**Analisa dan Sintesa Rangkaian Logika**

**I. Umum**

Analisa rangkaian logika adalah suatu proses untuk mempelajari perilaku/karakteristik dari rangkaian logika. Berikut langkah-langkah proses analisa dari rangkaian logika :

* Menurunkan persamaan fungsi dari rangkaian logika tersebut yang menyatakan hubungan antara output dengan inputnya.
* Menyusun tabel kebenaran yang menginterprestasikan fungsi dari rangkaian logika tersebut.

**II. Penurunan persamaan fungsi Rangkaian logika**

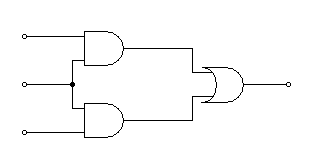
**biasanya terdiri dari :**

* Gerbang-gerbang logika yang berlainan
* Gerbang-gerbang logika yang sama

Rangkaian terdiri dari gerbang-gerbang logika yang berlainan.

Penurunan persamaan fungsi dari rangkaian ini dilakukan penelusuran sinyal/variabel dari sisi input ke sisi output.

Contoh :



A.B

Y =

A.C + A.B

A.C

B

A

C

Rangkaian Gerbang Logika Kombinasi

Jadi persamaan fungsi rangkaian logika di atas adalah Y = AC + AB

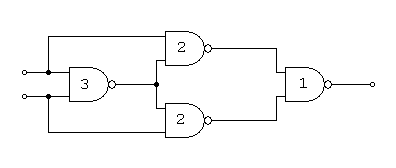
**Rangkaian terdiri dari Gerbang-gerbang Logika yang sama**

Penurunan persamaan fungsi dari rangkaian dari gerbang logika yang sama dilakukan **penelusuran dari sisi output ke sisi input**. Posisi gerbang gerbang logikanya dikelompokan dalam posisi ganjil atau genap dihitung dari sisi output ke sisi input dengan penomoran 1,2,3,4.....dst.

Rangkaian seperti ini biasanya terdiri dari gerbang-gerbang NAND atau gerbang-gerbang NOR. Adapun peraturan atau persyaratan yang harus diperhatikan dalam penurunan persamaan fungsi dari rangkaian logika yang mempunyai gerbang yang sama.

**A. Aturan Rangkaian dengan Gerbang-gerbang NAND**

1. Gerbang-gerbang NAND ditentukan posisinya, dimana NAND bisa di posisi ganjil atau genap.
2. Gerbang NAND pada posisi ganjil berfungsi sebagai gerbang OR, sedangkan di posisi genap berfungsi sebagai gerbang AND.
3. Setiap variabel yang langsung masuk ke gerbang NAND posisi ganjil, outputnya harus dikomplemenkan.



A

B

d

c

b

a

Rangkaian logika dengan gerbang-gerbang NAND

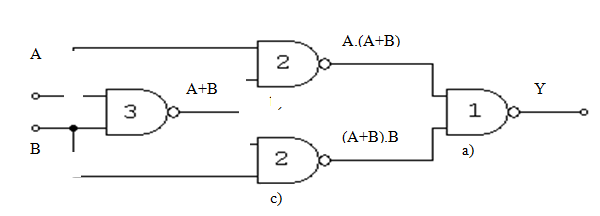
Di atas adalah rangkaian gerbang-gerbang NAND. Untuk rangkaian seperti ini penelusuran dari sisi output ke sisi input. Diasumsikan :

1. gerbang ( b ) dan ( c ) posisi genap dan berfungsi sebagai AND
2. gerbang ( a ) dan ( d ) posisi ganjil dan berfungsi sebagai OR
3. dan variabel input pada gerbang NAND di gerbang ( d ), outputnya dikomplemenkan karena langsung masuk ke gerbang posisi ganjil.

Sehingga output rangkaian atau persamaan fungsinya adalah



Contoh Lain : Turunkan Ekspresi Aljabar untuk rangkaian NAND Berikut ini :



d

c

a

b

B . C

C.D

A . (B + C )

B+C

1

2

2

2/3

D

C

A

B



Pada rangkaian tersebut yang perlu dperhatikan adalah posisi gerbang (C) yang dapat memiliki posisi ganjil dan dapat juga pada posisi genap, tergantung dari sinyal yang melewatinya.

* **Sinyal 1 Akan melewati :**

gerbang (a) : posisi ganjil f (gerbang OR),

gerbang (b) : posisi genap f (gerbang AND), dan

gerbang (c) : posisi ganjil f (gerbang OR yang inputnya

dikomplemenkan).

