

LAPORAN SEDERHANA
SISTEM INFORMASI AKADEMIK



Nama : Setya Handika

NRP : 5024211029

Departemen : Teknik Komputer

Mata Kuliah : Pemrograman Lanjutan

Kelas : A

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2022

Pembahasan

<https://github.com/HandikaSetya70/Tugas-SIM-Akademik-2>

Pada tugas ini, mahasiswa diminta untuk membuat SIM Akademik mulai dari 0. Tugas dikerjakan oleh Setya Handika NRP 5024211029.

Pada hasil ini, dibuatlah beberapa kelas. Yang pertama adalah kelas `user_data`, kelas sebagai berikut:

```
50
51 ~ class user_data{
52     public:
53         string name;
54         string dob;
55         string logid;
56         string pass;
57         string gender;
58
59 ~     void changepass(){
60         string bro;
61         cout << "Input new password: ";
62         cin >> bro;
63         pass = bro;
64         cout << "Password successfully changed!\n New password: " << pass << endl;
65         pause();
66         return;
67     }
68 };
69
```

Pada kelas ini, didapat member yaitu:

- name : Nama
- dob : Tempat tanggal lahir
- logid : Login id
- pass : Password
- gender : Jenis kelamin

Terdapat pula fungsi dalam kelas, yaitu `changepass`. Tujuan utama `changepass` disini adalah sebagai fungsi untuk membantu user mengganti password.

Kemudian, terdapat kelas Mahasiswa. Data dari mahasiswa ini juga mengambil data dari kelas sebelumnya, yaitu `user_data`.

```
class mahasiswa : public user_data {
    public:

        string NRP;
        string major;

        mahasiswa(){};
};
```

Terdapat 2 member untuk mahasiswa, yaitu NRP dan major. NRP adalah NRP dari mahasiswa tersebut, dan major adalah jurusan.

Kemudian terdapat pendeklarasian untuk input data kedalam kelas mahasiswa tersebut:

```
mahasiswa(string a, string c, string d, string e,
string f, string g, string h){
    NRP = a;
    major = c;
    name = d;
    dob = e;
    logid = f;
    pass = g;
    gender = h;
}
```

Maka akan mempermudah input untuk kelas mahasiswa. Kemudian terdapat beberapa fungsi di kelas tersebut:

```
void print(){
    cout << "Data Mahasiswa" << endl << endl;
    cout << "Nama" : " << name <<
endl;

    cout << "Date of birth : " << dob << endl;
    cout << "Gender" : " << gender <<
endl;

    cout << "NRP" : " << NRP << endl;
```

```

        cout << "Major          : " << major <<
endl;
        cout << "logid          : " << logid <<
endl;
        cout << "pass          : " << pass <<
endl;
        return;
    }

```

Fungsi diatas, print, bertujuan untuk mengoutputkan data mahasiswa.

```

void input(){
    string name_a;
    string dob_a;
    string logid_a;
    string pass_a;
    string gender_a;
    string NRP_a;
    string major_a;
    cin.ignore();
    cout << "Name          : ";
    getline(cin, name_a);
    cout << "Date of birth : ";
    getline(cin, dob_a);
    cout << "Create Login ID: ";
    cin >> logid_a;
    cout << "Create password: ";
    cin >> pass_a;
    cout << "Gender          : ";
    cin >> gender_a;
    cout << "NRP            : ";
    cin >> NRP_a;
    cin.ignore();
    cout << "Major          : ";
    getline(cin, major_a);
}

```

```

        name = name_a;
        dob = dob_a;
        logid = logid_a;
        pass = pass_a;
        gender = gender_a;
        NRP = NRP_a;
        major = major_a;
        return;
    }

```

Fungsi diatas, input, bertujuan untuk memasukan data namun secara manual dan bertahap.

