

Diktat Mata Kuliah

PERANCANGAN SISTEM DIGITAL

Sujono, ST. MT.

Program Studi Teknik Elektro
Fakultas Teknik
Universitas Budi Luhur
Juli 2011



KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas segala limpahan karunia-Nya, sehingga penyusunan buku bahan ajar mata kuliah Sistem Digital ini dapat terselesaikan.

Terimakasih kami sampaikan kepada semua pihak yang telah membantu dalam penyelesaian penyusunan diktat ini, banyak masukan yang kami ambil dari berbagai artikel yang kami dapatkan di internet, sekali lagi terima kasih kepada para penulis artikel tersebut yang tidak dapat kami sebutkan satu-persatu.

Dalam pembahasan materi, diawali dengan pengenalan konsep dasar teknik digital, sistem bilangan, gerbang logika. Flip-flop dan aplikasinya juga dibahas sebagai salah satu bagian penting dalam teknik digital. Pada bagian akhir dari buku ini dibahas tentang counter dan shift register serta aplikasinya dalam sistem digital.

Kami sadar bahwa penyusunan buku ajar ini masih jauh dari sempurna. Akhir kata semoga buku ajar ini bisa bermanfaat dalam menunjang mahasiswa dalam menempuh studinya, khususnya dalam mempelajari tentang analisa rangkaian listrik.

Jakarta, Agustus 2011

Sujono, ST. MT.

DAFTAR ISI

Kata Pengantar	i
Daftar Isi	ii
Daftar Gambar	vii
Daftar Tabel	x
Bab I PENGANTAR KONSEP DIGITAL.....	1
1.1. Representasi Numerik.....	2
1.1.1. Representasi Analog	2
1.1.2. Representasi Digital	2
1.2. Keuntungan dan Keterbatasan Teknik DIGITAL.....	4
12.1. Keuntungan Teknik Digital	4
1.2.2. Keterbatasan Teknik Digital	4
1.3. Sistem Bilangan Digital (Digital Number System)	6
1.3.1. Sistem Bilangan Desimal (Decimal System).....	6
1.3.2. Sistem Bilangan Biner (Binary System).....	6
1.4. Representasi Kuantitas Biner	8
1.5. Quis 1	9
Bab II SISTEM BILANGAN DAN KODE	10
2.1. Konversi Bilangan Biner ke Bilangan Desimal (<i>Binary-To-Decimal Conversion</i>)	11
2.2. Konversi Bilangan Desimal ke Bilangan Biner (<i>Decimal-To-Binary Conversion</i>)	12
2.2.1. Metode Penguraian (<i>Revese of Binary-To-Digital Method</i>)	12
2.2.2. Metode Pembagian Secara Berulang (<i>Repeat Division Methode</i>)	12
2.3. Sistem Bilangan Oktal (Octal Number System)	14
2.3.1. Konversi bilangan octal ke decimal (<i>Octal to Decimal Conversion</i>)	14

2.3.2. Konversi Bilangan Biner ke Bilangan Oktal dan Oktal ke Biner	14
2.3.3. Konversi Bilangan Desimal ke Oktal Dengan Pembagian Berulang (<i>Repeat Division</i>)	15
2.4. Sistem Bilangan Heksadesimal (Hexadecimal Number System)	15
2.4.1. Konversi Heksadesimal ke Desimal (<i>Hexadecimal to Decimal Conversion</i>)	15
2.4.2. Konversi Bilangan Desimal ke Bilangan Heksadesimal dengan Pembagian Berulang (<i>Repeat Division</i>)	16
2.4.3. Konversi Bil. Biner ke Bil. Heksa. dan sebaliknya (<i>Binary-To-Hexa/Hexa-To-Binary Conversion</i>)	16
2.4.4. Konversi Bil. Oktal ke Bil. Heksa. dan Sebaliknya (<i>Octal-To-Hexa. /Hexa.-To-Octal Conversion</i>)	17
2.5. Quis 2	19
Bab III GERBANG LOGIKA (<i>Logic Gates</i>)	21
3.1. Variabel Boolean dan Tabel Kebenaran	22
3.2. Operasi OR (<i>OR Operation</i>)	23
3.4. Operasi NOT (<i>NOT Operation</i>)	26
3.5. Operasi NOT OR (<i>NOR Operation</i>)	27
3.6. Operasi NAND (<i>NAND Operation</i>)	28
3.7. Quis 3	29
Bab IV ALJABAR BOOLEAN	32
4.1. Menyatakan Rangkaian Logika Secara Aljabar Boolean (Describing Logic Circuits Algebraically)	33
4.2. Analisa Keluaran Rangkaian Logika (Evaluating Logic Circuit Outputs)	34
4.3. Implementasi Rangkaian Gerbang Logika terhadap Ekpresi Boolean (Implementing Circuits From Boolean Expression)	35
4.4. Teorema Boolean (Boolean Theorems)	37

4.5.	Teorema DeMorgans (DeMorgan's Theorem)	38
4.5.1.	Teorema DeMorgan 3 Variabel (Three Variables DeMorgan's Theorem)	38
4.5.2.	Teorema Implikasi DeMorgans (Implications of DeMorgan's Theorem)	39
4.6.	Penyeragaman Gerbang NAND-NOR (Universality of NAND & NOR Gates)	39
4.7.	Kesamaan Representasi Gerbang Logika (Alternate Logic Gate Representations)	40
4.8.	Interpretasi Simbol Logika (Logic Symbol Interpretation)	42
4.9.	Quis 4	43
Bab V	FLIP-FLOP	46
5.1.	Pendahuluan	47
5.2.	Edge-Triggered Flip-flops	47
5.2.1.	Edge-triggered S-R flip-flop	48
5.2.2.	Edge-triggered J-K flip-flop	49
5.2.3.	Edge-triggered D flip-flop	50
5.3.	Pulse-Triggered (Master-Slave) Flip-flops	51
5.4.	Data Lock-Out Flip-flops	52
5.5.	Applications Flip-Flop	53
5.5.1.	Frequency Division	53
5.5.2.	Parallel Data Storage	54
5.5.3.	Counting	54
Bab VI	COUNTER	56
6.1.	Pendahuluan	57
6.2.	Asynchronous(Ripple) Counters	57
6.3.	Asynchronous Decade Counters	59
6.4.	Asynchronous Up-Down Counters	61
6.5.	Synchronous Counters	62
6.6.	Synchronous Decade Counters	63

6.7. Synchronous Up-Down Counters	64
6.8. Aplikasi	65
Bab VII REGISTER GESER (SHIFT REGISTER)	67
7.1. Pendahuluan	68
7.2. Serial In - Serial Out Shift Registers	68
7.3. Serial In - Parallel Out Shift Registers	70
7.4. Parallel In - Serial Out Shift Registers	70
7.5. Parallel In - Parallel Out Shift Registers	71
7.6. Bidirectional Shift Registers	72
7.7. Shift Register Counters	73
7.7.1. Ring Counters	73
7.7.2. Johnson Counters	74
7.8. Aplikasi Shift Register	74

