



PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI  
DAN INFORMATIKA

MODUL PRAKTIKUM

# SISTEM DIGITAL

Yusuf Sulistyo Nugroho, ST., M.Eng.  
Jan Wantoro, S.T., M.Eng.  
Diah Priyawati, S.T., M.Eng.  
Maryam, ST, M.Eng  
Maulana Luthfi Sholihin



# PETUNJUK PRAKTIKUM

# SISTEM DIGITAL

**Disusun oleh :**

Yusuf Sulistyo Nugroho, ST., M.Eng.

Jan Wantoro, S.T., M.Eng.

Diah Priyawati, S.T., M.Eng.

Maryam, ST, M.Eng

Maulana Luthfi Sholihin



2020

# **SISTEM DIGITAL**

## **Modul Praktikum**

Penulis : 1. Yusuf Sulistyo Nugroho, ST., M.Eng.  
2. Jan Wantoro, S.T., M.Eng.  
3. Diah Priyawati, S.T., M.Eng.  
4. Maryam, ST, M.Eng  
5. Maulana Luthfi Sholihin

Layouter : T. Santosa

Desain cover : T. Santosa

ISBN: 978-602-361-271-0

Cetakan 1, Januari 2020

©2019 Hak cipta pada penulis dan dilindungi undang-undang

Penerbit Muhammadiyah University Press

Universitas Muhammadiyah Surakarta

Gedung i Lantai 1

Jl. A Yani Pabelan Tromol Pos 1 Kartasura Surakarta 57102

Jawa Tengah - Indonesia

Telp: (0271) 717417 Eks. 2172

Email: [mupress@ums.ac.id](mailto:mupress@ums.ac.id)

## KATA PENGANTAR

---

Alhamdulillah, puji dan syukur dipanjatkan ke hadirat Allah SWT yang telah memberikan kemampuan dan kesempatan kepada Penulis untuk menyusun petunjuk praktikum “system digital” ini. Buku petunjuk praktikum ini disusun berdasarkan kurikulum 2015 dan diberlakukan mulai tahun ajaran 2016-2017. Buku petunjuk praktikum system digital ini mengalami beberapa pengembangan di tahun-tahun berikutnya.

Buku petunjuk praktikum ini menggunakan aplikasi Proteus sebagai aplikasi pendukung untuk simulasi rangkaian system digital. Proteus memudahkan mahasiswa memahami beberapa komponen rangkaian gerbang logika hingga kombinasi kompleks. Database komponen Proteus juga cukup lengkap. Kemudian mahasiswa dapat merangkai komponen-komponen system digital pada ISIS *schematic* Proteus sampai output rangkaian ketika dijalankan.

Secara umum buku petunjuk praktikum system digital ini membahas rangkaian gerbang logika dan implementasinya seperti rangkaian flip-flop, decoder, counter, register hingga aplikasi tingkat lanjut seperti penghitung waktu. Modul praktikum ini diawali dari pengenalan aplikasi pendukung, mengenal alat ukur rangkaian digital, mengenal gerbang logika dan kombinasinya, mengenal rangkaian flip-flop, mengenal rangkaian decoder dan counter, mengenal rangkaian register, hingga mahasiswa mampu membuat aplikasi stopwatch atau jam digital.

Penulis berharap buku ini dapat bermanfaat bagi praktikan dan kalangan lain yang berkaitan dengan praktikum system digital. Tak lupa penulis mengucapkan terima kasih kepada pihak-pihak yang membantu terselesainya buku ini, mulai dari pimpinan Program Studi Informatika, sejawat dosen terutama pengajar mata kuliah “Sistem Digital”, asisten dan staf. Terakhir, kritik dan saran sangat penulis harapkan demi kesempurnaan buku petunjuk ini.

Surakarta, November 2019

Penulis

## **DAFTAR ISI**

---

Sampul .....	i
Kata Pengantar.....	III
Daftar Isi .....	iv
<b>MODUL 1 : MENGGUNAKAN PROTEUS 8 .....</b>	<b>1</b>
<b>Tujuan Praktikum.....</b>	<b>1</b>
<b>Teori.....</b>	<b>1</b>
ISIS Schematic Capture .....	2
<b>Kegiatan Praktikum .....</b>	<b>6</b>
Percobaan 1 : Latihan Membuat Rangkaian .....	6
<b>Tugas.....</b>	<b>11</b>
<b>MODUL 2 : PENGENALAN SINYAL .....</b>	<b>13</b>
<b>Tujuan Praktikum.....</b>	<b>13</b>
<b>Teori.....</b>	<b>13</b>
Osiloskop .....	13
<b>Kegiatan Praktikum .....</b>	<b>18</b>
Percobaan 1 : Latihan Jenis-jenis Sinyal .....	18
Percobaan 2 : Latihan Range Sinyal Digital .....	21
<b>MODUL 3 : GERBANG LOGIKA.....</b>	<b>23</b>
<b>Tujuan Praktikum.....</b>	<b>23</b>
<b>Teori.....</b>	<b>23</b>
<b>Kegiatan Praktikum .....</b>	<b>25</b>
Percobaan 1 : Gerbang AND.....	25
Percobaan 2 : Gerbang OR.....	26
Percobaan 3 : Gerbang NOT .....	27
Percobaan 4 : Gerbang NAND .....	28
Percobaan 5 : Gerbang NOR.....	29
Percobaan 6 : Gerbang XOR .....	30
Percobaan 7 : Gerbang XNOR.....	31

<b>MODUL 4 : KOMBINASI GERBANG LOGIKA .....</b>	33
<b>Tujuan Praktikum.....</b>	33
<b>Teori.....</b>	33
<b>Kegiatan Praktikum .....</b>	36
Percobaan 1 : Substitusi Pengganti Gerbang NOT .....	36
Percobaan 2 : Substitusi Pengganti Gerbang OR.....	37
Percobaan 3 : Substitusi Pengganti Gerbang AND .....	38
Percobaan 4 : Substitusi Pengganti Gerbang XNOR.....	39
Percobaan 5 : Merancang fungsi boolean ke dalam rangkaian ....	40
 <b>MODUL 5 : ALJABAR BOOLEAN.....</b>	41
<b>Tujuan Praktikum.....</b>	41
<b>Teori.....</b>	41
Aljabar Boolean .....	41
Pengenalan Peta Karnaugh.....	44
<b>Kegiatan Praktikum .....</b>	46
Percobaan 1 .....	46
Percobaan 2 .....	47
Percobaan 3 .....	48
 <b>MODUL 6 : PETA KARNAUGH.....</b>	51
<b>Tujuan Praktikum.....</b>	51
<b>Teori.....</b>	51
Peta Karnaugh.....	51
<b>Kegiatan Praktikum .....</b>	55
Percobaan 1 .....	55
Percobaan 2 .....	56
Percobaan 3 .....	57
Percobaan 4 .....	58
 <b>MODUL 7 : FLIP-FLOP DASAR.....</b>	59
<b>Tujuan Praktikum.....</b>	59

<b>Teori.....</b>	59
RS Flip Flop (RS – FF) .....	61
RS Flip Flop dengan Clock.....	62
D Flip Flop (D – FF) .....	63
JK Flip Flop (JK – FF) .....	64
JK Flip Flop Master/Slave.....	65
T Flip Flop (T – FF).....	66
<b>Kegiatan Praktikum .....</b>	67
Percobaan 1 : NOR Latch .....	67
Percobaan 2 : NAND Latch.....	68
Percobaan 3 : Flip Flop RS.....	69
Percobaan 4 : Flip Flop D .....	70
Percobaan 5 : Flip Flop JK .....	71
 <b>MODUL 8 : APLIKASI FLIP-FLOP .....</b>	73
<b>Tujuan Praktikum.....</b>	73
<b>Teori.....</b>	73
<b>Kegiatan Praktikum .....</b>	78
Percobaan 1 : Membuat Counter JK – FF .....	78
Percobaan 2 : Counter Mod10.....	79
Percobaan 3 : Membuat Register JK-FF.....	80
 <b>MODUL 9 : DECODER .....</b>	81
<b>Tujuan Praktikum.....</b>	81
<b>Teori.....</b>	81
Decoder BCD-To_Seven Segment.....	82
<b>Kegiatan Praktikum .....</b>	84
Percobaan 1 : Membuat Rangkaian Decoder Sederhana .....	84
Percobaan 2 .....	86
Percobaan 3 .....	86
Percobaan 4 : IC 7442 Decoder BCD-to-Decimal .....	87
<b>Tugas.....</b>	89

<b>MODUL 10 : IMPLEMENTASI DECODER .....</b>	91
<b>Tujuan Praktikum.....</b>	91
<b>Teori.....</b>	91
<b>Kegiatan Praktikum .....</b>	93
Percobaan 1 : Rangkaian Clock Counter.....	93
Percobaan 2 : Penambahan Decoder BCD-to-7segment .....	94
Percobaan 3 : Melihat di dalam BCD-to-7segment Decoder .....	96
<b>Tugas.....</b>	98
<b>MODUL 11 : REGISTER.....</b>	99
<b>Tujuan Praktikum.....</b>	99
<b>Teori.....</b>	99
Register Geser Seri .....	99
Register Geser Pararel.....	101
Register Geser Universal IC 74194.....	102
<b>Kegiatan Praktikum .....</b>	104
Percobaan 1 : Rangkaian Register 4-bit.....	104
Percobaan 2 : Rangkaian Register IC 74194 .....	105
Percobaan 3 : Rangkaian Register Geser Kiri 1 .....	106
Percobaan 4 : Rangkaian Register Geser Kiri 2 .....	108
<b>MODUL 12 : TINGKAT LANJUT (PENGHITUNG WAKTU) ....</b>	111
<b>Tujuan Praktikum.....</b>	111
<b>Teori.....</b>	111
<b>Kegiatan Praktikum .....</b>	113
Percobaan 1 : Rangkaian Modulus 10 dan Decoder untuk Stopwatch .....	113
Percobaan 2 : Rangkaian Modulus 6 dan Decoder untuk Stopwatch .....	114
Percobaan 3: Jam Digital.....	115
<b>DAFTAR PUSTAKA.....</b>	117

©Muhammadiyah University Press

# MODUL I

## MENGGUNAKAN PROTEUS 8

### TUJUAN PRAKTIKUM

Membiasakan mahasiswa dengan dasar-dasar desain skematik, memilih komponen dari pustaka, menempatkan komponen pada skema dan menghubungkan komponen.

### TEORI

Proteus 8 adalah aplikasi tunggal yang mempunyai banyak modul layanan yang menawarkan fungsi berbeda-beda seperti desain skematik, layout PCB, dan lain-lain. Aplikasi ini terdiri dari tiga bagian utama yang memungkinkan beberapa alat untuk berkomunikasi satu sama lain. Tiga bagian utama tersebut adalah sebagai berikut :

#### 1. *Application Framework*

Proteus 8 terdiri dari sebuah aplikasi tunggal (PDS.EXE). *Framework* atau dapat disebut juga sebagai kontainer dimana semua fungsi-fungsi Proteus berada. ISIS, ARES, 3DV merupakan fungsi-fungsi tersebut yang akan aktif pada *framework* ini dan mempunyai akses ke *common database*.

#### 2. *Common Database*

*Common database* terdiri dari informasi tentang bagian-bagian yang digunakan dalam project. Bagian ini terdiri dari komponen skematik dan PCB serta sistem properti pengguna (user). Pengaksesan pada database ini dapat digunakan bersama-sama oleh semua modul aplikasi sehingga memungkinkan adanya berbagai fitur-fitur baru.

#### 3. *Live Netlist*

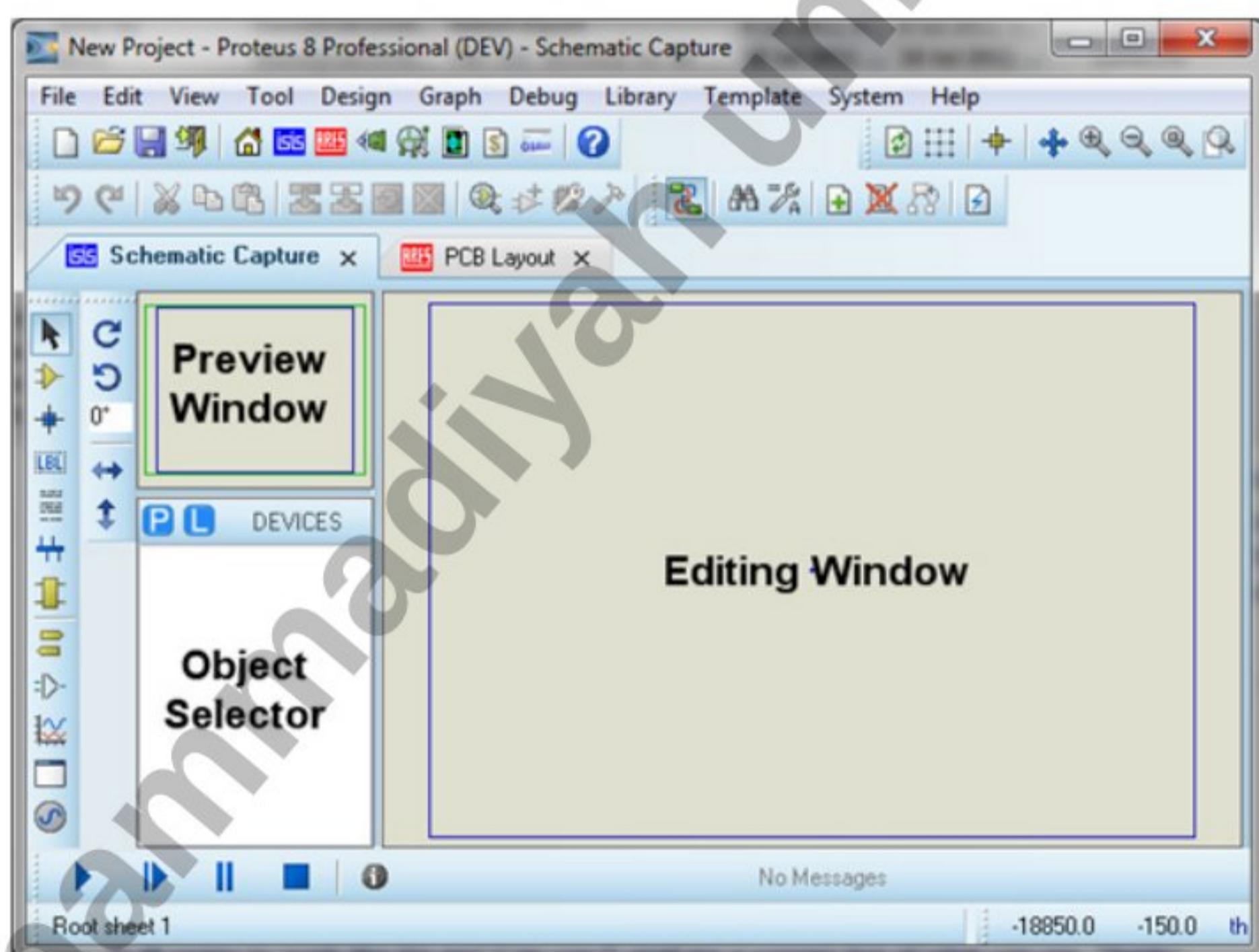
Bersama dengan *common database*, *live netlist* memperbolehkan semua modul yang aktif untuk dapat berubah secara otomatis. Dalam hal ini contoh paling jelas adalah pemasangan pada ISIS yang menghubungkan juga pada ARES. Bagian baru dari modul ini yaitu

mengandung alat untuk melihat komponen (*viewer*) secara langsung dan melihat komponen secara 3D.

## ISIS Schematic Capture

Modul skematik pada Proteus yang dikenal dengan ISIS terletak di jantung sistem Proteus. ISIS tidak seperti paket skematik pada umumnya, karena ISIS mempunyai desain kombinasi yang handal untuk menentukan aspek-aspek dalam menggambar tampilan. ISIS juga sesuai untuk membuat desain yang komplek dengan simulasi dan layout PCB.

ISIS mempunyai *editing window* yang merupakan jendela terbesar untuk melakukan perancangan dan penggambaran yang mana komponen diletakkan dan dihubungkan satu sama lain. *Overview window* merupakan jendela kecil berada di pojok kiri atas berfungsi sebagai gambaran dari seluruh perancangan. Kotak biru memperlihatkan garis tepi dari project yang aktif dan sementara itu kotak hijau ditampilkan pada *editing window*. Saat komponen dipilih dari jendela *object selector*, *overview window* beralih fungsi menjadi penampilan gambar dari komponen tersebut.

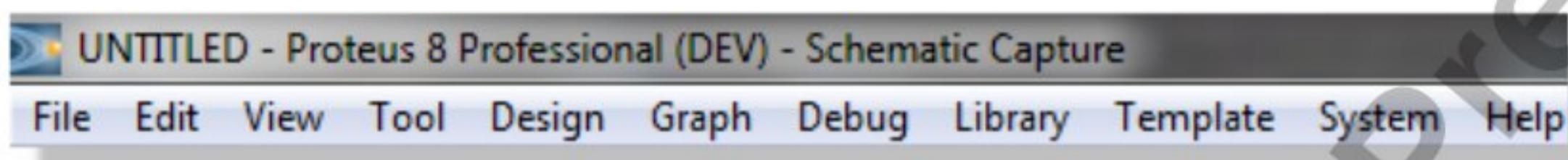


Gambar 1.1. Jendela desain ISIS

Apabila anda tidak menyukai tampilan standar dari *toolbar* diatas maka anda mengaturnya sendiri dengan cara mengambil yang diperlukan dan menempatkannya di salah satu dari empat sisi aplikasi. Begitu juga dengan jendela *object selector* dan *overview* yang dapat dipindahkan ke kanan dengan cara menariknya melewati sisi yang lain.

## 1. Menu Bar

Menubar berada di baris paling atas pada jendela utama yang berisi perintah-perintah dari menu program seperti aplikasi pada umumnya. Selain itu, area baru yang berada di atas nama-nama menu digunakan untuk menampilkan pesan cepat yang mengindikasikan program telah memasuki tahap editing atau perancangan.



Gambar 1.2. Menu bar pada ISIS

## 2. ToolBar

Seperti halnya jendela aplikasi lainnya, ISIS menyediakan akses ke sejumlah *toolbar* seperti *toolbar command*, *mode selector*, atau *orientation*. *Toolbar* tersebut dapat digunakan dengan menarik item-item yang ada didalamnya ke dalam jendela aplikasi ISIS. Berikut ini adalah ketiga *toolbar* tersebut :

### a. *Toolbar Command*

Toolbar ini umumnya berada pada bagian atas sepanjang jendela program sebagai cara lain mengakses menu. Berikut ini adalah bagian-bagian dari toolbar ini.

TITLE	TOOLBAR
File / Project Commands	
Application Commands	
Display Commands	
Editing Commands	
Design Tools	

Gambar 1.3. *Toolbar Command*

### b. *Toolbar Mode Selector*

Sedangkan toolbar mode selector berada di sebelah kiri dari jendela utama dan menjelaskan kondisi yang terjadi ketika mouse memilih komponen pada *editing window*. Berikut ini adalah contoh *toolbar mode selector*.

TITLE	TOOLBAR
Main Modes	
Gadgets	
2D Graphics	

Gambar 1.4. Mode Selector Toolbar

### c. Toolbar Orientasi

Toolbar orientasi berfungsi untuk menampilkan dan mengontrol gerakan memutar (rotasi) serta menjadi refleksi komponen saat diletakkan pada *layout*. Berikut adalah toolbar orientasi. Dalam menu rotasi, memungkinkan untuk memasukkan sudut rotasi secara langsung namun dalam ISIS hanya sudut ortogonal yang dapat dimasukkan. Dan ketika sebuah komponen dipilih, ikon rotasi dan refleksi berubah menjadi merah yang menunjukkan bahwa menu-menu tersebut akan mengubah orientasi komponen pada *layout*-nya.

TITLE	TOOLBAR
Rotation	
Reflection	

Gambar 1.5. Toolbar Orientasi

## 3. Navigasi

Terdapat dua bentuk navigasi yang ditampilkan pada jendela pengeditan (*editing window*) yaitu pengaturan skala gambar (*zooming*) dan pengaturan area gambar yang ditampilkan (*panning*). Berikut ini adalah penjelasan secara rinci dari teknik-teknik tersebut.

### a. Zooming

Ada beberapa cara untuk memperbesar atau memperkecil area dari skematik, antara lain:

- Arahkan mouse pada area yang ingin diperbesar atau diperkecil, kemudian putar tombol tengah mouse (putaran kedepan untuk memperbesar, dan putaran kebelakang untuk memperkecil)
- Arahkan mouse pada area yang ingin diperbesar atau diperkecil, kemudian tekan tombol F6 atau F7.
- Tekan dan tahan tombol Shift kemudian geser keluar kotak menggunakan tombol kiri pada mouse di daerah yang ingin diperbesar. Langkah ini disebut shift zoom.
- Langkah termudah dengan menggunakan toolbar zoom yang terdiri dari memperbesar (*zoom in*), memperkecil (*zoom out*), memperbesar seluruh komponen (*zoom all*), dan memperbesar area tertentu (*zoom area*). Gambar 1.6. merupakan *toolbar* untuk *zooming*.



Gambar 1.6. Zoom Icons

#### b. *Panning*

Sama halnya seperti *zooming*, ada beberapa pilihan untuk *panning* di jendela *editing*. Berikut ini adalah cara-cara untuk melakukan *panning* atau mengatur skala area gambar.

- Tekan tombol tengah mouse untuk masuk ke mode *track pan*. Mode ini menempatkan ISIS ke dalam sebuah mode yang mana seluruh lembar proyek akan dapat digerakkan mengikuti gerakan mouse. Kursor juga akan memberikan tanda bahwa kita telah memasuki mode ini. Untuk keluar dari mode track pan dengan menekan kembali tombol kiri pada mouse.
- Langkah mudah mengatur jendela pengeditan atas, bawah, kiri atau kanan, dapat dilakukan dengan memposisikan letak mouse pada bagian yang diinginkan kemudian tekan tombol F5.
- Tekan dan tahan tombol shift kemudian arahkan mouse agar menyentuh tepi jendela pengeditan untuk mengatur ke atas, bawah, kiri, atau kanan. Hal ini dikenal dengan *Shift Pan*.

- Apabila ingin memindahkan window *editing* ke bagian yang berbeda dari gambar, langkah tercepat adalah dengan menunjuk ke pusat area baru pada *overview window* dan klik kiri pada mouse.
- Dapat juga dengan menggunakan ikon *Pan* pada *toolbar*.

ISIS didesain agar lebih mudah dipakai (*user friendly*) dan menyediakan cara-cara untuk membantu mengetahui apa yang terjadi selama proses mendesain seperti komponen akan dikelilingi garis putus-putus ketika mouse mengarah di atasnya, atau kursor akan berubah sesuai dengan fungsinya. Gambar 1.7 menjelaskan tentang bentuk-bentuk kursor yang diikuti dengan fungsinya.

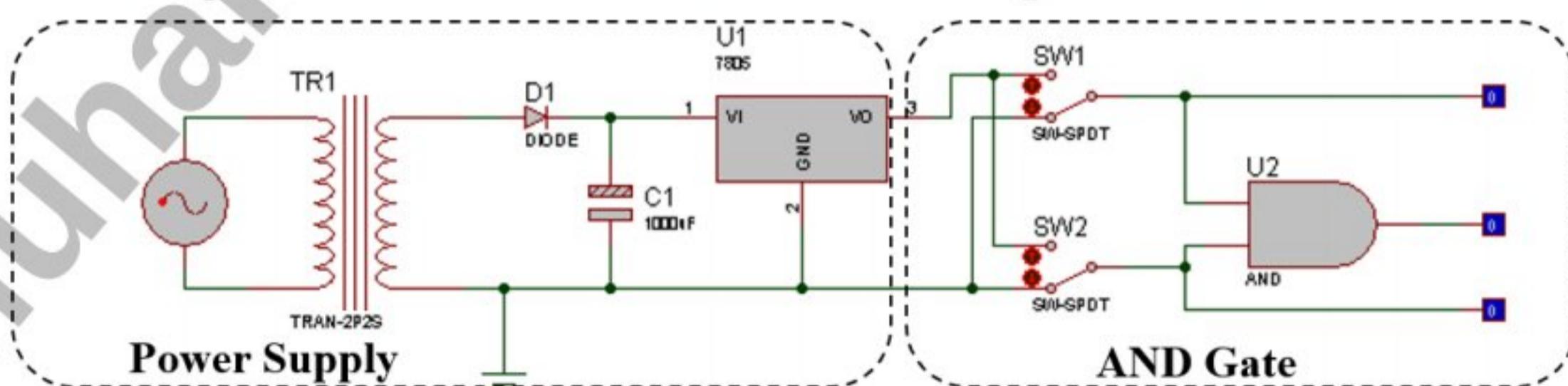
Cursor	Description
	Standard Cursor - Used in selection mode when not over a 'hot' object.
	Placement Cursor - Placement of an object will commence on a left click of the mouse.
	Hot Placement Cursor - Appears green when placement of a wire is available on lift click of the mouse.
	Bus Placement Cursor - Appears blue when placement of a BUS is available on lift click of the mouse.
	Selection Cursor - Object under the mouse will be selected on a left click of the mouse.
	Move Cursor - The currently selected object can be moved.
	Drag Cursor - The wire or 2D graphic can be dragged by holding the left mouse button down.
	Assignment Cursor - When over an object (having set the Property Assignment Tool) You can assign the property by left clicking the mouse button.

Gambar 1.7. Bentuk-bentuk cursor pada Proteus 8

## KEGIATAN PRAKTIKUM

### Percobaan 1. Latihan Membuat Rangkaian

1. Buat rangkaian di Gambar 1.8 dalam menerapkan Proteus.

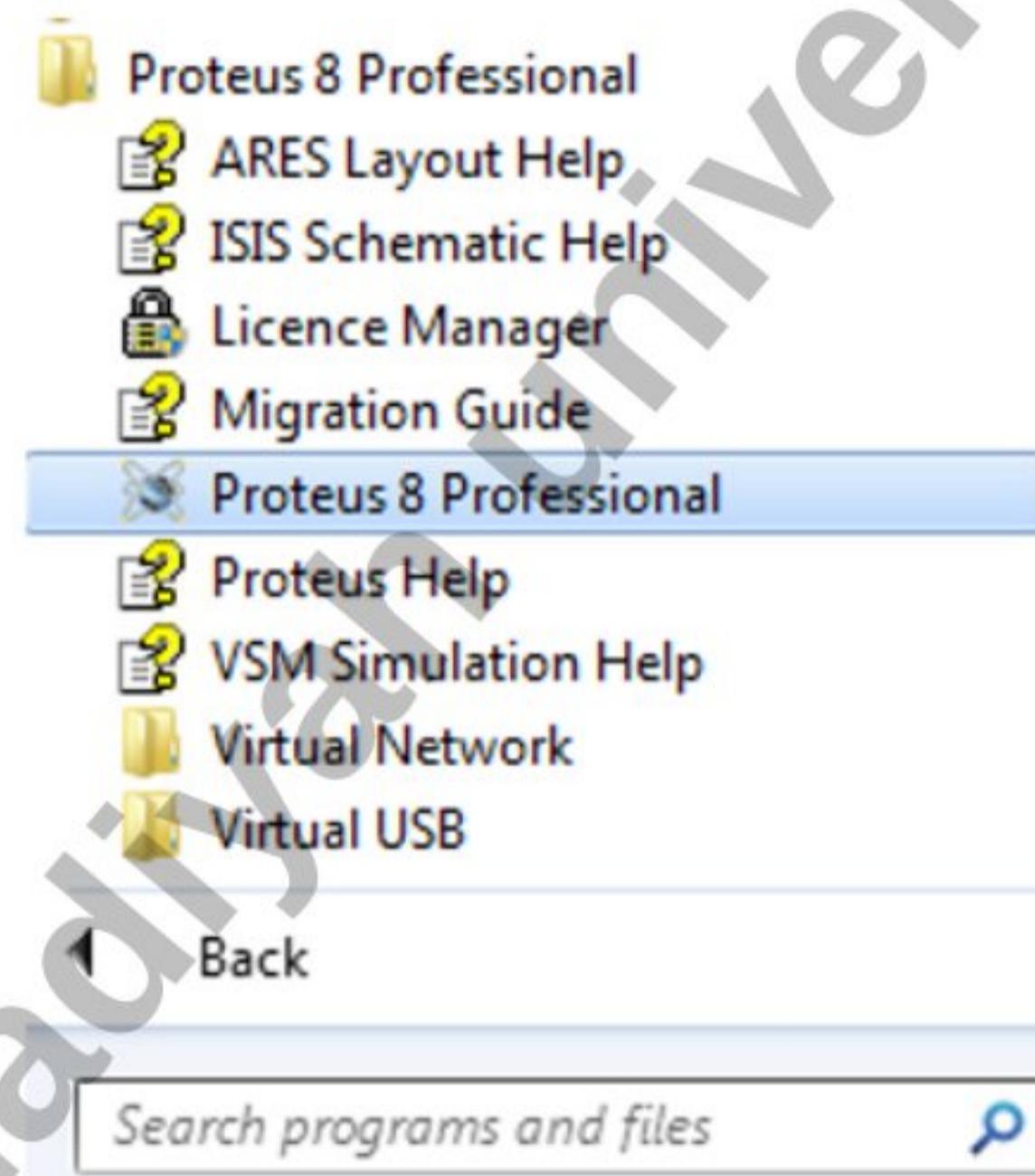


Gambar 1.8. Power Supply dasar dan rangkaian digital dasar

Tabel 1.1. Komponen yang diperlukan untuk membuat rangkaian

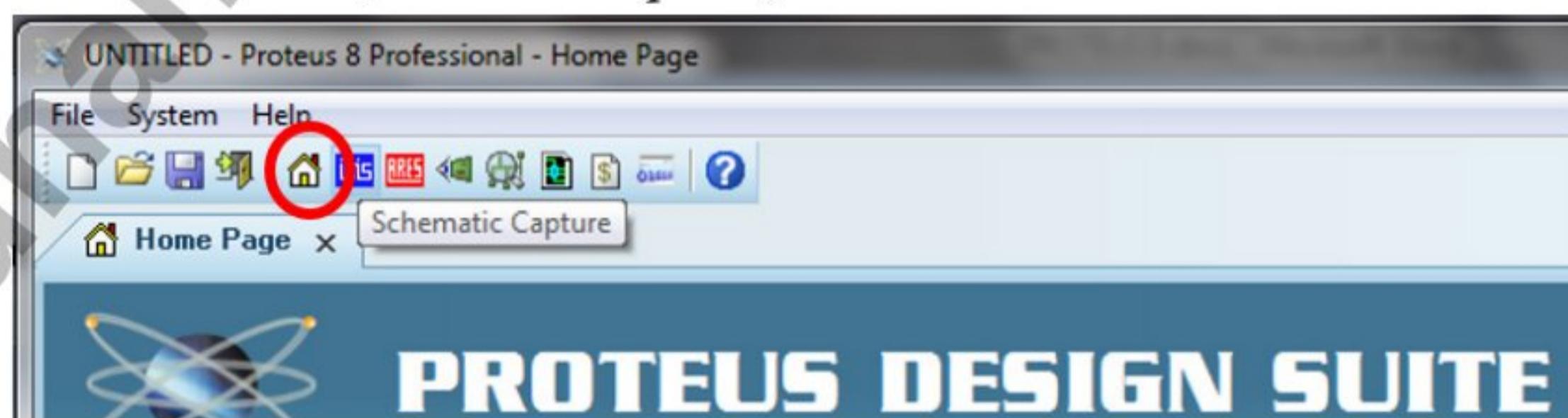
No	Device	Properties
1	ALTERNATOR	Voltage: 220V, Frequency: 50Hz
2	TRAN-2P2S	Coupling Factor: 0.05
3	DIODE	-
4	CAP-ELEC	Capacitance: 1000uf
5	7805	-
6	SW-SPDT	-
7	AND gate	-
8	Logic Probe	-

2. Login di komputer kemudian klik “All Program” → “Proteus 8 Professional” → “Proteus 8 Professional”



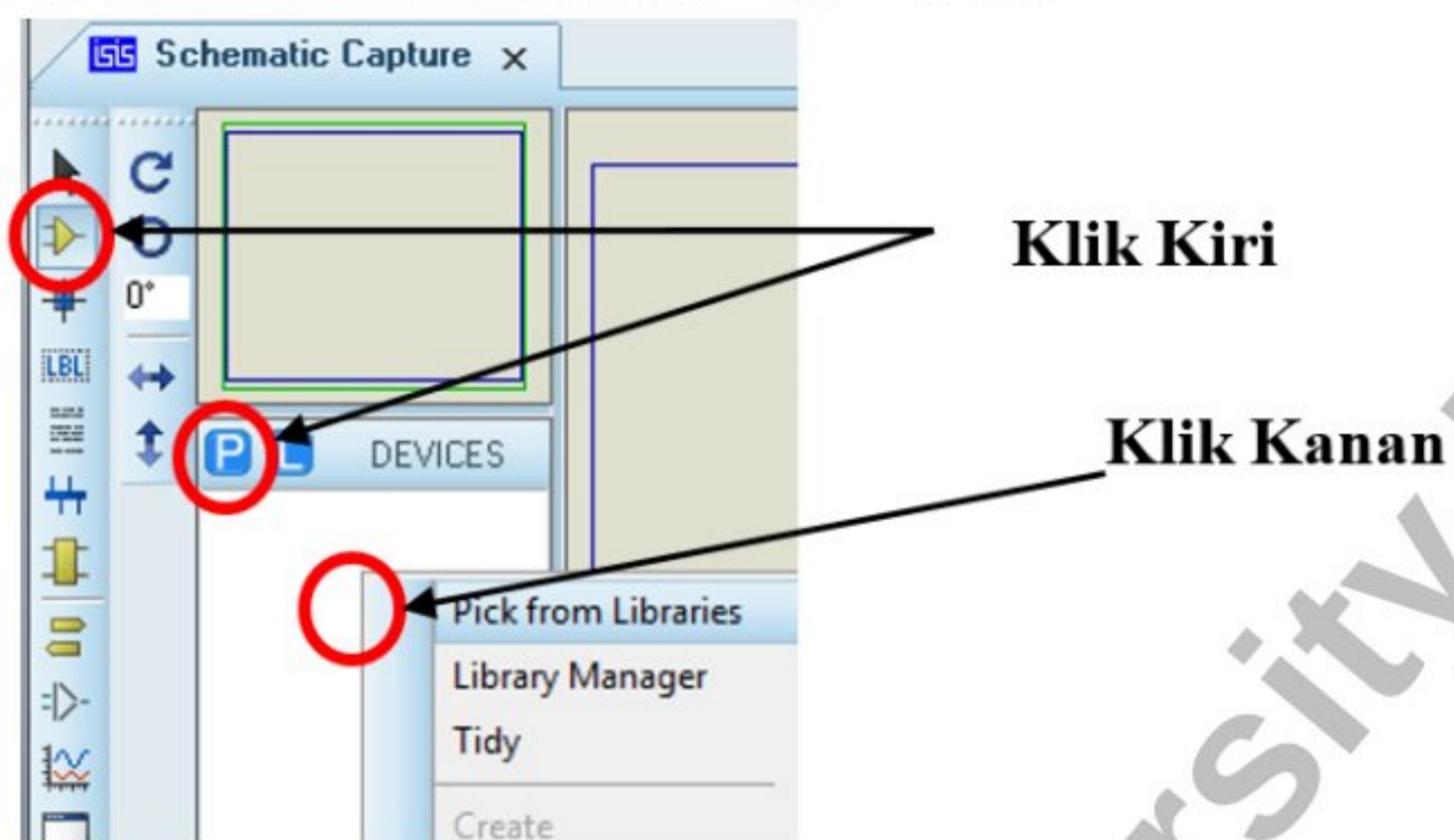
Gambar 1.9. Aplikasi proteus utama

3. Klik icon “ISIS (schematic capture)”



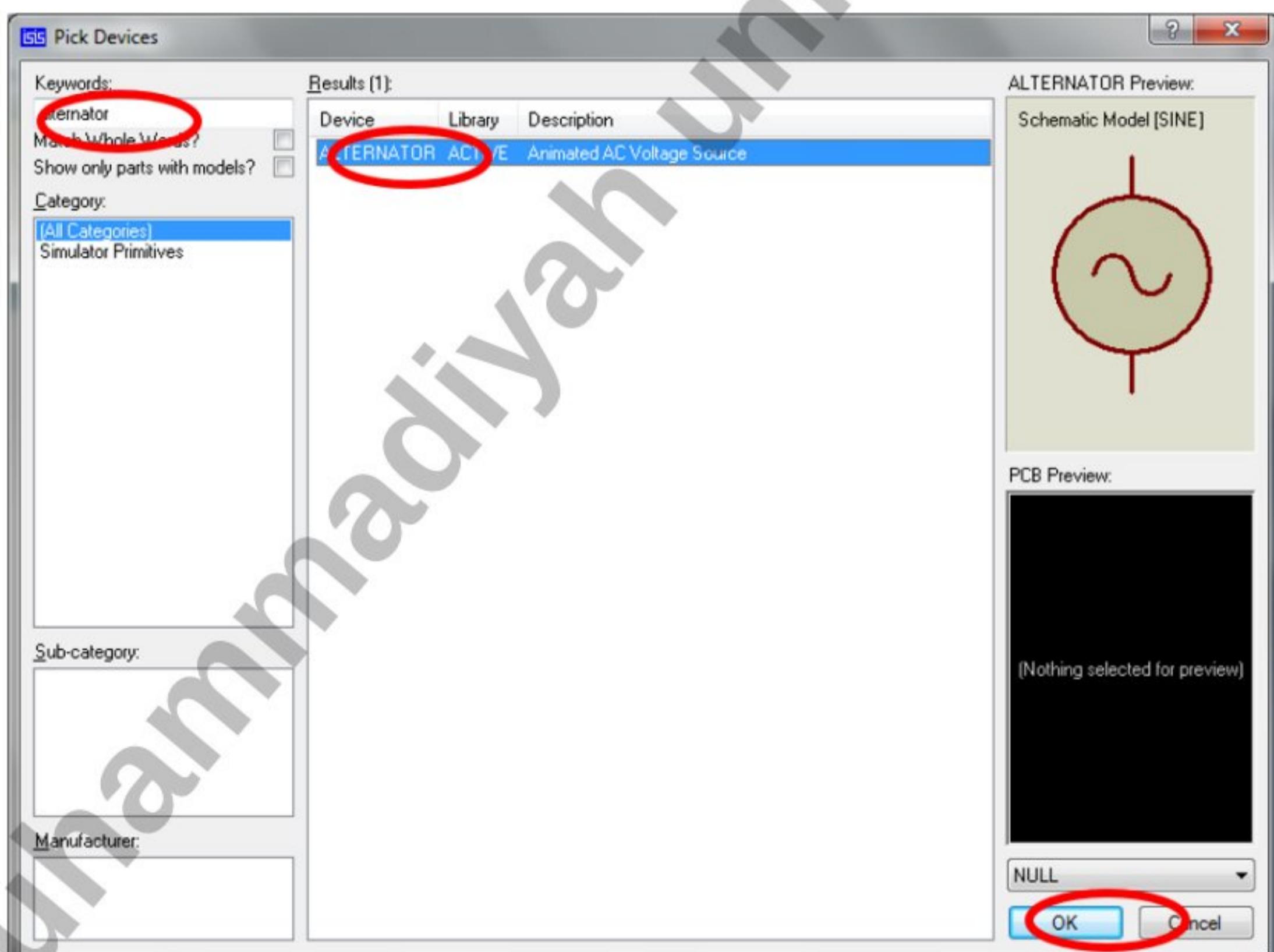
Gambar 1.10. Icon ISIS (schematic capture) pada window proteus

4. Setelah Window ISIS Schematic Capture muncul, klik “Komponen Mode” kemudian Klik kiri pada icon “P” di object selector. Atau dengan cara lain yaitu klik kanan di object selector → Pick from Libraries untuk membuka window *Pick Devices*



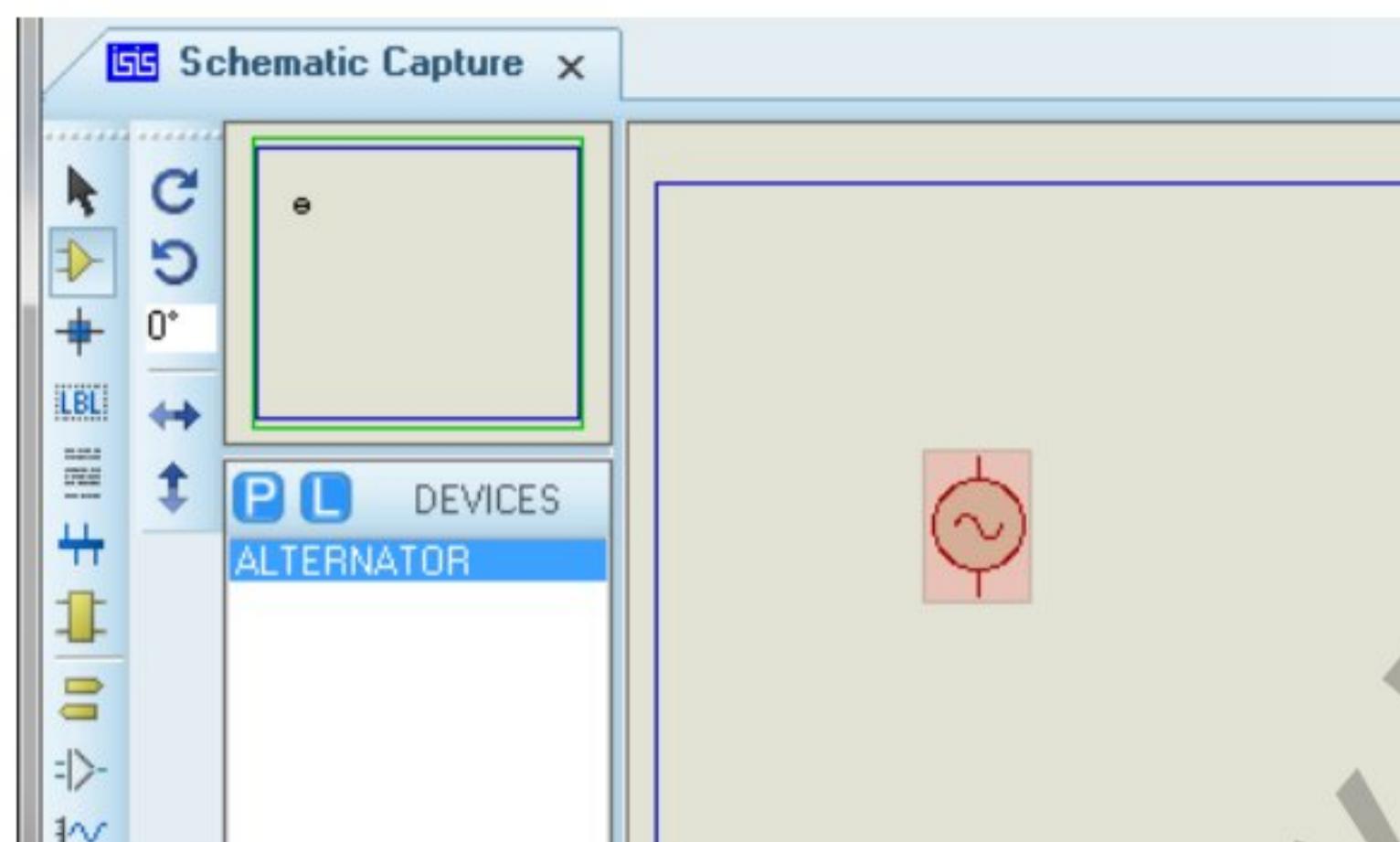
Gambar 1.11. Membuka pick device pada object selector

5. Ketik pada keyword “Alternator”, kemudian pilih *device* pada result, dan kemudian klik ”OK”



Gambar 1.12. Pick Device

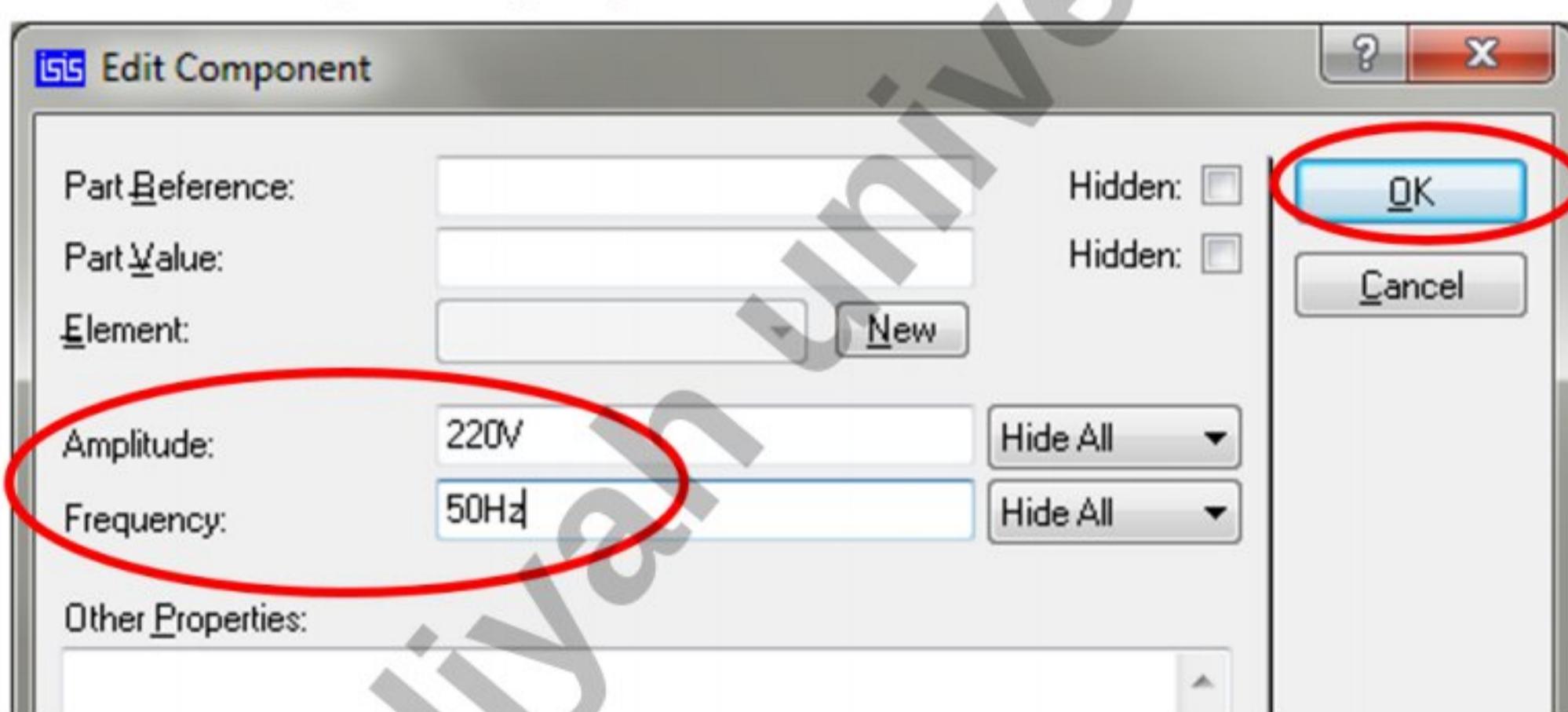
6. Letakkan *placement cursor* di *window editing* dan kemudian “klik kiri”



Gambar 1.13. Meletakan device alternator

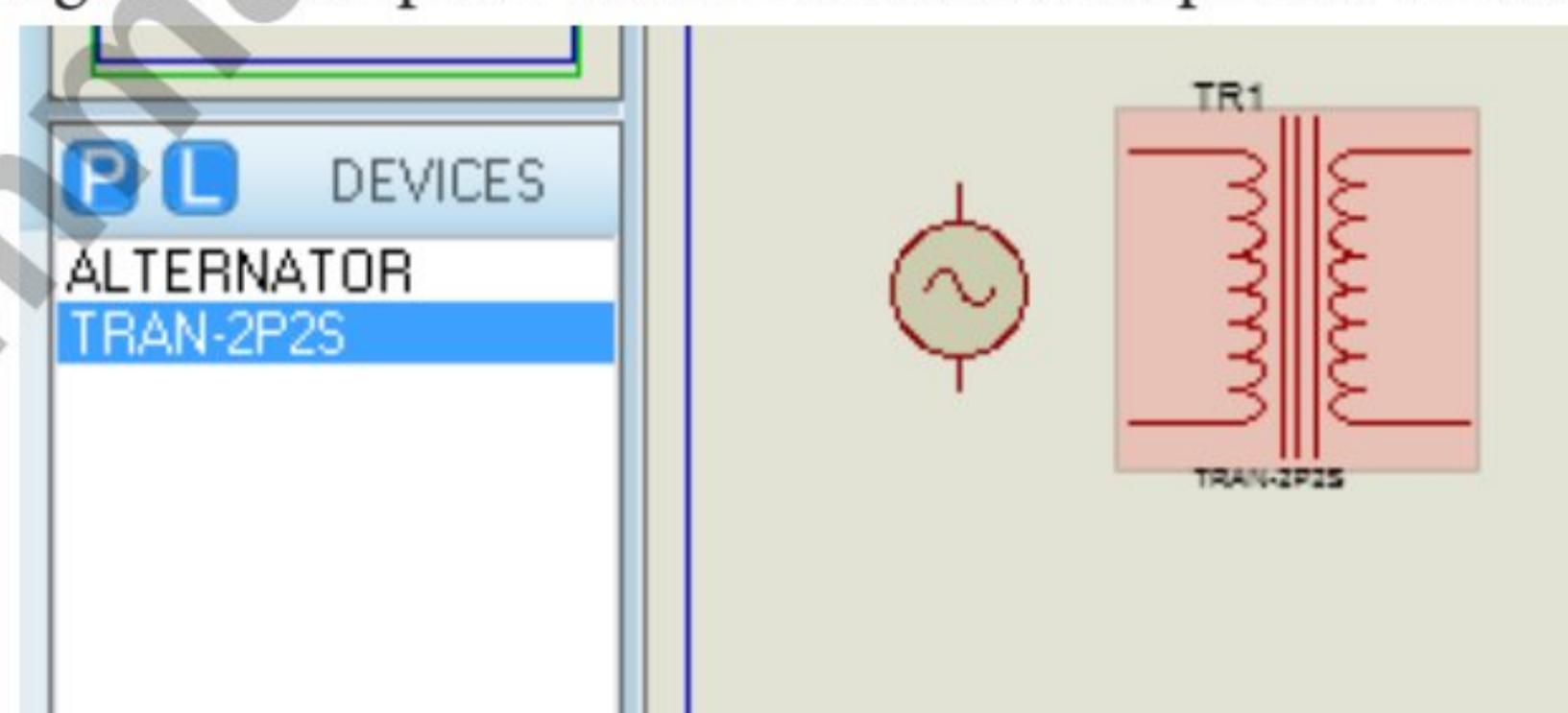
7. Klik kanan pada *device alternator* → edit properties → rubah nilainya (amplitude 220V, frequency 50Hz) dan kemudian klik “OK”.

Alternator ini seperti tegangan listrik 220V AC di sekitar kita.



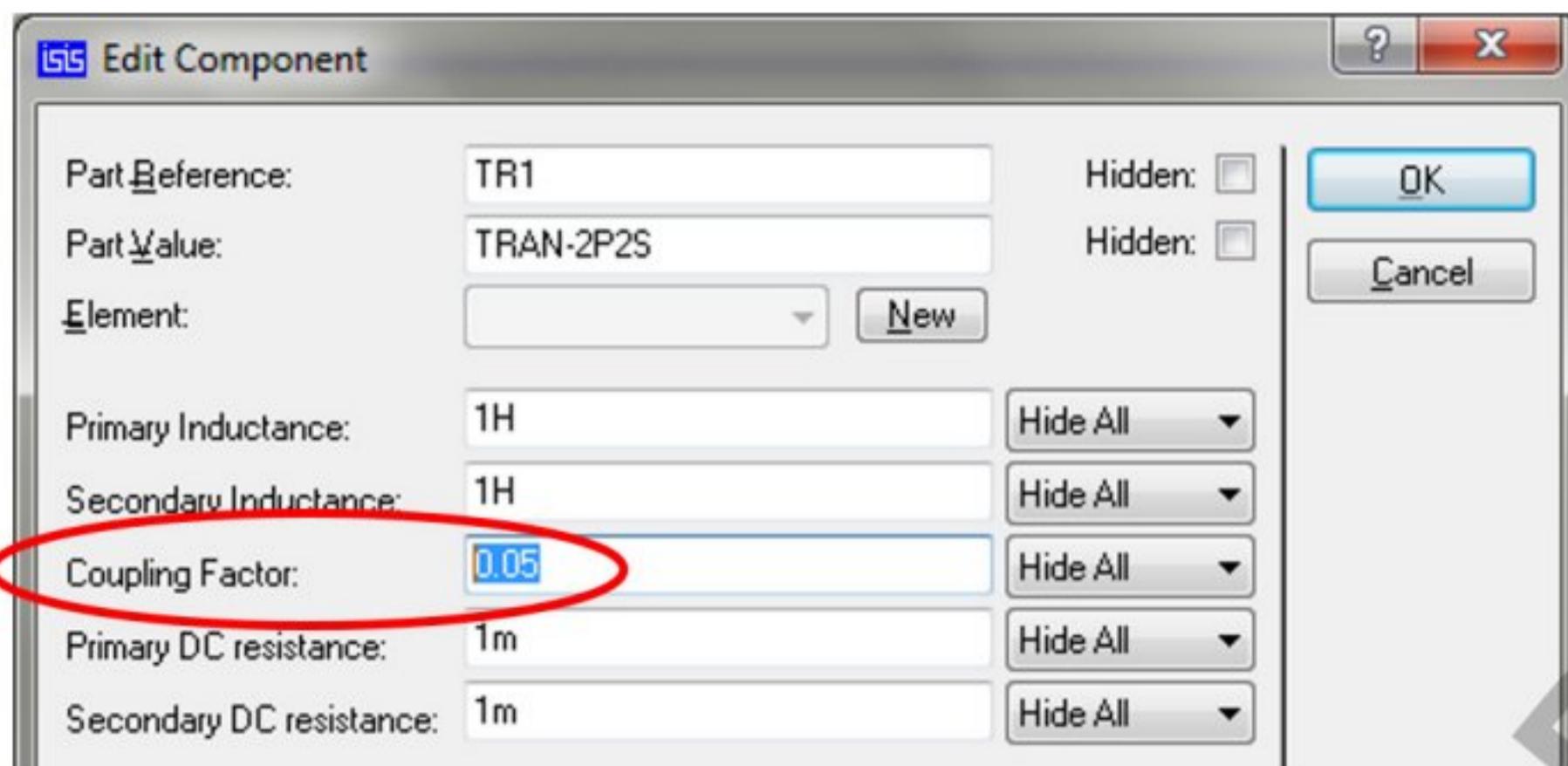
Gambar 1.14. Edit komponen

8. Ulangi langkah 4 sampai 6 untuk memilih komponen TRAN-2P2S



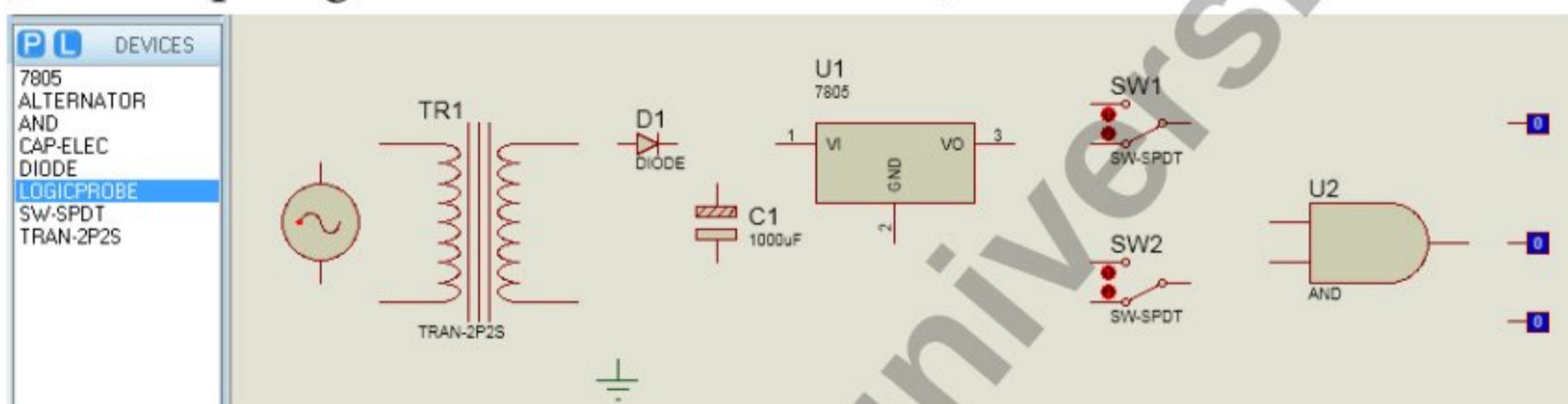
Gambar 1.15. Memilih Komponen Transformator Sederhana

9. Edit properties pada TRAN-2P2S (coupling factor 0.05 ). Perubahan ini untuk membuat trafo stepdown, yang dapat menurunkan tegangan AC



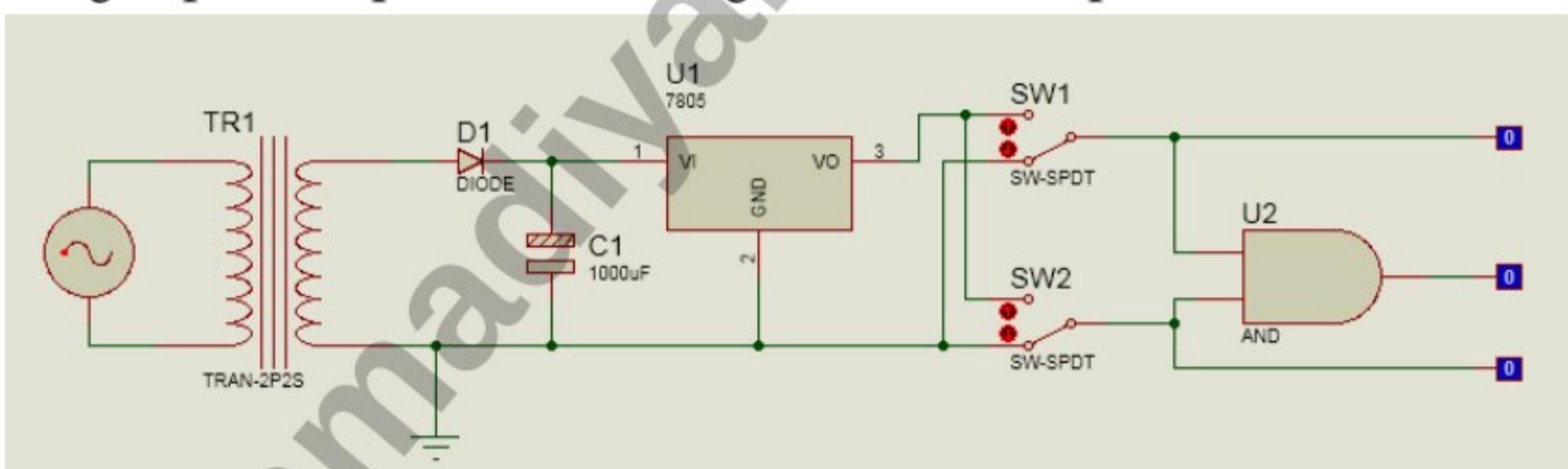
Gambar 1.16. Edit komponen

- Pilih device / komponen lain menggunakan langkah yang sama.  
(catatan: pilih ground di “Terminal mode”)



Gambar 1.17. Pilih komponen

- Hubungkan komponen bersama dengan klik kiri pin komponen dengan pin komponen lain sebagaimana dilihat pada Gambar 1.18.



