

# **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 SIM Akademik Bagian 2

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA**

**TEKNIK KOMPUTER**

**2022**

Nama akun : M4\$ \$3ÑĐÝ (SendyPrismanaNurferian)  
Link akun : <https://github.com/SendyPrismanaNurferian>  
Link Repo : [https://github.com/SendyPrismanaNurferian/Proglan-Siakad\\_Pt2](https://github.com/SendyPrismanaNurferian/Proglan-Siakad_Pt2)

## 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*. Dan dalam penambahan tugas di part 2 ini mahasiswa melanjutkan program yang sudah jadi untuk ditambahkan menu mata kuliah. Hal ini selaras dengan SIM yang ada di ITS Siakad karena bisa mengakses untuk melihat maupun menambahkan dari mata kuliah.

Seperti sebelumnya, 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 menggunakan 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 inisialisasi 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 outputnya 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 ada penambahan seperti pada gambar dibawah ini

```
class mahasiswa : public person {
private:
    std::string nrp;
    std::string departemen;
    int tahunmasuk;
    int semesterke;
    int skslulus;
    float ipk;
    std::vector<float> ips;
    int skstempuhsemesterini;
    std::vector<std::string> DaftarMataKuliah;
```

```

//Fungsi yang saya jalankan di program kedua ini yaitu ada:
void hitungIPK();
float getIPK(); //Fungsi untuk mengambil IPK

void setIPS(int semester, float ips);
float getIPS(int semester);
std::vector<float> getAllIPS();

int getSKSTempuh();
void TambahMataKuliah(std::string MataKuliah, int SKS);
void TampilDaftarMatkul();
};

#endif

```

Lalu saya menginisialisasi fungsi-fungsi tersebut ke file mahasiswa.cpp di dalam folder src. Saya juga menambahkan beberapa fungsi di mahasiswa.cpp dan mahasiswa.hpp. Karena ada matakuliah yang pastinya ada beban sks dan nanti beban sks di konversi setiap semester. Maka jika tidak ada perubahan di mahasiswa maka program tidak bisa sinkron nantinya jika user meminta sks dalam matakuliah yang saat ini sedang dijalani oleh mahasiswa. Dan saya menambahkan fungsi untuk IPS dan IPK agar bisa terakumulasi dan bisa ditampilkan saat user meminta. Berikut beberapa bagian yang saya tambahkan dalam mahasiswa.cpp seperti gambar dibawah ini

```

void mahasiswa::hitungIPK()
{ float total = 0;
  for(long unsigned int i = 0; i < ips.size(); i++){
    total += ips[i];
  }
  total /= (this->getSemester()-1);
  this->ipk = total;
}

void mahasiswa::setIPS(int semester, float ips)
{
  // semester mulai dari 1
  if (semester < 15) {
    this->ips[semester-1] = ips;
    this->hitungIPK();
  }
}

float mahasiswa::getIPS(int semester)
{
  if (semester < 15)
    return this->ips[semester-1];
  return -1.0;
}

```

```

int mahasiswa::getSKSTempuh()
{
    return this->skstempuhsemesterini;
}

void mahasiswa::TambahMataKuliah(std::string MataKuliah, int sks)
{
    if ((this->skstempuhsemesterini + sks) > 24){
        std::cout << "Mohon maaf SKS sudah melebihi batas maksimal, mohon di cek kembali!" << std::endl;
    } else{
        this->skstempuhsemesterini += sks;
        this->DaftarMataKuliah.push_back(MataKuliah + " (" + std::to_string(sks) + " sks)");
    }
}

void mahasiswa::TampilDaftarMatkul()
{
    for(unsigned int i = 0; i < this->DaftarMataKuliah.size(); i++){
        std::cout << this->DaftarMataKuliah[i];
        if(i != (this->DaftarMataKuliah.size()-1)){
            std::cout << ", ";
        }
    }
}

```

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 mnginisialisasi 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.

