

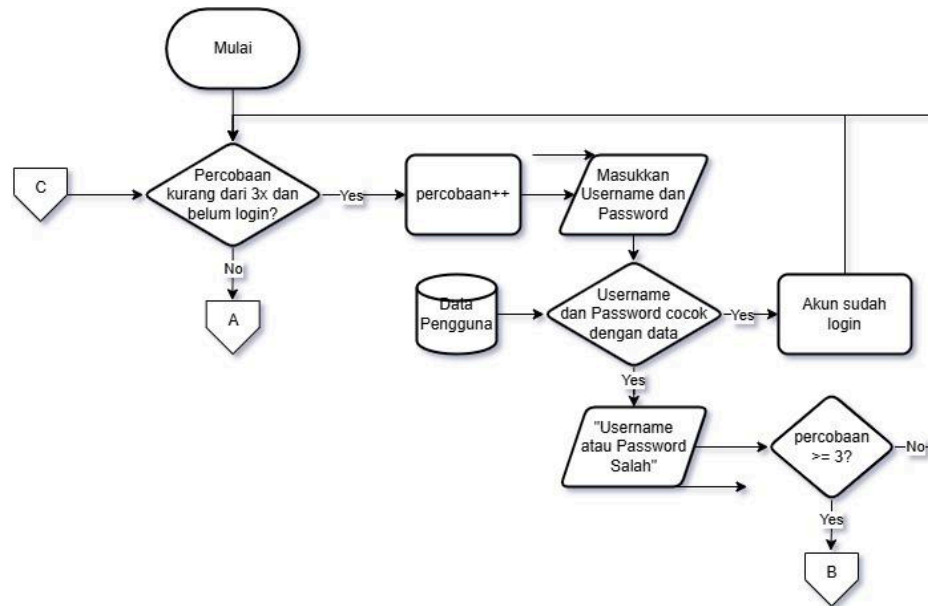
**LAPORAN PRAKTIKUM**  
**POSTTEST 3**  
**ALGORITMA PEMROGRAMAN LANJUT**



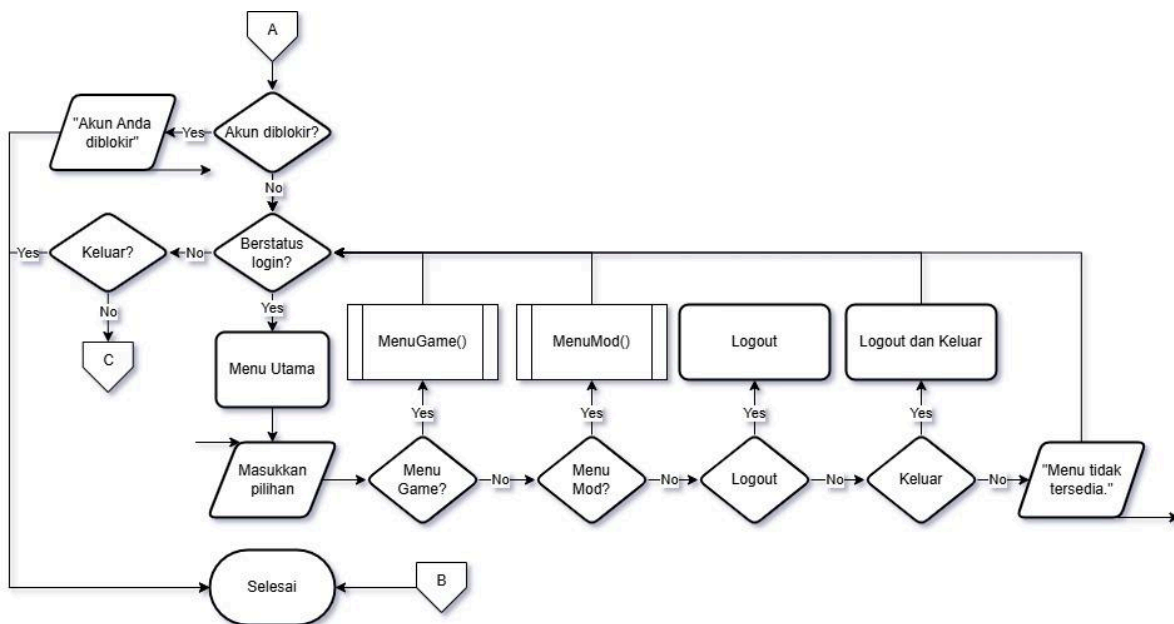
**Disusun oleh:**  
**Much. Trigusni Hermawan (2409106060)**  
**Kelas (B1 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

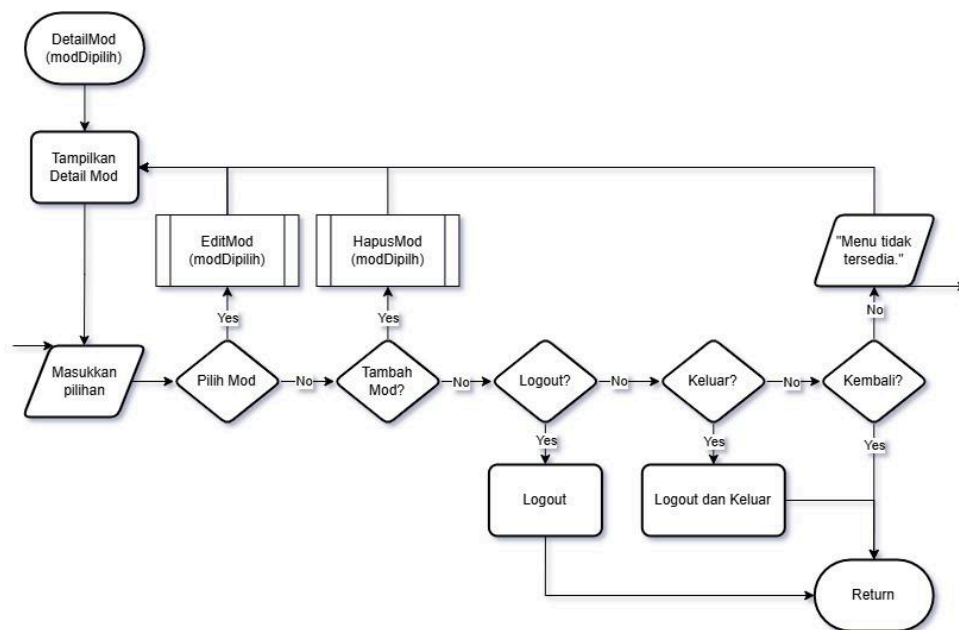
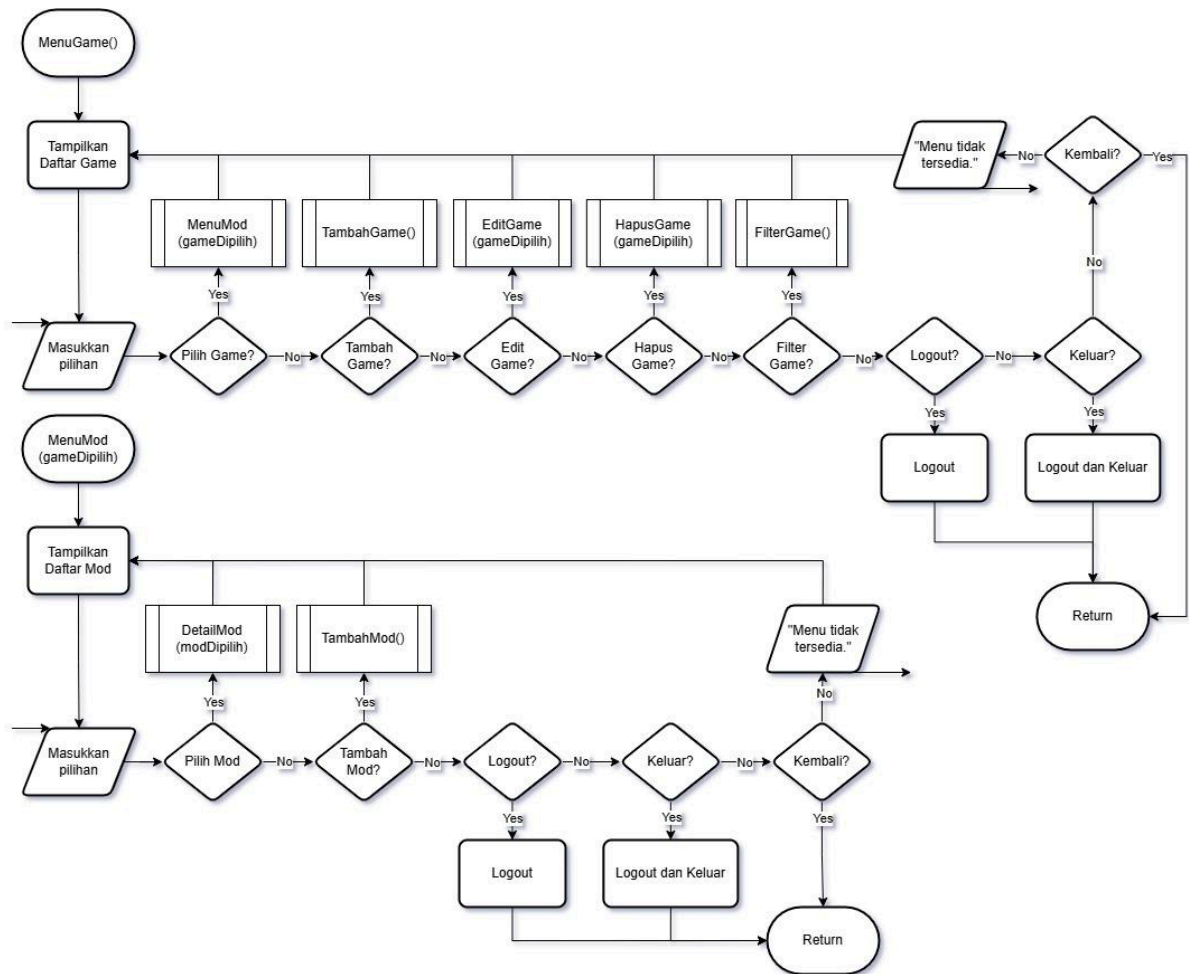
## 1. Flowchart

