



# LAPORAN PRAKTIKUM

IF310303

## PRAKTIKUM SISTEM DIGITAL

MODUL: 8

*COUNTER*

NAMA : Muhammad Alwiza Ansyar

NIM : M0520051

HARI : Jumat

TANGGAL : 4 Desember 2020

WAKTU : 10.15 – 11.05 WIB

ASISTEN : Akhtar Bariq Rahman

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SEBELAS MARET

2020

# Modul 8

## COUNTER

**Muhammad Alwiza Ansyar** (M0520051) / 4 Desember 2020

Email : alwiza21@student.uns.ac.id

Asisten : Akhtar Bariq Rahman

**Abstraksi**— Berikut merupakan laporan praktikum untuk modul 8 yang memiliki fokus bahasan tentang *counter*. *Counter* adalah salah satu rangkaian logika sekuensial yang [ISI]. Pada laporan praktikum ini, akan dibahas tentang pengekspresian *up counter* dan *down counter* baik secara sinkron maupun asinkron dalam rangkaian logika serta tabel kebenarannya.

**Kata kunci**— *counter*, *up counter*, *down counter*, *counter sinkronus*, *counter asinkronus*

### I. PENDAHULUAN

Pada sistem digital, diperlukan alat untuk melacak dan mencatat masukan dari pengguna supaya alat dapat berfungsi seperti seharusnya. Contoh sederhananya adalah sistem volume pada TV. Masukan dari pengguna saat menekan tombol *volume up* dan *volume down* haruslah disimpan dan dicatat oleh sistem pada TV supaya dapat memproses besar volume yang seharusnya. Apabila pada keadaan awal, volume sebesar 7 bar, lalu pengguna menekan tombol *volume up* sebanyak 3x, maka besar volume akan menjadi 7+3 yaitu 10 bar, yang lalu diproses dan hasil tadi disimpan pada sistem TV.

*Counter* adalah rangkaian sekuensial yang dapat melakukan pencacahan/penghitungan. *Counter* dapat dijumpai hampir di setiap alat elektronik karena fungsi dari *counter* sendiri bersifat fundamental/hal yang mendasar.

### II. DASAR TEORI

#### 2.1 *Counter* secara umum

*Counter* atau pencacah adalah rangkaian sekuensial memiliki fungsi untuk melakukan penghitungan/pencacahan berdasarkan banyaknya *pulse* dari clock. Karena *counter* akan menghitung sesuatu secara bertahap/sekuensial (yaitu berdasarkan keadaan jumlah *pulse* yang diberikan), maka diperlukan rangkaian memori untuk menyimpan informasi jumlah *pulse* sehingga *counter* adalah terdiri dari satu atau lebih rangkaian flip-flop. Flip-flop yang biasa digunakan yaitu JK Flip-flop.

Hampir seluruh peralatan elektronik yang menerapkan sistem digital memiliki unsur *counter* di dalamnya. Kegunaan *counter* secara umum ialah:

- Operasi aritmetika
- Pembagi frekuensi
- Pengurut alamat

*Counter* dapat memiliki rentang clock yang beragam sesuai dari bagaimana *counter* didesain. Jumlah clock yang sering dipakai adalah  $2^n$  yaitu 0-3, 0-7, 0-15, dst. tetapi kita juga bisa merancang *supaya* clock dapat menghitung diluar dari  $2^n$  seperti 0-2 dan 2-6.

Clock pada *counter* bekerja secara *negative trigger/falling edge* sehingga perubahan *output* terjadi saat keadaan clock berubah dari *high* (1) menjadi *low* (0).

## 2.2 Jenis Counter Berdasarkan Distribusi Pulse

*Counter* dibagi menjadi dua yaitu asinkronus dan sinkronus. *Counter* asinkronus atau sering disebut *ripple counter* adalah *counter* yang clocknya hanya disambung pada Flip-flop LSB/urutan pertama, yang kemudian *output* dari Flip-flop tersebut akan menjadi clock untuk Flip-flop berikutnya. Dengan ini, *pulse* dari clock disebarkan secara merambat dan tidak serentak. *Counter* sinkronus adalah *counter* yang clocknya disambung ke setiap Flip-flop yang digunakan sehingga *pulse* dari clock disebarkan secara serentak ke setiap Flip-flop.

