# TUGAS INDIVIDU

# C:\Users\JHOY SANTOS0917\Desktop\0981e16ae953ac6df50158cb84eed80e.jpeg

# NAMA : ANJO DOS SANTOS

# NIM : 2244190025

# SEMESTER : I

# JURUSAN : INFORMATIKA

# MATA KUALIAH : MATEMATIKA DISKRIT

**FAKULTAS TEKNIK**

**UNIVERSITAS PERSADA INDONESIA Y.A.I**

**(UPI Y.A.I)**

**2022**

# Aplikasi Aljabar Boolean

* 1. **Jaringan Pensaklaran (*Switching Network*)**

Saklar: objek yang mempunyai dua buah keadaan: buka dan tutup. Tiga bentuk gerbang paling sederhana:

1. *a x b*

*Output b* hanya ada jika dan hanya jika *x* dibuka  *x*

1. *a x y b*

*Output b* hanya ada jika dan hanya jika *x* dan *y* dibuka  *xy*

1. *a x*

*c*

*b y*

*Output c* hanya ada jika dan hanya jika *x* atau *y* dibuka  *x* + *y* 33

Contoh rangkaian pensaklaran pada rangkaian listrik:

1. Saklar dalam hubungan SERI: logika AND

Lampu

*A B*



Sumber tegangan

1. Saklar dalam hubungan PARALEL: logika OR

*A*

Lampu

*B*



Sumber Tegangan

* 1. **Rangkaian Logika**

*x x*

*x*

*xy x+ y x'*

*y y*

Gerbang AND Gerbang OR Gerbang NOT (*inverter*)

**Contoh.** Nyatakan fungsi *f*(*x*, *y*, *z*) = *xy* + *x*’*y* ke dalam rangkaian logika.

Jawab: (a) Cara pertama

*x y*



*xy*

*x'*

*x'y*

*xy+x'y*

*x y*

1. Cara kedua

*x y*

*xy*

*x*'*y*

*xy+x*'*y*

*x*'

1. Cara ketiga

*x y*

*xy+x'y*



*xy*

*x'*

*x'y*

Gerbang turunan

*x x*



+

(*xy*)' *x y*

*y y*

Gerbang NAND Gerbang XOR

*x x*



(*x+y*)'

*y y*

(*x y*)'

Gerbang NOR Gerbang XNOR



+

*x* (*x* + *y*)' ekivalen dengan *x*



*x* + *y*

*y y*

(*x* + *y*)'

*x*' *x*'*y*' ekivalen dengan *x*



*y*' *y*

(*x+y*)'

*x*



*x*' *x*' + *y*' ekivalen dengan

*y*' *y*

(*xy*)'

### Penyederhanaan Fungsi Boolean

**Contoh.** *f*(*x*, *y*) = *x*’*y* + *xy*’ + *y*’

disederhanakan menjadi

*f*(*x*, *y*) = *x*’ + *y*’

Penyederhanaan fungsi Boolean dapat dilakukan dengan 3 cara:

1. Secara aljabar
2. Menggunakan Peta Karnaugh
3. Menggunakan metode Quine Mc Cluskey (metode Tabulasi)

### Penyederhanaan Secara Aljabar

**Contoh**:

* 1. *f*(*x*, *y*) = *x* + *x*’*y*

= (*x* + *x*’)(*x* + *y*)

= 1  (*x* + *y* )

= *x* + *y*

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

= *x*’*z*(*y*’ + *y*) + *xy*’

= *x*’*z* + *xz*’

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

= *xy* + *x*’*z* + *xyz* + *x*’*yz*

= *xy*(1 + *z*) + *x*’*z*(1 + *y*) = *xy* + *x*’*z* 41

### Peta Karnaugh

1. *Peta Karnaugh dengan dua peubah*

*y*

0 1

*x* 0

|  |  |
| --- | --- |
| *m*0 | *m*1 |
| *m*2 | *m*3 |

|  |  |
| --- | --- |
| *x*’*y*’ | *x*’*y* |
| *xy*’ | *xy* |

1

1. *Peta 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*’*y*’*z*’ | *x*’*y*’*z* | *x*’*yz* | *x*’*yz*’ |
| *xy*’*z*’ | *xy*’*z* | *xyz* | *xyz*’ |

