

# QUEUE AND CIRCULAR



## QUEUE Kelompok 3

Alfiana Rahmawati (22040700076)

Gunawan (22040700083)

Muhammad Akhyar (22040700098)

(22040700064)

Muhammad Daffa (2018470071)

Fadilah

Nur Alfi Syahri

# TOPIC OF PRESENTATION

01  
Pengertian Queue

02  
Karakteristik Queue

03  
Operasi Queue

04  
Fungsi dan Kegunaan Queue

05  
Circular Queue



# Pengertian Queue

Queue/antrian adalah suatu kumpulan data yang mana penambahan data hanya bisa dilakukan pada satu sisi yaitu belakang (tail) dan penghapusan data dilakukan lewat sisi depan (head). Berbeda dengan stack yang bersifat LIFO (last in first out), maka queue bersifat FIFO (First in First Out) yaitu data yang masuk pertamakali akan keluar terlebih dahulu dan data yang terakhir masuk akan keluar terakhir.

## Ilustrasi dari Queue



# Karakteristik

## Queue

1. Elemen antrian yaitu item item data yang terdapat di elemen antrian
2. Head / front (elemen terdepan dari antrian)
3. Tail /Real (elemen terakhir dari antrian)
4. Jumlah elemen pada antrian (count)
5. Untuk menghapus elemen terakhir dari queue, semua elemen yang dimasukkan sebelum elemen tersebut harus dihilangkan atau dihapus
6. Queue adalah daftar berurutan dari elemen-elemen dari tipe data yang serupa
7. Status / kondisi antrian



Kondisi antrian yang menjadi perhatian, yaitu :

## 1. Penuh

Bila elemen di antrian mencapai kapasitas maksimum antrian maka kondisi tidak mungkin dilakukan penambahan ke antrian. Penambahan elemen menyebabkan kondisi kesalahan Overflow.

## 2. Kosong

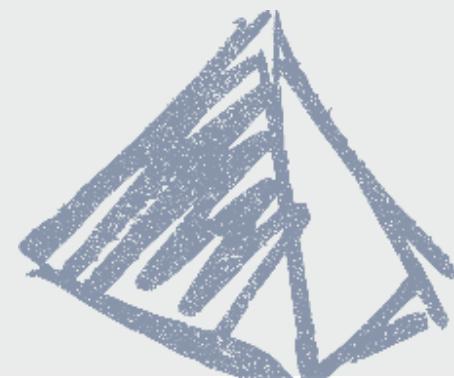
Bila tidak ada elemen di antrian maka kondisi tidak mungkin dilakukan pengambilan elemen dari antrian. Pengambilan elemen menyebabkan kondisi kesalahan Underflow.



# OPERASI OPERASI DASAR

- **Create** : Membuat dan ~~memeriksa~~ **QUEUE** Queue dengan cara membuat Head dan Tail = -1
- **IsEmpty** : Memeriksa apakah antrian kosong
- **IsFull** : Memeriksa apakah antrian sudah penuh
- **Enqueue** : Menambahkan elemen ke akhir antrian
- **Dequeue** : Menghapus elemen dari depan antrian
- **Tampil** : Menampilkan nilai nilai elemen antrian
- **Clear** : Menghapus elemen elemen dengan cara membuat Head dan Tail = -1

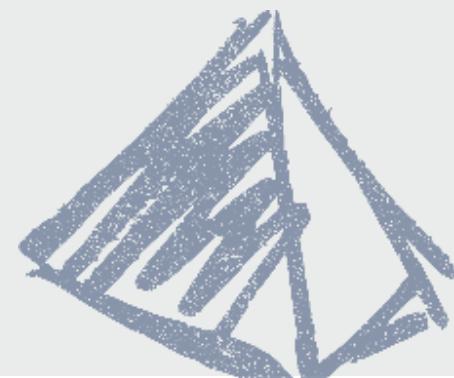
Namun, secara umum antrian memiliki 2 operasi utama, yaitu **enqueue** dan **dequeue**.



