

## BAB. III

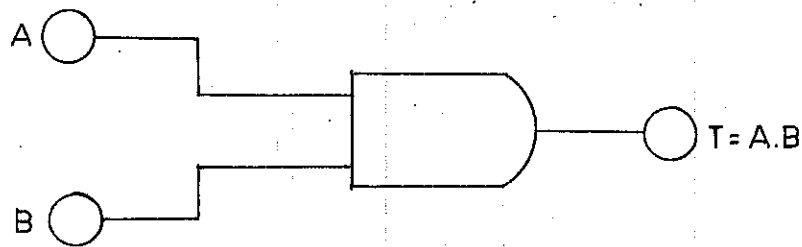
### RANGKAIAN LOGIKA, FLIP FLOP DAN REGISTER

#### 3.1. RANGKAIAN LOGIKA

##### 3.1.1. GERBANG DAN ( AND GATE )

Gerbang DAN ( *AND GATE* ) atau dapat pula disebut pintu DAN, adalah suatu rangkaian logika yang mempunyai beberapa jalan masuk ( *input* ) dan hanya mempunyai satu jalan keluar ( *output* ).

Simbol atau tanda gambarnya ditunjukkan pada gambar 3.1 di bawah ini.



Gambar 3.1. Simbol Gerbang DAN

Tabel 3.1. Kebenaran :

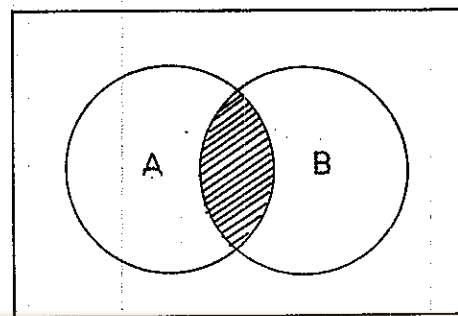
A	B	T
0	0	0
1	0	0
0	1	0
1	1	1

Dari gambar di atas dapat dilihat bahwa *output* gerbang DAN merupakan fungsi perkalian logika dari *variable inputnya*.

*Output* T = "1", hanya bila A = "1" dan B = "1".

Fungsi DAN juga dapat digambarkan dengan diagram Venn.

Jika A menyatakan himpunan dari swit "lampu yang tersambung" dan B himpunan dari "kunci kontak yang tersambung" maka T merupakan himpunan dari lampu yang menyala dan kunci kontak tersambung dan disajikan oleh bidang bergaris yang dicakup baik oleh A maupun oleh B.



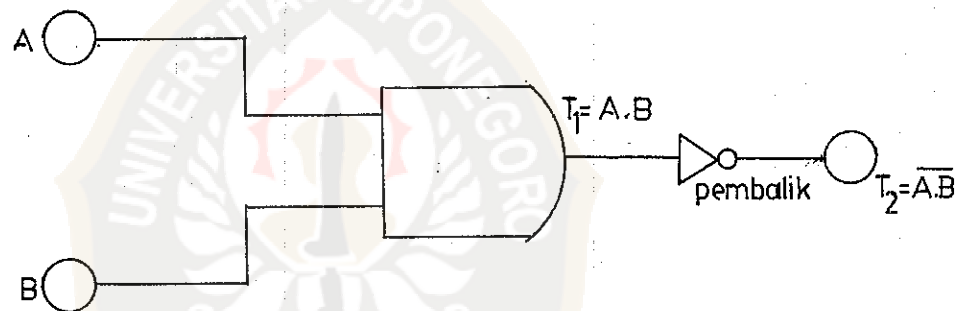
Gambar 3.2. Diagram Venn.

### 3.1.2. GERBANG NAND ( NAND GATE )

Kata NAND diambil dari 2 kata Inggris *NOT - AND* ( tidak DAN, bukan DAN ).

Bentuk dasar NAND adalah penyangkalan dari jalinan AND ( DAN ).

Realisasinya diwujudkan dengan mudah dengan rangkaian seri gerbang AND dan *Inverter* ( pem-balik ).