Daftar Pustaka

DAFTAR GAMBAR

<i>Gambar 1.1. Diagram Sinyal Tegangan Analog Terhadap Waktu</i>	<i>2</i>
<i>Gambar 1.2. Diagram Sinyal Tegangan Digital Terhadap Waktu</i>	<i>3</i>
<i>Gambar 1.3. Diagram Blok Proses Yang Dilengkapi Dengan Konversi Analog-Digital-Analog</i>	<i>5</i>
<i>Gambar 1.4. Tingkat Tegangan Untuk Tiap Nilai Logika Digital (LOW dan HIGH)</i>	<i>8</i>
<i>Gambar 2.1. Diagram alir metode pembagian berulang</i>	<i>13</i>
<i>Gambar 3.1. Nilai Variabel Boolean</i>	<i>22</i>
<i>Gambar 3.2. Gerbang Logika dengan 2 masukan dan tabel kebenarannya</i>	<i>23</i>
<i>Gambar 3.3. Gerbang Logika OR dengan 2 masukan dan tabel kebenarannya</i>	<i>24</i>
<i>Gambar 3.4. Gerbang Logika OR dengan 3 masukan dan tabel kebenarannya</i>	<i>24</i>
<i>Gambar 3.5. Gerbang Logika AND dengan 2 masukan dan tabel kebenarannya</i>	<i>25</i>
<i>Gambar 3.6. Gerbang Logika AND dengan 3 masukan dan tabel kebenarannya</i>	<i>25</i>
<i>Gambar 3.7. Gerbang Logika NAND dengan 2 masukan dan tabel kebenarannya</i>	<i>38</i>
<i>Gambar 3.8. Gerbang Logika NOR 2 masukan dan tabel kebenarannya</i>	<i>27</i>
<i>Gambar 3.9. Gerbang OR 3 masukan yang disusun dari gerbang NOR dan NOT</i>	<i>28</i>
<i>Gambar 3.10. Gerbang Logika NOT dan tabel kebenarannya</i>	<i>26</i>
<i>Gambar 4.1. Rangkaian gerbang dan ekspresi aljabar booleannya</i>	<i>33</i>
<i>Gambar 4.2. Rangkaian dengan gerbang NOT dan ekspresi aljabar booleannya</i>	<i>33</i>
<i>Gambar 4.3. Penentuan ekspresi aljabar terhadap rangkaian gerbang logika..</i>	<i>35</i>
<i>Gambar 4.4. Penentuan ekspresi aljabar terhadap gerbang logika</i>	<i>36</i>
<i>Gambar 4.5. Penentuan ekspresi aljabar terhadap gerbang logika</i>	<i>36</i>

Gambar 4.6. Penentuan ekspresi aljabar terhadap gerbang logika dengan gerbang NOT	36
Gambar 4.7. Diagram Implikasi De Morgan $(x + y)' = x' \cdot y'$	39
Gambar 4.8. Diagram Implikasi De Morgan $(x \cdot y)' = x' + y'$	39
Gambar 4.9. Ekuivalensi gerbang logika dengan gerbang NAND	40
Gambar 4.10. Ekuivalensi gerbang logika dengan gerbang NOR	40
Gambar 4.11. Ekuivalensi gerbang logika NAND dan NOR	41
Gambar 4.12. Interpretasi 2 buah simbol untuk gerbang	42
Gambar 5.1. Edge-Triggered Flip-flops	47
Gambar 5.2. Edge-triggered S-R flip-flop	48
Gambar 5.3. Diagram waktu untuk Edge-triggered S-R flip-flop	49
Gambar 5.4. Edge-triggered J-K flip-flop	50
Gambar 5.5. Edge-triggered D flip-flop	50
Gambar 5.6. Pulse-Triggered (Master-Slave) Flip-flops	51
Gambar 5.7. Pulse-Triggered (Master-Slave) S-R Flip-flops	52
Gambar 5.8. Data Lock-Out Flip-flops	52
Gambar 5.9. Rangkaian Pembagi Frekuensi dengan J-K Flip-Flop	53
Gambar 5.10. Diagram Waktu Pembagi Frekuensi dengan J-K Flip-Flop	53
Gambar 5.11. Rangkaian Paralel Data Storage dengan D Flip-Flop	54
Gambar 5.12. Pencacah 2 bit dengan J-K Flip-Flop	55
Gambar 5.13. Diagram Waktu Pencacah 2 bit dengan J-K Flip-Flop	55
Gambar 6.3. Pencacah tak-sinkron 3 bit	59
Gambar 6.4. Diagram Waktu Pencacah tak-sinkron 3 bit	59
Gambar 6.5. Asynchronous Decade Counters	60
Gambar 6.6. Asynchronous Up-Down Counters	61
Gambar 6.7. Synchronous Counters	62
Gambar 6.8. Synchronous Decade Counters	63
Gambar 6.9. Synchronous Up-Down Counters	64
Gambar 6.10. Aplikasi Counter pada Konversi Data Paralel ke Data Serial	65
Gambar 7.1. Shift Register 4 jenis Serial input-Serial Output	68
Gambar 7.2. Non-destructively readout pada Shift Register 4 jenis Serial input-Serial Output	69

<i>Gambar 7.3. 4 Bits Serial In - Parallel Out Shift Registers</i>	<i>70</i>
<i>Gambar 7.4. 4 Bits Parallel In - Serial Out Shift Registers</i>	<i>71</i>
<i>Gambar 7.5. 4 Bits Parallel In - Parallel Out Shift Registers</i>	<i>71</i>
<i>Gambar 7.6. 4 Bits Serial In - Serial Out Bidirectional Shift Registers</i>	<i>72</i>
<i>Gambar 7.7. 4 Bits Ring Counter</i>	<i>73</i>
<i>Gambar 7.8. Hasil pencacahan 4 Bits Ring Counter</i>	<i>73</i>
<i>Gambar 7.9. 4 Bits Johnson Counter</i>	<i>74</i>
<i>Gambar 7.10. Hasil pencacahan 4 Bits Johnson Counter</i>	<i>74</i>

DAFTAR TABEL

<i>Tabel 1.1. Bobot Tiap Digit Pada Bilangan Desimal</i>	<i>6</i>
<i>Tabel 1.2. Bobot Tiap Digit Pada Bilangan Desimal</i>	<i>7</i>
<i>Tabel 1.3. Urutan Cacahan Bilangan Biner</i>	<i>7</i>
<i>Tabel 5.1. Tabel Kebenaran S-R Flip-Flop</i>	<i>49</i>
<i>Tabel 5.2. Tabel Kebenaran J-K Flip-Flop</i>	<i>50</i>
<i>Tabel 5.3. Tabel Kebenaran D Flip-Flop</i>	<i>51</i>
<i>Tabel 6.1. Pencacahan pada Asynchronous Decade Counters</i>	<i>60</i>
<i>Tabel 6.2. Pencacahan pada Asynchronous Up-Down Counters</i>	<i>61</i>
<i>Tabel 6.3. Pencacahan pada Synchronous Counters</i>	<i>62</i>
<i>Tabel 6.4. Pencacahan pada Synchronous Decade Counters</i>	<i>63</i>
<i>Tabel 6.5. Pencacahan pada Synchronous Decade Counters</i>	<i>64</i>

Bab I

PENGANTAR KONSEP DIGITAL

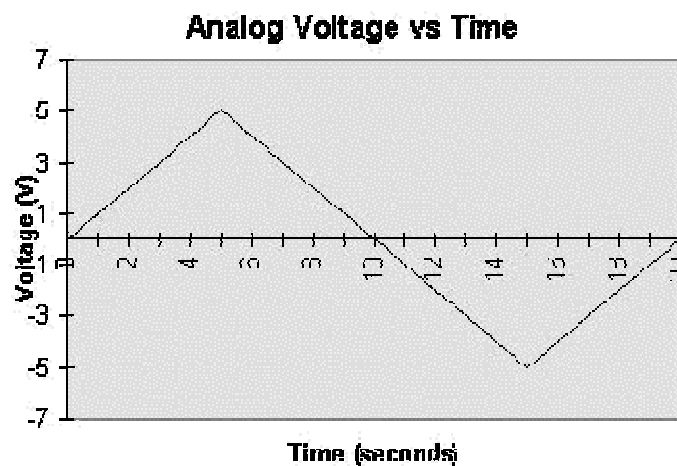
- 1.1. Representasi Numerik**
- 1.2. Keuntungan dan keterbatasan Teknik Digital**
- 1.3. Sistem Bilangan Digital**
- 1.4. Representasi Nilai Biner**

1.1. Representasi Numerik

Dalam berbagai hal di bidang keilmuan, teknologi dan bidang yang lain kita sering kali dituntut menyatakan suatu nilai atau kuantitas dengan jelas. Kuantitas adalah sesuatu yang didapatkan dari mengukur, memonitor, merekam, memanipulasi secara aritmatika, pengamatan atau cara lain yang biasa digunakan dalam sistem fisik pada umumnya. Adalah hal yang sangat penting untuk dapat menyatakan kuantitas secara efektif dan efisien. Ada 2 cara untuk menyatakan kuantitas secara numerik, yaitu secara analog dan secara digital.

