

RANGKAIAN KOMBINASI dan ALJABAR BOOLEAN



Mata Kuliah : Logika Digital dan Sistem Digital

Dosen Pengampu : Agus Pramono

Disusun Oleh

Nama : Muhammad Ali 18.11.0244

Reska Arosandi 18.11.0087

Rizqi Susetyo 18.11.0213

Kelas : TI 18 E

**STMIK AMIKOM PURWOKERTO
PURWOKERTO**

2018

RINGKASAN

Rangkaian Kombinasional adalah rangkaian yang outputnya hanya tergantung pada input "pada saat itu". Pada prinsipnya, rangkaian kombinasional merupakan penerapan dan penerjemah langsung dari aljabar boole, yang biasanya dinyatakan sebagai fungsi logika. Contoh dari rangkaian kombinasional : *Adder, Decoder, Encoder, Multiplexer dan Demultiplexer*.

Aljabar Boole adalah salah satu bentuk aljabar yang digunakan untuk menganalisis rangkaian digital dengan dasar logika variabel 1 dan 0. Aljabar boole dapat digunakan sebagai salah satu cara untuk menganalisis rangkaian dan menyatakan operasinya secara matematik, terutama untuk mendapatkan konfigurasi rangkaian yang paling sederhana (paling sedikit jumlah komponennya).

BAB 1

PENDAHULUAN

1. Latar Belakang

Perkembangan elektronika berkembang dengan cepatnya sejak ditemukan semikonduktor. Perkembangan yang sangat pesat akhir-akhir ini memberikan tantangan yang hebat bagi ahli-ahli elektronika. Sebelumnya, perangkat radio, tape dan televisi dibuat menggunakan tabung vakum dan transistor sebagai otak dan jantungnya. Hingga pada tahun 1960an IC dikembangkan oleh Jack S Kilby di Lab Texas Instrument. Kekompakan IC yang bisa mengintegrasikan beberapa transistor menjadi satu, memberi peluang peningkatan yang signifikan terutama di bidang elektronika digital.

Elektronika yang merupakan dasar dari teknologi digital saat ini berkembang sejak sistem digital diperkenalkan. Sistem elektronika sebelumnya, yaitu analog merupakan sistem yang kontinyu, baik masukan maupun keluarannya merupakan bilangan yang linier, baik berupa tegangan maupun arus. Memang tidak sepenuhnya, linier namun tetap kontinyu. Berbeda dengan sistem digital. Dalam suatu sistem digital hanya dikenal kondisi yang bersifat diskret (non kontinyu). Dengan demikian dapat dikatakan bahwa analog dan digital dibedakan berdasarkan bagaimana besaran tersebut ditampilkan. Misal saja dalam speedometer, yang berupa digital menampilkan berupa angka-angka sedangkan analog dengan jarum. Hal ini tentu saja berimbas pada pentingnya untuk meningkatkan pengetahuan di bidang elektronika, terlebih bidang digital.

Perlunya peningkatan pengetahuan di bidang digital adalah meningkatkan teknologi yang telah ada berkaitan dengan digital. Jika diperhatikan secara seksama, kecenderungan piranti elektronika sekarang ini menuju pada otomatisasi (komputersisasi), minimalisasi (kecil,kompak), dan digitalisasi.

Dengan otomatisasi segala pekerjaan dapat diselesaikan dengan mudah dan akurat. Dengan minimalisasi, bentuk fisik peranti elektronik menjadi semakin kecil dan kompak, tidak banyak menempati ruang tetapi kinerjanya sangat handal. Sedangkan digitalisasi memungkinkan pengolahan data (sinyal, Informasi) menjadi semakin menguntungkan. (Sumarna 2006: 3) “Kecenderungan pengolahan informasi dalam bentuk digital memiliki beberapa kelebihan, diantaranya adalah:

1. Lebih tegas (tidak mendua), karena sinyal hanya ditampilkan dalam salah satu bentuk, YA atau TIDAK, HIDUP atau MATI.

2. Informasi digital lebih mudah dikelola (mudah disimpan dalam memori, mudah ditransmisikan, mudah dimunculkan kembali dan mudah diolah tanpa menurunkan kualitas)
3. Lebih tahan terhadap gangguan dalam arti lebih sedikit gangguan. Jika kena gangguan lebih mudah dikembalikan
4. Konsumsi daya lebih rendah”

Pentingnya sistem digital di atas, maka sangat diperlukan pengetahuan mengenai sistem digital. Salah satu aspek yang menjadi point penting adalah minimalisasi rangkaian. Ada beberapa cara dasar dalam minimalisasi rangkaian, salah satunya dengan menggunakan Rangkaian Kombinasional dan Aljabar Boolean.

2. Rumusan Masalah

1. Apakah Rangkaian Kombinasional?
2. Kenapa disebut Rangkaian Kombinasional?
3. Apa yang membedakan Rangkaian Kombinasional dengan rangkaian lain?
4. Apa saja yang termasuk rangkaian Kombinasional?
5. Apakah Aljabar Boole?
6. Bagaimana teorema Aljabar Boole?
7. Bagaimana minimalisasi suatu rangkaian dengan Aljabar Boole?
8. Bagaimana hubungan minimalisasi Aljabar Boole dengan teorema minimalisasi lain?

3. Tujuan

1. Mengetahui pengertian rangkaian Kombinasional
2. Mengetahui jenis rangkaian Kombinasional
3. Mengetahui pengertian mengenai Aljabar Boole.
4. Mengetahui teorema aljabar Boole.
5. Mengetahui cara minimalisasi suatu rangkaian menggunakan Aljabar Boolean.
6. Mengetahui hubungan minimalisasi Aljabar Boole dengan teorema minimalisasi lain.

BAB 2

PEMBAHASAN

1. Rangkaian Kombinasional

Menurut Albert Paul Malvino, Ph.D. “Semua rangkaian logika dapat digolongkan atas dua jenis, yaitu rangkaian kombinasi (*combinational circuit*) dan rangkaian berurut (*sequential circuit*). Perbedaan kedua jenis rangkaian ini terletak pada sifat keluarannya. Keluaran suatu rangkaian kombinasi setiap saat hanya ditentukan oleh masukan yang diberikan saat itu. Keluaran rangkaian berurut pada setiap saat, selain ditentukan oleh masukannya saat itu, juga ditentukan oleh keadaan keluaran saat sebelumnya, jadi juga oleh masukan sebelumnya. Jadi, rangkaian berurut tetap mengingat keluaran sebelumnya dan dikatakan bahwa rangkaian ini mempunyai ingatan (*memory*). Kemampuan mengingat pada rangkaian berurut ini diperoleh dengan memberikan tundaan waktu pada lintasan balik (umpan balik) dari keluaran ke masukan. Secara

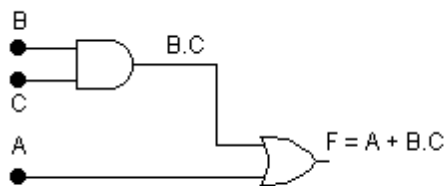
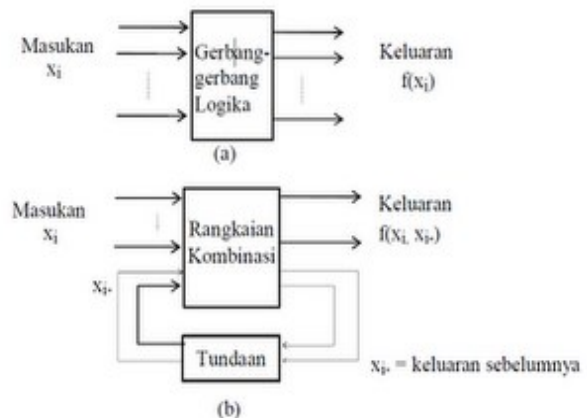


diagram blok, kedua jenis rangkaian logika ini dapat digambarkan seperti pada Gambar 1.”

