LAPORAN SEDERHANA

SISTEM INFORMASI AKADEMIK



Nama : Demitrias Wilbert

NRP : 5024211017

Departemen : Teknik Komputer

Mata Kuliah : Pemrogaman Lanjutan

Kelas : A

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2022

**Pembahasan**

Program yang dibuat pada tugas ini adalah lanjutan implementasi dan simulasi sederhana dari sebuah sistem informasi akademik yang terdapat pada sebuah institusi akademik.

Tugas sebelumnya adalah tugas implementasi sederhana dari sistem informasi akademik dari sebuah institusi, yang yang merupakan fork dari link github dosen pembimbing di <https://github.com/rezafuad/prolan-sim>,

Sebagai tugas lanjutan, Penulis telah menambahkan source, memodifikasi source, dan menghapus source yang terasa tidak relevan terhadap tugas ini. Seperti penambahan class baru seperti kelas, matkul, dan form, penghapusan source dan header tendik, dan modifikasi source dan header yang ada agar class-class baru terintegrasi dengan baik.

Untuk file driver di main.cpp, penulis juga telah melakukan perubahan total terhadap file main.cpp di tugas sebelumnya, yakni memperbaiki cara input data menggunakan class form yang menggunakan fungsi regex, menambah banyak fungsi agar asosiasi antarkelas dapat terjalan baik dengan input seperti template untuk print, fungsi input handler, dan sebagainya.

Berikut merupakan beberapa penambahan yang dibuat :

1. **Penambahan class Form**

Class Form adalah kelas yang ditambah untuk membantu penginputan data ke sistem. Class ini terdiri dari prompt (const char\*), buffer string (std::string\*), dan regex (const char\*). Fungsi ini terdapat metode eval (Form::eval()) yang dipanggil untuk mencetak isi prompt, kemudian menunggu input yang akan dimasukkan ke buffer string tersebut, dan kemudian dievaluasikan dengan ekspresi regex yang diberikan. Jika input yang diberikan bernilai “kembali”, method itu akan return nilai 0. Jika formatnya tidak sesuai (std::regex\_match mengembalikan nilai false), method akan mengembalikan nilai 1, dan jika inputnya sesuai, akan mengembalikan nilai 2. Untuk menggunakan class ini, terdapat fungsi luar yang dinamakan fillSingle. Fungsi ini menggunakan while loop untuk memanggil method Form::eval. Sehingga ketika input tidak sesuai dengan regex, akan mencetak “[ERROR] Format Salah”, dan penginputan data dapat diulang kembali. Lalu terdapat fungsi fillForm, yaitu merupakan fungsi yang menerima vector dari form, dan mengiterasikan fungsi fillSingle terhadap setiap elemen di vector itu.

1. **Penambahan class Matkul.**

Class Matkul merepresentasikan sebuah mata kuliah, yaitu kelas yang mengandung id (std::string), nilai sks (float), dan set pointer Kelas (std::set<Kelas\*>).

Pada class Matkul ini, terdapat method untuk mengasosiasikan dirinya dengan sebuah instansi Kelas (Matkul::addKelas(Kelas&)). Fungsi ini ketika dipanggil tidak hanya mengecek dan menambahkan pointer Kelas itu ke set didalamnya, tetapi juga memanggil method setter di class Kelas tersebut (Kelas::setMatkul(Matkul&)) untuk set pointer Matkul ini di kelas tersebut. Begitu juga sebaliknya pada metode pemutusan asosiasi (Matkul::removeKelas(Kelas&)) ketika dipanggil akan menghapus pointer Kelas di argumennya dari set didalamnya, serta memanggil method remove di class Kelas tersebut (Kelas::removeMatkul(Matkul&)). Artinya, hanya dengan satu panggilan method addKelas atau removeKelas, kedua instansi telah dapat berasosiasi atau berhenti berasosiasi.

