

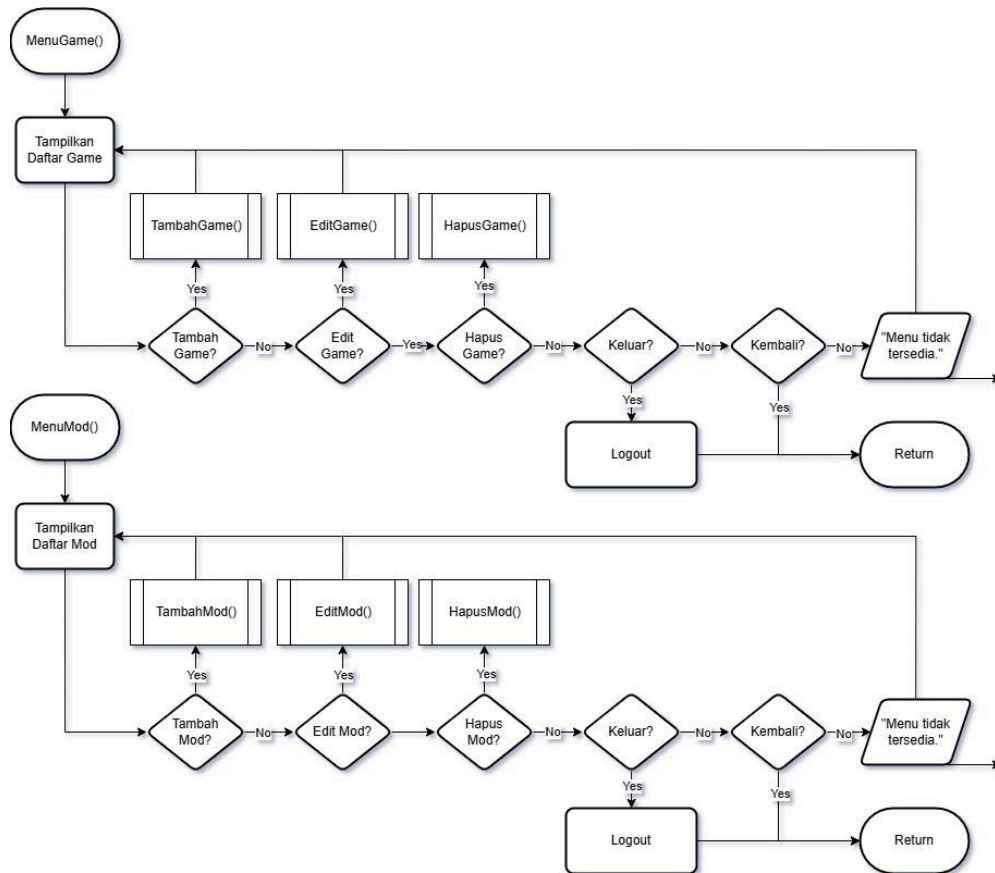
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



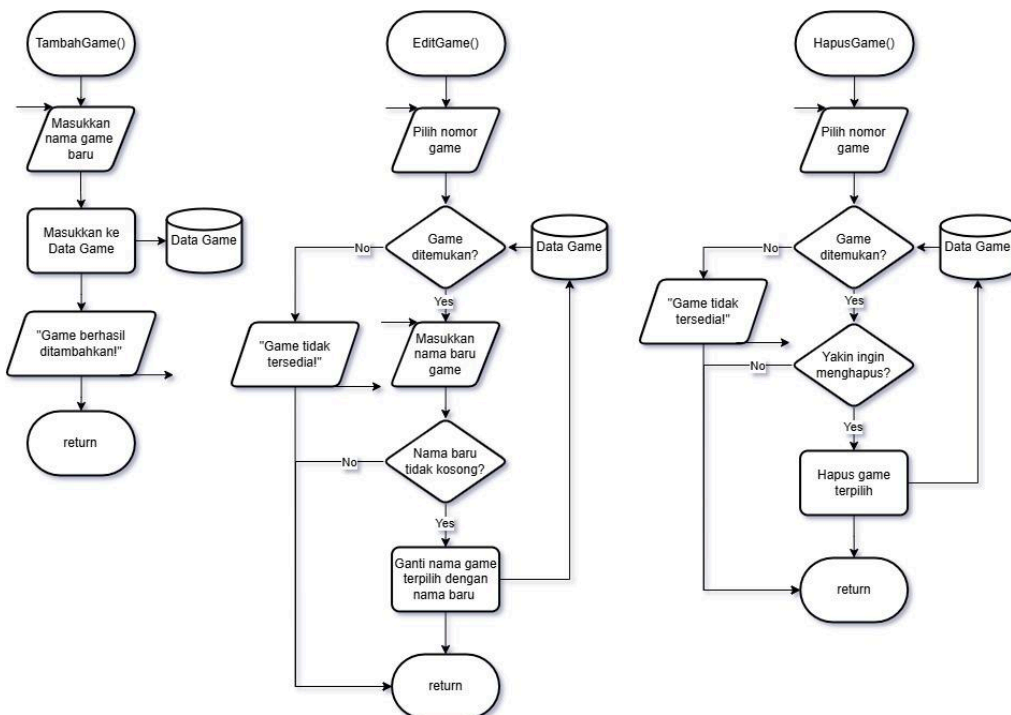
Disusun oleh:
Much. Trigusni Hermawan (2409106060)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

