Using C++ In Creating Crud Applications

Aditya Aji Saputra

AAS Institute of Business Technology Indonesia
ajis6231@gmail.com
* Corresponding author

ABSTRACT

C ++ means computer programming language designed by Bjarne Stroustrup is a development of the origin of C language developed at Bell Labs (Dennis Ritchie). In the early 1970s, the language meant adding to the origin of the previous language, C. [1] C++ in object-oriented programming languages (OOP) there is the word group Library. Class library is a function designed to make it easier for programmers to produce software. In experiments, the CRUD Library group is divided into 4 parts, namely: group Create, class Read, group Update, group Delete. As a result of this research, the Library group can make it easier to make CRUD software.

KEYWORDS
C++
Class Library
CRUD
OOP

1. Introduction

Computer programming today is increasingly becoming an interest for people in all walks of life. Since the development of IT is increasingly accelerating, almost everything has been digitized and the use of applications / software is a common activity for all circles

Object Oriented Programming (OOP) is a method of creating programs (programming paradigm) using the concept of "objects" that have data and code. This data is available in the attribute or property form, as well as code in the mechanism form.

The advantages of C ++ are, faster performance process, protection of many function libraries and classes to allow the creation of macro software, is reusable that can be reused in other projects using header files and libraries, programming languages that are multi-paradigm or moving forward, writing short code compared to other programming languages, has conveyable properties that can be used for all types of computers, The language is generally ANSI, therefore C ++ can be used on various platforms, one of the advantages possessed is very potential If you can make graphic processor software using high quality, C ++ language compilers are mostly open source, more systematic so that they can support OOP (objected - oriented - programming.

2. Method

In this research, the author formed a CRUD event based on the topology that you can see in the picture table 1. 1
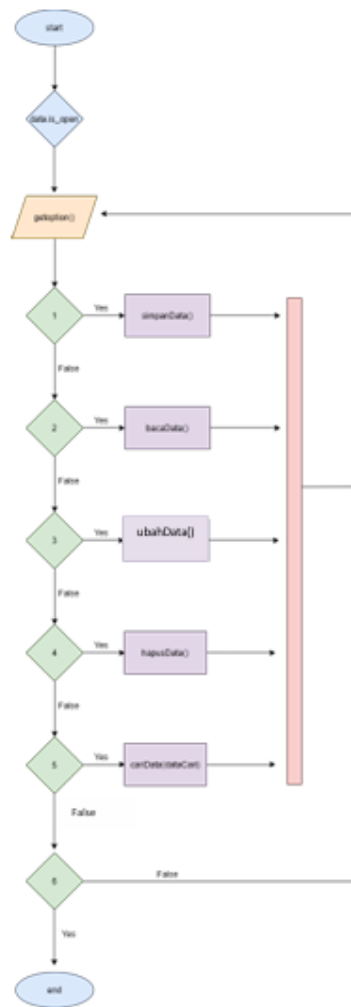
Figure 1. System Implementation Flow

Here we need a file header to perform operations against external archives. At the beginning of the program we will be presented a menu in the form like:

Figure 1. Data CRUD 1

Then we have to enter input based on the menu and will process the functions that we have set up. If we enter input with a value of 1, it will process the Save Data() function, similar to the image below:


```
Program CRUD data mahasiswa
=============================
1. Tambah data mahasiswa
2. Tampilkan data mahasiswa
3. Ubah data mahasiswa
4. Hapus data mahasiswa
5. Search data by NIM
6. Selesai
=============================
pilih [1-6]? : 1

Menambah data mahasiswa
NIM : 206
Nama : Rukman
Tugas : 80
UTS : 90
UAS : 100
```

Figure 2. Example Results 1

Code for the saveData() function:

```
void simpanData(){

GradesStudents grades;

nilai.inputNilai();

Save the object to a file

ofstream f;

f.open(namaFile, ios::binary|ios::app);

f.write( (char*) &nilai, sizeof(nilai));

f.close();

}
```

The process that happens to that function is, initially we enter our data in value.inputValue(); Then the data will be saved to a file using Ofstream with ios::binary mode to write in binary form, and ios::app to write at the end of the line. If we enter input with a value of 2, it processes the readData() function to display the data.

