

QUEUE-ANTRIAN

STRUKTUR DATA

ANTRIAN(QUEUE)

- ◉ Suatu bentuk khusus dari linear list, dengan operasi penyisipan (insertion) hanya diperbolehkan pada salah satu sisi, yang disebut **REAR**, dan operasi penghapusan (deletion) hanya diperbolehkan pada sisi yang lainnya, yang disebut **FRONT** dari list.
- ◉ Antrean $Q = [Q_1, Q_2, \dots, Q_N]$
- ◉ $\text{Front}(Q) = Q_1$ bagian depan antrean
- ◉ $\text{Rear}(Q) = Q_N$ bagian belakang antrean
- ◉ $\text{Noel}(Q) = N$ jumlah elemen dalam antrean

- ◉ Operasi Antrean : ***FIFO (First In First Out)***
Elemen yang pertama masuk merupakan elemen yang pertama keluar.

- ◉ Operator :
 - Penyisipan : Insert
 - Penghapusan : Remove

- ◉ Empat operasi dasar antrean, yaitu :
 1. CREATE
 2. ISEMPY
 3. INSERT
 4. REMOVE

◉ CREATE (Q)

Operator yang menunjukkan suatu antrean hampa Q.

Berarti :

Noel (Q) = 0

Front (Q) & Rear (Q) = tidak terdefinisi

◉ ISEMPY (Q)

Operator yang menunjukkan apakah antrean Q hampa.

Operand : tipe data antrean

Hasil : tipe data boolean

ISEMPY (CREATE (Q)) = True

- ◉ **INSERT (E, Q)**

Operator yang memasukkan elemen E ke dalam antrian Q.
E ditempatkan di bagian belakang antrian.

Hasil : antrian yang lebih besar.

$\text{REAR}(\text{INSERT}(E, Q)) = E$

$\text{ISEMPTY}(\text{INSERT}(E, Q)) = \text{False}$

- ◉ **REMOVE (Q)**

Operator yang menghapus elemen bagian depan dari antrian Q.

Hasil : antrian yang lebih pendek.

Pada setiap operasi, $\text{Noel}(Q)$ berkurang 1 dan elemen ke-2 menjadi elemen terdepan.

Jika $\text{Noel}(Q) = 0$ maka $Q = \text{hampa}$

Remove (Q) = kondisi error (underflow condition)

Remove (Create (Q)) = kondisi error (underflow condition)

◉ PENYAJIAN DARI ANTREAN

1. One Way List (Linear Linked List)
2. Array

◉ Array Queue

Kalau tidak disebutkan lain, maka Antrean disajikan dalam Array Queue, dilengkapi 2 variabel penunjuk :

FRONT (elemen depan antrean)

REAR (elemen belakang antrean)

