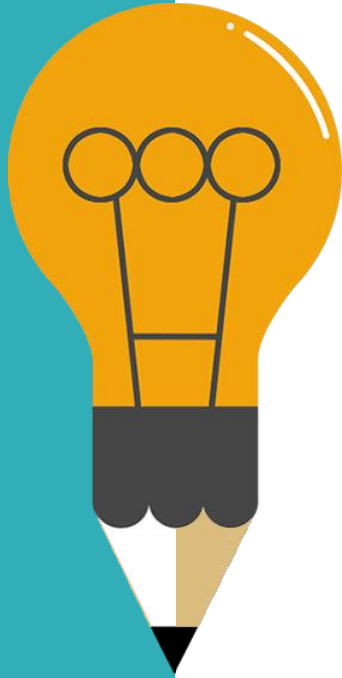


ALGORITMA & STRUKTUR DATA (IFUWP3337)

Dosen Pengampu:
Yosep Septiana, S.Kom., M.Kom.



QUEUE (ANTRIAN)



01

Struktur Data Queue (Antrean)

02

Operasi Dasar pada Queue (Antrean)

03

Ilustrasi Queue (Antrean)

04

Tugas 6



Struktur Data

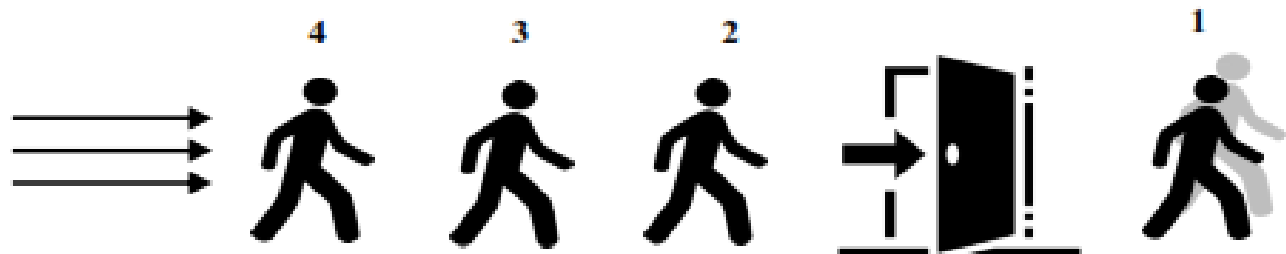
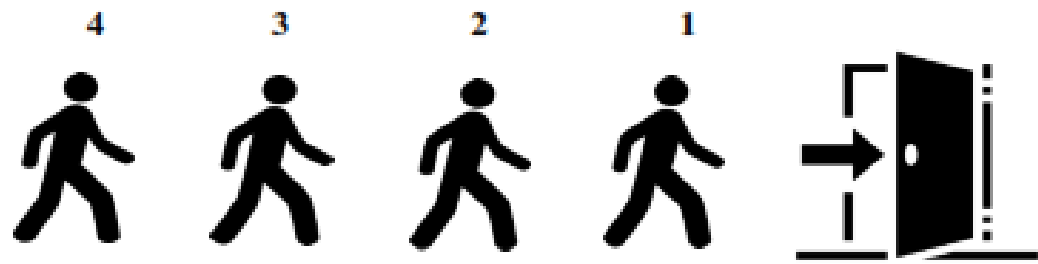
Queue (Antrean)

Konsep Dasar Queue (Antrean)

- ❑ Queue disebut juga antrian dimana data masuk di satu sisi dan keluar di sisi yang lain.
- ❑ Konsep penyimpanan data, terdiri dari beberapa data, dengan aturan data yang baru masuk akan diletakkan di data sebelumnya, data yang keluar adalah data sebelumnya, data yang keluar adalah data yang masuk pertama kali (FIFO, First in – First Out).

Konsep Dasar Queue (Antrean)

- ❑ Antrian (Queue) merupakan suatu kumpulan data yang penambahan elemennya (masuk antrian) hanya bisa dilakukan pada suatu ujung (disebut dengan sisi belakang/rear) atau disebut juga enqueue yaitu apabila data masuk ke dalam sebuah antrian.
- ❑ Jika data keluar dari antrian/penghapusan (pengambilan elemen) dilakukan lewat ujung yang lain (disebut dengan sisi depan/front) atau disebut juga dequeue yaitu apabila seseorang keluar dari antrian.



