



BUKU AJAR

SISTEM DIGITAL

UNTUK TEKNIK INFORMATIKA

Sri Mulyati

BUKU AJAR

SISTEM DIGITAL

UNTUK TEKNIK INFORMATIKA

Sri Mulyati

BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

Penulis:
Sri Mulyati

Desain Cover:
Ridwan

Tata Letak:
Handariri Rohana

Editor:
Neneng Sri Wahyuni

ISBN:
978-623-5811-52-9

Cetakan Pertama:
Desember, 2021

Hak Cipta 2021, Pada Penulis

Hak Cipta Dilindungi Oleh Undang-Undang

Copyright © 2021

by Penerbit Widina Bhakti Persada Bandung

All Right Reserved

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit.

PENERBIT:
WIDINA BHAKTI PERSADA BANDUNG
(Grup CV. Widina Media Utama)

Komplek Puri Melia Asri Blok C3 No. 17 Desa Bojong Emas
Kec. Solokan Jeruk Kabupaten Bandung, Provinsi Jawa Barat

Anggota IKAPI No. 360/JBA/2020
Website: www.penerbitwidina.com
Instagram: @penerbitwidina

PRAKATA

Alhamdulillah saya panjatkan puji syukur kehadiran Allah SWT yang telah memberikan nikmat sehat nikmat iman dan islam, sholawat serta salam saya sampaikan kepada Nabi Muhammad Rasullulloh SAW berserta para ulama, tabiin dan pengikutnya, semoga kita medapat syafaat di yaumul akhir. Aamiiin.

Alhamdulillah buku ini telah selesai ditulis sebagai buku pertama dan edisi pertama, tentu belum sempurna, dan in sya Allah akan ada penyempurnaan yang disesuaikan pada edisi selanjutnya. Oleh karena hal tersebut, saya menerima kritik dan saran sebagai bekal dalam menyempurnakan buku ini pada edisi selanjutnya. Semoga buku ini dapat bermanfaat, sebagai dasar dalam mempelajari mata kuliah sistem digital, khususnya untuk mahasiswa saya di Fakultas Teknik Universitas Muhammadiyah Tangerang.

Terimakasih kepada rekan dosen UMT yang telah memberikan support terhadap buku ini, dan juga suami serta anak anaku yang selalu memberikan semangat serta menjadi motivasi dalam belajar.

Demikian prakata ini saya sampaikan, terimakasih

Penulis

Sri Mulyati

DAFTAR ISI

PRAKATA	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
BAB 1 SISTEM DIGITAL	1
A. Pengertian Sistem Digital	2
B. Rangkaian Analog dan Rangkaian Digital	8
C. Representasi Besaran Digital	10
BAB 2 GERBANG LOGIKA	17
A. Gerbang AND	20
B. Gerbang OR	22
C. Gerbang NOT	25
D. Gerbang NAND	27
E. Gerbang NOR	30
F. Gerbang X-OR	34
G. Gerbang X-NOR	36
BAB 3 SISTEM BILANGAN DIGITAL DAN KONVERSI BILANG	41
A. Pengertian Sistem Bilangan	42
B. Biner	42
C. Oktal	43
D. Desimal	43
E. Hexadesimal	43
F. Konversi Bilangan	43
1. Biner ke Desimal	43
2. Biner ke Oktal	44
3. Biner ke HexaDesimal	45
4. Oktal ke Biner	45
5. Oktal ke Desimal	45
6. Oktal ke Hexadesimal	46
7. Desimal ke Biner	47
8. Desimal ke Oktal	48
9. Desimal ke Hexadesimal	48
10. Hexadesimal ke Biner	49

11. Hexadesimal ke Desimal	49
12. Hexadesimal ke Oktal	49
BAB 4 MULTIPLEXER	51
A. Pengertian Multiplexer	52
B. Fungsi Multiplexer	54
C. Tabel Kebenaran Multiplexer	54
D. Aljabar Boolean Multiplexer	56
E. Penggunaan Multiplexer	56
BAB 5 PENGERTIAN DEMULTIPLEXER	59
A. Demultiplexer 1 ke 2	67
B. 1-To-4 Demultiplexer	69
C. Demultiplexer 1 Ke 8	72
D. 1-To-8 Demux Using Two 1-To- 4 Demultiplexers	75
E. Implementasi Subtraktor Penuh Menggunakan Demux 1-Ke-8	76
F. Aplikasi Demultiplexer	78
BAB 6 FLIP FLOP	80
A. Pengertian FLIP-FLOP	81
B. FLIP-FLOP SR	82
C. FLIP-FLOP JK	86
D. D FLIP FLOP	89
E. T FLIP FLOP	91
F. Working	97
G. Applications	97
BAB 7 ALJABAR BOOLEAN	99
A. Pengertian Aljabar Boolean	102
B. Tabel Kebenaran Hukum Aljabar Boolean	104
C. Fungsi Aljabar Boolean	105
D. Operasi Aljabar Boolean Dua Variabel dan Tiga Variabel	106
E. Prioritas Operasi Aljabar Boolean	108
F. Bentuk Standar Fungsi Boolean	116
BAB 8 KARNAUGH MAP	121
A. Karnaugh Map Beserta Penjelasannya	123
B. Langkah – Langkah Pemetaan K-Map Secara Umum :	124
C. Penyederhanaan Dua Variabel	124
D. Penyederhanaan Tiga Variabel	132

E. Penyederhanaan Empat Variabel	133
F. Langkah-Langkah Dalam Menggunakan K-Map Adalah Sebagai Berikut	135
G. Rumus Menentukan Jumlah Kotak Dalam K-Map	139
DAFTAR PUSTAKA	142
INDEX	143
PROFIL PENULIS	145

DAFTAR GAMBAR

Gambar 1. 1 Jam tangan analog dan jam tangan digital [14]	4
Gambar 1. 2 Gelombang Analog dalam satu periode	4
Gambar 1. 3 Gelombang Digital dalam satu perioda	5
Gambar 1. 4 Alat ukur Ampere meter analog	5
Gambar 1. 5 Prinsip kerja pada Ampere meter analog	6
Gambar 1. 6 Kontruksi alat ukur analog	6
Gambar 1. 7 Multitester atau AVO-meter analog dan AVO-meter digital	7
Gambar 1. 8 IC Digital	7
Gambar 1. 9 Rangkaian Analog	9
Gambar 1. 10 Rangkaian digital	10
Gambar 1. 11 Rangkaian digital representasi nilai logika 0	11
Gambar 1. 12 Dioda dengan reverse bias	12
Gambar 1. 13 Daerah Cut Off sebuah transistor	13
Gambar 1. 14 Saklar dalam keadaan terbuka	13
Gambar 1. 15 LED	14
 Gambar 2. 1 Rangkaian ekuivalen gerbang AND	20
Gambar 2. 2 Simbol gerbang logika AND	21
Gambar 2. 3 Tabel kebenaran gerbang AND	21
Gambar 2. 4 Ekspresi Boolean gerbang logika AND	22
Gambar 2. 5 Rangkaian ekuivalen gerbang OR	22
Gambar 2. 6 Simbol gerbang OR	23
Gambar 2. 7 Tabel kebenaran gerbang OR	23
Gambar 2. 8 Ekspresi Boolean gerbang OR	24
Gambar 2. 9 Gerbang OR dengan masukan lebih dari dua	24
Gambar 2. 10 Tabel kebenaran gerbang OR dengan 3 masukan	25
Gambar 2. 11 Rangkaian ekivalen gerbang NOT	25
Gambar 2. 12 Simbol gerbang NOT	26
Gambar 2. 13 Tabel Kebenaran NOT	27
Gambar 2. 14 Rangkaian ekivalen gerbang NAND	27
Gambar 2. 15 Simbol gerbang NAND	28
Gambar 2. 16 Tabel kebenaran gerbang NAND	29

Gambar 2. 17 Ekspresi Boolean gerbang NOR.....	29
Gambar 2. 18 Gerbang NAND dengan 3 masukan	29
Gambar 2. 19 Ekspresi Boolean gerbang NAND	30
Gambar 2. 20 Tabel kebenaran gerbang NAND dengan 3 masukan	30
Gambar 2. 21 Rangkaian ekuivalen gerbang NOR	31
Gambar 2. 22 Struktur logika gerbang NOR	31
Gambar 2. 23 Tabel Kebenaran gerbang NOR.....	32
Gambar 2. 24 Ekspresi Boolean NOR.....	32
Gambar 2. 25 Simbol gerbang NOR dengan 3 masukan.....	33
Gambar 2. 26 Ekspresi Boolean gerbang NOR tiga masukan	33
Gambar 2. 27 Tabel kebenaran gerbang NOR tiga masukan.....	33
Gambar 2. 28 Rangkaian ekuivalen gerbang X-OR	34
Gambar 2. 29 Simbol gerbang X-OR.....	35
Gambar 2. 30 Tabel kebenaran X-OR	35
Gambar 2. 31 Ekspresi Boolean gerbang X-OR	36
Gambar 2. 32 Simbol X-OR masukan lebih dari dua.....	36
Gambar 2. 33 Ekspresi Boolean gerbang X-OR masukan lebih dari dua	36
Gambar 2. 34 Rangkaian ekivalen gerbang X-NOR.....	36
Gambar 2. 35 Simbol gerbang X-NOR.....	37
Gambar 2. 36 Tabel kebenaran gerbang X-NOR.....	38
Gambar 2. 37 Ekspresi Boolean gerbang X-NOR	38
Gambar 4. 1 Analogi multiplexer menggunakan saklar.....	53
Gambar 4. 2 Multiplexer dengan kombinasi gerbang logika	53
Gambar 4. 3 Blok 4X1 Multiplexer.....	54
Gambar 4. 4 Tabel kebenaran 4X1 Multiplexer	55
Gambar 4. 5 Rangkaian Logika	55
Gambar 4. 6 Aplikasi dari multiplexer	56
Gambar 5. 1 <i>Demultiplexer</i> 8 arah.....	60
Gambar 5. 2 Rangkaian dasar demultiplexer.....	61
Gambar 5. 3 Rangkaian Logika <i>Demultiplexer</i>	62
Gambar 5. 4 Tabel Kebenaran	63
Gambar 5. 5 Implementasi persamaan $Y(A, B, C) = \sum(2, 4, 5, 7)$	64
Gambar 5. 6 Diagram Dasar Multiplexer	65

Gambar 5. 7 Rangkaian IC TTL 74153 Multiplexer 2 keluaran	65
Gambar 5. 8 Multiplexer 2 Masukan	66
Gambar 5. 9 Demultiplexer 1 ke 2	69
Gambar 5. 10 1-to-4 <i>Demultiplexer</i>	79
Gambar 5. 11 rangkaian Logika	71
Gambar 5. 12 IC 74139	72
Gambar 5. 13 blok <i>demultiplexer</i> 1-ke-8	73
Gambar 5. 14 <i>demultiplexer</i> 1-ke-8	74
Gambar 5. 15 IC logika	75
Gambar 5. 16 1-to-8 DEMUX using Two 1-to- 4 <i>Demultiplexers</i>	76
Gambar 5. 17 Demux 1 ke 8	78
Gambar 5. 18 Demux 1 [16]	79
Gambar 5. 19 Demux 1[17]	79
Gambar 6. 1 Flip Flop SR	82
Gambar 6. 2 Blok Diagram Flip Flop SR	83
Gambar 6. 3 Diagram Rangkaian Flip Flop SR	84
Gambar 6. 4 Status Tabel Flip Flop SR	85
Gambar 6. 5 Flip Flop SR	85
Gambar 6. 6 Flip Flop JK	86
Gambar 6. 7 Flip Flop Master-Budak JK	87
Gambar 6. 8 Diagram Waktu dari Master Flip Flop	88
Gambar 6. 9 Flip flop D	89
Gambar 6. 10 Flip-flop T	91
Gambar 6. 11 Flip-flop T	92
Gambar 6. 12 T Flip Flop" menggunakan "SR Flip Flop	93
Gambar 6. 13 T-Flip Flop" menggunakan "D Flip Flop	93
Gambar 6. 14 T-Flip Flop" menggunakan "JK Flip Flop	94
Gambar 6. 15 Flip flop D	96
Gambar 6. 16 Diagram Rangkaian Flip Flop D	96
Gambar 6. 17 Cara kerja	97
Gambar 6. 18 Flip flop T	97
Gambar 6. 19 Flip Flop T	97
Gambar 6. 20 Flip flop T Paralel	98

Gambar 7. 1 Tabel aljabar boolean	101
Gambar 7. 2 Tabel AND	101
Gambar 7. 3 Tabel OR.....	102
Gambar 7. 4 Hukum Aljabar Boolean	104
Gambar 7. 5 Fungsi Aljabar Boolean	105
Gambar 8. 1 Tabel Kebenaran Karnaugh Map	123
Gambar 8. 2 Dua Variabel.....	124
Gambar 8. 3 K-Map dua variabel.....	124
Gambar 8. 4 K-Map Model-1 untuk dua variabel	125
Gambar 8. 5 pemetaan K-Map dua variabel	125
Gambar 8. 6 K-Map Model-1 untuk 3 variabel	126
Gambar 8. 7 K-Map Model-1 untuk 3 variabel	127
Gambar 8. 8 K-Map Model-1 untuk 3 variabel	128
Gambar 8. 9 K-Map untuk 3 variabel.....	128
Gambar 8. 10 K-Map Model-1 untuk 4 variabel	129
Gambar 8. 11 Tabel dari K-Map 2 variabel	130
Gambar 8. 12 Tabel dari K-Map 2 variabel	130
Gambar 8. 13 Tabel dari K-Map 2 variabel	131
Gambar 8. 14 Tabel dari K-Map 2 variabel	131
Gambar 8. 15 Tiga Variabel	132
Gambar 8. 16 Tiga Variabel	132
Gambar 8. 17 Tiga Variabel	132
Gambar 8. 18 K-Map 4 variabel	133
Gambar 8. 19 K-Map 4 variabel	133
Gambar 8. 20 K-Map 4 variabel	134
Gambar 8. 21 Model K-map tiga variabel.....	135
Gambar 8. 22 Hasil Pemetaan	136
Gambar 8. 23 Setelah dilakukan peng-cover an mainterm	136
Gambar 8. 24 Rangkaian penyederhanaan [13]	136
Gambar 8. 25 Tabel Kebenaran ke K-Map (Kotak-kotak)	137
Gambar 8. 26 Penyelesaian menggunakan K-Maps [5]	138
Gambar 8. 27 Sistem pemetaan pada karnaugh map	138
Gambar 8. 28 $\sum (0); K = 0$	139

Gambar 8. 29 $\Sigma (1,2,3,4); K = 1$	140
Gambar 8. 30 $\Sigma (1,4); K = A' B' + AB$	140
Gambar 8. 31 $\Sigma (1); K = A' B'$	140
Gambar 8. 32 $\Sigma (2,3,4); K = A + B [11, 12]$	141



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 1: SISTEM DIGITAL

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 1

SISTEM DIGITAL

A. PENGERTIAN SISTEM DIGITAL

1. Capaian Pembelajaran :

Setelah membaca Bab I ini, mahasiswa mampu menjelaskan pengertian sistem digital dan menyebutkan contoh-contoh yang termasuk kedalam sistem digital tersebut untuk bidang teknik komputer.

2. Tujuan Instruksional Umum (TIU) :

Mahasiswa mampu menjelaskan pengertian sistem digital dengan tepat.

3. Tujuan Instruksional Khusus (TIK) :

- Mahasiswa dapat menyebutkan contoh-contoh sistem digital pada bidang teknik komputer.
- Mengetahui prinsip kerja analog dan digital
- Membedakan gelombang analog dan gelombang digital
- Mengetahui keuntungan sistem digital

4. Pendahuluan

Sistem digital merupakan pemahaman yang paling sederhana dan mendasar dalam pembelajaran di bidang teknik informatika. Dalam Sistem digital akan membahas tentang sistem bilangan, dan konversi bilangan, gerbang logika dan rangkaian logika, aljabar boole dan Peta karnaugh. Inilah hal-hal yang paling utama dalam sistem digital.

5. Materi Utama

Perkembangan teknologi berjalan begitu cepat, terbukti dengan terciptanya komponen-komponen yang berbasis nano teknologi. Pada jaman dahulu komponen-komponen utama peralatan listrik terbuat dari lampu tabung, kemudian berkembang menjadi komponen transistor, selanjutnya berkembang lagi menjadi komponen transistor logic (TTL), kemudian berkembang menjadi komponen-komponen digital berupa gerbang-gerbang logika dan nano teknologi.

Pengertian sistem digital adalah suatu sistem yang terdiri dari komponen-komponen elektronika dan gerbang-gerbang logika yang diproses secara biner oleh bilangan biner yaitu bilangan 0 (nol) dan bilangan 1 (satu), serta bilangan 0 dan 1 tersebut merupakan dasar pembentukan angka-angka digital mulai dari 0 sampai dengan 9.

Contoh alat yang menggunakan sistem digital adalah jam tangan digital, timbangan digital, papan reklame digital, alat ukur digital, timbangan bayi digital, TV digital, radio digital dan lain sebagainya.

Kita mengenal istilah analog dan istilah digital. Analog menunjukkan suatu peralatan yang bekerja dengan prinsip Galvanometer, dimana arus yang mengalir pada suatu kumparan medan magnet dapat menggerakkan jarum pada angka tertentu. Sedangkan digital menunjukkan pada suatu peralatan yang berkerja pada prinsip gerbang-gerbang logika dengan keluaran berupa angka-angka digital.

Keberadaan sistem analog dan sistem digital dapat dilihat dari sisi keluaran yang dihasilkan pada suatu peralatan. Keberadaan sistem analog dapat berupa arus, tegangan atau gerakan jarum meter, contoh speedometer pada sebuah kendaraan mobil. Keberadaan sistem digital dapat berupa nilai digit, contoh adalah jam digital.

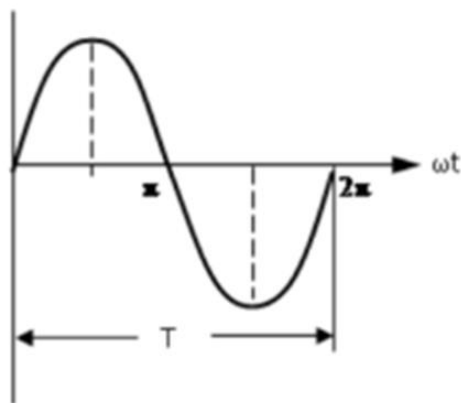
Marilah kita mengamati sebuah jam tangan analog dan jam tangan digital. Pada peralatan analog, misalnya sebuah jam, kita akan menemukan perubahan dari waktu ke waktu secara kontinue, sedangkan pada peralatan digital perubahan dari waktu ke waktu tidak terjadi secara kontinue tetapi terjadi secara diskrit.



Gambar 1. 1 Jam tangan analog dan jam tangan digital [14]

Kita dapat membedakan antara gelombang signal analog dan gelombang signal digital. Kata analog dapat kita asumsikan dengan berlanjut atau continue, sedangkan digital kita asumsikan dengan istilah step by step atau sering dikenal dengan nama diskrit. Marilah kita lihat kedua bentuk gelombang tersebut.

6. Bentuk Gelombang Sinyal Analog

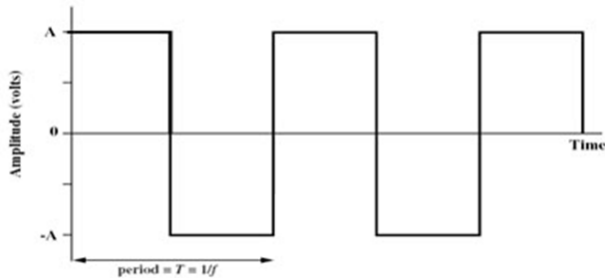


Gambar 1. 2 Gelombang Analog dalam satu periode

Bentuk gelombang signal digital berupa step by step atau diskrit, dan sering kita mengenalnya dengan sebutan gelombang kotak.

7. Bentuk Gelombang Sinyal Digital

Gelombang tersebut dapat kita lihat sebagai berikut.



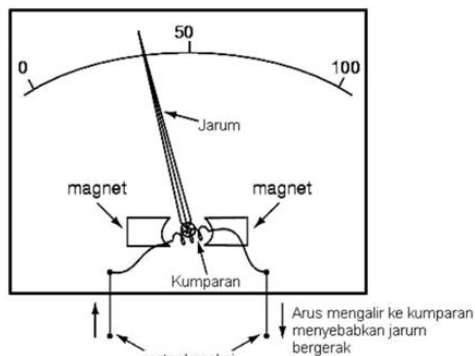
Gambar 1. 3 Gelombang Digital dalam satu perioda

Pada jaman sekarang hampir semua benda atau alat ukur terbentuk dengan menggunakan prinsip prinsip sistem digital. Sebagai contoh, pada awal teknologi analog, penggunaan alat ukur Ampere meter menggunakan kumparan medan magnet, dan hasilnya berupa gerakan jarum yang menunjuk ke nilai tertentu. Sedangkan pada jaman digital alat ukur Ampere meter sudah berubah dan ada yang tidak menggunakan medan magnet dan jarum, namun menggunakan teknologi gerbang gerbang logika yang hasilnya langsung berupa angka angka digital. Berikut adalah contoh sederhana alat ukur Amper-meter analog.

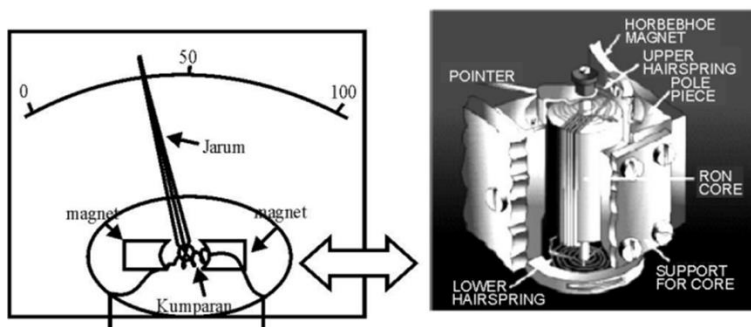


Gambar 1. 4 Alat ukur Ampere meter analog

Dalam alat ukur tersebut mengandung mekanik yang terdiri dari kumparan dan menggerakkan jarum pada nilai arus tertentu dengan satuan Amper. Amper meter diatas merupakan ampere meter arus bolak balik (*Alternating Current*). Arus bolak balik berupa gelombang yang berbentuk sinusoidal. Selain ampere meter arus bolak balik, kita juga mengenal alat ukur ampere meter arus searah (*Direct Current*). Gelombang arus berupa garis lurus.



Gambar 1. 5 Prinsip kerja pada Ampere meter analog



Gambar 1. 6 Kontruksi alat ukur analog

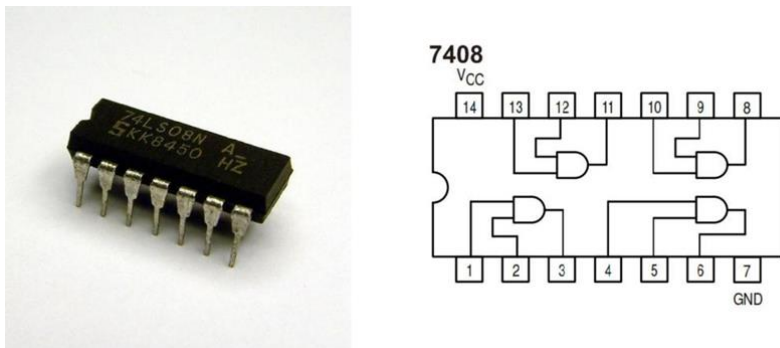
Pada alat ukur ampere meter tersebut diatas menggunakan prinsip Galvanometer, yaitu adanya suatu simpangan jarum ukur jika terdapat arus yang mengalir pada kumparan medan magnet. Prinsip Galvanometer ini digunakan pada berbagai alat ukur analog. Contoh alat ukur Ampere meter, Ohm meter, Volt meter dan lain sebagainya.

Marilah kita bedakan suatu alat ukur multitester atau yang sering disebut dengan nama AVO meter yaitu singkatan dari Ampere, Volt dan Ohm meter, antara AVO meter analog dan AVO meter digital.



Gambar 1. 7 Multitester atau AVO-meter analog dan AVO-meter digital

Yang membedakan antara alat ukur analog dan digital adalah komponen yang digunakan pada masing-masing alat tersebut, dimana pada alat ukur analog menggunakan prinsip Galvanometer dengan keluaran berupa gerakan jarum, sedangkan pada alat ukur digital menggunakan komponen komponen gerbang logika yang menghasilkan keluaran berupa digit angka angka.



Gambar 1. 8 IC Digital

Gambar diatas merupakan contoh sebuah IC 7480 yang dapat digunakan untuk rangkaian digital, dimana di dalam IC tersebut terdapat 4 gerbang AND. Keempat gerbang AND tersebut dapat digunakan jika IC tersebut diberikan sumber tegangan, yaitu pada pin 14 diberikan tegangan positif dan pada pin 7 diberikan tegangan negatif [15].

B. RANGKAIAN ANALOG DAN RANGKAIAN DIGITAL

1. Capaian Pembelajaran :

Mahasiswa dapat menjelaskan, membaca, memahami, memberi contoh, menerapkan, memeriksa, , mengidentifikasi, sistem digital.

2. Tujuan Instruksional Umum (TIU) :

- 1) Mahasiswa dapat memahami rangkaian analog dan rangkaian digital
- 2) Dapat membaca rangkaian analog dan digital
- 3) Dapat memberi contoh rangkaian analog dan digital

3. Tujuan Instruksional Khusus (TIK) :

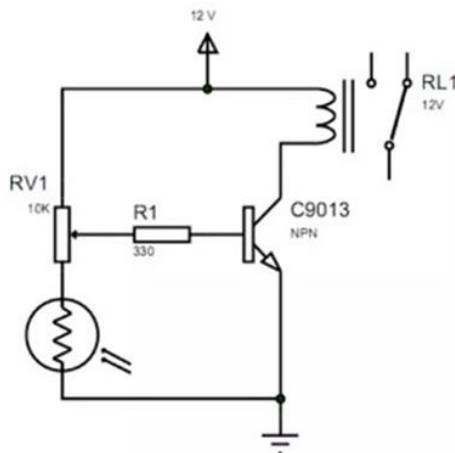
- 1) Mahasiswa dapat menyebutkan (C1) contoh Representasi nilai logika 0 dan logika 1.
- 2) Dapat menyebutkan (C1) komponen komponen yang memiliki Representasi nilai logika 0 dan 1.
- 3) Dapat menjelaskan (C1) keuntungan keuntungan sistem digital.
- 4) Mahasiswa dapat menerapkan (C3) Representasi logika 0 dan 1 pada rangkaian digital
- 5) Mahasiswa dapat memeriksa (C4) rangkaian Representasi bernilai logika 0 dan Representasi bernilai logika 1.

4. Materi

- **rangkaiian elektronika** berfungsi mengolah signal
- **Komponen elektronika pasif** tidak memerlukan sumber arus atau tegangan
- **Komponen elektronika aktif** memerlukan sumber arus atau tegangan

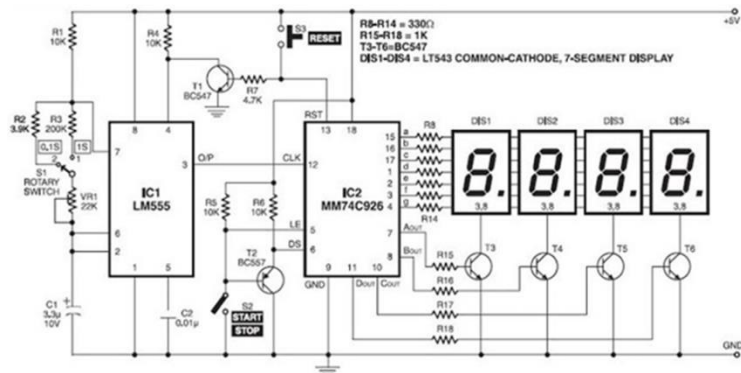
- **Rangkaian analog** signal bersifat kontinyu
- **rangkaian digital** signal bersifat diskrit

Perhatikan salah satu contoh yang termasuk kedalam rangkaian analog. Dalam rangkaian analog memiliki komponen elektronika aktif dan elektronika pasif, namun tidak memiliki komponen gerbang gerbang logika. Rangkaian analog di bawah ini, tersusun dari komponen : potensiometer (RV1), Resistor (R1), Transistor (C9013), LDR, dan Relay. Rangkaian analog berfungsi untuk mengolah sinyal kontinyu.