# OPERASI ENQUEUE

Di bawah ini adalah langkah-langkah untuk enqueue (memasukkan) data ke dalam antrian :

1. Periksa apakah antrian sudah penuh atau tidak.
2. Jika antrian penuh – cetak kesalahan *overflow* dan keluar dari program.
3. Jika antrian tidak penuh – naikkan pointer belakang untuk menunjuk ke ruang kosong berikutnya.
4. Tambahkan elemen pada posisi yang ditunjuk oleh pointer belakang.
5. Kembalikan status bahwa penambahan telah berhasil



# Pseuode untuk operasi enqueue

```
procedure enqueue (data)

    if queue is full

        return overflow

    endif

    rear ← rear + 1

    queue[rear] ← data

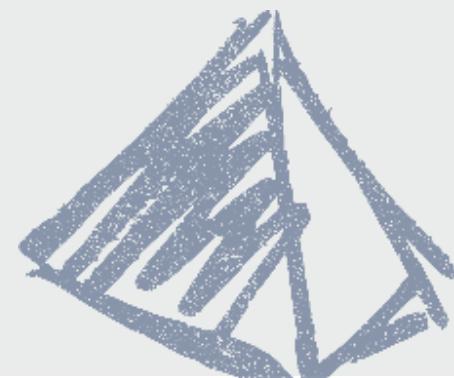
    return true

end procedure
```

# OPERASI DEQUEUE

Langkah-langkah untuk melakukan operasi dequeue, yaitu :

1. **Periksa apakah antrian sudah penuh atau tidak.**
2. **Jika antrian kosong – cetak kesalahan underflow dan keluar dari program.**
3. **Jika antrian tidak kosong – akses elemen data yang ditunjuk oleh pointer depan.**
4. **Geser pointer depan untuk menunjuk ke elemen data berikutnya yang tersedia.**
5. **Kembalikan status bahwa operasi penghapusan telah berhasil**



# Pseuode untuk operasi dequeue

```
procedure dequeue

    if queue is empty

        return underflow

    end if

    data = queue[front]

    front ← front + 1

    return true

end procedure
```

# Fungsi dan kegunaan queue

Berikut ini adalah beberapa fungsi queue yang paling umum dalam struktur data:

- Queue banyak digunakan untuk menangani lalu lintas (traffic) situs web.
- Membantu untuk mempertahankan playlist yang ada pada aplikasi media player
- Queue digunakan dalam sistem operasi untuk menangani interupsi.
- Membantu dalam melayani permintaan pada satu sumber daya bersama, seperti printer, penjadwalan tugas CPU, dll.
- Digunakan dalam transfer data asinkronus misal pipeline, IO file, dan socket.

# Kelebihan Queue

Kelebihan queue di antaranya, yaitu :

- Data dalam jumlah besar dapat dikelola secara efisien.
- Operasi seperti penyisipan dan penghapusan dapat dilakukan dengan mudah karena mengikuti aturan masuk pertama keluar pertama.
- Queue berguna ketika layanan tertentu digunakan oleh banyak konsumen.
- Queue cepat untuk komunikasi antar-proses data.
- Queue dapat digunakan dalam implementasi struktur data lainnya.

# Kekurangan Queue

Kelemahan struktur data queue sebagai berikut :

- Operasi seperti penyisipan dan penghapusan elemen dari tengah cenderung banyak memakan waktu.
- Dalam queue konvensional, elemen baru hanya dapat dimasukkan ketika elemen yang ada dihapus dari antrian.
- Mencari elemen data pada struktur queue membutuhkan time complexity  $O(N)$ .
- Ukuran maksimum antrian harus ditentukan sebelumnya.



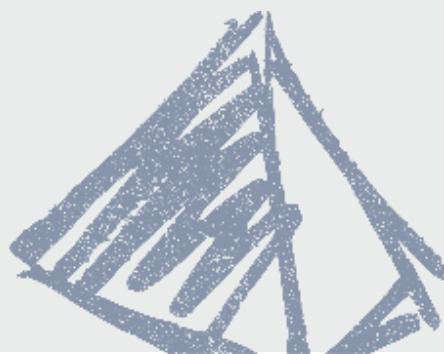
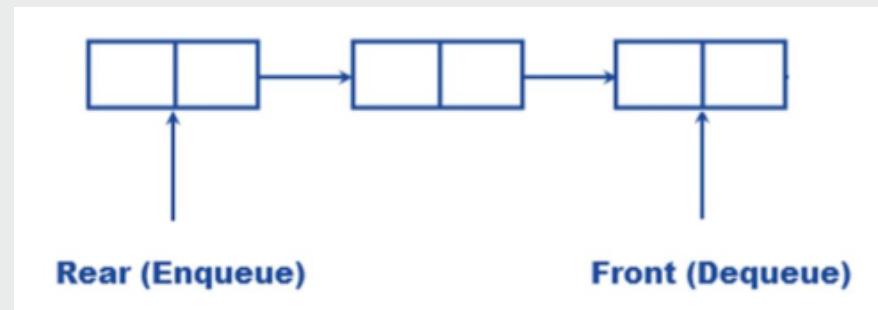
## Jenis Jenis Queue

