
MODUL PRAKTIKUM SISTEM DIGITAL I



**LABORATORIUM SISTEM PENGATURAN & KOMPUTER
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SUMATERA UTARA
2017**

Bab 1

Gerbang Logika Dasar Aljabar Boole

1.1 Tujuan

- ✓ Mempelajari gerbang logika dasar pada masing – masing tipe IC.
- ✓ Mempelajari cara kerja dan karakteristik gerbang logika.
- ✓ Menganalisis dan merancang rangkaian logika pada sebuah sistem.

1.2 Teori

Masyarakat modern sudah banyak yang beralih pada rangkaian digital. Hal ini disebabkan oleh beberapa keuntungan yang diperoleh. Dalam dunia digital, umumnya komponen – komponen yang digunakan (gerbang – gerbang logika) berbentuk rangkaian terintegrasi (*Integrated Circuit*). tipe – tipe *IC* yang banyak digunakan, antara lain : *TTL*, *CMOS*, dan lain – lain. Karakteristik suatu *IC* dapat dilihat pada **Data Sheet**. Rangkaian digital ini dibentuk dari persamaan aljabar yang lazim kita sebut dengan **Aljabar Boole**. Hasil ekspresi dari boolean tidak akan dapat berguna jika tidak bisa diimplementasikan dalam bentuk rangkaian logika. Untuk membantu mengekspresikan aljabar boole, maka pada setiap sistem terdapat **truth table** (tabel kebenaran) yang dapat membantu dalam penyederhanaan rangkaian. Ada beberapa mekanisme singkat yang dapat membantu proses penyederhanaan yang lain, diantaranya :

- Peta Karnaugh
- Tabel Quinn - McCluskey

Untuk mendapatkan persamaan aljabar yang lebih sederhana, maka digunakanlah beberapa hukum dasar, diantaranya :

1. Operasi dengan 0 dan 1	$x + 0 = x$	$x \cdot 0 = 0$
	$x + 1 = 1$	$x \cdot 1 = x$
2. Hukum Idempoten	$x + x = x$	
	$x \cdot x = x$	
3. Hukum Komplement	$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$
4. Hukum De Morgan	$\overline{(x + y + z)} = \bar{x} \cdot \bar{y} \cdot \bar{z}$	
	$\overline{(x \cdot y \cdot z)} = \bar{x} + \bar{y} + \bar{z}$	
5. Teorema Konsensus	$xy + yz + \bar{x}z = xy + \bar{x}z$	
	$(x + y)(\bar{x} + z) = xz + \bar{x}y$	

1.3 Percobaan

1.3.1 Pengenalan Gugus Gerbang Logika dalam IC

- Alat dan Bahan
 - ⊖ IC tipe :
 - 7404
 - 7408
 - 7432
 - 74HC08
 - 74S03
 - ⊖ Data sheet

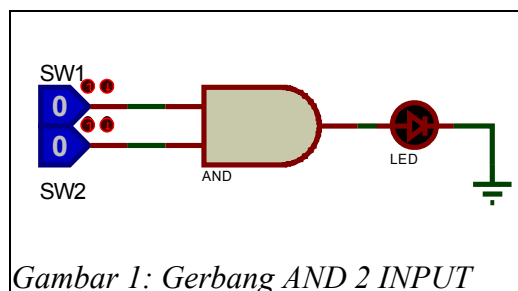
Sebutkan Fungsi dari tipe IC dibawah ini (lihat datasheet):

7404	
7408	
7432	
74HC08	
74S03	

1.3.2 Pengujian Gerbang Logika Dasar pada Tabel Kebenaran

- Alat dan Bahan
 - ⊖ Box Panel Digital.
 - ⊖ IC TTL Tipe :
 - 7404
 - 7408
 - 7432
- Prosedur
 - ⊖ Pengujian gerbang AND 2 INPUT.

(1) Rangkailah IC 7408 pada Box Panel Digital seperti pada Gambar 1.



Gambar 1: Gerbang AND 2 INPUT

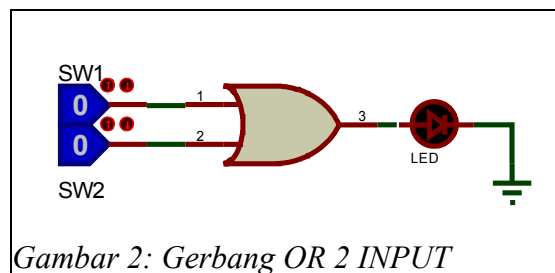
- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output LED pada tabel.

Tabel 1.1 AND gate 2 INPUT

SW1	SW2	Led
0	0	
0	1	
1	0	
1	1	

⇒ **Pengujian gerbang OR 2 INPUT.**

- (1) Rangkailah IC 7432 pada Box Panel Digital seperti pada Gambar 2.



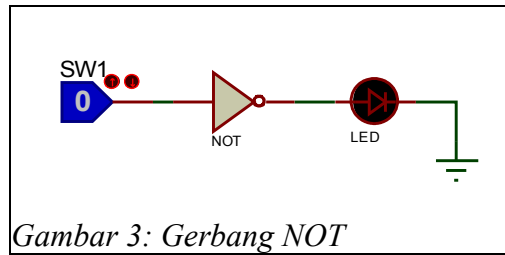
- (2) Nyalakan Box Panel, pastikan semua switch dalam keadaan low.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan cata hasil output LED pada tabel.

Tabel 1.2 OR gate 2 INPUT

SW1	SW2	Led
0	0	
0	1	
1	0	
1	1	

⇒ **Pengujian NOT gate**

- (1) Rangkailah IC 7404 pada Box Panel Digital seperti pada Gambar 3.
- (2) Nyalakan Box Panel, pastikan semua switch dalam keadaan low.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan cata hasil output LED pada tabel.



