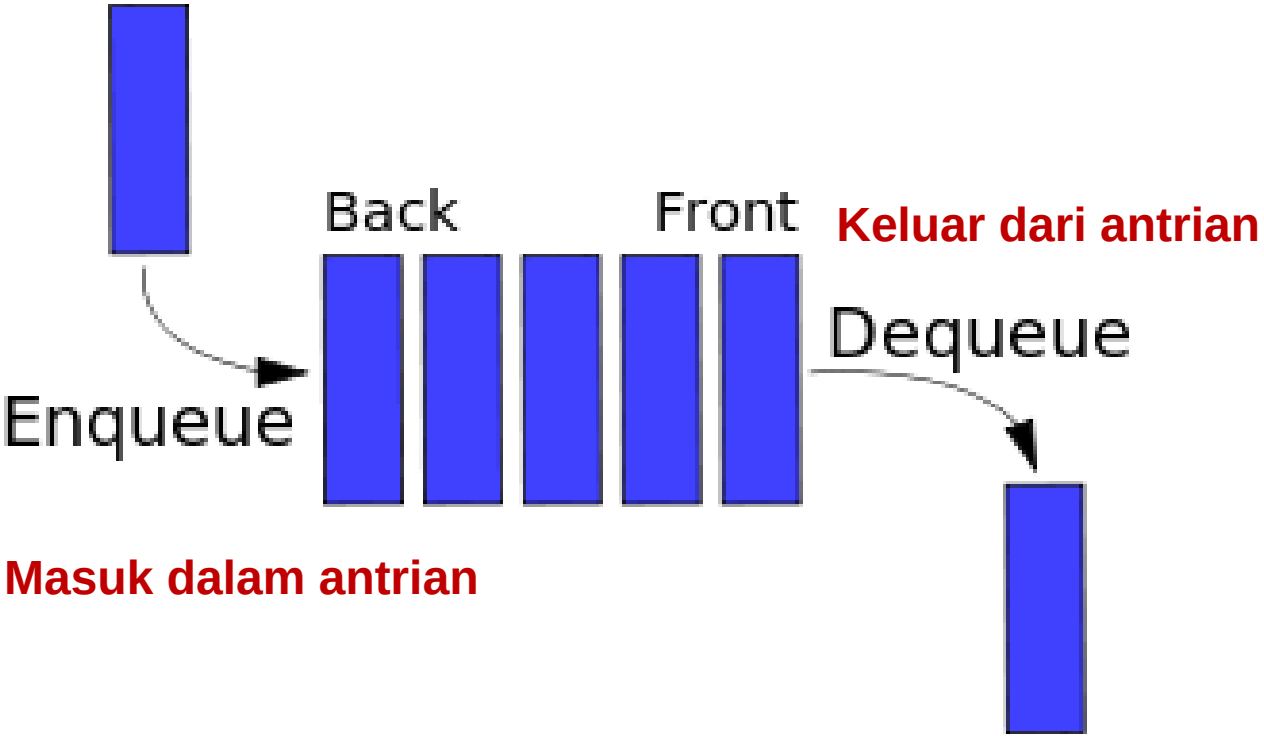
queue (antrian)

Tenia Wahyuningrum, S.Kom., M.T

Queue bersifat FIFO (First In First Out)

“elemen pertama yang ditempatkan pada queue adalah yang pertama dipindahkan”



Representasi Antrian

Operasi-operasi antrian

- **CREATE**

  Untuk menciptakan dan menginisialisasi queue dengan cara membuat Head dan Tail = -1

