

LAPORAN TUGAS BESAR
PEMROGRAMAN BERORIENTASI OBJEK
PROGRAM PERPUSTAKAAN RHEG



Disusun oleh :

| | |
|------------------------------|--------------------|
| Retno Wulan Sari | (201402024) |
| Muhammad Arif Fadhlán | (201402054) |
| Goklas Damendra Sitio | (201402102) |
| Harry Fauzan Hasibuan | (201402105) |

Diajukan sebagai syarat untuk memenuhi tugas besar mata kuliah
pemrograman berorientasi objek

Dosen Pengampu: Dani Gunawan, S.T., M.T.

UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER-TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNOLOGI INFORMASI

2020/2021

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan tugas besar ini dengan judul “Program Perpustakaan RHEG” Adapun tujuan dari penulisan laporan ini adalah untuk memenuhi tugas besar dari Dani Gunawan, S.T., M.T. selaku Dosen Pengampu Mata Kuliah Pemrograman Berorientasi Objek.

Kami selaku penulis mengucapkan terima kasih kepada Dani Gunawan, S.T., M.T. yang telah memberikan tugas besar ini sehingga kami dan para pembaca dapat menambah wawasan dan pengetahuan terhadap topik mengenai pemrograman berorientasi objek yang akan diterapkan dalam program yang kami buat.

Penulis memaklumi bahwa makalah ini masih banyak kesalahan dan kekurangan sehingga dengan kerendahan hati kami sebagai penulis menerima berbagai **kritikan** dan **masukan** dari Dani Gunawan, S.T., M.T. untuk kesempurnaan makalah ini.

Medan, 15 Juni 2021

Kelompok 1

DAFTAR ISI

| | |
|--------------------------------------|----|
| KATA PENGANTAR..... | I |
| DAFTAR ISI..... | II |
| BAB I..... | 1 |
| PENDAHULUAN | 1 |
| 1.1 Deskripsi Judul Tugas Besar..... | 1 |
| BAB II | 2 |
| PEMBAHASAN | 2 |
| 2.1 Setup Awal | 2 |
| 2.2 Tampilan | 3 |
| 2.3 Fitur-fitur | 5 |
| 2.4 Penjelasan Source Code..... | 7 |
| BAB III..... | 18 |
| PENUTUP..... | 18 |
| 3.1. Kesimpulan | 18 |
| 3.2. Saran..... | 18 |

BAB I

PENDAHULUAN

1.1 Deskripsi Judul Tugas Besar

Dalam tugas besar ini kami dari kelompok 1 membuat sebuah program berjudul “**Program Perpustakaan RHEG**” untuk memenuhi tugas besar mata kuliah Pemrograman Berorientasi Objek. Program ini dibuat dengan menggunakan konsep- konsep Pemrograman Berorientasi Objek, yaitu abstraksi, enkapsulasi, inheritance (pewarisan), database (file), dan polimorfisme. Program ini juga menggunakan library seperti fstream untuk menggunakan file sebagai database, vektor, sstream, dan conio.h.

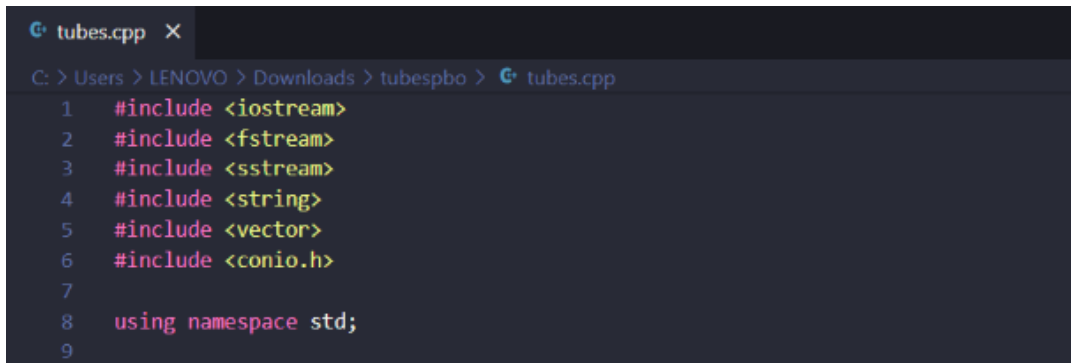
Program ini mempunyai beberapa fitur, yaitu penambahan buku, penyimpanan buku, dan pengembalian buku. Pada penambahan buku hanya bisa dilakukan oleh admin karena kami membuat 2 level pada program ini, yakni admin dan user.