Gambar 1. 9 Rangkaian Analog

Berikutnya adalah salah satu contoh rangkaian digital. Rangkaian digital terdiri dari komponen komponen elektronika aktif dan komponen elektronika pasif, dan dilengkapi dengan komponen gerbang logika. Jika kita perhatikan dalam rangkaian di bawah ini, tersusun dari Resistor R1-R18), Kapasitor (C1-C2), Transistor (T1-T6), *seven segment*, dan IC gerbang logika (IC1 dan IC2) dan saklar. Rangkaian ini berfungsi untuk mengolah sinyal diskrit.



Gambar 1. 10 Rangkaian digital

C. REPRESENTASI BESARAN DIGITAL

1. Capaian Pembelajaran :

Mahasiswa dapat menjelaskan, membaca, memahami, memberi contoh, menerapkan, memeriksa, , mengidentifikasi, sistem digital.

2. Tujuan Instruksional Umum (TIU) :

- 1) Mahasiswa dapat memahami rangkaian analog dan rangkaian digital
- 2) Dapat membaca rangkaian analog dan digital
- 3) Dapat memberi contoh rangkaian analog dan digital

3. Tujuan Instruksional Khusus (TIK) :

- 1) Mahasiswa dapat menyebutkan (C1) contoh Representasi nilai logika 0 dan logika 1.
- 2) Dapat menyebutkan (C1) komponen komponen yang memiliki Representasi nilai logika 0 dan 1.
- 3) Dapat menjelaskan (C1) keuntungan keuntungan sistem digital.
- 4) Mahasiswa dapat menerapkan (C3) Representasi logika 0 dan 1 pada rangkaian digital
- 5) Mahasiswa dapat memeriksa (C4) rangkaian Representasi bernilai logika 0 dan Representasi bernilai logika 1.

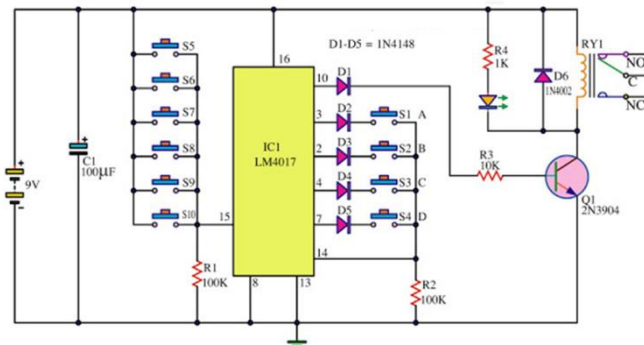
4. Materi Utama

a. Representasi Nilai Logika 0

Nilai logika 0 pada beberapa komponen rangkaian listrik dapat dinyatakan dengan besarnya suatu nilai, atau bekerja tidaknya suatu komponen elektronika dalam suatu rangkaian. Komponen-komponen tersebut dapat berupa komponen sebagai berikut : sumber tegangan, titik sasis atau grounding, dioda, transistor, dan Light Emitting Diode (LED).

Komponen pertama adalah suatu sumber tegangan dalam suatu rangkaian digital. Apabila suatu rangkaian digital yang terdiri dari beberapa komponen, diantaranya terdapat sumber tegangan, dan jika sumber tegangan tersebut bernilai antara 0 – 0,8 Volt, maka nilai tersebut menyatakan nilai logika 0, atau dengan kata lain representasi sumber tegangan tersebut bernilai 0 (nol).

Perhatikan Rangkaian digital dibawah ini.

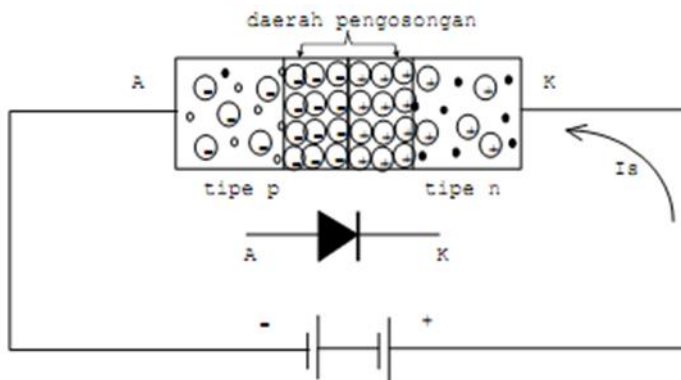


Gambar 1. 11 Rangkaian digital representasi nilai logika 0

Pada Gambar rangkaian digital di atas, jika kita perhatikan, dan kita pelajari jalannya arus dari sebuah sumber tegangan searah *Direct Current* (DC), maka akan bekerja jika rangkaian terpasang dengan benar dan memiliki sumber tegangan sebesar 9 Volt. Bagaimana jika sumber tegangan bernilai antara 0 – 0,8 Volt ? apa yang terjadi pada rangkaian digital tersebut ? Tentunya nilai tegangan tersebut merupakan representasi nilai logika 0 (nol).

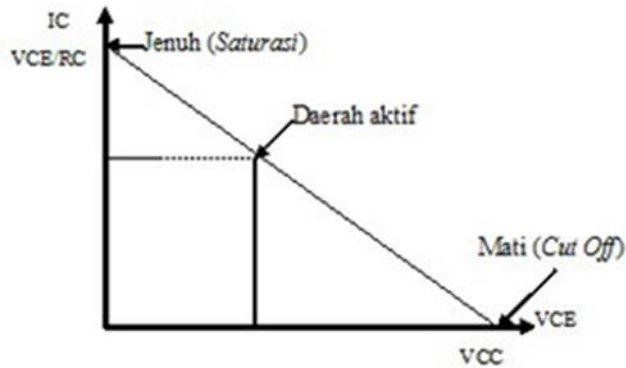
Komponen kedua, lebih tepatnya dikatakan titik *grounding*/sasis atau pertahanan pada suatu rangkaian. Pada Gambar di atas terdapat sebuah titik yang terpasang dengan *ground*, dengan simbol (). Titik *ground* adalah sebuah titik yang tidak mempunyai nilai tegangan, yaitu nilai potensial paling rendah dan bernilai 0 (nol). Dengan demikian sebuah *grounding* merupakan representasi nilai logika 0.

Komponen ketiga, adalah dioda. Dioda (D1-D6) yang terdapat pada rangkaian digital di atas dipasang reverse bias, artinya dioda tersebut tidak menghantarkan arus. Jika sebuah dioda dalam suatu rangkaian tidak berfungsi mengalirkan arus listrik, atau terpasang secara reverse bias, maka komponen dioda dalam rangkaian tersebut merupakan representasi nilai logika 0.



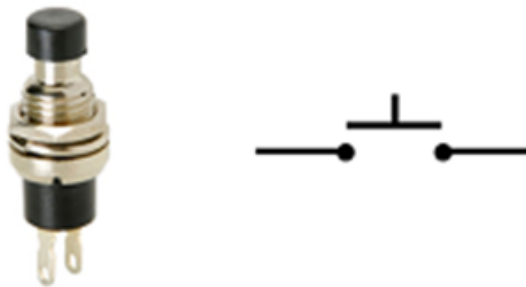
Gambar 1. 12 Dioda dengan reverse bias

Komponen keempat yaitu daerah cut off transistor. Pada rangkaian diatas terdapat transistor (Q1). Jika transistor dalam keadaan cut off maka transistor sama dengan saklar yang terbuka, dan merupakan representasi transistor nilai logika 0.



Gambar 1. 13 Daerah Cut Off sebuah transistor

Komponen kelima yaitu saklar. Saklar pada rangkaian di atas (S1-S4) dalam keadaan terbuka tidak dapat menghantarkan arus listrik, sehingga bisa dikatakan saklar dalam rangkaian tersebut representasi nilai logika 0.



Gambar 1. 14 Saklar dalam keadaan terbuka

Komponen terakhir adalah Light Emitting Diode (LED). Pada rangkaian diatas terdapat LED yang terpasang dengan benar. Jika sumber tegangan bernilai logika 1 maka lampu LED akan menyala, namun apabila lampu LED tersebut dalam keadaan tidak menyala, maka dikatakan bahwa dioda tersebut representasi nilai logika 0.



Gambar 1. 15 LED

Dengan mempelajari representasi nilai logika 0 diatas, tentunya kita dapat menentukan komponen komponen dalam rangkaian representasi nilai logika 1. Representasi nilai logika 1 adalah sebagai berikut : Tegangan listrik 2 – 5 Volt, Titik potensial catu daya (+Vcc), Dioda dengan *forward* bias, Transistor dalam keadaan jenuh (*saturated*), Saklar dalam keadaan tertutup, Light Emiting Diode dalam keadaan menyala.

b. Kelebihan yang dimiliki pada Sistem Digital adalah sebagai berikut :

- 1) Dalam merancang dengan Sistem digital lebih mudah dilakukan.
- 2) Memudahkan dalam penyimpanan informasi
- 3) Sistem digital memiliki ketelitian yang lebih besar
- 4) Sistem kerja pada sistem digital dapat dilakukan melalui pemograman
- 5) Sistem digital dapat memperkecil derau (*noise*)
- 6) Beberapa komponen digital dapat dikemas menjadi sebuah *integrated Circuit* (IC).

(Sumber : <https://berbagiilmuprogram.blogspot.com/2011/10/sistem-digital.html>)

c. Penutup

Sebagai kesimpulan bahwa semua peralatan yang mengandung gerbang gerbang logika dan menghasilkan keluaran berupa angka angka merupakan peralatan yang menggunakan sistem digital. Selain sistem digital tergolong ke dalam sistem analog yang berkaitan dengan medan magnet, dan berprinsip kepada Galvanometer.

d. Evaluasi

1. Apa yang dimaksud dengan Sistem Digital
2. Apa yang dimaksud dengan perubahan continue ? jelaskan dengan sebuah contoh !
3. Apa yang dimaksud dengan perubahan diskrit ? jelaskan dengan sebuah contoh !
4. Apa bedanya antara sistem analog dan sistem digital ?
5. Sebutkan sepuluh contoh alat-alat yang menggunakan prinsip analog !
6. Sebutkan sepuluh contoh alat-alat yang menggunakan prinsip digital !
7. Jelaskan bentuk gelombang analog !
8. Jelaskan bentuk gelombang digital !
9. Buatlah contoh rangkaian analog, jelaskan cara kerjanya !
10. Buatlah contoh rangkaian digital, jelaskan cara kerjanya !
11. Sebutkan sebuah jenis IC Digital, jelaskan gerbang gerbang yang terdapat pada Bagian ID tersebut !
12. Apa yang dimaksud dengan Representasi nilai logika 0 dan Representasi nilai logika 1 ?
13. Jelaskan pengertian Representasi bernilai logika 0 dan Representasi bernilai logika 1 pada sebuah rangkaian !
14. Jelaskan beberapa keuntungan dari sistem digital.



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 2: GERBANG LOGIKA

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 2

GERBANG LOGIKA

Gerbang logika atau gerbang logik adalah suatu entitas dalam elektronika dan matematika Boolean yang mengubah satu atau beberapa masukan logik menjadi sebuah sinyal keluaran logik. Gerbang logika terutama diimplementasikan secara elektronis menggunakan diode atau transistor, akan tetapi dapat pula dibangun menggunakan susunan komponen-komponen yang memanfaatkan sifat-sifat elektromagnetik (relay), cairan, dan optik.

- **Pengertian Gerbang Logika Dasar dan Jenis-jenisnya**

Gerbang Logika atau dalam bahasa Inggris disebut dengan Logic Gate adalah dasar pembentuk Sistem Elektronika Digital yang berfungsi untuk mengubah satu atau beberapa Masukan (masukan) menjadi sebuah sinyal Keluaran (Keluaran) Logis. Gerbang Logika beroperasi berdasarkan sistem bilangan biner yaitu bilangan yang hanya memiliki 2 kode simbol yakni 0 dan 1 dengan menggunakan Teori Aljabar Boolean.

Gerbang Logika yang diterapkan dalam Sistem Elektronika Digital pada dasarnya menggunakan Komponen-komponen Elektronika seperti *Integrated Circuit* (IC), Dioda, Transistor, Relay, Optik maupun Elemen Mekanikal.

- **Jenis-jenis Gerbang Logika Dasar dan Simbolnya**

Terdapat 7 jenis Gerbang Logika Dasar yang membentuk sebuah Sistem Elektronika Digital, yaitu :

1. Gerbang AND
2. Gerbang OR
3. Gerbang NOT
4. Gerbang NAND
5. Gerbang NOR
6. Gerbang X-OR (Exclusive OR)
7. Gerbang X-NOR (Exclusive NOR)

Tabel yang berisikan kombinasi-kombinasi Variabel Masukan (Masukan) yang menghasilkan Keluaran (Keluaran) Logis disebut dengan **“Tabel Kebenaran”** atau **“Truth Table”**.

Masukan dan Keluaran pada Gerbang Logika hanya memiliki 2 level. Kedua Level tersebut pada umumnya dapat dilambangkan dengan :

- HIGH (tinggi) dan LOW (rendah)
- TRUE (benar) dan FALSE (salah)
- ON (Hidup) dan OFF (Mati)
- 1 dan 0

Contoh Penerapannya ke dalam Rangkaian Elektronika yang memakai Transistor TTL (Transistor-transistor Logic), maka 0V dalam Rangkaian akan diasumsikan sebagai **“LOW”** atau **“0”** sedangkan 5V akan diasumsikan sebagai **“HIGH”** atau **“1”**.

- **Gerbang Elektronika**

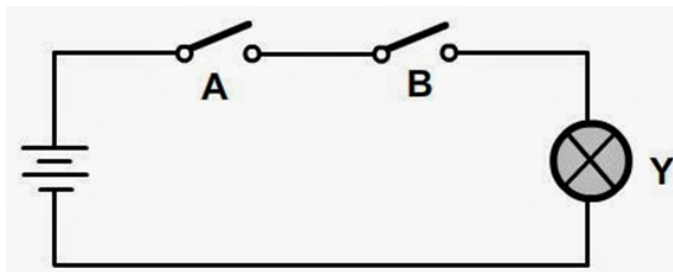
Untuk membangun sistem logika yang berfungsi secara penuh, relay, tabung hampa, atau transistor dapat digunakan. Contoh gerbang logika yaitu logika resistor-transistor (resistor-transistor logic / RTL), logika diode-transistor (diode-transistor logic / DTL), logika transistor-transistor (transistor-transistor logic / TTL), dan logika complementary metal-oxide-semiconductor (CMOS).

Sumber : Menyunting Gerbang logika - Wikipedia bahasa Indonesia, ensiklopedia bebas

A. GERBANG AND

1. Dasar Gerbang AND

Rangkaian dasar gerbang AND dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, dua buah saklar (A, B) dan LED (Y). Dua buah saklar A dan B disusun seri seperti pada gambar. Kedua saklar paralel tersebut dihubungkan dengan sumber tegangan, dan LED secara seri.



Gambar 2. 1 Rangkaian ekuivalen gerbang AND

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika kedua saklar dalam keadaan terhubung, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika salah satu saklar atau kedua saklar terbuka maka LED tidak akan mendapatkan arus listrik searah, sehingga LED menjadi mati, karena arus listrik terputus oleh salah satu atau kedua saklar tersebut.

2. Simbol Gerbang AND

Secara umum simbol gerbang AND dapat dilihat pada gambar dibawah ini. Simbol gerbang AND dibawah ini terdiri dari dua buah masukan yaitu A dan B dan sebuah keluaran yaitu Y.



Gambar 2. 2 Simbol gerbang logika AND

3. Tabel Kebenaran Gerbang AND

Perhatikan tabel kebenaran dibawah untuk menjelaskan gerbang AND dengan dua buah masukan dan satu keluaran. Tabel kebenaran dibuat berdasarkan rangkaian ekivalen diatas.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika masukan atau kedua saklar dalam kondisi terhubung atau bernilai logika 1. Dengan demikian **keluaran akan bernilai logika 1 jika kedua masukan bernilai 1, selain itu keluaran bernilai 0**. Masukan di asumsikan sebagai A dan B, sedangkan keluaran diasumsikan sebagai Y.

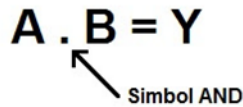
A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Gambar 2. 3 Tabel kebenaran gerbang AND

4. Ekspresi Boolean Gerbang AND

Ekspresi Boolean dari gerbang AND dapat ditulis dengan tanda titik () seperti pada gambar dibawah ini.

$$A \cdot B = Y$$

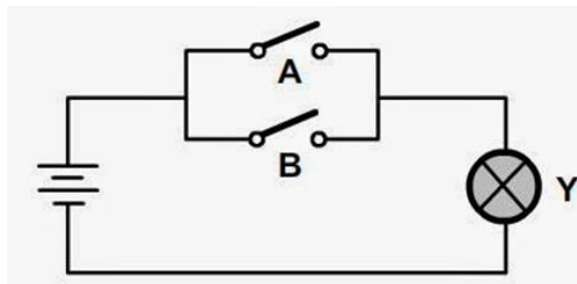

 Simbol AND

Gambar 2. 4 Ekspresi Boolean gerbang logika AND

B. GERBANG OR

1. Dasar Gerbang OR

Rangkaian dasar gerbang OR dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, dua buah saklar (A, B) dan LED (Y). Dua buah saklar A dan B disusun paralel seperti pada gambar. Kedua saklar paralel tersebut dihubungkan dengan sumber tegangan, dan LED secara seri.



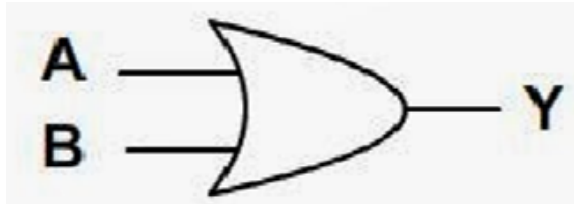
Gambar 2. 5 Rangkaian ekuivalen gerbang OR

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika salah satu saklar atau kedua saklar dalam keadaan terhubung, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika kedua saklar terbuka maka LED tidak akan mendapatkan arus listrik searah, sehingga LED menjadi mati, karena arus listrik terputus oleh kedua tersebut.

2. Simbol Gerbang OR

Secara umum simbol gerbang OR dapat dilihat pada gambar dibawah ini. Simbol gerbang OR dibawah ini terdiri dari dua buah masukan yaitu A dan B dan sebuah keluaran yaitu Y.



Gambar 2. 6 Simbol gerbang OR

3. Tabel Kebenaran Gerbang OR

Perhatikan tabel kebenaran dibawah untuk menjelaskan gerbang OR dengan dua buah masukan dan satu keluaran. Tabel kebenaran dibuat berdasarkan rangkaian ekivalen diatas.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika salah satu atau kedua masukan atau kedua saklar dalam kondisi terhubung atau bernilai logika 1. Dengan demikian keluaran akan bernilai logika 1 jika salah satu atau kedua masukan bernilai 1, selain itu keluaran bernilai 0. Masukan di asumsikan sebagai A dan B, sedangkan keluaran diasumsikan sebagai Y.

A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Gambar 2. 7 Tabel kebenaran gerbang OR

4. Aljabar Boolean untuk Gerbang OR

Ekspresi Boolean dari gerbang OR dapat ditulis dengan tanda tambah (+) seperti pada gambar dibawah ini.

$$A + B = Y$$

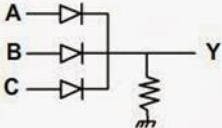
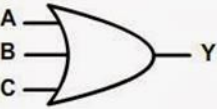
↖
Simbol OR

Gambar 2. 8 Ekspresi Boolean gerbang OR

Perlu dicatat bahwa tanda tambah (+) merupakan simbol Boolean untuk OR.

Tabel kebenaran OR dapat juga ditulis dengan menggunakan notasi aljabar Boolean seperti pada gambar dibawah ini.

Gerbang OR Dengan Masukan Lebih Dari Dua

RANGKAIAN	SIMBOL	FUNGSI
		$A = A + B + C$

Gambar 2. 9 Gerbang OR dengan masukan lebih dari dua

5. Tabel Kebenaran Gerbang OR untuk tiga masukan dan satu keluaran.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika salah satu atau kedua masukan atau kedua saklar dalam kondisi terhubung atau bernilai logika 1. Dengan demikian keluaran akan bernilai logika 1 jika salah satu atau kedua

masukan bernilai 1, selain itu keluaran bernilai 0. Masukan di asumsikan sebagai A, B dan C, sedangkan keluaran diasumsikan sebagai Y.

A	B	C	$Y = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Gambar 2. 10 Tabel kebenaran gerbang OR dengan 3 masukan

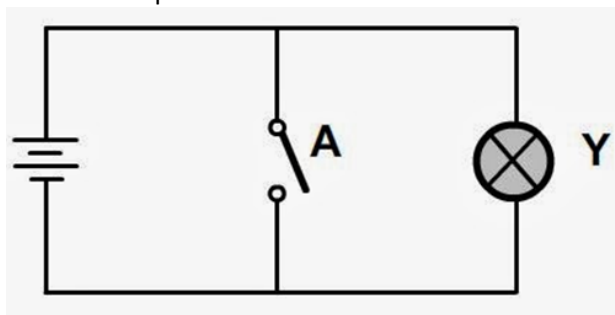
Dalam penyusunan tabel kebenaran (*truth table*), jumlah masukan terdiri dari atau $2n=8$ bit masukan (jika $n=3$).

Sumber : Gerbang OR ~ Portal Materi Dasar Elektronika
(dasarelektronika.blogspot.com)

C. GERBANG NOT

Dasar Gerbang NOT

Rangkaian dasar gerbang NOT dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, sebuah saklar (A) dan LED (Y). Saklar tersebut dihubungkan dengan sumber tegangan, dan LED secara paralel.



Gambar 2. 11 Rangkaian ekivalen gerbang NOT

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika saklar dalam keadaan terbuka, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika saklar terhubung maka LED tidak akan mendapatkan arus listrik searah, sehingga LED menjadi mati, karena arus listrik dihubung singkat oleh saklar tersebut.

6. Simbol Gerbang NOT

Secara umum simbol gerbang NOT dapat dilihat pada gambar dibawah ini. Simbol gerbang NOT dibawah ini terdiri dari sebuah masukan yaitu A dan sebuah keluaran yaitu Y.



Gambar 2. 12 Simbol gerbang NOT

7. Tabel kebenaran gerbang NOT

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika salah satu atau kedua masukan atau kedua saklar dalam kondisi terhubung atau bernilai logika 1. Dengan demikian keluaran akan bernilai logika 1 saklar bernilai 0, selain itu keluaran bernilai 1. Masukan di asumsikan sebagai A, sedangkan keluaran diasumsikan sebagai Y.

A	$Y = \bar{A}$
0	1
1	0

Gambar 2. 13 Tabel Kebenaran NOT

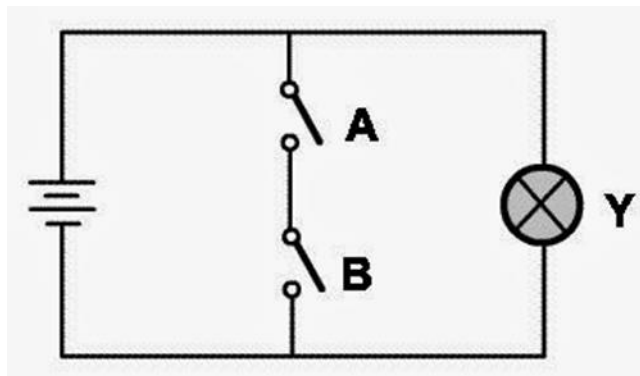
Contoh soal

<https://dasarelektroika2.blogspot.com/2014/02/gerbang-not.html>

D. GERBANG NAND

1. Dasar Gerbang NAND

Rangkaian dasar gerbang NAND atau NOT AND dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, dua buah saklar (A, B) dan LED (Y). Dua buah saklar A dan B disusun seri seperti pada gambar. Kedua saklar tersebut dihubungkan dengan sumber tegangan, dan LED secara paralel.



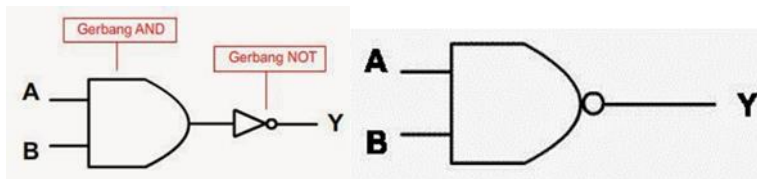
Gambar 2. 14 Rangkaian ekuivalen gerbang NAND

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika salah satu saklar atau kedua saklar dalam keadaan terbuka, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika kedua saklar terhubung maka LED tidak akan mendapatkan arus listrik searah, sehingga LED menjadi mati, karena arus listrik dihubung singkat oleh kedua saklar tersebut.

2. Simbol Gerbang NAND

Secara umum simbol gerbang NAND dapat dilihat pada gambar dibawah ini. Simbol gerbang NAND dibawah ini terdiri dari dua buah masukan yaitu A dan B dan sebuah keluaran yaitu Y.



Gambar 2. 15 Simbol gerbang NAND

3. Tabel Kebenaran Gerbang NAND

Perhatikan tabel kebenaran dibawah untuk menjelaskan gerbang AND dengan dua buah masukan dan satu keluaran. Tabel kebenaran dibuat berdasarkan rangkaian ekivalen diatas.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika masukan atau kedua saklar dalam kondisi terbuka atau bernilai logika 0. Dengan demikian keluaran akan bernilai logika 1 jika salah satu atau kedua masukan bernilai 0, selain itu keluaran bernilai 1. Masukan diasumsikan sebagai A dan B, sedangkan keluaran diasumsikan sebagai Y.

A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	0
1	0	0
1	1	0

Gambar 2. 16 Tabel kebenaran gerbang NAND

Aljabar Boolean

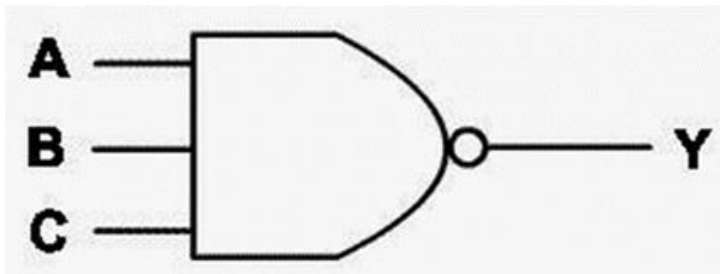
Ekspresi Boolean dari gerbang NAND dapat ditulis dengan tanda titik dan garis atas () seperti pada gambar dibawah ini. Persamaan Boolean gerbang NAND dibaca NOT AND B.

$$Y = \overline{A \cdot B}$$

Gambar 2. 17 Ekspresi Boolean gerbang NOR

4. Gerbang NAND Dengan Masukan Lebih Dari Dua

Terlepas dari berapa jumlah masukannya, sebuah gerbang NAND selalu ekuivalen dengan gabungan gerbang AND dan gerbang NOT. Pada gambar berikut menunjukkan gerbang NAND dengan 3 masukan.