Gambar 3.3a. Gerbang NAND.

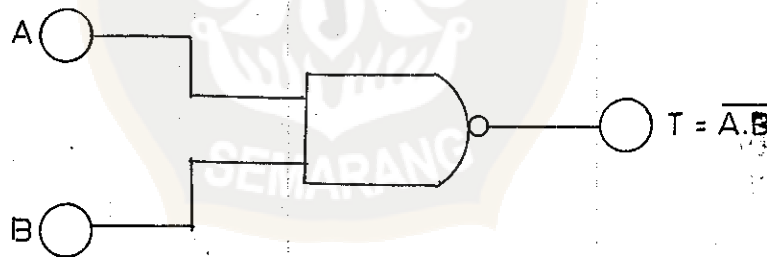
Tabel 3.2. Kebenaran :

A	B	T1	T2
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

Sesuai gambar dan tabel kebenaran di atas, *output* gerbang NAND akan ada ( $T_2 = 1$ ) bila salah satu atau kedua variabel *inputnya* adalah "1", maka *output*  $T_2 = 0$ .

Fungsi  $T_1 = A.B$  adalah konjungsi Inverter yang berikut membalik jalinan AND.A.B menjadi A.B. selaras dengan disatukannya dua kata *NOT AND* menjadi satu kata NAND.

Dalam praktek direalisasikan juga kedua komponen tersebut menjadi satu komponen logika. Rangkaian elektronika demikian disebut gerbang NAND dan disimbolkan seperti pada gambar berikut :



Gambar 3.3b. Gerbang NAND.

### 3.1.3. GERBANG ATAU ( OR GATE )

Disamping jalinan DAN yang telah dikenal, pengkaitan dapat juga dilakukan dengan kata "ATAU". Jalinan "ATAU" menghubungkan dua variabel secara disjungtip ( memisah, menceraikan ) untuk memberi suatu besaran keluaran. Jalinan "ATAU" disebut juga

DISJUNGSI, dan secara aljabar disebut fungsi OR.

Dapat dibedakan dua macam disjungsi atau fungsi OR, yaitu :

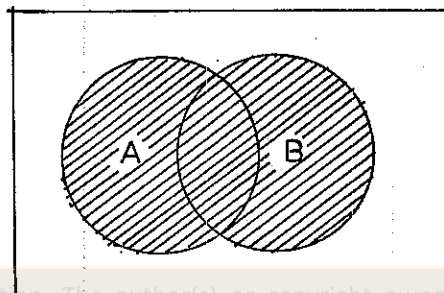
- a. Disjungsi Inklusip, atau Fungsi OR Inklusip.
- b. Disjungsi Eksklusip, atau Fungsi OR Eksklusip.

a. Tabel. 3.3. Kebenaran dari Disjungsi Inklusip atau Fungsi OR :

A	B	$T = A + B$
0	0	0
1	0	1
0	1	1
1	1	1

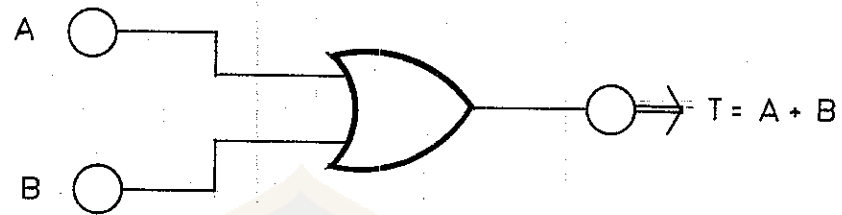
Fungsi OR Inklusip akan menghasilkan logika 1, bila paling sedikit satu variabel berlogika 1.

Diagram Venn untuk Fungsi OR adalah sebagai berikut :



Gambar 3.5. Diagram Venn.