Gambar 3: Gerbang NOT

Tabel 1.3 NOT gate

SW1	Led
0	
1	

→ Kesimpulan

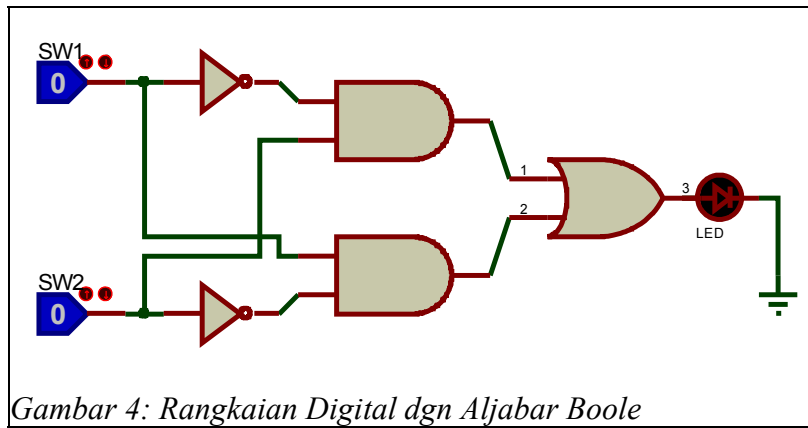
1.3.3 Analisis Rangkaian Digital dalam Aljabar Boole

→ Alat dan Bahan

- ⊖ AND gate 2 input 2 gerbang.
- ⊖ NOT gate 2 input 2 gerbang.
- ⊖ OR gate 1 gerbang.
- ⊖ Box Panel Digital.

→ Prosedur

- (1) Rangkailah gerbang tersebut pada Box Panel Digital seperti pada Gambar 4.
- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led pada tabel.



Tabel 1.4

SW1	SW2	Led
0	0	0
0	1	1
1	0	1
1	1	0

(5) Buatlah persamaan rangkaian dalam bentuk aljabar boole.

(6) Ubahlah rangkaian logika di atas dengan menggunakan gerbang NAND saja dan gerbang NOR saja.

1.3.4 Perancangan Rangkaian Logika

➔ **Alat dan Bahan**

- ⊖ Box Panel Digital.
- ⊖ Gerbang logika dasar

→ **Prosedur**

(1) Carilah persamaan boole dari tabel kebenaran berikut :

Tabel 1.5

Input			Output
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Input			Output
A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

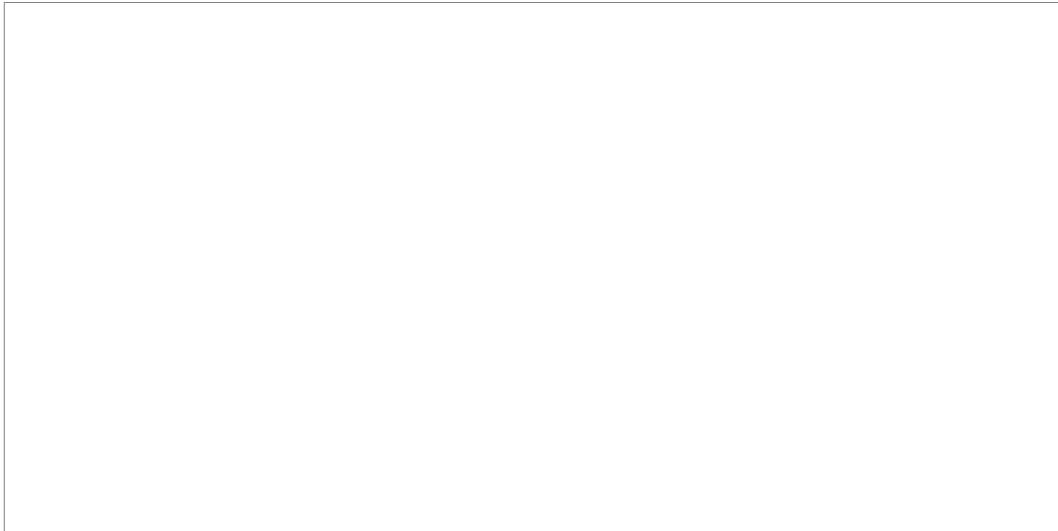
(2) Gambarkan persamaan tersebut dalam rangkaian logika.

(3) Buktikan kebenaran rangkaian tersebut pada Digital Panel Box, tulis output rangkaian pada tabel 1.5

(4) Sederhanakanlah persamaan diatas dengan peta karnaugh dibawah ini.

		ab			
		00	01	11	10
c	0				
	1				

→ Kesimpulan



1.4 Tugas Laporan

1. Sederhanakan fungsi berikut ini dengan menggunakan aljabar boole :

$$X = \overline{AC}(\overline{ABD}) + \overline{ABC}\overline{D} + \overline{ABC}$$

$$Y = \overline{ABC} + \overline{ABD} + (\overline{CD})$$

$$Z = (\overline{C+D}) + \overline{ACD} + \overline{ABC} + \overline{ABCD} + \overline{ACD}$$

Gambarlah bentuk gelombang Input dan Output pada diagram untuk masing – masing gerbang logika.

Untuk soal 2 s.d 7 mengacu pada tabel 1.6.

2. Carilah persamaan boole dari tabel kebenaran berikut:

Tabel 1.6

Input			Output
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	X
1	0	0	X
1	0	1	1
1	1	0	1
1	1	1	1

3. Gambarkan persamaan tersebut dalam rangkaian logika.
4. Buktikan kebenaran rangkaian tersebut pada Digital Panel Box, tulis output rangkaian pada tabel.
5. Sederhanakanlah persamaan diatas dengan peta karnaugh.
6. Gambarkan persamaan tersebut. dalam rangkaian logika.
7. Buktikan kebenaran rangkaian tersebut pada Digital Panel Box, tulis output rangkaian pada tabel.

Bab 2

Adder & Komparator

2.1 Tujuan

- ✓ Mempelajari bentuk-bentuk rangkaian logika kombinasional.
- ✓ Mempelajari cara kerja dan karakteristik Adder dan Komparator.
- ✓ Menganalisis dan merancang rangkaian logika kombinasional.

2.2 Teori

Rangkaian kombinasional adalah rangkaian yang keluarannya ditentukan langsung oleh kombinasi variable masukan tanpa memperhatikan variable masukan sebelumnya. Ada beberapa bentuk rangkaian kombinasi yang biasa digunakan dalam kehidupan : Komparator, Adder, Decoder-Encoder, Multiplexer-Demultiplexer. Rangkaian komparator adalah rangkaian logika yang berfungsi membandingkan keadaan logika input – inputnya. Rangkaian Adder merupakan rangkaian logika kombinasi yang berfungsi melakukan operasi penjumlahan bilangan biner. Adder terbagi menjadi dua, *half Adder* dan *full Adder*. Rangkaian half Adder tidak menyertakan bawaan sebelumnya (previous carry) pada inputnya sedangkan rangkaian full Adder menyertakan bawaan sebelumnya (previous carry).

