**MODUL PERKULIAHAN**

**EDISI 1**

**MATEMATIKA DISKRIT**



Penulis :

Nelly Indriani Widiastuti S.Si., M.T.

JURUSAN TEKNIK INFORMATIKA

UNIVERSITAS KOMPUTER INDONESIA

BANDUNG

2011

|  |
| --- |
| ALJABAR BOOLEAN **9** |
| JUMLAH PERTEMUAN : 1 PERTEMUAN  TUJUAN INSTRUKSIONAL KHUSUS : |

**Materi :**

## **Pendahuluan**

Adalah aljabar logika. Sifat biner proposisi / dalil logis (TRUE or FALSE) menunjukkan mempunyai aplikasi dalam komputasi.

Pelopornya George Boole

Misalkan terdapat

* Dua operator biner: + dan ⋅
* Sebuah operator uner: ’.
* *B* : himpunan yang didefinisikan pada operator +, ⋅, dan ’
* 0 dan 1 adalah dua elemen yang berbeda dari *B*.

## Tupel

(*B*, +, ⋅, ’)

disebut **aljabar Boolean** jika untuk setiap *a*, *b*, *c* ∈ *B* berlaku aksioma-aksioma atau **postulat Huntington** berikut:

1. *Closure*: (i) *a* + *b* ∈ *B*

(ii) *a* ⋅ *b* ∈ *B*

2. Identitas: (i) *a* + 0 = *a*

(ii) *a* ⋅ 1 = *a*

3. Komutatif: (i) *a* + *b* = *b* + *a*

(ii) *a* ⋅ *b* = *b* . *a*

4. Distributif: (i) *a* ⋅ (*b* + *c*) = (*a* ⋅ *b*) + (*a* ⋅ *c*)

(ii) *a* + (*b* ⋅ *c*) = (*a* + *b*) ⋅ (*a* + *c*)

5. Komplemen[[1]](#footnote-2): (i) *a* + *a*’ = 1

(ii) *a* ⋅ *a*’ = 0

Untuk mempunyai sebuah aljabar Boolean, harus diperlihatkan:

1. Elemen-elemen himpunan *B*,
2. Kaidah operasi untuk operator biner dan operator uner,

Memenuhi postulat Huntington.

## **ALJABAR BOOLEAN DUA-NILAI**

Aljabar Boolean dua-nilai:

* *B* = {0, 1}
* operator biner, + dan ⋅
* operator uner, ’
* Kaidah untuk operator biner dan operator uner:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *a* | *b* | *a* ⋅ *b* |  | *a* | *b* | *a* + *b* |  | *a* | *a*’ |
| 0 | 0 | 0 |  | 0 | 0 | 0 |  | 0 | 1 |
| 0 | 1 | 0 |  | 0 | 1 | 1 |  | 1 | 0 |
| 1 | 0 | 0 |  | 1 | 0 | 1 |  |  |  |
| 1 | 1 | 1 |  | 1 | 1 | 1 |  |  |  |

Cek apakah memenuhi postulat Huntington:

1. *Closure* : jelas berlaku
2. Identitas: jelas berlaku karena dari tabel dapat kita lihat bahwa:

(i) 0 + 1 = 1 + 0 = 1

(ii) 1 ⋅ 0 = 0 ⋅ 1 = 0

1. Komutatif: jelas berlaku dengan melihat simetri tabel operator biner.
2. Distributif: (i) *a* ⋅ (*b* + *c*) = (*a* ⋅ *b*) + (*a* ⋅ *c*) dapat ditunjukkan benar dari tabel operator biner di atas dengan membentuk tabel kebenaran:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *a* | *b* | *c* | *b* + *c* | *a* ⋅ (*b* + *c*) | *a* ⋅ *b* | *a* ⋅ *c* | (*a* ⋅ *b*) + (*a* ⋅ *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 |

(ii) Hukum distributif *a* + (*b* ⋅ *c*) = (*a* + *b*) ⋅ (*a* + *c*) dapat ditunjukkan benar dengan membuat tabel kebenaran dengan cara yang sama seperti (i).

1. Komplemen: jelas berlaku karena Tabel 7.3 memperlihatkan bahwa:

(i) *a* + *a*‘ = 1, karena 0 + 0’= 0 + 1 = 1 dan 1 + 1’= 1 + 0 = 1

(ii) *a* ⋅ *a* = 0, karena 0 ⋅ 0’= 0 ⋅ 1 = 0 dan 1 ⋅ 1’ = 1 ⋅ 0 = 0

Karena kelima postulat Huntington dipenuhi, maka terbukti bahwa *B* = {0, 1} bersama-sama dengan operator biner + dan ⋅ operator komplemen ‘ merupakan aljabar Boolean.