BAB II

PEMBAHASAN

2.1 Setup Awal

1. Header



```
tubes.cpp X
C: > Users > LENOVO > Downloads > tubespbo > tubes.cpp
1  #include <iostream>
2  #include <fstream>
3  #include <sstream>
4  #include <string>
5  #include <vector>
6  #include <conio.h>
7
8  using namespace std;
9
```

File header adalah file yang berisi deklarasi, baik berupa konstanta, fungsi kelas, namespace, dan sebagainya. Dan harus dipanggil menggunakan directive `#include`. Pada program ini, kami menggunakan header sebagai berikut :

- `<iostream>`
Digunakan untuk menampilkan perintah `cin`, `cout`, dan `endl`.
- `<fstream>`
Digunakan untuk membuat dan menulis file.
- `<sstream>`
untuk menggunakan class `stringstream` untuk membentuk suatu buffer string.
- `<string>`
Untuk menjalankan perintah perintah tipe data string.
- `<vector>`
untuk menggunakan Standard Template Library `vector`.
- `<conio.h>`
Merupakan File Header yang berfungsi untuk menampilkan hasil antarmuka kepada pengguna.
- `using namespace std;`
Statement tersebut menunjukkan bahwa kita menggunakan Bahasa C++ standar di dalam kode program yang kita tulis.

2.2 Tampilan

1. Menu awal

Menu awal berisi pilihan register atau login

```
C:\Users\LENOVO\Downloads\tubespbo\tubes.exe
1. Register Akun
2. Login Akun
3. Exit
Masukkan pilihan anda : _
```

2. Register

Proses registrasi merupakan tahapan awal bagi user yang belum memiliki akun. Data akun yang berhasil melakukan registrasi akan masuk ke dalam dataAkun.csv.

```
C:\Users\LENOVO\Downloads\tubespbo\tubes.exe
=====
Form Registrasi
=====
Masukkan Username : resha
Masukkan Password : re123
Selamat! Akun anda telah diregistrasi!_
```

3. Login

- Login as user

Pada proses login user harus memasukkan nama dan password dengan benar. Jika password atau nama yang dimasukkan salah akan terjadi kesalahan login.

```
=====
Form Login
=====
Masukkan Username : resha
Masukkan Password : re12
Data tidak ditemukan!_
```

Tampilan login berhasil :

```
=====
Form Login
=====
Masukkan Username : resha
Masukkan Password : re123
Akun Anda telah diverifikasi!_
```

Jika berhasil login, user akan diarahkan ke menu utama aplikasi

- Login as admin

Pada proses login admin harus memasukkan nama dan password dengan benar. Jika password atau nama yang dimasukkan salah akan terjadi kesalahan login.

```

=====
Form Login
=====
Masukkan Username : Rene
Masukkan Password : admin22

Data tidak ditemukan!_

```

Tampilan login berhasil :

```

=====
Form Login
=====
Masukkan Username : Rene
Masukkan Password : admin2

Akun Anda telah diverifikasi!_

```

4. Menu utama

- Menu utama user

Jika berhasil login sebagai user, tampilan menu utama yang muncul adalah sebagai berikut. Pada menu utama terdapat pilihan pinjam buku atau pengembalian buku yang bisa di pilih oleh user.

```

C:\Users\LENOVO\Downloads\tubespbo\tubes.exe
1. Peminjaman Buku
2. Pengembalian
3. Exit

Masukkan pilihan anda : _

```

- Menu utama admin

```

1. Penambahan buku
2. Exit

Masukkan pilihan anda : _

```

Jika sudah berhasil login, admin dapat menambahkan buku.