Secara praksis, *counter* asinkronus memiliki *propagatin delay* yang lebih lama daripada *counter* sinkronus. Hal ini dikarenakan pada *counter* asinkronus, *pulse* harus bergerak dari Flip-flop pertama, lalu Flip-flop kedua, lalu Flip-flop ketiga, dst. Sedangkan pada *counter* sinkronus, *pulse* bergerak secara serentak dan langsung ke setiap Flip-flop. Pada rangkaian digital yang semakin besar dan kompleks, kekurangan dari *counter* asinkronus akan dapat dirasakan.

## 2.3 Jenis Counter Berdasarkan Urutan Hitung

*Counter* dibagi menjadi *up counter/counter* naik dan *down counter/counter* turun. *Up counter* adalah *counter* yang melakukan penghitungan dari kecil ke besar. Contohnya pada *up counter* 2 Bit maka akan dilakukan penghitungan 0, 1, 2, dan 3. *Down counter* adalah *counter* yang melakukan penghitungan dari besar ke kecil. Contohnya pada *down counter* 2 Bit maka akan dilakukan penghitungan 3, 2, 1, dan 0. Penggunaan *up* dan *down counter* ini sesuai dengan kebutuhan.

# III. ALAT DAN LANGKAH PERCOBAAN

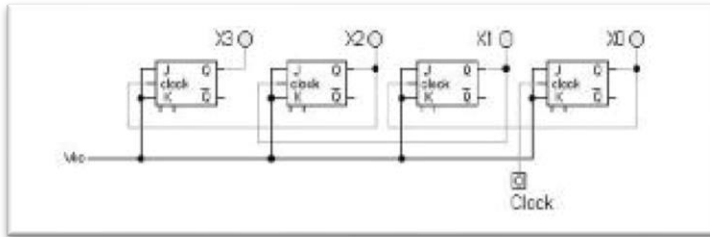
## 3.1 Alat

1. PC/Laptop
2. Aplikasi Digital Works

## 3.2 Langkah Percobaan

### **Counter naik asinkron**

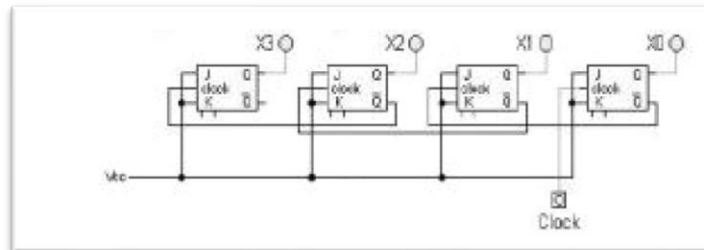
1. Buka aplikasi Digital Works
2. Buatlah rangkaian berikut



