

PERTEMUAN 6:

LINEAR QUEUE

A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai antrian data (Queue) yang terdapat pada struktur data. Di modul ini, Anda harus mampu:

6.1 Merepresentasikan Queue dalam bahasa pemrograman .

B. URAIAN MATERI

Tujuan Pembelajaran 6.1:

Aplikasi Queue

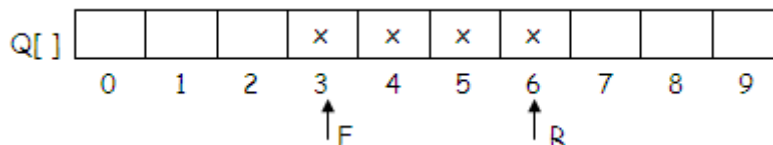
Queue menggunakan array 1D:

1. Linear Queue (Antrian Lurus)
2. Circular Queue (Antrian Melingkar)
3. Double Ended Queue/Deque (Antrian dengan ujung ganda)

1. Linear Queue

I. Ilustrasi

Misal $n = 10$



F (Front) : menunjuk pengantri paling depan/siap untuk keluar/ siap untuk dilayani

R (Rear) : menunjuk pengantri paling belakang/paling akhir masuk

$R = 6$, artinya :

- Pernah masuk 7 pengantri dengan urutan masuk $Q[0]$, $Q[1]$, $Q[2]$, $Q[3]$, $Q[4]$, $Q[5]$, $Q[6]$

$F = 3$, artinya :

- Sudah keluar sebanyak 3 pengantri dengan urutan keluar $Q[0]$, $Q[1]$, $Q[2]$

II. Prinsip : FIFO(First In First Out) atau
FIFS (First In First Serve)

III. Proses :

- AWAL (Inisialisasi)
- INSERT (Sisip, Masuk, Simpan, Tulis)
- DELETE (Hapus, Keluar, Ambil, Dilayani)
- RESET (Kembali ke keadaan awal)

a) Fungsi dasar untuk proses AWAL :

```
void AWAL(void)
{
    F = 0;
    R = -1;
}
```

b) Fungsi dasar proses INSERT :

```
void INSERT(void)
{
    R = R + 1;
    Q[R] = x;
}
```

atau

```
void INSERT(void)
{
    Q[++R] = x;
}
```

c) Fungsi dasar proses DELETE :

```
void DELETE(void)
{
    x = Q[F];
    F = F + 1;
}
```

atau

```
void DELETE(void)
{
    x = Q[F++];
}
```

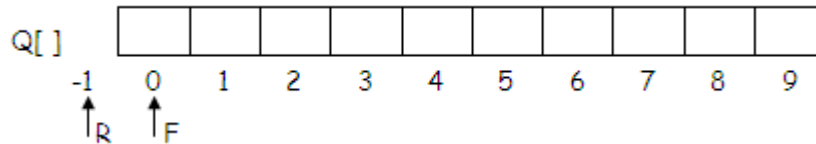
d) Fungsi dasar proses RESET :

```
void RESET(void)
{
    F = 0;
    R = -1;
}
```

IV. Kondisi antrian (n : jml elemen array)

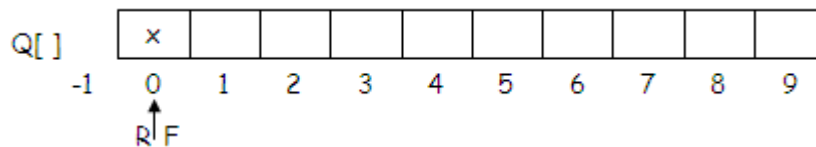
1. Kondisi awal

- $F = 0, R = -1$ kondisi awal
- $F = R + 1$ antrian kosong
- $R < n - 1$ antrian bisa diisi



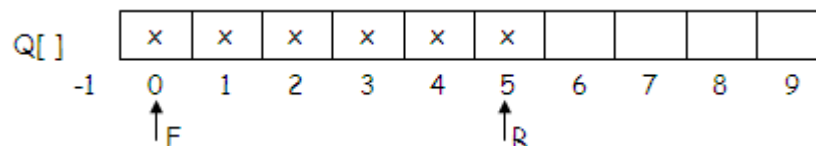
2. Misal masuk 1 pengantri, belum ada yang keluar

- $F = 0$ belum ada yang keluar
- $F < R + 1$ antrian ada isinya
- $R < n - 1$ antrian bisa diisi



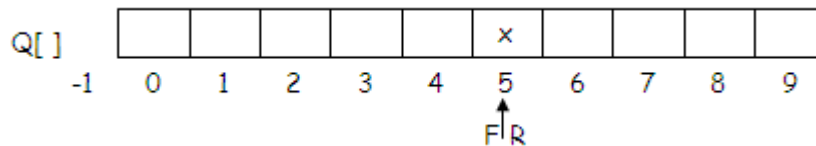
3. Misal masuk lagi 5 pengantri, belum ada yang keluar

- $R = 5$ sudah pernah masuk 6
- $F = 0$ belum ada yang keluar
- $F < R + 1$ antrian ada isinya
- $R < n - 1$ antrian bisa diisi



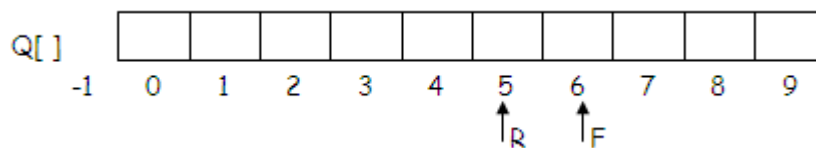
4. Misal keluar 5 pengantri

- $F = 5$ sudah keluar 5 pengantri
- $F = R$ tinggal 1 pengantri
- $F < R + 1$ antrian ada isinya
- $R < n - 1$ antrian bisa diisi