2.3 Percobaan

2.3.1 Komparator

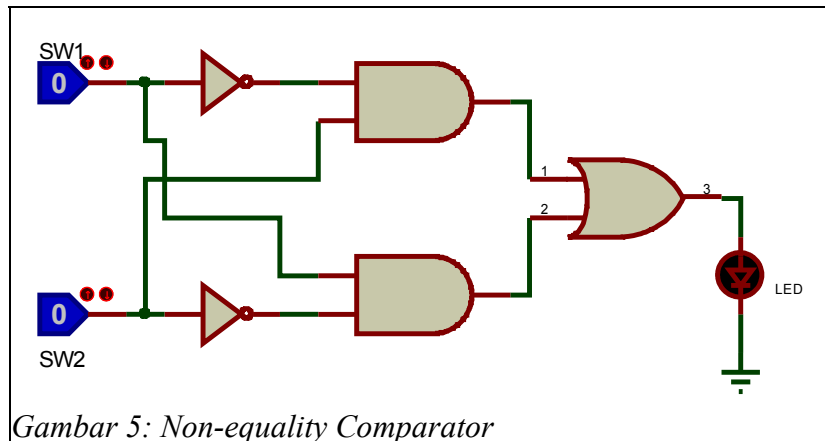
→ **Alat dan Bahan**

- ⊖ Box Panel Digital
- ⊖ IC TTL Tipe :
 - 7404
 - 7408
 - 7432
 - 74HC368

→ **Prosedur**

⊖ **Pengujian Non-Equality Comparator.**

- (1) Rangkailah pada Box Panel Digital seperti pada gambar 5.
- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led pada tabel.

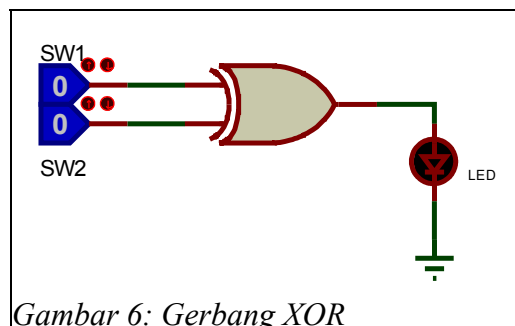


Gambar 5: Non-equality Comparator

Tabel 2.1

SW1	SW2	Led
0	0	
0	1	
1	0	
1	1	

(1) Rangkailah pada Box Panel Digital seperti pada gambar 6.



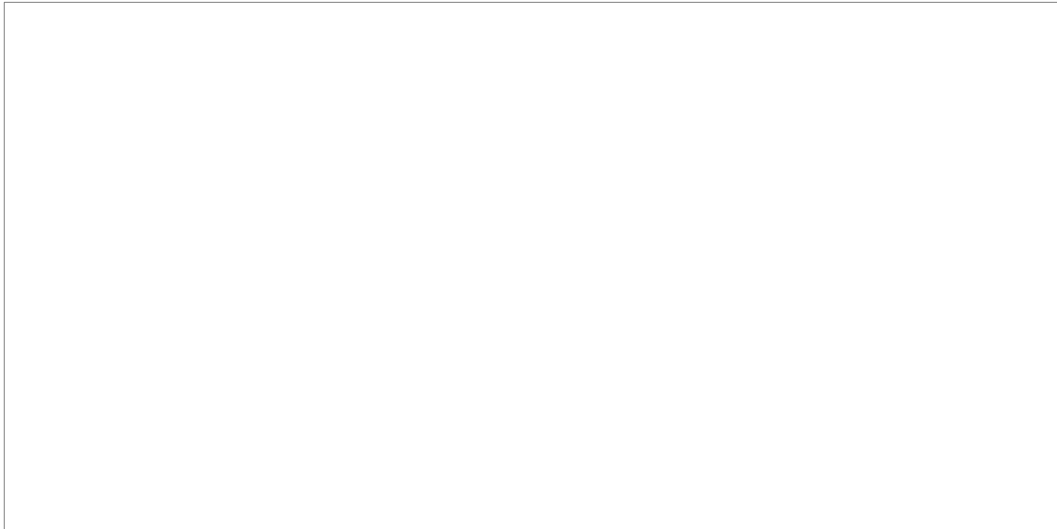
Gambar 6: Gerbang XOR

- (2) Nyalakan (Switch ON) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led pada tabel.

Tabel 2.2

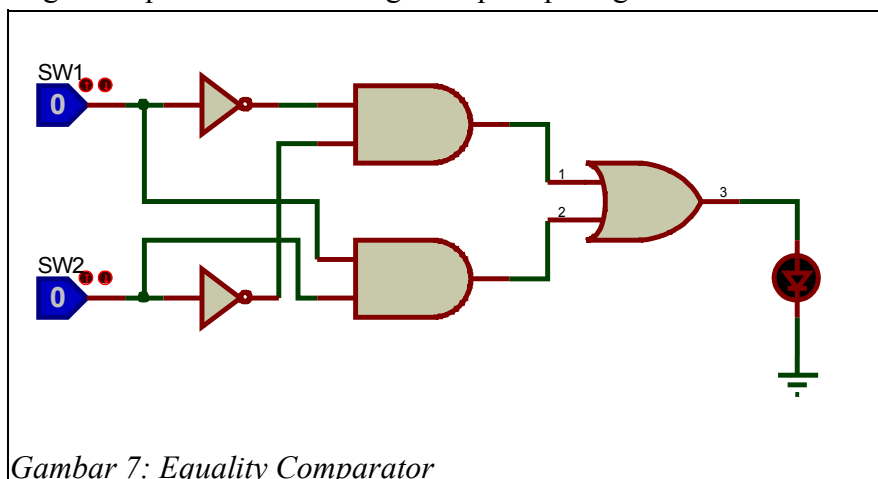
SW1	SW2	Led
0	0	
0	1	
1	0	
1	1	

⇒ Kesimpulan



⇒ Pengujian Equality Comparator.

(1) Rangkailah pada Box Panel Digital seperti pada gambar 7.



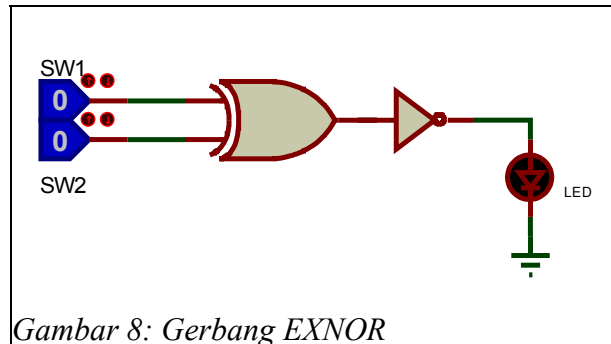
Gambar 7: Equality Comparator

- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led pada tabel.

