

# APLIKASI JK FLIP-FLOP UNTUK MERANCANG *DECADE COUNTER* ASINKRON

Oleh: Lilik Eko Nuryanto

Staf Pengajar Jurusan Teknik Elektro – Politeknik Negeri Semarang  
Jl. Prof. H. Soedarto, SH, Tembalang Semarang – 50275  
Email: lilikpolines@gmail.com

Abstrak

*Counter* (Rangkaian Pencacah atau penghitung) yaitu rangkaian logika sekuensial yang digunakan untuk menghitung jumlah pulsa yang diberikan pada bagian masukan. *Counter* digunakan untuk berbagai operasi aritmatika, pembagi frekuensi, penghitung jarak (*odometer*), penghitung kecepatan (*speedometer*), yang pengembangannya digunakan luas dalam aplikasi perhitungan pada instrumen ilmiah, kontrol industri, komputer, perlengkapan komunikasi, dan sebagainya. Dengan demikian penundaan *counters* adalah sama dengan penundaannya *flip-flop*. Pencacah sinkron memerlukan sirkuit lonceng/ clock yang berdaya tinggi, sebab lonceng harus menggerakkan semua *flip-flop* secara serempak. Sedangkan *Counter Asinkron*, yang mendapatkan input pulsa clock hanya *Flip-flop* yang bobotnya paling rendah. *Flip-flop* yang bobotnya lebih tinggi mendapatkan pulsa clock dari output *Flip-flop* yang bobotnya satu tingkat lebih rendah, begitu seterusnya. *Decade Counter* adalah counter yang mempunyai urutan hitungan biner mulai dari 0 s/d 9 dan mengulang (*re-cycle*).

**Kata Kunci** : *Flip-flop, Counter, Asinkron, Sinkron, Pulsa Clock, Decade, JK*

## 1. Pendahuluan

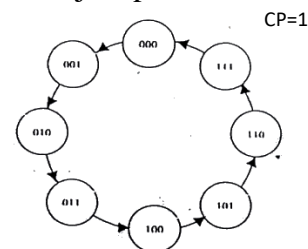
Rangkaian berurutan (*Sequential Circuit*) yang beroperasi melalui urutan kondisi tertentu berdasarkan aplikasi dari pulsa-pulsa inputnya dinamakan “pencacah” (*counter*).

Pulsa-pulsa input tersebut, dinamakan pulsa penghitung (*count pulses*), bisa berupa pulsa clock, atau pulsa-pulsa input tersebut dapat berasal dari “external source” dan bisa terjadi pada interval waktu tertentu atau random.

Dalam suatu counter, urutan kondisi tersebut dapat mengikuti hitungan biner atau urutan kondisi yang lain. Counter ditemukan hampir disemua peralatan yang berisi logika digital. Counter tersebut digunakan untuk menghitung jumlah peristiwa dari suatu kejadian dan berguna untuk menimbulkan urutan waktu untuk mengontrol operasi-operasi di dalam sistem digital.

Counter yang mengikuti urutan biner dinamakan binary counter (pencacah biner) Pencacah biner n-bit terdiri dari n *flip-flop* dan dapat menghitung dalam biner dari 0 ke  $2^n - 1$ .

Sebagai contoh, diagram kondisi (*state diagram*) dari counter 3-bit ditunjukkan pada Gambar 1. Sebagaimana terlihat dari kondisi-kondisi biner yang ditunjukkan di dalam lingkaran, output-output *flip-flop* mengulangi urutan hitungan biner dengan kembali ke 000 setelah 111. Ingat bahwa transisi-transisi kondisi di dalam “*clocked sequential circuits*” terjadi selama pulsa clock; *flip-flop* tetap pada “*present state*”-nya jika tidak terjadi pulsa.



Gambar 1. State Diagram dari Counter Biner 3-bit

“*Next state*” suatu counter tergantung sepenuhnya pada “*present state*”-nya, dan transisi kondisi terjadi setiap kali terjadi pulsa. Karena sifat ini, suatu counter sepenuhnya ditentukan oleh daftar dari urutan hitungan (*count sequence*), yaitu urutan dari kondisi-kondisi biner yang dilalui.