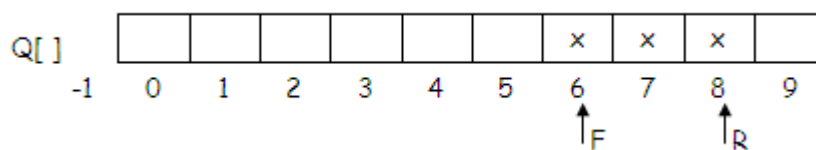
5. Misal keluar lagi satu pengantri

- $F = 6$ sudah keluar 6 pengantri
- $R = 5$
- $F = R + 1$ antrian kosong
- $R < n + 1$ antrian bisa diisi



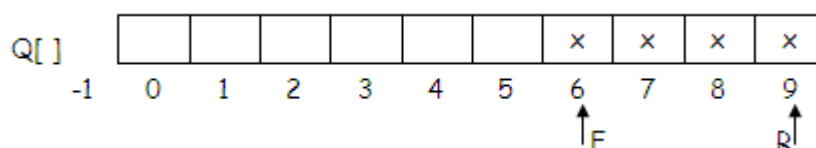
6. Misal masuk lagi 3 pengantri

- $F < R + 1$ antrian ada isinya
- $R < n - 1$ antrian masih bisa diisi



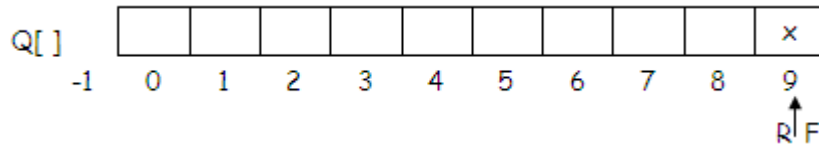
7. Misal masuk lagi 1 pengantri

- $F < R + 1$ antrian ada isinya
- $R = n - 1$ antrian penuh



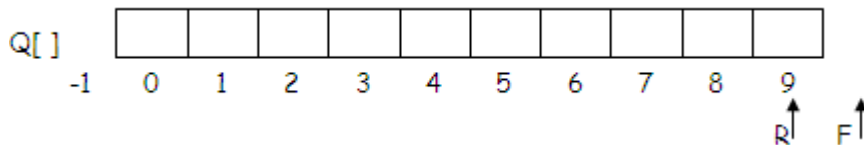
8. Misal keluar 3 pengantri

- $F < R + 1$ antrian ada isinya
- $F = R$ antrian sisa 1 pengantri
- $R = n - 1$ antrian penuh



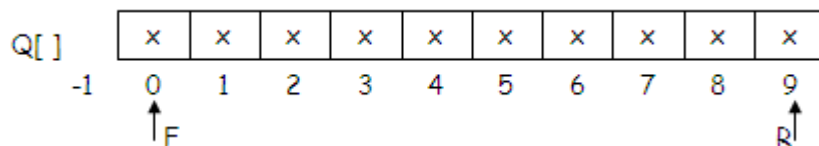
9. Misalkan keluar 1 pengantri

- $F = n$ semua antrian sudah keluar
- $F = R + 1$ antrian kosong
- $R = n - 1$ antrian penuh (disebut penuh)
- $F = R + 1$ dan $R = n - 1$ kondisi khusus :
 - Kosong karena tidak ada isinya
 - Penuh karena tidak bisa diisi
 - Perlu direset(kembali ke posisi awal)



10. Kondisi khusus lainnya :

- a. Antrian penuh tapi belum ada yang keluar
 - i. $F = 0$ belum ada yang keluar
 - ii. $R = n - 1$ antrian penuh



Kondisi antrian:

	Kondisi	Ciri
a	Kosong	$F = R + 1$ dimana saja
b	Penuh	$R = n - 1$
c	Bisa diisi	$R < n - 1$
d	Ada isinya	$F < R + 1$
e	Perlu direset	$F = R + 1$ dan $R = n - 1$

V. Fungsi INSERT dan DELETE lengkap

```
void INSERT(void)
{
    if(R < n - 1)
        Q[++R] = x;
    else
        cout<<"Antrian penuh";
}

void DELETE(void)
{
    if(F < R + 1)
    {
        x = Q[F++];
        if(F == n)
        {
            F = 0;
            R = -1;
        }
    }
    else
        cout<<"Antrian kosong";
}
```

Soal

1. Buatlah suatu program Animasi Antrian dengan 4 buah pilihan : INSERT, DELETE, CETAK ANTRIAN, QUIT.

Jika dipilih INSERT : program akan meminta user untuk menginput sebuah karakter yang akan dimasukan kedalam antrian

Jika dipilih DELETE : maka karakter pertama masuk akan dikeluarkan dari antrian

Jika dipilih CETAK ANTRIAN : komputer menampilkan karakter yang ada pada antrian

Jika dipilih QUIT : program keluar

C. DAFTAR PUSTAKA

Buku

1. Esakov, Jeffrey, Tom Weiss, Data Structures An Advanced Approach Using C, Prentice-Hall, Inc. 1989
2. Hariyanto, Bambang, Struktur Data, Informatika Bandung, Pebruari 2000
3. Kadir, Abdul, Pemrograman Dasar Turbo C, Andi Offset, Yogyakarta, 1991
4. Kruse, Robert L. Data Structures & Program Design, Prentice-Hall, Inc. 1987
5. Standish, Thomas A. Data Structures, Algorithms & Software Principles In C, Addison Wesley, 1995