Figure 3. Data View 1

Code for the readData() function:

void bacaData(){

GradesStudents grades;

ifstream f;

f.open (namefile, ios:binary);

cout << "NIM\t" << "Nama\t" << "Tugas\t" << "UTS\t" << "UAS\t" << "Nilai;

cout <<End\t" << "Letter Value\t" <<endl;

while (f.read((char*)&nilai, sizeof(nilai))){

nilai.outputNilai();

}

f.close();

}

In this process, looping will be done with the condition that as long as the data is read to the end, then after the data is read it will be displayed. If we select three, it will process to replace the data. As shown below:

Figure 4. Change Data 1

Code for the ubahData() function:

```
void ubahData(int nimDicari){

GradesStudents grades;

fstream f;

f.open(namaFile, ios::in|ios::out);

while (f.read((char*) &nilai, sizeof(nilai))) {

if(nilai.getNim() == nimDicari){

Show data before the update

cout << "NIM\t" << "Nama\t" << "Tugas\t" << "UTS\t" << "UAS\t";

cout << "Final Value\t" << "Letter Value\t" <<endl;

nilai.outputNilai();

New data input

cout << "Please enter new data" << endl;

nilai.inputNilai();

int pos = -1 * sizeof(nilai);

f.seekp(pos, ios::cur);

f.write ( (char*) &nilai, sizeof(nilai));

break;

}

}
```

f.close();

}

In this process, initially the data is checked whether there is or not, if there will be displayed before being changed, then the new data is accommodated at the value.inputValue(), then the data is deleted according to the region from which we entered. If we enter the value 4 it will delete the data.
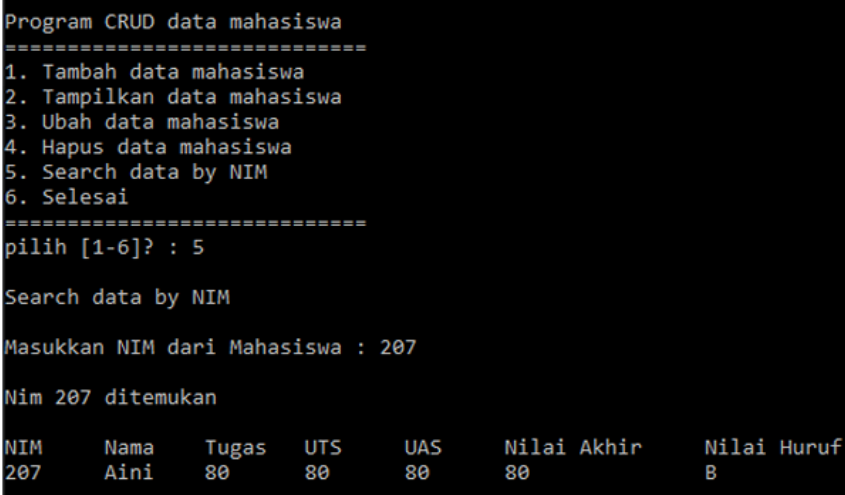

Figure 5. Clear Data 1

In this process, we need to use ifstream and ofstream. Here will be carried out the loping process with conditions during reading the data to completion. In looping there is a decision process, if the value of the NIM data derived in the index is not the same using the NIM we are looking for, then the original data will be written to the ad interim archive, here I use "tmp.dat" to hold new data, then the original data will be deleted, then the data "tmp.dat" will be renamed using the name of the original archive. If we enter input using the value five, it will display data according to NIM.

```
void hapusData(int nimDicari){

GradesStudents grades;

ifstream fi;

fi.open (namefile, ios:binary);

ofstream fo;

fo.open("tmp.dat", ios::out|ios::binary);

int meet = 0;

while(fi.read((char*) &nilai, sizeof(nilai))) {

if(nilai.getNim() != nimDicari){

fo.write((char*) &nilai, sizeof(nilai));

}

}
```

```
else {

}

fi.close();

fo.close();

ketemu = 1;

remove(namaFile);

rename("tmp.dat", FileName);

if (meet == 1){

cout << "Data with NIM " << nimSearch << " has been deleted!" << endl;

} else {

cout << "Data with NIM" << nimSearched << "Not found" << endl;

}

}
```