Urutan menghitung dari counter biner 3-bit diberikan dalam Tabel 1. Bilangan berikutnya didalamnya urutan merupakan kondisi berikutnya (*next state*) yang dicapai oleh rangkaian setelah penerapan pulsa menghitung (*count pulse*). Urutan menghitung terulang setelah urutan tersebut mencapai harga terakhir, sehingga kondisi 000 adalah kondisi berikutnya setelah 111. “*Count sequence*” memberi semua informasi yang diperlukan untuk merancang rangkaian. Kondisi berikutnya (*next state*) tidak perlu didaftar dalam kolom terpisah sebab *next state* dapat dibaca dari bilangan berikutnya di dalam urutan.

Tabel 1. Tabel eksitasi untuk counter biner 3-bit

Count Sequence (Urutan Hitungan)			Input-input Flip-flop		
A2	A1	A0	TA2	TA1	TA0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	1

Tabel 1 adalah tabel eksitasi untuk counter biner 3-bit. Ketiga flip-flop diberi penandaan variabel A2, A1, A0. Counter-counter biner paling efisien dibuat dengan T flip-flop.

Eksitasi flip-flop untuk input-input T didapatkan dari tabel eksitasi dari T flip-flop dan dari inspeksi transisi kondisi dari hitungan yang diberikan (*present state*) ke kondisi berikutnya (*next state*).

Sebagai ilustrasi, pelajari proses input flip-flop untuk baris 001. *Present state* di sini adalah 001 dan *next state*nya adalah 010, yang mana adalah hitungan berikutnya di dalam urutan. Dengan memperhatikan kedua hitungan ini, kita catat bahwa A2 tidak berubah dari 0 ke 0; maka TA2 ditandai dengan 0 sebab flip-flop A2 harus

tetap tidak berubah jika terjadi pulsa clock. A1 berubah dari 0 ke 1; maka TA1 ditandai dengan 1 sebab flip-flop ini harus terkomplemen dalam pulsa clock berikutnya. Dengan cara yang sama, A0 berubah dari 1 ke 0, menandakan bahwa FF harus terkomplemen; maka TA0 ditandai dengan 1.

*Counter* juga disebut pencacah atau penghitung yaitu rangkaian logika sekuensial yang digunakan untuk menghitung jumlah pulsa yang diberikan pada bagian masukan. *Counter* digunakan untuk berbagai operasi aritmatika, pembagi frekuensi, penghitung jarak (odometer), penghitung kecepatan (speedometer), yang pengembangannya digunakan luas dalam aplikasi perhitungan pada instrumen ilmiah, kontrol industri, komputer, perlengkapan komunikasi, dan sebagainya .

*Counter* tersusun atas sederetan *flip-flop* yang dimanipulasi sedemikian rupa dengan menggunakan peta Karnough sehingga pulsa yang masuk dapat dihitung sesuai rancangan. Dalam perancangannya *counter* dapat tersusun atas semua jenis flip-flop, tergantung karakteristik masing-masing flip-flop tersebut.

Dilihat dari arah cacahan/ hitungan, rangkaian pencacah dibedakan atas pencacah naik (*Up Counter*) dan pencacah turun (*Down Counter*). Pencacah naik melakukan cacahan dari kecil ke arah besar, kemudian kembali ke cacahan awal secara otomatis. Pada pencacah menurun, pencacahan dari besar ke arah kecil hingga cacahan terakhir kemudian kembali ke cacahan awal. Secara umum *counter* terbagi atas 2 jenis, yaitu: Pencacah sinkron (*Synchronous Counter*) dan Pencacah Asinkron (*Asynchronous counter*). Perbedaan kedua jenis *counter* ini adalah pada pemicuannya. Pada *Synchronous counter* pemicuan flip-flop dilakukan serentak (dipicu oleh satu sumber clock) susunan flip-flopnya paralel. Sedangkan pada *Asynchronous counter*, minimal ada

salah satu flip-flop yang clock-nya dipicu oleh keluaran flip-flop lain atau dari sumber clock lain, dan susunan flip-flopnya seri. Dengan memanipulasi koneksi flip-flop berdasarkan peta karnough atau *timing diagram* dapat dihasilkan *counter* acak, *shift counter* (*counter* sebagai fungsi register) atau juga *up-down counter*.