2.3 Fitur-fitur

1. Peminjaman buku

```
Peminjaman Buku

=====
Judul          Harga          Jumlah
=====
Laskar Pelangi 80000          10
Bumi Manusia   120000         20
Noceur          75000           3
=====

Masukkan nama peminjam : Viki
Masukkan nama buku yang ingin dipinjam (Case Sensitive) : Bumi Manusia
Masukkan jumlah buku yang ingin dipinjam : 3_
```

```
Konfirmasi Pinjaman

=====
Nama peminjam      : Viki
Judul buku         : Bumi Manusia
Jumlah buku yang dipinjam : 3
Total Harga        : 360000

Apakah data Anda sudah benar (y/n) : y

Data telah disimpan!

Jangan lupa untuk mengembalikan buku ya!_
```

| | A | B | C | D |
|---|------|----------|---|---|
| 1 | Viki | Bumi Man | 3 | |
| 2 | | | | |
| 3 | | | | |

Pada tampilan peminjaman buku, akan ditampilkan data buku yang telah diinputkan oleh admin. User diminta untuk memasukkan nama peminjam, buku yang ingin dipinjam, dan jumlah buku yang ingin dipinjam, setelah itu akan keluar halaman konfirmasi pinjaman. Pada halaman konfirmasi pinjaman, jika data user diyakini benar, maka user berhasil meminjam buku tanpa ada batasan waktu dan data peminjam akan masuk ke file dataPeminjam.csv. Jika tidak maka user bisa menginputkan karakter 'n' dan akan Kembali ke menu user.

2. Pengembalian buku

```
Pengembalian Buku

=====

Masukkan nama peminjam : Viki_
```

```
Konfirmasi Pengembalian

=====
Nama peminjam      : Viki
Judul buku         : Bumi Manusia
Jumlah buku yang dipinjam : 3

Apakah data Anda sudah benar (y/n) : y

Data telah disimpan!

Terima kasih telah mengembalikan buku!_
```


| | A | B | C |
|---|---|---|---|
| 1 | | | |
| 2 | | | |

| | A | B | C |
|---|------------|--------|----|
| 1 | Laskar Pel | 80000 | 10 |
| 2 | Bumi Man | 120000 | 20 |
| 3 | Noceur | 75000 | 3 |
| 4 | | | |

Pada tampilan pengembalian buku, user diminta untuk memasukkan nama peminjam. setelah itu, akan keluar halaman konfirmasi pengembalian beserta data peminjam dan buku yang dipinjam. User diminta konfirmasi apakah data sudah benar. jika data user benar maka dan user menginputkan 'Y' atau 'y' maka pengembalian telah berhasil dan akan mengembalikan semua buku yang telah dipinjam. Setelah itu, data peminjam di dalam file dataPeminjam.csv akan dihapus dan jumlah buku di dalam file dataBuku.csv akan bertambah setelah dikembalikan.

3. Penambahan buku

```

=====
Penambahan Buku
=====
Masukkan Nama Buku : noceur
Harga Buku : 75000
Jumlah Buku : 3

Data buku berhasil ditambahkan!_

```

Pada halaman penambahan buku. Admin diminta untuk memasukkan judul buku, harga buku dan jumlah buku yang ingin ditambahkan. Selanjutnya buku yang ditambahkan akan masuk kedalam file dataBuku.csv.

| | A | B | C | D |
|---|------------|--------|----|---|
| 1 | Laskar Pel | 80000 | 10 | |
| 2 | Bumi Man | 120000 | 20 | |
| 3 | Noceur | 75000 | 3 | |
| 4 | | | | |
| 5 | | | | |

2.4 Penjelasan Source Code

- a. Class autentikasi
 1. Fungsi register

```
1  class Autentikasi {
2      private:
3          string usernameRegistrasi, passwordRegistrasi, usernameLogin, passwordLogin;
4
5      public:
6          void registrasi() {
7              system("cls");
8
9              string tempData, line;
10
11              cout << "======" << endl;
12              cout << "Form Registrasi" << endl;
13              cout << "======" << endl;
14              cout << "Masukkan Username : ";
15              cin.ignore();
16              getline(cin, usernameRegistrasi);
17              cout << "Masukkan Password : ";
18              istream& ignore (streamsize n = 1, int delim = EOF);
19              getline(cin, passwordRegistrasi);
20
21              tempData = usernameRegistrasi + "," + passwordRegistrasi + ",user";
22
23              ifstream file("dataAkun.csv", ios::in | ios::out);
24              if (file.is_open()) {
25                  getline(file, line, '\n');
26
27                  while (getline(file, line, '\n')) {
28                      if (tempData == line) {
29                          cout << "\nAkun sudah ada!";
30                          getch();
31                          break;
32                      } else {
33                          if (!file.eof()) {
34                              continue;
35                          } else {
36                              file.close();
37                              ofstream file("dataAkun.csv", ios::app);
38                              if (file.is_open()) {
39                                  file << "\n" << tempData;
40                                  cout << "\nSelamat! Akun anda telah diregistrasi!";
41                                  getch();
42
43                                  file.close();
44                                  break;
45                              } else {
46                                  cout << "\nDatabase tidak dapat dibuka";
47                                  getch();
48                              }
49                          }
50                      }
51                  }
52              } else {
53                  cout << "\nDatabase tidak dapat dibuka";
54                  getch();
55              }
56          }
57      }
```