Tabel 2.3

SW1	SW2	Led
0	0	1
0	1	0
1	0	0
1	1	1

- (1) Rangkailah pada Box Panel Digital seperti pada gambar 8.
- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led pada tabel.



Tabel 2.4

SW1	SW2	Led
0	0	
0	1	
1	0	
1	1	

⇒ **Kesimpulan**

2.3.2 Adder

→ **Alat dan Bahan**

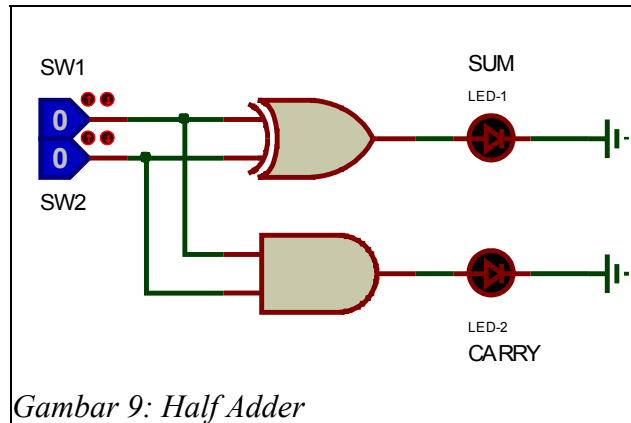
- ⇒ Box Panel Digital
- ⇒ IC TTL Tipe :
 - 7408

- 7432
- 74HC368

→ **Prosedur**

⇒ **Pengujian Half Adder**

(1) Rangkailah pada Box Panel Digital seperti pada gambar 9.



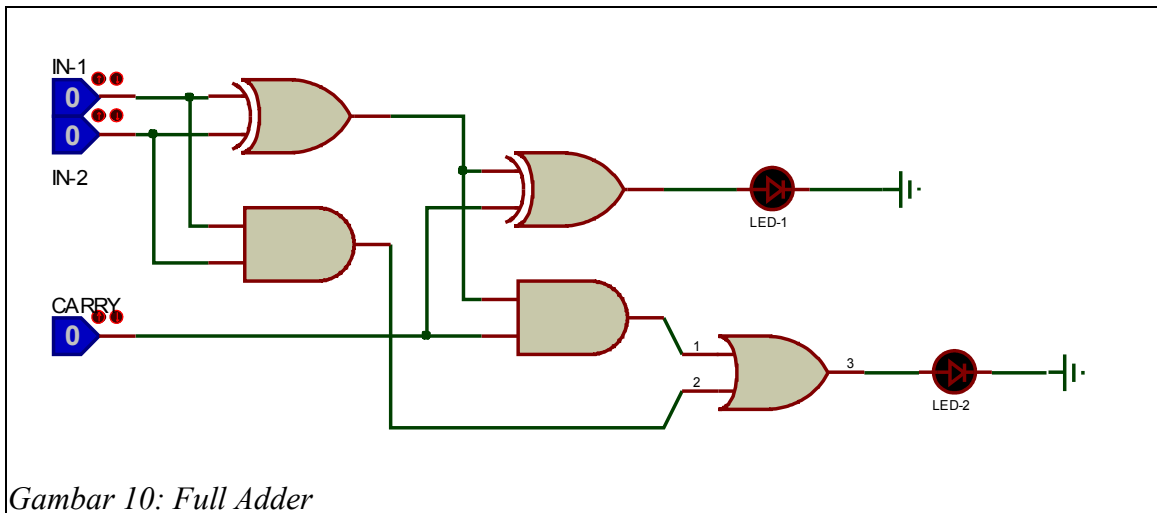
- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led-1 (sum) dan led-2 (carry) pada tabel.

Tabel 2.5

INPUT		OUTPUT	
SW1	SW2	Led-1	Led-2
0	0		
0	1		
1	0		
1	1		

⇒ **Pengujian Full Adder**

- (1) Rangkailah pada Box Panel Digital seperti pada gambar 10.
- (2) Nyalakan (*Switch ON*) Box Panel Digital, pastikan semua switch input (SW1 & SW2) dalam keadaan LOW.
- (3) Atur switch input sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led-1 (sum) dan led-2 (carry) pada tabel.



Gambar 10: Full Adder

Tabel 2.6

INPUT			OUTPUT	
IN-1	IN-2	CARRY	Led-1	Led-2
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

⇒ Kesimpulan

2.4 Tugas Laporan

1. Rangkailah sebuah penjumlah 3 bit dengan OUTPUT S0, S1, dan S2. A0, A1, A2, B0, B1, dan B2 sebagai INPUT-an. C0 sebagai Carry Out.
2. Masukkan beberapa data pada tabel dibawah. Perhatikan output pada led S0, S1, S2, dan C0 lalu catat hasilnya pada tabel.
3. Jelaskan cara kerja Half Adder dan Full Adder.

Tabel 2.7

INPUT A			INPUT B			OUTPUT			
A0	A1	A2	B0	B1	B2	S0	S1	S2	C0
0	1	0	1	0	1				
0	0	1	1	1	1				
0	1	0	1	0	0				
1	1	0	1	1	1				

4. Perhatikan peta karnaugh berikut ini :

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

Implementasikanlah peta karnaugh di samping dengan menggunakan gerbang XOR.

5. Buatlah rangkaian Full Adder 6 bit, lakukan operasi aritmatika berikut (anggap $C_i = 0$).
 - a. $12+4$
 - b. $14+12$
 - c. $8+2$
 - d. $9+7$

Bab 3

Decoder – Encoder Multiplexer - Demultiplexer

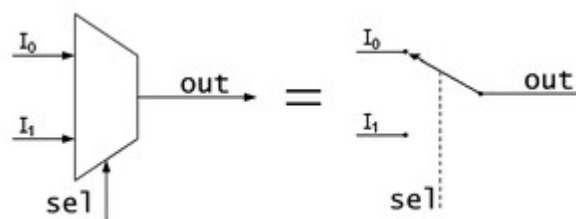
7.1 Tujuan

- ✓ Mempelajari karakteristik Decoder – Encoder.
- ✓ Mempelajari karakteristik Multiplexer – Demultiplexer.
- ✓ Menganalisis dan merancang rangkaian.

7.2 Teori

Decoder merupakan rangkaian logika kombinasi yang berfungsi mengkode ulang atau menafsirkan kode – kode biner yang ada pada inputnya menjadi data asli pada outputnya. Rangkaian decoder dapat dibedakan atas “non-inverted output” dan inverted-output”. Rangkaian decoder “inverted-output” menghasilkan satu dari sejumlah keluarannya yang berlogika 0 sedangkan keluaran lain semua berlogika 1. sedangkan decoder *non-inverted output* sebaliknya. Bentuk decoder antara lain, decoder biner ke octal, BCD ke decimal.

