

LAPORAN PRAKTIKUM
Algoritma dan Struktur Data
Week 3 : QUEUE



MUHAMMAD FARIS ISA
D4LJ – Teknik Informatika
3122640005

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2022

1. Latihan Program Parkir

Membuat sebuah program parkir.

- Source code :

```
#include <stdio.h>
#include <math.h>
#define MAX 100

typedef int Itemtype;
typedef struct{
    Itemtype item[MAX];
    int count;
    int front;
    int rear;
} Queue;
typedef struct{
    Itemtype item[MAX];
    int count;
} Stack;

void Inisialisasi_q(Queue *q){
    q->count = q->front = q->rear = 0;
}

int Penuh_q(Queue *q){
    return q->count == MAX;
}

int Kosong_q(Queue *q){
    return q->count == 0;
}

void Inisialisasi_s(Stack *s){
    s->count=0;
}

int Penuh_s(Stack *s){
    return s->count==MAX;
}

int Kosong_s(Stack *s){
    return s->count==0;
}
```

```

int Penuh_s(Stack *s){
    return s->count==MAX;
}

int Kosong_s(Stack *s){
    return s->count==0;
}

void Push(Stack *s, Itemtype x){
    if(Penuh_s(s)) printf("Stack penuh!\n");
    else{
        s->item[s->count]=x;
        s->count++;
    }
}

Itemtype Pop(Stack *s){
    Itemtype temp=0;
    if(Kosong_s(s)) printf("Stack kosong!\n");
    else{
        s->count--;
        temp=s->item[s->count];
    }
    return temp;
}

void Enqueue(Queue *q, Itemtype x){
    if(Penuh_q(q)) printf("Queue Penuh ! \n");
    else {
        q->item[q->rear] = x;
        q->rear = (q->rear+1)%MAX;
        q->count++;
    }
}

Itemtype Dequeue(Queue *q){
    Itemtype temp = -1;
    if(Kosong_q(q)) printf("Queue Kosong ! \n");
    else {
        temp = q->item[q->front];
        q->front = (q->front+1)%MAX;
        q->count--;
    }
    return temp;
}

```

```

void cetak_parkir(Queue q){
    int i = q.front;
    printf("Urutan Parkir : ");
    while( i != q.rear ){
        printf("%d ", q.item[i]);
        i = ( i + 1 )%MAX;
    }
    printf("\n");
}

main(){
    Queue ant;
    Stack tump;
    int no_parkir = 1, no_keluar, pilih;
    char lagi = 'y';
    Inisialisasi_q(&ant);
    Inisialisasi_s(&tump);
    while( lagi == 'y' ){
        printf("----- PROGRAM PARKIR ----- \n");
        printf("1. Masuk \n");
        printf("2. Keluar \n");
        printf("Pilih Menu : ");scanf("%d", &pilih);
        switch (pilih){
            case 1 :   printf("Nomor parkir yang masuk : %d \n",
no_parkir);
                    Enqueue(&ant, no_parkir);
                    no_parkir++;
                    break;
            case 2 :   printf("Nomor parkir yang keluar : ");scanf("%d",
&no_keluar);
                    if ( ant.item[ant.front] == no_keluar )
                        printf("Parkir yang keluar %d\n",
Dequeue(&ant));
                    else {
                        while (ant.item[ant.front] != no_keluar ){
                            Push(&tump, Dequeue(&ant));
                        }
                        no_keluar = Dequeue(&ant);
                        while (!Kosong_s(&tump)){
                            ant.count++;
                            ant.item[(ant.front-1)%MAX] = Pop(&tump);
                            ant.front = (ant.front - 1 )%MAX;
                        }
                    }
                    break;
        }
        cetak_parkir(ant);
    }
}

```

```
        printf("Apakah ingin kembali ke menu ?  
");fflush(stdin);scanf("%c", &lagi);  
  
    }  
}
```

- Hasil :

```
Latihan3.exe"  
----- PROGRAM PARKIR -----  
1. Masuk  
2. Keluar  
Pilih Menu : 1  
Nomor parkir yang masuk : 1  
Urutan Parkir : 1  
Apakah ingin kembali ke menu ? y  
----- PROGRAM PARKIR -----  
1. Masuk  
2. Keluar  
Pilih Menu : 1  
Nomor parkir yang masuk : 2  
Urutan Parkir : 1 2  
Apakah ingin kembali ke menu ? y
```

```
----- PROGRAM PARKIR -----  
1. Masuk  
2. Keluar  
Pilih Menu : 2  
Nomor parkir yang keluar : 1  
Parkir yang keluar 1  
Urutan Parkir : 2 3 4 5 6
```

```
----- PROGRAM PARKIR -----  
1. Masuk  
2. Keluar  
Pilih Menu : 2  
Nomor parkir yang keluar : 4  
Urutan Parkir : 2 3 5 6
```

- Analisa :

Saat program dijalankan, penulis diminta untuk melakukan input menu. Pengguna dapat memilih diantara kedua menu yang disediakan yaitu ada menu masuk dan menu keluar. Jika penulis memilih menu masuk maka akan diinputkan sebuah urutan nomor parkir dengan auto increment. Jadi setiap memilih menu masuk maka akan bertambah satu seperti contoh hasil program diatas.

Setelah itu penulis memilih menu keluar. Pada menu keluar, jika nomor input yang diinputkan saat memilih nomor keluar berada di paling awal masuk maka nomor tersebut akan langsung dicetak. Sedangkan apabila nomor yang akan dikeluarkan berada ditengah-tengah maka nomor yang dicari ini ditelusuri dulu melalui Queue. Jika nomor belum ditemukan saat menelusuri melalui Queue, maka nilai akan di push ke dalam stack. Jika nilai sudah ditemukan, maka nilai akan di Dequeue dan setelah itu nilai yang berada di stack akan di pop hingga stack kosong.

KESIMPULAN

1. Tidak seperti Stack, Queue menggunakan FIFO(First In First Out) Order.
2. Queue dapat handle penyisipan dari depan dan/atau belakang antrian.