

LAPORAN PROJECT AKHIR
PRAKTIKUM STRUKTUR DATA
PROGRAM PENDAFTARAN MASUK UNIVERSITAS



Anggota Kelompok:

Muhammad Azka Hartami / 12320005

Ryzal Fadhillah / 123200053

Program Studi Informatika

Jurusan Teknik Informatika

Fakultas Teknik Industri

Univeritas Pembangunan Nasional “Veteran” Yogyakarta

2021

Latar Belakang

Seiring berkembangnya zaman maka semakin berkembang pula ilmu pengetahuan dan teknologi yang dihasilkan oleh manusia. perkembangan yang sangat signifikan yaitu dalam bidang pemrograman. Banyak macam bahasa pemrograman yang telah diciptakan di dunia ini salah satunya yaitu bahasa C++.

Dalam sebuah universitas pasti terdapat seleksi untuk mendata calon mahasiswa yang mendaftar di universitas tersebut. Di situasi pandemic seperti ini hampir semua kegiatan seharusnya dilakukan secara tatap muka menjadi dilakukan secara daring. Maka dari itu dibutuhkan seseorang yang ahli dalam pemrograman untuk membuat program untuk mendata para pendaftar universitas yang efisien untuk membantu dalam proses pendataan tersebut agar tidak terjadi kesalahan yang dapat memberatkan naik pendaftar maupun universitasnya.

Dengan adanya program ini kita dapat memudahkan calon mahasiswa atau pendaftar untuk mendaftarkan dirinya tanpa harus datang ke universitasnya. Program ini juga memudahkan sebuah universitas untuk merekap data pendaftar dan siapa saja yang sudah melakukan ujian atau seleksi masuk universitas. Oleh karena itu dalam proyek akhir ini kami membahas tentang program pendaftaran masuk universitas.

Listing Program

```
#include <fstream>
#include <iostream>
#include <conio.h>
#include <windows.h>
#include <string.h>
#include <malloc.h>
#include <string>
using namespace std;

typedef struct typequeue *typeptr;
struct pendaftar{
    char nama[30];
    char jurusan[30];
};
struct typequeue{
    pendaftar info;
    typeptr next;
};
typeptr qdepan,qbelakang;
typeptr bottom, top;

int antrian=0;
char ulang;
void pendaftar(),lihatpendaftar(),filependaftar(),hapuspendaftar();
void ujian(),listujian();
void buatqueue(),buatstack();
void tumpuk(pendaftar IB);
int queuekosong();

main(){
    char pilihan;
    buatqueue();
    buatstack();
    filependaftar();
    menu:
    do{system("cls");

        cout << " ===== \n";
        cout << " | Menu utama Program | \n";
        cout << " ===== \n";
        cout << " |1. Pendaftaran | \n";
        cout << " |2. Ujian | \n";
        cout << " |3. Keluar | \n";
        cout << " ===== \n";
        cout << " pilih : ";cin>>pilihan;

        switch (pilihan){
            //Menu 1 Pendaftaran
            case '1' :
                system("cls");
                cout << " ===== \n";
                cout << " | Menu Pendaftaran | \n";
                cout << " ===== \n";
                cout << " |1. Pendaftaran Data | \n";
                cout << " |2. Lihat Data | \n";
                cout << " |3. Hapus Data | \n";
                cout << " |4. Kembali | \n";
```

```

        cout << " ===== \n";
        cout << " pilih : ";cin>>pilihan;
        if (pilihan=='1'){pendaftaran();}
        else if (pilihan=='2'){lihatpendaftar();system("pause");}
        else if (pilihan=='3'){hapuspendaftar();}
        else if (pilihan=='4'){goto menu;}
        else {
            cout << "pilihan anda salah !!\n";system("pause");}
        break;

//menu 2 ujian
case '2' :
    system("cls");
    cout << " ===== \n";
    cout << " | Menu Ujian      | \n";
    cout << " ===== \n";
    cout << " |1. lakukan Ujian    | \n";
    cout << " |2. Lihat Data      | \n";
    cout << " |3. Kembali        | \n";
    cout << " ===== \n";
    cout << " pilih : ";cin>>pilihan;
    if (pilihan=='1'){ujian();system("pause");}
    else if (pilihan=='2'){listujian();system("pause");}
    else if (pilihan=='3'){goto menu;}
    else {
        cout << "pilihan anda salah !!\n";system("pause");}
    break;
    }
}while(pilihan!='3');
}
// void untuk membuat stak
void buatstack(){
    typeptr NS;
    NS = (typequeue*)malloc(sizeof(typequeue));
    NS = NULL;
    bottom = NS;
    top = bottom;
}
//void untuk membuat antrian atau queue
void buatqueue(){
    qdepan=(typequeue *) malloc(sizeof(typequeue));
    qdepan=NULL;
    qbelakang=NULL;
}
//void untuk mengecek isi queue
int queuekosong(){
    if(qdepan == NULL){
        return (1);
    }else{
        return (0);
    }
}

void tumpuk(pendaftar IB){
    ofstream stackfile;
    stackfile.open("ujian.txt", ios::app);
    stackfile << IB.nama << " " << IB.jurusan << endl;
    typeptr NS;
    NS = (typequeue*)malloc(sizeof(typequeue));
    NS->info = IB;