Lalu tambahan dari program ini adalah saya membuat 1 kelas baru di masing – masing folder src dan include. Yaitu matakuliah.hpp di folder include dan juga matakuliah.cpp di folder src. Seperti sebelumnya yang dijelaskan masing – masing fungsi untuk file yang berformat .cpp dan .hpp untuk kelas matakuliah. Untuk detail isi dari matakuliah.cpp dan matakuliah.hpp seperti gambar dihalaman selanjutnya.

```

include > C:\matakuliah.hpp > ...
1  #ifndef __MATAKULIAH_HPP__
2  #define __MATAKULIAH_HPP__
3
4  #include <iostream>
5  #include <vector>
6
7  struct DataMhs{
8      std::string nama;
9      std::string nrp;
10     std::string departemen;
11 };
12
13 struct DataDsn{
14     std::string nama;
15     std::string npp;
16     std::string departemen;
17 };
18
19 class MataKuliah{
20 private:
21     std::string id;
22     std::string nama_matakuliah;
23     int bebansks;
24     std::vector <DataMhs> databaseMhs;
25     std::vector <DataDsn> databaseDsn;
26
27 public:
28     MataKuliah(std::string id, std::string nama, int sks);
29
30     void fillDatabaseMhs(DataMhs inputData);
31     void fillDatabaseDsn(DataDsn inputData);
32     std::string getId();
33     std::string getName();
34     int getBebanSKS();
35     std::vector<DataDsn> getDatabaseDsn();
36     std::vector<DataMhs> getDatabaseMhs();
37 };
38
39 #endif

```

Gambar di atas adalah fungsi inisiasi saya di matakuliah.hpp. Di dalam matakuliah.hpp saya beri inisiasi untuk dosen dan mahasiswa. Dengan struct, sangat mudah nantinya menjadikan dosen dan mahasiswa sebagai kelompok variabel yang difungsikan menjadi sebuah tipe data baru yang bisa tersambung dengan database yang sudah saya buat dahulu. Lalu ada didalam struct beberapa string yang sebagai inisiasi untuk struct itu sendiri. Tak lupa saya menambahkan beban sks yang pastinya dalam mata kuliah adalah hal yang wajib untuk di masukkan agar nantinya user tahu bahwa ada mata kuliah yang mungkin ada 2 sks atau 3 sks (*opsional tergantung kemauan user tersebut*). Dan beban sks tadi saya inisiasikan sebagai tipe data int. Lalu ada beberapa *vector* untuk database data dari mahasiswa dan dosen yang telah masuk dalam suatu mata kuliah/kelas matkul. Nah ini sangat berfungsi ketika user mencari matkul yang nantinya akan tertampil nama dari dosen dan juga mahasiswa yang bergabung dalam mata kuliah tersebut. Penjelasan selanjutnya saya tambahkan void fillDatabase\*\*\*\* untuk memasukkan data dari struct tadi bisa di masukkan di database dan juga bisa tertampil nantinya.

Lalu saya membuat matakuliah.cpp dengan fungsi seperti gambar dibawah ini

```
src > matakuliah.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include "include/matakuliah.hpp"
4
5  Matakuliah::Matakuliah(std::string id, std::string nama, int sks)
6  |   : id(id), nama_matakuliah(nama), bebansks(sks) {}
7
8  void Matakuliah::fillDatabaseMhs(DataMhs inputData){
9  |   this->databaseMhs.push_back(inputData);
10 }
11
12 void Matakuliah::fillDatabaseDsn(DataDsn inputData){
13 |   this->databaseDsn.push_back(inputData);
14 }
15
16 std::string Matakuliah::getId(){
17 |   return this->id;
18 }
19
20 std::string Matakuliah::getNama(){
21 |   return this->nama_matakuliah;
22 }
23
24 int Matakuliah::getBebanSKS(){
25 |   return this->bebansks;
26 }
27
28 std::vector<DataDsn> Matakuliah::getDatabaseDsn(){
29 |   return this->databaseDsn;
30 }
31
32 std::vector<DataMhs> Matakuliah::getDatabaseMhs(){
33 |   return this->databaseMhs;
34 }
```

Didalam main.cpp saya juga menambahkan beberapa fungsi yang dibutuhkan program agar program bisa mengerti dan juga berjalan tanpa ada halangan. Seperti gambar dibawah ini.

```
//Fungsi untuk mencari nama ke database Dosen, Mahasiswa yang berada di database Mata Kuliah
template <typename T> int DatabaseCariNama(T &rec, string inputNama){
    string inputNama_hurufbesar, inputNama_hurufkecil, nama_hurufkecil, nama_hurufbesar;

    inputNama_hurufkecil = inputNama;
    transform(inputNama_hurufkecil.begin(), inputNama_hurufkecil.end(), inputNama_hurufkecil.begin(), ::tolower);
    inputNama_hurufbesar = inputNama;
    transform(inputNama_hurufbesar.begin(), inputNama_hurufbesar.end(), inputNama_hurufbesar.begin(), ::toupper);

    for(int i = 0; i < rec.size(); i++){
        nama_hurufkecil = rec[i].getNama();
        transform(inputNama_hurufkecil.begin(), inputNama_hurufkecil.end(), inputNama_hurufkecil.begin(), ::tolower);
        nama_hurufbesar = rec[i].getNama();
        transform(inputNama_hurufbesar.begin(), inputNama_hurufbesar.end(), inputNama_hurufbesar.begin(), ::toupper);

        if((nama_hurufkecil == inputNama_hurufkecil) || (nama_hurufbesar == inputNama_hurufbesar)){
            return i;
        }
    }
    return -1;
}
```