Rangkaian elektronika yang mewujudkan jalinan ATAU disebut gerbang ATAU atau OR GATE, simbol penyambungan untuk Gerbang ATAU adalah :



Gambar 3.6. Gerbang ATAU

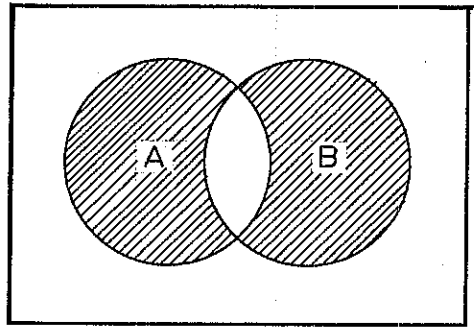
b. Tabel 3.4. Kebenaran dari Disjungsi Eksklusip.

A	B	$T = A + B$
0	0	0
1	0	1
0	1	1
1	1	0

Fungsi T hanya bernilai 1, bila kedua variabel mempunyai nilai yang berbeda.

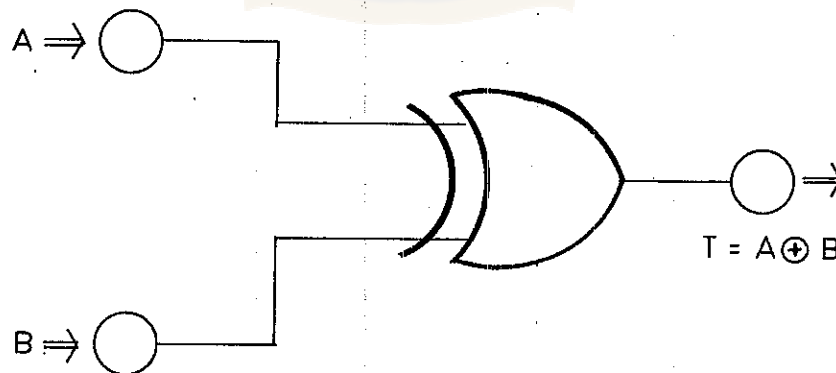
Diagram Venn dari Fungsi EXOR adalah sebagai berikut :

lihat gambar 3.7. Diagram Venn pada halaman berikut.



Gambar 3.7. Diagram Venn.

Rangkaian elektronika yang mewujudkan jalinan ATAU Eksklusip OR kadang-kadang juga disederhanakan dengan sebutan EXOR.

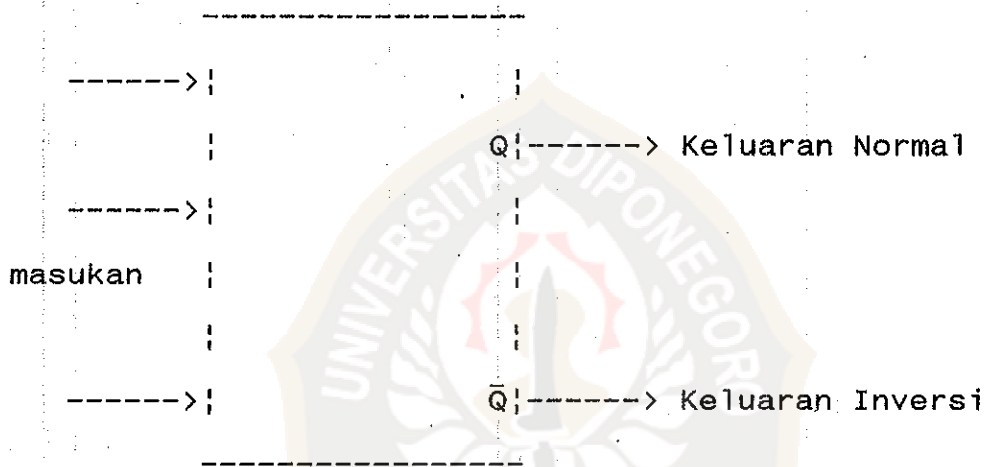


Gambar 3.8. Gerbang EXOR

### 3.2. FLIP FLOP

Flip Flop ( disingkat FF ) adalah sebuah rangkaian logika dengan dua keluaran yang satu kebalikan dari yang lain.

