

# **LAPORAN PRAKTIKUM**

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

## **RANGKAIAN SEKUENSIAL – FLIP FLOP DASAR**

**Dosen: Arif R. Dwiyanto ST., MTI.**



**Disusun oleh:**

**Khalila indana (202410715127)**

**F3A6**

**PROGRAM STUDI INFORMATIKA FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BHAYANGKARA JAKARTA RAYA**

**2025**

## BAB I

### PENDAHULUAN

#### 1.1 Tujuan Praktikum

Mahasiswa diharapkan mampu:

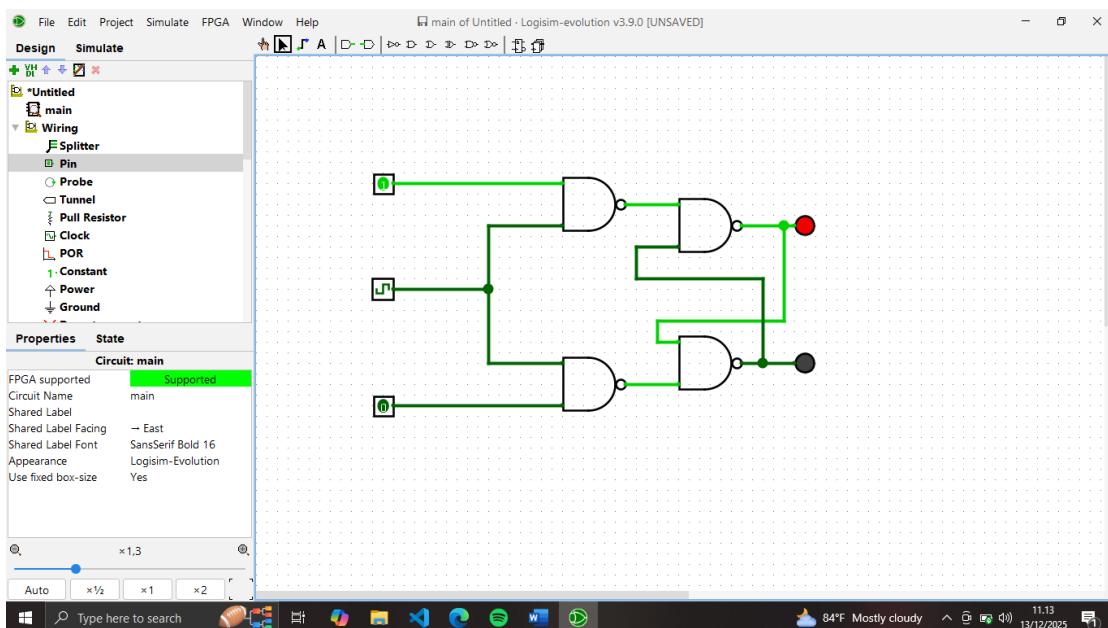
- a. Memahami konsep rangkaian sekuensial.
- b. Merancang dan mensimulasikan *flip-flop* dasar.
- c. Menganalisis pengaruh *clock* dan *input* terhadap *output flip-flop*.
- d. Memahami peran *flip-flop* sebagai memori satu bit digital.

#### 1.2 Langkah-langkah Praktikum

- a. SR Flip-Flop: Rangkaian dibangun menggunakan gerbang NOR/NAND yang dihubungkan umpan balik, dipasangi LED pada *output Q* dan  $\bar{Q}$ , kemudian diuji menggunakan kombinasi input S dan .
- b. D Flip-Flop: Rangkaian SR yang sudah ada dimodifikasi dengan menambahkan gerbang NOT pada input R agar output selalu mengikuti input D saat *clock* aktif, dan hasil simulasinya diverifikasi dengan tabel kebenaran.
- c. JK Flip-Flop: Rangkaian dibangun di atas dasar SR dengan penambahan rangkaian pengendali, lalu diuji secara bertahap menggunakan sinyal *clock* untuk memverifikasi semua kombinasi input J dan K, termasuk mode *toggle*.
- d. T Flip-Flop: Komponen *flip-flop* dipilih, disambungkan dengan sumber *clock*, dan hasil simulasi dijalankan untuk mengamati *output Q* yang berubah (*toggle*) setiap kali *clock* diaktifkan.