Kemudian terdapat kelas tendik atau tenaga kependidikan.

```

class tendik : public user_data {
    public:

        string profession;
        string ID;
        int year_of_entry;
        string education;
        int jml_kelas;

        tendik(){};

        tendik(string a, int b, string c, string d,
string e, string f, string g, string h, string i){
            ID = a;
            year_of_entry = b;
            education = c;
            name = d;
            dob = e;
            logid = f;
            pass = g;

```

```

        gender = h;
        profession = i;
    }

    void print(){
        cout << "Data Tenaga Kependidikan" << endl
<< endl;
        cout << "Nama          : " << name <<
endl;
        cout << "Date of birth : " << dob << endl;
        cout << "Gender       : " << gender <<
endl;
        cout << "ID           : " << ID << endl;
        cout << "Year of entry : " <<
year_of_entry << endl;
        cout << "Education    : " << education <<
endl;
        cout << "logid        : " << logid <<
endl;
        cout << "pass         : " << pass <<
endl;
        return;
    }

    void input(){
        string name_a;
        string dob_a;
        string logid_a;
        string pass_a;
        string gender_a;
        string ID_a;
        double year_of_entry_a;
        string education_a;
    }

```

```
string profession_a;

cin.ignore();
cout << "Name          : ";
getline(cin, name_a);
cout << "Date of birth  : ";
getline(cin, dob_a);
cout << "Create Login ID: ";
cin >> logid_a;
cout << "Create password: ";
cin >> pass_a;
cout << "Gender          : ";
cin >> gender_a;
cout << "ID              : ";
cin >> ID_a;
cout << "Year of entry   : ";
cin >> year_of_entry_a;
cin.ignore();
cout << "Education        : ";
getline(cin, education_a);
cout << "Profession       : ";
cin >> profession_a;

name = name_a;
dob = dob_a;
logid = logid_a;
pass = pass_a;
gender = gender_a;
ID = ID_a;
year_of_entry = year_of_entry_a;
education = education_a;
profession = profession_a;

return;
```

```
}  
  
};
```

Tenaga kependidikan memiliki tambahan member:

- profession : profesi
- ID : ID
- year_of_entry : tahun masuk
- education : Pendidikan
- jml_kelas : Jumlah kelas dari tendik

Kelas tersebut juga memiliki fungsi yang tidak jauh berbeda dengan kelas mahasiswa.

Kemudian terdapat kelas croom (classroom).

```
class croom{  
    public:  
        int jumlah_mhs = 0;  
        tendik dosen;  
        string subjek;  
        map<string,mahasiswa> Mhs;  
        map<string,int> nilai;  
        //string: NRP  
  
        croom(){}  
  
        croom(tendik a, string b){  
            dosen = a;  
            subjek = b;  
        }  
}
```

Pada kelas ini terdapat beberapa member:

- jumlah_mhs : Jumlah mahasiswa
- dosen : Data dosen (dengan kelas tendik)

- subjek : Subjek dari kelas tersebut
- Mhs : List mahasiswa dengan indeks merupakan NRP
- nilai : List nilai dari mahasiswa dengan indeks merupakan NRP

Kemudian terdapat beberapa fungsi,

```
int getgradebynrp(string k){  
    if(Mhs[k].name==""){  
        return 0;  
    }  
    return nilai[k];  
}  
  
void showgradebynrp(string k){  
    if(Mhs[k].name==""){  
        return;  
    }  
    cout << endl;  
    cout << "Class " << subjek << endl;  
    cout << "Name : " << Mhs[k].name << endl;  
    cout << "NRP : " << Mhs[k].NRP << endl;  
    cout << "Grade : " << nilai[k] << endl;  
    return;  
}
```

Kedua fungsi ini memiliki fungsi yang sama, yaitu untuk mendapatkan nilai, namun fungsi yang pertama guna mendapat nilai return berupa nilai, kemudian fungsi kedua menampilkan data secara full, yaitu nama, nrp, dan nilai.

```
void input_mhs(mahasiswa a){  
    Mhs[a.NRP] = a;  
    nilai[a.NRP] = 0;  
    jumlah_mhs++;  
    return;  
}
```

```
}
```

Fungsi input_mhs ini berguna untuk menambahkan mahasiswa dalam kelas tersebut.

```
void get_mhs(){
    map<string,mahasiswa>::iterator it;
    map<string,int>::iterator ti;

    cout << "Jumlah mahasiswa    : " <<
jumlah_mhs << endl;
    int k = 1;
    for(it = Mhs.begin(), ti = nilai.begin();
it != Mhs.end() && k <= jumlah_mhs; ti++, it++, k++){
        cout << k << ". " << it->second.NRP <<
"    " << ti->second << "    \t" << it->second.name <<
endl;
    }
    cout << endl;
    return;
}
```

