

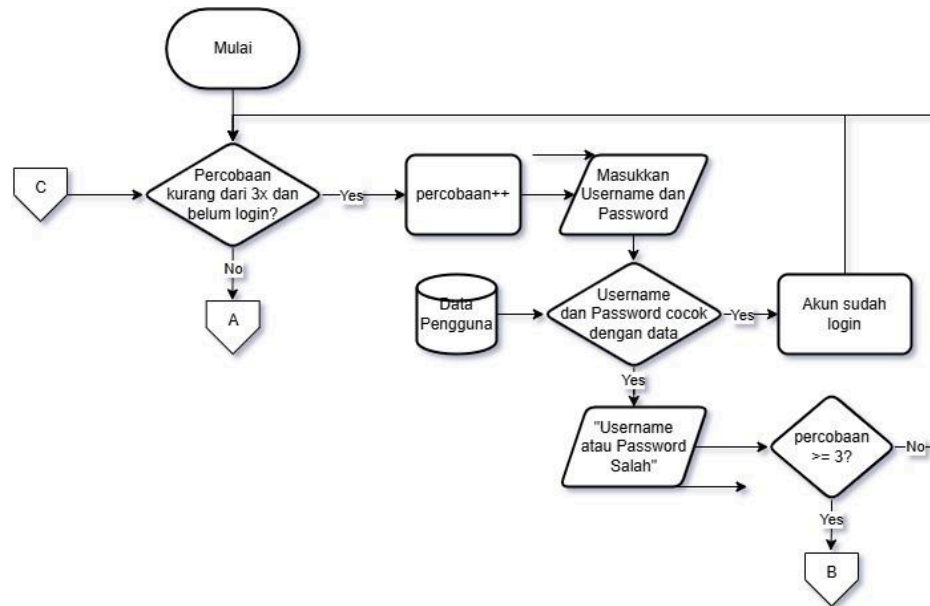
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT



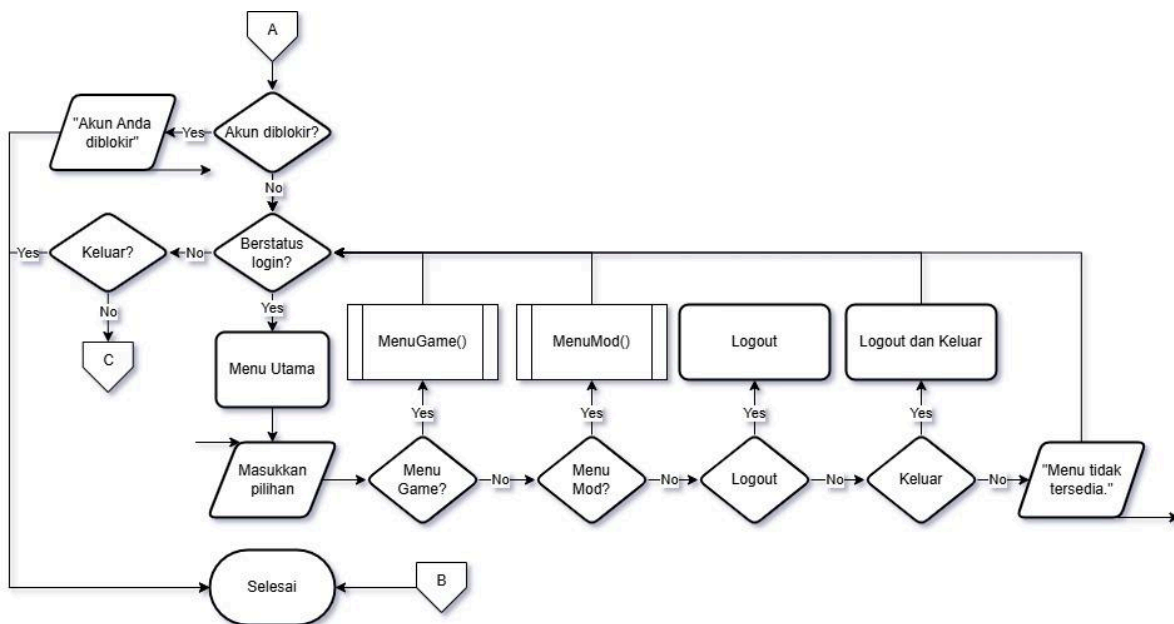
Disusun oleh:
Much. Trigusni Hermawan (2409106060)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