Konsep Dasar Queue (Antrean)

- ❑ Perbedaan antara stack dan queue terdapat pada aturan penambahan dan penghapusan elemen. Pada stack, operasi penambahan dan penghapusan elemen dilakukan di satu tempat atau satu ujung (LIFO).
- ❑ Sementara pada queue operasi penambahan dapat dilakukan di tempat yang berbeda. Penambahan elemen dilakukan di salah satu ujung, biasa disebut dengan elemen depan, dan ujung lainnya, yang biasa disebut dengan ujung belakang digunakan untuk penghapusan data (FIFO).

Depan/Head

←	23	30	34	50	120	←
	30	34	50	120		
	30	34	50	120	170	

Belakang/Tail

Item masuk dari belakang,
keluar dari depan

Setelah keluar

Input data lagi (170)



Operasi Dasar pada Queue (Antrean)

Operasi Dasar Queue (Antrean)

- ❑ **Init** : Membuat Queue (Inisialisasi)
- ❑ **InQueue** : Memasukan elemen ke dalam Queue (Insert Queue)
- ❑ **DeQueue** : Menghapus Elemen dari Queue (Delete Queue)
- ❑ **Full** : Mengecek apakah Queue penuh
- ❑ **Empty** : Mengecek apakah Queue kosong

Operasi Dasar Queue (Antrian)

- ❑ **Init**, merupakan prosedur untuk membuat queue pada kondisi awal, yaitu queue yang masih kosong (belum mempunyai elemen).

```
void init(void)
{
    antre.awal= -1;
    antre.akhir= -1;
}
```

Operasi Dasar Queue (Antrian)

- ❑ **InQueue**, Insert Queue merupakan prosedur untuk memasukkan sebuah elemen baru pada queue. Jumlah elemen Queue akan bertambah satu dan elemen tersebut merupakan elemen belakang.

```
void inQueue(){
    tampilData();
    int elemen;

    if(!full()){
        cout<<"Data yang akan dimasukkan : ";
        cin>>elemen;
        cout<<"Data berhasil ditambahkan\n";
        antrean.data[antrean.akhir]=elemen;
        antrean.akhir++;
    } else {
        cout<<"Queue penuh\n";
    }
    getch();
}
```

Operasi Dasar Queue (Antrian)

- ❑ **DeQueue**, Delete Queue merupakan prosedur untuk menghapus/mengambil sebuah elemen dari queue. Elemen yang diambil adalah elemen depan dan jumlah elemen queue akan berkurang satu.

```
void deQueue() {  
    int i;  
  
    if(!empty()) {  
        for(i=antre.awal; i<antre.akhir; i++){  
            antre.data[i]=antre.data[i+1];  
        }  
        antre.akhir--;  
    } else {  
        cout<<"Antrean kosong";  
    }  
}
```

Operasi Dasar Queue (Antrian)

- ❑ **Full**, merupakan prosedur untuk mengetahui apakah Queue penuh atau tidak. Prosedur ini hanya berlaku untuk queue yang jumlahnya terbatas.

```
bool full(void)
{
    if(antre.akhir== MAX-1){
        return true;
    } else {
        return false;
    }
}
```

Operasi Dasar Queue (Antrian)