Fungsi diatas berguna untuk menampilkan jumlah mahasiswa dan data dari mahasiswa kelas tersebut.

```
void input_nilaiauto(string k, int bro){
//string = NRP
    int be = 0;
    nilai[k] = bro;
    return;
}

void input_nilamanual(){
    int n;
```

```

        string b;
        get_mhs();
        cout << "NRP nilai yang di input: ";
        cin >> b;
        cout << "Jumlah nilai: ";
        cin >> n;
        ;
        nilai[b] = n;
        cout << "Input successful!\nName: " <<
Mhs[b].name << "\nGrade: " << nilai[b] << endl;
        return;
    }

```

Kedua fungsi tersebut berguna untuk menginput nilai. Fungsi pertama secara otomatis dan kedua secara manual.

```

    void getinfo(){
        cout << "Kelas          : " << subjek
<< endl;
        cout << "Dosen          : " <<
dosen.name << endl;
        return;
    }

```

Fungsi diatas getInfo adalah untuk menampilkan informasi tentang kelas yaitu subjek dan dosen.

Kemudian untuk main menu awal, maka ditampilkan 3 menu

```
C:\Users\Sgtsa\Documents\Pelajaran\Dasar Pemrograman\Program iki\programs
ssssss ii   aaaa   kk   kk   aaaa   ddddd
ss  ss  ii   aaaa   kk  kk   aaaa   dd  dd
  ss   ii   aa  aa   kkkk   aa  aa   dd  dd
    ss   ii   aaaaaa  kkk   aaaaaa  dd  dd
      ss   ii   aa   aa   kkkk   aa   aa   dd  dd
ss  ss  ii   aa   aa   kk  kk   aa   aa   dd  dd
ssssss ii   aa   aa   kk   kk   aa   aa   ddddd

Menu:
1. Login
2. Create new user
3. exit
input:
```

Pilihan 1 untuk **login**, pilihan 2 untuk **membuat user baru**, dan pilihan 3 untuk **exit**.

Untuk pilihan 1, maka ditampilkan sebagai berikut:

```
Menu:
1. Login
2. Create new user
3. exit
input: 1
ID      : handika
Password : handika_
```

Akan diminta input berupa ID dan password.

Apabila salah, maka muncul

```
2. Create new user
3. exit
input: 1
ID      : handika
Password : handika1
incorrect ID / password. Please try again!
Press any key to continue . . .
```

Dengan kode:

```
cout << "ID      : ";
cin >> gol;
```

```

cout << "Password : ";
cin >> pas;
if(pas == dataMhs[gol].pass){
    cout << "login successful!\n";
    bru = true; //mhs
} else if(pas == dataTendik[gol].pass){
    cout << "login successful!\n";
    bru = false; //tendik
} else if(k<3){
    cout << "incorrect ID / password. Please try
again!\n";
    k++;
    pause();
    goto blok;
} else {
    cout << "You failed to login 3x. Please try
again later.\n";
    pause();
    exit(1);
}

```

Memanfaatkan fungsi if dan else if, maka akan di cek password tersebut sesuai atau tidak.

Apabila selesai login, maka muncul tampilan sebagai berikut:

```
pause();
exit(1);

C:\Users\Sgtsa\Documents\Pelajaran\Dasar Pemrograman\Program iki\program...

ssssss ii   aaaa   kk  kk   aaaa   ddddd
ss  ss  ii   aaaa   kk  kk   aaaa   dd  dd
ss      ii   aa  aa   kkkk   aa  aa   dd  dd
ss      ii   aaaaaa  kkk   aaaaaa  dd  dd
ss      ii   aa   aa   kkkk   aa   aa  dd  dd
ss  ss  ii   aa   aa   kk  kk   aa   aa  dd  dd
ssssss ii   aa   aa   kk  kk   aa   aa  ddddd

Welcome to SIM Akademik!

Main menu:
1. Create new user
2. User information
3. Change user password
4. Classroom
5. logout
6. exit

Input:
```

Menu utama memanfaatkan switch case dengan case sesuai dengan fungsinya masing-masing.