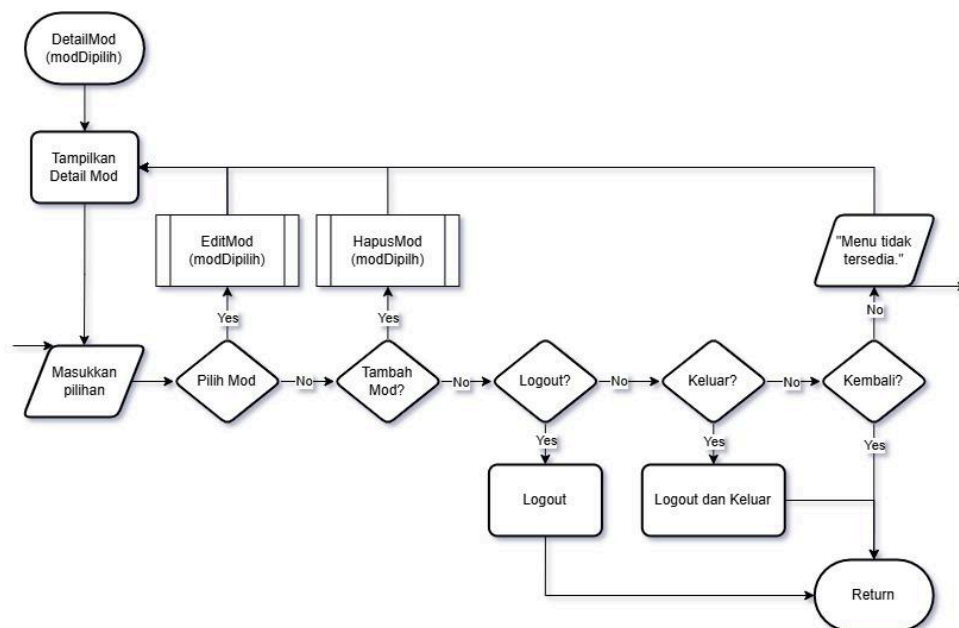
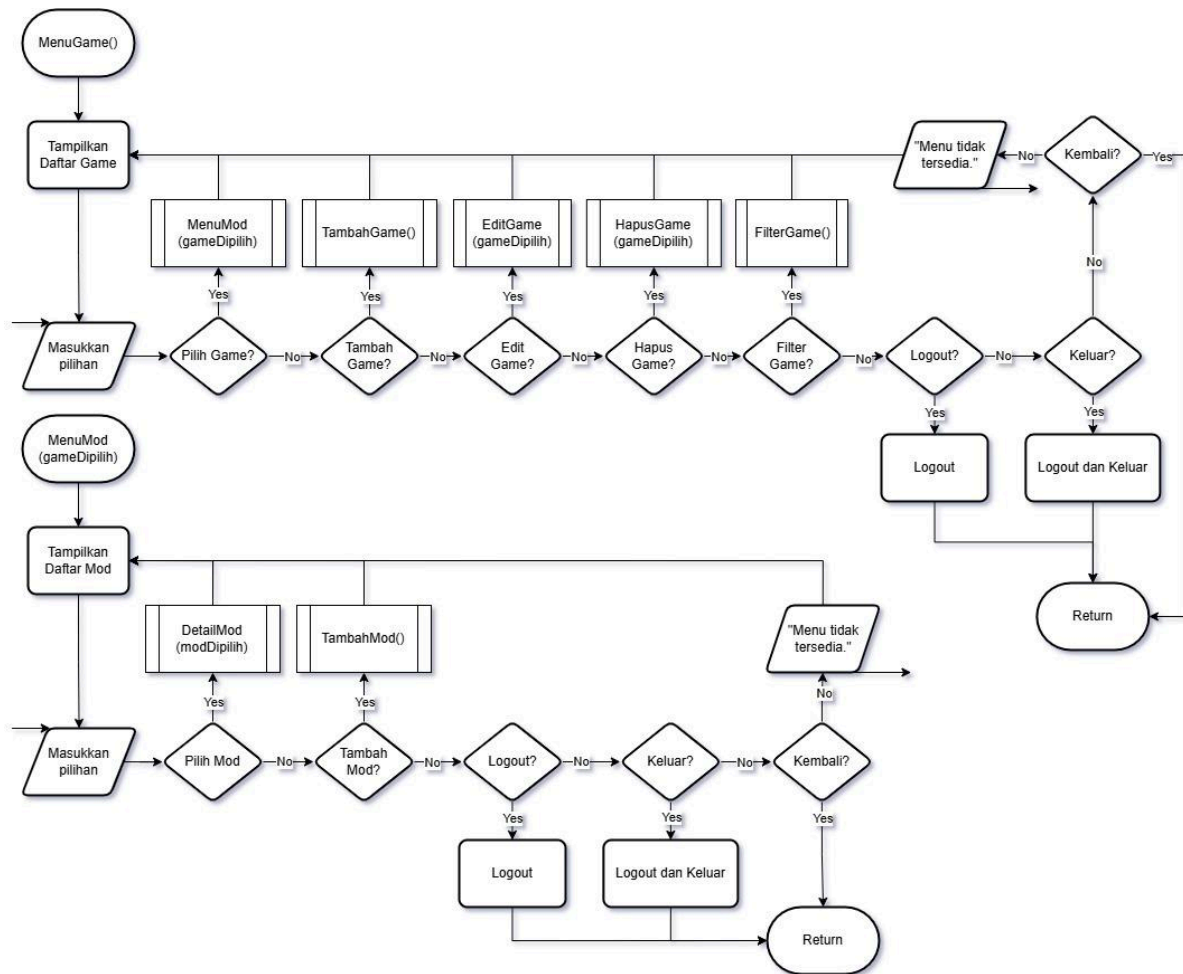
1. Flowchart