Saya menuliskan fungsi tersebut untuk memudahkan nantinya program agar bisa mencari data dosen dan mahasiswa yang berada di database matakuliah. Tulisan dari nama dosen maupun mahasiswa dari yang huruf kecil dan besar dan sebaliknya agar program tidak error dan pastinya bisa di cari dengan mudah nantinya.

```

while(1){
    //system akan membersihkan layar konsol
    system("clear");
    cout << "Selamat datang di Universitas Kembang Lambe Tunas Bangsa" << endl << endl;
    cout << "Data Statistik :" << endl;
    cout << " * Jumlah Mahasiswa          : " << recMhs.size() << " mahasiswa" << endl;
    cout << " * Jumlah Dosen                : " << recDosen.size() << " dosen" << endl;
    cout << " * Jumlah Tenaga Kependidikan  : " << recTendik.size() << " tenaga kependidikan" << endl;
    cout << " * Jumlah Mata Kuliah          : " << recMatkul.size() << " mata kuliah" << endl;
    cout << endl;
    cout << "Silahkan Pilih Salah Satu Menu";
    cout << "Menu: " << endl;
    cout << " 1. Tambah Mahasiswa" << endl;
    cout << " 2. Tambah Dosen" << endl;
    cout << " 3. Tambah Tenaga Kependidikan" << endl;
    cout << " 4. Tampilkan semua Mahasiswa" << endl;
    cout << " 5. Tampilkan semua Dosen" << endl;
    cout << " 6. Tampilkan semua Tenaga Kependidikan" << endl;
    cout << " 7. Tambah Mata Kuliah" << endl;
    cout << " 8. Tambah Dosen mengisi Mata Kuliah" << endl;
    cout << " 9. Tambah Mahasiswa di Mata Kuliah" << endl;
    cout << " 10. Tampilkan Semua Mata Kuliah" << endl;

    cin >> menu_terpilih;
}

```

Dan penjelasan selanjutnya saya mengubah tampilan menu untuk user saat program dijalankan seperti gambar diatas. Terdapat tambahan terutama untuk mata kuliah, dari menambahkan, menampilkan dan juga memasukkan mahasiswa maupun dosen untuk di dalam mata kuliah dalam program. Lalu saya menambahkan juga di ID Mata Kuliah dan juga *vector* untuk mata kuliah. Dan gambar di bawah ini adalah ada tambahan dari saya untuk fungsi di case memasukkan data mahasiswa untuk user.

```

for(int i = 1; i < semesterke; i++){
    cout << "Masukkan IPS Semester anda sekarang/ke=" << i << " : ";
    cin >> ips_temp; cin.ignore();
    ips.push_back(ips_temp);
}

//Menetapkan inputan yang di masukkan ke object penyimpan yaitu inputMhs
mahasiswa inputMhs = mahasiswa(id, nama, jeniskelamin, dd, mm, yy, nrp, departemen, tahunmasuk);
//Fungsi logika untuk semester, sks lulus, dan IPS
inputMhs.setSemester(semesterke);
inputMhs.setSKSLulus(skslulus);
for(int i = 1; i < semesterke; i++){
    inputMhs.setIPS(i, ips[i-1]);
}

