PERTEMUAN 6

STRUKTUR DATA

QUEUE

Disusun oleh:

Fauzi Ikhsan Fajar Muzaqi 2022320018

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS INFORMATIKA

UNIVERSITAS BINA INSANI

BEKASI

2022

1. PENJELASAN SYNTAX

A. HEADER

```
strukturdata_p6 > C++ queue.cpp > main()
1  #include <iostream>
2  #include <conio.h>
3  #include <stdlib.h>
4  #define MAX 75
5
6  using namespace std;
7
```

Seperti biasa, kita import bagian bagian yang ingin kita pakai, seperti : *<iostream>* , *<conio.h>*, *<stdlib.h>*. ini adalah library yang akan kita gunakan untuk membuat program queue.

Dalam **Queue,** ada yang Namanya head dan tail, kita definisikan dengan tipe data integer. Variable num yang tipe datanya integer diambil dari MAX yang telah di definisikan dengan nilai 75 diatas.

```
int num[MAX];
int head = -1;
int tail = -1;
```

B. FUNCTION

Dalam membuat Program kali ini, kita akan belajar membuat beberapa function yang nantinya akan digunakan, beberapa function itu seperti ***isFull()***, ***isEmpty()***, ***Entry()***, dan function lainnya, langsung saja kita buat sebagai berikut :

```
bool IsEmpty()
{
    if (tail == -1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

buat fungsi *isEmpty* dengan tipe data Boolean yang akan mengembalikan nilai true atau false, maka kita gunakan if dan else sebagai isinya, di *isEmpty* jika tail sama dengan -1, maka akan mengembalikan nilai true, dan jika tidak akan mengembalikan nilai false.

```
bool IsFull()
{
    if (tail == MAX - 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

lalu kita buat fungsi isFull dengan tipe data yang sama, Boolean, denga nisi jika tail (panggil nilai MAX) -1, maka akan mengembalikan nilai true, jika sebaliknya akan mengembalikan nilai false. Kita akan banyak menggunakan fungsi if else pada program kali ini.

```
void Entry(int no)
{
    if (IsEmpty())
    {
        head = tail = 0;
    }
    else
    {
        tail++;
    }
    num[tail] = no;
}
```

Ada fungsi void *Entry*, maksud dari void adalah fungsi tidak akan minta return di akhir. *Entry* mempunyai isi sebagai berikut: menggunakan if else untuk membuat percabangan, dengan memanggil fungsi *isEmpty* yang telah dibuat sebelumnya, kita panggil fungsi head, cara bacanya yaitu, jika kosong, atau *isEmpty* maka akan mengembalikan nilai head dan tail berupa 0; namun jika data nanti ada isinya, tail akan ditambah, dibawah dimasukkan variable num yang diambil dari tail.

```
void Exit()
{
    if (IsEmpty())
    {
        cout << "Antrian Kosong ! ";
        getch();
    }
    else
    {
        for (int a = head; a < tail; a++)
        {
            num[a] = num[a + 1];
        }
        tail--;
        if (tail == -1)
        {
            head = -1;
        }
    }
}
```

Membuat fungsi *Exit*, cara membaca fungsi disamping yaitu, jika kondisi nya *isEmpty* (panggil fungsi isEmpty), maka akan mengembalikan print berupa "Antrian Kosong!" dan ada perintah ***getch()***, apa itu? **getch**, digunakan untuk mendapatkan 1 (satu) karakter dari user. Ingat! Hanya 1 (satu) karakter. Dan, karena karakteristiknya tersebut, seringkali perintah getch() hanya digunakan untuk melakukan "penundaan" sebelum perintah berikutnya dieksekusi.

Berlanjut di else {}, kita akan mengulang nilai dari head dan tail, kita akan buat variable dengan tipe data int untuk menampung head, cara membacanya adalah : Nilai awal dari a adalah nilai yang didapat dari variable head;, maka jika a (head) kurang dari tail; maka head akan ditambah (increament). Setelah diulang, action yang akan dilakukan adalah variable num yang didapat dari a (head) akan ditambah 1. Maka tail akan di kurang (decreament). Kemudian panggil fungsi if lagi untuk mengecek tail yang telah dikurang, jika tail sama dengan -1, maka head akan dikurang 1. Selanjutnya :

```
void Clear()
{
    head = tail = -1;
}

void View()
{
    if (IsEmpty())
    {
        cout << "Antrian Kosong ! ";
    }
    else
    {
        system("cls");
        for (int a = head; a <= tail; a++)
        {
            cout << "============================="
                 << "\n >> Nomor Antrian : [" << num[a] << "]"
                 << "\n=============================" << endl;
        }
    }
}
```

Fungsi *Clear* cukup mudah, dia akan menghapus head dan tail dengan cara dikurang 1.