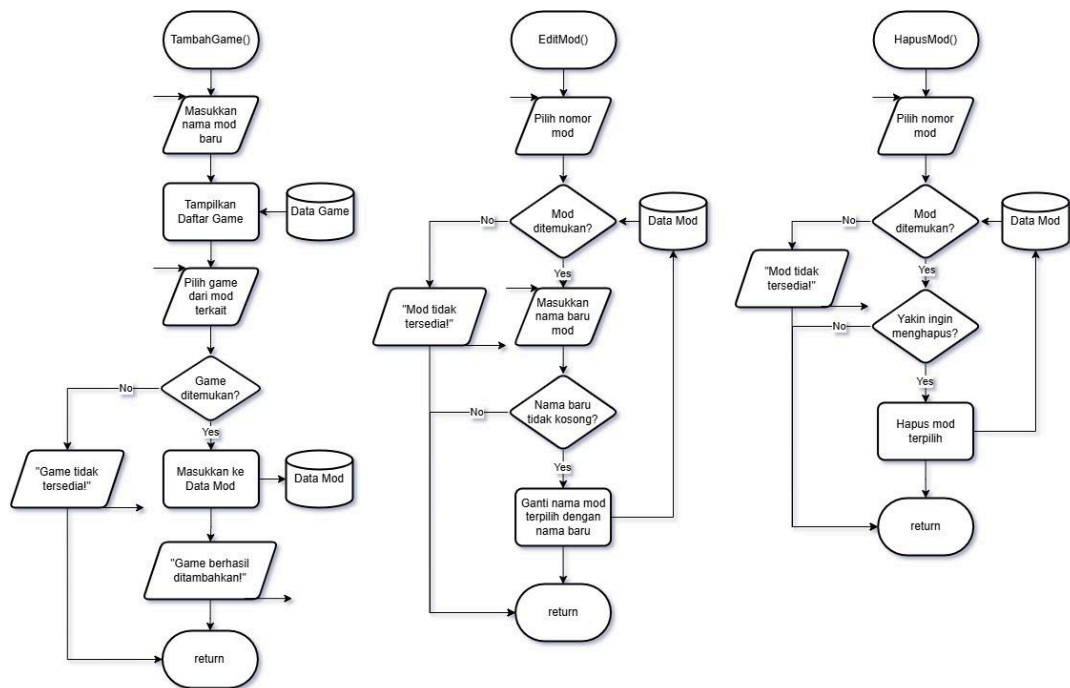
1. Flowchart



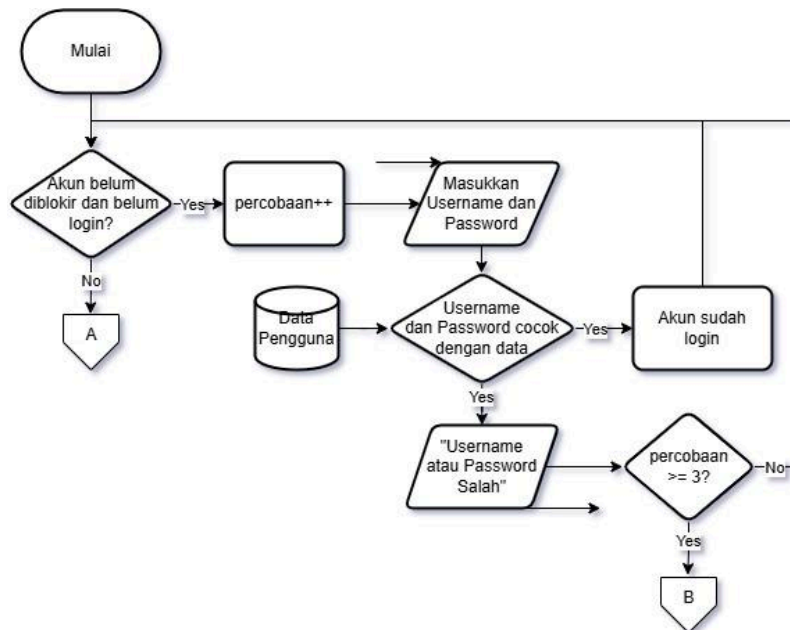
Gambar 1.1 Fungsi Halaman Menu



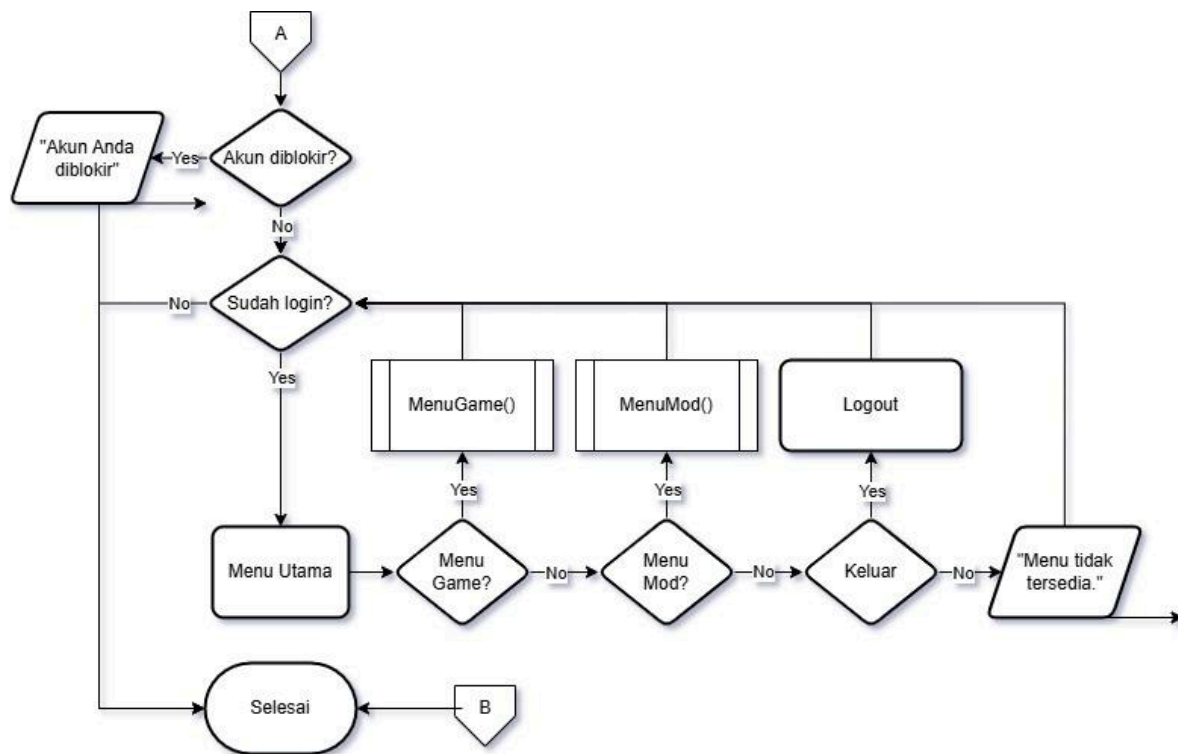
Gambar 1.2 Fungsi Menu Game



Gambar 1.3 Fungsi Menu Mod



Gambar 1.4 Main Flow Bagian 1



Gambar 1.5 Main Flow Bagian 2

2. Analisis Program

2.1 Deskripsi Singkat Program

Program Pengelolaan *Video Game Modding* ini dibuat bertujuan sebagai sebuah wadah bagi para komunitas penggemar sebuah *video game* membagikan dan mencari konten lebih dari para penggemar lainnya. Program ini diharapkan dapat memudahkan para *modder* menjadi lebih terbantu untuk membagikan kreasi *mod* buatannya dan begitu juga para *player* yang ingin dengan mudah mencari sekumpulan *mod* untuk dimainkan.