Karakteristik penting dari pada Counter :

- Kerjanya sinkron atau tak sinkron.
- mencacah maju atau mundur.
- sampai beberapa banyak ia dapat mencacah (modulo pencacah).
- Dapat berjalan terus ( free running ) ataukah dapat berhenti sendiri ( Self Stopping )

Macam-macam penggunaan *counter* :

- Penggunaan pencacah dalam teknologi industri. Dalam hal ini pencacah dioperasikan untuk menghitung obyek (barang produksi) dengan tujuan untuk mencapai kecepatan dan kecermatan penghitungan.
- Digunakan sebagai pembagi frekuensi.
- Untuk mengukur besarnya frekuensi.
- mengukur waktu interval anta dua pulsa.
- Untuk mengukur jarak.
- Untuk mengukur kecepatan.
- Penggunaan dalam digital komputer.
- Untuk mengubah sinyal analog menjadi digital ( Analog to Digital Converters / ADC ) maupun untuk mengubah sinyal digital ke analog ( *Digital to Analog Converter / DAC* ) .

## 2. Merancang Counter Asinkron

Mengacu pada Gambar 2 (a), masing-masing FF mempunyai input-input J dan K nya pada level 1, sehingga FF tersebut akan berubah kondisi (*toggle*) kapan saja jika sinyal pada input CLK-nya bergerak dari Tinggi (*High*) ke rendah (*Low*). Deretan pulsa clock diberikan hanya pada input CLK dari FF X<sub>0</sub>. Output X<sub>0</sub> dihubungkan pada input CLK dari FF X<sub>1</sub> dan output X<sub>1</sub> dihubungkan pada input CLK dari FF X<sub>2</sub>. Bentuk gelombang pada Gambar 2 (b)

menunjukkan bagaimana FF-FF tersebut berubah kondisi sesaat pulsa clock diberikan. Hal-hal penting berikut harus dicatat :

- FF X<sub>0</sub> berubah kondisi (*toggle*) pada transisi ke-negatif (*negative-going transition*) dari masing-masing input pulsa clock. Jadi, bentuk gelombang output X<sub>0</sub> mempunyai frekuensi tepat  $\frac{1}{2}$  nya dari frekuensi pulsa clock.
- FF X<sub>1</sub> berubah kondisi setiap kali output X<sub>0</sub> bergerak dari High to Low. Bentuk gelombang X<sub>1</sub> mempunyai frekuensi tepat sama dengan  $\frac{1}{2}$  dari frekuensi output X<sub>0</sub> dan sehingga  $\frac{1}{4}$  dari frekuensi clock.
- FF X<sub>2</sub> berubah kondisi setiap kali output X<sub>1</sub> bergerak dari High to Low. Jadi, bentuk gelombang X<sub>2</sub> mempunyai  $\frac{1}{2}$  frekuensi dari X<sub>1</sub> dan sehingga  $\frac{1}{8}$  dari frekuensi clock.

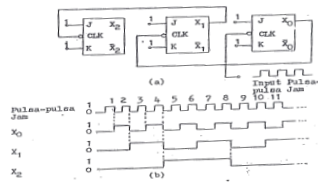
Sebagaimana dideskripsikan di atas, masing-masing FF membagi frekuensi dari inputnya dengan 2. Jadi, seandainya kita tambahkan FF yang ke empat pada rangkaian tersebut, FF yang ke-4 tersebut akan mempunyai frekuensi  $\frac{1}{16}$  dari frekuensi clock; dan seterusnya (menggunakan n FF akan menghasilkan frekuensi Output FF yang terakhir sebesar  $\frac{1}{2^n}$  dari frekuensi input.

Selain sebagai pembagi frekuensi (*frequency divider*). Rangkaian pada Gambar. 2 juga beroperasi sebagai pencacah biner (*binary counter*). Ini dapat diperlihatkan dengan menguji urutan kondisi FF setelah terjadinya pulsa clock.