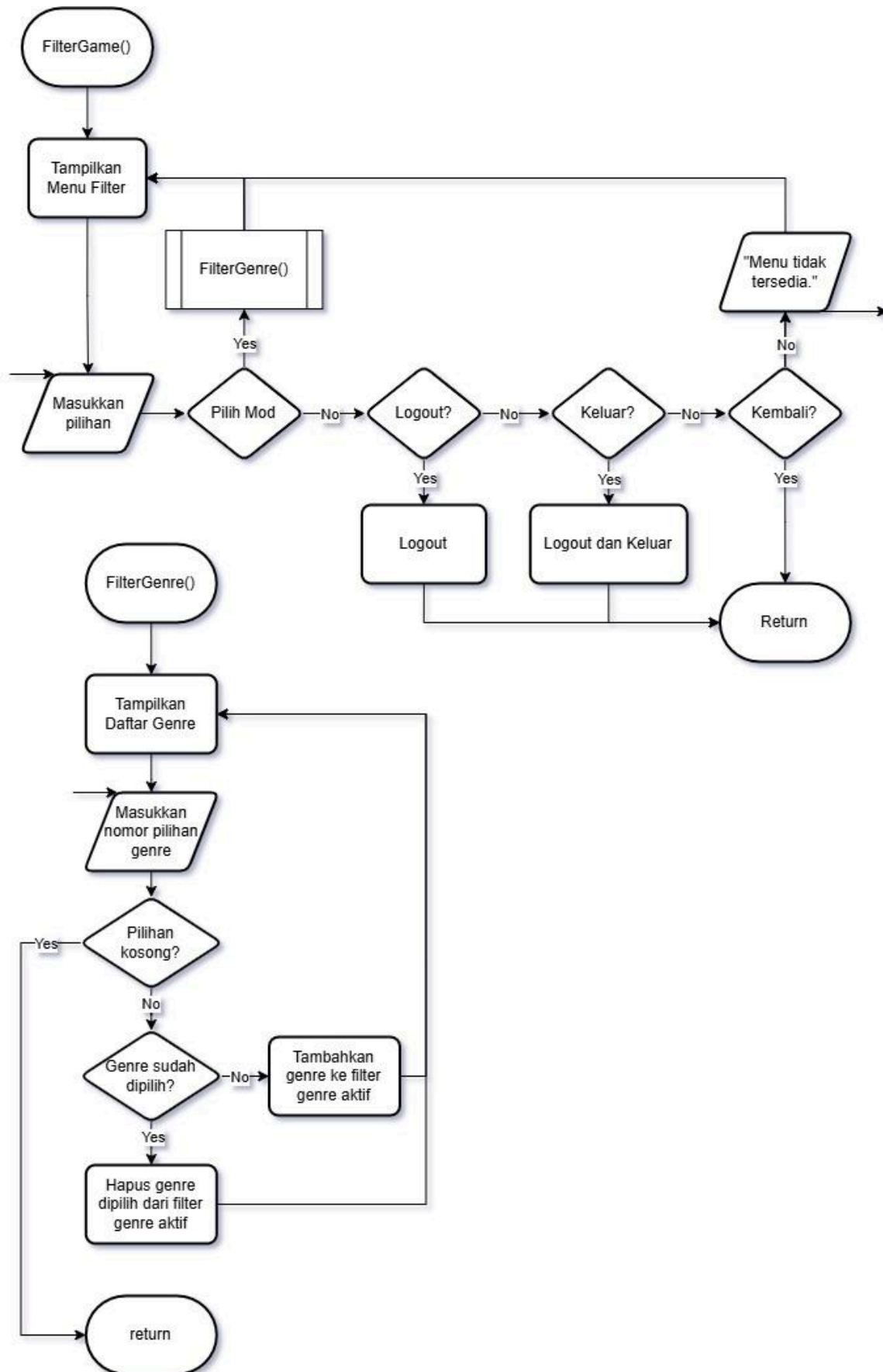


Gambar 1.1 Main Flow Bagian 1

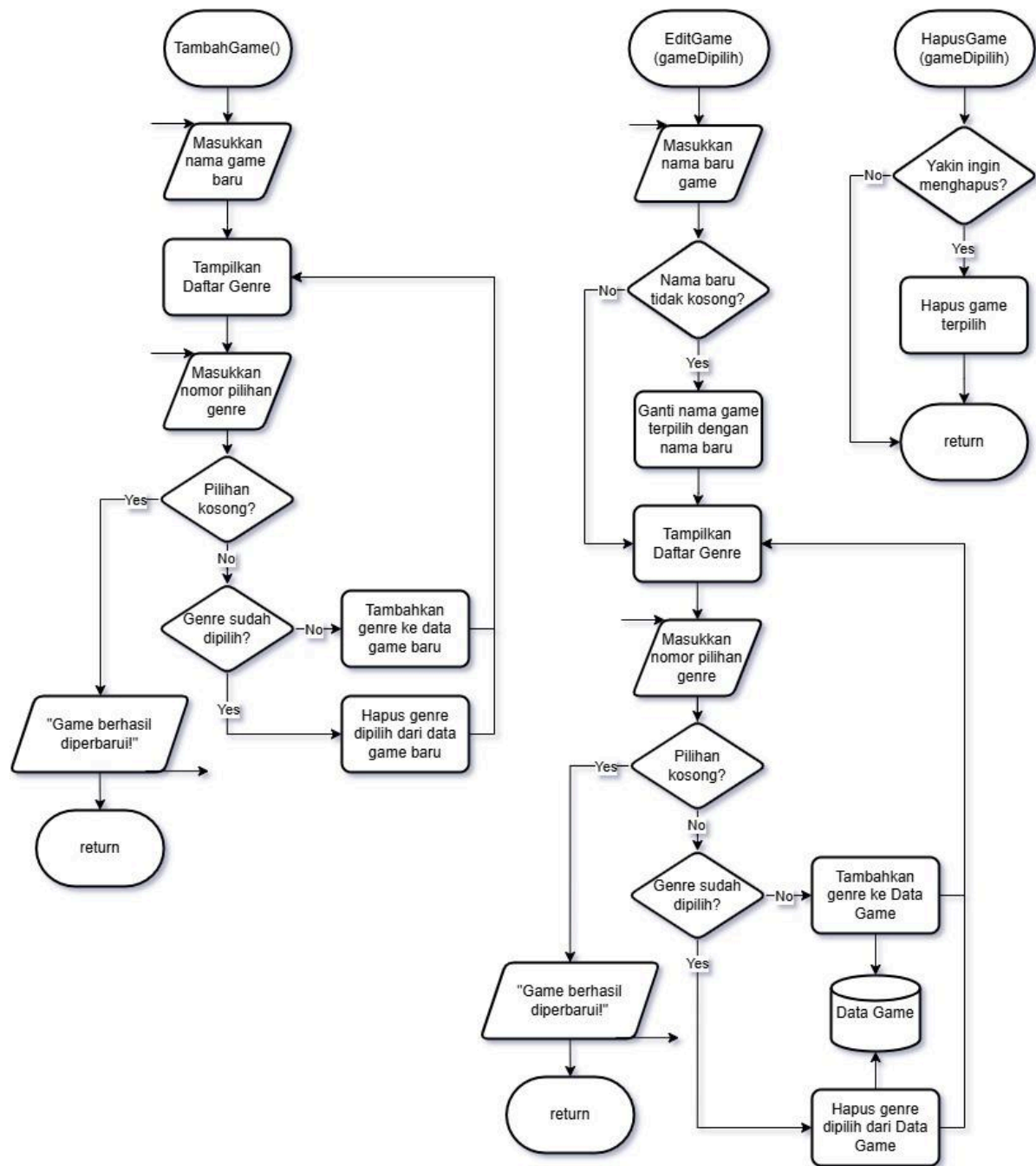


Gambar 1.2 Main Flow Bagian 2

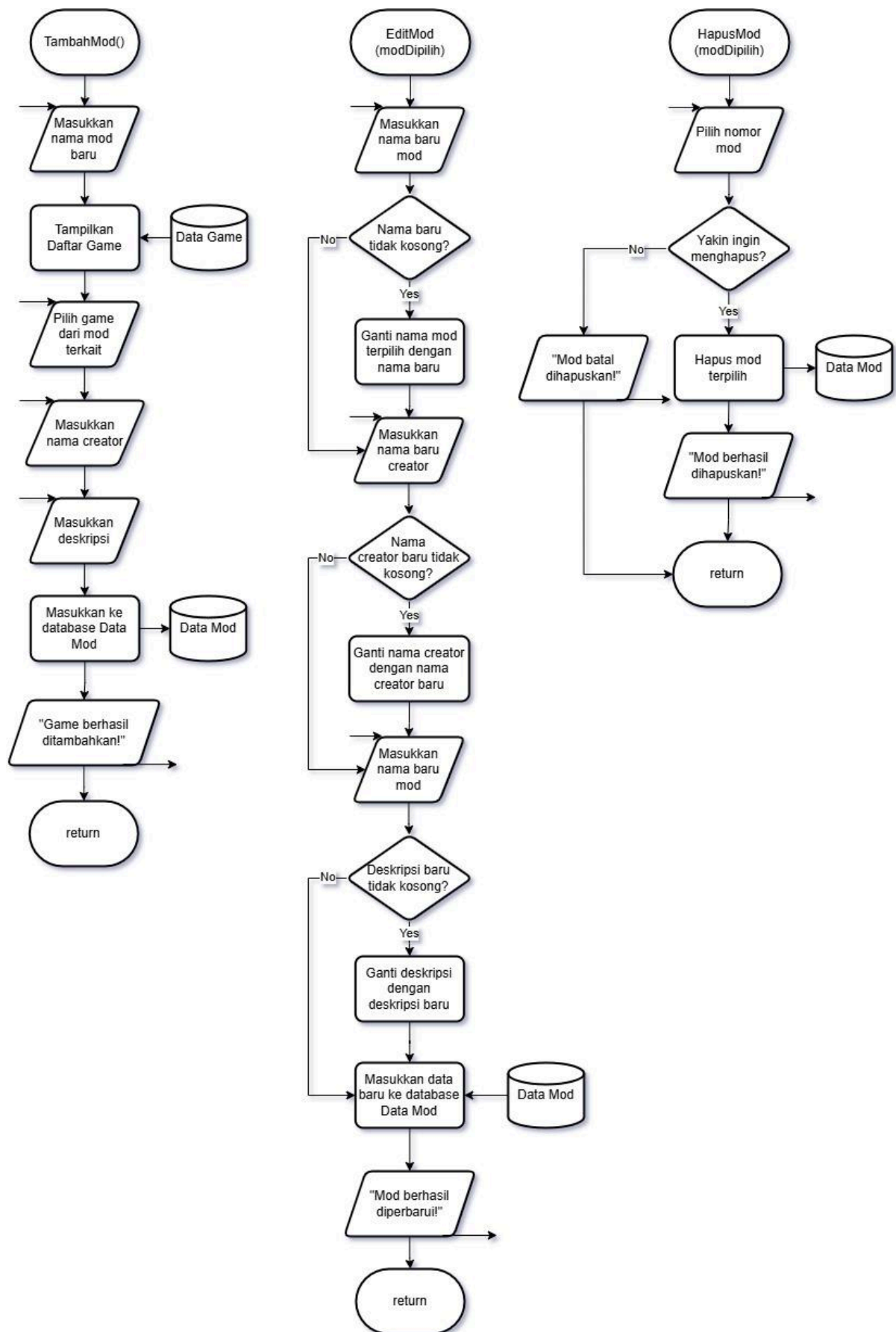




Gambar 1.5 Predefined Process Menu FilterGame() dan FilterGenre()



Gambar 1.6 Predefined Process TambahGame(), EditGame(), dan HapusGame (CRUD Game)



Gambar 1.7 Predefined Process TambahMod(), EditMod(), dan HapusMod() (CRUD Mod)

## 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.

## 3. Source Code

### A. Sistem Login

Pada awal program, pengguna akan berada dalam menu Sistem Login, di mana pengguna dapat memilih untuk melakukan login atau registrasi jika belum memiliki akun.

**Source Code:**

```
// Sistem Login
cout << "Sistem Pengelolaan Mod" << endl;
cout <<
"_____
_____ " << endl;
cout << "1 > Login" << endl;
cout << "2 > Register" << endl;
cout <<
"_____
_____ " << endl;
cout << "Q > Keluar" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);
```

## B. Login

Login digunakan untuk pengguna melanjutkan ke Menu Utama program. Di proses login, pengguna perlu memasukkan username dan password dari akun yang pernah dibuat pada saat registrasi. Maksimal percobaan hingga program pengguna dihentikan adalah sebanyak 3 kali.

