Pertemuan 6

QUEUE (ANTREAN)

PENGERTIAN QUEUE (ANTREAN)

Struktur Data Antrean (Queue) adalah suatu bentuk khusus dari List Linier dengan operasi pemasukan data hanya diperbolehkan pada salah satu sisi, yang disebut sisi Belakang / ekor (Tail) dan operasi penghapusan hanya diperbolehkan pada sisi lainnya yang disebut sisi Depan / kepala (Head) dari LinkedList.

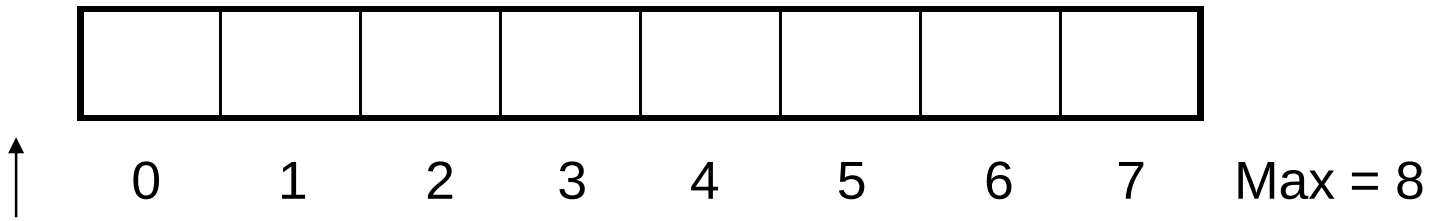
Prinsip Antrean : FIFO (First In First Out)
FCFS (First Come First Serve)

"Yang Tiba lebih awal Maka akan dilayani Terlebih Dahulu"