3. Amati dan catat *output* terhadap kombinasi keadaan *input*

**Counter turun asinkron**

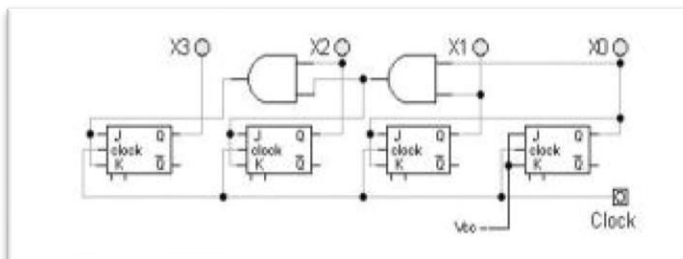
4. Buatlah rangkaian berikut



5. Amati dan catat *output* terhadap kombinasi *input*

**Counter naik sinkron**

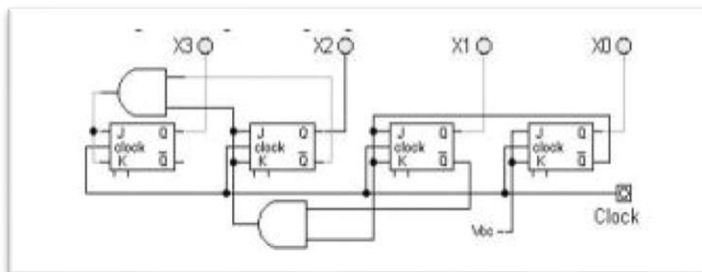
6. Buatlah rangkaian berikut



7. Amati dan catat *output* terhadap kombinasi *input*

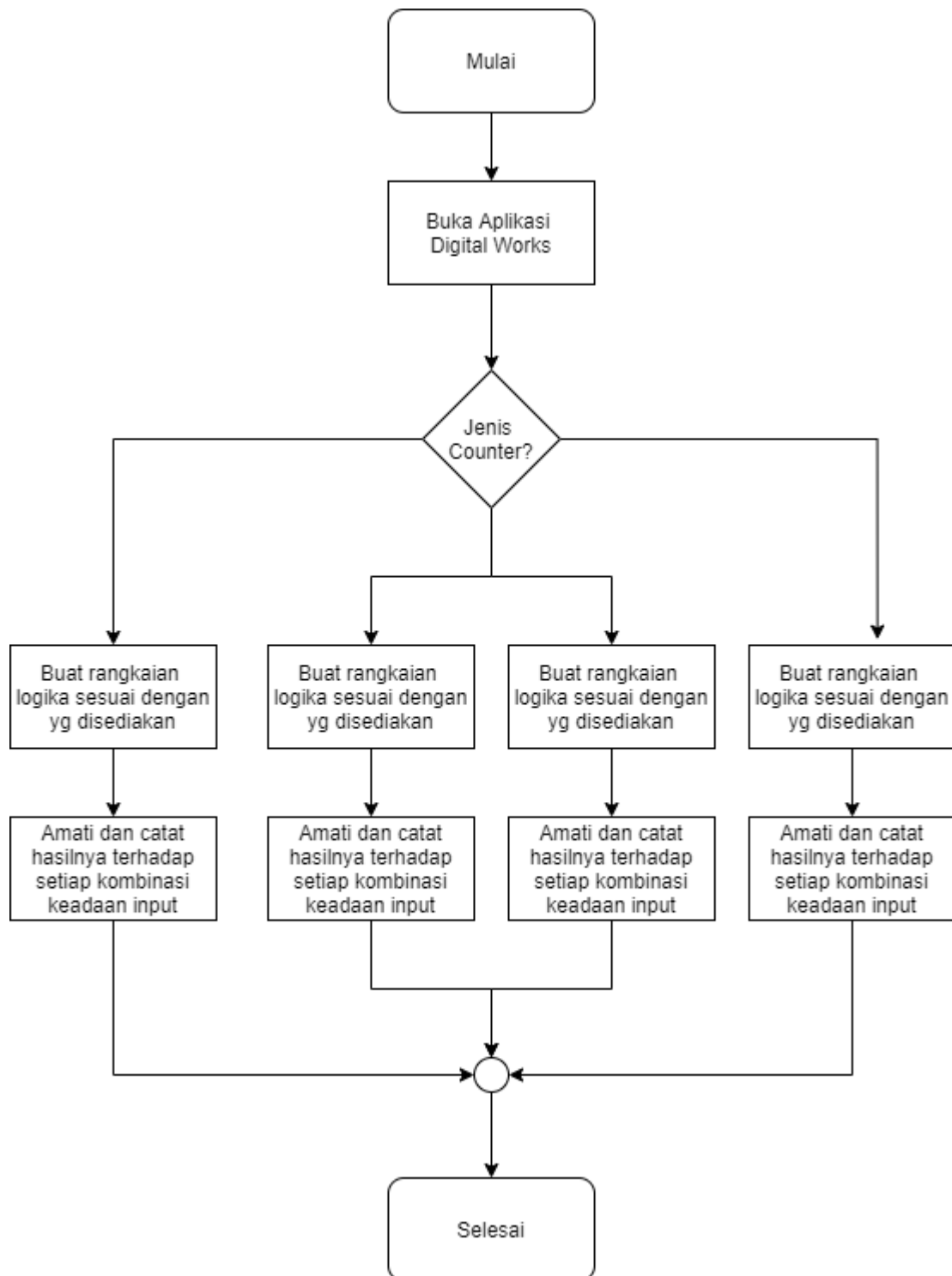
**Counter turun sinkron**

8. Buatlah rangkaian berikut



9. Amati dan catat *output* terhadap kombinasi *input*