Untuk **create new user**, memanfaatkan fungsi input dalam kelas mahasiswa dan tendik. Apabila dipilih menu pertama, akan muncul

```
5. logout
6. exit

Input: 1
Masukan 1 apabila mahasiswa dan 2 apabila tenaga pendidikan.
input:
```

Maka akan dapat membuat *mahasiswa* maupun *tendik*.

Kemudian untuk pilihan **user information**, akan menampilkan informasi akun user tersebut. Contoh sebagai berikut:

```
Input: 2
Data Mahasiswa

Nama      : Setya Handika
Date of birth : 28 November 2002
Gender    : male
NRP       : 5024211029
Major     : Teknik Komputer
logid     : handika
pass      : handika
Press any key to continue . . .
```

Kemudian untuk pilihan ketiga yaitu **change user password**, tampilannya sebagai berikut:

```
5. logout
6. exit

Input: 3
Input new password: aersk123
Password successfully changed!
New password: aersk123
Press any key to continue . . .
```

Akan diminta input password baru berupa string. Kemudian setelah berhasil diganti akan ditampilkan seperti diatas. Kodenya:

```
case 3:
{
    if (bru){
        dataMhs[gol].changepass();
    } else {
        dataTendik[gol].changepass();
    }
    break;
}
```

Memanfaatkan fungsi changepass pada kelas *user_data*.

Kemudian terdapat pilihan **classroom**. Pada pilihan ini, terdapat 2 macam layout. Apabila anda adalah *mahasiswa*, maka tampilan sebagai berikut:

```
C:\Users\Sgtisa\Documents\Pelajaran\Dasar Pemrograman\Program Ikt\programa
ssssss ii      aaaa  kk  kk      aaaa  ddddd
ss  ss  ii      aaaa  kk  kk      aaaa  dd  dd
  ss    ii      aa  aa  kkkk      aa  aa  dd  dd
    ss    ii      aaaaaa kkk      aaaaaa dd  dd
      ss    ii      aa  aa  kkkk      aa  aa  dd  dd
ss  ss  ii      aa  aa  kk  kk      aa  aa  dd  dd
ssssss ii      aa  aa  kk  kk      aa  aa  ddddd

Welcome to the classroom menu!

Option:
1. Show classes
2. Join a class (requires permission)
3. Show my grades
4. Back
Option:
```

Pada pilihan pertama, **show classes**, berfungsi untuk menunjukan semua kelas yang ada. Contoh tampilan adalah sebagai berikut:

```
C:\Users\Sgtisa\Documents\Pelajaran\Dasar Pemrograman\Program Ikt\programs rea
ssssss ii      aaaa  kk  kk      aaaa  ddddd
ss  ss  ii      aaaa  kk  kk      aaaa  dd  dd
  ss    ii      aa  aa  kkkk      aa  aa  dd  dd
    ss    ii      aaaaaa kkk      aaaaaa dd  dd
      ss    ii      aa  aa  kkkk      aa  aa  dd  dd
ss  ss  ii      aa  aa  kk  kk      aa  aa  dd  dd
ssssss ii      aa  aa  kk  kk      aa  aa  ddddd

Class number 1
Kelas          : Fisika
Dosen           : Bruno Tyson
Jumlah mahasiswa : 3
1. 5024211029   50      Setya Handika
2. 5024211030   100     Leo Kumara
3. 5024211032   90      Benjamin Aldrin

Class number 2
Kelas          : Kimia
Dosen           : Tyson Iron
Jumlah mahasiswa : 2
1. 5024211029   75      Setya Handika
2. 5024211031   80      Ardian Kumparan

Press any key to continue . . .
```


