3 Teknik Reduksi

Bahasan :

3 Teknik Reduksi

3.1 Tujuan :

Setelah mempelajari Topik ini, anda diharapkan dapat :

- Dapat menerjemahkan suatu rangkaian menjadi suatu persamaan, atau sebaliknya menerjemahkan suatu persamaan menjadi suatu rangkaian.
- Dapat menyederhanakan persamaan menggunakan aljabar boolean.
- Memahami prinsip de Morgan dan menggunakannya dalam system reduksi “Buble”
- Dapat menyederhanakan persamaan menggunakan peta Karnough.

3.2 Pendahuluan

Dalam Bab ini, anda akan lebih memperdalam sistem logika kombinasi. Logika kombinasi merupakan suatu rangkaian digital yang mempergunakan 2 atau lebih gerbang-gerbang logika. Kombinasi beberapa gerbang logika dapat menjadi suatu rangkaian digital yang sangat komplek. Pada dasarnya kompleksitas suatu rangkaian digital dapat diserderhanakan sehingga rangkaian digital tersebut dapat memanfaatkkan gerbang yang lebih sedikit.

Penyederhanaan rangkaian digital tersebut dikenal sebagai teknik reduksi. Teknik Reduksi yang akan dibahas dalam bab ini antara lain : Teknik reduksi menggunakan aljabar Bolean, Teknik reduksi menggunakan teorema de morgan, dan teknik reduksi menggunakan peta karnough.

Selain itu, sebelum membahas teknik reduksi lebih mendalam, dalam bab ini akan dibahas konversi rangkaian digital menjadi suatu persamaan logika dan konversi suatu persamaan logika menjadi suatu rangkaian digital.

3.3 Rangkaian Digital dan Persamaan Digital

Dalam sub bab ini, kita akan mempelajari konversi rangkaian digital menjadi suatu persamaan logika dan konversi suatu persamaan logika menjadi suatu rangkaian digital. Sebagai dasar menusun dan menyederhanakan rangkaian.

Suatu rangkaian digital sebenarnya merupakan realisasi sistem yang bekerja secara digital, sistem digital secara sederhana dapat dinyatakan sebagai suatu sistem ON/OFF yaitu sistem yang bekerja dengan 2 kondisi, yaitu HIDUP atau MATI.

Contoh 1:

Buatlah suatu rangkaian digital dari sistem alarm rumah yang mempunyai sistem kerja sebagai berikut :

Sistem alarm (AL) akan aktif bila:

Jika di dalam rumah ada asap (A) dan suhu naik (S) dari kondisi suhu rata-rata. Asap di dalam rumah dideteksi oleh sensor asap, sedangkan suhu dideteksi oleh sensor suhu.

Atau

Jika Sistem kunci rumah aktif (K) dan ada jendela yang terbuka (J).

Penyelesaian 1:

Dengan mengingat pembahasan simbol dalam bab 1, maka kondisi di atas dapat diterjemahkan sebagai berikut ;

Kondisi 1 : ada asap (A) DAN suhu naik (S) $\rightarrow$ A . S

Kondisi 2 : Kunci aktif (K) DAN jendela terbuka (J) $\rightarrow$ K . J

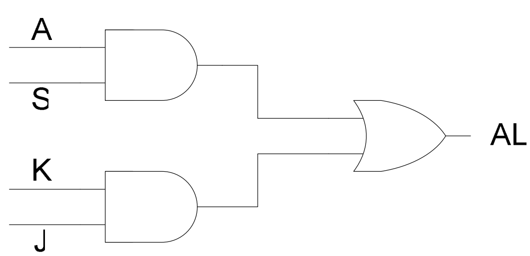
Sistem alarm aktif (AL) jika kondisi 1 ATAU kondisi 2 $\rightarrow$ kondisi 1 + kondisi 2