```

```

        if(bottom == NULL){
            bottom = NS;
        }else{
            top->next = NS;
        }
        top = NS;
        top->next = NULL;
        stackfile.close();
    }
//void untuk melakukan pendaftaran
void pendaftaran(){

    typeptr NB;
    typequeue data;
    ofstream myfile;
    myfile.open("pendaftar.txt", ios::app);

    do{system("cls");
        cout << "INPUT DATA PENDAFTAR DIBAWAH INI\n\n ";
        cout << "===== \n";cin.ignore();
        cout << " Nama          : ";
        cin.getline(data.info.nama, sizeof(data.info.nama));
        cout << " Jurusan dipilih : ";
        cin.getline(data.info.jurusan, sizeof(data.info.jurusan));
        cout<<endl;

        myfile << data.info.nama << " " << data.info.jurusan << endl;
        antrian=antrian+1;
        NB=(typequeue *) malloc(sizeof(typequeue));
        NB->info=data.info;
        if(qdepan==NULL)
            qdepan=NB;
        else
            qbelakang->next=NB;
        qbelakang=NB;
        qbelakang->next=NULL;
        cout<<endl;
        cout << " Input Data Pendaftar Lagi?(y/n) "; cin>>ulang;
    }while(ulang=='y' || ulang=='Y');
    myfile.close();
}
//void untuk melihat data pendaftar
void lihatpendaftar(){
    system("cls");
    typeptr NB;
    typequeue data;
    buatqueue();
    ifstream myfile("pendaftar.txt");
    while(myfile >> data.info.nama >> data.info.jurusan ){
        NB=(typequeue *) malloc(sizeof(typequeue));
        NB->info=data.info;
        if(qdepan==NULL)
            qdepan=NB;
        else
            qbelakang->next=NB;
        qbelakang=NB;
        qbelakang->next=NULL;
    }
    if(queuekosong())
        cout<<"Tidak Ada Pendaftar"<<endl;

```

```

else{
    typeptr bantu;
    bantu=qdepan;
    int i=1;

    cout<<"=== List Pendaftar ===\n\n";
    do{
        cout<<"Pendaftar ke-"<<i<<<endl;
        cout<<" Nama          : "<<bantu->info.nama<<endl;
        cout<<" Jurusan Yang Dipilih : "<<bantu->info.jurusan<<endl;
        cout<<endl;
        i++;
        bantu=bantu->next;
    }while(bantu!=NULL);
}
myfile.close();
}
//void untuk menghapus data pendaftar
void hapuspendaftar(){
    ifstream myfile("pendaftar.txt");
    char carihapus[30];
    typeptr bantu, hapus, NB;
    typequeue data;

    buatqueue();
    while(myfile >> data.info.nama >> data.info.jurusan){
        NB=(typequeue *) malloc(sizeof(typequeue));
        NB->info=data.info;
        if(qdepan==NULL)
            qdepan=NB;
        else
            qbelakang->next=NB;
        qbelakang=NB;
        qbelakang->next=NULL;
    }

    system("cls");
    if(queuekosong()) //cek apakah ada antrian
        cout<<"Tidak Ada Antrian"<<endl;
    else{
        cout<<"=== Hapus Data Pendaftar ==="<<endl;
        cout << "Masukkan Nama yang akan dihapus : "; cin.ignore();cin.getline(carihapus,
sizeof(carihapus));
        if (strcmp(qdepan->info.nama,carihapus)==0) //hapus awal list
        {
            hapus=qdepan;
            cout << "Pendaftar atas nama " << hapus->info.nama << " berhasil dihapus" <<
endl;

            qdepan=hapus->next;
            free(hapus);
            antrian--;
        }
        else
        {
            bantu=qdepan; // cari list yang akan dihapus
            while ((strcmp(bantu->next->info.nama,carihapus)!=0) && (bantu-
>next!=NULL))
            {
                bantu=bantu->next;
            }
            hapus=bantu->next;
            if (hapus==NULL)
            {
                cout<<"List tidak ditemukan\n"; }
            else