- **ISEMPTY**

  Untuk memeriksa apakah queue kosong

- **ISFULL**

  Untuk memeriksa apakah queue sudah penuh

Operasi-operasi antrian

- **ENQUEUE**

Untuk menambahkan item pada posisi paling belakang

- **DEQUEUE**

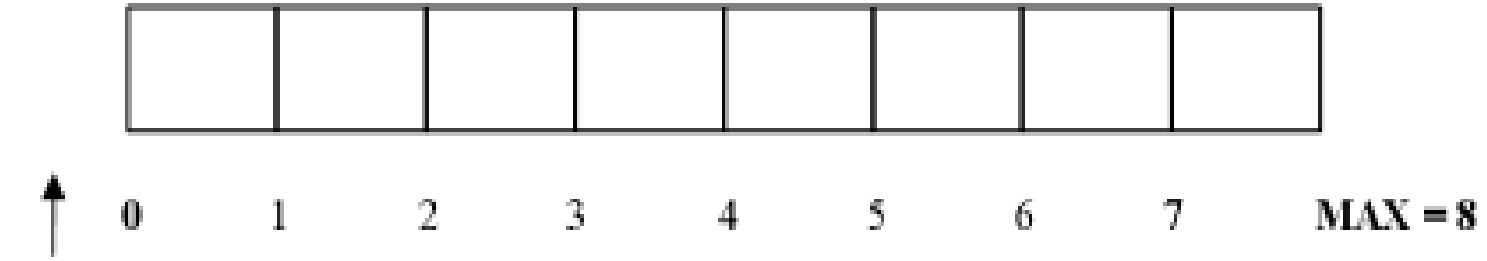
Untuk menghapus item dari posisi paling depan

- **CLEAR**

Untuk mengosongkan queue

Queue Linier Array

- Terdapat satu buah pintu masuk di suatu ujung dan satu buah pintu keluar di ujung satunya
- Sehingga membutuhkan 2 variabel: **Head dan Tail**



Head = -1

Tail = -1

Queue (2)

DEKLARASI QUEUE

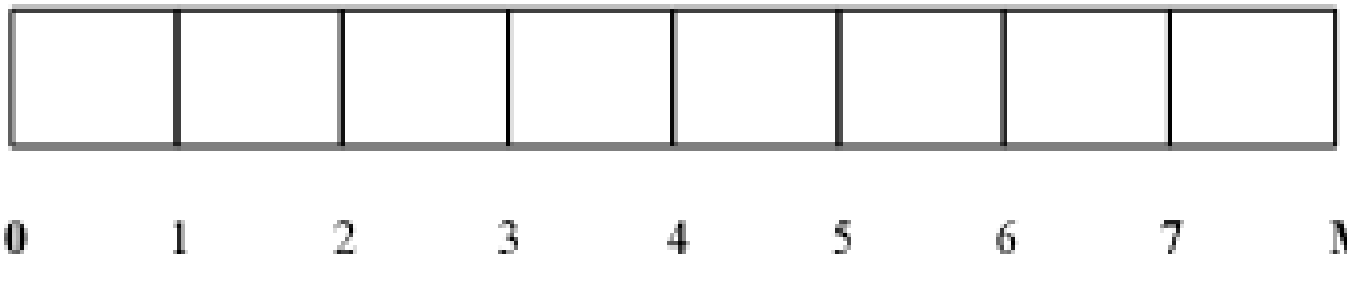
```
#define MAX 8
typedef   struct{
                    int data[MAX];
                    int head;
                    int tail;
                } Queue;

Queue antrian;
```

- Operasi-operasi:

Create()

  - Untuk menciptakan dan menginisialisasi Queue
  - Dengan cara membuat **Head dan Tail = -1**

Queue (3)



Head = -1

Antrian pertama kali

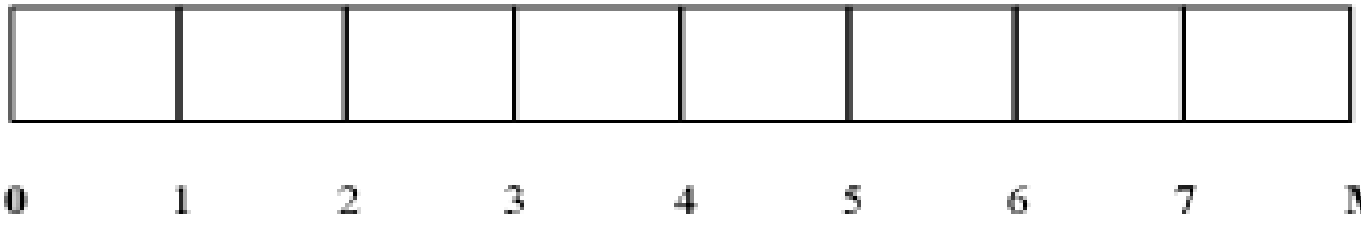
Tail = -1

```
void Create(){
     antrian.head=antrian.tail=-1;
}
```

Queue (4)

- **IsEmpty()**
  - Untuk memeriksa apakah **Antrian sudah penuh atau belum**
  - Dengan cara memeriksa nilai Tail, jika Tail = -1 maka **empty**
  - Kita tidak memeriksa Head, karena **Head adalah tanda untuk kepala antrian** (elemen pertama dalam antrian) yang tidak akan berubah-ubah
  - Pergerakan pada Antrian terjadi dengan penambahan elemen Antrian kebelakang, yaitu menggunakan **nilai Tail**