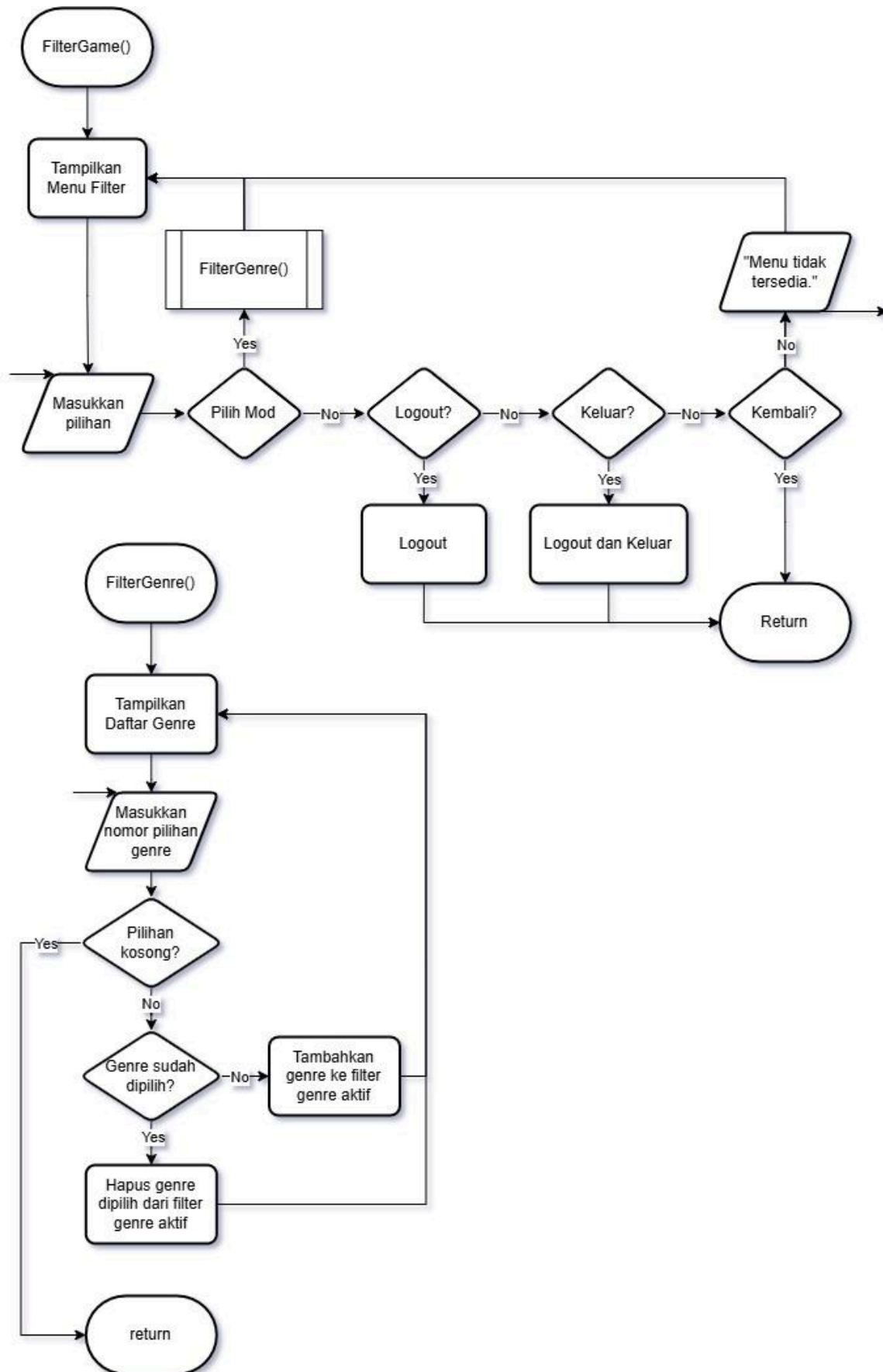


Gambar 1.1 Main Flow Bagian 1

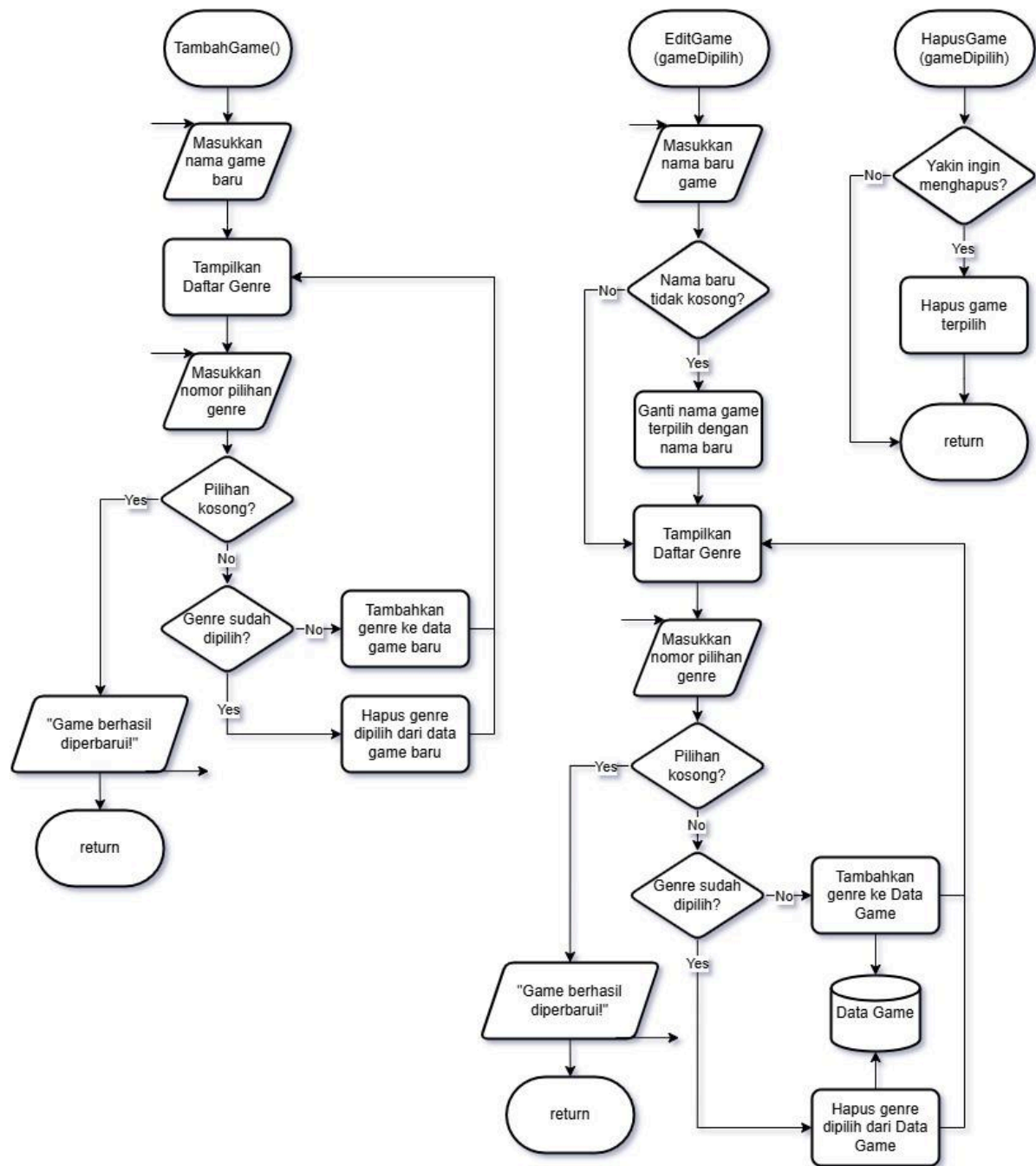


Gambar 1.2 Main Flow Bagian 2

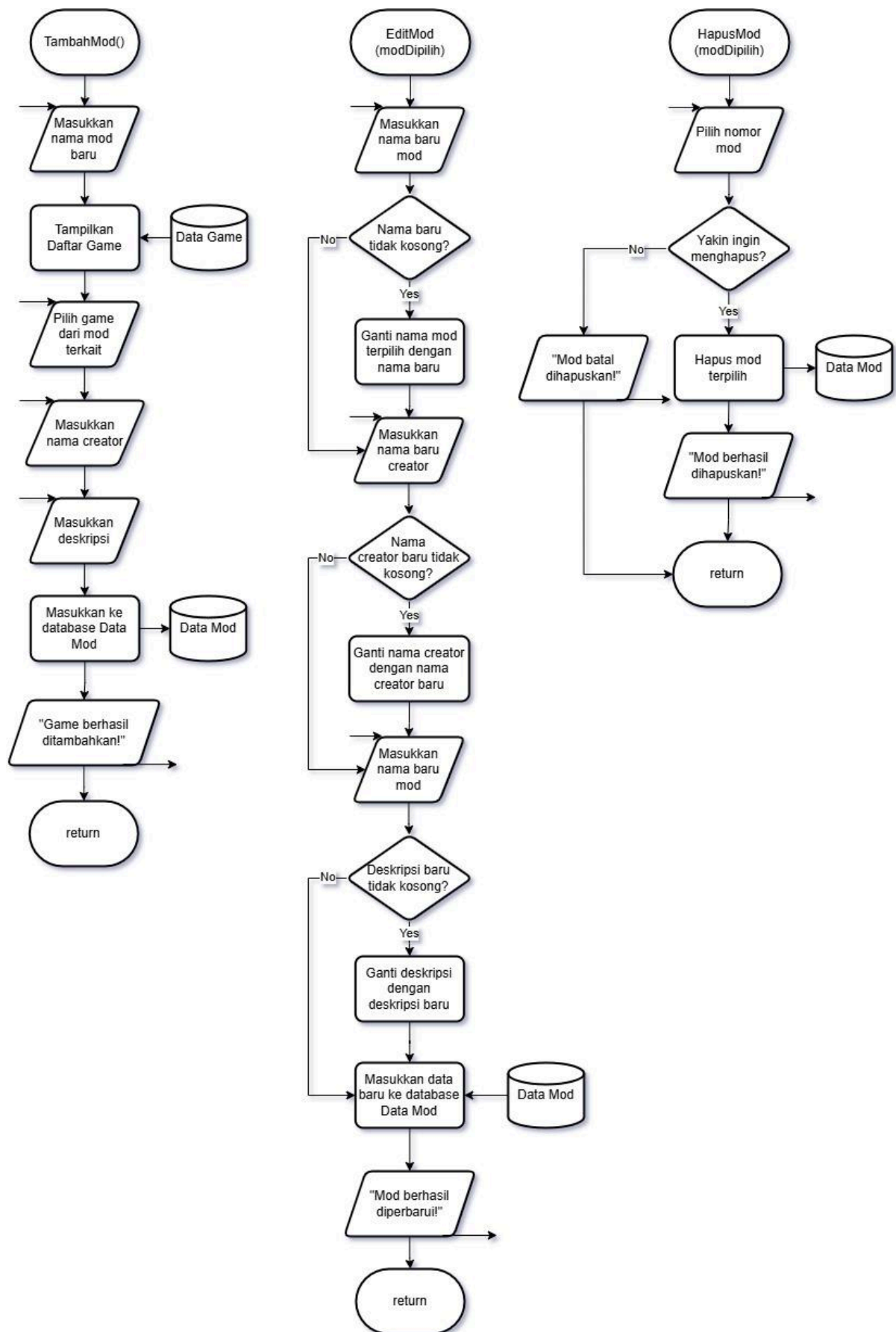




Gambar 1.5 Predefined Process Menu FilterGame() dan FilterGenre()



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. PilihanCheckbox

Fungsi utama yang digunakan di beberapa fungsi lainnya, seperti untuk memilih beberapa filter genre.

Source Code:

```
void PilihanCheckbox(string selectedData[], int &selectedDataCount, string
options[], int optionCount, string title)
{
    string _temp;
    int selectedIndex = -1;

    while (true)
    {
        cout << title << endl;
        for (int i = 0; i < optionCount; i++)
        {
            bool isSelected = false;

            for (int j = 0; j < selectedDataCount; j++)
            {
                if (options[i] == selectedData[j])
                {
                    isSelected = true;
                    break;
                }
            }
            cout << i + 1 << ". " << options[i] << (isSelected ? " (Dipilih)" :
            "") << endl;
```

```

    }

    cout << endl;
    cout << "(Kosongkan input untuk mengakhiri pilihan.)" << endl;
    cout << "Pilih pilihan: ";
    getline(cin, _temp);

    if (_temp == "")
        break;

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

    if (selectedIndex >= 0 && selectedIndex < optionCount)
    {
        bool alreadySelected = false;

        for (int i = 0; i < selectedDataCount; i++)
        {