*x* 0

1

**Contoh.** Diberikan tabel kebenaran, gambarkan Peta Karnaugh.

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

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |

*x* 0

1

b. *Peta dengan empat peubah*

*wx* 00

|  |  |  |  |
| --- | --- | --- | --- |
| *m*0 | *m*1 | *m*3 | *m*2 |
| *m*4 | *m*5 | *m*7 | *m*6 |
| *m*12 | *m*13 | *m*15 | *m*14 |
| *m*8 | *m*9 | *m*11 | *m*10 |

01

11

10

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| *w*’*x*’*y*’*z*’ | *w*’*x*’*y*’*z* | *w*’*x*’*yz* | *w*’*x*’*yz*’ |
| *w*’*xy*’*z*’ | *w*’*xy*’*z* | *w*’*xyz* | *w*’*xyz*’ |
| *wxy*’*z*’ | *wxy*’*z* | *wxyz* | *wxyz*’ |
| *wx*’*y*’*z*’ | *wx*’*y*’*z* | *wx*’*yz* | *wx*’*yz*’ |

**Contoh**. Diberikan tabel kebenaran, gambarkan Peta Karnaugh.

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

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 |

*wx* 00

01

11

10

**Teknik Minimisasi Fungsi Boolean dengan Peta Karnaugh**

1. *Pasangan*: dua buah 1 yang bertetangga

*yz*

00 01 **1**1 **1**0

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |

*wx* 00

01

**11**

10

*Sebelum disederhanakan*: *f*(*w*, *x*, *y*, *z*) = *wxyz* + *wxyz*’ *Hasil Penyederhanaan*: *f*(*w*, *x*, *y*, *z*) = *wxy*

Bukti secara aljabar:

*f*(*w*, *x*, *y*, *z*) = *wxyz* + *wxyz*’

= *wxy*(*z* + *z*’)

= *wxy*(1)

= *wxy* 46

2. *Kuad*: empat buah 1 yang bertetangga

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

*wx* 00

01

**11**

10

*Sebelum disederhanakan*: *f*(*w*, *x*, *y*, *z*) = *wxy*’*z*’ + *wxy*’*z* + *wxyz* + *wxyz*’ *Hasil penyederhanaan*: *f*(*w*, *x*, *y*, *z*) = *wx*

Bukti secara aljabar:

*f*(*w*, *x*, *y*, *z*) = *wxy*’ + *wxy*

= *wx*(*z*’ + *z*)

= *wx*(1)

= *wx*

*wx* 00

01

11

10

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

Contoh lain:

*yz*

**0**0 **0**1 11 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | | 0 | | 0 | 0 |
| 0 | | 0 | | 0 | 0 |
|  | |  | | 0 | 0 |
|  | 1 | 1 |  |
|  | 1 | 1 |  | 0 | 0 |
|  |  |

*wx* 00

01

**1**1

**1**0

Sebelum disederhanakan: *f*(*w*, *x*, *y*, *z*) = *wxy*’*z*’ + *wxy*’*z* + *wx*’*y*’*z*’ + *wx*’*y*’z *Hasil penyederhanaan*: *f*(*w*, *x*, *y*, *z*) = *wy*’

1. *Oktet*: delapan buah 1 yang bertetangga

*yz*

00 01 11 10

*wx* 00

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | | 0 | 0 | 0 | |
| 0 | | 0 | 0 | 0 | |
|  | |  |  |  | |
|  | 1 | 1 | 1 | 1 |  |
|  | 1 | 1 | 1 | 1 |  |
|  | |  |  |  | |

01

**1**1

**1**0

*Sebelum disederhanakan*: *f*(*a*, *b*, *c*, *d*) = *wxy*’*z*’ + *wxy*’*z* + *wxyz* + *wxyz*’ +

*wx*’*y*’*z*’ + *wx*’*y*’*z* + *wx*’*yz* + *wx*’*yz*’

*Hasil penyederhanaan*: *f*(*w*, *x*, *y*, *z*) = *w*

Bukti secara aljabar:

*f*(*w*, *x*, *y*, *z*) = *wy*’ + *wy*

= *w*(*y*’ + *y*)

= *w*

*wx* 00

01

11

10

*yz*

00 01 11 10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | | 0 | | 0 | | 0 | |
| 0 | | 0 | | 0 | | 0 | |
|  | |  | |  | |  | |
|  | 1 | 1 |  |  | 1 | 1 |  |
|  | 1 | 1 |  |  | 1 | 1 |  |
|  | |  | |  | |  | |

**Contoh 5.12.** Andaikan suatu tabel kebenaran telah diterjemahkan ke dalam Peta Karnaugh. Sederhanakan fungsi Boolean yang bersesuaian sesederhana mungkin.