### Source Code:

```
// Data Login
string inputUsername = "";
string inputPassword = "";
int batasPercobaan = 3;
int percobaanGagal = 0;
do
{
    cout << "Username\t: ";
    getline(cin, inputUsername);
    if (inputUsername == "")
        break;

    cout << "Password\t: ";
    getline(cin, inputPassword);
    if (inputPassword == "")
        break;

    // Proses validasi Login
    for (int i = 0; i < userCount; i++)
    {
        if (inputUsername == users[i].username && inputPassword ==
users[i].password)
        {
            currentUser = users[i];
            break;
        }
    }

    if (currentUser.id > 0)
    {
        cout << "Login berhasil!" << endl;
        riwayatHalamanCount = 0;
    }
    else
    {
        percobaanGagal++;
        cout << "Username atau Password salah!" << endl;
    }
    getline(cin, _temp);
} while (percobaanGagal < batasPercobaan && currentUser.id <= 0);
```



```

if (percobaanGagal >= batasPercobaan)
{
    // Menghentikan user ketika mencapai batas percobaan login
    cout << "Anda melampaui batas percobaan login!" << endl;
    break;
}

```

### C. Registrasi

Registrasi digunakan pengguna untuk membuat akun baru. Dalam proses registrasi pengguna perlu memasukkan username dan password yang akan digunakan dalam proses login.

#### Source Code:

```

// Registrasi
if (userCount < MAX_DATA)
{
    // Proses registrasi dapat berjalan selama user belum mencapai maks
    User newUser;

    cout << "Registrasi" << endl;
    cout << endl;
    cout << "Username\t: ";
    getline(cin, newUser.username);

    for (int i = 0; i < userCount; i++)
    {
        if (newUser.username != "" && newUser.username == users[i].username)
        {
            newUser.username = "";
            cout << "Username sudah ada!";
            getline(cin, _temp);
            break;
        }
    }

    if (newUser.username == "")
        break;

    cout << "Password\t: ";
    getline(cin, newUser.password);

    if (newUser.password == "")
        break;

    // Menambahkan user baru
}

```

```

    newUser.id = userCount + 1;
    newUser.level = "user";
    users[userCount] = newUser;
    userCount++;

    cout << "Registrasi berhasil!";
    getline(cin, _temp);
}
else
{
    cout << "Registrasi gagal! Coba dalam beberapa saat lagi.";
    getline(cin, _temp);
}

```

#### D. Logout

Fitur “Logout” digunakan untuk kembali ke menu “Sistem Login”, sehingga pengguna dapat melakukan registrasi atau login kembali tanpa harus memulai ulang program. Fitur ini dapat digunakan di setiap menu setelah login.

##### Source Code:

```

// Logout
currentUser = emptyUser;
percobaanGagal = 0;
riwayatHalamanCount = 0;
halaman_menu = '0';

```

#### D. Menu Utama

Menu Utama menampilkan daftar dari fitur utama program yang dapat pengguna pilih, yaitu “Menu Game” dan “Menu Mod.” Menu ini diakses setelah pengguna berhasil melakukan login.

##### Source Code:

```

// Menu Utama
cout << "Menu Utama" << endl;
cout <<
"_____
_____ " << endl;

```

```

cout << "Selamat datang, " << currentUser.username << "!" << endl;
cout << endl;
cout << "1 > Games" << endl;
cout << "2 > Mods" << endl;
cout <<
"_____
_____ " << endl;
cout << "Q > Keluar | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

```

## E. Menu Game

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

### Source Code:

```

// Menampilkan Menu Games
cout << "Games" << endl;
cout <<
"_____
_____ " << endl;
if (currentUser.level == "admin")
{
    cout << "C > Tambah Game | U > Edit Game | D > Hapus Game | ";
}
cout << "F > Filter Game" << endl;
cout <<
"_____
_____ " << endl;
if (gameListMenuCount > 0)
{
    cout << left << setw(6) << "No."
        << setw(40) << "Game"
        << endl;
    cout <<
"_____
_____ " << endl;

    for (int i = 0; i < gameListMenuCount; i++)
    {
        cout << left << setw(4) << i + 1 << setw(2) << ">"
            << setw(40) << gameListMenu[i].nama
            << endl;
    }
}

```

```

    }
}
else
{
    cout << "Belum ada data game!" << endl;
}
cout <<
"_____
_____ " << endl;
cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

```

## F. Tambah Game

Fitur “Tambah Game” memungkinkan admin untuk menambahkan *game* baru ke dalam daftar game. Dalam proses penambahan game, masukan yang perlu dipenuhi terdiri dari nama dan genre game.

### Source Code:

```

// Tambah Game
Game gameBaru;

if (gameCount <= MAX_DATA)
{
    cout <<
"_____
_____ " << endl;

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

    // Input Nama Game
    cout << "Nama Game\t: ";
    getline(cin, gameBaru.nama);

    for (int i = 0; i < gameCount; i++)
    {
        if (gameBaru.nama != "" && games[i].nama == gameBaru.nama)
        {
            gameBaru.nama = "";
            cout << "Game sudah ada!";

```

```

        getline(cin, _temp);
        break;
    }
}

if (gameBaru.nama == "")
    continue;

// Input Genre
while (true)
{
    cout << "Genre" << endl;
    for (int i = 0; i < genreCount; i++)
    {
        bool isSelected = false;
        for (int j = 0; j < gameBaru.genreCount; j++)
        {
            if (dataArray[i].genre == gameBaru.genre[j])
            {
                isSelected = true;
                break;
            }
        }
        cout << i + 1 << ". " << dataArray[i].genre << (isSelected ? "
(Dipilih)" : "") << endl;
    }

    cout << endl;
    cout << "(Kosongkan input untuk kembali.)" << endl;
    cout << "Pilih Genre\t: ";
    getline(cin, _temp);
    cout << endl;

    if (_temp == "")
        break;

    for (int i = 0; i < genreCount; i++)
    {
        bool isSelected = false;

        if (to_string(i + 1) == _temp)
        {
            for (int j = 0; j < gameBaru.genreCount; j++)
            {
                if (dataArray[i].genre == gameBaru.genre[j])
                {
                    isSelected = true;
                    for (int k = j; k < gameBaru.genreCount - 1; k++)
                    {
                        gameBaru.genre[k] = gameBaru.genre[k + 1];
                    }
                }
            }
        }
    }
}

```

```

        gameBaru.genreCount--;
        break;
    }
}

if (isSelected)
    break;

if (gameBaru.genreCount < SMOL_DATA)
{
    gameBaru.genre[gameBaru.genreCount] = dataArray[i].genre;
    gameBaru.genreCount++;
}
else
{
    cout << "Mencapai batas maksimum filter genre!";
    getline(cin, _temp);
}
}
}

// Prsoses penambahan game
games[gameCount] = gameBaru;
gameCount++;
cout << "Game berhasil ditambahkan!";
getline(cin, _temp);
}
else
{
    cout << "Game penuh! Tidak dapat menambah game lagi.";
    getline(cin, _temp);
}
}

```

## G. Edit Game

Fitur “Edit Game” memungkinkan admin untuk memperbarui data game yang sudah terdaftar, seperti mengganti nama atau genre dari suatu game.