Gambar 3.1. menunjukkan kedua keluaran tersebut sebagai Q dan  $\bar{Q}$  ( sebenarnya dapat digunakan sembarang huruf, tetapi Q adalah yang paling umum ).



Gambar Simbol Flip Flop Umum

Keluaran Q disebut Keluaran FF Normal dan  $\bar{Q}$  adalah Keluaran FF Inversi.

Apabila dikatakan FF berada dalam keadaan tinggi ( 1 ) atau keadaan rendah ( 0 ), yang dimaksudkan adalah keadaan pada keluaran Q.

Dengan sendirinya, keluaran  $\bar{Q}$  selalu merupakan kebalikkan Q.

FF memiliki dua keadaan kerja, yaitu :

1.  $Q = 0, \bar{Q} = 1$
2.  $Q = 1, \bar{Q} = 0$

FF mempunyai satu masukan atau lebih, yang digunakan untuk mengoperasikan FF silih berganti antara dua keadaan tersebut.



Sekali sebuah sinyal masukkan telah mengoperasikan FF menuju satu keadaan tertentu, FF tersebut akan tetap berada pada keadaan itu meskipun setelah masukannya terputus. Keadaan ini merupakan karakteristik memori rangkaian FF.

Flip Flop juga dikenal dengan beberapa nama lain, yaitu *Bistable Multivibrator*, *Latch* dan *Binary*, tetapi umumnya akan disebut Flip Flop, karena istilah ini paling banyak digunakan dalam bidang digital.

Flip Flop adalah memori yang paling baik, karena kecepatan operasinya yang tinggi, informasinya mudah disimpan dan mudah dihubungkan dengan *gate-gate* logika.

Ada 3 ( tiga ) jenis dasar dari Flip Flop, yaitu :

1. S - R Flip - Flop ( Set - Reset )
2. D Flip - Flop
3. J - K Flip - Flop

Perbedaan utama antara berbagai jenis Flip Flop itu adalah :

banyaknya masukkan yang dimiliki dan perilaku bagaimana masukkan itu mempengaruhi keadaan biner dalam Flip Flop tersebut.

### 3.2.1. SET - RESET FLIP FLOP ( S - R FF )

Set - Reset Flip Flop adalah rangkaian Flip Flop yang paling sederhana dibanding Flip Flop lainnya. Nama lain Flip Flop ini adalah "*Bis Table*" atau dapat disebut pula "*One Bit Memory*".

Rangkaian ini dapat dibangun dari dua buah gerbang NAND yang saling dihubungkan

menyilang, yaitu *output* dari *gate* yang pertama masuk sebagai *input gate* yang kedua, dan *output gate* yang kedua masuk sebagai *input gate* yang pertama.

Dua buah jalan masukannya ( *input* ) masing-masing diberi tanda S ( Set ) dan R ( Reset ), sedangkan dua buah *outputnya* diberi tanda "Q" dan " $\bar{Q}$ ".

Set - Reset mempunyai pengertian sebagai berikut :

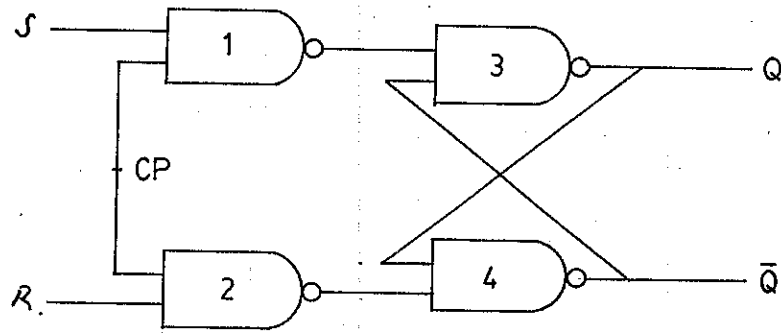
Men - Set ( memasang ) Flip Flop, berarti membuat supaya Q = "1".