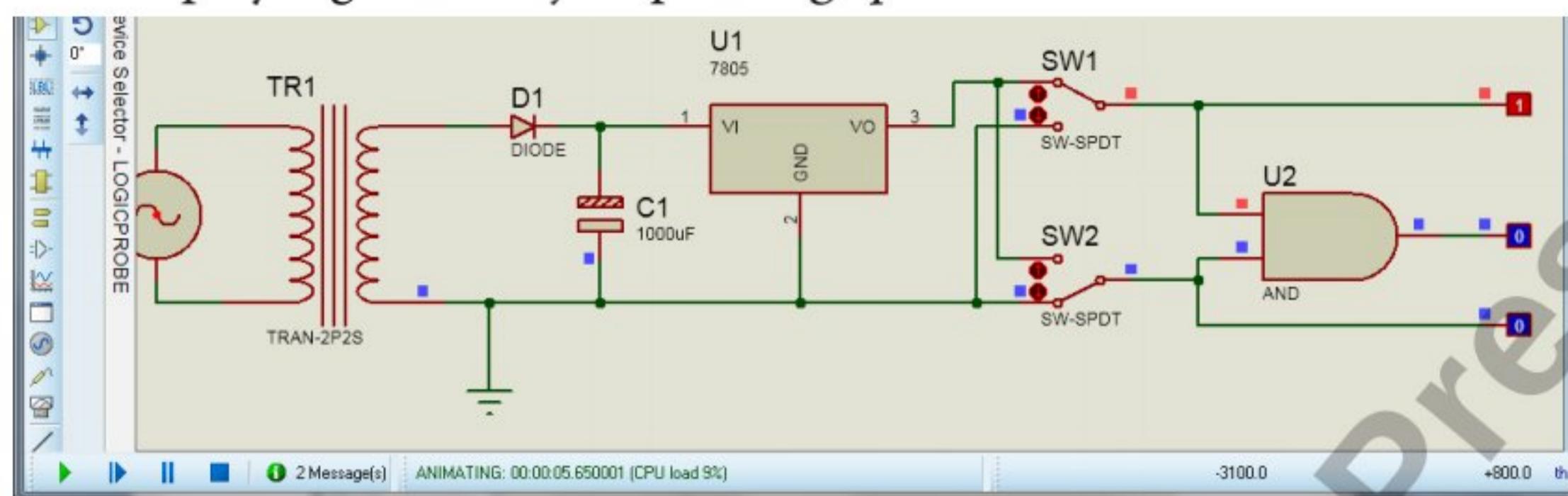
Gambar 1.18. Menyambung komponen

- Untuk melakukan simulasi klik “run the simulation”, untuk menghentikan pilih “stop button”.



Gambar 1.19. Tombol Simulasi

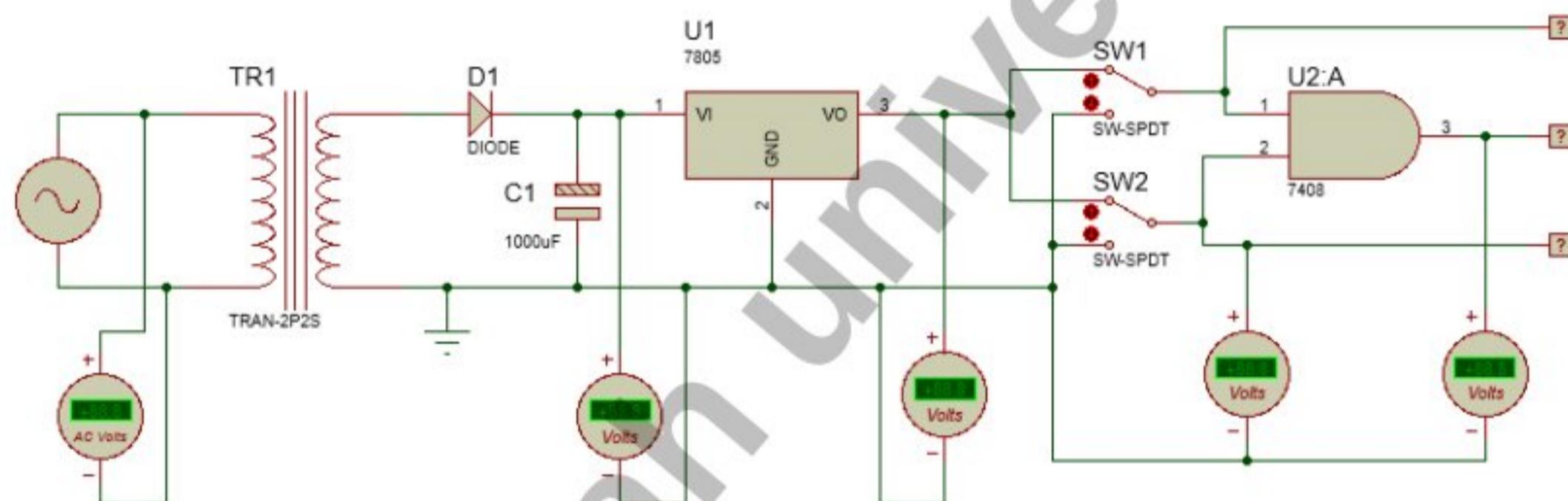
13. Simulasikan gerbang AND dengan melakukan klik pada SW1 dan SW2. Apa yang akan terjadi pada logicprobe?



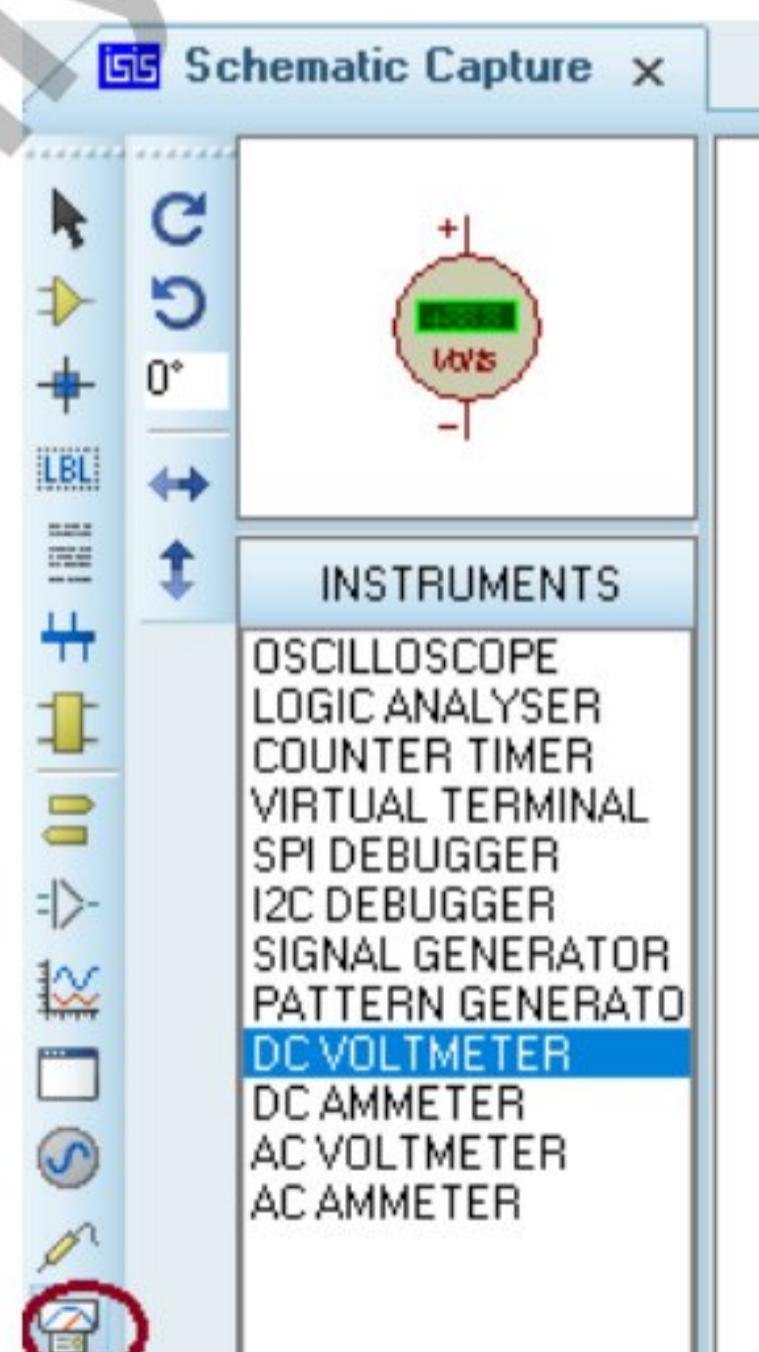
Gambar 1.20. Simulasi gerbang AND

## TUGAS

1. Buat rangkaian Gambar 1.21, dan simulasi dangan klik “run the simulation”!



Gambar 1.21. Rangkaian catu daya (power supply) terhubung dengan gerbang logika



Gambar 1.22. Instrument selector

**2. Catat pengukuran anda!**

Voltmeter AC : ..... Volt

Voltmeter DC 1 : ..... Volt

Voltmeter DC 2 : ..... Volt

Voltmeter DC 3 : ..... Volt

Voltmeter DC 4 : ..... Volt

**3. Jawab pertanyaan dibawah ini!**

- a. Apa perbedaan tegangan AC dan DC?

.....  
.....

- b. Bagaimana karakter tegangan pada masing-masing Voltmeter?

1) Tegangan di Voltmeter AC : ....(AC/DC).... Dan memiliki karakter : .....

2) Tegangan di Voltmeter DC 1 : ....(AC/DC).... Dan memiliki karakter: .....

3) Tegangan di Voltmeter DC 2 : ....(AC/DC).... Dan memiliki karakter : .....

4) Tegangan di Voltmeter DC 3 : ....(AC/DC).... Dan memiliki karakter : .....

5) Tegangan di Voltmeter DC 4 : ....(AC/DC).... Dan memiliki karakter : .....

**4. Buat kembali rangkaian Gambar 1.21, namun atur rangkaian sehingga mengasilkan nilai output dari catu daya 9 Volt DC!**

Nilai output gerbang logika tetap 5 Volt.

- a. Jelaskan fungsi dari transformator stepdown dalam rangkaian tersebut?

.....

- b. Jelaskan fungsi dari IC 78XX dalam rangkaian tersebut?

.....

# MODUL 2

## PENGENALAN SINYAL

### TUJUAN PRAKTIKUM

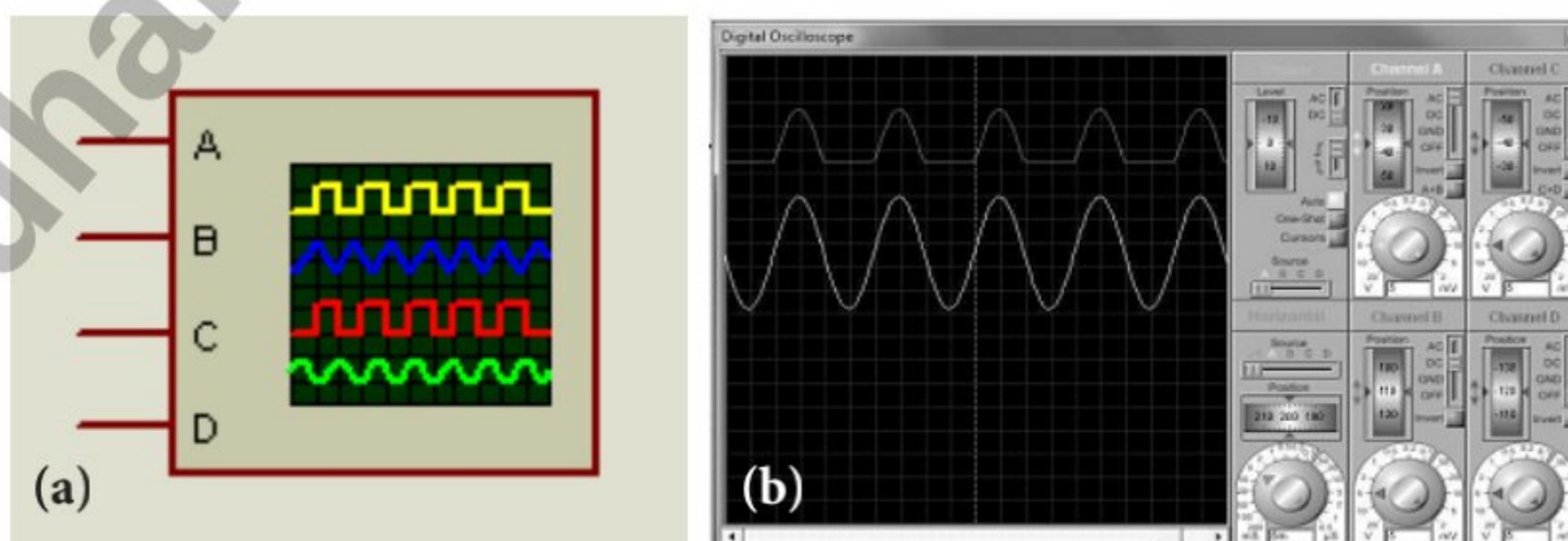
Mahasiswa dapat memahami tentang jenis-jenis pengukuran sinyal dan simulasinya.

### TEORI

Rangkaian elektronika tidak akan terlepas dari berbagai alat ukur elektronika, baik untuk mengukur besaran listrik maupun dalam bentuk sinyal listrik. Dua jenis alat ukur yaitu alat ukur analog dan alat ukur digital.

### Osiloskop

Osiloskop atau oscilloscope merupakan alat ukur elektronika yang digunakan untuk merepresentasikan bentuk sinyal listrik sehingga dapat dilihat dan dipelajari. Kondisi fisik dari osiloskop sendiri dilengkapi dengan tabung sinar katoda. Piranti pemancar elektron memproyeksikan sorotan elektro ke layar tabung sinar katoda sehingga membekas pada layar. Rangkaian khusus pada osiloskop menyebabkan sorotan tersebut bergerak berulang-ulang dari kiri ke kanan. Pengulangan inilah yang menyebabkan bentuk sinyal kontinyu sehingga dapat dipelajari. Gambar 2.1 (a) simbol osiloskop pada ISIS dan Gambar 2.1 (b) adalah tampilan osiloskop dalam program simulasi. Gambar 2.2 merupakan menu-menu yang ada dalam osiloskop. Table 2.1. merupakan penjelasan detail mengenai panel-panel osiloskop pada Proteus 8.



Gambar 2.1. Osiloskop. (a) symbol osiloskop pada ISIS. (b) simulasi program.

Panel-panel pada sebuah osiloskop mempunyai fungsi masing-masing. Berikut ini adalah karakteristik dari panel osiloskop (Manik, 2006):

**1. *Volt/div (atau volts/cm)***

Panel ini menunjukkan besarnya tegangan yang terlihat pada layar. Ada dua tombol konsentris, tombol yang ditempatkan pada kedudukan maksimum ke kanan (searah dengan jarum jam) menyatakan osiloskop dalam keadaan terkalibrasi untuk pengukuran. Kedudukan tombol di luar menyatakan besar tegangan pada layer (per cm) dalam arah vertical.

**2. *Time/div (atau Time/cm)***

Panel ini menyatakan faktor pengali waktu untuk mendapatkan periode atau frekuensi. Ada dua tombol konsentris, tombol di tengah pada kedudukan maksimum ke kanan (searah dengan jarum jam) menyatakan osiloskop dalam keadaan terkalibrasi untuk pengukuran. Kedudukan tombol di luar menyatakan faktor pengali untuk waktu dari gambar pada layar dalam arah horizontal.

**3. *Sinkronisasi***

Panel ini berfungsi untuk mengatur supaya pada layar diperoleh gambar yang tidak bergerak.

**4. *Slope***

Panel ini berfungsi untuk mengatur saat trigger dilakukan, yaitu pada waktu sinyal naik (+) atau pada waktu sinyal turun (-)

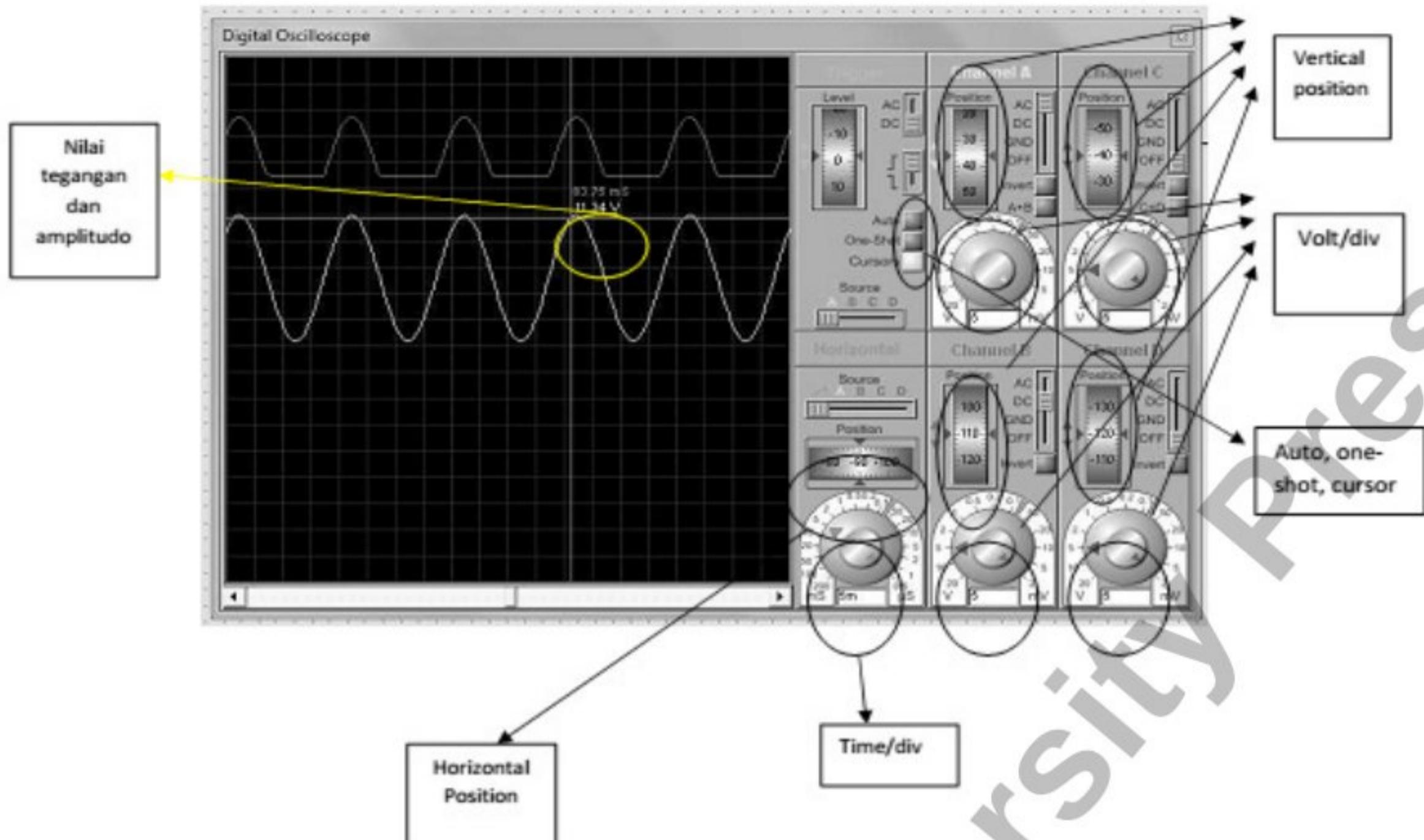
**5. *Kopling***

Panel ini menunjukkan hubungan dengan sinyal searah atau bolak-balik

**6. *Trigger ‘Ext’ atau ‘Int’ adalah***

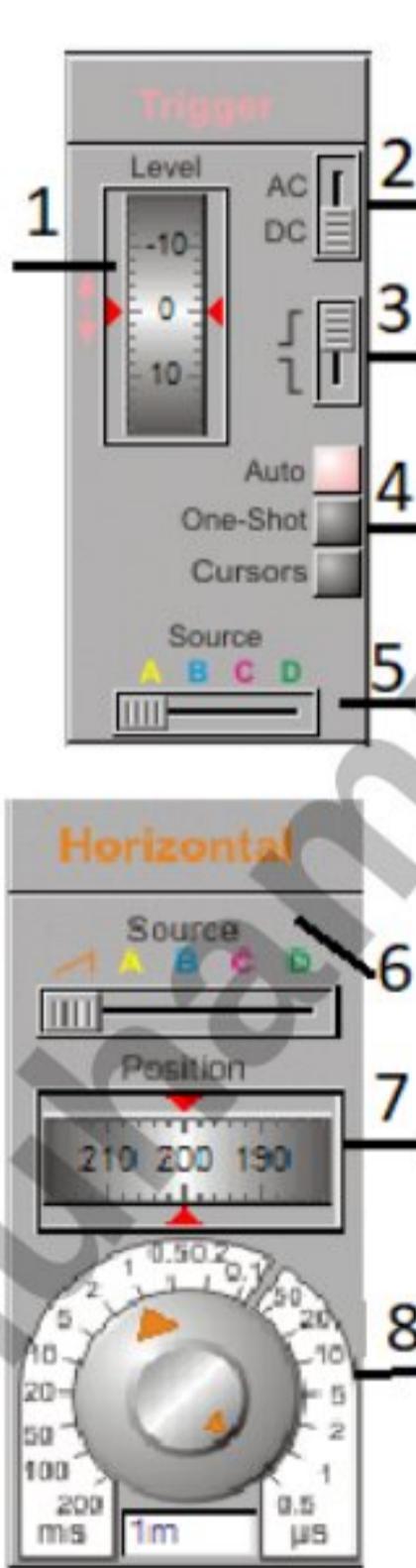
Untuk ‘Ext’ adalah trigger yang dikendalikan oleh rangkaian di luar osiloskop. Pada kedudukan ini fungsi tombol ‘sinkronisasi’, ‘slope’ dan ‘kopling’ tidak dapat dipergunakan.

Untuk ‘Int’ adalah trigger yang dikendalikan oleh rangkaian di dalam osiloskop. Pada kedudukan ini fungsi tombol ‘sinkronisasi’, ‘slope’ dan ‘kopling’ dapat dipergunakan.



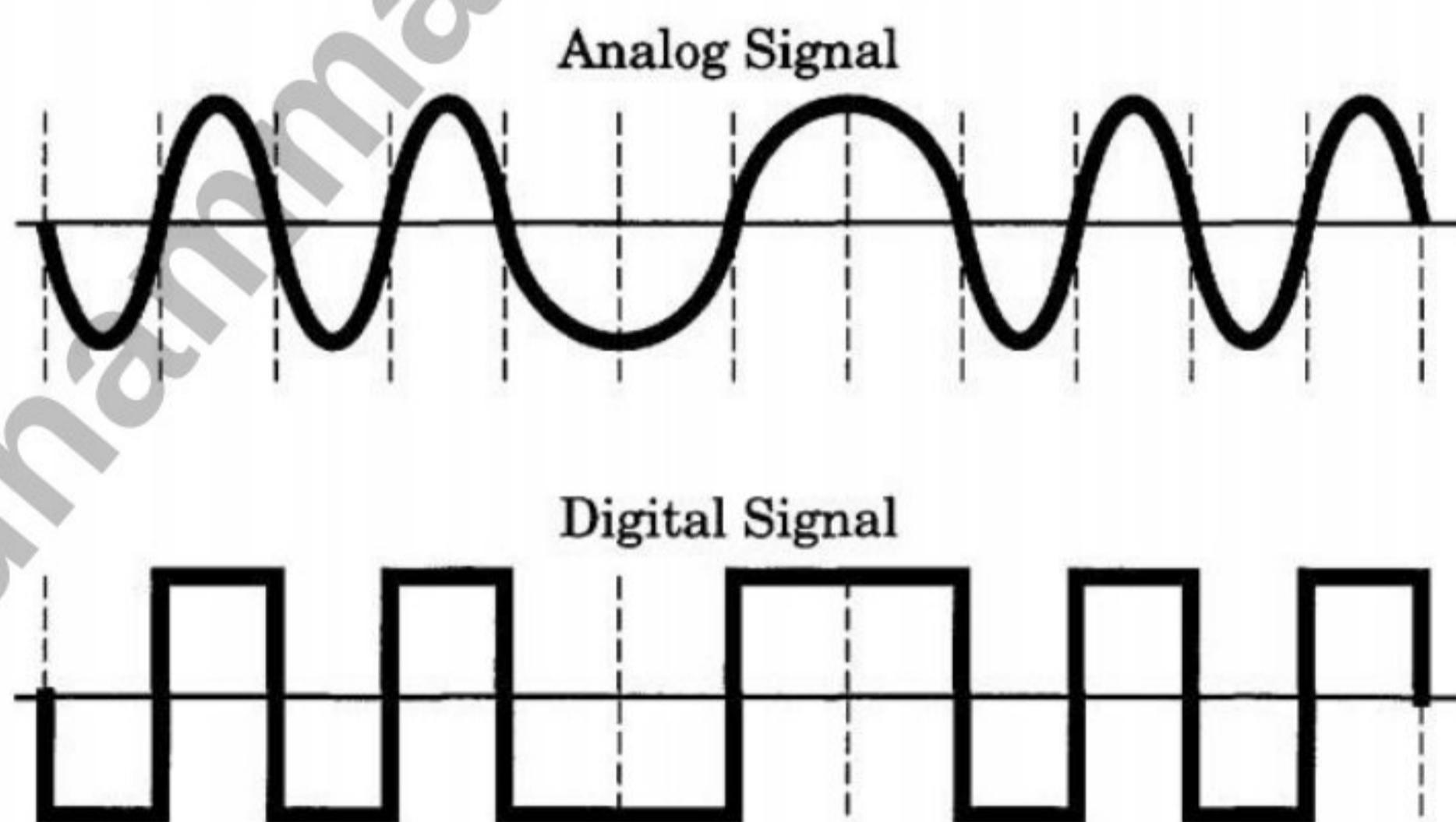
Gambar 2.2. Menu osiloskop pada Proteus 8

Table 2.1. panel-panel osiloskop pada Proteus 8.

Panel Osiloskop	Keterangan
	<ol style="list-style-type: none"> <li>1. Level, digunakan untuk mengatur posisi di sumbu-X</li> <li>2. AC-DC, digunakan untuk mengatur tegangan yang digunakan</li> <li>3. Untuk mengubah puncak menjadi lembah atau sebaliknya</li> <li>4. Mode operasi : <ul style="list-style-type: none"> <li>- Auto = hasil pengukuran otomatis,</li> <li>- One-shot = secara one-shot bentuk sinyal berada dalam posisi diam,</li> <li>- Cursor = nilai tegangan dan waktu dari sinyal input/output.</li> </ul> </li> <li>5. Source (pada trigger), digunakan untuk menentukan channel yang digunakan sebagai sinyal input/output</li> <li>6. Source (horizontal), digunakan untuk mengatur posisi pada arah horizontal untuk channel tertentu</li> <li>7. Position (horizontal), digunakan untuk mengatur posisi sinyal pada arah horizontal.</li> <li>8. Timebase, adalah merupakan panel Time/div yang digunakan untuk mengatur periode / time base yang besarnya mulai dari <math>0,5 \mu\text{s}/\text{div}</math> sampai <math>200 \text{ ms}/\text{div}</math>. <ul style="list-style-type: none"> <li>- Sebagai contoh dalam gambar disamping, terlihat ukuran Time/div sebesar 1 ms.</li> </ul> </li> </ol>

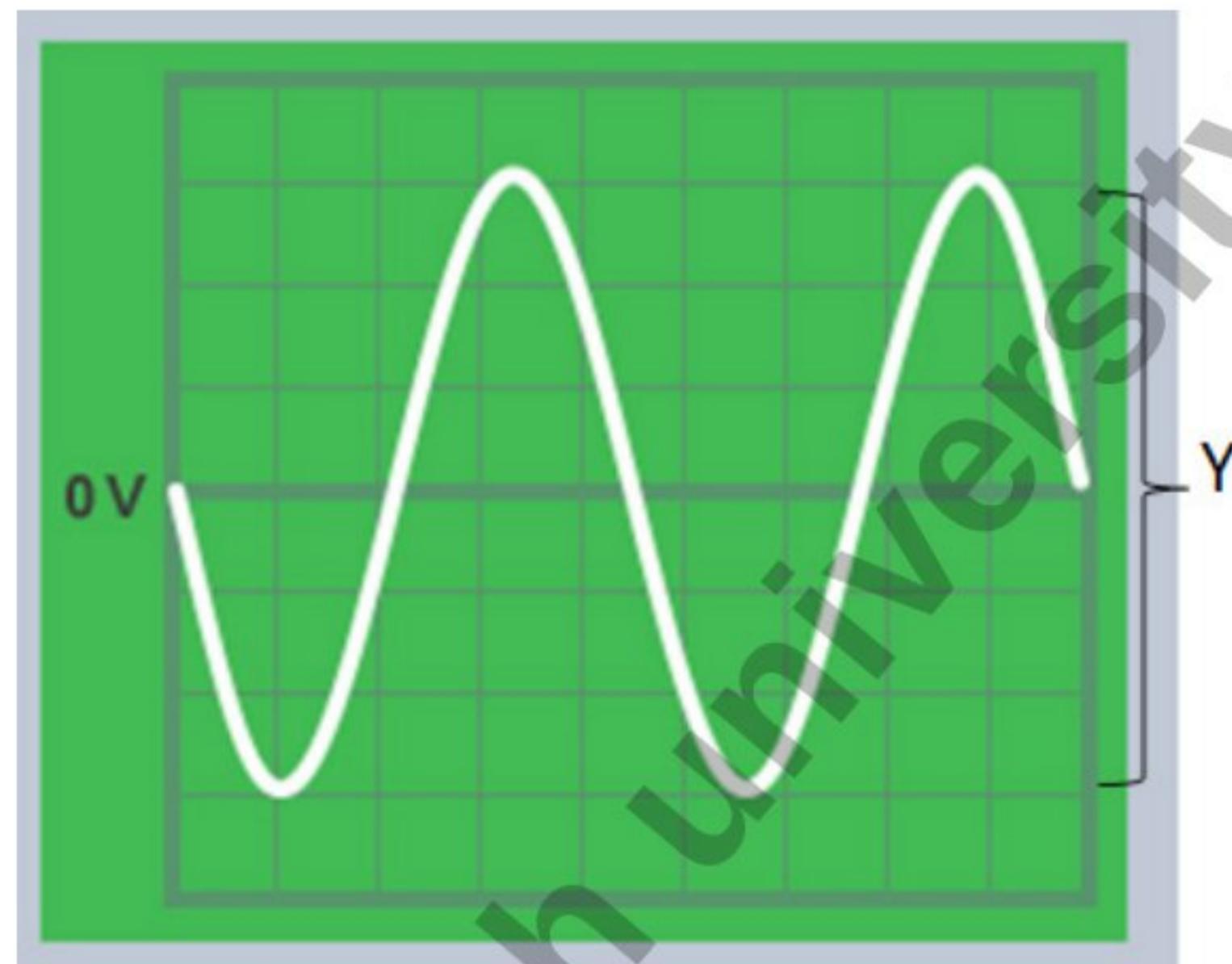
	<ol style="list-style-type: none"> <li>9. Position (horizontal), digunakan untuk mengatur posisi sinyal pada arah horizontal.</li> <li>10. AC/DC/GND/OFF, merupakan panel yang digunakan untuk menentukan fungsi dari channel yang dipakai. Apakah sinyal digunakan untuk mengukur sinyal AC, DC, dll</li> <li>11. Invert, merupakan panel untuk membalikkan sinyal</li> <li>12. A+B atau C+D, digunakan untuk menggabungkan sinyal dari channel A atau C dengan channel B atau D sehingga menghasilkan superposisi gelombang</li> <li>13. Gain channel, digunakan untuk mengatur penguatan yang nilainya mulai dari 2 mV/div sampai 20 V/div</li> </ol>
--	--

Sinyal terbagi menjadi dua jenis yaitu sinyal analog dan sinyal digital. Sinyal analog adalah sinyal yang berupa gelombang kontinyu mempunyai amplitude, frekuensi, dan phase. Amplitudo adalah ukuran tinggi rendahnya tegangan dari sinyal analog. Frekuensi merupakan jumlah gelombang sinyal analog dalam satuan detik. Dan phase adalah besarnya sudut sinyal analog pada waktu tertentu. Gelombang sinyal analog mengalami perubahan secara bertahap setiap ada perubahan. Misalnya satu gelombang penuh diawali dari voltase nol menuju voltase tertinggi dan turun hingga voltase terendah kemudian kembali ke voltase nol. Sedangkan sinyal digital adalah sinyal dengan bentuk pulsa yang mengalami perubahan tiba-tiba dengan besaran sinyal 0 dan 1. Dua keadaan ini, yaitu 0 dan 1 menyebabkan sinyal digital tidak mudah terpengaruh terhadap noise, namun transmisi sinyal digital untuk pengiriman jarak dekat. Gambar 2.3 merupakan tampilan sinyal analog dan sinyal digital.

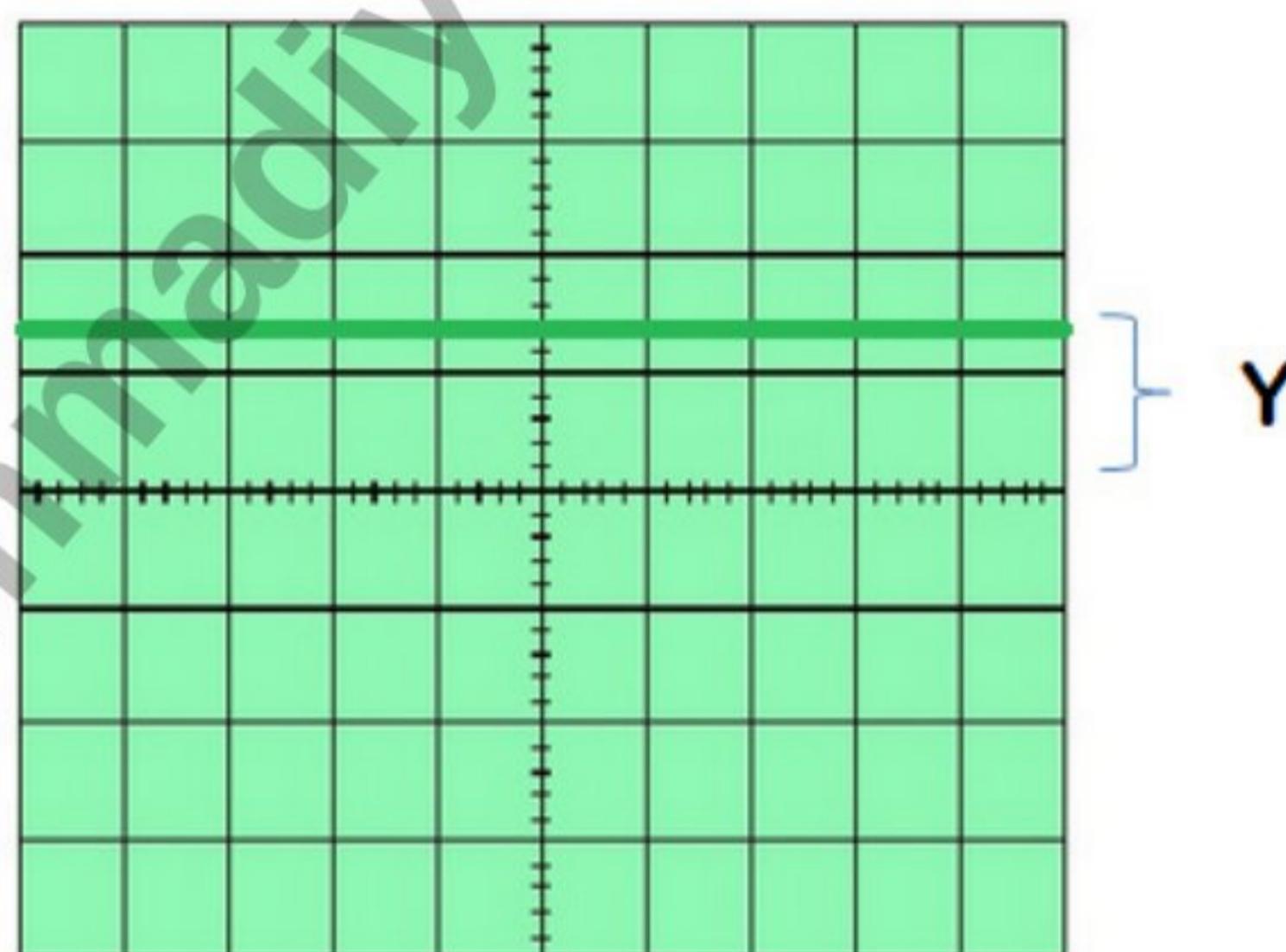


Gambar 2.3. Tampilan output sinyal analog dan digital

Kedua sinyal tersebut mempunyai keunggulan masing-masing. Kelebihan sinyal analog diantaranya adalah (1) sangat baik untuk komunikasi data dengan pengiriman jarak jauh. (2) antenna penerima RF mengambil sinyal nirkabel hanya berupa sinyal analog. (3) sinyal analog merupakan sinyal alamiah, seperti sinyal suara, dan seismograph. Kelebihan sinyal digital diantaranya adalah (1) pemakaian ruangan yang lebih kecil dan konsumsi daya yang lebih rendah. (2) memberikan keandalan (reability) yang lebih baik dan fleksibilitas. (3) memungkinkan penambahan layanan baru dan semakin berkembang.



Gambar 2.4. Contoh gelombang sinus dari sumber AC



Gambar 2.5. Sinyal output dari sumber DC

Setelah rangkaian selesai dibuat, maka langkah selanjutnya adalah pengamatan berdasarkan hasil simulasi rangkaian tersebut. Gambar 2.3. merupakan contoh sinyal sinus atau sinyal keluaran dari sumber AC

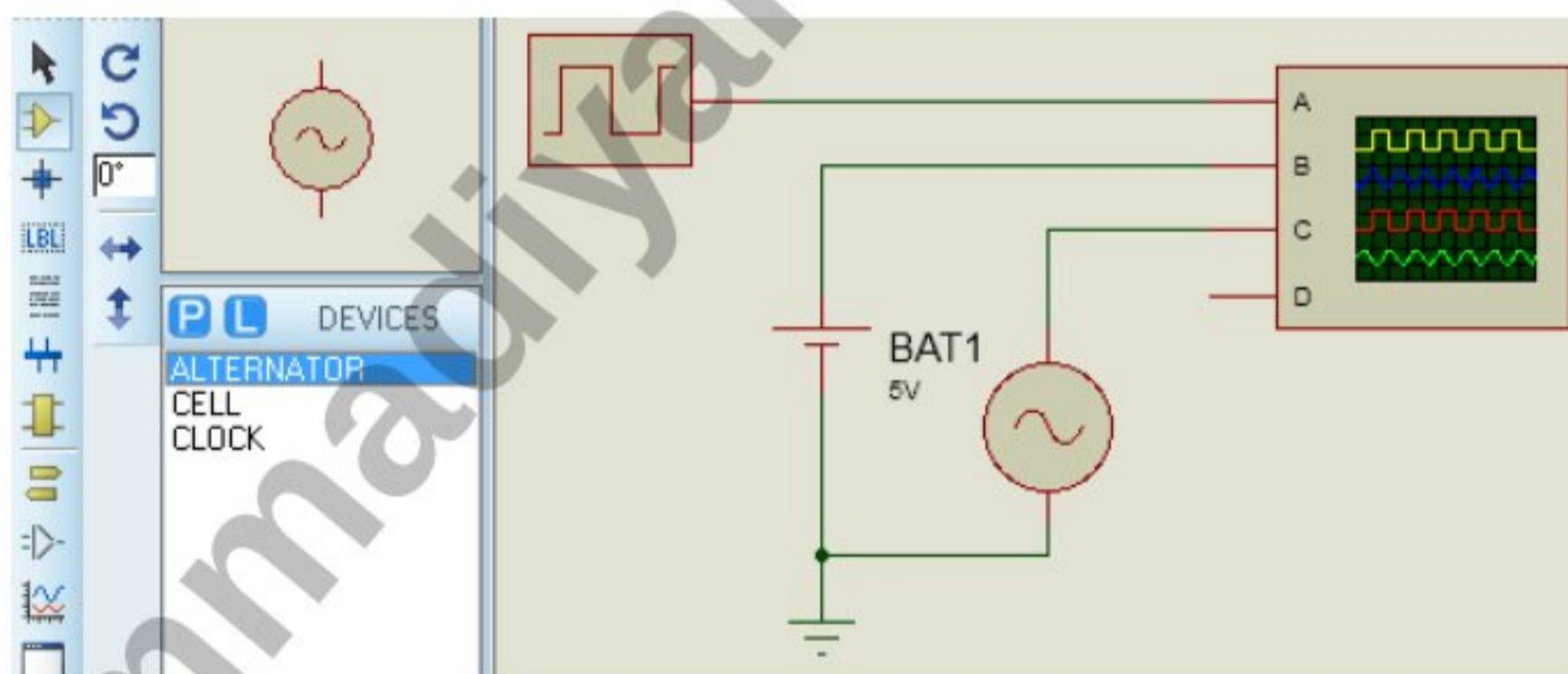
(Alternating Current). Berdasarkan Gambar 2.4, diketahui ukuran Volts/div pada osiloskop adalah 2V/div dan Time/div adalah 4,5 ms. Tegangan sebuah sinyal didapatkan dari perkalian antara Volts/div dengan jarak puncak-lembah sinyal. Sedangkan periode sebuah sinyal didapatkan dari perkalian antara Time/div dengan Panjang gelombang (/lamda). Sehingga dari Gambar 2.4, tegangan sinyal sebesar dan periodenya sebesar . Sedangkan frekuensi dari gelombang adalah berbanding terbalik dengan periode, sehingga frekuensi sinyal pada Gambar 2.3 adalah .

Gambar 2.5 merupakan contoh sinyal keluaran dari sumber DC (Direct Alternating) stabil dari sebuah baterai. Berdasarkan Gambar 2.5, misalnya nilai Volts/div pada osiloskop menunjukkan 5V/div. Tegangan sinyal didapatkan dari perkalian Volts/div dengan jarak puncak-lembah sinyal. Sehingga tegangan dari sinyal tersebut sebesar . Nilai periode dan frekuensi dari sinyal tersebut dapat dilakukan dengan cara yang sama seperti uraian sumber tegangan AC. Volts/div dan Time/div mempunyai peranan penting dalam mengatur keluaran sinyal dari sebuah rangkaian elektronik.

## KEGIATAN PRAKTIKUM

### Percobaan 1. Latihan Jenis-jenis Sinyal

- Buat rangkaian pada Gambar 2.3 dan edit properties masing-masing komponen sebagaimana terlihat pada Tabel 1.

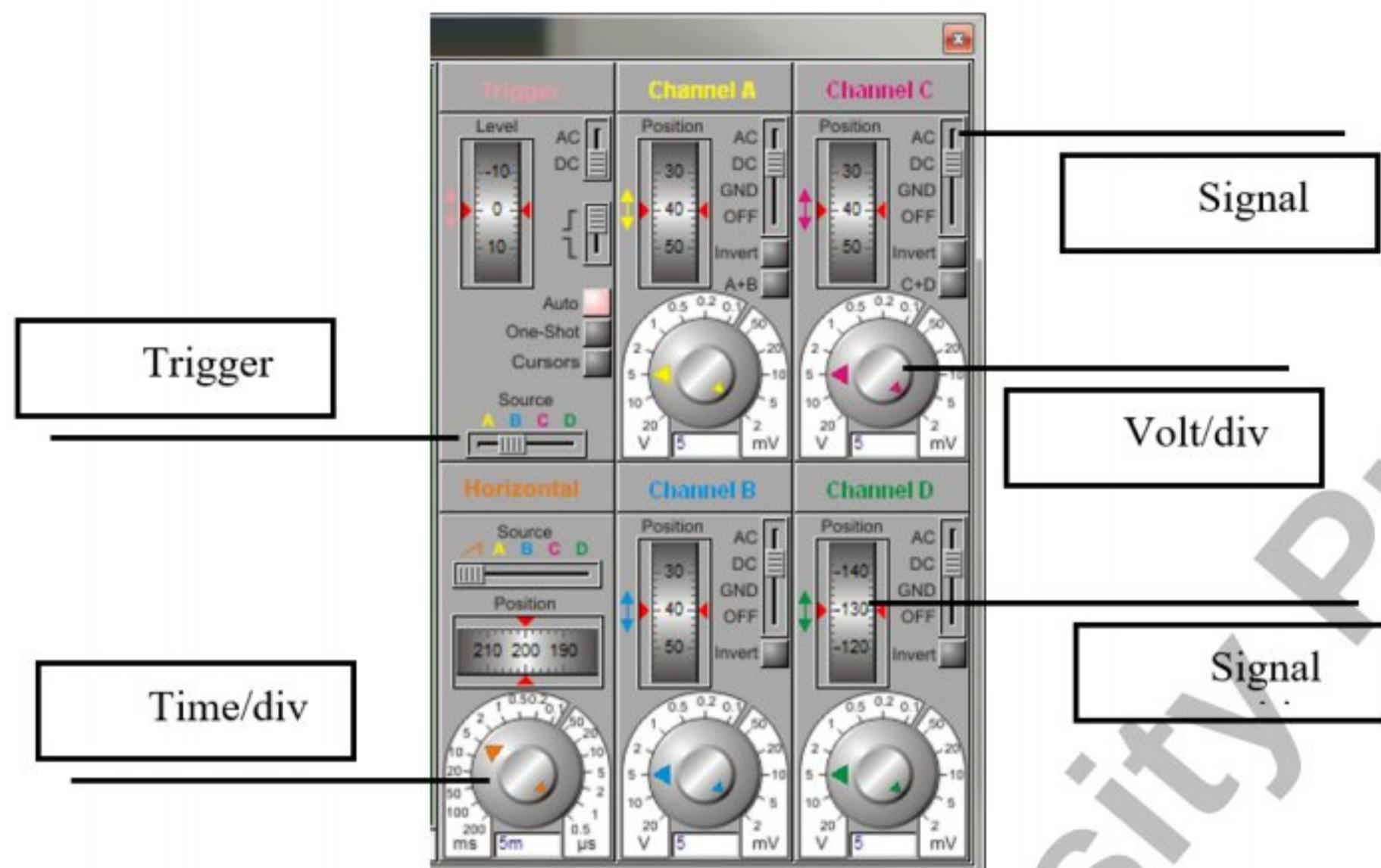


Gambar 2.3. Latihan macam sinyal

Tabel 1. Komponen pada rangkaian

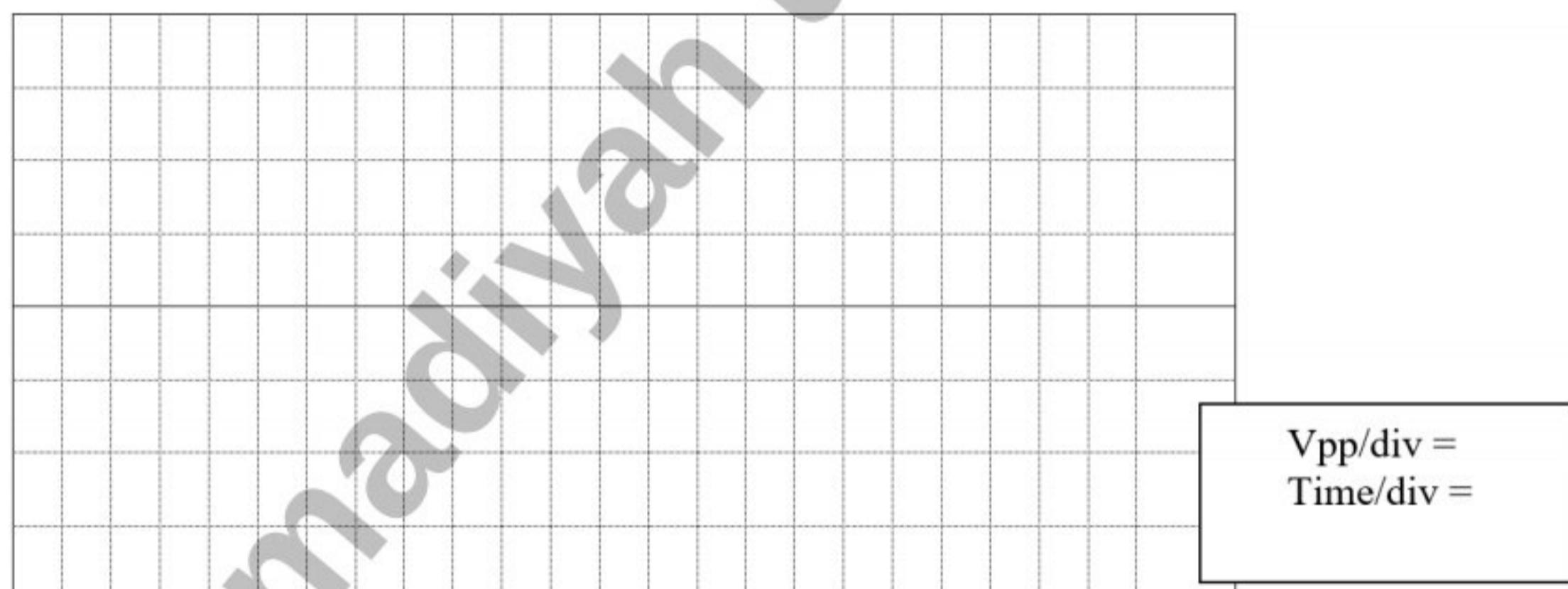
No	Device	Information
1	Alternator	V= 5 Volt, f= 100Hz
2	Cell	V= 5 Volt
3	Clock	F= 100Hz
4	Ground	Pick from Terminals
5	Osiloskop	Pick from Instrument

2. Simulasikan! Kemudian akan muncul osiloskop window



Gambar 2.4. Memahami Osiloskop

3. Coba pahami tentang *trigger source*, *signal type*, *volt/div*, *signal position* dan *time/div* dengan mengatur tiap *switch*. Kemudian atur switch sebagaimana di Gambar 2.4.
4. Simulasi akan menujukan pada kita garis sinyal dari Baterai, *Clock* dan Alternator. Gambarlah hasil simulasi anda simulation!



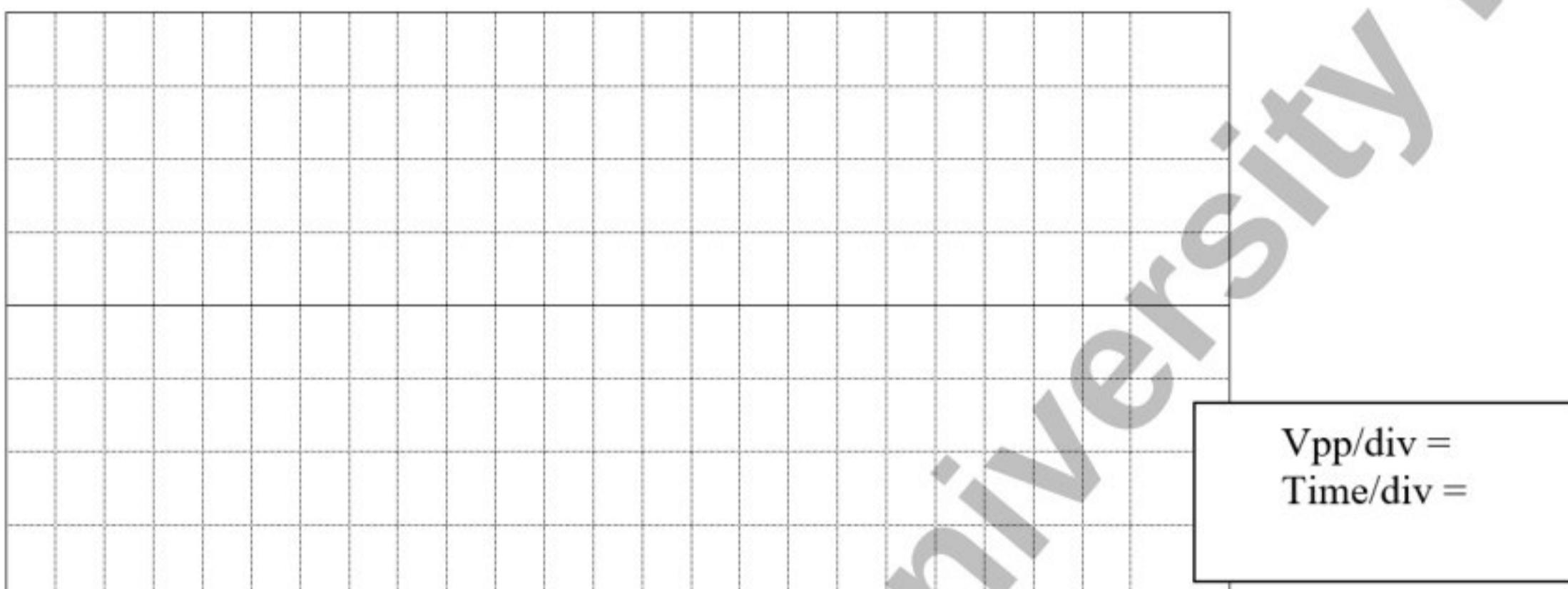
Dan berikan penjelasan!

5. Matikan simulasi! Kemudian edit komponen anda sebagaimana pada Tabel 2 berikut.

Tabel 2. Properties komponen

No	Device	Information
1	Alternator	$V = 10 \text{ Volt}, f = 50\text{Hz}$
2	Cell	$V = 7 \text{ Volt}$
3	Clock	$F = 200\text{Hz}$

6. Jalankan simulasi! Gambarlah hasil simulasi anda simulation!



Dan berikan penjelasan!

7. Jawab pertanyaan-pertanyaan ini!

- a. Jawab dengan singkat, perbedaan antara sinyal analog dan digital?

.....  
.....  
.....

- b. Bagaimana karakter sinyal pada masing-masing komponen?

Sinyal dari Alternator : ... (Analog / digital) ... Karena .....

.....  
.....

Sinyal dari Batery : ... (Analog / digital) ... Karena .....

.....  
.....

Sinyal dari Clock source : ... (Analog / digital) ... Karena .....

8. Buat kesimpulan berdasarkan pengamatan anda pada percobaan macam-macam sinyal.

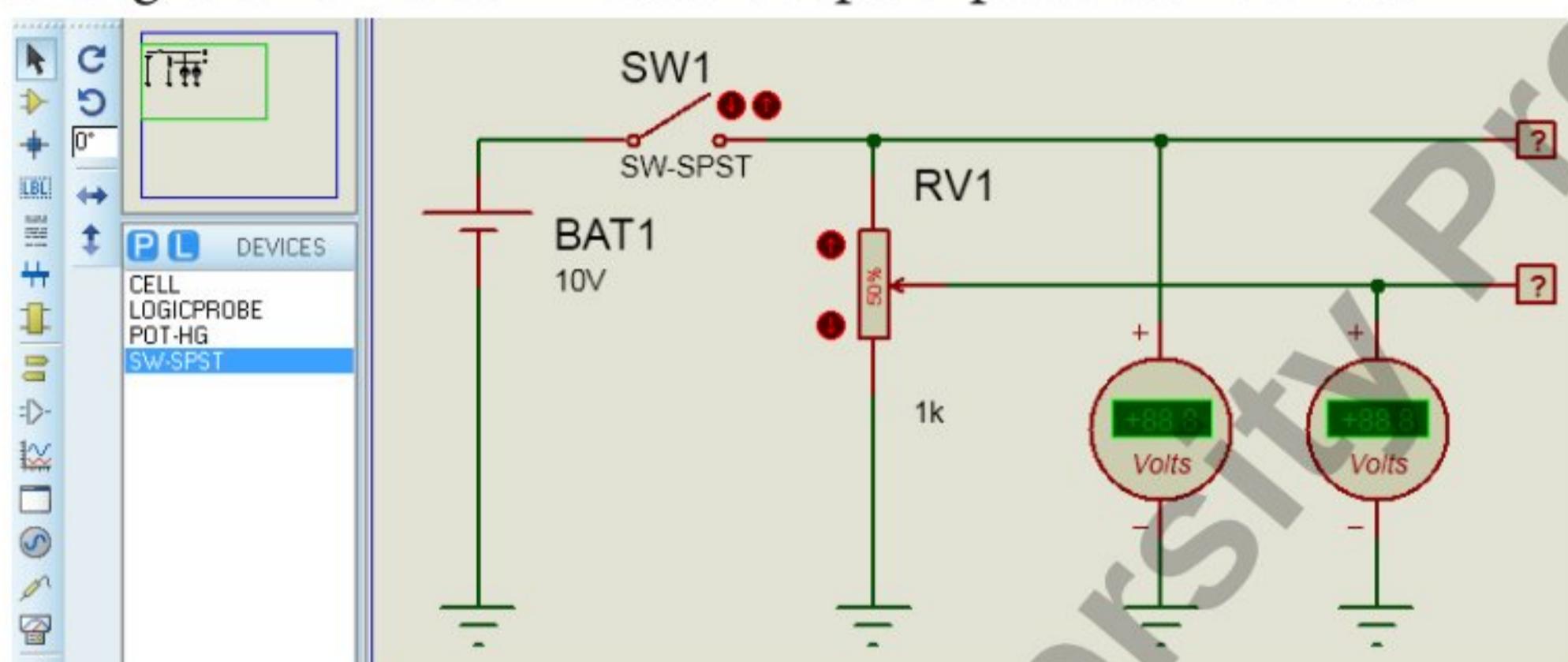
.....  
.....

Catatan :

- Parameter Digital mempunyai nilai diskrit yang tetap
- Parameter Analog mempunyai range nilai yang kontinyu

### Percobaan 2. Latihan Range Sinyal Digital

- Buat rangkaian simulasi Proteus 8 seperti pada Gambar 2.5.

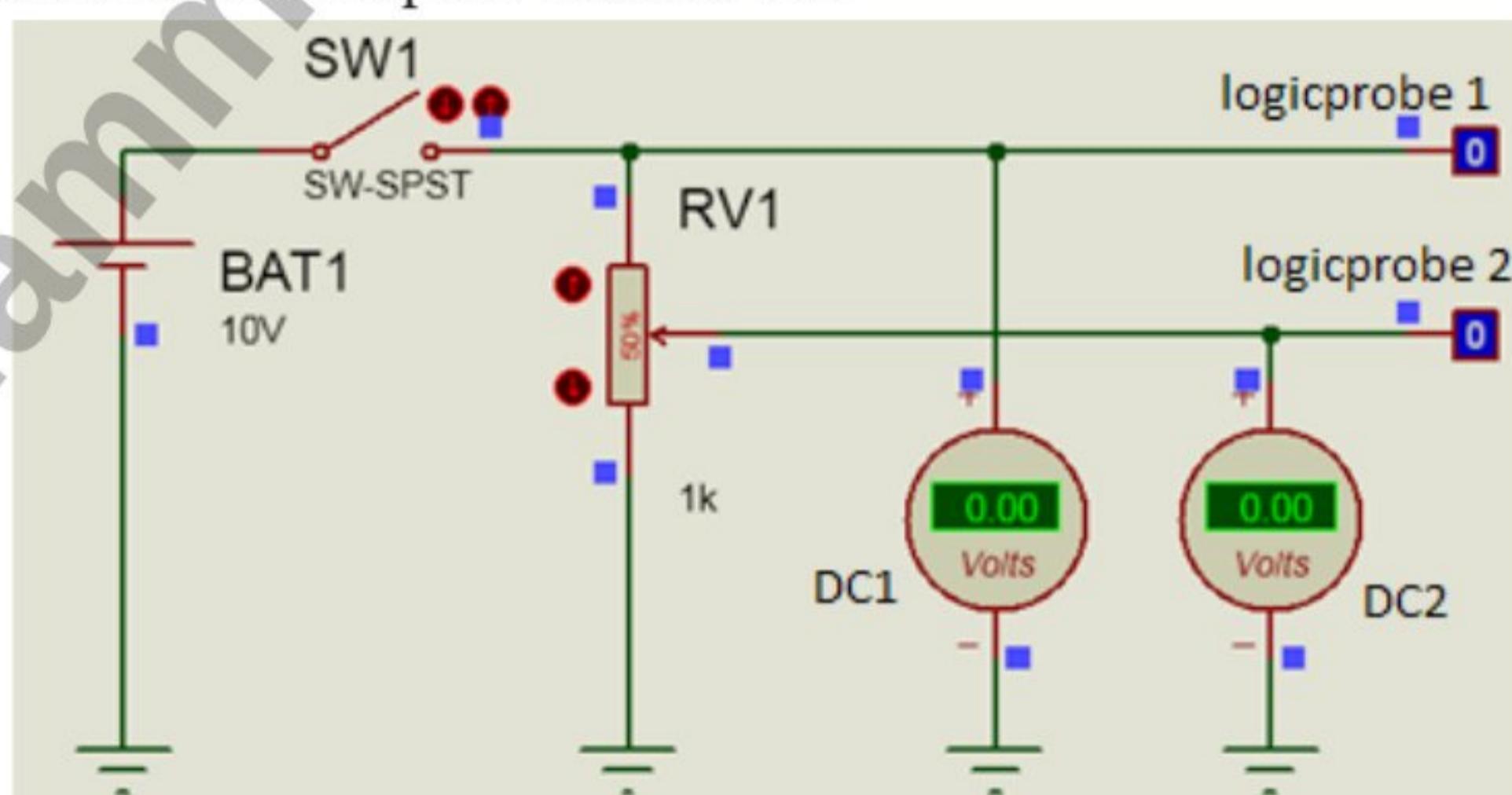


Gambar 2.5. Latihan range sinyal digital

Tabel 2.3. Komponen untuk rangkaian

No	Device	Information
1	Cell	Edit to 10V
2	SW-SPST	
3	POT-HG	
4	Logicprobe	
5	Ground	Pick from Terminals
6	DC Voltmeter	Pick from Instrument

- Jalankan Simulasi seperti Gambar 2.6!