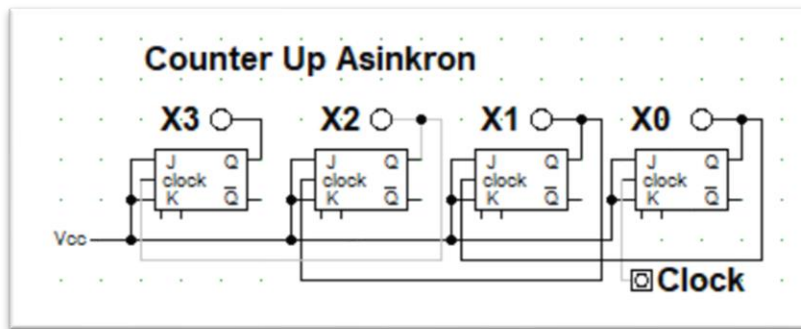
Diagram alur:



#### IV. HASIL DAN ANALISIS PERCOBAAN

##### 4.1 Up counter 4-bit asinkronus

Rangkaian logika:



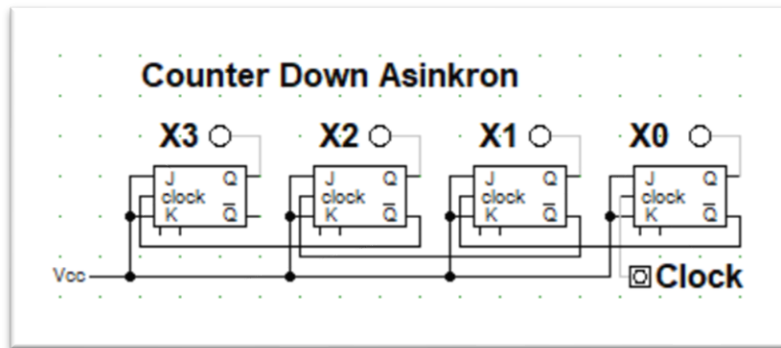
Tabel kebenaran:

Jumlah pulse	Mod-16	X3	X2	X1	X0	Desimal
0	0	0	0	0	0	0
1	1	0	0	0	1	1
2	2	0	0	1	0	2
3	3	0	0	1	1	3
4	4	0	1	0	0	4
5	5	0	1	0	1	5
6	6	0	1	1	0	6
7	7	0	1	1	1	7
8	8	1	0	0	0	8
9	9	1	0	0	1	9
10	10	1	0	1	0	10
11	11	1	0	1	1	11
12	12	1	1	0	0	12
13	13	1	1	0	1	13
14	14	1	1	1	0	14
15	15	1	1	1	1	15
16	0	0	0	0	0	0