### Source Code:

```

// Edit Game
int _selectedIndex = -1;

// Input pilihan game
cout << "Pilih Game\t: ";
getline(cin, _temp);

```

```

if (_temp == "")
    continue;

for (int i = 0; i < gameListMenuCount; i++)
{
    if (to_string(i + 1) == _temp)
    {
        for (int j = 0; j < gameCount; j++)
        {
            if (gameListMenu[i].nama == games[j].nama)
            {
                _selectedIndex = j;
                break;
            }
        }
        break;
    }
}

if (gameCount > _selectedIndex >= 0 )
{
    // Lanjutkan proses edit game jika indeks valid
    Game gameBaru = games[_selectedIndex];
    cout <<
    "
    _____" << endl;

    cout << "Edit Game \"" << games[_selectedIndex].nama << "\"" << endl;
    cout <<
    "
    _____" << endl;

    // Input Nama Game
    cout << "Nama Game Baru\t: ";
    getline(cin, _temp);
    cout << endl;

    // Validasi nama game
    for (int i = 0; i < gameCount; i++)
    {
        if (_temp != "" && games[i].nama == _temp)
        {
            _temp = "";
            cout << "Game sudah ada!";
            getline(cin, _temp);
            break;
        }
    }

    if (_temp != "")
        gameBaru.nama = _temp;
}

```

```

// Input Genre
while (true)
{
    cout << "Genre" << endl;
    for (int i = 0; i < genreCount; i++)
    {
        bool isSelected = false;
        for (int j = 0; j < gameBaru.genreCount; j++)
        {
            if (dataArray[i].genre == gameBaru.genre[j])
            {
                isSelected = true;
                break;
            }
        }
        cout << i + 1 << ". " << dataArray[i].genre << (isSelected ? "
(Dipilih)" : "") << endl;
    }

    cout << endl;
    cout << "(Kosongkan input untuk kembali.)" << endl;
    cout << "Pilih Genre\t: ";
    getline(cin, _temp);
    cout << endl;

    if (_temp == "")
        break;

    for (int i = 0; i < genreCount; i++)
    {
        bool isSelected = false;

        if (to_string(i + 1) == _temp)
        {
            for (int j = 0; j < gameBaru.genreCount; j++)
            {
                if (dataArray[i].genre == gameBaru.genre[j])
                {
                    isSelected = true;
                    for (int k = j; k < gameBaru.genreCount - 1; k++)
                    {
                        gameBaru.genre[k] = gameBaru.genre[k + 1];
                    }
                    gameBaru.genreCount--;
                    break;
                }
            }
        }

        if (isSelected)
            break;
    }
}

```



```

        if (gameBaru.genreCount < SMOL_DATA)
        {
            gameBaru.genre[gameBaru.genreCount] = dataArray[i].genre;
            gameBaru.genreCount++;
        }
        else
        {
            cout << "Mencapai batas maksimum filter genre!";
            getline(cin, _temp);
        }
    }
}

// Proses perubahan data game
games[_selectedIndex] = gameBaru;
cout << "Game berhasil diperbarui!";
getline(cin, _temp);
}
else
{
    cout << "Game tidak ditemukan!";
    getline(cin, _temp);
}
}

```

## H. Hapus Game

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

### Source Code:

```

// Menampilkan Menu Mod
if (selectedGame.nama != "")
    cout << "Mod " << selectedGame.nama << endl;
else
    cout << "Semua Mod" << endl;
cout <<
"
" << endl;
cout << "C > Upload Mod | U > Edit Mod | D > Hapus Mod" << endl;
cout <<
"
" << endl;
if (modListMenuCount > 0)
{

```

```

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

for (int i = 0; i < modListMenuCount; i++)
{
    cout << left << setw(4) << i + 1 << setw(2) << ">"
        << setw(40) << modListMenu[i].nama
        << setw(40) << modListMenu[i].game.nama
        << setw(20) << modListMenu[i].creator
        << endl;
}
}
else
{
    cout << "Belum ada data mod!" << endl;
}
cout <<
"
" << endl;
cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

```

## I. Menu Filter Game

Menu “Filter Game” menyediakan cara untuk melihat daftar game lebih spesifik sesuai dengan tipe game yang dicari pengguna. Dalam menu ini terdapat salah satunya adalah filter “Genre.”

### Source Code:

```

cout << "Filter Game" << endl;
cout <<
"
" << endl;
cout << "1. Genre" << endl;
cout <<
"
" << endl;
cout << "Q > Keluar | B > Kembali | L > Logout" << endl;

```

```
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);
```

## J. Filter Genre Game

Fitur “Filter Genre Game” memungkinkan pengguna untuk memilih beberapa genre untuk hanya melihat game dengan genre terpilih saja yang ditampilkan dalam “Menu Game.”

### Source Code:

```
// Menampilkan daftar genre
cout << "Genre" << endl;
for (int i = 0; i < genreCount; i++)
{
    bool isSelected = false;
    for (int j = 0; j < SMOL_DATA; j++)
    {
        if (dataArray[i].genre == selectedGenres[j])
        {
            isSelected = true;
            break;
        }
    }
    cout << i + 1 << ". " << dataArray[i].genre << (isSelected ? " (Dipilih)" :
    "") << endl;
}

cout << endl;
cout << "(Kosongkan input untuk kembali.)" << endl;
cout << "Pilih Genre\t: ";
getline(cin, _temp);

if (_temp == "")
    break;

for (int i = 0; i < genreCount; i++)
{
    bool isSelected = false;

    if (to_string(i + 1) == _temp)
    {
        // Mengeliminasi genre jika sudah dipilih
        for (int j = 0; j < SMOL_DATA; j++)
        {
            if (dataArray[i].genre == selectedGenres[j])
```

```

        {
            isSelected = true;
            for (int k = j; k < SMOL_DATA - 1; k++)
            {
                selectedGenres[k] = selectedGenres[k + 1];
            }
            selectedGenreCount--;
            break;
        }
    }

    // Menambahkan genre jika belum dipilih
    if (isSelected)
        break;

    if (selectedGenreCount < SMOL_DATA)
    {
        selectedGenres[selectedGenreCount] = dataArray[i].genre;
        selectedGenreCount++;
    }
    else
    {
        cout << "Mencapai batas maksimum filter genre!";
        getline(cin, _temp);
    }
}
}

```

## K. Menu Mod

Di “Menu Mod”, program menampilkan daftar mod yang tersedia. Program juga menyediakan pengguna fitur untuk meng-*upload* suatu mod..

### Source Code:

```

// Mods
cout << "Mods" << endl;
cout <<
"_____ " << endl;
cout << "C > Upload 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 | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

```

## L. Upload Mod

Fitur “Upload Mod” memungkinkan program untuk menambahkan *mod* baru dari suatu *game* yang telah terdaftar. Dalam proses upload mod, masukan yang diperlukan terdiri dari nama, *video game*, nama *creator*, dan deskripsi dari mod yang di-*upload*-nya.