## **EKSPRESI BOOLEAN**

Misalkan (*B*, +, ⋅, ’) adalah sebuah aljabar Boolean. Suatu ekspresi Boolean dalam (*B*, +, ⋅, ’) adalah:

(i) setiap elemen di dalam *B*,

(ii) setiap peubah,

(iii) jika *e*1 dan *e*2 adalah ekspresi Boolean, maka *e*1 + *e*2, *e*1 ⋅ *e*2, *e*1’ adalah ekspresi Boolean

Contoh:

0

1

*a*

*b*

*c*

*a* + *b*

*a* ⋅ *b*

*a*’⋅ (*b* + *c*)

*a* ⋅ *b*’ + *a* ⋅ *b* ⋅ *c*’ + *b*’, dan sebagainya

**Mengevaluasi Ekspresi Boolean**

* Contoh: *a*’⋅ (*b* + *c*)

jika *a* = 0, *b* = 1, dan *c* = 0, maka hasil evaluasi ekspresi:

0’⋅ (1 + 0) = 1 ⋅ 1 = 1

* Dua ekspresi Boolean dikatakan **ekivalen** (dilambangkan dengan ‘=’) jika keduanya mempunyai nilai yang sama untuk setiap pemberian nilai-nilai kepada *n* peubah.

Contoh:

*a* ⋅ (*b* + *c*) = (*a* . *b*) + (*a* ⋅ *c*)

**Contoh.** Perlihatkan bahwa *a* + *a*’*b* = *a* + *b* .

Penyelesaian:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *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 |

* Perjanjian: tanda titik (⋅) dapat dihilangkan dari penulisan ekspresi Boolean, kecuali jika ada penekanan:

(i)  *a*(*b* + *c*) = *ab* + *ac*

1. *a* + *bc* = (*a* + *b*) (*a* + *c*)
2. *a* ⋅ 0 , bukan a0

## **DUALITAS**

* Misalkan *S* adalah kesamaan (*identity*) di dalam aljabar Boolean yang melibatkan operator +, ⋅, dan komplemen, maka jika pernyataan *S*\* diperoleh dengan cara mengganti

⋅ dengan +

+ dengan ⋅

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) (*a* ⋅ 1)(0 + *a*’) = 0 dualnya (*a* + 0) + (1 ⋅ *a*’) = 1

(ii) *a*(*a*‘ + *b*) = *ab* dualnya *a* + *a*‘*b* = *a* + *b*

**Hukum-hukum Aljabar Boolean**

|  |  |
| --- | --- |
| 1. Hukum identitas:  (i) *a* + 0 = *a*  (ii) *a* ⋅ 1 = *a* | 2. Hukum idempoten:  (i) *a* + *a* = *a*  (ii) *a* ⋅ *a* = *a* |
| 3. Hukum komplemen:  (i) *a* + *a*’ = 1  (ii) *aa*’ = 0 | 4. Hukum dominansi:  (i) *a* ⋅ 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*) = *a* *b* + *a* *c* | 10. Hukum De Morgan:  (i) (*a* + *b*)’ = *a*’*b*’  (ii) (*ab*)’ = *a*’ + *b*’ |
| 1. Hukum 0/1   (i) 0’ = 1  (ii) 1’ = 0 |  |

**Contoh 7.3.** Buktikan (i) *a* + *a*’*b* = *a* + *b* dan (ii) *a*(*a*’ + *b*) = *ab*

Penyelesaian:

(i) *a* + *a*’*b* = (*a* + *ab*) + *a*’*b* (Penyerapan)

= *a* + (*ab* + *a*’*b*) (Asosiatif)

= *a* + (*a* + *a*’)*b* (Distributif)

= *a* + 1 • *b* (Komplemen)

= *a* + *b* (Identitas)

(ii) adalah dual dari (i)

## **FUNGSI BOOLEAN**

* **Fungsi Boolean** (disebut juga fungsi biner) adalah pemetaan dari *Bn* ke *B* melalui ekspresi Boolean, kita menuliskannya sebagai

*f* : *Bn*→ *B*

yang dalam hal ini *Bn* adalah himpunan yang beranggotakan pasangan terurut ganda-*n* (*ordered n-tuple*) di dalam daerah asal *B*.

* Setiap ekspresi Boolean tidak lain merupakan fungsi Boolean.
* Misalkan sebuah fungsi Boolean adalah

*f*(*x*, *y*, *z*) = *xyz* + *x*’*y* + *y*’*z*

Fungsi *f* memetakan nilai-nilai pasangan terurut ganda-3

(*x*, *y*, *z*) ke himpunan {0, 1}.