Kode dari pilihan diatas adalah

```
vector<croom>::iterator it;
int b = 0;
it = datacroom.begin();
while(it!=datacroom.end()){
    cout << "Class number " << b+1 <<
endl;

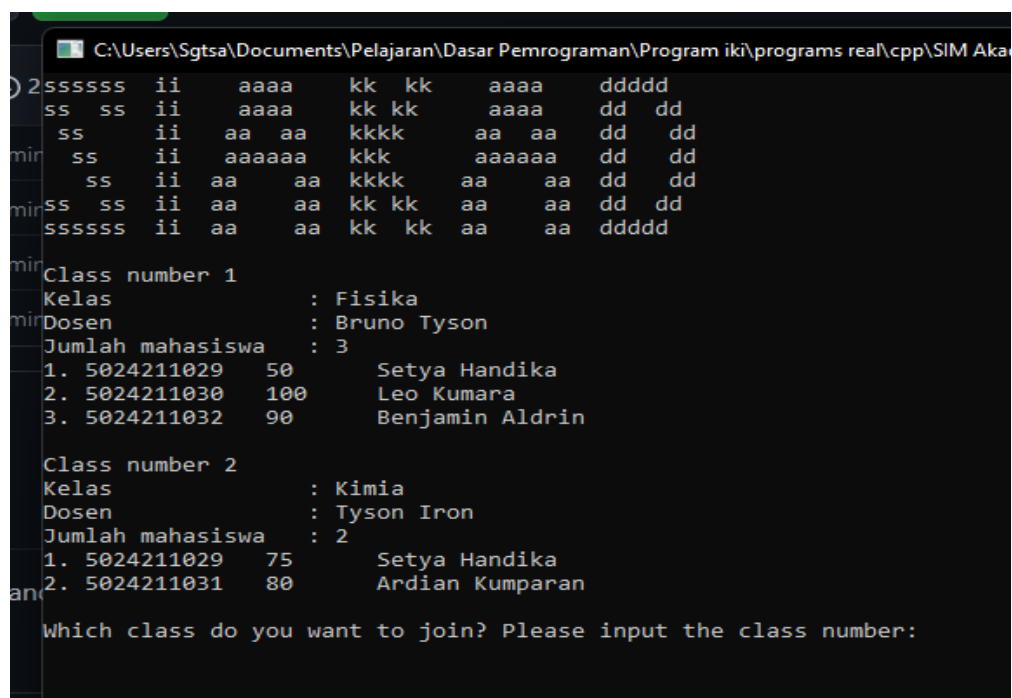
    it->getinfo();
    it->get_mhs();
    b++;
    it++;
}
```

Memfaatkan iterator terhadap data classroom yang telah di declare sebelumnya,

```
vector<croom> datacroom;
```

maka akan menunjukan semua classroom yang ada.

Kemudian pilihan kedua, **join a class** (requires permission), dengan tampilan sebagai berikut:



The screenshot shows a terminal window with the following output:

```
C:\Users\Sgtsa\Documents\Pelajaran\Dasar Pemrograman\Program iki\programs real\cpp\SIM Aka
2 ssssss ii aaaa kk kk aaaa dddd
ss ss ii aaaa kk kk aaaa dd dd
ss ii aa aa kkkk aa aa dd dd
mir ss ii aaaaaa kkk aaaaaa dd dd
ss ii aa aa kkkk aa aa dd dd
mir ss ss ii aa aa kk kk aa aa dd dd
sssss ii aa aa kk kk aa aa dddd
mir
Class number 1
Kelas : Fisika
Dosen : Bruno Tyson
Jumlah mahasiswa : 3
1. 5024211029 50 Setya Handika
2. 5024211030 100 Leo Kumara
3. 5024211032 90 Benjamin Aldrin
Class number 2
Kelas : Kimia
Dosen : Tyson Iron
Jumlah mahasiswa : 2
1. 5024211029 75 Setya Handika
an 2. 5024211031 80 Ardian Kumparan
Which class do you want to join? Please input the class number:
```

Akan ditampilkan semua kelas kemudian diminta input berupa nomor kelas dimana user ingin gabung. Kodenya adalah sebagai berikut

```
        vector<croom>::iterator it;
        int bro = 0, in, b = 0;
        it = datacroom.begin();
        while(it!=datacroom.end()){
            map<string,mahasiswa>::iterator w;
            for(w = it->Mhs.begin(); w != it-
>Mhs.end(); w++){
                if(dataMhs[gol].NRP == w-
>first){
                    goto kek;
                }
            }
            cout << "Class number " << b+1 <<
endl;

            it->getinfo();
            it->get_mhs();
            bro++;
            kek:
            b++;
            it++;
        }
        if(bro == 0){
            cout << "\nYou are already a
student of all available class!\n" << endl;

        } else {
            cout << "Which class do you want
to join? Please input the class number: ";
            cin >> in;
            datpermit[gol] = datacroom[in-1];
            clear();
            logo();
        }
    }
}
```

```

        cout << "\nRequest to join class
sent!\nPlease wait until confirmation!\n Class info:\n
Class number " << in << endl;
        datacroom[in-1].getinfo();
        datacroom[in-1].get_mhs();
        cout << endl;
    }

