

**Mata Kuliah**  
**Pengantar Sistem Digital**

**Rangkaian Sekuensial – Flip – Flop Dasar**

**Dosen Pengampu:**  
**Arif Rifai Dwiyanto ST., MTI**



**Disusun Oleh:**  
**Inez Agatha (202410715023)**  
**F3A7**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS BHAYANGKARA JAKARTA**  
**RAYA**  
**2025**

## **DAFTAR ISI**

<b>BAB I PENDAHULUAN.....</b>	<b>3</b>
<b>1.1    Latar Belakang.....</b>	<b>3</b>
<b>1.2    Tujuan Praktikum .....</b>	<b>3</b>
<b>BAB II PERANCANGAN .....</b>	<b>4</b>
<b>2.1    Langkah-Langkah.....</b>	<b>4</b>
<b>2.1.1.    Membuat Rangkaian SR (Set-Reset) dan Clocked Flip-Flop .....</b>	<b>4</b>
<b>2.1.2.    Membuat Rangkaian D Flip-Flop .....</b>	<b>4</b>
<b>2.1.3.    Membuat Rangkaian JK Flip-Flop .....</b>	<b>5</b>
<b>2.1.4.    Membuat Rangkaian T (Toggle) Flip-Flop.....</b>	<b>5</b>
<b>BAB III HASIL DAN PEMBAHASAN .....</b>	<b>7</b>
<b>3.1    Rangkaian SR-Flip-Flop dan Clocked Flip Flop.....</b>	<b>7</b>
<b>3.2    Rangkaian D Flip-Flop .....</b>	<b>9</b>
<b>3.3    Rangkaian JK Flip-Flop.....</b>	<b>10</b>
<b>3.4    Rangkaian T (Toggle) .....</b>	<b>11</b>
<b>BAB IV KESIMPULAN.....</b>	<b>12</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam sistem digital sendiri, ada dua jenis rangkaian utama: rangkaian kombinasi dan rangkaian sekuensial. Rangkaian kombinasi hanya bergantung pada input yang masuk saat itu juga, seperti Dekoder, Enkoder, Multiplexer, Demultiplexer termasuk beberapa rangkaian logika kombinasional.

Sementara itu, contoh rangkaian sekuensial salah satunya adalah Flip-Flop. Flip-flop adalah rangkaian digital yang dipakai untuk menyimpan satu bit data. Data ini bisa tetap tersimpan selama belum ada sinyal yang menyuruh untuk mengubah atau menghapusnya. Secara umum, flip-flop dibuat dari komponen dasar seperti transistor, resistor, dan diode yang disusun menjadi gerbang logika sehingga bisa bekerja secara sekuensial atau mempunyai ingatan.

Rangkaian sekuensial punya kemampuan menyimpan informasi melalui elemen memori. Kemampuan untuk menyimpan *state* ini sangat penting agar suatu rangkaian dapat melakukan fungsi seperti counter, register, memori, sampai rangkaian kontrol yang lebih rumit. Tanpa flip-flop, semua rangkaian yang butuh pencatatan keadaan sebelumnya tidak akan bisa dijalankan.

Sementara itu, flip-flop juga sering digunakan sebagai blok dasar dalam membuat rangkaian yang lebih besar, seperti clock divider, shift register, dan berbagai jenis pencacah. Maka flip-flop bisa digunakan sebagai pondasi dari banyak sistem digital modern. Dengan memahami cara kerja flip-flop, kita bisa lebih mudah memahami rangkaian-rangkaian digital yang sifatnya lebih kompleks.

### **1.2 Tujuan Praktikum**

- 1.2.1 Mahasiswa memahami konsep rangkaian sequential.
- 1.2.2 Mahasiswa mampu merancang dan mensimulasikan flip-flop dasar (SR, D, dan JK).
- 1.2.3 Mahasiswa dapat menganalisis hasil simulasi dari setiap jenis flip-flop.

