TUGAS PENGGANTI PRAKTIKUM STRUKTUR DATA

Nama : Patricia Joanne

NPM : 140810160065

1. Jelaskan perbedaan antara Stack dan Queue, kemudian beri contoh impelemtasinya pada kehidupan sehari-hari!

   Jawab:

   Stack pada Struktur Data adalah sekumpulan data yang seolah-olah diletakkan di atas data yang lain atau suatu urutan elemen yang elemennya dapat diambil dan ditambah hanya pada posisi akhir (top) saja. Sedangkan Queue pada Struktur Data atau antrean adalah sekumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung disebut dengan sisi belakang (rear), dan penghapusan (pengambilan elemen) dilakukan lewat ujung lain (disebut dengan sisi depan atau front).

   Pada stack atau tumpukan menggunakan prinsip "Masuk terakhir keluar pertama" atau LIFO (Last In First Out), sedangkan pada queue atau antrean prinsip yang digunakan adalah "Masuk Pertama Keluar Pertama" atau FIFO (First In First Out).

   Stack dan queue banyak kita jumpai dalam kehidupan sehari-hari, untuk stack misalnya setumpuk buku, dimana buku yang paling terakhir ditaruh di atas tumpukan yang dapat dilihat. Sementara untuk queue, pemakaian sistem komputer berbagi waktu (time-sharing computer system) dimana ada sejumlah pemakai yang akan menggunakan sistem tersebut secara serempak.

2. Jelaskan perbedaan Stack dan Queue yang menggunakan array dan list!

   Jawab:

   1) Stack array

      - Pendeklarasian

```
#include<iostream>
using namespace std;

const int maxElemen = 255;
```

```
struct stack{
    int isi[maxElemen];
    int top;
};
stack s;

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

- Fungsi void push & pop

```
void push(stack& T, char X){
    if (T.atas == maxElemen-1){
        cout << "TUMPUKAN SUDAH PENUH" << endl;
    }
    else {
        T.Atas = T.Atas+1;
        T.Isi[T.Atas] = X;
    }
}
```

```
void pop(stack& T){
    if (T.Atas == 0){
        cout << "TUMPUKAN SUDAH KOSONG" << endl;
    }
    else {
        T.Atas = T.Atas-1;
    }
}
```

2) Stack list

- Pendeklarasian

```
#include <iostream.h>
using namespace std;

const int maxElmt=255;

struct elmtList{
    char isi;
    elmtList*next;
};

typedef elmtList* pointer;
typedef pointer list;
```

```
void createStack(list&top){
    top=NULL;
}

void createElmt(pointer&pBaru,char k){
    pBaru=new elmtList;
    pBaru->isi=k;
    cout<<pBaru->isi;
    pBaru->next=NULL;
}
```

- Fungsi void push & pop

```
void push(list&top, pointer pBaru){
    if (top==NULL)
        top=pBaru;
    else{
        pBaru->next=top;
        top=pBaru;
    }
}
```

```
void pop(list&top, pointer&pHapus){
    if (top==NULL){
        pHapus=NULL;
        cout<<"stack kosong, tidak ada yang dihapus"<<endl;
    }
    else if(top->next==NULL){
        pHapus=top;
        top=NULL;
    }
    else{
        pHapus=top;
        top=top->next;
        pHapus->next=NULL;
    }
    cout<<pHapus->isi;
}
```

3) Queue array

- Pendeklarasian

```
#include<iostream>
using namespace std;

const int maxElemen = 255;
```

```
struct Queue{
    char isi[maxElemen];
    int head;
    int tail;
};

void createQueue(Queue& Q){
Q.head = 0;
Q.tail = -1;
}
```

- Fungsi void insert & delete

```
void insertQArray(arrayQ& aq, char baru){
    if(aq.tail == maxElem-1){
        cout<<"Antrian penuh"<<endl;
    }
    else{
        aq.tail = aq.tail+1;
        aq.isi[aq.tail] = baru;
    }
}
```

