

TUGAS PEMROGRAMAN LANJUT



Disusun Oleh :

Nama : Sendy Prisma Nurferian
NRP : 5024211012
Departemen : Teknik Komputer
Kelas : (A) Pemrograman Lanjut
Dosen : Bapak Reza Fuad Rachmadi
Tugas : Membuat Laporan Penjelasan Code SIM Akademik

INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

TEKNIK KOMPUTER

2022

Nama akun Github : M4\$ \$3ÑĐÝ (SendyPrismanaNurferian)
Link akun Github : <https://github.com/SendyPrismanaNurferian>
Link Repo Tugas : <https://github.com/SendyPrismanaNurferian/Tugas-Proglan-Siakad>

PENJELASAN TUGAS

Dalam penugasan ini diberikan repository project SIM akademik dalam universitas sederhana yang *terconsole language programing* dalam bahasa C++ yang berasal dari Repository GitHub Bapak Fuad selaku dosen mata kuliah Pemrograman Lanjut. Akan tetapi, dalam repository tersebut code-code nya masih error dan harus diperbaiki oleh mahasiswa yang diajalkan terutama di kelas `main.cpp` yang difungsikan sebagai program. Mahasiswa diharapkan bisa mengerjakan kode-kode tersebut hingga tidak ada error. Di dalam file *directory* yang bernama `main.cpp` saya menambahkan kode-kode program di *case 1* sampai *case 6*.

Dalam repository terdapat 3 folder yaitu `include` yang nantinya berisi `.hpp`, lalu ada folder yang bernama `src` yang berisi file `.cpp` dan terakhir `makefile`. Dalam `makefile` ini sudah dibuat oleh bapak Fuad yang bertujuan untuk mengatur format dari jalannya program ini. Dan didalam `.hpp` maupun `.cpp` berisi Mahasiswa, Dosen, Person, Tendik untuk `.hpp` lalu untuk Mahasiswa, Dosen, Tendik, Main, Person, Tendik berada di `.cpp`. Dari *case 1* sampai *case 3* adalah berfungsi sebagai fitur untuk user menginput data, baik data mahasiswa, data dosen, maupun data tendik. Jadi dapat digaris besarkan bahwa didalam setiap *case* saya mendeklarasikan beberapa variabel terlebih dahulu untuk menampung data yang dimasukkan, seperti *nama*, *dd*, *mm*, *yy*, dan sebagainya. Setelah deklarasi saya membuat kode supaya user bisa memasukkan datanya ke dalam variabel-variabel yang difungsikan sebagai penampung. Lalu untuk data yang hanya terdiri 1 kata, saya menggynakan fungsi `std::cin`. Lalu untuk data yang terdiri dari lebih dari 1 kata disini saya menggunakan fungsi lain yaitu `getline()` dan tepat sebelum fungsi tadi saya menambahkan `cin.ignore()` untuk menghilangkan sifat fungsi `'\n'` dari fungsi `cin` sebelumnya.

Sedangkan untuk data yang jenisnya `id` saya disini menggunakan urutan kapan data tersebut akan dimasukkan dan ditambahkan oleh user dengan jenis data tersebut di awalnya. Untuk kelas mahasiswa memiliki kode berupa "mhsw", untuk dosen kode berupa "dsn" dan juga untuk tendik kode berupa "tndk". Untuk lebih paham dapat dijabarkan seperti ini, jika ada mahasiswa yang input pertama kali, maka ID dari mahasiswa tersebut adalah "mhsw1". Untuk urutan kedua adalah "mhsw2", dan seterusnya sampai data yang dirasa cukup. Dan sifat data tersebut berlaku ke semuanya yaitu dosen dan tendik. Setelah selesai mengatur ID saya kemudian membuat deklarasi dan inialisasi sebuah object yang nantinya digunakan untuk menampung setiap variabel-variabel tersebut. Tempat atau object ini berbentuk vector, dan saya beri nama vector `recMhs` untuk mahasiswa, `recDosen` untuk dosen, `recTendik` untuk tendik. Langkah terakhir adalah saya menggunakan fungsi `push_back` untuk menambahkan setiap object yang sudah dibuat ke dalam vector `recMhs` atau `recDosen`, `recTendik` tadi.