Gambar 1: Rangkain Gerbang Logika.



Gambar 2. Model Umum Rangkaian Logika

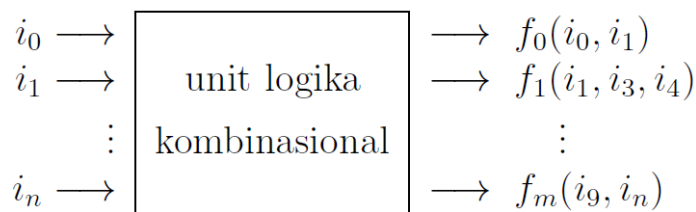
(a) Rangkaian Kombinasional

(b) Rangkaian Berurut

1.1. Unit Logika Kombinasional

Unit logika kombinasional (ULK) adalah unit yang menerjemahkan sederetan masukan menjadi sederetan keluaran menggunakan fungsi-fungsi tertentu. Keluaran yang dihasilkan hanya merupakan fungsi dari masukan, dan begitu nilai masukan berubah maka nilai keluaran akan menyesuaikan. Bentuk umum dari unit logika kombinasional tercantum pada Gambar 3 Sederetan masukan $i_0 - i_n$ diumpungkan ke ULK, yang menghasilkan sederetan keluaran sesuai dengan fungsi $f_0 - f_m$. Tidak ada umpan balik dari keluaran ke masukan dalam rangkaian logika kombinasional.

Masukan dan keluaran untuk ULK secara normal mempunyai 2 nilai yaitu: tinggi dan rendah. Jika sinyal (nilai) berupa nilai yang dimabil dari anggota himpunan berhingga, rangkaianannya disebut digital. Rangkaian elektronika digital menerima masukan dan keluaran dalam nilai 0 atau 1. Nilai 0 yang berarti 0 volt disebut sebagai nilai rendah dan nilai 1 yang biasanya mengacu pada 5 volt disebut nilai tinggi. Kesepakatan ini tidak berlaku di semua keadaan.



Gambar 3 : Unit logika kombinasi, jika dilihat dari luar

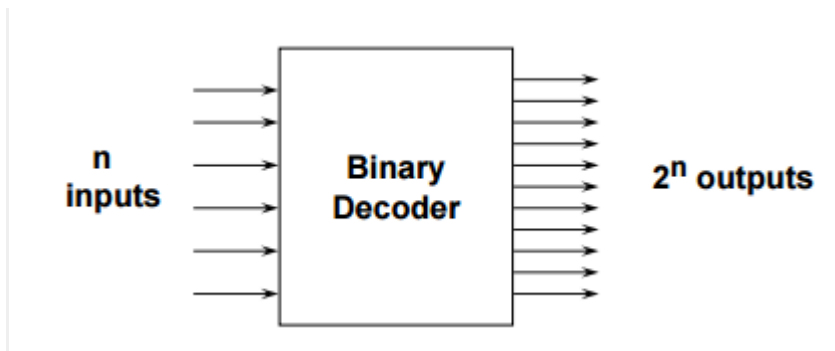
Walaupun sebagian besar komputer digital adalah komputer biner, namun rangkaian yang menggunakan multi-nilai juga ada. Jalur yang mengirimkan data dengan multi-nilai menjadi lebih efisien daripada menggunakan 2 nilai saja. Rangkaian digital multi-nilai berbeda dengan rangkaian analog karena rangkaian digital multi-nilai mempunyai variasi nilai terhingga sedangkan sinyal analog mempunyai nilai kontinu. Secara teori penggunaan rangkaian digital multi-nilai adalah menguntungkan. Namun dalam praktiknya sulit untuk membuat rangkaian multi-nilai yang handal dalam membedakan nilai lebih dari 2 macam. Oleh karena itu, logika multi-nilai saat ini digunakan secara terbatas, Contoh dari rangkaian kombinasional : *Adder, Decoder, Encoder, Multiplexer dan Demultiplexer*.

A. Decoder

Decoder adalah rangkaian digital yang dapat mengubah bilangan biner menjadi bilangan desimal, dimana rangkaian ini akan menghasilkan output high (1) pada jalur yang sesuai dengan yang ditunjuk oleh selector. Artinya input decoder merupakan bilangan biner, dan outputnya pun berbentuk biner. Namun akan menunjukkan bilangan decimal, yaitu menentukan output manakah yang aktif (Y_0 , Y_1 , Y_2 , atau Y_3).

A.1. Binary Decoding

Mengkonversi sebuah n -bit code ke dalam sebuah 1 (satu) output yang aktif (low/high). Rangkaianya dapat dibentuk menggunakan AND atau OR gate. Serta jumlah masukan (input) < Jumlah Keluaran (Output)

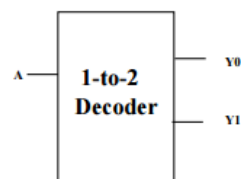


Input berupa (n) maka akan menghasilkan output (2^n). Yang nantinya hanya akan ada satu output yang aktif (low/high) dari banyak input yang diberikan.

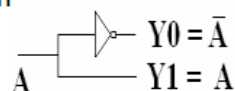
1. 1-to-2 Binary Decoder

Tabel 1 to 2

A	Y_0	Y_1
0	1	0
1	0	1



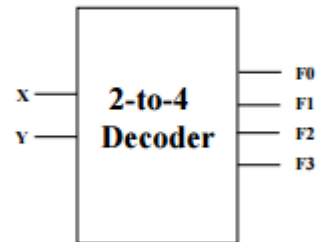
Rangkaian



2. 2-to-4 Binary Decoder

Tabel Kebenaran:

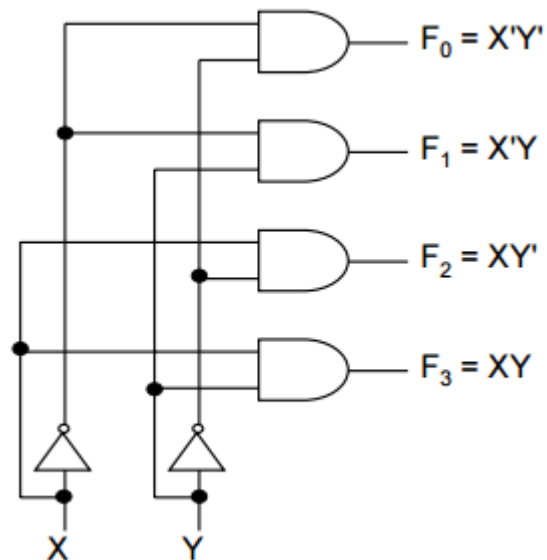
X	Y	F ₀	F ₁	F ₂	F ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Dari tabel kebenaran 2 to 4 diperoleh persamaan :

2-variable minterm ($X'Y'$, $X'Y$, XY' , XY)