## Source Code:

```
// Upload Mod
if (modCount < MAX_DATA)
{
    // Tambahkan mod jika belum penuh
    Mod modBaru;

    cout <<
    "
    " << endl;

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

    cout << "(Kosongkan input untuk kembali.)" << endl;

    // Input nama mod
    cout << "Nama Mod\t: ";
    getline(cin, modBaru.nama);
    cout << endl;

    // Validasi nama mod
    for (int i = 0; i < modCount; i++)
    {
        if (modBaru.nama != "" && modBaru.nama == mods[i].nama)
        {
            modBaru.nama = "";
            cout << "Nama mod sudah ada!";
            getline(cin, _temp);
            break;
        }
    }

    if (modBaru.nama == "")
        continue;

    // Input creator
    cout << "Creator\t\t: ";
    getline(cin, modBaru.creator);
    cout << endl;

    if (modBaru.creator == "")
        continue;

    // Input pilihan game
    cout << "Game" << endl;
    for (int i = 0; i < gameCount; i++)
        cout << i + 1 << ". " << games[i].nama << endl;
    cout << "Pilih Game\t: ";
```

```

getline(cin, _temp);
cout << endl;

if (_temp == "")
    continue;

// Mencari dan validasi game yang dipilih
for (int i = 0; i < gameCount; i++)
{
    if (to_string(i + 1) == _temp)
    {
        modBaru.game = games[i];
        break;
    }
}

if (modBaru.game.nama == "")
{
    cout << "Game tidak ditemukan!";
    getline(cin, _temp);
    continue;
}

// Input deskripsi
cout << "Deskripsi\t: ";
getline(cin, modBaru.deskripsi);
cout << endl;

if (modBaru.deskripsi == "")
    continue;

// Proses penambahan mod
modBaru.id += 1;
mods[modCount] = modBaru;
modCount++;
cout << "Mod berhasil diupload!";
getline(cin, _temp);
}
else
{
    cout << "Mod penuh! Tidak dapat mengupload mod lagi.";
    getline(cin, _temp);
}
}

```

## M. Detail Mod

Menu “Detail Mod” menampilkan informasi lengkap dari suatu *mod* yang dipilih dari “Menu Mod.” Dalam menu ini menyediakan fitur bagi *uploader* untuk mengedit atau menghapus mod yang sedang dilihatnya saat ini.

### Source Code:

```
// Menampilkan detail mod terpilih
cout << selectedMod.nama << endl;
cout <<
"_____
_____ " << endl;
if (currentUser.id == selectedMod.uploader.id)
{
    cout << "U > Edit | D > Hapus" << endl;
    cout <<
"_____
_____ " << endl;
}
cout << "Creator\t\t: " << selectedMod.creator << endl;
cout << "Uploader\t: " << selectedMod.uploader.username << endl;
cout << "Game\t\t: " << selectedMod.game.nama << endl;
cout <<
"_____
_____ " << endl;
cout << selectedMod.deskripsi << endl;
cout <<
"_____
_____ " << endl;
cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);
```

## N. Edit Mod

Fitur “Edit Mod” memungkinkan *uploader* untuk memperbarui data *mod* yang sedang dibuka di menu “Detail Mod.” *Uploader* dapat memperbarui nama, nama *creator*, dan deskripsi dari mod yang telah dipilihnya.



## Source Code:

```
// Edit Mod terpilih
cout << "(Kosongkan untuk melewati salah satu pengeditan.)" << endl;
cout << "Nama Baru\t:";
getline(cin, _temp);

for (int i = 0; i < modCount; i++)
{
    if (_temp != "" && mods[i].nama == _temp)
    {
        _temp = "";
        cout << "Nama mod sudah ada!";
        getline(cin, _temp);
        break;
    }
}

if (_temp != "")
    selectedMod.nama = _temp;

cout << "Creator Baru\t:";
getline(cin, _temp);
if (_temp != "")
    selectedMod.creator = _temp;

cout << "Deskripsi Baru\t:";
getline(cin, _temp);
if (_temp != "")
    selectedMod.deskripsi = _temp;

for (int i = 0; i < modCount; i++)
{
    if (mods[i].id == selectedMod.id)
    {
        mods[i] = selectedMod;
    }
}

cout << "Mod berhasil diperbarui!";
getline(cin, _temp);
```

## O. Hapus Mod

Fitur “Edit Mod” memungkinkan *uploader* untuk menghapus *mod* yang sedang dibuka di menu “Detail Mod.”

## Source Code:

```
// Menghapus Mod saat ini
cout << "Apakah Anda yakin ingin menghapus mod \"" << selectedMod.nama << "\"?
[Y/N] ";
getline(cin, _temp);

if (_temp == "")
    break;
else if (_temp == "Y" || _temp == "y")
{
    // Proses penghapusan mod
    for (int i = 0; i < modCount; i++)
    {
        if (mods[i].id == selectedMod.id)
        {
            for (int j = i; j < modCount; j++)
            {
                mods[j] = mods[j + 1];
            }
            modCount--;
            break;
        }
    }
    cout << "Mod berhasil dihapuskan!";
    getline(cin, _temp);
    selectedMod = emptyMod;
    halaman_menu = '2';
}
else if (_temp == "N" || _temp == "n")
{
    // Batalkan penghapusan mod
    cout << "Mod batal dihapuskan!";
    getline(cin, _temp);
}
else
{
    cout << "Input tidak valid!";
    getline(cin, _temp);
}
```

#### 4. Hasil Output

```
=====
Sistem Pengelolaan Mod
=====
1 > Login
2 > Register
=====
Q > Keluar

Pilih Menu      : █
```

Gambar 4.1 Halaman “Sistem Login” untuk Login atau Register

```
Registrasi

Username       : Trigusni
Password       : trigusni123
Registrasi berhasil!
```

Gambar 4.2 Proses Registrasi berhasil

```
Login

Username       : Trigusni
Password       : trigusni123
Login berhasil!
```

Gambar 4.3 Proses Login berhasil dengan akun baru

```
=====
Menu Utama
=====
Selamat datang, Trigusni!

1 > Games
2 > Mods
=====
Q > Keluar | L > Logout

Pilih Menu      : █
```

Gambar 4.4 Halaman “Menu Utama”

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

Gambar 4.5 Halaman “Menu Game” sebagai Admin

```
=====
Games
F > Filter Game
No.  Game
1   > Stardew Valley
2   > Undertale
3   > Terraria
4   > Fallout 4
5   > Skyrim
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : 
```

Gambar 4.6 Halaman “Menu Game” sebagai User

```
Tambah Game
Nama Game      : Geometry Dash
Genre
1. Action
2. Adventure
3. RPG
4. Simulation
(Kosongkan input untuk kembali.)
Pilih Genre    : 1
Genre
1. Action (Dipilih)
2. Adventure
3. RPG
4. Simulation
(Kosongkan input untuk kembali.)
Pilih Genre    :
Game berhasil ditambahkan!
=====
```

Gambar 4.7 Proses tambah game

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

Gambar 4.8 Tampilan “Menu Game” setelah penambahan game baru

```
=====
Edit Game "Undertale"
Nama Game Baru : Deltarune
Genre
1. Action
2. Adventure
3. RPG (Dipilih)
4. Simulation
(Kosongkan input untuk kembali.)
Pilih Genre :
Game berhasil diperbarui!
=====
```

Gambar 4.9 Proses edit game “Undertale”

```
=====
Games
C > Tambah Game | U > Edit Game | D > Hapus Game | F > Filter Game
No.  Game
1   > Stardew Valley
2   > Deltarune
3   > Terraria
4   > Fallout 4
5   > Skyrim
6   > Geometry Dash
Q > Keluar | B > Kembali | L > Logout
Pilih Menu : █
```

Gambar 4.10 Tampilan “Menu Game” setelah edit game

```
Pilih Menu : d
Pilih Game : 5
Apakah Anda yakin ingin menghapus "Skyrim"? [Y/N] : Y
Game berhasil dihapus!
```

Gambar 4.11 Proses hapus game “Skyrim”

```
=====
Games
-----
C > Tambah Game | U > Edit Game | D > Hapus Game | F > Filter Game
-----
No.   Game
-----
1    > Stardew Valley
2    > Deltarune
3    > Terraria
4    > Fallout 4
5    > Geometry Dash
-----
Q > Keluar | B > Kembali | L > Logout
-----
Pilih Menu      : 
```

Gambar 4.12 Tampilan “Menu Game” setelah penghapusan game.

```
=====
Filter Game
-----
1. Genre
-----
Q > Keluar | B > Kembali | L > Logout
-----
Pilih Menu      : 
```