Figure 6. Searching Data 1

Code for the searchData() function;

```
void cariData(int nimDicari){

GradesStudent grades, grades found;

int meet = 0;

ifstream f;

f.open (namefile, ios:binary);

while(f.read((char*)&value, sizeof(value)){

if(nilai.getNim() == nimDicari){

nilaiketemu = nilai;
```

```
ketemu = 1;

break;

}

}

if (meet == 0){

}else {

}

f.close();

}
```

In this process we have a variable as a value for the decision later, here I have the variable int met with a value of 0 to be a marker if this current data has not been found. then looping occurs to check one by one in the data, whether the index of the NIM value matches the NIM we are looking for, If so, the original value of the variable found will be changed to 1, otherwise it will remain 0. then after looping, a decision occurs using the condition If the original value of the meeting is 0 it will notify that the data sought does not exist, If the value of the meeting is 1 it will display the data.

2. Results and Discussion

We can use functions in the main function using in combination with WHILE so that the program does not last one way, but can be used continuously as long as we want to use, and so that it runs following the command.

```
while(pilihan != FINISH){

    switch(pilihan){
        case CREATE:
            cout << "\nMenambah data mahasiswa" << endl;
            simpanData();
            break;
        case READ:
            cout << "\nTampilkan data mahasiswa" << endl;
            bacaData();
            break;
        case UPDATE:
            cout << "\nUbah data mahasiswa" << endl;
            cout << "\nMasukkan NIM dari Mahasiswa : ";
            cin >> dataCari;
            ubahData(dataCari);
            break;
        case DELETE:
            cout << "\nHapus data mahasiswa" << endl;
            cout << "\nMasukkan NIM dari Mahasiswa : ";
            cin >> dataCari;
            hapusData(dataCari);
            break;
        case SEARCH:
            cout << "\nSearch data by NIM" << endl;
            cout << "\nMasukkan NIM dari Mahasiswa : ";
            cin >> dataCari;
            cariData(dataCari);
            break;
        default:
            cout << "\nPilihan tidak ditemukan" << endl;
            break;
    }
```

Figure 7. Coding 1

in this program will create a CRUD (Create, Read, Update, Delete) program in the case of entering the value of student origin, in this method we can also form a program using other cases.

3. Conclusion

As a result of research and project tests, a CRUD program has been obtained that can be run on the Terminal or Commnd Promp. with the creation of several functions this library can make the process of making CRUD applications as simpler and faster because programmers do not need to take a long time for a code to be used repeatedly. This project can be a training material for reading and writing programs into files in the form of CRUD. The drawback of the program this time is the limited table made, here we only make tables for students. It is needed on the next occasion to be developed again, for example plus several other tables.

Reference

[1]   "C++ - Wikipedia Indonesian, the free encyclopedia." https://id.wikipedia.org/wiki/C%2B%2B (accessed Jul. 14, 2021).

[2]   N. G. A. P. Harry Saptarini, R. A. Hidayat, and P. I. Ciptayani, "AJARINCODE: WEB-BASED PROGRAMMING LANGUAGE LEARNING APPLICATION," Just TI (Journal of Science Terap. Techno. Information), vol. 10, no. 2, p. 21, Jul. 2019, doi: 10.46964/JUSTTI. V10I2.106.

[3]   "Understanding Object-Oriented Programming C++ | Duniailkom." https://www.duniailkom.com/tutorial-belajar-oop-c-plus-plus-pengertian object-oriented programming/ (accessed Jul. 14, 2021).

[4]   "The Disadvantages and Advantages That C++ Has - Josi Kie." https://josikie.com/kekurangan-dan-kelebihan-yang-dimiliki-c/ (accessed Jul. 14, 2021).

[5]   F. File, "File & Stream," vol. 2, no. 5, pp. 1–5.