Encoder merupakan rangkaian logika kombinasi yang fungsinya kebalikan dari fungsi decoder. Multiplexer merupakan rangkaian logika kombinasi yang berfungsi sebagai pemilih data yang ada pada inputnya untuk disalurkan ke outputnya dengan bantuan sinyal control. Kata multiplexer sering juga disingkat dengan MUX. Jumlah input multiplexer adalah 2^n ($n=1,2,3,\dots$) dengan n merupakan jumlah bit sinyal pemilih, sehingga terdapat MUX 2 ke-1, MUX 4 ke-1, MUX 8 ke-1, dst.



Gambar 11: Multiplexer

Demultiplexer merupakan rangkaian logika kombinasi yang berfungsi sebagai menyalurkan data yang ada pada inputnya ke salah satu dari beberapa outputnya dengan bantuan sinyal control. Kata demultiplexer sering juga disingkat dengan DEMUX. DEMUX kebalikan dari MUX.

7.3 Percobaan

7.3.1 Decoder

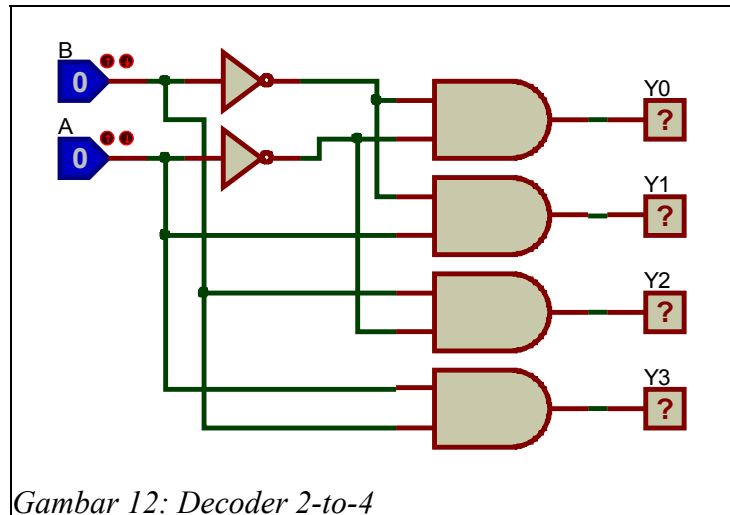
→ **Alat dan Bahan**

- ⊖ Software ISIS-Proteus.

→ **Prosedur**

- ⊖ **Pengujian Decoder 2 X 4**

(1) Rangkailah pada work area seperti pada gambar 12.



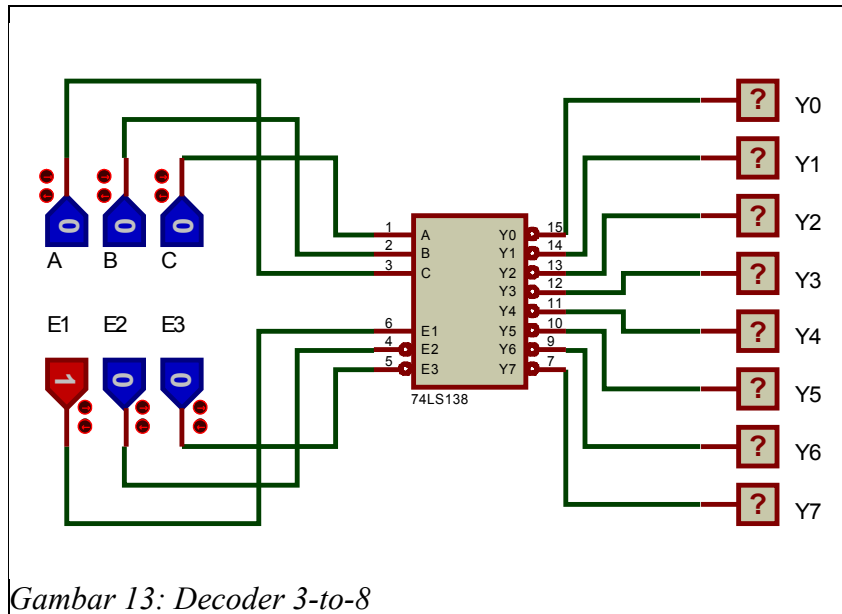
- (2) Lakukan Debugging/Start simulasi.
- (3) Atur switch input (A dan B) sesuai dengan tabel kebenaran dibawah ini.
- (4) Lakukan percobaan berikut dan catat hasil output led (Y0, Y1, Y2, Y3) pada tabel.

Tabel 2.8

INPUT		OUTPUT			
A	B	Y0	Y1	Y2	Y3
0	0				
0	1				
1	0				
1	1				

- ⊖ **Pengujian Decoder 3 X 8**

- (1) Rangkailah pada work area seperti pada gambar dibawah.
- (2) Atur switch enable (E1, E2, E3) sesuai dengan konfigurasi berikut E1 = **High**, E2 = **Low**, E3 = **Low**.
- (3) Lakukan Debugging/Start simulasi.
- (4) Atur switch input (A, B, dan C) sesuai dengan tabel kebenaran dibawah ini.



Gambar 13: Decoder 3-to-8

(5) Lakukan percobaan berikut dan catat hasil output led (Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7) pada tabel.