Jadi AL = kondisi 1 + kondisi 2

AL = A . S + K . J

Jadi pernyataan di atas dapat diterjemahkan menjadi persamaan AL = A . S + K . J

Persamaan AL dapat diubah menjadi suatu rangkaian logika kombinasi dengan mengubah tanda (.) dengan gerbang AND, sedangkan tanda (+) diubah dengan gerbang OR, sehingga persamaan di atas merupakan rangkaian sederhana yang memanfaatkan 2 gerbang AND dan 1 gerbang OR. Persamaan di atas dapat digambarkan menjadi :



Gambar 3.1 Rangkaian penyelesaian contoh 1

Contoh 2 :

Konversi persamaan **(A + B)’ . C + C’.B** menjadi rangkaian digital.

Penyelesaian 2:

Untuk membuat rangkaian di atas maka tanda (**+**) atau (**.**) atau (‘) diganti dengan gerbang-gerbang logika, yaitu :

Tanda (+) diganti dengan gerbang OR

Tanda (.) diganti dengan gerbang AND

Tanda (‘) diganti dengan gerbang NOT,

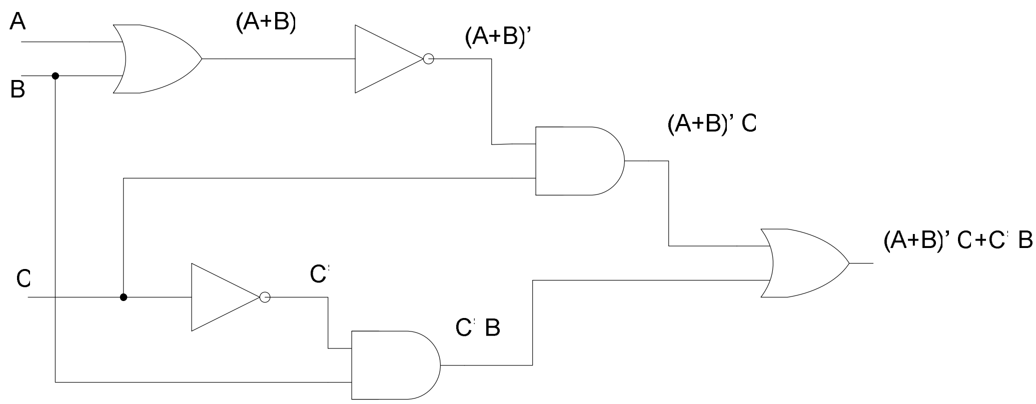
Selain itu urutan prioritas tanda akan sangat mempengaruhi rangkaian. Urutan prioritas ini adalah

1. (') atau NOT,
2. (.) atau AND,
3. (+) atau OR

sehingga persamaan tersebut dapat diubah menjadi :

{(**NOT** (A **OR** B)) **AND** C} **OR** {(**NOT** C) **AND** B}

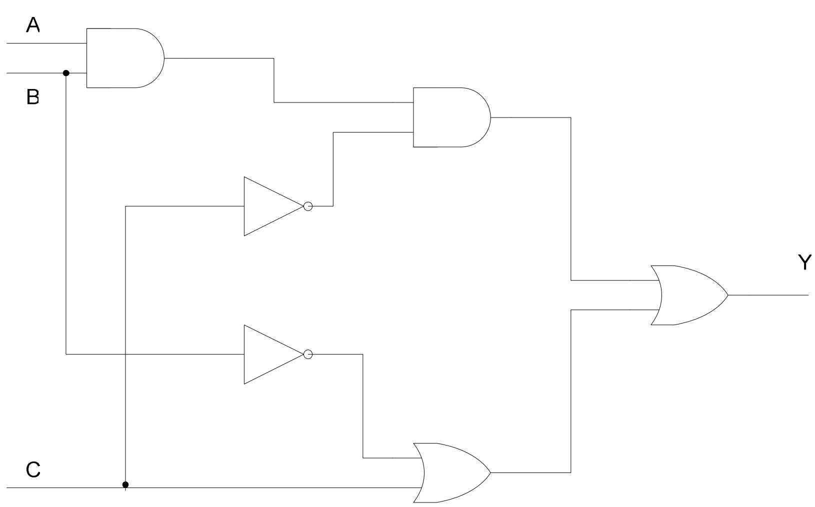
sehingga dapat digambarkan menjadi sebagaimana gambar 3.2.