2.2 Penjelasan Alur & Algoritma

1. Program dimulai pada halaman login yang meminta username dan password,
2. Pengguna memasukkan nama panggilan sebagai username dan nim sebagai password masuk ke dalam program,
3. Jika pengguna salah memasukkan username atau password sebanyak tiga kali, maka program dihentikan,

4. Pengguna masuk ke menu utama setelah berhasil memasukkan username dan password.
5. Selama di menu utama, pengguna dapat memilih untuk melihat menu *game* atau *mod*,
6. Jika pengguna memilih menu *game*, maka program akan menampilkan daftar game yang tersedia dalam program. Selama di menu *game*, pengguna dapat memilih menu untuk “Tambah Game”, “Edit Game”, dan “Hapus Game”. Pada menu ini pengguna juga dapat kembali ke Menu Utama atau langsung keluar dari program.
7. Jika pengguna memilih menu “Tambah Game”, maka program akan meminta input nama *game* baru dan memasukkannya ke dalam daftar *game*.
8. Jika pengguna memilih menu “Edit Game”, maka program akan meminta input pilihan nomor game yang berdasarkan tabel. Jika pengguna memasukkan nomor game yang tersedia, program akan meminta nama baru untuk game yang dipilih. Jika pengguna mengisi nama baru maka *game* yang dipilih pengguna akan di-*update* menjadi nama baru yang diisi oleh pengguna, jika pengguna tidak mengisi nama baru mod, maka akan membiarkan bawaan nama *game* yang dipilih.
9. Jika pengguna memilih menu “Hapus Game”, maka program akan meminta input pilihan nomor *game* yang berdasarkan tabel. Jika pengguna memasukkan nomor game yang tersedia, program akan meminta konfirmasi penghapusan, dan jika pengguna setuju, maka *game* pada nomor yang dipilih akan dihapus,
10. Setelah menyelesaikan pengisian formulir dari “Tambah Game”, “Edit Game”, dan “Hapus Game”, pengguna akan diarahkan kembali ke menu *game*.
11. Jika pengguna kembali ke menu utama dan pengguna memilih menu *mod*, maka program akan menampilkan daftar *mod* yang tersedia dalam program. Selama di menu *mod*, pengguna dapat memilih menu untuk “Upload Mod”, “Edit Mod”, dan “Hapus Mod”. Pada menu ini pengguna juga dapat kembali ke Menu Utama atau langsung keluar dari program.
12. Jika pengguna memilih menu “Tambah Mod”, maka program akan meminta input nama *mod* baru dan memilih *game* dari mod yang dibuatnya.
13. Jika pengguna memilih menu “Edit Mod”, maka program akan meminta input pilihan nomor *mod* yang berdasarkan tabel. Jika pengguna memasukkan nomor game yang tersedia, program akan meminta nama baru untuk *mod* yang dipilih. Jika pengguna mengisi nama baru maka *mod* yang dipilih pengguna akan di-*update* menjadi nama baru *mod* yang diisi oleh pengguna, jika pengguna tidak mengisi nama baru mod, maka akan membiarkan bawaan nama *mod* yang dipilih.

14. Jika pengguna memilih menu “Hapus Mod”, maka program akan meminta input pilihan nomor *mod* yang berdasarkan tabel. Jika pengguna memasukkan nomor *mod* yang tersedia, program akan meminta konfirmasi penghapusan, dan jika pengguna setuju, maka *mod* pada nomor yang dipilih akan dihapus,
15. Setelah menyelesaikan pengisian formulir dari “Tambah Mod”, “Edit Mod”, dan “Hapus Mod”, pengguna akan diarahkan kembali ke menu *mod*.
16. Pengguna memilih pilihan keluar untuk menghentikan program.

3. Source Code

A. Sistem Login

Pada awal program, pengguna akan diarahkan ke sistem login untuk melanjutkan ke menu program. Data yang harus diberikan dalam proses login adalah username dan password. Maksimal percobaan hingga program pengguna dihentikan adalah sebanyak 3 kali.

Source Code:

```
// Data Login
string inputUsername = "";
string inputPassword = "";
int batasPercobaan = 3;
int percobaan = 0;
do
{
    percobaan++;
    cout << endl;
    cout << "Username\t: ";
    getline(cin, inputUsername);
    cout << "Password\t: ";
    getline(cin, inputPassword);
```

```

    if (inputPassword != password || inputUsername != username)
    {
        cout << "Username atau Password salah!" << endl;
        getline(cin, _temp);
    }
    else
    {
        cout << "Login berhasil!" << endl;
        isLogin = true;
        getline(cin, _temp);
    }
} while (percobaan < batasPercobaan && !isLogin);

```

B. Menu Utama

Menu utama menampilkan fitur utama program yaitu daftar *Game* dan *Mod*. Pengguna dapat salah satu menu untuk melihat masing-masing dari sistem pengelolaannya.

Source Code:

```

// Menu Utama
cout << "Menu Utama" << endl;
cout <<
"_____ " << endl;
cout << "1 > Games" << endl;
cout << "2 > Mods" << endl;
cout <<
"_____ " << endl;
cout << "Q > Keluar" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

if (_temp == "1")
{
    menu = "1";
}
else if (_temp == "2")
{
    menu = "2";
}
else if (_temp == "Q" || _temp == "q")
{
    isLogin = false;
}
else

```

```
{
    cout << "Menu tidak ditemukan!" << endl;
    getline(cin, _temp);
}
```

C. Menu Game

Di “Menu Game”, program menampilkan daftar game yang tersedia dalam bentuk tabel. Program juga menyediakan fitur untuk menambahkan, memperbarui, dan menghapus *game*.