Lalu fungsi *View* akan menampilkan jika *isEmpty()*, maka akan menampilkan “Antrian Kosong” pada console. Jika tidak, program akan dihentikan dan menjalankan perulangan yang sama seperti diatas, lalu akan menampilkan nomer antrian sesuai dari nilai [a].

C. BODY

```
int main()
{
    int pilih, p = 1, urut;
    do
    {
        system("cls");
        cout << "\n\n========== ANTRIAN PAJAK =========\n"
             << "\n================================"
             << "\n| 1. Tambah Antrian            |"
             << "\n| 2. Panggil Antrian           |"
             << "\n| 3. Daftar Antrian            |"
             << "\n| 4. Bersihkan Antrian         |"
             << "\n| 5. Keluar                    |"
             << "\n================================";
        cout << "\nPilih : ";
        cin >> pilih;
        cout << "\n\n";
```

Di sini kita akan melakukan perualang do while, dan pada bagian ini kita akan menampilkan halaman utama dari program.

```
if (pilih == 1)
{
    if (IsFull())
    {
        cout << "Antrian sudah penuh, mohon tunggu beberapa saat lagi ";
    }
    else
    {
        urut = p;
        Entry(urut);
        cout << "--------------------------------" << endl;
        cout << "|         NOMOR ANTRIAN        |" << endl;
        cout << "|               " << p << "              |" << endl;
        cout << "--------------------------------" << endl;
        cout << "|       Silahkan Mengantri     |" << endl;
        cout << "|       Menunggu " << tail << " Antrian      |" << endl;
        cout << "--------------------------------" << endl;
        p++;
    }
}
```

Selanjutnya adalah menu 1, disini kita menggunakan if else, jika queue penuh maka akan menampilkan cout pada bagian if. Apabila queue belum penuh, maka akan memanggil function entry dan menampilkan cout pada bagian else yang di dalamnya memanggil variabel p dan tail.

```
else if (pilih == 2)
{
    cout << "================================" << endl;
    cout << " Nomor Antrian : [" << num[head] << "]";
    cout << "\n================================" << endl;
    Exit();
}
```

Selanjutnya adalah menu 2, apabila kita menginput menu 2, maka akan menampilkan cout seperti di atas, dan akan menampilkan nomor antrian num[head] berdasarkan function exit.

Selanjutnya adalah menu 3, apabila kita menginput menu 3, maka akan menampilkan function view.

```
else if (pilih == 3)
{
    View();
}
```

```
else if (pilih == 4)
{
    if (IsEmpty())
    {
        cout << "Antrian Kosong ! ";
    }
    else
    {
        Clear();
        cout << "Antrian telah dibersihkan ! ";
    }
}
```

Selanjutnya adalah menu 4, di sini kita menggunakan if else, jika queue kosong, maka akan menampilkan cout di bagian if. Apabila queue masih terisi, maka akan memanggil function clear untuk membersihkan queue.

Selanjutnya adalah menu 5, apabila kita menginput menu 5 maka program akan ditutup. Apabila kita menginput selain menu 1 -5, maka di bagian ini akan memproses dan akan menampilkan cout seperti di gambar.

```
        else if (pilih == 5)
        {
        }
        else
        {
            cout << "Masukan anda salah ! \n"
                 << endl;
        }
        getch();
    } while (pilih != 5);
}
```

2. OUTPUT

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih :
```

Halaman Utama

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 1


--------------------------------
|         NOMOR ANTRIAN        |
|              1               |
--------------------------------
|       Silahkan Mengantri     |
|       Menunggu 0 Antrian     |
--------------------------------
```

Menu 1

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 2


================================
 Nomor Antrian : [1]
================================
```

Menu 2

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 2


================================
 Nomor Antrian : [1878372360]
================================
Antrian Kosong !
```

Menu 2 jika queue kosong

```
================================
 >> Nomor Antrian : [1]
================================
================================
 >> Nomor Antrian : [2]
================================
================================
 >> Nomor Antrian : [3]
================================
```

Menu 3

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 3


Antrian Kosong !
```

Menu 3 jika queue kosong

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 4


Antrian telah dibersihkan !
```

Menu 4

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 4


Antrian Kosong !
```

Menu 4 jika queue kosong

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 6


Masukan anda salah !
```

Jika menginput menu yang tidak ada

```
========= ANTRIAN PAJAK =========

================================
| 1. Tambah Antrian            |
| 2. Panggil Antrian           |
| 3. Daftar Antrian            |
| 4. Bersihkan Antrian         |
| 5. Keluar                    |
================================
Pilih : 5
```

Menu 5 Exit