Contohnya, (1, 0, 1) yang berarti *x* = 1, *y* = 0, dan *z* = 1

sehingga f(1, 0, 1) = 1 ⋅ 0 ⋅ 1 + 1’ ⋅ 0 + 0’⋅ 1 = 0 + 0 + 1 = 1 .

**Contoh.** Contoh-contoh fungsi Boolean yang lain:

1. *f*(*x*) = *x*
2. *f*(*x*, *y*) = *x*’*y* + *xy*’+ *y*’
3. *f*(*x*, *y*) = *x*’ *y*’
4. *f*(*x*, *y*) = (*x* + *y*)’
5. *f*(*x*, *y*, *z*) = *xyz*’

* Setiap peubah di dalam fungsi Boolean, termasuk dalam bentuk komplemennya, disebut **literal**.

Contoh: Fungsi *h*(*x*, *y*, *z*) = *xyz*’ pada contoh di atas terdiri dari 3 buah literal, yaitu *x*, y, dan *z*’.

**Contoh.** Diketahui fungsi Booelan *f*(*x*, *y*, *z*) = *xy z*’, nyatakan *h* dalam tabel kebenaran.

Penyelesaian:

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | *y* | *z* | *f*(*x*, *y*, *z*) = *xy z*’ |
| 0  0  0  0  1  1  1  1 | 0  0  1  1  0  0  1  1 | 0  1  0  1  0  1  0  1 | 0  0  0  0  0  0  1  0 |

## **KOMPLEMEN FUNGSI**

Cara pertama: menggunakan hukum De Morgan

Hukum De Morgan untuk dua buah peubah, *x*1 dan *x*2, adalah

**Contoh.** Misalkan *f*(*x*, *y*, *z*) = *x*(*y*’*z*’ + *yz*), maka

*f* ’(*x*, *y*, *z*) = (*x*(*y*’*z*’ + *yz*))’

= *x*’ + (*y*’*z*’ + *yz*)’

= *x*’ + (*y*’*z*’)’ (*yz*)’

= *x*’ + (*y* + *z*) (*y*’ + *z*’)

Cara kedua: menggunakan prinsip dualitas.

Tentukan dual dari ekspresi Boolean yang merepresentasikan *f*, lalu komplemenkan setiap literal di dalam dual tersebut.

**Contoh.** Misalkan *f*(*x*, *y*, *z*) = *x*(*y*’*z*’ + *yz*), maka

dual dari *f*: *x* + (*y*’ + *z*’) (*y* + *z*)

komplemenkan tiap literalnya: *x*’ + (*y* + *z*) (*y*’ + *z*’) = *f* ’

Jadi, *f* ‘(*x*, *y*, *z*) = *x*’ + (*y* + *z*)(*y*’ + *z*’)

## **BENTUK KANONIK**

* Jadi, ada dua macam bentuk kanonik:

1. Penjumlahan dari hasil kali (*sum-of-product* atau SOP)
2. Perkalian dari hasil jumlah (*product-of-sum* atau POS)

Contoh: 1. *f*(*x*, *y*, *z*) = *x*’*y*’*z* + *xy*’*z*’ + *xyz* 🡪 SOP

Setiap suku (*term*) disebut *minterm*

2. *g*(*x*, *y*, *z*) = (*x* + *y* + *z*)(*x* + *y*’ + *z*)(*x* + *y*’ + *z*’)

(*x*’ + *y* + *z*’)(*x*’ + *y*’ + *z*) 🡪 POS

Setiap suku (*term*) disebut *maxterm*

* Setiap *minterm*/*maxterm* mengandung literal lengkap

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | *Minterm* | | *Maxterm* | |
| *x* | *y* | | Suku | Lambang | Suku | Lambang |
| 0  0  1  1 | | 0  1  0  1 | *x*’*y*’  *x*’*y*  *xy*’  *x y* | *m*0  *m*1  *m*2  *m*3 | *x* + *y*  *x* + *y*’  *x*’ + *y*  *x*’ + *y*’ | *M*0  *M*1  *M*2  *M*3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *Minterm* | | *Maxterm* | |
| *x* | *y* | *z* | Suku | Lambang | Suku | Lambang |
| 0  0  0  0  1  1  1  1 | 0  0  1  1  0  0  1  1 | 0  1  0  1  0  1  0  1 | *x*’*y*’*z*’  *x*’*y*’*z*  *x*‘*y* *z*’  *x*’*y* *z*  *x* *y*’*z*’  *x y*’*z*  *x* *y* *z*’  *x y z* | *m*0  *m*1  *m*2  *m*3  *m*4  *m*5  *m*6  *m*7 | *x* + *y* + *z*  *x* + *y* + *z*’  *x* + *y*’+*z*  *x* + *y*’+*z*’  *x*’+ *y* + *z*  *x*’+ *y* + *z*’  *x*’+ *y*’+ *z*  *x*’+ *y*’+ *z*’ | *M*0  *M*1  *M*2  *M*3  *M*4  *M*5  *M*6  *M*7 |

