

**LAPORAN RENCANA PRAKTIKUM
STRUKTUR DATA**



NAMA : INDRA FIQI RIPANI
NIM : 213010503002
KELAS : F
MODUL : 2 (QUEUE DAN STACK)

Program Studi S1 Teknik Informatika
Fakultas Teknik
Universitas Palangka Raya
Palangka Raya, Kalimantan Tengah
2022

MODUL 2

QUEUE DAN STACK

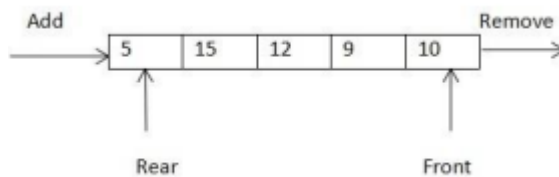
I. TUJUAN PRAKTIKUM

1. Mahasiswa memahami konsep dari Queue dan Stack.
2. Mahasiswa mampu mengimplementasikan Queue dan Stack.

II. Landasan Teori

1. Queue

Queue (antrian) dapat diartikan sebagai suatu kumpulan data yang seolah-olah terlihat seperti ada data yang diletakkan di sebelah data yang lain seperti pada gambar 1, dimana data masuk melalui lorong di sebelah kanan dan masuk dari terowongan sebelah kiri. Hal ini membuat antrian bersifat FIFO (*First In First Out*), beda dengan stack yang berciri LIFO.



Gambar 1.1 Ilustrasi Queue.

Contoh yang paling populer untuk membayangkan sebuah queue adalah antrian pada kasir sebuah bank. Ketika seorang pelanggan datang, akan menuju ke belakang dari antrian. Setelah pelanggan dilayani, antrian yang berada di depan akan maju. Pada saat menempatkan elemen pada ujung (tail) dari queue disebut dengan enqueue, pada saat memindahkan elemen dari kepala (head) sebuah queue disebut dengan dequeue. Pada gambar 2 diperlihatkan sebuah queue serta proses enqueue dan dequeue.

Karakteristik penting dari antrian adalah:

- a. Elemen, antrian yaitu item-item data yang terdapat didalam antrian.
- b. Front (elemen terdepan dari antrian).
- c. Rear (elemen terakhir dari antrian).

d. Count (jumlah elemen pada antrian).

Operasi yang dapat dilakukan pada queue pada umumnya terdapat 6 operasi yaitu create, initialization, isEmpty, isFull, clear, enqueue dan juga dequeue. Namun hanya tiga operasi utama saja pada queue yang akan dijelaskan, yaitu:

a. Create

Untuk menciptakan queue menggunakan perintah struct dan menginisialisasi Queue dengan cara membuat Head dan Tail = -1. Beberapa orang memulai index queue dari -1 dan ada juga yang dari 0.

```
#define MAX 8
Struct Queue{
    Int elemen[MAX]
    Int front,rear,count;
```

b. Enqueue

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

```
void Enqueue(int data, Queue *Q){
    Q->elemen[Q->rear] = data;
    Q->rear++;
    Q->count++;
}
```

c. Dequeue

Digunakan untuk menghapus elemen terdepan/pertama (head) dari antrian dengan cara menggeser semua elemen antrian ke depan dan mengurangi tail dengan satu penggeseran yang dilakukan dengan menggunakan looping.

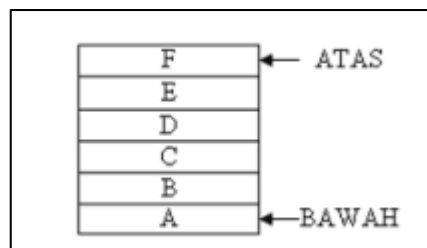
```
void Dequeue(Queue *Q){
    int keeper = Q->elemen[Q->front];
```

```
for(int i = Q->front; i<Q->rear; i++){  
    Q->elemen[i]=Q->elemen[i+1];  
}  
Q->rear--;  
Q->count--;
```

2. Stack

Pengertian Stack pada Struktur Data adalah sebagai tumpukan dari benda, sekumpulan data yang seolah-olah diletakkan di atas data yang lain, koleksi dari objek-objek homogen, atau suatu urutan elemen yang mana elemennya dapat diambil dan ditambah hanya pada posisi akhir (top) saja.

Stack pada Struktur Data dapat diilustrasikan dengan dua buah kotak yang ditumpuk, kotak yang satu akan ditumpuk diatas kotak yang lainnya. Jika kemudian stack 2 kotak tadi, ditambah kotak ketiga, keempat, kelima, dan seterusnya, maka akan diperoleh sebuah stack kotak yang terdiri dari N kotak.



Gambar 2.2 Ilustrasi Stack.

Stack pada Struktur Data dapat diilustrasikan dengan dua buah kotak yang ditumpuk, kotak yang satu akan ditumpuk diatas kotak yang lainnya. Jika kemudian stack 2 kotak tadi, ditambah kotak ketiga, keempat, kelima, dan seterusnya, maka akan diperoleh sebuah stack kotak yang terdiri dari N kotak.

Stack bersifat LIFO (Last In First Out) artinya Benda yang terakhir masuk ke dalam stack akan menjadi yang pertama keluar dari stack. Operasi-operasi yang biasanya terdapat pada Stack mirip dengan queue. Adapun 2 operasi utama pada stack yaitu:

- a. Push: digunakan untuk menambah item pada stack pada tumpukan paling atas.

```
void Push(int data, Queue *Q){  
    Q->elemen[Q->top] = data;  
    Q->top++;  
}
```

- b. Pop: digunakan untuk mengambil item pada stack pada tumpukan paling atas.

```
void pop(Queue *Q){  
    int keeper=Q->elemen[Q->top];  
    Q->top--;  
}
```

Cara mendefenisikan Stack dengan Array of Struct yaitu dengan menggunakan struct, stack hanya memiliki karakteristik berupa top dan bottom sebagai penanda awal dan akhir serta elemennya itu sendiri. Definisikan konstanta MAX_STACK untuk menyimpan maksimum isi stack kemudian buatlah variabel array data sebagai implementasi stack.

```
#define MAX_STACK 10  
struct Stack{  
    int elemen_array[MAX_STACK];  
    int top;  
    int bottom;  
};
```

III. TUGAS

Terdapat sebuah struct data dari mahasiswa, seperti nama, NIM, dsb. Buatlah sebuah program yang mampu melakukan Dequeue (FIFO) dan Pop (LIFO) pada struct data mahasiswa tersebut. Berikut adalah operasi/fungsi yang akan dibuat:

1. Create, sebuah procedure untuk membuat struct/record mahasiswa serta menginisialisasikan nilai awalnya.
2. Input, procedure untuk memasukan data mahasiswa.
3. Dequeue/FIFO, untuk melakukan pengambilan data secara FIFO.

4. Pop/LIFO, untuk melakukan pengambilan data secara LIFO.
5. Clear, untuk menghapus semua data didalam struct/record mahasiswa.
6. Print, mencetak seluruh nama mahasiswa.