Gambar 2.6. Jalankan simulasi

3. Klik SW1! Berdasarkan simulasi anda, isi titik-titik dibawah ini!
    - a. Voltmeter DC 1 : ..... Volt
    - b. Voltmeter DC 2 : ..... Volt
    - c. Logicprobe 1 menunjukan kondisi logika :.....
    - d. Logicprobe 2 menunjukan kondisi logika :.....
  4. Klik komponen RV1 (resistor variable/POT-HG) naik dan turun! Dan kemudian isi titik-titik dibawah ini!
    - a. Logicprobe 2 menunjukan kondisi logika **1 (High)**, jika Voltmeter DC 2 : ..... Volts sampai ..... Volts
    - b. Logicprobe 2 menunjukan kondisi logika **0 (Low)**, jika Voltmeter DC 2 : ..... Volts sampai ..... Volts
  5. Buat kesimpulan berdasarkan analisis anda di latihan range sinyal digital!
- .....  
.....  
.....  
.....

**Catatan :**

1. Logicprobe menunjukan apakah suatu tegangan termasuk dalam range tegangan digital.
2. Hanya dua kondisi tegangan yang diperbolehkan pada tegangan digital yaitu **0 Volt** dan **5 Volt**! (dengan toleransi)
3. Sinyal digital tidak diperkenankan melalui tegangan batas (seperti pada Logicprobe 1)

# MODUL 3

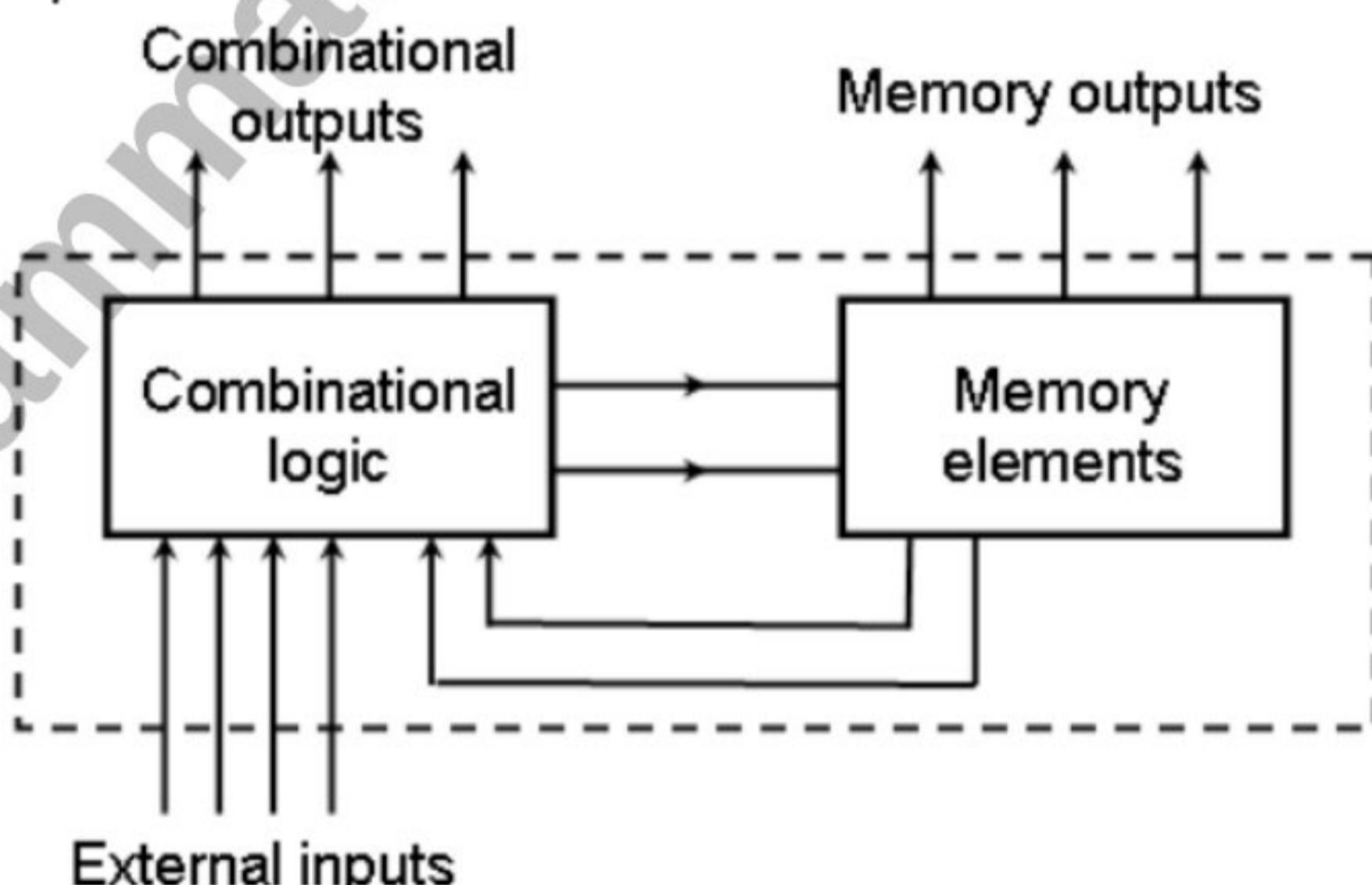
## GERBANG LOGIKA

### TUJUAN PRAKTIKUM

Mahasiswa dapat membangun rangkaian gerbang logika dan mampu menganalisa bagaimana cara kerja gerbang logika.

### TEORI

Gerbang logika adalah sebuah perangkat yang mengimplementasikan fungsi boolean. Fungsi boolean melakukan operasi logika pada satu atau lebih masukan dan menghasilkan keluaran sebuah logika baru. Dengan menggunakan kombinasi benar atau salah seperti halnya cara penyelesaian masalah sehari-hari. Dapat juga dengan menggunakan langkah-langkah berpikir logis berdasarkan keputusan-keputusan masa lalu (memory) untuk menyelesaikan masalah, sehingga disebut dengan logika sekuensial (terurut) seperti pada Gambar 2.1. gerbang logika mempunyai table kebenaran. Table kebenaran ini menyediakan daftar setiap kombinasi yang mungkin dari masukan biner dan hasil keluaran terkait. Tabel 3.1. memperlihatkan tipe-tipe gerbang logika disertai simbol dan tabel kebenarannya.

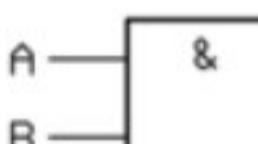


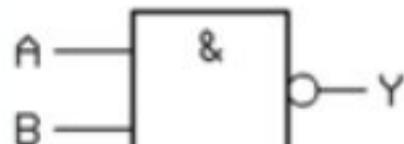
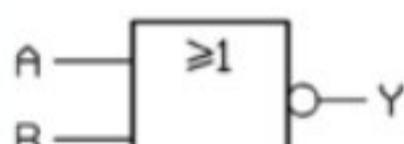
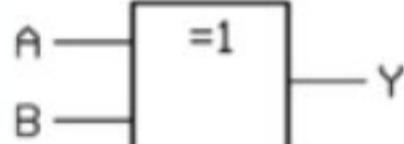
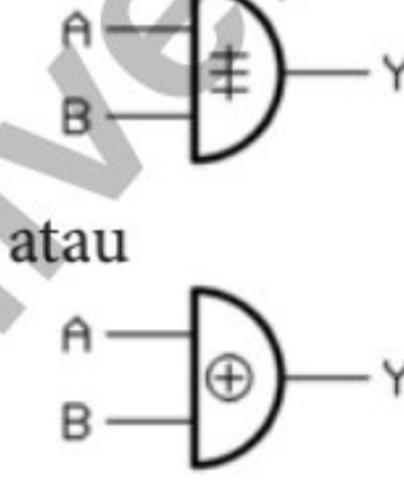
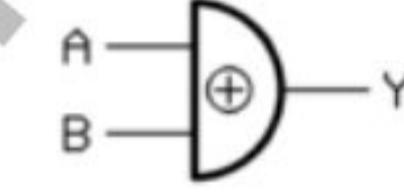
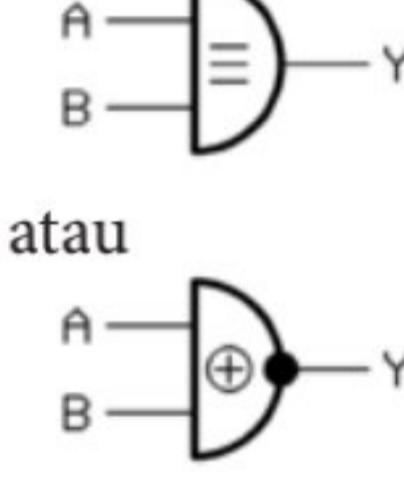
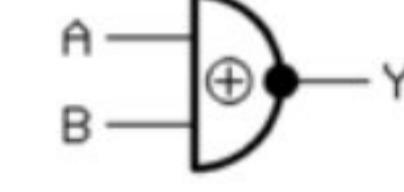
Gambar 2.1. Representasi digital

Dioda dan transistor komponen penyusun paling umum untuk mengimplementasikan gerbang logika yaitu sebagai saklar elektronik. Selain itu gerbang logika juga dibangun menggunakan relai eletromagnetik, logika fluida, logika pneumatik, optik, molekul, atau bahkan elemen mekanik. Gerbang logika dapat berjalan dengan cara yang sama dimana fungsi boolean berada di dalamnya, memungkinkan semua algoritma dan operasi matematikanya dapat dijelaskan dengan logika boolean.

Peralatan yang mengandung gerbang logika seperti *multiplexer*, *register*, *arithmetic logic units* (ALU), dan memori sebuah komputer, yang terdapat lebih dari 100 juta gerbang. Gerbang-gerbang tersebut terbuat dari efek medan transistor atau dikenal dengan *field-effect transistor* (FET), dan khususnya terbuat dari MOSFET (metal-oxide-semiconductor field-effect transistor). Untuk gerbang logika kompleks seperti AND-OR-Invert (AOI) dan OR-AND-Invert (OAI) lebih sering digunakan. Hal ini karena konstruksi gerbang logika tersebut menggunakan MOSFET yang lebih sederhana dan efisien dibanding jumlah gerbang tunggal.

Tabel 3.1. Jenis-jenis gerbang logika, symbol dan table kebenarannya

Name and Function	Symbol			Truth Table
	IEC 60617-12	US-Norm	DIN 40700 (Before 1976)	
<u>AND - Gate</u>  (AND) $Y = A \wedge B$ $Y = A \cdot B$ $Y = AB$	 A —> & —> Y	 A —> —> Y	 A —> —> Y	A    B    Y 0    0    0 0    1    0 1    0    0 1    1    1
<u>OR - Gate</u>  (OR) $Y = A \vee B$ $Y = A + B$	 A —> ≥1 —> Y	 A —> —> Y	 A —> —> Y	A    B    Y 0    0    0 0    1    1 1    0    1 1    1    1
<u>NOT - Gate</u>  (NOT, Complement - Gate, Inverter) $Y = \overline{A}$ $Y = \neg A$	 A —> 1 —> Y	 A —> O —> Y	 A —> —> Y	A    Y 0    1 1    0

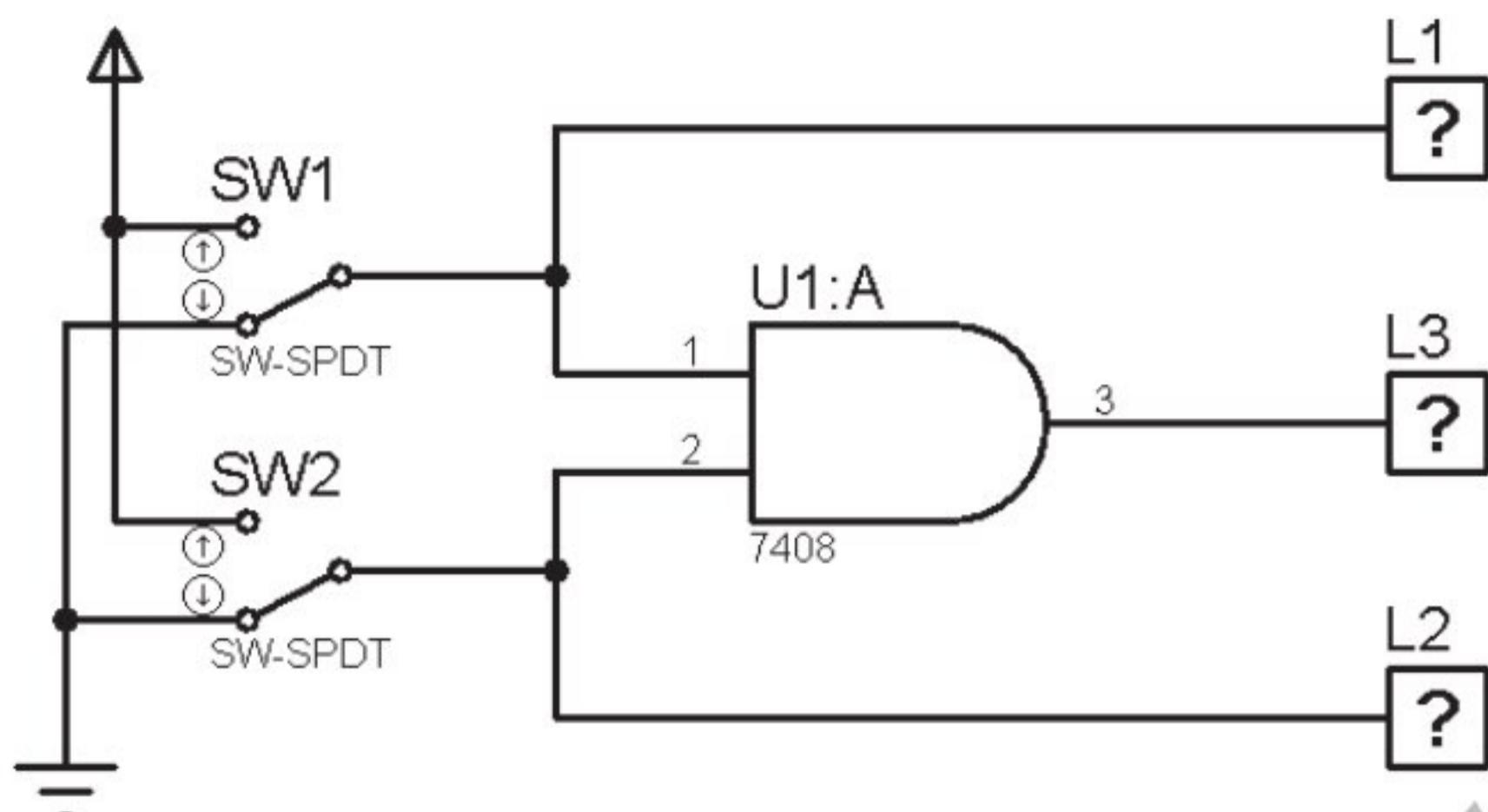
Name and Function	Symbol			Truth Table															
	IEC 60617-12	US-Norm	DIN 40700 (Before 1976)																
<u>NAND - Gate</u> (Not-AND) $Y = \overline{A \wedge B}$ $Y = A \bar{\wedge} B$ $Y = \overline{AB}$				<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <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>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
<u>NOR - Gate</u> (Not-OR) $Y = \overline{A \vee B}$ $Y = A \bar{\vee} B$ $Y = \overline{A+B}$				<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</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>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
<u>XOR - Gate</u> (Antivalen, Exclusive-OR) $Y = A \underline{\vee} B$ $Y = A \oplus B$			 atau 	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</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>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
<u>XNOR - Gate</u> (Ekuivalen, Not-Exclusive- OR) $Y = \overline{A \vee B}$ $Y = A \bar{\vee} B$ $Y = \overline{A \oplus B}$			 atau 	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</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>1</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

## KEGIATAN PRAKTIKUM

### Percobaan 1. Gerbang AND

#### 1. Buat rangkaian pada Gambar 3.2!

Buat dengan menggunakan gerbang AND (IC 7408), SW-SPDT, dan logic probe! Pilih VCC and ground dari terminal mode. SW1 dan SW2 sebagai input, L1 dan L2 berperan sebagai indikator inpur, dan L3 perperan sebagai output dari gerbang AND.

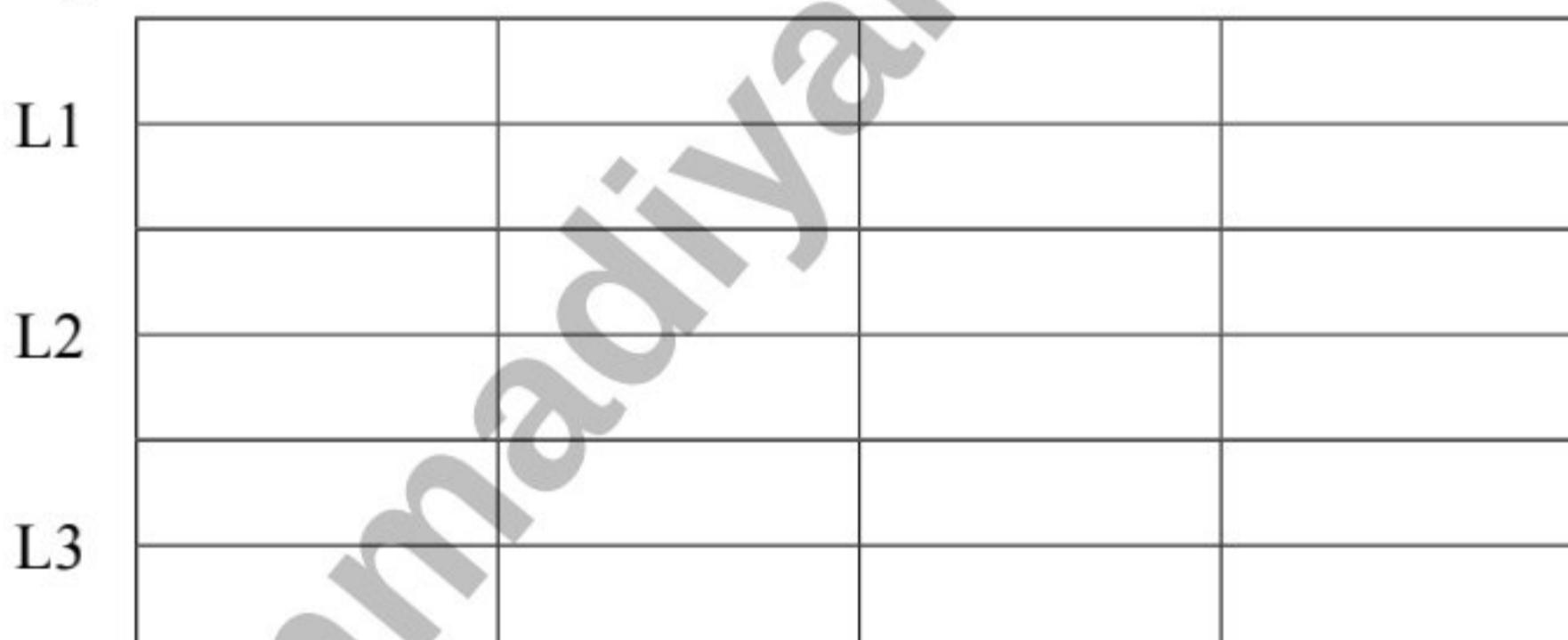


Gambar 3.2.Gerbang AND

2. Fungsi Boolean :  $L_3 = L_1 \cdot L_2$  atau  $L_3 = L_1 \cdot L_2$
3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

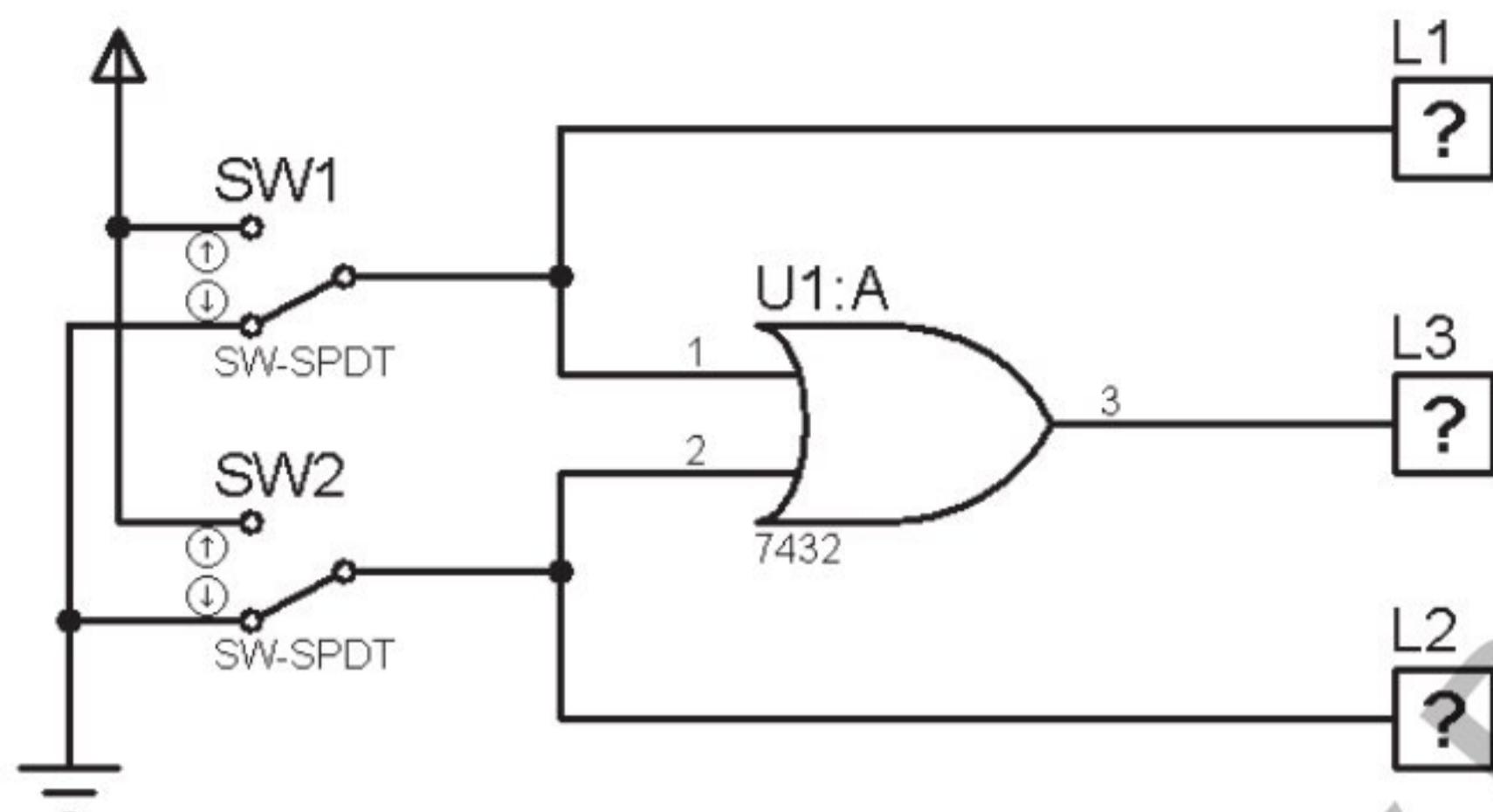
4. Diagram waktu



### Percobaan 2. Gerbang OR

1. Buat rangkaian pada Gambar 3.3!

Buat dengan menggunakan gerbang OR (IC 7432), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode. SW1 dan SW2 sebagai input, L1 dan L2 berperan sebagai indikator input, dan L3 berperan sebagai output dari gerbang OR.



Gambar 3.3. Gerbang OR

2. Fungsi Boolean :  $L_3 = L_1 + L_2$
3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

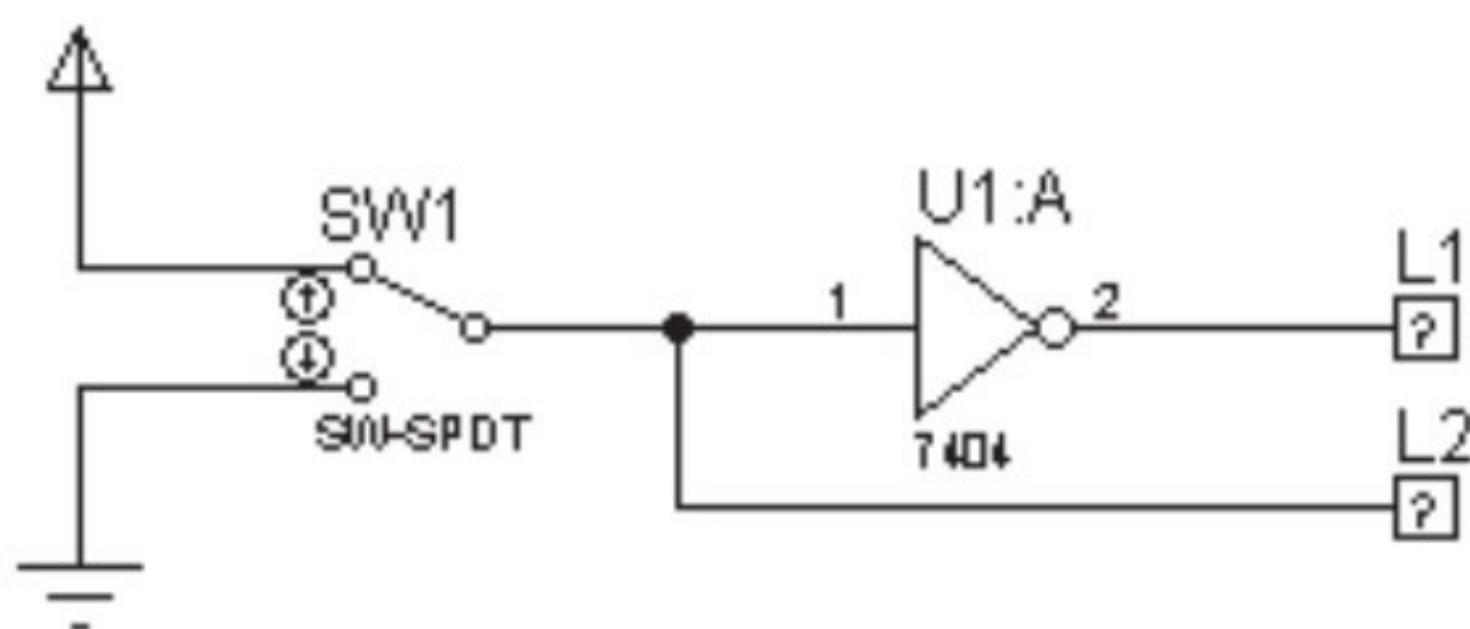
4. Diagram Waktu



### Percobaan 3. Gerbang NOT

1. Buat rangkaian pada Gambar 3.4!

Buat dengan menggunakan gerbang NOT(IC 7404), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode. SW1 sebagai input, L2 sebagai indikator input, dan L3 sebagai output dari Gerbang NOT.



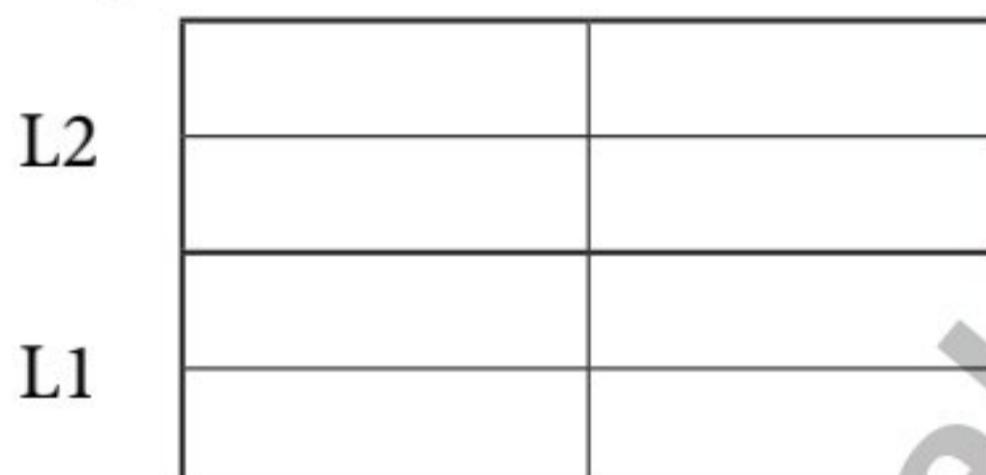
Gambar 3.4. Gerbang NOT

2. Fungsi Boolean :  $L_1 = \sim L_2$  atau  $L_1 = \overline{L_2}$

### 3. Tabel kebenaran

SW1	L2	L1
0		
1		

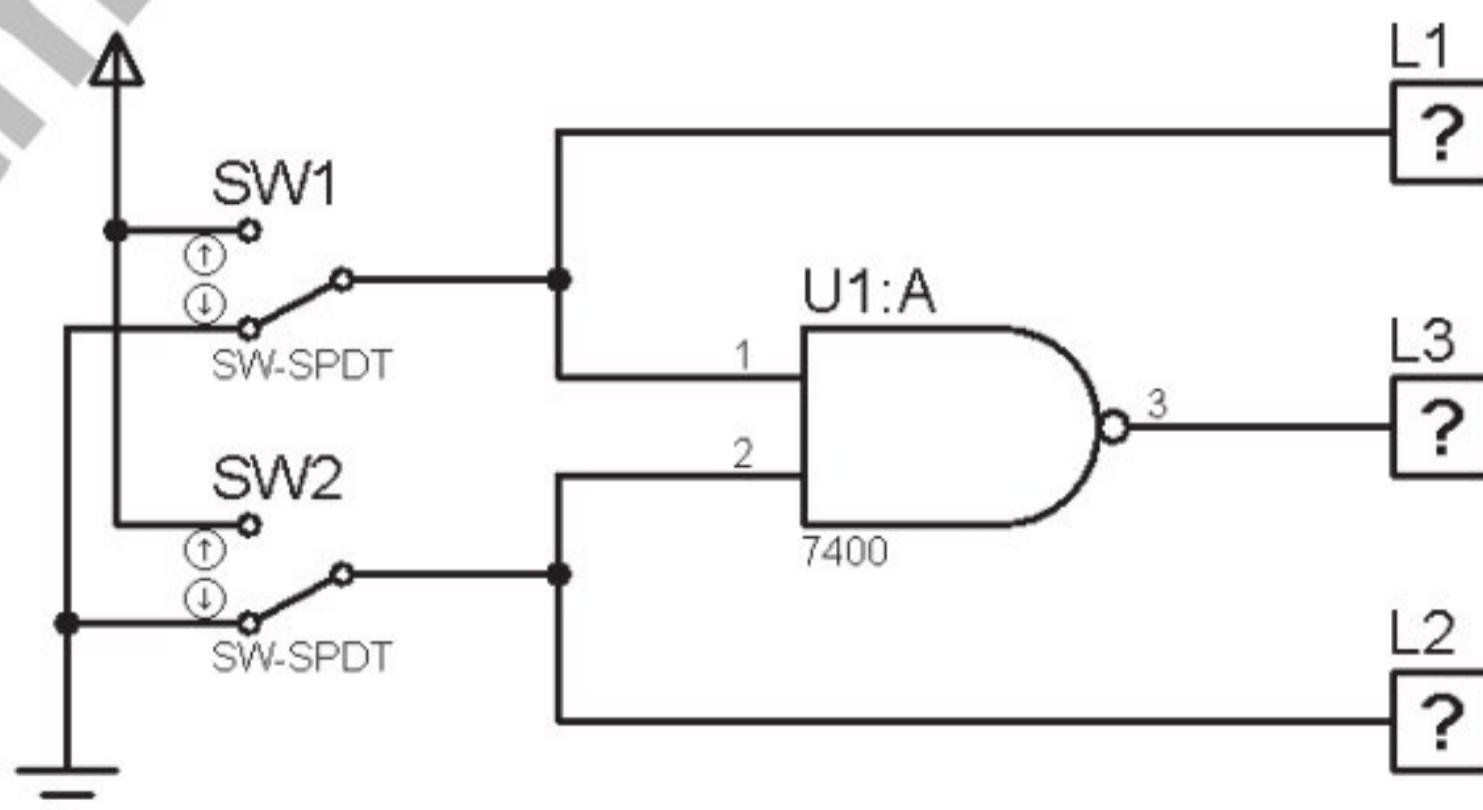
#### 4. Diagram waktu



## Percobaan 4. Gerbang NAND

1. Buat rangkaian pada Gambar 3.5!

Buat dengan menggunakan gerbang NAND (IC 7400), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode. SW1 dan SW2 sebagai input, L1 dan L2 berperan sebagai indikator input, dan L3 sebagai Output dari gerbang NAND.



Gambar 3.5. Gerbang NAND

2. Fungsi Boolean :  $L3 = \overline{L1L2}$  atau  $L3 = \overline{L1} \cdot \overline{L2}$

3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

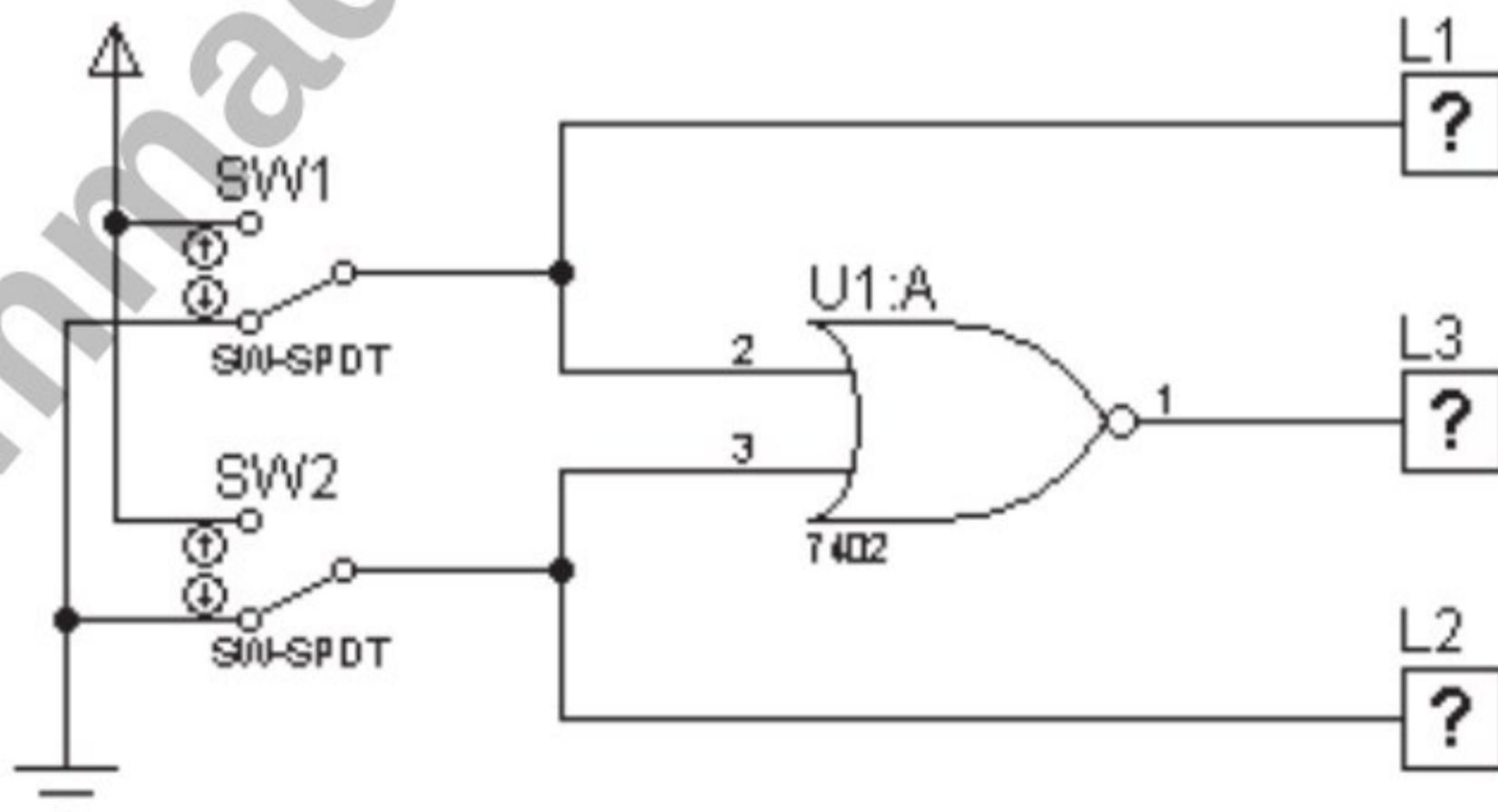
4. Diagram waktu



### Percobaan 5. Gerbang NOR

1. Buat rangkaian pada Gambar 3.6!

Buat dengan menggunakan gerbang NOR(IC 7402), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode. SW1 dan SW2 sebagai input, L1 dan L2 sebagai indikator input, dan L3 sebagai output dari gerbang NOR.



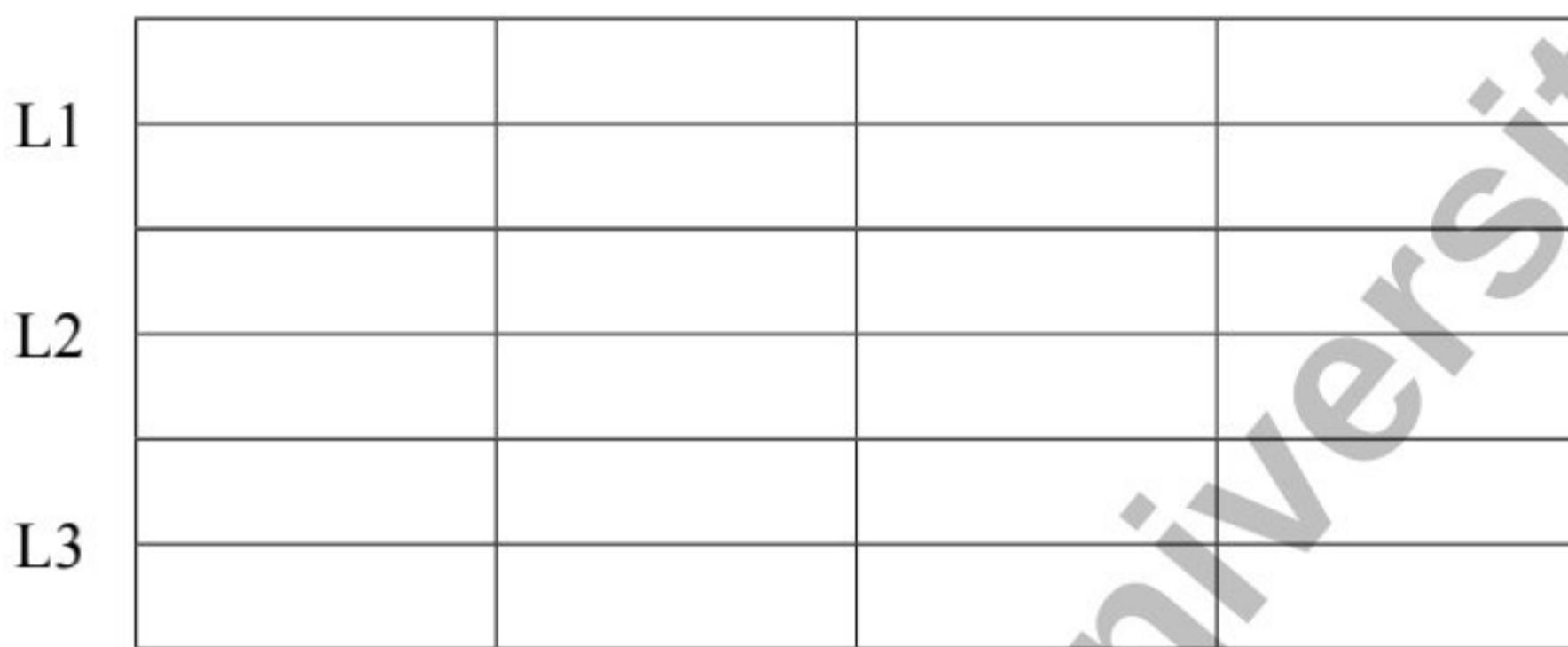
Gambar 3.6. Gerbang NOR

2. Fungsi Boolean :  $L3 = \overline{L1 + L2}$

3. Tabel kebenaran

SW1	SW1	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

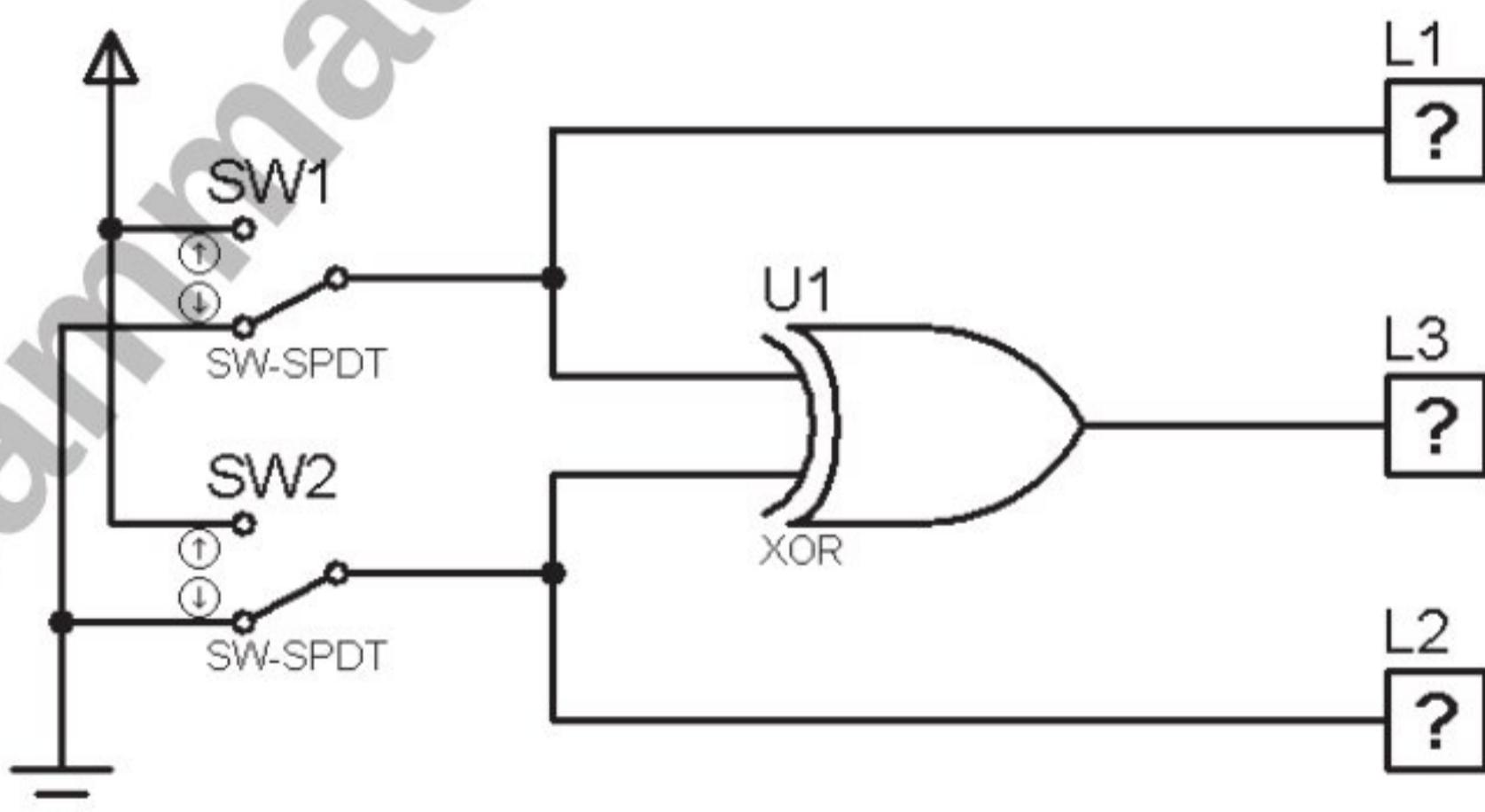
4. Diagram waktu



### Percobaan 6. Gerbang XOR

1. Buat rangkaian pada Gambar 7!

Buat dengan menggunakan gerbang XOR (IC 4070), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode. SW1 dan SW2 sebagai input, L1 dan L2 berperan sebagai indikator input, dan L3 berperan sebagai output gerbang XOR.



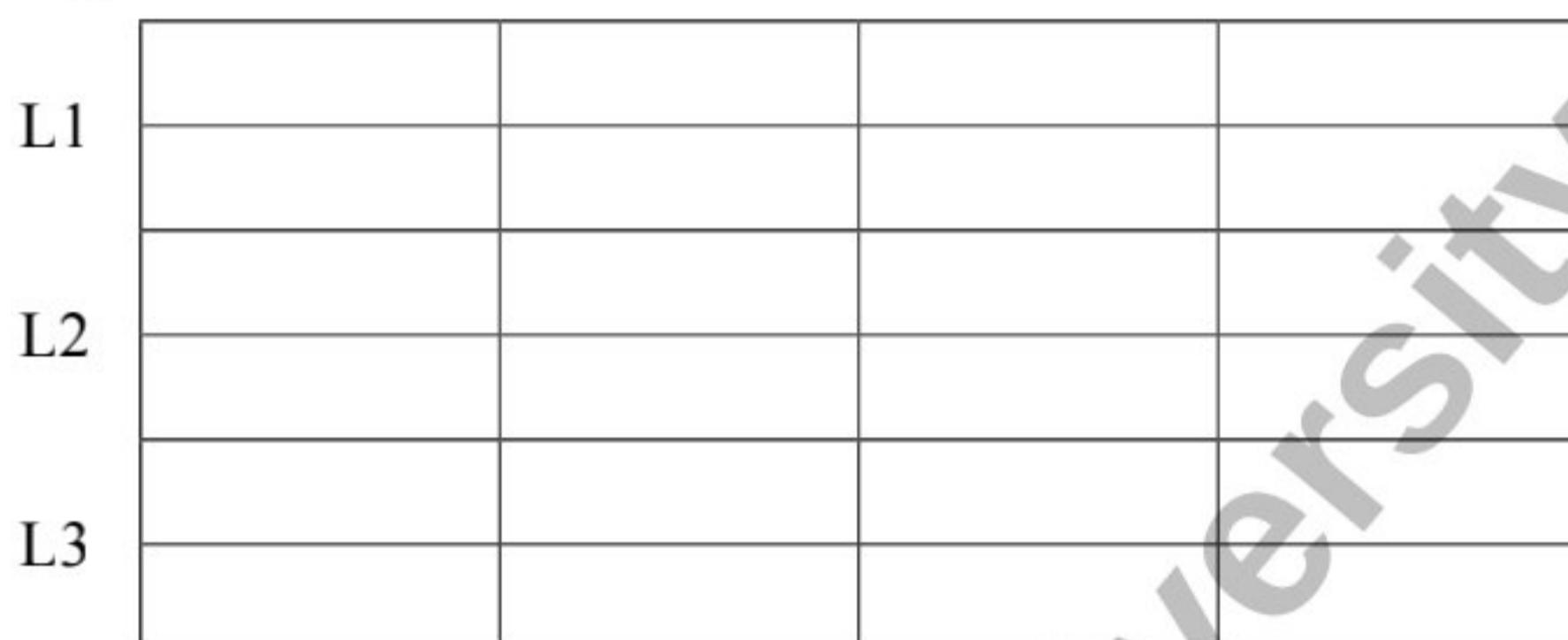
Gambar 3.7. Gerbang XOR

2. Fungsi Boolean  $L3 = L1 \oplus L2$

3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

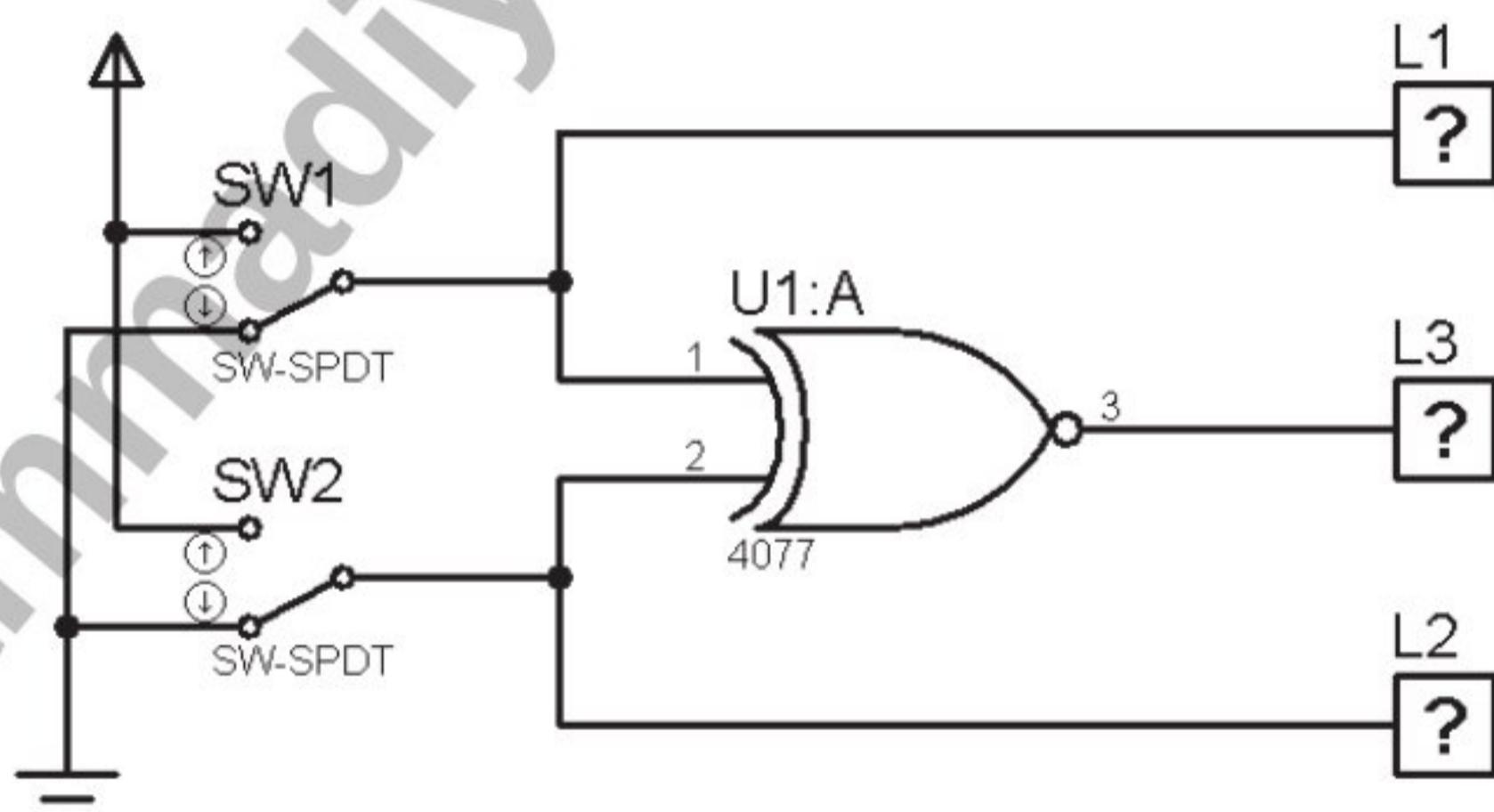
4. Diagram waktu



**Percobaan 7. Gerbang XNOR**

1. Buat rangkaian pada Gambar 3.8!

Buat dengan menggunakan gerbang XNOR (IC 4077), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode. SW1 dan SW2 sebagai input, L1 dan L2 berperan sebagai indikator input, dan L3 sebagai output dari gerbang XNOR.



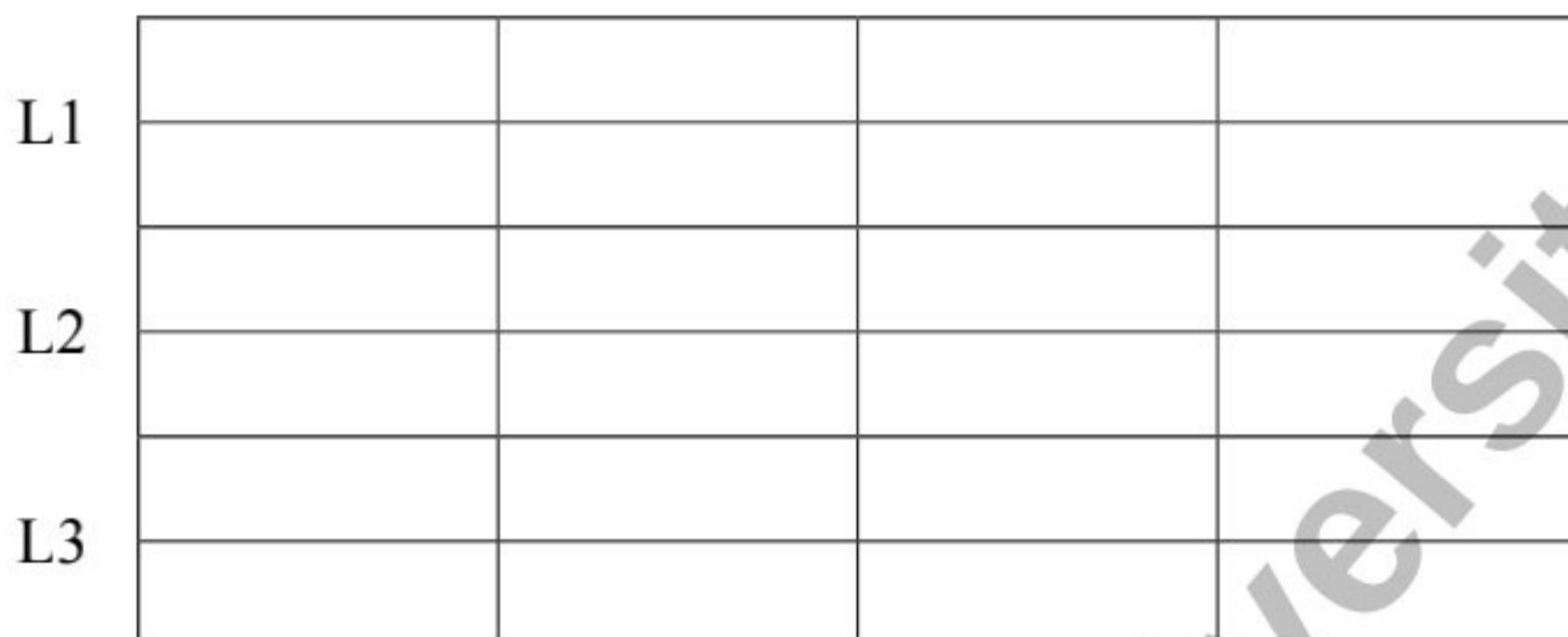
Gambar 3.8. Gerbang XNOR

2. Fungsi Boolean :  $L3 = \overline{L1 \oplus L2}$

3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

4. Diagram waktu



# MODUL 4

## KOMBINASI GERBANG LOGIKA

### TUJUAN PRAKTIKUM

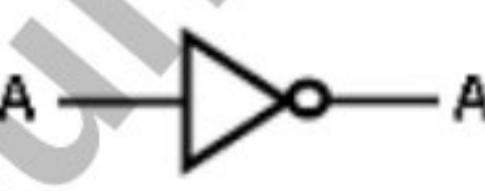
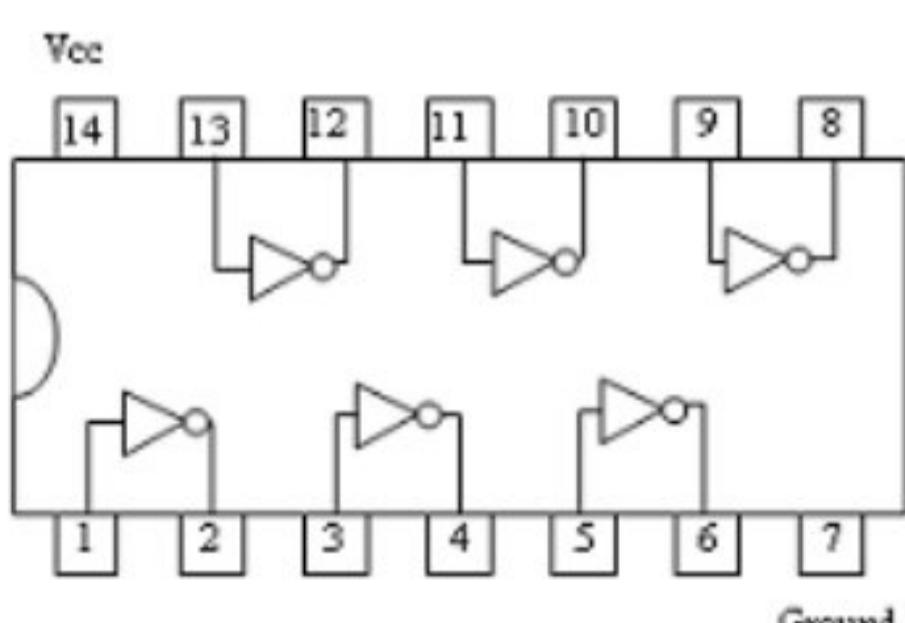
Mahasiswa dapat membangun logika menggunakan kombinasi gerbang logika lain.