            if (selectedData[i] == options[selectedIndex])
            {
                for (int j = i; j < selectedDataCount - 1; j++)
                {
                    selectedData[j] = selectedData[j + 1];
                }

                selectedDataCount--;
                alreadySelected = true;
                break;
            }
        }

        if (!alreadySelected && selectedDataCount < SMOL_DATA)
        {
            selectedData[selectedDataCount] = options[selectedIndex];
            selectedDataCount++;
        }
    }
    else
    {
        cout << "Pilihan tidak valid!" << endl;
    }
}
}

```


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:

```
void Login(int &batasPercobaan, int &percobaanGagal, User *users, int userCount,
User *currentUser)
{
    string inputUsername = "";
    string inputPassword = "";

    do
    {
        cout << "Username\t: ";
        getline(cin, inputUsername);
        if (inputUsername == "")
            break;

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

        // Proses validasi login
        *currentUser = GetUserLogin(users, userCount, inputUsername,
inputPassword);

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

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:

```
void Register(User *userPtr, int &userCount)
{
    User newUser;
    string _temp = "";

    if (userCount < MAX_DATA)
    {
        // Proses registrasi dapat berjalan jika user belum mencapai maks
        cout << "Registrasi" << endl;
        cout << endl;
        cout << "Username\t: ";
        getline(cin, newUser.username);

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

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

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

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

        // Default Value
        newUser.id = userCount + 1;
        newUser.level = "user";

        // Menambahkan user baru
        userPtr[userCount] = newUser;
        userCount++;
    }
}
```

```

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

```

D. 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:

```

void MenuSistemLogin(int &batasPercobaan, int &percobaanGagal, User *usersPtr,
int &userCount, User *currentUser)
{
    string _temp;

    // Sistem Login
    Border("Sistem Pengelolaan Mod");

    cout << "1 > Login" << endl;
    cout << "2 > Register" << endl;
    Border("-", 106);
    cout << "Q > Keluar" << endl;
    cout << endl;
    cout << "Pilih Menu\t: ";
    getline(cin, _temp);
    Border("-", 106);
    if (_temp == "1")
    {
        Login(batasPercobaan, percobaanGagal, usersPtr, userCount, currentUser);
    }
    else if (_temp == "2")
    {
        Register(usersPtr, userCount);
    }
    else if (_temp != "Q" && _temp != "q")
    {
        cout << "Menu tidak ditemukan!";
        getline(cin, _temp);
    }
}

```

E. 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:

```
void MenuUtama(char &halaman, string &pilihan_menu, User *currentUser)
{
    string _temp = "";