Deklarasi Queue

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

Queue antrian;
```



head = -1

tail = -1

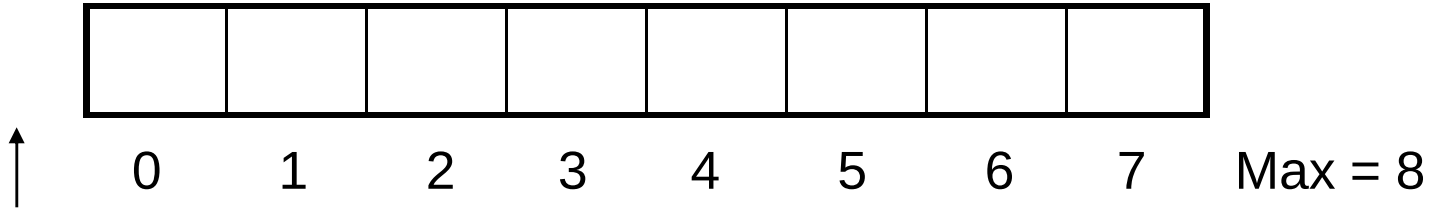
OPERASI QUEUE

- **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
- **ENQUEUE**
  Untuk menambahkan item pada posisi paling belakang
- **DEQUEUE**
  Untuk menghapus item dari posisi paling depan
- **CLEAR**
  Untuk mengosongkan queue

Fungsi Create

- Digunakan untuk membentuk dan menunjukan awal terbentuknya suatu Antrean / Queue

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



head = -1

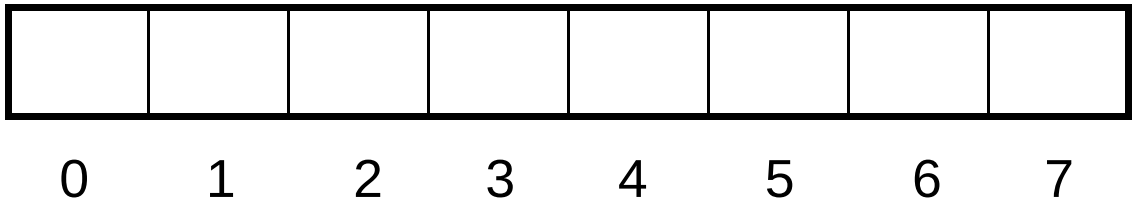
tail = -1

Antrian pertama kali

Fungsi IsEmpty

- Untuk memeriksa apakah Antrian penuh atau kosong
- Dengan cara memeriksa nilai Tail, jika Tail = -1 maka antrian kosong (empty)
- 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

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



head = -1

tail = -1

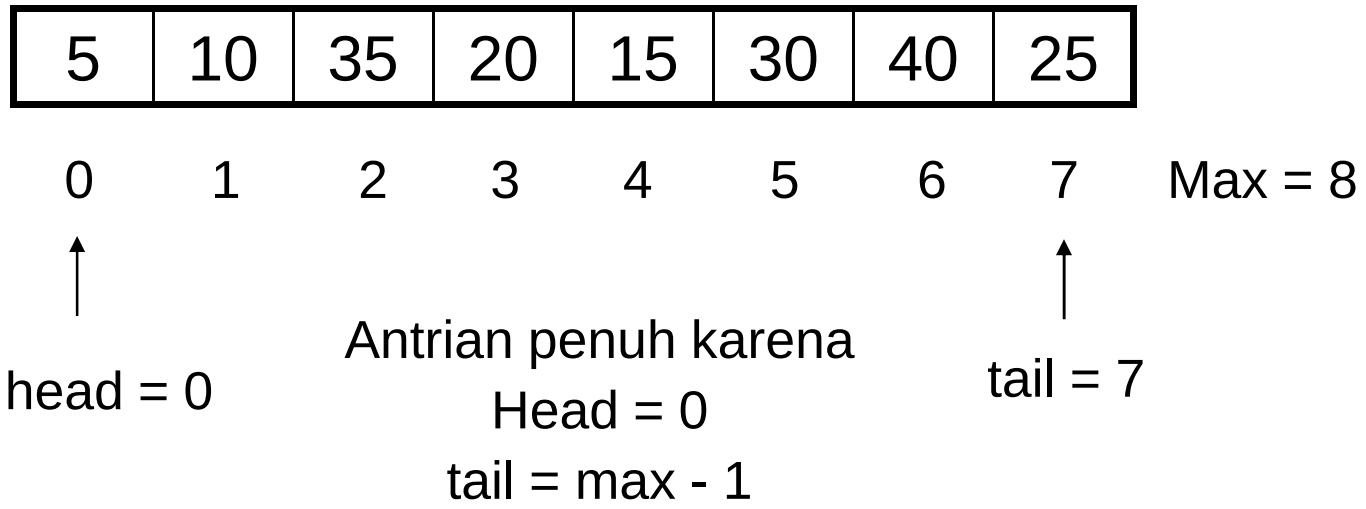
Antrian kosong

Karena tail = -1

Fungsi IsFull

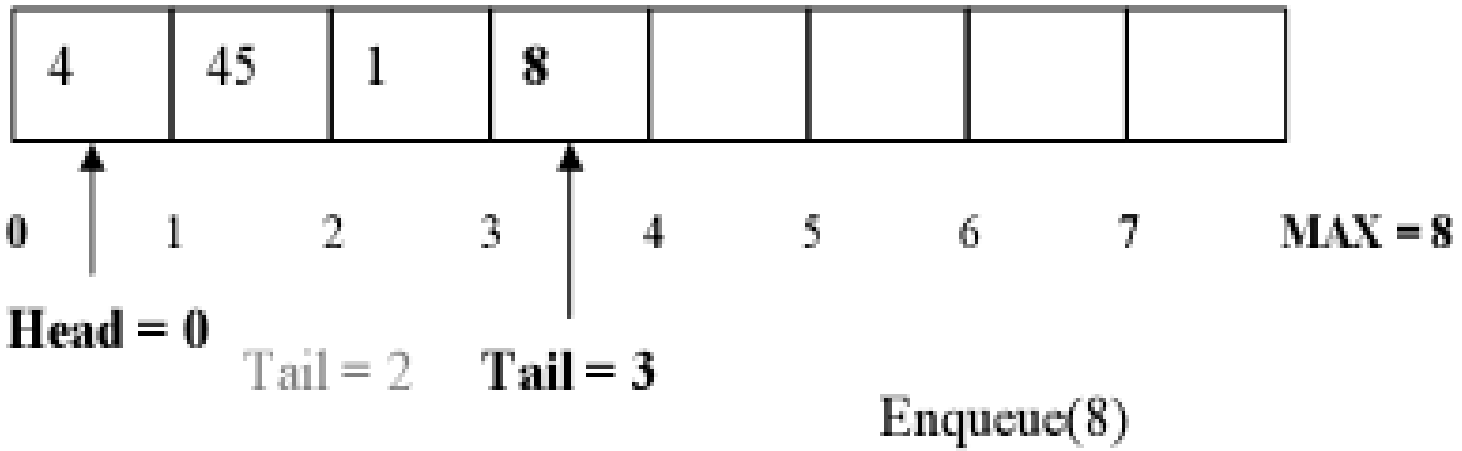
- Untuk mengecek apakah Antrian sudah penuh atau belum
- Dengan cara :
  - Mengecek nilai Tail
  - Jika tail = MAX-1 berarti antrian sudah penuh (MAX-1 adalah batas elemen array dalam program C++)

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



Fungsi Enqueue

- Untuk menambahkan elemen ke dalam Antrian, penambahan elemen selalu dilakukan pada elemen paling **belakang**
- Penambahan elemen selalu menggerakan variabel Tail dengan cara menambahkan Tail terlebih dahulu

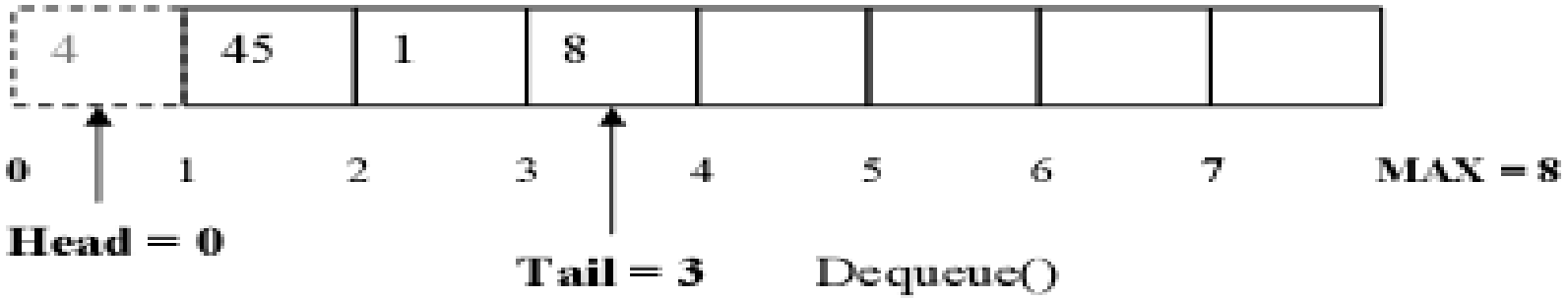


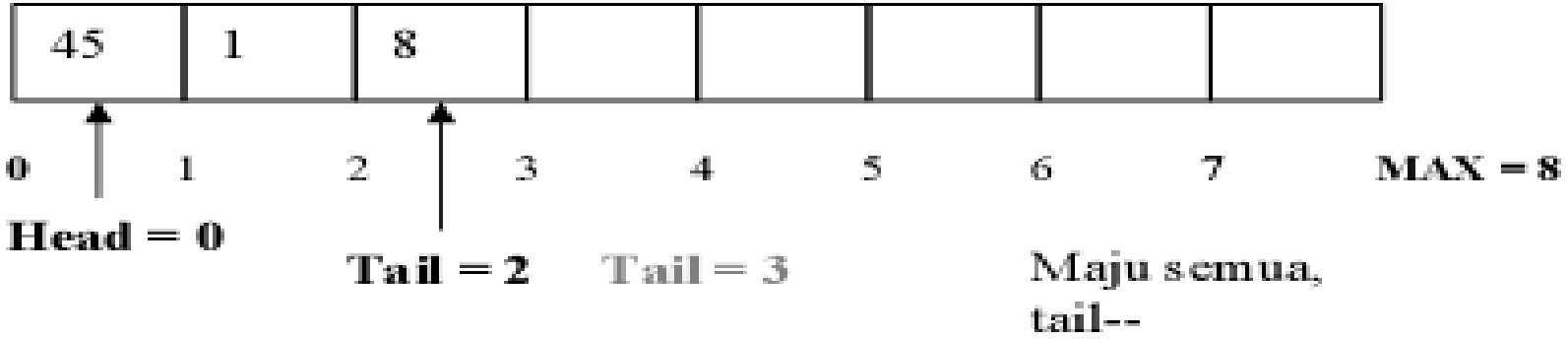
```
void Enqueue(int data){
     if(IsEmpty()==1){
          antrian.head=antrian.tail=0;
          antrian.data[antrian.tail]=data;
          printf("%d masuk!",antrian.data[antrian.tail]);
     } else
     if(IsFull()==0){
          antrian.tail++;
          antrian.data[antrian.tail]=data;
      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