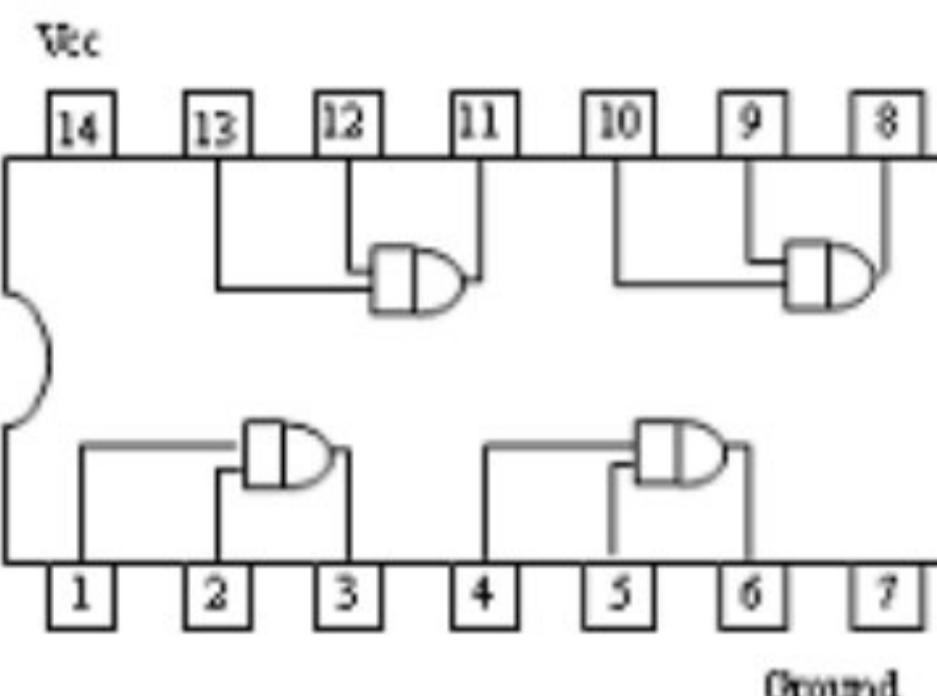
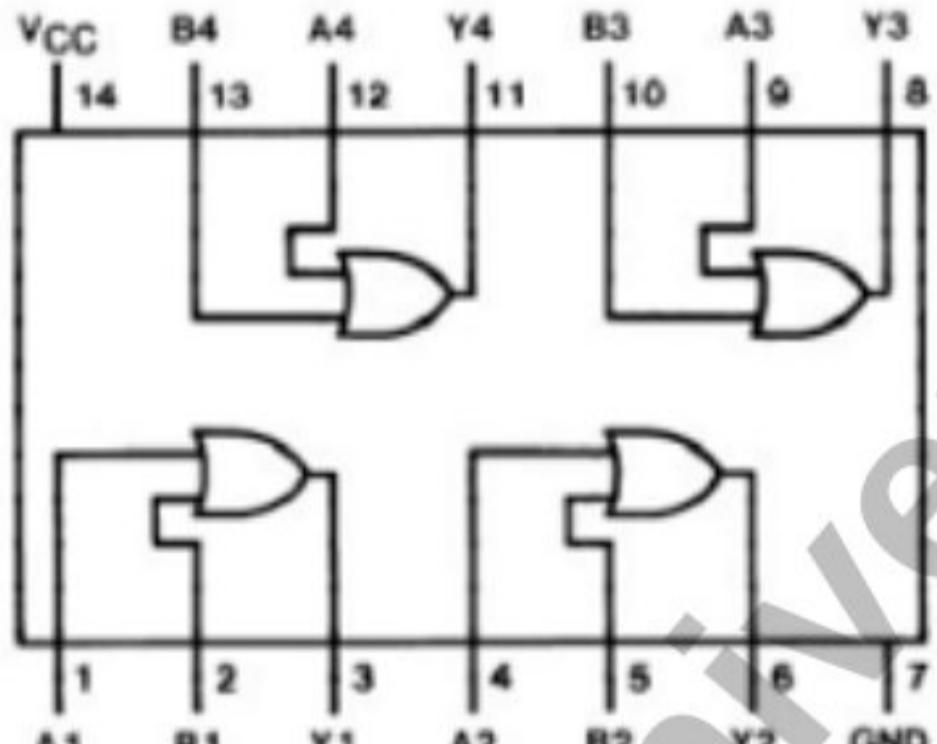
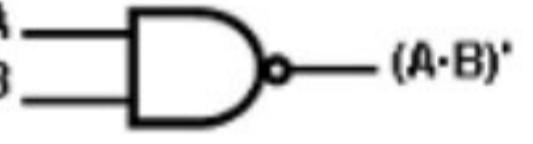
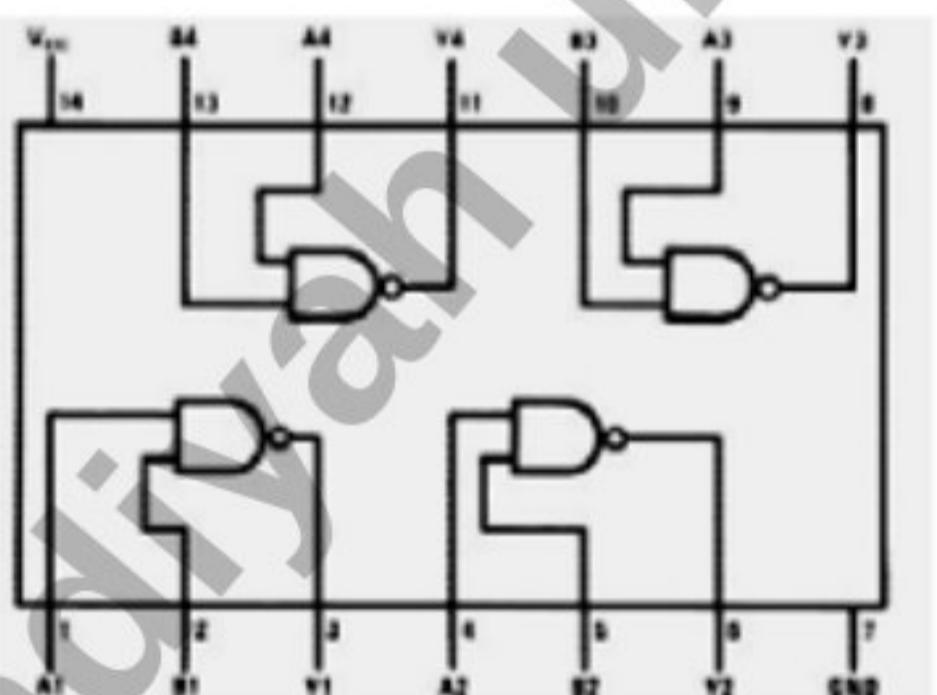
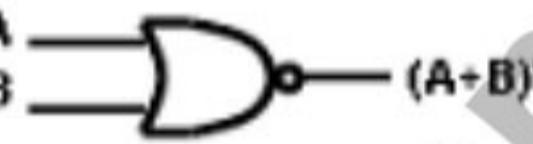
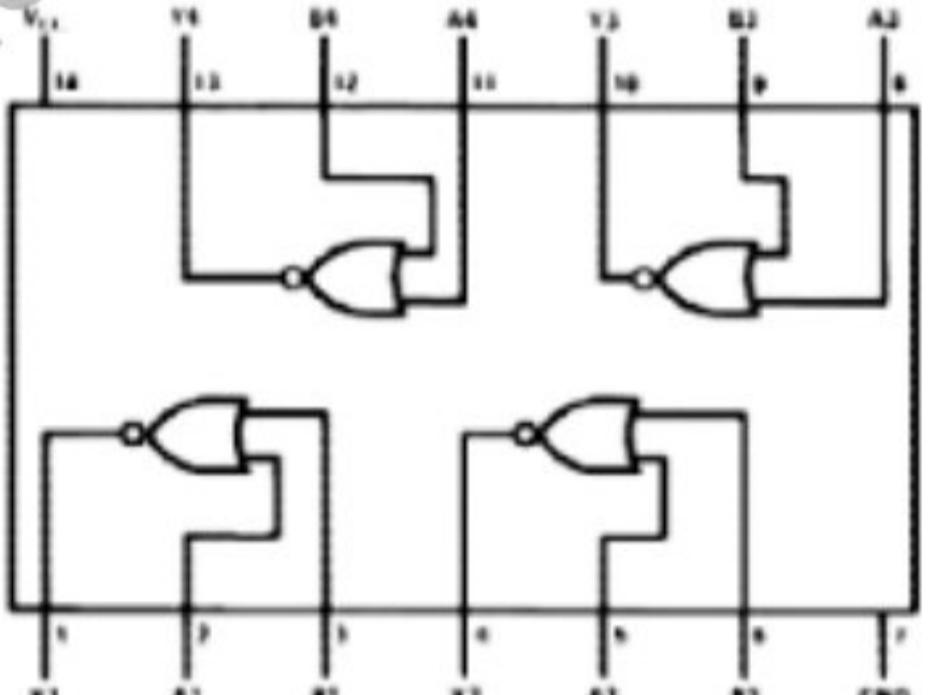
### TEORI

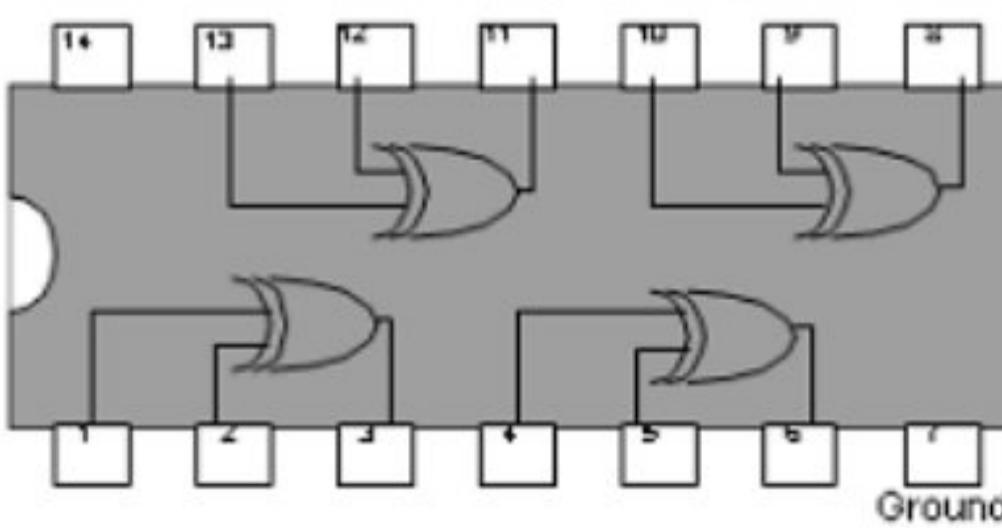
Dengan mengacu pada modul sebelumnya, pada modul 4 ini merupakan pemahaman lanjutan dengan mengkombinasikan gerbang-gerbang logika. Seperti telah diketahui bahwa komponen penyusun gerbang logika dapat berupa transistor atau diode. Family dari jenis TTL (Transistor-Transistor Logic) merupakan gerbang logika yang tersusun dari transistor BJT. Walaupun namanya TTL, namun tidak hanya tersusun dari transistor saja tetapi terdapat komponen-komponen pendukung di dalamnya yaitu resistor dan PN-junction diode. Gerbang-gerbang logika dari family TTL ini banyak digunakan dalam perancangan IC digital pada level LSI.

Gerbang-gerbang logika yang umumnya tersedia dibuat dengan teknologi *Integrated Circuit* (IC). Integrasi gerbang-gerbang logika dasar seperti NOT, AND, OR, NAND, NOR, dan XOR pada umumnya dibuat dalam skala kecil (*small scale Integration/ SSI*) yang terkandung dua sampai enam gerbang dalam setiap produk. Tabel 4.1. merupakan kumpulan gerbang logika pada IC TTL 74LS.

Tabel 4.1. kumpulan gerbang logika pada IC TTL 74LS.

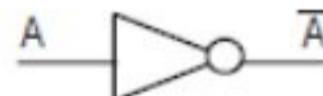
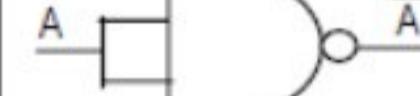
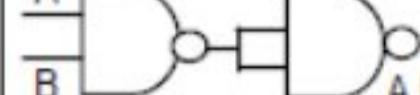
Gerbang logika	TTL 74LS Family	Keterangan
		Gerbang 'NOT' diwakili oleh IC 47LS04 Hex Inverter

Gerbang logika	TTL 74LS Family	Keterangan
		Gerbang 'AND' diwakili oleh IC 47LS08 Quad 2-input AND Gate
		Gerbang 'OR' diwakili oleh IC 47LS32 Quad 2-input OR Gate
		Gerbang 'NAND' diwakili oleh IC 47LS00 Quad 2-input NAND Gate
		Gerbang 'NOR' diwakili oleh IC 47LS02 Quad 2-input NOR Gate

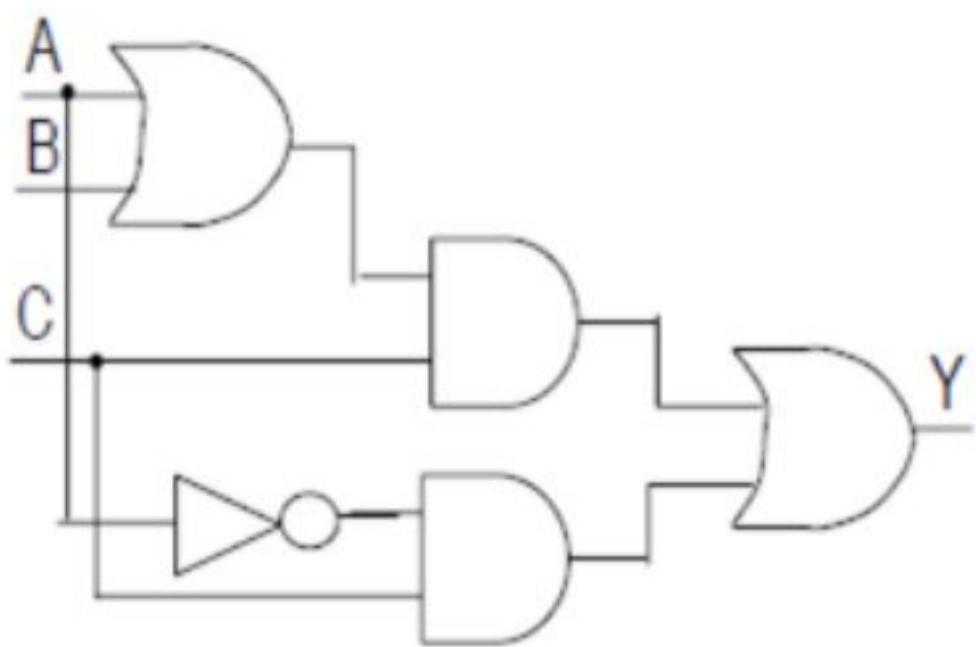
Gerbang logika	TTL 74LS Family	Keterangan
		Gerbang ‘XOR’ diwakili oleh IC 47LS86 Quad 2-input XOR Gate

Dalam kenyataannya, gerbang-gerbang logika dasar tersebut berbeda atau tidak dapat memenuhi kebutuhan dalam perancangan real pembuatan rangkaian digital. Gerbang yang paling banyak tersedia di pasaran adalah gerbang dengan dua atau tiga masukan. Apabila dalam sebuah rancangan membutuhkan gerbang dengan empat atau lima masukan maka akan mengalami kesulitan. Oleh karena itu perlu merubah dan mengkombinasikan gerbang-gerbang dasar sesuai dengan kebutuhan.

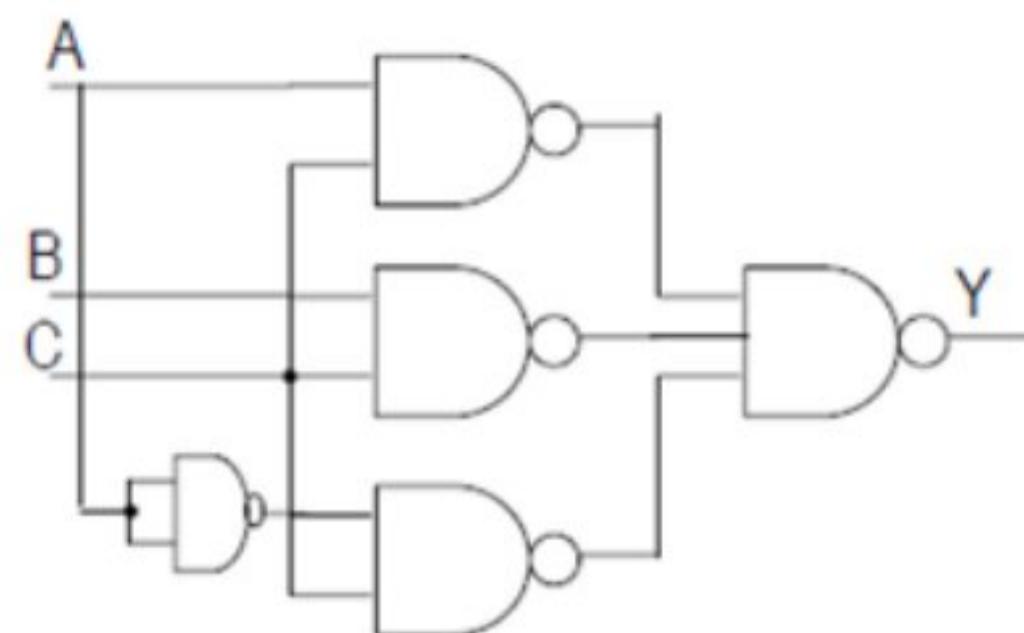
Gerbang NAND dan NOR merupakan gerbang universal, artinya hanya dengan menggunakan jenis gerbang ini saja dapat menggantikan fungsi dari tiga gerbang dasar yang lain seperti AND, OR, NOT. Gerbang NAND dan NOR juga disebut gerbang multilevel karena dengan mengimplementasikan gerbang ini akan ada banyak tingkatan mulai dari sistem input sampai sistem output. Keuntungan dari penggunaan salah satu jenis gerbang ini adalah mengoptimalkan pemakaian seluruh gerbang yang terdapat dalam sebuah IC sehingga menghemat biaya. Untuk memudahkan mempelajari kombinasi gerbang logika, dianjurkan telah paham mengenai gerbang logika (Gambar 3.1 pada Modul 3). Gambar 4.1 merupakan contoh gerbang NAND. Gambar 4.2 merupakan implementasi dari gerbang NAND.

	Bentuk Asal	Dengan NAND
NOT		
AND		
OR		

Gambar 4.1. Substitusi gerbang dasar menggunakan gerbang NAND



Rangkaian Asal



Rangkaian Dengan NAND saja

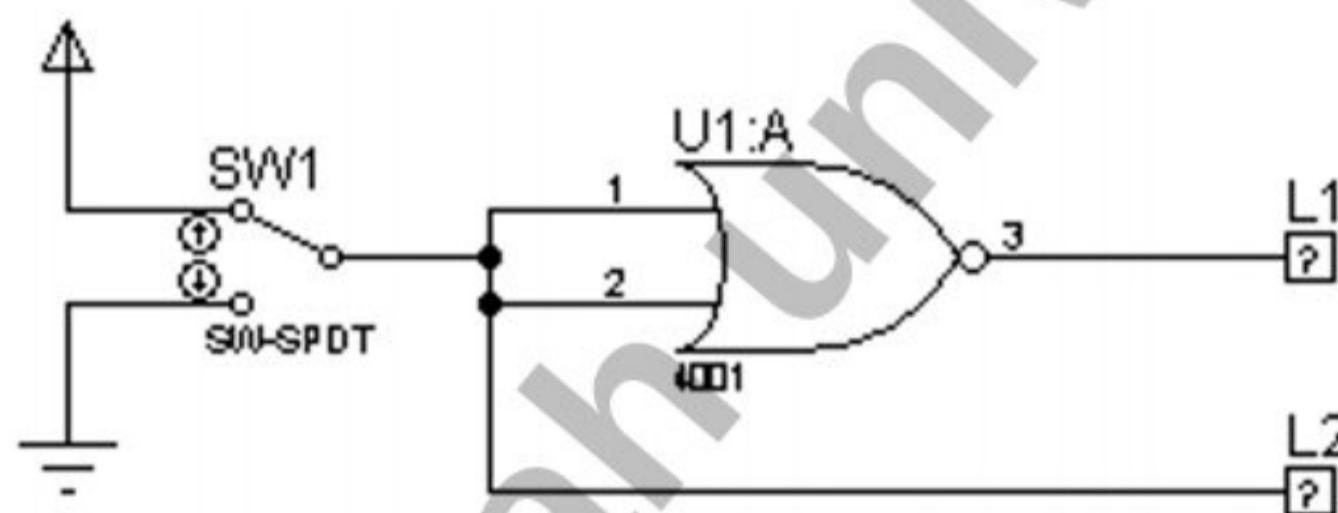
Gambar 4.2. Implementasi dari gerbang NAND

## KEGIATAN PRAKTIKUM

### Percobaan 1 : Substituti Pengganti Gerbang Logika

- Buat rangkaian pada Gambar 4.3!

Buat dengan menggunakan gerbang NOR (IC 4001), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode.



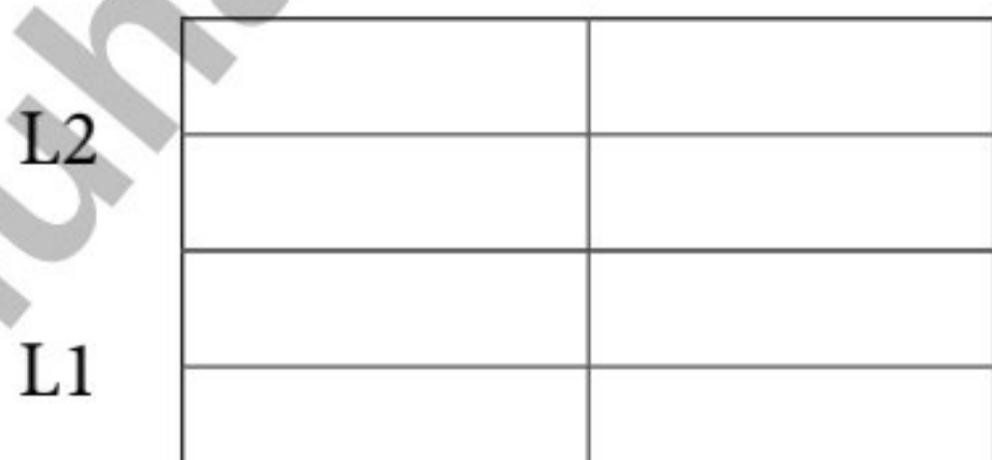
Gambar 4.3. Variasi Gerbang 1

- Fungsi Boolean :  $L_1 = \overline{L_2 + L_2} = \overline{L_2}$

- Tabel kebenaran

SW1	L2	L1
0		
1		

- Diagram waktu



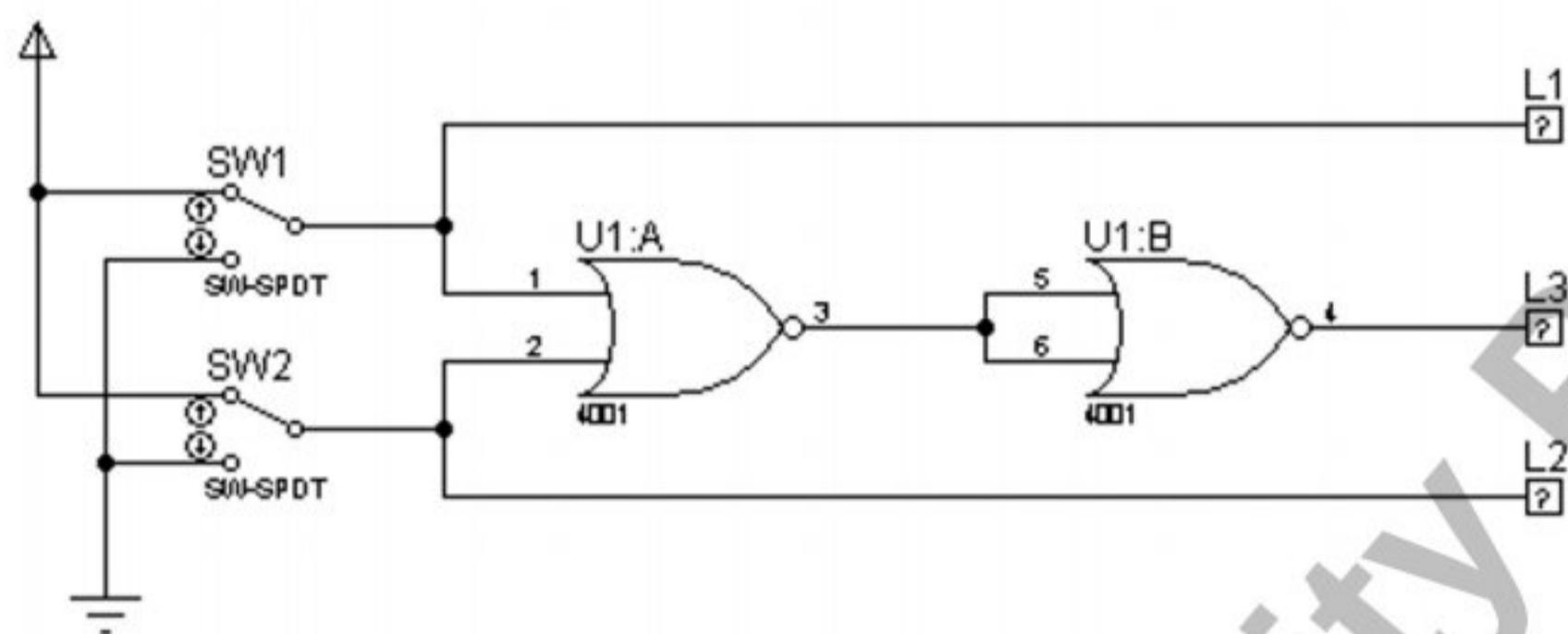
- Kesimpulan :

Gerbang NOR pada Gambar 4.3 membentuk logika dari gerbang .....

## Percobaan 2 : Substituti Pengganti Gerbang Logika

1. Buat rangkaian pada Gambar 4.4!

Buat dengan menggunakan gerbang NOR (IC 4001), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode.



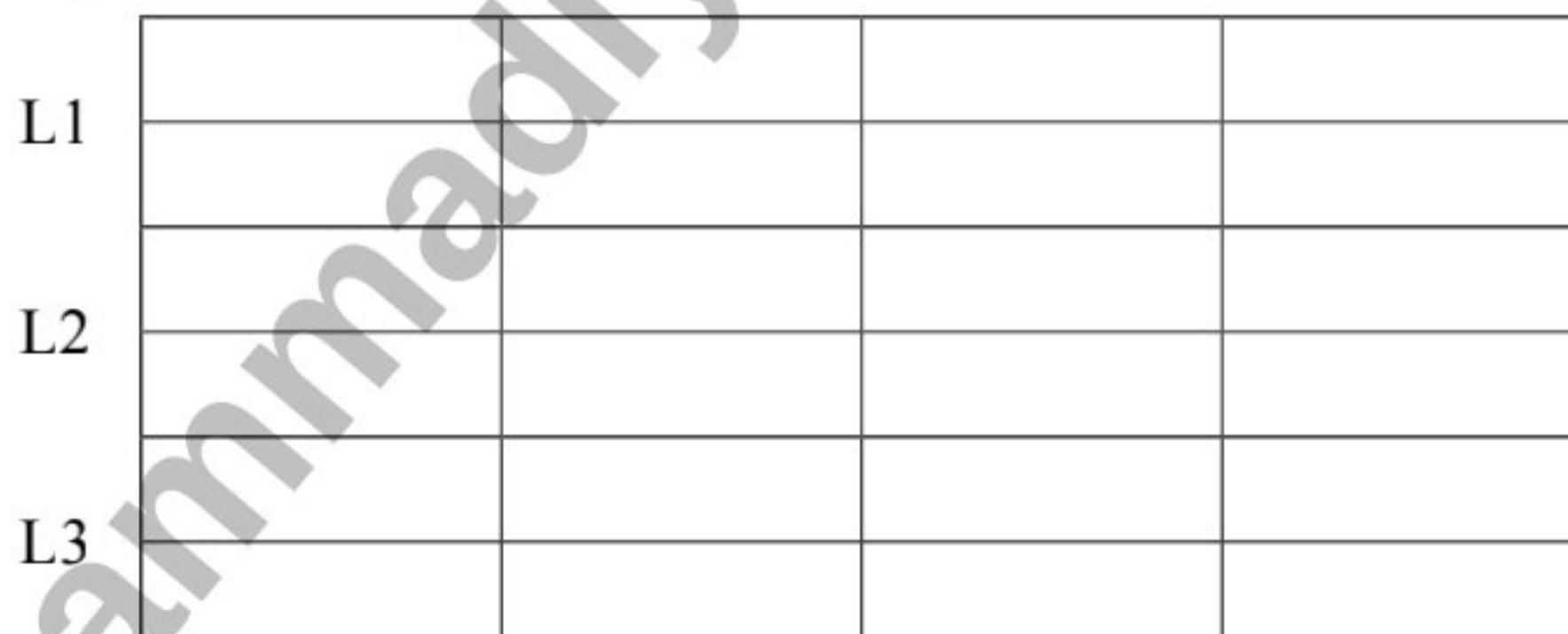
Gambar 4.4. Variasi Gerbang 2

2. Fungsi Boolean :  $L_3 = \overline{\overline{L_1 + L_2}} = L_1 + L_2$

3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

4. Diagram waktu



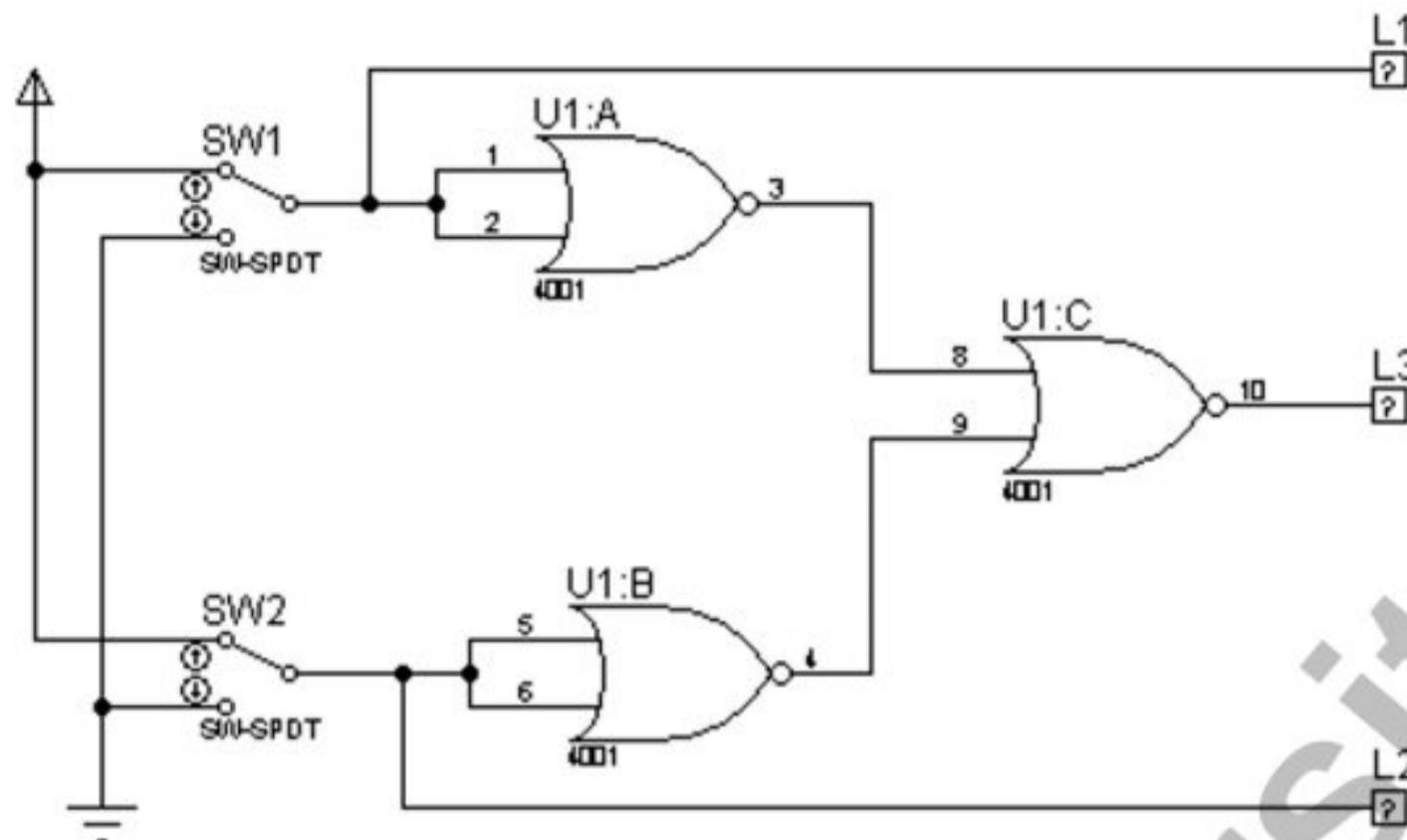
5. Kesimpulan:

Gerbang NOR pada Gambar 4.4 membentuk logika dari gerbang.....

### Percobaan 3 : Substituti Pengganti Gerbang Logika

- Buat rangkaian pada Gambar 4.5!

Buat dengan menggunakan gerbang NOR (IC 4001), SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode.



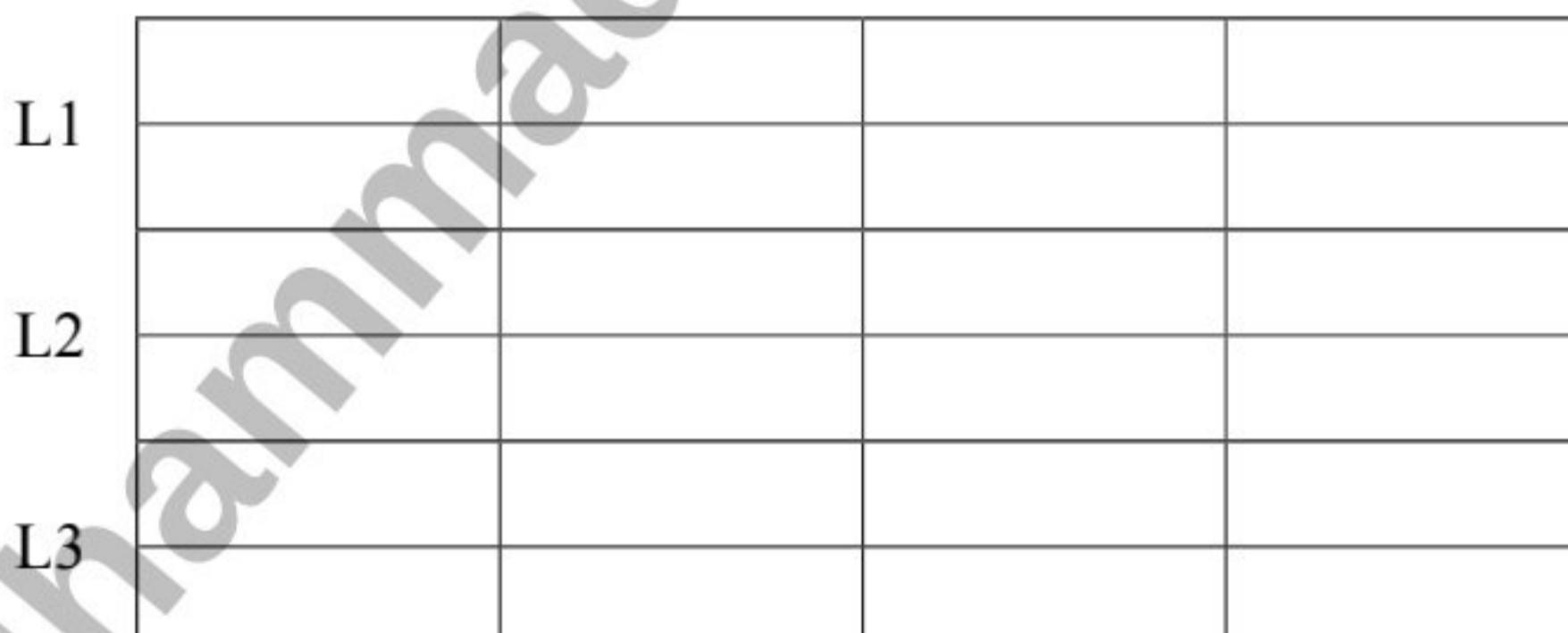
Gambar 4.5. Variasi Gerbang 3

- Fungsi Boolean :  $L3 = \overline{\overline{L1}} + \overline{\overline{L2}} = \dots\dots$

- Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

- Diagram waktu



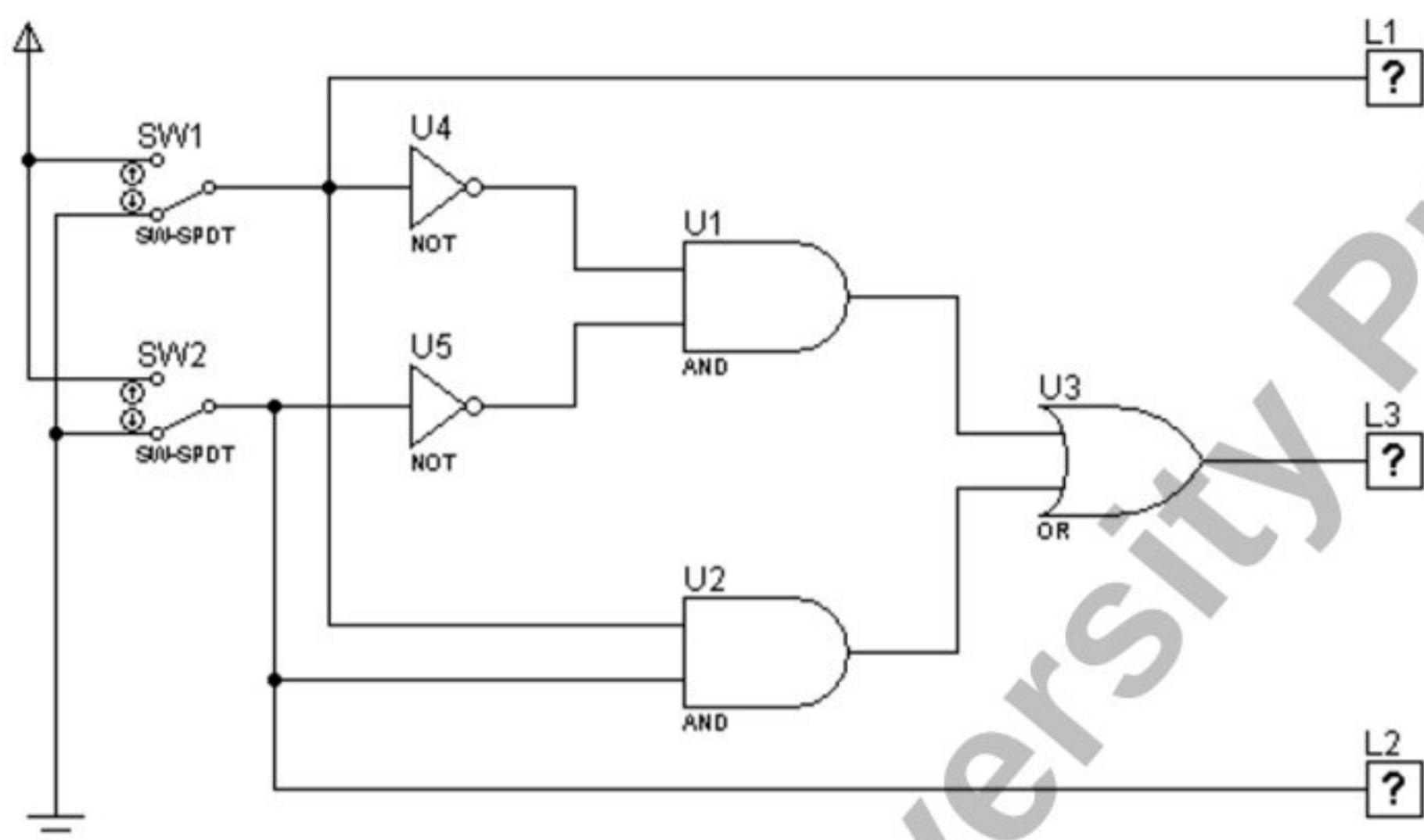
- Kesimpulan :

Gerbang NOR pada Gambar 4.5 membentuk logika dari gerbang .....  
.....

## Percobaan 4 : Substituti Pengganti Gerbang Logika

- Buat rangkaian pada Gambar 4.6!

Buat dengan menggunakan gerbang AND, NOT, OR, SW-SPDT, dan logic probe! Pilih VCC dan ground dari terminal mode.



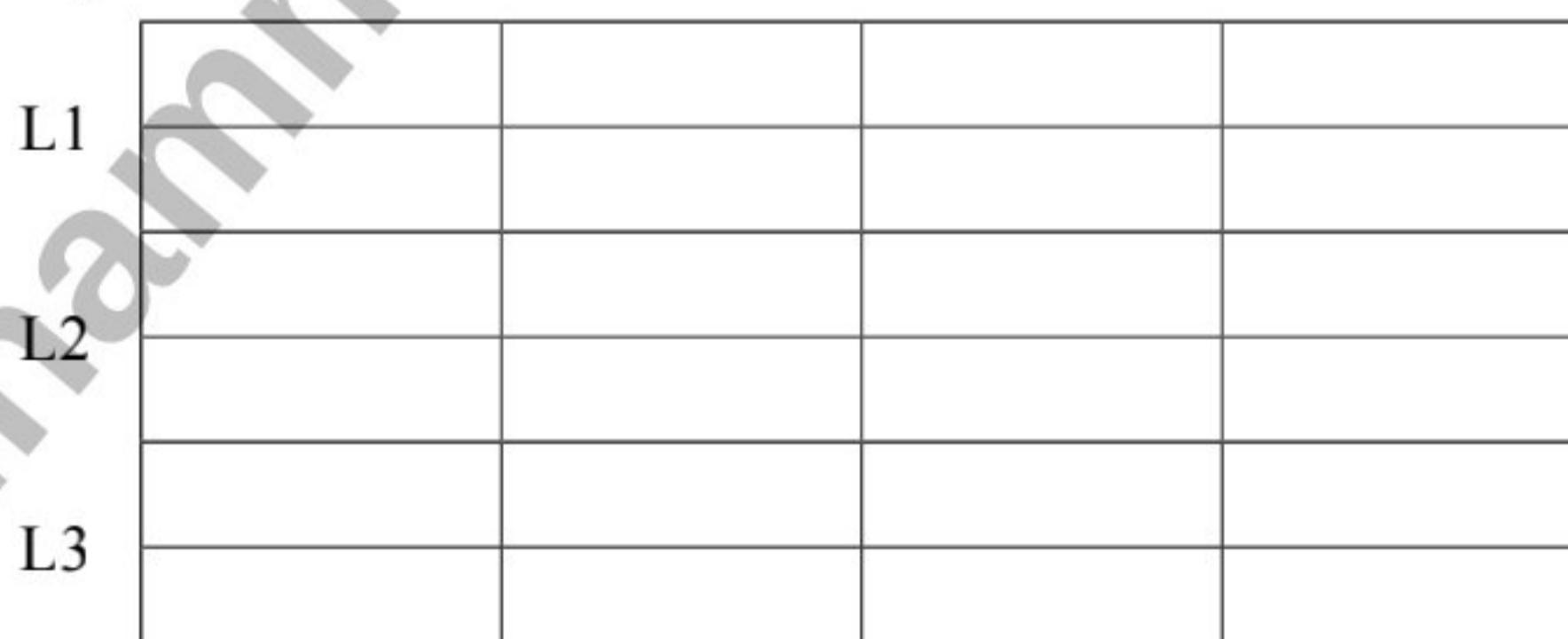
Gambar 4.6. Variasi Gerbang 4

- Fungsi Boolean :  $L_3 = \overline{L_1 L_2} + L_1 L_2 = \dots$

- Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

- Diagram waktu



- Kesimpulan :

Kombinasi gerbang pada gambar 4.6 membentuk logika dari gerbang .....  
.....

### Percobaan 5 : Merancang fungsi Boolean ke dalam rangkaian

1. Buat kombinasi gerbang logika berdasarkan Fungsi Boolean  
 $L3 = \overline{L1}L2 + L1\overline{L2}$

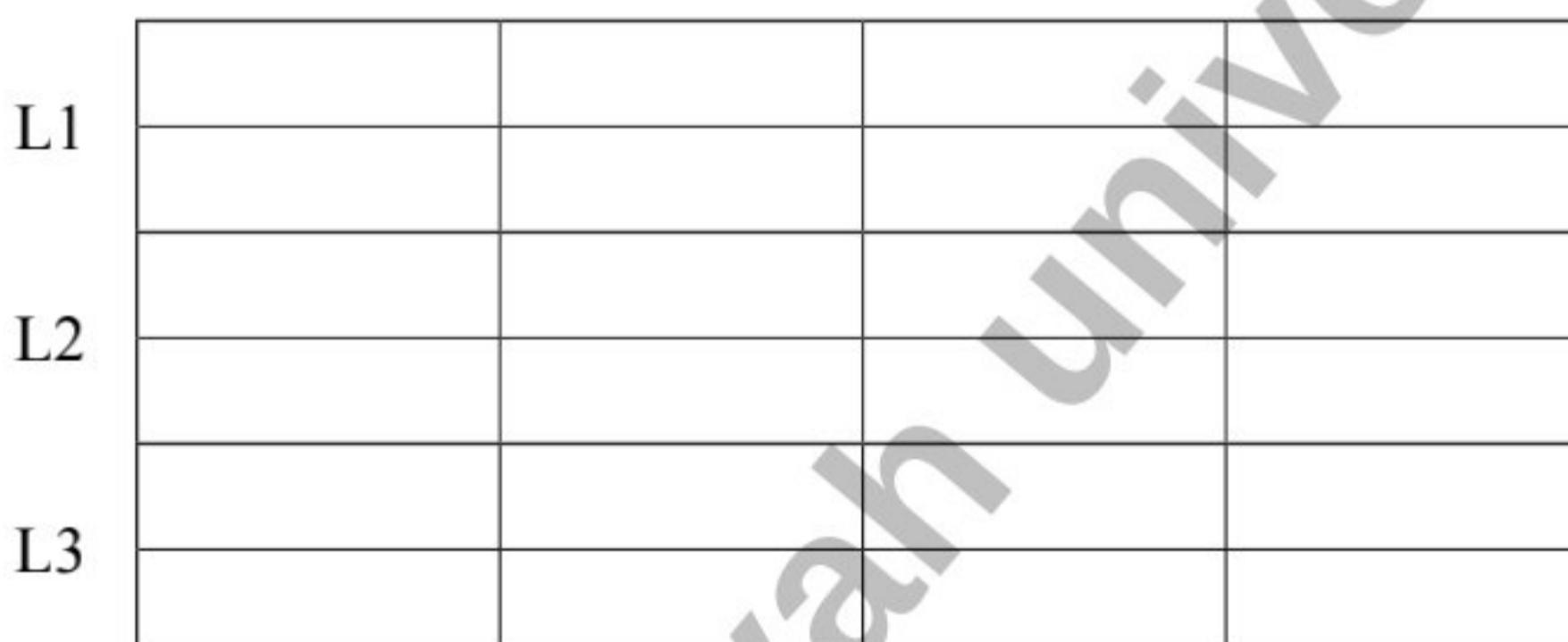
2. Gambar kombinasi gerbang logika nya!

Gambar dari fungsi  $L3 = \overline{L1}L2 + L1\overline{L2}$

3. Tabel kebenaran

SW1	SW2	L1	L2	L3
0	0			
1	0			
0	1			
1	1			

4. Diagram waktu



5. Kesimpulan :

Kombinasi gerbang akan membentuk logika dari gerbang .....

# MODUL 5

## ALJABAR BOOLEAN

### TUJUAN PRAKTIKUM

Mahasiswa mampu mengimplementasikan aljabar *boolean* dalam proyek simulasi.

### TEORI

#### Aljabar Boolean

Aljabar boolean atau *Boolean algebra* telah menjadi dasar dari perkembangan ilmu komputer dan logika digital. Dalam ilmu matematika dan logika matematika, aljabar boolean merupakan sub area dari aljabar yang mana variable-variablenya bernilai benar (*true*) atau salah (*false*), atau umumnya dinotasikan dengan 1 dan 0. Hubungan logika antar variable biner dalam aljabar boolean menggunakan tiga operasi dasar yaitu AND (*conjunction*), OR (*disjunction*), dan NOT (*negation*). Formula-formula yang digunakan untuk operasi aljabar Boolean untuk logika AND dapat dilihat dalam Tabel 5.1 dan logika OR pada Tabel 5.2. Hukum distribusi pada Tabel 5.3, hukum redudansi pada Tabel 5.4, dan hukum de morgan seperti Tabel 5.5 mempunyai tujuan membantu mengurangi kerumitan fungsi aljabar Boolean dan meningkatkan “area dan kecepatan” rangkaian digital.

Tabel 5.1. Teori dan penjelasan logika AND

Teori		Penjelasan	Contoh
1.	$A \bullet 0 = 0$	Any input “AND”ed with 0 = 0, since both inputs in an AND gate must be 1 for the result to be 1	$1 \bullet 0 = 0$ $0 \bullet 0 = 0$
2.	$A \bullet 1 = A$	Any input “AND”ed with 1 = the original input	$1 \bullet 1 = 1$ $0 \bullet 1 = 0$
3.	$A \bullet A = A$	Any input “AND”ed with itself = itself	$1 \bullet 1 = 1$ $0 \bullet 0 = 0$

4.	$A \bullet \bar{A} = 0$	Any input “AND”ed with the opposite of that input will always be zero.	$1 \bullet 0 = 0$ $0 \bullet 1 = 0$
5.	$A \bullet B = B \bullet A$	Commutative Law - the order of inputs to an AND gate is irrelevant.	
6.	$A \bullet B \bullet C = (A \bullet B) \bullet C = A \bullet (B \bullet C)$	Associative Law - the order in which successive ANDs are performed is irrelevant.	

### Contoh 1

$$\begin{aligned} Y &= (A + B) \bullet (A + B) + C \\ &= A + B + C \quad (\text{Thm #3}) \end{aligned}$$

### Contoh 2

$$\begin{aligned} Y &= A \bullet (B + C) \bullet \bar{A} \bullet \bar{B} \bullet D \\ &= A \bullet \bar{A} \bullet (B + C) \bullet \bar{B} \bullet D \quad (\text{Thm #5}) \\ &= 0 \bullet (B + C) \bullet \bar{B} \bullet D \quad (\text{Thm #4}) \\ &= 0 \quad (\text{Thm #1}) \end{aligned}$$

Tabel 5.2. Teori dan penjelasan logika OR

Teori		Penjelasan	Contoh
7.	$A + 0 = A$	Any input “OR”ed with 0 = the original input	$1 + 0 = 1$ $0 + 0 = 0$
8.	$A + 1 = 1$	Any input “OR”ed with 1 = 1	$1 + 1 = 1$ $0 + 1 = 1$
9.	$A + A = A$	Any input “OR”ed with itself = itself	$1 + 1 = 1$ $0 + 0 = 0$
10.	$A + \bar{A} = 1$	Any input “OR”ed with the opposite of that input will always be 1.	$1 + 0 = 1$ $0 + 1 = 1$
11.	$A + B = B + A$	Commutative Law - the order of inputs to an OR gate is irrelevant.	
12.	$A + B + C = (A + B) + C = A + (B + C)$	Associative Law - the order in which successive ORs are performed is irrelevant.	

**Contoh 3**

$$Y = A \bullet \bar{C} + A \bullet \bar{C} \bullet D + 1 \\ = 1 \quad (\text{Thm } \#8)$$

**Contoh 4**

$$Y = A \bullet B + \overline{A \bullet B} + C \\ = 1 + C \quad (\text{Thm } \#10) \\ = 1 \quad (\text{Thm } \#8)$$

Tabel 5.3. Hukum Distribusi

Teori		Penjelasan
13.	$A \bullet (B + C) = A \bullet B + A \bullet C$	Any AND operation can be distributed over an OR operation.
14.	$A + (B \bullet C) = (A + B) \bullet (A + C)$	Any OR operation can be distributed over an AND operation (This does not apply in normal algebra).

**Contoh 5**

$$X = A \bullet B \bullet C + A \bullet C \\ = A \bullet C \bullet (B + 1) \quad (\text{Thm } \#13) \\ = A \bullet C \quad (\text{Thm } \#8)$$

**Contoh 6**

$$X = (A + B) \bullet (A + C) \\ = A \bullet A + A \bullet C + B \bullet A + B \bullet C \quad (\text{Thm } \#13) \\ = A + A \bullet C + B \bullet A + B \bullet C \quad (\text{Thm } \#3) \\ = A \bullet (1 + C + B) + B \bullet C \quad (\text{Thm } \#13) \\ = A \bullet 1 + B \bullet C \quad (\text{Thm } \#8) \\ = A + B \bullet C \quad (\text{Thm } \#2)$$

Tabel 5.4. Hukum Redudansi

Teori	Penjelasan
15a. $A + (A \bullet B) = A$	Can be proven using previous theorems: $\begin{aligned} A + (A \bullet B) &= (A \bullet 1) + (A \bullet B) && (\text{Thm 2}) \\ &= A \bullet (1 + B) && (\text{Thm 13}) \\ &= A \bullet 1 && (\text{Thm 8}) \\ &= A && (\text{Thm 2}) \end{aligned}$
15b. $A \bullet (A + B) = A$	Can be proven using previous theorems: $\begin{aligned} A \bullet (A + B) &= (A + 0) \bullet (A + B) && (\text{Thm 7}) \\ &= A + (0 \bullet B) && (\text{Thm 14}) \\ &= A + 0 && (\text{Thm 1}) \\ &= A && (\text{Thm 7}) \end{aligned}$
16a. $A + (\bar{A} \bullet B) = A + B$	Can be proven using the previous theorems: $\begin{aligned} A + (\bar{A} \bullet B) &= (A + \bar{A}) \bullet (A + B) && (\text{Thm 14}) \\ &= 1 \bullet (A + B) && (\text{Thm 10}) \\ &= A + B && (\text{Thm 2}) \end{aligned}$
16b. $A \bullet (\bar{A} + B) = A \bullet B$	Can be proven using the previous theorems: $\begin{aligned} A \bullet (\bar{A} + B) &= (A \bullet \bar{A}) + (A \bullet B) && (\text{Thm 13}) \\ &= 0 + (A \bullet B) && (\text{Thm 4}) \\ &= A \bullet B && (\text{Thm 7}) \end{aligned}$

Tabel 5.5. Hukum Demorgan

Teori	Penjelasan
17. $\overline{(A + B)} = \bar{A} \bullet \bar{B}$	For any two inputs A, B - the result of the inverse of A <b>or</b> B is the same as the inverse of A <b>and</b> the inverse of B.
18. $\overline{(A \bullet B)} = \bar{A} + \bar{B}$	For any two inputs A, B - the result of the inverse of A <b>and</b> B is the same as the inverse of A <b>or</b> the inverse of B.

### Pengenalan Peta Karnaugh

Peta karnaugh digunakan untuk menyederhanakan fungsi boolean baik untuk suku *minterm* atau *maxterm*. Gambar 5.1 merupakan contoh dari peta Karnaugh dua variable. Gambar 5.2 merupakan contoh peta Karnaugh tiga variable. Gambar 5.3 merupakan peta Karnaugh empat variable. Yang perlu diperhatikan adalah deretan nomor pada kotak peta Karnaugh bukan diurutkan berdasarkan angka biner, melainkan berdasarkan deretan *GrayCode*. Peta Karnaugh digambarkan dengan kotak

bujur sangkar dengan tiap kotak merepresentasikan minterm (SoP). Jumlah kotak adalah dua pangkat jumlah variablenya (, dengan  $n = 0,1,2,3$ , dst). Cara menentukan bentuk sederhana dari peta Karnaugh adalah mencari variable yang memiliki nilai yang sama (tidak berubah) dari kelompok tersebut. Pemahaman tentang peta Karnaugh akan dijelaskan lebih dalam pada modul 6.

	$y$	0	1	
$x$		$x'y'$	$x'y$	
0				
1		$xy'$	$xy$	

	$y$	0	1	
$x$		$m_0$	$m_1$	
0				
1		$m_2$	$m_3$	

Gambar 5.1. Peta Karnaugh dengan dua variable

	$yz$	00	01	11	10	
$x$		$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$	
0						
1		$xy'z'$	$xy'z$	$xyz$	$xyz'$	

	$yz$	00	01	11	10	
$x$		$m_0$	$m_1$	$m_3$	$m_2$	
0						
1		$m_4$	$m_5$	$m_7$	$m_6$	

Gambar 5.2. Peta Karnaugh dengan tiga variable

	$yz$	00	01	11	10	
$wx$		$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$	
00						
01		$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$	
11		$Wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$	
10		$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$	

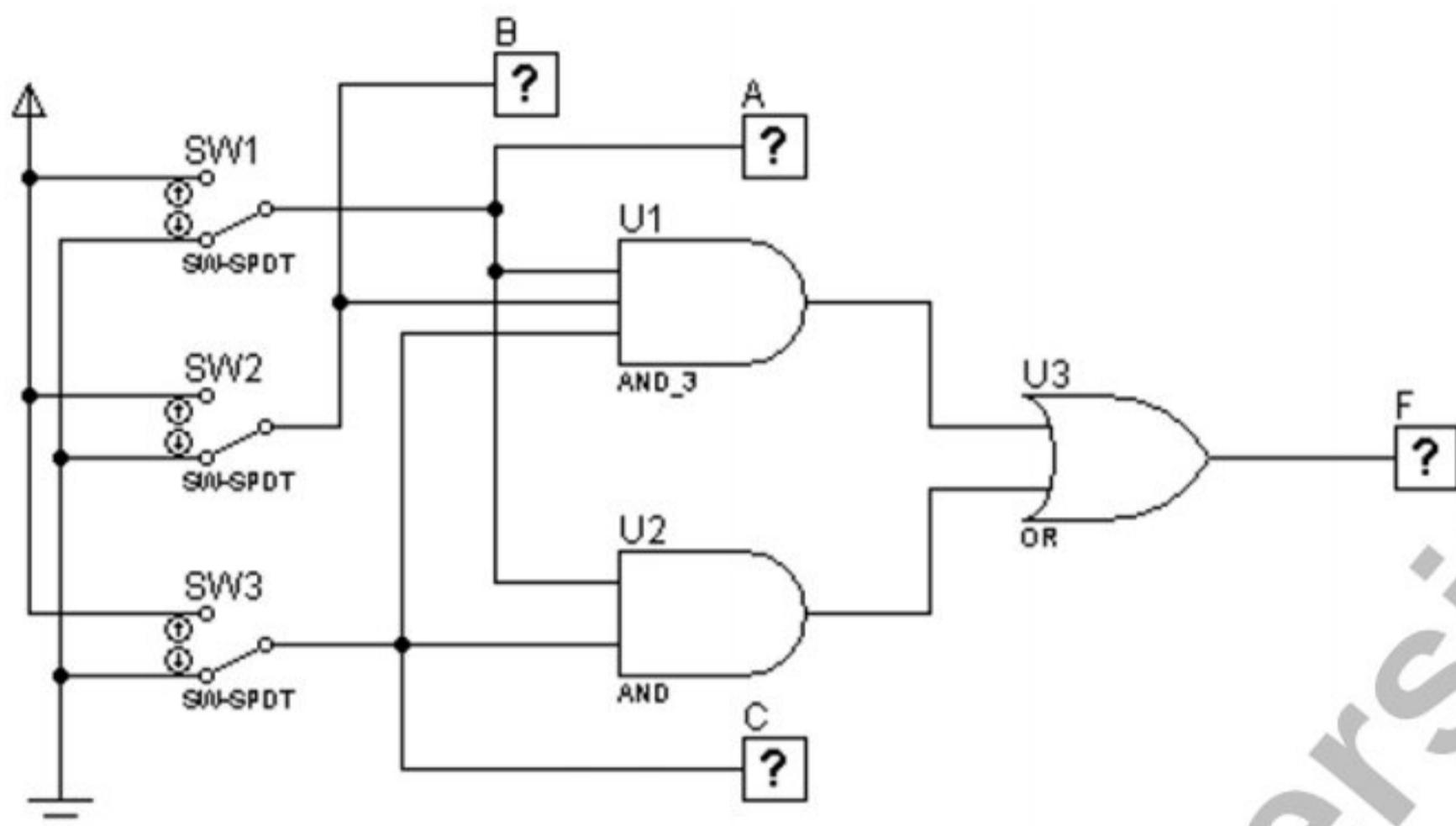
	$yz$	00	01	11	10	
$wx$		$m_0$	$m_1$	$m_3$	$m_2$	
00						
01		$m_4$	$m_5$	$m_7$	$m_6$	
11		$m_{12}$	$m_{13}$	$m_{15}$	$m_{14}$	
10		$m_8$	$m_9$	$m_{11}$	$m_{10}$	

Gambar 5.3. Peta Karnaugh dengan empat variable

## KEGIATAN PRAKTIKUM

### Percobaan 1

- Buat kombinasi gerbang logika sebagaimana pada gambar di bawah ini!



- Fungsi boolean :  $F = ABC + AC$
- Isi titik-titik dalam tabel!