Dan saya juga menambahkan sebuah file lagi di dalam file `person`, mahasiswa, dosen dan `tendik.hpp` yaitu jenis kelamin. Saya menggunakan string sama halnya dengan nama dan lain-lain dalam menginisialisasi. Namun disini dalam jenis kelamin ini saya terdapat kendala untuk menentukan secara khusus untuk `out[utnya` dan saya tambahkan lagi kedalam `main.cpp`.

Saya tampilkan perintah untuk user agar menginputkan jenis kelamin ini pria dan wanita karena dalam fungsi jenis kelamin ini tidak saya beri *getline()*. Jika saya beri maka program bisa menampilkan output lebih dari dua kata nantinya. Contoh Laki-Laki maka akan tertampil tulisan Laki-Laki, namun jika saya tidak beri *getline()* maka akan tertampil Laki saja.

Untuk case 4 sampai case 6 difungsikan sebagai fitur untuk menampilkan data, baik data dari mahasiswa, data dari dosen maupun data dari tendik. Dalam melakukan hal tersebut saya menggunakan bentuk perulangan *for* yang berguna untuk tiap-tiap iterasinya. Sehingga untuk tiap iterasinya setara dengan *size* dari database vector *recMhs*, *recDosen*, *recTendik*. Maka setiap iterasinya juga dapat disimpan atau diambil dan ditampilkan data-data tiap dari object sesuai dengan urutan. Cara untuk mengakses data-data di subkelas *private* dengan menggunakan fungsi-fungsi seperti *getDepartemen()*, *getNRP()*, untuk mengakses atau mengambil izin di *private*.

Untuk data yang bersifat *vector* maka saya menggunakan bentuk perulangan *for* supaya bisa mengakses data. Saya juga membuat kode *looping* agar program memberikan pertanyaan ke user untuk melanjutkan/menjalankan programnya lagi atau tidak. Dengan *case* yaitu apabila user tersebut menjawab dengan menekan 'Y' atau 'y' maka user akan dilanjutkan *looping* programnya. Namun juga sebaliknya agar user bisa mengakhiri program dengan cara tekan 'N' atau 'n' untuk menghentikan atau *break looping* programnya. Sedangkan jika user tidak mengetik atau mmenginputkan kata sesuai perintah maka program akan menampilkan pesan "Input yang Anda masukkan Salah". Dan program akan menanyakan ulang terkait pilihan user. Disini saya menggunakan *flow-control* yang bernama *goto* agar program bisa menanyakan user seperti yang sudah saya jelaskan.

Di dalam file *main.cpp* ini saya menambahkan fungsi-fungsi yang sekiranya menurut saya kurang di file *.hpp* di folder *include* yang berfungsi untuk membantu mengakses data-data yang bersifat *private*. Alasannya pada file *.hpp* ini yang diclone dari repository GitHub Pak Fuad belum ada beberapa fungsi tersebut. Contoh di file *mahasiswa.hpp* saya membuat deklarasi fungsi *getNRP()*, *getDepartemen()*, *getTahunmasuk()*, *getIPK()* dan lain-lain. Lalu saya menginisialisasi fungsi-fungsi tersebut ke file *mahasiswa.cpp* di dalam folder *src*. Tak hanya di file *mahasiswa*, saya juga menambahkan beberapa fungsi di *dosen.hpp* dan *dosen.cpp*. Untuk *dosen.hpp* ini saya menambahkan *getNPP()*, dan juga *getDepartemen()*. Saya lakukan hal yang sama seperti sebelumnya yaitu menginisialisasi di file *dosen.cpp* di dalam folder *src*. Dan di file *tendik* di bagian *tendik.hpp* saya menambahkan *getUnit()*. Lalu saya inisialisasikan dalam file *tendik.cpp* yang berada di folder *src*.

Database dalam project ini yang digunakan adalah berupa data berbentuk *vector*. Maka jika user nanti *merestart* program maka hasilnya data akan hilang dan mengulang atau memulai dari awal lagi. Agar hal ini tidak terjadi user bisa menyimpan data yang diinputkan disimpan di dalam suatu file eksternal berupa format *txt* ataupun lainnya.

Itulah penjelasan dari saya untuk tugas Pemrograman Lanjut tentang SIM sederhana dengan *console C++* yang telah saya buat. Mohon maaf atas kurang dan lebihnya dalam penyusunan laporan ini dan apabila dalam program yang saya buat masih kurang baik. Terima kasih atas perhatiannya.