*yz*

00 01 11 10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | | 1 | | 1 |  |  |  |
| 1 |
| 0 | | 0 | | 0 |  | 1 |  |
|  | |  | | 0 |  | 1 |  |
|  | 1 | 1 |  |
|  | 1 | 1 |  | 0 |  | 1 |  |
|  | |  | |
|  |

*wx* 00

01

11

10

Jawab: (lihat Peta Karnaugh) *f*(*w*, *x*, *y*, *z*) = *wy*’ + *yz*’ + *w*’*x*’*z*

**Contoh 5.13.** Minimisasi fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

*yz*

00 01 11 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | | 0 | 0 | 0 | |
| 0 | | 1 | 0 | 0 | |
|  | |  |  |  | |
|  | 1 | 1 | 1 | 1 |  |
|  | 1 | 1 | 1 | 1 |  |
|  | |  |  |  | |

*wx* 00

01

11

10

Jawab: (lihat Peta Karnaugh) *f*(*w*, *x*, *y*, *z*) = *w* + *xy*’*z*

Jika penyelesaian Contoh 5.13 adalah seperti di bawah ini:

*yz*

00 01 11 10

*wx* 00

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | | 0 | 0 | 0 | |
| 0 | | 1 | 0 | 0 | |
|  | |  |  |  | |
|  | 1 | 1 | 1 | 1 |  |
|  | 1 | 1 | 1 | 1 |  |
|  | |  |  |  | |

01

11

10

maka fungsi Boolean hasil penyederhanaan adalah

*f*(*w*, *x*, *y*, *z*) = *w* + *w*’*xy*’*z* (jumlah literal = 5)

yang ternyata masih belum sederhana dibandingkan *f*(*w*, *x*, *y*, *z*) = *w* + *xy*’*z*

(jumlah literal = 4).

**Contoh 5.14.** (Penggulungan/*rolling*) Sederhanakan fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 |

*wx* 00

01

11

10

Jawab: *f*(*w*, *x*, *y*, *z*) = *xy*’*z*’ + *xyz*’ ==> belum sederhana

Penyelesaian yang lebih minimal:

*wx* 00

0**1**

1**1**

0

0

0

0

1

1

0

0

0

0

0

0

0

0

1

1

*yz*

0**0** 01 11 1**0**

10

*f*(*w*, *x*, *y*, *z*) = *xz*’ ===> lebih sederhana

**Contoh 5.11.** Sederhanakan fungsi Boolean *f*(*x*, *y*, *z*) = *x*’*yz* + *xy*’*z*’ + *xyz* +

*xyz*’.

Jawab:

Peta Karnaugh untuk fungsi tersebut adalah:

*yz*

00 01 11 10

1

1

1

1

*x*

Hasil penyederhanaan: *f*(*x*, *y*, *z*) = *yz* + *xz*’

**contoh 5.15**: (Kelompok berlebihan) Sederhanakan fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |

*wx* 00

01

11

10

Jawab: *f*(*w*, *x*, *y*, *z*) = *xy*’*z* + *wxz* + *wyz*  masih belum sederhana.

Penyelesaian yang lebih minimal:

*yz*

00 01 11 10

*wx* 00

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |

01

11

10

*f*(*w*, *x*, *y*, *z*) = *xy*’*z* + *wyz* ===> lebih sederhana

**Contoh 5.16.** Sederhanakan fungsi Boolean yang bersesuaian dengan Peta Karnaugh di bawah ini.

*cd*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

*ab* 00

01

11

10

Jawab: (lihat Peta Karnaugh di atas) *f*(*a*, *b*, *c*, *d*) = *ab* + *ad* + *ac* + *bcd*

**Contoh 5.17.** Minimisasi fungsi Boolean *f*(*x*, *y*, *z*) = *x*’*z* + *x*’*y* +*xy*’*z* + *yz*

Jawab:

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

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

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

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

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

Peta Karnaugh untuk fungsi tersebut adalah:

*yz*

00 01 11 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  | |  | | 1 |
|  | 1 | 1 |  |
| 0 |  | 1 | 1 |  | 0 |
|  | |  | |

*x* 0

1

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

Peta Karnaugh untuk lima peubah

000 001 011 010 110 111 101 100

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *m*0 | *m*1 | *m*3 | *m*2 | *m*6 | *m*7 | *m*5 | *m*4 |
| *m*8 | *m*9 | *m*11 | *m*10 | *m*14 | *m*15 | *m*13 | *m*12 |
| *m*24 | *m*25 | *m*27 | *m*26 | *m*30 | *m*31 | *m*29 | *m*28 |
| *m*16 | *m*17 | *m*19 | *m*18 | *m*22 | *m*23 | *m*21 | *m*20 |

00

01

11

10

Garis pencerminan

**Contoh 5.21.** (Contoh penggunaan Peta 5 peubah) Carilah fungsi sederhana dari *f*(*v*, *w*, *x*, *y*, *z*) =  (0, 2, 4, 6, 9, 11, 13, 15, 17, 21, 25, 27, 29, 31)

Jawab:

Peta Karnaugh dari fungsi tersebut adalah:

*xyz*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 00 |  |  | 00 | 01 | 01 | 11 | 11 | 10 |  |  | 10 |
| 0 |  |  | 1 | 1 | 0 | 0 | 1 | 1 |  |  | 0 |
| *vw*  00 | 1 |  |  |  |  | 1 | 1 |  |  |  |  | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01 |  |  |  | 1 | 1 |  |  | 1 | 1 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  | 1 | 1 |  |  | 1 | 1 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  | 1 |  |  |  |  | 1 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

Jadi *f*(*v*, *w*, *x*, *y*, *z*) = *wz* + *v*’*w*’*z*’ + *vy*’*z*

Kondisi *Don’t care*

**Tabel 5.16**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *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 | *don’t care* |
| 1 | 0 | 1 | 1 | *don’t care* |
| 1 | 1 | 0 | 0 | *don’t care* |
| 1 | 1 | 0 | 1 | *don’t care* |
| 1 | 1 | 1 | 0 | *don’t care* |
| 1 | 1 | 1 | 1 | *don’t care* |

**Contoh 5.25.** Diberikan Tabel 5.17. Minimisasi fungsi *f* sesederhana mungkin.

**Tabel 5.17**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *a* | *b* | *C* | *d* | *f*(*a*, *b*, *c*, *d*) |
| 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* |

Jawab: Peta Karnaugh dari fungsi tersebut adalah:

*cd*

00 01 11 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0 | | 1 | | 0 |
| 1 |  | |  | | 0 |
|  | 1 | 1 |  |
| *X* |  | *X* | *X* |  | *X* |
| *X* | 0 | | *X* | | *X* |

*ab*

00

01

11

10

Hasil penyederhanaan: *f*(*a*, *b*, *c*, *d*) = *bd* + *c*’*d*’ + *cd*

**Contoh 5.26.** Minimisasi fungsi Boolean *f*(*x*, *y*, *z*) = *x*’*yz* + *x*’*yz*’ + *xy*’*z*’ +

*xy*’*z*. Gambarkan rangkaian logikanya.

Jawab: Rangkaian logika fungsi *f*(*x*, *y*, *z*) sebelum diminimisasikan adalah seperti di bawah ini:

*x y z*

*x*'*yz*

*x*'*yz*'

*xy*'*z*'

*xy*'*z*

Minimisasi dengan Peta Karnaugh adalah sebagai berikut:

*yz*

00 01 11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

*x* 0

1

Hasil minimisasi adalah *f*(*x*, *y*, *z*) = *x*’*y* + *xy*’.