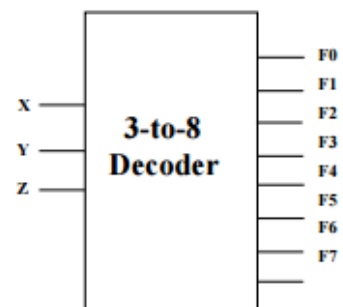
Rangkaian 2-to-4 Binary Decoder



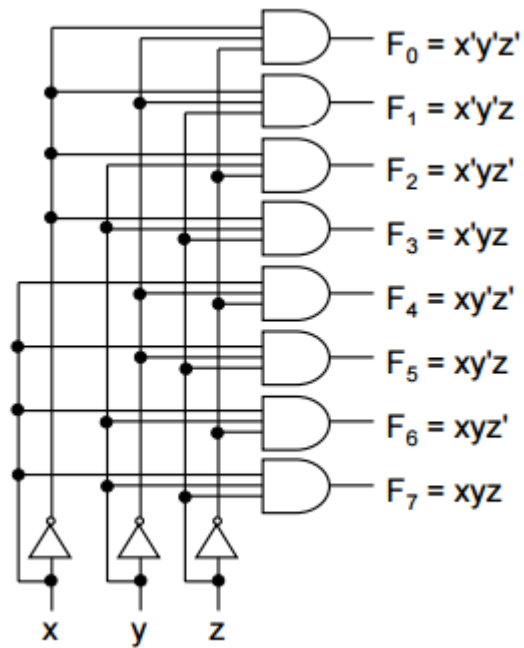
3. 3-to-8 Binary Decoder

Tabel Kebenaran

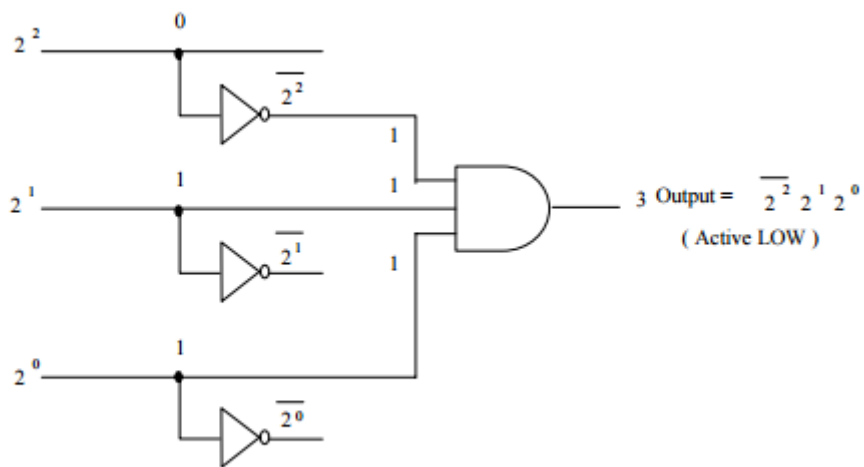
x	y	z	F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Rangkaian 3-to-8 Binary Decoder



Rangkaian penghasil output '3' (active HIGH) untuk input 0 1 1



A.2. Binary to Octal Decoding

**Tabel kebenaran Active High output
3-Bit Biner to Octal Decoder**

Input			Output							
2^2	2^1	2^0	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Aktif High menggunakan AND

**Tabel kebenaran Active Low output
3-Bit Biner to Octal Decoder**

Input			Output							
2^2	2^1	2^0	0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

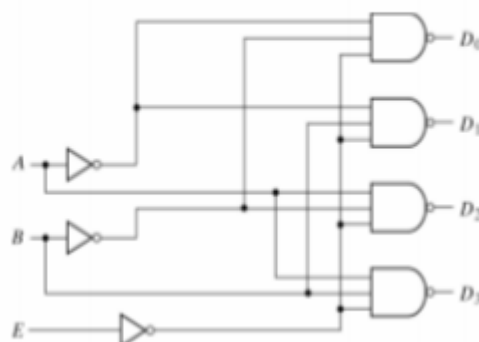
Aktif Low menggunakan NAND

1. 2 to 4 Decoder dengan Enable Input

Tabel Kebenaran :

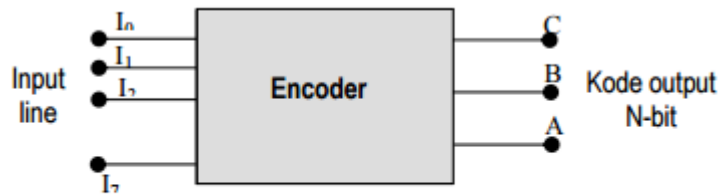
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Diagram Logic:



B. Encoder

Sebuah rangkaian Encoder menterjemahkan keaktifan salah satu inputnya menjadi urutan bit-bit biner. Encoder terdiri dari beberapa input line, hanya salah satu dari input-input tersebut diaktifkan pada waktu tertentu, yang selanjutnya akan menghasilkan kode output Nbit. Gambar 12-1 menunjukkan blok diagram dari sebuah encoder.

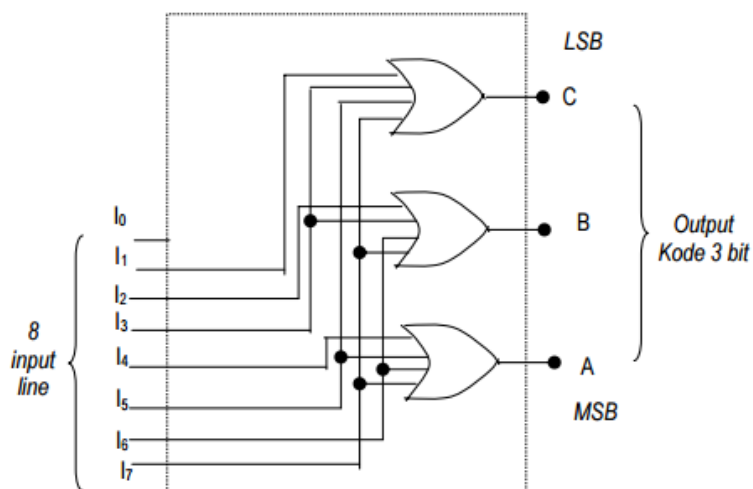


Hanya salah satu bernilai HIGH pada waktu tertentu

Tabel Kebenaran Encoder

INPUT								OUTPUT		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Berdasarkan output dari tabel kebenaran di atas maka rangkaian encodernya adalah sebagai berikut



B.1. Priority Encoder

Sebuah Priority Encoder adalah rangkaian Encoder yang mempunyai fungsi prioritas. Operasi dari rangkaian Priority Encoder adalah sebagai berikut : Jika ada dua atau lebih input bernilai “1” pada saat yang sama, maka input yang mempunyai prioritas tertinggi yang akan diambil. Tabel Kebenaran Priority Encoder diberikan pada Tabel 12-2. Kondisi ‘x’ adalah kondisi don’t care, yang menyatakan nilai input bisa “1” atau “0”. Input D3 mempunyai prioritas tertinggi, sehingga bila input ini bernilai “1” maka output X dan Y keduanya akan bernilai “1” (11 menyatakan biner dari 3). Input D2 mempunyai prioritas kedua, dengan output X dan Y bernilai 10 menyatakan biner 2, dimana input D2 = “1” dan D3=“0”. Input D1 adalah prioritas ketiga dengan output X dan Y bernilai 01 menyatakan biner 1, dimana input D1 =“1”, sedangkan D2= D3=“0”. Prioritas terendah adalah input D0, yang akan memberikan output X dan Y = 00 (menyatakan biner 0), jika input D1 bernilai “1”, sedang ketiga input lainnya bernilai “0”.

Tabel kebenaran Priority Encoder

INPUT				OUTPUT		
D ₃	D ₂	D ₁	D ₀	Q ₁	Q ₀	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Dari Tabel Kebenaran di atas, kemudian dibuat K-Map seperti gambar 12-3 untuk masing-masing output X, Y dan V (V adalah nilai output Validitas, yang akan bernilai “1” jika satu atau lebih inputnya bernilai “1”, dan bernilai “0” jika tidak ada inputnya yang bernilai “1”).