```

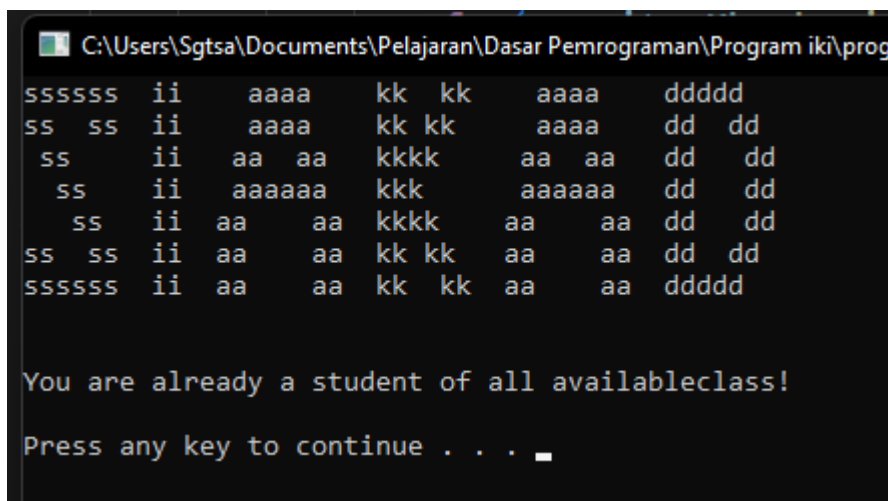
Maka dalam kode tersebut, digunakan beberapa perulangan seperti while dan for. Kemudian memanfaatkan pointer yang dideklarasikan dengan cara berikut.

```

map<string,mahasiswa>::iterator w;

```

Inti dari kode diatas adalah untuk mengecek apakah mahasiswa tersebut sudah berada di dalam kelas tersebut atau belum, apabila belum maka akan dapat dilanjutkan dengan menampilkan kelas tersebut sebagai salah satu pilihan. Apabila tidak terdapat kelas yang ditampilkan maka didapat tampilan:



```

C:\Users\Sgtsa\Documents\Pelajaran\Dasar Pemrograman\Program iki\prog
ssssss ii  aaaa  kk  kk  aaaa  ddddd
ss ss ii  aaaa  kk kk  aaaa  dd  dd
ss  ii  aa  aa  kkkk  aa  aa  dd  dd
ss  ii  aaaaaa kkk  aaaaaa dd  dd
ss  ii  aa  aa  kkkk  aa  aa  dd  dd
ss ss ii  aa  aa  kk kk  aa  aa  dd  dd
ssssss ii  aa  aa  kk  kk  aa  aa  ddddd

You are already a student of all availableclass!

Press any key to continue . . .

```

Kemudian untuk pilihan **show my grades**, akan ditampilkan sebagai berikut:

```

ssssss ii    aaaa    kk kk    aaaa    dddd
ss ss ii    aaaa    kk kk    aaaa    dd dd
  ss ii    aa aa    kkkk    aa aa    dd dd
    ss ii    aaaaaa    kkk    aaaaaa    dd dd
      ss ii    aa aa    kkkk    aa aa    dd dd
ss ss ii    aa aa    kk kk    aa aa    dd dd
ssssss ii    aa aa    kk kk    aa aa    dddd

```

```

Class Fisika
Name : Setya Handika
NRP : 5024211029
Grade : 50

```

```

Class Kimia
Name : Setya Handika
NRP : 5024211029
Grade : 75

```

```

IP: 2.8

```

```

Press any key to continue . . .

