Nama : Helmi Efendi Lubis
NIM : 1301223338
Kelas : IF-46-08

Tugas Pendahuluan Modul 10

Queue.h

```h
#ifndef QUEUE_H_INCLUDED
#define QUEUE_H_INCLUDED

#include <iostream>
using namespace std;

#define Nil NULL
#define info(P) (P)->info
#define next(P) (P)->next
#define head(Q) ((Q).head)
#define tail(Q) ((Q).tail)

typedef bool boolean;
typedef int infotype;
typedef struct elmQ *address;

struct elmQ{
    infotype info;
    address next;
};

struct queue{
    address head, tail;
};

address alokasi_1301223338(infotype info);
address findElmt_1301223338(queue Q, infotype num);
bool queueEmpty_1301223338(queue Q);
void createQueue_1301223338(queue &Q);
void dealokasi_1301223338(address P);
void enQueue_1301223338(queue &Q, address P);
void deQueue_1301223338(queue &Q, address P);
void printInfo_1301223338(queue Q);
int nbOfElm_1301223338(queue Q);
void ganjilGenap_1301223338(queue &Q, queue &QGanjil, queue &QGenap);

#endif // QUEUE_H_INCLUDED
```

Queue.cpp

```cpp
#include "queue.h"

address alokasi_1301223338(infotype info){
    address p = new elmQ;
    info(p) = info;
    next(p) = Nil;
    return p;
}

address findElmt_1301223338(queue Q, infotype num){
    address p;
    boolean found;

    p = head(Q);
    found = false;

    while(p != Nil && found == false){
        if(info(p) == num){
            found = true;
        }else{
            p = next(p);
        }
    }
    return p;
}

bool queueEmpty_1301223338(queue Q){
    return head(Q) == Nil;
}

void createQueue_1301223338(queue &Q){
    head(Q) = Nil;
}

void dealokasi_1301223338(address P){
    delete P;
}

void enQueue_1301223338(queue &Q, address P){
    if(queueEmpty_1301223338(Q)){
        head(Q) = P;
        tail(Q) = P;
    }else {
        next(tail(Q))= P;
        tail(Q) = P;
```

```
    }
}

void deQueue_1301223338(queue &Q, address P){
    if(queueEmpty_1301223338(Q)){
        cout << "Queue kosong" << endl;
    }else if(next(head(Q)) == Nil){
        P = head(Q);
        head(Q) = Nil;
        dealokasi_1301223338(P);
    }else {
        P = head(Q);
        head(Q) = next(P);
        next(P) = Nil;
    }
}

void printInfo_1301223338(queue Q){
    address P;
    int i = 1;

    P = head(Q);

    if(P == Nil){
        cout << "Queue kosong" << endl;
    }else{
        while(P != Nil){
            cout << "Antrian ke-" << i << ":" << info(P) << endl;
            P = next(P);
            i++;
        }
    }
}

int nbOfElm_1301223338(queue Q){
    int numOfElmt = 0;

    address P;
    P = head(Q);

    while(P != Nil){
        numOfElmt++;
        P = next(P);
    }
    return numOfElmt;
}

void ganjilGenap_1301223338(queue &Q, queue &QGanjil, queue &QGenap){
```

```
    address p;
    while (!queueEmpty_1301223338(Q)){
        p = head(Q);
        if (info(p) % 2 == 0){
            deQueue_1301223338(Q, p);
            enQueue_1301223338(QGenap, p);
        }else
        {
            deQueue_1301223338(Q, p);
            enQueue_1301223338(QGanjil, p);
        }
    }
}
```

Main.cpp

```
#include "queue.h"
#include "queue.cpp"
#include <iostream>

using namespace std;

int main()
{
    queue Q, Qodd, Qeven;
    createQueue_1301223338(Q);

    infotype uInp;
    address p;

    int i = 1;
    while(i <= 10){
        cout << "Input ke-" << i << ":";
        cin >> uInp;
        p = alokasi_1301223338(uInp);
        enQueue_1301223338(Q, p);
        i++;
    }
    cout << endl;
    cout << "Kondisi awal" << endl;
    printInfo_1301223338(Q);

    cout << endl;
    ganjilGenap_1301223338(Q, Qodd, Qeven);
    cout << "Kondisi akhir" << endl;
    cout << "Queue Utama" << endl;
```

```
    printInfo_1301223338(Q);

    cout << endl;
    cout << "Queue Ganjil" << endl;
    printInfo_1301223338(Qodd);

    cout << endl;
    cout << "Queue Genap" << endl;
    printInfo_1301223338(Qeven);
}
```

Output