```
void deleteQArray(arrayQ& aq, char& hapus){
    if(aq.head > aq.tail){
        cout<<"Antrian Kosong"<<endl;
    }
    else{
        hapus = aq.isi[aq.head];
        for(int i=0; i<aq.tail; i++){
            aq.isi[aq.tail] = aq.isi[aq.tail+1];
        }
        aq.tail = aq.tail-1;
    }
}
```

4) Queue list

- Pendeklarasian

```
#include<iostream>
using namespace std;

struct elqueue{
    char info;
    elqueue* next;
};

typedef elqueue* pointer;
typedef pointer list;

struct queue{
    list head;
    list tail;
};
queue Q;
```

- Fungsi void insert & delete

```
void insertQueue(Queue& Q, pointer pBaru){
    if (Q.Head==NULL && Q.Tail==NULL) {
        Q.Head = pBaru;
        Q.Tail = pBaru;
    }
    else {
        Q.Tail->next = pBaru;
        Q.Tail = pBaru;
    }
}
```

```
void deleteQueue(Queue& Q, pointer& pHapus){
    cout<<"Delete Queue"<<endl;
    if (Q.Head==NULL && Q.Tail==NULL) {
        pHapus=NULL;
        cout<<"List Queue kosong "<<endl;
    }

    else if (Q.Head->next==NULL) {
        pHapus=Q.Head;
        Q.Head=NULL;
        Q.Tail=NULL;
    }

    else {
        pHapus=Q.Head;
        Q.Head=Q.Head->next;
        pHapus->next=NULL;
    }
}
```

3. Buatlah program Queue List priority antrian rumah sakit:

   a. Procedure yang harus ada :

   - void createQueue(Queue &Q);
   - void createElement(pointer& pBaru);
   - void insertLastQueue(Queue& Q,pointer pBaru);
   - void insertFirstQueue(Queue& Q,pointer pBaru);
   - void DeleteQueue(Queue &Q,pointer &pHapus);
   - void insertBeforeQueue(Queue &Q, pointer pBaru, pointer pBantu);

- void traversal(Queue Q);

b. Tambahkan menu untuk melakukan input, delete data, dan menampilkan output

c. F.S. Priority Queue adalah penyisipan elemen P ke dalam antrian dengan aturan khusus sbb:
   - Elemen P dengan prioritas tertentu (10 - tertinggi, 1 - terendah) akan menyisip sebelum elemen dengan prioritas yang lebih rendah darinya.
   - Jika prioritas sama maka P akan menyisip tepat dibelakang deretan elemen yang sama prioritasnya
   - P akan menyisip dibelakang antrian jika prioritasnya paling kecil atau sama dengan prioritas Tail.

Jawab:

- Codingan

```
#include<iostream>
using namespace std;

struct elqueue{
    char nama[20];
    int prior;
    elqueue* next;
};

typedef elqueue* pointer;
typedef pointer list;

struct queue{
    list head;
    list tail;
};
queue Q;

void createQueue(queue& Q){
    Q.head = NULL;
    Q.tail = NULL;
}
```

```
void createElement(pointer& pBaru){
    pBaru=new elqueue;
    cout<<"Masukkan nama pasien\t: " ;
    cin>>pBaru->nama;
    cout<<"Masukkan nomor antrian\t: " ;
    cin>>pBaru->prior;
    pBaru->next=NULL;
}

void insertFirstQueue(queue& Q,pointer pBaru){
    pBaru->next=Q.head;
    Q.head=pBaru;
}

void insertLastQueue(queue& Q,pointer pBaru){
    pointer last;
    last=Q.head;
    while(last!=Q.tail){
        last=last->next;
    }
    last->next=pBaru;
    Q.tail=pBaru;
}
```

```
void insertBeforeQueue(queue& Q,pointer pBaru,pointer pBantu){
    pBantu=Q.head;
    pointer prev;
    while(pBantu->prior>pBaru->prior){
        prev=pBantu;
        pBantu=pBantu->next;
    }
    pBaru->next=pBantu;
    prev->next=pBaru;
}
```