```

```

        {
            if (hapus==qbelakang)    //hapus list akhir
            {
                qbelakang=bantu;
                qbelakang->next=NULL;
            }
            else                      //hapus list tengah
            {

                bantu->next=hapus->next;
            }
            cout << "Pendaftar atas nama " << hapus->info.nama << " berhasil
dihapus" << endl;

            free(hapus);
            antrian--;
        }
    }
    bantu=qdepan;
    myfile.close();
    ofstream newfile;
    newfile.open("pendaftar.txt", ios::trunc);
    while(bantu!=NULL){
        data.info = bantu->info;
        newfile << data.info.nama << " " << data.info.jurusan << endl;
        bantu=bantu->next;
    }
    newfile.close();
}
system("pause");main();
}
//void untuk melakukan ujian
void ujian(){
    typequeue data;
    typeptr NB, bantu;
    pendaftar stack;
    ifstream myfile("pendaftar.txt");
    buatqueue();
    while(myfile >> data.info.nama >> data.info.jurusan){
        NB=(typequeue *) malloc(sizeof(typequeue));
        NB->info=data.info;
        if(qdepan==NULL)
            qdepan=NB;
        else
            qbelakang->next=NB;
        qbelakang=NB;
        qbelakang->next=NULL;
    }

    typeptr selesai;
    system("cls");
    if(queuekosong())
        cout << "Tidak Ada Pendaftar" << endl;
    else{
        selesai=qdepan;
        stack = selesai->info;
        qdepan=selesai->next;
        cout << "Pendaftar dengan Data dibawah ini Sudah melakukan Ujian\n\n";
        cout << "Nama          : " << selesai->info.nama << endl;
        cout << "Jurusan Yang Dipilih : " << selesai->info.jurusan << endl;
        tumpuk(stack);
        free(selesai);
    }
}

```

```

        antrian--;

        bantu=qdepan;
        myfile.close();
        ofstream newfile;
        newfile.open("pendaftar.txt", ios::trunc);
        while(bantu!=NULL){
            data.info = bantu->info;
            newfile << data.info.nama << " " << data.info.jurusan << endl;
            bantu=bantu->next;
        }
        newfile.close();
    }
}

//void untuk melihat yang sudah ujian
void listujian(){
    pendaftar data;
    ifstream stackfile("ujian.txt");
    int no=0;
    system("cls");
    if(queuekosong())
        cout << "Belum ada Yang melakukan ujian ";
    else {
        cout << "=== List Yang sudah Melakukan Ujian ===" << endl;
        cout << endl;
        while(stackfile >> data.nama >> data.jurusan){
            cout << no+1 << ".";
            cout << " Nama          : " << data.nama << endl;
            cout << " Jurusan Yang Dipilih : " << data.jurusan << endl;
            cout << endl;
            no++;
        }
        stackfile.close();
    }
}

void filependaftar(){
    typequeue data;
    ifstream myfile("pendaftar.txt");
    while(myfile >> data.info.nama >> data.info.jurusan){
        antrian=antrian+1;
    }
    myfile.close();
}

```


Penjelasan Program

Berikut adalah penjelasan program yang sudah kami buat:

1. Tampilan ketika program pertama kali di buka

```
=====
| Menu utama Program |
=====
|1. Pendaftaran      |
|2. Ujian            |
|3. Keluar           |
=====
pilih : 
```