Gambar 2. 18 Gerbang NAND dengan 3 masukan

Sinyal-sinyal dari ke 3 masukan di AND kan kemudian hasilnya di NOT-kan. Persamaan aljabar Boole untuk 3 masukan atau lebih adalah :

$$Y = \overline{A \cdot B \cdot C \cdot \dots}$$

Gambar 2. 19 Ekspresi Boolean gerbang NAND

Analisa rangkaiannya adalah sebagai berikut. Jika salah satu atau lebih masukannya rendah, hasil operasi AND terhadap masukan-masukannya rendah, ini berarti keluaran akhir tinggi. Jika semua masukannya dalam keadaan tinggi, hasil operasi AND adalah tinggi dan keluaran akhir menjadi rendah.

Tabel Kebenaran

Tabel berikut membuktikan semua kemungkinan operasi gerbang NAND dengan 3 masukan. Jelas bahwa rangkaian ini hanya mengenali semua masukan yang mengandung bitnya 0. Artinya bahwa untuk mendapatkan keluaran yang tinggi (1), salah satu masukannya harus rendah (1).

A	B	C	$Y = \overline{A \cdot B \cdot C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

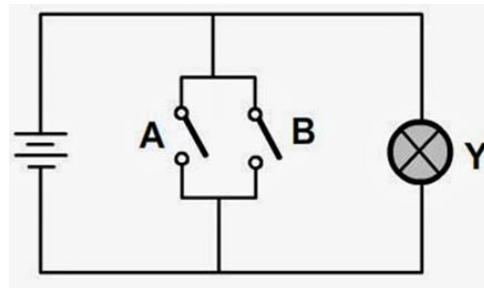
Gambar 2. 20 Tabel kebenaran gerbang NAND dengan 3 masukan

E. GERBANG NOR

1. Dasar Gerbang NOR

Rangkaian dasar gerbang NOR dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, dua buah saklar (A, B) dan LED (Y). Dua buah saklar A dan B disusun paralel

seperti pada gambar. Kedua saklar tersebut dihubungkan dengan sumber tegangan, dan LED secara paralel.



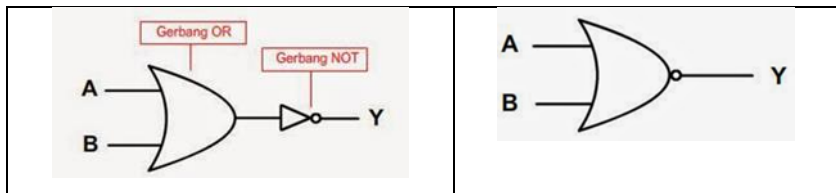
Gambar 2. 21 Rangkaian ekuivalen gerbang NOR

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika kedua saklar dalam keadaan terbuka, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika salah satu saklar (atau duanya) terhubung maka LED tidak akan mendapatkan arus listrik searah, sehingga LED menjadi mati, karena arus listrik dihubung singkat oleh salah satu saklar (atau kedua saklar) tersebut.

Simbol

Secara umum simbol gerbang NOR dapat dilihat pada gambar dibawah ini. Simbol gerbang NOR dibawah ini terdiri dari dua buah masukan yaitu A dan B dan sebuah keluaran yaitu Y.



Gambar 2. 22 Struktur logika gerbang NOR

2. Tabel Kebenaran Gerbang NOR

Perhatikan tabel kebenaran dibawah untuk menjelaskan gerbang AND dengan dua buah masukan dan satu keluaran. Tabel kebenaran dibuat berdasarkan rangkaian ekivalen diatas.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika masukan atau kedua saklar dalam kondisi terbuka atau bernilai logika 0. Dengan demikian **keluaran akan bernilai logika 1 jika kedua masukan bernilai 0**, selain itu keluaran bernilai 1. Masukan diasumsikan sebagai A dan B, sedangkan keluaran diasumsikan sebagai Y.

A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Gambar 2. 23 Tabel Kebenaran gerbang NOR

3. Aljabar Bolean

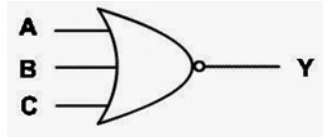
Ekspresi Boolean dari gerbang NOR dapat ditulis dengan tanda tambah dan garis atas () seperti pada gambar dibawah ini. Persamaan Boolean gerbang NOR dibawah dibaca Y adalah NOT A OR B.

$$Y = \overline{A + B}$$

Gambar 2. 24 Ekspresi Boolean NOR

4. Gerbang NOR dengan Masukan Lebih Dari Dua

Berapapun jumlah masukan yang terdapat pada sebuah gerbang NOR, struktur logikanya tetap ekuivalen dengan gabungan sebuah gerbang OR dan NOT. Pada gambar berikut menunjukkan gerbang NOR dengan 3 masukan.



Gambar 2. 25 Simbol gerbang NOR dengan 3 masukan

Sinyal-sinyal dari ke 3 masukan di-OR-kan kemudian hasilnya di-NOT-kan. Persamaan aljabar Boole untuk 3 masukan atau lebih adalah :

$$Y = \overline{A + B + C + \dots}$$

Gambar 2. 26 Ekpresi Boolean gerbang NOR tiga masukan

5. Tabel Kebenaran

A	B	C	$Y = \overline{A + B + C}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

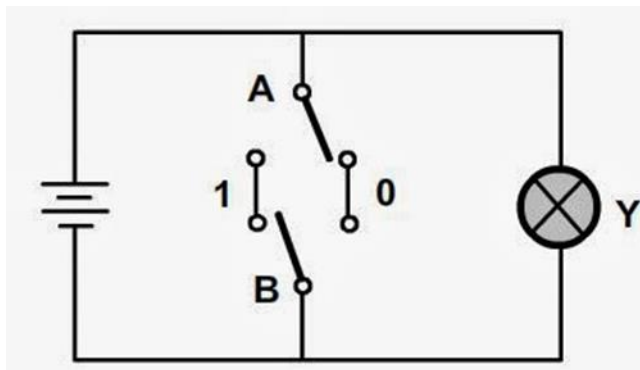
Gambar 2. 27 Tabel kebenaran gerbang NOR tiga masukan

<https://dasarelektronika2.blogspot.com/2014/02/gerbang-nor.html>

F. GERBANG X-OR

1. Dasar Gerbang X-OR

Rangkaian dasar gerbang X-OR dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, dua buah saklar (A, B) dan LED (Y). Dua buah saklar A dan B disusun silang seperti pada gambar. Kedua saklar tersebut dihubungkan dengan sumber tegangan, dan LED secara paralel.



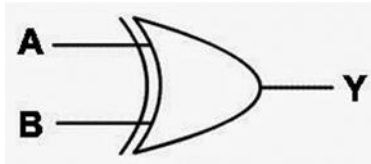
Gambar 2. 28 Rangkaian ekuivalen gerbang X-OR

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika kedua saklar dalam keadaan silang terbuka, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika kedua saklar terhubung pada posisi yang sama, yaitu keduanya pada 1 atau 0, maka LED tidak akan mendapatkan arus listrik searah, sehingga LED menjadi mati, karena arus listrik dihubungkan oleh salah satu saklar (atau kedua saklar) tersebut.

2. Simbol Gerbang X-OR

Secara umum simbol gerbang X-OR dapat dilihat pada gambar dibawah ini. Simbol gerbang X-OR dibawah ini terdiri dari dua buah masukan yaitu A dan B dan sebuah keluaran yaitu Y.



Gambar 2. 29 Simbol gerbang X-OR

3. Tabel Kebenaran Gerbang X-OR

Perhatikan tabel kebenaran dibawah untuk menjelaskan gerbang X-OR dengan dua buah masukan dan satu keluaran. Tabel kebenaran dibuat berdasarkan rangkaian ekivalen diatas.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika masukan atau kedua saklar dalam kondisi silang yaitu saklar A pada posisi terminal 1 dan posisi saklar B pada terminal 0, atau sebaliknya. Dengan demikian **keluaran akan bernilai logika 1 jika kedua saklar bersilang atau berbeda posisi, selain itu keluaran bernilai 0**. Masukan diasumsikan sebagai A dan B, sedangkan keluaran diasumsikan sebagai Y.

	A	B	$Y = A \oplus B$
Genap →	0	0	0
Ganjil →	0	1	1
Genap →	1	0	1
Ganjil →	1	1	0

Gambar 2. 30 Tabel kebenaran X-OR

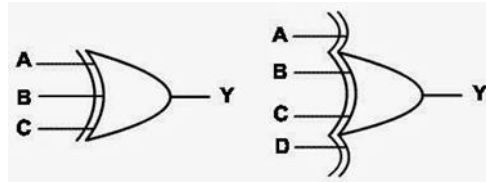
4. Aljabar Boolean Gerbang X-OR

Ekspresi Boolean dari gerbang NOR dapat ditulis dengan tanda tambah dan garis atas () seperti pada gambar dibawah ini. Persamaan Boolean gerbang X-OR dibawah dibaca Y adalah **A X-OR B**.

$$Y = A \oplus B$$

Gambar 2.31 Ekspresi Boolean gerbang X-OR

5. Gerbang X-OR Dengan Masukan Lebih Dari Dua



Gambar 2. 32 Simbol X-OR masukan lebih dari dua

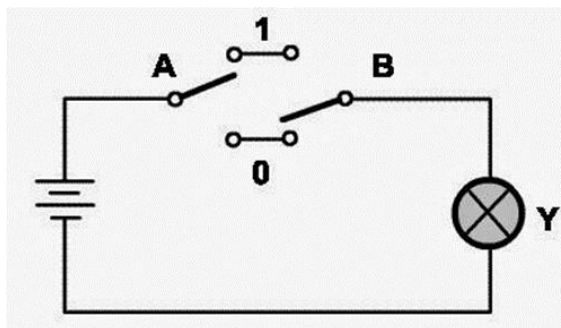
$$Y = A \oplus B \oplus C \oplus \dots$$

Gambar 2. 33 Ekspresi Boolean gerbang X-OR masukan lebih dari dua

G. GERBANG X-NOR

Dasar gerbang X-NOR

Rangkaian dasar gerbang X-OR dapat dilihat pada gambar seperti dibawah ini. Rangkaian terdiri dari sumber tegangan arus searah, dua buah saklar (A, B) dan LED (Y). Dua buah saklar A dan B disusun silang seperti pada gambar. Kedua saklar tersebut dihubungkan dengan sumber tegangan, dan LED secara seri.



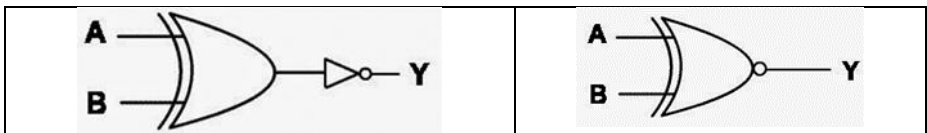
Gambar 2. 34 Rangkaian ekivalen gerbang X-NOR

Prinsip kerja dari rangkaian ekuivalen diatas, adalah jika kedua saklar dalam keadaan silang terhubung, maka LED akan menyala. Kenapa menyala ? LED menyala karena mendapatkan arus listrik searah dari sumber tegangan. LED bekerja dengan sempurna, yaitu sumber arus yang keluar dari sumber tegangan menuju bagian atas LED, dan bagian bawah LED terhubung dengan bagian negatif sumber tegangan.

Pada Rangkaian diatas, jika kedua saklar terhubung pada posisi yang sama, yaitu keduanya pada 1 atau 0, maka LED akan mendapatkan arus listrik searah, sehingga LED menjadi menyala, karena arus listrik dihubungkan singkat oleh salah satu saklar (atau kedua saklar) tersebut

1. Simbol gerbang X-NOR

Secara umum simbol gerbang X-NOR dapat dilihat pada gambar dibawah ini. Simbol gerbang X-NOR dibawah ini terdiri dari dua buah masukan yaitu A dan B dan sebuah keluaran yaitu Y.



Gambar 2. 35 Simbol gerbang X-NOR

2. Tabel kebenaran gerbang X-NOR

Perhatikan tabel kebenaran dibawah untuk menjelaskan gerbang X-OR dengan dua buah masukan dan satu keluaran. Tabel kebenaran dibuat berdasarkan rangkaian ekivalen diatas.

Rangkaian ekivalen diatas dapat kita dianalogikan bahwa saat saklar terhubung, adalah representasi nilai logika tinggi atau bernilai logika 1, sedangkan pada saat saklar terbuka merupakan representasi nilai logika rendah atau nilai logika 0. Pada rangkaian ekivalen diatas, keluaran atau LED akan menyala atau bernilai 1 jika masukan atau kedua saklar dalam kondisi atau pada posisi terminal yang sama yaitu 1 atau 0, atau sebaliknya. Dengan demikian **keluaran akan bernilai logika 1** jika kedua saklar pada posisi yang sama, **selain itu keluaran bernilai 0**. Masukan diasumsikan sebagai A dan B, sedangkan keluaran diasumsikan sebagai Y.

A	B	$Y = A \oplus B$
0	0	1
0	1	0
1	0	0
1	1	1

Masukan identik (sama)

Gambar 2. 36 Tabel kebenaran gerbang X-NOR

3. Aljabar Boolean gerbang X-NOR

Ekspresi Boolean dari gerbang X-NOR dapat ditulis dengan tanda lingkaran tambah dan garis atas () seperti pada gambar dibawah ini.

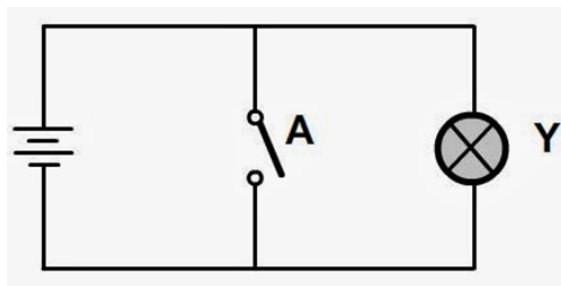
Persamaan Boolean gerbang X-NOR dibawah dibaca Y adalah A X-NOR B.

$$Y = \overline{A \oplus B}$$

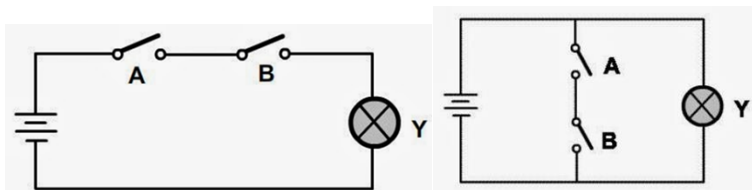
Gambar 2. 37 Eskpresi Boolean gerbang X-NOR

Sumber : [Pengertian Gerbang Logika Dasar beserta Jenis dan simbolnya \(teknikelektronika.com\)](http://teknikelektronika.com)

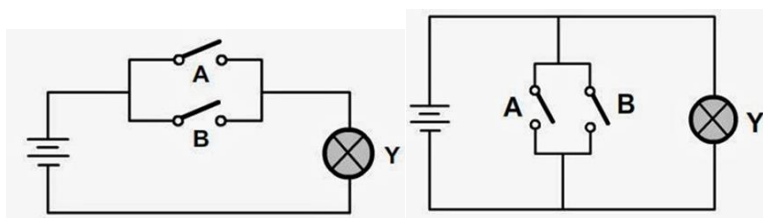
4. Rangkuman dalam gambar rangkaian ekuivalen



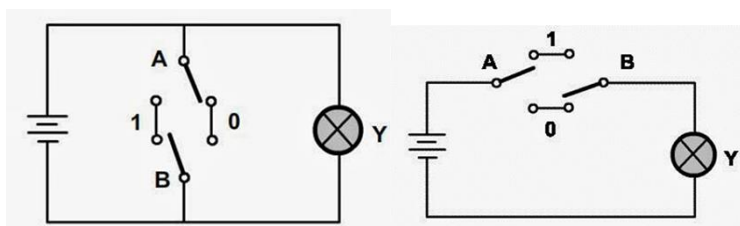
Gerbang NOT



Gerbang AND dan NAND



Gerbang OR dan NOR



Gerbang X-OR dan X-NOR



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 3: SISTEM BILANGAN DIGITAL DAN KONVERSI BILANG

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 3

SISTEM BILANGAN DIGITAL DAN KONVERSI BILANG

A. PENGERTIAN SISTEM BILANGAN

Sistem bilangan merupakan dasar semua bilangan dan merupakan dasar sebagai pembuat bahasa pemrograman tingkat-tingkat rendah, yang biasa kita kenal dengan bahasa mesin atau bahasa assembler. Kita mengenal empat jenis bilangan utama, yaitu bilangan Biner, bilangan Desimal, bilangan Oktal dan bilangan Hexa Desimal.

Dengan memahami ke empat jenis bilangan tersebut, kita dapat melakukan konversi bilangan, dari bilangan satu ke bilangan lainnya. Misalnya konversi dari bilangan biner ke desimal, dari bilangan biner ke Oktal, dari bilangan biner ke hexa desimal. Itu adalah contoh konversi bilangan dari benier ke bilangan yang lainnya. Demikian bilangan Oktal, Desimal, hexadesimal bisa dilakukan konversi bilangan antara satu dan yang lainnya.

Dengan mengetahui sistem bilangan akan memudahkan kita dalam belajar di teknik informatika. Terutama sistem digital, dan aplikasinya.

B. BINER

Sistem bilangan biner terdiri dari dua yaitu angka 0 dan 1. Angka 0 dan 1 atau biner tersebut merupakan dasar untuk semua bilangan. Dengan adanya biner bisa membuat bilangan digital dari 0 sampai dengan 9. Bilangan biner kita kenal dengan nama bilangan berbasis 2. Contoh Bilangan biner adalah 1100112, 010102 dan seterusnya.

C. OKTAL

Sistem bilangan oktal terdiri dari delapan angka yaitu angka 0, 1, 2, 3, 4, 5, 6, dan 7. Bilangan oktal tidak memiliki angka 8. Bilangan Oktal tersebut merupakan bilangan yang lebih lengkap dibanding bilangan biner. Dengan adanya bilangan oktal lebih memudahkan lagi dalam mengkonversi suatu nilai bilangan. Bilangan oktal kita kenal dengan nama bilangan berbasis 8. Contoh Bilangan biner adalah 12345678, 772198, 2013768 dan seterusnya.

D. DESIMAL

Bilangan desimal adalah bilangan yang biasa kita gunakan sehari-hari. Bilangan desimal memiliki angka sebagai berikut 0,1, 2, 3, 4, 5, 6, 7, 8, dan 9. Bilangan Desimal merupakan bilangan yang paling mudah digunakan, karena tidak berbasis. Namun bilangan desimal dapat mempermudah konversi bilangan.

E. HEXADESIMAL

Bilangan hexadesimal adalah bilangan yang terdiri dari 16 simbol bilangan yaitu 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,dan F.

F. KONVERSI BILANGAN

Konversi Bilangan digunakan untuk mengubah suatu bilangan dari suatu sistem bilangan menjadi bilangan dalam sistem bilangan yang lain.

1. Biner ke Desimal

Contoh :

$$1100012 = \dots? \dots_{10}$$

$$\begin{aligned} 1100012 &= (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 32 + 16 + 0 + 0 + 0 + 1 \\ &= 49 \end{aligned}$$

$$\text{Jadi, } 110012 = 49$$

$$010111012 = \dots? \dots_{10}$$

$$\begin{aligned} 01011101 &= (0 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) \\ &\quad + (0 \times 2^1) + (1 \times 2^0) \end{aligned}$$

$$= (0 \times 128) + (1 \times 64) + (0 \times 32) + (1 \times 16) + (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1)$$

$$= + 64 + 0 + 16 + 8 + 4 + 0 + 1$$

Jawaban yang benar ialah 93

2. Biner ke Oktal

Cara mengubah bilangan Biner menjadi bilangan Oktal dengan mengambil 3 digit bilangan dari kanan.

Contoh : 111100110012 dapat dikonversi ke bilangan oktal, dilakukan dengan mengubah menjadi sistem bilangan biner terlebih dahulu menjadi basis delapan, atau Oktal dengan cara sebagai berikut.

$$\underline{11\ 110\ 011\ 001}_2 = \dots? \dots_8$$

$$\text{➤ } 11_2 = 2^1 + 2^0 = 3_8$$

$$\text{➤ } 110_2 = 2^2 + 2^1 = 6_8$$

$$\text{➤ } 011_2 = 2^1 + 2^0 = 3_8$$

$$\text{➤ } 001_2 = 2^0 = 1_8$$

$$\text{Jadi, } 11110011001_2 = 3631_8$$

$$1001110_2 = \dots(8)$$

kalaupun susah rubah dulu saja ke (10) sehingga menjadi 78 yang diperoleh dari:

$$0 \times 2^0 = 0 \text{ (lihat dari belakang terus kedepan)}$$

$$1 \times 2^1 = 2$$

$$1 \times 2^2 = 4$$

$$1 \times 2^3 = 8$$

$$0 \times 2^4 = 0$$

$$0 \times 2^5 = 0$$

$$1 \times 2^6 = 64$$

$$\text{sehingga ke decimal} = 64 + 0 + 0 + 8 + 4 + 2 + 0 = 78$$

$$78_{10} = \dots_8$$

$$78/8 = 9 \text{ sisa } 6$$

$$9/8 = 1 \text{ sisa } 1$$

$$\text{Jadi hasilnya } 78_{10} = 116_8, \text{ atau } 1001110_2 = 116_8$$

3. Biner ke HexaDesimal

Cara mengubah Biner menjadi bilangan HexaDesimal dengan mengambil 4 digit bilangan dari kanan .

Contoh: 0100111101011100₂ dapat dikonversi ke bilangan hexadesimal, dilakukan dengan mengubah menjadi sistem bilangan biner terlebih dahulu menjadi basis enambelas, atau hexadesimal dengan cara sebagai berikut.

$$\underline{0100\ 1111\ 0101\ 1100} = \dots? \dots_{16}$$

- $0100_2 = 2^2 = 4_{16}$
- $1111_2 = 2^3 + 2^2 + 2^1 + 2^0 = 15 = F_{16}$
- $0101_2 = 2^2 + 2^0 = 5_{16}$
- $1100_2 = 2^3 + 2^2 = 12 = C_{16}$

Jadi, **0100111101011100₂ = 4F5C₁₆**

4. Oktal ke Biner

Cara mengubah bilangan Oktal menjadi Biner dengan menjadikan satu persatu angka bilangan Oktal menjadi bilangan Biner dahulu kemudian di satukan. Untuk bilangan Oktal haruslah memiliki 3 digit bilangan Biner sehingga jika hanya menghasilkan kurang dari 3 digit maka didepannya ditambahkan bilangan 0.

Contoh :

$$261_8 = \dots? \dots_2$$

$$= 2_8 = 010_2$$

$$= 6_8 = 110_2$$

$$= 1_8 = 001_2$$

$$\text{Jadi, } 261_8 = 010110001_2$$

5. Oktal ke Desimal

Cara mengubah bilangan Oktal menjadi bilangan Desimal dengan mengubah bilangan Oktal tersebut menjadi bilangan Biner terlebih dahulu baru kita ubah menjadi bilangan Desimal.

Contoh :

$$261_8 = \dots? \dots_{10}$$

Langkah 1 : mengubah ke bilangan Biner

261

➤ $2_8 = 010_2$

➤ $6_8 = 110_2$

➤ $1_8 = 001_2$

Jadi, $261_8 = 010110001_2$

Langkah 2 : mengubah bilangan Biner menjadi Desimal

$$010110001_2 = (0 \times 2^8) + (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$= 0 + 128 + 0 + 32 + 16 + 0 + 0 + 0 + 1$$

$$= 177$$

Jadi, $261_8 = 177$

$$324(8) = \dots (10)$$

Caranya:

$$324(8) = 3 \times 8^2 + 2 \times 8^1 + 4 \times 8^0$$

$$= (3 \times 64) + (2 \times 8) + (4 \times 1)$$

$$= 192 + 16 + 4$$

$$= 212(10)$$

Jadi, $324(8) = 212(10)$

6. Oktal ke Hexadesimal

Cara mengubah bilangan **Oktal menjadi bilangan HexaDesimal** dengan mengubah bilangan Oktal tersebut menjadi bilangan Biner terlebih dahulu baru kita ubah menjadi bilangan Desimal. Lalu kita ubah lagi menjadi bilangan HexaDesimal.

Contoh : $261_8 = \dots? \dots_{16}$

Langkah 1 : mengubah ke bilangan Biner

261

➤ $2_8 = 010_2$

➤ $6_8 = 110_2$

➤ $1_8 = 001_2$

Jadi, $261_8 = 010110001_2$

Langkah 2 : mengubah bilangan Biner menjadi Desimal

$$010110001_2 = (0 \times 2^8) + (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 0 + 128 + 0 + 32 + 16 + 0 + 0 + 0 + 1 = 177$$

Langkah 3 : mengubah bilangan Desimal menjadi HexaDesimal

177 kita bagi dengan 16 - $177:16 = 11$ sisa 1

$11/16 = 0$ sisa 11 - B

dibaca dari bawah maka menjadi B1

Jadi **$261_8 = B1_{16}$**

$46_8 = \dots ? \dots_{16}$

Diubah ke Biner

$4 = 100$

$6 = 110$

Diubah Ke Hexadecimal

$0010 = 2$

$0110 = 6$

Hasil Konversi = 26

7. Desimal ke Biner

Contoh :

$25 = \dots ? \dots_2$

$25/2 = 12$ sisa 1

$12/2 = 6$ sisa 0

$6/2 = 3$ sisa 0

$3/2 = 1$ sisa 1

$1/2 = 0$ sisa 1

maka ditulis 11001

Jadi **$25 = 11001_2$**

Contoh $88_{(10)} = \dots ? \dots_{(2)}$

$88/2 = 44$ sisa 0

$$44/2 = 22 \text{ sisa } 0$$

$$22/2 = 11 \text{ sisa } 0$$

$$11/2 = 5 \text{ sisa } 1$$

$$5/2 = 2 \text{ sisa } 1$$

$$2/2 = 1 \text{ sisa } 0$$

Hasil Konversi bolangan disimal ke biner adalah : 1011000_2

8. Desimal ke Oktal

Contoh :

$$80 = \dots ? \dots_8$$

$$80/8 = 10 \text{ sisa } 0$$

$$10/8 = 1 \text{ sisa } 2$$

$$1/8 = 0 \text{ sisa } 1$$

maka ditulis 120

$$\text{Jadi } 80 = 120_8$$

9. Desimal ke Hexadesimal

$$275 = \dots ? \dots_{16}$$

$$275/16 = 17 \text{ sisa } 3$$

$$17/16 = 1 \text{ sisa } 1$$

$$1/16 = 0 \text{ sisa } 1$$

maka ditulis 113

$$\text{Jadi } 275 = 113_{16}$$

$$130/16=8 \text{ sisa } 2$$

$$8/16 = 0 \text{ sisa } 8$$

$$\text{Jadi } 130_{10} = 82_{16}$$

$$1583_{10} = \dots ? \dots_{16}$$

Caranya:

$$1583/16 = 98 \text{ sisa } 15 = F$$

$$98/16 = 6 \text{ sisa } 2$$

$$\text{Sehingga } 1583_{10} = 62F_8$$

10. Hexadesimal ke Biner

Contoh :

$$4DA2_{16} = \dots? \dots_2$$

$$4DA2 = 4_{16} = 0100_2$$

$$= D_{16} = 1101_2$$

$$= A_{16} = 1010_2$$

$$= 2_{16} = 0010_2$$

$$\text{Jadi } 4DA2_{16} = 0100110110100010_2$$

11. Hexadesimal ke Desimal

Contoh : $3C2_{16}$ diubah menjadi bilangan Desimal

$$3C2_{16} = (3 \times 16^2) + (C(12) \times 16^1) + (2 \times 16^0)$$

$$= 768 + 192 + 2$$

$$= 962$$

$$\text{Jadi } 3C2_{16} = 962$$

$$B6A(16) = \dots (10)$$

Caranya:

$$B6A = (11 \times 16^2) + (6 \times 16^1) + (10 \times 16^0)$$

$$= (11 \times 256) + (6 \times 16) + (10 \times 1)$$

$$= 2816 + 96 + 10$$

$$= 2922$$

$$\text{Sehingga } B6A(16) = 2922(10)$$

12. Hexadesimal ke Oktal

$$3C2_{16} = \dots\dots_8$$

Langkah 1: Mengubah bilangan HexaDesimal menjadi Desimal

$$3C2_{16} = (3 \times 16^2) + (C(12) \times 16^1) + (2 \times 16^0)$$

$$= 768 + 192 + 2$$

$$= 962$$

Langkah 2 : Mengubah bilangan Desimal menjadi Oktal

$$962/8 = 120 \text{ sisa } 2$$

$$120/8 = 15 \text{ sisa } 0$$

$$15/8 = 1 \text{ sisa } 7$$

$$1/8 = 0 \text{ sisa } 1$$

maka ditulis 1702

Jadi $3C2_{16} = 1702_8$

<https://tutorial-nanang.blogspot.com/2016/02/sistem-bilangan-digital-dan-konversi.html>

<https://quipper.co.id/sistem-bilangan/>



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 4: MULTIPLEXER

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 4

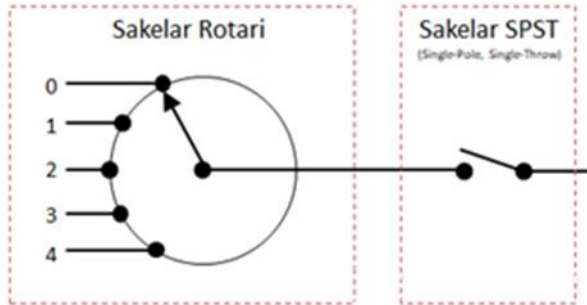
MULTIPLEXER

A. PENGERTIAN MULTIPLEXER

Multiplexer (MUX) adalah suatu rangkaian digital yang terdiri dari beberapa masukan signal digital dan diproses oleh komponen *processing* (selektor) untuk menghasilkan satu keluaran. Rangkaian multiplexer merupakan rangkaian yang terdiri dari beberapa komponen digital pada bagian masukan. Rangkaian multiplexer bisa terdiri dari beberapa gerbang NOT, beberapa gerbang AND dan sebuah gerbang OR. Multiplexer merupakan sebuah rangkaian terintegrasi (*Integrated Circuit*), atau sebuah modul, atau secara umum dikenal dengan istilah IC multiplexer.