1.1.1. Representasi Analog

Dalam representasi secara analog, kuantitas suatu besaran dinyatakan dengan tegangan, arus atau jarak perpindahan yang sebanding dengan nilai dari kuantitas tersebut. Kuantitas analog memiliki nilai yang bermacam-macam disepanjang rentang nilai. Dibawah ini adalah diagram sinyal tegangan analog vs waktu :

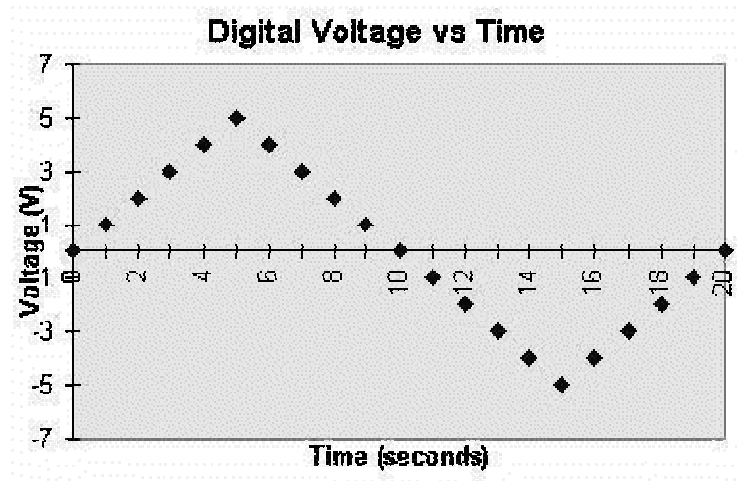


Gambar 1.1. Diagram Sinyal Tegangan Analog Terhadap Waktu

1.1.2. Representasi Digital

Dalam representasi digital, sebuah kuantitas dinyatakan tidak secara proporsional terhadap nilainya, tetapi dinyatakan dengan menggunakan sebuah simbol yang dinamakan digit. Seperti pada jam digital yang akan menghasilkan tanda waktu dalam bentuk digit desimal yang menyatakan jam, menit bahkan detik. Sebagaimana kita ketahui waktu berubah secara kontinyu, sedangkan

tampilan jam digital tidak berubah secara kontinyu, tetapi secara step dan mempunyai step sebesar satu menit (terkadang 1 detik). Dibawah ini adalah contoh diagram hubungan antara perubahan tegangan digital terhadap waktu :



Gambar 1.2. Diagram Sinyal Tegangan Digital Terhadap Waktu

Secara sederhana perbedaan utama antara kuantitas analog dan kuantitas digital adalah bahwa :

Analog = kontinyu

Digital = diskrit (step by step)

1.2. Keuntungan dan Keterbatasan Teknik DIGITAL

Teknik digital memiliki beberapa keuntungan maupun keterbatasan dalam penerapannya. Sehingga diperlukan kecermatan dalam analisa penerapannya.

12.1. Keuntungan Teknik Digital

Keuntungan teknik digital antara lain sebagai berikut :

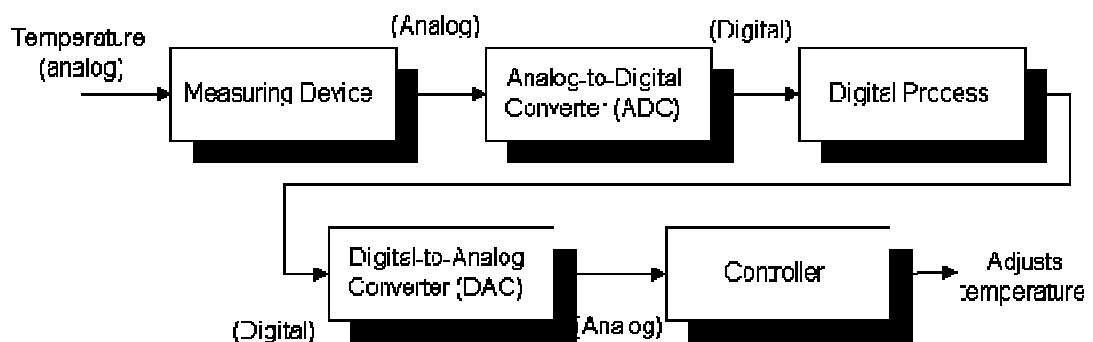
1. Mudah dalam segi disain, dikarenakan nilai pasti dari sebuah tegangan dan arus bukanlah suatu hal yang mutlak/penting, akan tetapi hanya rentang nilai dimana akan termasuk ke dalam logika TINGGI/HIGH atau logika RENDAH/LOW.
2. Penyimpanan Data sangat mudah dibanding dengan analog.
3. Lebih bagus dalam hal akurasi dan presisi.
4. Teknik Digital sangatlah Programmable. Walaupun sistem analog juga dapat diorogram, akan tetapi sangat bervariasi dan kompleks sehingga akan diperlukan suatu program yang rumit.
5. Rangkaian digital lebih kebal terhadap gangguan (noise). Dalam aplikasi sistem digital sampai saat ini jenis gangguan yang ada belum cukup kuat untuk mempengaruhi logika data HIGH maupun LOW.
6. Lebih banyak sistem digital yang telah diproduksi pabrik dalam satu kemasan berupa chips IC dibandingkan dengan sistem analog.
7. Tingkat keberhasilan dalam eksperimen biasanya lebih tinggi dibandingkan dengan sistem analog.

1.2.2. Keterbatasan Teknik Digital

Keterbatasan utama dari sistem digital adalah bahwa mayoritas sistem yang ada saat ini adalah analog (*The real world is mainly analog*). Mayoritas kuantitas besaran fisik yang ada di alam adalah merupakan kuantitas analog. Kuantitas inilah yang akan berperan sebagai input suatu sistem, selanjutnya diproses, dimonitor, dikendalikan, dan akan menghasilkan output. Sehingga untuk mengaplikasikan teknik digital dalam hal ini harus dilakukan 3 tahapan utama yaitu seperti berikut ini :

1. Rubahlah input dalam besaran analog menjadi dalam bentuk digital. Untuk ini diperlukan ADC (*Analog to Digital Converter*).
2. Proses input yang sudah berbentuk digital tersebut secara digital pula. Dari proses ini akan didapatkan output dalam bentuk digital.
3. Rubahlah bentuk besaran output dalam bentuk digital tersebut ke bentuk analog. Untuk ini diperlukan DAC (*Digital to Analog Converter*).

Berikut ini adalah diagram yang menunjukkan sistem kendali suhu yang memerlukan konversi/perubahan analog menjadi digital sehingga dapat dilakukan pemrosesan data secara digital.



Gambar 1.3. Diagram Blok Proses Yang Dilengkapi Dengan Konversi Analog-Digital-Analog

1.3. Sistem Bilangan Digital (*Digital Number System*)

Ada beberapa sistem bilangan yang dapat digunakan pada teknik digital. Yang paling utama adalah sistem bilangan desimal, biner, oktal dan heksadesimal. Sistem bilangan desimal paling mudah difahami karena telah kita gunakan bersama dalam kehidupan sehari-hari. Dengan mempelajari beberapa karakteristik yang dimiliki sistem bilangan desimal tersebut akan sangat membantu kita untuk mempelajari sistem bilangan yang lain.

1.3.1. Sistem Bilangan Desimal (*Decimal System*)

Sistem bilangan desimal tersusun atas 10 macam simbol angka. 10 macam simbol angka tersebut adalah 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Dengan menggunakan simbol-simbol tersebut untuk menyatakan digit-digit dari sebuah bilangan kita dapat menyatakan nilai dari sebuah bilangan desimal. Sistem bilangan desimal juga biasa disebut sebagai suatu sistem bilangan berbasis 10, hal ini dikarenakan pada sistem bilangan ini memiliki 10 simbol digit.