Tabel 2.9

INPUT			OUTPUT							
A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

→ Kesimpulan

7.3.2 Encoder

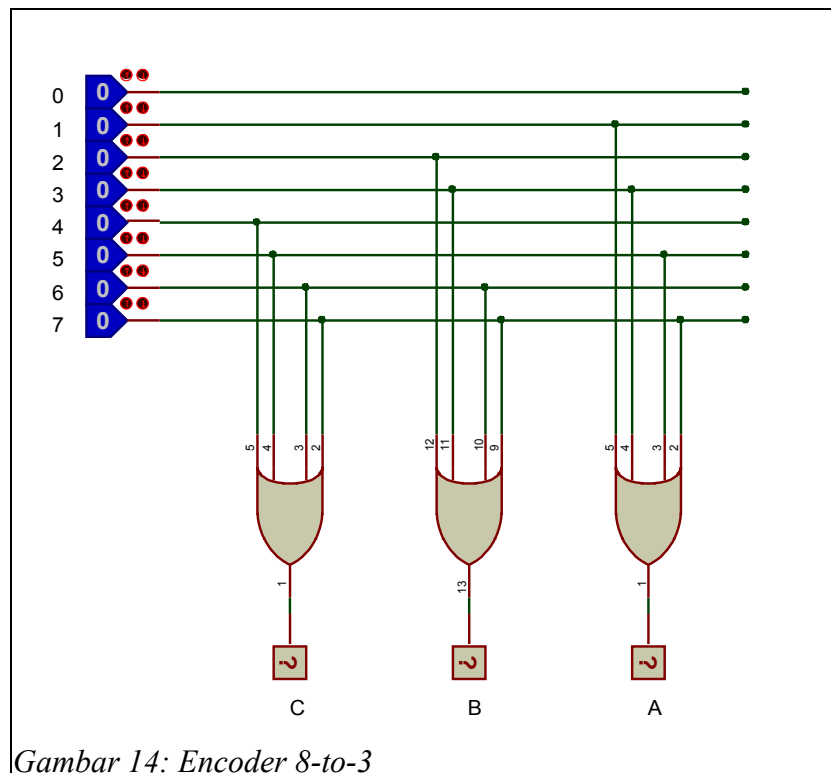
→ Alat dan Bahan

- ⊖ Software ISIS-Proteus.

→ Prosedur

- ⊖ Pengujian Encoder 8 ke 3

- (1) Rangkailah pada work area seperti pada gambar 14.
- (2) Pastikan semua switch input dalam keadaan **Low**.
- (3) Lakukan Debugging/Start simulasi.
- (4) Atur switch input (0..7) sesuai dengan tabel kebenaran dibawah ini.



Gambar 14: Encoder 8-to-3

- (5) Lakukan percobaan berikut dan catat hasil output led (A, B, C) pada tabel.

Tabel 3.0

INPUT			OUTPUT							
A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
			1	0	0	0	0	0	0	0
			0	1	0	0	0	0	0	0
			0	0	1	0	0	0	0	0
			0	0	0	1	0	0	0	0
			0	0	0	0	1	0	0	0
			0	0	0	0	0	1	0	0
			0	0	0	0	0	0	1	0
			0	0	0	0	0	0	0	1

7.3.3 Multiplexer

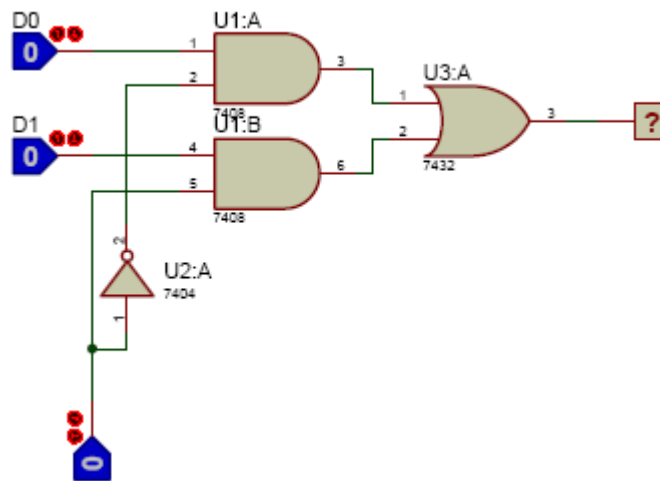
→ **Alat dan Bahan**

- ⊖ Software ISIS-Proteus.

→ **Prosedur**

- ⊖ **Pengujian Multiplexer**

- (1) Rangkailah pada work area seperti pada gambar 15.
- (2) Pastikan semua switch input dalam keadaan **Low**.
- (3) Lakukan Debugging/Start simulasi.
- (4) Atur switch input (D0, D1, dan Clk) sesuai dengan tabel kebenaran dibawah ini.
- (5) Lakukan percobaan berikut dan catat hasil pada tabel



Gambar 15: Multiplexer

Tabel 34

INPUT			OUTPUT
D0	D1	Clk	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

7.4 Tugas Laporan

1. Rangkailah sebuah Decoder biner ke octal, BCD ke decimal, dan BCD ke 7-segment.
2. Rangkailah sebuah Encoder octal ke biner.
3. Buktikan kebenaran keluarannya.
4. Catat hasilnya dalam table berikut.

INPUT			OUTPUT							
A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
			1	0	0	0	0	0	0	0
			0	1	0	0	0	0	0	0
			0	0	1	0	0	0	0	0
			0	0	0	1	0	0	0	0
			0	0	0	0	1	0	0	0
			0	0	0	0	0	1	0	0
			0	0	0	0	0	0	1	0
			0	0	0	0	0	0	0	1

5. Rangkailah sebuah multiplexer 4 X 1

Bab 4

Flip - Flop

4.1 Tujuan

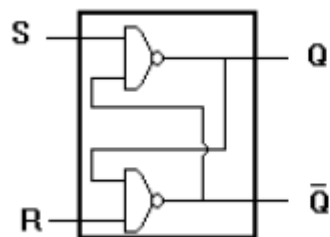
- ✓ Mengetahui, mengerti dan memahami operasi dasar rangkaian Flip – Flop.
- ✓ Mengetahui berbagai macam IC Flip – Flop.

4.2 Teori

Pemahaman terhadap rangkaian Flip-Flop (FF) ini sangat penting karena FF merupakan satu sel memori. Keadaan keluaran FF dapat berada dalam keadaan tinggi atau keadaan rendah, untuk selang waktu yang dikehendaki. Biasanya untuk mengubah keadaan tersebut diperlukan suatu masukan pemicu. Berikut ini akan diuraikan secara singkat tentang berbagai tipe FF.

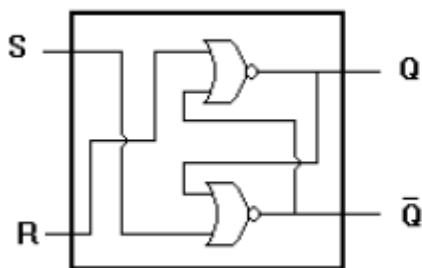
4.2.1 Flip-Flop SR

Flip-Flop SR merupakan rangkaian dasar untuk menyusun berbagai jenis FF yang lainnya. FF-SR dapat disusun dari dua gerbang NAND atau dua gerbang NOR.