```

Selanjutnya saya juga menambahkan fungsi di case menampilkan data dari mahasiswa agar fungsi di case input data mahasiswa nanti bisa tertampil dan tidak error. Saya menambahkan beberapa fungsi tersebut seperti gambar dibawah ini



```

// menampilkan data
for(int i = 0; i < IDMhs; i++){
    cout << i+1 << ". Data Mahasiswa dengan ID : " << recMhs[i].getId() << endl;
    cout << "\tNama      : " << recMhs[i].getNama() << endl;
    cout << "\tJenis Kelamin  : " << recMhs[i].getjeniskelamin() << endl;
    cout << "\tdtdd/mm/yy    : " << recMhs[i].getTglLahir() << "/" << recMhs[i].getBulanLahir() << "/" << recMhs[i].getTahunLahir() << endl;
    cout << "\tNRP         : " << recMhs[i].getNRP() << endl;
    cout << "\tDepartemen    : " << recMhs[i].getDepartemen() << endl;
    cout << "\tTahun Masuk   : " << recMhs[i].getTahunmasuk() << endl;
    cout << "\tSemester ke   : " << recMhs[i].getSemester() << endl;
    cout << "\tSKS Lulus     : " << recMhs[i].getSKSLulus() << endl;
    printf ("\tIPK      : %.2f\n", recMhs[i].getIPK());
    // menampilkan detail ips
    if(tampilkanIPS == 'y' || tampilkanIPS == 'Y'){
        cout << "    Rincian IPS :\n";
        for(int j = 1; j < recMhs[i].getSemester(); j++){
            cout << "\tsem" << j;
        }
        cout << endl;
        for(int j = 1; j < recMhs[i].getSemester(); j++){
            cout << "\t" << recMhs[i].getIPS(j);
        }
        cout << endl;
    }
    // menampilkan matkul dan sks semester ini
    cout << "\tSKS Tempuh Semester ini : ";
    if(recMhs[i].getSKSTempuh() == 0){
        cout << "Mahasiswa Belum Menempuh SKS di Semester Ini" << endl;
    } else{
        cout << recMhs[i].getSKSTempuh() << endl;
    }
    recMhs[i].TampilDaftarMatkul();
    cout << endl;
}

```

Disini saya juga memberikan hasil dari program yang sudah saya buat :

Selamat datang di Universitas Kembang Lambe Tunas Bangsa

Data Statistik :

```

* Jumlah Mahasiswa      : 0 mahasiswa
* Jumlah Dosen          : 0 dosen
* Jumlah Tenaga Kependidikan : 0 tenaga kependidikan
* Jumlah Mata Kuliah     : 0 mata kuliah

```

Silahkan Pilih Salah Satu MenuMenu:

1. Tambah Mahasiswa
2. Tambah Dosen
3. Tambah Tenaga Kependidikan
4. Tampilkan semua Mahasiswa
5. Tampilkan semua Dosen
6. Tampilkan semua Tenaga Kependidikan
7. Tambah Mata Kuliah
8. Tambah Dosen mengisi Mata Kuliah
9. Tambah Mahasiswa di Mata Kuliah
10. Tampilkan Semua Mata Kuliah

7

'clear' is not recognized as an internal or external command,  
operable program or batch file.

Masukkan Nama Mata Kuliah : DD

Masukkan Beban SKS dari 2-6 : 3

Apakah Anda akan menjalankan atau melanjutkan program? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : y

'clear' is not recognized as an internal or external command,  
operable program or batch file.

```

Selamat datang di Universitas Kembang Lambe Tunas Bangsa

Data Statistik :
* Jumlah Mahasiswa      : 0 mahasiswa
* Jumlah Dosen          : 0 dosen
* Jumlah Tenaga Kependidikan : 0 tenaga kependidikan
* Jumlah Mata Kuliah     : 1 mata kuliah

Silahkan Pilih Salah Satu MenuMenu:
1. Tambah Mahasiswa
2. Tambah Dosen
3. Tambah Tenaga Kependidikan
4. Tampilkan semua Mahasiswa
5. Tampilkan semua Dosen
6. Tampilkan semua Tenaga Kependidikan
7. Tambah Mata Kuliah
8. Tambah Dosen mengisi Mata Kuliah
9. Tambah Mahasiswa di Mata Kuliah
10. Tampilkan Semua Mata Kuliah
10
'clear' is not recognized as an internal or external command,
operable program or batch file.
Apakah Anda Ingin Melihat list Nama Dosen dan Mahasiswa yang ada dalam Mata Kuliah? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : n
1. Data Mata Kuliah dengan ID : Matkul1
   Nama Mata Kuliah      : DD
   Beban SKS Mata Kuliah ini : 3

Apakah Anda akan menjalankan atau melanjutkan program? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : y
'clear' is not recognized as an internal or external command,
operable program or batch file.
Selamat datang di Universitas Kembang Lambe Tunas Bangsa

Data Statistik :
* Jumlah Mahasiswa      : 0 mahasiswa
* Jumlah Dosen          : 0 dosen
* Jumlah Tenaga Kependidikan : 0 tenaga kependidikan
* Jumlah Mata Kuliah     : 1 mata kuliah

Silahkan Pilih Salah Satu MenuMenu:
1. Tambah Mahasiswa
2. Tambah Dosen
3. Tambah Tenaga Kependidikan
4. Tampilkan semua Mahasiswa
5. Tampilkan semua Dosen
6. Tampilkan semua Tenaga Kependidikan