Dibagian awal, kami membuat fungsi registrasi didalam class autentifikasi. User diminta untuk memasukkan username dan password, setelah itu inputan akan digabung. Setelah itu, akan dicek apakah data sudah ada di dalam file, jika tidak maka data akan dimasukkan kedalam file dataAkun.csv.

2. Fungsi login

```
1  string login() {
2      system("cls");
3
4      string level;
5
6      cout << "=====" << endl;
7      cout << "Form Login" << endl;
8      cout << "=====" << endl;
9      cout << "Masukkan Username : ";
10     cin.ignore();
11     getline(cin, usernameLogin);
12     cout << "Masukkan Password : ";
13     istream& ignore (streamsize n = 1, int delim = EOF);
14     getline(cin, passwordLogin);
15
16     string x = usernameLogin + "," + passwordLogin;
17
18     fstream file;
19     file.open("dataAkun.csv", ios::in);
20
21     vector<string> row;
22     string line, data;
23
24     while (getline(file, line)) {
25         row.clear();
26
27         istringstream s(line);
28         while (getline(s, data, ',')) {
29             row.push_back(data);
30         }
31
32         if (x == row[0] + "," + row[1]) {
33             if (row[2] == "admin") {
34                 cout << "\nAkun Anda telah diverifikasi!";
35                 getch();
36
37                 goto LoginAdmin;
38             } else if (row[2] == "user") {
39                 cout << "\nAkun Anda telah diverifikasi!";
40                 getch();
41
42                 goto LoginUser;
43             }
44         } else {
45             if (!file.eof()) {
46                 continue;
47             } else {
48                 goto GagalLogin;
49             }
50         }
51     }
52
53     LoginAdmin:
54     file.close();
55     level = "admin";
56     goto BerhasilLogin;
57
58     LoginUser:
59     file.close();
60     level = "user";
61     goto BerhasilLogin;
62
63     GagalLogin:
64     file.close();
65     level = "gagal";
66     goto BerhasilLogin;
67
68     BerhasilLogin:
69     return level;
70 }
```

Di fungsi login user diminta untuk memasukkan username dan password yang telah diregistrasi, setelah itu inputan akan digabung. Setelah itu, akan dicek apakah data sudah ada di dalam file, jika tidak maka akan muncul pemberitahuan, jika ada maka

akan di cek levelnya apakah admin atau user. Jika admin maka akan beralih halaman admin, jika user maka akan beralih ke halaman user.

b. Class penambahan

1. Fungsi tambahData

```
1  class Penambahan {
2      private:
3          int hargaBuku, banyakBuku;
4          string namaBuku;
5
6      public:
7          void tambahData() {
8              int hargaBuku, JumlahBuku;
9              string namaBuku;
10
11              fstream fout;
12              fout.open("dataBuku.csv", ios::out | ios::app);
13
14              if (fout.is_open()) {
15                  cout << "Masukkan Nama Buku : ";
16                  cin.ignore();
17                  getline(cin, namaBuku);
18                  cout << "Harga Buku : ";
19                  cin >> hargaBuku;
20                  cout << "Jumlah Buku : ";
21                  cin >> JumlahBuku;
22
23                  fout << namaBuku << "," << hargaBuku << "," << JumlahBuku << "\n";
24                  cout << "\nData buku berhasil ditambahkan!";
25                  getch();
26              } else {
27                  cout << "Database buku tidak ditemukan!";
28                  getch();
29              }
30          }
31  };
```

Class penambahan berisi fungsi tambahData() yang berfungsi untuk menambahkan data buku, dan jumlah buku yang ingin ditambahkan oleh admin.