Pada dasarnya analogi multiplexer sama dengan kumpulan saklar, dimana pada bagian masukan terdapat banyak saklar yang bisa kita pilih (seleksi) satu yang sesuai, sedangkan pada bagian keluaran hanya terdapat satu saklar. Bagian masukan kita gunakan saklar atau yang biasa dikenal dengan nama saklar putar atau saklar rotari. Bagian keluaran menggunakan saklar tunggal atau yang biasa dikenal dengan nama saklar SPST atau single pole single throw. Bentuk multiplexer dengan analogi saklar tersebut dapat dilihat pada gambar dibawah ini.

Pada saklar terdapat pemilih atau selektor. Selektor inilah yang berperan untuk menghasilkan aljabar Boolean.

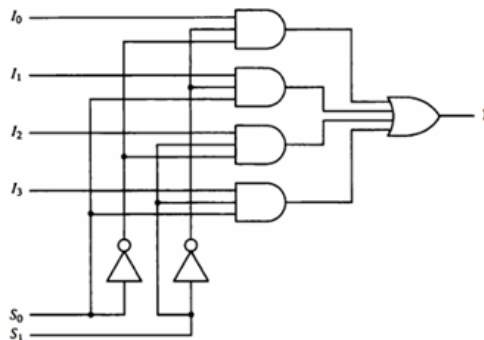


Gambar 4. 1 Analogi multiplexer menggunakan saklar

Proses yang dilakukan oleh multiplexer disebut dengan multiplexing. Dalam proses ini semua signal digital yang masuk diproses dan dipilih menjadi sebuah signal keluaran. Dilihat dari cara kerja multiplexer tersebut, lebih mudah kita memahami, bahwa multiplexer adalah jumlah signal masukan lebih dari satu dan signal keluaran menjadi satu signal. Saluran yang masuk pada multiplexses dapat pula berupa transmisi data.

Analogi multiplexer diatas dapat pula digambar dengan bentuk rangkaian digital. Bentuk rangkaian digital multiplexer dapat dilihat pada gambar dibawah ini. Dimana multiplexer terdiri dari gerbang NOT, AND dan OR.

salah satu contoh gambar rangkaian multiplexer dapat dilihat seperti pada gambar dibawah ini.



Gambar 4. 2 Multiplexer dengan kombinasi gerbang logika

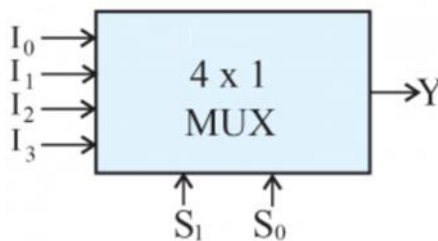
B. FUNGSI MULTIPLEKSER

Multiplexer merupakan penghubung atau interface dari beberapa masukan menjadi satu keluaran. Masukan multiplexer yang paling cocok adalah signal digital, karena komponen utamanya merupakan gerbang gerbang digital. Apakah masukan signal analog dapat di proses melalui multiplexer ? Signal analog yang mau diproses oleh multiplexer harus dilakukan perubahan terlebih dahulu dengan menggunakan Analog to Digital Converter (ADC).

Manfaat multiplexer selain memproses beberapa signal, menjadi satu, juga dapat meningkatkan efisiensi penggunaan data. Proses multiplexer atau Multiplexing sangat banyak ragamnya, diantaranya adalah : *Frequency Division Multiplexing (FDM)*, *Time Division Multiplexing (TDM)*, *Code Division Multiplexing (CDM)*.

C. TABEL KEBENARAN MULTIPLEKSER

Sebuah multiplexer dengan istilah **4X1 multiplexer**, artinya terdapat 4 signal data masukan dan terdapat 1 signal data keluaran. Pada gambar dibawah ini adalah sebuah contoh blok dari dari **4X1 multiplexer**. Dengan masukan 4 buah yaitu I_0 , I_1 , I_2 , I_3 , dan 1 keluaran yaitu Y . Kita perlu memperhatikan **nilai n pada 2^n** . Jika terdapat 4 signal data berapa nilai n-nya ?. Nilai n merupakan nilai untuk menentukan **jumlah selektor** pada bagian masukan multiplexer. Dengan demikian 4 signal masukan sama dengan $2^n=4$. Dalam hal ini nilai n adalah 2. Dengan demikian pada gambar tersebut dibawah ini terdapat 4 masukan signal dan 2 selektor serta satu keluar Y . Nilai n ini yang harus bisa kita tentukan, sehingga kita bisa membuat persamaan aljabar Boolean multiplexer dengan benar.



Gambar 4. 3 Blok 4X1 Multiplexer

Dalam menentukan tabel kebenaran 4X1 multiplexer, kita dapat menentukan jumlah kombinasi data. Dalam tabel kebenaran disini terdapat 2 selektor yaitu S_0 dan S_1 , sehingga jumlah kombinasinya sebanyak $2^n = 2^2 = 4$ kombinasi. Perhatikan tabel kebenaran 4X1 multiplexer berikut ini, dengan masukan S_1 dan S_0 , dan keluaran Y .

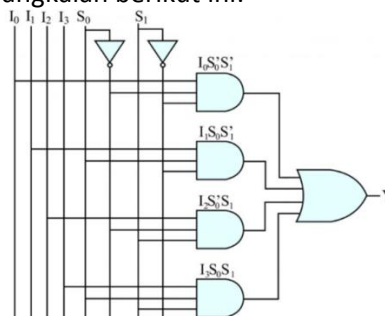
Garis seleksi		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Gambar 4. 4 Tabel kebenaran 4X1 Multiplexe

Dari tabel kebenaran di atas, kita dapat menuliskan fungsi Booleannya untuk *output* Y . Dipilih bentuk perkalian jumlah (ekspansi sukumin), maka yang bernilai 0 akan dikomplemenkan. Untuk kombinasi pertama, S_0 dan S_1 bernilai 0 maka bagian pertama *outputnya* adalah $I_0S_0'S_1'$, begitu seterusnya sehingga persamaan Booleannya menjadi:

$$Y = I_0S_0'S_1' + I_1S_0S_1' + I_2S_0'S_1 + I_3S_0S_1$$

Dari persamaan di atas, selanjutnya kita gambar rangkaian gerbang logikanya. Perhatikan rangkaian berikut ini:



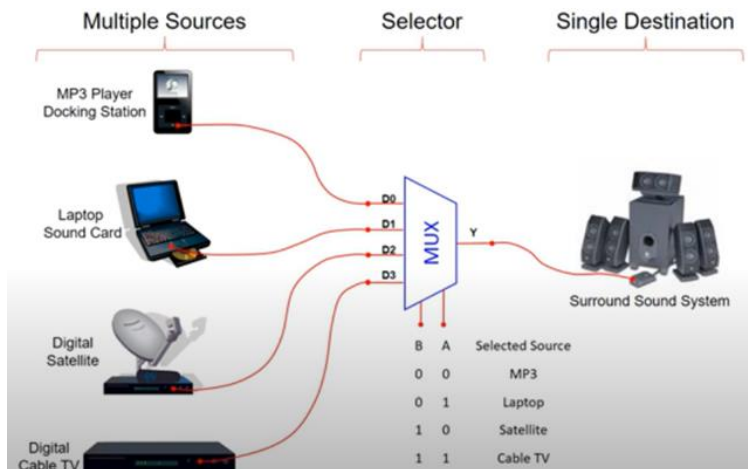
Gambar 4. 5 Rangkaian Logika

Sumber : <http://pauzan.com/multiplexer/>

D. ALJABAR BOOLEAN MULTIPLEKSER

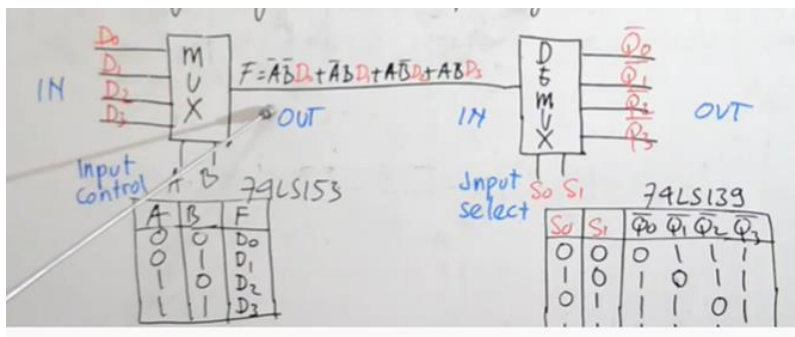
$$Y = I_0 S_0' S_1' + I_1 S_0 S_1' + I_2 S_0' S_1 + I_3 S_0 S_1$$

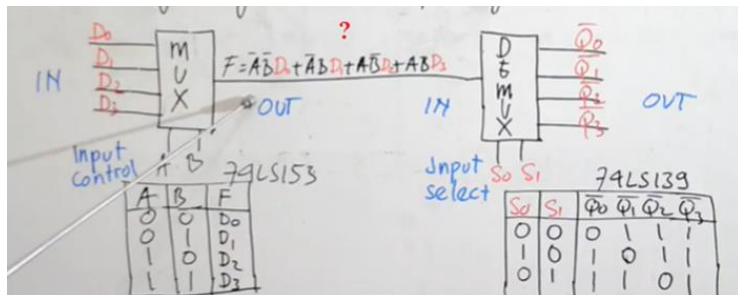
E. PENGGUNAAN MULTIPLEKSER



Gambar 4. 6 Aplikasi dari multiplexer

Soal





Sumber <https://www.youtube.com/watch?v=VZ-IQ7XHQSI>
 Perbedaan Multiplexer dan Demultiplexer ~ Berbagi Ilmu Informatika
 (berbagiilmuprogram.blogspot.com)



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 5: PENGERTIAN *DEMULTIPLEXER*

Sri Mulyati

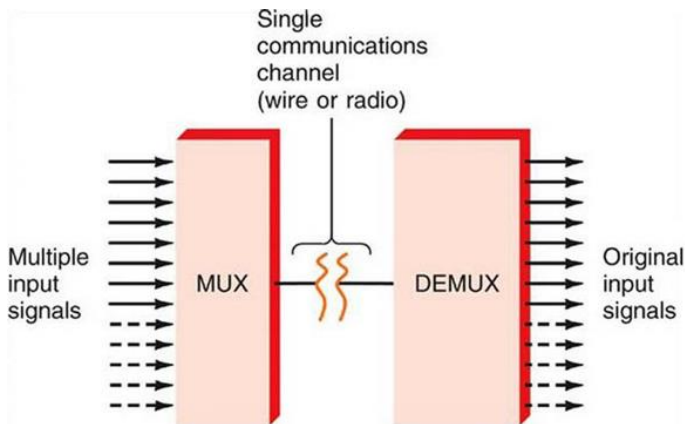
Universitas Muhammdiyah Tangerang

BAB 5

PENGERTIAN *DEMULTIPLEXER*

Fungsi *demultiplexer*

Perangkat elektronik yang fungsi untuk memilih salah satu yang ada pada data yang diberikan pada data yang menggunakan bentuk dari suatu masukan. *Demultiplexer* memang banyak disebut dengan perangkat yang sedikit masukan dan banyak nya keluaran, karena yang berfungsi memilih saluran keluaran yang banyak di jalur masukan yang sedikit . Contoh yang diaplikasikan di digital (TTL) yang berada di IC yang fungsinya sebagai *demultiplexer* seperti **IC74LS138** yang berbentuk *demultiplexer* 8 arah. *Demultiplexer* 74LS138 yang banyak



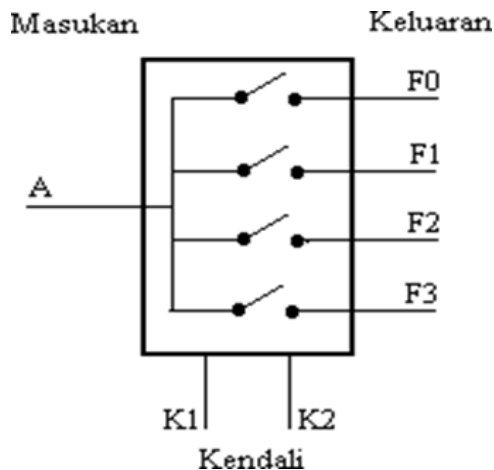
Gambar 5. 1 *Demultiplexer* 8 arah

<https://anotherorion.com/pengertian-multiplexer-demultiplexer-decoder-dan-encoder/>

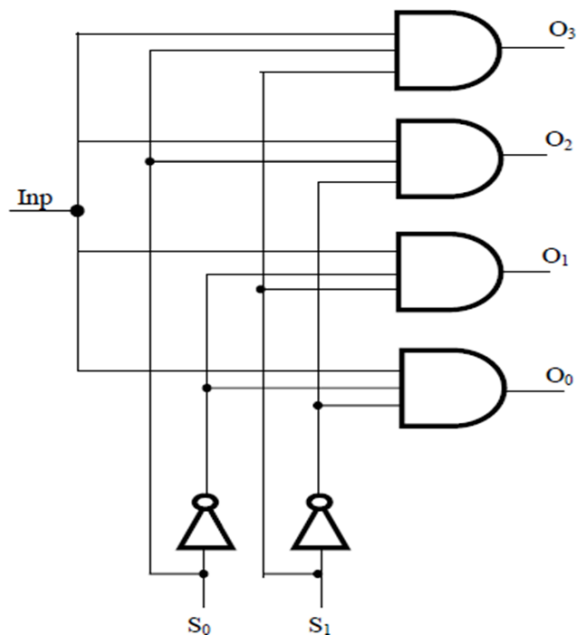
Demultiplexer dan Multiplexer Oleh :

DEMULTIPLEXER

Sebuah *Demultiplexer* adalah rangkaian logika yang menerima satu masukan data dan mendistribusikan masukan tersebut ke beberapa keluaran yang tersedia. Kendali pada demultiplexer akan memilih saklar mana yang akan dihubungkan. Pemilihan keluarannya dilakukan melalui masukan penyeleksi. Seleksi data-data masukan dilakukan oleh selector line, yang juga merupakan masukan dari demultiplexer tersebut. Pada demultiplexer saluran kendali sebanyak "n" saluran dapat menyeleksi 2^n saluran keluaran. Secara bagan, kerja demultiplexer dapat digambarkan sebagai berikut :



Gambar 5. 2 Rangkaian dasar demultiplexer



Gambar 5. 3 Rangkaian Logika *Demultiplexer*

Contoh soal :

Sebuah DE-MUX 4 kanal di realisasikan pada rangkaian berikut :

tentukan persamaan keluaran (F) dari rangkaian tersebut !

Penyelesaian :

Tabel kebenaran :

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0

0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Gambar 5. 4 Tabel Kebenaran

Dari tabel kebenaran diatas dapat dibuatkan persamaan logika berikut:
 $F = ABC \quad (ABC \text{ negasi})$

Buatlah rangkaian logika yang mempunyai persamaan Boole sebagai
 $Y(A, B, C) = \sum(2, 4, 5, 7)$

serta tuliskan tabel kebenarannya?

Jawab :

Sesuai persamaan yang diberikan, maka tabel kebenaran rangkaian tersebut dan aljabar boole rangkaian tersebut adalah:

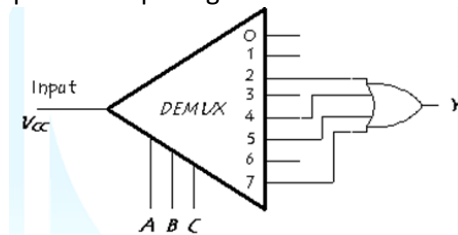
Tabel Kebenaran

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

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0

1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

sedangkan untuk rangkaian *demultiplexer* yang menggunakan persamaan diatas dapat dilihat pada gambar 4.



Gambar 5. 5 Implementasi persamaan $Y(A, B, C) = \sum(2, 4, 5, 7)$

Mengapa suatu *demultiplexer* disebut juga dengan penyalur data ?

Jawab :

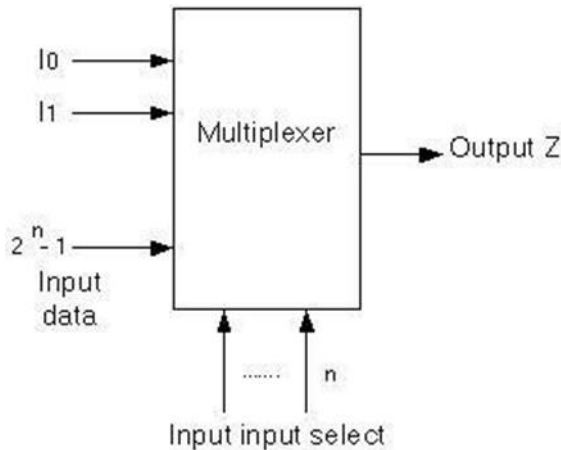
Karena *Demultiplexer* merupakan rangkaian logika yang berfungsi menyalurkan data yang ada pada masukannya ke salah satu dari beberapa keluarannya dengan bantuan sinyal pemilih atau sinyal kontrol

MULTIPLEXER

Tabel 2. Tabel Kebenaran Multiplexer

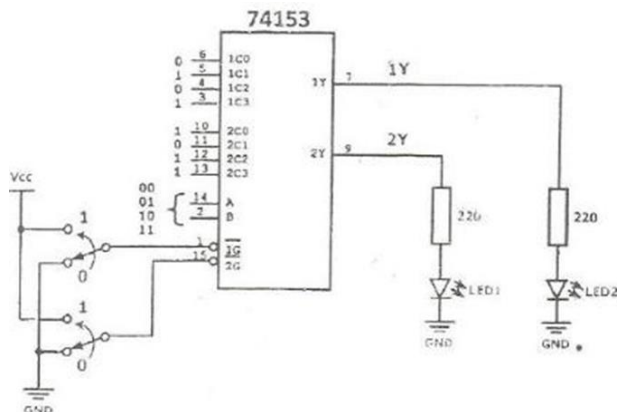
Gambar 5. Rangkaian Multiplexer.

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



Gambar 5. 6 Diagram Dasar Multiplexer

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

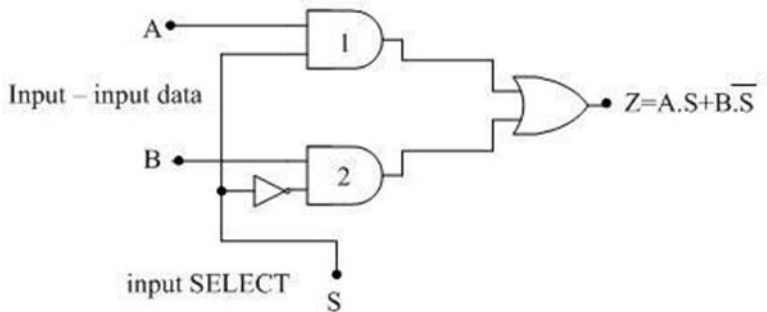


Gambar 5. 7 Rangkaian IC TTL 74153 Multiplexer 2 keluaran

Contoh soal:

Gambar 6 menunjukkan rangkaian logika untuk multiplexer dua masukan (dua chanel) dengan masukan masukan data A dan B dan masukan SELECT. Level logika yang diberikan ke masukan S menentukan AND gate mana yang di –enable sehingga masukan datanya lewat melalui OR gate ke keluaran Z. Tuliskan ekspresi boolean untuk keluaran tersebut beserta rangkaian logikanya.

Jawab : persamaan boolean untuk keluaran tersebut $Z = A.S + B.S'$ sedangkan untuk rangkaian logikanya dapat di tunjukkan pada Gambar 8.



Gambar 5. 8 Multiplexer 2 Masukan

Dari tabel kebenaran tunjukkan bahwa $Z=B$. Tabel Kebenaran Multiplexer 2 Masukan

Input select (S)	Output
0	$Z = B$
1	$Z = A$

Dengan $S = 0$, ekspresi ini menjadi:

$$Z = A.0 + B.1$$

$$= B$$

Tunjukkan bahwa Z akan identik dengan sinyal masukan B yang merupakan suatu level logika tetap atau sinyal logika yang berubah – ubah menurut waktu. Dimana $S = 1$

jawab :

$$Z = A.1 + B.0$$

$$= A$$

Sehingga menunjukkan bahwa keluaran Z akan identik dengan sinyal masukan A. Rangkaian rangkaian multiplexer sangat banyak di temukan dalam berbagai macam pemakaian dalam sistem-sistem digital. Pemakaian pemakaian ini termasuk seleksi data, data routing, pengurutan operasi (*operation sequencing*), konversi paralel ke seri dan lain sebagainya.

<http://staff.unila.ac.id/junaidi/files/2013/06/MULTIFLEXER-DAN-DEMULTIFLEXER.pdf>

A. DEMULTIPLEXER 1 KE 2

Demultiplexer 1-ke-2 terdiri dari satu saluran masukan, dua saluran keluaran, dan satu saluran pilih. Sinyal pada jalur pilih membantu mengalihkan masukan ke salah satu dari dua keluaran. Gambar di bawah menunjukkan diagram blok *demultiplexer* 1-ke-2 dengan tambahan masukan masukan.

Pada gambar, hanya ada dua cara yang mungkin untuk menghubungkan masukan ke saluran keluaran, sehingga hanya satu sinyal pilihan yang cukup untuk melakukan operasi demultiplexing. Bila masukan yang dipilih LOW, maka masukan tersebut akan diteruskan ke Y0 dan jika masukan yang dipilih HIGH, maka masukan tersebut akan diteruskan ke Y1.

Tabel kebenaran *demultiplexer* 1-ke-2 ditunjukkan di bawah ini, dimana masukan dirutekan ke Y0 dan Y1 tergantung pada nilai masukan S yang dipilih.

S	D	Y1	Y0
0	0	0	0

0	1	0	1
1	0	0	0
1	1	1	0

Kita dapat menurunkan Ekspresi Boolean untuk keluaran sebagai berikut:

Asumsikan S adalah Pilih yang Masuk, D adalah Data yang Masuk dan Y0 dan Y1 adalah keluaran dari *Demultiplexer* 1-ke-2. Dari tabel di atas, keluaran Y0 aktif ketika kombinasi saluran pilih dan saluran masukan masing-masing aktif rendah dan tinggi, yaitu,

$$S D = 0 \ 1 .$$

Oleh karena itu, ekspresi untuk keluaran Y0 adalah

$$Y0 = S D$$

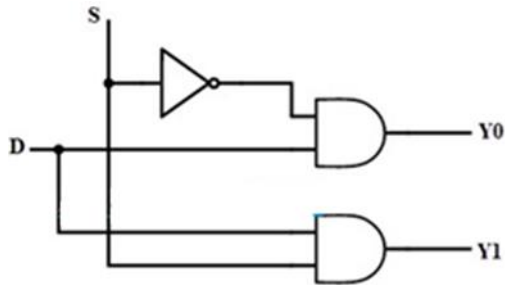
Demikian pula, keluaran Y1 aktif ketika kombinasi saluran pilih dan saluran masukan aktif tinggi, yaitu $S D = 1 \ 1$.

Oleh karena itu, ekspresi untuk keluaran Y0 adalah

$$Y1 = S D$$

Dari tabel kebenaran di atas dan Ekspresi Boolean yang diturunkan, diagram logika *demultiplexer* 1-ke-2 dapat dirancang menggunakan dua gerbang AND dan satu gerbang NOT seperti yang ditunjukkan pada gambar di bawah ini. Ketika jalur pilih $S = 0$, gerbang AND pertama (A1) diaktifkan, sedangkan gerbang AND kedua (A2) dinonaktifkan.

Kemudian, data dari masukan tersebut mengalir ke jalur keluaran Y0. Demikian pula, ketika $S = 1$, gerbang AND kedua (A2) diaktifkan dan gerbang AND pertama (A1) dinonaktifkan, dengan demikian data diteruskan ke keluaran Y1.

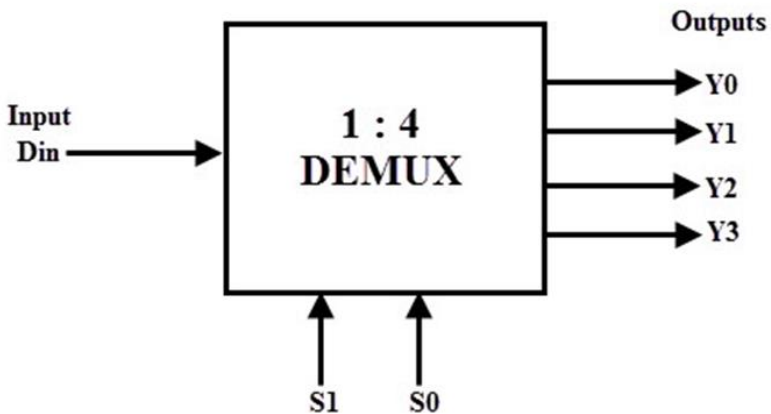


Gambar 5. 9 Demultiplexer 1 ke 2

B. 1-TO-4 DEMULTIPLEXER

A 1-to-4 demultiplexer has a single masukan (D), two selection lines (S1 and S0) and four keluarans (Y0 to Y3). The masukan data goes to any one of the four keluarans at a given time for a particular combination of select lines.

This demultiplexer is also called as a 2-to-4 Demultiplexer, which means that it has two select lines and 4 keluaran lines. The block diagram of a 1:4 DEMUX is shown below.



Gambar 5. 10 1-to-4 Demultiplexer

The truth table of this type of demultiplexer is given below. From the truth table it is clear that, when $S_0 = 0$ and $S_1 = 0$, the data masukan is connected to keluaran Y_0 and when $S_0 = 0$ and $S_1 = 1$, the data masukan is connected to keluaran Y_1 .

Similarly, other keluarans are connected to the masukan for the other two combinations of select lines.

