

**Pertemuan 09**

# **QUEUE**

# Learning Outcomes

Pada akhir pertemuan ini, diharapkan mahasiswa akan mampu :

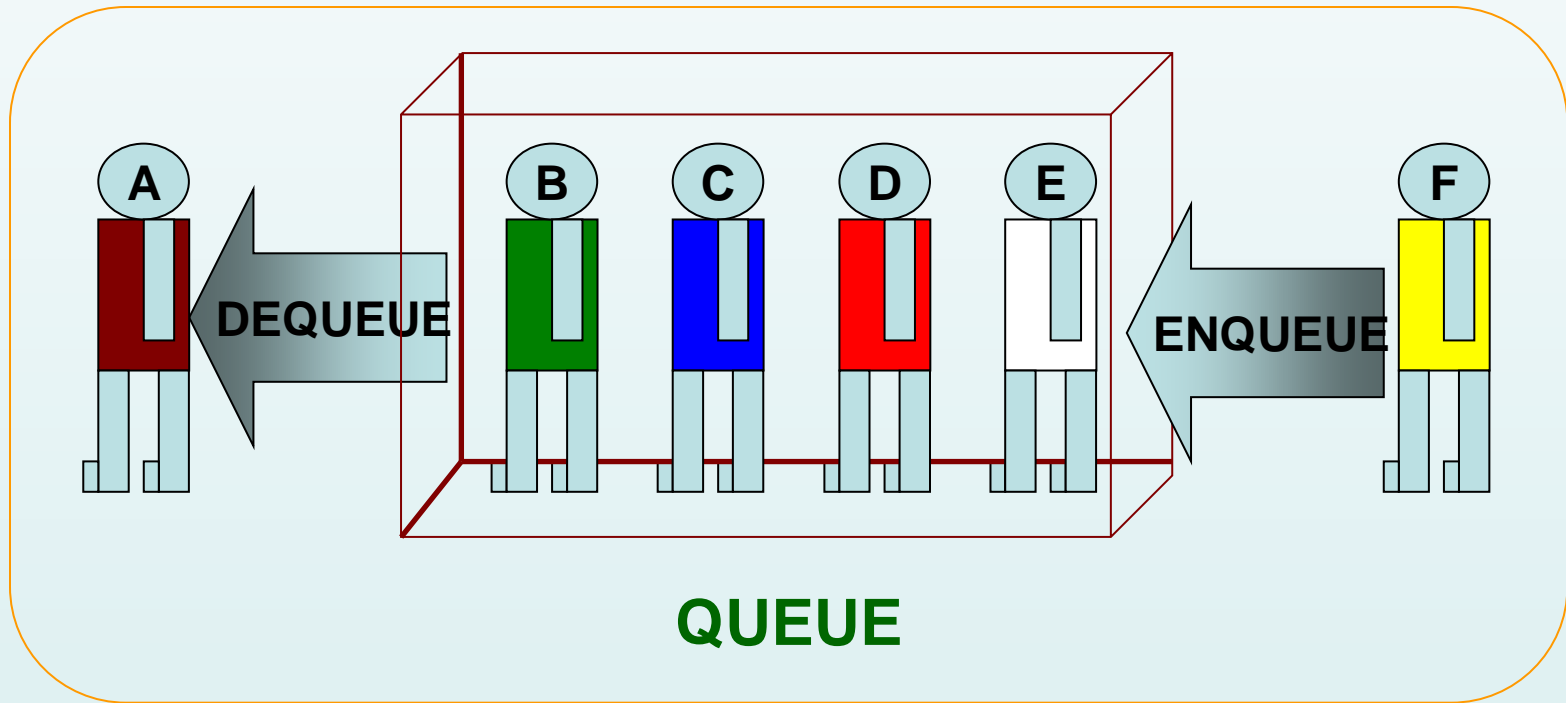
- mendemonstrasikan TDA Queue.
- menerapkan TDA Queue pada program aplikasi komputer.