D ₃ D ₂	D ₁ D ₀			
	00	01	11	10
00	x	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$Q_1 = D_3 + D_2$$

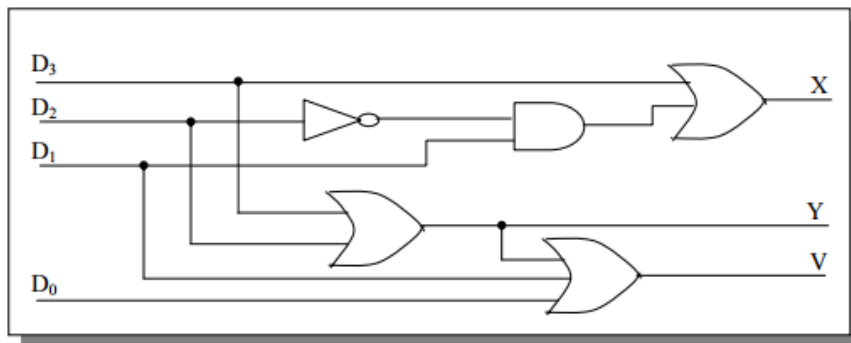
D ₃ D ₂	D ₁ D ₀			
	00	01	11	10
00	x	0	1	1
01				
11	1	1	1	1
10	1	1	1	1

$$Q_0 = D_3 + \overline{D_2}D_1$$

		D ₁ D ₀			
		00	01	11	10
D ₃ D ₂	00	0	1	1	1
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

$$V = D_0 + D_1 + D_2 + D_3$$

Rangkaian Priority Encoder



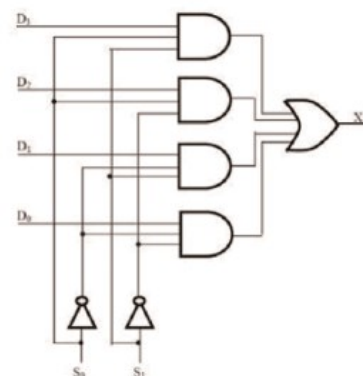
C. Multiplexer

Multiplexer adalah suatu rangkaian yang mempunyai banyak input dan hanya mempunyai satu output. Dengan menggunakan selektor, dapat dipilih salah satu inputnya untuk dijadikan output. Sehingga dapat dikatakan bahwa multiplexer ini mempunyai n-input, m-selector, dan 1 output. Biasanya jumlah inputnya adalah 2^m selektornya. Adapun macam dari multiplexer ini adalah sebagai berikut:

- Multiplexer 4x1 atau 4 to 1 multiplexer
- Multiplexer 8x1 atau 8 to 1 multiplexer
- Multiplexer 16x1 atau 16 to 1 multiplexer dsb

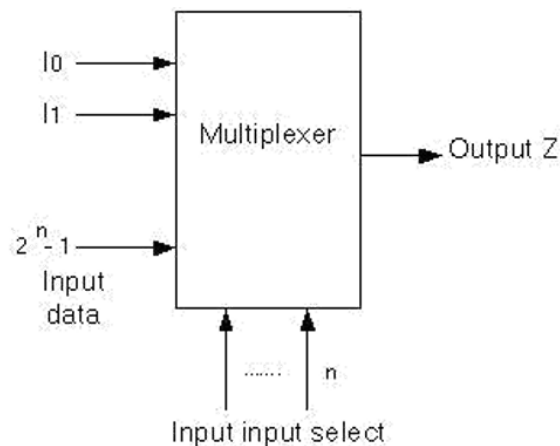
Tabel 2.1 Tabel Kebenaran Multiplexer

INPUT						OUTPUT	
S ₀	S ₁	D ₀	D ₁	D ₂	D ₃	X	Ket
0	0	0	x	x	x	0	D ₀
0	0	1	x	x	x	1	
0	1	x	0	x	x	0	D ₁
0	1	x	1	x	x	1	
1	0	x	x	0	x	0	D ₂
1	0	x	x	1	x	1	
1	1	x	x	x	0	0	D ₃
1	1	x	x	x	1	1	



Gambar 2.1 Rangkaian Multiplexer

Multiplexer atau selektor data adalah suatu rangkaian logika yang menerima input data dan untuk suatu saat tertentu hanya mengijinkan satu dari data input tersebut untuk lewat mencapai output. Jalan yang akan ditempuh dari input data yang diinginkan ke output dikontrol oleh input – input SELECT (kadang – kadang disebut input ADDRESS). Di bawah ini merupakan gambar diagram dasar multiplexer.

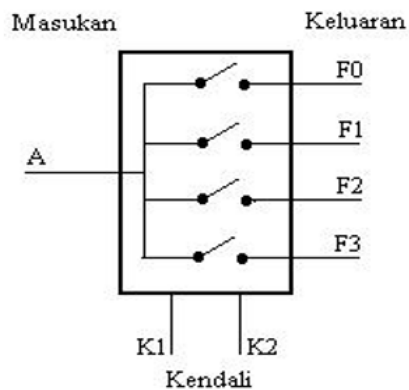


Gambar 2.2 Diagram Dasar Multiplexer

Multiplexer bekerja seperti sebuah saklar (switch) multi posisi yang dikontrol secara digital, dimana kode digital yang diberikan ke input - input SELECT mengontrol input - input data mana yang di switch ke output. misalnya, pada multiplexer dua input, output z akan sama dengan input data I_0 untuk kode input SELECT berlogik 1, Z akan sama dengan I_1 untuk kode input SELECT berlogik 0. Dengan kata lain multiplexer memilih 1 dari N data input dan menyalurkan data yang terpilih ke suatu channel output tunggal.

D. Demultiplexer

Sebuah *Demultiplexer* adalah rangkaian logika yang menerima satu input data dan mendistribusikan input tersebut ke beberapa output yang tersedia. Kendali pada *demultiplexer* akan memilih saklar mana yang akan dihubungkan. Pemilihan keluarannya dilakukan melalui masukan penyeleksi. Seleksi data-data input dilakukan oleh *selector line*, yang juga merupakan input dari *demultiplexer* tersebut. Pada *demultiplexer* saluran kendali sebanyak "n" saluran dapat menyeleksi saluran keluaran. Secara bagan, kerja *demultiplexer* dapat digambarkan sebagai berikut :

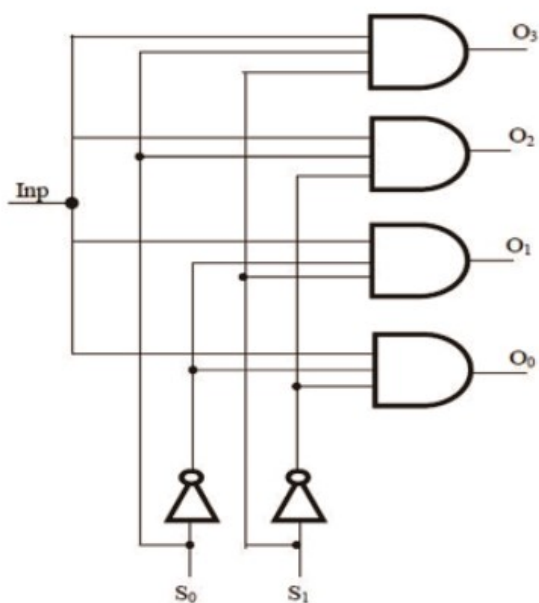


Gambar 2.3 Rangkaian dasar demultiplexer

Pada *demultiplexer*, masukan data dapat terdiri dari beberapa bit. Keluarannya terdiri dari beberapa jalur, masing-masing jalur terdiri dari satu atau lebih dari satu bit. Masukan selector terdiri dari satu atau lebih dari satu bit tergantung pada banyaknya jalur keluaran.

Sedangkan tabel kebenaran sebuah *demultiplexer* dengan 2 *select line* dapat ditunjukkan pada Tabel berikut.