Me - Reset ( melepaskan ) Flip Flop, berarti membuat supaya Q = "0".

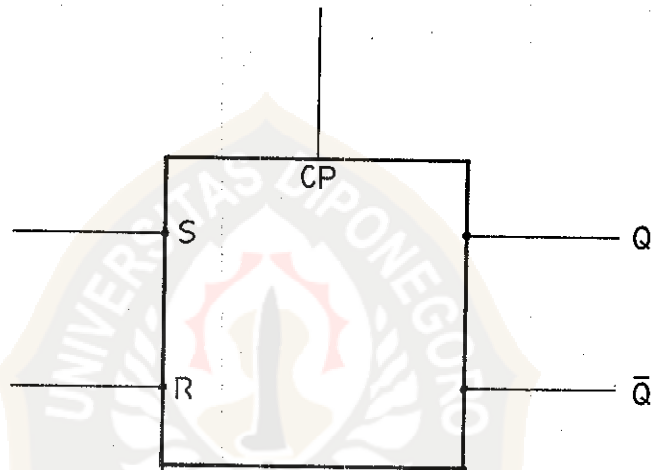
Selanjutnya untuk membuat agar keluarannya berubah secara bersamaan dan pada saat-saat tertentu yang dikehendaki, digunakan sinyal pengontrol yang disebut ( CP ). Dengan adanya sinyal pengontrol tersebut, maka keluaran akan berubah hanya pada saat pulsa detak ( CP = 1 ).

Jika pulsa detak dihentikan ( CP = 0 ), maka keluaran ( *output* ) tidak akan mengalami perubahan.

Gambar Diagram Logika Flip Flop S - R yang dilengkapi dengan CP dapat dilihat pada gambar 3.9. Dimana menggunakan tambahan dua buah gerbang NAND, masukan akan diteruskan bila CP = 1 dan dihentikan bila CP = 0.



Gambar 3.9a. Diagram Logika Flip Flop S - R



Gambar 3.9b. Simbol Flip Flop S - R

Tabel 3.5. Kebenaran : Clock S - R Flip Flop

S	R	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	Diabaikan

Keterangan :

 $Q_n$  = adalah Q sebelum *clock pulse* $Q_{n+1}$  = adalah Q sesudah *clock pulse*

Pada Flip Flop ini bila masukkan ( S ) diberi logik ( asumsi ) 1 ( satu ), maka keadaan berikutnya pasti 1 ( satu ) artinya Flip Flop diset dan bila R diberi 1, maka Flip Flop menjadi 0 ( nol ) artinya direset.

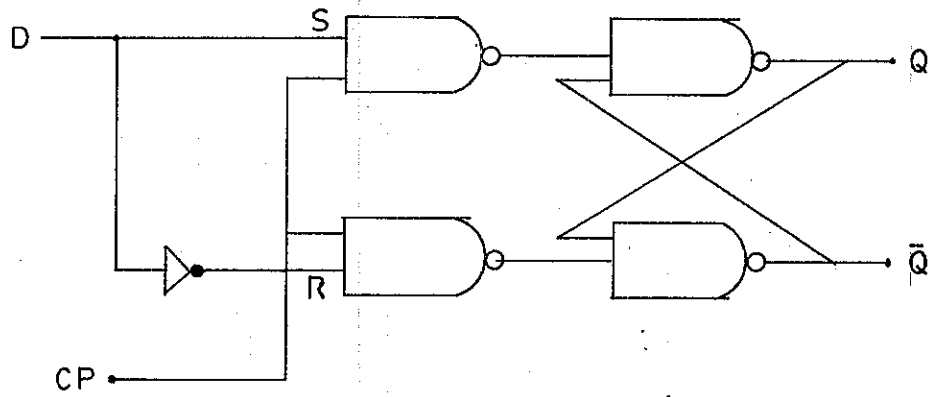
Dan bila S dan R keduanya diberi 1 ( satu ), keadaan berikutnya tidak dapat dipastikan, dapat 0 ( nol ) atau 1 ( satu ).