# Materi

- Basis Operasi
- Abstraksi
  - Obyek
  - Operasi/fungsi
- Implementasi
  - Array
    - Linier
    - Circular
  - Linked List

# Basis Operasi

Basis operasi FIFO (First In First Out), Elemen yang diambil / dihapus adalah elemen yang pertama dimasukkan / diinsert



# TDA Queue

## ADT Queue

**Structure** *Queue* is

**Objects** : a finite ordered list with zero or more elements.

**function** :

For all *queue*  $\in$  *Queue*, *item*  $\in$  *element*, *max\_queue\_size*  $\in$  *positive integer*.

Queue Create(*max\_queue\_size*) ::= create an empty *queue* whose maximumsize is *max\_queue\_size*.

Boolean IsFull(*queue*, *max\_queue\_size*) ::= if (number of elements in *queue* == *max\_queue\_size* ) return TRUE else return FALSE

Queue Enqueue(*queue*, *item*) ::= if (IsFull(*queue*)) *queue-full* else insert *item* at rear of *queue* and return *queue*

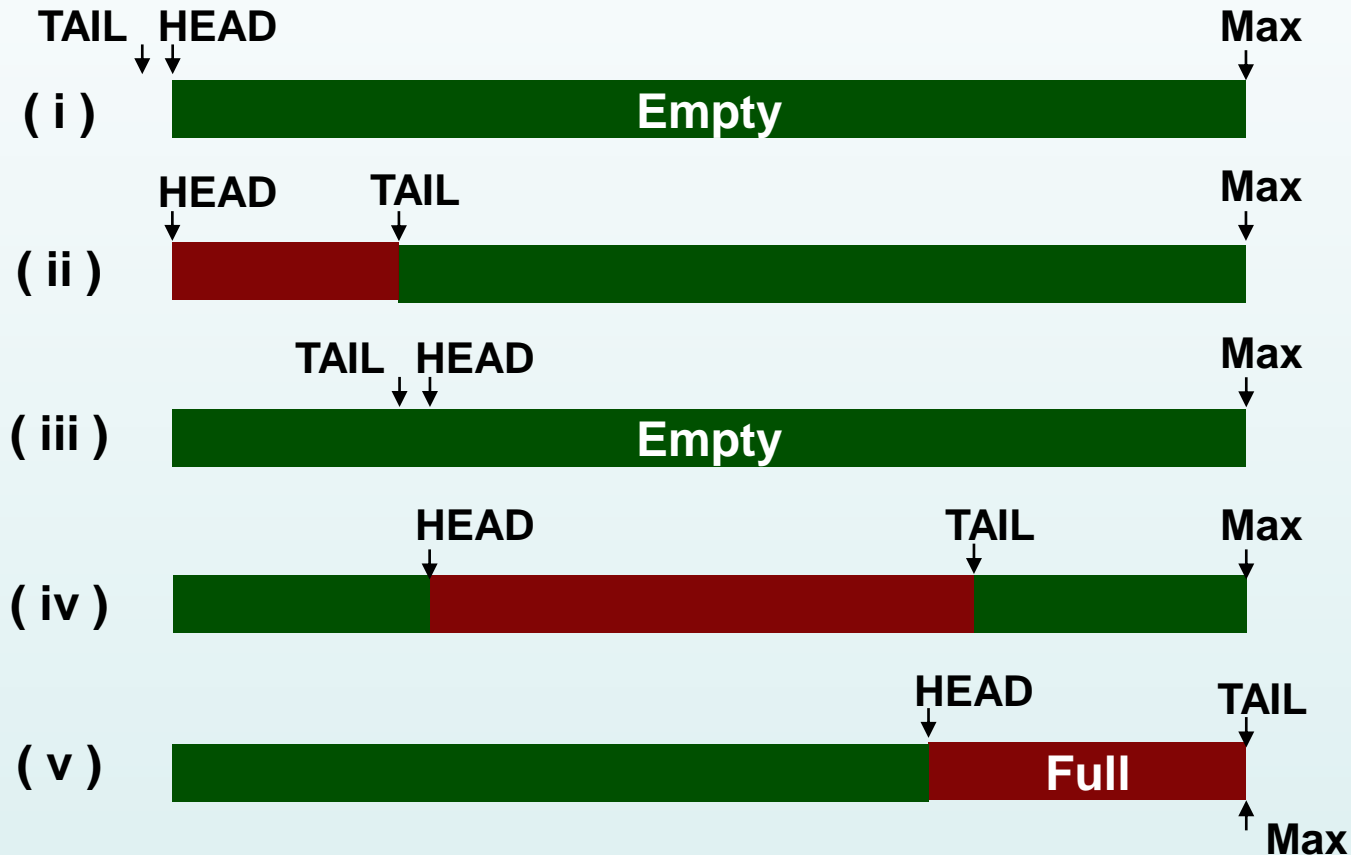
Boolean IsEmptyQ(*queue*) ::= if (*queue* == Create(*max\_queue\_size*)) return TRUE else return FALSE