    // Menu Utama
    cout << "Menu Utama" << endl;
    Border("-", 106);
    cout << "Selamat datang, " << currentUser->username << "!" << endl;
    cout << endl;
    cout << "1 > Games" << endl;
    cout << "2 > Mods" << endl;
    Border("-", 106);
    cout << "Q > Keluar | L > Logout" << endl;
    cout << endl;
    cout << "Pilih Menu\t: ";
    getline(cin, _temp);

    if (_temp == "1")
    {
        halaman = '1';
    }
    else if (_temp == "2")
    {
        halaman = '2';
    }
    else if ((_temp == "Q" || _temp == "q") || (_temp != "B" || _temp != "b") ||
(_temp == "L" || _temp == "l"))
    {
        pilihan_menu = _temp;
    }
    else
    {
        cout << "Menu tidak ditemukan!";
        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:

```
void TambahGame(Game *gamesPtr, int &gameCount, GabunganArray *dataArrayPtr, int
&genreCount)
{
    string _temp = "";
    Game gameBaru;

    if (gameCount <= MAX_DATA)
    {
        Border("-", 106);
        cout << "Tambah Game" << endl;
        Border("-", 106);

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

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

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

        // Input Genre
        string genres[SMOL_DATA];
        for (int i = 0; i < genreCount; i++)
        {
            genres[i] = dataArrayPtr[i].genre;
        }
        PilihanCheckbox(gameBaru.genre, gameBaru.genreCount, genres, genreCount,
"Genre");
        cout << endl;
```

```

        // Prsoses penambahan game
        gamesPtr[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:

```

void EditGame(Game gameListMenu[], int gameListMenuCount, Game *gamesPtr, int
&gameCount, GabunganArray *dataArrayPtr, int &genreCount)
{
    string _temp = "";
    int _selectedIndex = -1;

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

    if (_temp == "")
        return;

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

```

```

if (gameCount > _selectedIndex >= 0)
{
    // Lanjutkan proses edit game jika indeks valid
    Game gameBaru = gamesPtr[_selectedIndex];
    Border("-", 106);
    cout << "Edit Game \"" << gamesPtr[_selectedIndex].nama << "\"" << endl;
    Border("-", 106);

    // 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 != "" && gamesPtr[i].nama == _temp && gameBaru.nama !=
_temp)
        {
            _temp = "";
            cout << "Game sudah ada!";
            getline(cin, _temp);
            break;
        }
    }

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

    // Input Genre
    string genres[SMOL_DATA];
    for (int i = 0; i < genreCount; i++)
    {
        genres[i] = dataArrayPtr[i].genre;
    }
    PilihanCheckbox(gameBaru.genre, gameBaru.genreCount, genres, genreCount,
"Genre");
    cout << endl;

    // Proses perubahan data game
    gamesPtr[_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:

```
void HapusGame(Game gameListMenu[], int gameListMenuCount, Game *gamesPtr, int &gameCount, Mod *mods, int &modCount)
{
    string _temp = "";
    int _selectedIndex = -1;

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

    if (_temp == "")
        return;

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

    if (gameCount > _selectedIndex >= 0)
    {
        // Lanjutkan proses penghapusan game jika indeks valid
        cout << endl;
        cout << "Apakah Anda yakin ingin menghapus \"" <<
gamesPtr[_selectedIndex].nama << "\"? [Y/N]\t: ";
        getline(cin, _temp);

        if (_temp == "Y" || _temp == "y")
        {
            // Proses penghapusan game
            for (int i = _selectedIndex; i < gameCount - 1; i++)
            {
```



```

        gamesPtr[i] = gamesPtr[i + 1];
    }
    gameCount--;
    cout << "Game berhasil dihapus!";
    getline(cin, _temp);

    for (int i = 0; i < modCount; i++)
    {
        if (mods[i].game->nama == gamesPtr[_selectedIndex].nama)
        {
            mods[i].game = new Game();
        }
    }
}
else
{
    // Batalkan proses penghapusan game
    cout << "Game dibatalkan dihapus!";
    getline(cin, _temp);
}
}
else
{
    cout << "Game tidak ditemukan!";
    getline(cin, _temp);
}
}
}

```

I. 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:

```

void MenuGame(char &halaman, string &pilihan_menu, Game &selectedGame, string
selectedGenres[], int &selectedGenreCount, Game *gamesPtr, int &gameCount, Mod
*modsPtr, int &modCount, GabunganArray *dataArrayPtr, int &genreCount, User
*currentUser)
{
    string _temp;
    int i;

    Game gameListMenu[MAX_DATA];
    int gameListMenuCount = 0;

    selectedGame = Game();
}