Gambar 4.13 Halaman “Menu Filter.”

```
Genre
1. Action
2. Adventure
3. RPG
4. Simulation

(Kosongkan input untuk kembali.)
Pilih Genre      : 1
Genre
1. Action (Dipilih)
2. Adventure
3. RPG
4. Simulation

(Kosongkan input untuk kembali.)
Pilih Genre      : 
=====
```

Gambar 4.14 Memilih filter “Action” di fitur “Filter Genre Game”

```

Genre
1. Action
2. Adventure
3. RPG
4. Simulation

(Kosongkan input untuk kembali.)
Pilih Genre      : 1
Genre
1. Action (Dipilih)
2. Adventure
3. RPG
4. Simulation

(Kosongkan input untuk kembali.)
Pilih Genre      :
=====

```

Gambar 4.15 Memilih filter “Action” di fitur “Filter Genre Game”

```

=====
Games
C > Tambah Game | U > Edit Game | D > Hapus Game | F > Filter Game
-----
No.   Game
-----
1    > Terraria
2    > Geometry Dash
-----
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : █

```

Gambar 4.16 Tampilan “Menu Game” setelah memilih filter genre “Action”

```

=====
Games
C > Tambah Game | U > Edit Game | D > Hapus Game | F > Filter Game
-----
No.   Game
-----
1    > Terraria
2    > Geometry Dash
-----
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : █

```

Gambar 4.17 Tampilan “Menu Game” setelah memilih filter genre “Action”

```

=====
Mod Terraria
C > Upload Mod
-----
No.   Nama Mod                               Game                               Creator
-----
1    > Calamity Mod                           Terraria                           Fabsol
-----
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : █

```

Gambar 4.18 Tampilan “Menu Mod”, jika dibuka dengan memilih game dari “Menu Game”

```

=====
Semua Mod
=====
C > Upload Mod
=====
No.    Nama Mod                                Game                                Creator
=====
1  > Stardew Valley Expanded                Stardew Valley                    FlashShifter
2  > Undertale Bed Lump                     Undertale                        TheMaxine
3  > Calamity Mod                           Terraria                         Fabsol
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : 

```

Gambar 4.19 Tampilan “Menu Mod”, jika dibuka melalui “Menu Utama”

```

=====
Semua Mod
=====
C > Upload Mod
=====
No.    Nama Mod                                Game                                Creator
=====
1  > Stardew Valley Expanded                Stardew Valley                    FlashShifter
2  > Undertale Bed Lump                     Undertale                        TheMaxine
3  > Calamity Mod                           Terraria                         Fabsol
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : 

```

Gambar 4.20 Tampilan “Menu Mod”, jika dibuka melalui “Menu Utama”

```

=====
Undertale Bed Lump
=====
U > Edit | D > Hapus
=====
Creator      : TheMaxine
Uploader     : Hermawan
Game         : Undertale
=====
Based on that one 20-second gag from Undertale... the BED LUMP finally gets their own adventure.

No fighting. No thinking. No worrying. Only dancing. Get to the Barrier.
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : 

```

Gambar 4.21 Tampilan Detail Mod “Undertale Bed Lump”, jika sebagai *uploader*



```
=====
Calamity Mod
=====
D > Hapus
=====
Creator      : Fabsol
Uploader     : user
Game         : Terraria
=====
The Calamity Mod is a large content mod for Terraria which adds many hours of endgame content and dozens of enemies
and bosses dispersed throughout the vanilla game's progression.
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu  : █
```

Gambar 4.22 Tampilan Detail Mod “Calamity Mod”, jika sebagai *admin* dan bukan sebagai *uploader*

```
=====
Undertale Bed Lump
=====
Creator      : TheMaxine
Uploader     : Hermawan
Game         : Undertale
=====
Based on that one 20-second gag from Undertale... the BED LUMP finally gets their own adventure.

No fighting. No thinking. No worrying. Only dancing. Get to the Barrier.
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu  : █
```

Gambar 4.23 Tampilan Detail Mod “Undertale Bed Lump”, jika bukan sebagai admin dan *uploader*

```
=====
(Kosongkan untuk melewati salah satu pengeditan.)
Nama Baru      :Deltarune Bad Lump
Creator Baru   :
Deskripsi Baru :Ini deskripsi baru.
Mod berhasil diperbarui!
=====
```

Gambar 4.24 Proses edit mod “Undertale Bed Lump”

```
=====
Deltarune Bad Lump
=====
U > Edit | D > Hapus
=====
Creator      : TheMaxine
Uploader     : Hermawan
Game         : Undertale
=====
Ini deskripsi baru.
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu  : █
```

Gambar 4.25 Tampilan Detail Mod setelah pengeditan mod

```

=====
Semua Mod
=====
C > Upload Mod
=====
No.   Nama Mod                                Game                                Creator
-----
1   > Stardew Valley Expanded                Stardew Valley                    FlashShifter
2   > Deltarune Bad Lump                     Undertale                        TheMaxine
3   > Calamity Mod                           Terraria                         Fabsol
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : █

```

Gambar 4.26 Tampilan “Menu Mod” setelah pengeditan mod

```

=====
Upload Mod
=====
(Kosongkan input untuk kembali.)
Nama Mod      : Thorium Mod
Creator       : DivermasSam
Game
1. Stardew Valley
2. Undertale
3. Terraria
4. Fallout 4
5. Skyrim
Pilih Game    : 3
Deskripsi     : Deskripsi baru
Mod berhasil diupload!
=====

```

Gambar 4.27 Proses *upload mod*

```

=====
Semua Mod
=====
C > Upload Mod
=====
No.   Nama Mod                                Game                                Creator
-----
1   > Stardew Valley Expanded                Stardew Valley                    FlashShifter
2   > Deltarune Bad Lump                     Undertale                        TheMaxine
3   > Calamity Mod                           Terraria                         Fabsol
4   > Thorium Mod                           Terraria                         DivermasSam
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu      : █

```

Gambar 4.28 Tampilan “Menu Mod” setelah meng-*upload mod*

```

=====
Thorium Mod
=====
U > Edit | D > Hapus
=====
Creator      : DivermanSam
Uploader     : Trigusni
Game         : Terraria
=====
Deskripsi baru.
=====
Q > Keluar | B > Kembali | L > Logout
Pilih Menu   : █