c. Class peminjaman

1. Fungsi tampilDataBuku

Fungsi tampilDataBuku() berfungsi untuk menampilkan judul buku, jumlah buku, dan harga buku yang dapat dipinjam, berdasarkan data file dataBuku.csv

2. Fungsi pinjamBuku

```

1  template <class T>
2      T getValue(string s, int col)
3      {
4          T val;
5          int counter = 0;
6          string str;
7          stringstream ss(s);
8
9          while (counter++ < col) {
10             getline(ss, str, ',');
11         }
12
13         stringstream sstr(str);
14         sstr >> val;
15         return val;
16     }
17
18     void pinjamBuku() {
19         fstream fin, fout, dataBukuBaru;
20         fout.open("dataPeminjam.csv", ios::out | ios::app);
21         dataBukuBaru.open("dataBukuBaru.csv", ios::out);
22         if (!dataBukuBaru.is_open()) {
23             dataBukuBaru.close();
24             dataBukuBaru.open("dataBukuBaru.csv", ios::out);
25         }
26
27         fin.open("dataBuku.csv", ios::in);
28         vector<string> row;
29         string line, data;
30
31         if (fin.is_open()) {
32             isiData:
33             cout << "\nMasukkan nama peminjam : ";
34             cin.ignore();
35             getline(cin, namaPeminjam);
36             cout << "Masukkan nama buku yang ingin dipinjam (Case Sensitive) : ";
37             istream& ignore (streamsize n = 1, int delim = EOF);
38             getline(cin, namaBuku);
39             cout << "Masukkan jumlah buku yang ingin dipinjam : ";
40             cin >> banyakBuku;
41
42             system("cls");
43
44             while (getline(fin, line)) {
45                 row.clear();
46
47                 stringstream s(line);
48                 while (getline(s, data, ',')) {
49                     row.push_back(data);
50                 }
51
52                 if (row[0] == namaBuku) {
53                     int totalHarga = getValue<int>(line, 2);
54                     int jumlah = getValue<int>(line, 3);
55                     totalHarga *= banyakBuku;
56                     jumlah -= banyakBuku;
57
58                     cout << "===== << endl;
59                     cout << "Konfirmasi Pinjaman << endl;
60                     cout << "===== << endl;
61                     cout << "Nama peminjam : " << namaPeminjam << endl;
62                     cout << "Judul buku : " << namaBuku << endl;
63                     cout << "Jumlah buku yang dipinjam : " << banyakBuku << endl;
64                     cout << "Total Harga : " << totalHarga << endl;
65
66                     cout << "\nApakah data Anda sudah benar (y/n) : ";
67                     cin >> konfirmasiPinjaman;
68
69                     if (konfirmasiPinjaman == 'Y' | konfirmasiPinjaman == 'y') {
70                         fout << namaPeminjam << "," << namaBuku << "," << banyakBuku << endl;
71                         dataBukuBaru << row[0] << "," << row[1] << "," << jumlah << endl;
72                         cout << "\nData telah disimpan!\n << endl;
73                         cout << "Jangan lupa untuk mengembalikan buku ya!";
74                         getch();
75                     } else if (konfirmasiPinjaman == 'N' | konfirmasiPinjaman == 'n') {
76                         cout << "\nPemesanan dibatalkan!";
77                         getch();
78
79                         goto selesai;
80                     } else {
81                         cout << "\nInput yang anda masukkan salah!";
82                         getch();
83
84                         system("cls");
85                         tampilDataBuku();
86
87                         goto isiData;
88                     }
89                 } else {
90                     if (!fin.eof()) {
91                         for (int i = 0; i < row.size() - 1; i++) {
92                             dataBukuBaru << row[i] << ",";
93                         }
94                         dataBukuBaru << row[row.size() - 1] << "\n";
95                     }
96                 }
97
98                 if (fin.eof()) {
99                     break;
100                 }
101             }
102         }
103         fin.close();
104         fout.close();
105         dataBukuBaru.close();
106
107         remove("dataBuku.csv");
108         rename("dataBukuBaru.csv", "dataBuku.csv");
109
110         selesai:
111         fin.close();
112         fout.close();
113         dataBukuBaru.close();
114     }
115 };