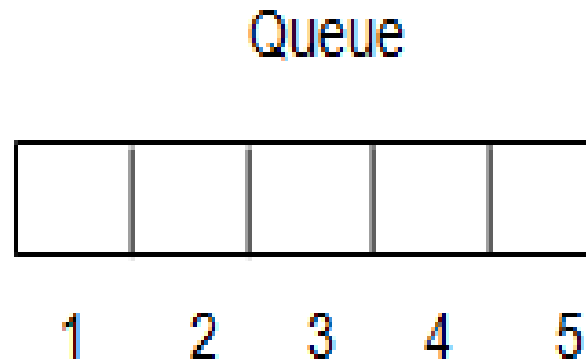
- ◉ **Contoh :**

- ◉ Antrean dalam array queue dengan 5 lokasi memori

1. Pada awal antrean hampa

$F = 0$

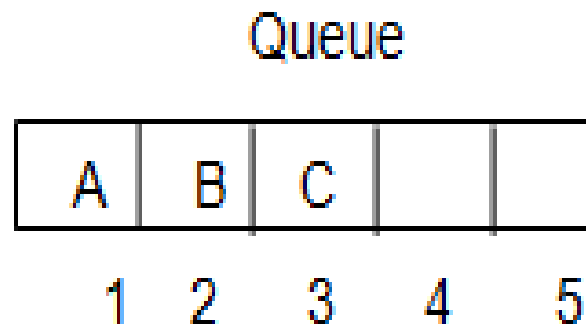
$R = 0$



2. A, B dan C dimasukkan

$F = 1$

$R = 3$



3. Hapus 1 elemen : A dihapus

$F = 2$

$R = 3$

Queue

	B	C		
1	2	3	4	5

4. D dan E dimasukkan

$F = 2$

$R = 5$

Queue

	B	C	D	E
1	2	3	4	5

5. Hapus 1 elemen : B dihapus

$F = 3$

$R = 5$

Queue

		C	D	E
1	2	3	4	5

- ◉ Untuk setiap pemasukan elemen, nilai $\text{Rear} + 1$
Penghapusan elemen, nilai $\text{Front} + 1$
- ◉ Akibatnya, setelah pemasukan elemen ke-5 maka lokasi Queue (5) telah diduduki mungkin saja tidak sebanyak 5 elemen ada dalam antrean, karena sudah dilakukan beberapa penghapusan. Untuk pemasukan elemen berikutnya, yakni memasukkan elemen ITEM, gunakan lokasi QUEUE (1), dan seterusnya. Array Sirkular yaitu elemen Queue (1) datang sesudah Queue (N) di dalam array, maka $\text{Rear} = 1$. Jika $\text{Front} = N$, dilakukan penghapusan maka $\text{Front} = 1$, bukan $N + 1$.

◉ Contoh :

◉ Array Sirkular dengan 5 lokasi memori

1. Pada awal antrean hampa

$F = 0$

$R = 0$

Queue

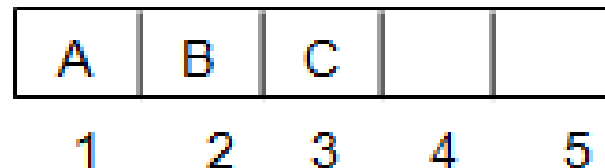


2. A, B dan C dimasukkan

$F = 1$

$R = 3$

Queue



3. Hapus 1 elemen : A dihapus

$F = 2$

$R = 3$

Queue



4. D dan E dimasukkan

$F = 2$

$R = 5$

Queue

	B	C	D	E
1	2	3	4	5

5. Hapus 1 elemen : B dihapus

$F = 3$

$R = 5$

Queue

		C	D	E
1	2	3	4	5

6. G dimasukkan

$F = 3$

$R = 1$

Queue

G		C	D	E
1	2	3	4	5

◉ QINSERT (QUEUE, N, FRONT, DATA)

- ◉ 1. {Apakah antrean penuh}
Jika $\text{Front} = 1$ dan $\text{Rear} = N$, atau Jika $\text{Front} = \text{Rear} + 1$, maka write overflow, return
- ◉ 2. Jika $\text{Front} = \text{Null}$, maka $\text{Front} := 1$
 $\text{Rear} := 1$
 Dalam hal lain
 Jika $\text{Rear} = N$, maka $\text{Rear} := 1$
 Dalam hal lain
 $\text{Rear} := \text{Rear} + 1$
- ◉ 3. $\text{Queue}(\text{Rear}) := \text{Data}$ (masukkan elemen baru)
- ◉ 4. Return.

- ◉ QDELETE (QUEUE, N, FRONT, REAR, DATA)
- ◉ 1. {Apakah antrean kosong}
Jika Front := Null, maka write underflow,
return
- ◉ 2. Data := Queue (Front)
- ◉ 3. (Front mendapat nilai baru)
Jika Front := Rear, maka
Front := Null,
Dalam hal lain
Jika Front = N, maka Front := 1
Dalam hal lain
Front := Front + 1
- ◉ 4. Return.

- **Latihan :**

- **1. Apa output dari program dibawah ini :**

- Program antrian_1;
- uses crt;
- type
- data= array [1..3] of string;
- var
- d: data;
- i, antri: integer;
- temp: string;
- begin
- antri:=0;
- *{untuk input}*
- For i:=1 to 3 do
- begin
- Writeln('masukkan nama ke',' ',i);
- Readln(d[i]);
- antri:=antri+1;
- end;
- *{untuk Output}*
- for i=1 to 3 do
- begin
- temp:=d[i];
- antri:=antri-1;
- end;
- *{lihat output di var temp setelah pengambilan }*
- writeln('hasil var temp=',temp);
- readln;
- end.

- 2. Diketahui Array Sirkular dengan 6 lokasi memori, dimana awal antrean hampa.
Buatlah ilustrasi untuk :
 - (a) Insert F, G, H, I
 - (b) Insert J
 - (c) Remove
 - (d) Insert K
 - (e) Insert L