Gambar 3.2 Rangkaian konversi untuk Contoh 3.2

Contoh 3 :

Konversi rangkaian berikut menjadi persamaan logika



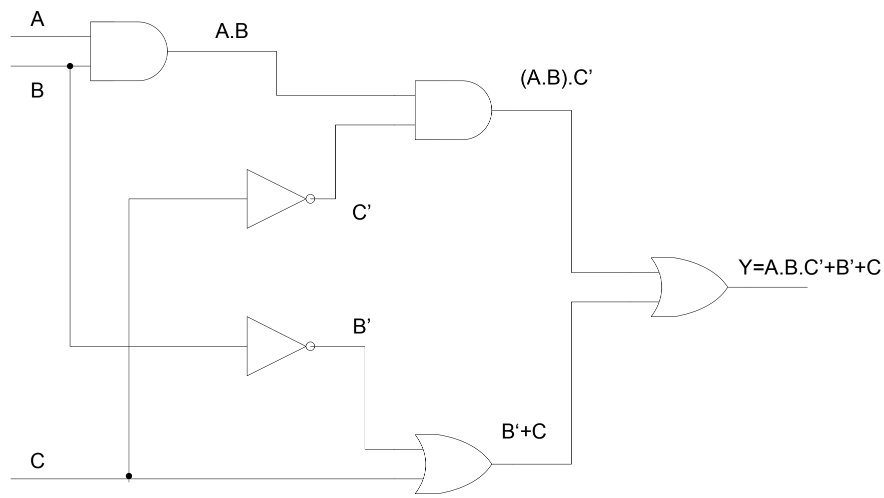
Gambar 3.3 Rangkaian Untuk Contoh 3.2

Penyelesaian 3:

Untuk membuat persamaan logika dari rangkaian digital di atas dilakukan langkah-langkah sebagai berikut ;

1. memberi persamaan pada keluaran masing-masing gerbang logika berdasar masukannya, urut mulai dari depan.
2. mengkonversi persamaan akhir dari gerbang terakhir dengan menghilangkan tanda kurung yang tidak berfungsi berdasar urutan prioritas.

Dari rangkaian di atas diperoleh keluaran gerbang-gerbang sebagai berikut ;



Gambar 3.4 Rangkaian penyelesaian contoh 3

sehingga diperoleh persamaan dari gerbang terakhir adalah ((A.B).C')+(B'+C) = Y

dari persamaan tersebut tanda kurung dapat dieliminasi, sebagai berikut :

((A . B) . C') dapat dieliminasi menjadi (A . B . C'), karena mempunyai prioritas yang sama (bersifat komutatif), kemudian

(A . B . C') + (B' + C) dapat dieliminasi menjadi (A . B . C') + B ' + C, kemudian

karena perkalian mempunyai prioritas mendahului, maka persamaan dapat disederhanakan menjadi A.B.C' + B' + C, sehingga diperoleh persamaan akhir :

Y = A . B . C' + B' + C

3.4 Aljabar Boolean

Salah satu teori dasar teknik reduksi adalah aljabar Boolean. Sebagaimana matematika aljabar, aljabar boolean mempunyai hukum-hukum dan aturan-aturan. Dalam aljabar Boolean ada 3 hukum dan 10 aturan yang digunakan.