```

Untuk meminjam buku, user diminta mengisi nama peminjam, judul buku yang mau dipinjam, dan jumlah buku yang mau dipinjam. Selanjutnya program akan menampilkan data yang sudah di isi oleh user, jika user mengkonfirmasi data tersebut sudah benar maka peminjaman buku berhasil. Namun jika user mengkonfirmasi ada kesalahan pada data maka peminjaman akan dibatalkan.

d. Class pengembalian

1. fungsi kembalikanBuku

```
1  class Pengembalian {
2      private:
3          int banyakBuku;
4          string namaBuku, namaPeminjam;
5          char konfirmasiPengembalian;
6
7      public:
8          template <class T>
9              T getValue(string s, int col)
10             {
11                 T val;
12                 int counter = 0;
13                 string str;
14                 stringstream ss(s);
15
16                 while (counter++ < col) {
17                     getline(ss, str, ',');
18                 }
19
20                 stringstream sstr(str);
21                 sstr >> val;
22                 return val;
23             }
24
25         void kembalikanBuku() {
26             fstream fin, fout, dataPeminjamBaru, dataBukuBaru;
27             fout.open("dataPeminjam.csv", ios::in | ios::out | ios::app);
28             dataPeminjamBaru.open("dataPeminjamBaru.csv", ios::out);
29             if (!dataPeminjamBaru.is_open()) {
30                 dataPeminjamBaru.close();
31                 dataPeminjamBaru.open("dataPeminjamBaru.csv", ios::out);
32             }
33             fin.open("dataBuku.csv", ios::in);
34             dataBukuBaru.open("dataBukuBaru.csv", ios::out);
35             if (!dataBukuBaru.is_open()) {
36                 dataBukuBaru.close();
37                 dataBukuBaru.open("dataBukuBaru.csv", ios::out);
38             }
39
40             vector<string> rowPeminjam;
41             vector<string> rowBuku;
42             string linePeminjam, lineBuku, dataPeminjam, dataBuku;
```

```

1  if (fout.is_open()) {
2      isiData:
3          cout << "\nMasukkan nama peminjam : ";
4          cin.ignore();
5          getline(cin, namaPeminjam);
6
7          while (getline(fout, linePeminjam)) {
8              rowPeminjam.clear();
9
10             stringstream s(linePeminjam);
11             while (getline(s, dataPeminjam, ',')) {
12                 rowPeminjam.push_back(dataPeminjam);
13             }
14
15             if (rowPeminjam[0] == namaPeminjam) {
16                 int jumlahPinjaman = getValue<int>(linePeminjam, 3);
17
18                 while (getline(fin, lineBuku)) {
19                     rowBuku.clear();
20
21                     stringstream str(lineBuku);
22                     while (getline(str, dataBuku, ',')) {
23                         rowBuku.push_back(dataBuku);
24                     }
25
26                     if (rowPeminjam[1] == rowBuku[0]) {
27                         banyakBuku = getValue<int>(lineBuku, 3);
28                         banyakBuku += jumlahPinjaman;
29
30                         system("cls");
31
32                         cout << "===== " << endl;
33                         cout << "Konfirmasi Pengembalian" << endl;
34                         cout << "===== " << endl;
35                         cout << "Nama peminjam          : " << namaPeminjam << endl;
36                         cout << "Judul buku           : " << rowPeminjam[1] << endl;
37                         cout << "Jumlah buku yang dipinjam : " << jumlahPinjaman << endl;
38
39                         cout << "\nApakah data Anda sudah benar (y/n) : ";
40                         cin >> konfirmasiPengembalian;
41
42                         if (konfirmasiPengembalian == 'Y' | konfirmasiPengembalian == 'y') {
43                             dataBukuBaru << rowBuku[0] << "," << rowBuku[1] << "," << banyakBuku << endl;
44                             cout << "\nData telah disimpan!\n" << endl;
45                             cout << "Terima kasih telah mengembalikan buku!";
46                             getch();
47                         } else if (konfirmasiPengembalian == 'N' | konfirmasiPengembalian == 'n') {
48                             cout << "\nPengembalian dibatalkan!";
49                             getch();
50
51                             goto selesai;
52                         } else {
53                             cout << "\nInput yang anda masukkan salah!";
54                             getch();
55
56                             system("cls");
57
58                             goto isiData;
59                         }
60                     } else {
61                         if (!fin.eof()) {
62                             for (int i = 0; i < rowBuku.size() - 1; i++) {
63                                 dataBukuBaru << rowBuku[i] << ",";
64                             }
65                             dataBukuBaru << rowBuku[rowBuku.size() - 1] << "\n";
66                         }
67                     }
68
69                     if (fin.eof()) {
70                         break;
71                     }
72                 } else {
73                     if (!fout.eof()) {
74                         for (int i = 0; i < rowPeminjam.size() - 1; i++) {
75                             dataPeminjamBaru << rowPeminjam[i] << ",";
76                         }
77                         dataPeminjamBaru << rowPeminjam[rowPeminjam.size() - 1] << "\n";
78                     }
79                 }
80             }
81
82             if (fout.eof()) {
83                 break;
84             }
85         }
86     }
87     fin.close();
88     fout.close();
89     dataBukuBaru.close();
90     dataPeminjamBaru.close();
91
92     remove("dataBuku.csv");
93     remove("dataPeminjam.csv");
94     rename("dataBukuBaru.csv", "dataBuku.csv");
95     rename("dataPeminjamBaru.csv", "dataPeminjam.csv");
96
97     selesai:
98     fin.close();
99     fout.close();
100    dataBukuBaru.close();
101    dataPeminjamBaru.close();
102 }
103 };