Gambar 3 menunjukkan hasilnya di dalam bentuk tabel. Dimisalkan nilai-nilai X<sub>2</sub>X<sub>1</sub>X<sub>0</sub> mewakili bilangan biner dimana X<sub>2</sub> adalah posisi 2<sup>2</sup>, X<sub>1</sub> adalah posisi 2<sup>1</sup>, dan X<sub>0</sub> adalah posisi 2<sup>0</sup>. Delapan kondisi X<sub>2</sub>X<sub>1</sub>X<sub>0</sub> yang pertama di dalam tabel dikenal sebagai pencacah biner urut dari 000 sampai 111 (dalam desimal urut dari 0 sampai 7).

Dapat dilihat dari tabel pada Gambar 3 bahwa setelah pulsa yang ke-8 FF-FF

tersebut kembali ke kondisi 000, dan urutan biner mengulang sendiri untuk pulsa-pulsa berikutnya.



Gambar 2. J-K FF dirangkai sebagai Counter biner 3-bit (MOD-8)

$2^2$	$2^1$	$2^0$	
$X_2$	$X_1$	$X_0$	
0	0	0	Sebelum diberikan pulsa-pulsa jam
0	0	1	Setelah pulsa no. 1
0	1	0	Setelah pulsa no. 2
0	1	1	Setelah pulsa no. 3
1	0	0	Setelah pulsa no. 4
1	0	1	Setelah pulsa no. 5
1	1	0	Setelah pulsa no. 6
1	1	1	Setelah pulsa no. 7
0	0	0	Setelah pulsa no. 8 mengulang dari 000
0	0	1	Setelah pulsa no. 9
0	1	0	Setelah pulsa no. 10
0	1	1	Setelah pulsa no. 11
...	...	...	...

Gambar 3. Urutan kondisi FF menunjukkan urutan hitungan biner

### 2.1. Bilangan MOD

Counter pada 5.12 mempunyai  $2^3 = 8$  kondisi yang berbeda (000 sampai 111). Counter tersebut dapat dinamakan sebagai counter MOD-8, dimana bilangan MOD menunjukkan jumlah kondisi di dalam urutan hitungan. Jika FF yang ke-4 ditambahkan, urutan kondisi akan menghitung di dalam biner dari 0000 sampai 1111, totalnya adalah 16 kondisi. Ini akan dinamakan counter MOD-16.

Secara umum, jika N FF dihubungkan di dalam aturan pada Gb. 5.12, counter tersebut akan mempunyai  $2^N$  kondisi yang berbeda, sehingga counter tersebut adalah counter MOD- $2^N$ . Counter tersebut akan dapat menghitung naik sampai  $2^N-1$  sebelum kembali ke kondisi nol-nya.

Bilangan MOD pada counter juga menunjukkan pembagian frekuensi didapat dari FF yang terakhir. Sebagai contoh, counter 4-bit mempunyai 4 FF, masing-masing FF mewakili satu digit biner (binary digit/bit), sehingga ini adalah counter MOD- $2^4 = \text{MOD-16}$ . Sehingga counter tersebut dapat menghitung naik sampai 15 ( $= 2^4-1$ ). Counter tersebut juga dapat

digunakan untuk membagi frekuensi pulsa input dengan faktor 16 (bilangan MOD-nya).

MOD Number tersebut dari suatu Counter dapat ditentukan dengan rumus :

$$\text{MOD Number (maks)} = 2^N$$

N : Banyaknya FF yang dipakai untuk menyusun counter

Untuk mengetahui frekuensi output suatu counter digunakan rumus :

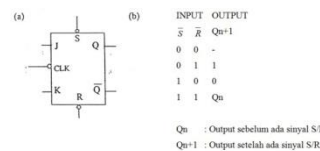
$$f_o = \frac{f_i}{\text{Mod Number}} \quad \dots\dots\dots (1)$$

$f_o$  = Frekuensi output

$f_i$  = Frekuensi input

### 2.2. Counter Asinkron Dengan MOD Tertentu

Untuk membuat suatu counter menghitung ke bilangan tertentu, maka diperlukan adanya rangkaian tambahan yang berfungsi untuk mereset counter bila hitungan counter telah mencapai angka yang diinginkan. Hal ini dapat dicapai dengan cara memanfaatkan fasilitas pin RESET pada FF pembentuk Counter.