Queue (5)



Head = -1

Tail = -1

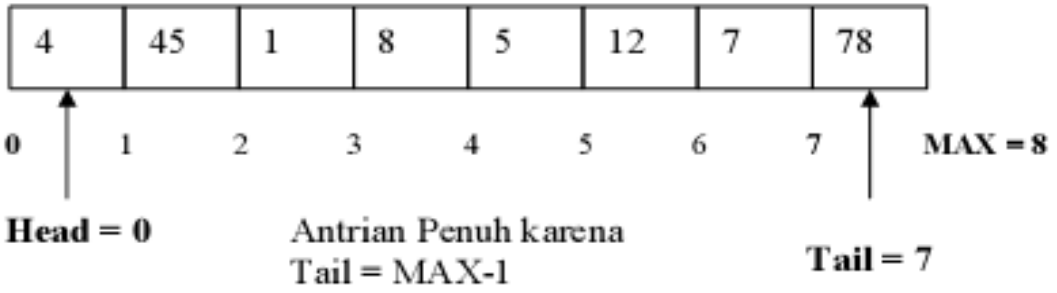
Antrian Kosong
karena Tail = -1

```
int IsEmpty(){
      if(antrian.tail==-1)
            return 1;
      else
            return 0;
}
```

Queue (6)

Fungsi IsFull

- Untuk mengecek apakah Antrian sudah **penuh atau belum**
- Dengan cara mengecek nilai Tail, **jika Tail >= MAX-1** (karena MAX-1 adalah batas elemen array pada C) berarti **sudah penuh**




```
int IsFull(){
     if(antrian.tail==MAX-1) return 1;
     else return 0;
}
```

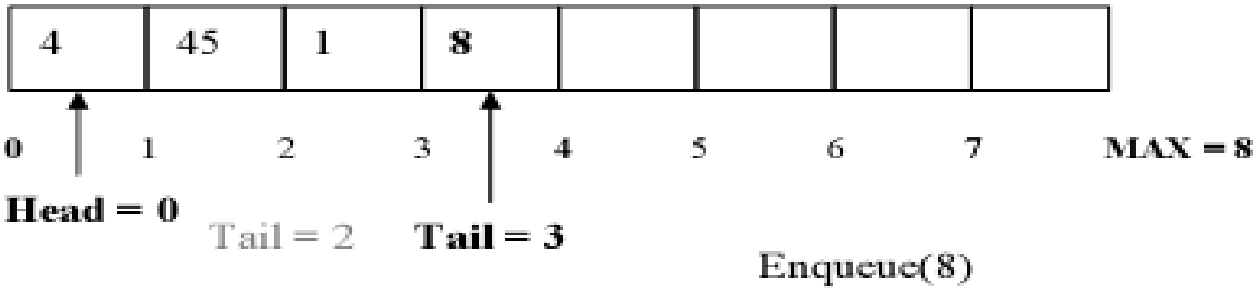
Queue (7)

Enqueue

– Untuk menambahkan elemen ke dalam Antrian, penambahan elemen selalu ditambahkan di elemen paling **belakang**

– Penambahan elemen selalu menggerakan variabel Tail dengan cara **increment counter Tail** terlebih dahulu

Queue (8)