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

- Isi titik-titik dalam karnaugh map

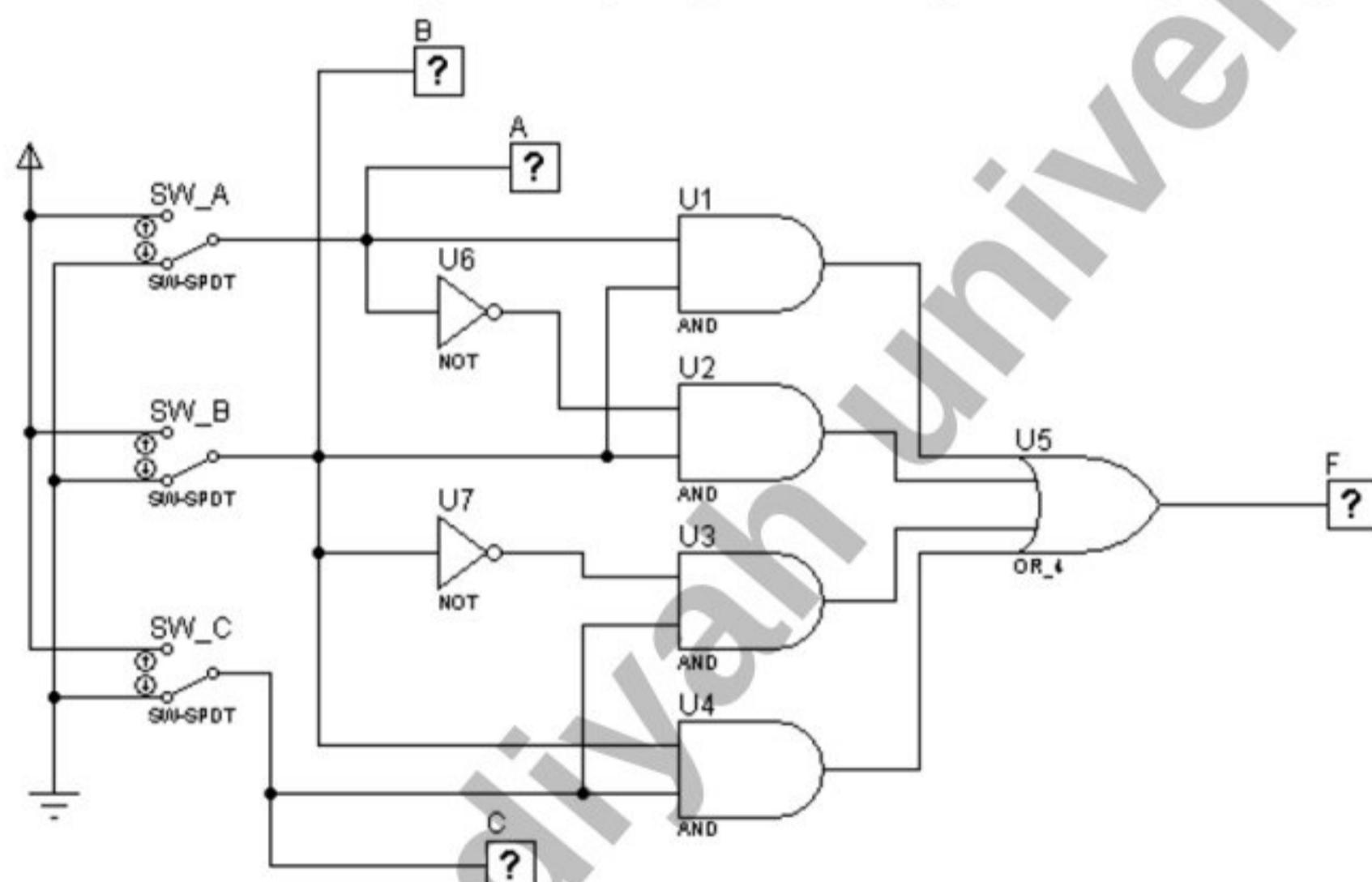
		AB			
		00	01	11	10
C	0	.....	.....	.....	.....
	1	.....	.....	.....	.....

- Sederhanakan Fungsi boolean berdasarkan karnaugh map :  
 $F = \dots$

6. Buat kombinasi gerbang logika berdasarkan fungsi boolean baru anda! Gambar gerbang logika dalam kotak dibawah!

### Percobaan 2

1. Buat kombinasi gerbang logika sebagaimana pada gambar!



2. Fungsi boolean :  $F = (A + B) \cdot (A + C)$   
 3. Isi titik-titik dalam tabel kebenaran

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

4. Isi titik-titik dalam karnaugh map

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

5. Sederhanakan Fungsi boolean berdasarkan karnaugh map :

$$F = \dots$$

6. Buat kombinasi gerbang logika berdasarkan fungsi boolean baru anda! gambar gerbang logika dalam kotak dibawah!

### Percobaan 3

- Buat kombinasi gerbang logika sebagaimana pada gambar!
- Fungsi boolean :  $F = AB + A'B + B'C + BC$
- Isi titik-titik dalam tabel kebenaran

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

4. Isi titik-titik dalam karnaugh map

		AB			
		01	11	10	
C	0	.....	.....	.....	.....
	1	.....	.....	.....	.....

5. Sederhanakan Fungsi boolean berdasarkan karnaugh map :

$$F = \dots$$

6. Buat kombinasi gerbang logika baru! Gambar dalam kotak dibawah ini!



©Muhammadiyah University Press

# MODUL 6

## PETA KARNAUGH

### TUJUAN PRAKTIKUM

Mahasiswa mampu menyederhanakan fungsi Boolean menggunakan peta Karnaugh.

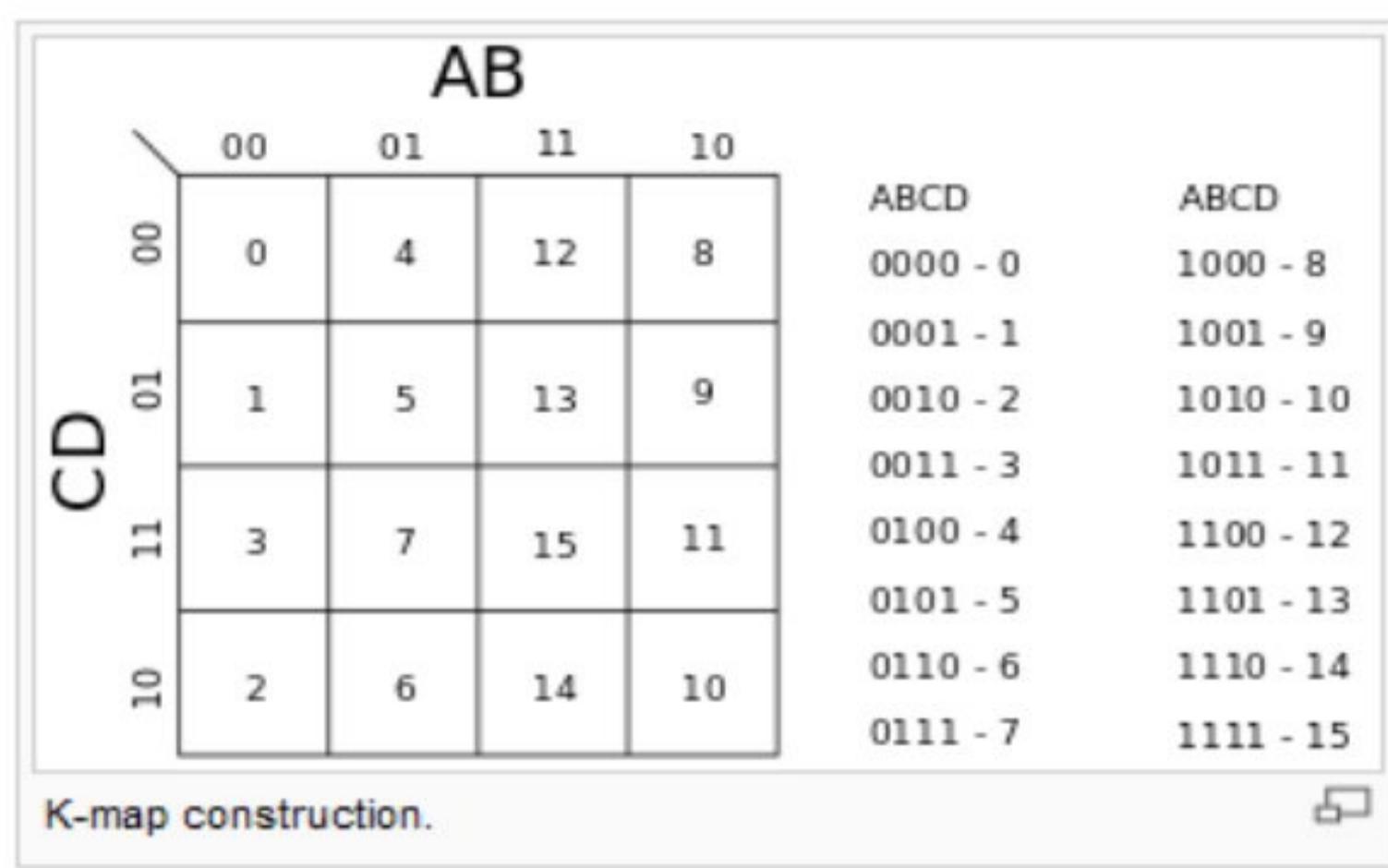
### TEORI

#### Peta Karnaugh

Peta Karnaugh atau dikenal juga sebagai K-map merupakan sebuah metode untuk menyederhanakan ekspresi dari aljabar boolean. Peta Karnaugh dikenalkan oleh Maurice Karnaugh tahun 1953 sebagai penyempurnaan dari diagram Veitch milik Edward Veitch tahun 1952. Peta Karnaugh mengurangi kebutuhan kalkulasi yang kompleks dengan mengambil keuntungan dari kemampuan manusia dalam mengenali pola.

Hasil boolean yang dibutuhkan ditransfer dari tabel kebenaran ke dalam grid dua dimensi kemudian diolah pada pengkodean Gray seperti Gambar 6.1. Selanjutnya setiap posisi sel merepresentasikan satu kombinasi dari kondisi masukan, dan sementara setiap nilai sel mewakili nilai keluaran yang sesuai. Kelompok optimal 1 atau 0 yang telah diidentifikasi merupakan syarat dari bentuk standar (canonical) logika dari tabel kebenaran.

Peta Karnaugh digunakan untuk menyederhanakan kebutuhan logika dalam dunia sehari-hari sehingga dapat diimplementasikan menggunakan angka minimal dari gerbang logika fisik. Ekspresi jumlah hasil (sum-of-product) umumnya diimplementasikan menggunakan gerbang AND yang memasuki gerbang OR. Dan ekspresi hasil jumlah (product-of-sums) diimplementasikan dengan gerbang OR yang memasuki gerbang AND. Peta Karnaugh juga dapat digunakan untuk menyederhanakan ekspresi logika dalam desain software. Kondisi Boolean yang umumnya digunakan dalam pernyataan bersyarat, dapat menjadi sangat rumit sehingga membuat kode sulit untuk dibaca.



Gambar 6.1. Konstruksi peta Karnaugh

### Contoh 1. Boolean function

$$F(A,B,C,D) = E(6,8,9,10,11,12,13,14)$$

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

Langkah Penyelesaian

Kelompok 1 :  $F = \underline{AC'}$

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

**Kelompok 2 :  $F = AC' + \underline{AB'}$**

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

**Kelompok 3 :  $F = AC' + AB' + \underline{AD'}$**

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

**Kelompok 4 :  $F = AC' + AB' + AD' + \underline{BCD'}$**

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

$$F(A,B,C,D) = E(6,8,9,10,11,12,13,14)$$

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

Hasil penyederhanaannya :

$$F = AC' + AB' + AD' + BCD'$$

AD' mempunyai fungsi untuk menghindari "race Hazard" pada implementasi interface

Atau menggunakan kombinasi yang lain yaitu POS (OR-AND) menghasilkan bentuk sederhana  $F = (A+B)(A+C)(B'+C'+D')(A+D')$

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

Fungsi inverse diselesaikan dengan cara yang sama yaitu melalui pengelompokan nilai 0, didapatkan bentuk sederhana  $F' = A'B' + A'C' + A'D + BCD$

## Contoh 2. Don't Care

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

Penambahan dari “Don’t care (X)” dapat dengan mudah diselesaikan

$$F = A + BCD'$$

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

Inverse

$$F' = A'B' + A'C' + A'D$$

## Contoh 3. 2 x 2 Peta Karnaugh

		A	
		0	1
B	0	1	1
	1	1	0

$$F(A,B) = E(1,2,3)$$

Fungsi Boolean:

$$F = A' + B'$$

$$F' = AB$$

## Contoh 4. 2 x 4 Peta Karnaugh

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

$$F(A,B,C) = E(3,4,5,6,7,8)$$

Fungsi Boolean:

$$F = C + A$$

$$F' = A'C'$$

## KEGIATAN PRAKTIKUM

### Percobaan 1

- Buat kombinasi gerbang logika berdasarkan peta karnaugh berikut.

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

2. Fungsi boolean :  $F = \dots$
3. Buat gerbang logika berdasarkan fungsi boolean anda! Gambar dalam kotak dibawah ini!

### Percobaan 2

1. Buat kombinasi gerbang logika berdasarkan peta karnaugh berikut!

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

2. Fungsi boolean :  
 $F = \dots$  (AND-OR)  
 $F = \dots$  (OR-AND)
3. Buat gerbang logika berdasarkan fungsi boolean anda! Gambar dalam kotak dibawah ini!



Apakah kedua kombinasi memberikan hasil yang sama? Ya / tidak

### Percobaan 3

1. Fungsi boolean :  $F = XYZ + XYZ' + XY'Z + X'YZ + X'YZ' + XY'Z' + X'Y'Z'$ .
2. Berdasarkan fungsi boolean, isi titik-titik dalam peta karnaugh berikut!

		XY			
		00	01	11	10
Z	0	....	....	....	....
	1	....	....	....	....

3. Sederhanakan fungsi boolean :  $F = \dots$
4. Buat gerbang logika berdasarkan fungsi boolean anda! Gambar dalam kotak dibawah ini!



#### Percobaan 4

1. Fungsi boolean :  $F = AD' + ABC + ABC' + BCD + BC'D' + AB'CD'$ .
2. Berdasarkan fungsi boolean, isi titik-titik dalam peta karnaugh berikut!

		AB			
		00	01	11	10
CD	00	....	....	....	....
	01	....	....	....	....
	11	....	....	....	....
	10	....	....	....	....

3. Sederhanakan fungsi boolean :  $F = \dots$
4. Buat gerbang logika berdasarkan fungsi boolean anda! Gambar dalam kotak dibawah ini!

#### Percobaan 5

1. Tabel Fungsi boolean :

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

A	B	C	D	F
1	1	1	0	0
0	0	0	1	<b>1</b>
1	0	0	1	<b>1</b>
0	1	0	1	0
1	1	0	1	<b>1</b>
0	0	1	1	<b>1</b>
1	0	1	1	0
0	1	1	1	<b>1</b>
1	1	1	1	0

2. Berdasarkan tabel di atas, Isi titik-titik dalam karnaugh map berikut!

		AB			
		01	11	10	
CD	00	....	....	....	....
	01	....	....	....	....
	11	....	....	....	....
	10	....	....	....	....

3. Fungsi Boolean sederhana:

$$F = \dots$$

4. Buat gerbang logika berdasarkan fungsi boolean anda! Gambar dalam kotak dibawah ini!

©Muhammadiyah University Press

# MODUL 7

## FLIP – FLOP DASAR

### TUJUAN PRAKTIKUM

Mahasiswa mampu membuat flip-flop dasar dan memahami cara kerjanya.

### TEORI

Dalam dunia elektronik, sebuah flip-flop (FF) atau dikenal dengan pengancing (latch) merupakan sebuah rangkaian yang mempunyai dua kondisi stabil dan dapat digunakan untuk menyimpan informasi. Sebuah flip-flop adalah multivibrator dengan kondisi ganda. Rangkaianya dapat dibuat untuk mengubah kondisi melalui sinyal-sinyal yang diaplikasikan ke satu atau lebih kontrol input dan akan mempunyai satu atau dua output. Inilah elemen penyimpanan dasar dalam logika sekuensial. Flip-flop atau latch merupakan blok rangkaian dasar dari sistem elektronik digital yang telah digunakan dalam komputer, sistem komunikasi, dan berbagai sistem lain.

Flip flop atau latch umumnya digunakan sebagai elemen penyimpan data. Penyimpanan data tersebut dapat digunakan untuk penyimpanan memori dan sirkuitnya dijelaskan melalui logika sekuensial. Ketika digunakan dalam mesin terbatas, output flip-flop dan keadaan selanjutnya akan tergantung pada masukan dan keadaan saat itu juga (current input and current state). Flip-flop juga dapat digunakan untuk menghitung pulsa dan untuk sinkronisasi sinyal masukan variable waktu. Berdasarkan rangkaianya, terdapat beberapa jenis flip-flop diantaranya adalah flip-flop (FF), RS-FF, D-FF, JK-FF, dan T-FF. Pembahasan dan percobaan materi akan dibagi menjadi dua modul, yaitu modul 7 dan modul 8.

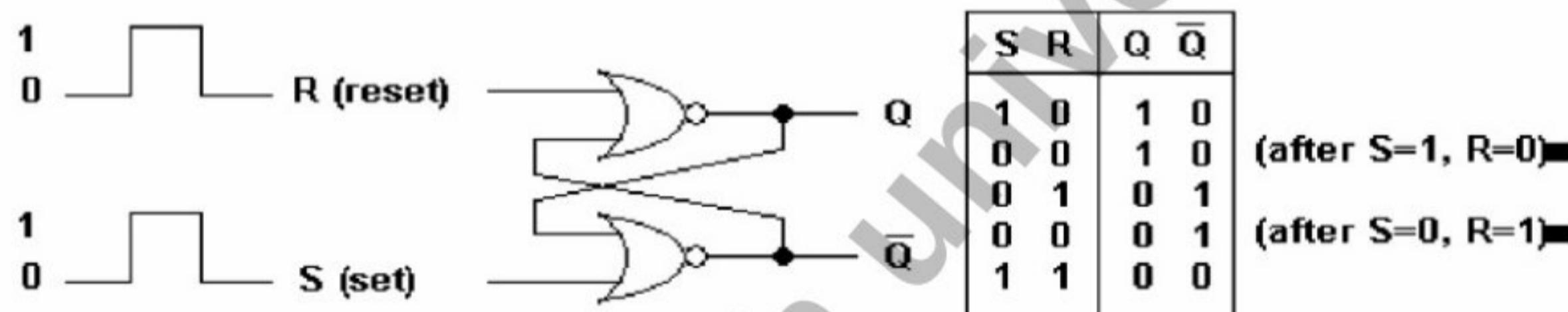
### RS Flip-Flop (RS-FF)

Flip-flop atau latch terdiri dari dua gerbang NAND atau NOR seperti diperlihatkan Gambar 7.1 dan Gambar 7.2. Ketika menggunakan gerbang statis sebagai blok rangkaian, latch dasar adalah latch S-R dasar yang mana S dan R ditujukan untuk set dan reset. Hal ini dapat dibangun dari

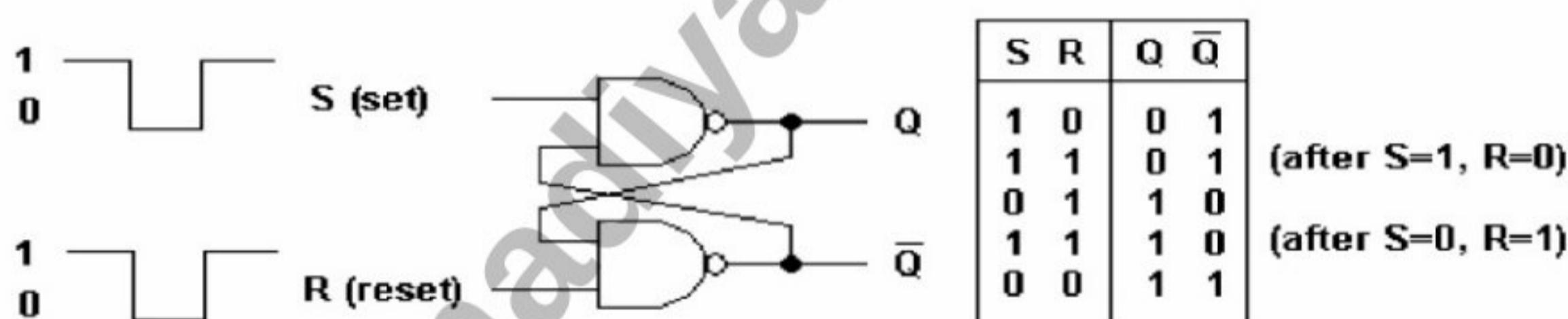
penggabungan dua gerbang logika NOR yang saling disilangkan. Bit yang tersimpan berada pada output yang ditandai dengan Q.

Disaat input S dan R dalam keadaan low, umpan balik untuk dan berada pada keadaan konstan dengan adalah komplemen dari . Jika S (set) dalam keadaan high, sementara R (reset) dalam keadaan low, sehingga output akan menjadi high dan akan tetap high ketika S kembali bernilai rendah. Hal yang sama juga berlaku ketika R bernilai high sementara S bernilai low, maka output Q akan dipaksa rendah dan tetap rendah apabila R kembali ke posisi low.

Kombinasi  $R = S = 1$  disebut kombinasi terbatas atau keadaan dilarang karena baik gerbang nol dan output yang sama-sama bernilai nol, dapat mematahkan persamaan logika  $Q = \text{not } S$ . Kombinasi ini juga tidak sesuai dalam rangkaian yang kedua input bernilai rendah secara bersamaan. Output akan terkunci pada nilai 1 dan 0 tergantung pada hubungan waktu propagasi antar gerbang logika.



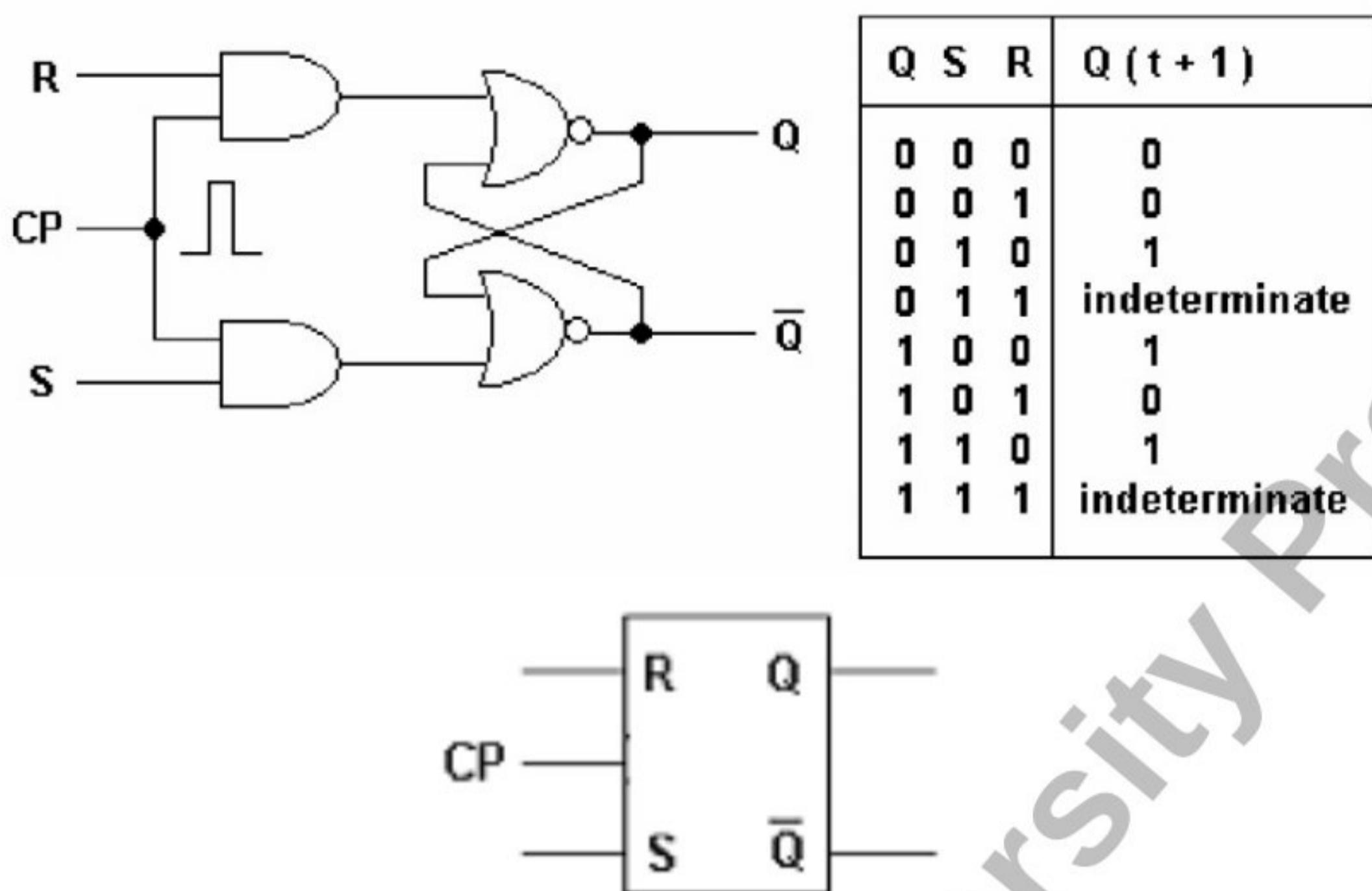
Gambar 7.1.NOR Latch



Gambar 7.2. NAND Latch

### RS - Flip flop dengan Clock

Dengan menambahkan beberapa gerbang pada dasar rangkain input, flip-flop hanya merespon selama ada pulsa clock. Output dari flip-flop tidak akan berubah selama pulsa clock bernilai 0 meskipun ada perubahan dalam inputnya. Output flip-flop hanya akan berubah dengan inputan saat pulsa clock bernilai 1.



Gambar 7.3. RS flip-flop dengan clock

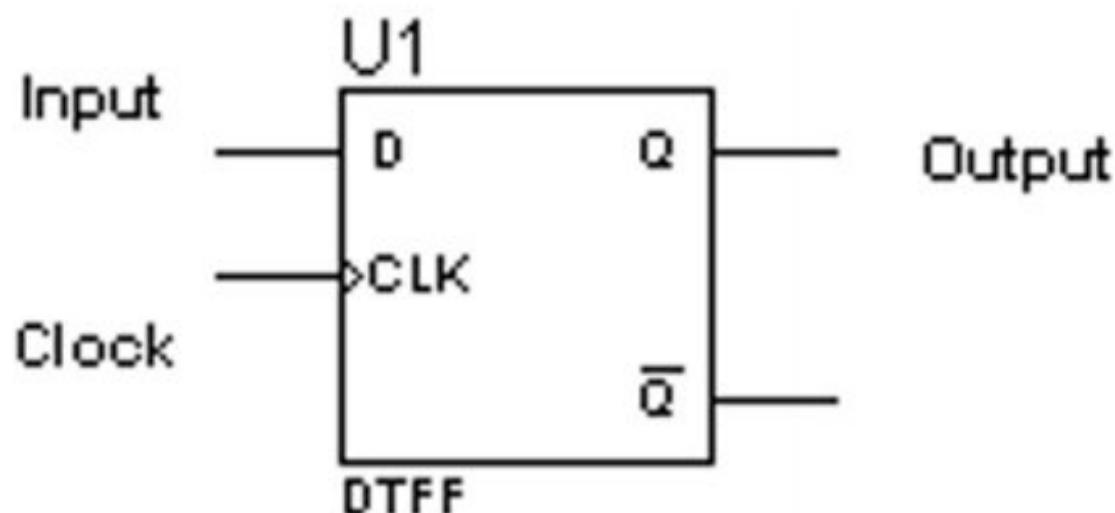
Flip-flop tipe SR atau RS merupakan jenis flip-flop yang memiliki masukan asinkron S (set) atau R (reset) atau keduanya, dengan output Q dan . Input S dan R dapat disinkronkan dengan menambahkan input sebuah clock pada rangkaiannya seperti ditunjukkan pada Gambar 7.3 diatas. Output Q tidak dapat merespon input S dan R sebelum input clock. Konsep RS-FF yang perlu diingat adalah sebagai berikut :

- Apabila R dan S bernilai rendah (low) artinya output Q tetap dalam keadaan terakhir tanpa batas karena tindakan latch internal.
- Input S bernilai high akan menyebabkan Q bernilai 1, kecuali jika output telah berada dalam keadaan high. Dalam hal ini output tidak berubah, bahkan apabila input S kembali ke kondisi rendah.
- Input R bernilai high dapat menyebabkan Q bernilai 0, kecuali jika output telah berada dalam keadaan low. Output Q selanjutnya tetap dalam kondisi rendah, walaupun masukan R berubah ke keadaan low.
- Input R dan S tidak diperbolehkan dalam kondisi yang bernilai high di waktu yang sama.

### D Flip-Flop (D-FF)

Simbol logika untuk D flip-flop ditunjukkan pada Gambar 7.4. D flip-flop hanya memiliki satu input data dan satu input clock. ‘D’ disini disebut sebagai delay atau penundaan flip-flop. ‘Delay’ menjelaskan apa

yang terjadi pada data yang tersimpan, atau pada input D. Data (0 atau 1) tertunda sebesar 1 pulsa clock dari input ke output Q. Tabel 1 merupakan tabel kebenaran untuk D flip-flop yang telah disederhanakan.

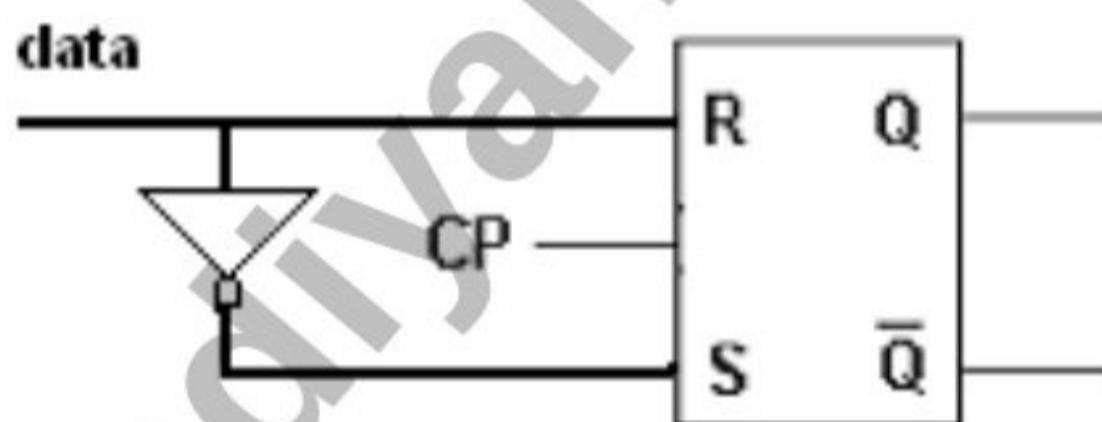


Gambar 7.4. Simbol D Flip-flop

Tabel 1. Tabel Kebenaran D Flip-flop

Input	Output
D	$Q^{n+1}$
0	0
1	1

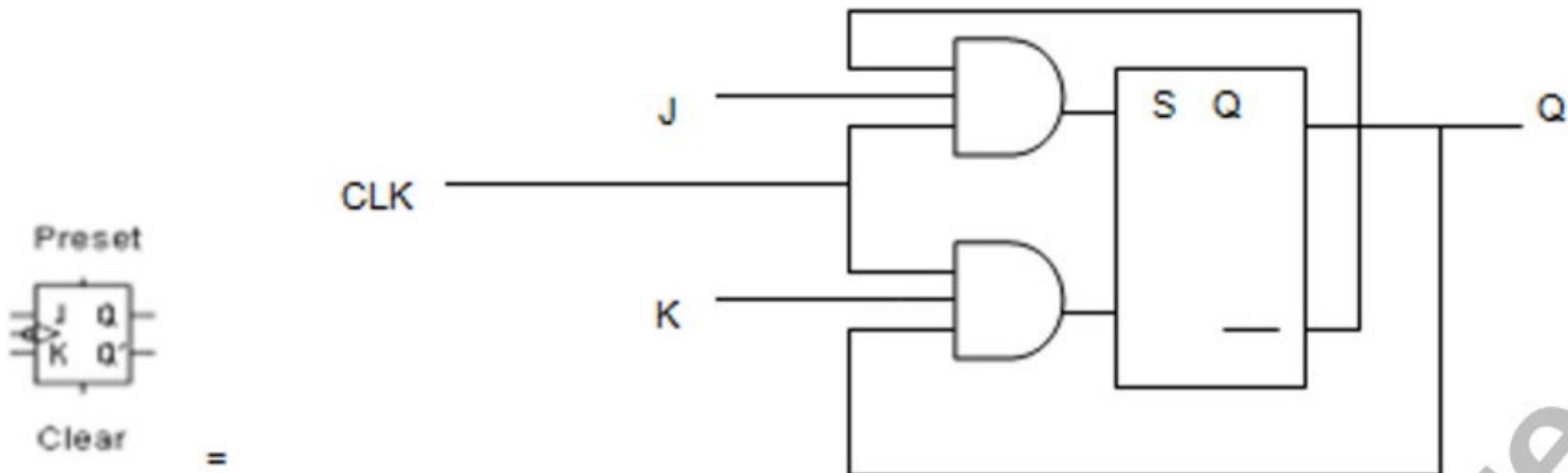
Terdapat banyak cara untuk mendesain rangkaian D flip flop ini. Pada dasarnya, D flip-flop adalah multivibrator dengan keadaan ganda (bistable) yang input D ditransfer ke output setelah menerima pulsa clock. Gambar 7.5 menjelaskan rangkaian D flip-flop.



Gambar 7.5. Rangkaian D flip-flop

### JK Flip-flop (JK-FF)

JK flip-flop merupakan jenis flip-flop yang sering digunakan, memiliki sifat flip-flop dari semua jenis flip-flop. Simbol logika dari JK flip-flop dan rangkaian JK flip-flop juga ditunjukkan pada Gambar 7.6. J dan K disebut sebagai input pengendali untuk menentukan apa yang akan dilakukan oleh flip-flop ketika menerima pulsa clock yang meningkat. Rangkaian RC mempunyai konstanta waktu yang singkat sehingga mengubah pulsa clock menjadi impuls yang sempit. Ketika J dan K bernilai 0, output Q tetap pada nilai atau keadaan terakhir.



Gambar 7.6.Simbol dan rangkaian JK flip-flop

Saat J bernilai low (0) dan K bernilai high (1), gerbang akan tertutup dan memungkinkan untuk mengatur flip-flop. Pada waktu Q lebih tinggi, gerbang yang lebih rendah akan segera memicu reset dan melewatkkan penerimaan pulsa clock positif yang datang selanjutnya. Hal ini mendorong Q menjadi low. Sehingga untuk  $J = 0$  dan  $K = 1$  mengandung arti bahwa meningkatnya pulsa clock dapat me-reset flip-flop. Selain itu, apabila J bernilai high dan K bernilai low dapat menyebabkan gerbang yang lebih rendah tertutup. Dan apabila J dan K, keduanya bernilai high, maka dapat mengatur flip flop atau me-reset-nya. Tabel 2 merupakan tabel kebenaran dari JK flip-flop.

Tabel 2. Tabel kebenaran JK flip-flop

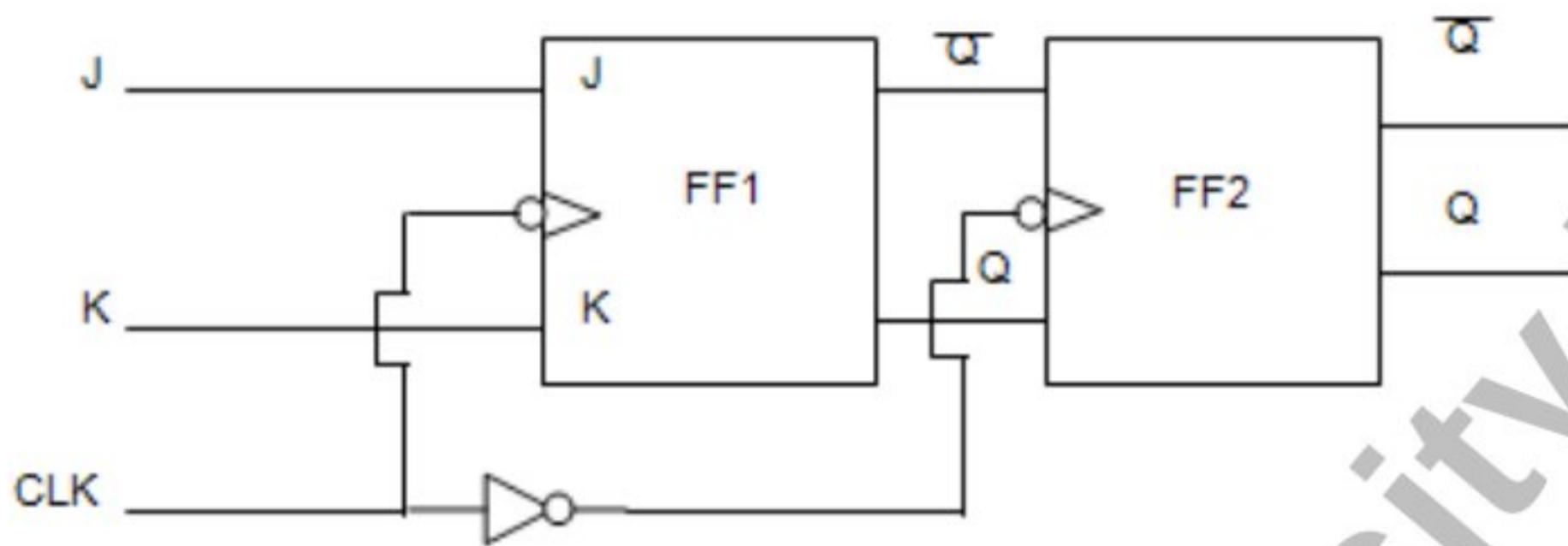
CLK	J	K	Q
0	0	0	Keadaan terakhir
↑	0	1	0
↑	1	0	1
↑	1	1	Keadaan terakhir

### JK Flip-flop Master/Slave

Gambar 7.7 memperlihatkan salah satu cara untuk membangun rangkaian JK flip-flop master/slave. Cara kerja dari flip flop jenis ini adalah : (1) flip-flop master dipicu oleh sisi naik dan flip-flop dipicu oleh sisi turun. Sehingga master akan merespon input J dan K sebelum slave. Apabila  $J=1$  dan  $K=0$ , flip-flop diatur pada sisi naik pulsa clock. Output Q high pada master menjadi input slave, sehingga ketika sisi turun clock tiba, slave dapat diatur dan sesuai dengan kinerja master.

Apabila  $J=0$  dan  $K=1$ , master direset melalui sisi naik pulsa clock. Output Q high pada master akan dilanjutkan ke slave. Oleh karena itu, kehadiran sisi turun akan membuat slave ter-reset. Hal ini juga sesuai

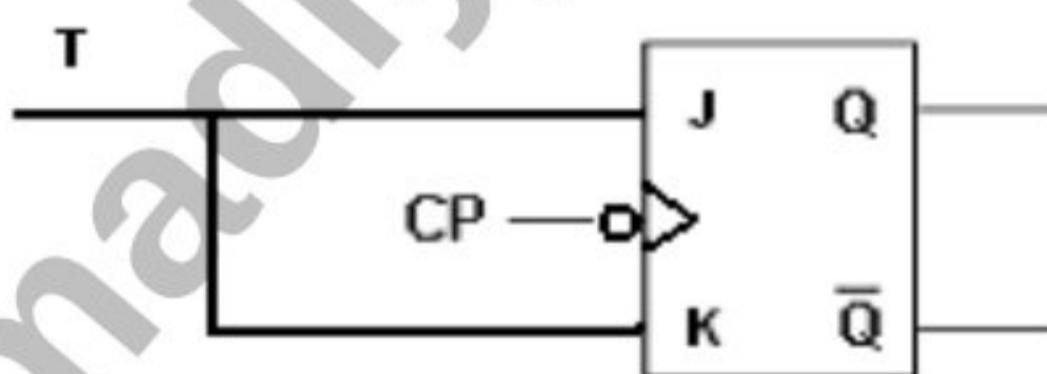
dengan kinerja dari master. Apabila input J dan K pada master bernilai high, flip-flop akan berubah ketika menemui sisi naik clock dan slave akan berubah pada pulsa sisi turun. Sehingga apapun yang dilakukan oleh master, akan dilakukan juga oleh slave. Apabila master di-set, maka slave juga akan di-set. Begitu juga ketika master di-reset maka slave pun akan di-reset.



Gambar 7.7. Rangkaian JK flip-flop master/slave

### T Flip-Flop (T-FF)

T flip-flop atau togel flip-flop adalah flip-flop yang input J dan K digabungkan menjadi satu sehingga hanya ada satu pintu masuk. Karakteristik flip-flop ini adalah kondisi dari output akan selalu berubah atau selalu berlawanan dengan kondisi sebelumnya ketika diberi input bernilai 1. Sementara kondisi keluaran akan sama atau tetap dengan kondisi keluaran sebelumnya saat diberi input bernilai 0. Gambar 7.8 merupakan simbol T flip-flop. Sebagai pembanding, Tabel 3 merupakan tabel kebenaran untuk semua flip flop.



Gambar 7.8. Simbol T flip-flop

Table 3. table kebenaran semua jenis flip flop

S-R Flip Flop				D Flip Flop		
Q (t)	Q (t+1)	S	R	Q (t)	Q (t+1)	D
0	0	0	X	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	0
1	1	X	0	1	1	1

J-K Flip Flop			
Q (t)	Q (t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

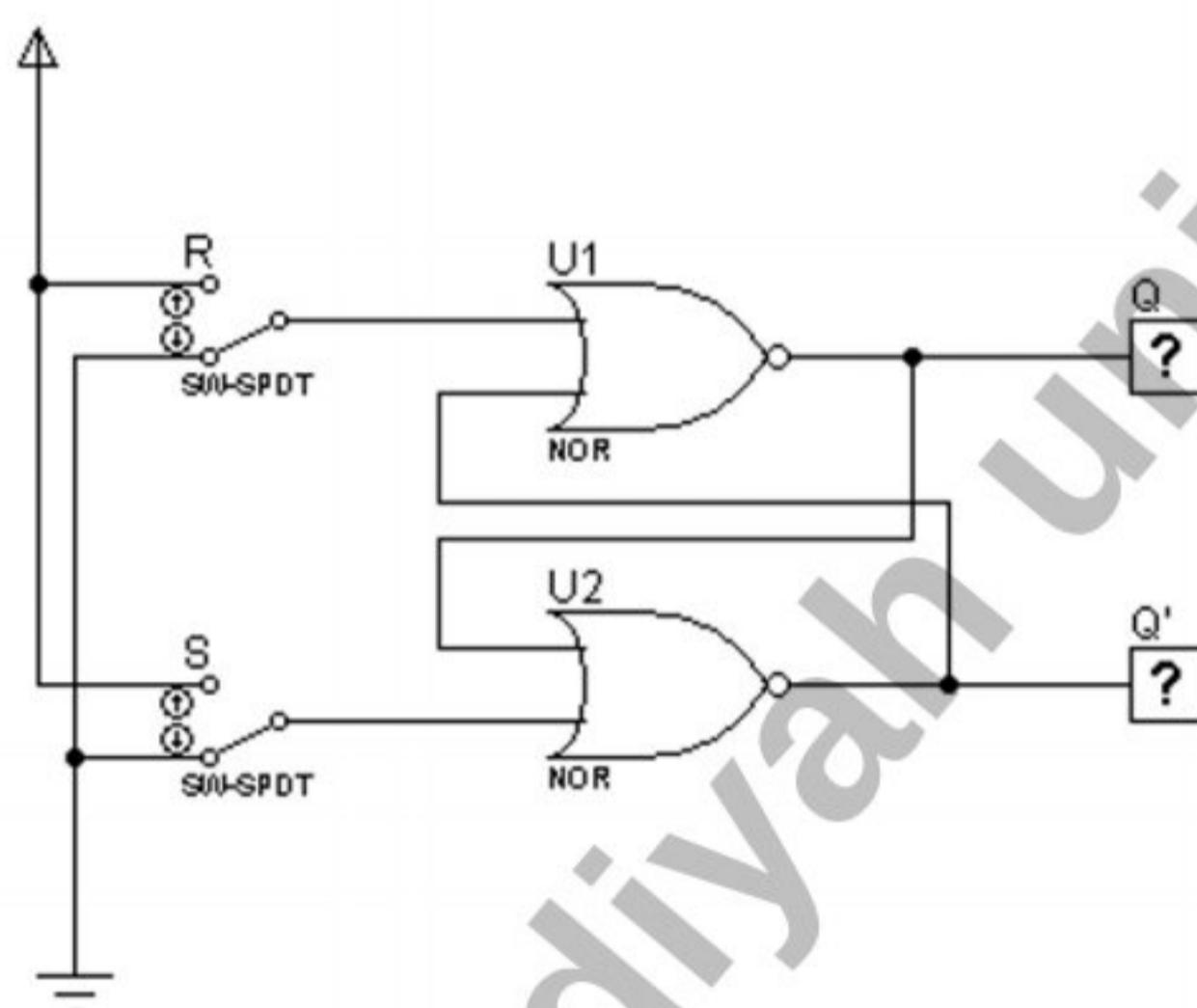
T Flip Flop		
Q (t)	Q (t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Keterangan : Nilai X dapat diisi dengan nilai 0 atau 1

## KEGIATAN PRAKTIKUM

### Percobaan 1. NOR Latch

- Buat dan simulasikan NOR latch seperti pada gambar! Dan kemudian klik switch SW-SPDT untuk mengoperasikan Latch.



- Berdasarkan simulasi anda, isi titik-titik pada tabel berikut!

	S (Set)	R (Reset)	Output		Mode
			Q	Q'	
1	0	1			
2	0	0			
3	1	0			
4	0	0			
5	1	1			

Catatan : Kolom mode menjelaskan kondisi kinerja dari FF tersebut, seperti 'memori', 'set', 'reset', atau 'togle'.

3. Jawab pertanyaan berikut!

- a. Apa yang akan terjadi jika kita berikan kondisi  $S = R = 0$ ?

.....

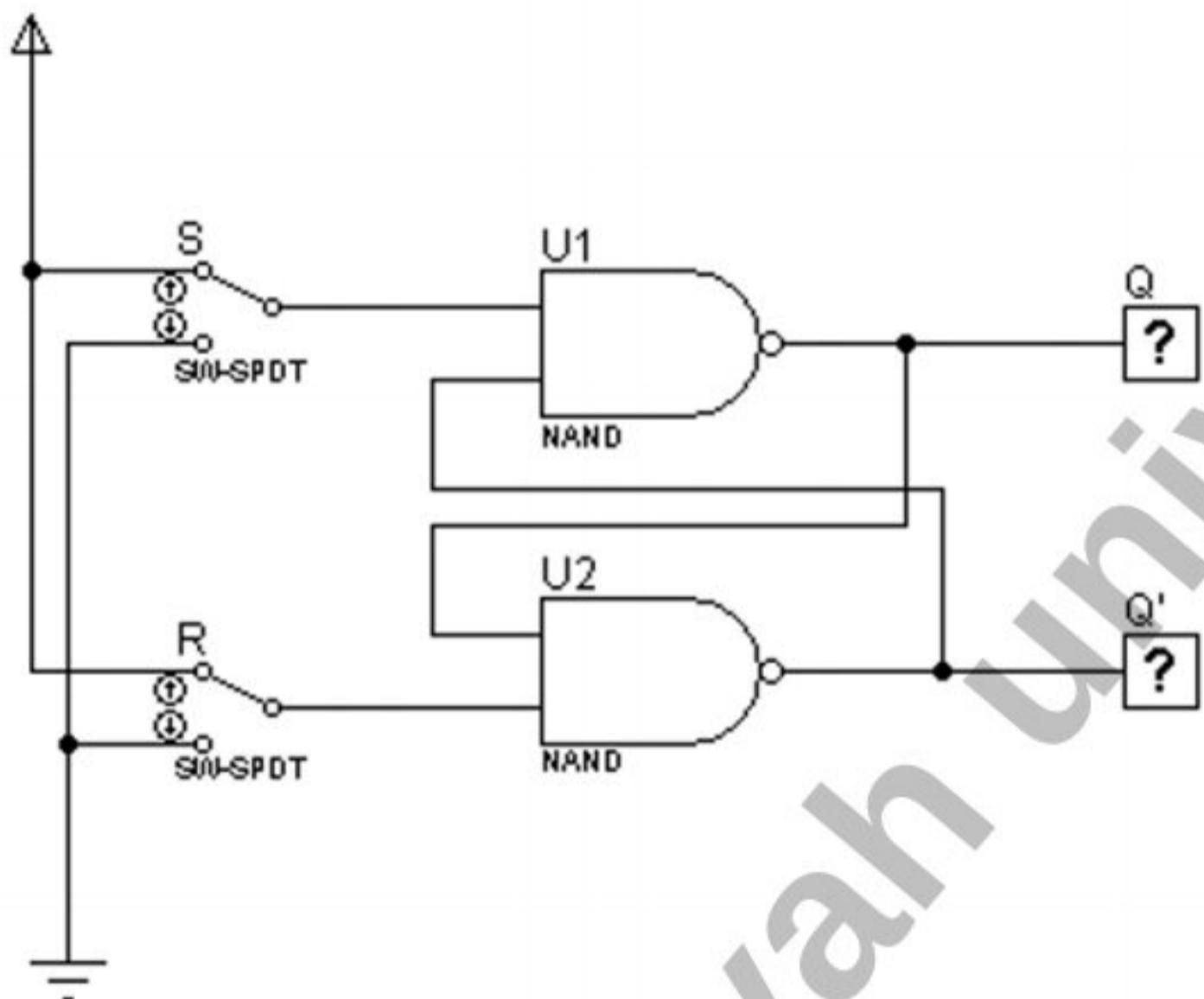
.....

- b. Kenapa kondisi  $S = R = 1$  tidak diperbolehkan?

.....

### Percobaan 2. NAND Latch

1. Buat dan simulasi NOR latch seperti pada gambar! Dan kemudian klik switch SW-SPDT untuk mengoperasikan Latch.



2. Berdasarkan simulasi anda, isi titik-titik pada tabel berikut!

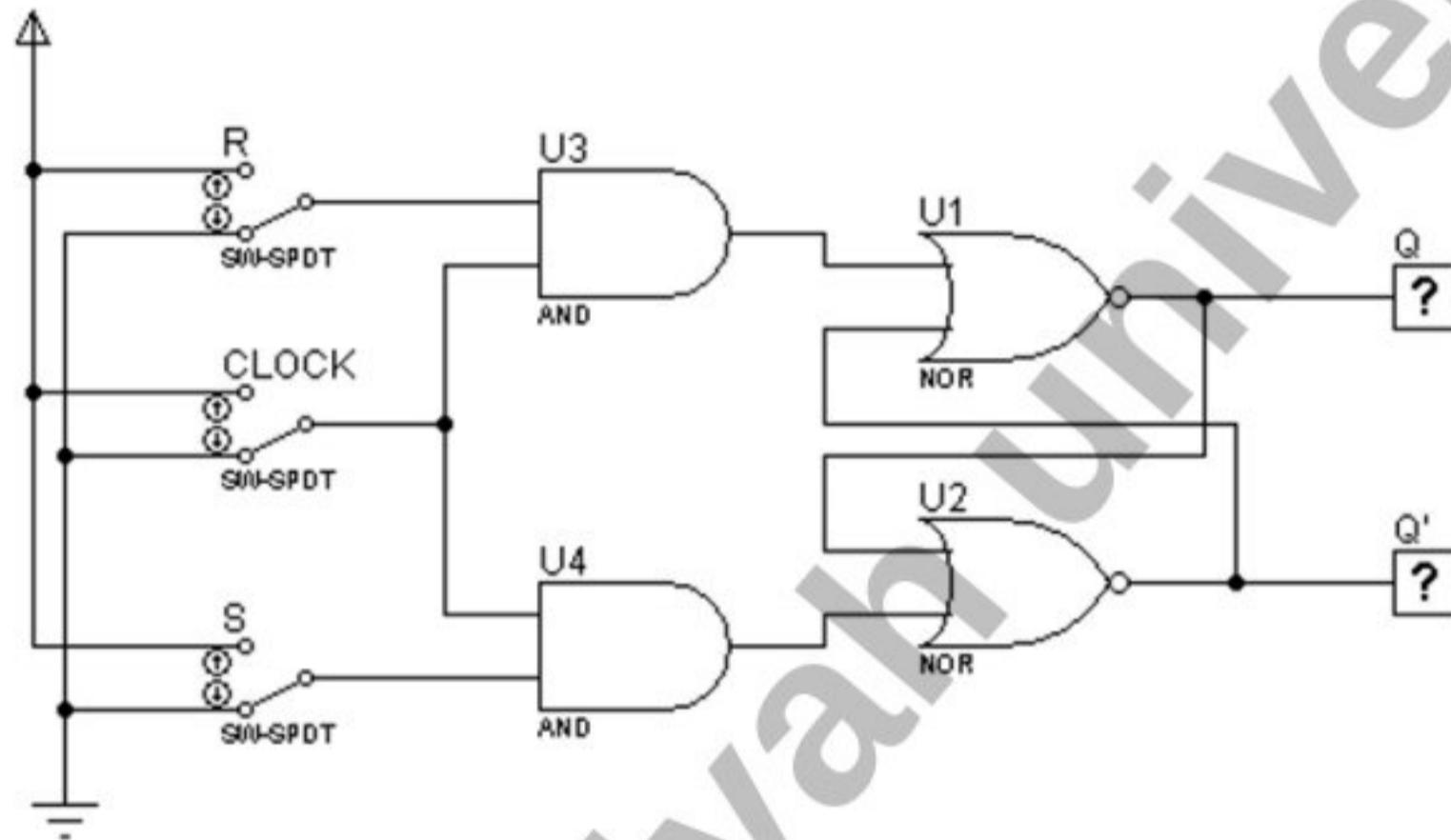
	S (Set)	R (Reset)	Output		Mode
			Q	Q'	
1	0	1			
2	1	1			
3	1	0			
4	1	1			
5	0	0			

Catatan : Kolom mode menjelaskan kondisi kinerja dari FF tersebut, seperti 'memori', 'set', 'reset', atau 'togle'

3. Jawab pertanyaan berikut!
- Apa yang akan terjadi jika kita berikan kondisi  $S = R = 1$ ?  
.....
  - Kenapa kondisi  $S = R = 0$  tidak diperbolehkan?  
.....
4. Berdasarkan analisis rangkaian flip-flop di atas, apa opini/pendapat anda tentang pernyataan “Flip-flop dan latch digunakan sebagai elemen penyimpanan data” :
- .....

### Percobaan 3. Flip-Flop RS

1. Buat dan simulasikan Flip-Flop RS seperti pada gambar! Dan kemudian klik switch SW-SPDT untuk mengoperasikan Flip-Flop RS.



2. Berdasarkan simulasi anda, isi titik-titik pada tabel berikut!

	S (Set)	R (Reset)	CLOCK	Output		Mode
				Q	$Q_{(t+1)}$	
1	0	0	0			
2	0	0	1			
3	0	1	0			
4	0	1	1			
5	1	0	0			
6	1	0	1			
7	1	1	0			
8	1	1	1			

Catatan : Kolom mode menjelaskan kondisi kinerja dari FF tersebut, seperti ‘memori’, ‘set’, ‘reset’, atau ‘togle’

3. Jawab pertanyaan berikut!

- a. Apa yang akan terjadi jika kita beri kondisi  $S = R = 1$  dan clock berubah dari 1 ke 0?

.....

- b. Bagaimana kondisi diatas dapat terjadi?

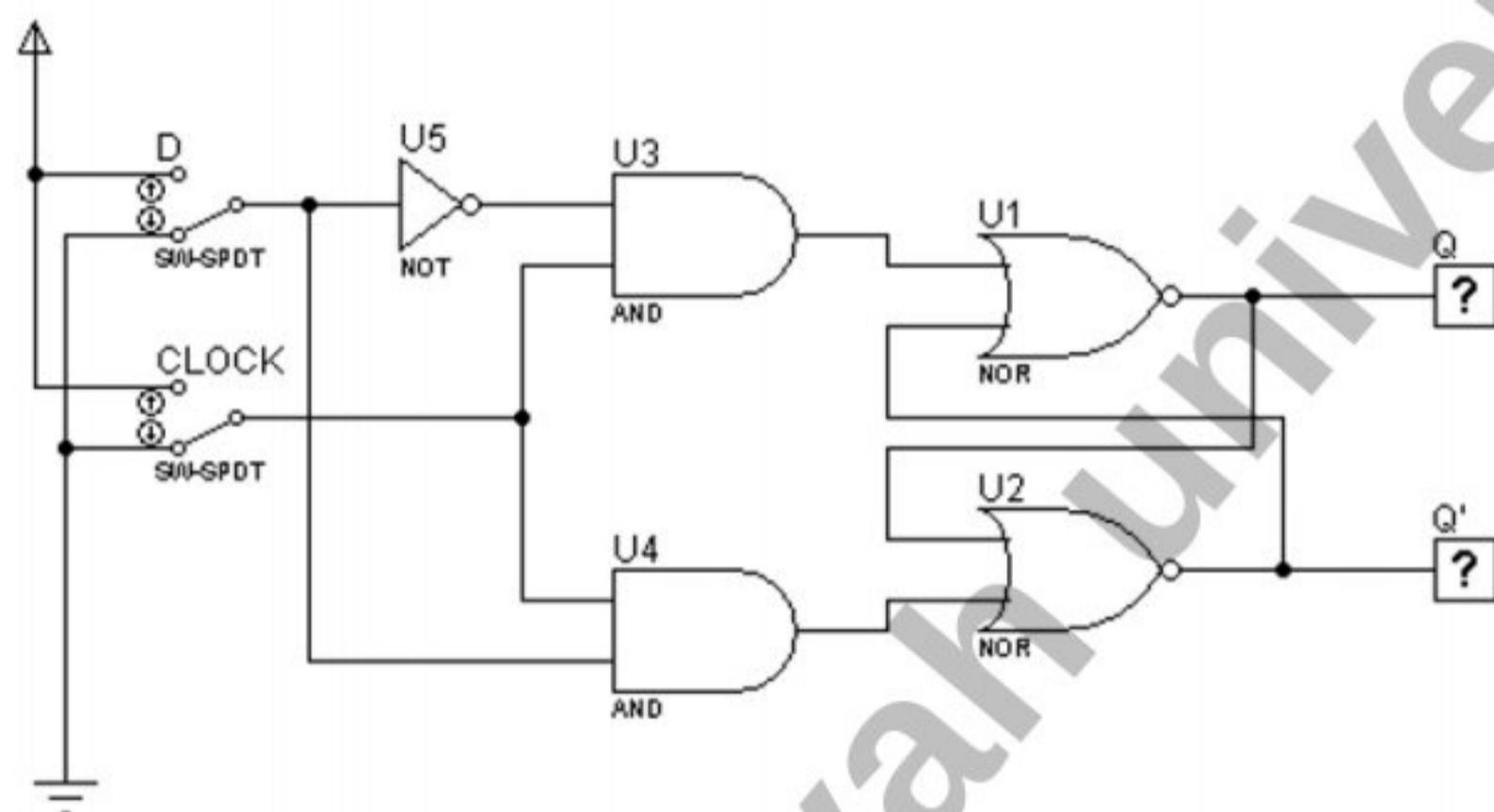
.....

- c. Jelaskan bagaimana Flip-flop RS bekerja!

.....

#### Percobaan 4. Flip-Flop D

1. Buat dan simulasikan Flip-Flop D seperti pada gambar! Dan kemudian klik switch SW-SPDT untuk mengoperasikan flip-flop.



2. Berdasarkan simulasi anda, isi titik-titik pada tabel berikut!

	D	CLOCK	Output		Mode
			Q	Q <sub>(t+1)</sub>	
1	0	0			
2	0	1			
3	1	0			
4	1	1			
5	0	0			
6	0	1			
7	1	0			
8	1	1			

Catatan : Kolom mode menjelaskan kondisi kinerja dari FF tersebut, seperti 'memori' atau 'Data in/masuk'.

3. Jelaskan bagaimana Flip-flop D bekerja!

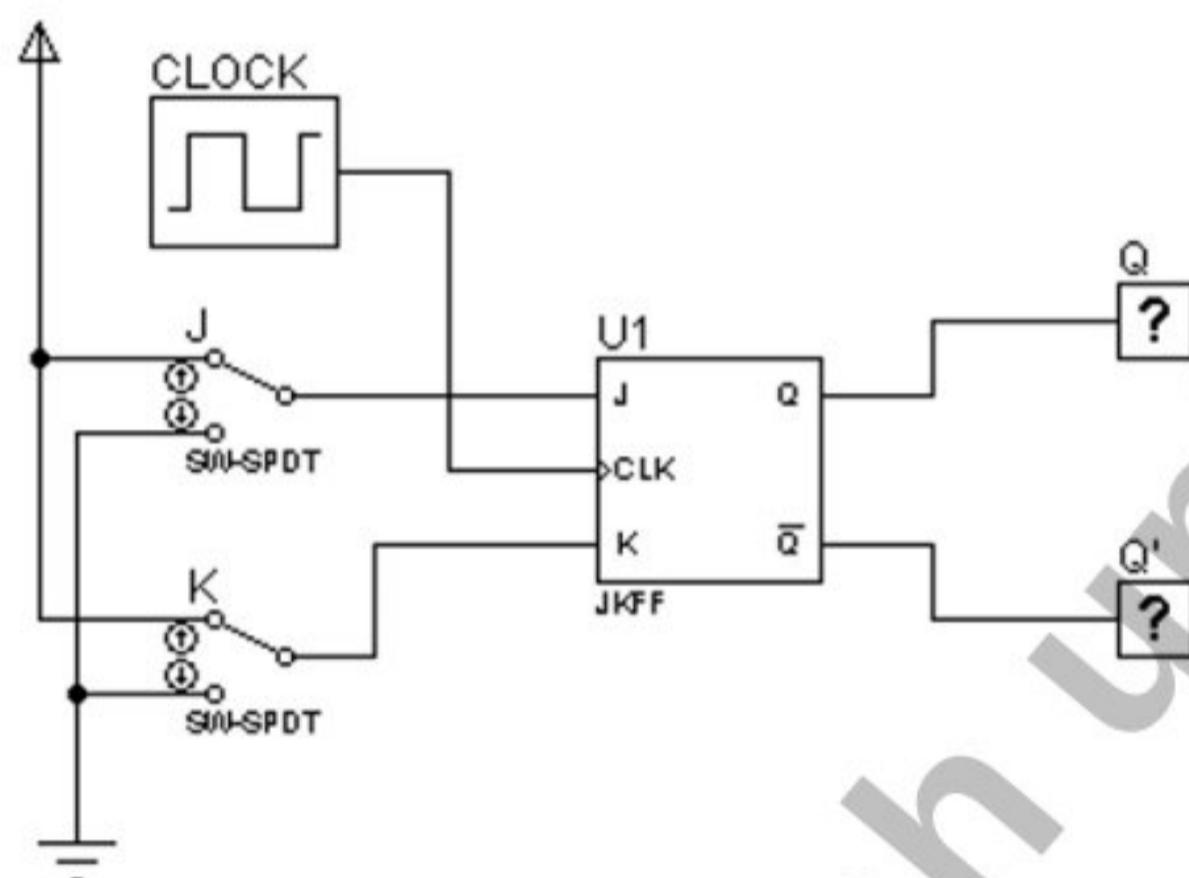
.....  
.....