## **BAB II**

### **PERANCANGAN**

#### **2.1 Langkah-Langkah**

##### **2.1.1. Membuat Rangkaian SR (Set-Reset) dan Clocked Flip-Flop**

1. Buka software Logisim Evolution.
2. Membuat rangkaian SR Flip-Flop menggunakan dua gerbang logika NOR dari library gates.
3. Tambahkan dua input pin untuk S(Set) dan R(Reset).
4. Hubungkan pin S dan R ke gerbang logika NOR masing-masing.
5. Hubungkan output NOR pertama ke input NOR kedua, dan output NOR kedua ke input NOR pertama (umpan balik).
6. Tambahkan dua LED dan hubungkan ke output Q dan Q' sebagai output dari masing-masing NOR.
7. Hubungkan output NOR pertama ke input NOR kedua, dan output NOR kedua ke input NOR pertama (umpan balik).

##### **Rangkaian Clocked**

1. Tambahkan hasil rangkaian SR ke area kerja.
2. Tambahkan dua gerbang logika AND yang tersusun dibelakang rangkaian SR.
3. Tambahkan input untuk S, R dan Clock.
4. Lalu hubungkan input S dan R ke salah satu gerbang AND, dan input clock dihubungkan ke gerbang AND.
5. Hubungkan output AND ke input S dan R pada rangkaian SR flip-flop

##### **2.1.2. Membuat Rangkaian D Flip-Flop**

1. Gunakan rangkaian SR flip-flop yang sudah dibuat sebelumnya.
2. Tambahkan dua gerbang logika AND yang tersusun di depan rangkaian SR flip-flop.

3. Tambahkan input untuk D(data) dan Clock.
4. Tambahkan satu gerbang NOT.
5. Hubungkan:
  - Input D ke salah satu gerbang AND
  - Output NOT ke gerbang AND lainnya
  - Input clock ke dua gerbang AND
6. Hubungkan output AND ke input S dan R pada rangkaian SR flip-flop.

### **2.1.3. Membuat Rangkaian JK Flip-Flop**

1. Tambahkan rangkaian SR Flip-Flop yang sudah dibuat sebelumnya.
2. Tambahkan dua gerbang AND di depan input S dan R, lalu ubah inputan menjadi 3.
3. Tambahkan input J, K, dan Clock
4. Hubungkan:
  - J ke salah satu input AND pertama
  - K ke salah satu input AND kedua
  - Clock hubungkan ke dua gerbang AND
5. Hubungkan output Q' ke input J sebagai umpan balik.
6. Hubungkan output Q ke input K sebagai umpan balik.
7. Hubungkan output J dan K ke rangkaian SR AND.

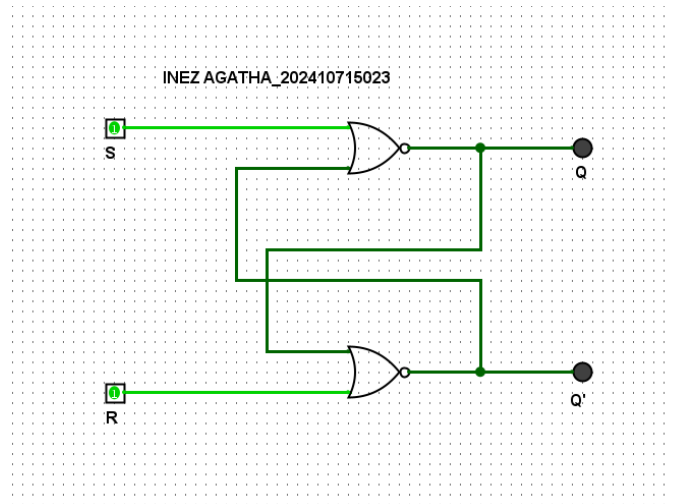
### **2.1.4. Membuat Rangkaian T (Toggle) Flip-Flop**

1. Tambahkan rangkaian SR Flip-Flop yang sudah dibuat sebelumnya.
2. Tambahkan dua input pin untuk T dan Clock.
3. Tambahkan dua gerbang AND.
4. Hubungkan:
  - Input T ke dua gerbang AND

- Input Clock ke dua gerbang AND
5. Hubungkan AND pertama ke output NOR pertama sebagai output Q
  6. Hubungkan AND kedua ke output NOR kedua sebagai output Q'
  7. Tambahkan dua LED dan hubungkan ke output Q dan Q' sebagai output dari masing-masing NOR.