Secara umum ada 4 jenis struktur data queue, meliputi :

1. Simple Queue
2. Circular Queue
3. Priority Queue
4. Double-Ended Queue (Dequeue)

# SIMPLE Queue

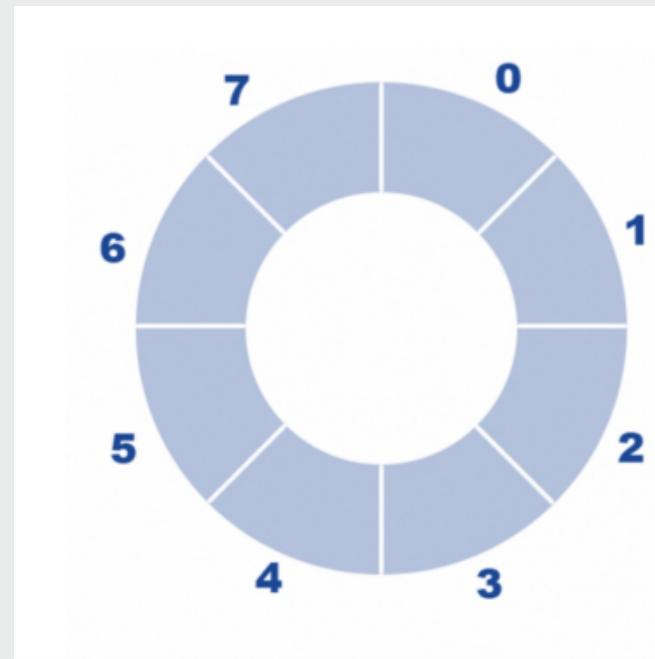
Simple Queue adalah Struktur data queue paling dasar di queue dimana penyisipan item dilakukan di simpul belakang (Rear atau Tail) dan penghapusan terjadi di simpul depan (Front atau Head).



# Circular Queue

Pada circular queue, simpul terakhir terhubung ke simpul pertama. Queue jenis ini juga dikenal sebagai Ring Buffer karena semua ujungnya terhubung ke ujung yang lain.

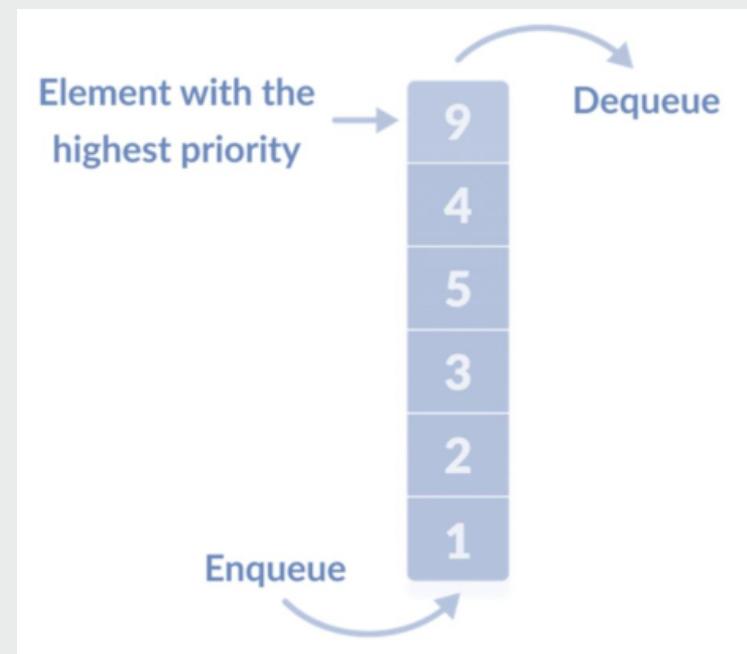
Penyisipan terjadi di akhir antrian dan penghapusan di depan antrian.