S1	S0	D	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

From the above truth table, we can derive the Boolean Expressions for the keluarans as follows:

$$Y_0 = S_1 S_0 D$$

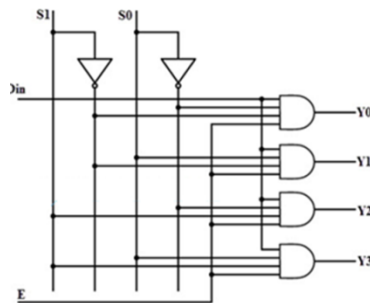
$$Y_1 = S_1 S_0 D$$

$$Y_2 = S_1 S_0 D$$

$$Y_3 = S_1 S_0 D$$

Dimana D adalah data masukan, Y0 sampai Y3 adalah jalur keluaran dan S0 & S1 adalah jalur pilihan. Dari ekspresi Boolean di atas, *demultiplexer* 1-ke-4 dapat diimplementasikan dengan menggunakan empat gerbang AND 3-masukan dan dua gerbang NOT seperti yang ditunjukkan pada gambar di bawah. Dua garis pilih mengaktifkan gerbang AND tertentu pada suatu waktu. Selain itu, terdapat Enable / Strobe Masukan yang berfungsi sebagai global enable masukan yaitu keluaran hanya aktif saat bit 'E' TINGGI.

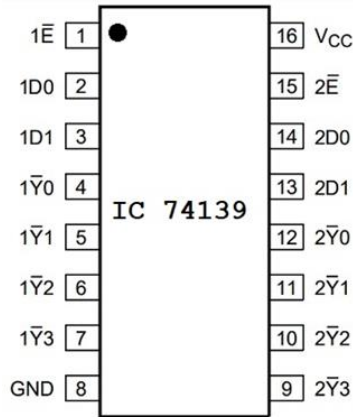
Jadi tergantung pada kombinasi masukan tertentu, masukan data dilewatkan melalui gerbang terpilih ke keluaran terkait.



Gambar 5. 11 rangkaian Logika

Jenis *demultiplexer* ini tersedia dalam bentuk sirkuit terpadu sebagai IC 74139. Ini adalah salah satu IC *demultiplexer* yang paling umum digunakan dan merupakan IC *demultiplexer* ganda 1-ke-4 yaitu, berisi dua blok *demultiplexer* 1-ke-4 independen dalam satu IC. Setiap DEMUX menerima dua masukan biner sebagai jalur terpilih dan empat keluaran aktif-rendah yang saling eksklusif.

Kedua *demultiplexer* memiliki set baris pilih individu sehingga mereka dapat bertindak sebagai Demux yang benar-benar independen. Selain itu, setiap *demultiplexer* terdiri dari pin enable khusus, yang dapat bertindak sebagai masukan data untuk operasi *demultiplexer*. Pin Aktifkan aktif RENDAH.



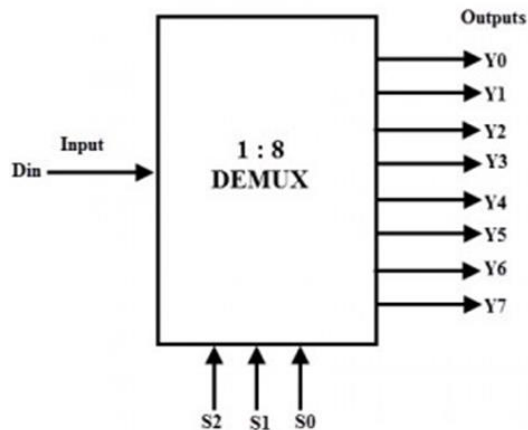
Gambar 5. 12 IC 74139

The keluarans are active LOW i.e., they are HIGH by default. So, if the Enable pin is HIGH, all the keluarans are HIGH and if Enable is LOW, then based on the Select Pins, only the corresponding keluaran pin becomes LOW.

C. DEMULTIPLEXER 1 KE 8

Gambar di bawah menunjukkan diagram blok *demultiplexer* 1-ke-8 yang terdiri dari satu masukan D, tiga masukan pilihan S2, S1 dan S0 dan delapan keluaran dari Y0 ke Y7.

Disebut juga sebagai 3-to-8 *demultiplexer* karena memiliki tiga saluran masukan dan 8 saluran keluaran pilihan. Ini mendistribusikan satu saluran masukan ke salah satu dari 8 saluran keluaran tergantung pada kombinasi masukan tertentu.



Gambar 5. 13 blok *demultiplexer* 1-ke-8

Tabel kebenaran untuk *demultiplexer* 1-ke-8 ditunjukkan di bawah ini. Masukan 'D' dihubungkan dengan salah satu dari delapan keluaran dari Y0 hingga Y7 berdasarkan jalur pilihan S2, S1 dan S0.

Misalnya, jika S2 S1 S0 = 0 0 0, maka masukan D dihubungkan dengan keluaran Y0 dan seterusnya.

S2	S1	S0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	D
0	0	1	0	0	0	0	0	0	D	0
0	1	0	0	0	0	0	0	D	0	0
0	1	1	0	0	0	0	D	0	0	0
1	0	0	0	0	0	D	0	0	0	0
1	0	1	0	0	D	0	0	0	0	0
1	1	0	0	D	0	0	0	0	0	0
1	1	1	D	0	0	0	0	0	0	0

Dari tabel kebenaran di atas, ekspresi Boolean semua keluaran dapat ditulis sebagai berikut.

$$Y_0 = S_2 S_1 S_0 D$$

$$Y_1 = S_2 S_1 S_0 D$$

$$Y_2 = S_2 S_1 S_0 D$$

$$Y_3 = S_2 S_1 S_0 D$$

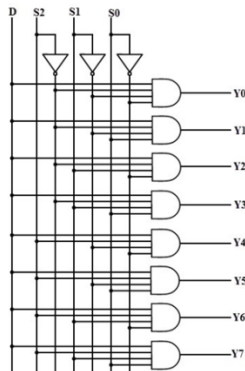
$$Y_4 = S_2 S_1 S_0 D$$

$$Y_5 = S_2 S_1 S_0 D$$

$$Y_6 = S_2 S_1 S_0 D$$

$$Y_7 = S_2 S_1 S_0 D$$

Dari persamaan yang diperoleh, diagram logika *demultiplexer* ini dapat diimplementasikan dengan menggunakan delapan gerbang AND 4-masukan dan tiga gerbang NOT seperti yang ditunjukkan pada gambar di bawah ini. Kombinasi yang berbeda dari jalur pilih mengaktifkan satu gerbang AND pada waktu tertentu, sehingga masukan data akan muncul pada keluaran yang sesuai.

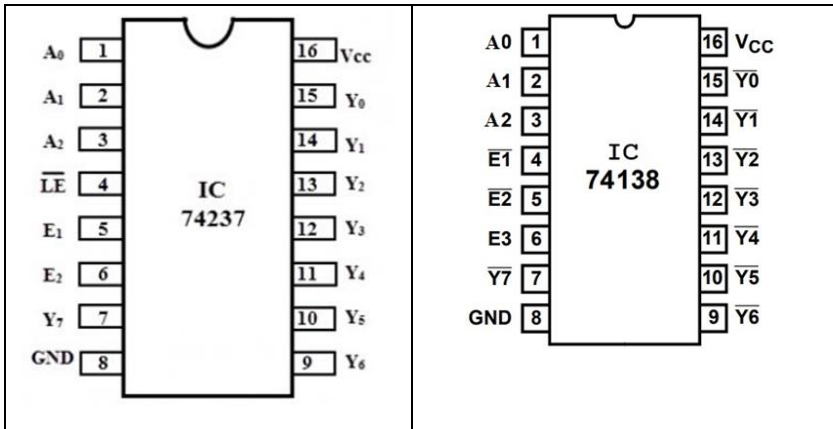


Gambar 5. 14 *demultiplexer* 1-ke-8

Ada dua rangkaian terintegrasi *demultiplexer* 1-ke-8 yang populer. Salah satunya adalah IC 74237, yang terdiri dari kait pada tiga masukan pilih. Pin keluar dari IC ini diberikan di bawah ini.

Pin A0 hingga A2 adalah *input* data, Y0 hingga Y7 adalah keluaran *demultiplexer*, E1&E2 masing-masing adalah pin active-low data enable dan active-high data enable, LE adalah *output latch enable* masukan, Vcc

dan GND adalah tegangan suplai positif dan terminal ground. IC ini mengkombinasikan 3-bit *storage latch* dengan 3-to-8 fungsi decoder.



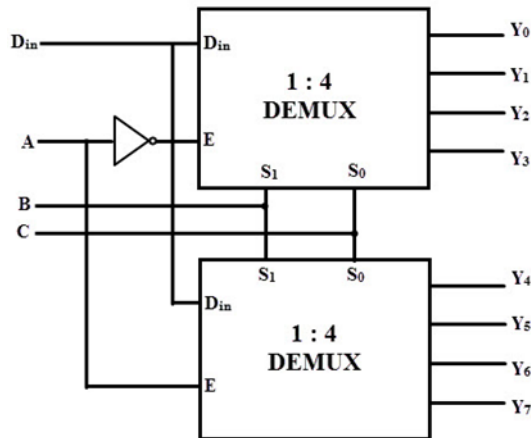
Gambar 5. 15 IC logika

Rangkaian terintegrasi *demultiplexer* 1-ke-8 lainnya yang umum digunakan adalah IC 74138. Prinoutnya sangat mirip kecuali tidak ada Latch Enable masukan (karena semua pin enable adalah pin enable normal – dua aktif LOW dan satu aktif HIGH) dan *outputnya* aktif LOW. Gambar berikut menunjukkan pinout dari IC 74138.

D. 1-TO-8 DEMUX USING TWO 1-TO-4 DEMULTIPLEXERS

Ketika aplikasi membutuhkan *demultiplexer* orde tinggi dengan jumlah pin keluaran yang lebih banyak, maka kami tidak dapat mengimplementasikannya dengan satu sirkuit terpadu. Jika diperlukan lebih dari 16 pin keluaran, maka dua atau lebih IC *demultiplexer* diturunkan untuk memenuhi persyaratan. Misalnya, jika aplikasi membutuhkan 32 baris keluaran dari DEMUX, maka kita cascade dua *demultiplexer* 1:16 atau tiga *demultiplexer* 1:8. Oleh karena itu, dengan mengalirkan dua atau lebih *demultiplexer*, sebuah *demultiplexer* besar dapat diimplementasikan.

Pertimbangkan kasus bahwa *demultiplexer* 1-ke-8 dapat diimplementasikan dengan menggunakan dua *demultiplexer* 1-ke-4 dengan cascading yang tepat.



Gambar 5. 16 1-to-8 DEMUX using Two 1-to- 4 Demultiplexers

Pada gambar di atas, bit A signifikan tertinggi dari masukan pilihan dihubungkan ke masukan yang diaktifkan sedemikian rupa sehingga dilengkapi sebelum menghubungkan ke satu DEMUX dan yang lain terhubung langsung.

Dengan konfigurasi ini, bila A diatur ke nol, salah satu jalur keluaran dari Y_0 ke Y_3 dipilih berdasarkan kombinasi jalur pilih B dan C . Demikian pula, bila A diatur ke satu, berdasarkan jalur pilih salah satu jalur keluaran dari Y_4 ke Y_7 akan dipilih.

E. IMPLEMENTASI SUBTRAKTOR PENUH MENGGUNAKAN DEMUX 1-KE-8

Mirip dengan multiplexer, pada *demultiplexer* juga digunakan untuk implementasi fungsi Boolean serta desain kombinasi rangkaian. Kita dapat merancang *demultiplexer* untuk menghasilkan keluaran tabel kebenaran apa pun dengan mengontrol garis-garis yang dipilih dengan benar.

Pertimbangkan kasus penerapan rangkaian *demultiplexer* untuk menghasilkan keluaran subtraktor penuh. Tabel kebenaran di bawah ini menunjukkan keluaran dari sebuah subtraktor penuh.

Tabel Kebenaran Demux 1 ke 8 [8]

A	B	B _{IN}	D	B _{OUT}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Dari tabel diatas, *the full subtractor* keluaran D dapat ditulis sebagai berikut

$$D = f(A, B, B_{IN})$$

$$D = \sum m(1, 2, 4, 7)$$

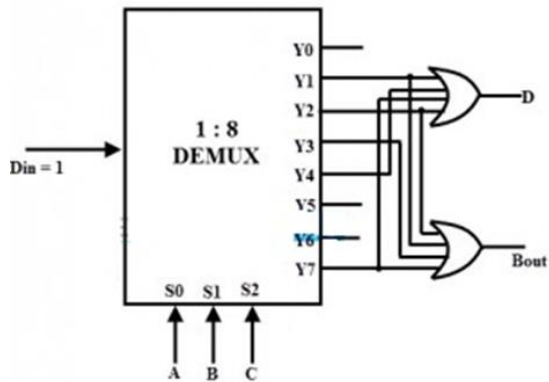
$$D = A \bar{B} B_{IN} + A \bar{B} \bar{B}_{IN} + A B B_{IN} + A B \bar{B}_{IN}$$

Dan keluaran pinjam dapat di ekspresikan sebagai berikut

$$B_{OUT} = f(A, B, B_{IN}) = \sum m(1, 2, 3, 7)$$

$$B_{OUT} = A \bar{B} B_{IN} + A \bar{B} \bar{B}_{IN} + A B B_{IN} + A B \bar{B}_{IN}$$

Dari ekspresi Boolean ini, sebuah *demultiplexer* untuk menghasilkan keluaran subtraktor penuh dapat dibangun dengan mengkonfigurasi DEMUX 1-ke-8 dengan benar, sehingga dengan masukan $D = 1$, memberikan minterm pada keluaran. Dan dengan mengORing secara logis minterms ini, keluaran dari selisih dan pinjam dapat diperoleh seperti yang ditunjukkan pada gambar.



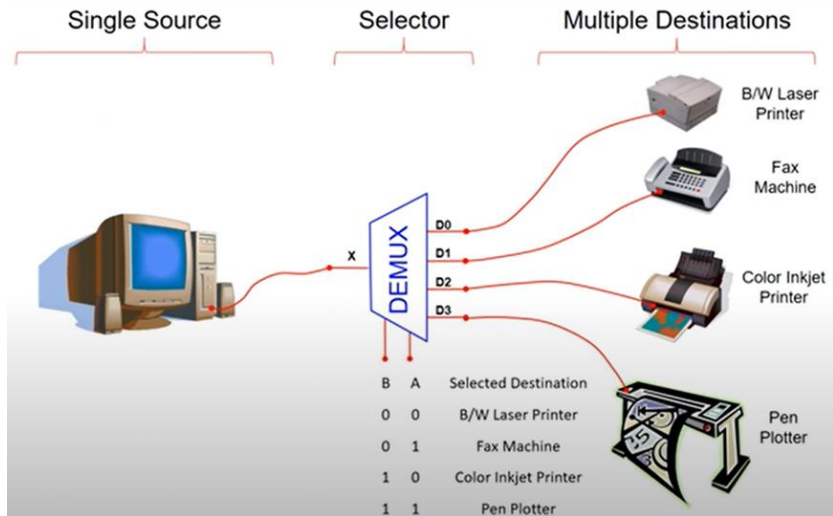
Gambar 5. 17 Demux 1 ke 8

F. APLIKASI DEMULTIFLEXER

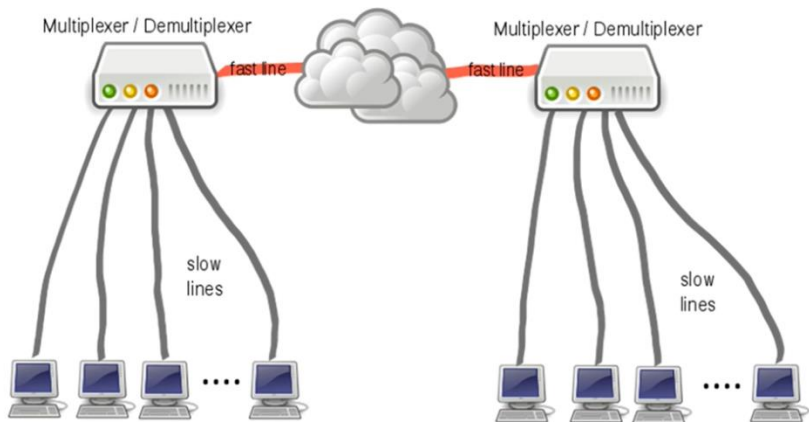
Karena *demultiplexer* digunakan untuk memilih atau mengaktifkan satu sinyal dari banyak sinyal, maka dengan demikian banyak digunakan dalam mikroprosesor atau sistem kontrol komputer seperti:

- Memilih perangkat IO yang berbeda untuk transfer data (Perutean Data).
- Memilih bank memori yang berbeda (Memory Decoding).
- Tergantung pada alamatnya, memungkinkan baris yang berbeda dari chip memori.
- Mengaktifkan unit fungsional yang berbeda.
- Selain ini, *demultiplexer* dapat ditemukan dalam berbagai macam aplikasi seperti:
 - Sistem transmisi data sinkron
 - Implementasi fungsi Boolean
 - Sistem akuisisi data
 - Desain kombinasi rangkaian
 - Sistem peralatan uji otomatis
 - Sistem pemantauan keamanan (untuk memilih kamera pengintai tertentu pada suatu waktu), dll.

Perhatikan contoh pemakaian dari *demultiplexer* pada gambar dibawah ini.



Gambar 5. 18 Demux 1 [16]



Gambar 5. 19 Demux 1[17]



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 6: FLIP FLOP

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 6

FLIP FLOP

A. PENGERTIAN FLIP-FLOP

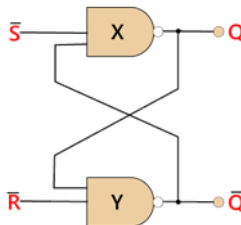
Flip-flop adalah suatu rangkaian elektronika yang memiliki dua kondisi stabil dan dapat digunakan untuk menyimpan informasi. Flip Flop merupakan pengaplikasian gerbang logika yang bersifat Multivibrator Bistabil. Dikatakan **Multivibrator Bistabil** karena kedua tingkat tegangan keluaran pada Multivibrator tersebut adalah stabil dan hanya akan mengubah situasi tingkat tegangan keluarannya saat dipicu (trigger). Flip-flop mempunyai dua Keluaran yang salah satu keluarannya merupakan komplemen keluaran yang lain.

Flip-flop Elektronik yang pertama kali ditemukan oleh dua orang ahli fisika Inggris **William Eccles and F. W. Jordan** pada tahun 1918 ini merupakan dasar dari penyimpanan data memory pada komputer maupun smartphone. Flip-flop juga dapat digunakan sebagai penghitung detak dan sebagai pensinkronisasi *input* sinyal waktu variabel untuk beberapa sinyal waktu referensi. [1]

Rangkaian yang memiliki dua keadaan stabil diperlakukan sebagai flip flop. Status stabil ini digunakan untuk menyimpan data biner yang dapat diubah dengan menerapkan *input* yang bervariasi. Sandal jepit adalah blok bangunan fundamental dari sistem digital. Flip flop dan latch adalah contoh elemen penyimpanan data. Dalam rangkaian logika sekuensial, flip flop adalah elemen penyimpanan dasar. **Kait dan sandal jepit** adalah elemen penyimpanan dasar tetapi berbeda dalam bekerja. Berikut adalah jenis-jenis flip flop:

B. FLIP-FLOP SR

Flip Flop SR adalah flip flop yang paling umum digunakan dalam sistem digital. Dalam flip flop SR, ketika *input* yang disetel "S" benar, *output* Y akan tinggi, dan Y' akan rendah. Hal ini diperlukan bahwa pengkabelan rangkaian dipertahankan ketika *output* ditetapkan. Kita kondisikan sampai *input* set atau reset menjadi tinggi, atau daya dimatikan.



Gambar 6. 1 Flip Flop SR

1. Tabel kebenaran S-R:

Tabel 6. 1 Tbel Kebenaran SR

S	R	Y	Y'
0	0	0	1
0	1	0	1
1	0	1	0
1	1	∞	∞

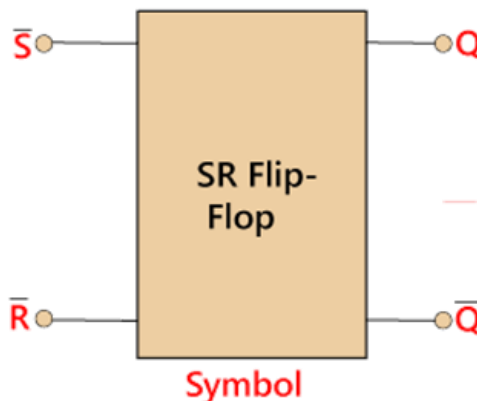
Flip flop SR adalah perangkat bistable memori 1-bit yang memiliki dua *input*, yaitu, SET dan RESET. *Input* SET 'S' mengatur perangkat atau menghasilkan *output* 1, dan *input* RESET 'R' mengatur ulang perangkat atau menghasilkan *output* 0. *Input* SET dan RESET diberi label masing-masing sebagai S dan R.

Flip flop SR adalah singkatan dari flip flop "Set-Reset". *Input* reset digunakan untuk mengembalikan flip flop ke keadaan semula dari keadaan saat ini dengan *output* 'Q'. *Output* ini tergantung pada kondisi set dan reset, baik pada level logika "0" atau "1".

2. Flip-Flop Gerbang NAND SR

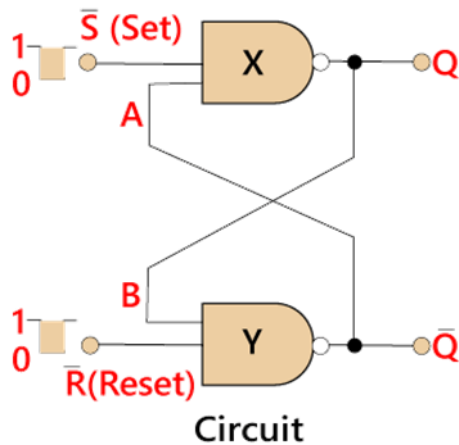
Flip flop SR gerbang NAND adalah flip flop dasar yang memberikan umpan balik dari kedua *output*nya kembali ke *input* yang berlawanan. Rangkaian ini digunakan untuk menyimpan bit data tunggal di rangkaian memori. Jadi, flip flop SR memiliki tiga *input*, yaitu, 'S' dan 'R', dan *output* saat ini 'Q'. *Output* 'Q' ini terkait dengan riwayat atau status saat ini. Istilah "flip-flop" berkaitan dengan operasi perangkat yang sebenarnya, karena dapat "dibalik" ke keadaan set logika atau "di-flop" kembali ke keadaan reset logika yang berlawanan.

Kita dapat mengimplementasikan flip flop set-reset dengan menghubungkan dua gerbang NAND 2-*input cross-coupled* bersama-sama. Dalam rangkaian flip flop SR, dari setiap *output* ke salah satu *input* gerbang NAND lainnya, umpan balik terhubung. Jadi, perangkat memiliki dua *input*, yaitu Set 'S' dan Reset 'R' dengan dua *output* Q dan $\bar{Q}$ masing-masing. Di bawah ini adalah diagram blok dan diagram rangkaian flip flop SR.



Gambar 6. 2 Blok Diagram Flip Flop SR

4. Diagram Rangkaian Flip Flop SR :



Gambar 6. 3 Diagram Rangkaian Flip Flop SR

5. Keadaan yang Ditetapkan

Pada diagram di atas, ketika *input* R diset false atau 0 dan *input* S diset true atau 1, gerbang NAND Y memiliki *input* 0, yang akan menghasilkan *output* Q' 1. Nilai Q' adalah memutar ke gerbang NAND 'X' sebagai *input* 'A', dan sekarang kedua *input* gerbang NAND 'X' adalah 1(S=A=1), yang akan menghasilkan *output* 'Q' 0.

Sekarang, jika *input* R diubah menjadi 1 dengan 'S' tersisa 1, *input* gerbang NAND 'Y' adalah R=1 dan B=0. Di sini salah satu *input*nya juga 0, sehingga *output* Q' adalah 1. Jadi, rangkaian flip-flop diset atau di-latched dengan Q=0 dan Q'=1.

Setel Ulang Status

Keluaran Q' adalah 0, dan keluaran Q adalah 1 dalam keadaan stabil kedua. Diberikan oleh R = 1 dan S = 0. Salah satu *input* gerbang NAND 'X' adalah 0, dan *output* Q-nya adalah 1. *Output* Q dipudarkan ke gerbang NAND Y sebagai *input* B. Jadi, kedua *input* ke Gerbang NAND Y diatur ke 1, oleh karena itu, Q' = 0.

Sekarang, jika *input* S diubah menjadi 0 dengan 'R' tersisa 1, *output* Q' akan menjadi 0 dan tidak ada perubahan status. Jadi, keadaan reset dari rangkaian flip flop telah terkunci, dan tindakan set/reset didefinisikan dalam tabel kebenaran berikut:

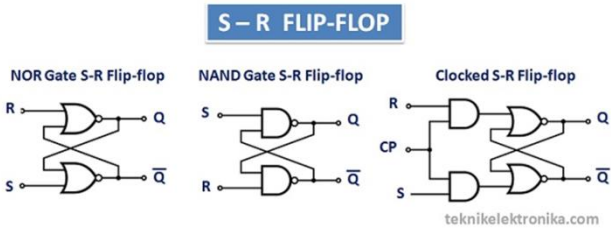
State	S	R	Q	Q'	Description
Set	1	0	0	1	Set Q'>>1
	1	1	0	1	No change
Reset	0	1	1	0	Reset Q'>>0
	1	1	1	0	No change
Invalid	0	0	1	1	Invalid Condition

Gambar 6. 4 Status Tabel Flip Flop SR

Dari tabel kebenaran di atas, kita dapat melihat bahwa ketika *input* 'S' dan reset 'R' diset ke 1, *output* Q dan Q' akan menjadi 1 atau 0. *Output* ini bergantung pada status *input* S atau R sebelumnya kondisi *input* ada. Jadi, ketika *input*nya adalah 1, status *output*nya tetap tidak berubah.

Kondisi di mana kedua status *input* diatur ke 0 diperlakukan sebagai tidak valid dan harus dihindari. S-R adalah singkatan dari “Set” dan “Reset”. Sesuai dengan namanya, S-R Flip-flop ini terdiri dari dua masukan (*INPUT*) yaitu S dan R. S-R Flip-flop ini juga terdapat dua Keluaran (*OUTPUT*) yaitu Q dan Q'. Rangkaian S-R Flip-flop ini umumnya terbuat dari 2 gerbang logika NOR ataupun 2 gerbang logika NAND. Ada juga S-R Flip-flop yang terbuat dari gabungan 2 gerbang Logika NOR dan NAND.

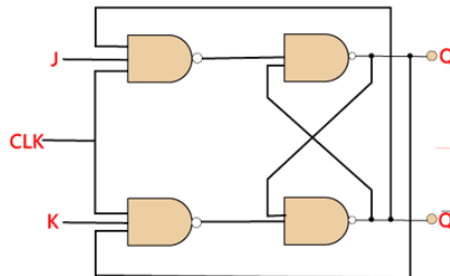
Berikut ini adalah diagram logika NOR Gate S-R Flip-flop, NAND Gate S-R Flip-Flop dan *Clocked* S-R Flip-flop (gabungan gerbang logika NOR dan NAND).



Gambar 6. 5 Flip Flop SR

C. FLIP-FLOP JK

Flip Flop JK digunakan untuk menghilangkan kelemahan flip flop SR, yaitu keadaan tidak terdefinisi. Flip flop JK dibentuk dengan melakukan modifikasi pada flip flop SR. Flip flop SR ditingkatkan untuk membangun flip flop JK. Ketika *input* S dan R disetel ke *true*, flip flop SR memberikan hasil yang tidak akurat. Tetapi dalam kasus JK flip flop, ia memberikan *output* yang benar.