* **Sinyal 2 Akan melewati :**

gerbang (c) : posisi genap f (gerbang AND).

* **Sinyal 3 Akan melewati :**

gerbang (d) : posisi ganjil f (gerbang AND).

Ekspresi aljabar rangkaian NAND diatas adalah :

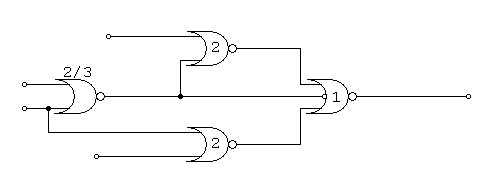
Y = 1 + 2 + 3



**B. Aturan Rangkaian dengan Gerbang-gerbang NOR**

1. Gerbang-gerbang NOR ditentukan posisi genap atau ganjil.
2. Gerbang NOR pada posisi ganjil berfungsi sebagai gerbang AND.
3. Grbang NOR pada posisi genap berfungsi sebagai gerbang OR.
4. Setiap variabel yang masuk langsung ke gerbang NOR pada posisi ganjil, outputnya harus dikomplemenkan.

Sebagai contoh : Rangkaian dengan Gerbang-gerbang NAND.



D

C

B

A

Rangkaian logika dengan gerbang-gerbang NOR.

Penurunan persamaan fungsi pada rangkaian dengan gerbang-gerbang NOR di atas sama dengan rangkaian gerbang-gerbang NAND, yang beda adalah pada posisi ganjil atau genap fungsi gerbang apa yang dipakai.

1. Gerbang logikanya dikelompokan posisi genap atau ganjil dalam nomor 1,2,3,4........dst
2. Variabel input yang masuk pada posisi ganjil outputnya harus dikomplemenkan.
3. Gerbang (1) = posisi ganjil berfungsi sebagai gerbang AND.
4. Gerbang (2) = posisi genap berfungsi sebagai gerbang OR.
5. Gerbang (3) = posisi ganjil terhadap gerbang (2) fungsinya sebagai AND,
6. Gerbang (2) = posisi genap terhadap gerbang (1) fungsinya sebagai OR.

Sehingga persamaan fungsi dari rangkaian di atas adalah



**III. Menyusun Tabel Kebenaran**

Menyusun Tabel kebenaran suatu fungsi logika adalah dengan cara mendaftarkan semua kombinasi nilai variabel input, yang disusun dengan urutan nilai ekivalen desimalnya yang dimulai dari angka 0 sampai nilai tetinggi yang sesuai dengan karakteristik atau perilaku dari rangkaian logikanya.

Nilai variabel output diisi sesuai dengan persamaan fungsi logikanya.

Sebagai contoh Y = AB+AB , maka tabel kebenarannya,

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Y** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Dari tabel kebenaran diatas dapat dilihat perilaku rangkaian logikan yang bersangkutan bahwa rangkaian akan memberikan output 1, jika kedua inputnya berbeda satu sama lainnya, dan akan memberikan output 0, jika semua input sama.

1. **SINTESA RANGKAIAN LOGIKA**
2. **Umum**

Sintesa rangakaian logika sama dengan proses analisa, tetapi sintesa adalah kebalikan dari analisa rangkaian logika, jadi dibalik. Diawali menyusun tabel kebenaran yang merupakan terjemahan dari perilaku rangkaian logika yang dipersyaratkan kedalam nilai variabel input dan output. Kemudian diturunkan persamaan fungsi logikanya yang merupakan hubungan antara variabel input dan output.

Yang kemudian diimplementasikan kedalam rangkaian logika dengan menggunakan gerbang-gerbang logika.

**b**  **Penterjemahan Perilaku yang Dipersyaratkan Kedalam Tabel Kebenaran.**

Dalam hal ini Tabel Kebenaran disusun berdasarkan persyaratan perilaku yang ditentukan bagi Rangkaian Logika yang akan dibuat.