## BAB III HASIL DAN PEMBAHASAN

### 3.1 Rangkaian SR-Flip-Flop dan Clocked Flip Flop



*Gambar 3.1 Rangkaian SR-Flip-Flop*

*Tabel 3.1 Truth Table SR-Flip-Flop 1*

S	R	Q	Q'	Ket
0	0	Q(t)	Q(t)	Hold
0	1	1	0	set
1	0	0	1	Reset
1	1	0	0	Invalid

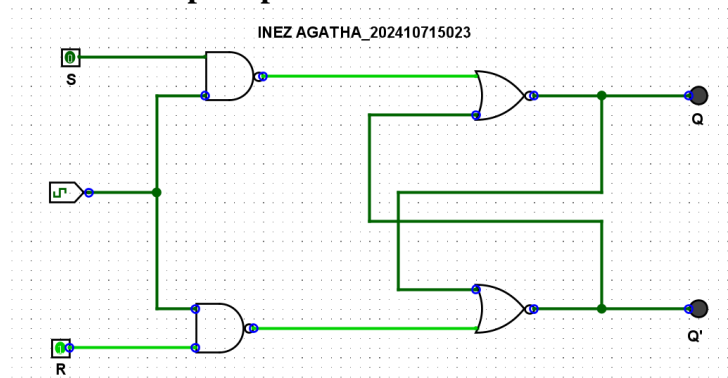
Pada rangkaian SR-Flip-Flop yang berfungsi sebagai menyimpan satu bit data. Praktikum ini membuat rangkaian dengan menggunakan gerbang logika NOR yang akan menghubungkan secara umpan balik.

Rangkaian ini menghasilkan sebuah tabel kebenaran, dimana kondisi  $S = 1$  dan  $R = 0$ , rangkaian akan berada pada kondisi *set*, sehingga output Q bernilai 1 dan Q' bernilai 0. Sedangkan,  $S = 0$  dan  $R = 1$ , rangkaian berada pada kondisi *reset*, sehingga output Q bernilai 0 dan Q' bernilai 1. Kedua kondisi ini menunjukkan bahwa SR Flip-Flop dapat menyimpan keadaan sesuai dengan input yang diberikan.

S dan R bernilai 0 maka output Q maupun Q' akan bernilai no change(hold), hal ini terjadi karena tidak ada proses yang diberikan yang menyebabkan output tetap menggunakan nilai sebelumnya. Namun, pada simulasi menggunakan logisim

evolution, akan menemukan kondisi dimana S dan R 0 akan menghasilkan output  $Q=0$  dan  $Q'=1$ . Hal ini dapat terjadi karena Logisim menentukan kondisi awal rangkaian. Sedangkan pada kondisi S dan R bernilai 1 maka output akan menghasilkan 0 (invalid), karena kedua output bernilai 0 yang tidak saling umpan balik, maka ini dilarang agar tidak menyebabkan ketidakstabilan, sehingga kondisi ini tidak digunakan.

### Rangkaian Clocked SR-Flip-Flop



Gambar 3.1 Rangkaian Clock SR-Flip-Flop

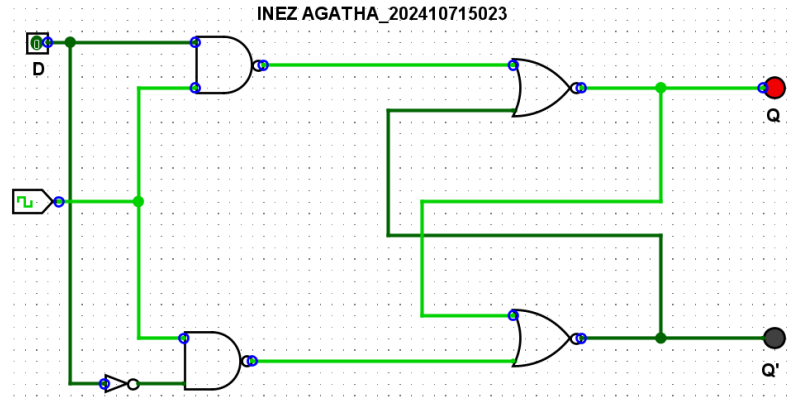
Tabel 3.1 Truth Table Clock SR-Flip-Flop