Gambar 6. 6 Flip Flop JK

Pada flip flop JK, jika kedua *inputnya* berbeda, nilai J pada tepi clock berikutnya diambil oleh *output* Y. Jika kedua *inputnya* rendah, maka tidak terjadi perubahan, dan jika tinggi pada tepi clock, kemudian dari satu state ke state lainnya, *output* akan di-toggle. JK Flip Flop adalah Set atau Reset Flip flop dalam sistem digital.

Tabel kebenaran Flip Flop JK :

J	K	Y	Y'
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	0

Tabel 6. 3 Tabel Kebenaran JK

Master-Slave Flip Flop JK

Dalam "JK Flip Flop", ketika *input* dan CLK diatur ke 1 untuk waktu yang lama, maka *output* Q beralih hingga CLK adalah 1. Dengan demikian, dihasilkan *output* yang tidak pasti atau tidak dapat diandalkan. Masalah ini disebut sebagai kondisi *race-round* di JK flip-flop dan dihindari dengan memastikan bahwa CLK disetel ke 1 hanya untuk waktu yang sangat singkat.

Penjelasan

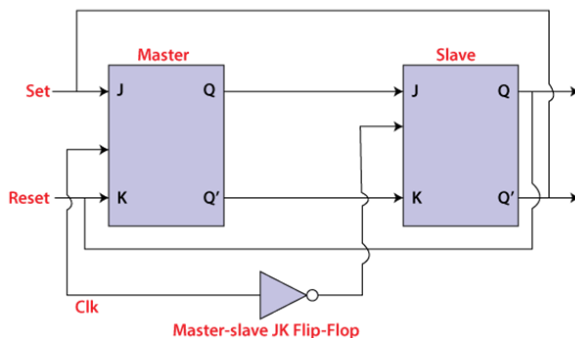
Flip flop master-slave dibangun dengan menggabungkan dua flip flop JK

Sandal jepit ini dihubungkan dalam konfigurasi seri. Dalam dua flip flop ini, flip flop pertama berfungsi sebagai "master", disebut master flip flop, dan yang ke-2 berfungsi sebagai "slave", disebut slave flip flop. Flip flop master-slave dirancang sedemikian rupa sehingga *output* dari flip flop "master" dilewatkan ke kedua *input* dari flip flop "slave".

Output dari flip flop "slave" dilewatkan ke *input* dari master flip flop.

Dalam "flip flop master-slave", selain dari dua flip flop ini, sebuah gerbang inverter atau NOT juga digunakan. Untuk melewatkan pulsa clock terbalik ke flip flop "slave", inverter terhubung ke pulsa clock. Dengan kata sederhana, ketika CP disetel ke false untuk "master", maka CP disetel ke true untuk "slave", dan ketika CP disetel ke true untuk "master", maka CP disetel ke false untuk "slave".

Flip Flop Master-Budak JK



Gambar 6. 7 Flip Flop Master-Budak JK

Bekerja:

Ketika pulsa *clock* benar, flip flop budak akan berada dalam keadaan terisolasi, dan keadaan sistem dapat dipengaruhi oleh *input* J dan K. "*slave*" tetap terisolasi sampai CP adalah 1. Ketika CP diatur ke 0, master flip-flop meneruskan informasi ke *slave* flip-flop untuk mendapatkan *output*.

Flip flop master merespons terlebih dahulu dari *slave* karena flip flop master adalah pemicu level positif, dan flip flop *slave* adalah pemicu level negatif.

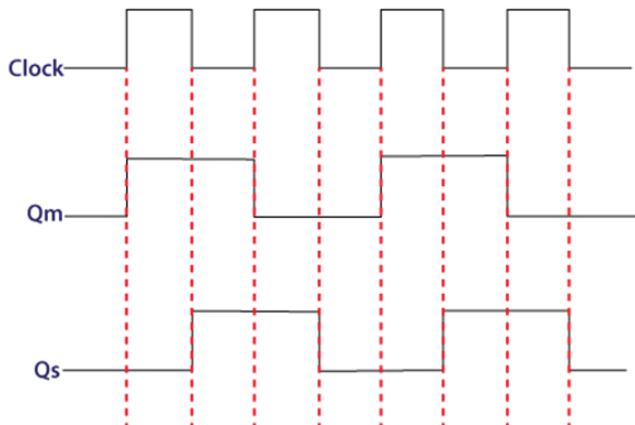
Keluaran $Q'=1$ dari flip flop master dilewatkan ke flip flop *slave* sebagai *input* K ketika *input* J diatur ke 0 dan K disetel ke 1. Jam memaksa flip flop *slave* bekerja sebagai reset, dan kemudian budak menyalin master flip flop.

Ketika $J=1$, dan $K=0$, *output* $Q=1$ dilewatkan ke *input* J dari *slave*. Transisi negatif jam mengatur budak dan menyalin master.

Flip flop master beralih pada transisi positif jam ketika *input* J dan K diatur ke 1. Pada saat itu, flip flop budak beralih pada transisi negatif jam.

Flip flop akan dinonaktifkan, dan Q tetap tidak berubah ketika kedua *input* dari JK flip flop diatur ke 0.

Diagram Waktu dari Master Flip Flop:



Gambar 6. 8 Diagram Waktu dari Master Flip Flop

Flip Flop Master-Budak JK

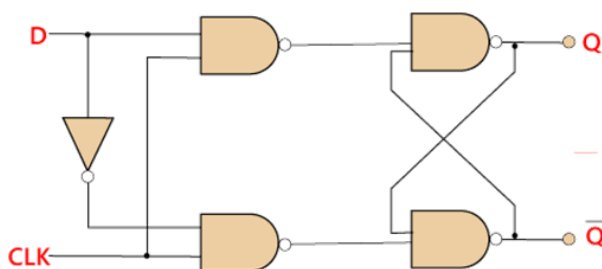
Ketika pulsa clock diset ke 1, *output* dari master flip flop akan menjadi satu sampai *input* clock tetap 0. Ketika pulsa clock menjadi high kembali, maka *output* master adalah 0, yang akan di set menjadi 1 saat clock menjadi satu lagi.

Flip flop master beroperasi ketika pulsa clock adalah 1. Keluaran *slave* tetap 0 sampai jam tidak disetel ke 0 karena flip flop *slave* tidak beroperasi.

Flip flop *slave* beroperasi ketika pulsa clock adalah 0. *Output* dari master tetap satu sampai clock tidak disetel ke 0 lagi. Toggling terjadi selama seluruh proses karena *output* berubah sekali dalam siklus. [2].

D. D FLIP FLOP

D flip flop adalah flip flop yang banyak digunakan dalam sistem digital. Flip flop D banyak digunakan dalam shift-register, counter, dan sinkronisasi *input*.



Gambar 6. 9 Flip flop D

Tabel kebenaran D Flip Flop :

Clock	D	Y	Y'
↓ » 0	0	0	1
↑ » 1	0	0	1
↓ » 0	1	0	1
↑ » 1	1	1	0

Kita tahu bahwa SR flip-flop membutuhkan dua *input*, yaitu, satu untuk "SET" *output* dan satu lagi untuk "RESET" *output*. Dengan menggunakan inverter, kita dapat mengatur dan mengatur ulang *output* hanya dengan satu *input* karena sekarang kedua sinyal *input* saling melengkapi. Di SR flip flop, ketika kedua *input*nya 0, keadaan itu tidak mungkin lagi. Ini adalah ambiguitas yang dihilangkan oleh komplemen di D-flip flop.

Dalam D flip flop, *input* tunggal "D" disebut sebagai *input* "Data". Ketika *input* data diatur ke 1, flip flop akan diatur, dan ketika diatur ke 0, flip flop akan berubah dan menjadi reset. Namun, ini akan sia-sia karena *output* dari flip flop akan selalu berubah pada setiap pulsa yang diterapkan pada *input* data ini.

Input "CLOCK" atau "ENABLE" digunakan untuk menghindari hal ini karena mengisolasi *input* data dari rangkaian penguncian flip flop. Ketika *input* clock disetel ke true, kondisi *input* D hanya disalin ke *output* Q. Ini membentuk dasar perangkat sekuensial lain yang disebut D Flip Flop.

Ketika *input* jam diatur ke 1, *input* "set" dan "reset" dari flip-flop keduanya diatur ke 1. Jadi tidak akan mengubah keadaan dan menyimpan data yang ada pada *output*nya sebelum transisi jam terjadi. Dengan kata sederhana, *output* "terkunci" pada 0 atau 1.

Tabel Kebenaran untuk Flip Flop tipe-D

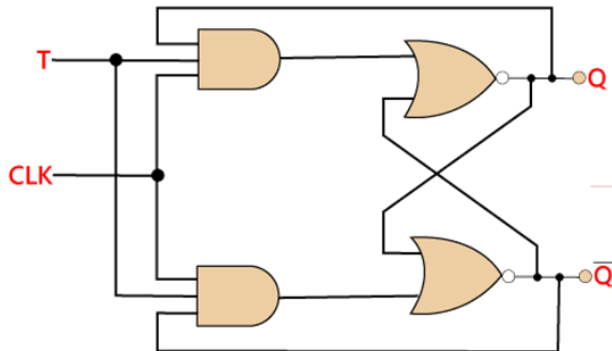
Clock	D	Q	Q'	Description
↓ » 0	X	Q	Q'	Memory no change
↑ » 1	0	0	1	Reset Q » 0
↑ » 1	1	1	0	Set Q » 1

Simbol dan menunjukkan arah pulsa clock. Flip flop tipe-D mengasumsikan simbol-simbol ini sebagai pemicu tepi.[9]

D flip flop adalah flip flop yang banyak digunakan dalam sistem digital. Flip flop D banyak digunakan dalam shift-register, counter, dan sinkronisasi *input*.

E. T FLIP FLOP

Sama seperti JK flip-flop, T flip-flop digunakan. Tidak seperti flip flop JK, pada flip flop T, hanya ada *input* tunggal dengan *input* clock. Flip flop T dibangun dengan menghubungkan kedua *input* JK flip flop bersama-sama sebagai *input* tunggal.



Gambar 6. 10 Flip-flop T

Flip-flop T juga dikenal sebagai Toggle flip-flop. Flip-flop T ini dapat menemukan komplemen dari keadaannya.

Tabel kebenaran T FF:

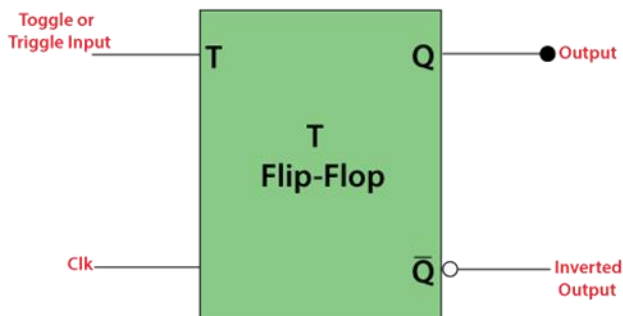
T	Y	Y (t+1)
0	0	0
1	0	1
0	1	1
1	1	0

T Flip-flop merupakan bentuk sederhana dari J-K Flip-flop. Kedua *Input* J dan K dihubungkan sehingga sering disebut juga dengan Single J-K Flip-Flop. Berikut ini adalah diagram logika T flip-flop.

Dalam T flip flop, "T" mendefinisikan istilah "Toggle". Di SR Flip Flop, kami hanya menyediakan satu *input* yang disebut *input* "Toggle" atau "Trigger" untuk menghindari terjadinya keadaan perantara. Sekarang, flip-flop ini berfungsi sebagai sakelar Toggle. Keadaan keluaran berikutnya diubah dengan komplemen dari keluaran keadaan sekarang. Proses ini dikenal sebagai "Toggling".

Kita dapat membuat "T Flip Flop" dengan membuat perubahan pada "JK Flip Flop". "T Flip Flop" hanya memiliki satu *input*, yang dibangun dengan menghubungkan *input* JK flip-flop. Masukan tunggal ini disebut T. Dengan kata sederhana, kita dapat membuat "T Flip Flop" dengan mengubah "JK Flip Flop". Terkadang "T Flip Flop" disebut sebagai *input* tunggal "JK Flip Flop".

Diagram blok "T-Flip Flop" diberikan di mana T mendefinisikan "*input* Toggle", dan CLK mendefinisikan *input* sinyal clock.



Gambar 6. 11 Flip-flop T

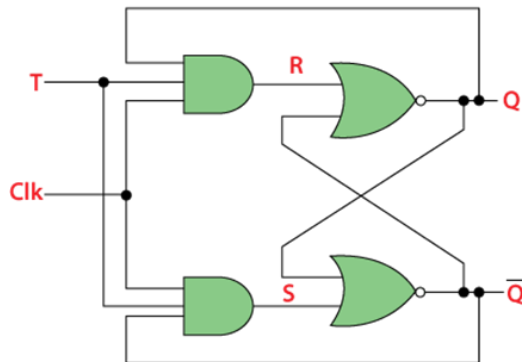
Ada dua metode berikut yang digunakan untuk membentuk "T Flip Flop":

Dengan menghubungkan umpan balik *output* ke *input* di "SR Flips Flop".

Kami melewati *output* yang kami dapatkan setelah melakukan operasi XOR dari *output* T dan QPREV sebagai *input* D di D Flip Flop.

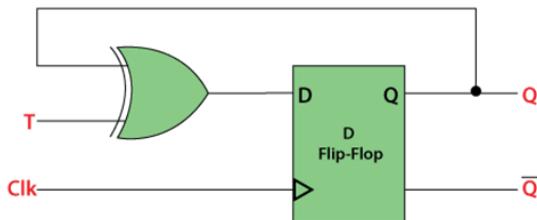
Konstruksi

"T Flip Flop" dirancang dengan melewati *output* gerbang AND sebagai *input* ke gerbang NOR dari "SR Flip Flop". *Input* dari gerbang "AND", status keluaran saat ini Q , dan komplementennya Q' dikirim kembali ke setiap gerbang AND. *Input* toggle dilewatkan ke gerbang AND sebagai *input*. Gerbang ini terhubung ke sinyal Clock (CLK). Dalam "T Flip Flop", rangkaian pulsa pemicu sempit dilewatkan sebagai *input* sakelar, yang mengubah status keluaran flip flop. Diagram rangkaian "T Flip Flop" menggunakan "SR Flip Flop" diberikan di bawah ini:



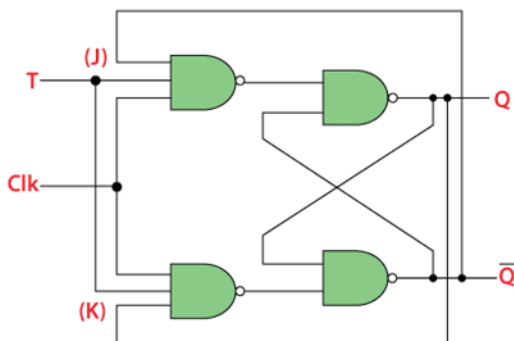
Gambar 6. 12 T Flip Flop" menggunakan "SR Flip Flop

"T Flip Flop" dibentuk menggunakan "D Flip Flop". Pada flip-flop D, *output* setelah melakukan operasi XOR dari *input* T dengan *output* "QPREV" dilewatkan sebagai *input* D. Rangkaian logis dari "T-Flip Flop" menggunakan "D Flip Flop" diberikan di bawah ini:



Gambar 6. 13 T-Flip Flop" menggunakan "D Flip Flop

Konstruksi paling sederhana dari D Flip Flop adalah dengan JK Flip Flop. Kedua *input* dari "JK Flip Flop" terhubung sebagai *input* tunggal T. Di bawah ini adalah rangkaian logis dari T Flip Flop yang terbentuk dari "JK Flip Flop":



Gambar 6. 14 T-Flip Flop menggunakan "JK Flip Flop

Tabel Kebenaran T Flip Flop

T	Previous		Next	
	Q	Q'	Q	Q'
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

Gerbang NAND atas diaktifkan, dan gerbang NAND bawah dinonaktifkan ketika *output* Q diatur ke 0. membuat flip flop dalam "status set ($Q = 1$)", pemicu melewati *input* S di flip flop.

Gerbang NAND atas dinonaktifkan, dan gerbang NAND bawah diaktifkan ketika *output* Q diatur ke 1. Pemicu melewati *input* R di flip flop untuk membuat flip flop dalam keadaan reset ($Q = 0$).

Pengoperasian T-Flip Flop

Keadaan selanjutnya dari flip flop T serupa dengan keadaan saat ini ketika *input* T disetel ke false atau 0.

Jika *input* toggle diatur ke 0 dan status saat ini juga 0, status selanjutnya adalah 0.

Jika *input* toggle diatur ke 0 dan status saat ini adalah 1, status selanjutnya adalah 1.

Keadaan flip flop selanjutnya berlawanan dengan keadaan saat ini ketika *input* sakelar diatur ke 1.

Jika *input* toggle diatur ke 1 dan status saat ini adalah 0, status selanjutnya adalah 1.

Jika *input* toggle diatur ke 1 dan status saat ini adalah 1, status berikutnya adalah 0.

"T Flip Flop" diaktifkan ketika set dan reset *input* secara alternatif diubah oleh pemacu yang masuk. "T Flip Flop" membutuhkan dua pemacu untuk menyelesaikan siklus penuh bentuk gelombang keluaran. Frekuensi *output* yang dihasilkan oleh "T Flip Flop" adalah setengah dari frekuensi *input*. "T Flip Flop" berfungsi sebagai "Rangkaian Pembagi Frekuensi".

Sumber : <https://www.javatpoint.com/t-flip-flop-in-digital-electronics>
diakses 1 nov 21 jam 7.00 WIB.

Dalam rangkaian Bistabil Gerbang SR NAND, kondisi *input* yang tidak ditentukan dari SET = "0" dan RESET = "0" dilarang. Ini adalah kelemahan dari SR flip flop. negara ini:

Ganti tindakan penguncian umpan balik.

Paksa kedua *output* menjadi 1.

Kehilangan kontrol oleh *input*, yang pertama masuk ke 1, dan *input* lainnya tetap "0" di mana status kait yang dihasilkan dikendalikan.

Kami membutuhkan inverter untuk mencegah hal ini terjadi. Kami menghubungkan inverter antara *input* Set dan Reset untuk menghasilkan jenis lain dari rangkaian flip flop yang disebut D flip flop, Penundaan flip flop, D-type Bistabil, D-type flip flop.

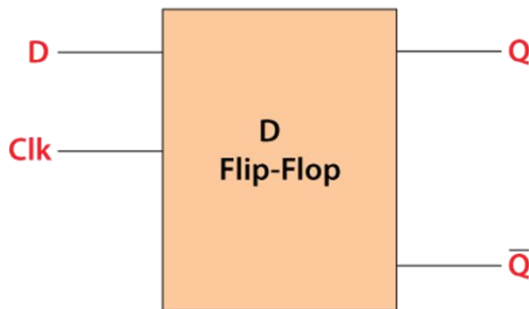
Flip flop D adalah flip flop yang paling penting dari jenis clock lainnya. Ini memastikan bahwa pada saat yang sama, kedua *input*, yaitu, S dan R, tidak pernah sama dengan 1. Flip-flop penundaan dirancang menggunakan flip-flop SR yang terjaga keamanannya.

dengan inverter yang terhubung di antara *input* yang memungkinkan satu *input* D (Data).

Input data tunggal ini, yang diberi label sebagai "D" digunakan sebagai pengganti *input* "Set" dan untuk *input* "Reset" pelengkap, digunakan inverter. Dengan demikian, flip flop tipe-D atau D yang peka-tingkat dibangun dari flip-flop SR yang peka-tingkat.

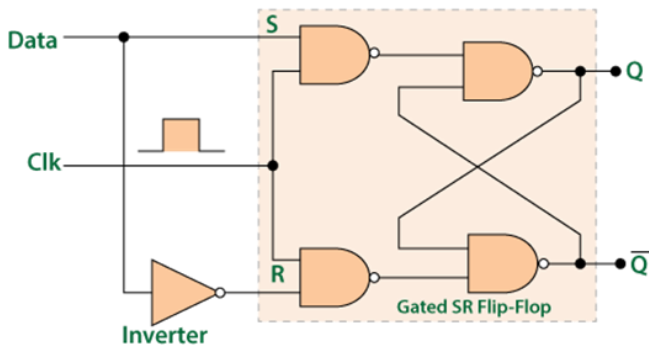
Jadi, di sini $S=D$ dan $R=\sim D$ (pelengkap dari D)

Diagram Blok



Gambar 6. 15 Flip flop D

Diagram Rangkaian

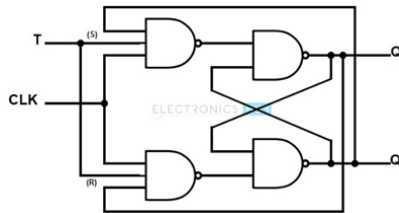


Gambar 6. 16 Diagram Rangkaian Flip Flop D

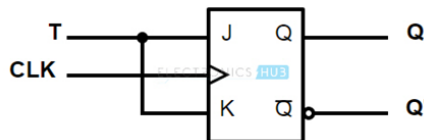
F. WORKING

T flip – flop is an edge triggered device i.e., the low to high or high to low transitions on the masukan clock signal will cause a change in the keluaran state of the flip – flop.

For example, consider a T flip – flop made of NAND SR latch as shown below.



Gambar 6. 17 Cara kerja



Gambar 6. 18 Flip flop T

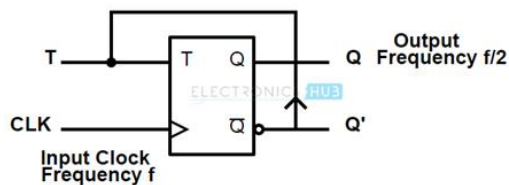
G. APPLICATIONS

Let us now see a couple of important applications of T flip-flop.

- 1-Frequency Division Circuit
- 2 – Bit Parallel Load Registers

1. Frequency Division Circuit

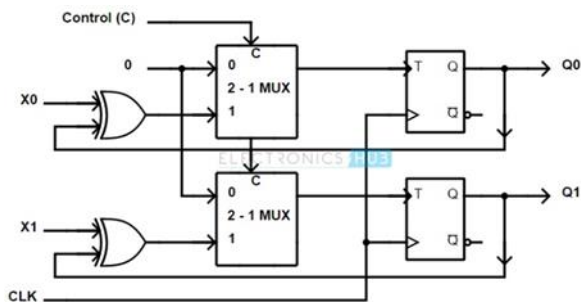
The logic symbol of a frequency divider using T flip – flop is shown below.



Gambar 6. 19 Flip Flop T

2. Bit Parallel Load Register using T flip – flops

The circuit of a 2 – bit parallel load register is shown below.



Gambar 6. 20 Flip flop T Paralel

Gambar [10]



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 7: ALJABAR BOOLEAN

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 7

ALJABAR BOOLEAN

Aljabar Boolean adalah aljabar matematika yang hanya memiliki 2 bilangan yaitu 0 dan 1 (bilangan biner), aljabar boolean disebut juga penjabaran teknik logika x dan y dimana di dalam programnya terdapat diantaranya : AND, OR, dan NOT yang nanti akan saya jelaskan dibawah.

FUNGSI LOGIKA

Mungkin fungsi logika pada matematika ada banyak, diantaranya permutasi, kombinasi, komutatif, asosiatif, dan lain lain yang sering kita pelajari di sekolah, namun pada fungsi logika yang akan kita bahas adalah fungsi logika yang terdapat dalam CU (*control unit*) pada CPU (komputer).

Komputer memiliki fungsi logika yang sama seperti logika matematika, hanya saja fungsi ini hanya menjawab 2 jawaban saja yakni true atau false (0 atau 1), contoh : $1+1=0$ (simpan 1), ini menandakan angka 1 jika di AND dengan 1 hasilnya 0 (true) bukan 2, hal ini bersifat sama seperti bilang biner.

Negasi dari suatu aljabar boolean menunjukkan bahwa perubahan perintah dari true menjadi false seperti $\sim x$ (negasi x), namun jika dinegasikan lagi maka akan kembali seperti awal seperti $\sim(\sim x)=x$, hal ini disebut involution, ini terdapat pada hukum De' Morgan.

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\rightarrow$	0	1
0	1	1
1	0	1

$\oplus$	0	1
0	0	1
1	1	0

Figure 1. Truth tables

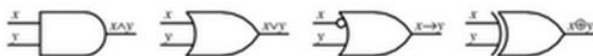


Figure 2. Logic gates

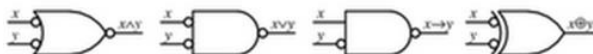


Figure 3. De Morgan equivalents



Figure 4. Venn diagrams

Gambar 7. 1 Tabel aljabar boolean

FUNGSI AND DAN XAND

Fungsi AND dan XAND adalah fungsi dimana x dan y ditambahkan seperti tabel diatas:

TABEL FUNGSI AND

Ini menandakan bahwa pada fungsi AND nilai x dan y jika salah satunya true maka akan true, sedangkan jika dua-duanya false maka jawabannya false, fungsi AND diibaratkan dengan penggabungan 2 buah nilai yang berbeda, sedangkan pada XAND adalah sebuah fungsi negasi dari AND.

AND	0	1
0	0	0
1	0	1

Gambar 7. 2 Tabel AND

FUNGSI OR DAN XOR

Fungsi ini sering disebut fungsi perbandingan, karena pada fungsi ini jikalau salah satunya false maka hasilnya akan false pula, seperti tabel berikut:

OR	0	1
0	0	1
1	1	1

Gambar 7. 3 Tabel OR

Dan XOR adalah negasi dari OR.

Seperti yang telah kita ketahui diatas, bahwa aljabar boolean sama seperti aljabar yang lainnya, seperti fungsi logika yang memiliki AND, OR, PERMUTASION dan COMBINATION (hanya saja permutasi dan kombinasi tidak akan saya jelaskan), dan seperti yang kita ketahui lagi ternyata komputer pun memiliki sistem prosesnya melalui logika yang sering kita gunakan.

Menguraikan Rangkaian-Rangkaian Logika Secara Aljabar Setiap rangkaian logika, bagaimanapun kompleksnya, dapat diuraikan secara lengkap dengan menggunakan operasi-operasi Boolean yang telah didefinisikan sebelumnya, karena rangkaian OR gate, AND gate, dan NOT gate merupakan blok-blok bangun dasar dari system-sistem digital. Perhatikan rangkaian pada gambar 4.1. Rangkaian ini tiga masukan, A, B, dan C dan suatu keluaran X. Dengan menggunakan ekspresi Boolean untuk tiap gate dengan mudah dapat ditentukan ekspresi keluarannya.

Rangkaian-rangkaian yang menggunakan Inverter Jika suatu Inverter ada disuatu rangkaian logika, ekspresi keluarannya secara sederhana sama dengan ekspresi masukan dengan tandan string (') diatasnya.

A. PENGERTIAN ALJABAR BOOLEAN

Aljabar boolean adalah aljabar yang berhubungan dengan variabel biner dan operasi logik, dimana aljabar boolean adalah sistem matematika yang terbentuk dari 3 operator logika berupa "negasi", Logika "AND" dan "OR". Selain simbol logika "0" dan "1" yang digunakan untuk merepresentasikan masukan atau keluaran digital, kita juga dapat menggunakannya sebagai konstanta pada rangkaian terbuka atau rangkaian tertutup secara permanen. Serangkaian aturan dari ekspresi Aljabar Boolean telah diciptakan untuk membantu mengurangi jumlah

gerbang logika dasar yang dibutuhkan dalam melakukan operasi logika tertentu sehingga akan dihasilkan daftar fungsi atau teorema yang dikenal umum sebagai Hukum Aljabar Boolean.

Aljabar Boolean adalah operasi matematika yang berguna dalam menganalisis gerbang dan rangkaian digital, dengan menggunakan "Hukum Boolean" ini maka akan dapat mengurangi atau menyederhanakan ekspresi Boolean yang kompleks dengan maksud untuk mengurangi jumlah gerbang logika yang diperlukan. Oleh sebab itu, Aljabar Boolean adalah sistem matematika yang didasarkan pada logika yang memiliki seperangkat aturan atau hukum yang berguna dalam menentukan, mengurangi atau menyederhanakan ekspresi Boolean.