Tabel 1.1. Bobot Tiap Digit Pada Bilangan Desimal

10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}
=1000	=100	=10	=1	.	=0.1	=0.01	=0.001
Ribuan	Ratusan	Puluhan	Satuan		Per sepuluhan	Per seratusan	Per Seribuhan
Most Significant Digit				Decimal point			Least Significant Digit

1.3.2. Sistem Bilangan Biner (*Binary System*)

Dalam sistem bilangan biner, hanya ada 2 simbol digit bilangan yaitu 0 dan 1. Dari hal tersebut sistem bilangan ini juga sering dinamakan sistem bilangan berbasis 2. Sistem bilangan berbasis 2 ini bisa digunakan untuk menyatakan berbagai nilai bilangan desimal atau bilangan lain yang menggunakan sistem bilangan selain desimal dan biner.

Tabel 1.2. Bobot Tiap Digit Pada Bilangan Biner

2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
=8	=4	=2	=1	.	=1/2	=1/4	=1/8
Delapanan	Empatan	Duaan	Satuan		Per duaan	Per empatan	Per delapanan
Most Significant Bit				Binary point			Least Significant Bit

Cacahan Bilangan Biner (Binary Counting)

Urutan cacahan bilangan biner adalah sebagaimana seperti yang ditunjukkan pada tabel berikut ini :

Tabel 1.3. Urutan Cacahan Bilangan Biner

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	Decimal Equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

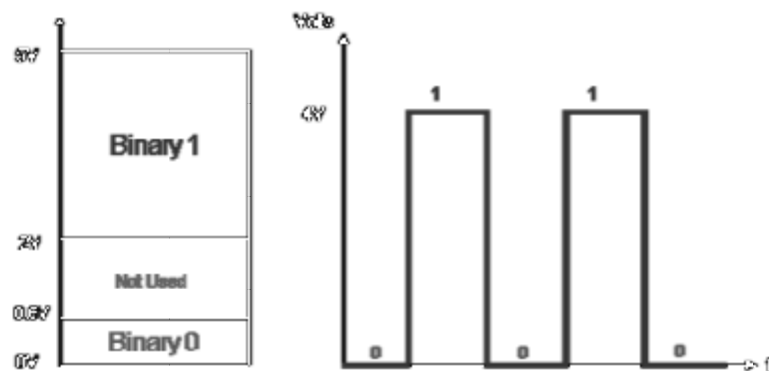
1.4. Representasi Kuantitas Biner (*Representing Binary Quantities*)

Dalam sistem digital, sebuah informasi yang akan diproses biasanya dinyatakan kedalam bentuk biner. Kuantitas biner dapat menyatakan berbagai peralatan yang hanya memiliki 2 kemungkinan kondisi operasi. Sebagai contoh adalah sebuah saklar hanya memiliki 2 kondisi operasi yaitu ON (tertutup) dan OFF (terbuka). Dalam hal ini saklar yang terbuka dapat diwakilkan dengan biner 0 (logika LOW), dan saklar yang tertutup dapat dinyatakan dengan biner 1 (logika HIGH).

Dalam teknik digital logika 1 (HIGH) dinyatakan dengan tegangan antara 2 volt sampai dengan 5 volt. Sedangkan logika 0 (LOW) dinyatakan dengan tegangan antara 0 volt sampai dengan 0,8 volt. Untuk tegangan antara 0,8 volt sampai dengan 2 volt tidak dipergunakan atau sering disebut kondisi logika mengambang. Hal ini dikarenakan tidak termasuk ke dalam logika LOW maupun HIGH.

Biner 1: Berbagai tegangan yang nilainya antara 2Volt sampai 5Volt.

Biner 0: Berbagai tegangan yang nilainya antara 0Volt sampai 0,8 Volt.



Gambar 1.4. Tingkat Tegangan Untuk Tiap Nilai Logika Digital (LOW dan HIGH)

Dari hal tersebut kita dapat melihat dengan jelas perbedaan teknik digital dengan analog. Dalam teknik digital, nilai eksak/pasti dari tegangan bukanlah suatu hal yang mutlak, sebagai contoh tegangan 3,6 volt akan dianggap sama dengan tegangan 4,3 volt yaitu sama-sama memiliki nilai logika HIGH. Hal tersebut tidak berlaku dalam sistem analog. Dalam sistem analog nilai eksak/pasti tegangan adalah suatu hal yang sangat penting.

QUIZ 1

Pilihlah jawaban yang benar dari pertanyaan berikut ini .

1. Apa perbedaan antara teknik digital dan analog ?
 - a. Kuantitas digital bisa bernilai dalam suatu range yang kontinyu
 - b. Kuantitas digital bisa bernilai pada suatu nilai diskrit dalam suatu range
 - c. Pada dasarnya adalah sama, hanya saja teknik digital adalah teknologi yang lebih baru.
 - d. Tidak ada jawaban yang benar.

2. Pilih yang merupakan kuantitas digital :
 - a. Saklar pemilih 10 posisi.
 - b. Besarnya arus listrik
 - c. Suhu
 - d. Pengatus volume radio

3. Dibawah ini adalah kelebihan teknik digital, kecuali :
 - a. Sistem digital dapat dirancang dengan lebih mudah
 - b. Lebih akurat dan presisi
 - c. Lebih tahan terhadap noise atau gangguan
 - d. Kuantitas digital lebih banyak ditemui dalam kasus nyata

4. Pilih angka terbesar yang bisa dinyatakan ke dalam 8 digit biner ?
 - a. 128
 - b. 255.
 - c. 256
 - d. 1024

5. Pilih batasan nilai tegangan yang tidak digunakan dalam teknik digital !
 - a. 0.4V - 1.2V
 - b. 0.8V - 2V
 - c. 0.8V - 2.4V
 - d. 1V - 2.4V

Bab II

SISTEM BILANGAN DAN KODE

- 2.1. Binary-to-Decimal Conversion**
- 2.2. Decimal-to-Binary Conversion**
- 2.3. Octal Number System**
- 2.4. Hexadecimal Number System**

Sistem bilangan biner adalah yang paling penting dalam sistem digital diantara sistem bilangan lain yang ada. Akan tetapi sistem bilangan yang juga penting. Sistem bilangan desimal penting dalam sistem digital, hal ini dikarenakan dipakai secara umum untuk menyatakan kuantitas untuk hal-hal diluar sistem digital. Hal ini mengakibatkan bahwa suatu saat akan diperlukan untuk dapat mengkonversikan nilai suatu bilangan desimal ke dalam bilangan biner sebelum diproses secara digital oleh sistem digital.

Dalam hal lain, ada juga sistem bilangan selain bilangan biner dan desimal, yaitu bilangan oktal (berbasis 8) dan heksadesimal (berbasis 16) yang sering dipergunakan untuk menuliskan bilangan biner yang besar dengan lebih efisien. Pada bab ini akan dijelaskan bagaimana cara melakukan konversi dari satu sistem bilangan ke dalam sistem bilangan yang lain.

2.1. Konversi Bilangan Biner ke Bilangan Desimal

(Binary-To-Decimal Conversion)

Sebuah bilangan biner dapat dikonversi menjadi bilangan desimal dengan cara menjumlahkan bobot dari masing-masing digit 1 yang terdapat pada bilangan biner tersebut. Sebagai contoh adalah sebagai berikut ini dijelaskan konversi bilangan biner 5 digit menjadi bilangan desimal yang sesuai :

$$\begin{aligned} 1 \quad 1 \quad 0 \quad 1 \quad 1 & \quad (\text{binary}) \\ 2^4 + 2^3 + 0 + 2^1 + 2^0 & = 16+8+0+2+1 \\ & = 27_{10} \text{ (decimal)} \end{aligned}$$

Dan

$$\begin{aligned} 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 & \quad (\text{binary}) \\ 2^7 + 0 + 2^5 + 2^4 + 0 + 2^2 + 0 + 2^0 & = 128+0+32+16+0+4+0+1 \\ & = 181_{10} \text{ (decimal)} \end{aligned}$$

Nampak bahwa semua digit 1 pada bilangan biner dibobotkansesuai dengan posisi digitnya, kemudian hasil pembobotan tersebut dijumlahkan secara keseluruhan.