Gambar 4. (a) JK FF (Toggle) dengan pin Set, Reset (b) Tabel Kebenaran

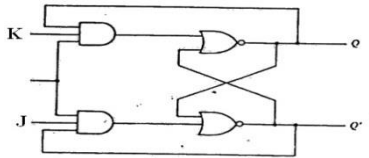
### 2.3. Mengenal JK Flip-flop

JK Flip-flop adalah perbaikan dari RS Flip-flop di dalam hal bahwa kondisi tidak menentu (*indeterminate state*) dari jenis RS adalah didefinisikan di dalam jenis JK.

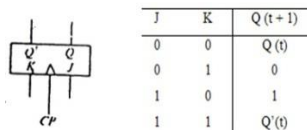
Input-input J dan K bertindak seperti input-input S dan R untuk men-set dan meng-clear flip-flop. Jika input-input diberikan ke kedua J dan K secara bersamaan, FF tersebut berubah ke kondisi komplementnya, yaitu, jika Q = 1, FF tersebut berubah ke Q = 0, dan sebaliknya.

Clocked JK Flip-flop ditunjukkan pada Gambar 5 (a). Output Q di-AND-kan dengan input K dan CP sehingga FF adalah

“clear” selama pulsa clock hanya jika Q sebelumnya adalah 1. Dengan hal yang sama, output Q’ di-AND-kan dengan input J dan CP sehingga FF adalah “set” dengan pulsa clock hanya jika Q’ sebelumnya adalah 1.



(a) Diagram Logika JK – FF



(b) Simbol Grafik dan Tabel Karakteristik

Gambar 5. Clocked JK Flip-flop

### 3. Merancang Decade Counter Asinkron Dengan JK Flip-flop

#### 3.1. Counter Dengan Mod-10 (Decade)

Untuk merancang Decade Counter (Counter Mod-10), perlu dibuat suatu rangkaian pengendali untuk mereset counter, bila hitungan counter mulai akan mencapai angka desimal 10, seperti ditunjukkan pada Tabel 2.

Tabel 2. Urutan hitungan Decade Counter

QD	QC	QB	QA	DESIMAL
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10

Kondisi untuk reset

Pada saat counter mencapai hitungan desimal 10, maka output  $Q_D = 1$ ,  $Q_C = 0$ ,  $Q_B = 1$ ,  $Q_A = 0$ . Output ini digunakan sebagai input bagi rangkaian pengendali reset. Dengan keadaan output seperti di atas, maka output pengendali akan menghasilkan logika 0. Output hitungan desimal 10 hanya terjadi sesaat, karena adanya sinyal reset yang menyebabkan counter reset ke kedudukan awal 0000.

Dengan menggunakan persamaan logika didapatkan.

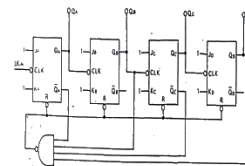
$$Y = \overline{Q_D} \overline{Q_C} Q_B \overline{Q_A}$$

Realisasi rangkaian digital dari persamaan di atas dapat dilihat pada Gambar 6.



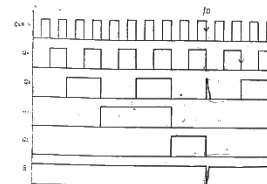
Gambar 6. Pengendali Reset Decade Counter

Diagram rangkaian Decade Counter Asinkron dapat dilihat pada Gambar 7. Counter tersebut dibuat dengan menggunakan 4 buah Flip-flop jenis JK dengan input J dan K berlogika 1. Dan yang mendapatkan input pulsa clock / pulsa hitungan adalah Flip-flop yang bobotnya paling rendah (LSB / Least Significant Bit), yaitu Flip-flop A.