Input			Output				
K1	K2	Inp	F0	F1	F2	F3	
0	0	0	0	x	x	x	
0	0	1	1	x	x	x	
0	1	0	x	0	x	x	
0	1	1	x	1	x	x	
1	0	0	x	x	0	x	
1	0	1	x	x	1	x	
1	1	0	x	x	x	0	
1	1	1	x	x	x	1	



Tabel 2.2 Tabel kebenaran demultiplexer dengan 2 select line dan Rangkaian Logika Demultiplexer

2. ALJABAR BOOLEAN

A. Pengertian aljabar boole

Aljabar merupakan nama salah satu ilmuwan Islam dalam bidang matematika. Banyak teori yang dikemukakan olehnya. Banyak jenis aljabar yang dikenal dalam bidang matematika, seperti aljabar biasa, aljabar himpunan, aljabar vektor, aljabar group, aljabar boole dan lain-lain. Aljabar bisa disebut juga ilmu hitung. Dalam setiap aljabar memiliki postulat, teorema, dan operasi sendiri-sendiri. Aljabar Boole beda dengan aljabar yang lain. Aljabar boole ditemukan oleh George Boole di abad 19 sebagai suatu sistem untuk menganalisis secara matematis mengenai logika. Aljabar boole didasarkan pada pernyataan logika bernilai benar atau salah. Ternyata, aljabar boole ini menjadi alat yang sangat ampuh untuk merancang maupun menganalisis rangkaian digital.

Sistem Aljabar Boole yaitu suatu sistem aljabar yang hanya memiliki dua macam konstanta, yaitu '0' dan '1'. Dimana dua konstanta ini (0 dan 1) digunakan untuk menggambarkan (mewakili) keadaan (state) suatu terminal. Keadaan (state ini) pada umumnya dianalogikan dengan level tegangan. Dalam Aljabar Boolean, besaran yang dapat berubah/ variable (dapat bernilai '0' atau '1') dituliskan dengan simbol huruf misal A, B, C dan sebagainya. Variable aljabar boole sering digunakan untuk menyajikan suatu tingkat tegangan pada terminal suatu rangkaian. Terminal itu dapat berupa kawat atau saluran masukan/ keluaran suatu rangkaian. Misalnya 0 sering digunakan untuk menandai suatu jangkauan tegangan dari 0 volt sampai dengan 0.8 volt. Sedangkan 1 sering digunakan untuk jangkauan tegangan dari 2 volt hingga 5 volt. Dengan demikian, tanda 0 dan 1 tidak menggambarkan bilangan yang sebenarnya tetapi menyatakan keadaan suatu variable tegangan. Setiap keluaran dari suatu atau kombinasi beberapa buah gerbang dapat digunakan dalam suatu rangkaian logika yang disebut ungkapan Boole seperti gerbang AND, OR, dan NOT.

Aljabar boole digunakan untuk menyatakan pengaruh berbagai rangkaian digital pada masukan-masukan logika, dan untuk memanipulasi variable logika dalam menentukan cara terbaik pada pelaksanaan (kinerja) fungsi rangkaian tertentu. Oleh karena hanya ada dua nilai yang mungkin, aljabar boole lebih cocok digunakan untuk rangkaian digital dibandingkan dengan aljabar yang lain. Dalam aljabar boole tidak ada pecahan, decimal, bilangan negative, akar kuadrat, akar pangkat tiga, logaritma, bilangan imajiner, dan sebagainya.

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

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

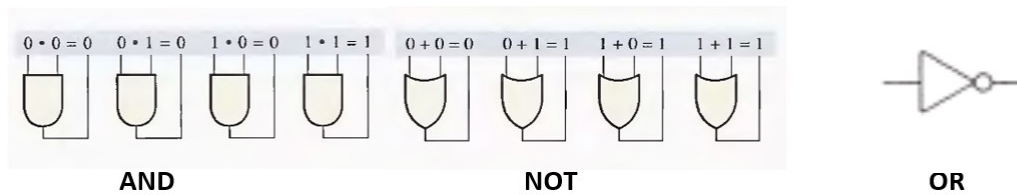
Dalam rangkaian kombinasi, keluaran rangkaian pada setiap saat hanya ditentukan oleh masukannya pada saat itu. Pada rangkaian berurut, selain ditentukan oleh masukan saat itu, keluaran juga ditentukan oleh keadaan keluaran sebelumnya. Jadi, rangkaian berurut mempunyai kemampuan untuk mengingat keadaan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

Teori-teori aljabar boolean ini merupakan aturan-aturan dasar hubungan antara variabel variabel boolean. Aturan ini digunakan untuk memanipulasi dan menyederhanakan suatu rangkaian logika ke dalam bentuk yang bervariasi. Variasi yang dilakukan mengenal tiga operasi dasar yaitu:

- 1) penjumlahan logika atau OR dengan simbol operasi “+”
- 2) perkalian logika atau AND dengan simbol operasi “.”
- 3) Komplementasi atau NOT atau inversi dengan simbol operasi “ $\bar{}$ ” dengan garis di atas variable

Aturan operasi OR, AND dan NOT pada dua tingkat logika 0 dan 1 dapat dirangkum sebagai berikut:

OR	AND	NOT
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\bar{1} = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\bar{0} = 1$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	



Aljabar boole dapat digunakan sebagai salah satu cara untuk menganalisis rangkaian dan menyatakan operasinya secara matematik, terutama untuk mendapatkan konfigurasi rangkaian yang paling sederhana (paling sedikit jumlah komponennya).

B. Teorema aljabar boole

Setiap peubah Boole hanya dapat berkeadaan satu dari dua keadaan 0 atau 1. Jadi, kalau satu peubah di-OR-kan dengan 0 maka hasilnya tidak akan 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 satu, 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**:

$$\begin{aligned} X + 0 &= X & X \cdot 0 &= 0 \\ X + 1 &= 1 & X \cdot 1 &= X \end{aligned}$$

Jika suatu peubah di-OR-kan dengan dirinya sendiri, maka hasilnya akan 0 bila keadaan variable itu adalah 0 dan hasilnya akan 1 bila keadaan variable itu adalah 1. Jadi, peng-OR-an satu variable dengan dirinya sendiri menghasilkan keadaan sama dengan keadaan variable itu. Keadaan serupa berlaku untuk operasi AND. Ini disebut **hukum idempotent**:

$$X + X = X \quad X \cdot X = X$$

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:

$$\overline{\overline{X}} = X$$

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:

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

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

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

1. $A + B = B + A$ (komutatif OR)
2. $A \cdot B = B \cdot A$ (komutatif AND)
3. $A + (B + C) = (A + B) + C = A + B + C$ (asosiatif OR)
4. $A(BC) = (AB)C = ABC$ (asosiatif AND)
5. $A(B + C) = AB + AC$ (distributive OR)
6. $(A + B)(C + D) = AC + BC + AD + BD$ (distributive AND)
7. $A + AB = A$
8. $A + \overline{A}B = A + B$

Adapun matematikawan De Morgan menyumbangkan teoremanya dalam aljabar boole, atau yang lebih dikenal dengan **teorema de Morgan**, yaitu:

$$\overline{XY} = \overline{X} + \overline{Y}$$

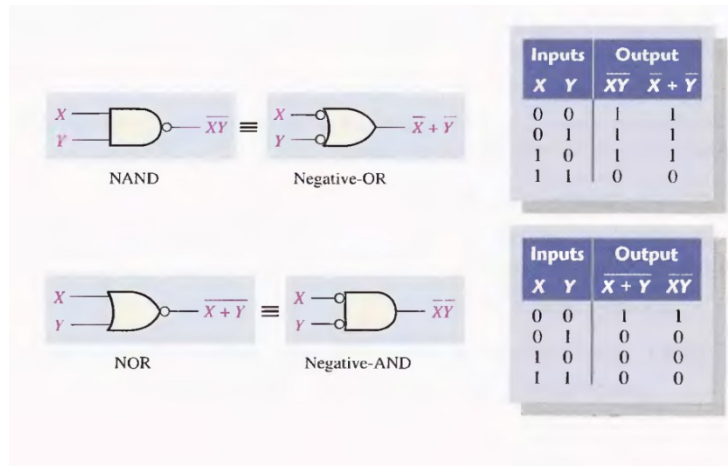
Nilai komplemen dari dua variable yang di-AND-kan akan sama nilainya dengan jumlah dari dua variable yang di komplemenkan masing-masing.

$$\overline{\overline{X} + \overline{Y}} = \overline{\overline{X}} \cdot \overline{\overline{Y}}$$

Nilai komplemen dari dua variable yang di-OR-kan akan sama nilainya dengan perkalian dari dua variable yang di komplemenkan masing-masing.

