**MODUL ENAM MINGGU KE TUJUH**

**TUMPUKAN DAN ANTRIAN**

Kompetensi mahasiswa : Mampu memahami, menerapkan dan mengendalikan penggunaan struktur antrian (queue) dan tumpukan dalam suatu kasus yang lebih kompleks.

Dalam kasus yang lazim ditangani, maka ada kalanya kita harus menyelesaikan kasus antrian dan tumpukan yang tergabung menjadi satu. Missal dalam pengeloaan data pendaftar mahasiswa baru, maka pendaftar akan antre untuk membeli formulir atau mengisi formulir, kemudian formulir yang sudah diserahkan akan disimpan dalam tumpukan. Perhatikan contoh program berikut yang mengelola hal tersebut.

#include <iostream>

using namespace std;

#define max 10

// membuat record beli

struct beli{

int nomor[max];

string nama[max];

int harga[max];

};

// membuat record stack

struct stack{

int no[max];

string nama[max];

int tgl[max];

int tos; //titik teratas

};

// membuat record antrian

struct antrian{

int nomor[max];

int front;

int rear;

};

struct stack s;

struct beli b;

struct antrian a;

int n = 0,nn = 0;

// function untuk mengecek apakah stack penuh ?

bool isFull(){

if(s.tos == max-1) return true;

else return false;

}

// function untuk mengecek apakah stack kosong ?

bool isEmpty(){

if(s.tos == -1) return true;

else return false;

}

// function untuk mengecek apakah antrian penuh ?

bool penuh(){

if(a.rear == max-1) return true;

else return false;

}

// function untuk mengecek apakah antrian kosong ?

bool kosong(){

if(a.rear == -1) return true;

else return false;

}

// function untuk beli formulir

void beliFormulir(){

int nf;

ulang:

cout<<"masukkan nomor formulir yang akan dibeli : ";

cin>>nf;

// verifikasi nomor formulir apakah ada di database

for(int i = 0;i<=n;i++){

if(b.nomor[i] == nf){

cout<<"anda telah membeli formulir nomor "<<nf<<endl;

goto ulang;

}

}

// jika verifikasi berhasil

b.nomor[n] = nf;

cout<<"masukkan nama pembeli :";

cin>>b.nama[n];

b.harga[n] = 75;

n++;

cin.get();cin.ignore();

}

// function untuk antrian

void pushAntrian(){

nn++;

a.rear++;

a.nomor[a.rear] = nn;

cout<<"anda mendapatkan nomor antrian "<<a.nomor[a.rear]<<endl;

cin.ignore();cin.get();

}

// function untuk layanan antrian

void layanan\_antrian(){

bool ketemu,ada;int pos;

cout<<"melayani pengembalian formulir"<<endl;

cout<<"----------------------------------------------------"<<endl;

// antrian awal di layani

int noambil = a.nomor[a.front];

for(int i = 0;i<a.rear;i++){

a.nomor[i] = a.nomor[i+1];

}

a.rear--;

cout<<"sekarang melayani pengembalian formulir nomor"<<noambil<<endl;

cout<<"masukkan nomor formulir yang akan dikembalikan : ";

int nf;

cin>>nf;

// di cek apakah nomor formulir sama dengan yg ada di database?

for(int i = 0;i<=n;i++){

if(b.nomor[i] == nf){

ketemu = true;

pos = i;

}

}

// jika ada

if (ketemu){

ada = false;

// cek di stack apakah ada nomor formulir yang sama

for(int i = 0;i<=s.tos;i++){

if(s.no[i] == nf){

ada = true;

}

}

if(ada) cout<<"formulir sudah pernah dikembalikan"<<endl;

else {

s.no[s.tos] = nf;

s.nama[s.tos] = b.nama[pos];

cout<<"masukkan tanggal ";

cin>>s.tgl[s.tos];

s.tos++;

}

}else cout<<"nomor formulir tidak ditemukan"<<endl;

cin.ignore();cin.get();

}

// function untuk mencetak daftar formulir yang sudah dibeli

void cetakForm(){

cout<<"daftar formulir yang telah dibeli"<<endl;

cout<<"-----------------------------------------------"<<endl;

cout<<"nomor nama harga"<<endl;

cout<<"-----------------------------------------------"<<endl;

for(int i = 0;i<n;i++){

cout<<b.nomor[i]<<" "<<b.nama[i]<<" "<<b.harga[i]<<endl;

