah suku abaikan (don't care)