Teorema de Morgan itu tidak hanya berlaku untuk dua variable, selain A dan B masing-masing terdiri dari satu variable tetapi operasi OR atau AND dapat diteruskan pada variable berikutnya.

Berikut table kebenaran dari teorema de Morgan:



C. Minimalisasi rangkaian

Fungsi tertentu dari sebuah rangkaian logika pada umumnya tidak unik, maksudnya tidak berpengaruh konfigurasinya namun menghasilkan fungsi yang sama. 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. Tentu saja dalam perancangannya dicari sebuah konfigurasi yang minimal, sehingga tidak rumit, akan meminimalkan kemungkinan terjadinya kesalahan, mengurangi efek pembebanan dan sebagainya. 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. Banyak yang mencari model penyederhanaan dalam sebuah rangkaian digital. Penyederhanaan fungsi Boole dapat dilakukan dengan beberapa cara/ metoda, antara lain:

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

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 penjelasan sebelumnya dapat diperoleh:

- $XY + X\bar{Y} = X(Y + \bar{Y}) = X.1 = X$
- $X + X\bar{Y} = X(1 + \bar{Y}) = X.1 = X$
- $(X+Y)(X+\bar{Y}) = X.X + X(Y+\bar{Y}) + Y.\bar{Y}$
 $= X + X.1 + 0$
 $= X + X$
 $= X$
- $X(X+Y) = X + XY$
 $= X.1 + XY$
 $= X(1+Y)$
 $= X$
- $(X+\bar{Y})Y = XY + Y\bar{Y}$
 $= XY$
- $XY + \bar{Y} = (X+Y)(\bar{Y} + \bar{Y})$ (hukum distributif)
 $= X + Y$
- Teorema Konsensus: $xy + yz + \bar{x}z = xy + \bar{x}z$

$$(x+y)(y+z)(x+z) = (x+y)(\bar{x}+z)$$

$$(x+y)(\bar{x}+z) = xz + xy$$

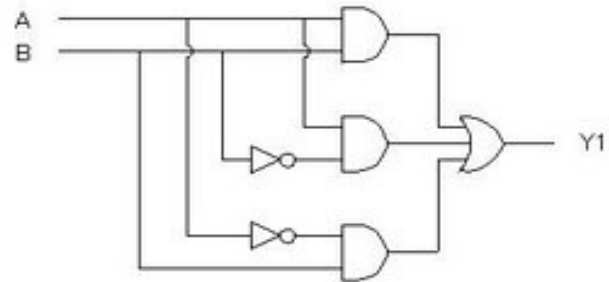
Banyak yang mencari metode terbaik untuk keperluan penyederhanaan itu. Diantara metode-metode yang telah disebutkan diatas ada salah satu metode penyederhanaan rangkaian logika yang akan dipelajari lebih jauh yaitu metode analitis. Metode analitis ini banyak menggunakan teorema-teorema aljabar boole.

Sebagai ilustrasi marilah mencari bentuk yang paling sederhana atau setidaknya lebih sederhana dari pernyataan atau fungsi logika beberapa contoh di bawah ini.

***) Minimalisasi Rangkaian Logika :**

$$\begin{aligned}
 Y &= AB + \bar{A}\bar{B} + \bar{A}B \\
 &= A(B + \bar{B}) + \bar{A}B \\
 &= A \cdot 1 + \bar{A}B \\
 &= A + \bar{A}B
 \end{aligned}$$

$$Y = A + B$$



Rangkaian Asli

$$Y1 = AB + \bar{A}\bar{B} + \bar{A}B$$



Rangkaian Yg Telah Di Minimalisasikan

$$Y2 = A + B$$

Jika dibandingkan lebih seksama, rangkaian yang ditunjukkan pada gambar kedua jauh lebih sederhana dari rangkaian pada gambar pertama padahal kedua rangkaian memiliki fungsi yang sama. Pada rangkaian awal yang rumit terdiri dari 6 jumlah gerbang, 3 jenis gerbang, dan 3 tingkat. Sedangkan setelah disederhanakan rangkaian kedua menjadi rangkaian sederhana yang terdiri dari 1 jumlah gerbang, 1 jenis gerbang, dan 1 tingkat. Rangkaian tersebut lebih sederhana dari selumnya.

Contoh lain dari penyederhanaan rangkaian logika yaitu sebagai berikut :

Sederhanakan pernyataan :

$$Y = ABC + A\bar{B}(\bar{A}\bar{C})$$

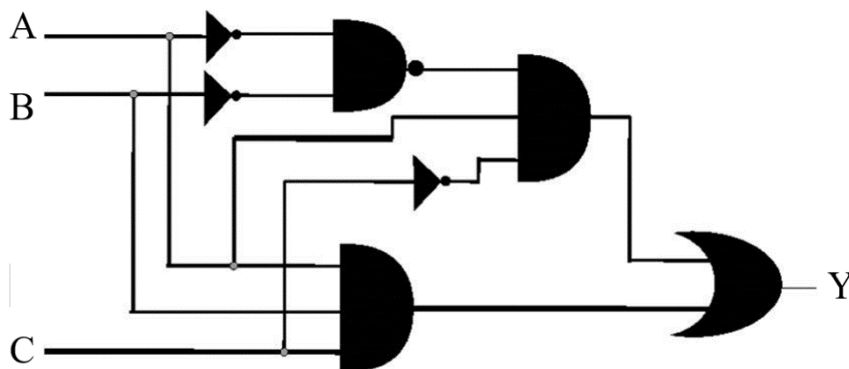
Penyelesaian :

$$\begin{aligned}
 Y &= ABC + A\bar{B}(\bar{A}\bar{C}) \\
 &= ABC + A\bar{B}(\bar{A} + \bar{C}) \\
 &= ABC + A\bar{B}(A + C) \\
 &= ABC + AA\bar{B} + A\bar{B}C \\
 &= ABC + A\bar{B} + A\bar{B}C \\
 &= AC(B + \bar{B}) + A\bar{B} \\
 &= AC(1) + A\bar{B} \\
 &= A(C + \bar{B})
 \end{aligned}$$

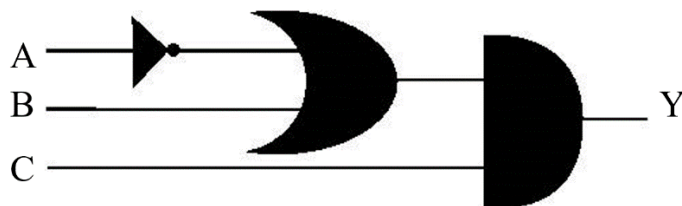
Jadi $Y = ABC + A\bar{B}(\bar{A}\bar{C})$

Memiliki bentuk yang lebih sederhana yaitu $Y = A(C + \bar{B})$

Rangkaian $Y = ABC + A\bar{B}(\bar{A}\bar{C})$ dapat direalisasikan dalam bentuk rangkaian sebagai berikut:



Sedangkan realisasi dari penyederhanaan dari penyelesaian soal diatas dapat di rangkai menjadi



Jika dibandingkan lebih seksama, rangkaian yang ditunjukkan pada gambar kedua jauh lebih sederhana dari rangkaian pada gambar pertama padahal kedua rangkaian memiliki fungsi yang sama. Pada rangkaian awal yang rumit terdiri dari 7 jumlah gerbang, 4 jenis gerbang, dan 4 tingkat. Sedangkan setelah disederhanakan rangkaian kedua menjadi rangkaian sederhana yang terdiri dari 3 jumlah gerbang, 3 jenis gerbang, dan 3 tingkat. Setidaknya rangkaian tersebut lebih sederhana dari selumnya.