3.4.1 Hukum Aljabar Boolean

Hukum aljabar Boolean antara lain :

1. Hukum Komutatif

- Pada perkalian

  A . B = B . A

- Pada penjumlahan

  A + B = B + A

2. Hukum Assosiatif

- Pada perkalian

  A . (B . C) = (A . B) . C

- Pada penjumlahan

  A + (B + C) = (A + B) + C

3. Hukum Distributif

- Distributif 1

  A . (B + C) = A . B + A . C

- Distributif 2

  (A + B) . (C + D) = A . C + A . D + B . C + B . D

3.4.2 Aturan – aturan Aljabar Boolean

Aturan- aturan dalam aljabar Boolean antara lain :

1. A . 1 = A

2. A . 0 = 0

3. A + 1 = 1

4. A + 0 = A

5. A . A = A

6. A + A = A

7. A . A’ = 0

8. A + A’ = 1

9. (A’) ’ = A

10. a. A + A’ . B = A + B

b. A ‘ + A . B = A’ + B

Aturan – aturan di atas (Aturan 1 – 9) dapat dibuktikan dengan menggunakan tabel kebenaran sebagai berikut :

A	A’	A . 1	A . 0	A + 1	A + 0	A.A	A + A	A . A’	A + A’	(A’)’
0	1	0	0	1	0	0	0	0	1	0
1	0	1	0	1	1	1	1	0	1	1

Hasil	A	0	1	A	A	A	0	1	A
Aturan	1	2	3	4	5	6	7	8	9

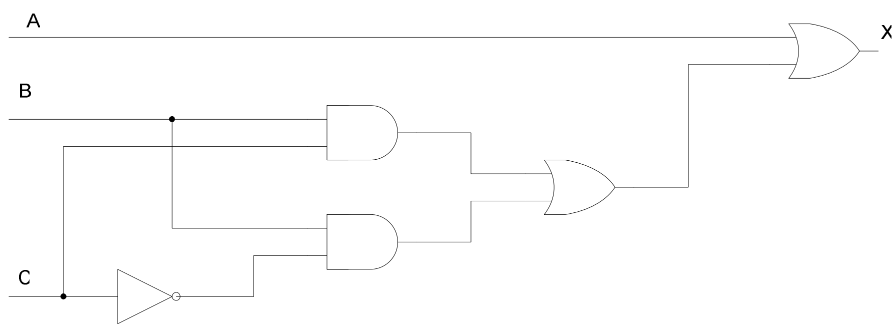
Sedangkan untuk aturan ke 10.a dan 10.b dapat dibuktikan dengan tabel kebenaran sebagai berikut :

A	**B**	**A'**	**B'**	**A' . B**	**A + A' . B**	**A + B**	**A . B**	**A' + A . B**	**A' + B**
0	0	1	1	0	0	0	0	1	1
0	1	1	0	1	1	1	0	1	1
1	0	0	1	0	1	1	0	0	0
1	1	0	0	0	1	1	1	1	1
Hasil					A + A'.B = A + B			A' + A.B = A' + B	
Aturan					10.a			10.b	

Aplikasi aljabar boolean dalam menyelesaikan suatu persamaan logika dapat dilihat dalam contoh berikut.

<u>Contoh 4.</u>

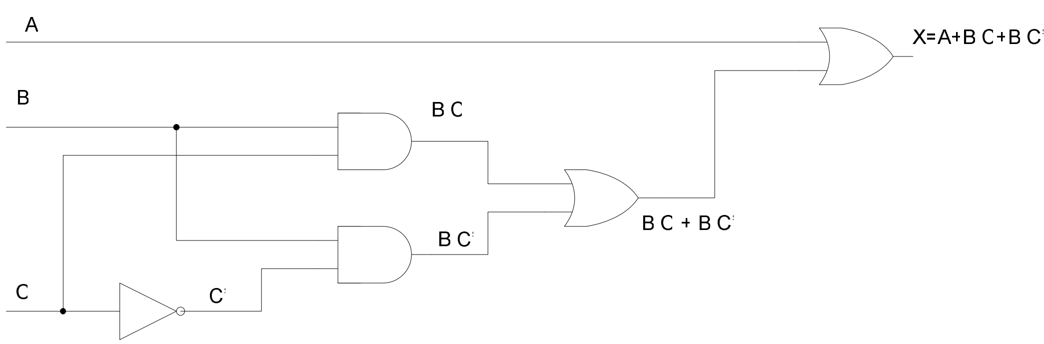
Suatu Rangkaian Digital diperlihatkan dalam gambar 3.5. sederhanakan rangkaian tersebut menggunakan persamaan aljabar boolean.