## 1.3 Hasil Simulasi

### 1. Rangkaian SR Flip-Flop



Rangkaian hasil praktikum SR Flip-Flop berclock berbasis gerbang NAND. Dua gerbang NAND di sebelah kanan disusun saling silang (cross-coupled) sehingga membentuk elemen penyimpanan data dengan keluaran  $q$  dan  $q'$ . Dua gerbang NAND di sebelah kiri berfungsi sebagai pengendali input set (S) dan reset (R) yang digabungkan dengan sinyal clock.

Sinyal clock masuk ke kedua gerbang NAND pengendali, sehingga input S dan R hanya diteruskan ke latch saat clock bernilai aktif (1). Ketika clock tidak aktif, maka output dari gerbang pengendali tidak berubah, sehingga rangkaian mempertahankan kondisi sebelumnya (hold). Dengan demikian, perubahan pada set atau reset tidak langsung memengaruhi output jika clock tidak aktif.

Saat clock aktif, kombinasi input S dan R akan menentukan keadaan output. Jika S aktif maka rangkaian berada pada kondisi set, jika R aktif maka berada pada kondisi reset, dan jika S dan R tidak aktif maka output akan mempertahankan keadaan sebelumnya.

TABEL KEBENARAN

- Saat clock = 1 (aktif)

S	R	Q(t+1)	Ket
0	0	Q(t)	Hold/tidak berubah
0	1	0	Reset
1	0	1	Set
1	1	x	Kondisi terlarang

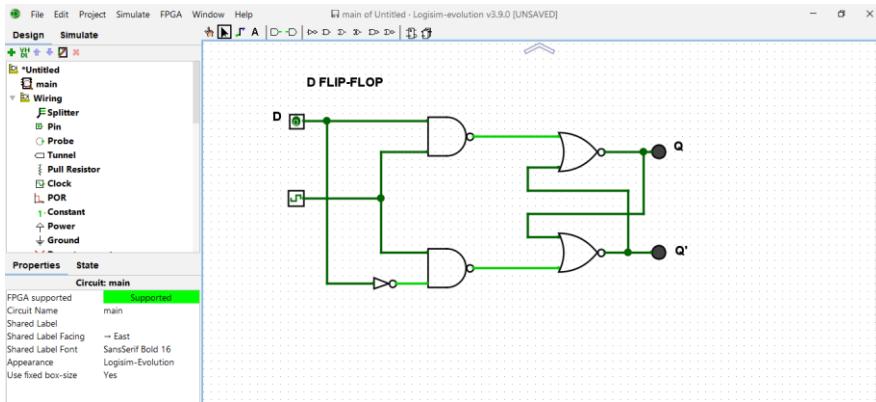
Pada saat clock bernilai 1 atau aktif, SR Flip-Flop berada dalam kondisi bekerja. Jika  $S = 0$  dan  $R = 0$ , maka tidak ada perubahan pada output sehingga flip-flop mempertahankan keadaan sebelumnya (hold). Jika  $S = 0$  dan  $R = 1$ , maka rangkaian berada pada kondisi reset sehingga keluaran q menjadi 0. Jika  $S = 1$  dan  $R = 0$ , maka rangkaian berada pada kondisi set sehingga keluaran q menjadi 1. Sedangkan kondisi  $S = 1$  dan  $R = 1$  merupakan kondisi terlarang karena dapat menyebabkan output tidak terdefinisi.

- Saat clock = 0 (tidak aktif)

S	R	Q(t+1)	Ket
x	x	Q(t)	Hold

Pada saat clock bernilai 0 atau tidak aktif, SR Flip-Flop merespons perubahan input S dan R. Berapapun nilai S dan R, output q akan tetap mempertahankan keadaan sebelumnya. Kondisi ini menunjukkan fungsi clock sebagai pengendali waktu perubahan output, sehingga flip-flop hanya berubah saat clock aktif.