```

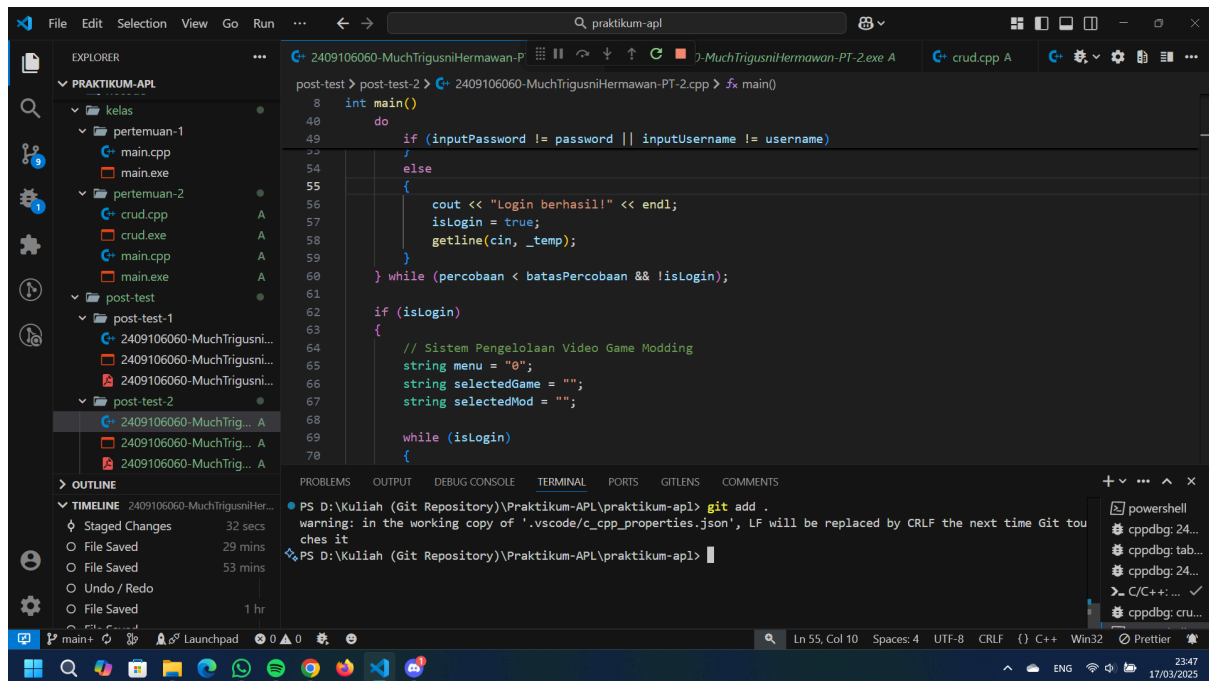
Gambar 4.29 Tampilan detail mod “Thorium Mod” setelah di-*upload*

## 5. Langkah-Langkah Git

### A. Git Init

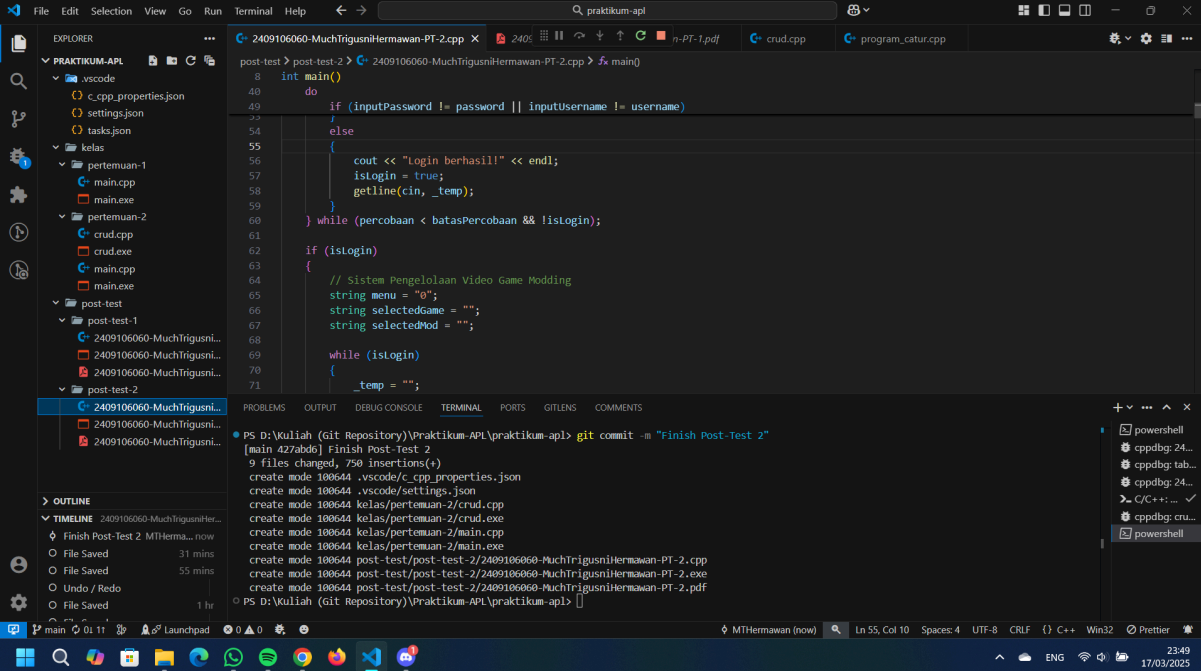
Karena fungsi git init untuk membuat repository kosong di lokal pada saat pertama kali, sementara repository lokal telah dibuat pada saat post-test pertama, maka tidak perlu dilakukan lagi.

### 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 C++ code. The terminal window at the bottom shows the command `git commit -m "Finish Post-Test 2"` being executed, followed by the output of the command, which lists the files changed and the commit hash.

```
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 = "";
```

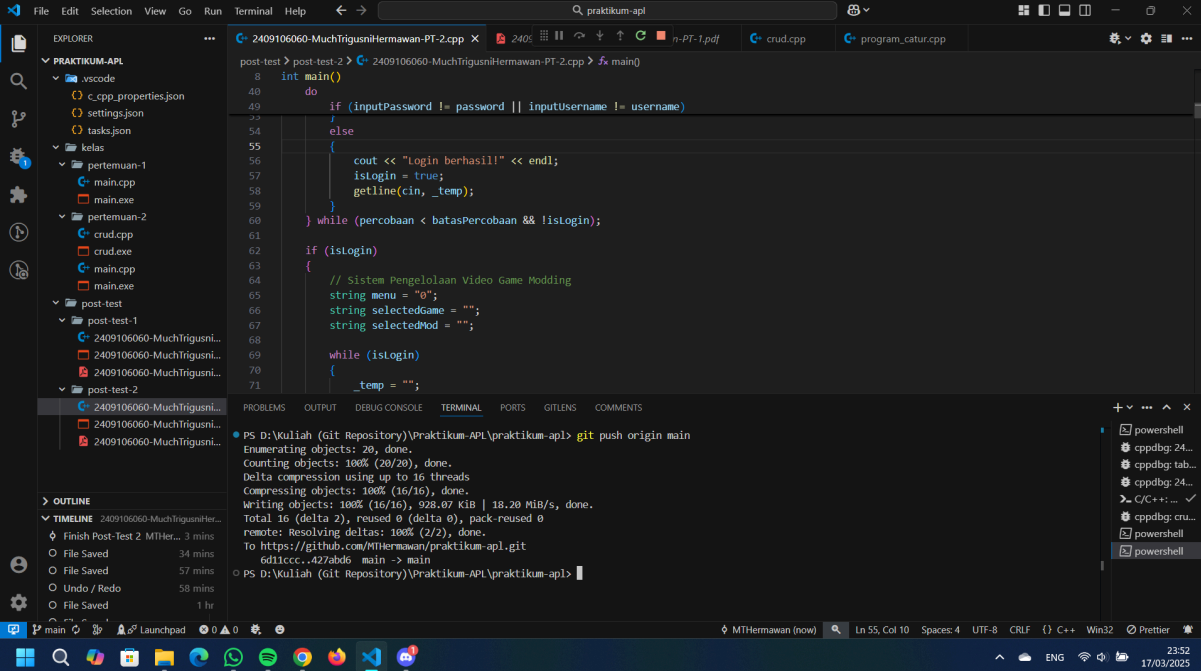
```
PS D:\Kuliah (git Repository)\Praktikum-APL\praktikum-apl> git commit -m "Finish Post-Test 2"
[main 427abde] Finish Post-Test 2
9 files changed, 750 insertions(+)
create mode 100644 .vscode/c_cpp_properties.json
create mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan-2/crud.cpp
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 various files and folders. The main editor window shows a C++ file named '2409106060-MuchTrigusniHermawan-PT-2.cpp'. The code includes a login function and a menu system. The Terminal panel at the bottom shows the output of a 'git push' command, indicating a successful push to the 'origin main' branch. The output includes details about object enumeration, compression, and writing to the 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.