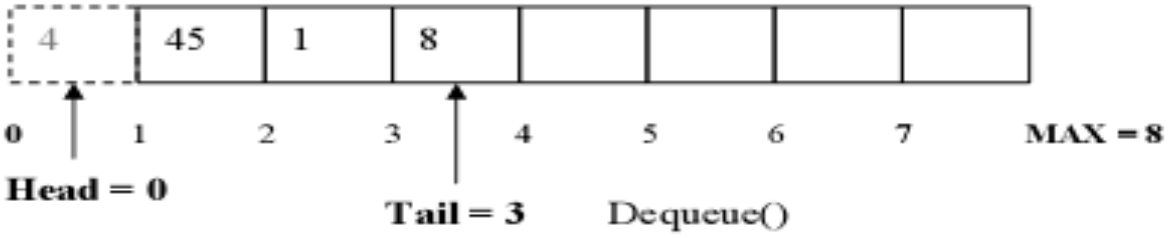


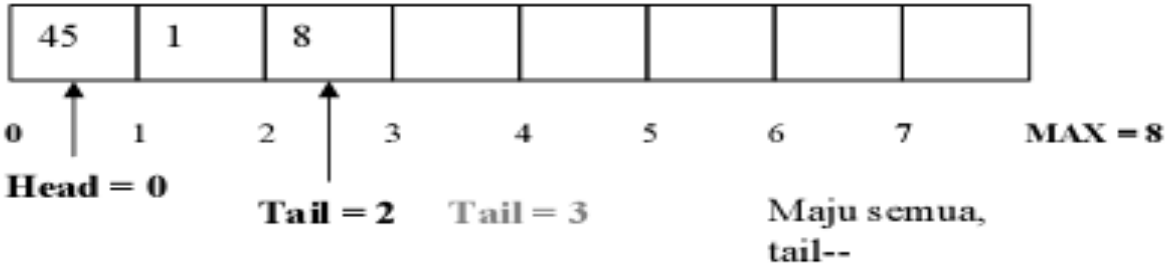
```
void Enqueue(int data){
      if(IsEmpty()==1){
            antrian.head=antrian.tail=0;
            antrian.data[antrian.tail]=data;
            printf("%d masuk!",antrian.data[antrian.tail]);
      } else
      if(IsFull()==0){
            antrian.tail++;
            antrian.data[antrian.tail]=data;
            printf("%d masuk!",antrian.data[antrian.tail]);
      }
}
```

Queue (9)

- **Dequeue()**
  - Digunakan untuk **menghapus** elemen terdepan/pertama (head) dari Antrian
  - Dengan cara **menggeser** semua elemen antrian kedepan dan **mengurangi Tail dgn 1**
  - Penggeseran dilakukan dengan **menggunakan looping**

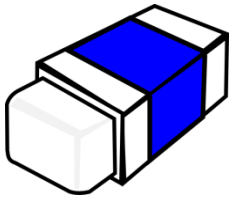
Queue (10)





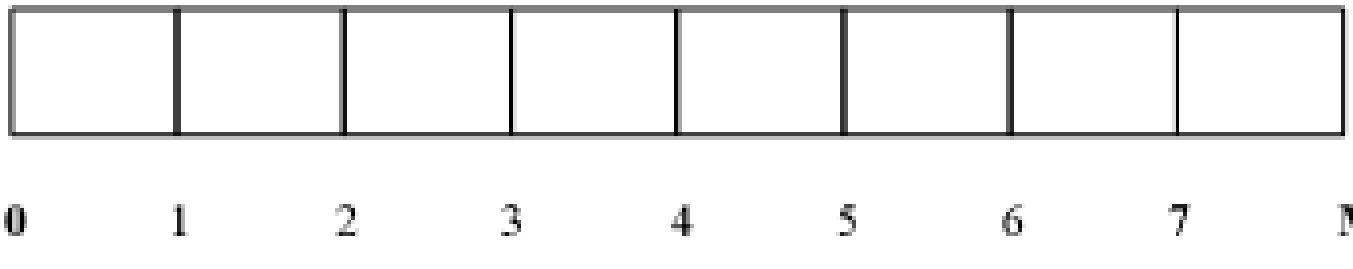
```
int Dequeue(){
      int i;
      int e = antrian.data[antrian.head];
      for(i=antrian.head;i<=antrian.tail-1;i++){
            antrian.data[i] = antrian.data[i+1];
      }
      antrian.tail--;
      return e;
}
```

Queue (11)



- **Clear()**
  - Untuk **menghapus elemen-elemen Antrian** dengan cara membuat Tail dan Head = -1
  - Penghapusan elemen-elemen Antrian sebenarnya tidak menghapus array-nya, namun hanya **mengeset indeks pengaksesan-nya ke nilai -1** sehingga elemen-elemen Antrian tidak lagi terbaca

Queue (12)



Head = -1

Tail = -1

```
void Clear(){
     antrian.head=antrian.tail=-1;
     printf("data clear");
}
```

Queue (13)

- **Tampil()**
  - Untuk **menampilkan** nilai-nilai elemen Antrian
  - Menggunakan **looping dari head s/d tail**

```
void Tampil(){
     if(IsEmpty()==0){
          for(int i=antrian.head;i<=antrian.tail;i++){
               printf("%d ",antrian.data[i]);
          }
     }else printf("data kosong!\n");
}
```

Ada pertanyaan?

Telkom Foundation

Telkom Indonesia
the world in your hand

Jadi, apa perbedaan stack dan queue?

Kenapa head dan tail pertama kali harus =-1?

Terima kasih, sampai jumpa !