1. **Penambahan class Kelas.**

Class Kelas merepresentasikan kelas virtual, seperti container yang mengandung id (std::string), set pointer mahasiswa (std::set<Mahasiswa\*>), set pointer dosen (std::set<Dosen>) dan pointer mata kuliah (Matkul\*).

Seperti pada class Matkul, class Kelas juga memiliki method untuk mengasosiasikan dirinya dengan sebuah instansi Matkul (Kelas::setMatkul(Matkul&)). Fungsi ini ketika dipanggil akan set nilai pointer argumennya ke nilai pointer matkul dirinya, serta memanggil fungsi (Matkul::addKelas(Kelas&)) untuk menambahkan pointer Kelas ini di set didalam class Matkul, dan juga method pemutusan asosiasi (Kelas::removeMatkul(Matkul&)) yang akan memanggil fungsi (Matkul::removeKelas(Kelas&)) untuk menghapus pointer Kelas ini dari set didalam Matkul tersebut, dan set nilai pointer matkul di class Kelas ini menjadi NULL. Artinya hanya dengan pemanggilan method setMatkul atau removeMatkul, kedua fungsi dapat berasosiasi atau berhenti berasosiasi.

Hal yang sama juga diimplementasikan antara kelas ini dengan class Dosen dan class Mahasiswa pemanggilan method Kelas::addMahasiswa(Mahasiswa&) atau Kelas::addDosen(Dosen&) akan menginsert pointer itu ke set mahasiswa atau dosen, serta memanggil method Mahasiswa::addKelas(Kelas&) atau Dosen::addKelas(Kelas&). Begitu juga dengan pemanggilan Kelas::removeMahasiswa(Mahasiswa&) atau Kelas::removeDosen(Dosen&). Melalui class ini, dosen yang terasosiasi di class ini dapat memberikan penilaian terhadap mahasiswa-mahasiswa yang terasosiasi dengan mengakses map Kelas dan float di instansi mahasiswa itu.

1. **Perubahan class Mahasiswa dan Dosen.**

Untuk mengasosiasikan class-class ini dengan class Kelas, juga ditambahkan container pointer kelas di masing-masing class ini. Yaitu map pointer Kelas dan Nilai untuk class Mahasiswa, dan set pointer Kelas untuk class Dosen. Alasan digunakan map untuk kelas Mahasiswa adalah untuk sekaligus memegang nilai mahasiswa di kelas tersebut. Seperti class Kelas, juga diberikan fungsi asosiatif Mahasiswa::addKelas(Kelas&), Mahasiswa::removeKelas(Kelas&), dan Dosen::addKelas(Kelas&), Dosen::removeKelas(Kelas&) yang juga akan memanggil Kelas::addMahasiswa(Mahasiswa&), Kelas::removeMahasiswa(Mahasiswa&), dan Kelas::addDosen(Dosen&), Kelas::removeDosen(Dosen&). Sehingga satu panggilan asosiatif dapat mengasosiasikan kedua kelas.

Dosen dapat memberikan nilai ke mahasiswa melalui kelas yang terdaftar oleh dosen itu. Dengan mengambil instansi Kelas yang terdaftar di dosen itu, dapat dipanggil metode Kelas::getRecMhs() yang akan mengembalikan reference set pointer mahasiswa (std::set<Mahasiswa\*>&) yang kemudian dapat dicari mahasiswa yang perlu penilaian, dan digunakan map di mahasiswa tersebut, untuk di set nilainya.

1. **Fungsi main**

Pada fungsi main, dilakukan demo terhadap semua perubahan yang dibuat. Program awalnya akan mendisplay jumlah mahasiswa, dosen, dan kelas di sistem, kemudian akan ditampilkan menu :

1. Add Mahasiswa
2. Add Dosen
3. Add Matkul
4. Select Mahasiswa
5. Select Dosen
6. Select Matkul
7. Remove Mahasiswa
8. Remove Dosen
9. Remove Mata Kuliah
10. Print Mahasiswa
11. Print Dosen
12. Print Kelas
13. Keluar