### 3.2.2 D FLIP FLOP

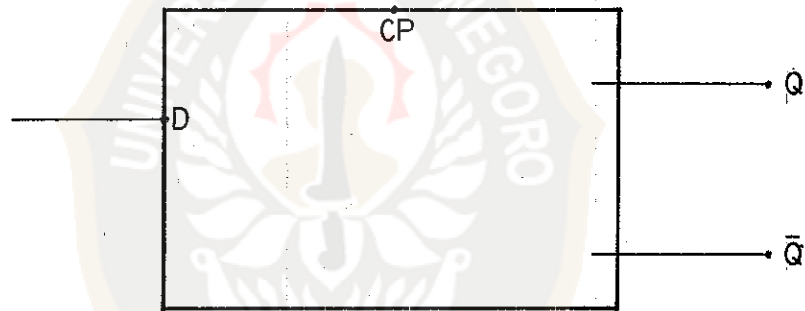
Dari tabel kebenaran-kebenaran clock S - R Flip Flop dapat dilihat, bahwa  $Q_{n+1}$  pada baris no.2 dan baris no.3, kondisinya sama dengan *input* S. Pada saat CP = "1" bila S = 0 maka  $Q_{n+1}$  juga = "1".

Kondisi *output* yang selalu mengikuti *input* tersebut digunakan sebagai dasar pembentukan D Flip Flop ( Delay Flip Flop ) atau sering disebut juga "D - Latch".

Diagram logika "D - Latch" sama dengan clock S - R Flip Flop, hanya *input* S diganti dengan D *input* R dengan fungsi NOT dari D ( menggunakan satu *inverter* ) seperti terlihat pada gambar 3.10a. berikut :



Gambar 3.10a. Diagram Logika D Flip Flop



Gambar 3.10b. Simbol D Flip Flop

Tabel 3.6. Kebenaran : D Flip Flop

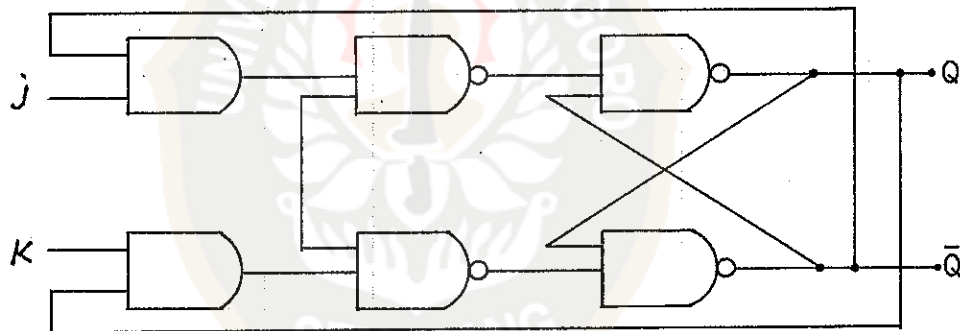
D	Q <sub>n+1</sub>
0	0
1	1

Dari tabel kebenaran terlihat bahwa *output* D Flip Flop adalah keadaan masukan ( *input* ) yang sekarang.

### 3.2.3. J - K FLIP FLOP

J - K Flip Flop merupakan peningkatan dari Flip Flop S - R, dimana dua buah S - R Flip Flop disambungkan menjadi satu, yaitu : kedua *output* Flip Flop yang pertama masuk sebagai *input* Flip Flop yang kedua, dan kedua *output* Flip Flop yang kedua masuk sebagai *input* Flip Flop yang pertama.