```

Dengan kode:

```

        int ind = 0;
        double ip = 0;
        int jmlip = 0;
        vector<croom>::iterator klok =
dataacroom.begin();
        while(klok!=dataacroom.end()){
            klok-
>showgradebynrp(dataMhs[gol].NRP);
            int b =
dataacroom[ind].getgradebynrp(dataMhs[gol].NRP);
            if(b!=0){
                ip += grade(b);
                jmlip++;
            }
            ind++;
            klok++;
        }

```

```
cout <<"\n IP: " << ip/jmlip << endl  
<< endl;
```

Maka pada kode tersebut, menggunakan perulangan untuk mencari nilai dari semua kelas terkhusus nilai dari user tersebut, menggunakan fungsi showgradebynrp dan getgradebynrp. Kemudian akan dihitung sehingga didapat indeks IP, dan menggunakan fungsi untuk menampilkan IP menggunakan kode:

```
double grade(int b){  
    double k;  
    if(b >= 86){  
        k = 4;  
    } else if(b > 75 && b < 86){  
        k = 3.5;  
    } else if(b > 65 && b < 76){  
        k = 3.2;  
    } else if(b > 60 && b < 66){  
        k = 3;  
    } else if(b > 55 & b < 61){  
        k = 2.7;  
    } else if(b > 40 && b < 56){  
        k = 2.4;  
    } else if(b < 41){  
        k = 2;  
    }  
    return k;  
}
```

Kemudian untuk tampilan **classroom** sebagai *tendik* sebagai berikut:

```
C:\Users\SgtSa\Documents\Pelajaran\Dasar Pemrograman\Program iki\
ssssss ii   aaaa   kk  kk   aaaa   ddddd
ss  ss  ii   aaaa   kk  kk   aaaa   dd  dd
  ss    ii   aa  aa   kkkk   aa  aa   dd  dd
    ss   ii   aaaaaa  kkk   aaaaaa  dd  dd
      ss  ii   aa  aa   kkkk   aa  aa   dd  dd
ss  ss  ii   aa  aa   kk  kk   aa  aa   dd  dd
ssssss ii   aa  aa   kk  kk   aa  aa   ddddd

Welcome to the classroom menu!

Option:
1. Show classes
2. Create a new class
3. Input grades
4. Class admission request
5. Back
Option: _
```

Pilihan pertama sama dengan mahasiswa, yaitu menampilkan kelas.
Pilihan kedua, **Create a new class**, untuk membuat kelas baru.
Tampilan sebagai berikut:

```
Option: 2
Please input class's subject: MateMATika
Success!
Class info:
Kelas           : MateMATika
Dosen            : Tyson Iron
Jumlah mahasiswa : 0

Press any key to continue . . .
```

Dengan kode:

```
string subj;
cout << "Please input class's subject:
";

cin >> subj;
croom blob(dataTendik[gol],subj);
datacroom.push_back(blob);
cout << "Success!\n Class info:\n";
blob.getinfo();
blob.get_mhs();
```

Maka memanfaatkan fungsi `push_back`, menambahkan data *croom* pada *datacroom*. Kemudian setelah selesai dibuat kelas baru, maka data kelas baru tersebut akan ditampilkan.

Kemudian untuk pilihan ketiga, **input grades**, akan didapat tampilan:

```
C:\Users\SgtSa\Documents\Pelajaran\Dasar Pemrograman\Program Iki\progr
ssssss ii      aaaa      kk kk      aaaa      ddddd
ss ss ii      aaaa      kk kk      aaaa      dd dd
ss ii      aa aa      kkkk      aa aa      dd dd
ss ii      aaaaaa      kkk      aaaaaa      dd dd
ss ii      aa aa      kkkk      aa aa      dd dd
ss ss ii      aa aa      kk kk      aa aa      dd dd
ssssss ii      aa aa      kk kk      aa aa      ddddd

Class number 2
Kelas          : Kimia
Dosen           : Tyson Iron
Jumlah mahasiswa : 2
1. 5024211029 75      Setya Handika
2. 5024211031 80      Ardian Kumparan

Class number 3
Kelas          : MateMATika
Dosen           : Tyson Iron
Jumlah mahasiswa : 0

Which class number?
Option: 2
Jumlah mahasiswa : 2
1. 5024211029 75      Setya Handika
2. 5024211031 80      Ardian Kumparan

NRP nilai yang di input: 5024211029
Jumlah nilai: 100
```

Dengan kode:

```
int x, b = 0;
clear();
logo();
vector<croom>::iterator it;
for(it = datacroom.begin(); it !=
datacroom.end(); it++, b++){
    if(it->dosen.ID !=
dataTendik[gol].ID){
        continue;
```

```

    }
    cout << "\nClass number " << b+1
<< endl;

    it->getinfo();
    it->get_mhs();
}
cout << "Which class number? \n
Option: ";

cin >> x;
datacroom[x-1].input_nilamanual();

```

Maka memanfaatkan fungsi dari kelas *croom* yaitu *input_nilaimanual* untuk menginput nilai secara manual.