Tabel 1. Penyederhanaan Aljabar Boole pada Rangkaian $Y = ABC + AB(\overline{AC})$

Aspek	Sebelum	Sesudah
Jumlah gerbang	7	3
Jenis gerbang	4	3
Tingkat	4	4

Selanjutnya, untuk lebih memahami penggunaan postulat Boole dalam rangka penyederhanaan suatu pernyataan logika, perhatikanlah dengan seksama contoh-contoh berikutnya.

Contoh1 Sederhanakan : $A \cdot (A \cdot B + C)$

$$\begin{aligned}
 \text{Penyelesaian : } A \cdot (A \cdot B + C) &= A \cdot A \cdot B + A \cdot C \\
 &= A \cdot (B + C) \\
 &= A \cdot B + A \cdot C
 \end{aligned}$$

Contoh 2 Sederhanakan : $\bar{A} \cdot B + A \cdot B + \bar{A} \cdot \bar{B}$

$$\begin{aligned}
 \text{Penyelesaian : } \bar{A} \cdot B + A \cdot B + \bar{A} \cdot \bar{B} &= (\bar{A} + A) \cdot B \\
 &= 1 \cdot B + \bar{A} \cdot \bar{B} + \bar{A} \cdot \bar{B} \\
 &= B + \bar{A} \cdot \bar{B} \\
 &= B + \bar{A}
 \end{aligned}$$

Contoh3 Sederhanakan : $A + A \cdot \bar{B} + \bar{A} \cdot B$

$$\begin{aligned}
 \text{Penyelesaian : } A + A \cdot \bar{B} + \bar{A} \cdot B &= (A + A \cdot \bar{B}) + \bar{A} \cdot B \\
 &= A + \bar{A} \cdot B \\
 &= A + B
 \end{aligned}$$

Contoh 4 Sederhanakan : $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C}$

Penyelesaian : $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C} = \bar{A}(\bar{B}\bar{C} + B\bar{C}) + A(\bar{B}\bar{C} + B\bar{C})$

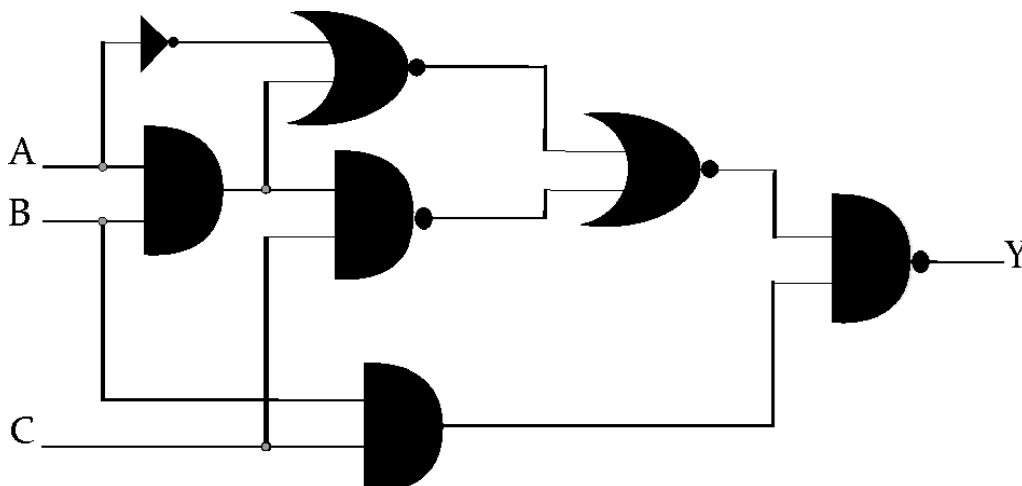
$$= \bar{A}\{\bar{C}(\bar{B} + B)\} + A\{\bar{C}(\bar{B} + B)\}$$
$$= \bar{A}\bar{C} + A\bar{C}$$
$$= \bar{C}$$

Jadi $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C} = \bar{C}$

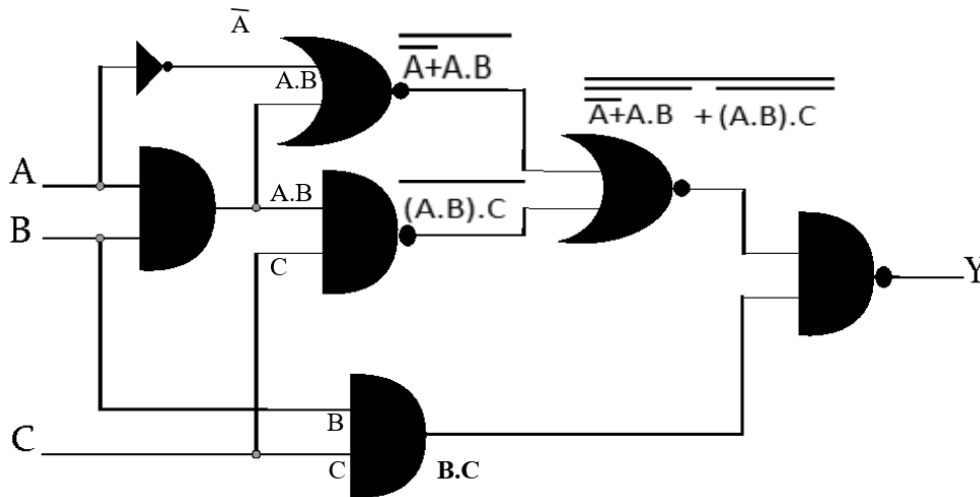
Selanjutnya kita akan mempelajari cara untuk membuat pernyataan logika yang direalisasikan dari sebuah gambar rangkaian logika. Untuk membuat pernyataan boolean dari sebuah rangkaian logika diperlukan langkah-langkah sebagai berikut:

- Mengetahui dari jenis gerbang apa saja yang membentuk rangkaian logika tersebut. Gerbang AND, NOT, atau OR.
- Mengurutkan mulai dari aljabar masukan pada awal rangkaian menuju gerbang pertama yang mengawali rangkaian.
- Tulis hasilnya pada keluaran gerbang tersebut
- Gunakan cara yang sama untuk masukan-masukan aljabar lainnya, (karena masukan biasanya lebih dari 2 masukan)
- Teruslah melanjutkan hasil dari keluaran beberapa gerbang yang menyusun rangkaian itu
- Lalu menggabungkan beberapa hasil keluaran menjadi satu hasil pada keluaran yang terakhir.