4. Apa fungsi NOT gate pada Flip-Flop D dibandingkan dengan Flip-Flop SR!

.....  
.....

### Percobaan 5. Flip-Flop JK

1. Buat dan simulasi JK Flip-Flop seperti pada gambar! Dan kemudian klik switch SW-SPDT untuk mengoperasikan flip-flop.



2. Berdasarkan simulasi anda, isi titik-titik pada tabel berikut!

	J	K	CLOCK	Output		Mode
				Q	$Q_{(t+1)}$	
1	0	0	0			
2	0	0	1			
3	0	1	0			
4	0	1	1			
5	1	0	0			
6	1	0	1			
7	1	1	0			
8	1	1	1			

Catatan : Kolom mode menjelaskan kondisi kinerja dari FF tersebut, seperti 'memori', 'set', 'reset', atau 'togle'

3. Jawab pertanyaan berikut ini
- Apa yang akan terjadi jika  $J = K = 0$ , dan clock rise up (change from 0 to 1)?

.....

.....

- Apa yang akan terjadi jika  $J = K = 1$ , dan clock rise up?

.....

.....

4. Jelaskan bagaimana Flip-flop JK bekerja!

.....

.....

# MODUL 8

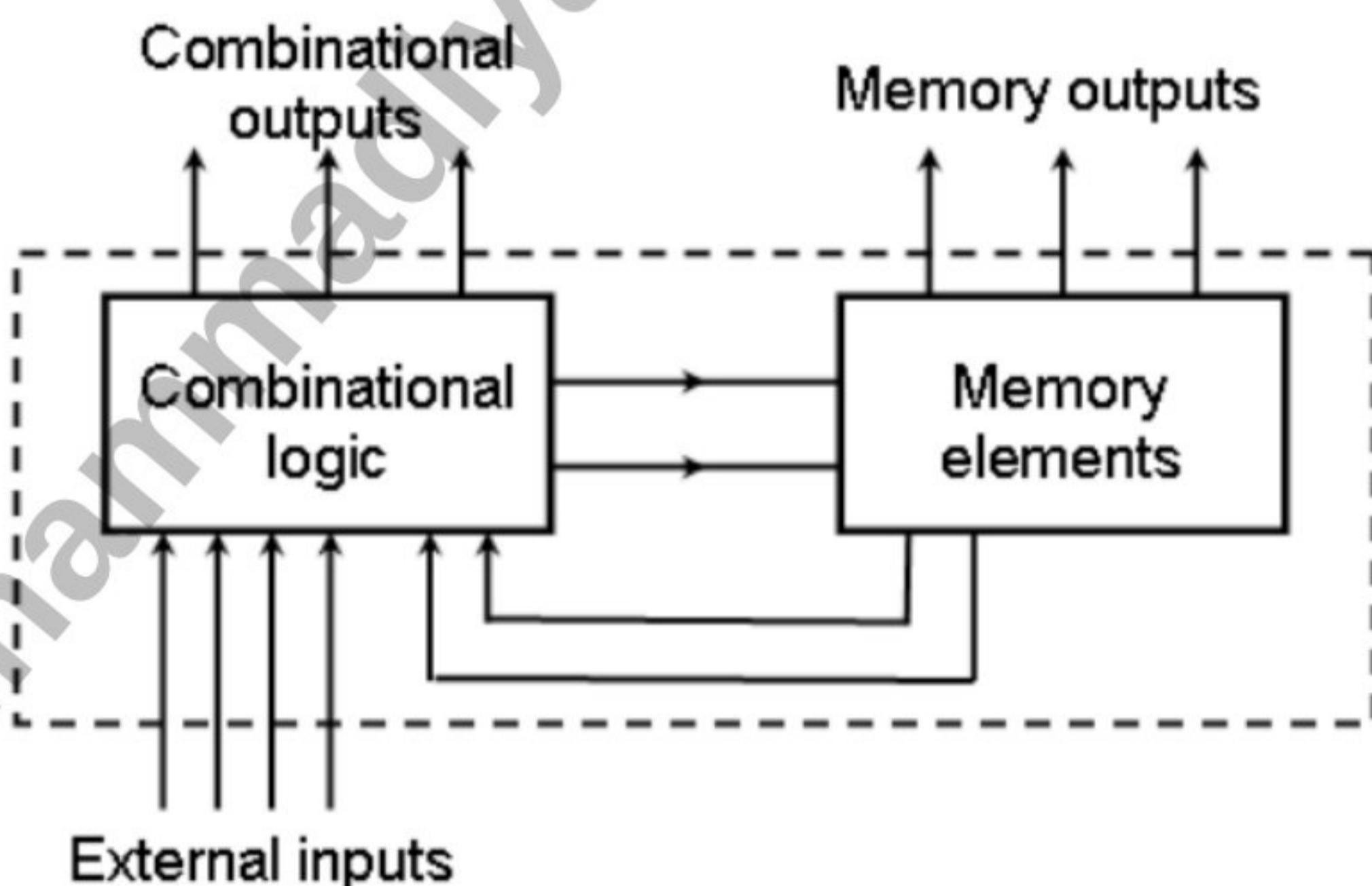
## APLIKASI FLIP – FLOP

### TUJUAN PRAKTIKUM

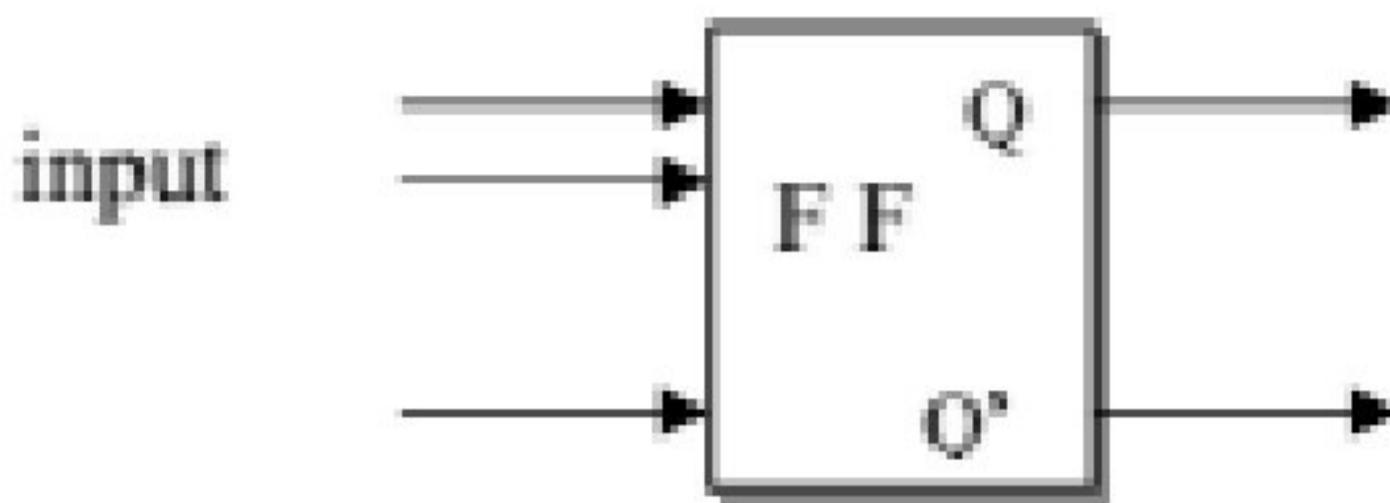
Mahasiswa mampu memahami dan membuat beberapa contoh aplikasi flip-flop.

### TEORI

Rangkaian logika yang telah dibahas dalam praktikum adalah rangkaian logika kombinatorial dengan level output tergantung dengan level inputannya. Keadaan level input ynf tidak mempunyai pengaruh terhadap level outputnya merupakan rangkaian kombinatorial tanpa memori. System digital umumnya terdiri dari dua bagian tersebut, yaitu rangkaian kombinatorial dan elemen memori seperti ditunjukkan pada Gambar 8.1. system digital tersebut menggabungkan gerbang logika dan elemen memori yang menerima sinyal logika dari masukan (input) eksternal dan dari output memori.



Gambar 8.1. Diagram sistem digital umum



Gambar 8.2. Flip flop secara umum

Flip flop (FF) merupakan contoh elemen memori yang banyak digunakan. FF merupakan rangkaian gerbang logika dengan dua output yang berlawanan seperti ditunjukkan Gambar 8.2. Dua keadaan kerja FF yang mungkin terjadi adalah (1)  $Q = 0, Q' = 1$ . (2)  $Q = 1, Q' = 0$ . FF mempunyai satu input atau lebih. Input tersebut berfungsi untuk mengoperasikan FF bolak-balik antara dua keadaan tersebut. Sebuah sinyal input menyebabkan FF beroperasi menuju suatu keadaan tertentu, dan FF tersebut akan tetap berada pada keadaan terakhir meskipun sinyal inputnya terputus. Hal ini merupakan karakteristik dari rangkaian FF. Tabel 8.1. merupakan tabel kebenaran dari empat jenis FF yang perlu dipahami karakteristik dan kinerjanya.

FF dapat digunakan di dalam counter biner sehingga keadaan output dari FF dapat menyatakan suatu bilangan biner yang sama dengan jumlah pulsa-pulsa clock yang datang. Rangkaian counter adalah rangkaian elektronika yang berfungsi untuk melakukan penghitungan angka secara berurutan baik perhitungan maju ataupun perhitungan mundur. Perhitungan maju dimaksudkan menghitung mulai dari angka kecil ke besar. Begitu juga sebaliknya yang dimaksud dengan perhitungan mundur. Manfaat rangkaian counter dalam dunia elektronika digital sangat besar. Hampir semua rangkaian digital memerlukan rangkaian counter. Hal ini disebabkan rangkaian counter dapat digunakan untuk melakukan fungsi perhitungan angka atau operasi matematika.

Tabel 8.1. Tabel kebenaran empat jenis Flip Flop

**S-R Flip Flop**

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

**D Flip Flop**

$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

### J-K Flip Flop

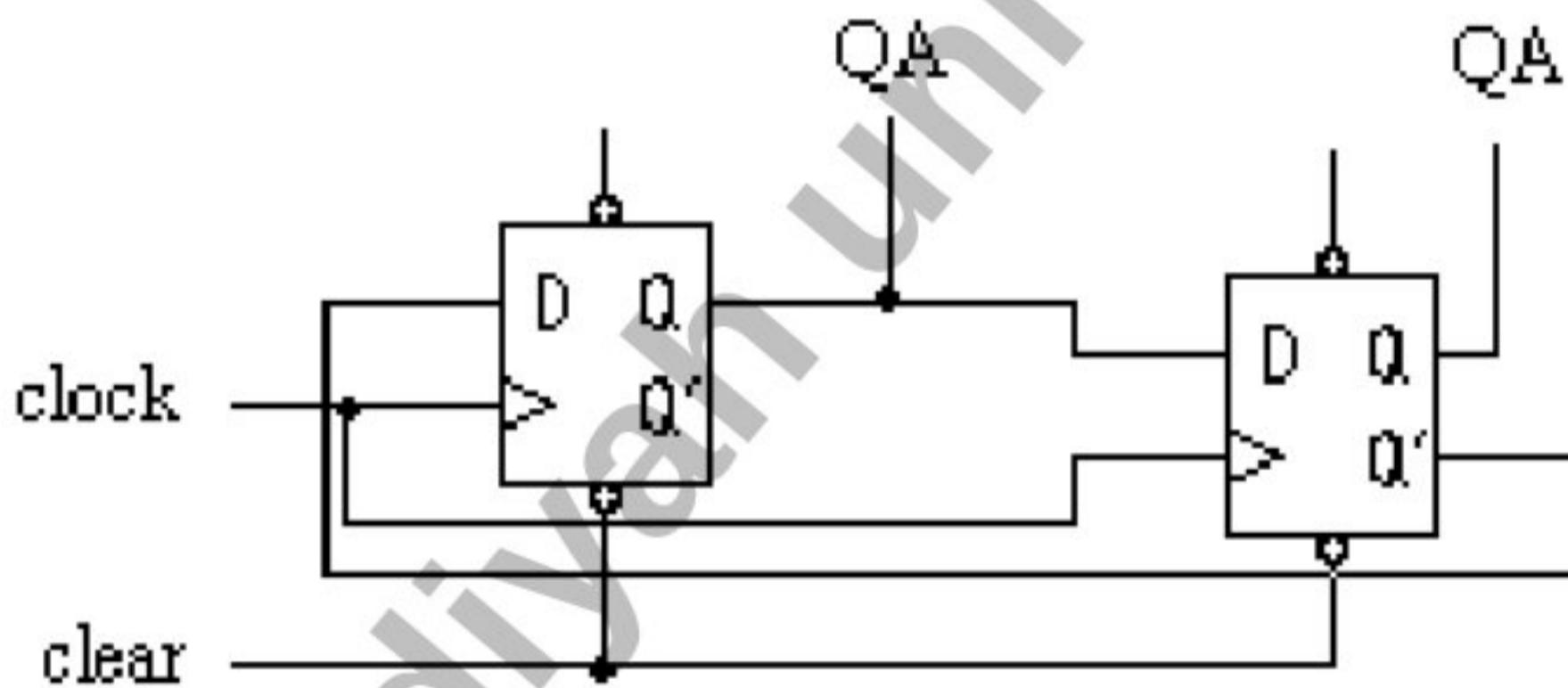
$Q(t)$	$Q(t+1)$	$J$	$K$
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

### T Flip Flop

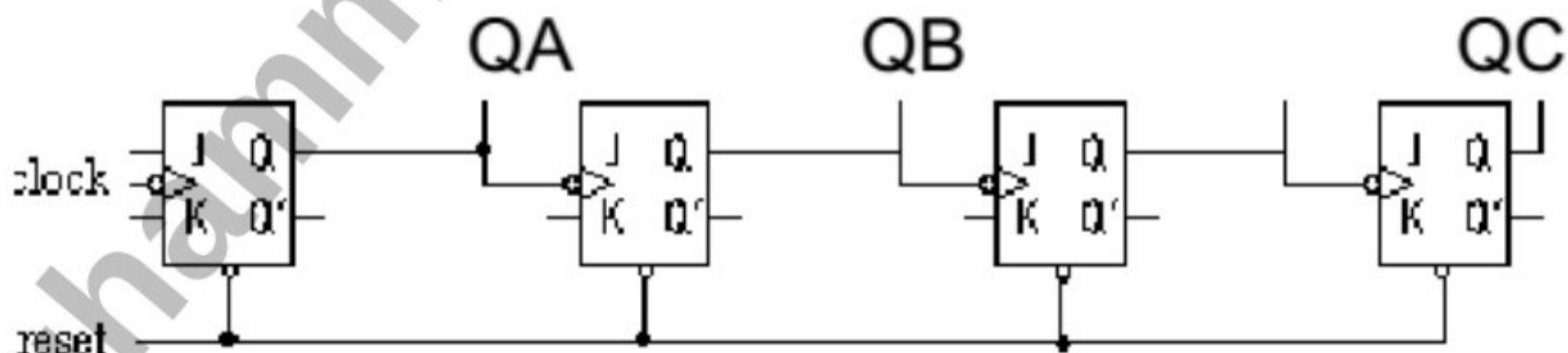
$Q(t)$	$Q(t+1)$	$T$
0	0	0
0	1	1
1	0	1
1	1	0

Catatan : Nilai X dapat diisi dengan nilai 0 atau 1.

Counter dibagi menjadi dua yaitu counter sinkron (*synchronous counter*) dan counter tak sinkron (*asynchronous counter*). Counter sinkron merupakan counter yang bekerja serentak dengan pulsa clock seperti ditunjukkan pada Gambar 8.3. Counter sinkron disebut juga pencacah deret atau *parallel counter*. Sedangkan counter tak sinkron adalah counter yang bekerja tidak serentak dengan pulsa clock. Hanya FF pertama yang dikendalikan oleh sinyal clock, sedangkan masukan FF lainnya diambil dari FF sebelumnya seperti ditunjukkan pada Gambar 8.4.



Gambar 8.3. counter sinkron 2 bit menggunakan flip flop D

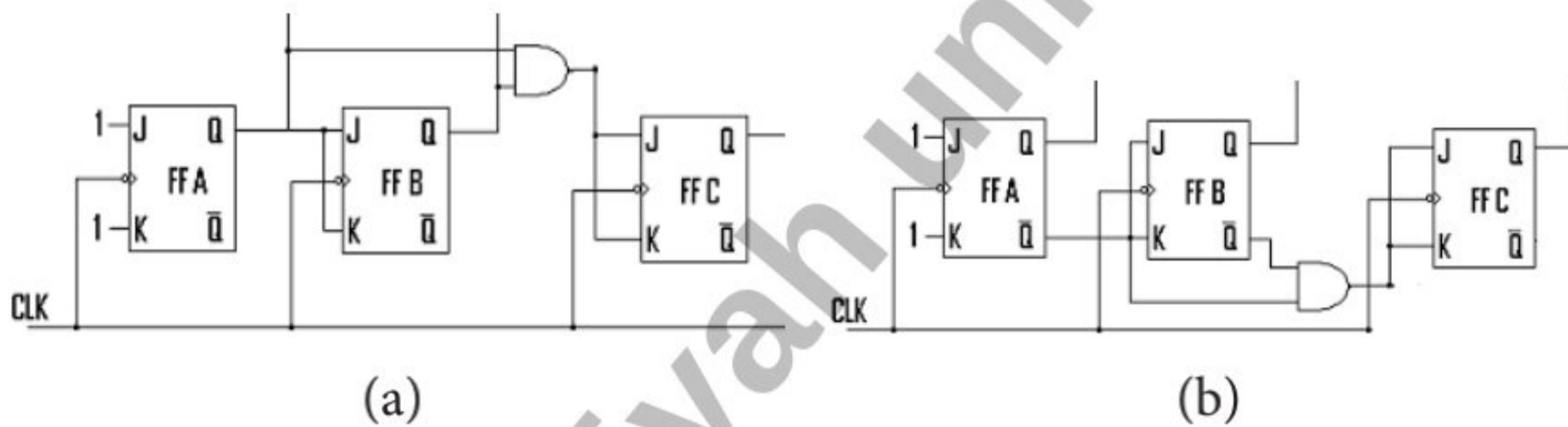


Gambar 8.4. Counter tak sinkron menggunakan flip flop JK

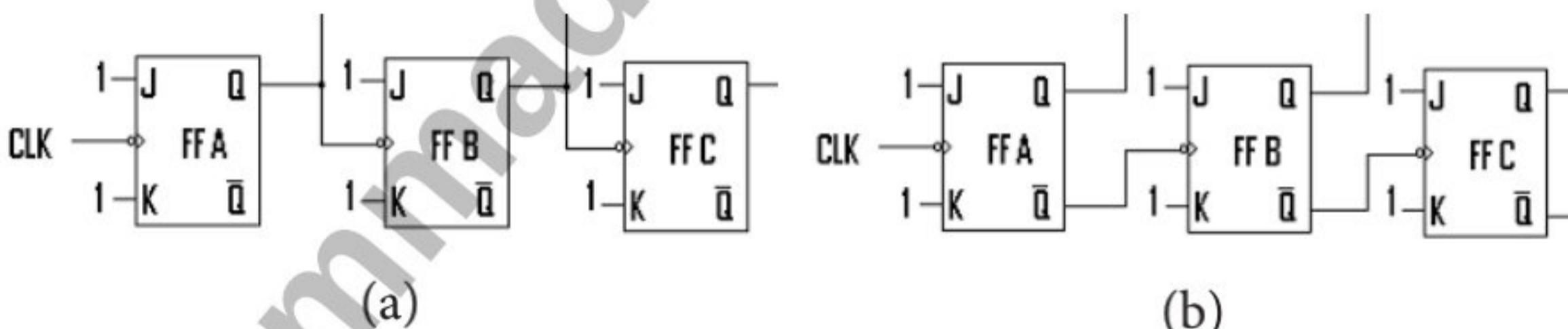
Counter adalah rangkaian elektronika digital yang menggunakan logika digital yang dipicu oleh pulsa clock atau dikenal juga sebagai rangkaian sekuensial. Counter biasanya mencacah atau menghitung dalam biner yang

dapat dihentikan atau mengulang perhitungan dari awal setiap saat. Untuk counter yang berulang, jumlah kondisi biner yang berbeda menunjukkan modulus (MOD) counter. Contohnya, MOD-8 atau modulus 8 merupakan counter yang mencacah dari 0-1-2-3-4-5-6-7 secara berulang. Rangkaian dasar counter terdiri dari beberapa flip-flop yang jumlahnya bergantung pada modulus yang diperlukan.

Berdasarkan arah cacahan, rangkaian counter dibedakan menjadi dua yaitu up counter (pencacah naik) dan down counter (pencacah turun). Up counter melakukan perhitungan dari kecil ke besar kemudian kembali ke nilai awal secara otomatis. Sedangkan down counter melakukan perhitungan dari besar ke kecil hingga nilai terakhir dan kembali ke nilai awal secara otomatis. Untuk membangun up/down counter melalui tiga faktor diantaranya (1) pemilihan transisi FF yang aktif, (2) pemilihan output FF yang dijadikan input untuk FF berikutnya, dari output Q atau  $\bar{Q}$ . (3) pemilihan indicator hasil cacahan yang dijadikan output, Q atau  $\bar{Q}$ . Gambar 8.5 merupakan contoh rangkaian counter sinkron up/down. Gambar 8.6 merupakan contoh rangkaian tak sinkron up/down.

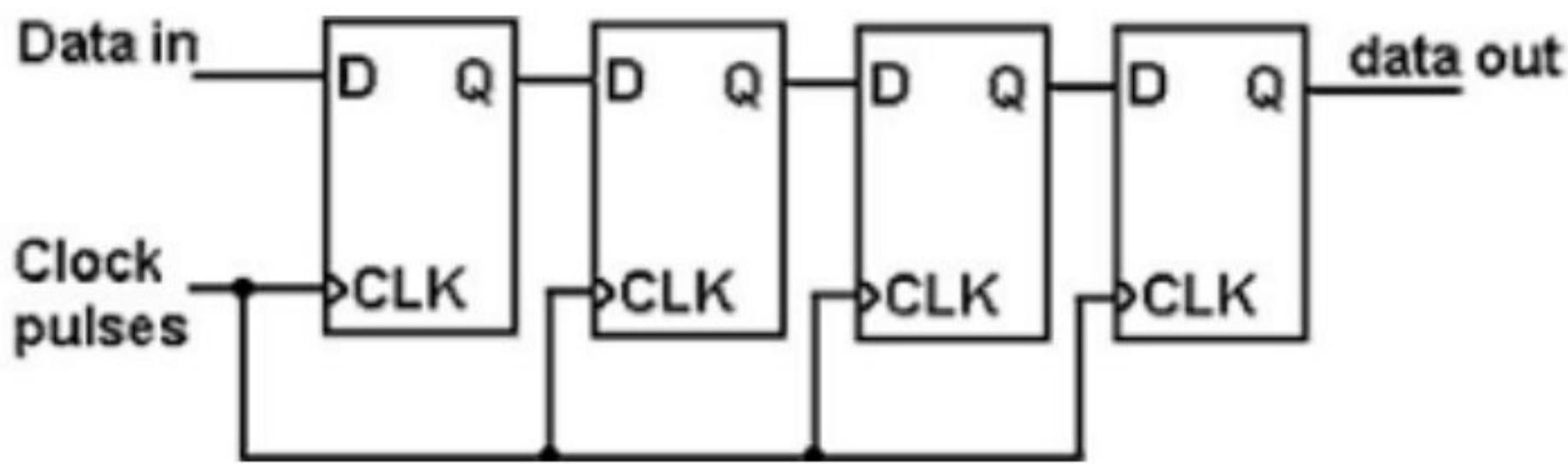


Gambar 8.4. Counter sinkron (a) up counter, (b) down counter



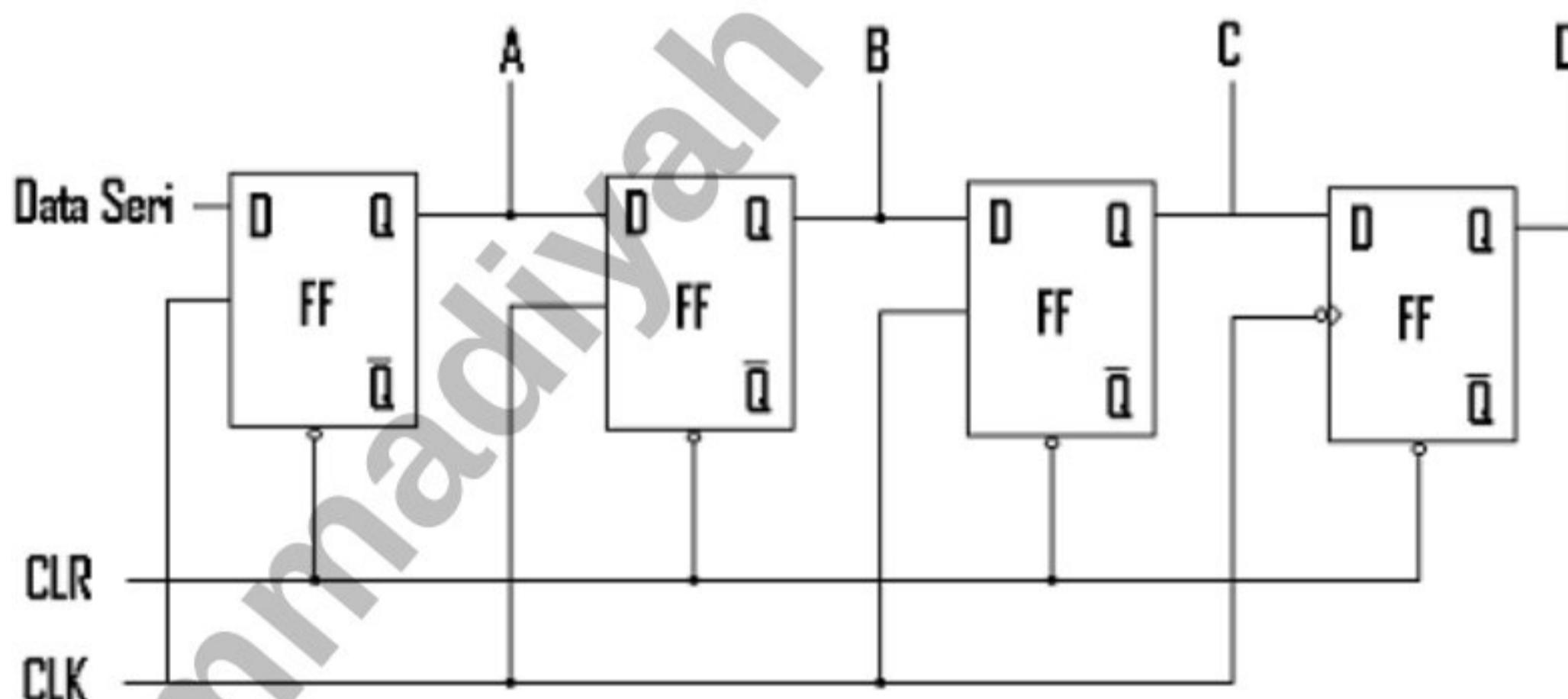
Gambar 8.5. Counter tak sinkron (a) up counter, (b) down counter

Implementasi FF yang lain yaitu berupa register. Register adalah kumpulan elemen memori yang bekerja bersama-sama menjadi satu buah unit sebagai penyimpan bit/memori. Data biner yang dimasukkan dapat secara seri atau pararel dan output data dapat dikeluarkan secara seri atau pararel. Beberapa jenis register diantaranya register buffer dan register geser. Register buffer adalah jenis register yang sederhana untuk menyimpan data digital. Contoh register buffer terlihat pada Gambar 8.6 yang terdiri dari 4 buah D-FF.



Gambar 8.6 Register buffer terdiri dari 4 buah D-FF

Register geser merupakan jenis register yang menyimpan data dan dapat bergeser ke kiri atau ke kanan. Fungsi lain dari register geser yaitu untuk mengubah data dari seri ke pararel atau sebaliknya. Penggunaan register geser ini dapat ditemukan pada kinerja layar sebuah kalkulator, dimana angka dapat bergeser ke kiri setiap kali ada angka baru yang dimasukkan. Gambar 8.7. merupakan contoh rangkaian register geser menggunakan 4 buah D-FF. Cara kerja register geser yaitu apabila pulsa clock masuk, maka data dari input D masing-masing FF akan di transfer ke output Q. Jika nilai 1 merupakan input FF pertama, maka pada pulsa berikutnya nilai 1 tersebut akan di transfer ke FF yang pertama dan sekaligus menjadi input untuk FF kedua dan seterusnya. Register geser terbagi menjadi beberapa jenis diantaranya register geser beban seri, register geser pararel, dan register geser universal.

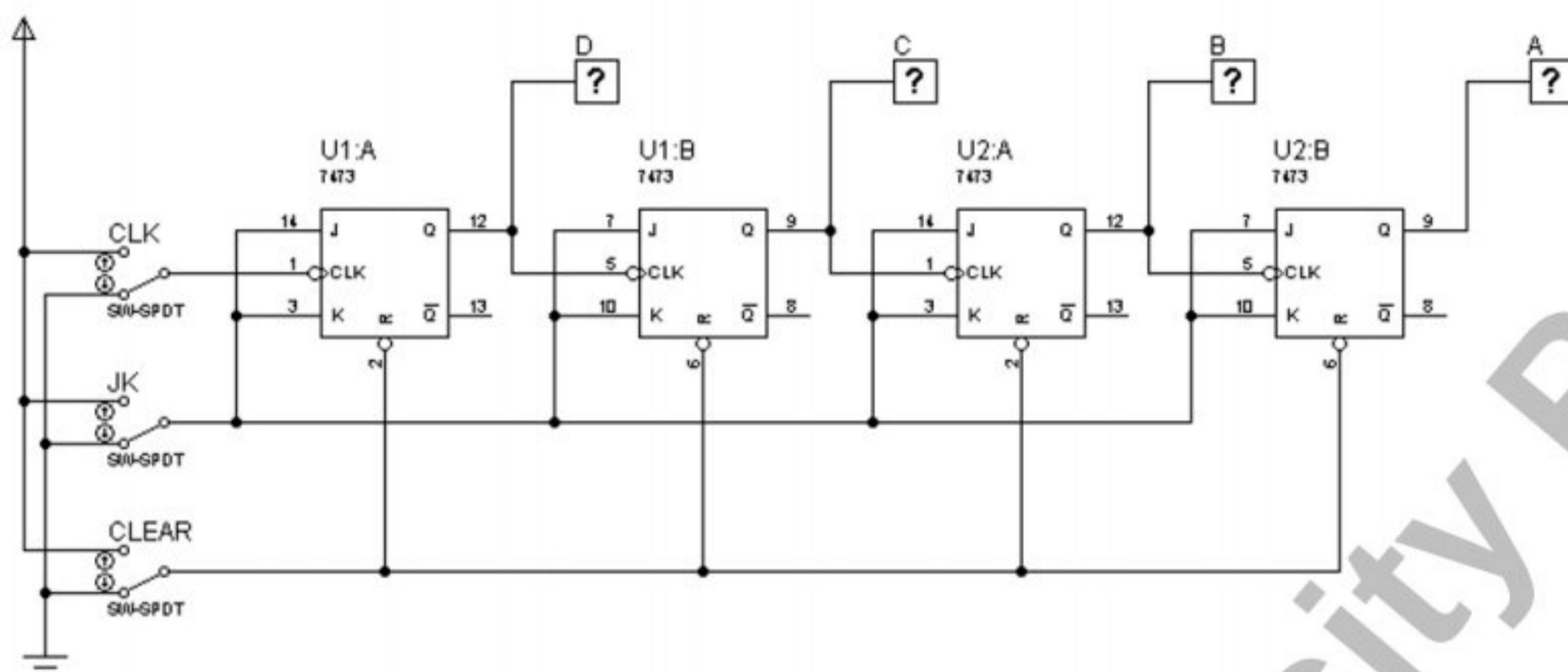


Gambar 8.7. Register geser.

## KEGIATAN PRAKTIKUM

### Percobaan 1. Membuat Counter JK-FF

- Buat kombinasi flip-flop JK seperti pada gambar!



- Simulasikan rangkaian anda!

Klik pada switch berdasarkan pada tabel dan isi kolom kosong pada tabel!

NO	INPUT			OUTPUT			
	CLR	JK	CLK	A	B	C	D
1	1	1	0				
2	1	1	1				
3	1	1	0				
4	1	1	1				
5	1	1	0				
6	1	1	1				
7	1	1	0				
8	1	1	1				
9	1	1	0				
10	1	1	1				
11	1	1	0				

NO	INPUT			OUTPUT			
	CLR	JK	CLK	A	B	C	D
12	1	1	1				
13	1	1	0				
14	1	1	1				
15	1	0	0				
16	1	0	1				
17	1	1	0				
18	1	1	1				
19	0	1	0				
20	0	1	1				

- Apa fungsi dari :

- Switch CLK:

.....

- Switch JK:

.....

- Switch CLEAR:

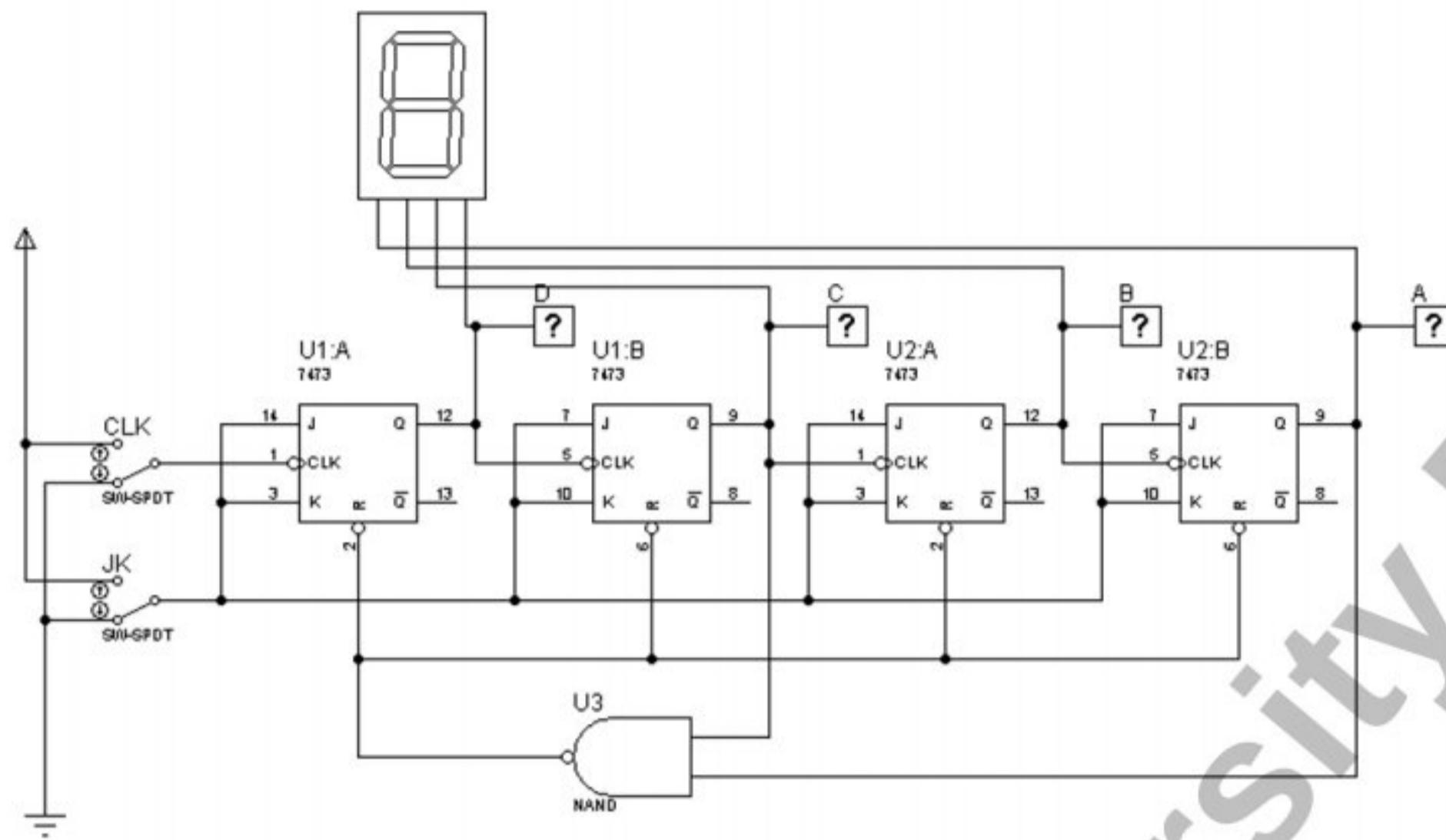
.....

- Kesimpulan:

.....

## Percobaan2. Counter Mod 10

- Buat kombinasi flip-flop JK seperti pada gambar!



- Simulasikan rangkaian anda!

Klik pada switch berdasarkan pada tabel dan isi kolom kosong pada tabel!

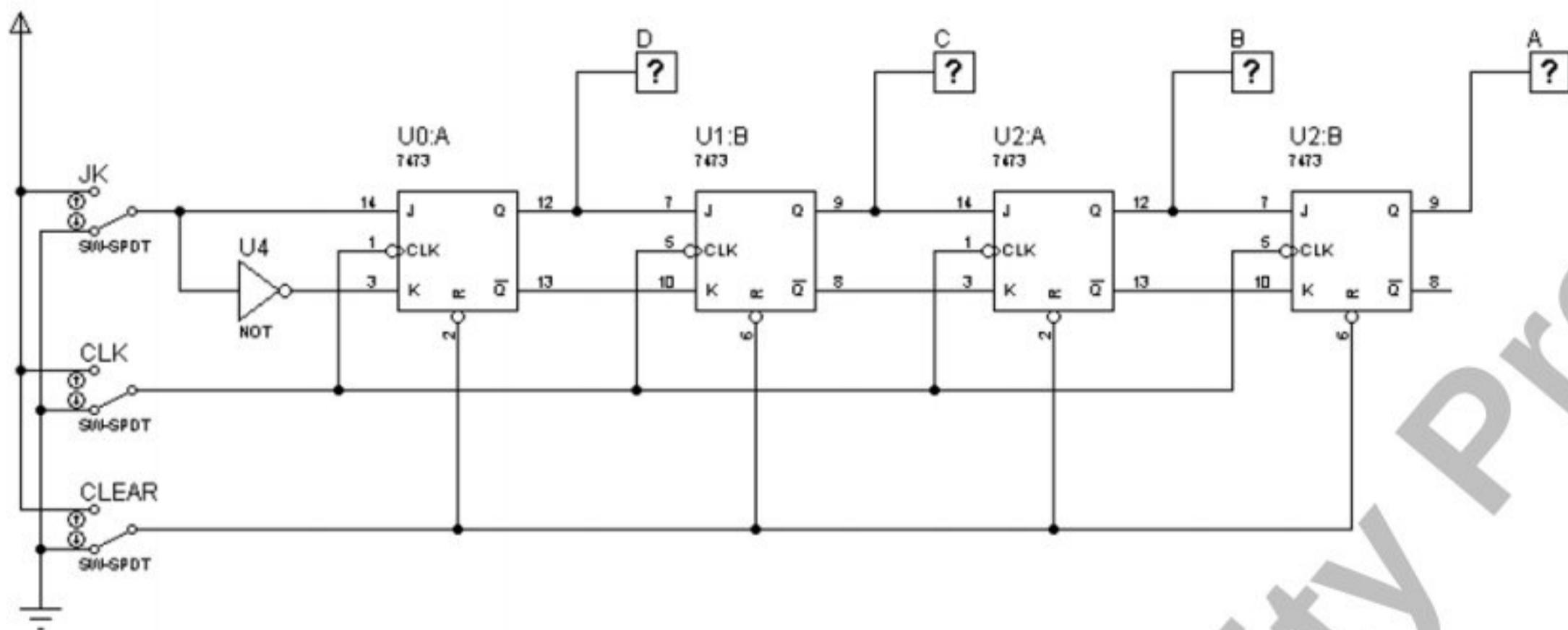
NO	INPUT		OUTPUT			
	JK	CLK	A	B	C	D
1	1	0				
2	1	1				
3	1	0				
4	1	1				
5	1	0				
6	1	1				
7	1	0				
8	1	1				
9	1	0				
10	1	1				
11	1	0				
12	1	1				

NO	INPUT		OUTPUT			
	JK	CLK	A	B	C	D
13	1	0				
14	1	1				
15	1	0				
16	1	1				
17	1	0				
18	1	1				
19	1	0				
20	1	1				
21	0	0				
22	0	1				
23	1	0				
24	1	1				

Kesimpulan:

### Percobaan 3. Membuat Register JK-FF

- Buat kombinasi flip-flop JK seperti pada gambar!



- Simulasikan rangkaian anda!

Klik pada switch berdasarkan pada tabel dan isi kolom kosong pada tabel

NO	INPUT			OUTPUT			
	CLR	JK	CLK	A	B	C	D
1	0	X	-				
2	1	1	-				
3	1	1	1				
4	1	1	2				
5	1	1	3				
6	1	0	4				
7	1	0	5				
8	1	0	6				

NO	INPUT			OUTPUT			
	CLR	JK	CLK	A	B	C	D
9	1	0	7				
10	1	0	8				
11	1	0	9				
12	1	0	10				
13	1	0	11				
14	1	0	12				
15	1	0	13				

Kesimpulan :

.....

.....

.....

.....

Catatan :

- Flip-flop adalah elemen dasar untuk membuat counter dan register, yang merupakan fundamental building block sangat penting pada sistem elektronik digital yang digunakan dalam sistem komputer.

# MODUL 9

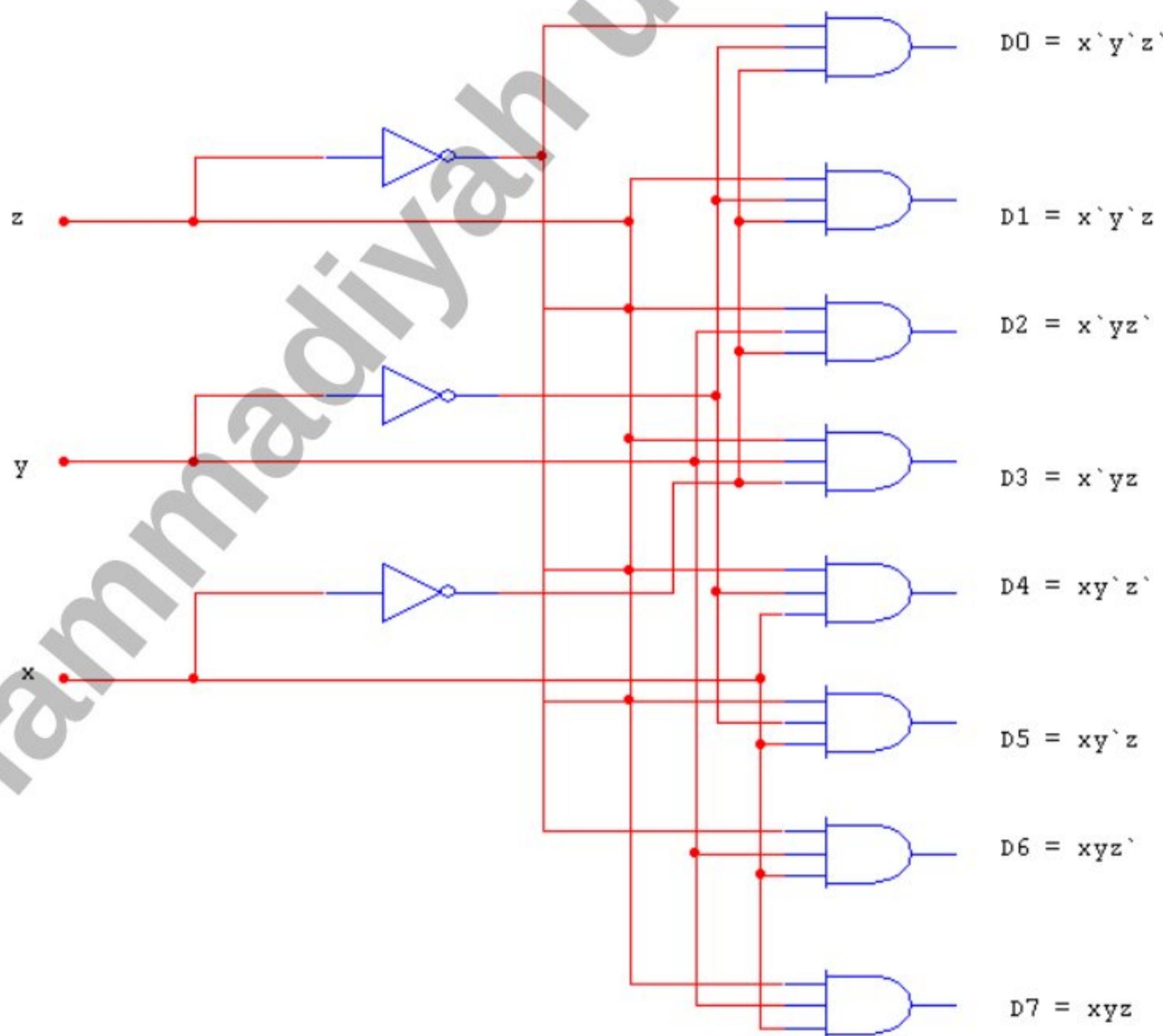
## DECODER

### TUJUAN PRAKTIKUM

Mahasiswa mampu membuat decoder dan memahami cara kerjanya.

### TEORI

Decoder merupakan rangkaian kombinasi yang berfungsi untuk mengkonversi informasi biner dari baris input n kedalam output yang unik, dengan decoder basis disebut juga n-to-m. Nama decoder juga digunakan untuk beberapa kode konversi seperti decoder BCD-to-7 segment. Gambar 9.1 merupakan contoh rangkaian decoder 3 ke 8 yaitu mempunyai 3 input dan 8 output () .



Gambar 9.1. Rangkaian decode 3 ke 8

Tiga masukan tersebut di-decode menjadi 8 output. Setiap output menyatukan kombinasi dari tiga variable input. Gerbang NOT menyediakan tiga pelengkap dari input, dan masing-masing gerbang menghasilkan sebuah kombinasi. Contoh aplikasi khusus dari decoder adalah proses konversi dari biner ke oktal. Variable input bersatu dan menghasilkan 8 digit output dalam sistem bilangan oktal. Operasi decoder dapat dilihat pada hubungan input-output seperti terlihat pada Tabel 9.1. Variabel output sangat eksklusif karena hanya satu output yang dibandingkan dengan tiga kombinasi masukan.

Tabel 9.1. Tabel kebenaran decoder 3 ke 8 ( 0 = mati, 1 = nyala)

Inputs			Outputs							
X	y	z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

## DECODER BCD-TO-SEVEN SEGMENT

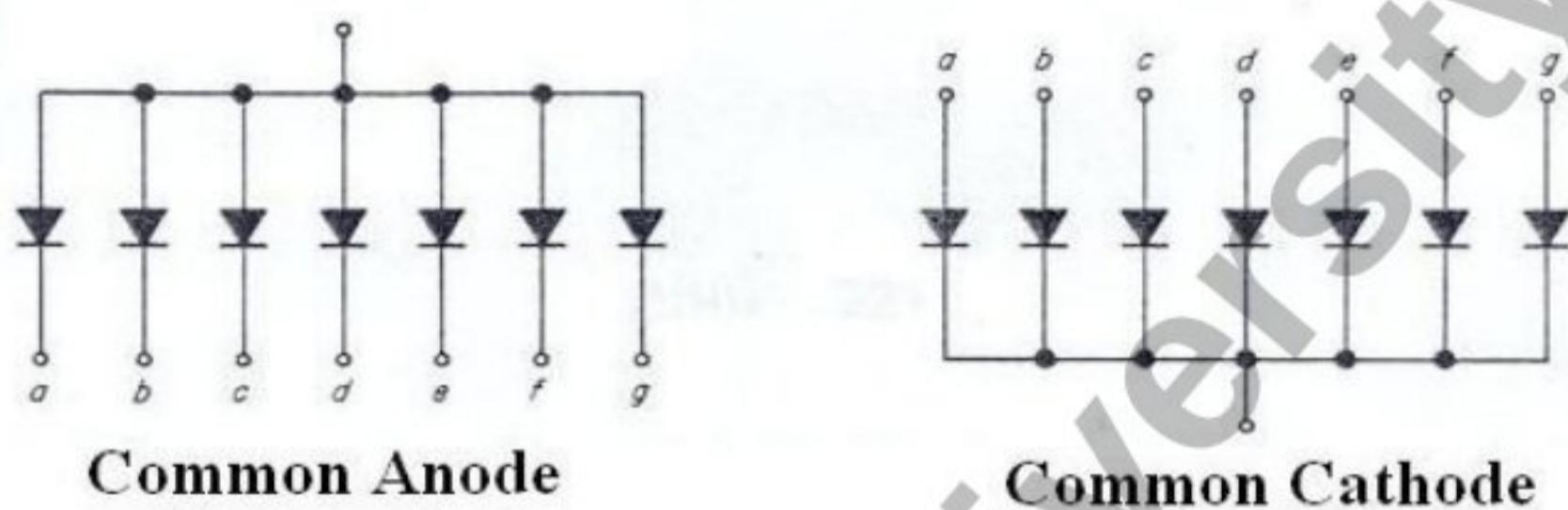
Salah satu fungsi dari decoder adalah untuk mengubah kode dari sinyal input menjadi sinyal output khusus. Penggunaan decoder salah satunya untuk antarmuka atau tampilan (display). Display decoder adalah sebuah perangkat yang mengubah sinyal input BCD menjadi sinyal output sesuai kebutuhan untuk mendorong beberapa display numerik atau alpha numerik. Pada Tabel 2 menjelaskan beberapa jenis display numerik dengan kode input dan tegangan suplai relatifnya.

Tabel 9.2. Display numerik.

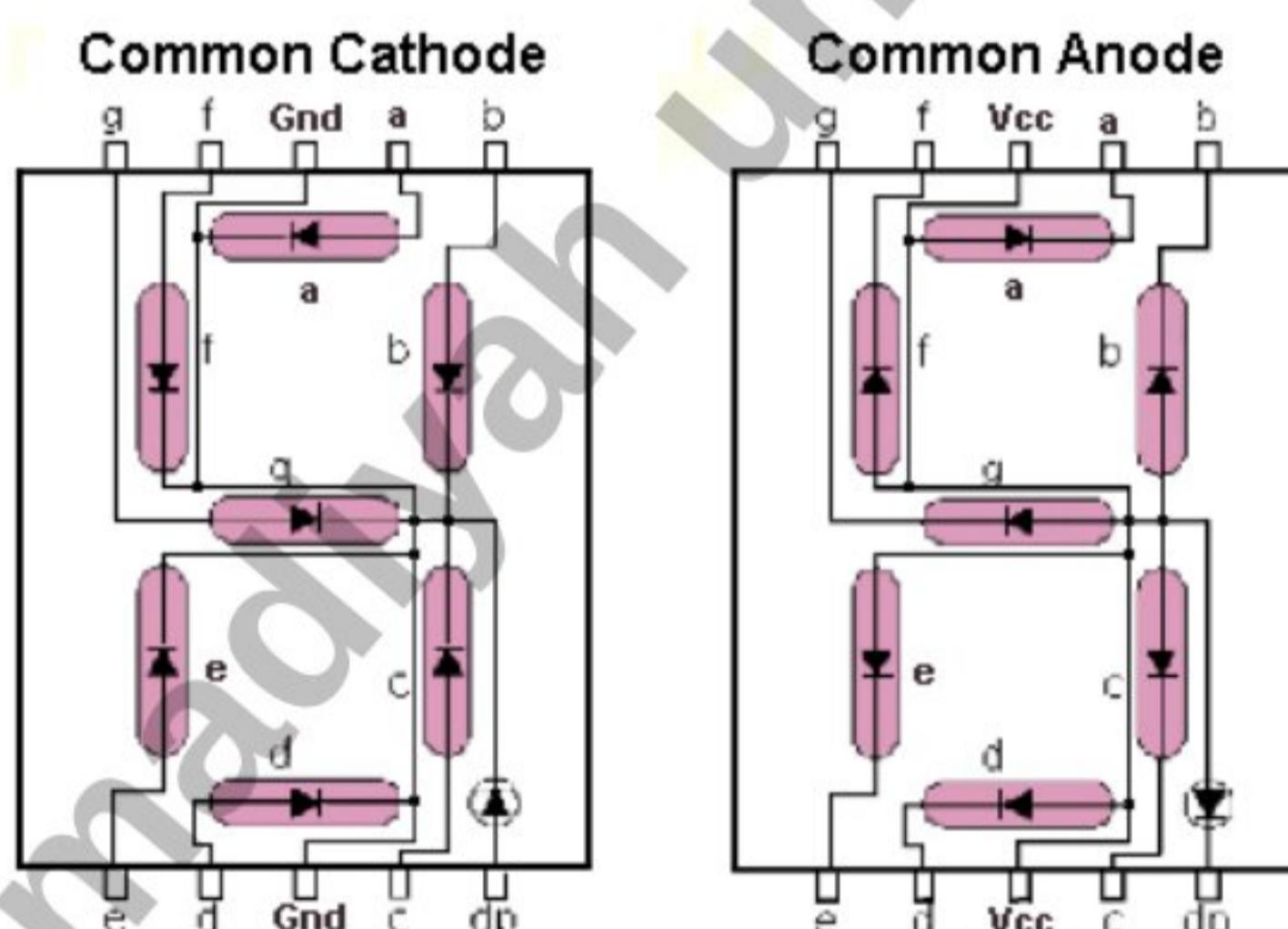
Display	Input code	Relative Supply voltage
Cold cathode		
Older type	1 of 10	High dc
Newer type	7-segment	High dc

Display	Input code	Relative Supply voltage
Gallium arsenide		
Phosphide (LED)	7-segment	Low dc
Incandescent	7-segment	Low dc
Liquid crystal	7-segment	Low dc

Pemilihan decoder tergantung pada pilihan display 7-segment, apakah jenis common-anoda atau common-katoda. Gambar 9.2 memperlihatkan jenis common anode dan common katode. Sedangkan Gambar 9.3 menunjukkan tampilan 7-segment.



Gambar 9.2. Jenis decoder



Gambar 9.3. Tampilan visual 7-semen

Tabel 2 memperlihatkan decoder / BCD to 7 segmen pada IC 4511 yang dihubungkan ke display LED 7 segmen commong katoda. Pemilihan nilai resistor (R) tergantung pada tegangan dan arus untuk display LED 7 segmen. Nilai Resistor terlihat jelas pada Gambar 5 yaitu sekitar 470  $\Omega$ . Tabel 3 merupakan tabel kebenaran dari Decoder / driver BCD ke 7

segment untuk IC 4511.

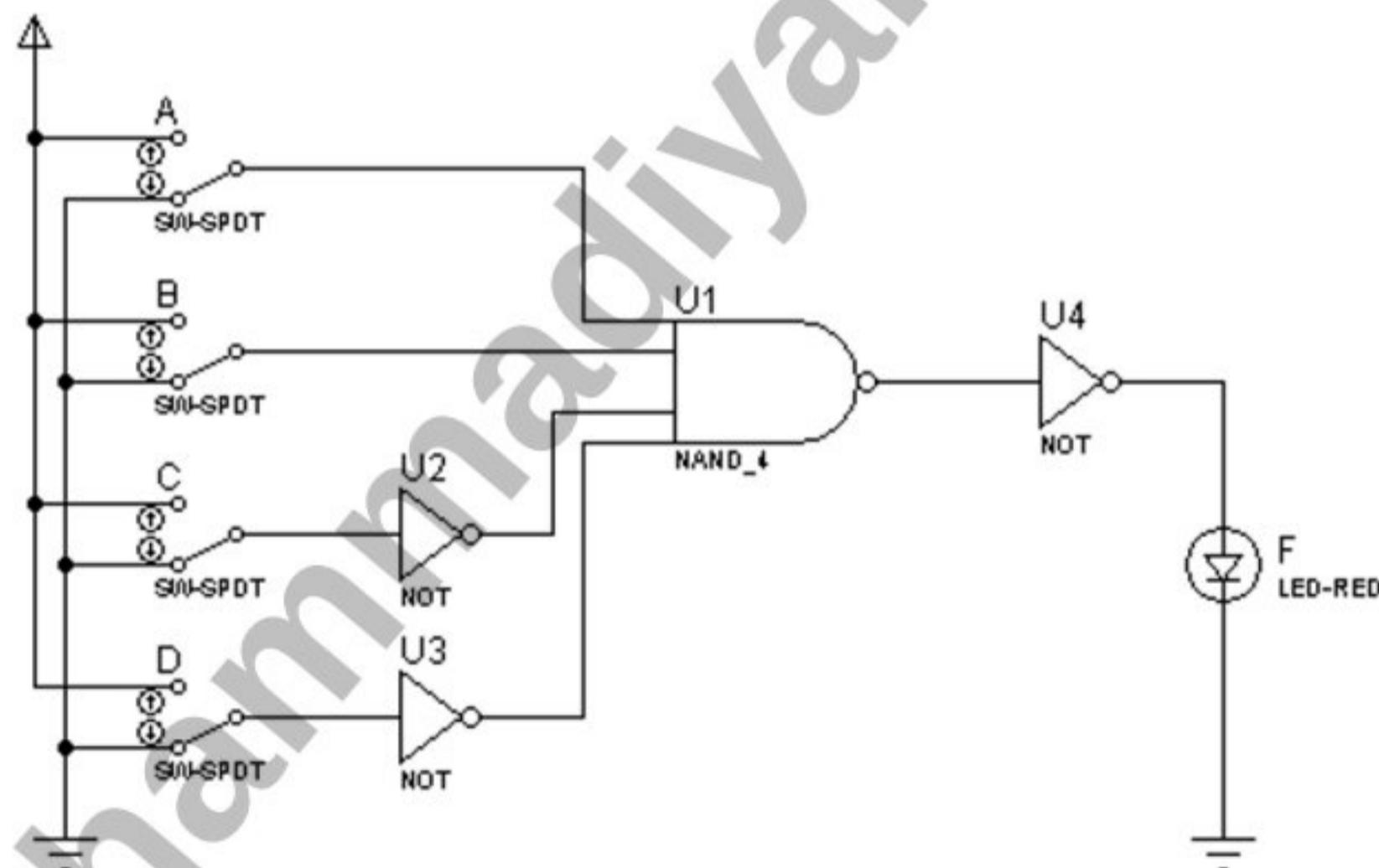
Tabel 9.3. Tabel kebenaran “High-Low” IC 4511 dan outputnya

Decimal Digit	Input					Output							Display Output
	LT	D	C	B	A	a	b	c	d	e	f	g	
0	H	L	L	L	L	H	H	H	H	H	H	L	0
1	H	L	L	L	H	L	H	H	L	L	L	L	1
2	H	L	L	H	L	H	H	L	H	H	L	H	2
3	H	L	L	H	H	H	H	H	H	L	L	H	3
4	H	L	H	L	L	L	H	H	L	L	H	H	4
5	H	L	H	L	H	H	L	H	H	L	H	H	5
6	H	L	H	H	L	L	L	H	H	H	H	H	6
7	H	L	H	H	H	H	H	H	L	L	L	L	7
8	H	H	L	L	L	H	H	H	H	H	H	H	8
9	H	H	L	L	H	H	H	H	L	L	H	H	9
LT	L	X	X	X	X	H	H	H	H	H	H	H	8

## KEGIATAN PRAKTIKUM

### Percobaan 1. Membuat Rangkaian Decoder Sederhana

- Buat decoder seperti pada gambar berikut!