**Contoh 5.28.** Berbagai sistem digital menggunakan kode *binary coded decimal* (BCD). Diberikan Tabel 5.19 untuk konversi BCD ke kode *Excess*- 3 sebagai berikut:

**Tabel 5.19**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Masukan BCD | | | | Keluaran kode *Excess*-3 | | | |
| *w* | *x* | *y* | *z* | *f*1(*w*, *x*, *y*, *z*) | *f*2(*w*, *x*, *y*,*z*) | *f*3(*w*, *x*, *y*, *z*) | *f*4(*w*, *x*, *y*, *z*) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

1. *f*1(*w*, *x*, *y*, *z*)

*yz*

00 01 11 10

*wx* 00

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  | |
|  | | 1 | 1 | 1 | |
|  | |  |  |  | |
|  | *X* | *X* | *X* | *X* |  |
|  | 1 | 1 | *X* | *X* |  |

01

11

10

*f*1(*w*, *x*, *y*, *z*) = *w* + *xz* + *xy* = *w* + *x*(*y* + *z*)

1. *f*2(*w*, *x*, *y*, *z*)

*yz*

00

01

1

11

1

10

1

1

*X*

*X*

*X*

*X*

1

*X*

*X*

*wx* 00

01

11

10

*f*2(*w*, *x*, *y*, *z*) = *xy*’*z*’ + *x*’*z* + *x*’*y* = *xy*’*z*’ + *x*’(*y* + *z*)

*x*,

(c) *f*3(*w*,

*wx* 00

*y*, *z*) *yz* 00

01

11 10

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | 1 |  |
| 1 |  | 1 |  |
| *X* | *X* | *X* | *X* |
| 1 |  | *X* | *X* |

01

11

10

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

1. *f*4(*w*, *x*, *y*, *z*)

*yz*

00 01 11 10

*wx* 00 1

*X*

*X*

*X*

X

1

1

01

1

11

*X*

10 1

*X*

*f*4(*w*, *x*, *y*, *z*) = *z*’

*w x y z*



*f*4

*f*3

*f*2

*f*1

**Contoh 7.43**

Minimisasi fungsi Boolean berikut (hasil penyederhanaan dalam bentuk baku SOP dan bentuk baku POS):

*f*(*w*, *x*, *y*, *z*) =  (1, 3, 7, 11, 15)

dengan kondisi *don’t care* adalah *d*(*w*, *x*, *y*, *z*) =  (0, 2, 5)

Penyelesaian:

Peta Karnaugh dari fungsi tersebut adalah:



*yz*

*wx*

00

01 11

10

00

01

11

0

0

10

0

0

0

1

0

1

0

1

*X*

0

*X*

*X*

1

1

Hasil penyederhanaan dalam bentuk SOP

*f*(*w*, *x*, *y*, *z*) = *yz* + *w*’*z* (SOP) (garis penuh) dan bentuk baku POS adalah

*f*(*w*, *x*, *y*, *z*) = *z* (*w*’ + *y*) (POS) (garis putus2)

# Metode Quine-McCluskey

* Metode Peat Karnaugh tidak mangkus untuk jumlah peubah > 6 (ukuran peta semakin besar).
* Metode peta Karnaugh lebih sulit diprogram dengan komputer karena diperlukan pengamatan visual untuk mengidentifikasi *minterm-minterm* yang akan dikelompokkan.
* Metode alternatif adalah metode Quine- McCluskey . Metode ini mudah diprogram.

**Contoh 7.46**

Sederhanakan fungsi Boolean *f*(*w*, *x*, *y*, *z*) =  (0, 1, 2, 8, 10, 11, 14, 15).

Penyelesaian:

1. Langkah 1 sampai 5:
   1. (b) (c)

*term w x y z term w x y z term w x y z*

0 0 0 0 0  0,1 0 0 0 - 0,2,8,10 - 0 - 0

0,2 0 0 - 0  0,8,2,10 - 0 - 0

1 0 0 0 1  0,8 - 0 0 0 

2 0 0 1 0  10,11,14,15 1 - 1 -

8 1 0 0 0  2,10 - 0 1 0  10,14,11,15 1 - 1 -

8,10 1 0 - 0 

10 1 0 1 0 

10,11 1 0 1 - 

11 1 0 1 1  10,14 1 - 1 0 

14 1 1 1 0 

11,15 1 - 1 1 

15 1 1 1 1  14,15 1 1 1 - 

(i) Langkah 6 dan 7:

*minterm*

Bentuk prima 0 1 2 8 10 11 14 15

 0,1  

 0,2,8,10    

 10,11,14,15    

\* \* \* \* \* \*

       

Bentuk prima yang terpilih adalah:

|  |  |  |
| --- | --- | --- |
| 0,1 | yang bersesuaian dengan *term* | *w*’*x*’*y* |
| 0, 2, 8, 10 | yang bersesuaian dengan *term* | *x*’*z*’ |
| 10, 11, 14, 15 | yang bersesuaian dengan *term* | *wy* |

Semua bentuk prima di atas sudah mencakup semua *minterm* dari fungsi Boolean semula. Dengan demikian, fungsi Boolean hasil penyederhanaan adalah *f*(*w*, *x*, *y*, *z*) = *w*’*x*’*y*’ + *x*’*z*’ + *wy*.