**Contoh 7.10.** Nyatakan tabel kebenaran di bawah ini dalam bentuk kanonik SOP dan POS.

**Tabel 7.10**

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

Penyelesaian:

1. 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 = ∑ (1, 4, 7)

(b) POS

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 = ∏(0, 2, 3, 5, 6)

**Contoh 7.11.** Nyatakan fungsi Boolean *f*(*x*, *y*, *z*) = *x* + *y*’*z* dalam bentuk kanonik SOP dan POS.

Penyelesaian:

(a) SOP

*x* = *x*(*y* + *y*’)

= *xy* + *xy*’

= *xy* (*z* + *z*’) + *xy*’(*z* + *z*’)

= *xyz* + *xyz*’ + *xy*’*z* + *xy*’*z*’

*y*’*z* = *y*’*z* (*x* + *x*’)

= xy’z + x’y’z

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*

atau *f*(*x*, *y*, *z*) = *m*1 + *m*4 + *m*5 + *m*6 + *m*7 = Σ (1,4,5,6,7)

(b) POS

*f*(*x*, *y*, *z*) = *x* + *y*’*z*

= (*x* + *y*’)(*x* + *z*)

*x* + *y*’ = *x* + *y*’ + *zz*’

= (*x* + *y*’ + *z*)(*x* + *y*’ + *z*’)

*x* + *z* = *x* + *z* + *yy*’

= (*x* + *y* + *z*)(*x* + *y*’ + *z*)

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*’)

atau *f*(*x*, *y*, *z*) = *M*0*M*2*M*3 = ∏(0, 2, 3)

**Konversi Antar Bentuk Kanonik**

Misalkan

*f*(*x*, *y*, z) = Σ (1, 4, 5, 6, 7)

dan *f* ’adalah fungsi komplemen dari *f*,

*f* ’(*x*, *y*, *z*) = Σ (0, 2, 3) = *m*0+ *m*2 + *m*3

Dengan menggunakan hukum De Morgan, kita dapat memperoleh fungsi *f* dalam bentuk POS:

*f* ’(*x*, *y*, *z*) = (*f* ’(*x*, *y*, *z*))’ = (*m*0 + *m*2 + *m*3)’

= *m*0’ . *m*2’ . *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

= ∏ (0,2,3)

Jadi, *f*(*x*, *y*, z) = Σ (1, 4, 5, 6, 7) = ∏ (0,2,3).

Kesimpulan: *mj*’ = *Mj*

**Contoh.**  Nyatakan

*f*(*x*, *y*, *z*)= ∏ (0, 2, 4, 5) dan

*g*(*w*, *x*, y, z) = Σ(1, 2, 5, 6, 10, 15)

dalam bentuk SOP.

Penyelesaian:

*f*(*x*, *y*, z) = Σ (1, 3, 6, 7)

*g*(*w*, *x*, *y*, *z*)= ∏ (0, 3, 4, 7, 8, 9, 11, 12, 13, 14)

**Contoh.** Carilah bentuk kanonik SOP dan POS dari *f*(*x*, *y*, *z*) = *y*’ + *xy* + *x*’yz’

Penyelesaian:

(a) SOP

*f*(*x*, *y*, z) = *y*’ + *xy* + *x*’*yz*’

= *y*’ (*x* + *x*’) (*z* + *z*’) + *xy* (*z* + *z*’) + *x*’*yz*’

= (*xy*’ + *x*’*y*’) (*z* + *z*’) + *xyz* + *xyz*’ + *x*’*yz*’

= *xy*’*z* + *xy*’*z*’ + *x*’*y*’*z* + *x*’*y*’*z*’ + *xyz* + *xyz*’ + *x*’*yz*’

atau *f*(*x*, *y*, z) = *m*0+ *m*1 + *m*2+ *m*4+ *m*5+ *m*6+ *m*7

(b) POS

*f*(*x*, *y*, z) = *M*3 = *x* + *y*’ + *z*’

## **Bentuk Baku**

Contohnya,

*f*(*x*, *y*, *z*) = *y*’ + *xy* + *x*’*yz* (bentuk baku SOP

*f*(*x*, *y*, *z*) = *x*(*y*’ + *z*)(*x*’ + *y* + *z*’) (bentuk baku POS)

1. [↑](#footnote-ref-2)