- Isi kolom kosong pada tabel!

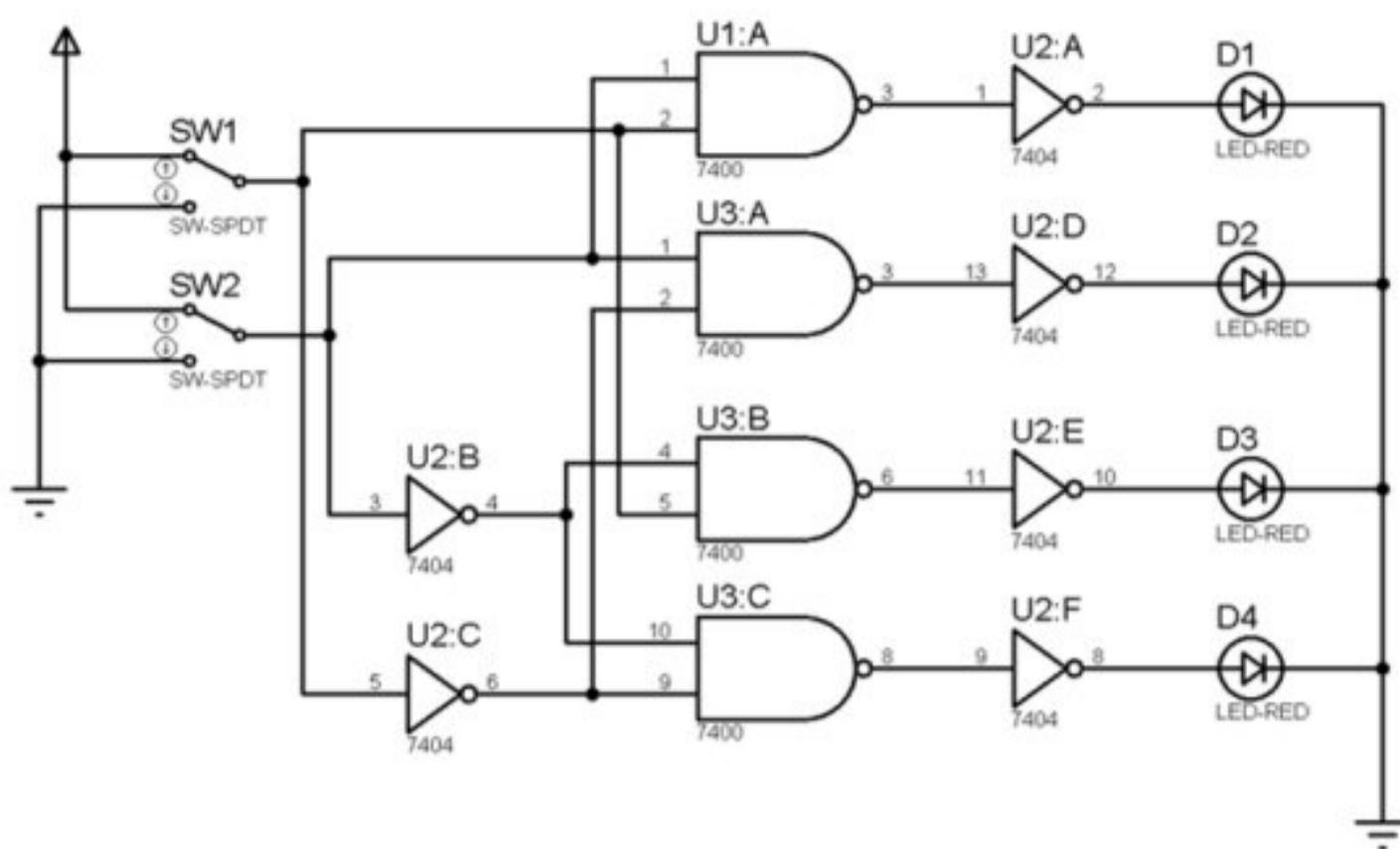
NO	A	B	C	D	F
1	0	0	0	0	
2	1	0	0	0	
3	0	1	0	0	

<b>NO</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>F</b>
4	1	1	0	0	
5	0	0	1	0	
6	1	0	1	0	
7	0	1	1	0	
8	1	1	1	0	
9	0	0	0	1	
10	1	0	0	1	
11	0	1	0	1	
12	1	1	0	1	
13	0	0	1	1	
14	1	0	1	1	
15	0	1	1	1	
16	1	1	1	1	

3. Decoder (F) hanya bekerja (ON) ketika : A = ...., B = ...., C = ...., dan D = ....
4. Berdasarkan rangkaian dan tabel diatas, coba buat decoder yang mempunyai output sebagaimana fungsi decoder berikut :  
 $F = 1$ , jika kondisi  $A = 0, B = 1, C = 1, D = 1$ .      ( $F = A'BCD$ )

## Percobaan 2

1. Buat kombinasi gerbang logika seperti pada gambar!



2. Isi kolom kosong dalam tabel!

Input		Output LED			
SW1	SW2	D1	D2	D3	D4
0	0	1			
0	1				
1	0				
1	1				

3. Masing-masing dioda(LED) menunjukkan hasil output dari rangkaian kombinasi :

$$D1 = SW1 \cdot SW2$$

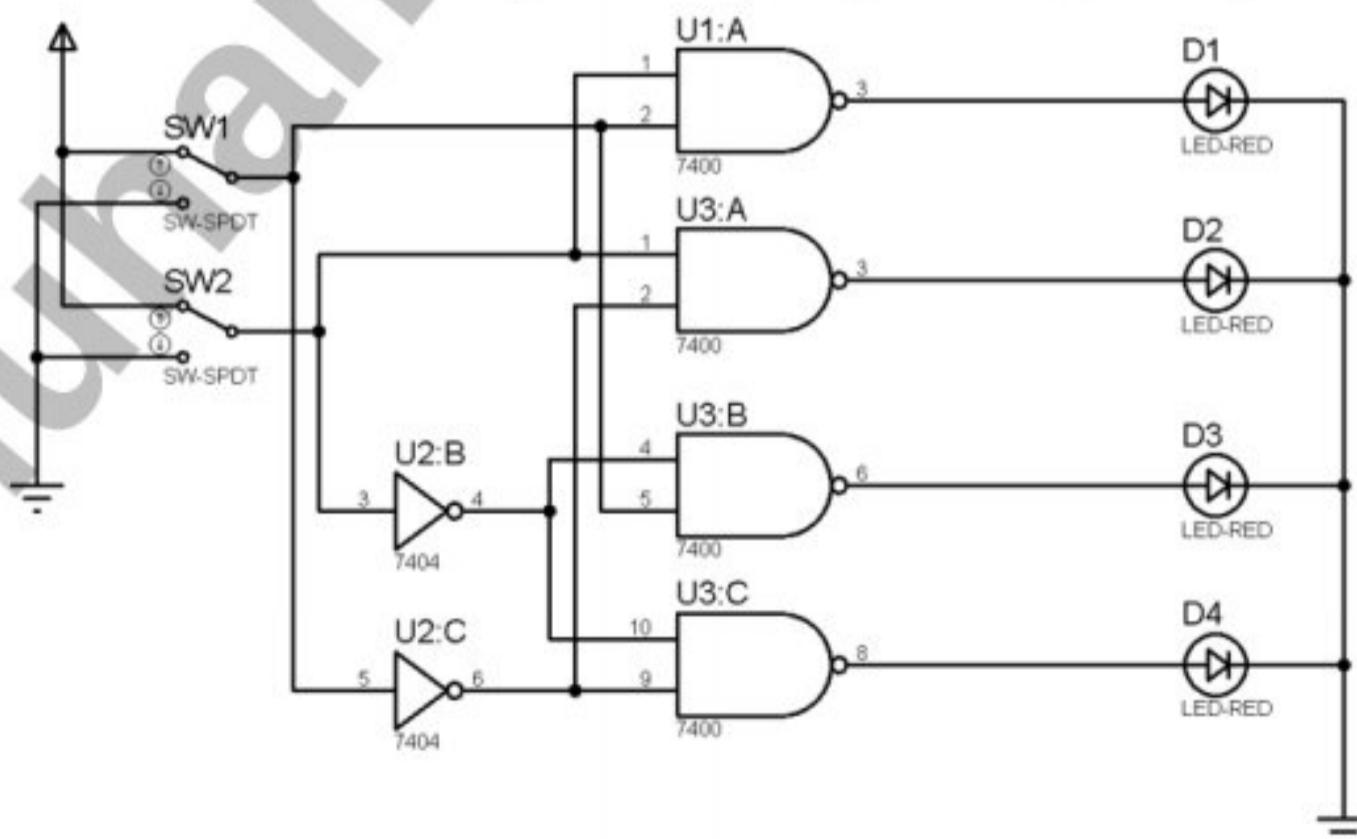
$$D2 = \dots$$

$$D3 = \dots$$

$$D4 = \dots$$

## Percobaan 3

1. Buat kombinasi gerbang logika seperti pada gambar!



2. Isi kolom kosong pada tabel

Input		Output LED			
SW1	SW2	D1	D2	D3	D4
0	0				
0	1				
1	0				
1	1				

3. Masing-masing dioda(LED) menunjukkan hasil output dari rangkaian kombinasi :

D1 = .....

D2 = .....

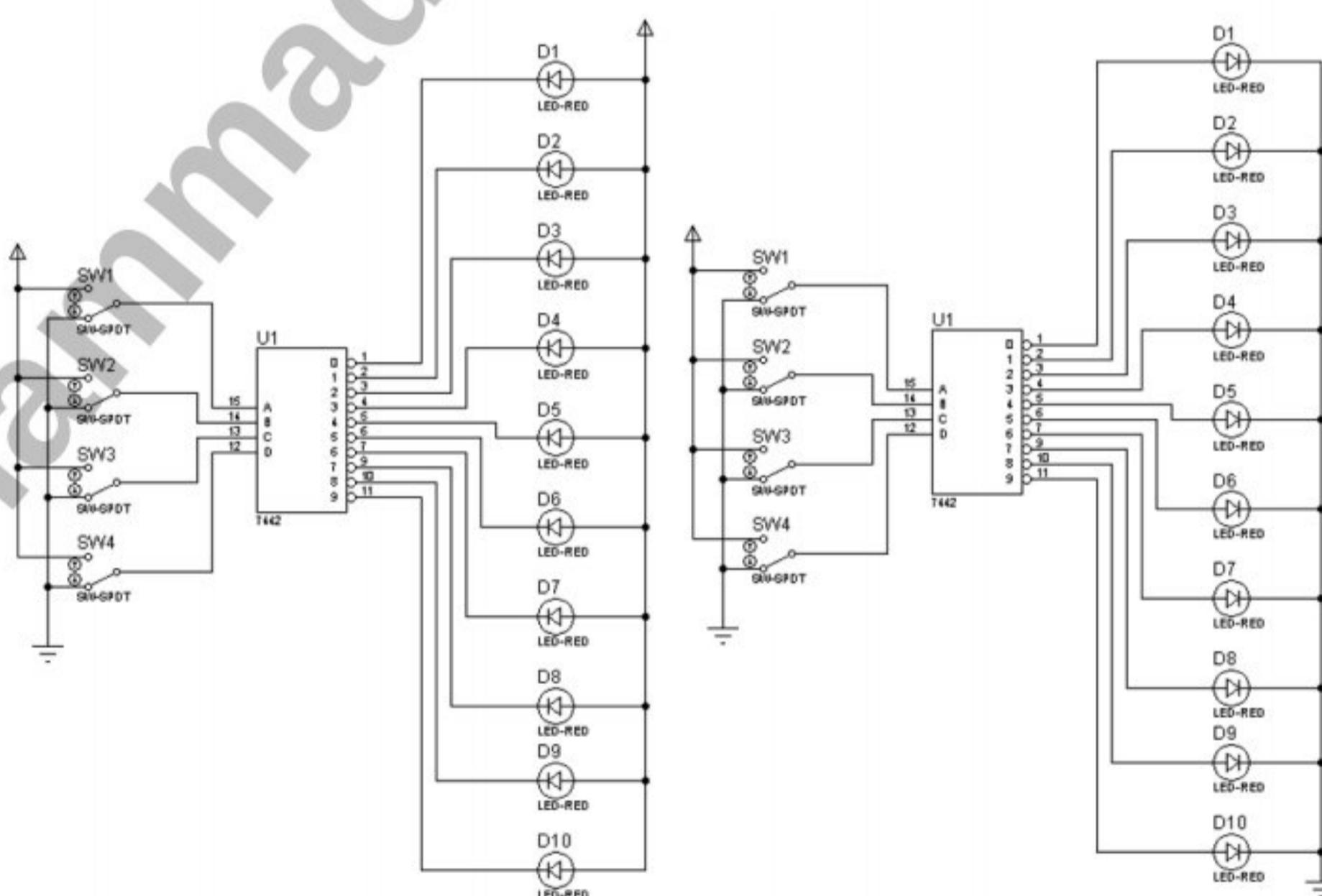
D3 = .....

D4 = .....

4. Bandingkan hasil tabel kebenaran pada percobaan 2 dan percobaan 3! Buat kesimpulan dari percobaan 2 and 3.
- .....  
.....  
.....  
.....

#### Percobaan 4. IC 7442 Decoder BCD-to-decimal

1. Buat rangkaian sebagaimana terlihat pada gambar!gunakan IC 7442 (BCD to decimal decoder)



Decoder 7442 : Common anode LED circuit

Common Cathode LED circuit

2. Isi kolom kosong pada tabel kebenaran decoder 7442 dibawah ini :

a. Rangkaian common anode LED

NO	INPUT				OUTPUT									
	SW4	SW3	SW2	SW1	0	1	2	3	4	5	6	7	8	9
1	0	0	0	0										
2	0	0	0	1										
3	0	0	1	0										
4	0	0	1	1										
5	0	1	0	0										
6	0	1	0	1										
7	0	1	1	0										
8	0	1	1	1										
9	1	0	0	0										
10	1	0	0	1										
11	1	0	1	0										
12	1	0	1	1										
13	1	1	0	0										
14	1	1	0	1										
15	1	1	1	0										
16	1	1	1	1										

b. Rangkaian common cathode LED

NO	Input				Output									
	SW4	SW3	SW2	SW1	0	1	2	3	4	5	6	7	8	9
1	0	0	0	0										
2	0	0	0	1										
3	0	0	1	0										
4	0	0	1	1										
5	0	1	0	0										
6	0	1	0	1										
7	0	1	1	0										
8	0	1	1	1										
9	1	0	0	0										
10	1	0	0	1										

NO	Input				Output								
	SW4	SW3	SW2	SW1	0	1	2	3	4	5	6	7	8
11	1	0	1	0									
12	1	0	1	1									
13	1	1	0	0									
14	1	1	0	1									
15	1	1	1	0									
16	1	1	1	1									

## TUGAS

1. Cari datasheet dari IC 7442! cari schematic yang menunjukan gerbang logika penyusun IC 7442 !

©Muhammadiyah University Press

# MODUL 10

## IMPLEMENTASI DECODER

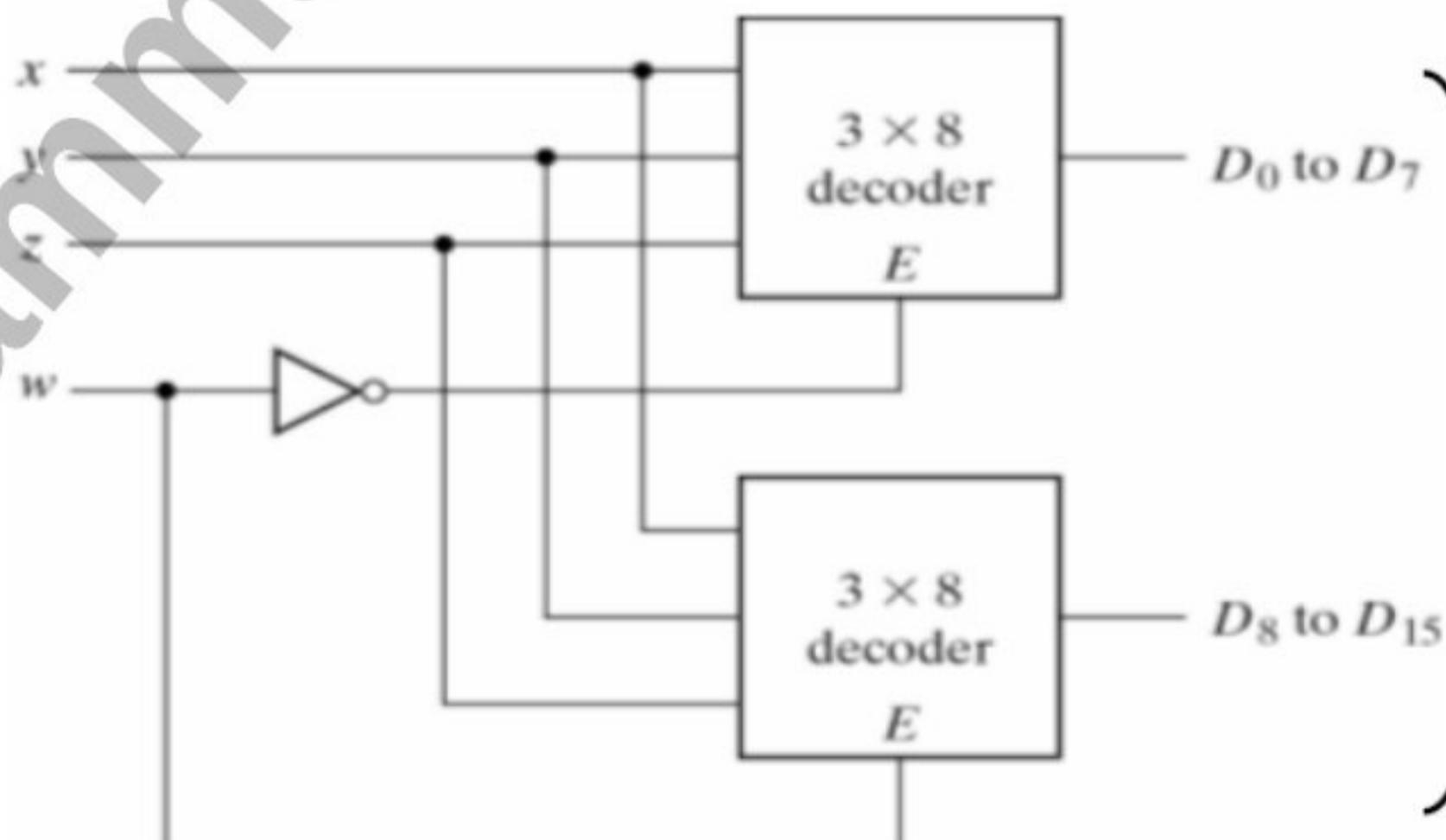
### TUJUAN PRAKTIKUM

Mahasiswa memahami implementasi decoder di dunia nyata.

### TEORI

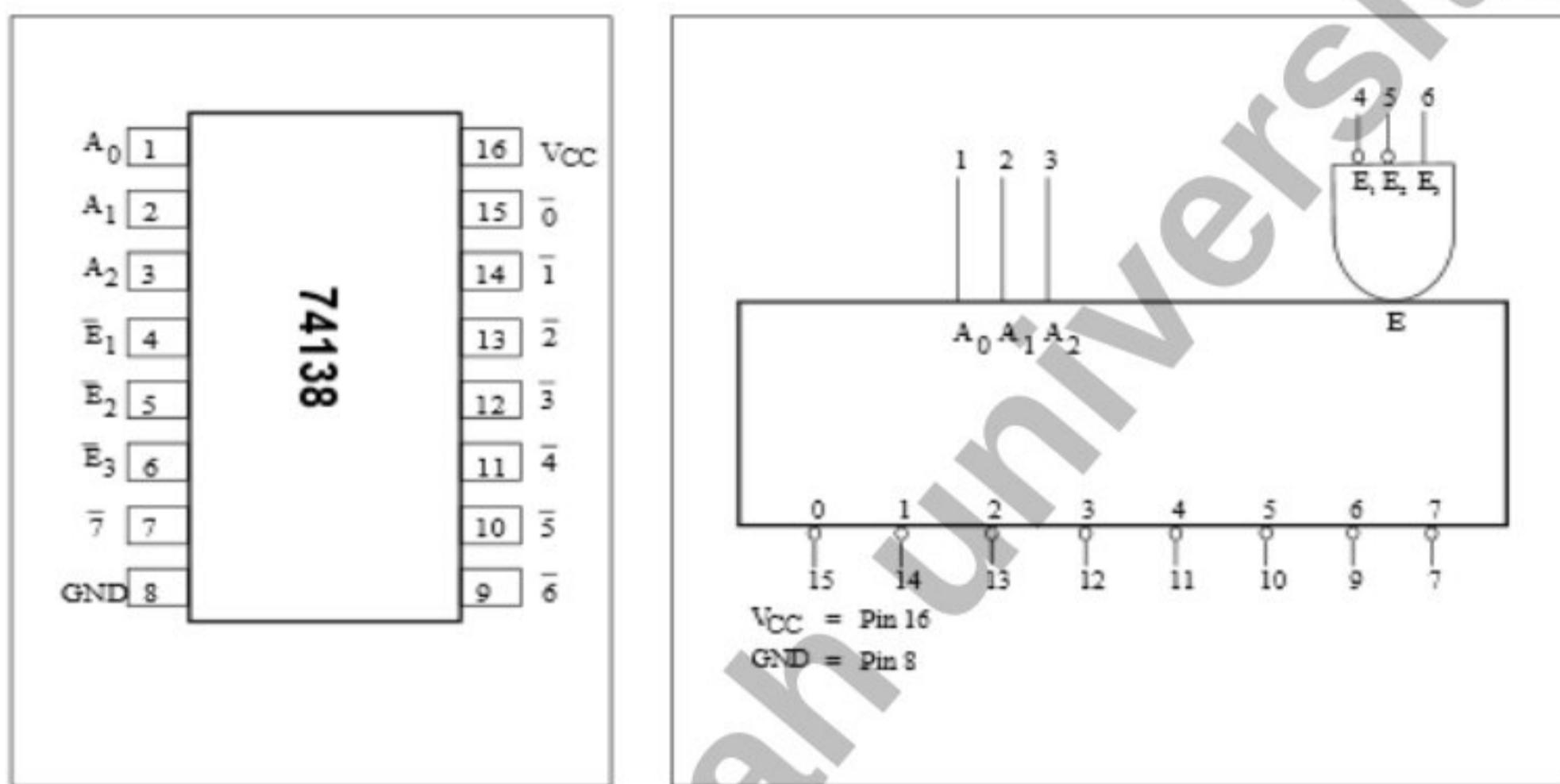
Decoder adalah suatu rangkaian logika kombinasional yang mampu mengubah masukan kode biner n-bit ke m-saluran keluaran sehingga setiap keluaran hanya satu yang akan aktif dari beberapa kemungkinan kombinasi. Setiap n masukan dapat berisi logika 1 atau 0, dan terdapat  $2^N$  kombinasi yang mungkin terjadi dari masukan kode-kode. Dan setiap kombinasi masukan tersebut hanya satu keluaran yang akan aktif atau berlogika 1. Sedangkan keluaran yang lain berlogika 0.

Manfaat decoder yang paling fundamental adalah untuk memudahkan dalam menyalakan seven segment. Seperti pembahasan pada bab sebelumnya, output decoder maksimum adalah  $2^n$ . Sehingga apabila ingin membuat decoder 4 ke 16 dapat menggunakan dua buah decoder 3 ke 8 yang dapat ditunjukkan pada Gambar 10.1. Pada contoh gambar tersebut, hanya satu decoder yang dapat aktif setiap saat. Rangkaian Gambar 10.1 mempunyai empat input yaitu x, y, z dan w dan output sebanyak 16.

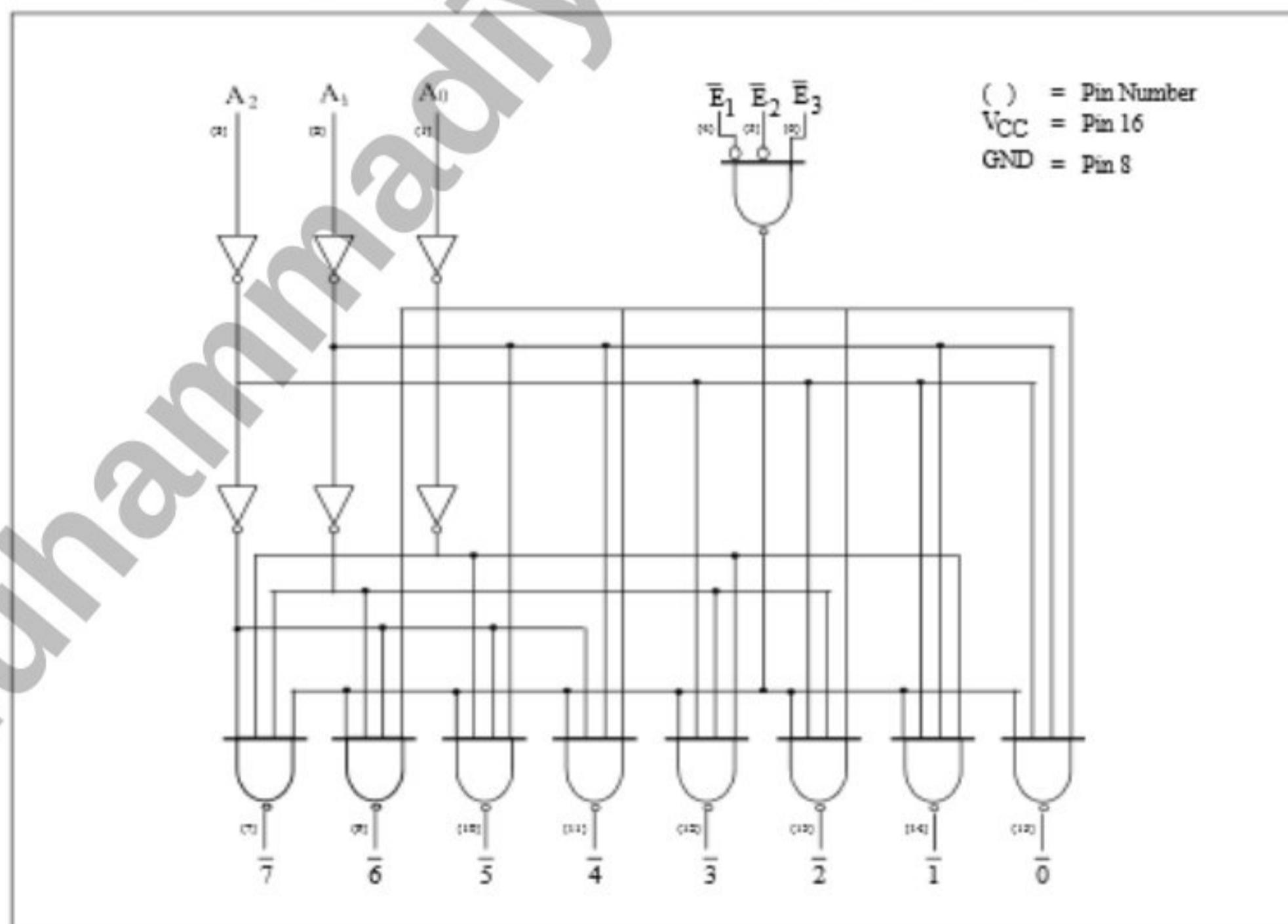


Gambar 10.1. Rangkaian decoder 4 ke 16 yang tersusun dari dua buah decoder 3 ke 8.

Rangkaian decoder yang umumnya ditemui adalah decoder jenis 3x8, decoder jenis 4x16, decoder BCD-to-7 segmen (input 4 bit dan output 8). Karakteristik decoder BCD-to-7 segment sedikit berbeda yaitu kombinasi setiap inputnya dapat mengaktifkan beberapa output. Salah satu jenis IC decoder yang umum dipakai adalah 74138 karena IC ini mempunyai 3 input dan 8 output. IC 74138 adalah sebuah decoder octal yang konfigurasi pin dan simbolnya ditunjukkan pada Gambar 10.2. Disamping itu diagram logika dalam kinerja IC 74138 diperlihatkan pada Gambar 10.3. Tabel 1 merupakan tabel kebenaran untuk IC 74138. Decoder untuk tipe perangkat lain diantaranya adalah 7442 merupakan decoder BCD 4 ke 10, 74154 merupakan decoder hex 4 ke 16, dan 7447 yang merupakan decoder BCD-to-7 segmen.



Gambar 10.2. Konfigurasi pin (kiri) dan simbol logika (kanan) untuk IC 74138.



Gambar 10.3. Diagram logika IC 74138

Tabel 10.1. Tabel kebenaran dari IC 74138

Inputs						Outputs							
		E3	A0	A1	A2	D0	D1	D2	D3	D4	D5	D6	D7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

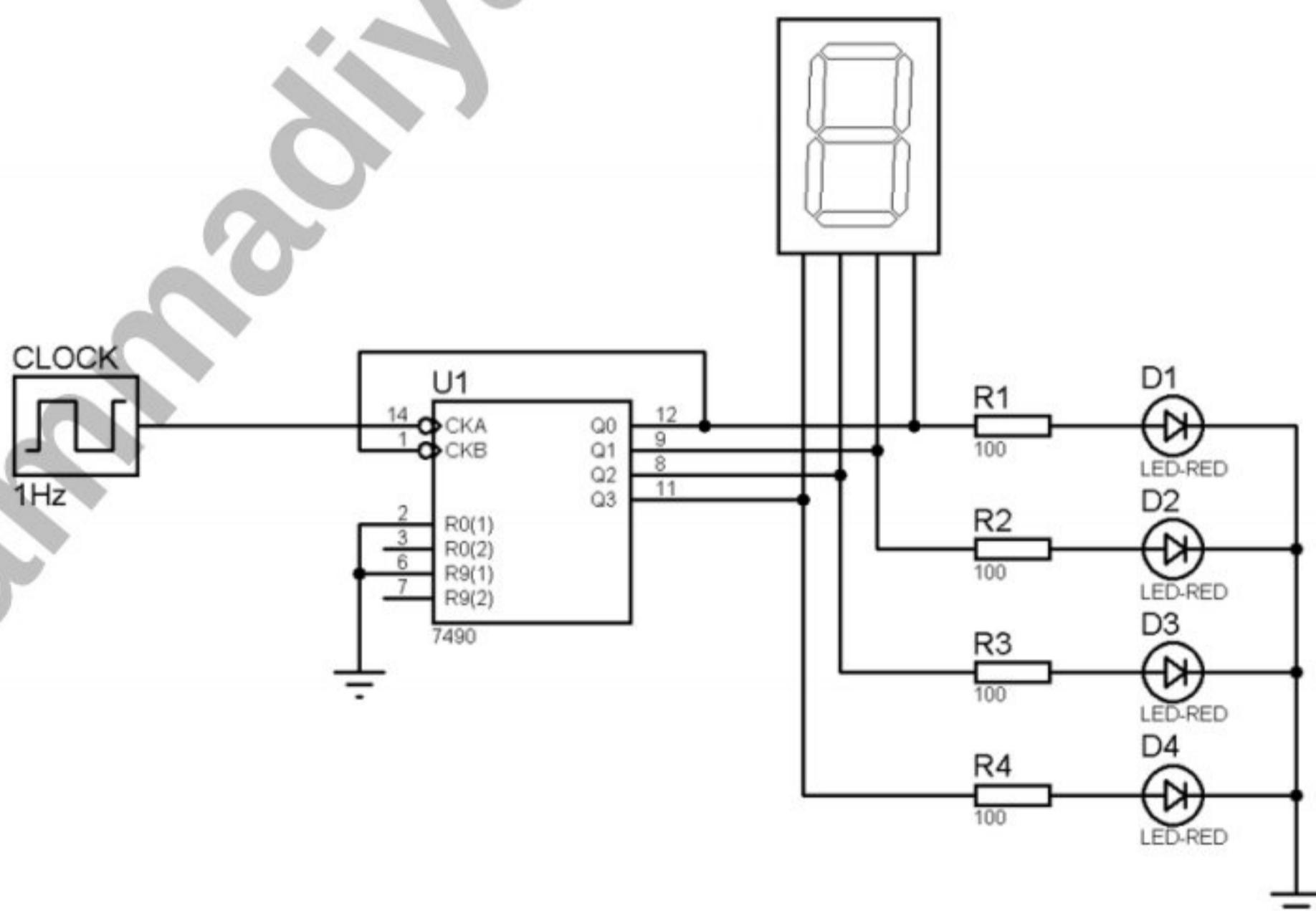
H = level tegangan high, L = level tegangan low, dan X = high/low

## KEGIATAN PRAKTIKUM

### Percobaan 1. Rangkaian Clock Counter

- Buat Counter dibawah!

Gunakan IC 7490 (Decade Counter), 7segment(7seg-BCD-red), resistor (res)



2. Isi kolom kosong pada Tabel!

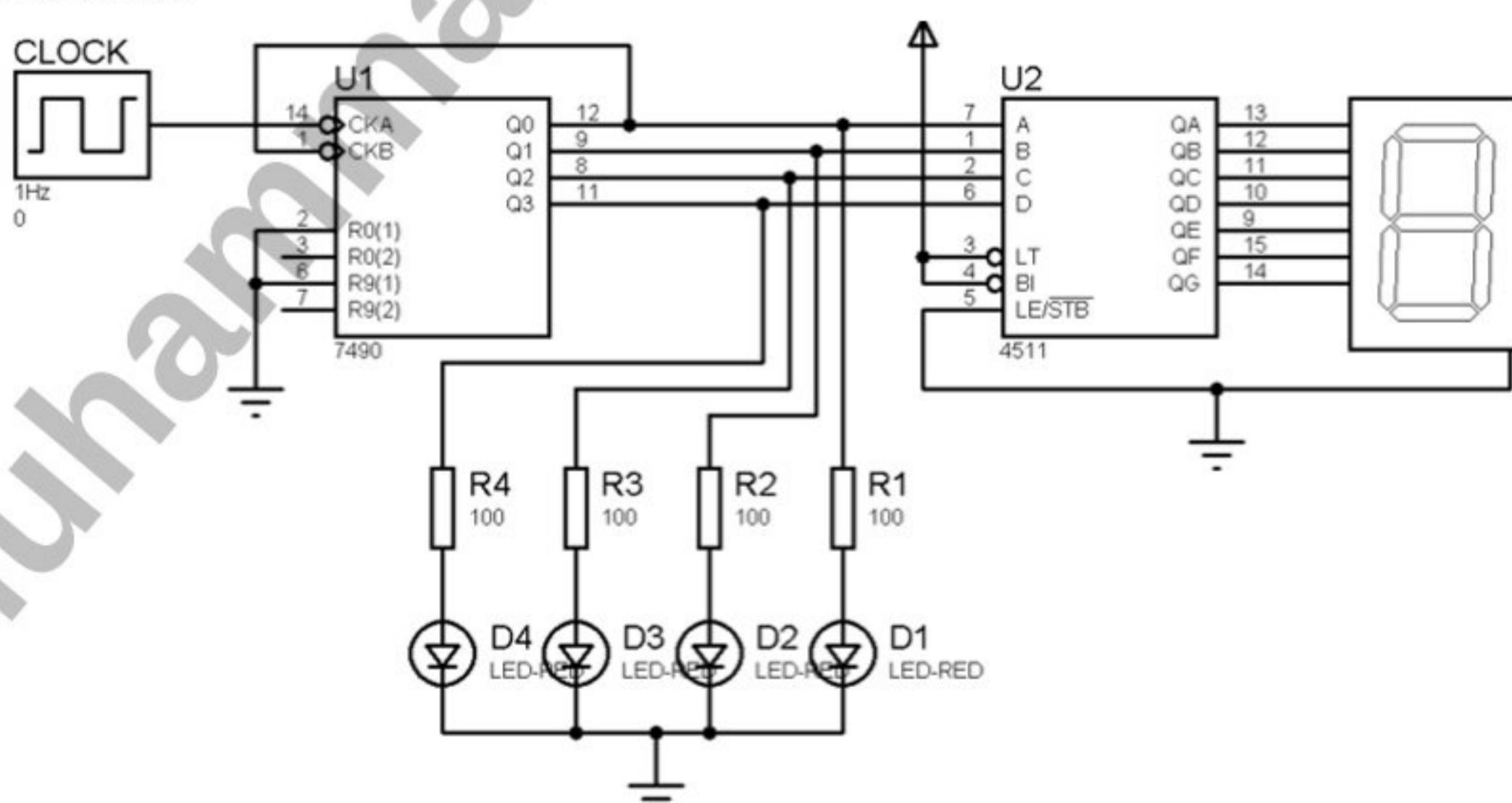
Input Clock	Output LED				Output Seven Segment
	D1	D2	D3	D4	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

3. Tugas : cari referensi datasheet mengenai IC 7490! Lihat gerbang logika penyusunnya dan laporkan ke dosen pengampu / asisten praktikum!

### Percobaan2. Penambahan Decoder BCD-to-7segment

1. Buat rangkaian seperti pada percobaan 1

Tambahkan rangkaian dengan IC 4511 dan 7segment common cathode!

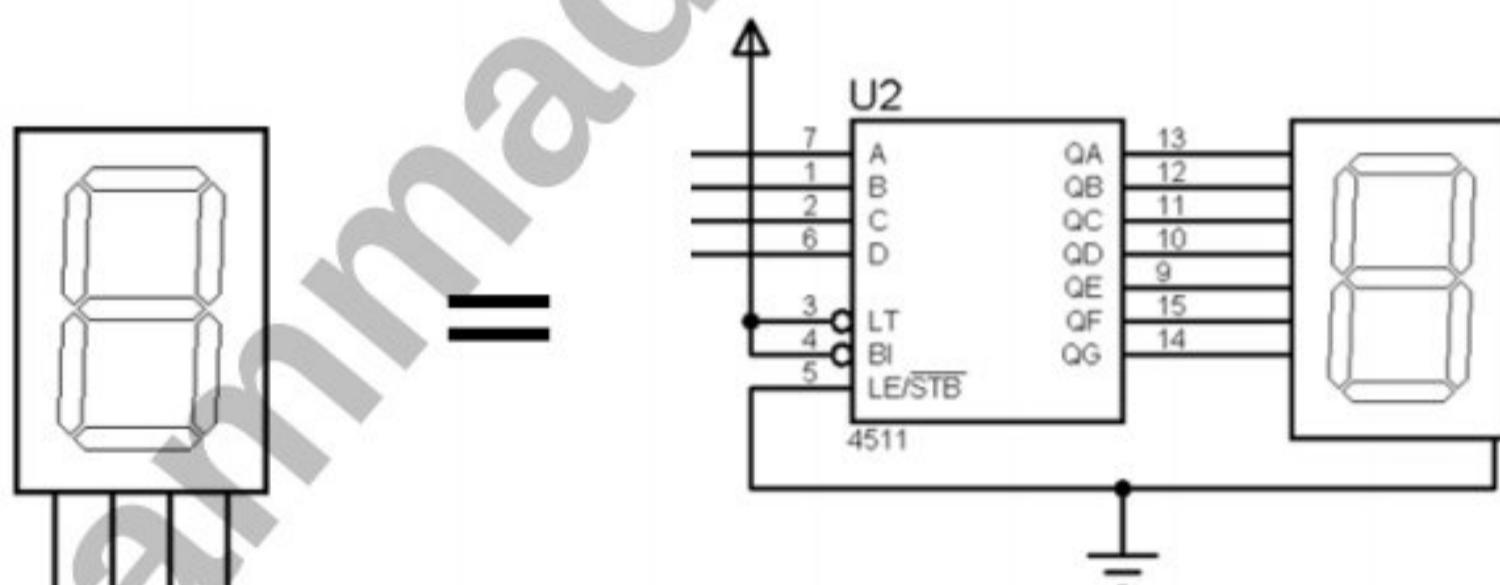


2. Isi kolom kosong pada tabel!

Input Clock	Output LED				Output Seven Segment
	D1	D2	D3	D4	
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

3. Bandingkan percobaan 1 dan percobaan 2! Dapatkan anda melihat persamaannya?
- .....  
.....  
.....

4. Apakah benar bahwa 7seg-BCD sama dg BCD-to-7segment decoder?  
(...Yes... / ...No...)

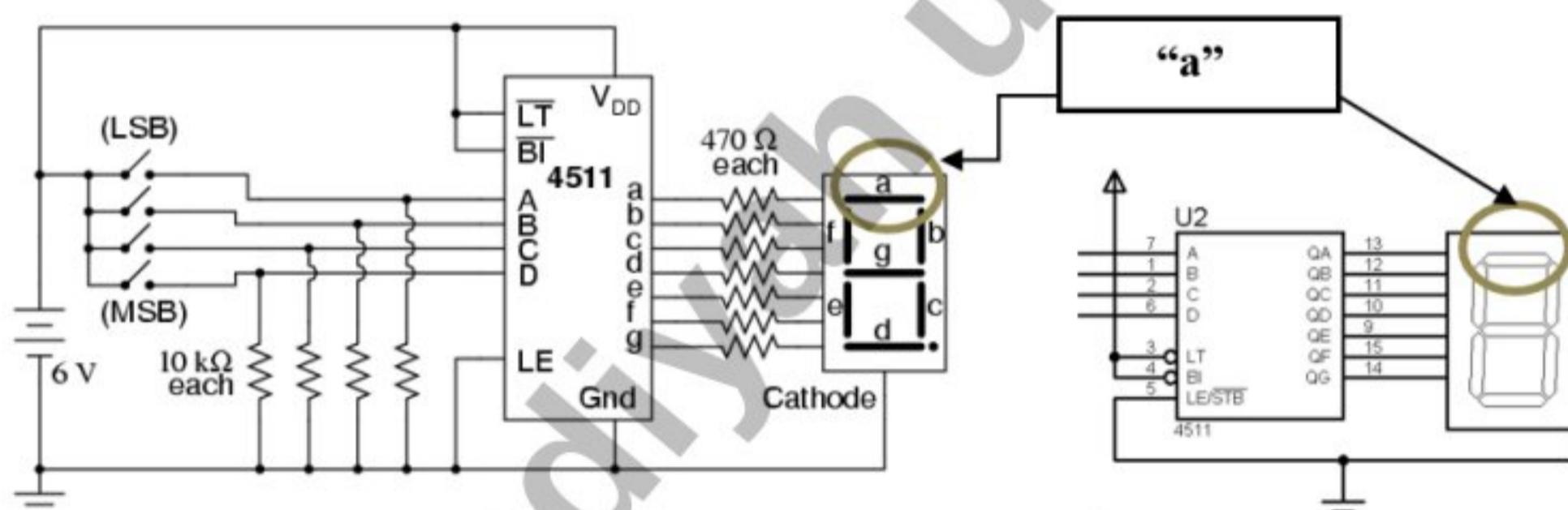


### Percobaan 3. Melihat di dalam BCD-to-7segment Decoder

- Perhatikan fungsi tabel IC 4511

Decimal Digit	Input					Output							Display Output
	LT	D	C	B	A	a	b	c	d	e	f	g	
0	H	L	L	L	L	H	H	H	H	H	H	L	0
1	H	L	L	L	H	L	H	H	L	L	L	L	1
2	H	L	L	H	L	H	H	L	H	L	H	H	2
3	H	L	L	H	H	H	H	H	H	L	L	H	3
4	H	L	H	L	L	L	H	H	L	L	H	H	4
5	H	L	H	L	H	H	L	H	H	L	H	H	5
6	H	L	H	H	L	L	L	H	H	H	H	H	6
7	H	L	H	H	H	H	H	H	L	L	L	L	7
8	H	H	L	L	L	H	H	H	H	H	H	H	8
9	H	H	L	L	H	H	H	H	L	L	H	H	9
LT	L	X	X	X	X	H	H	H	H	H	H	H	8

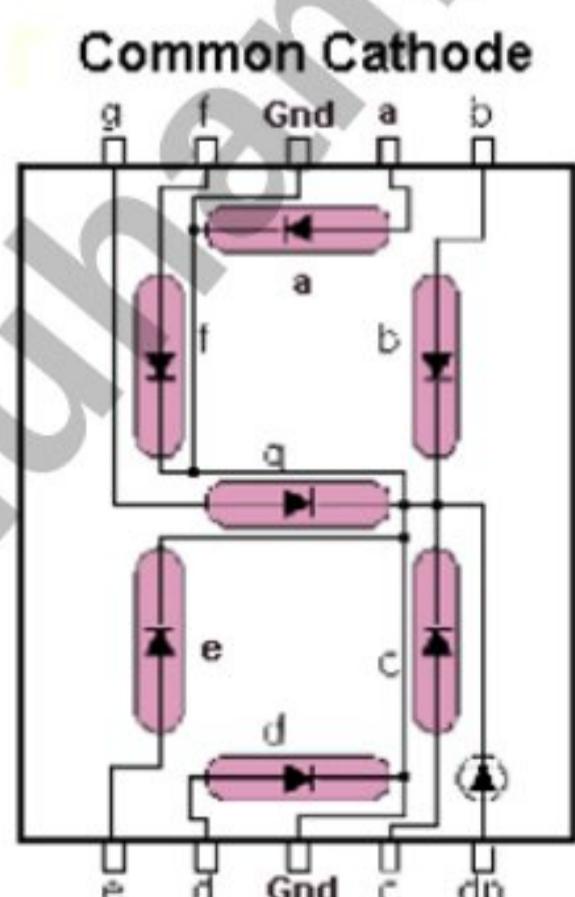
- Output "a" (highlight) pada tabel menunjukkan kerjanya LED di seven segment Common cathode dibawah.



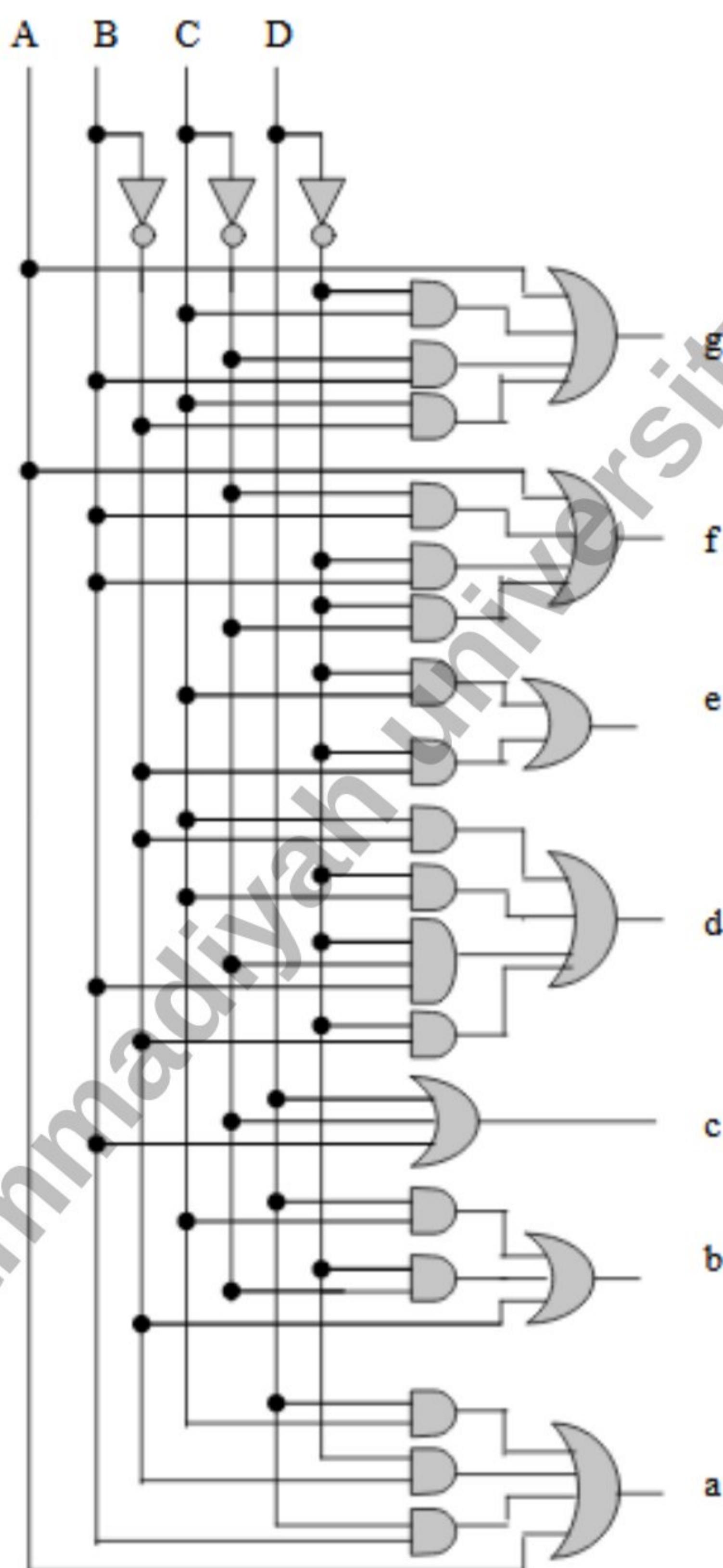
output "a" pada teori

output "a" pada percobaan anda

- Tiap output menunjukkan keadaan LED dari seven segment berbagai kondisi



4. Masing-masing LED dikendalikan oleh kombinasi gerbang logika. Diagram logic lengkap dari decoder BCD-to-7segment ditunjukkan pada gambar berikut ini.



## TUGAS

Coba buat rangkaian decoder BCD-to-7Segmen di atas, dan bandingkan dengan tabel kebenaran pada point no 1!

©Muhammadiyah University Press

# MODUL 11

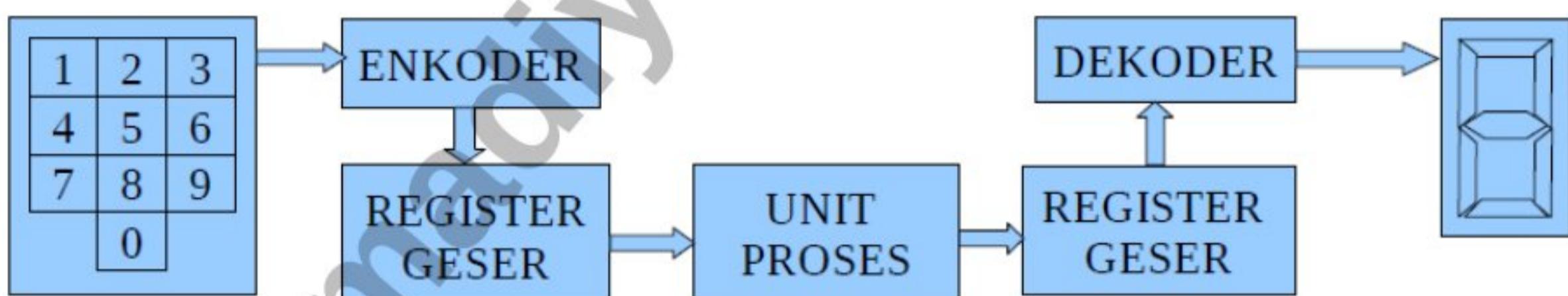
## REGISTER

### TUJUAN PRAKTIKUM

Mahasiswa mampu memahami bagaimana rangkaian register bekerja.

### TEORI

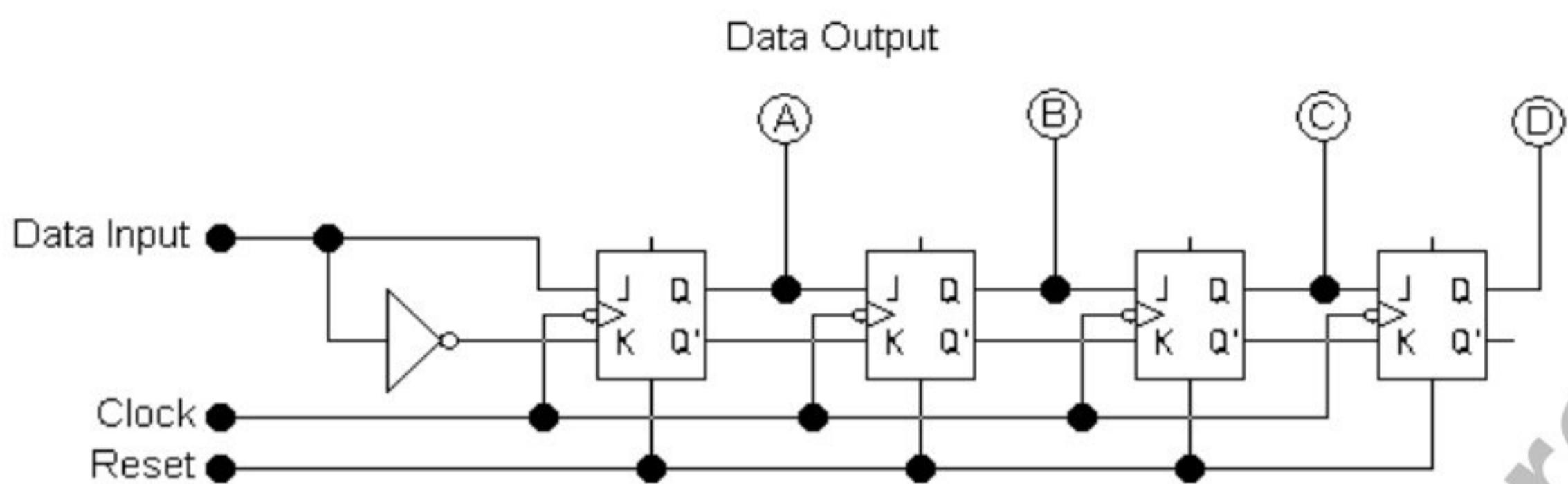
Dalam percobaan sebelumnya telah diperlihatkan bahwa flip-flop mempunyai karakteristik sebagai penyimpanan memori. Karakteristik memori yang digunakan untuk membentuk sebuah register geser (shift register). Register geser dikembangkan menggunakan flip-flop. Fungsi register geser adalah untuk menyimpan data sementara. Gambar 11.1 menunjukkan penggunaan register yang dapat ditemukan di mesin kalkulator. Sebagai catatan, register geser digunakan untuk menyimpan informasi encoder sebelum dikirim ke unit pengolahan. Register geser juga digunakan sebagai tempat penyimpanan sementara sebelum hasil pengolahannya dikirim ke decoder.



Gambar 11.1. Contoh penggunaan register geser pada mesin kalkulator

### Register Geser Seri

Register geser beban seri mempunyai rangkaian register dasar seperti terlihat pada Gambar 2. Register geser ini terdiri dari empat buah JK-FF, sehingga dikenal juga sebagai register geser 4-bit yang dapat menyimpan empat buah data 1-bit : A, B, C, dan D. Sistem kerja dari register geser ini dapat dijelaskan melalui rangkaian pada Gambar 11.2.



Gambar 11.2. Register geser beban seri

Tabel 11.1. Tabel operasi register geser beban seri

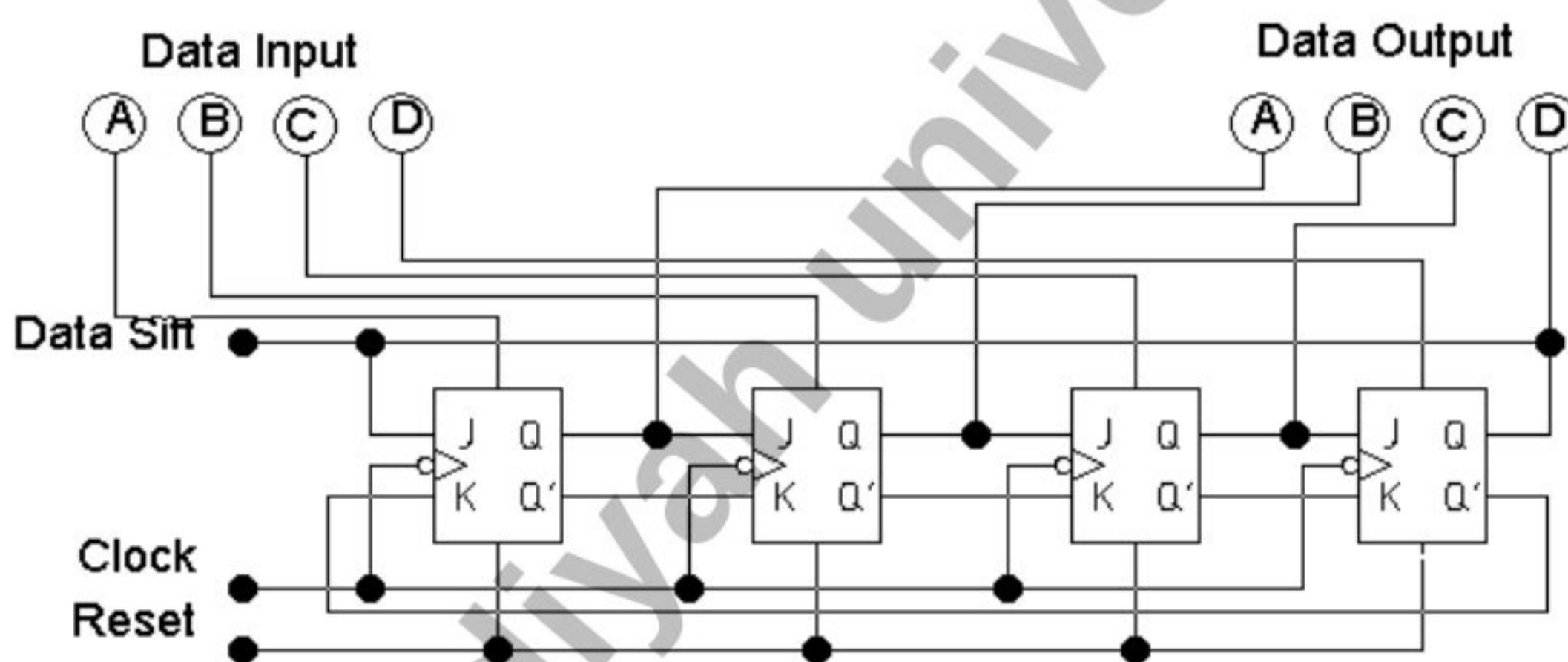
#	Input			Output			
	Reset	Data Input	Clock	A	B	C	D
1	0	x	-	0	0	0	0
2	1	1	-	1	0	0	0
3	1	1	1	1	1	0	0
4	1	1	2	1	1	1	0
5	1	1	3	1	1	1	1
6	1	0	4	0	1	1	1
7	1	0	5	0	0	1	1
8	1	0	6	0	0	0	1
9	1	0	7	0	0	0	0
10	1	0	8	0	0	0	0
11	1	1	9	1	0	0	0
12	1	0	10	0	1	0	0
13	1	0	11	0	0	1	0
14	1	0	12	0	0	0	1
15	1	0	13	0	0	0	0

Semua output Q flip flop pada Tabel 1 terjadi setelah terdapat pulsa clock. Karena data yang memasuki register geser satu per satu, sehingga disebut dengan register geser beban seri. Cara yang lain yaitu dengan memasukkan data secara pararel, yang akan dijelaskan pada sub bab berikutnya. Data sebanyak 4 bit dan muncul pada entri yang sama pada output setiap flip-flop. Proses ini dilakukan dalam pulsa clock tunggal. Arah pergeseran data pada penjelasan di atas adalah ke kanan, sehingga dikenal

dengan register geser kanan. Dalam percobaan juga akan ditunjukkan bagaimana data dapat bergeser ke kiri.

### Register Geser Pararel

Pada register geser beban seri, data hanya 1 bit. Sehingga untuk data 4 bit, contohnya pada IC 1101, data harus dimasukkan secara bergantian dengan urutan, 1-clock-0-clock-1-clock-1-clock-1-clock. Apabila clock tidak berhenti maka data akan terus bergeser dan akhirnya menghilang. Gambar 11.3 memperlihatkan sebuah blok register geser yang memungkinkan memasukkan data ke dalam register 4-bit sekaligus hanya dengan satu pulsa clock. Dengan proses berulang (recycling), output data flip-flop terakhir dapat kembali ke dalam flip flop. Hal ini dilakukan bertujuan untuk mencegah kehilangan data. Gambar 11.3 juga memperlihatkan daur ulang proses recycling register menggunakan JK flip flop. Tabel 2 membantu pemahaman dalam menjelaskan register geser bekerja.



