

# PERTEMUAN MINGGU KE-2



---

LEVEL GATE



# KOMPONEN LEVEL GATE

---

## **1. Rangkaian Kombinatorial**

Komponennya terdiri dari : Logic Gate (Gerbang Logika)

## **2. Rangkaian Sequential**

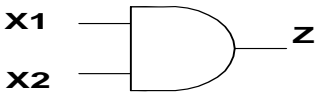
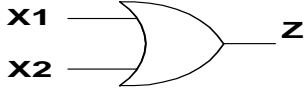
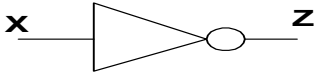
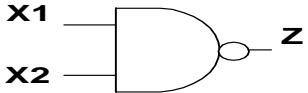
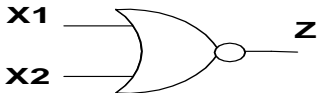

Komponennya terdiri dari : Flip-Flop



# Logic Gate (Gerbang Logika)

---

- Logic Gate (Gerbang Logika) adalah merupakan dasar pembentuk sistem digital
- Logic Gate mempunyai gerbang logika dasar yaitu NOT, AND dan OR.
- Dari 3 gerbang logika dasar dibentuk 4 gerbang logika tambahan yaitu NAND, NOR, EX-OR, dan EX- NOR

Name	Circuit symbol	Truth table	Equation															
<b>AND</b>		<table border="1"> <thead> <tr> <th>X1</th> <th>X2</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	X1	X2	Z	0	0	0	0	1	0	1	0	0	1	1	1	$Z = X1 \cdot X2$
X1	X2	Z																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
<b>OR</b>		<table border="1"> <thead> <tr> <th>X1</th> <th>X2</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	X1	X2	Z	0	0	0	0	1	1	1	0	1	1	1	1	$Z = X1 + X2$
X1	X2	Z																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
<b>NOT</b>		<table border="1"> <thead> <tr> <th>X</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	X	Z	0	1	1	0	$Z = \overline{X}$									
X	Z																	
0	1																	
1	0																	
<b>NAND</b>		<table border="1"> <thead> <tr> <th>X1</th> <th>X2</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	X1	X2	Z	0	0	1	0	1	1	1	0	1	1	1	0	$Z = \overline{X1 \cdot X2}$
X1	X2	Z																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
<b>NOR</b>		<table border="1"> <thead> <tr> <th>X1</th> <th>X2</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	X1	X2	Z	0	0	1	0	1	0	1	0	0	1	1	0	$Z = \overline{X1 + X2}$
X1	X2	Z																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
<b>EXCLUSIVE-OR</b>		<table border="1"> <thead> <tr> <th>X1</th> <th>X2</th> <th>Z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	X1	X2	Z	0	0	0	0	1	1	1	0	1	1	1	0	$Z = X1 \oplus X2$
X1	X2	Z																
0	0	0																
0	1	1																
1	0	1																
1	1	0																



# Perubahan Gerbang Dengan Menggunakan Pembalik

---

1. Gerbang Asal + Pembalik Pada Output = Gerbang Baru
2. Pembalik Pada Input + Gerbang Asal = Gerbang Baru
3. Pembalik Pada Input + Gerbang Asal + Pembalik Pada Output = Gerbang Baru



# ALJABAR BOOLEAN

---

- Aljabar Boolean merupakan cara yang ekonomis untuk menjelaskan fungsi rangkaian digital, bila fungsi yang diinginkan telah diketahui, maka aljabar boolean dapat digunakan untuk membuat implementasi fungsi tersebut dengan cara yang lebih sederhana.



# HUKUM DAN TEOREMA ALJABAR BOOLEAN

---

1.  $A + 0 = A$

$A \cdot 1 = A$

2.  $A + A' = 1$

$A \cdot A' = 0$

3.  $A + 1 = 1$

$A \cdot 0 = 0$

4.  $A + A = A$

$A \cdot A = A$

5.  $(A')' = A$

6.  $A + B = B + A$

$A \cdot B = B \cdot A$

7.  $A + (B \cdot C) = (A + B) \cdot (A + C)$

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

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

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

9.  $(A \cdot B)' = A' + B'$

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

**IDENTITAS**

**INVERS**

**KOMUTATIF**

**DISTRIBUTIF**

**ASOSIATIF**

**DE MORGAN**

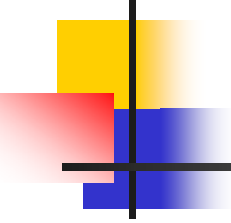


# PETA KARNAUGH

---

- Salah satu teknik yang paling mudah untuk penyederhanaan rangkaian logika adalah dengan menggunakan peta karnaugh.
- **Peta karnaugh dapat digunakan untuk menyusun :**
- Aljabar Boolean Minterm (Sum Of Product)
- Aljabar Boolean Maksterm (Pos Of Product)



- 
- 
- Penusunan Peta Karnaugh menggunakan urutan **Sandi Gray** yaitu :
  - **00, 01, 11, 10** atau  **$A'B'$  ,  $A'B$ ,  $AB$ ,  $AB'$**



## KONDISI TAK PEDULI (DON'T CARE)

---

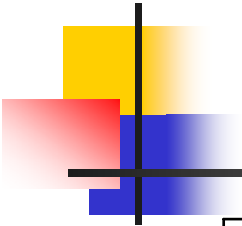
- Bilangan BCD (Binary Code Desimal) dibatasi pada bilangan 4 bit yaitu dari 0000 sampai 1001, 1010 sampai 1111 tidak mungkin terjadi pada operasi normal.
- Karena masukkan BCD yang terlarang tidak terjadi di bawah kondisi operasi normal, maka ruang-ruang kosong dapat dipandang 1 atau 0 tergantung mana yang lebih menguntungkan.
- Untuk menunjukkan hal ini diberikan tanda X
- Tanda X ini disebut tak peduli (Don't Care)



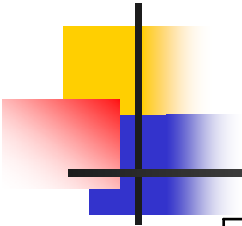
# PELINGKARAN YANG TIDAK BIASA

<b>1</b>	
<b>1</b>	
	<b>1</b>

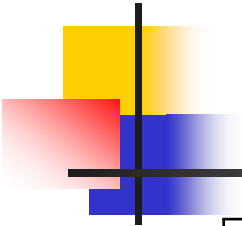
		<b>1</b>	
		<b>1</b>	



<b>1</b>			<b>1</b>
<b>1</b>			<b>1</b>



	<b>1</b>	<b>1</b>	
	<b>1</b>	<b>1</b>	



<b>1</b>			<b>1</b>
<b>1</b>			<b>1</b>



# FLIP-FLOP

---

- **RANGKAIAN LOGIKA SEKUENSIAL**

Adalah rangkaian dimana outputnya tidak hanya tergantung pada input waktu itu saja, tetapi juga pada keadaan input sebelumnya.

- Contoh rangkaian sekuensial yang paling sederhana adalah Flip-flop/FF.
- Flip-flop adalah perangkat bistabil, hanya dapat berada pada salah satu statusnya saja, jika input tidak ada, FF tetap mempertahankan statusnya. Maka FF dapat berfungsi sebagai memori 1-bit.
- Flip-Flop disebut juga kancing, multivibrator, biner