}

cout<<"-----------------------------------------------"<<endl;

cout<<"anda telah membeli "<<n<<" formulir"<<endl;

cin.ignore();cin.get();

}

// function untuk mencetak daftar antrian

void cetakAntrian(){

cout<<" Daftar antrian"<<endl;

cout<<"-----------------------------------------------"<<endl;

cout<<"Antrian\_ke nomor\_urut\_antrian"<<endl;

cout<<"-----------------------------------------------"<<endl;

for(int i=0;i<=a.rear;i++){

cout<<i+1<<" "<<a.nomor[i]<<endl;

}

cout<<"-----------------------------------------------"<<endl;

cout<<"Jumlah yang sedang antri saat ini : "<<a.rear<<endl;

cin.ignore();cin.get();

}

// function untuk mencetak daftar antrian yang yang sudah dilayani/formulir sudah dikembalikan

void cetak\_stack(){

cout<<" Daftar Formulir yang sudah dikembalikan"<<endl;

cout<<"-----------------------------------------------"<<endl;

cout<<"no nomor\_form nama\_pembeli tanggal\_kembali"<<endl;

cout<<"-----------------------------------------------"<<endl;

for(int i = s.tos-1 ;i >=0;i--){

cout<<i+1<<" "<<s.no[i]<<" "<<s.nama[i]<<" "<<s.tgl[i]<<endl;

}

}

int main(){

int pil;

a.front = 0;

a.rear = -1;

do

{

system("clear");

cout<<"LAYANAN PEMBELIAN DAN PENGEMBALIAN FORMULIR MABA"<<endl;

cout<<"=================================================="<<endl;

cout<<"1. Beli Formulir"<<endl;

cout<<"2. Antri penyerahan"<<endl;

cout<<"3. Layanan Penyerahan"<<endl;

cout<<"4. Cetak daftar pembelian formulir"<<endl;

cout<<"5. Cetak antrian yang akan menyerahkan formulir"<<endl;

cout<<"6. Cetak tumpukan formulir yang sudah diserahkan"<<endl;

cout<<"0. Keluar"<<endl;

cout<<"=================================================="<<endl;

cout<<"Masukkan pilihan anda : ";

cin>>pil;

// switch case untuk memilih menu

switch (pil)

{

case 1:

beliFormulir();

break;

case 2:

if (penuh()){

cout<<"Antrian sudah penuh"<<endl;

cin.get();cin.ignore();

} else pushAntrian();

break;

case 3:

if (kosong()){

cout<<"Antrian sudah kosong"<<endl;

cin.get();cin.ignore();

}

else layanan\_antrian();

cin.get();cin.ignore();

break;

case 4:

if (n == 0){

cout<<"anda belum membeli formulir"<<endl;

cin.get();cin.ignore();

}else cetakForm();

cin.get();cin.ignore();

break;

case 5:

if (kosong()){

cout<<"Antrian kosong"<<endl;

cin.get();cin.ignore();

}else cetakAntrian();

break;

case 6:

if(s.tos == -1){

cout<<"Belum ada yang menyerahkan formulir"<<endl;

cin.get();cin.ignore();

}else cetak\_stack();

cin.get();cin.ignore();

break;

default:

cout<<"anda salah pilih menu"<<endl;

cin.get();cin.ignore();

break;

}

} while (pil != 0);

}

**latihan :**

Buatlah program untuk mengelola Dealer suatu kendaraan dengan ketentuan sbb:

1. Data kendaraan disimpan dengan atribut no\_plat (unik),nama kendaraan dan harga
2. Data pembelian disimpan dengan atribut nama pembeli, no\_plat kendaraan yang dibeli dan tanggal pembelian

Menu yang diinginkan adalah input data kendaraan, cetak kendaraan, antrean pembelian kendaraan, layanan pembelian (dari kendaraan yang telah diinputkan dan pembeli diambil dari antrian) dan cetak penjualan kendaraan

diupload di google classroom dgn kode : ujfxdvm

1. Dikirim dalam bentuk pdf dengan nama file latihan\_NIM\_Nama
2. code + output