Gambar 1. 1 Menu Utama

2. Tampilan ketika user memilih menu 1.pendaftaran

```
=====
| Menu Pendaftaran   |
=====
|1. Pendaftaran Data |
|2. Lihat Data       |
|3. Hapus Data       |
|4. Kembali          |
=====
pilih : 
```

Gambar 1. 2 Menu Pendaftaran

3. Jika user menginput angka 1 maka akan masuk ke halaman pendaftaran. Di menu pendaftaran tersebut user diminta untuk mengisi nama dan jurusan.

```
INPUT DATA PENDAFTAR DIBAWAH INI

=====
Nama           : Azka
Jurusan dipilih : Informatika

Input Data Pendaftar Lagi?(y/n) 
```

Gambar 1. 3 Input data pendaftar

4. Jika pada menu pendaftaran user menginputkan angka 2, maka akan masuk ke halaman lihat data pendaftar seperti berikut.

```

=== List Pendaftar ===

Pendaftar ke-1
Nama           : Ryzal
Jurusan Yang Dipilih : Informatika

Pendaftar ke-2
Nama           : Azka
Jurusan Yang Dipilih : Informatika

Press any key to continue . . . █

```

Gambar 1. 4 Lihat data pendaftar

5. Jika pada menu pendaftaran user menginputkan angka 3, maka akan masuk ke halaman hapus data pendaftar dengan menginputkan nama pendaftar yang akan di hapus

```

=== Hapus Data Pendaftar ===
Masukkan Nama yang akan dihapus : Ryzal
Pendaftar atas nama Ryzal berhasil dihapus
Press any key to continue . . . █

```

Gambar 1. 5 Hapus data pendaftar

6. Namun , jika pada menu pendaftaran user menginputkan angka 4. Maka akan kembali ke menu utama.
7. Pada menu utama, jika user menginput angka 2 maka akan masuk kehalaman ujian.

```

=====
| Menu Ujian          |
=====
| 1. lakukan Ujian    |
| 2. Lihat Data       |
| 3. Kembali         |
=====
pilih : █

```

Gambar 1. 6 Menu ujian

8. Jika pada menu ujian user menginput angka 1, maka akan masuk ke halaman untuk melakukan ujian.

```

Pendaftar dengan Data dibawah ini Sudah melakukan Ujian

Nama           : Azka
Jurusan Yang Dipilih : Informatika
Press any key to continue . . . █

```

Gambar 1. 7 Menu melakukan ujian

9. Jika pada menu ujian user menginput angka 2, maka akan masuk ke halaman lihat data yang sudah melakukan ujian.

```
=== List Yang sudah Melakukan Ujian ===  
  
1. Nama           : Azka  
   Jurusan Yang Dipilih : Informatika  
  
Press any key to continue . . . █
```

Gambar 1. 8 Lihat pendaftar yang sudah ujian

10. Namun , jika pada menu ujian user menginput angka 3 maka akan kembali ke menu utama.
11. Pada menu utama. Jika user menginput angka 3 maka akan keluar dari program.

Kesimpulan

Dari program yang telah di buat, dapat disimpulkan bahwa program ini dapat memudahkan calon mahasiswa atau pendaftar dalam melakukan pendaftaran untuk masuk ke sebuah universitas. Namun dalam program ini tidak terdapat fitur untuk melakukan seleksinya, hanya terdapat fitur untuk mendata siapa saja yang sudah dan belum melakukan seleksi atau tes.

Untuk pengembangan lebih lanjut, maka penulis memberikan saran yang dapat bermanfaat dan membantu nantinya baik untuk calon mahasiswa maupun universitas di masa yang akan datang, yaitu :

Perlunya penambahan menu untuk melakukan seleksi sehingga tidak perlu membuka banyak program untuk melakukan tes lagi. Dan perlunya perbaikan tampilan agar pengguna lebih nyaman ketika menggunakannya.

Pembagian Tugas

Nama Tugas	Minggu 1	Minggu 2	Minggu 3	Minggu 4	Keterangan
Perancangan program					Azka, Ryzal
Pembuatan program					Azka, Ryzal
Pembuatan laporan dan presentasi					Azka, Ryzal