S	R	Clk	$Q(t+1)$	Ket
x	x	0	$Q(t)$	Tidak berubah
0	0	1	$Q(t)$	Hold
0	1	1	0	Reset
1	0	1	1	Set
1	1	1	0	Invalid

Rangkaian Clock SR-Flip-Flop berfungsi untuk mengubah atau menyimpan data. Saat clock bernilai 0. Rangkaian akan dalam kondisi tidak aktif, dan perubahan pada input S dan R tidak akan mempengaruhi output. Karena rangkaian hanya menyimpan data yang sudah ada sebelumnya. Sedangkan rangkaian clock bernilai 1, input S dan R dapat mempengaruhi output, seperti S bernilai 1 dan R bernilai 0, maka output bernilai 1. Jika S bernilai 0 dan R bernilai 1, maka bernilai 0. Clock akan mempengaruhi output jika ia bernilai 1.



### 3.2 Rangkaian D Flip-Flop



Gambar 3.2 Rangkaian D Flip-Flop

Table 3.2 Truth Table D flip-flop

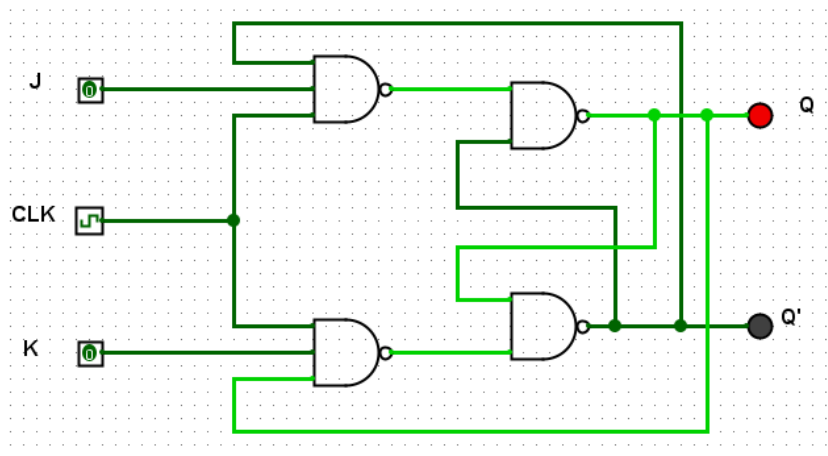
Clk	D	Q(t+1)	Ket
0	0	Q(t)	Hold
0	1	Q(t)	Hold
1	0	0	0
1	1	1	1

Rangkaian D flip-flop yang terdiri dari NOT, AND, dan NOR. Dimana Not digunakan untuk membedakan kondisi  $D=1$  dan  $D=0$ , yang akan menghasilkan kebalikan dari  $D$  menjadi  $D'$  atau yang disebut sebagai umpan balik (feedback). Rangkaian ini bekerja dengan cara menyimpan nilai input  $D$  ke output  $Q$  berdasarkan sinyal clock.

Dengan kondisi clock yang aktif akan menghasilkan nilai output yang sama dengan nilai input  $D$ . Artinya, jika nilai  $D$  berubah-ubah, maka output  $Q$  akan mengikuti saat clock dalam kondisi aktif. Hal ini digambarkan pada tabel kebenaran, jika  $D$  bernilai 1, maka output akan bernilai 1. Sedangkan  $D$  bernilai 0, maka output akan bernilai 0. Dimana output akan selalu menghasilkan nilai yang sama dengan  $D$ .

Ketika clock bernilai 0, rangkaian tidak akan mempengaruhi perubahan input  $D$ . Output  $Q$  akan menyimpan nilai sebelumnya. Dijelaskan pada tabel, meskipun  $D$  bernilai 1/0 tetapi clock bernilai 0, output akan menghasilkan nilai sebelumnya atau kondisi hold.