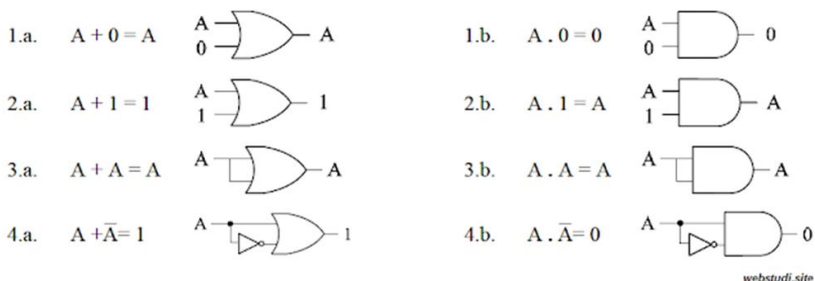
Variabel yang digunakan dalam Aljabar Boolean hanya memiliki dua kemungkinan yaitu logika "0" dan logika "1" tetapi ekspresi jumlah variabel yang dihasilkan tak terbatas yang semuanya dilabeli secara individual untuk mewakili masukan ke ekspresi. Sebagai contoh, sebuah variabel A, B, C dll, dihasilkan sebuah ekspresi logis yaitu $A + B = C$, tetapi setiap variabel HANYA dapat berupa 0 atau 1.

Perhatikanlah tabel di bawah ini yang menjelaskan mengenai hukum boolean, aturan dan teorema Aljabar Boolean

B. TABEL KEBENARAN HUKUM ALJABAR BOOLEAN

Boolean Expression	Description	Equivalent Switching Circuit	Boolean Algebra Law or Rule
$A + 1 = 1$	A in parallel with closed = "CLOSED"		Annulment
$A + 0 = A$	A in parallel with open = "A"		Identity
$A \cdot 1 = A$	A in series with closed = "A"		Identity
$A \cdot 0 = 0$	A in series with open = "OPEN"		Annulment
$A + A = A$	A in parallel with A = "A"		Idempotent
$A \cdot A = A$	A in series with A = "A"		Idempotent
$\text{NOT } \overline{\overline{A}} = A$	NOT NOT A (double negative) = "A"		Double Negation
$A + \overline{A} = 1$	A in parallel with NOT A = "CLOSED"		Complement
$A \cdot \overline{A} = 0$	A in series with NOT A = "OPEN"		Complement
$A + B = B + A$	A in parallel with B = B in parallel with A		Commutative
$A \cdot B = B \cdot A$	A in series with B = B in series with A		Commutative
$\overline{A+B} = \overline{A} \cdot \overline{B}$	invert and replace OR with AND		de Morgan's Theorem
$\overline{A \cdot B} = \overline{A} + \overline{B}$	invert and replace AND with OR		de Morgan's Theorem

Gambar 7. 4 Hukum Aljabar Boolean



C. FUNGSI ALJABAR BOOLEAN

Berdasarkan penjelasan sebelumnya, maka gerbang dasar AND, OR, NOT 2 masukan akan menghasilkan 16 fungsi yang ditunjukkan pada tabel dibawah

Function	Description	Expression
1.	NULL	0
2.	IDENTITY	1
3.	Input A	A
4.	Input B	B
5.	NOT A	$\bar{A}$
6.	NOT B	$\bar{B}$
7.	A AND B (AND)	$A \cdot B$
8.	A AND NOT B	$A \cdot \bar{B}$
9.	NOT A AND B	$\bar{A} \cdot B$
10.	NOT AND (NAND)	$\overline{A \cdot B}$
11.	A OR B (OR)	$A + B$
12.	A OR NOT B	$A + \bar{B}$
13.	NOT A OR B	$\bar{A} + B$
14.	NOT OR (NOR)	$\overline{A + B}$
15.	Exclusive-OR	$A \cdot \bar{B} + \bar{A} \cdot B$
16.	Exclusive-NOR	$A \cdot B + \bar{A} \cdot \bar{B}$

Gambar 7. 5 Fungsi Aljabar Boolean

D. OPERASI ALJABAR BOOLEAN DUA VARIABEL DAN TIGA VARIABEL

Dalam menyelesaikan permasalahan yang berkaitan dengan aljabar Boolean dua dan tiga variabel maka digunakan sifat identitas, disamping itu prinsip dualitas juga digunakan pada sifat-sifat identitas.

Jika A, B, dan C adalah Variabel Boole maka berlaku hukum/sifat berikut

Hukum Komutatif

Hukum Komutatif menjelaskan bahwa penukaran atau perubahan urutan variabel masukan atau sinyal masukan sama sekali tidak mempengaruhi variabel keluaran suatu rangkaian logika.

komutatif (pertukaran):

$$- A + B = B + A$$

$$- A \cdot B = B \cdot A$$

webstudi.site

Hukum Asosiatif

Hukum Asosiatif menjelaskan bahwa perubahan urutan penyelesaian operasi pada variabel tidak akan mempengaruhi variabel keluaran suatu rangkaian logika.

asosiatif (pengelompokan):

$$- A + (B + C) = (A + B) + C$$

$$- A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

webstudi.site

Hukum Distributif

Hukum Distributif menyatakan bahwa variabel masukan pada operasi aljabar Boolean dapat disebarkan tempatnya tanpa mengubah variabel hasil dari keluaran suatu rangkaian logika.

distributif (penyebaran):

$$- A \cdot (B + C) = A \cdot B + A \cdot C$$

$$- A + B \cdot C = (A + B) \cdot (A + C)$$

webstudi.site

Hukum Absorpsi, Kombinasi, dan Konsensus

absorpsi (penghilangan):

$$- A + A \cdot B = A$$

$$- A \cdot (A + B) = A$$

kombinasi:

$$- A \cdot B + A \cdot \bar{B} = A$$

$$- (A + B) \cdot (A + \bar{B}) = A$$

konsensus:

$$- A \cdot B + B \cdot C + \bar{A} \cdot C = A \cdot B + \bar{A} \cdot C$$

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

Sifat Identitas lain

sifat-sifat identitas lain:

$$- A + \bar{A} \cdot B = A + B$$

$$- A \cdot (\bar{A} + B) = A \cdot B$$

Beberapa sifat pada teorema Boolean dapat dipergunakan untuk menyederhanakan suatu pernyataan logika, dengan kata lain suatu pernyataan kompleks akan dapat disederhanakan dengan menggunakan konsep teorema Boolean tanpa mengubah fungsi logikanya.

E. PRIORITAS OPERASI ALJABAR BOOLEAN

Pada teorema Aljabar Boolean dikenal 3 operasi logika yaitu operasi logika OR, AND, dan NOT sehingga dapat dihasilkan berbagai bentuk fungsi logika. Demi memudahkan dalam pengoperasiannya maka dipergunakan tanda kurung untuk memberikan prioritas. Pada dasarnya konsep prioritas operasi ini tidak ada bedanya dengan konsep prioritas pada operasi aritmatika.

Berikut aturan prioritas operasi Aljabar Boolean

1. Bila terdapat tanda kurung maka diselesaikan terlebih dahulu.
2. Bila tidak terdapat tanda kurung, maka suatu pernyataan logika diselesaikan dengan urutan : NOT, AND setelah itu OR.

Contoh :

$$AB + \overline{A}\overline{B}$$

1. pertama kerjakan terlebih dahulu negasi A dan B
2. Lanjutkan dengan operasi
 $A \text{ AND } B \text{ dan } \overline{A} \text{ AND } \overline{B}$
3. Kemudian baru lanjutkan dengan operasi
 $(AB) \text{ OR } (\overline{A}\overline{B})$

Contoh Soal Aljabar Boolean

Berikut ini kami sajikan beberapa contoh soal Aljabar Boolean

1. **Suatu lokasi memori mempunyai nilai 1000 1100. Jadikan bit ke-5 dan ke-4 menjadi '1', dan bit ke-3 dan ke-2 menjadi '0' tanpa mengubah bit-bit yang lain!**

Jawab:

Untuk menjadikan bit ke-5 dan ke-4 menjadi '1' dapat dilakukan dengan operasi OR.

$$\begin{array}{r}
 1000\ 1100 \\
 0011\ 0000 \\
 \hline
 1011\ 1100
 \end{array}
 \quad \text{OR}$$

Untuk menjadikan bit ke-3 dan ke-2 menjadi '0' dapat dilakukan dengan operasi AND.

$$\begin{array}{r}
 1011\ 1100 \\
 1111\ 0011 \\
 \hline
 1011\ 0000
 \end{array}
 \quad \text{AND}$$

2. Sederhanakan pernyataan logika berikut:

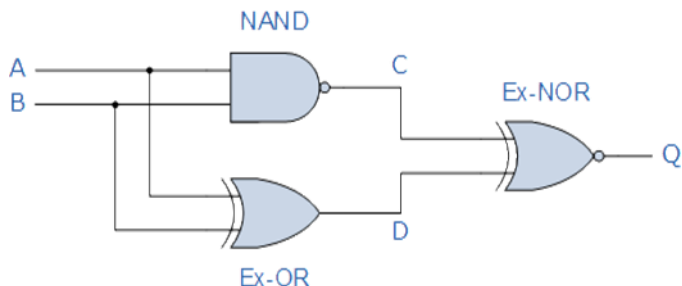
$$Y = \overline{A}BC + A\overline{B}C, Y = \overline{A}BCD + \overline{A}BC\overline{D} + A\overline{B}CD + A\overline{B}\overline{C}D, \text{ dan } Y = (\overline{A} + B)(A + B)$$

Jawab:

$$\begin{aligned}
 \blacksquare Y &= \overline{A}BC + A\overline{B}C \\
 &= (\overline{A} + A)BC && \text{(sifat distributif)} \\
 &= BC && \text{(sifat identitas ke-4)} \\
 \blacksquare Y &= \overline{A}BCD + \overline{A}BC\overline{D} + A\overline{B}CD + A\overline{B}\overline{C}D \\
 &= \overline{A}BC(D + \overline{D}) + A\overline{B}D(C + \overline{C}) && \text{(sifat distributif)} \\
 &= \overline{A}BC + A\overline{B}D && \text{(sifat identitas ke-4)} \\
 \blacksquare Y &= (\overline{A} + B)(A + B) \\
 &= \overline{A}A + \overline{A}B + BA + BB \\
 &= 0 + \overline{A}B + AB + B \\
 &= \overline{A}B + AB + B \\
 &= B(\overline{A} + A + 1) \\
 &= B
 \end{aligned}$$

webstudi.site

3. Buatlah Tabel Kebenaran untuk fungsi logika pada titik C, D dan Q pada rangkaian berikut serta identifikasi gerbang logika tunggal yang dapat digunakan untuk mengganti seluruh rangkaian tersebut.



Jawab :

Dari gambar rangkaian diatas dapat kita ketahui bawah terdapat 3 gerbang yaitu gerbang NAND 2-masukan, gerbang EX-OR 2-masukan dan terakhir gerbang EX-NOR 2-masukan pada bagian keluaran.

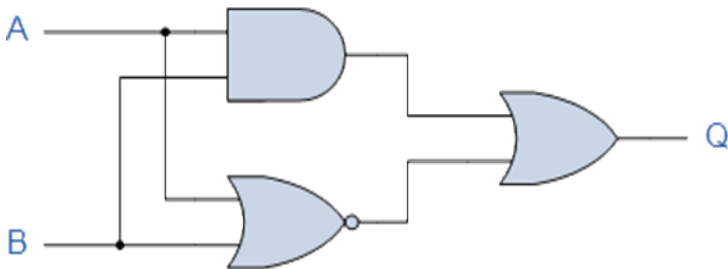
Karena hanya ada 2 masukan ke rangkaian A dan B, maka ada 4 kemungkinan kombinasi masukan (2²) yaitu : 0-0, 0-1, 1-0 dan 1-1. Berdasarkan kombinasi masukan tersebut maka akan dibuat tabel kebenaran.

Inputs		Output at		
A	B	C	D	Q
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	0	1

Dari tabel kebenaran di atas, kolom C mewakili fungsi keluaran yang dihasilkan oleh gerbang NAND, sedangkan kolom D mewakili fungsi keluaran dari gerbang Ex-OR. Dari kedua ekspresi keluaran ini kemudian menjadi kondisi masukan untuk gerbang Ex-NOR.

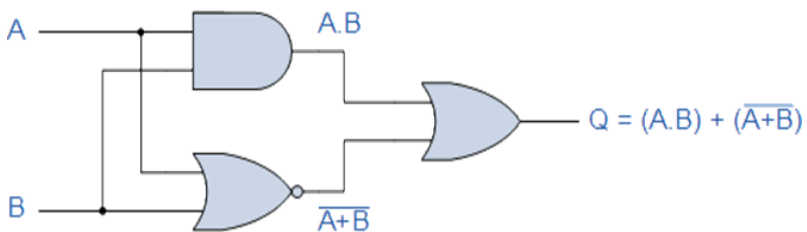
Dapat dilihat dari tabel kebenaran bahwa keluaran di Q akan bernilai 1 apabila salah satu dari dua masukan A atau B berada di logika 1. Satu-satunya tabel kebenaran yang memenuhi kondisi ini yaitu tabel kebenaran gerbang OR. Oleh karena itu, seluruh rangkaian di atas dapat diganti dengan gerbang OR 2 masukan.

4. Temukan ekspresi Aljabar Boolean untuk sistem berikut.



Jawab :

Untuk sistem diatas terdiri dari Gerbang AND, Gerbang NOR dan Gerbang OR, ekspresi untuk gerbang AND adalah $A.B$, dan ekspresi untuk gerbang NOR adalah $\overline{A+B}$. Dari kedua ekspresi ini juga memiliki masukan yang terpisah ke gerbang OR yang didefinisikan sebagai $A+B$. Jadi ekspresi keluaran akhir adalah sebagai berikut:

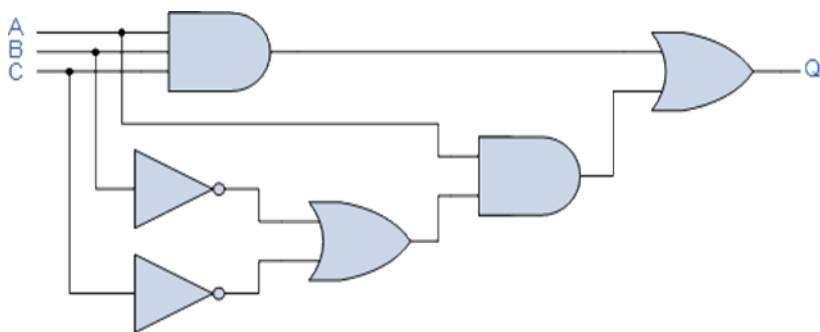


Keluaran pada Q yaitu $Q = (A.B) + (\overline{A+B})$, tetapi notasinya sama seperti notasi De'Morgan $\overline{A.B}$ maka notasi $\overline{A+B}$ diganti kedalam ekspresi keluaran sehingga dihasilkan notasi output $Q = (A.B) + \overline{A.B}$, dimana merupakan notasi Boolean untuk Gerbang Exclusive-NOR.

Inputs		Intermediates		Output
B	A	AB	$\overline{A+B}$	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Dengan kata lain, rangkaian diatas dapat diganti dengan gerbang Exclusive-NOR.

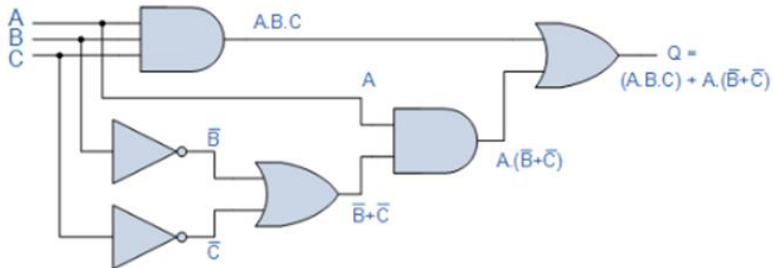
5. Temukan ekspresi Aljabar Boolean untuk sistem berikut



Jawab :

Pada sistem rangkaian diatas mungkin terlihat lebih rumit daripada dua rangkaian sebelumnya. Jika dianalisa maka rangkaian logika diatas hanya terdiri dari gerbang dasar AND, OR dan NOT yang dihubungkan bersama.

Seperti contoh soal Aljabar Boolean sebelumnya, kita dapat menyederhanakan rangkaian rangkaian dengan menuliskan notasi Boolean pada setiap fungsi gerbang logika secara bergantian, agar didapatkan ekspresi akhir untuk keluaran pada Q.



Keluaran dari gerbang AND 3-masukan yaitu logika "1" ketika semua masukan gerbang adalah "HIGH" pada level logika "1" ($A.B.C$). Keluaran dari gerbang OR yang lebih rendah hanya "1" ketika satu atau kedua masukan B atau C berada pada level logika "0".

Keluaran dari gerbang AND 2-masukan adalah "1" ketika masukan A adalah "1" dan masukan B atau C berada pada "0". Maka keluaran pada Q hanya "1" ketika masukan $A.B.C$ sama dengan "1" atau A sama dengan "1" dan keduanya masukan B atau C sama dengan "0", $A.(B + C)$.

Dengan menggunakan "teorema de Morgan", masukan B dan masukan C dapat diputus untuk menghasilkan keluaran pada Q, yang mana dapat berupa logika "1" atau pada logika "0". Maka satu-satunya masukan yaitu masukan A, sebagai satu-satunya masukan yang diperlukan untuk memberikan keluaran pada Q seperti yang ditunjukkan pada tabel di bawah ini.

Inputs			Intermediates					Output
C	B	A	A.B.C	$\overline{B}$	$\overline{C}$	$\overline{B}+\overline{C}$	$A.(\overline{B}+\overline{C})$	Q
0	0	0	0	1	1	1	0	0
0	0	1	0	1	1	1	1	1
0	1	0	0	0	1	1	0	0
0	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	0	0
1	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1

Berdasarkan rangkaian logika di atas maka dapat disimpulkan bahwa masukan pada rangkaian tersebut dapat diganti dengan hanya satu masukan tunggal yaitu masukan "A" sehingga akan mengurangi jumlah sirkutnya dengan hanya satu kawat, (atau Buffer).

Dengan menggunakan Aljabar Boolean maka melakukan analisis pada jenis rangkaian akan sangat cepat sehingga ketika dalam praktiknya kita dapat mengeliminasi gerbang logika yang tidak perlu dalam desain logika digital sehingga mengurangi jumlah gerbang yang diperlukan, konsumsi daya rangkaian serta biaya.

<https://www.webstudi.site/2019/02/aljabar-boolean.html>

FUNGSI BOOLEAN

Dalam Aljabar Boolean, variable x disebut peubah Boolean. Fungsi Boolean adalah ekspresi yang dibentuk dari peubah Boolean melalui operasi penjumlahan, perkalian, atau komplemen.

Contoh:

$$f(x) = x$$

$$f(x,y) = x'y + x$$

$$3. g(x,y,z) = (x + y)' + xyz'$$

Selain dengan cara aljabar, fungsi Boolean dapat dinyatakan dalam bentuk tabel kebenaran. Tabel kebenaran adalah suatu tabel yang menyatakan seluruh kemungkinan nilai peubah dari fungsinya. Jika suatu fungsi Boolean memuat n peubah, maka banyaknya baris dalam tabel kebenaran ada 2^n .

$$\text{Contoh: } f(x,y,z) = xyz' + x$$

x	y	z	$f(x,y,z) = xyz' + x$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Fungsi Boolean tidak unik (tunggal), artinya dua fungsi yang ekspresinya berbeda dikatakan sama jika keduanya mempunyai nilai yang sama pada tabel kebenaran untuk setiap kombinasi peubah-peubahnya.

$$\text{Contoh: } f(x,y,z) = x'y'z + x'yz + xy' \text{ dengan } g(x,y,z) = x'z + xy'$$

KOMPLEMEN FUNGSI

Ada dua cara untuk mencari komplemen suatu fungsi Boolean.

- Cara 1: Dengan menggunakan hukum De Morgan berulang-ulang
Hukum De Morgan yang diperluas:

$$(A_1 + A_2 + A_3 + \dots + A_n)' = A_1' \cdot A_2' \cdot A_3' \cdot \dots \cdot A_n'$$

$$(A_1 \cdot A_2 \cdot A_3 \cdot \dots \cdot A_n)' = A_1' + A_2' + A_3' + \dots + A_n'$$

Contoh: $f(x,y,z) = x(y'z' + yz)$, maka

$$f'(x,y,z) = (x(y'z' + yz))'$$

$$= x' + (y'z' + yz)'$$

$$= x' + (y'z')' (yz)'$$

$$= x' + (y + z) (y' + z')$$

- Cara 2: Menggunakan dual fungsi Langkah – langkah:

Tentukan dual f.

Komplemen dari f didapat dengan cara mengkomplemenkan setiap variabel yang ada pada dual f.

$$\text{Contoh: } f(x,y,z) = x(y'z' + yz) \text{ Dual } f = x + (y' + z') (y + z)$$

Maka $f' = x' + (y + z) (y' + z')$ (setiap variabel pada dual f dikomplemenkan)

F. BENTUK STANDAR FUNGSI BOOLEAN

Ada 2 bentuk standar fungsi Boolean:

Sum of Product (SOP) / Minterm

Product of Sum (POS) / Maxterm

Sum of Product Ciri – ciri:

Dalam setiap suku operasi variabelnya adalah perkalian.

Setiap suku (term) dijumlahkan.

Setiap suku mengandung semua variabel. Contoh:

$$f(x,y) = xy + x'y$$

$$g(x,y,z) = x'yz + xyz + x'y'z'$$

Cara membaca:

Variabel tanpa komplemen dianggap bernilai 1.

Variabel dengan komplemen dibaca 0.

Product of Sum Ciri – ciri:

Dalam setiap suku operasi variabelnya adalah penjumlahan.

Setiap suku (term) dikalikan.

Setiap suku mengandung semua variabel. Contoh:

$$f(x,y) = (x' + y) (x + y')$$

$$g(x,y,z) = (x + y + z') (x' + y' + z) (x + y + z)$$

Cara membaca:

Variabel tanpa komplemen dianggap bernilai 0.

Variabel dengan komplemen dibaca 1.

Minterm dan maxterm dengan 2 peubah:

x	y	Minterm		Maxterm	
		suku	simbol	suku	simbol
0	0	$x'y'$	m0	$x + y$	M0
0	1	$x'y$	m1	$x + y'$	M1
1	0	xy'	m2	$x' + y$	M2
1	1	xy	m3	$x' + y'$	M3

Minterm dan maxterm dengan 3 peubah:

x	y	z	Minterm		Maxterm	
			suku	simbol	suku	simbol
0	0	0	$x'y'z'$	m0	$x + y + z$	M0
0	0	1	$x'y'z$	m1	$x + y + z'$	M1
0	1	0	$x'yz'$	m2	$x + y' + z$	M2
0	1	1	$x'yz$	m3	$x + y' + z'$	M3

1	0	0	$xy'z'$	m4	$x' + y + z$	M4
1	0	1	$xy'z$	m5	$x' + y + z'$	M5
1	1	0	xyz'	m6	$x' + y' + z$	M6
1	1	1	xyz	m7	$x' + y' + z'$	M7

KONVERSI BENTUK FUNGSI

Konversi ke bentuk SOP Contoh1: $f(x,y) = x + x'y$

$= x.1 + x'y$ (lengkapi variabel disetiap suku dgn cara mengalikan dengan 1)

$= x(y + y') + x'y$ (ganti 1 dengan menggunakan hukum komplemen untuk memunculkan variabel yang belum ada)

$= xy + xy' + x'y$ (gunakan hukum distributif)

$= m3 + m2 + m1$

$= \Sigma(1,2,3)$

Contoh 2: $f(x,y,z) = x + y'z$

$= x.1.1 + 1.y'z$

$= x(y + y')(z + z') + (x + x')y'z$

$= x(yz + yz' + y'z + y'z') + xy'z + x'y'z$

$= xyz + xyz' + xy'z + xy'z' + x'y'z + x'y'z'$

$= m7 + m6 + m5 + m4 + m5 + m1$

$= \Sigma(1,4,5,6,7)$

Konversi ke bentuk POS Contoh 1 : $f(x,y) = x + x'y$

$= (x + x')(x + y)$ (gunakan hukum distributif untuk menghilangkan perkalian pada setiap suku)

$= 1. (x + y)$

$= (x + y)$

$= M0$

$= \Sigma(0)$

Contoh 2: $f(x,y,z) = x + y'z$
 $= (x + y') (x + z)$
 $= ((x + y') + 0) ((x+z) + 0)$ (hukum identitas)
 $= ((x + y') + zz') ((x+z) + yy')$ (hukum komplemen untuk memunculkan variabel yg belum ada)
 $= (x + y' + z) (x + y' + z') (x + y + z) (x + y' + z)$
 $= M2. M3. M0. M2$
 $= \Sigma(0,2,3)$

Dengan hukum De Morgan diperoleh bahwa:

$f'(x,y,z) = x'yz' + x'yz + x'y'z' + x'yz'$
 $= m2+m3+m0+m2$
 $= \Sigma(0,2,3)$

Kesimpulan:

$m_j' = M_j$

Tabel Kebenaran untuk mencari Bentuk SOP & POS

Perhatikan tabel kebenaran berikut:

x	y	z	F(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Cara membaca:

Bentuk SOP : baca baris bernilai '1'

Kombinasi peubahnya, tulis dalam bentuk minterm Jadi,

$$f(x,y,z) = x'y'z + xy'z' + xyz = \Sigma(1,4,7)$$

Bentuk POS : baca baris bernilai '0'

Kombinasi peubahnya, tulis dalam bentuk maxterm Jadi, $f(x,y,z) = (x$

$$+ y + z)(x + y' + z)(x + y' + z')(x' + y + z')$$

$$(x' + y' + z)$$

$$= (0,2,3,5,6)[7]$$



BUKU AJAR SISTEM DIGITAL UNTUK TEKNIK INFORMATIKA

BAB 8: KARNAUGH MAP

Sri Mulyati

Universitas Muhammdiyah Tangerang

BAB 8

KARNAUGH MAP

Peta Karnaugh digunakan untuk mempermudah penyederhanaan fungsi aljabar Boolean. Misalnya, perhatikan fungsi Boolean yang dijelaskan oleh tabel kebenaran berikut.

<i>Truth table of a function</i>					
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F(A,B,C,D)</i>
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0

8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Gambar 8. 1 Tabel Kebenaran Karnaugh Map

Berikut ini adalah dua notasi berbeda yang menjelaskan fungsi yang sama dalam aljabar Boolean tak disederhanakan, menggunakan variabel Boolean A, B, C, D dan inversnya.

$f(A, B, C, D) = \sum m_i, i \in \{6, 8, 9, 10, 11, 12, 13, 14\}$ di mana minterms untuk dipetakan (yaitu, baris yang memiliki keluaran 1 dalam tabel kebenaran).

$f(A, B, C, D) = \prod M_i, i \in \{0, 1, 2, 3, 4, 5, 7, 15\}$ di mana maxterms untuk dipetakan (yaitu, baris yang memiliki keluaran 0 dalam tabel kebenaran).

A. KARNAUGH MAP BESERTA PENJELASANNYA

Karnaugh Map atau K-Map adalah suatu teknik penyederhanaan fungsi logika dengan cara pemetaan. K-Map terdiri dari kotak-kotak yang jumlahnya terdiri dari jumlah variable dan fungsi logika atau jumlah masukan dari rangkaian logika yang sedang kita hitung.

Rumus untuk menentukan jumlah kotak pada K-Map adalah 2^n yang mana adalah banyaknya variabel / masukan.