Source Code:

```
// Games
cout << "Games" << endl;
cout <<
"_____
_____
" << endl;
cout << "C > Tambah Game | U > Edit Game | D > Hapus Game" << endl;
cout <<
"_____
_____
" << endl;
if (gameCount > 0)
{
    cout << left << setw(6) << "No."
        << setw(40) << "Game"
        << endl;
    cout <<
"_____
_____
" << endl;

    for (int i = 0; i < gameCount; i++)
    {
        cout << left << setw(4) << i + 1 << setw(2) << ">"
            << setw(40) << games[i][1]
            << endl;
    }
}
else
{
    cout << "Belum ada data game!" << endl;
}
cout <<
"_____
_____
" << endl;
cout << "Q > Keluar | B > Kembali" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);
```


D. Tambah Game

Fitur “Tambah Game” memungkinkan program untuk menambahkan *game* baru ke dalam daftar game.

Source Code:

```
// Tambah Game
if (gameCount <= MAX_DATA)
{
    cout <<
    "
    " << endl;

    cout << "Tambah Game" << endl;
    cout <<
    "
    " << endl;

    cout << "Nama Game\t: ";
    getline(cin, _temp);
    games[gameCount][0] = to_string(gameCount + 1);
    games[gameCount][1] = _temp;
    gameCount++;
    cout << "Game berhasil ditambahkan!" << endl;
    getline(cin, _temp);
}
else
{
    cout << "Game penuh! Tidak dapat menambah game lagi." << endl;
    getline(cin, _temp);
}
```

E. Edit Game

Fitur “Edit Game” memungkinkan program untuk memperbarui nama game yang sudah terdaftar.

Source Code:

```
// Edit Game
int _selectedIndex = -1;

cout << "Pilih Game\t: ";
getline(cin, _temp);
for (int i = 0; i < gameCount; i++)
{
    if (to_string(i + 1) == _temp)
```

```

    {
        _selectedIndex = i;
        break;
    }
}

if (_selectedIndex >= 0)
{
    cout <<
    "
    " << endl;

    cout << "Edit Game \"" << games[_selectedIndex][1] << "\"\" << endl;
    cout <<
    "
    " << endl;

    cout << "Nama Game Baru\t: ";
    getline(cin, _temp);
    games[_selectedIndex][1] = _temp != "" ? _temp : games[_selectedIndex][1];
    cout << "Game berhasil diperbarui!" << endl;
    getline(cin, _temp);
}
else
{
    cout << "Game tidak ditemukan!" << endl;
    getline(cin, _temp);
}
}

```

F. Hapus Game

Fitur “Hapus Game” memungkinkan program untuk menghapus suatu game yang telah terdaftar.

Source Code:

```

// Hapus Game
int _selectedIndex = -1;

cout << "Pilih Game\t: ";
getline(cin, _temp);
for (int i = 0; i < gameCount; i++)
{
    if (to_string(i + 1) == _temp)
    {
        _selectedIndex = i;
        break;
    }
}
}

```

```

if (_selectedIndex >= 0)
{
    cout << endl;
    cout << "Apakah Anda yakin ingin menghapus \"" << games[_selectedIndex][1]
<< "\"? (Y/N)\t: ";
    getline(cin, _temp);

    if (_temp == "Y" || _temp == "y")
    {
        for (int i = _selectedIndex; i < gameCount - 1; i++)
        {
            games[i][0] = games[i + 1][0];
            games[i][1] = games[i + 1][1];
        }
        gameCount--;
        cout << "Game berhasil dihapus!" << endl;
        getline(cin, _temp);
    }
    else
    {
        cout << "Game dibatalkan dihapus!" << endl;
        getline(cin, _temp);
    }
}
else
{
    cout << "Game tidak ditemukan!" << endl;
    getline(cin, _temp);
}
}

```

G. Menu Mod

Di “Menu Mod”, program menampilkan daftar mod yang tersedia dalam bentuk tabel. Program juga menyediakan fitur untuk menambahkan, memperbarui, dan menghapus *mod*.

Source Code:

```

// Mods
cout << "Mods" << endl;
cout <<
"_____
_____
" << endl;
cout << "C > Upload Mod | U > Edit Mod | D > Hapus Mod" << endl;
cout <<
"_____
_____
" << endl;
if (modCount > 0)
{

```

```

cout << left << setw(6) << "No."
    << setw(40) << "Nama Mod"
    << setw(40) << "Game"
    << setw(20) << "Creator"
    << endl;
cout <<
"_____
_____ " << endl;

for (int i = 0; i < modCount; i++)
{
    string game = "";
    for (int j = 0; j < gameCount; j++)
    {
        if (mods[i][1] == games[j][0])
        {
            game = games[j][1];
            break;
        }
    }

    cout << left << setw(4) << i + 1 << setw(2) << ">"
        << setw(40) << mods[i][2]
        << setw(40) << game
        << setw(20) << mods[i][3]
        << endl;
}
}
else
{
    cout << "Belum ada data mod!" << endl;
}
cout <<
"_____
_____ " << endl;
cout << "Q > Keluar | B > Kembali" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

```

H. Tambah Mod

Fitur “Tambah Game” memungkinkan program untuk menambahkan *game* baru ke dalam daftar game.

Source Code:

```
// Upload Mod
```

```

if (modCount < MAX_DATA)
{
    string tempName = "";
    int tempIndexGame = -1;
    cout <<
    "
    _____" << endl;

    cout << "Upload Mod" << endl;
    cout <<
    "
    _____" << endl;

    cout << "Nama Mod\t: ";
    getline(cin, tempName);
    cout << endl;
    for (int i = 0; i < gameCount; i++)
    {
        cout << i + 1 << ". " << games[i][1] << endl;
    }
    cout << "Pilih Game\t: ";
    getline(cin, _temp);
    cout << endl;

    if (_temp == "")
        continue;

    for (int i = 0; i < gameCount; i++)
    {
        if (to_string(i + 1) == _temp)
        {
            tempIndexGame = i;
            break;
        }
    }

    if (tempIndexGame >= 0)
    {
        mods[modCount][0] = to_string(modCount + 1);
        mods[modCount][1] = games[tempIndexGame][0];
        mods[modCount][2] = tempName;
        mods[modCount][3] = username;
        modCount++;
        cout << "Mod berhasil diupload!" << endl;
        getline(cin, _temp);
    }
    else
    {
        cout << "Game tidak ditemukan!" << endl;
        getline(cin, _temp);
    }
}
else

```

```
{
    cout << "Mod penuh! Tidak dapat mengupload mod lagi." << endl;
    getline(cin, _temp);
}
```

I. Edit Mod

Fitur “Edit Mod” memungkinkan program untuk menambahkan *mod* baru ke dalam daftar mod.