- Dapat dilihat pada rangkaian bahwa clock hanya disambungkan pada JK Flip-flop pertama (di bawah X0) lalu *outputnya* menjadi clock untuk Flip-flop berikutnya sehingga *counter* ini bekerja secara asinkronus.
- Karena *counter* dapat menghitung hingga 4-bit, *counter* ini sering juga disebut dengan *counter* mod-16. Rangkaian membaca jumlah clock secara mod-16 sehingga saat clock berupa 0 atau kelipatan 16, *counter* akan direset menjadi 0000.
- *Output counter* sama dengan hasil clock mod-16 dan naik dari kecil ke besar, atau  $Output = (clock) \text{ Mod } 16$  sehingga *counter* ini mencacah secara naik (*up counter*)

## 4.2 Down counter 4-bit asinkronus

Rangkaian logika:



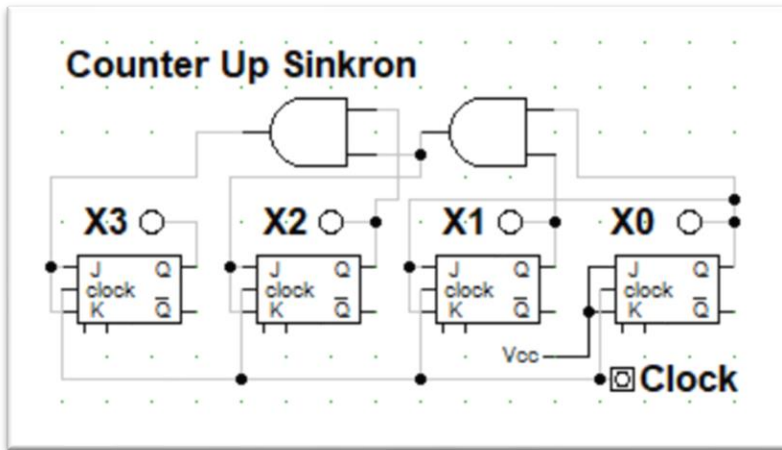
Tabel kebenaran:

Jumlah <i>pulse</i>	Mod-16	X3	X2	X1	X0	Desimal
0	0	0	0	0	0	0
1	1	1	1	1	1	15
2	2	1	1	1	0	14
3	3	1	1	0	1	13
4	4	1	1	0	0	12
5	5	1	0	1	1	11
6	6	1	0	1	0	10
7	7	1	0	0	1	9
8	8	1	0	0	0	8
9	9	0	1	1	1	7
10	10	0	1	1	0	6
11	11	0	1	0	1	5
12	12	0	1	0	0	4
13	13	0	0	1	1	3
14	14	0	0	1	0	2
15	15	0	0	0	1	1
16	0	0	0	0	0	0

- Dapat dilihat pada rangkaian bahwa clock hanya disambungkan pada JK Flip-flop pertama (di bawah X0) lalu *outputnya* menjadi clock untuk Flip-flop berikutnya sehingga *counter* ini bekerja secara asinkronus.
- Karena *counter* dapat menghitung hingga 4-bit, *counter* ini sering juga disebut dengan *counter* mod-16. Rangkaian membaca jumlah clock secara mod-16 sehingga saat clock berupa 0 atau kelipatan 16, *counter* akan direset menjadi 0000.
- *Output counter* untuk selain dari kondisi diatas (yang memiliki hasil  $\text{clock} \times \text{mod}16 = 0$ ) yaitu 1-15 adalah berkebalikan dengan *up counter*, yaitu dari besar ke kecil, atau 
$$\text{Output} = 16 - (\text{clock})\text{Mod}16, \quad (\text{clock})\text{Mod}16 \neq \{0\}$$
 sehingga *counter* ini mencacah secara turun (*down counter*)

### 4.3 Up counter 4-bit sinkronus

Rangkaian logika:



Tabel kebenaran:

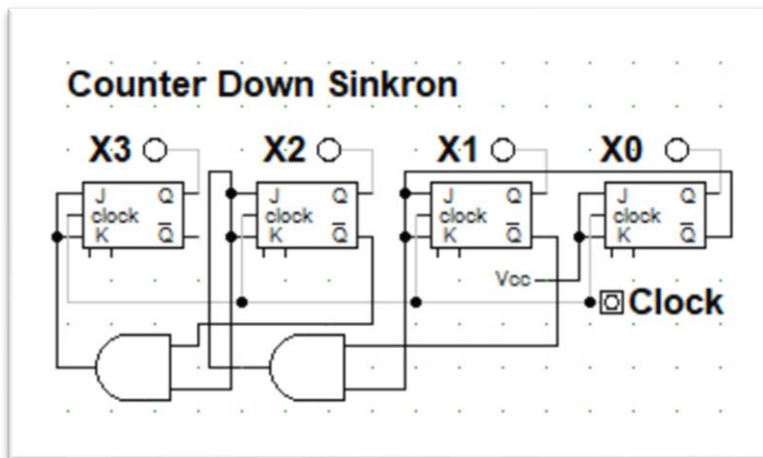
Jumlah <i>pulse</i>	Mod-16	X3	X2	X1	X0	Desimal
0	0	0	0	0	0	0
1	1	0	0	0	1	1
2	2	0	0	1	0	2
3	3	0	0	1	1	3
4	4	0	1	0	0	4
5	5	0	1	0	1	5
6	6	0	1	1	0	6
7	7	0	1	1	1	7
8	8	1	0	0	0	8
9	9	1	0	0	1	9
10	10	1	0	1	0	10
11	11	1	0	1	1	11
12	12	1	1	0	0	12
13	13	1	1	0	1	13
14	14	1	1	1	0	14
15	15	1	1	1	1	15
16	0	0	0	0	0	0

- Dapat dilihat pada rangkaian bahwa clock disambungkan pada setiap JK Flip-flop yang ada sehingga *counter* ini bekerja secara sinkronus.
- Karena *counter* dapat menghitung hingga 4-bit, *counter* ini sering juga disebut dengan *counter* mod-16. Rangkaian membaca jumlah clock secara mod-16 sehingga saat clock berupa 0 atau kelipatan 16, *counter* akan direset menjadi 0000.
- *Output counter* sama dengan hasil clock mod-16 dan naik dari kecil ke besar, atau  $Output = (clock) \text{ Mod } 16$  sehingga *counter* ini mencacah secara naik (*up counter*)



#### 4.4 Down counter 4-bit sinkronus

Rangkaian logika:



Tabel kebenaran:

Jumlah <i>pulse</i>	Mod-16	X3	X2	X1	X0	Desimal
0	0	0	0	0	0	0
1	1	1	1	1	1	15
2	2	1	1	1	0	14
3	3	1	1	0	1	13
4	4	1	1	0	0	12
5	5	1	0	1	1	11
6	6	1	0	1	0	10
7	7	1	0	0	1	9
8	8	1	0	0	0	8
9	9	0	1	1	1	7
10	10	0	1	1	0	6
11	11	0	1	0	1	5
12	12	0	1	0	0	4
13	13	0	0	1	1	3
14	14	0	0	1	0	2
15	15	0	0	0	1	1
16	0	0	0	0	0	0

- Dapat dilihat pada rangkaian bahwa clock disambungkan pada setiap JK Flip-flop yang ada sehingga *counter* ini bekerja secara sinkronus.
- Karena *counter* dapat menghitung hingga 4-bit, *counter* ini sering juga disebut dengan *counter* mod-16. Rangkaian membaca jumlah clock secara mod-16 sehingga saat clock berupa kelipatan 16 atau 0, *counter* akan direset menjadi 0000.
- *Output counter* untuk selain dari kondisi diatas (yang memiliki hasil  $\text{clock} \times \text{mod}16 = 0$ ) yaitu 1-15 adalah berkebalikan dengan *up counter*, yaitu dari besar ke kecil, atau  
$$\text{Output} = 16 - (\text{clock})\text{Mod}16, \quad (\text{clock})\text{Mod}16 \neq \{0\}$$
sehingga *counter* ini mencacah secara turun (*down counter*)

## V.KESIMPULAN

*Counter* merupakan rangkaian sekuensial yang terdiri dari beberapa Flip-flop yang dapat melakukan penghitungan/pencacahan. Fungsi tersebut adalah fundamental dalam sistem digital sehingga *counter* dapat dengan mudah ditemukan di berbagai alat elektronik. *Counter* menghitung *pulse* yang berasal dari clock yang lalu ditampilkan jumlah tersebut dalam *output*nya. Clock pada *counter* bekerja secara *negative trigger/falling trigger*.