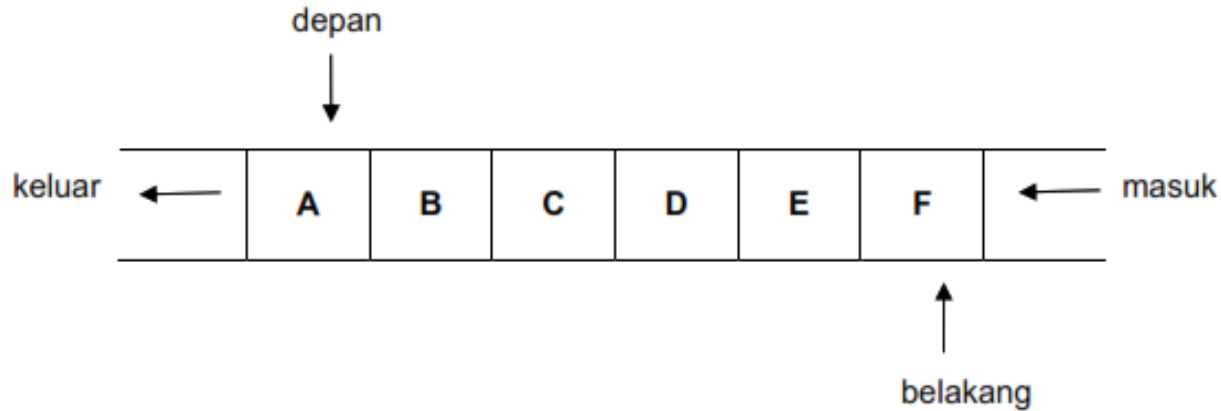
- ❑ **Empty**, yaitu prosedur untuk mengetahui apakah queue dalam keadaan kosong atau tidak. Dengan status ini maka dapat dicegah dilakukannya operasi Dequeue dari suatu queue yang kosong.

```
bool empty(void)
{
    if(antre.akhir== -1){
        return true;
    } else {
        return false;
    }
}
```



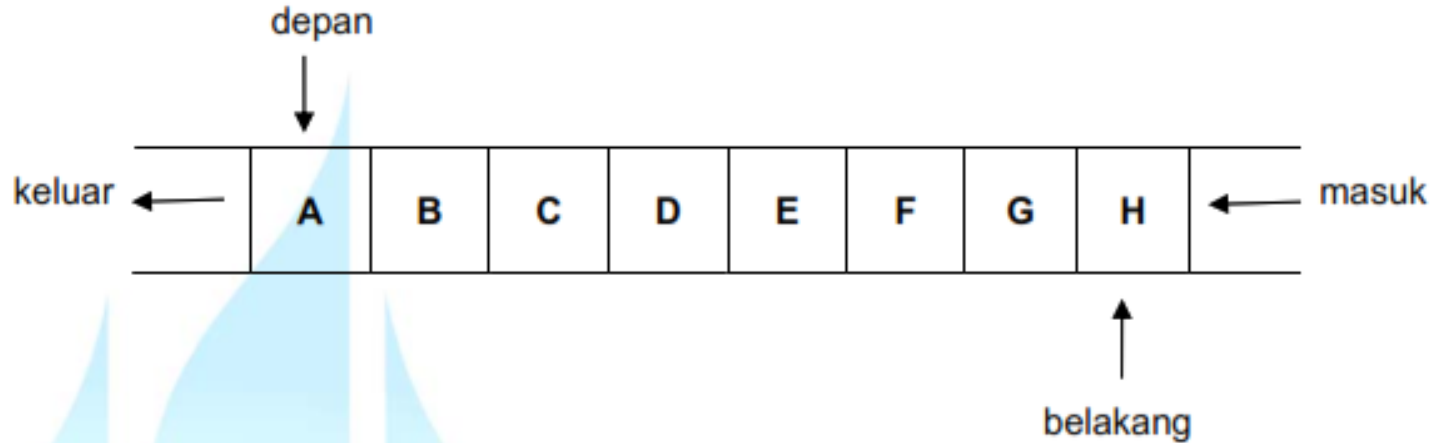

Ilustrasi Queue (Antrean)

Ilustrasi Queue (Antrian)



Gambar tersebut menunjukkan contoh penyajian antrian menggunakan array. Antrian tersebut berisi 6 elemen, yaitu A, B, C, D, E dan F. Elemen A terletak di bagian depan antrian dan elemen F terletak di bagian belakang antrian. Jika ada elemen baru yang akan masuk, maka elemen tersebut akan diletakkan di sebelah kanan F. Jika ada elemen yang akan dihapus, maka A akan dihapus terlebih dahulu.

Ilustrasi Queue (Antrian)



Elemen A terletak di bagian depan, kemudian disusul elemen B dan elemen yang paling akhir atau paling belakang adalah elemen F.

Misalkan ada elemen baru yang akan masuk maka akan terletak di belakang elemen F, sehingga elemen baru akan menempati posisi yang paling belakang (penambahan elemen G dan H).



Sekian dan Terima Kasih



TUGAS 6