```


Untuk mengembalikan buku, user diminta mengisi nama peminjam. Selanjutnya program akan mengecek apakah data peminjam ada atau tidak, kemudian akan mengecek data buku yang dipinjam peminjam ada atau tidak. Jika ada maka akan menampilkan konfirmasi data berupa data yang sudah di isi oleh user, jika user mengkonfirmasi data tersebut sudah benar maka pengembailan buku berhasil dan data peminjam akan dihapuskan serta jumlah buku akan bertambah sesuai buku yang telah dipinjam.

e. class menuOpsiAdmin

1. Fungsi menu

```
1  class MenuOpsiAdmin : public Penambahan {
2      private:
3          int pilihanMenuOpsiAdmin;
4
5      public:
6          void menu() {
7              system("cls");
8
9              cout << "1. Penambahan buku" << endl;
10             cout << "2. Exit" << endl;
11             cout << "\nMasukkan pilihan anda : ";
12
13             cin >> pilihanMenuOpsiAdmin;
14             pilihanOpsiAdmin(pilihanMenuOpsiAdmin);
15         }
16
17         void pilihanOpsiAdmin(int pilihan) {
18             while (!(pilihanMenuOpsiAdmin < 1) || !(pilihanMenuOpsiAdmin > 2)) {
19                 switch(pilihanMenuOpsiAdmin){
20                     case 1:
21                         system("cls");
22                         cout << "======" << endl;
23                         cout << "Penambahan Buku" << endl;
24                         cout << "======" << endl;
25                         tambahData();
26                         menu();
27                         break;
28
29                     case 2:
30                         exit(0);
31                         break;
32
33                     default:
34                         cout << "\nPilihan Anda tidak valid!";
35                         menu();
36                         break;
37                 }
38             }
39         }
40     };
```

Pada class menuOpsiAdmin terdapat fungsi menu() yang berfungsi untuk menampilkan menu utama jika login sebagai admin, dimana admin dapat menambahkan buku ketika menginputkan angka 1 untuk mengakses opsi pertama. Buku yang ditambahkan oleh admin akan masuk ke dalam file dataBuku.csv