Contoh : Tabel Kebenaran

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **Y** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1  **A B C** |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1    **A B C** |
| 1 | 1 | 0 | 1  **A B C** |
| 1 | 1 | 1 | 1  A B C |

Tabel Kebenaran diatas akan dibuat suatu rangkaian logika yang memiliki 3 buah input dan 1 buah output. Dengan menggunakan **Metode Standard Sum of Product.**

1. Rangkaian akan memberikan output berlogik / bernilai = 1, jika mayoritas inputnya berlogik/bernilai = 1.
2. Nilai kebenaran variabel output (Y) = 1, jika :

* Pada baris-baris dimana 2 dari ketiga variabel

inputnya berlogik/bernilai =1. atau

* Ketiga variabel inpunya berlogik/bernilai =1

1. Dari Tabel Kebenaran ini kemudian diturunkan persamaan / ekspresi Aljabarnya sebagai berikut :



**3.3 MINIMISASI RANGKAIAN LOGIKA**

Ada beberapa proses minimisasi suatu rangkaian logika tetapi yang paling banyak digunakan adalah

1. Manipulasi Aljabar Boolen
2. Peta karnaug

**a** **Manipulasi Aljabar Boolen**

Manipulasi aljabar boolen dilakukan dengan menerapkan berbagai teorema aljabar untuk mengeliminir beberapa variabel.

**Tujuannya adalah**

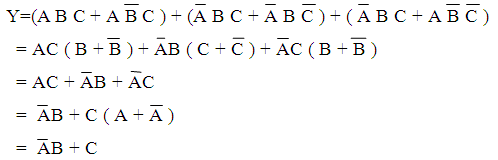
* Mengurangi jumlah suku
* Mengurangi Variabel

Dimana semua tujuan diatas agar harga rangkaian menjadi lebih murah dengan tidak mengubah rangkaian tersebut.

Contoh : suatu persamaan rangkaian logikanya sebagai berikut :



Persamaan y terdiri dari 5 suku product, berarti membutuhkan 5 buah gerbang AND terus membutuhkan 1 buah gerbang OR. Dengan menggunakan teorema komplementasi aljabar boolen X+X =1, persamaan diatas dapat dimanipulasi sbb :



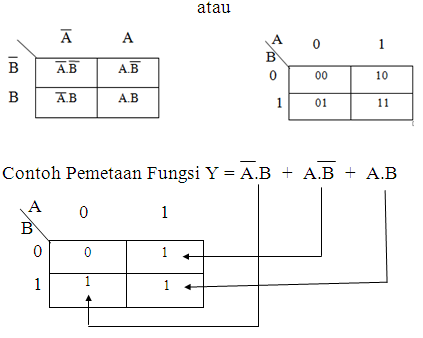
Terlihat jumlah suku productnya berkurang dari 5 suku menjadi 1 suku, jadi gerbang AND yang diperlukan berkurang 4 buah dan jumlah variabel dalam sukuk berkurang dari 3 variabel menjadi 2 variabel. Terminal input pada gerbang AND berkurang, sehingga harga rangkaian lebih murah.

**b Metoda Peta Karnaugh**

Metoda ini sama didasari teorema komplementasi X+X = 1. Berarti apabila suatu variable muncul bersamaan dengan komplemennya dalam peta Karnaugh, maka nilainya = 1, atau variablenya hilang. Peta Karnaugh terdiri atas kotak-kotak untuk memetakan Kombinasi nilai dari variable variable yang ada jumlahnya = 2, dimana n jumlah variabel.

1. **Peta Karnaugh untuk 2 variabel :**

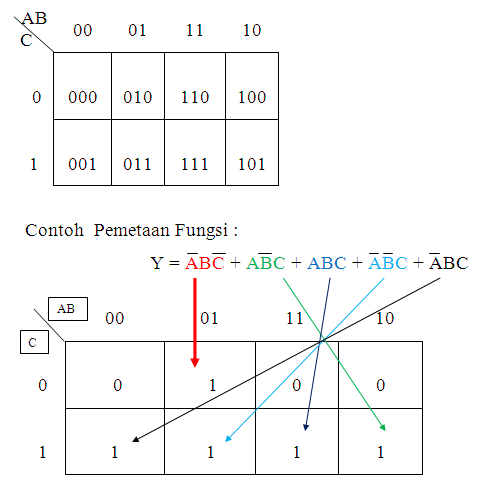
Jumlah kombinasi nilai dari 2 variabel = 22 =4, maka peta karnaugh untuk 2 variabel memiliki 4 buah kotak disusun sebagai berikut :



Setiap suku product dipetakan sebagai angka 1 pada kotak yang sesuai dengan kombinasi variabel suku yang bersangkutan.