## 2. Rangkaian D Flip-Flop



Rangkaian pada hasil praktikum D Flip-Flop ber-clock yang dirangkai dari kombinasi gerbang logika SR latch berbasis NOR. Input utama rangkaian adalah D (data) dan clock, sedangkan output-nya berupa q dan q' yang saling berkomplemen.

Sinyal D dibagi menjadi dua jalur, yaitu jalur langsung dan jalur yang melewati gerbang not (inverter) sehingga menghasilkan d dan d'. Kedua sinyal ini kemudian masing-masing digabungkan dengan sinyal clock melalui gerbang logika di bagian kiri. Hasil dari proses ini membentuk sinyal set (S) dan reset (R) yang saling berlawanan, sehingga kondisi terlarang pada SR Flip-Flop dapat dihindari.

Dua gerbang NOR dibagian kanan disusun secara silang membentuk SR latch yang berfungsi sebagai penyimpanan data. Ketika clock aktif, nilai D akan menentukan keadaan output. Jika D = 1, maka rangkaian berada pada kondisi set sehingga Q = 1 dan Q' = 0. Jika D = 0, maka rangkaian berada pada kondisi reset sehingga Q = 0 dan Q' = 1. Saat clock tidak aktif, sinyal set dan reset tidak berubah sehingga SR latch mempertahankan keadaan sebelumnya.

TABEL KEBENARAN

- Saat clock = 1 (aktif)

D	Q(t+1)	Ket
0	0	Reset
1	1	Set

Pada saat clock bernilai 1 atau aktif, D Flip-Flop akan membaca nilai input D dan langsung menyimpan ke output-mya. Jika D = 0, maka flip-flop berada pada kondisi

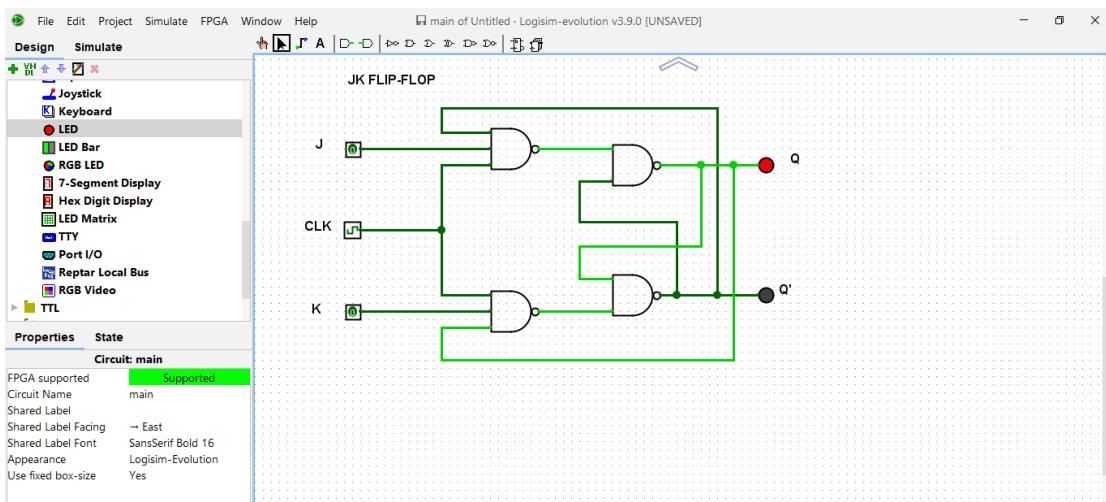
reset sehingga output q menjadi 0. Jika  $D = 1$ , maka flip-flop berada pada kondisi set sehingga output q menjadi 1. Pada kondisi ini, nilai output selalu mengikuti input D.

- Saat clock = 0 (tidak aktif)

D	$Q(t+1)$	Ket
x	$Q(t)$	Hold / tidak berubah

Pada saat clock bernilai 0 atau tidak aktif, D Flip-Flop tidak merespons perubahan pada input D. Berapapun nilai D, output q akan tetap mempertahankan keadaan sebelumnya. Kondisi ini menunjukkan fungsi flip-flop sebagai elemen penyimpan data, di mana nilai hanya berubah ketika clock aktif dan disimpan ketika clock tidak aktif.

