

## Modul Praktikum 1 : Stack dan Queue menggunakan Sturktur Data Sederhana

### Kompetensi:

Mahasiswa mampu mengimplementasikan Stack dan Queue menggunakan Sturktur Data Sederhana.

### Tujuan Instruksional Khusus:

1. Mahasiswa mampu mengimplementasikan Stack menggunakan sturktur data Array.
2. Mahasiswa mampu mengimplementasikan Queue menggunakan sturktur data Array.

### Praktikum:

1. Peralatan
  - a. Perangkat komputer
  - b. Sistem Operasi Windows
  - c. C++ Compiler (MinGW)
  - d. Text Editor (Geany, Notepad++, Visual Studio Code, SublimeText, dll)
2. Prosedur
  - a. Baca dan pahami semua tahapan praktikum dengan cermat.
  - b. Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
  - c. Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
  - d. Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
  - e. Pastikan diri anda tidak menyentuh sumber listrik.

3. Teori

- a. Stack

Struktur Stack :

```
struct Stack{
    int elemen[MAX_STACK];
    int top;
};
```

Pembuatan Stack baru:

```
void createStack(Stack &stack){
    stack.top = -1;
}
```

Pengecekan jika stack kosong:

```
bool isEmpty(Stack stack){
    return stack.top == -1;
}
```

Pengecekan jika stack penuh:

```
bool isFull(Stack stack){
    return stack.top == MAX_STACK-1;
}
```

Proses Push:

```
void push(Stack &stack, int i){
    stack.top++;
    stack.elemen[stack.top] = i;
}
```

Proses pop:

```
void pop(Stack &stack, int &i){
    i = stack.elemen[stack.top];
    stack.top--;
}
```

#### b. Queue

Struktur Queue :

```
struct Queue{
    int elemen[1000];
    int rear;
    int front;
    int jml;
};
```

Pembuatan Queue baru:

```
void createQ(Queue &q){
    q.front = 0;
    q.rear = -1;
    q.jml = 0;
}
```

Pengecekan jika queue penuh:

```
bool isFull(Queue q){
    return q.jml == MAX_QUEUE;
}
```

Pengecekan jika queue kosong:

```
bool isEmpty(Queue q){
    return q.jml == 0;
}
```

Proses enqueue:

```
void enQ(Queue &q, int i){
    if(isFull(q)) return;

    q.rear++;
    q.jml++;
    q.elemen[q.rear] = i;
}
```

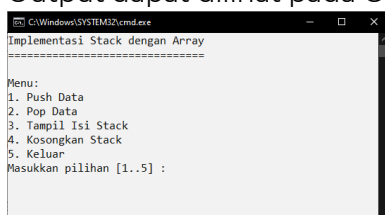
Proses dequeue:

```
void deQ(Queue &q, int &i){
    if(isEmpty(q)) return;

    i = q.elemen[q.front];
    q.front++;
    q.jml--;
}
```

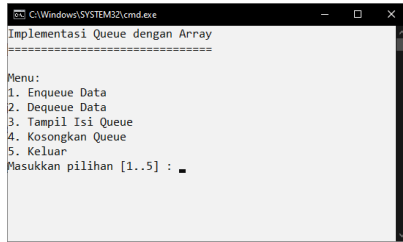
#### 4. Kegiatan Praktikum:

- Buat program untuk implementasi stack menggunakan struktur data array dengan kapasitas  $n=5$ . Output dapat dilihat pada Gambar di bawah. Demo program dapat diakses di sini.



Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.

- b. Buat program untuk implementasi queue menggunakan struktur data array dengan kapasitas  $n=5$ . Output dapat dilihat pada Gambar di bawah.



```
C:\Windows\SYSTEM32\cmd.exe
Implementasi Queue dengan Array
=====
Menu:
1. Enqueue Data
2. Dequeue Data
3. Tampil Isi Queue
4. Kosongkan Queue
5. Keluar
Masukkan pilihan [1..5] : █
```

Untuk lebih jelasnya, silakan buka demo program yg telah disediakan.