Source Code:

```
// Edit Mod
int _selectedIndex = -1;

cout << "Pilih Mod\t: ";
getline(cin, _temp);
for (int i = 0; i < modCount; i++)
{
    if (to_string(i + 1) == _temp)
    {
        _selectedIndex = i;
        break;
    }
}

if (_selectedIndex >= 0)
{
    cout <<
    "_____ " << endl;

    cout << "Edit Mod \"< mods[_selectedIndex][2] << "\"< endl;
    cout <<
    "_____ " << endl;

    cout << "Nama Mod Baru\t: ";
    getline(cin, _temp);
    if (sizeof(_temp) > 37)
    {
        cout << "Nama Mod terlalu panjang!" << endl;
        getline(cin, _temp);
        continue;
    }
    cout << endl;

    mods[_selectedIndex][2] = _temp != "" ? _temp : mods[_selectedIndex][2];
    cout << "Game berhasil diperbarui!" << endl;
    getline(cin, _temp);
```

```

}
else
{
    cout << "Mod tidak ditemukan!" << endl;
    getline(cin, _temp);
}

```

J. Hapus Mod

Fitur “Hapus Mod” memungkinkan program untuk menambahkan *mod* baru ke dalam daftar *mod*.

Source Code:

```

// Hapus Mod
int _selectedIndex = -1;

cout << "Pilih Mod\t: ";
getline(cin, _temp);
for (int i = 0; i < modCount; i++)
{
    if (to_string(i + 1) == _temp)
    {
        _selectedIndex = i;
        break;
    }
}

if (_selectedIndex >= 0)
{
    cout << endl;
    cout << "Apakah Anda yakin ingin menghapus \"" << mods[_selectedIndex][2] <<
    "\"? (Y/N)\t: ";
    getline(cin, _temp);

    if (_temp == "Y" || _temp == "y")
    {
        for (int i = _selectedIndex; i < modCount - 1; i++)
        {
            mods[i][0] = mods[i + 1][0];
            mods[i][1] = mods[i + 1][1];
            mods[i][2] = mods[i + 1][2];
            mods[i][3] = mods[i + 1][3];
        }
        modCount--;
        cout << "Mod berhasil dihapus!" << endl;
        getline(cin, _temp);
    }
}

```

```

else
{
    cout << "Mod dibatalkan dihapus!" << endl;
    getline(cin, _temp);
}
}
else
{
    cout << "Mod tidak ditemukan!" << endl;
    getline(cin, _temp);
}
}

```

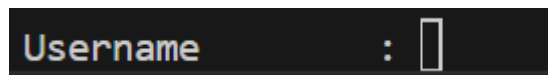
4. Uji Coba dan Hasil Output

4.1 Uji Coba

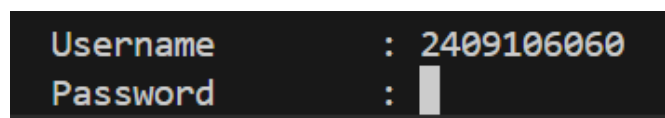
1. Login Berhasil.
2. Login Gagal
3. Menambah, mengedit, dan menghapus game.
4. Menambah mod, mengedit game dan mengecek kembali mod

4.2 Hasil Output

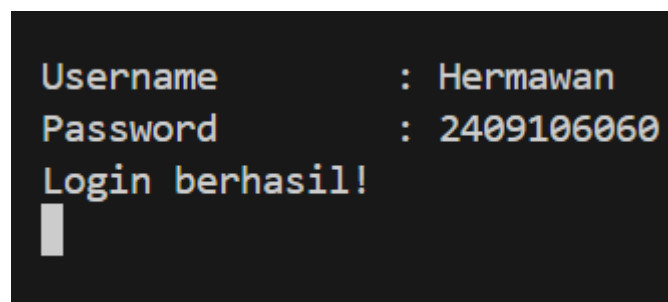
A. Login Berhasil



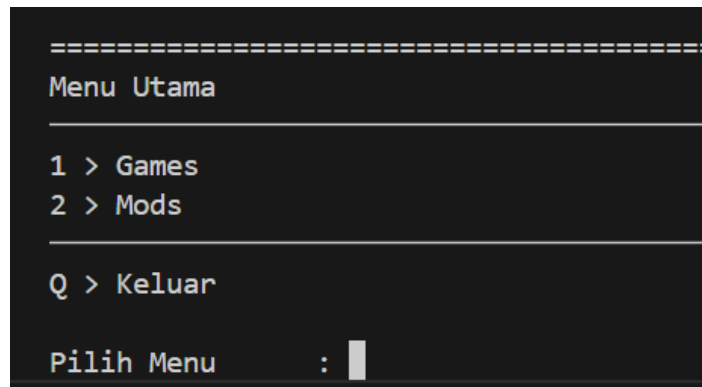
Gambar 4.1 Program meminta input username