**Contoh 7.47**

Sederhanakan

Penyelesaian: fungsi Boolean *f*(*w*, *x*, *y*, *z*) =  (1,4,6,7,8,9,10,11,15)

1. Langkah 1 sampai 5:
   1. (b) (c)

*term w x y z term w x y z term w x y z*

1 0 0 0 1  1,9 - 0 0 1 8,9,10,11 1 0 - -

4 0 1 0 0  4,6 0 1 - 0 8,10,9,11 1 0 - -

8 1 0 0 0  8,9 1 0 0 - 

8,10 1 0 - 0 

6 0 1 1 0 

9 1 0 0 1  6,7 0 1 1 -

10 1 0 1 0  9,11 1 0 - 1 

10,11 1 0 1 - 

7 0 1 1 1 

11 1 0 1 1  7,15 - 1 1 1

11,15 1 - 1 1

15 1 1 1 1 

1. Langkah 6 dan 7

*minterm*

Bentuk prima 1 4 6 7 8 9 10 11 15

 1,9  

 4,6  

6,7  

7,15  

11,15  

 8,9,10,11    

\* \* \* \*

      

Sampai tahap ini, masih ada dua *minterm* yang belum tercakup dalam bentuk prima terpilih, yaitu 7 dan 15. Bentuk prima yang tersisa (tidak terpilih) adalah (6,7), (7,15), dan (11, 15). Dari ketiga kandidat ini, kita pilih bentuk prima (7,15) karena bentuk prima ini mencakup *minterm* 7 dan 15 sekaligus.

*minterm*

Bentuk prima

1 4 6 7 8 9 10 11 15

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1,9 |  |  |  |  |  |  |  | | |
|  | 4,6 |  |  |  |  |  |  |
|  | 6,7 |  |  |  |  |  |  |
|  | 7,15 |  |  |  |  |  |  |  |  |  |
|  | 11,15 |  |  |  |  |  |  |  |  |  |
|  | 8,9,10,11 |  |  |  |  |  |  |  |  |  |
|  |  | \* | \* |  |  | \* |  | \* |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Sekarang, semua *minterm* sudah tercakup dalam bentuk prima terpilih. Bentuk prima yang terpilih adalah:

|  |  |  |
| --- | --- | --- |
| 1,9 | yang bersesuaian dengan *term* | *x*’*y*’*z* |
| 4,6 | yang bersesuaian dengan *term* | *w*’*xz*’ |
| 7,15 | yang bersesuaian dengan *term* | *xyz* |
| 8,9,10,11 | yang bersesuaian dengan *term* | *wx*’ |

Dengan demikian, fungsi Boolean hasil penyederhanaan adalah *f*(*w*, *x*, *y*, *z*) = *x*’*y*’*z* + *w*’*xz*’ + *xyz* + *wx*’.

## **Latihan soal**

1. Implementasikan fungsi *f*(*x*, *y*, *z*) =  (0, 6) dan hanya dengan gerbang NAND saja.
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).
3. Sebuah instruksi dalam sebuah program adalah

**if** A > B **then writeln**(A) **else writeln**(B);

Nilai *A* dan *B* yang dibandingkan masing-masing panjangnya dua bit (misalkan *a*1*a*2 dan *b*1*b*2).

* 1. Buatlah rangkaian logika (yang sudah disederhanakan tentunya) yang menghasilkan keluaran 1 jika *A* > *B* atau 0 jika tidak.
  2. Gambarkan kembali rangkaian logikanya jika hanya menggunakan gerbang *NAND* saja (petunjuk: gunakan hukum de Morgan)

5. Buatlah rangkaian logika yang menerima masukan dua-bit dan menghasilkan keluaran berupa kudrat dari masukan. Sebagai contoh, jika masukannya 11 (3 dalam sistem desimal), maka keluarannya adalah 1001 (9 dalam sistem desimal).