Gambar 3.5 Rangkaian untuk contoh 4

<u>Penyelesaian 4:</u>

Dengan menggunakan cara sebagaimana Contoh 3, maka diperoleh keluaran masing-masing gerbang sebagaimana gambar 3.6.



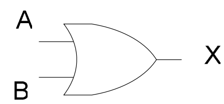
Gambar 3.6 Rangkaian contoh 4 dengan persamaan logika untuk masing-masing keluaran gerbang

Dari Gambar 3.6 diperoleh persamaan akhir : X = A + B.C + B.C'

Penyederhanaan menggunakan aljabar Boolean :

X = A + B.C + B.C'
= A + B.**(C + C')** {aturan 8}
= A + B.1 {aturan 1}
= A + B

jadi dari hasil penyederhanaan diperoleh persamaan X = A + B, yang diwakili oleh satu gerbang OR, dengan 2 masukan A dan B.



Gambar 3.7 Rangkaian penyederhanaan contoh 4

Contoh 5 :

Sederhanakan persamaan logika berikut ini :

a. Y = A + A'.B
b. Z = A' + A.B

Penyelesaian 5 :

a. Y = A + A'.B
= A.**1** + A'.B {aturan 1}
= A.**(B + B')** + A'.B {aturan 8}
= A.B + A.B' + A'.B {hukum asosiatif}
= **A.B + A.B** + A.B' + A'.B {aturan 6}
= A.B + A.B' + A.B + A'.B {hukum komutatif}
= A.**(B + B')** + **(A + A')**.B {hukum asosiatif, aturan 8}
= A.**1** + **1**.B {aturan 1}
= A + B

b. Z = A' + A.B
= A'.**1** + A.B {aturan 1}
= A'.**(B + B')** + A.B {aturan 8}
= A'.B + A'.B' + A.B {hukum asosiatif}
= **A'.B + A'.B** + A'.B' + A.B {aturan 6}
= A'.B + A'.B' + A'.B + A.B {hukum komutatif}
= A'.**(B+B')** + **(A'+A)**.B {hukum asosiatif, aturan 8}
= A'.**1** + **1**.B {aturan 1}
= A' + B

3.5 Teorema De Morgan

Teorema de Morgan dinyatakan sebagai :

$\overline{A+B} = \overline{A} \cdot \overline{B}$ atau **(A + B)' = A' . B'**

$\overline{A \cdot B} = \overline{A} + \overline{B}$ atau **(A . B)' = A' + B'**

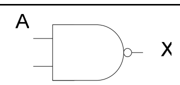
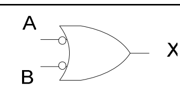
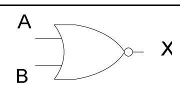
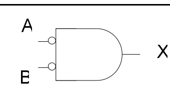
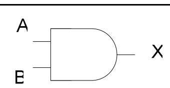
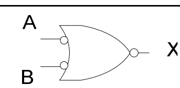
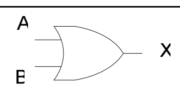
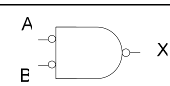
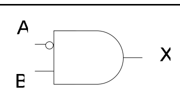
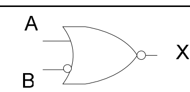
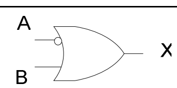
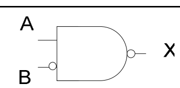
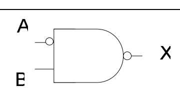
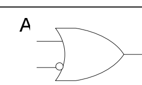
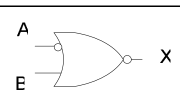
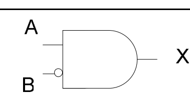
persamaan di atas merupakan persamaan de Morgan dengan 2 variabel. Untuk persamaan de Morgan yang lebih dari 2 variabel, juga menggunakan pinsip yang sama, yaitu (misal untuk 3 variabel) :