S	R	Q_{n+1}
0	1	1
1	0	0
1	1	Q_n
0	0	Don't Care

Gambar 16: Flip-Flop SR dgn gerbang NAND

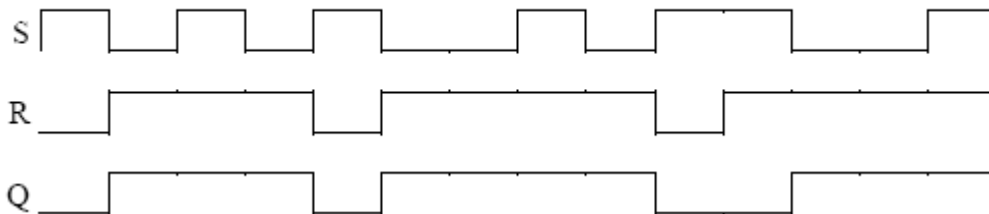


S	R	Q_{n+1}
0	1	0
1	0	1
0	0	Q_n
1	1	Don't Care

Gambar 17: Flip-Flop SR dgn gerbang NOR

Mengeset FF berarti membuat keluaran $Q = 1$ dan mereset FF berarti membuat keluaran $Q = 0$ dari kondisi stabil/ tak berubah. Mengeset FF dari gerbang NAND dapat dilakukan dengan membuat $S = 0$ dan mereset dilakukan dengan membuat $R = 0$. Sedangkan mengeset FF dari gerbang NOR dapat dilakukan dengan membuat $S = 1$ dan mereset dengan memberi nilai $R = 1$.

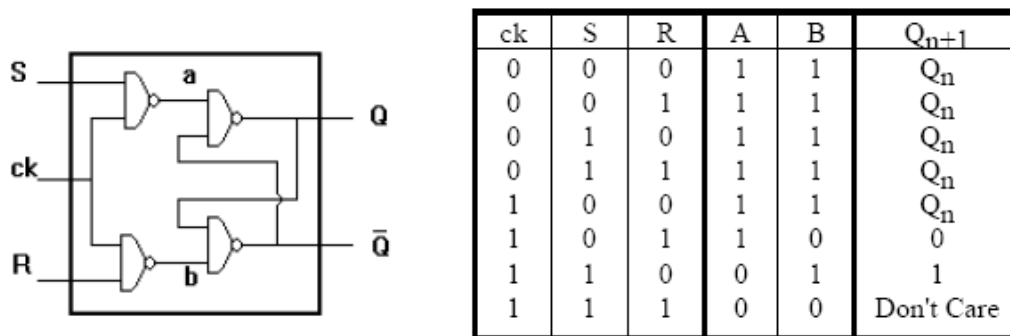
Gambar berikut ini melukiskan bentuk keluaran dari FF SR dengan menggunakan gerbang NAND.



Gambar 18: Diagram Waktu Flip-Flop SR

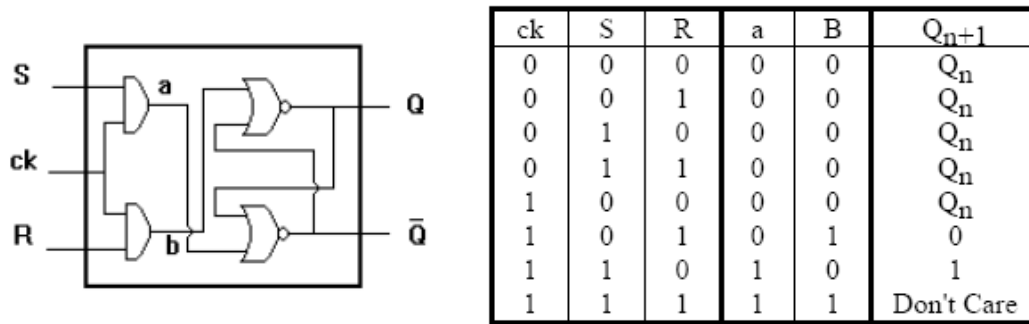
4.2.2 Clocked Flip-Flop SR

FF jenis ini dapat dirangkai dari FF-SR ditambah dengan dua gerbang AND atau NAND untuk masukan pemacu yang disebut dengan sinyal clock (ck).

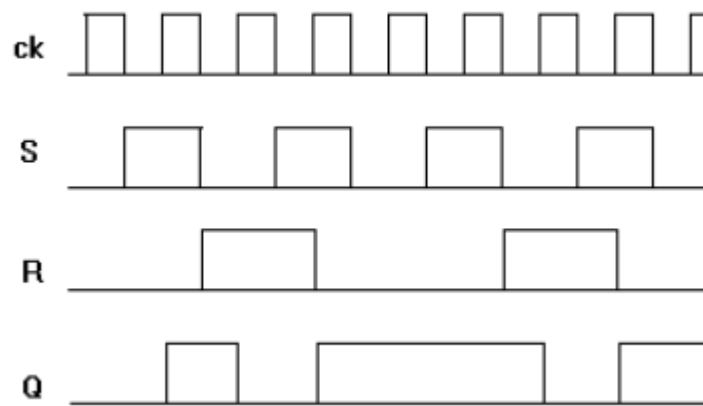


Gambar 19: Clocked Flip-Flop SR dgn Gerbang NAND

Dari tabel kebenaran kedua rangkaian diatas terlihat bahwa untuk sinyal clock yang tinggi, FF ini bekerja seperti FF-SR dari gerbang NOR, sedangkan untuk sinyal clock yang rendah, keluaran Q tidak bergantung kepada input R dan S, tetapi tetap mempertahankan keadaan terakhir sampai datangnya sinyal clock berikutnya. Sebagai ilustrasi, berikut ini diberikan contoh bentuk sinyal Q.



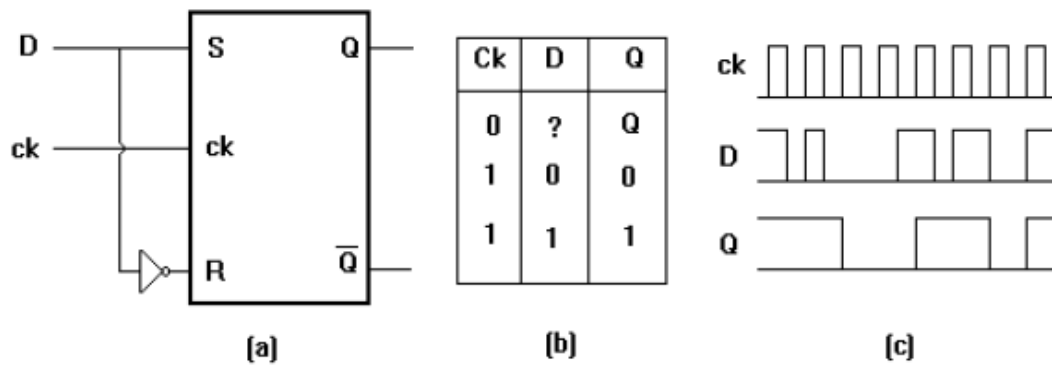
Gambar 20: Clocked Flip-Flop SR dgn gerbang NOR