# Priority Queue

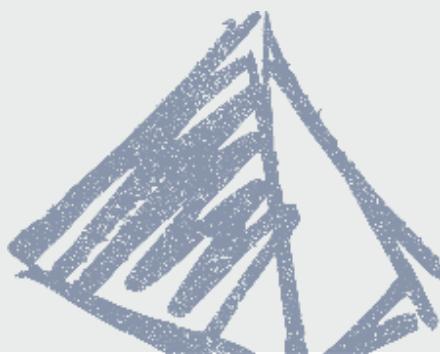
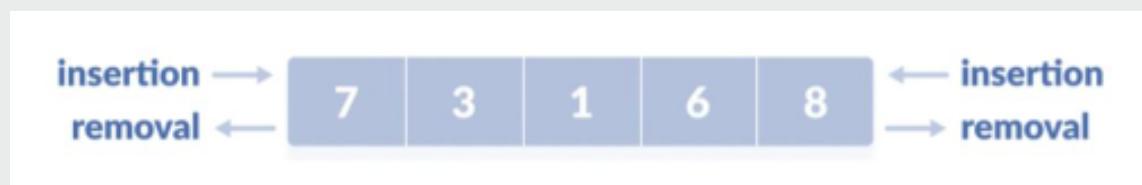
Priority Queue adalah Struktur data queue dimana simpul akan memiliki beberapa prioritas yang telah di tentukan. Simpul dengan prioritas terbesar akan menjadi yang pertama dihapus dari antrian.

Sedangkan penyisipan item terjadi sesuai urutan kedatangannya.



# DOUBLE ENDED Queue (DEQUEUE)

Double Ended Queue yaitu Struktur data queue yang operasi penyisipan dan penghapusan dapat terjadi di ujung depan dan belakang dari antrian.



```
1 #include<iostream>
2 #include<conio.h>
3 #include<stdlib.h>
4 #define MAX 10
5
6 using namespace std;
7
8 int nomor[MAX];
9 int head = -1;
10 int tail = -1;
11
12 bool IsEmpty() {
13     if(tail == -1){
14         return true;
15     }else{
16         return false;
17     }
18 }
19
20 bool IsFull(){
21     if (tail == MAX-1){
22         return true;
23     }else{
24         return false;
25     }
26 }
27 void AntrianMasuk(int urut){
28     if (IsEmpty()){
29         head=tail=0;
30     } else{
31         tail++;
32     }
33     nomor[tail]=urut;
34 }
35 void AntrianKeluar(){
36     if(IsEmpty()){
37         cout<<"Antrian sudah kosong! ";
38     }

```

```
39     getch();
40 }
41 else {
42     for (int a=head; a<tail; a++){
43         nomor[a]=nomor[a+1];
44     }
45     tail--;
46     if(tail == -1){
47         head = -1;
48     }
49 }
50 }
51 void Clear(){
52     head=tail= -1;
53 }
54
55 void View(){
56     if(IsEmpty()){
57         cout<<"Antrian kosong ! ";
58     } else
59     {
60         system("cls");
61         for(int a=head; a<=tail; a++){
62             cout<< "-----"
63             << "\n >> No. Antri : [" << nomor[a] << "]"
64             << "\n -----"
65             cout << "\n\n";
66         }
67     }
68 }
69
70 }
71 }
72
73 int main(){
74     system("color 79");
75     int choose, p=1, urut;
76 }
```

```
77
78
79     do{
80
81         system("cls");
82         cout << "\n\n          PROGRAM ANTRIAN MENGGUNAKAN C++ "
83         << "\n -----"
84         << "\n\n 1. Tambah Antrian"
85         << "\n 2. Panggil Antrian"
86         << "\n 3. Lihat daftar antrian"
87         << "\n 4. Format"
88         << "\n 5. Exit"
89         << "\n -----";
90
91         cout << "\n\n Masukkan Nomor : "; cin >> choose;
92         cout << "\n\n";
93
94
95         if(choose == 1){
96             if(IsFull()){
97                 cout<<"Antrian sudah penuh, mohon untuk tunggu kembali.... ";
98             }
99             else{
100
101                 urut=p;
102                 AntrianMasuk(urut);
103                 cout << "-----" << endl;
104                 cout << " | NO. ANTRIAN | " << endl;
105                 cout << " | " << p << " | " << endl;
106                 cout << "-----\n" << endl;
107                 cout << "-----" << endl;
108                 cout << " | Silahkan Mengantre | " << endl;
109                 cout << " | Menunggu " << tail << " Antrian | " << endl;
110                 cout << "-----" << endl;
111                 p++;
112             }
113         }
114         else if(choose == 2){
115             cout << "-----" << endl;
116             cout << " No. Antri : " << nomor[head] << " ";
117             cout << "\n -----" << endl;
118
119             AntrianKeluar();
120             cout << "\n\n Memanggil Antrian !" << endl;
121         }
122         else if(choose == 3){
123             View();
124         }
125         else if(choose == 4){
126             Clear();
127             cout<<"Antrian dikosongkan ! ";
128         }
129         else if(choose == 5){
130
131             cout << "Masukan anda salah ! \n" << endl;
132         }
133         getch();
134     }while(choose!=5);
135
136 }
```



# SESI TANYA JAWAB