Gambar 11.3. Diagram register geser pararel 4 bit (recycling)

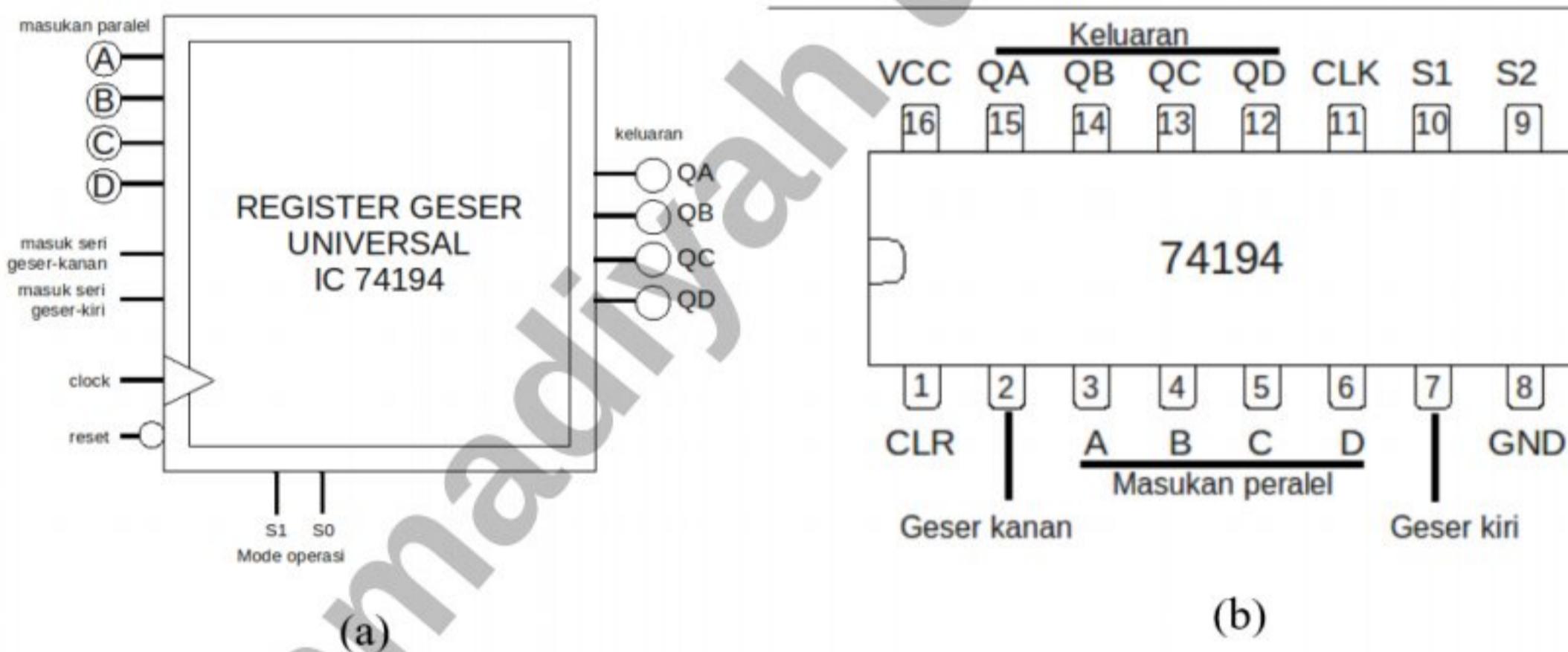
Tabel 11.2. Tabel operasi register geser pararel 4 bit (recycling)

#	Masukan					Clock	Keluaran			
	Reset	A	B	C	D		A	B	C	D
1	1	1	1	1	1	---	x	x	x	x
2	0	1	1	1	1	---	0	0	0	0
3	1	1	1	0	1	---	0	0	1	0
4	1	1	1	1	1	1	0	0	0	1
5	1	1	1	1	1	2	1	0	0	0
6	1	1	1	1	1	3	0	1	0	0
7	1	1	1	1	1	4	0	0	1	0

#	Masukan					Clock	Keluaran			
	Reset	A	B	C	D		A	B	C	D
8	1	1	1	1	1	5	0	0	0	1
9	0	1	1	1	1	---	0	0	0	0
10	1	1	0	0	1	---	0	1	1	0
11	1	1	1	1	1	6	0	0	1	1
12	1	1	1	1	1	7	1	0	0	1
13	1	1	1	1	1	8	1	1	0	0
14	1	1	1	1	1	9	0	1	1	0
15	1	1	1	1	1	10	0	0	1	1
16	0	1	1	1	1	---	0	0	0	0

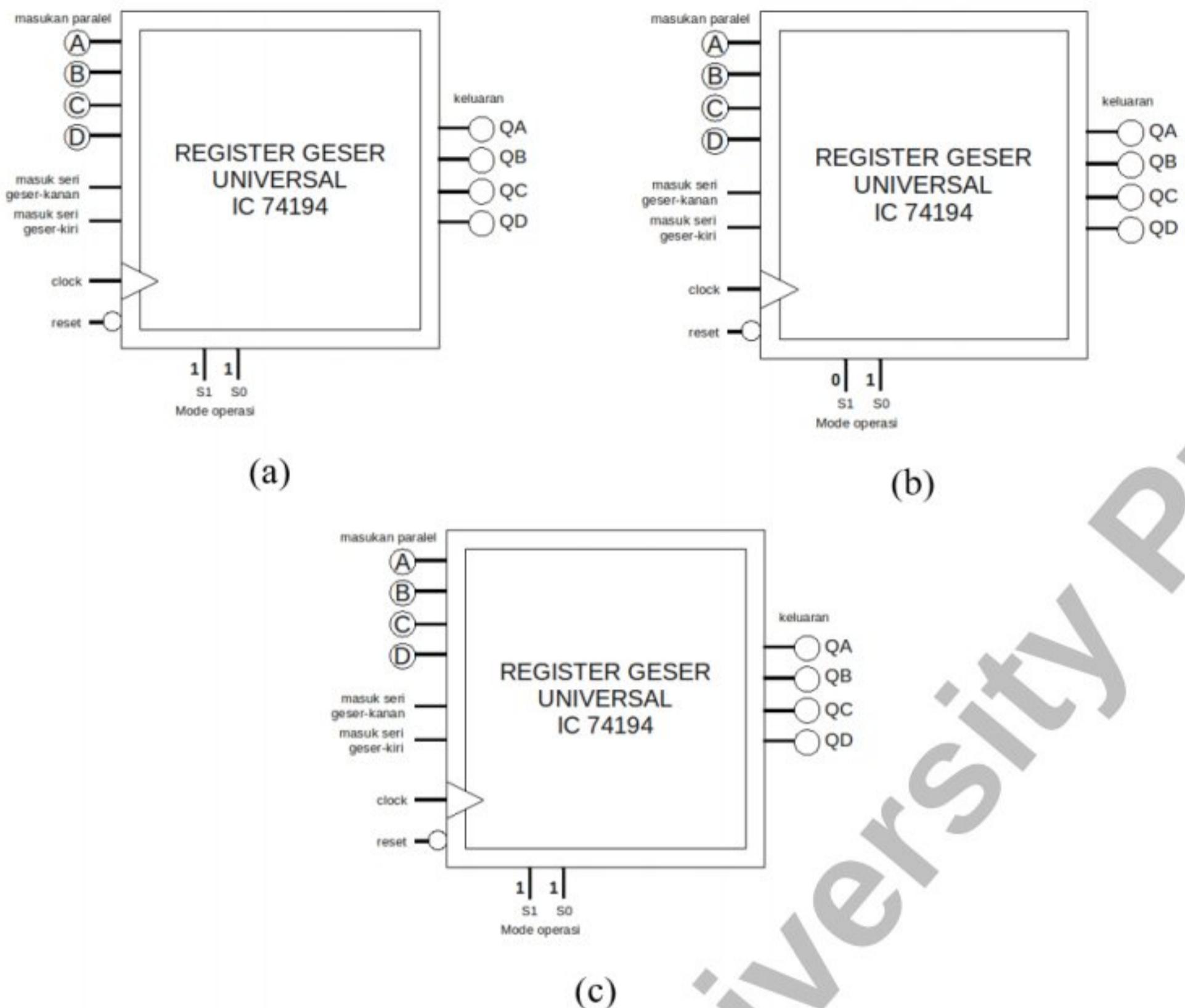
### Register Geser Universal IC 74194

Register jenis ini mampu melakukan entri data seri dan pararel, data bergeser ke kanan, dan data bergeser ke kiri. Salah satu jenisnya adalah register geser IC 74 194. Gambar 11.4(a) memperlihatkan gambar IC 74 194 dan fungsi blok diagram ditunjukkan pada Gambar 11.4(b).



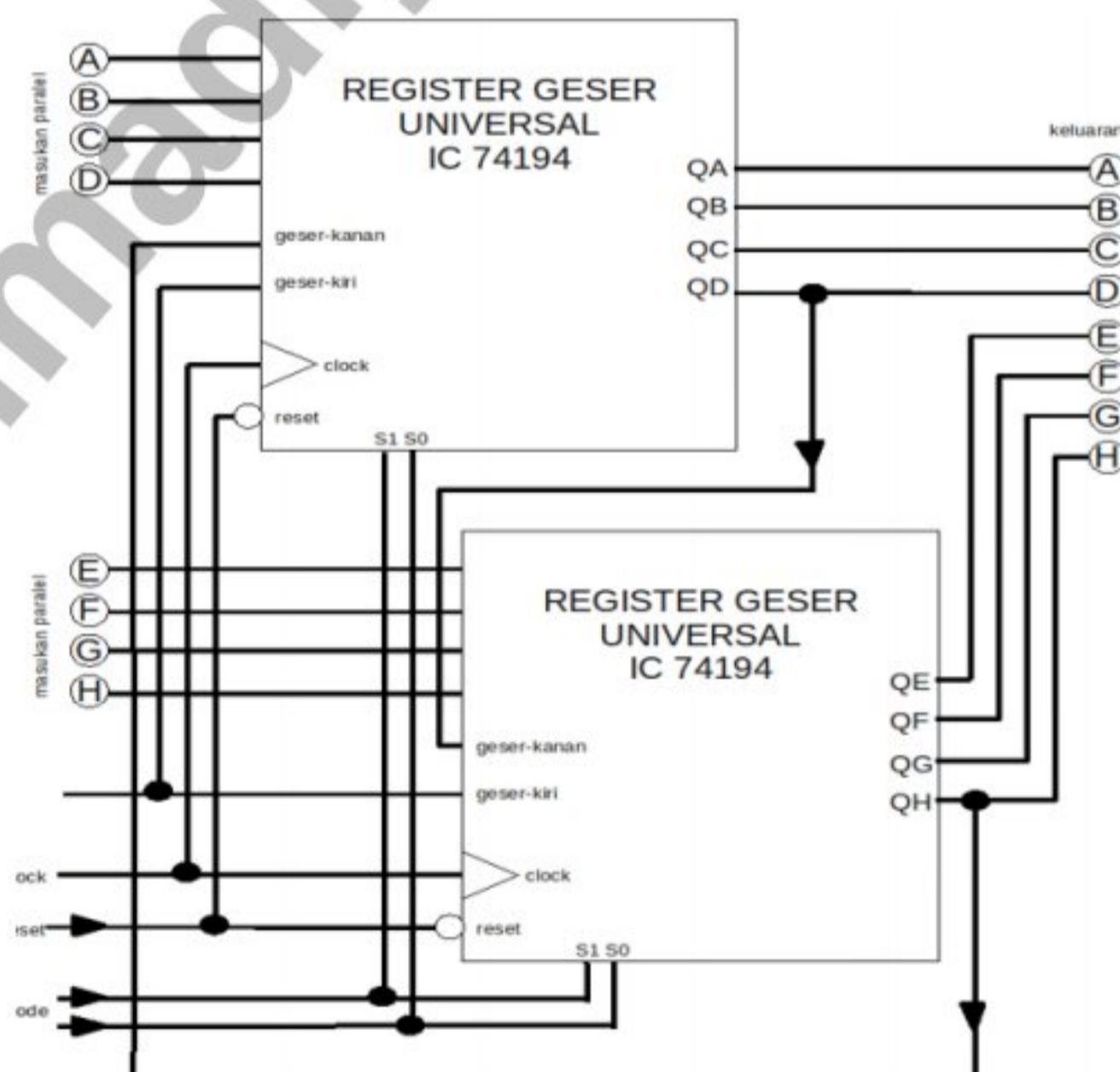
Gambar 11.4. Register geser universal. (a) IC 74 194. (b) blok diagram IC 74 194.

Untuk mengetahui bagaimana register ini bekerja dapat melihat dan mempelajari blok simbol dan pin dari IC tersebut. Data input sebesar 4 bit, dan data output juga sebesar 4 bit. Data yang dimasukkan ke dalam register dapat seri atau pararel. Pergeseran data dikenalikan oleh S0 dan S1. Sinkronisasi terjadi pada clock transisi dari 0 ke 1. Gambar 11.5 memperlihatkan operasi register dengan memperhatikan fungsi kaki-kaki IC pada masing-masing operasi.



Gambar 11.5. Operasi register geser universal. (a) pemasukan data serial, S1 = 1, S0 = 0. (b) pemasukan data serial S1 = 0, S0 = 1. (c) pemasukan data serial S1 = 1, S0 = 1.

Beberapa register 4 bit dapat ditumpuk untuk membentuk register IC 74194 8 bit atau lebih. Gambar 11.6 memperlihatkan dua buah IC 74 194 yang disusun untuk membentuk register geser 8 bit, dengan data dapat bergeser ke kanan dan berulang (recycling). Diperlukan tabel kebenaran dan diagram waktu agar dapat memahami dengan benar cara kerja dari register ini.



Gambar 11.6. Register geser 8 bit.

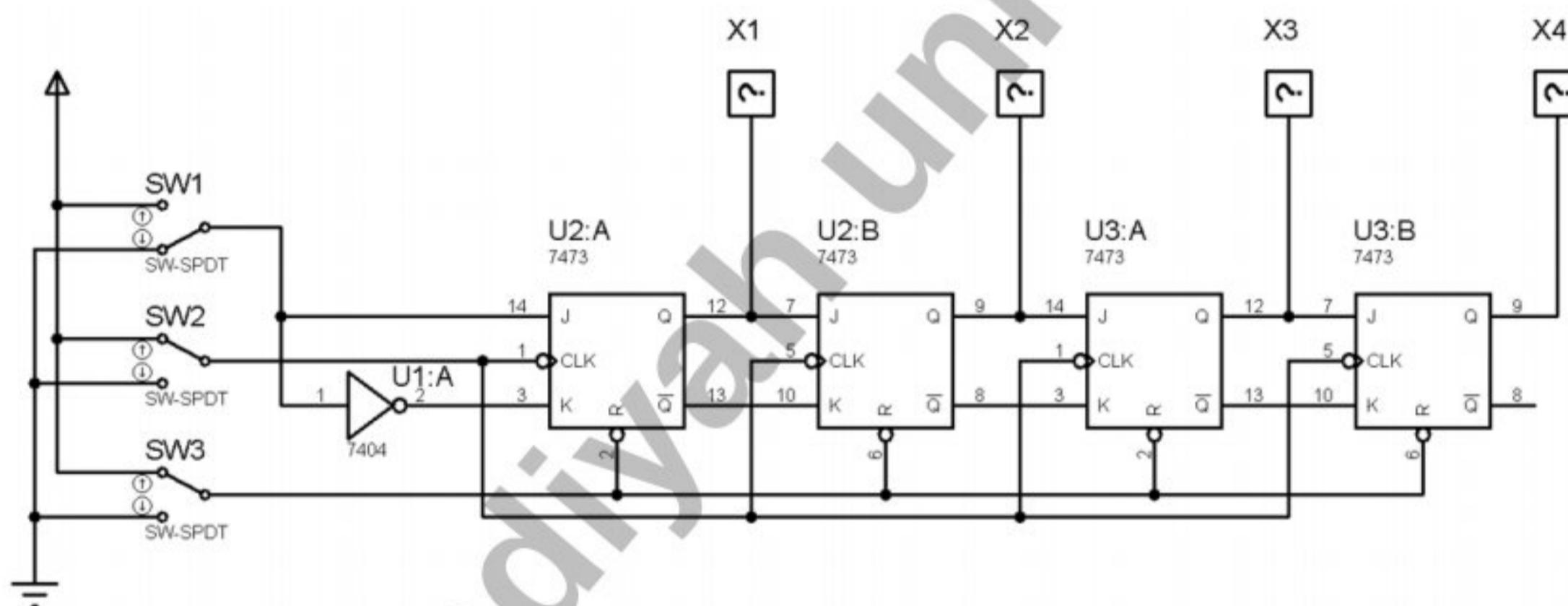
Tabel 11.3. Tabel register geser 8 bit

reset	mode		seri		clock	paralel				QA	QB	QC	QD
	S1	S0	kiri	kanan		A	B	C	D				
0	x	x	x	x	x	x	x	x	x	0	0	0	0
1	x	x	x	x	0	x	x	x	x	QA	QB	QC	QD
1	0	0	x	x	↑	x	x	x	x	QA	QB	QC	QD
1	1	1	x	x	↑	a	b	c	d	a	b	c	d
1	0	1	x	1	↑	x	x	x	x	1	QA	QB	QC
1	0	1	x	0	↑	x	x	x	x	0	QA	QB	QC
1	1	0	1	x	↑	x	x	x	x	QB	QC	QD	1
1	1	0	0	x	↑	x	x	x	x	QB	QC	QD	0

## KEGIATAN PRAKTIKUM

### Percobaan 1. Rangkaian Register 4-bit

- Buat rangkaian register dengan menggunakan JK-FF



- Jalankan Simulasi!
- Reset register geser dilakukan dengan mengaktifkan saklar SW3 (*open* kemudian *close* lagi).
- Set saklar SW1 pada posisi biner 1.
- Berikanlah 4 buah pulsa geser (0-1-0) melalui saklar SW2, amati keadaan register melalui kondisi PROBE.
- Catatlah bilangan biner dari isi register setelah 4 pulsa geser diberikan.  
**ABCD = .....**
- Selanjutnya, set saklar SW1 pada posisi biner 0, dan berikan lagi 4 buah pulsa geser melalui saklar SW2. Dan catatlah isi register.  
**ABCD = .....**

8. Dengan menggunakan saklar SW1 dan saklar SW2.

Berikan muatan pada register geser tahap demi tahap melalui prosedur dibawah ini:

SW1 = 1, kemudian berikan satu pulsa dari saklar SW2

SW1 = 0, kemudian berikan satu pulsa dari saklar SW2

SW1 = 1, kemudian berikan satu pulsa dari saklar SW2

SW1 = 0, kemudian berikan satu pulsa dari saklar SW2

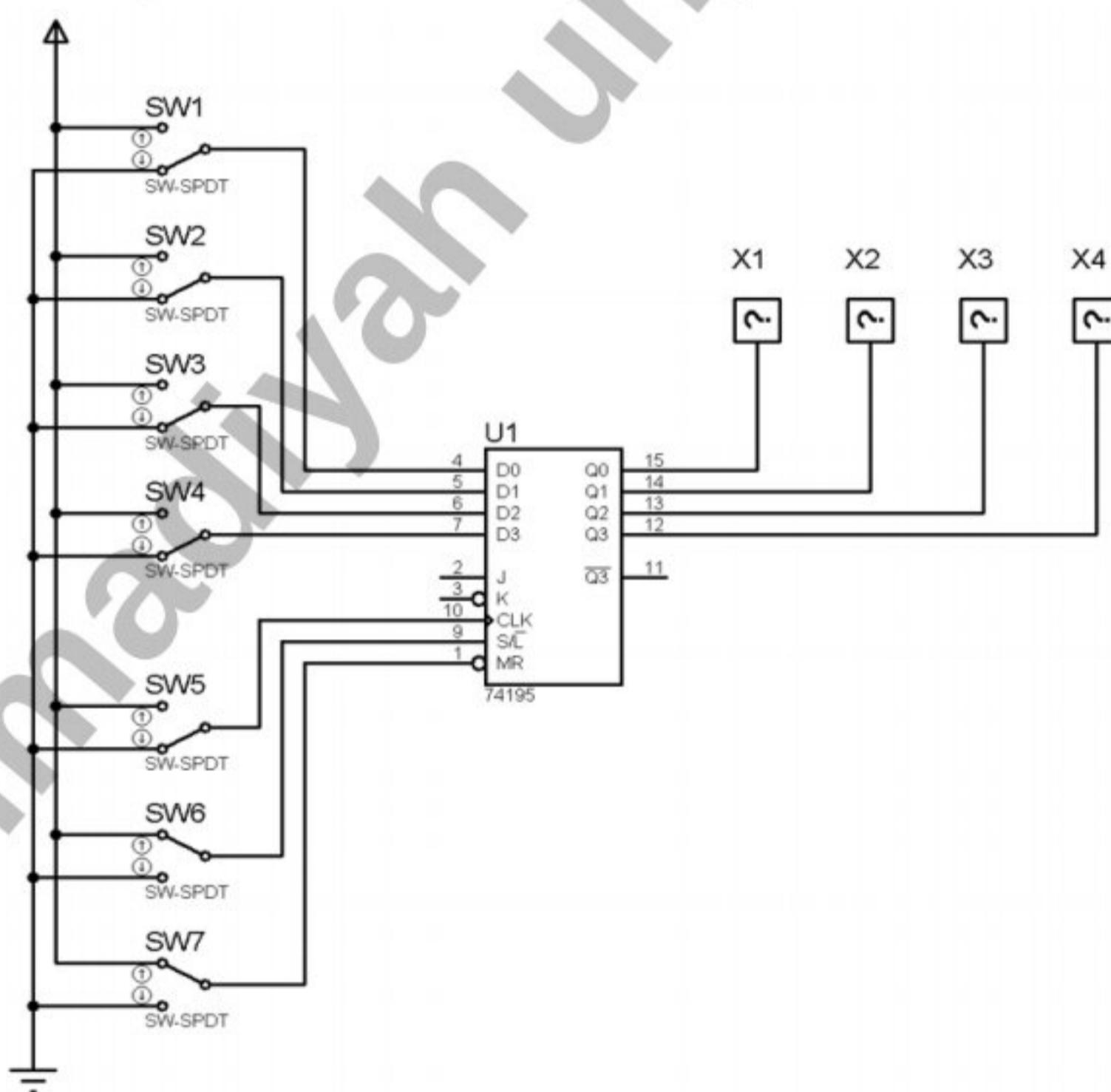
9. Setelah itu amatilah keadaan PROBE dan tulislah bilangan desimal yang ekuivalen dengan bilangan biner dalam register geser.

Bilangan Biner = ..... Bilangan Desimal = .....

### Percobaan 2. Rangkaian Register IC 74194

1. Buatlah rangkaian register geser seperti pada gambar, IC yang dipakai adalah 74195N yang telah siap pakai, yang akan mengurangi hubungan antar FF untuk membentuk sebuah register geser.

Fleksibilitas terminal masukan-keluaran memungkinkan bagi komponen ini dipakai untuk berbagai fungsi.



2. Saklar data SW1, SW2, SW3 dan SW4 dipakai sebagai sumber data paralel.

Saklar SW5 dipakai untuk membangkitkan pulsa geser.

Saklar SW6 berfungsi sebagai mode control untuk rangkaian.

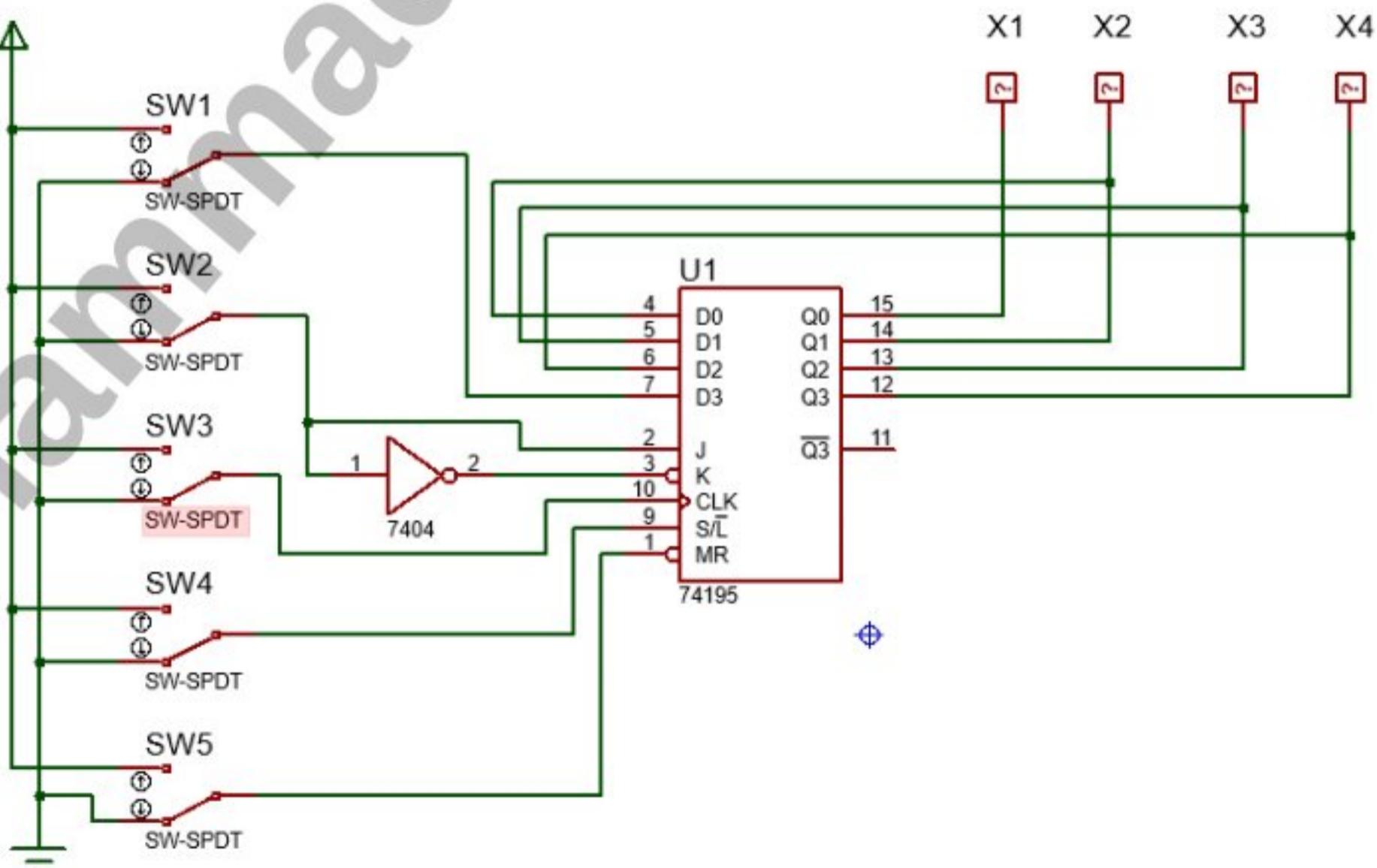
Saklar SW7 berfungsi sebagai reset data

3. Set semua saklar data (SW1 sampai SW4) ke biner 0.
4. Set saklar SW6 ke biner 0, selanjutnya saklar SW5 diset ke 0 kemudian set ke 1 lagi (memberikan pulsa clock), kemudian kembalikan saklar SW6 ke biner 1 lagi. Catatlah keadaan PROBE X1, X2, X3 dan X4.  
**ABCD = .....**
5. Selanjutnya, set semua saklar data (SW1 sampai SW4) ke biner 1.
6. Set saklar SW6 ke biner 0, selanjutnya saklar SW5 diset ke 0 kemudian set ke 1 lagi (memberikan pulsa clock), kemudian kembalikan saklar SW6 ke biner 1 lagi. Catatlah keadaan PROBE X1, X2, X3 dan X4.  
**ABCD = .....**
7. Berikan pulsa clock (set 0 kemudian set 1 lagi) pada saklar SW5 sebanyak 4 kali dan amatilah keadaan PROBE setelah anda memasukkan 4 pulsa geser kedalam shift-register dan catatlah hasilnya di bawah ini.  
Setelah pulsa 1: **ABCD = \_\_\_\_\_**  
Setelah pulsa 2: **ABCD = \_\_\_\_\_**  
Setelah pulsa 3: **ABCD = \_\_\_\_\_**  
Setelah pulsa 4: **ABCD = \_\_\_\_\_**

### Percobaan 3. Rangkaian Register Geser Kiri 1

1. Modifikasi rangkaian register geser kedalam bentuk seperti rangkaian pada gambar.

Di sini anda akan menghubungkan masukan data paralel kepada keluaran, yang memungkinkan register geser bekerja geser kiri (*shift left operation*).



2. Anda dapat menggunakan saklar SW1 untuk masukan data serial pada operasi geser kiri, saklar SW2 untuk masukan data serial pada operasi geser kanan. Saklar SW3 dipakai sebagai sumber clock register. Saklar SW4 dipakai sebagai pengendali mode (mode kontrol) rangkaian tersebut, yang akan menentukan operasi geser kiri atau kanan. Saklar SW5 untuk mereset register.
3. Jalankan Simulasi.
4. Set saklar pada kondisi biner berikut:  
 $SW1 = 0; SW2 = 0; SW3 = 1; SW4 = 0; SW5 = 1;$
5. Untuk mereset data, berikan masukan satu pulsa pada SW5 (0-1 ).
6. Atur SW1 ke nilai biner 1.
7. Berikan pulsa pada saklar SW3 sebanyak 5 kali kemudian catat hasil pada keluarannya.  
Sebelum pulsa diberikan : ABCD = .....  
Setelah pulsa 1 : ABCD = .....  
Setelah pulsa 2 : ABCD = .....  
Setelah pulsa 3 : ABCD = .....  
Setelah pulsa 4 : ABCD = .....  
Setelah pulsa 5 : ABCD = .....
8. Set saklar pada kondisi biner berikut:  
 $SW1 = 0 ; SW2 = 0 ; SW3 = 1 ; SW4 = 1 ; SW5 = 1 .$
9. Untuk reset data, berikan pulsa pada saklar SW5 (set 0 kemudian set 1 lagi).
10. Set saklar SW2 ke biner 1.
11. Berikan pulsa pada saklar SW3 sebanyak 5 kali kemudian catat hasil pada keluarannya.  
Sebelum pulsa diberikan : ABCD = .....  
Setelah pulsa 1 : ABCD = .....  
Setelah pulsa 2 : ABCD = .....  
Setelah pulsa 3 : ABCD = .....  
Setelah pulsa 4 : ABCD = .....  
Setelah pulsa 5 : ABCD = .....
12. Set saklar pada kondisi biner berikut:  
 $SW1 = 0 ; SW2 = 0 ; SW3 = 1 ; SW4 = 0 ; SW5 = 1 .$

13. Untuk mereset data, beri pulsa pada switch SW5 (set ke 0 lalu set 1 lagi).
14. Set saklar SW1 ke biner 1 .
15. Berikan pulsa pada saklar SW3, 1 kali dan kemudian catat hasil keluarannya.  
Sebelum pulsa diberikan : ABCD = .....  
Setelah pulsa diberikan : ABCD = .....
16. Set saklar SW1 ke biner 0.
17. Beri pulsa ke SW3 3 kali dan kemudian catat hasil keluarannya.  
Setelah pulsa 1 : ABCD = .....  
Setelah pulsa 2 : ABCD = .....  
Setelah pulsa 3 : ABCD = .....
18. Set saklar SW4 ke biner 0.
19. Beri pulsa ke saklar SW3 3 kali dan kemudian catat hasil keluarannya.  
Setelah pulsa 1 : ABCD = .....  
Setelah pulsa 2 : ABCD = .....  
Setelah pulsa 3 : ABCD = .....
20. Untuk lebih memahami percobaan 3 ini, cobalah dengan melakukan operasi-operasi pergeseran yang lain dengan mengubah-ubah kondisi saklar SW1, SW2, SW3, SW4 dan SW5 sehingga anda memahami fungsi masing-masing tombol.

#### Percobaan 4. Rangkaian Register Geser Kiri 2

1. Buat rangkaian pada percobaan 3 tanpa ada perubahan.
2. Set saklar seperti kondisi biner berikut :  
 $SW1 = 0 ; SW2 = 0 ; SW3 = 1 ; SW4 = 0 ; SW5 = 1 ;$   
Untuk mereset data, beri pulsa pada saklar SW5 (set ke 0 Lalu set 1 lagi).
3. Set saklar SW1 ke biner 1.
4. Beri pulsa pada SW3 2 kali dan kemudian catat hasil keluarannya.  
Sebelum pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....  
Setelah pulsa ke 1 : ABCD = ..... ; **bilangan desimal** = .....  
Setelah pulsa ke 2 : ABCD = ..... ; **bilangan desimal** = .....
5. Set saklar SW1 ke biner 0.

6. Beri pulsa ke SW3 1 kali dan kemudian catat hasil keluarannya.  
Setelah pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....
7. Pelajari data yang anda dapatkan pada prosedur diatas! Apa hubungan antara bilangan-bilangan yang diperoleh, ketika register dimuatkan dengan data dan register operasi geser kiri?  
.....
8. Operasi matematika apa yang dibentuk oleh geser kiri?  
.....
9. Set saklar pada biner berikut :  
 $SW1 = 0$  ;  $SW2 = 0$  ;  $SW3 = 1$  ;  $SW4 = 1$  ;  $SW5 = 1$  ; ..
10. Untuk mereset data, berikan pulsa pada saklar SW5 (set ke 0 lalu set 1 lagi).
11. Set saklar SW2 ke biner 1.
12. Berikan pulsa pada saklar SW3 sebanyak 1 kali kemudian catat hasil pada keluaranya.  
Sebelum pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....  
Setelah pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....
13. Set saklar SW2 ke biner 0.
14. Berikan pulsa pada saklar SW3 sebanyak 1 kali kemudian catat hasil pada keluaranya.  
Setelah pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....
15. Set saklar SW2 ke biner 1.
16. Berikan pulsa pada saklar SW3 sebanyak 1 kali kemudian catat hasil pada keluaranya.  
Setelah pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....
17. Berikan pulsa pada saklar SW3 sebanyak 1 kali kemudian catat hasil pada keluaranya.  
Setelah pulsa diberikan : ABCD = ..... ; **bilangan desimal** = .....
18. Pelajarilah data-data yang anda peroleh! Apa hubungan antara bilangan-bilangan yang diperoleh, ketika register dimuatkan dengan data dan register operasi geser kanan?  
.....
19. Fungsi matematika apakah yang terbentuk, saat terjadi operasi geser kanan?  
.....

©Muhammadiyah University Press

# MODUL 12

## TINGKAT LANJUT (PENGHITUNG WAKTU)

### TUJUAN PRAKTIKUM

Mahasiswa mampu mengimplementasi percobaan sebelumnya dengan membuat penanda waktu, misalnya sebuah stopwatch dan jam digital.

### TEORI

#### 1. Stopwatch

Jam randek atau biasa dikenal dengan stopwatch merupakan salah satu alat pengukur waktu dalam melakukan kegiatan tertentu yang dilengkapi dengan satuan detik, menit, terkadang hingga hitungan jam. Stopwatch dapat ditemukan dalam bentuk analog atau digital. Penggunaan stopwatch diawali dengan menekan bagian tertentu (tombol start) untuk mulai menghitung. Dan menekan tombol tertentu untuk menghentikan waktu sehingga dapat ditampilkan hitungan waktu yang telah berlalu.

Dalam percobaan praktikum ini, stopwatch yang digunakan adalah penghitung waktu berisi menit, detik, dan milidetik menggunakan sistem flip flop dalam pencacahan waktunya. Flip flop adalah rangkaian pengunci yang dapat menyimpan informasi. Flip-flop JK merupakan jenis flip-flop yang sering digunakan, memiliki sifat flip-flop dari semua jenis flip-flop. Tabel kebenaran flip flop JK ditunjukkan pada Tabel 12.1. J dan K disebut sebagai input pengendali untuk menentukan apa yang akan dilakukan oleh flip-flop ketika menerima pulsa clock yang meningkat.

Tabel 12.1. Tabel kebenaran JK flip-flop

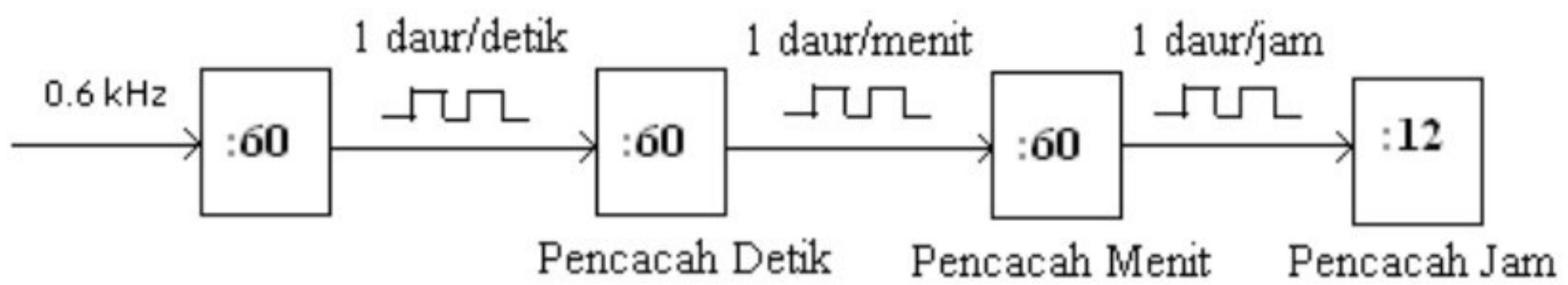
<b>CLK</b>	<b>J</b>	<b>K</b>	<b>Q</b>
0	0	0	Keadaan terakhir
↑	0	1	0
↑	1	0	1
↑	1	1	Keadaan terakhir

Cara kerja stopwatch pada praktikum ini adalah (1) power VCC terhubung ke dalam switch (SW-SPDT) sebagai masukan. (2) rangkaian stopwatch terhubung juga dengan clock yang berfrekuensi 250Hz. Kecepatan pergerakan bilangan cacah tersebut dipengaruhi oleh besar frekuensi pada clock (3) clock tersebut merupakan masukan keempat buah flip flop JK. (4) output flip flop JK merupakan masukan untuk seven segment. Seven segment menjalankan perhitungan dari daur millisecond hingga daur yang lebih besar. (5) setiap daur dilengkapi dengan dua buah seven segment. Misalnya modulus 10, setiap segmen telah mencapai angka 9, maka nilai dari seven segment yang lain akan bertambah 1, dan seterusnya. (6) untuk menghentikan rangkaian, pindah switch (SW-SPDT) yang awalnya terhubung ke power, di ubah ke ground. (7) SW-SPST merupakan switch yang digunakan untuk mereset data yang terlihat pada seven segment. Sehingga pada saat awal rangkaian ini akan dijalankan, pastikan SW-SPST tidak terhubung.

## 2. Jam digital

Suatu pencacah dan proses decode yang sangat menarik adalah dalam perancangan jam digital. Penyedia daya bagi sebuah jam biasa (jam, menit, detik) adalah daya komersial arus bolak-balik 120 Volt, 0.6 kHz. Untuk mendapatkan pulsa-pulsa yang berlangsung pada laju 1 detik diperlukan untuk membagi sumber daya 0.6 kHz dengan 60. Jika gelombang segi-empat 0.01kHz dibagi lagi dengan 60, dihasilkan gelombang segi-empat 1 daur permenit. Jika dibagi lagi dengan 60 maka akan menghasilkan gelombang segi-empat 1 daur perjam seperti ditunjukkan pada Gambar 12.1.

Rangkaian percobaan counter / pencacah yang telah dipelajari akan diterapkan di sini. MOD10 digunakan untuk satuan (jam, menit, dan detik), sedangkan MOD6 untuk puluhan (menit dan detik), dan MOD2 untuk puluhan (jam). Disamping itu decoder diterapkan dalam menampilkan waktu, dengan jenis komponen Binary to 7segment.

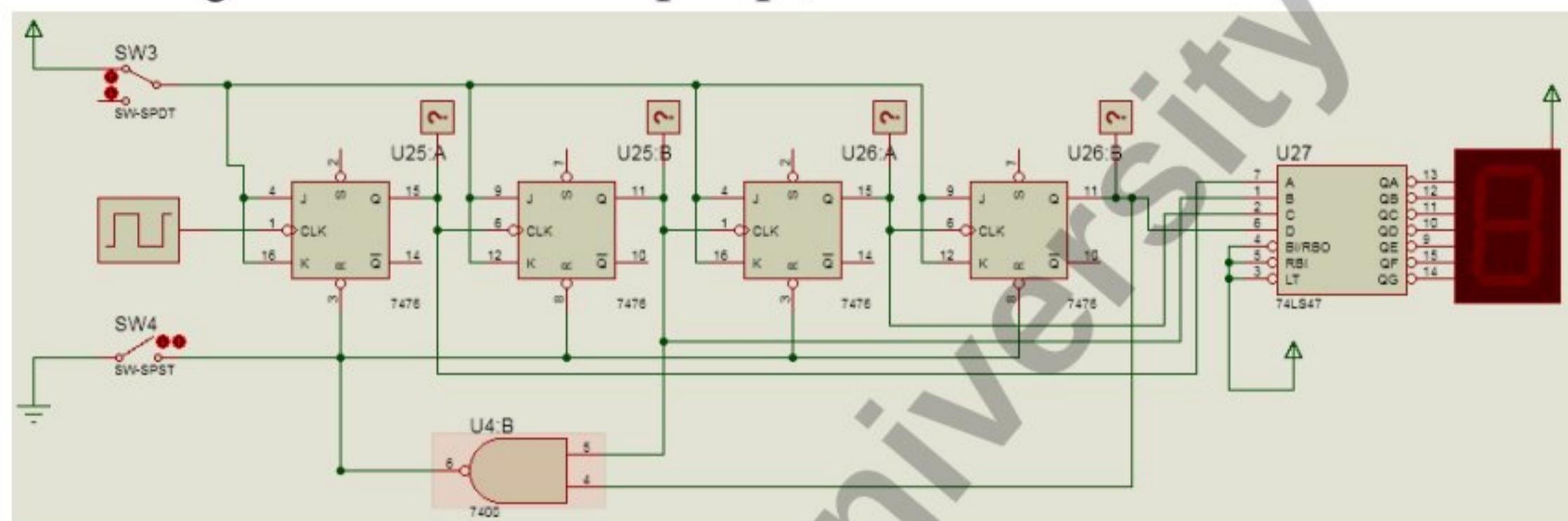


Gambar 12.1. Perhitungan waktu jam digital

## KEGIATAN PRAKTIKUM

### Percobaan 1. Rangkaian Modulus 10 dan Decoder untuk Stopwatch

- Buat rangkaian kombinasi flip flop JK berikut ini!



Gambar 12.2. MOD10 dan decoder

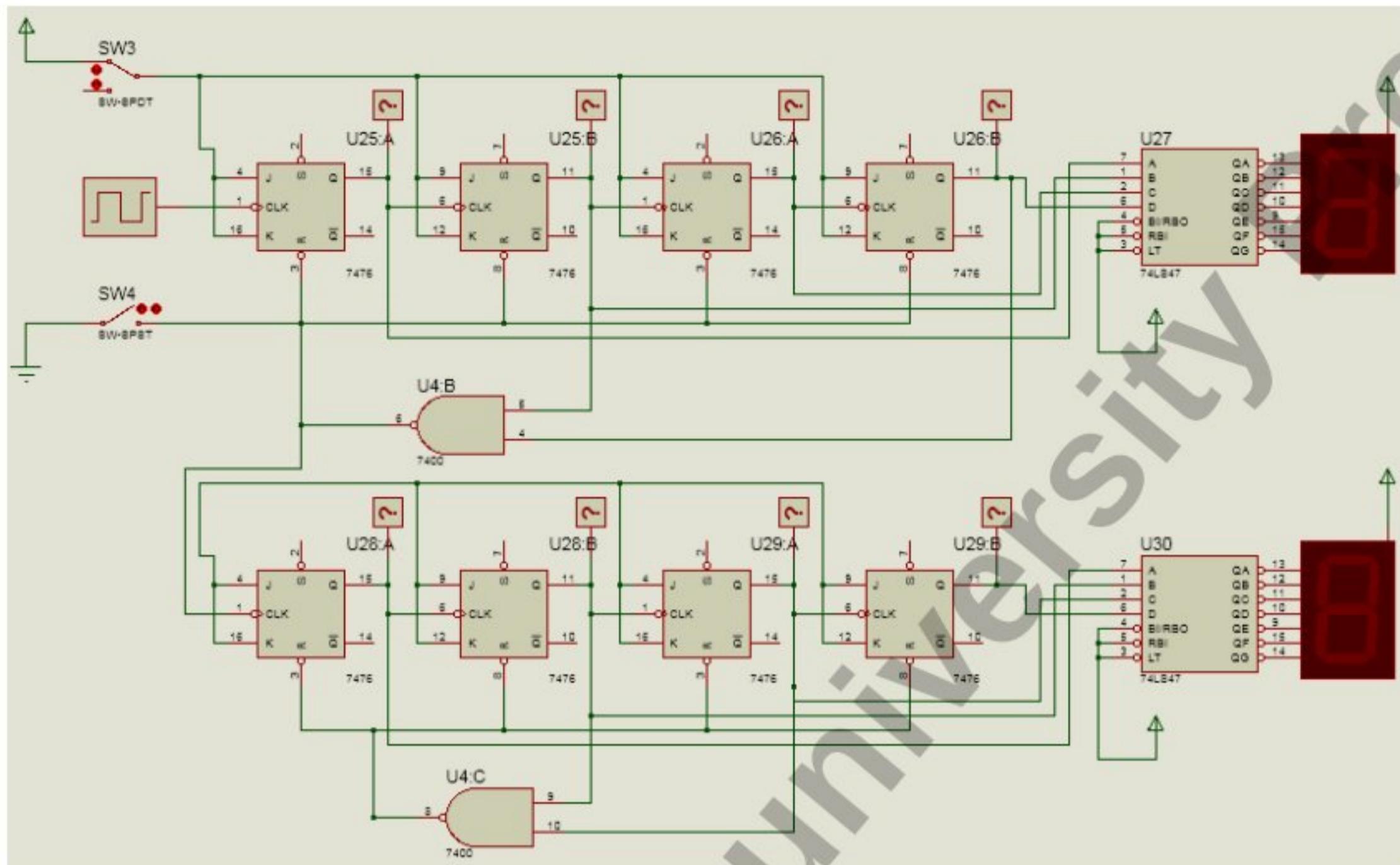
- Komponen yang diperlukan dalam rangkaian di atas dapat dilihat pada Tabel 1.

No	Device	Information
1	IC 74LS47	-
2	IC 7476	-
3	IC 7400	-
4	Logic Probe	-
5	Clock	Frekuensi = 50 Hz
6	Switch-SPDT	-
7	Switch-SPST	-
8	BCD-7Segment	-

- SW-SPST atur dalam keadaan tidak terhubung.
  - Jalankan rangkaian, dan bagaimana output yang terlihat pada seven segment!
- .....

## Percobaan 2. Rangkaian Modulus 6 dan Decoder untuk Stopwatch

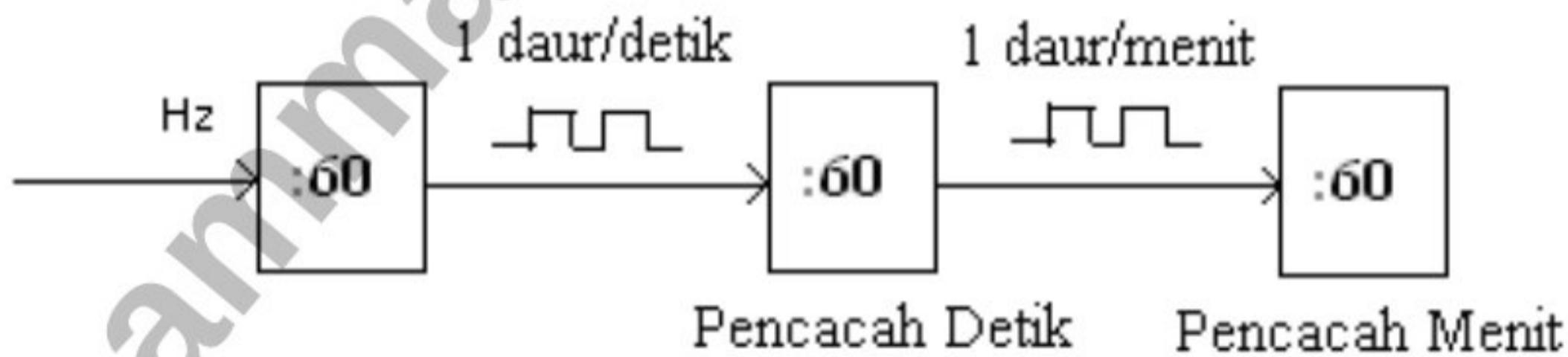
- Untuk mencapai hitungan detik, maka perlu ditambahkan rangkaian counter lainnya. gerbang NAND (percobaan pertama) akan mengirimkan clock pada flip flop JK pertama pada percobaan kedua jika output decoder sudah menampilkan biner 9.



Gambar 12.3. MOD10 dan MOD6 untuk membangkitkan daur detik

- Ubah frekuensi clock menjadi 250 Hz. Apakah fungsi clock pada rangkaian di atas?

- 
- Lanjutkan rangkaian di atas hingga perhitungan daur menit, seperti Gambar 12.3!

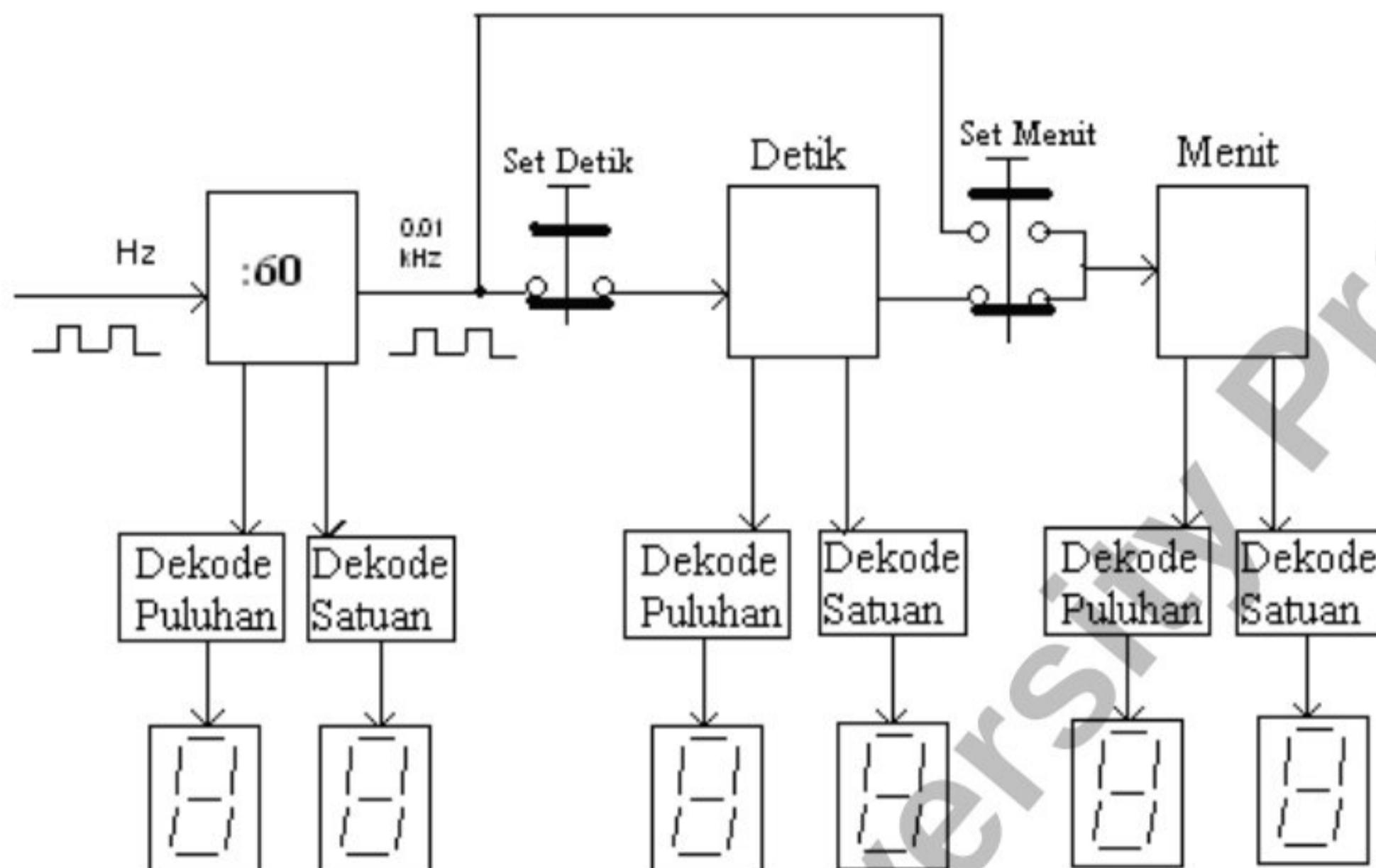


Gambar 12.4. perhitungan waktu daur detik dan daur menit

- Jelaskan kegunaan dari Switch-SPDT pada rangkaian di atas?

- 
- Jelaskan kegunaan dari Switch-SPST pada rangkaian di atas?

6. Laporkan hasil percobaan anda kepada dosen pengampu / asisten praktikum! Agar memudahkan pengamatan, atur seven segment seperti Gambar 12.4!

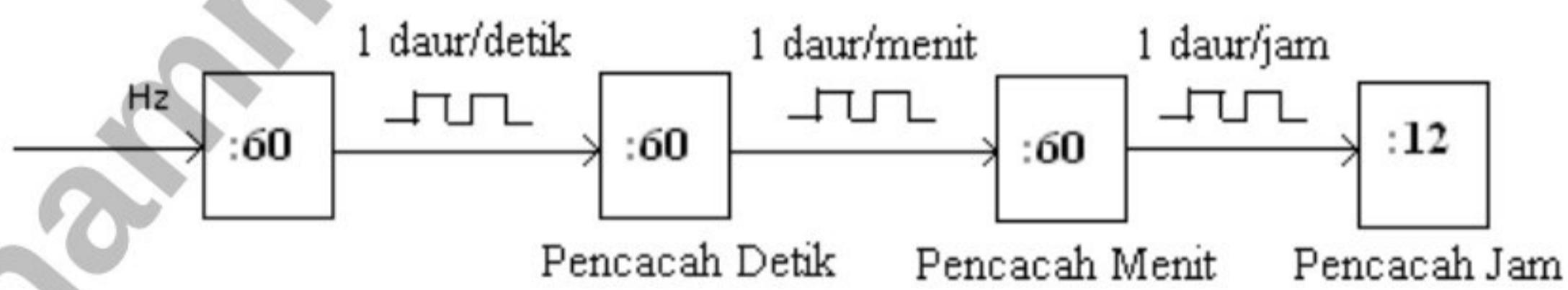


Gambar 12.5. seven segment stopwatch

### Percobaan 3. Jam Digital

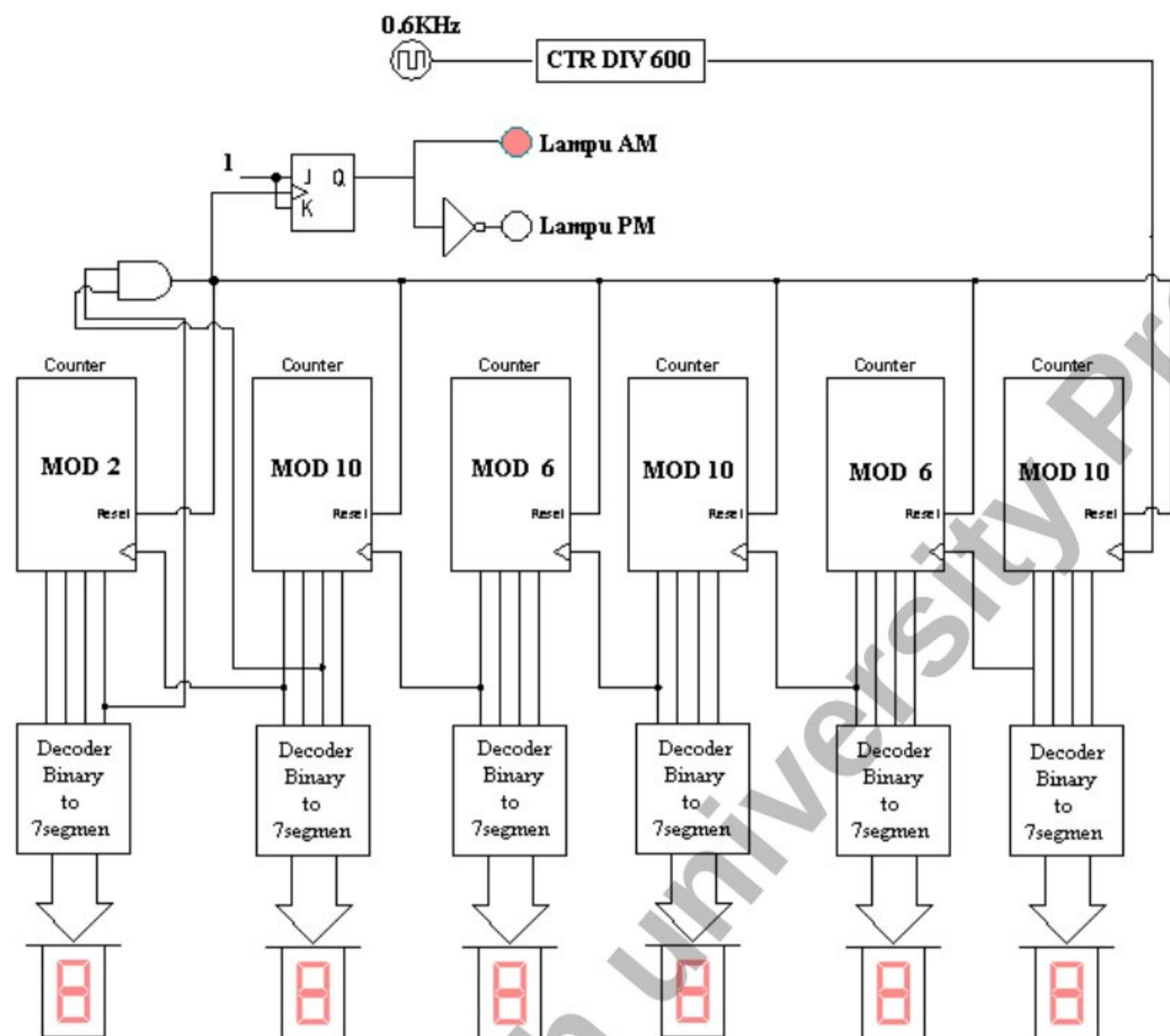
Pembuatan rangkaian jam digital merupakan implementasi akhir dari praktikum ini yaitu dengan penambahan daur Jam seperti terlihat pada Gambar 12.5. Langkah pembuatan jam digital dapat melalui dua tahap yaitu :

- Melanjutkan percobaan 2 diatas dengan menambahkan fungsi mod 2 pada daur jam.
- Penggunaan IC 74192 sebagai penghitung maju dan mundur (up / down counter). Lihat kembali prinsip kerja counter (Modul 8) dan register (Modul 11) pada buku ini.



Gambar 12.6. Pencacah jam digital

1. Buatlah rangkaian berdasarkan rangkaian berikut ini!



Gambar 12.7. Rancangan simulasi jam digital

2. Simulasikan dan perlihatkan hasilnya kepada dosen atau asisten praktikum!
  3. Atur masukan clock menjadi lebih besar, untuk memudahkan simulasi dari output daur jam!

## DAFTAR PUSTAKA

---

<https://www.electronicshub.org/digital-stopwatch-circuit/> diakses pada 11 November 2019

<https://mardhiyatulhayati2015.wordpress.com/post/stopwatch/> diakses pada 11 November 2019

Informatika. "Modul Praktikum Sistem Digital". Surakarta : Universitas Muhammadiyah Surakarta. 2016

Labcenter Electronics. "Proteus Design Suite Getting Started Guide". [www.labcenter.com](http://www.labcenter.com). Labcenter Elektronics Ltd 1990-2019

Manik, Henry. " Tutorial Osiloskop". Bogor : Institut Pertanian Bogor. 2006

©Muhammadiyah University Press

©Muhammadiyah University Press



ISBN: 978-602-361-271-0

A standard linear barcode is positioned in the center of a white rectangular box. The ISBN number "978-602-361-271-0" is printed vertically along the left side of the barcode.

9 786023 612710