```
void inputQueue(queue& Q,pointer& pBaru,pointer pBantu){
    createElement(pBaru);
    if(Q.head == NULL){
        Q.head = pBaru;
        Q.tail = pBaru;
    }
    else{
        if(pBaru->prior>Q.head->prior){
            insertFirstQueue(Q,pBaru);
        }
        else if(pBaru->prior<Q.tail->prior||pBaru->prior==Q.tail->prior){
            insertLastQueue(Q,pBaru);
        }
        else insertBeforeQueue(Q,pBaru,pBantu);
    }
}
```

```
void deleteQueue(queue& Q,pointer& pHapus){
    if(Q.head==NULL){
        cout<<"Antrian sudah kosong!";
    }
    else{
        if(Q.head->next==NULL){
            pHapus=Q.head;
            Q.head=NULL;
        }
        else{
            pHapus=Q.head;
            Q.head=pHapus->next;
            pHapus->next=NULL;
        }
    }
}
```

```
void traversal(queue Q){
    pointer pBantu;
    pBantu=Q.head;
    cout<<"Nama\tNo. Antrian"<<endl;
    do{
        cout<<pBantu->nama<<"\t"<<pBantu->prior<<endl;
        pBantu = pBantu->next;
    }
    while(pBantu!=NULL);
}
```

```
int main(){
    queue Q;
    pointer pBaru,pHapus,pBantu;
    char cek;
    int pilih;

    cout<<"DAFTAR REKAP PASIEN RS HARMONI 28/05/17"<<endl;
    cout<<"****************************************"<<endl;

    createQueue(Q);
    do{
        createElement(pBaru);
        insertFirstQueue(Q,pBaru);
        cout<<"(Y/N)\t: ";
        cin>>cek;
    }
    while(cek=='Y'||cek=='y');

    do{
        cout<<endl;
        cout<<"--------------- M E N U ---------------"<<endl;
        cout<<"1. Tampilkan antrian sementara (traversal)"<<endl;
        cout<<"2. Tambah pasien (input)"<<endl;
        cout<<"3. Hapus pasien yang sudah keluar (delete)"<<endl;
```

```
        cout<<endl<<"Masukkan Pilihan: ";
        cin>>pilih;

        switch(pilih){
            case 0: return 0;
            case 1: traversal(Q);
                break;
            case 2: do{
                        createElement(pBaru);
                        inputQueue(Q,pBaru,pBantu);
                        cout<<"Lagi? (Y/N)\t: ";
                        cin>>cek;
                    }
                    while(cek=='Y'||cek=='y');
                break;
            case 3: do{
                        deleteQueue(Q,pHapus);
                        traversal(Q);
                        cout<<"Lagi? (Y/N)\t: ";
                        cin>>cek;
                    }
                    while(cek=='Y'||cek=='y');
                break;
            default: cout<<"Maaf, pilihan tidak tersedia!";
                break;
        }
    }
    while(pilih!=0);
}
```

- Screenshot Compiling

```
DAFTAR REKAP PASIEN RS HARMONI 28/05/17
****************************************
Masukkan nama pasien    : iya
Masukkan nomor antrian  : 3
(Y/N)   : n

--------------- M E N U ---------------
1. Tampilkan antrian sementara (traversal)
2. Tambah pasien (input)
3. Hapus pasien yang sudah keluar (delete)

Masukkan Pilihan: 2
```

```
Masukkan nama pasien    : sip
Masukkan nomor antrian  : 6
Lagi? (Y/N)     : n

--------------- M E N U ---------------
1. Tampilkan antrian sementara (traversal)
2. Tambah pasien (input)
3. Hapus pasien yang sudah keluar (delete)

Masukkan Pilihan: 1
Nama    No. Antrian
sip     6
iya     3
```

```
--------------- M E N U ---------------
1. Tampilkan antrian sementara (traversal)
2. Tambah pasien (input)
3. Hapus pasien yang sudah keluar (delete)

Masukkan Pilihan: 3
Nama    No. Antrian
iya     3
Lagi? (Y/N)     : n

--------------- M E N U ---------------
1. Tampilkan antrian sementara (traversal)
2. Tambah pasien (input)
3. Hapus pasien yang sudah keluar (delete)

Masukkan Pilihan: 0


Terminated with return code 0
Press any key to continue ...
```