Cara penyelesaian diatas akan lebih jelas jika direalisasikan pada soal di bawah ini:
Tulislah persamaan Boolean (persamaan logika) untuk rangkaian digital berikut:



Dengan langkah yang sudah dikemukakan diatas maka dapat dilihat proses menyatakan pernyataan Boolean dari suatu gambar rangkaian, adalah sebagai berikut:



Dari langkah cara penyelesaian diatas, dengan menggabungkan beberapa hasil keluaran menjadi satu hasil pada keluaran yang terakhir sehingga diperoleh hasil akhir.

$$Y = \overline{A} + A.B + (A.B).C + B.C$$

Kemudian bentuk soal selanjutnya yaitu sebaliknya menggambar rangkaian logika dari pernyataan Boolean yang diketahui. Langkah untuk menyelesaikan kasus tersebut antara lain:

1. Menganalisis gerbang apa saja yang harus digunakan dari pernyataan Boolean yang ada.(misalnya saja ada perkalian jadi menggunakan gerbang AND)
2. Mengelompokkan beberapa fungsi boole yang menurut kita mudah
3. Menggambar rangkaian yang dimulai dari pernyataan boole yang terakhir
4. Mengkombinasikan beberapa gerbang yang sudah diketahui menjadi suatu rangkaian

Sampai semua tergambar.

Untuk lebih jelasnya kita mencoba menganalisis kasus berikut ini,

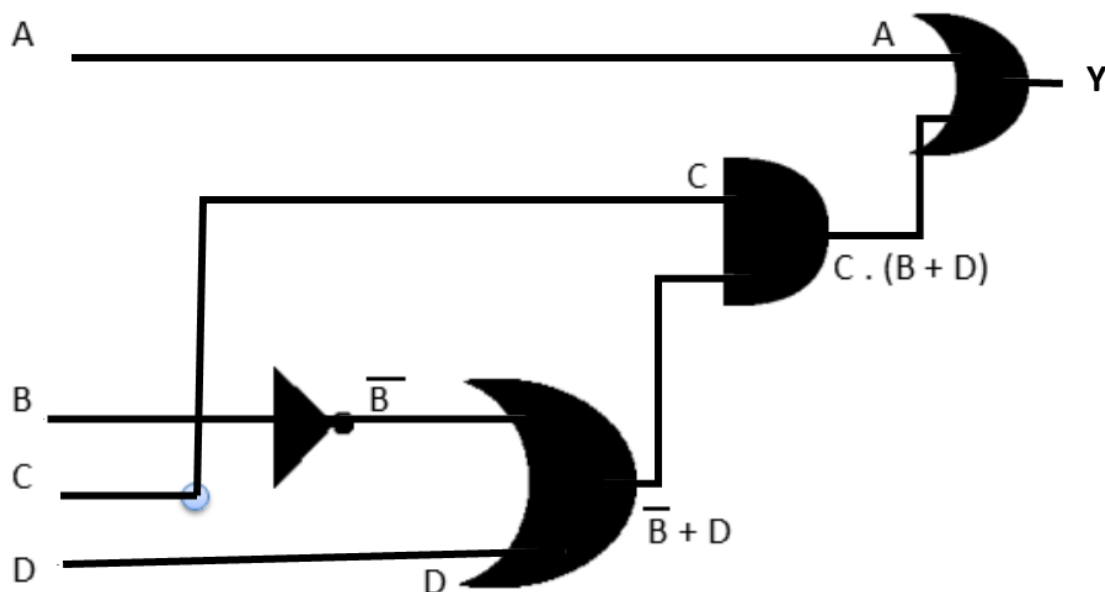
Gambarlah rangkaian digital untuk mengimplementasikan persamaan Boolean berikut:

$$Y = A + C(\bar{B} + D)$$

Penyelesaian: dengan langkah diatas dapat direalisasikan bahwa dari pernyataan boole atau persamaan logika tersebut terdapat fungsi OR, AND, dan NOT. Kemudian agar lebih mudah yaitu menggambar rangkaian yang dimulai dari pernyataan boole yang terakhir langkahnya sebagai berikut:

1. Pertama menggambar gerbang NOT untuk masukan B lebih dulu.
2. kedua dilanjutkan fungsi $(\bar{B}+D)$ digambar dengan gerbang OR
3. Ketiga gambarlah masukan C yang kemudian dikalikan dengan $(\bar{B}+D)$ menggunakan gerbang AND menjadi fungsi $C \cdot (\bar{B}+D)$
4. Terakhir menggambar masukan A yang kemudian ditambah dengan fungsi Menjadi $Y = A + C \cdot (\bar{B}+D)$

Dari langkah di atas maka hasil gambar dari persamaan logika $Y = A + C \cdot (\bar{B}+D)$



BAB III

PENUTUP

KESIMPULAN

1. Rangkaian Kombinasional adalah rangkaian yang outputnya hanya tergantung pada input "pada saat itu". Pada prinsipnya, rangkaian kombinasional merupakan penerapan dan penerjemah langsung dari aljabar boole, yang biasanya dinyatakan sebagai fungsi logika.
2. Encoder adalah rangkaian yang berfungsi untuk mengkodekan data input menjadi data bilangan dengan format tertentu. Encoder dalam rangkaian digital adalah rangkaian kombinasi gerbang digital yang memiliki input banyak dalam bentuk line input dan memiliki output sedikit dalam format bilangan biner
3. Decoder adalah suatu rangkaian logika yang berfungsi untuk mengkonversikan kode yang kurang dikenal manusia ke dalam kode yang lebih dikenal manusia.
4. Multiplexer (MUX) adalah suatu rangkaian logika yang menerima beberapa input data, dan untuk suatu saat tertentu hanya mengizinkan satu data input masuk ke output, yang diatur oleh input selector. Multiplexer juga dapat digunakan pada seleksi data, data routing (perjalanan data), konversi parallel ke seri, menghasilkan bentuk gelombang, menghasilkan fungsi logika, operation sequencing (pengurutan operasi)
5. Demultiplexer (De-Mux) atau disebut juga distributor data. De-Mux memiliki satu kanal input yang didistribusikan ke beberapa kanal output
6. Aljabar Boole adalah salah satu bentuk aljabar yang digunakan untuk menganalisis rangkaian digital dengan dasar logika variabel 1 dan 0.
Aljabar boole dapat digunakan sebagai salah satu cara untuk menganalisis rangkaian dan menyatakan operasinya secara matematik, terutama untuk mendapatkan konfigurasi rangkaian yang paling sederhana (paling sedikit jumlah komponennya).

7. Teorema Aljabar Boole meliputi :

Fungsi-fungsi boole dengan dua peubah atau lebih, dikenal juga dengan hukum-hukum kumulatif, assosiatif, dan distributive yang berlaku dalam aljabar biasa, yaitu:

1. $A + B = B + A$ (komutatif OR)
2. $A \cdot B = B \cdot A$ (komutatif AND)
8. $A + (B + C) = (A + B) + C = A + B + C$ (asosiatif OR)
9. $A(BC) = (AB)C = ABC$ (asosiatif AND)
10. $A(B + C) = AB + AC$ (distributive OR)
11. $(A + B)(C + D) = AC + BC + AD + BD$ (distributive AND)
12. $A + AB = A$
13. $A + \bar{A}B = A + B$

3. Minimalisasi suatu rangkaian dilakukan secara analitis berdasarkan dasar-dasar aljabar Boole

DAFTAR PUSTAKA

Sugiartowo , 2015 , Konsep Dasar Rangkaian Digital Teori dan Aplikasinya, UMJ Press, Jakarta, ISBN: 978-979-8823-89-3

Tocci et all, 2011, Digital System Principles and Applications. New Jersey: Prentice.

Anoname.Boolean Algebra. http://en.wikipedia.org/boolean_Algebra/

John Crowe and Barrie Hayes-GillJohn. 2003. Introduction to Digital Electronics.
Bristol : JW Arrowsmith and Barrie Hayes-Gill

Sumarna.2006.Elektronika Digital.Konsep dasar dan aplikasinya.
Yogyakarta: Graha Ilmu.

Sumarna. 2011. Modul Praktikum Sistem Digital. Yogyakarta : FMIPA UNY