Fungsi Dequeue

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

```
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;
}
```


4
45
1
8
0
1
2
3
4
5
6
7
MAX = 8
Head = 0
Tail = 3
Dequeue()


45
1
8
0
1
2
3
4
5
6
7
MAX = 8
Head = 0
Tail = 2
Tail = 3
Maju semua,
tail--

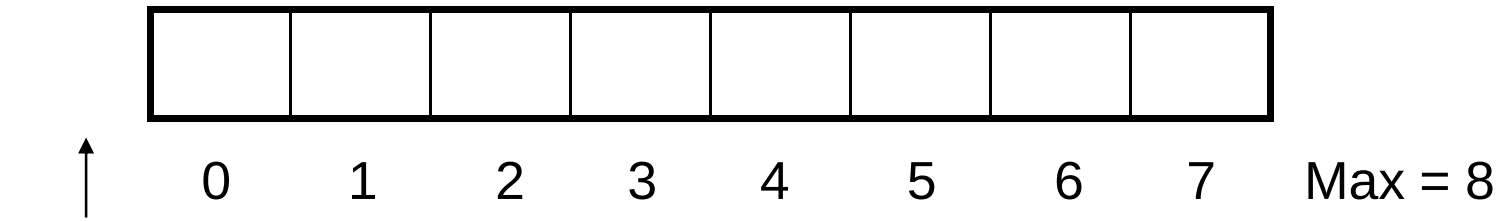
Fungsi Clear

- Untuk menghapus elemen-elemen Antrian dengan cara membuat Tail dan Head = -1

- Penghapusan elemen-elemen Antrian sebenarnya tidak menghapus arraynya, namun hanya mengeset indeks pengaksesan-nya ke nilai -1 sehingga elemen-elemen Antrian tidak lagi terbaca sehingga mengembalikan antrian seperti keadaan semula

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

Antrian setelah di lakukan Clear



head = -1

tail = -1

Antrian kosong

Karena tail = -1