$\overline{A+B+C} = \overline{A} \cdot \overline{B} \cdot \overline{C}$ atau **(A + B + C)' = A' . B' . C'**

$\overline{A \cdot B \cdot C} = \overline{A} + \overline{B} + \overline{C}$ atau **(A . B . C)' = A' + B' + C'**

Persamaan de Morgan merupakan persamaan yang unik, karena dengan menggunakan persamaan de Morgan maka semua logika dasar (OR, AND dan NOT) akan dipakai secara bersamaan. Dengan keistimewaan ini, maka persamaan de Morgan dapat juga digunakan untuk menyederhanakan suatu rangkaian logika kombinasi.

Berikut ditampilkan gerbang-gerbang dengan menggunakan prinsip de Morgan, yang dikenal sebagai teknik reduksi "buble"

Y = (A.B)'		⇔		X = A' + B'
Y = (A+B)'		⇔		X = A' . B'
Y = A.B		⇔		X = (A' + B')'
Y = A+B		⇔		X = (A' .B')'
Y = A'.B		⇔		X = (A + B')'
Y = A'+B		⇔		X = (A . B')'
Y = (A'.B)'		⇔		X = A + B'
Y = (A'+B)'		⇔		X = A.B'

Contoh 6:

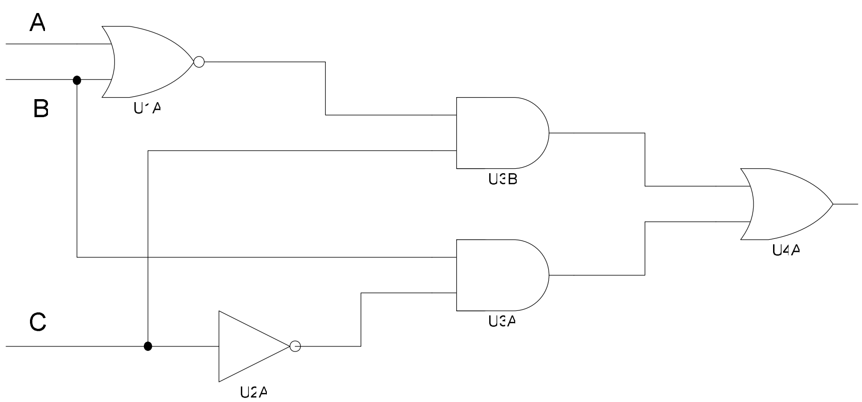
Buktikan bahwa $(A'.B + A.B')' = A.B + A'B'$

Penyelesaian 6:

$(A'.B + A.B')'$	$= (A'.B)'.(A.B')'$	{hukum de Morgan}
	$= ((A')'+B') \;.\; (A'+(B')')$	{hukum de Morgan, aturan 9}
	$= (A+B').(A'+B)$	{aturan 9}
	$= A.A' + A.B + B'A' + B'B$	{hukum distributif}
	$= 0 + AB + B'A' + 0$	{aturan 7}
	$= AB + A'B'$	{hukum komutatif}

Contoh 7:

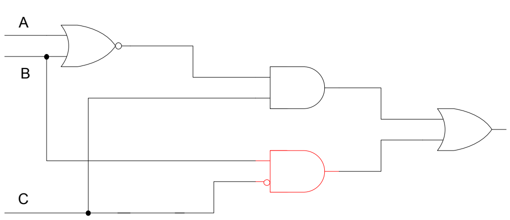
Reduksilah rangkaian gambar 3.8 dengan menggunakan teknik reduksi bubble.



