# RANGKUMAN TEORI QUEUE

## TEORI DASAR

### Pendahuluan

Queue atau Antrian merupakan kumpulan elemen dengan penyisipan dan penghapusan elemen yang dilakukan dari sisi/gerbang yang berbeda. Penyisipan dilakukan dari gerbang belakang dan penghapusan dilakukan dari gerbang depan. Hal ini menunjukan bahwa untuk Queue mempunyai dua gerbang yaitu gerbang depan dan gerbang belakang. Dengan demikian dapat dilihat bahwa Queue mempunyai sifat FIFO (first In Firs Out), yaitu elemen yang pertama masuk akan keluar pertama juga. Queue dapat direpresentasikan dengan menggunakan Array atau Linked List.

### Operasi-operasi pada Queue

Sama halnya dengan stack, operasi yang dapat dilakukan pada suatu Queue pada dasarnya adalah penyisipan elemen dan penghapusan elemen. Disamping itu juga dapat dilakukan operasi untuk mengecek apakah Queue kosong atau penuh, macam-macam operasi untuk mengecek Queue antara lain:

1. Operasi Inisialisasi
2. Operasi Queue Kosong
3. Operasi Queue Penuh
4. Operasi Mengosongkan Queue
5. Operasi Penyisipan Elemen Queue
6. Operasi Penghapusan Elemen Queue
7. Operasi Pencetakan Isi Queue

# TUGAS PENDAHULUAN QUEUE

1. Apa yang dimaksud dengan Queue!

Queue atau Antrian merupakan kumpulan elemen dengan penyisipan dan penghapusan elemen yang dilakukan dari sisi/gerbang yang berbeda.

1. Tuliskan Deklarasi sintaks Queue!

// Deklarasi struct antrian

struct Queue {

int front, rear, data[MAX];

}Q;

// cek apakah antrian penuh

bool isFull() {

return Q.rear == MAX;

}

// cek apakah antrian kosong

bool isEmpty() {

return Q.rear == 0;

}

// Menampilkan antrian

void printQueue() {

if (isEmpty()) {

cout << "Antrian kosong"<<endl;

}

else {

cout << "QUEUE: ";

for (int i = Q.front; i < Q.rear; i++)

cout << Q.data[i] <<",";

cout << endl;

}

}

// manambahkan data ke antrian

void enqueue() {

if (isFull())

{

cout << "Antrian penuh!"<<endl;

}

else {

int data;

// menambahkan data ke antrian

cout << "Masukkan Data : ";cin >> data;

Q.data[Q.rear] = data;

// menempatkan tail pada elemen data terakhir yang ditambahkan

Q.rear++;

cout << "Data ditambahkan\n";

printQueue();

}

}

// mengambil data dari antrian

void dequeue() {

if (isEmpty())

{

cout << "Antrian masih kosong"<<endl;

}

else{

cout << "Mengambil data \"" << Q.data[Q.front] << "\"..." << endl;

// menggeser antrian data ke head

for (int i = Q.front; i < Q.rear; i++)

Q.data[i] = Q.data[i + 1];

// menempatkan tail pada data terakhir yang digeser

Q.rear--;

printQueue();

}

}

1. Sebutkan dan Jelaskan operasi-operasi pada Queue!

Operasi-operasi pada Queue:

1. Create()

Untuk menciptakan dan menginisialisasi Queue dengan cara membuat Head dan Tail = -1.

2. IsEmpty()

Untuk memeriksa apakah Antrian sudah penuh atau belum dengan cara memeriksa nilai Tail, jika Tail = -1 maka empty

Kita tidak memeriksa Head, karena Head adalah tanda untuk kepala antrian (elemen pertama dalam antrian) yang tidak akan berubah-ubah pergerakan pada Antrian terjadi dengan penambahan elemen Antrian kebelakang, yaitu menggunakan nilai Tail.

3. IsFull()

Untuk mengecek apakah Antrian sudah penuh atau belum dengan cara mengecek nilai Tail, jika Tail >= MAX-1 (karena MAX-1 adalah batas elemen array pada C) berarti sudah penuh.

4. Enqueue()

Untuk menambahkan elemen ke dalam Antrian, penambahan elemen selalu ditambahkan di elemen paling belakang penambahan elemen selalu menggerakan variabel Tail dengan cara increment counter Tail terlebih dahulu.

5. Dequeue()

Digunakan untuk menghapus elemen terdepan/pertama (head) dari Antrian dengan cara menggeser semua elemen antrian kedepan dan mengurangi Tail dgn 1 penggeseran dilakukan dengan menggunakan looping.

6. Clear()

Untuk menghapus elemen-elemen Antrian dengan cara membuat Tail dan Head =-1.

Penghapusan elemen-elemen Antrian sebenarnya tidak menghapus arraynya, namun hanya mengeset indeks pengaksesan-nya ke nilai -1 sehingga elemen-elemen Antrian tidak lagi terbaca.

7. Tampil()

Untuk menampilkan nilai-nilai elemen Antrian menggunakan looping dari head s/d tail.

1. Jelaskan Aplikasi-Aplikasi Queue dalam dunia nyata!

Contoh aplikasi Queue dalam dunia nyata :

1. Aplikasi antrian di jalan Tol.

2. Aplikasi antrian saat mengantri di loket.

3. Aplikasi antrian reservasi tiket kereta api, dan lain-lain.

Semua itu menggunakan aturan FIFO (First In, First Out).