Gambar 4.2 Program meminta input password setelah username

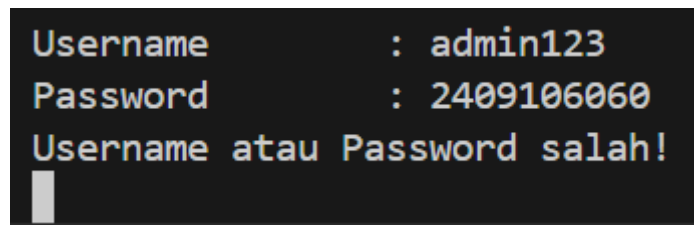


Gambar 4.3 Login Berhasil dengan username dan password yang benar

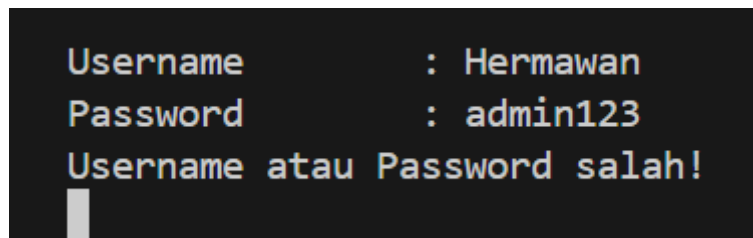


Gambar 4.4 Program diarahkan ke Menu Utama setelah login berhasil

B. Login Gagal



Gambar 4.5 Percobaan login pertama gagal dengan username salah



Gambar 4.6 Percobaan login kedua gagal dengan password salah

```
Username      : admin123
Password      : 2409106060
Username atau Password salah!

Username      : Hermawan
Password      : admin123
Username atau Password salah!

Username      : admin123
Password      : admin123
Username atau Password salah!

Program dihentikan...!
PS D:\Kuliah (Git Repository)\Praktikum-APL\praktikum-apl>
```

Gambar 4.7 Percobaan ketiga login program dihentikan

C. Menambah, Mengedit, dan Menghapus Game

```
=====
Menu Utama
-----
1 > Games
2 > Mods
-----
Q > Keluar

Pilih Menu   : 
```

Gambar 4.8 Program berhasil melakukan login dan berada di Menu Utama

```
=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game
-----
No.   Game
-----
1    > Stardew Valley
2    > Undertale
3    > Terraria
4    > Fallout 4
5    > Skyrim
-----
Q > Keluar | B > Kembali
Pilih Menu      : 
```

Gambar 4.9 Program masuk ke Menu Game

```
Tambah Game
-----
Nama Game      : Lethal Company
Game berhasil ditambahkan!

```

Gambar 4.10 Program memilih “Tambah Game” dan berhasil menambahkan game baru

```
=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game
-----
No.   Game
-----
1    > Stardew Valley
2    > Undertale
3    > Terraria
4    > Fallout 4
5    > Skyrim
6    > Lethal Company
-----
Q > Keluar | B > Kembali
Pilih Menu      : 
```

Gambar 4.11 Program menampilkan Menu Game dengan game baru pada nomor 6

```

Pilih Menu      : u
Pilih Game      : 6

Edit Game "Lethal Company"

Nama Game Baru  : Limbus Company
Game berhasil diperbarui!

```

Gambar 4.12 Program memilih Edit Game pada nomor 6 dan berhasil mengedit game baru

```

=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game

No.   Game
-----
1  > Stardew Valley
2  > Undertale
3  > Terraria
4  > Fallout 4
5  > Skyrim
6  > Limbus Company

Q > Keluar | B > Kembali

Pilih Menu      : 

```

Gambar 4.13 Program kembali ke Menu Game dengan nama baru pada game nomor 6

```

Pilih Menu      : d
Pilih Game      : 4

Apakah Anda yakin ingin menghapus "Fallout 4"? (Y/N) : y
Game berhasil dihapus!

```

Gambar 4.14 Program memilih hapus game nomor 4 dan berhasil menghapus

```

=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game
-----
No.   Game
-----
1    > Stardew Valley
2    > Undertale
3    > Terraria
4    > Skyrim
5    > Limbus Company
-----
Q > Keluar | B > Kembali
Pilih Menu : 

```

Gambar 4.14 Program kembali ke menu game dan hanya menampilkan 6 game

D. Menambah Mod, Memperbarui Game, dan Mengecek Mod

```

=====
Menu Utama
-----
1 > Games
2 > Mods
-----
Q > Keluar
Pilih Menu : 

```

Gambar 4.15 Program berhasil melakukan login dan berada di Menu Utama

```

=====
Mods
-----
C > Upload Mod | U > Edit Mod | D > Hapus Mod
-----
No.   Nama Mod                                Game                Creator
-----
1    > Undertale Bed Lump                     Undertale            TheMaxine
2    > Calamity Mod                           Terraria             Fabsol
3    > Stardew Valley Expanded                 Stardew Valley       FlashShifter
-----
Q > Keluar | B > Kembali
Pilih Menu : 

```

Gambar 4.16 Program memilih “Menu Mod” dan menampilkan daftar mod yang tersedia

```
Pilih Menu      : c
-----
Upload Mod
-----
Nama Mod       : SMAPI

1. Stardew Valley
2. Undertale
3. Terraria
4. Fallout 4
5. Skyrim
Pilih Game     : 1

Mod berhasil diupload!
```

Gambar 4.17 Program memilih “Upload Mod” dan berhasil menambahkan mod baru

```
=====
Mods
-----
C > Upload Mod | U > Edit Mod | D > Hapus Mod
-----


| No. | Nama Mod                  | Game           | Creator      |
|-----|---------------------------|----------------|--------------|
| 1   | > Undertale Bed Lump      | Undertale      | TheMaxine    |
| 2   | > Calamity Mod            | Terraria       | Fabsol       |
| 3   | > Stardew Valley Expanded | Stardew Valley | FlashShifter |
| 4   | > SMAPI                   | Stardew Valley | Hermawan     |


-----
Q > Keluar | B > Kembali
Pilih Menu    : 
```

Gambar 4.18 Program kembali ke menu mod dan dengan menampilkan mod baru

```
=====
Menu Utama
-----
1 > Games
2 > Mods
-----
Q > Keluar

Pilih Menu    : 
```

Gambar 4.19 Program kembali ke menu utama

```
=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game
-----
No.   Game
-----
1    > Stardew Valley
2    > Undertale
3    > Terraria
4    > Fallout 4
5    > Skyrim
-----
Q > Keluar | B > Kembali
Pilih Menu      : █
```

Gambar 4.20 Program memilih “Menu Mod”

```
=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game
-----
No.   Game
-----
1    > Sprout Valley
2    > Undertale
3    > Terraria
4    > Fallout 4
5    > Skyrim
-----
Q > Keluar | B > Kembali
Pilih Menu      : █
```