Gambar 7. Diagram Rangkaian Decade Counter Asinkron

Untuk menunjukkan bahwa rangkaian tersebut akan menghitung mulai dari 0 s/d 9, dan hitungan ke 10 hanya digunakan sesaat untuk mereset semua Flip-flop sehingga hitungan langsung kembali ke 0 (recycle) dapat dilihat pada Gambar 8,



Gambar 8. Diagram Waktu Decade Counter

#### 3.2. Perbedaan Counter Sinkron Dengan Counter Asinkron

Perbedaan kedua jenis counter ini adalah pemicuannya. Pada sinkron counter, pemicuan counter terjadi serempak (dipicu oleh satu sumber clock) susunan flip flop

nya paralel. Sedangkan pada *asinkron counter*, minimal ada satu flip flop dipicu oleh keluaran flip flop lain atau dari sumber *clock* lain dan susunan flip flop nya seri. Dengan memanipulasi flip flop berdasarkan peta karnough atau *timing diagram* dapat dihasilkan *counter* acak, *shift counter* (*counter* sebagai fungsi *register*) atau juga *up down counter*.

### 3.2.1. Pencacah sinkron (Serempak)

- Masukan untuk denyut lonceng/clock dikembalikan secara serempak.
- Waktu penundaan counter adalah sama dengan penundaan satu flip-flop.
- Memerlukan sirkit clock yang berdaya tinggi, sebab sirkit lonceng/clock tersebut harus menggerakkan semua flip-flop secara serentak.
- Sering juga dinamakan pancacah jajar/paralel.

### 3.2.2. Pencacah Asinkron (tak serempak)

- Masukan untuk denut lonceng/clock dikembalikan secara tak serempak atau tak berurutan.
- Waktu penundaan counter adalah waktu semua penundaan flip-flop dijumlahkan.
- Memerlukan sirkit clock yang berdaya rendah, sebab hanya flip-flop yang paling awal saja yang dikendalikan oleh flip-flop.
- Sering juga dinamakan pencacah seri/pencacah biner.

## 4. Penutup

Seperti tersebut pada bagian sebelumnya *Asynchronous counter* tersusun atas flip-flop yang dihubungkan seri dan pemicuannya tergantung dari flip-flop sebelumnya, kemudian menjalar sampai flip-flop MSB-nya. Karena itulah *Asynchronous counter* sering disebut juga sebagai *ripple-through counter*.

Sebuah Counter Asinkron (Ripple) terdiri atas sederetan Flip-flop yang dikonfigurasi dengan menyambung outputnya dari yang satu ke yang lain. Yang

berikutnya sebuah sinyal yang terpasang pada input Clock FF pertama akan mengubah kedudukan outputnya apabila tebing (Edge) yang benar yang diperlukan terdeteksi.

Output ini kemudian mentrigger input clock berikutnya ketika terjadi tebing yang seharusnya sampai. Dengan cara ini sebuah sinyal pada inputnya akan meripple (mentrigger input berikutnya) dari satu FF ke yang berikutnya sehingga sinyal itu mencapai ujung akhir deretan itu. Ingatlah bahwa FF T dapat membagi sinyal input dengan faktor 2 (dua). Jadi Counter dapat menghitung dari 0 sampai  $2^n - 1$  (dengan n sama dengan banyaknya Flip-flop dalam deretan itu).

Pencacah asinkron yaitu pencacah yang disebut juga ripple trough counter atau serial counter karena masing – masing flip flop yang digunakan akan berguling (berubah kondisi dari 0 ke 1) atau sebaliknya. Secara berurutan atau langkah demi langkah hal ini disebabkan karena hanya flip flop yang paling ujung saja yang dikendalikan oleh sinyal clock. Sedangkan sinyal untuk flip flop yang lain diambil dari masing – masing flip flop sebelumnya.

## DAFTAR PUSTAKA

- Lee, Samuel, dan Sutisno, 1994, *Rangkaian Digital dan Rancangan Logika*, Penerbit Erlangga, Jakarta.
- Hodges, David, G. Jackson, Horace, dan H. Nasution, Sofyan, 1987, *Analisis Dan Desain Rangkaian Terpadu Digital*, Penerbit Erlangga, Jakarta.
- Mano, M. Morris, 2007, *Digital Design*, 5<sup>TH</sup> Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Millman Jacob dan Halkias Christos C, *Elektronika Terpadu Jilid 2*, Erlangga, Jakarta 1985
- Zoe Far, 2014, “*Dasar Rangkaian Counters*”, <http://zoefar.blogspot.com/2014/04/dasar-rangkaian-counter.html>, - download bulan November 2014