```

```

Selamat datang di Universitas Kembang Lambe Tunas Bangsa

Data Statistik :
* Jumlah Mahasiswa      : 0 mahasiswa
* Jumlah Dosen          : 0 dosen
* Jumlah Tenaga Kependidikan : 0 tenaga kependidikan
* Jumlah Mata Kuliah     : 1 mata kuliah

Silahkan Pilih Salah Satu MenuMenu:
1. Tambah Mahasiswa
2. Tambah Dosen
3. Tambah Tenaga Kependidikan
4. Tampilkan semua Mahasiswa
5. Tampilkan semua Dosen
6. Tampilkan semua Tenaga Kependidikan
7. Tambah Mata Kuliah
8. Tambah Dosen mengisi Mata Kuliah
9. Tambah Mahasiswa di Mata Kuliah
10. Tampilkan Semua Mata Kuliah
2
'clear' is not recognized as an internal or external command,
operable program or batch file.
Masukkan Nama Dosen : re
Masukkan Jenis Kelamin (Pria/Wanita) : pria
Masukkan Tanggal Lahir : 11
Masukkan Bulan Lahir (Angka) : 02
Masukkan Tahun Lahir : 1989
Masukkan NPP : 15763517
Masukkan Departemen : Teknik
Masukkan Pendidikan : S1
Apakah Anda akan menjalankan atau melanjutkan program? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : y
'clear' is not recognized as an internal or external command,
operable program or batch file.
Selamat datang di Universitas Kembang Lambe Tunas Bangsa

Data Statistik :
* Jumlah Mahasiswa      : 0 mahasiswa
* Jumlah Dosen          : 1 dosen
* Jumlah Tenaga Kependidikan : 0 tenaga kependidikan
* Jumlah Mata Kuliah     : 1 mata kuliah

Silahkan Pilih Salah Satu MenuMenu:
1. Tambah Mahasiswa
2. Tambah Dosen
3. Tambah Tenaga Kependidikan

```

```

5. Tampilkan semua Dosen
6. Tampilkan semua Tenaga Kependidikan
7. Tambah Mata Kuliah
8. Tambah Dosen mengisi Mata Kuliah
9. Tambah Mahasiswa di Mata Kuliah
10. Tampilkan Semua Mata Kuliah
8
'clear' is not recognized as an internal or external command,
operable program or batch file.
Mohon Tulis Nama Dosen yang akan ditambahkan : re
Mohon Tulis Nama Mata Kuliah : DD
Apakah Anda akan menjalankan atau melanjutkan program? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : y
'clear' is not recognized as an internal or external command,
operable program or batch file.
Selamat datang di Universitas Kembang Lambé Tunas Bangsa

Data Statistik :
* Jumlah Mahasiswa      : 0 mahasiswa
* Jumlah Dosen          : 1 dosen
* Jumlah Tenaga Kependidikan : 0 tenaga kependidikan
* Jumlah Mata Kuliah     : 1 mata kuliah

Silahkan Pilih Salah Satu MenuMenu:
1. Tambah Mahasiswa
2. Tambah Dosen
3. Tambah Tenaga Kependidikan
4. Tampilkan semua Mahasiswa
5. Tampilkan semua Dosen
6. Tampilkan semua Tenaga Kependidikan
7. Tambah Mata Kuliah
8. Tambah Dosen mengisi Mata Kuliah
9. Tambah Mahasiswa di Mata Kuliah
10. Tampilkan Semua Mata Kuliah
10
'clear' is not recognized as an internal or external command,
operable program or batch file.
Apakah Anda Ingin Melihat list Nama Dosen dan Mahasiswa yang ada dalam Mata Kuliah? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : y
1. Data Mata Kuliah dengan ID : Matkul1
   Nama Mata Kuliah          : DD
   Beban SKS Mata Kuliah ini : 3
   Data dari Dosen pada Mata Kuliah ini : DD:
       1.re (15763517) - Teknik
   Data dari Mahasiswa pada Mata Kuliah ini : DD :

Apakah Anda akan menjalankan atau melanjutkan program? Tekan y/Y untuk Ya dan n/N untuk Tidak (y/n) : █

```

Itulah penjelasan dari saya untuk tugas Pemrograman Lanjut tentang SIM sederhana bagian ke 2 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.