### 3. Rangkaian JK Flip-Flop



Rangkaian hasil praktikum JK Flip-Flop ber-clock yang dirangkai dari kombinasi gerbang logika dan SR latch berbasis NAND. Rangkaian ini memiliki dua input utama, yaitu J dan K, serta satu sinyal clock sebagai pengendali waktu, dengan output q dan  $q'$  yang saling berkomplemen.

Pada bagian kiri rangkaian, input J dan K masing-masing digabungkan dengan sinyal clock dan umpan balik dari output melalui gerbang NAND. Umpan balik ini berfungsi untuk menghilangkan kondisi terlarang yang terdapat pada SR Flip-Flop. Hasil dari proses ini menghasilkan sinyal set dan reset yang terkontrol.

Dua gerbang NAND pada bagian kanan disusun secara silang membentuk SR latch yang berfungsi menyimpan keadaan logika. Ketika clock aktif, kombinasi input J dan K akan menentukan keadaan output. Jika  $J = 0$  dan  $K = 0$ , output akan mempertahankan

keadaan sebelumnya. Jika  $J = 1$  dan  $K = 0$ , flip-flop berada pada kondisi set sehingga  $q = 1$ . Jika  $J = 0$  dan  $K = 1$ , flip-flop berada pada kondisi reset sehingga  $q = 0$ . Sedangkan jika  $J = 1$  dan  $K = 1$ , output akan berpindah keadaan (toggle) pada setiap clock. Saat clock tidak aktif, sinyal set dan reset tidak berubah sehingga SR latch mempertahankan keadaan sebelumnya.

TABEL KEBENARAN

- Saat clock = 1 (aktif)

J	K	Q(t+1)	Ket
0	0	Q(t)	Hold /tidakberubah
0	1	0	Reset
1	0	1	Set
1	1	Q(t)	Toggle

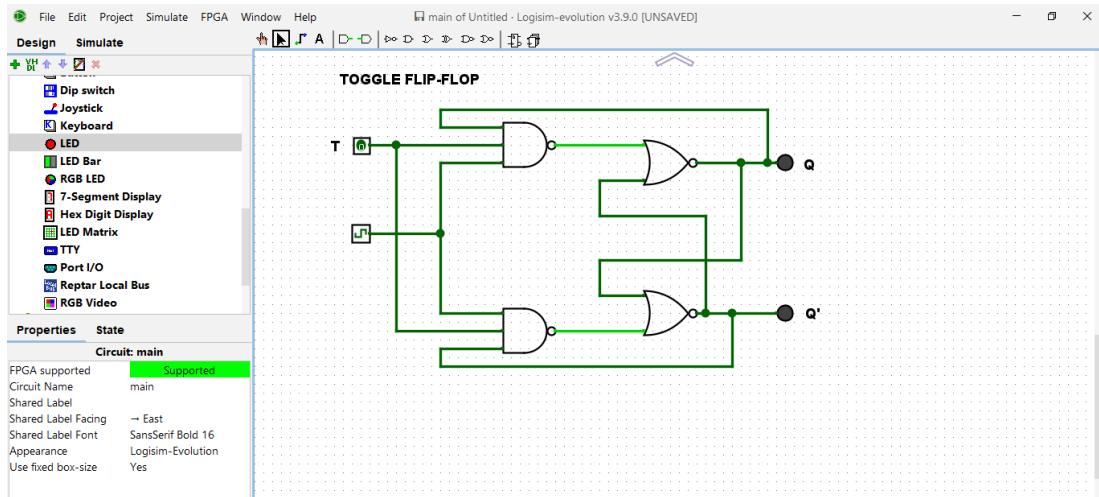
Pada saat clock bernilai 1 atau aktif, JK Flip-Flop merespons kombinasi input J dan K. Jika  $J = 0$  dan  $K = 0$ , output akan mempertahankan keadaan sebelumnya. Jika  $J = 1$  dan  $K = 0$ , flip-flop berada pada kondisi set sehingga  $q = 1$ . Jika  $J = 0$  dan  $K = 1$ , flip-flop berada pada kondisi reset sehingga  $q = 0$ . Sedangkan jika  $J = 1$  dan  $K = 1$ , output akan berpindah keadaan (toggle) pada setiap clock.