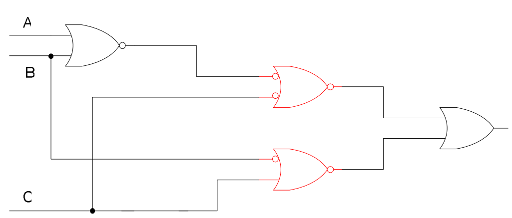
Gambar 3.8 Rangkaian untuk contoh 7

Penyelesaian 7:

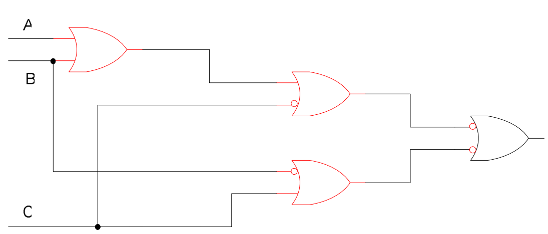
Ubah menjadi rangkaian bubble dengan urutan sebagai berikut :



(a) Menghapus gerbang NOT (U2A) menjadi masukan buble U3A



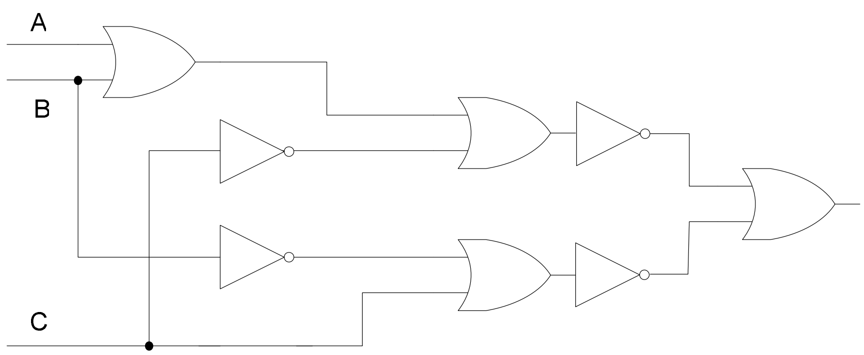
(b) mengubah gerbang AND (U3A dan U3B) menjadi NOR



(c) memindah bubble keluaran NOR (U3A & U3B) menjadi masukan OR (U4A), menghapus bubble keluaran U1A dan bubble masukan U3B

Hasil akhir rangkaian adalah sebagaimana Gambar 3.9. Dari hasil akhir dengan kondisi awal dapat diambil kesimpulan :

- Pada kondisi awal digunakan 5 gerbang dasar dengan 4 buah IC (OR, AND, NOR, dan NOT)
- Pada kondisi akhir digunakan 8 buah komponen, tetapi IC yang digunakan hanya 2 (NOT dan OR)
- sehingga dengan menggunakan metode bubble, dapat menghemat IC sekaligus menghemat biaya, karena dalam suatu IC 7432 terdapat 4 buah gerbang OR, dan 7404 terdapat 6 gerbang NOT.



Gambar 3.9 Hasil akhir contoh 7

3.6 Peta Karnough

Peta karnough merupakan metode penyederhanaan persamaan logika secara grafis. Penyelesaian dilakukan dengan mengisi sel-sel peta karnough. Tiap-tiap sel peta karnough yang bersisian hanya memiliki 1variabel yang berubah, misal dalam suatu persamaan logika ada 4 variabel (ABCD), maka dibagi 2 vertikal dan horisontal (AB\CD) sehingga terbentuk 16 sel sebagaimana dalam contoh 8, Jika dalam persamaan logika ada 5 variabel, maka bisa dibagi vertikal dan horisontal (ABC\DE atau AB\CDE).1 variabel berubah dari 3 variabel, misal : 000,001,011,010,110,111,101,100.