Element Dequeue(*queue*) ::= if (IsEmpty(*queue*)) return else remove and return the *item* at front of *queue*.

# Operasi

- **CREATE()**  
Untuk menciptakan QUEUE yg baru dan kosong.
- **CLEAR()**  
Untuk menghapus semua elemen dalam QUEUE.
- **ENQUEUE (ElementType e)**  
Untuk memasukkan 1 elemen ke dalam QUEUE.
- **DEQUEUE (ElementType \*e)**  
Untuk mengambil 1 elemen dari QUEUE, disebut juga operasi SERVE.
- **int EMPTY()**  
Untuk mengecek apakah QUEUE masih kosong atau sudah berisi data.
- **int FULL()**  
Untuk mengecek apakah QUEUE sudah penuh atau masih bisa menampung data.

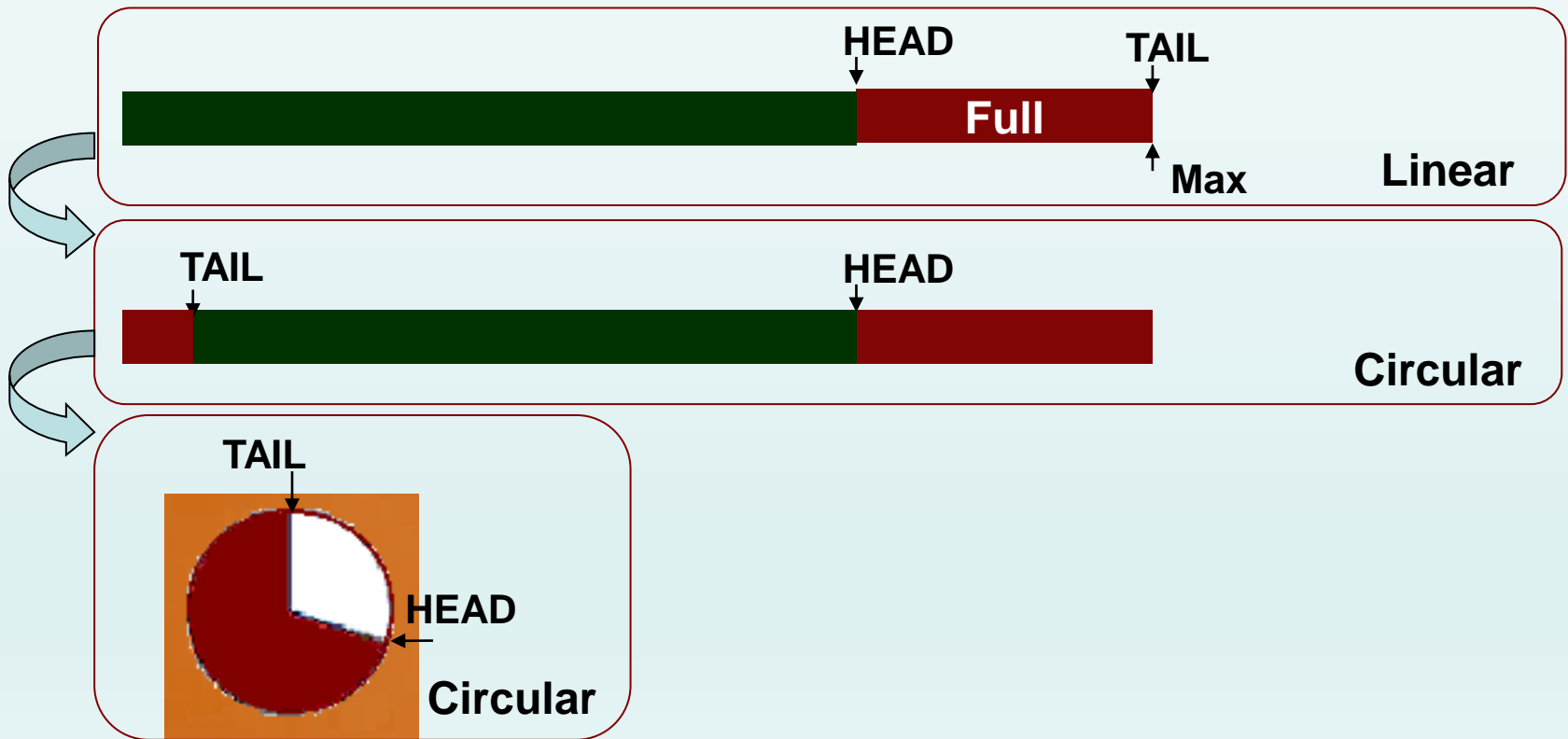
# Implementasi dg Array-Linear



**QUEUE sudah penuh meskipun masih ada tempat yg kosong di depan => ( v )**

# Implementasi dg Array-Circular

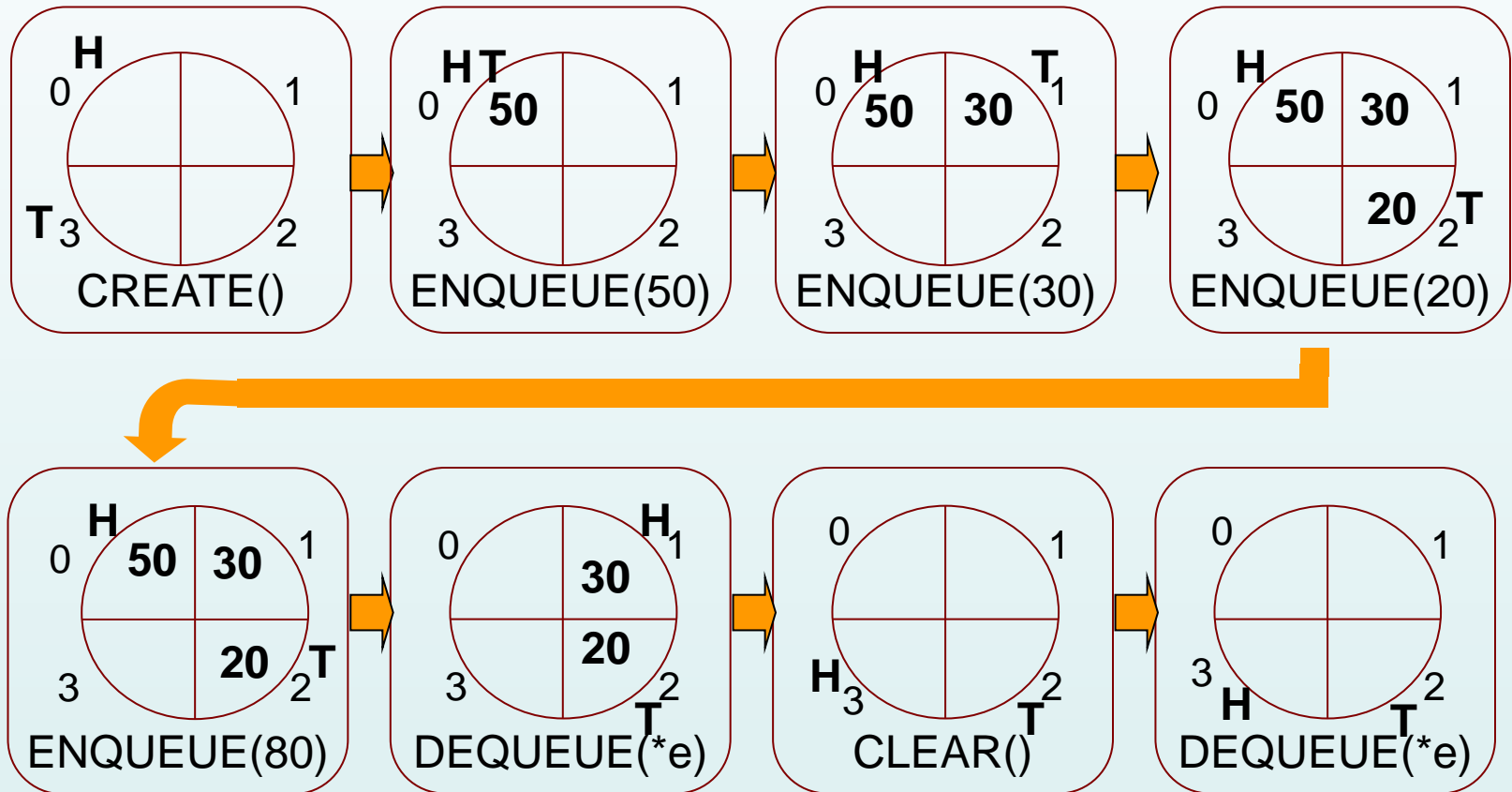
- Untuk mengoptimalkan QUEUE, maka ruang yg kosong bisa dimanfaatkan.
- Yg dilakukan adalah dg memindahkan / memutar posisi TAIL ke depan ( tempat kosong ) saat dilakukan ENQUEUE dan juga posisi HEAD saat dilakukan DEQUEUE