Gambar 21: Diagram waktu Clocked Flip-Flop SR

4.2.3 Flip-Flop D

Pada FF-SR ada nilai-nilai masukan yang terlarang. Untuk menghindari adanya nilai terlarang tersebut, disusun suatu jenis FF lain yang dinamakan FF Data. Rangkaian ini dapat diperoleh dengan menambahkan satu gerbang NOT pada masukan FF terlonceng sebagai berikut:

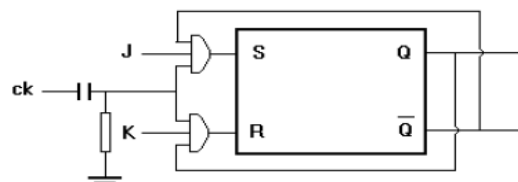


Gambar 22: Flip-Flop D

Dari gambar tersebut terlihat bahwa untuk sinyal clock yang rendah, keluaran Q akan tetap "terkunci" atau "tergerendel" pada nilai terakhirnya. Dalam hal ini dapat dikatakan bahwa pada saat kondisi clock rendah, sinyal masukan D tidak mempengaruhi keluaran Q. Sedangkan untuk sinyal clock yang tinggi, maka akan diperoleh keluaran sesuai dengan data D yang masuk saat itu.

4.2.4 Flip-Flop JK

FF JK mempunyai masukan "J" dan "K". FF ini "dipicu" oleh suatu pinggiran pulsa clock positif atau negatif. FF JK merupakan rangkaian dasar untuk menyusun sebuah pencacah. FF JK dibangun dari rangkaian dasar FF-SR dengan menambahkan dua gerbang AND pada masukan R dan S serta dilengkapi dengan rangkaian diferensiator pembentuk denyut pulsa clock seperti yang ditunjukkan pada gambar di bawah ini.



Gambar 23: Flip-Flop JK

Pada FF JK ini, masukan J dan K disebut masukan pengendali karena kedua masukan ini yang menentukan keadaan yang harus dipilih oleh FF pada saat pulsa clock tiba (dapat pinggiran positif atau negatif, tergantung kepada jenis FFnya). FF ini berbeda dengan FF-D karena pada FF-JK masukan clock adalah masukan yang dicacah, dan masukan J serta K adalah masukan yang mengendalikan FF itu. Cara kerja dari FF-JK adalah sebagai berikut :

- A. Pada saat J dan K keduanya rendah, gerbang AND tidak memberikan tanggapan sehingga keluaran Q tetap bertahan pada keadaan terakhirnya.
- B. Pada saat J rendah dan K tinggi, maka FF akan diseret hingga diperoleh keluaran $Q = 0$ (kecuali jika FF memang sudah dalam keadaan reset atau Q memang sudah pada keadaan rendah).
- C. Pada saat J tinggi dan K rendah, maka masukan ini akan mengeset FF hingga diperoleh keluaran $Q = 1$ (kecuali jika FF memang sudah dalam keadaan set atau Q sudah dalam keadaan tinggi).
- D. Pada saat J dan K kedua-duanya tinggi, maka FF berada dalam keadaan "toggle", artinya keluaran Q akan berpindah pada keadaan lawan jika pinggiran pulsa clocknya tiba.

4.3 Percobaan

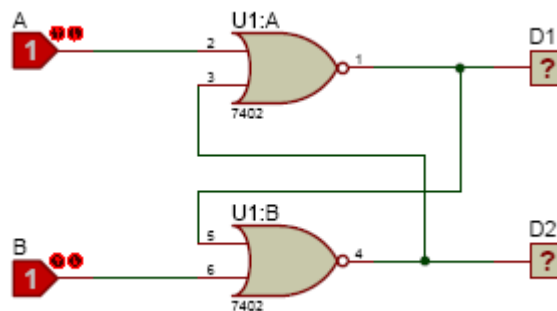
4.3.1 Flip-Flop SR

→ **Alat dan Bahan**

⇒ Software ISIS-Proteus.

→ **Prosedur**

- (1) Rangkailah pada work area seperti pada gambar dibawah ini.
- (2) Atur logicstate pada poisi 1
- (3) Lakukan Debugging/Start simulasi.
- (4) Atur switch input (A dan B) sesuai dengan tabel kebenaran dibawah ini.



Gambar 24: Pengujian Flip-Flop SR dgn gerbang NOR

Tabel 4.1

INPUT		OUTPUT	
A	B	D1	D2
0	0		
0	1		
1	0		
1	1		

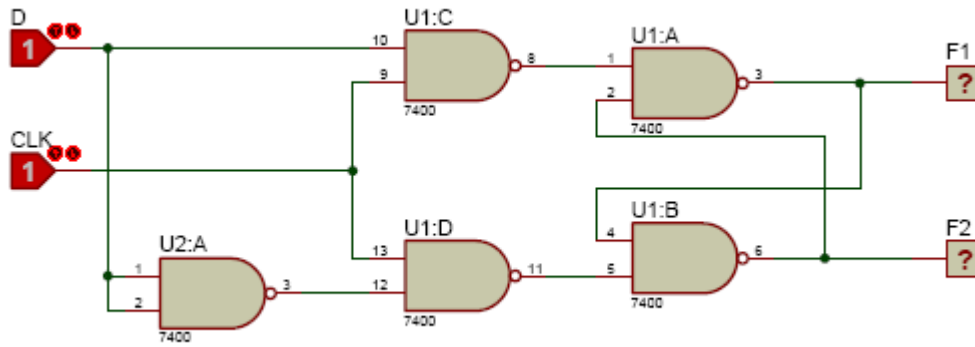
4.3.2 Flip-Flop D

→ **Alat dan Bahan**

⇒ Software ISIS-Proteus.

→ **Prosedur**

- (1) Rangkailah pada work area seperti pada gambar dibawah ini.
- (2) Atur logicstate pada poisi 1
- (3) Lakukan Debugging/Start simulasi.
- (4) Atur switch input (D dan Clk) sesuai dengan tabel kebenaran dibawah ini.



Gambar 25: Pengujian Flip-Flop D dgn gerbang NAND

Tabel 4.2

INPUT		OUTPUT	
A	B	D1	D2
0	0		
0	1		
1	0		
1	1		

→ Kesimpulan

4.4 Tugas Laporan

(1) Rangkailah sebuah Flipflop D Master Slave dengan menggunakan flipflop JK