Langkah-langkah penyederhanaan menggunakan peta karnough adalah sebagai berikut :

1. Ubah persamaan logika menjadi persamaan logika SOP (*Sum of Product*, Penjumlahan dari persamaan perkalian).
2. kelompokkan sel-sel yang berlogika 1 yang bersebelahan secara ber-2, ber-4, ber-8, atau ber-16. kelompokkan sel tersebut dengan ukuran yang terbesar.
3. carilah persamaan dari kelompok tersebut yang variabelnya tidak berubah, tulis dalam bentuk perkalian.
4. Jika terdapat lebih dari 1 kelompok, maka persamaan masing-masing kelompok dijumlahkan.

Contoh 8:

Sederhanakan persamaan berikut dengan menggunakan metode peta karnough

a. W = (A'B + AB).C' + A'BC + ABC +AB'D

b. X = A'B'D' + B'.(A'D + AC')

c. Y = B.(A'C'D + CD' +AC')

d. Z = B'.(A'D' + AC' + ACD')

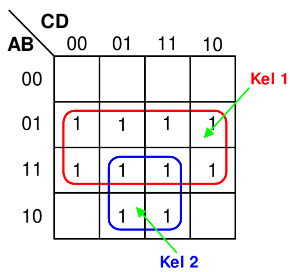
Penyelesaian 8:

a. W = (A'B + AB).C' + A'BC + ABC +AB'D

= A'BC' + ABC' + A'BC + ABC + AB'D

= A'BC' + A'BC + ABC' + ABC + AB'D

= A'BC'D' + A'BC'D + A'BCD + A'BCD' + ABC'D' + ABC'D + ABCD + ABCD' + AB'C'D + AB'CD



Kel. 1 : B

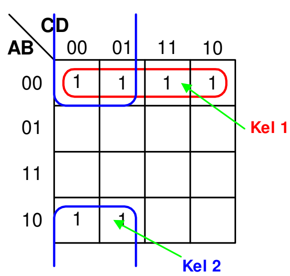
Kel. 2 : AD

Sehingga W = B + A.D

b. X = A'B'D' + B'.(A'D + AC')

= A'B'D' + A'B'D + AB'C'

= A'B'C'D' + A'B'CD' + A'B'C'D + A'B'CD + AB'C'D' + AB'C'D



Kel 1 : A'.B'

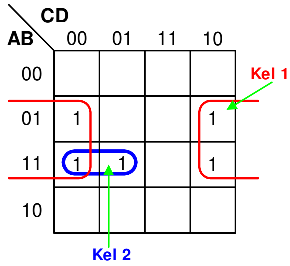
Kel 2 : B'.C'

Sehingga : X = A'B' + B'.C' = B'.(A' + C')

c. Y = B.(A'C'D + CD' +AC')

= A'BC'D + BCD' + ABC'

= A'BC'D' + A'BCD' + ABC'D' + ABCD' + ABC'D



Kel 1 : B.D’

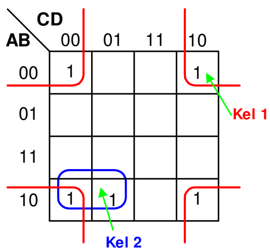
Kel 2 : A.B.C’

Sehingga : Y = B.D’ + A.B.C’

d. Z = B’.(A’D’ + AC’ + ACD’)

= A’B’D’ + AB’C’ + AB’CD’

= A’B’C’D’ + A’B’CD’ + AB’C’D’ + AB’C’D+ AB’CD’



Kel 1 : B’.D’

Kel 2 : A.B’.C’

Sehingga : Z = B’D’ + A.B’.C’