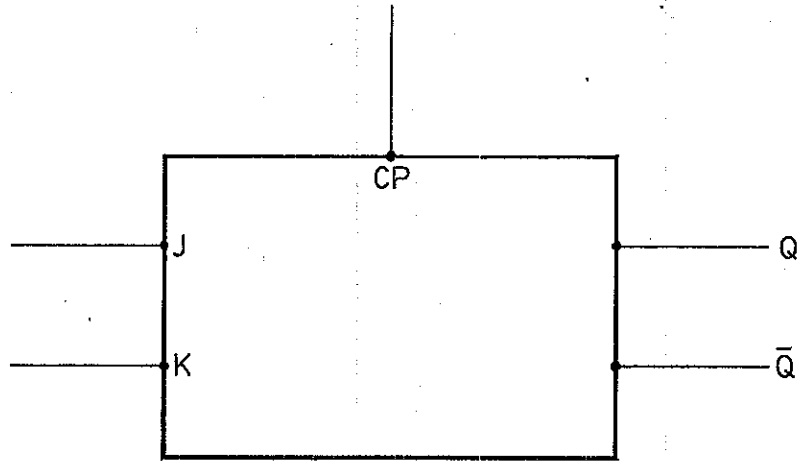
Gambar untuk skema logika J - K Flip Flop, simbol dan tabel kebenaran adalah sebagai berikut :



Gambar 3.11a. Skema Logika J - K Flip Flop

Tabel 3.7. Kebenaran : J - K Flip Flop

J	K	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\bar{Q}_n$



Gambar 3.11b. Simbol J - K Flip Flop.



### 3.3. REGISTER

Register adalah suatu rangkaian logika yang berfungsi untuk menyimpan data atau informasi.

Dengan telah mempelajari bermacam-macam Flip Flop, dapatlah dimengerti bahwa yang disebut Register itu tidak lain adalah alat untuk menyimpan data yang berupa satu atau beberapa Flip Flop yang digabungkan menjadi satu.

Hal ini dimaksudkan, bahwa Register yang paling sederhana hanya terdiri satu Flip Flop saja, yang berarti hanya dapat menyimpan data yang terdiri dari satu bilangan biner saja yaitu "0" atau "1".

Oleh karena itu untuk menyimpan data yang terdiri dari empat bit bilangan biner misalnya, diperlukan Flip Flop sebanyak empat buah.

Untuk mengisi Register atau untuk menyimpan data dapat dilakukan dengan 2 ( dua ) cara :

a. Dimasukkan Secara Jajar ( *Parallel In* ) :

Pada cara ini semua bagian register atau masing-masing Flip Flop diisi ( dimuati ) pada saat yang bersamaan. Atau output masing-masing Flip Flop akan respon sesuai data pada saat yang sama setelah diberikan Sinyal *Input Kontrol*.

Data	:	1	1	0	0	Output masing-masing
						Flip Flop akan respon
		↓	↓	↓	↓	Sesuai data secara se-
		-----				rempak setelah Input

Output : | 1 | 1 | 0 | 0 | Kontrol diberi nilai

----- Asumsi ( logika ) "1"



b. Dimasukkan Secara Deret ( Serial n ).

Pada cara ini data dimasukkan bit demi bit mulai dari Flip Flop yang paling ujung ( dapat dari kiri atau dari kanan ) dan digeser sampai semuanya terisi.

Penggeseran data di atas oleh Sinyal *Clock* tiap kali data dimasukkan satu persatu.

Bila data digeser dari kanan ke kiri disebut "Register Bergeser Kekiri" ( *Shift Left Register* ) sebaliknya bila data digeser dari kiri ke kanan disebut "Register Bergeser Ke kanan" ( *Shift Right Register* ).

Register Bergeser Ke kiri ( *Shift Left Register* )

-----	Data :	Keterangan :
1.   0   0   0   0	1101	1. Register dalam keadaan kosong data siap masuk menunggu clock.
-----		
2.   0   0   0   1	1010	2. Sinyal clock pertama, satu bit data mulai masuk.
-----		
3.   0   0   1   1	0100	3. Sinyal clock ke dua, data yang pertama bergeser, data ke-
-----		

dua mulai masuk

4. | 0 | 1 | 1 | 0 | 1000

4-5. Demikian seterusnya sampai semua Flip Flop terisi.

5. | 1 | 1 | 0 | 1 | 0000