f. class menuOpsidUser

1. Fungsi menu

```
1  class MenuOpsidUser : public Peminjaman, Pengembalian {
2      private:
3          int pilihanMenuOpsidUser;
4          enum option{PEMINJAMAN = 1, PENGEMBALIAN, KELUAR, BREAK};
5
6      public:
7          void menu() {
8              system("cls");
9
10             cout << "1. Peminjaman Buku" << endl;
11             cout << "2. Pengembalian Buku" << endl;
12             cout << "3. Exit" << endl;
13             cout << "\nMasukkan pilihan anda : ";
14
15             cin >> pilihanMenuOpsidUser;
16             pilihanOpsidUser(pilihanMenuOpsidUser);
17         }
18
19         void pilihanOpsidUser(int pilihan) {
20             while (!(pilihanMenuOpsidUser < 1) || !(pilihanMenuOpsidUser > 3)) {
21                 switch(pilihanMenuOpsidUser){
22                     case PEMINJAMAN:
23                         system("cls");
24                         cout << "=====" << endl;
25                         cout << "Peminjaman Buku" << endl;
26                         cout << "=====" << endl;
27                         tampilDataBuku();
28                         menu();
29                         break;
30
31                     case PENGEMBALIAN:
32                         system("cls");
33                         cout << "=====" << endl;
34                         cout << "Pengembalian Buku" << endl;
35                         cout << "=====" << endl;
36                         kembalikanBuku();
37                         menu();
38                         break;
39
40                     case KELUAR:
41                         exit(0);
42                         break;
43
44                     default:
45                         cout << "\nPilihan Anda tidak valid!";
46                         menu();
47                         break;
48                 }
49             }
50         }
51     };
```

Pada class MenuOpsidUser terdapat Fungsi void menu() berfungsi untuk menampilkan menu utama program jika login sebagai user dan terdapat menu peminjaman buku dan pengembalian buku.

g. Fungsi main

```
1  int main() {
2      fstream data;
3      Autentikasi auth;
4      MenuOpsiUser menuUser;
5      MenuOpsiAdmin menuAdmin;
6
7      cekFileDataAkun(data);
8      cekFileDataBuku(data);
9
10     menuAwal:
11     system("cls");
12     int pilihan;
13     string level;
14
15     cout << "1. Register Akun" << endl;
16     cout << "2. Login Akun" << endl;
17     cout << "3. Exit" << endl;
18     cout << "\nMasukkan pilihan anda : ";
19     cin >> pilihan;
20
21     switch (pilihan) {
22         case 1:
23             auth.registrasi();
24             main();
25             break;
26
27         case 2:
28             level = auth.login();
29             if (level == "admin") {
30                 menuAdmin.menu();
31             } else if (level == "user") {
32                 menuUser.menu();
33             } else if (level == "gagal") {
34                 cout << "\nData tidak ditemukan!";
35                 getch();
36
37                 goto menuAwal;
38             }
39             break;
40
41         case 3:
42             exit(0);
43             break;
44
45         default:
46             cout << "\nPilihan Anda tidak valid!";
47             getch();
48
49             goto menuAwal;
50             break;
51     }
52
53     return 0;
54 }
```

Pada fungsi main, terdapat opsi awal, yakni login dan register, serta keluar dari program. Jika user memilih opsi 1 maka akan beralih ke fungsi registrasi di class

Autentikasi, jika memilih opsi 2 maka akan beralih ke fungsi login di class Autentikasi.

h. Fungsi cekFileDataAkun dan Fungsi cekFileDataBuku



```
1 void cekFileDataAkun(fstream &data) {
2     data.open("dataAkun.csv", ios::out | ios::in | ios::binary);
3
4     if (!data.is_open()) {
5         data.close();
6         data.open("dataAkun.csv", ios::trunc | ios::out | ios::in | ios::binary);
7     }
8     data.close();
9 }
10
11 void cekFileDataBuku(fstream &data) {
12     data.open("dataBuku.csv", ios::out | ios::in | ios::binary);
13
14     if (!data.is_open()) {
15         data.close();
16         data.open("dataBuku.csv", ios::trunc | ios::out | ios::in | ios::binary);
17     }
18     data.close();
19 }
```

Fungsi ini digunakan untuk mengecek apakah file dataAkun dan file dataBuku ada atau tidak, jika tidak maka akan dibuatkan kedua file tersebut otomatis, jika tidak maka file tersebut tidak akan dibuat.

BAB III

PENUTUP

3.1. Kesimpulan

Pemrograman berorientasi objek adalah pemograman yang menggunakan objek dalam programnya. Di sini kami menggunakan pemrograman berorientasi objek dalam program kami untuk mempermudah dalam memprogram. Kami juga menggunakan beberapa library untuk mempermudah kami, seperti vector, iostream, sstream, conio.h, dan string.

3.2. Saran

Kami menyadari laporan dan program kami masih memiliki banyak kekurangan dan kelemahan. Oleh karena itu, kami bersedia untuk menerima kritikan dan saran dari pembaca agar kami dapat memperbaiki dan mengevaluasi kekurangan kami di lain waktu yang akan datang.