

# Struktur Data Algoritma

Muhammad Azhar Rasyad

0110217029

Teknik Informatika 1

Kelas Pagi

Sekolah Tinggi Teknologi Terpadu Nurul Fikri

Queue

# Definisi

Queue atau antrian merupakan kumpulan data yang tersusun secara berurutan serta memiliki satu ujung keluar dan satu ujung masuk sehingga data yang duluan masuk akan keluar duluan dan data yang masuk belakangan akan keluar belakangan.

# Deklarasi

Terdiri dari data, head, dan tail. Data berfungsi menyimpan informasi yang dibawa, Head berfungsi menunjukkan ujung antrian, dan tail berfungsi menunjukkan akhir antrian. Untuk mendeklarasikan antrian dapat menggunakan struct supaya tipe data yang dimasukkan berbeda dan memiliki batas dalam sebuah antrian

```
#define batas 6
```

```
struct Queue
```

```
{
```

```
    int data[batas];
```

```
    int head;
```

```
    int tail;
```

```
} antrian;
```

# Inisialisasi

Proses dalam membuat sebuah antrian kosong dengan menggunakan array, karena array index pertama adalah 0 maka head dan tail memiliki nilai awal -1 karena antrian yang digunakan masih kosong

```
void create()  
{  
    antrian.head = antrian.tail = -1;  
}
```

# Operasi Cek Kosong Antrian

Cek antrian kosong atau tidak berfungsi untuk proses dequeue yaitu proses mengeluarkan data dari antrian, jika data kosong maka tidak perlu menggunakan dequeue melainkan enqueue yaitu memasukkan data ke dalam antrian. Untuk mengecek antrian kosong atau tidak, dapat dilihat dari tailnya apakah bernilai -1 atau tidak

```
int cekKosong()  
{  
    if(antrian.tail == -1)  
        return 1;  
    else  
        return 0;  
}
```

# Operasi Cek Penuh Antrian

Cek antrian penuh atau tidak berfungsi untuk menghentikan proses enqueue agar tidak terjadinya antrian yang berlebihan sehingga proses yang diperlukan adalah proses dequeue. Untuk mengecek antrian penuh atau tidak, dapat dilihat dari tailnya apakah bernilai dari batas antrian -1 atau tidak

```
int cekPenuh()  
{  
    if(antrian.tail == batas-1)  
        return 1;  
    else  
        return 0;  
}
```

# Operasi Enqueue #1

Proses menambah data baru pada antrian, langkah yang harus dilakukan adalah mengecek antrian apakah kosong atau tidak kosong atau penuh supaya dapat menentukan kondisi yang akan dilakukan.

Jika sudah mengetahui kondisinya maka yang bergerak adalah head dengan tail supaya dapat berubah posisi dan data yang duluan masuk akan keluar duluan.



# Operasi Enqueue #2

```
void enqueue(int data)
{
    if(cekKosong() == 1)
    {
        antrian.head = antrian.tail = 0;
        antrian.data[antrian.tail] = data;
        cout << antrian.data[antrian.tail];
    }
    else
    {
        antrian.tail++;
        antrian.data[antrian.tail] = data;
        cout << antrian.data[antrian.tail];
    }
}
```

# Operasi Dequeue #1

Proses mengurangi data pada antrian, langkah yang harus dilakukan adalah menghapus data yang sudah selesai mengantri dan kemudian menggeser data-data lain kedepan antrian sehingga antrian terus berjalan

# Operasi Dequeue #2

```
int dequeue()
{
    int i
    int e = antrian.data[antrian.head];

    for(i = antrian.head; i <= antrian.tail-1; i++)
    {
        antrian.data[i] = antrian.data[i+1];
    }

    antrian.tail--;
    return e;
}
```

# Operasi Kosongkan

Berfungsi menghapus semua data pada antrian sehingga meskipun ada yang masih mengantri atau tidak maka antrian akan tetap dikosongkan dengan cara seperti membuat antrian yang baru

```
void kosongkan()  
{  
    antrian.head = antrian.tail = -1;  
    cout << "Data DiKosongkan";  
}
```

# Operasi Tampil

Berfungsi menampilkan nilai yang ada pada sebuah data di dalam antrian dengan cara melihat satu per satu data yang mengantri

```
void tampil()
{
    int i
    if(cekKosong() == 0)
        for(i = antrian.head; i <= antrian.tail; i++)
            cout << antrian.data[i] << " ";
    else
        cout << "Data Kosong";
}
```

# Kesimpulan

Queue atau antrian merupakan kumpulan data yang disusun secara berurutan yang memiliki satu pintu masuk dan satu pintu keluar sehingga proses yang terjadi adalah data saling menunggu gilirannya masing-masing dan dapat disimpulkan bahwa data yang duluan masuk maka akan keluar duluan.

# Referensi

<http://www.klikartikel.com/2015/01/mengenal-queue-antrian-pada-c.html>

Sekian