### 3.3 Rangkaian JK Flip-Flop



Gambar 3.3 Rangkaian JK Flip-Flop

Tabel 3.3 Truth Table JK Flip-Flop

J	K	Q(t+1)	KET
0	0	Q(t)	Hold
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Toggle

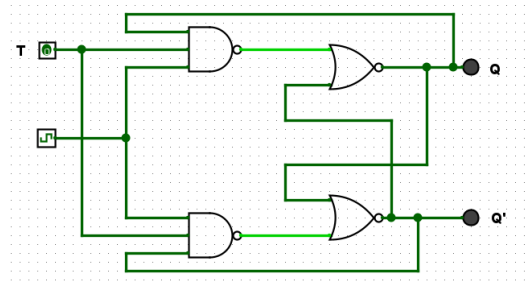
JK Flip-Flop adalah rangkaian yang bertujuan untuk menghilangkan kondisi invalid ketika kedua input aktif secara bersamaan. Pada rangkaian ini, input J sebagai Set, dan input K sebagai Reset. Ketika J dan K = 1 maka rangkaian akan tetap stabil, dan melakukan proses toggle dengan membalik nilai output sebelumnya. Dengan adanya fitur toggle, JK Flip-Flop sangat cocok digunakan pada rangkaian counter.

Pada kondisi clock tidak aktif (0), rangkaian tidak akan mengubah input J dan K. Dimana nilai J dan K berubah-ubah, output akan tetap menyimpan nilai sebelumnya. Dengan ini clock berperan sebagai pengunci (enable) agar data tetap stabil.

Berbeda pada kondisi clock aktif (1) yang dapat mempengaruhi nilai output, Jika J = 1 dan K = 0, output akan bernilai 1. Jika J = 0 dan K = 1, output akan bernilai 0. Saat J = 0 dan K = 0, output akan menyimpan nilai sebelumnya.

Sifat toggle memberikan rangkaian JK flip-flop sebagai rangkaian pencacah (counter) dan menjadikannya rangkaian yang lebih stabil dibandingkan SR flip-flop karena tidak ada kondisi invalid.

### 3.4 Rangkaian T (Toggle)



Gambar 3.4 Rangkaian Toggle

Table 3.4 Truth Table Rangkaian Toggle

T	Clock	Q(t+1)	KET
0	1	Q(t)	Hold
1	1	Q(t)	Toggle

Rangkaian T flip-flop bekerja berdasarkan sinyal clock dan satu input utama. Rangkaian ini digunakan untuk membalik nilai output sebelumnya (toggle) ketika kondisi sudah terpenuhi. Flip-flop ini digunakan pada rangkaian pencacah (counter). Saat clock 0, maka rangkaian tidak akan melakukan perubahan, dan output tidak akan berubah karena flip-flop belum diizinkan untuk memperbaharui datanya.

Sedangkan saat clock 1, dan  $T = 0$  output akan bernilai seperti sebelumnya, karena flip-flop hanya menyimpan data dan tidak melakukan perubahan. Dan kondisi  $T = 1$ , output akan dibalik dengan proses toggle menjadi  $Q = 0$  maka  $Q' = 1$  atau sebaliknya.

## **BAB IV**

### **KESIMPULAN**

Pada praktikum rangkaian sekuensial ini, dapat disimpulkan bahwa flip-flop adalah bagian yang sangat penting dalam sistem digital, karena mempunyai kemampuan untuk menyimpan data. Memiliki perbedaan dengan rangkaian kombinasi yang bergantung pada input saat itu, rangkaian sekuensial dipengaruhi kondisi sebelumnya karena ada memori. Dengan ini, rangkaian sekuensial dapat digunakan untuk pada rangkaian counter, rangkaian register, dan rangkaian kontrol.

Rangkaian SR flip-flop ditentukan oleh kombinasi input Set(S) dan Reset(R). Rangkaian ini dapat menyimpan data ketika kondisi input bernilai 0, jika kedua input bernilai 1 maka akan menghasilkan keadaan invalid yang mengakibatkan output tidak stabil. Pada rangkaian SR flip-flop, kondisi ini harus dihindari.

Pada Rangkaian D flip-flop hanya memiliki satu input data (D), yang tidak ada kondisi terlarang. Output Q akan mengikuti nilai input D saat sinyal clock aktif. Maka D flip-flop lebih stabil ketika digunakan sebagai penyimpan data.

Rangkaian JK flip-flop tidak memiliki kondisi invalid, dengan menghilangkan kondisi terlarang dan memiliki fungsi toggle ketika kedua input bernilai 1. Sedangkan T flip-flop bekerja dengan cara membalik output setiap clock aktif jika  $T = 1$ .