2.2. Konversi Bilangan Desimal ke Bilangan Biner

(Decimal-To-Binary Conversion)

Untuk mengkonversi bilangan desimal menjadi bilangan biner ada 2 cara yang umum dipakai. Yaitu :

1. Menguraikan bilangan desimal ke dalam jumlahan dari bobot digit bilangan biner (*Reverse of Binary to Digital Methode*)
2. Pembagian secara berulang (*Repeat Division Methode*)

2.2.1. Metode Penguraian (Reverse of Binary-To-Digital Method)

Cara ini dapat dijelaskan dengan contoh bila kita menginginkan konversi bilangan desimal 45 menjadi bilangan binernya sebagaimana yang diuraikan pada contoh berikut ini.

$$\begin{aligned}
 45_{10} &= 32 + 0 + 8 + 4 + 0 + 1 \\
 &= 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 \\
 &= 1\ 0\ 1\ 1\ 0\ 1_2
 \end{aligned}$$

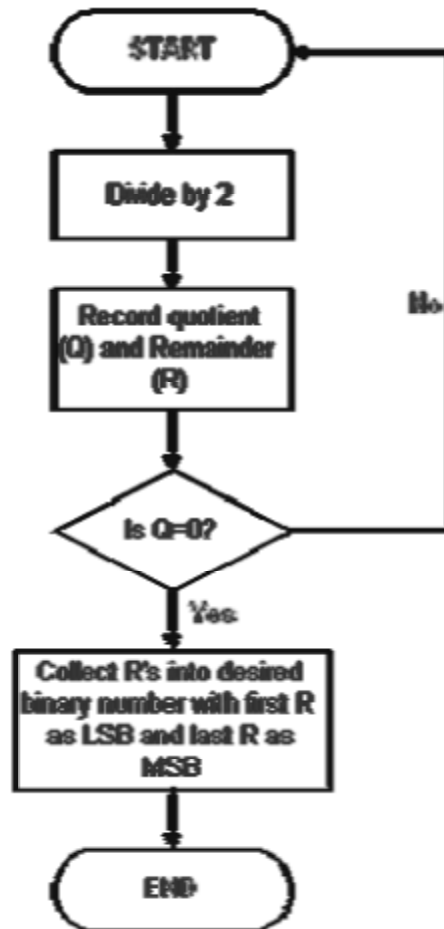
2.2.2. Metode Pembagian Secara Berulang (Repeat Division Methode)

Pada metode ini konversi bilangan desimal menjadi bilangan biner dilakukan dengan membagi bilangan desimal tersebut dengan angka 2 secara berulang-ulang dengan selalu mencatat hasil bagi dan sisa pembagiannya. Pengulangan pembagian tersebut dilakukan sampai pada akhirnya sudah tidak dapat dibagi lagi.

Berikut ini contoh bagaimana metode ini dilakukan untuk konversi bilangan desimal menjadi bilangan biner.

	Hasil	Sisa	Keterangan
25 / 2	12	1	(Least Significant Bit)
12 / 2	6	0	
6 / 2	3	0	
3 / 2	1	1	
1 / 2	0	1	(Most Significant Bit)
Result	25 ₁₀ =	1 1 0 0 1 ₂	

Diagram alir dari metode pembagian berulang ini dapat digambarkan sebagai berikut :



Gambar 2.1. Diagram alir metode pembagian berulang

2.3. Sistem Bilangan Oktal (Octal Number System)

Sistem bilangan oktal memiliki basis bilangan 8 dan memiliki simbol angka ada 8 macam yaitu 0,1,2,3,4,5,6 dan 7.

8^3	8^2	8^1	8^0	.	8^{-1}	8^{-2}	8^{-3}
=512	=64	=8	=1	.	=1/8	=1/64	=1/512
Most Significant Digit				Octal point			Least Significant Digit

2.3.1. Konversi bilangan octal ke decimal

(*Octal to Decimal Conversion*)

Konversi bilangan oktal ke desimal pada prinsipnya adalah sama seperti halnya konversi bilangan biner ke decimal. Yaitu pertama kali yang harus dilakukan adalah mengalikan tiap digit yang dengan bobot yang sesuai. Kemudian dilanjutkan dengan menjumlahkan secara keseluruhan hasil perkalian tersebut.

Contoh :

$$24.6_8 = 2 \times (8^1) + 4 \times (8^0) + 6 \times (8^{-1}) = 20.75_{10}$$

2.3.2. Konversi Bilangan Biner ke Bilangan Oktal dan Oktal ke Biner

(*Binary-To-Octal / Octal-To-Binary Conversion*)

Untuk mengkonversi bilangan biner ke bilangan oktal dapat dilakukan dengan cara mengelompokkan tiap digit yang ada pada bilangan biner ke dalam 3 digit untuk tiap kelompoknya. Kemudian mengkonversi tiap kelompok tersebut menjadi bilangan oktal yang sesuai. Yang harus diingat bahwa pengelompokan dilakukan urut dari digit yang bobotnya paling ringan (Least Significant Bit / LSB).

Octal Digit	0	1	2	3	4	5	6	7
Binary Equivalent	000	001	010	011	100	101	110	111

Dengan demikian untuk mengkonversikan bilangan oktal ke biner dapat dilakukan dengan kebalikan langkah sebagaimana disebutkan diatas, yaitu tiap digit pada bilangan oktal dinyatakan dengan tiga digit bilangan biner (3 bit).

Contoh :

$$100\ 111\ 010_2 = (100)\ (111)\ (010)_2 = 4\ 7\ 2_8$$

2.3.3. Konversi Bilangan Desimal ke Oktal Dengan Pembagian Berulang (*Repeat Division*)

Metode ini menggunakan pembagian berulang dengan faktor pembagi adalah 8. Sebagai contoh jika diinginkan mengkonversi bilangan 177_{10} menjadi bilangan oktal dan biner dapat dilakukan sebagai berikut :

$$177/8 = 22 + \text{sisa } \mathbf{1} \quad \mathbf{1 \text{ (Least Significant Bit)}}$$

$$22/8 = 2 + \text{sisa } \mathbf{6} \quad \mathbf{6}$$

$$2/8 = 0 + \text{sisa } \mathbf{2} \quad \mathbf{2 \text{ (Most Significant Bit)}}$$

$$\text{Hasilnya : } 177_{10} = \mathbf{2 \ 6 \ 1}_8$$

$$\text{Konversi ke Biner} = 010 \ 110 \ 001_2$$

2.4. Sistem Bilangan Heksadesimal (*Hexadecimal Number System*)

Sistem bilangan heksadesimal menggunakan basis 16. Sehingga memiliki 16 jenis simbol digit yang bisa dipergunakan. Ke-enambelas simbol tersebut adalah antara lain 0 , 1 , 2 , 3 , 9 , A , B , C , D , E dan F.

16^3	16^2	16^1	16^0		16^{-1}	16^{-2}	16^{-3}
=4096	=256	=16	=1	.	=1/16	=1/256	=1/4096
Most Significant Digit				Hexadec. point			Least Significant Digit

2.4.1. Konversi Heksadesimal ke Desimal

(*Hexadecimal to Decimal Conversion*)

Konversi bilangan heksadesimal ke desimal pada prinsipnya adalah sama seperti halnya konversi bilangan oktal ke decimal yang terdiri du tahapan. Yaitu pertama kali yang harus dilakukan adalah mengalikan tiap digit yang ada pada bilangan heksadesimal dengan bobot yang sesuai. Kemudian dilanjutkan dengan menjumlahkan secara keseluruhan hasil perkalian tersebut.

