

# Modul VIII

## Queue

### TUJUAN

1. Mahasiswa mampu menjelaskan definisi dan konsep dari Double Queue
2. Mahasiswa mampu menerapkan operasi tamba, menghapus pada Queue
3. Mahasiswa mampu menerapkan operasi tampil data pada queue

### Pengertian

Queue adalah struktur data yang digunakan untuk menyimpan data dengan metode FIFO (First-In First-Out). Data yang pertama dimasukkan ke dalam queue akan menjadi data yang pertama pula untuk dikeluarkan dari queue. Queue mirip dengan konsep antrian pada kehidupan sehari-hari.

### Implementasi

Implementasi queue dapat dilakukan dengan menggunakan array atau linked list. Struktur data queue terdiri dari dua pointer yaitu front dan rear. front adalah pointer ke elemen pertama dalam queue dan rear adalah pointer ke elemen terakhir dalam queue.

### Fungsi-fungsi pada Queue

Pada Queue, karena perubahan data selalu mengacu pada Head, maka hanya ada 1 jenis insert maupun delete. Prosedur ini sering disebut Enqueue dan Dequeue pada kasus Queue. Untuk Enqueue, cukup tambahkan elemen setelah elemen terakhir Queue, dan untuk Dequeue, cukup "geser"kan Head menjadi elemen selanjutnya.

### Operasi pada Queue

enqueue(): menambahkan data ke dalam queue.

dequeue(): mengeluarkan data dari queue.

peek(): mengambil data dari queue tanpa menghapusnya.

isEmpty(): mengecek apakah queue kosong atau tidak.

isFull(): mengecek apakah queue penuh atau tidak.

size(): menghitung jumlah elemen dalam queue.

## GUIDED 1

```
#include <iostream>
using namespace std;

//queue array
int maksimalQueue = 5;//maksimal antrian
int front = 0;//penanda antrian
int back = 0;//penanda
string queueTeller[5]; //fungsi pengecekan
bool isFull(){//pengecekan antrian penuh atau tidak
    if(back == maksimalQueue){
        return true;//=1
    }else{
        return false;
    }
}
//fungsi pengecekan
bool isEmpty(){//antriannya kosong atau tidak
    if(back==0){
        return true;
    }else{
        return false;
    }
}
//fungsi menambahkan antrian
void enqueueAntrian(string data){
    if(isFull()){
        cout << "antrian penuh"<<endl;
    }else{//nested if, nested for
        if(isEmpty()){//kondisi ketika queue kosong
            queueTeller[0]=data;
            front++;//front = front +1;
            back++;
        }else{//antriannya ada isi
            queueTeller[back]=data;//queueTeller[1]=data
            back++;//back=back+1; 2
        }
    }
}
//fungsi mengurangi antrian
void dequeueAntrian(){    if(isEmpty()){
    cout << "antrian kosong"<<endl;
}
}
else{
    for(int i=0; i<back; i++){
        queueTeller[i]=queueTeller[i+1];
    }back--;
}
}
//fungsi menghitung banyak antrian
```

```

int countQueue(){
    return back;
}
//fungsi menghapus semua antrian
void clearQueue(){
    if(isEmpty()){
        cout << "antrian kosong"<<endl;
    }else{
        for(int i=0; i<back; i++){
            queueTeller[i]="";
        }
        back=0;
        front=0;
    }
}
//fungsi melihat antrian
void viewQueue(){
    cout << "data antrian teller : "<<endl;
    for(int i =0; i<maksimalQueue; i++){
        if(queueTeller[i]!=""){
            cout << i+1 << ". " <<queueTeller[i]<<endl;
        }else{
            cout << i+1 << ". (kosong)" <<endl;
        }
    }
}

int main()
{
    enqueueAntrian("Andi");
    enqueueAntrian("Maya");
    viewQueue();
    cout << "jumlah antrian = " << countQueue()<<endl;
    dequeueAntrian();
    viewQueue();
    cout << "jumlah antrian = " << countQueue()<<endl;
    clearQueue();
    viewQueue();
    cout << "jumlah antrian = " << countQueue()<<endl;
    return 0;
}

```

## UNGUIDED

1. Ubahlah penerapan konsep queue pada bagian guided dari array menjadi linked list
2. Dari nomor 1 buatlah konsep antri dengan atribut Nama mahasiswa dan NIM Mahasiswa

Contoh hasil jadi:

```
1. Masukkan Data
2. Hapus Satu Data
3. Reset Data
4. Tampil Data
Masukkan Pilihan Anda: 1
```

```
Masukkan nama mahasiswa: bagas
Masukkan NIM mahasiswa: 21102198
```

```
1. Masukkan Data
2. Hapus Satu Data
3. Reset Data
4. Tampil Data
Masukkan Pilihan Anda: 4
```

```
Nama Mahasiswa  NIM
bagas           21102198
```