B. LANGKAH – LANGKAH PEMETAAN K-MAP SECARA UMUM :

- Menyusun aljabar Boolean terlebih dahulu
- Menggambar rangkaian digital
- Membuat Table Kebenarannya
- Merumuskan Tabel Kebenarannya
- Lalu memasukkan rumus Tabel Kebenaran ke K-Map (Kotak-kotak)

C. PENYEDERHANAAN DUA VARIABEL

Contoh 1.

$$F = AB + A'B + AB'$$

Penyelesaian:

Gambarkan K-Map Model-1 untuk dua variabel

Ganti kotak-kotak yang sesuai untuk AB, A'B, dan AB, dengan angka satu (1) dan sisanya dengan angka nol (0)

		A		0	1
B	0	A'B'	AB'		
	1	A'B	AB		

		A		0	1
B	0	0	1		
	1	1	1		

Gambar 8. 2 Dua Variabel

Gabungkan semua angka satu (1) sesederhana mungkin. Untuk mempermudah dapat menggunakan pemetaan K-Map dua variabel.

The diagram illustrates the construction of a 2x2 matrix for the XOR operation. The first matrix shows the headers A (0, 1) and B (0, 1) and is empty. The second matrix shows the same headers but contains the values 0, 1, 1, 1 in the cells, with red boxes highlighting the 1s. Red arrows indicate the placement of A and B.

		A		0	1
B	0				
	1				

		A		0	1
B	0	0	1		
	1	1	1		

LINKSUKSES.COM

Gambar 8. 3 K-Map dua variabel.

Hasil penyederhanaan dari $F = AB + A'B + AB'$ adalah $F = A + B$
 Perbandingan dengan cara Aljabar: $F = AB + A'B + AB' = A(B+B') + A'B = A(1) + A'B = A + A'B = A + B$ (Teorema T9)

Contoh 2.

$$F = AB' + A'B'$$

Penyelesaian:

Gambarkan K-Map Model-1 untuk dua variabel

Ganti kotak-kotak yang sesuai untuk AB' dan $A'B'$ dengan angka satu (1) dan sisanya dengan angka nol (0)

		A	
		0	1
B	0	$A'B'$	AB'
	1	$A'B$	AB

		A	
		0	1
B	0	1	1
	1	0	0

Gambar 8. 4 K-Map Model-1 untuk dua variabel

Gabungkan semua angka satu (1) sesederhana mungkin. Untuk mempermudah dapat menggunakan pemetaan K-Map dua variabel.

		A	
		0	1
B	0		
	1		

← B'

		A	
		0	1
B	0	1	1
	1	0	0

Gambar 8. 5 pemetaan K-Map dua variabel

Hasil penyederhanaan adalah $F = B'$

Perbandingan dengan cara Aljabar:

$$F = AB' + A'B'$$

$$F = (A+A') B'$$

$$F = (1) B'$$

$$F = B'$$

PENYEDERHANAAN TIGA VARIABEL

Contoh 1.

$$F = ABC' + AB'C' + AB'C + ABC$$

Penyelesaian:

Gambarkan K-Map Model-1 untuk 3 variabel

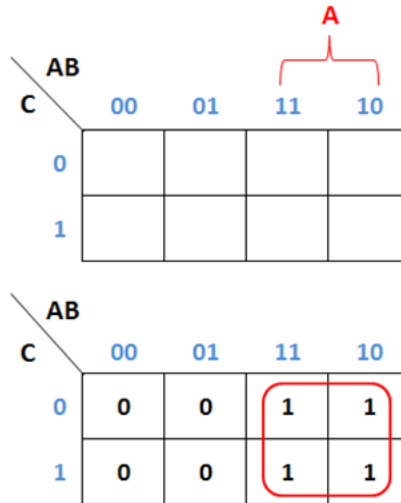
kemudian tandai dengan angka satu (1) setiap kotak yang mewakili ABC' , $AB'C'$, $AB'C$, dan ABC , sisanya diisi dengan angka nol (0).

		AB			
		00	01	11	10
C	0	$A'B'C'$	$A'BC'$	ABC'	$AB'C'$
	1	$A'B'C$	$A'BC$	ABC	$AB'C$

		AB			
		00	01	11	10
C	0	0	0	1	1
	1	0	0	1	1

Gambar 8. 6 K-Map Model-1 untuk 3 variabel

Gambarkan pemetaan K-Map untuk 3 variabel yang paling mendekati dan paling sederhana. Pada kasus ini area A pada K-Map dapat mewakili semua variabel dalam soal.



Gambar 8. 7 K-Map Model-1 untuk 3 variabel

Perbandingan dengan Aljabar:

$$F = ABC' + AB'C' + AB'C + ABC$$

$$F = AB (C' + C) + AB' (C' + C)$$

$$F = AB (1) + AB' (1)$$

$$F = AB + AB'$$

$$F = A (B + B')$$

$$F = A (1) = A$$

Contoh 2.

$$F = A'B'C + A'BC + AB'C + ABC + ABC'$$

Penyelesaian

Gambarkan K-Map Model-1 untuk 3 variabel

kemudian tandai dengan angka satu (1) setiap kotak yang mewakili $A'B'C$, $A'BC$, $AB'C$, dan ABC , dan ABC' , sisanya diisi dengan angka nol (0).

AB					
C		00	01	11	10
	0	$A'B'C'$	$A'BC'$	ABC'	$AB'C'$
	1	$A'B'C$	$A'BC$	ABC	$AB'C$

AB					
C		00	01	11	10
	0	0	0	1	0
	1	1	1	1	1

Gambar 8. 8 K-Map Model-1 untuk 3 variabel

Gambarkan pemetaan K-Map untuk 3 variabel yang paling mendekati dan paling sederhana. Pada kasus ini area AB dan area C pada K-Map dapat mewakili semua variabel dalam soal.

AB					
C		00	01	11	10
	0				
	1				

AB					
C		00	01	11	10
	0	0	0	1	0
	1	1	1	1	1

Gambar 8. 9 K-Map untuk 3 variabel

Penyederhanaan dari

$$F = A'B'C + A'BC + AB'C + ABC + ABC' \text{ adalah } F = AB + C$$

Perbandingan dengan Aljabar:

$$F = A'B'C + A'BC + AB'C + ABC + ABC'$$

$$F = (A'+A)B'C + (A'+A)BC + ABC'$$

$$F = (1) B'C + (1) BC + ABC' = B'C + BC + ABC'$$

$$F = (B'+B)C + ABC'$$

$$F = (1) C + ABC'$$

$$F = C + ABC'$$

$$F = C + AB \text{ (Teorema T9)}$$

$$F = AB + C$$

PENYEDERHANAAN EMPAT VARIABEL

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

Penyelsaian

Gambarkan K-Map Model-1 untuk 4 variabel

kemudian tandai dengan angka satu (1) setiap kotak yang mewakili $A'BC'D$, $ABC'D$, $A'BCD$, $ABCD$, sisanya diisi dengan angka nol (0).

		AB			
CD \		00	01	11	10
	00	$A'B'C'D'$	$A'BC'D'$	$ABC'D'$	$AB'C'D'$
	01	$A'BC'D$	$A'BC'D$	$ABC'D$	$AB'C'D$
	11	$A'B'CD$	$A'BCD$	$ABCD$	$AB'CD$
	10	$A'B'CD'$	$A'BCD'$	$ABCD'$	$AB'CD'$

		AB			
CD \		00	01	11	10
	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

Gambar 8. 10 K-Map Model-1 untuk 4 variabel

Hasil penyederhanaan dari

$$F = A'BC'D + ABC'D + A'BCD + ABCD \text{ adalah } F = BD$$

Perbandingan dengan Aljabar:

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

$$= (A'+A) BC'D + (A'+A) BCD$$

$$= (1) BC'D + (1) BCD$$

$$= BC'D + BCD$$

$$= BD (C'+C)$$

$$= BD (1)$$

$$= BD [4]$$

Catatan : Bar = '

Tabel dari K-Map 2 variabel adalah seperti dibawah ini

A \ B	0	1
0		
1		

Gambar 8. 11 Tabel dari K-Map 2 variabel

Contoh Soal :

$$H = AB + A'B + AB'$$

Maka cara pengerjaannya seperti dibawah ini

A \ B	0	1
0		A'B
1	AB'	AB

Gambar 8. 12 Tabel dari K-Map 2 variabel

Bar (') atau aksen biasanya ditulis kedalam angka 0 sedangkan angka 1 adalah tanpa Bar aksen.

Dan dapat dipermudah lagi menjadi dibawah ini :

A \ B		0	1
0			1
1	1		1

Gambar 8. 13 Tabel dari K-Map 2 variabel

Yang dapat disederhanakan dalam K-Map hanya 2 / kelipatan 2 dari kotak yang berdempetan dan sedangkan jika seperti kotak diatas maka penyederhanaannya:

A \ B		0	1
0			1
1	1		1

Gambar 8. 14 Tabel dari K-Map 2 variabel

Karena kolom ber angka 1 dan baris ber angka 1 memenuhi setiap garisnya, maka dapat disimpulkan kalau $H = AB + A'B + AB'$ K-Map nya adalah AB/BA

D. PENYEDERHANAAN TIGA VARIABEL

Catatan : Bar = '

Tabel dari K-Map 3 variabel adalah seperti dibawah ini

A \ BC				
	00	01	11	10
0				
1				

Gambar 8. 15 Tiga Variabel

Contoh Soal

$$H = ABC + A'BC + A'B'C + AB'C$$

Maka cara pengerjaannya seperti dibawah ini

A \ BC				
	00	01	11	10
0		A'B'C	A'BC	
1		AB'C	ABC	

Gambar 8. 16 Tiga Variabel

Dan dapat dipermudah lagi menjadi dibawah ini

A \ BC				
	00	01	11	10
0		1	1	
1		1	1	

Gambar 8. 17 Tiga Variabel

Sekarang kita lihat, karena yang memenuhi setiap kotaknya adalah baris 01 dan 11 sedangkan simbol 01 artinya adalah ($B'C$) dan 11 artinya adalah (BC) dan simbol yang tidak ada aksennya hanya C , maka $H = ABC + A'BC + A'B'C + AB'C$ adalah C .

E. PENYEDERHANAAN EMPAT VARIABEL

Catatan : Bar = '

Tabel dari K-Map 4 variabel adalah seperti dibawah ini :

AB \ CD		CD			
		00	01	11	10
AB	00				
	01				
	11				
	10				

Gambar 8. 18 K-Map 4 variabel

Contoh Soal

$$H = ABCD + ABCD' + AB'CD + ABC'D'$$

Maka cara pengerjaannya seperti dibawah ini

AB \ CD		CD			
		00	01	11	10
AB	00				
	01				
	11	$ABC'D'$		$ABCD$	$ABCD'$
	10			$AB'CD$	

Gambar 8. 19 K-Map 4 variabel

Dan dapat dipermudah lagi menjadi dibawah ini :

CD \ AB		CD			
		00	01	11	10
AB	00				
	01				
	11	1		1	1
	10			1	

Gambar 8. 20 K-Map 4 variabel

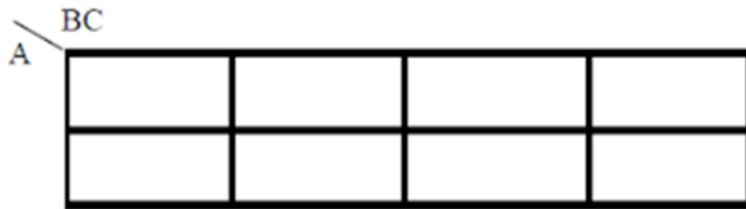
Karena yang ada angka 1 nya ada di kolom dan baris 1100, 1111, 1110 dan 1011, yaitu AB, ABCD, ABC dan ACD maka jika kita eliminasi dengan cara mengambil huruf yang sama saja menjadi $AB + ABC + ACD$.

<https://helmifadhiel.wordpress.com/2015/11/16/karnaugh-map-beserta-penjelasan/>

Karnaugh map (disingkat K-Map) adalah suatu metode untuk menjelaskan beberapa hal tentang penghitung aljabar boolean, metode ini telah ditemukan oleh Maurice Karnaugh pada tahun 1953. Karnaugh map ini sering digunakan untuk perhitungan yang menghitung sistem pola pikir manusia dengan hal-hal yang menguntungkan (sistem pemetaan peluang). Karnaugh Map atau K-Map adalah suatu teknik penyederhanaan fungsi logika dengan cara pemetaan. K-Map terdiri dari kotak-kotak yang jumlahnya terdiri dari jumlah variable dan fungsi logika atau jumlah masukan dari rangkaian logika yang sedang kita hitung.

Karnaugh Map (disingkat K-map) adalah sebuah peralatan grafis yang digunakan untuk menyederhanakan persamaan logika atau mengkonversikan sebuah Tabel Kebenaran menjadi sebuah rangkaian Logika. Blok diagram sebuah K-map seperti gambar 6.1 di bawah ini. AB dan C adalah variabel masukan, keluaran-keluaran berupa minterm-

minterm bernilai 1 diisikan pada sel K-map. Jumlah sel K-map adalah 2 jumlah variabel masukan.



Gambar 8. 21 Model K-map tiga variabel

F. LANGKAH-LANGKAH DALAM MENGGUNAKAN K-MAP ADALAH SEBAGAI BERIKUT

Logika Boolean Setelah cukup memahami teorema aljabar Boolean, penyederhanaan fungsi Boolean dengan aljabar, dan model-model Karnaugh Map beserta pemetaannya, kini saatnya mencoba menyelesaikan fungsi logika Boolean dengan Peta Karnaugh (Karnaugh Map/ K-Map). Jika suatu fungsi logika memiliki tiga atau empat variabel, maka penyelesaian dengan K-Map ini akan lebih mudah dibanding dengan penyederhanaan cara Aljabar. Dari beberapa model K-Map yang telah dibahas sebelumnya, penyederhanaan fungsi logika pada posting ini hanya akan menggunakan model-1 karena metode penyederhanaan dengan model-model K-Map lain pun hasilnya akan tetap sama.

Konversikan persamaan Boolean yang diketahui ke dalam bentuk persamaan SOP-nya (Sum of Product). Gunakan Tabel Kebenaran sebagai alat bantu. Gambarlah K-map, dengan jumlah sel = 2 jumlah variabel masukan. Isi sel K-map sesuai dengan minterm pada Tabel Kebenaran. Cover minterm-minterm bernilai 1 yang berdekatan, dengan aturan :

- hanya minterm berdekatan secara vertikal atau horizontal yang boleh di-cover.
- Jumlah minterm berdekatan yang boleh di-cover adalah : 2, 4, 8, 16, 32
- Buat persamaan SOP baru sesuai dengan hasil peng-cover-an minterm.
- Dari persamaan SOP yang didapatkan, bisa digambarkan rangkaian hasil penyederhanaannya.

Contoh :

Dari persamaan berikut ini, buatlah rangkaian penyederhanaannya.

$$X = ABC' + A'B + A'B'$$
 (Gambar 6.1)

Persamaan di atas dipetakan ke dalam K-Map menjadi seperti Gambar 6.2.

A \ BC				
	00	01	11	10
0	1	1	1	1
1	0	0	0	1

Gambar 8. 22 Hasil Pemetaan

Setelah dilakukan peng-cover-an minterm-minterm yang berdekatan :

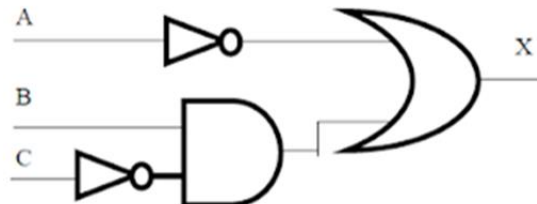
A \ BC				
	00	01	11	10
0	1	1	1	1
1	0	0	0	1

Gambar 8. 23 Setelah dilakukan peng-cover an mainterm

Dari hasil peng-cover-an didapatkan persamaan SOP :

$$X = A' + BC'$$

Rangkaian penyederhanaannya adalah seperti pada gambar 5-4



Gambar 8. 24 Rangkaian penyederhanaan [13]

berikut merupakan Langkah – langkah pemetaan K-Map secara umum :

- Menyusun aljabar Boolean terlebih dahulu
- Menggambar rangkaian digital
- Membuat Table Kebenarannya
- Merumuskan Tabel Kebenarannya
- Lalu memasukkan rumus Tabel Kebenaran ke K-Map (Kotak-kotak)

A \ BC				
	00	01	11	10
0		$A'B'C$	$A'BC$	
1		$AB'C$	ABC	

Dan dapat dipermudah lagi menjadi dibawah ini

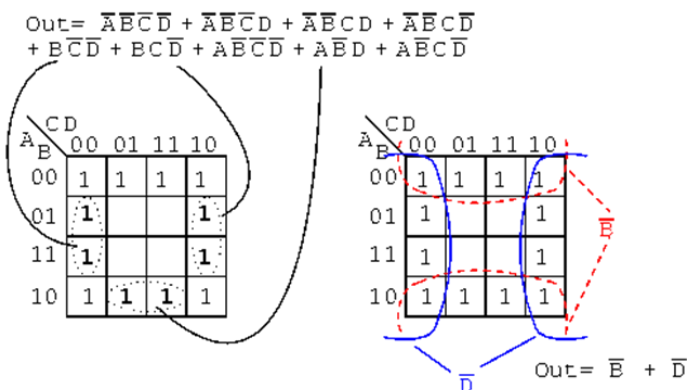
A \ BC				
	00	01	11	10
0		1	1	
1		1	1	

Gambar 8. 25 Tabel Kebenaran ke K-Map (Kotak-kotak)

Jenis-Jenis K-Map

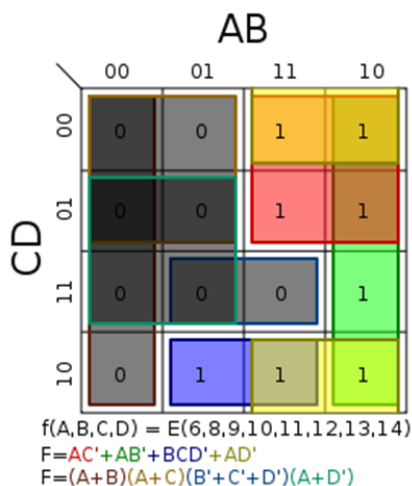
- K-Map 2 variabel
- K-Map 3 variabel
- K-Map 4 variabel
- K-Map 5 variabel
- K-Map 6 variabel

Salah satu contoh penerapan dari K-Maps dalam dunia aljabar Boolean adalah:



Gambar 8. 26 Penyelesaian menggunakan K-Maps [5]

Seperti gambar dibawah ini adalah sistem pemetaan pada bilang aljabar boolean :



Gambar 8. 27 Sistem pemetaan pada karnaugh map

pada gambar pemetaan diatas, variabel dari aljabar boolean ditransfer berdasarkan variabelnya masing-masing, dimana terjadi sistem perubahan pada beberapa kotak sehingga menghasilkan sebuah rumus 2^n dengan n adalah banyaknya kotak (1,2,3,4,...).

(Peta Karnaugh) K-Map merupakan penyederhanaan persamaan logika yang lebih sederhana dengan cara pemetaan yang terdapat kotak - kotak atau bujur sangkar yang jumlahnya tergantung dari banyaknya masukan dari rangkaian logikanya.

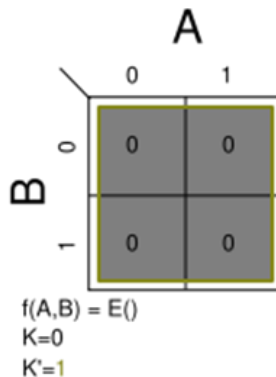
G. RUMUS MENENTUKAN JUMLAH KOTAK DALAM K-MAP

$N = 2$ dimana N = jumlah kotak dalam K-Map

N = banyaknya variabel/masukan

sebenarnya banyak sekali macam - macam k-map dari kmap dengan 2 variabel , 3 variabel dan 4 variabel. kita bisa melihat contoh k-map 2 variabel dibawah ini :

Jika terdapat 2 masukan variabel (X,Y atau A,B) dan 1 keluaran (z) maka . Dibawah sini ada beberapa sistem penghitungan aljabar boolean dengan menggunakan karnaugh map diantaranya :



Gambar 8. 28 $\Sigma (0)$; $K = 0$

		A	
		0	1
B	0	1	1
	1	1	1

$$f(A,B) = E(1,2,3,4)$$

$$K=1$$

$$K'=0$$

Gambar 8. 29 $\Sigma(1,2,3,4)$; $K = 1$

		A	
		0	1
B	0	1	0
	1	0	1

$$f(A,B) = E(1,4)$$

$$K = +A'B'AB$$

$$K' = +AB'A'B$$

Gambar 8. 30 $\Sigma(1,4)$; $K = A' B' + AB$

		A	
		0	1
B	0	1	0
	1	0	0

$$f(A,B) = E(1)$$

$$K = A'B'$$

$$K' = A+B$$

Gambar 8. 31 $\Sigma(1)$; $K = A' B'$

		A	
		0	1
B	0	0	1
	1	1	1

$$f(A,B) = \Sigma(2,3,4)$$

$$K = +AB$$

$$K' = A'B'$$

Gambar 8. 32 $\Sigma(2,3,4)$; $K = A + B$ [11, 12]

DAFTAR PUSTAKA

<https://teknikelektronika.com/pengertian-flip-flop-jenis-flip-flop/> diakses
tgl 2 Nov 2021 jam 5:47 WIB

<https://www.javatpoint.com/master-slave-jk-flip-flop-in-digital-electronics>

<https://dasarelelektronika2.blogspot.com/2014/02/gerbang-nor.html>

Source: [https://www.linksukses.com/2012/11/logika-boolean-karnaugh-
map.html](https://www.linksukses.com/2012/11/logika-boolean-karnaugh-map.html)

<https://binus.ac.id/bandung/2019/12/metode-k-maps/>

[http://staff.unila.ac.id/junaidi/files/2013/06/MULTIFLEXER-DAN-
DEMULTIFLEXER.pdf](http://staff.unila.ac.id/junaidi/files/2013/06/MULTIFLEXER-DAN-DEMULTIFLEXER.pdf)

[http://file.upi.edu/Direktori/FPMIPA/JUR._PEND._MATEMATIKA/KHUSNU
L_NOVIANIGSIH/FUNGSI_BOOLEAN.pdf](http://file.upi.edu/Direktori/FPMIPA/JUR._PEND._MATEMATIKA/KHUSNU_L_NOVIANIGSIH/FUNGSI_BOOLEAN.pdf)

<https://www.youtube.com/watch?v=kQv3PSGhEVo>

<https://www.javatpoint.com/d-flip-flop-in-digital-electronics>

<https://www.electronicshub.org/t-flip-flop/>

<http://berbagiilmuprogram.blogspot.com/2011/12/karnaugh-map.html>

[https://rangkaianlogikadasar.blogspot.com/2016/04/penyederhanaan-
dengan-karnaugh-map.html](https://rangkaianlogikadasar.blogspot.com/2016/04/penyederhanaan-dengan-karnaugh-map.html)

<https://nociepsikeluk.blogspot.com/2013/10/karnaugh-map-k-map.html>

digital<https://www.youtube.com/watch?v=EWj2TkiJ9W8>

[https://matakuliah45.blogspot.com/2015/02/pengantar-sistem-digital-
perbedaan.html](https://matakuliah45.blogspot.com/2015/02/pengantar-sistem-digital-perbedaan.html)

<https://www.electronicshub.org/demultiplexerdemux/>

<https://www.youtube.com/watch?v=HleQhZ9Gq5s>

INDEX

analog, 4

AND, 1, 3, 7, 15, 17, 18, 24, 25, 26, 27, 34, 42, 51, 53, 55, 57, 71, 77, 78, 81, 83, 84, 86, 88

Biner, 1, 36, 37, 38, 39, 40

Boolean, 2, 1, 2, 3, 15, 18, 20, 25, 26, 28, 29, 31, 33, 42, 43, 44, 46, 53, 54, 55, 57, 59, 60, 61, 77, 78, 79, 80, 81, 82, 83, 84, 86, 87, 88, 89, 90, 91, 92, 97, 98, 108, 110, 111

CMOS, 16

cut off, 12

D flip flop, 68, 69, 70, 74

Demultiplexer, 2, 45, 46, 47, 49, 52, 53, 54

Desimal, 1, 36, 38, 39, 40

digital, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 36, 41, 42, 43, 46, 50, 52, 63, 66, 68, 70, 73, 78, 79, 89, 98, 110, 114

Flip Flop, 2, 3, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75

Gelombang, 1, 4, 5

J-K Flip-Flop, 70

Konversi Bilangan, 1, 36

LED, 1, 10, 13, 17, 18, 19, 21, 22, 24, 25, 27, 28, 29, 30, 32, 33

Multiplexer, 1, 2, 42, 43, 45, 46, 50, 51, 52

NAND, 1, 2, 1, 15, 24, 25, 26, 34, 64, 65, 73, 75, 85, 86

NOR, 1, 2, 15, 25, 27, 28, 29, 31, 32, 33, 34, 65, 71, 85, 86, 87

NOT, 1, 15, 22, 23, 24, 25, 26, 28, 34, 42, 53, 55, 57, 67, 77, 78, 81, 83, 88

Oktal, 1, 36, 37, 38, 39, 40

OR, 1, 2, 3, 15, 18, 19, 20, 21, 28, 29, 30, 31, 32, 34, 42, 51, 77, 78, 81, 83,
84, 85, 86, 88

reverse, 1, 11, 12

RTL, 16

sistem, 1, 2, 3, 4, 8, 10, 13, 14, 15, 16, 36, 37, 41, 52, 61, 63, 66, 68, 70, 78,
79, 86, 87, 88, 108, 111, 112, 114

T Flip Flop, 70, 71, 72, 73

TTL, 2, 16, 46

PROFIL PENULIS

Ir. Sri Mulyati, S.Kom, M.Kom



Nama lengkap penulis adalah Ir. Sri Mulyati, S.Kom. Penulis lahir di Jakarta pada tanggal 25 Agustus 1965. Penulis adalah anak dari pasangan K.Marto Sudiro dan Winarti. Saat ini penulis sudah menikah dengan Dr. Ir. Sumardi Sadi, SPd., ST., MT. dan memiliki 3 anak yang bernama Dr. Nabila Octalia Fajarini, M. Arsy Kamal Faadhil dan Denafa Diena Sabrina. Pendidikan penulis S1 di Universitas Brawijaya Malang, S1 yang lainnya yaitu di STIMIK Pamitran Krawang dan melanjutkan pendidikan S2 di Budi Luhur Jakarta. Saat ini penulis adalah Dosen Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Tangerang.



BUKU AJAR

SISTEM DIGITAL

UNTUK TEKNIK INFORMATIKA

Istilah digital telah menjadi bagian dari perbendaharaan kata kita sehari-hari. Sistem digital telah menjadi sedemikian luas hampir semua bidang kehidupan, dari komputer, piranti otomatis, robot, ilmu dan teknologi kedokteran sampai kepada transportasi, hiburan, penjelajah ruang angkasa dan banyak lagi. Sistem Digital adalah suatu sistem yang berfungsi untuk mengukur suatu nilai yang bersifat tetap atau tidak teratur dalam bentuk diskrip berupa digit-digit atau angka-angka yang bersifat elektronika, sistem elektronika yang setiap rangkaian penyusunnya melakukan pengolahan sinyal diskrit. Selain itu Sistem Digital juga terdiri dari beberapa rangkaian digital ataupun logika, komponen elektroika, dan elemen gerbang logika untuk suatu tujuan pengalihan tenaga maupun energi.

Buku ini merupakan sebuah bentuk sumbangsih keilmuan yang dihasilkan dari bauh pikir para pakar terkemuka dibidangnya yang mampu memberikan perubahan sikap dan mental pada tatanan keilmuan bidang informatika khususnya. Oleh sebab itu buku ini hadir dihadapan sidang pembaca sebagai bagian dari upaya diskusi sekaligus dalam rangka melengkapi khazanah keilmuan dibidang teknik informatika, sehingga buku ini sangat cocok untuk dijadikan bahan acuan bagi kalangan intelektual dilngkungan perguruan tinggi ataupun praktisi yang berkecimpung langsung dibidang teknik informatika.