Sebagai contoh akan dilakukan konversi bilangan heksadesimal ke bilangan desimal yang sesuai :

$$2AF_{16} = 2 \times (16^2) + 10 \times (16^1) + 15 \times (16^0) = 687_{10}$$

2.4.2. Konversi Bilangan Desimal ke Bilangan Heksadesimal dengan Pembagian Berulang (*Repeat Division*)

Metode ini menggunakan pembagian berulang dengan faktor pembagi adalah 16. Sebagai contoh jika diinginkan mengkonversi bilangan 378_{10} menjadi bilangan heksadesimal dan biner dapat dilakukan sebagai berikut :

$$\begin{array}{ll} 378/16 = 23 + \text{sisa } \mathbf{10} & \mathbf{A \text{ (Least Significant Bit)}} \\ 23/16 = 1 + \text{sisa } 7 & 7 \\ 1/16 = 0 + \text{sisa } \mathbf{1} & \mathbf{1 \text{ (Most Significant Bit)}} \end{array}$$

$$\text{Hasilnya } 378_{10} = \mathbf{1 \quad 7 \quad A}_{16}$$

$$\text{Konversi ke Biner} = 0001 \ 0111 \ 1010_2$$

2.4.3. Konversi Bil. Biner ke Bil. Heksadesimal dan sebaliknya

(*Binary-To-Hexadecimal /Hexadecimal-To-Binary Conversion*)

Untuk mengkonversi bilangan biner ke bilangan heksadesimal dapat dilakukan dengan cara mengelompokkan tiap digit yang ada pada bilangan biner ke dalam 4 digit untuk tiap kelompoknya. Kemudian mengkonversi tiap kelompok tersebut menjadi bilangan heksadesimal yang sesuai. Yang harus diingat bahwa pengelompokan dilakukan urut dari digit yang bobotnya paling ringan (Least Significant Bit / LSB). Berikut ini disampaikan digit bilangan heksadesimal ekuivalen dengan digit bilangan biner. Tiap digit bilangan heksadesimal dinyatakan dalam empat digit biner (4 bit).

Hexadecimal Digit	0	1	2	3	4	5	6	7
Binary Equivalent	0000	0001	0010	0011	0100	0101	0110	0111

Hexadecimal Digit	8	9	A	B	C	D	E	F
Binary Equivalent	1000	1001	1010	1011	1100	1101	1110	1111

Sebagai contoh :

$$101100101111_2 = (1011) (0010) (1111)_2 = B 2 F_{16}$$

2.4.4. Konversi Bilangan Oktal ke Bilangan Heksadesimal dan Sebaliknya (*Octal-To-Hexadecimal /Hexadecimal-To-Octal Conversion*)

Untuk mengkonversi bilangan heksadesimal ke bilangan oktal dapat dilakukan dengan 2 tahapan yaitu :

1. Rubahlah bilangan heksadesimal yang ada menjadi bilangan biner terlebih dahulu
2. Kelompokkan bilangan biner tersebut menjadi masing-masing 3 digit biner dimulai dari Least Significant Bits
3. Rubahlah tiap kelompok yang ada ke dalam kode digit oktal yang sesuai

Sebagai contoh akan dirubah bilangan heksadesimal $5A8_{16}$ menjadi bilangan oktal yang sesuai. Hal ini bisa dijelaskan sebagai berikut :

$$5A8_{16} = \dots\dots\dots \text{(Oktal)}$$

Langkah 1 :

$$5A8_{16} = 0101 1010 1000 \text{ (Biner)}$$

$$\text{Langkah 2 :} \quad = 010 110 101 000$$

$$\text{Langkah 3 :} \quad = 2 \quad 6 \quad 5 \quad 0 \text{ (Oktal)}$$

Sedangkan untuk mengkonversi bilangan oktal ke bilangan heksadesimal dapat dilakukan dengan membalik urutan ketiga langkah diatas, menjadi :

1. Rubahlah bilangan oktal yang ada menjadi bilangan biner terlebih dahulu
2. Kelompokkan bilangan biner tersebut menjadi masing-masing 4 digit biner dimulai dari Least Significant Bits
3. Rubahlah tiap kelompok yang ada ke dalam kode digit heksadesimal yang sesuai

Sebagai contoh akan dirubah bilangan heksadesimal 2650_8 menjadi bilangan heksadesimal yang sesuai. Hal ini bisa dijelaskan sebagai berikut :

$$2650_8 = \dots\dots\dots \text{(heksadesimal)}$$

Langkah 1 :

$$2650_8 = 010\ 110\ 101\ 000 \text{ (biner)}$$

Langkah 2 : $= 0101\ 1010\ 1000$

Langkah 3 : $= 5\ A\ 8 \text{ (heksadesimal)}$

Quis 2 :

Pilih jawaban yang benar dari pertanyaan berikut ini!

1. Rubahlah bilangan $(63.25)_{10}$ ke bilangan biner !
 - a. 11111.11
 - b. 111001.01
 - c. 111111.01
 - d. 111111.1
 - e. NA

2. Rubahlah bilangan $(43.8125)_{10}$ ke bilangan biner !
 - a. 101011.1101
 - b. 110101.1101
 - c. 101011.1011
 - d. 110101.1011
 - e. NA

3. Rubahlah bilangan $(1001011.011)_2$ ke bilangan desimal !
 - a. 73.0375
 - b. 75.375
 - c. 91.375
 - d. 75.573
 - e. NA

4. Rubahlah bilangan $(110101.1011)_2$ ke bilangan desimal !
 - a. 53.6875
 - b. 53.6375
 - c. 52.6875
 - d. 55.6375
 - e. NA

5. Rubahlah bilangan $(11001.1)_2$ ke bilangan oktal !
 - a. $(62.4)_8$
 - b. $(62.1)_8$
 - c. $(31.1)_8$
 - d. $(31.2)_8$
 - e. $(31.4)_8$

6. Rubahlah bilangan $(25.6)_8$ ke bilangan biner !
 - a. $(10101.11)_2$
 - b. $(11101.10)_2$
 - c. $(10101.10)_2$
 - d. $(10010.11)_2$
 - e. $(11111.01)_2$

7. Rubahlah bilangan $(35.1)_8$ ke bilangan heksadesimal !

- a. $(17.4)_{16}$
- b. $(1D.1)_{16}$
- c. $(D1.2)_{16}$
- d. $(E8.1)_{16}$
- e. NA

8. Rubahlah bilangan $(39.A)_{16}$ ke bilangan oktal !

- a. $(35.5)_8$
- b. $(70.5)_8$
- c. $(71.5)_8$
- d. $(72.25)_8$
- e. $(75.5)_8$

9. Rubahlah bilangan $(485)_{10}$ ke bilangan heksadesimal !

- a. $(1E5)_{16}$
- b. $(231)_{16}$
- c. $(5E1)_{16}$
- d. $(15E)_{16}$
- e. NA

10. Rubahlah bilangan $(397)_{10}$ ke bilangan berbasis 3 !

- a. $(12310)_3$
- b. $(121201)_3$
- c. $(012211)_3$
- d. $(112201)_3$
- e. $(100202)_3$

Bab III

GERBANG LOGIKA

(*Logic Gates*)

3.1. Boolean Variables & Truth Tables

3.2. OR Operation

3.3. AND Operation

3.4. NOT Operation

3.5. NOR Operation

3.6. NAND Operation

3.1. Variabel Boolean dan Tabel Kebenaran

(Boolean Variables & Truth Tables)

Aljabar boolean sangat berbeda dengan aljabar pada umumnya. Dalam aljabar boolean variabelnya hanya memiliki 2 kemungkinan nilai yaitu 0 atau 1. Lihat sub bab 1.4. tentang bagaimana definisi nilai 0 dan 1. Nilai boolean 0 dan 1 bukanlah menyatakan nilai dari angka 0 dan 1 itu sendiri, akan tetapi menyatakan keadaan variabel tegangan atau lebih sering disebut sebagai level logic (0 = LOW dan 1 = HIGH).

Beberapa representasi tentang variabel boolean 0 dan 1 yang sering digunakan adalah seperti yang tercantum dalam tabel berikut ini.

