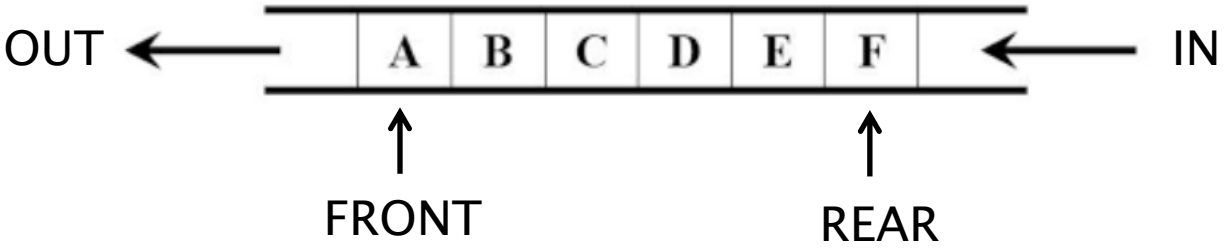
ANTRIAN (QUEUE)

Dosen : Sulistyowati, ST., M.Kom.

- Queue/antrian adalah barisan elemen yang apabila elemennya ditambah maka penambahannya berada di **belakang (*Rear*)** dan apabila dilakukan pengambilan maka elemen yang diambil adalah elemen yang posisinya paling **depan (*Front*).**

- Konsep ini dikenal dengan F**IFO (First In First Out)**

- Ilustrasi queue :



- Ada 2 operasi dasar yang bisa dilakukan pada queue yaitu :
  - **Enqueue** (menambah/menyisipkan data)
  - **Dequeue** (menghapus/mengurangi data)

- **Prinsip kerja Queue** :
  - Semula (pada saat queue kosong), Front dan Rear selalu berada di index 0.

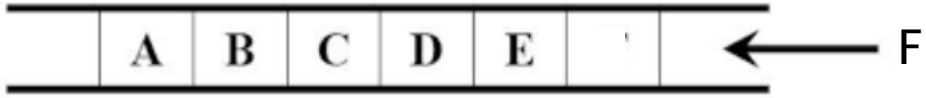
  - Pada saat diisi satu data, maka posisi Front dan Rear bergerak ke index 1. Jika diisi data lagi dan seterusnya, maka posisi Rear akan bergerak naik ke index berikutnya sedangkan posisi Front akan tetap di index 1.

  - Pada saat pengambilan/penghapusan sebuah data, maka posisi Rear akan bergerak turun ke index sebelumnya dan begitu seterusnya.

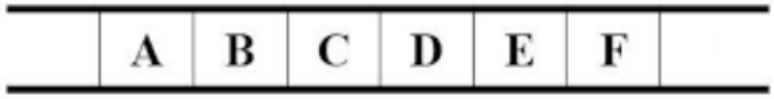
ENQUEUE

- Operasi Enqueue bisa dilakukan jika QUEUE BELUM PENUH

Queue semula :



Queue akhir :



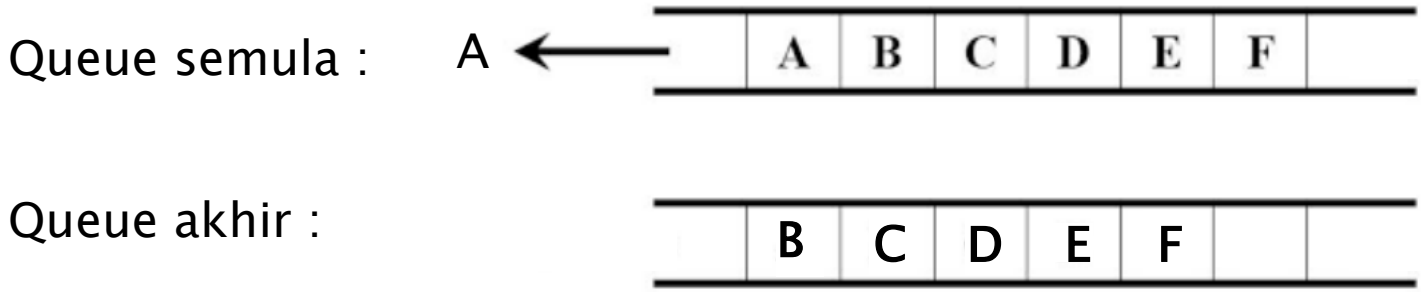
Pada gambar di atas mula-mula queue berisi data A, B, C, D dan E. Kemudian dilakukan operasi enqueue data F, sehingga antrian data pada queue menjadi A, B, C, D, E, F

- Algoritma ENQUEUE :

```
if (rear<maximum)
{
        rear++;
        cin>>queue[rear];
        if (rear == 1)
                front++;
}
else
        cout<<“Queue sudah penuh”;
```

DEQUEUE

- Ilustrasi operasi Dequeue pada queue adalah sebagai berikut :



Pada gambar di atas mula-mula queue berisi data A, B, C, D, E dan F. Kemudian dilakukan operasi dequeue yaitu data yang paling depan (A) dihapus, sehingga data pada antrian menjadi B, C, D, E, F

- Algoritma DEQUEUE :

```
if (rear>0)
{
        hapus=queue[front];
        for (i=1;i<=rear-1;i++)
                queue[i]=queue[i+1];
        rear--;
        if (rear=0)
                front--;
}
else
        cout<<“Queue telah kosong";
```