Setelah input nilai selesai maka akan ditampilkan:

```

Input successful!
Name: Setya Handika
Grade: 100
Press any key to continue . . .

```

Kemudian pilihan 4, **Class admission request**, apabila terdapat request dengan kelas yang bersangkutan dengan user (tendik) akan ditampilkan:

```

Request number      : 1
Student name       : Benjamin Aldrin
Student name       : 5024211032
Class subject      : Kimia

Request number to confirm :

```

Apabila tidak maka akan ditampilkan:

```

There are no request made.
Press any key to continue . . .

```

Dengan kode:

```

clear();
map<string,croom>::iterator it;

```



```

        int ok, ko = 0;
        for(it = datpermit.begin(); it !=
datpermit.end(); it++){
            if(it-
>second.dosen.ID==dataTendik[gol].ID){
                cout << "Request number      : "
<< ko+1 << endl;
                cout << " Student name      : "
<< dataMhs[it->first].name << endl;
                cout << " Student name      : "
<< dataMhs[it->first].NRP << endl;
                cout << " Class subject      : "
<< it->second.subjek << endl;
                cout << endl;
                ko++;
            }
        }
        if(ko==0){
            clear();
            logo();
            cout << endl;
            cout << "There are no request
made." << endl;
            pause();
            goto exfile;
        }
        cout << " Request number to confirm :
";

        cin >> ok;
        it = datpermit.begin();
        for(int j = 0; j != ok-1 && it !=
datpermit.end() ; it++){
            if(it-
>second.dosen.ID==dataTendik[gol].ID){

```

```

        j++;
    }
}
clear();
logo();
vector<croom>::iterator lo;
for(lo = datacroom.begin(); lo->subjek
!= it->second.subjek; lo++){
    lo->input_mhs(dataMhs[it->first]);
    datpermit.erase(it->first);
    cout << "Student permitted!" << endl;
    cout << "Class info: " << endl;
    lo->getinfo();
    lo->get_mhs();
}

```

Apabila terdapat request dengan dosen terkait, maka akan ditampilkan data yang bersangkutan, kemudian terdapat pilihan dengan tampilan:

```

Student permitted!
Class info:
Kelas          : Kimia
Dosen           : Tyson Iron
Jumlah mahasiswa : 3
1. 5024211029   75   Setya Handika
2. 5024211031   80   Ardian Kumparan
3. 5024211032   0    Benjamin Aldrin

Press any key to continue . . .

```

Maka mahasiswa yang bersangkutan akan berhasil ditambahkan dalam classroom dengan fungsi input mahasiswa pada kelas *croom*.

Kemudian untuk pilihan ke 5 yaitu **back**. Perintah dari pilihan ini adalah simple dengan menggunakan perintah sebagai berikut:

```

5
6  label:
7
8  goto label;
9

```

Kemudian untuk pilihan ke 5 pada menu utama, **logout**, adalah untuk me-logout user dan mengembalikannya ke halaman utama.

```

Welcome to SIM Akademik!

Main menu:
1. Create new user
2. User information
3. Change user password
4. Classroom
5. logout
6. exit

Input: 5
Do you want to log out of your account?
input 0 to go back and 1 to log out.
input: 1
logout successful!
Press any key to continue . . .

```

Dengan kode:

```

    int ram;
    cout << "Do you want to log out of your
account?\n input 0 to go back and 1 to log out.\n
input: ";

    cin >> ram;
    if(ram==1){

        cout << "logout successful!\n";
        pause();
        goto blok;
    } else if(ram==0){
        goto exfile;
    }

```

```
pause();  
break;
```

Apabila sukses logout, maka akan kembali ke menu awal.

Kemudian untuk pilihan ke 6, **exit**, akan mengeluarkan user dari program dengan kode:

```
return;
```