- Saat clock = 0 (tidak aktif)

J	K	Q(t+1)	Ket
x	x	Q(t)	Hold

Pada saat clock bernilai 0 atau tidak aktif, perubahan nilai J dan k tidak memengaruhi output. Flip-Flop akan tetap menyimpan keadaan terakhirnya yang menunjukkan fungsi clock sebagai pengendali perubahan output pada rangkaian sequential.

#### 4. Rangkaian T (Toggle)



Rangkaian hasil praktikum Toggle Flip-Flop ber-clock dirangkai dari kombinasi gerbang logika dan SR latch berbasis NAND dan NOR. Rangkaian ini memiliki satu input utama (T) serta satu sinyal clock sebagai pengendali waktu, dengan output q dan q' yang saling berkomplemen.

Pada bagian kiri rangkaian, input T digabungkan dengan sinyal clock melalui gerbang NAND sehingga membentuk sinyal kendali menuju bagian penyimpanan. Proses ini berfungsi untuk mengatur kapan flip-flop diperbolehkan mengubah keadaannya, yaitu hanya saat clock berada pada kondisi aktif.

Dua gerbang NOR pada bagian kanan disusun secara silang membentuk SR latch yang berfungsi menyimpan keadaan logika. Ketika clock aktif dan  $T = 1$ , output q akan berpindah ke keadaan kebalikan dari kondisi sebelumnya (toggle). Jika sebelumnya  $q = 0$ , maka akan menjadi 1, dan sebaliknya. Namun, ketika  $T = 0$ , output akan mempertahankan keadaan sebelumnya. Saat clock tidak aktif, sinyal kendali tidak memengaruhi SR latch sehingga rangkaian tetap menyimpan nilai terakhirnya.

#### TABEL KEBENARAN

- Saat clock 1 (aktif)

T	Q (t)	Q (t+1)	Ket
0	0	0	Hold/tidak berubah
0	1	1	Hold/tidak berubah
1	0	1	Toggle/ubah ke 1

1	1	0	Toggle/ubah ke 0
---	---	---	------------------

Pada saat *clock* bernilai 1 atau aktif, T Flip-Flop merespons input T. Jika T = 0, output mempertahankan keadaan sebelumnya (*hold*). Jika T = 1, output berbalik dari keadaan sebelumnya (*toggle*). Flip-flop hanya berubah saat ada pulsa *clock* aktif.

- Saat *clock* = 0 (tidak aktif)

T	Q(t)	Q(t+1)	Ket
X	0	0	Hold
X	1	1	Hold

Saat *clock* bernilai 0 atau tidak aktif, perubahan input T tidak memengaruhi output.

Flip-flop tetap menyimpan keadaan terakhir, menunjukkan fungsi *clock* sebagai pengendali perubahan output pada rangkaian sequential.

#### 1.4 Kesimpulan

Flip-flop adalah komponen utama rangkaian sekvensial yang keluarannya dipengaruhi oleh kombinasi antara input saat ini dan keadaan sebelumnya. Hasil praktikum menunjukkan fungsi spesifik masing-masing jenis *flip-flop*: SR Flip-Flop digunakan untuk menentukan kondisi *set*, *reset*, atau *hold*. Sementara itu, pada D Flip-Flop, nilai input D secara langsung akan direfleksikan pada *output* Q ketika *clock* aktif. JK Flip-Flop menawarkan fungsionalitas terlengkap dengan mampu melakukan *set*, *reset*, *hold*, dan *toggle*. Terakhir, T Flip-Flop menyederhanakan operasi menjadi hanya mempertahankan (*hold*) atau membalikkan (*toggle*) keadaan output. Kunci dari semua operasi ini adalah sinyal *clock*, yang berfungsi sebagai pengendali waktu untuk memastikan perubahan *output* pada seluruh *flip-flop* terjadi secara serempak hanya saat *clock* tersebut aktif.