Dari cara kerjanya, *counter* dibagi menjadi sinkronus dan asinkronus. *Counter* sinkronus menghubungkan clock ke setiap flip-flop. *Counter* asinkronus menghubungkan clock hanya ke flip-flop pertamanya yang lalu *output* dari flip-flop tersebut menjadi clock untuk flip-flop selanjutnya.

Dari urutan menghitungnya, *counter* dibagi menjadi *counter* naik dan *counter* turun. *Counter* naik menghitung dari kecil ke besar yaitu seperti 0, 1, 2, 3, 4, dst. sedangkan *counter* turun menghitung dari besar ke kecil yaitu seperti 5, 4, 3, 2, dst.

Setiap *counter* memiliki rentang yang terdiri dari minimal dan maksimal. Pada *up counter*, apabila *counter* melebihi rentang maksimal, maka akan direset menjadi keadaan minimal. Pada *down counter*, apabila *counter* kurang dari rentang minimal, maka akan direset menjadi keadaan maksimal.

Pada praktikum ini, dibuat *counter* 4-bit yang bekerja secara sinkronus dan asinkronus serta dihitung secara naik dan turun (total ada 4 jenis *counter* 4-bit). Rangkaian-rangkaian tersebut dibuat pada aplikasi Digital Works. Setelah rangkaian dibuat, dilakukan pengetesan dan pencatatan *output* sesuai dengan kombinasi *input* lalu hasil tersebut disajikan pada bagian pembahasan.

Counter yang bekerja secara asinkronus memiliki kelemahan yakni bekerja lebih lambat daripada counter yang bekerja secara sinkronus. Hal ini mungkin tidak terasa pada praktikum ini, namun semakin besar dan kompleks suatu rangkaian atau semakin cepat frekuensi dari *pulse*, maka counter asinkronus akan terasa kekurangannya. Kekurangan tersebut disebabkan oleh proses perambatan dari satu flip-flop ke flip-flop lain yang menyita banyak waktu.

## VI. DAFTAR PUSTAKA

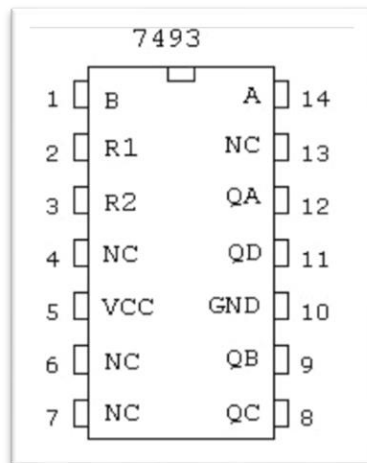
Hari. 2015. *Counter atau Rangkaian Pencacah*, diakses dari <https://www.uniksharianja.com/2015/05/counter-atau-rangkaian-pencacah.html#>, pada 6 Desember 2020

Rizki, Aditya. 2011. *Tutorial Teknik Digital : Rangkaian Pencacah (Counter)*, diakses dari <https://adityarizki.net/tutorial-teknik-digital-rangkaian-pencacah-counter/>, pada 6 Desember 2020

[Muhammad Alwiza Ansyar. Saya adalah seorang mahasiswa yang berasal dari Bogor. Saat ini, saya sedang menempuh pendidikan di Universitas Sebelas Maret jurusan Informatika.....]

## VII. LAMPIRAN

Pinout IC 7493 (4-Bit *Binary Ripple Counter*)



*Up counter* asinkronus menggunakan IC 7493