Logic 0	Logic 1
False	True
Off	On
Low	High
No	Yes
Open Switch	Close Switch

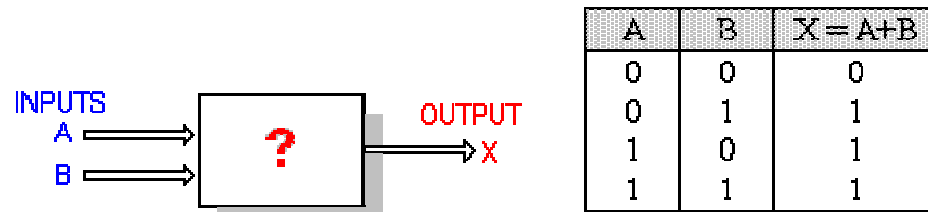
Gambar 3.1. Nilai Variabel Boolean

Dalam aljabar boolean, terdapat 3 operasi logika yang utama, yaitu OR, AND dan NOT. Ketiga operasi tersebut dinyatakan kedalam gerbang logika (Logic Gate). Gerbang logika tersebut adalah sebuah rangkaian digital yang tersusun dari sebuah dioda, transistor dan resistor yang terhubung sedemikian rupa sehingga akan menghasilkan keluaran sesuai operasi logika OR, AND dan NOT.

Tabel Kebenaran (*Truth Table*)

Tabel kebenaran adalah suatu tabel yang menyatakan hubungan input dengan output. Tabel ini menjelaskan bagaimana logika output yang terjadi tergantung pada logika input yang diberikan pada rangkaian.

Dalam contoh rangkaian logika dengan 2 masukan berikut ini diberikan tabel kebenaran yang menyatakan kombinasi logika input yang mungkin terjadi dan logika output yang sesuai untuk tiap kombinasi inputnya.



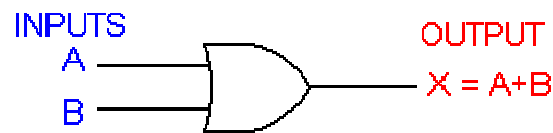
Gambar 3.2. Gerbang Logika dengan 2 masukan dan tabel kebenarannya

Dari tabel nampak bahwa logika output yang terjadi sangat tergantung pada logika input yang diberikan. Yaitu sebagai berikut :

- Ketika logika input A dan B keduanya memiliki logika 0, maka logika keluaran X adalah 0.
- Ketika logika input A bernilai 0 dan B bernilai 1, maka logika keluaran X adalah 1.
- Ketika logika input A bernilai 1 dan B bernilai 0, maka logika keluaran X adalah 1.
- Ketika kedua input A dan B memiliki logika 1, maka logika keluaran X adalah 1.

3.2. Operasi OR (*OR Operation*)

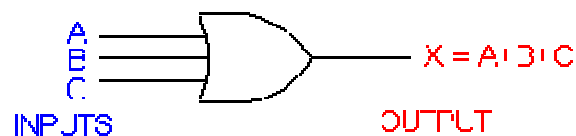
Ekpresi matematis dari logika OR adalah $X = A + B$ (dibaca : $X = A$ OR B). Tanda + menyatakan operasi logika OR bukan operasi penjumlahan angka biasa pada umumnya. Operasi logika OR akan menghasilkan logika 1 jika minimal ada salah satu variabel yang ada bernilai 1. Operasi logika OR akan menghasilkan logika 0 jika semua variabel yang ada bernilai 0.



A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

Gambar 3.3. Gerbang Logika OR dengan 2 masukan dan tabel kebenarannya

Berikut ini adalah contoh gerbang logika OR dengan 3 masukan. A, B dan C adalah masukan dan X adalah keluaran gerbang logika OR.



A	B	C	X = 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 3.4. Gerbang Logika OR dengan 3 masukan dan tabel kebenarannya

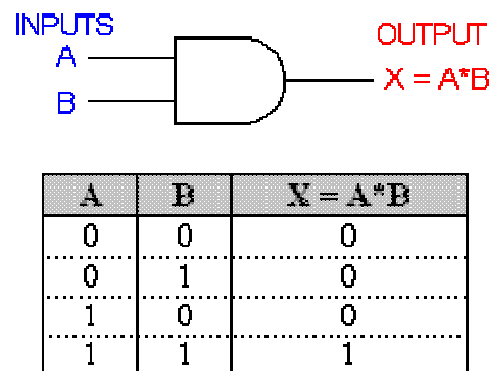
Catatan : Dengan menggunakan operasi OR maka $1 + 1 = 1$, $1 + 1 + 1 = 1$, demikian dan seterusnya.

3.3. Operasi AND (AND Operation)

Ekspresi operasi logika AND adalah $X = A * B$ (dibaca : $X = A \text{ AND } B$). Tanda perkalian pada operasi logika AND adalah sama dengan perkalian biasa terhadap angka 1 dan 0. Pada operasi logika AND akan menghasilkan logika

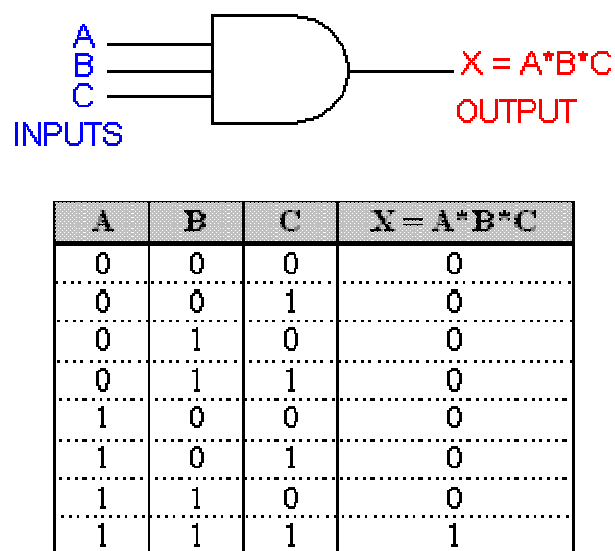
keluaran 1 jika dan hanya jika semua variabelnya memiliki nilai logika 1. Sebaliknya jika ada salah satu atau lebih dari variabel masukan yang memiliki logika 0 maka logika keluaran yang didapatkan akan bernilai 0.

Sebagai contoh berikut ini adalah sebuah gerbang logika AND dengan 2 masukan. Dari tabel kebenarannya tampak bahwa jika semua logika variabel masukan bernilai 1 maka logika keluaran akan bernilai 1. Sebaliknya jika minimal ada salah satu masukan memiliki logika 0, maka logika keluaran akan bernilai 0.



Gambar 3.5. Gerbang Logika AND dengan 2 masukan dan tabel kebenarannya

Untuk gerbang AND dengan 3 masukan dijelaskan sebagaimana berikut ini lengkap dengan tabel kebenarannya.



Gambar 3.6. Gerbang Logika AND dengan 3 masukan dan tabel kebenarannya

Dengan operasi logika AND maka : $1*1 = 1$, $1*1*1 = 1$ dan seterusnya.

3.4. Operasi NOT (*NOT Operation*)

Operasi NOT berbeda dengan operasi OR dan AND, yaitu bahwa operasi NOT dapat diaplikasikan terhadap variabel masukan tunggal. Sebagai contoh, jika terhadap variabel A diterapkan operasi NOT, hasil keluaran X dapat dinyatakan sebagai $X = A'$, dimana tanda (') menyatakan operasi NOT. Ekspresi ini dibaca sebagai :

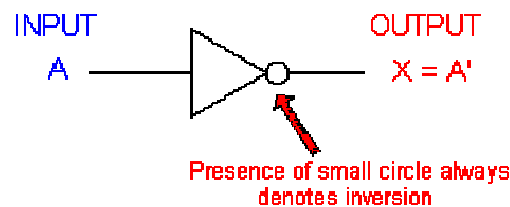
$$X = \text{NOT } A$$

X = inverse dari A , atau

X = komplement dari A

Ketiga cara baca tersebut diatas sangat sering digunakan dalam teknik digital, dan semuanya menghasilkan logika keluaran yang sama yaitu bahwa nilai logika X adalah kebalikan dari nilai logika A.