1. **Peta Karnaugh untuk 3 Variabel:**

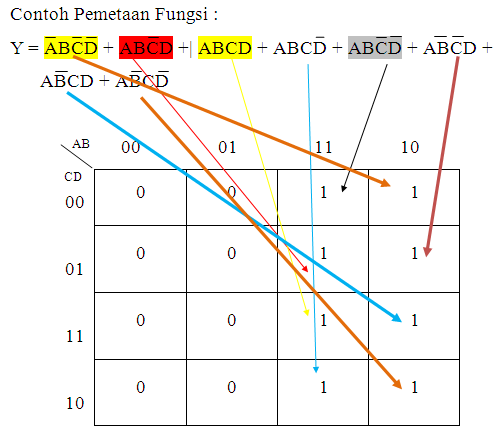
Jumlah kombinasi nilai untuk 3 variabel = 23 = 8, maka memiliki 8 kotak yang disusun sebagai berikut :



1. **Peta Karnaugh unruk 4 Variabel :**

Jumlah kombinasi nilai untuk 4 variabel = 24 = 16, maka memiliki 16 kotak. Dan disusun sebagai berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB | 00 | 01 | 11 | 10 |
| CD  00 | 0000 | 0100 | 1100 | 1000 |
| 01 | 0001 | 0101 | 1101 | 1001 |
| 11 | 0011 | 0111 | 1111 | 1011 |
| 10 | 0010 | 0110 | 1110 | 1010 |

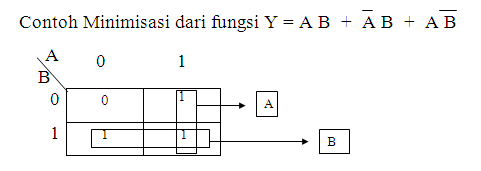


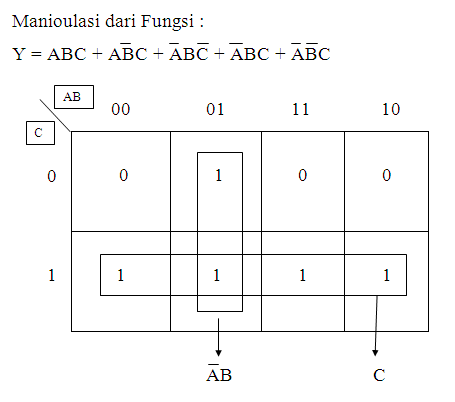
**4. Proses Minimisasi**

Langka-langkah proses minimasi dengan peta karnaugh adalah sebagai berikut :

* Angka-angka 1 saling bertetangga dikelompokan dalam sebuah loop dengan jumlah 2n dimana n = bilangan bulat positip.
* Loop yang berisi 21 =2 angka 1akan mengeleminir 1 buah variabel.
* Loop yang berisi 22 =4 angka 1akan mengeleminir 2 buah variabel.
* Loop yang berisi 23 =8 angka 1akan mengeleminir 3 buah variabel.

Secara Umum adalah : Loop yang berisi 2n angka 1, akan mengeliminir n buah variabel. Variabel yang tereliminir / hilang adalah variabel yang muncul bersamaan dengan variabel komplemennya, atau variabel yang kedua nilainya ( 0 dan 1 ) muncul semuanya, sesuai dengan teorema komplementasi aljabar X + X = 1 ( variabel X nya sudah tidak ada lagi ).

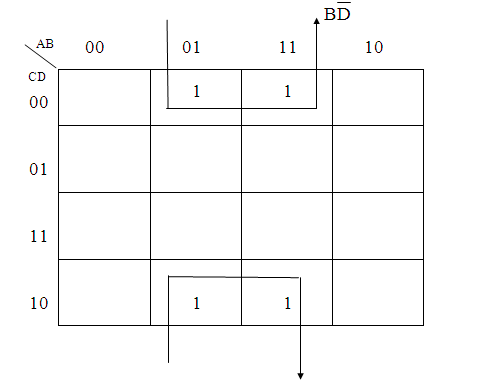




**Catatan :**

Dalam rangka menentukan angka 1 yang saling bertetanggaandidalam peta karnaugh perlu diperhatikan hal-hal sbb :

Peta karnaugh dapat digulung kearah vertical, sehingga sisi bawah saling bertemu dengan sisi atas. Dengan demikian angka-angka 1 yang ada dalam kotak-kotak sisi bawah menjadi saling bertetangga dengan angka-angka 1 pada kotak-kotak sisi paling atas, sehingga dapat dikelompokan.



**Keterangan :**

Dua angka 1 pada garis paling bawah dapat digabung dalam sebuah loop dengan dua angka 1 pada garis paling atas.

Gabungan 4 angka 1 ini akan mengeliminir 2 variabel, yaitu variable A dan C yang kedua nilainya. Hasil gabungan adalah BD.