# Simulasi Circular Queue

Contoh operasi pd Circular QUEUE yang memiliki 4 sel ( jumlah maksimum 3 ) :



# Implementasi Circular Queue

Implementasi Circular QUEUE dengan Bahasa C

```
void CREATE()
```

```
{  HEAD = 0;  
    TAIL  = PJG_MAX - 1;}
```

```
void ENQUEUE(ElemenType e)
```

```
{  if (FULL()) printf("Queue sudah penuh\n");  
    else{ TAIL = TAIL++ % PJG_MAX;  
        Q[TAIL]=e; }  
}
```

```
void DEQUEUE(ElemenType *e)
```

```
{  if (EMPTY()) printf("Queue kosong\n");  
    else{ *e = Q[HEAD];  
        HEAD = HEAD++ % PJG_MAX; }  
}
```

# Implementasi Circular Queue(2)

```
int EMPTY()
{   if (((TAIL+1) % PJG_MAX) == HEAD) return(1);
    else return(0);
}
```

```
int FULL()
{   int x;
    x = TAIL+2;
    x = x % PJG_MAX;
    if (x == HEAD) return(1);
    else return(0);
}
```

# Implementasi dg Linked List

```
#include <stdio.h>
#include <malloc.h>
typedef int ElementType;
struct Node{
    ElementType data;
    struct Node *next;
};
struct QUEUE{
    struct Node *Head;
    struct Node *Tail;
};
struct QUEUE Q;
void create()
{
    Q.Head = NULL;
    Q.Tail = NULL;
}
```

# Implementasi dg Linked List(2)

```
void enqueue(ElemenType e)
{
    struct Node *Temp;
    Temp=(struct Node*)malloc(sizeof(struct Node));
    Temp->data = e;
    Temp->next = NULL;
    if(empty()){
        Q.Head = Temp;
        Q.Tail = Temp; }
    else{
        Q.Tail->next = Temp;
        Q.Tail = Temp;
    };
};
```

# Implementasi dg Linked List(3)

```
void dequeue(ElementType *e)
{  struct Node *Temp;
   if(!empty()) {
       Temp = Q.Head;
       Q.Head = Q.Head->next;
       *e = Temp->data;
       free(Temp);
   };
};
```

# Implementasi dg Linked List(3)

```
void main ()
{   int i,j;
    int *k;
    enqueue(10);
    enqueue(125);
    enqueue(20);
    enqueue(225);
    enqueue(30);
    enqueue(325);
    for(i=50; i<80; i+=5;) enqueue( i );
    printf("Lihat isi Antruan \n\n");
    while(!empty()) {
        dequeue(&k);
        printf("   %5d   \n", k);
    }
}
```

# Selesai