Simbol gerbang logika dan tabel kebenaran dari sebuah operasi NOT adalah sebagai berikut :



A	X = A'
0	1
1	0

Gambar 3.7. Gerbang Logika NOT dan tabel kebenarannya

Catatan :

$1' = 0$ karena NOT 1 is 0

$0' = 1$ karena NOT 0 is 1

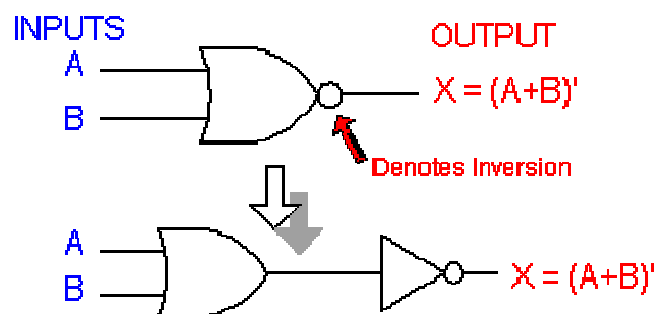
3.5. Operasi NOT OR (NOR Operation)

Gerbang NOT OR (NOR) dan gerbang NOT AND (NAND) sangat banyak digunakan dalam rangkaian digital. Gerbang ini mengkombinasikan gerbang AND, OR dan NOT, sehingga sangat mudah untuk memahaminya dalam menyatakan aljabar boolean.

Gerbang NOR memiliki simbol yang sama dengan gerbang OR dengan sedikit tambahan berupa lingkaran kecil pada sisi keluaran. Lingkaran tersebut menyatakan operasi pembalikan nilai logika (inversi).

Ekspresi untuk operasi logika NOR dengan 2 masukan adalah :

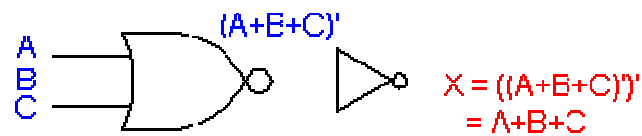
$$X = (A + B)' = \overline{(A + B)}$$



INPUTS		OR	NOR
A	B	$X = A+B$	$X = (A+B)'$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Gambar 3.8. Gerbang Logika NOR 2 masukan dan tabel kebenarannya

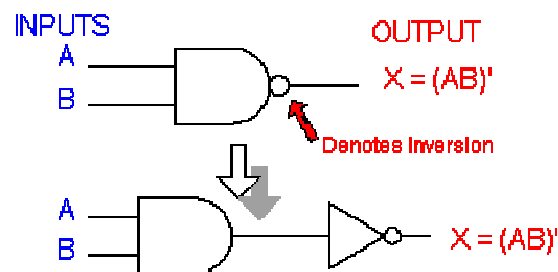
Sebuah contoh gerbang OR dengan 3 masukan dapat disusun dari sebuah gerbang NOR ditambah dengan gerbang NOT. Hal ini didasarkan pada prinsip logika boolean bahwa pembalikan nilai logika sebanyak dua kali tidak akan merubah nilai logika tersebut.



Gambar 3.9. Gerbang OR 3 masukan yang disusun dari gerbang NOR dan NOT

3.6. Operasi NAND (NAND Operation)

Simbol gerbang logika untuk operasi NAND adalah sama dengan AND ditambahkan dengan lingkaran kecil disisi keluaran yang menandakan pembalikan keluaran yang dihasilkan. Untuk gerbang NAND dengan 2 masukan dapat dinyatakan dengan ekspresi boolean $X = (AB)'$. Simbol gerbang logika NAND dengan 2 masukan dan tabel kebenarannya adalah seperti berikut ini.



INPUTS		AND	NAND
A	B	$X = AB$	$X = (AB)'$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

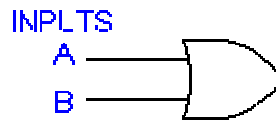
Gambar 3.10. Gerbang Logika NAND dengan 2 masukan dan tabel kebenarannya

QUIZ 3

Pilih jawaban yang tepat untuk soal-soal berikut ini !

1. Aljabar boolean berbeda dengan aljabar pada umumnya dalam hal :
 - a. Aljabar boolean dapat memiliki nilai lebih dari 1 nilai antara 0 dan 1
 - b. Aljabar boolean hanya memiliki 2 nilai diskrit yaitu 0 dan 1
 - c. Aljabar boolean dijelaskan dengan level tegangan sampai dengan 3 macam
 - d. Pada prinsipnya adalah sama
 - e. Tidak ada jawaban yang benar

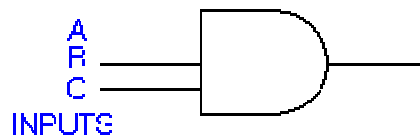
2. Pertanyaan soal nomor 2 dan 3 mengacu pada gambar berikut ini :



Tentukan keluaran X jika kedua masukan A dan B adalah 0 !

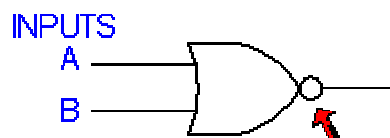
- a. 0
 - b. 1
 - c. Tidak Pasti
 - d. Tidak ada jawaban yang tepat
3. Tentukan keluaran X jika masukan A = 1 dan B = 0 !
 - a.
 - b. 0
 - c. 1
 - d. Tidak Pasti
 - e. Tidak ada jawaban yang tepat
 4. Untuk tiga masukan (A,B,C) pada gerbang logika OR, kombinasi mana yang menghasilkan keluaran bernilai 0 ?
 - a. A=0, B=0, C=1
 - b. A=0, B=1, C=0
 - c. A=1, B=1, C=1
 - d. A=0, B=0, C=0

5. Pertanyaan nomor 5 dan 6 mengacu pada gambar berikut ini :



Tentukan keluaran X jika masukan $A=1$, $B=0$ and $C=1$?

- a. 0
 - b. 1
 - c. Tidak tahu
 - d. Tidak ada jawaban yang benar
6. Kombinasi masukan yang menyebabkan keluaran $X = 1$ adalah ?
- a. $A=0, B=0, C=0$
 - b. $A=1, B=0, C=1$
 - c. $A=0, B=1, C=0$
 - d. $A=1, B=1, C=1$
 - e. Tidak ada jawaban yang benar
7. Pertanyaan nomor 7 dan 8 mengacu pada gambar berikut ini :

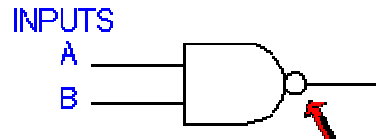


Tentukan keluaran jika masukan $A=0$, $B=1$?

- a. 0
 - b. 1
 - c. Tidak pasti
 - d. Tidak ada jawaban yang benar
8. Kombinasi masukan yang menyebabkan keluaran bernilai 1 adalah
- a. $A=0, B=0$
 - b. $A=0, B=1$
 - c. $A=1, B=0$
 - d. $A=1, B=1$

e. Tidak tahu

9. Pertanyaan berikut ini mengacu pada gambar berikut ini :



Kombinasi masukan yang menghasilkan keluaran bernilai 0 ?

- a. A=0, B=0
- b. A=0, B=1
- c. A=1, B=0
- d. A=1, B=1
- e. Tidak tahu

10. Berikut ini tabel kebenaran suatu gerbang logika yang belum diketahui :

A	B	C	X
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

Tentukan jenis gerbang logika tersebut :

- a. 3 Inputs OR
- b. 3 Inputs AND
- c. 3 Inputs NOR
- d. 3 Inputs NAND
- e. Tidak pasti

DAFTAR PUSTAKA

- [1] Wijaya Widjanarka, Ir., *Teknik Digital* , Erlangga, Jakarta, 2006.
- [2] Bartee, T.C., Ph.D., *Digital Computer Fundamental* , 6th Ed, Mc Graw Hill, Inc, 1985.
- [3] Tocci, R. J., *Digital System : Principle and Applications* , 4th ed, Prentice Hall Inc., New Jersey.
- [4] Tokheim, R.L., M.Sc., *Digital Principle* , 2nd ed., Mc graw Hill, Inc, Singapore, 1984.
- [5] Malvino, A. P., Ph.D., *Electronic Principles*, 3rd ed., Mc Graw Hill, Singapore, 1984.