Gambar 4.21 Program mengedit “Stardew Valley” menjadi “Sprout Valley”

```
=====
Menu Utama
-----
1 > Games
2 > Mods
-----
Q > Keluar
Pilih Menu      : █
```

Gambar 4.22 Program kembali ke “Menu Utama”

```
=====
Mods

C > Upload Mod | U > Edit Mod | D > Hapus Mod

No.    Nama Mod                                Game                                Creator
-----
1  > Undertale Bed Lump                       Undertale                           TheMaxine
2  > Calamity Mod                             Terraria                           Fabsol
3  > Stardew Valley Expanded                 Sprout Valley                       FlashShifter
4  > SMAPI                                    Sprout Valley                       Hermawan

Q > Keluar | B > Kembali

Pilih Menu : 
```

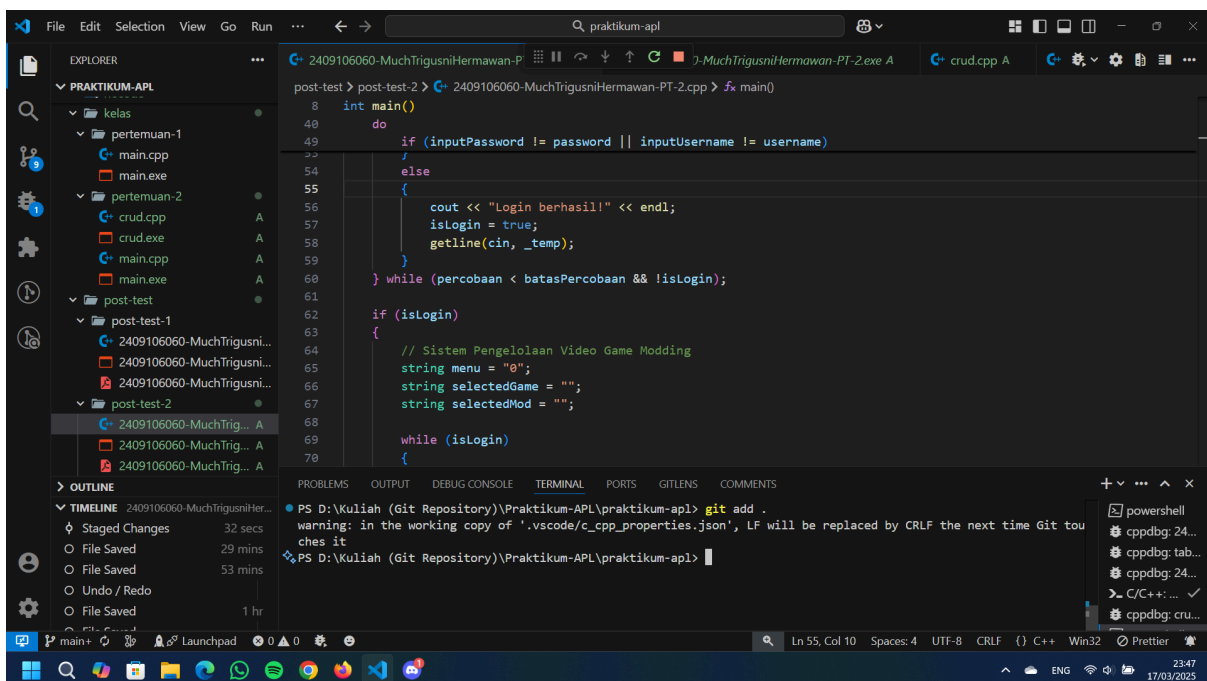
Gambar 4.23 Program mengecek kembali daftar mod pada game yang diperbarui

5. Langkah-Langkah Git

A. Git Init

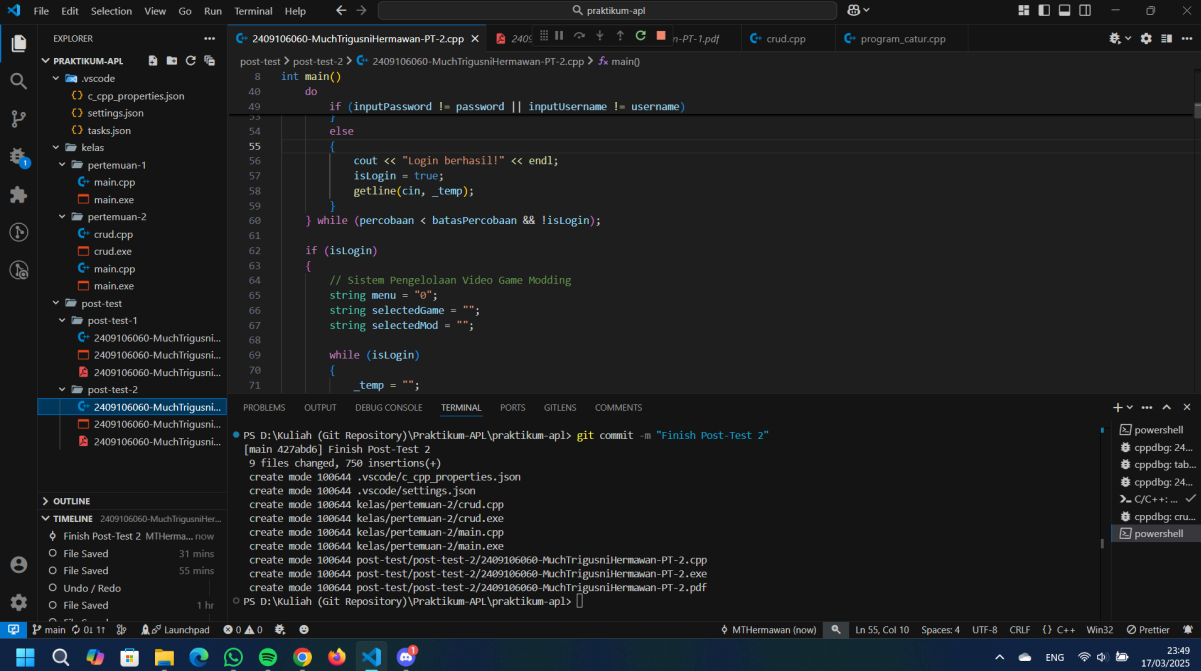
Karena fungsi git init untuk membuat repository kosong pertama kali, sementara repository telah dibuat pada saat post-test pertama, maka tidak perlu dilakukan lagi.

B. Git Add



Menambahkan semua perubahan file saat ini ke commit berikutnya dengan “git add .”

C. Git Commit



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a project structure with folders like 'praktikum-apl', 'kelas', and 'post-test'. The main editor window shows a C++ file named '2409106060-MuchTrigunihermawan-PT-2.cpp' with the following code:

```
8 int main()
48 do
49 if (inputPassword != password || inputUsername != username)
54 else
55 {
56 cout << "Login berhasil!" << endl;
57 islogin = true;
58 getline(cin, _temp);
59 }
60 while (percobaan < batasPercobaan && !islogin);
61
62 if (islogin)
63 {
64 // Sistem Pengelolaan Video Game Modding
65 string menu = "0";
66 string selectedGame = "";
67 string selectedMod = "";
68
69 while (islogin)
70 {
71 _temp = "";
```

The Terminal panel at the bottom shows the command prompt with the following output:

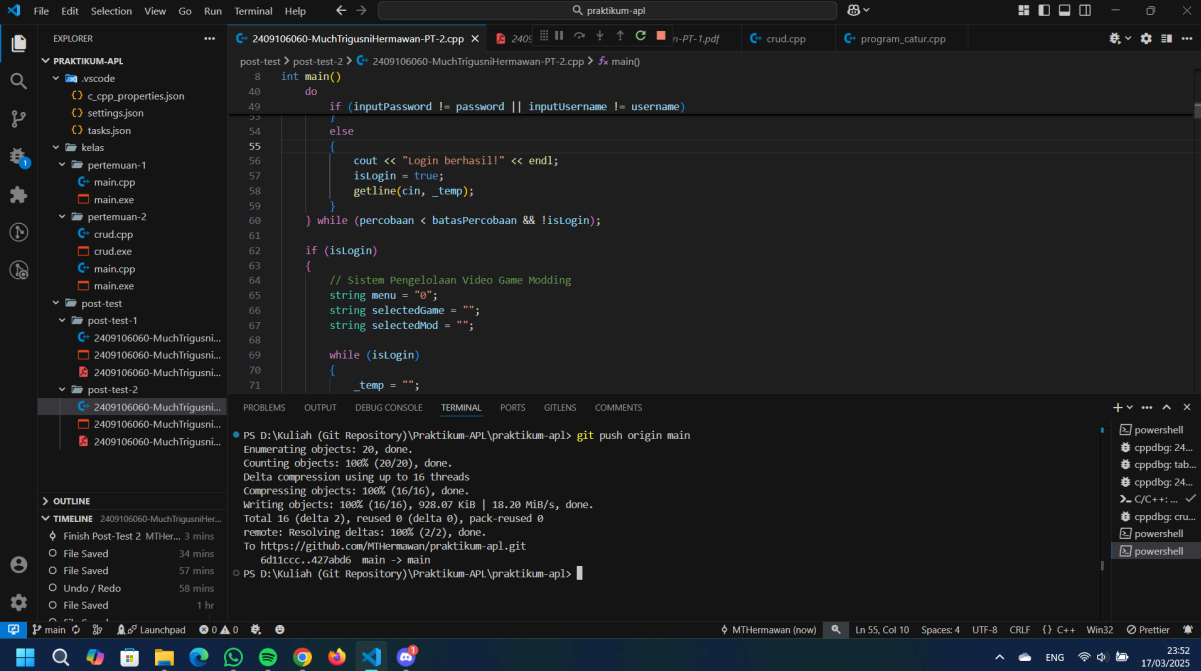
```
PS D:\Kuliah (git Repository)\Praktikum-APL\praktikum-apl> git commit -m "Finish Post-Test 2"
[main 427ab6] Finish Post-Test 2
9 files changed, 750 insertions(+)
create mode 100644 .vscode/c_cpp_properties.json
create mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan-2/crud.cpp
create mode 100644 kelas/pertemuan-2/crud.exe
create mode 100644 kelas/pertemuan-2/main.cpp
create mode 100644 kelas/pertemuan-2/main.exe
create mode 100644 post-test/post-test-2/2409106060-MuchTrigunihermawan-PT-2.cpp
create mode 100644 post-test/post-test-2/2409106060-MuchTrigunihermawan-PT-2.exe
create mode 100644 post-test/post-test-2/2409106060-MuchTrigunihermawan-PT-2.pdf
PS D:\Kuliah (git Repository)\Praktikum-APL\praktikum-apl>
```

Membuat checkpoint progres pada branch main dengan git commit dengan pesan “Finish Post-Test-2”.

D. Git Remote

Sama seperti git init, git remote memberi tujuan server repository online, sementara hal tersebut sudah dilakukan pada post-test pertama.

D. Git Push



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a project structure for 'PRAKTIKUM-APL' with folders like 'kelas', 'pertemuan-1', 'pertemuan-2', 'post-test', and 'post-test-2'. The main editor window shows a C++ file '2409106060-MuchTrigunihHermawan-PT-2.cpp' with code for a login system. The terminal window at the bottom shows the output of the 'git push origin main' command, indicating a successful push to the GitHub repository.

```
8 int main()
48 do
49 if (inputPassword != password || inputUsername != username)
54 else
55 {
56 cout << "Login berhasil!" << endl;
57 islogin = true;
58 getline(cin, _temp);
59 }
60 while (percobaan < batasPercobaan && !islogin);
61
62 if (islogin)
63 {
64 // Sistem Pengelolaan Video Game Modding
65 string menu = "0";
66 string selectedGame = "";
67 string selectedMod = "";
68
69 while (islogin)
70 {
71 _temp = "";
```

```
PS D:\Kuliah (Git Repository)\Praktikum-APL\praktikum-apl> git push origin main
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 16 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 928.07 KiB | 18.20 MiB/s, done.
Total 16 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/MTHERmawan/praktikum-apl.git
 6d1ccc..427abdc main -> main
PS D:\Kuliah (Git Repository)\Praktikum-APL\praktikum-apl>
```

Dengan melakukan git push, perubahan file saat ini telah disinkronisasikan di Repository GitHub.