```

```

// Menduplikat data game
for (int i = 0; i < gameCount; i++)
{
    gameListMenu[gameListMenuCount] = gamesPtr[i];
    gameListMenuCount++;
}

// Proses pemfilteran game
i = 0;
while (i < gameListMenuCount && selectedGenreCount > 0)
{
    bool genreDitemukan = false;

    // Mencocokkan Genre
    for (int j = 0; j < gameListMenu[i].genreCount; j++)
    {
        for (int k = 0; k < selectedGenreCount; k++)
        {
            if (gameListMenu[i].genre[j] == selectedGenres[k])
            {
                genreDitemukan = true;
                break;
            }
        }
        if (genreDitemukan)
            break;
    }

    // Mengeliminasi Game jika tidak memiliki Genre yang dicari
    if (!genreDitemukan)
    {
        for (int l = i; l < MAX_DATA - 1; l++)
        {
            gameListMenu[l] = gameListMenu[l + 1];
        }
        gameListMenuCount--;
    }
    else
        i++;
}

// Mengosongkan sisa data game
for (int i = gameListMenuCount; i < sizeof(gameListMenu) / sizeof(Game);
i++)
    gameListMenu[i] = Game();

// Menampilkan Menu Games
cout << "Games" << endl;
Border("-", 106);
if (currentUser->level == "admin")
{

```

```

        cout << "C > Tambah Game | U > Edit Game | D > Hapus Game | ";
    }
    cout << "F > Filter Game" << endl;
    Border("-", 106);
    if (gameListMenuCount > 0)
    {
        cout << left << setw(6) << "No."
            << setw(40) << "Game"
            << endl;
        Border("-", 106);

        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;
    }
    Border("-", 106);
    cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
    cout << endl;
    cout << "Pilih Menu\t: ";
    getline(cin, _temp);

    // Mencari pilihan game
    for (int i = 0; i < gameCount; i++)
    {
        if (to_string(i + 1) == _temp)
        {
            selectedGame = gameListMenu[i];
            break;
        }
    }

    if (selectedGame.nama != "")
    {
        // Pindah ke Menu Mod yang terkait dengan game
        halaman = '2';
        return;
    }

    // List Pilihan
    if (currentUser->level == "admin" && (_temp == "C" || _temp == "c"))
    {
        // Tambah Game
        TambahGame(gamesPtr, gameCount, dataArrayPtr, genreCount);
    }

```

```

    else if (currentUser->level == "admin" && _temp == "U" || _temp == "u")
    {
        // Edit Game
        EditGame(gameListMenu, gameListMenuCount, gamesPtr, gameCount,
dataArrayPtr, genreCount);
    }
    else if (currentUser->level == "admin" && _temp == "D" || _temp == "d")
    {
        // Hapus Game
        HapusGame(gameListMenu, gameListMenuCount, gamesPtr, gameCount, modsPtr,
modCount);
    }
    else if (_temp == "F" || _temp == "f")
    {
        // Pindah menu filter
        halaman = '3';
        return;
    }
    else if ((_temp == "Q" || _temp == "q") || (_temp != "B" || _temp != "b") ||
(_temp == "L" || _temp == "l"))
    {
        pilihan_menu = _temp;
    }
    else
    {
        cout << "Menu tidak ditemukan!";
        getline(cin, _temp);
    }
}

```

J. 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:

```

void MenuFilterGame(char &halaman, string &pilihan_menu, string
selectedGenres[], int &selectedGenreCount, GabunganArray *dataArrayPtr, int
&genreCount)
{
    string _temp = "";

    cout << "Filter Game" << endl;
    Border("-", 106);
    cout << "1. Genre" << endl;
    Border("-", 106);

```

```

cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);
Border("-", 106);

// List Pilihan
if (_temp == "1")
{
    // Filter Genre
    string genres[SMOL_DATA];
    for (int i = 0; i < genreCount; i++)
    {
        genres[i] = dataArrayPtr[i].genre;
    }
    PilihanCheckbox(selectedGenres, selectedGenreCount, genres, genreCount,
"Genre");
    cout << endl;
}
else if ((_temp == "Q" || _temp == "q") || (_temp != "B" || _temp != "b") ||
_temp == "L" || _temp == "l"))
{
    pilihan_menu = _temp;
}
else
{
    cout << "Menu tidak ditemukan!";
    getline(cin, _temp);
}
}

```

K. 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:

```

string genres[SMOL_DATA];
for (int i = 0; i < genreCount; i++)
{
    genres[i] = dataArrayPtr[i].genre;
}
PilihanCheckbox(gameBaru.genre, gameBaru.genreCount, genres, genreCount,
"Genre");
cout << endl;

```

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:

```
void UploadMod(Mod *modsPtr, int &modCount, Game *gamesPtr, int &gameCount, User
*currentUser)
{
    string _temp = "";

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

        Border("-", 106);
        cout << "Upload Mod" << endl;
        Border("-", 106);
        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 == modsPtr[i].nama)
            {
                modBaru.nama = "";
                cout << "Nama mod sudah ada!";
                getline(cin, _temp);
                break;
            }
        }

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

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

```

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

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

        if (_temp == "")
            return;

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

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

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

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

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

```

```
}  
}
```

M. 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:

```
void EditMod(Mod &selectedMod, Mod *modsPtr, int modCount)
{
    string _temp = "";

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

    for (int i = 0; i < modCount; i++)
    {
        if (_temp != "" && modsPtr[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 (modsPtr[i].id == selectedMod.id)
        {
            modsPtr[i] = selectedMod;
        }
    }

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

```

N. Hapus Mod

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

Source Code:

```

void HapusMod(char &halaman, Mod &selectedMod, Mod *modsPtr, int modCount)
{
    string _temp = "";
    // Menghapus Mod saat ini
    cout << "Apakah Anda yakin ingin menghapus mod \"" << selectedMod.nama <<
    "\"? [Y/N] ";
    getline(cin, _temp);

    if (_temp == "")
        return;

    if (_temp == "Y" || _temp == "y")
    {
        // Proses penghapusan mod
        for (int i = 0; i < modCount; i++)
        {
            if (modsPtr[i].id == selectedMod.id)
            {
                for (int j = i; j < modCount; j++)
                {
                    modsPtr[j] = modsPtr[j + 1];
                }
                modCount--;
                break;
            }
        }
        cout << "Mod berhasil dihapuskan!";
        getline(cin, _temp);
        selectedMod = Mod();
    }
}

```

```

        halaman = '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);
    }
}

```

O. Menu Mod

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

Source Code:

```

void MenuMod(char &halaman, string &pilihan_menu, Game selectedGame, Mod
&selectedMod, Game *gamesPtr, int &gameCount, Mod *modsPtr, int &modCount, User
*currentUser)
{
    string _temp = "";
    Mod modListMenu[MAX_DATA];
    int modListMenuCount = 0;

    selectedMod = Mod();

    // Memfilter mod berdasarkan game yang dipilih
    for (int i = 0; i < modCount; i++)
    {
        if (selectedGame.nama != "" && modsPtr[i].game->nama !=
selectedGame.nama)
            continue;

        modListMenu[modListMenuCount] = modsPtr[i];
        modListMenuCount++;
    }

    // Mengosongkan sisa data mod
    for (int i = modListMenuCount; i < sizeof(modListMenu) / sizeof(Mod); i++)
        modListMenu[i] = Mod();

    // Menampilkan Menu Mod

```

```

if (selectedGame.nama != "")
    cout << "Mod " << selectedGame.nama << endl;
else
    cout << "Semua Mod" << endl;
Border("-", 106);
cout << "C > Upload Mod" << endl;
Border("-", 106);
if (modListMenuCount > 0)
{
    // Daftar Mod
    cout << left << setw(6) << "No."
        << setw(40) << "Nama Mod"
        << setw(40) << "Game"
        << setw(20) << "Creator"
        << endl;
    Border("-", 106);

    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;
}
Border("-", 106);
cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
cout << endl;
cout << "Pilih Menu\t: ";
getline(cin, _temp);

// Pilih Mod
for (int i = 0; i < modListMenuCount; i++)
{
    if (to_string(i + 1) == _temp)
    {
        selectedMod = modListMenu[i];
    }
}

if (selectedMod.id > 0)
{
    halaman = '4';
    return;
}

```

```

// List Pilihan
if (_temp == "C" || _temp == "c")
{
    // Upload Mod
    UploadMod(modsPtr, modCount, gamesPtr, gameCount, currentUser);
}
else if ((_temp == "Q" || _temp == "q") || (_temp != "B" || _temp != "b") ||
(_temp == "L" || _temp == "l"))
{
    pilihan_menu = _temp;
}
else
{
    cout << "Menu tidak ditemukan!";
    getline(cin, _temp);
}
}

```

P. 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:

```

void DetailMod(char &halaman, string &pilihan_menu, Mod &selectedMod, Mod
*modsPtr, int &modCount, User *currentUser)
{
    string _temp = "";

    if (selectedMod.id < 1)
    {
        halaman = '2';
        return;
    }

    // Menampilkan detail mod terpilih
    cout << selectedMod.nama << endl;
    Border("-", 106);
    if (currentUser->id == selectedMod.uploader->id)
    {
        cout << "U > Edit | D > Hapus" << endl;
        Border("-", 106);
    }
    else if (currentUser->level == "admin")
    {
        cout << "D > Hapus" << endl;
    }
}

```

```

        Border("-", 106);
    }
    cout << "Creator\t\t: " << selectedMod.creator << endl;
    cout << "Uploader\t: " << selectedMod.uploader->username << endl;
    cout << "Game\t\t: " << selectedMod.game->nama << endl;
    Border("-", 106);
    cout << selectedMod.deskripsi << endl;
    Border("-", 106);
    cout << "Q > Keluar | B > Kembali | L > Logout" << endl;
    cout << endl;
    cout << "Pilih Menu\t: ";
    getline(cin, _temp);
    Border("-", 106);

    if (currentUser->id == selectedMod.uploader->id && (_temp == "U" || _temp ==
"u"))
    {
        // Edit Mod terpilih
        EditMod(selectedMod, modsPtr, modCount);
    }
    else if ((currentUser->id == selectedMod.uploader->id || currentUser->level
== "admin") && (_temp == "D" || _temp == "d"))
    {
        // Hapus Mod terpilih
        HapusMod(halaman, selectedMod, modsPtr, modCount);
    }
    else if ((_temp == "Q" || _temp == "q") || (_temp != "B" || _temp != "b") ||
(_temp == "L" || _temp == "l"))
    {
        pilihan_menu = _temp;
    }
    else
    {
        cout << "Menu tidak ditemukan!";
        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*

```

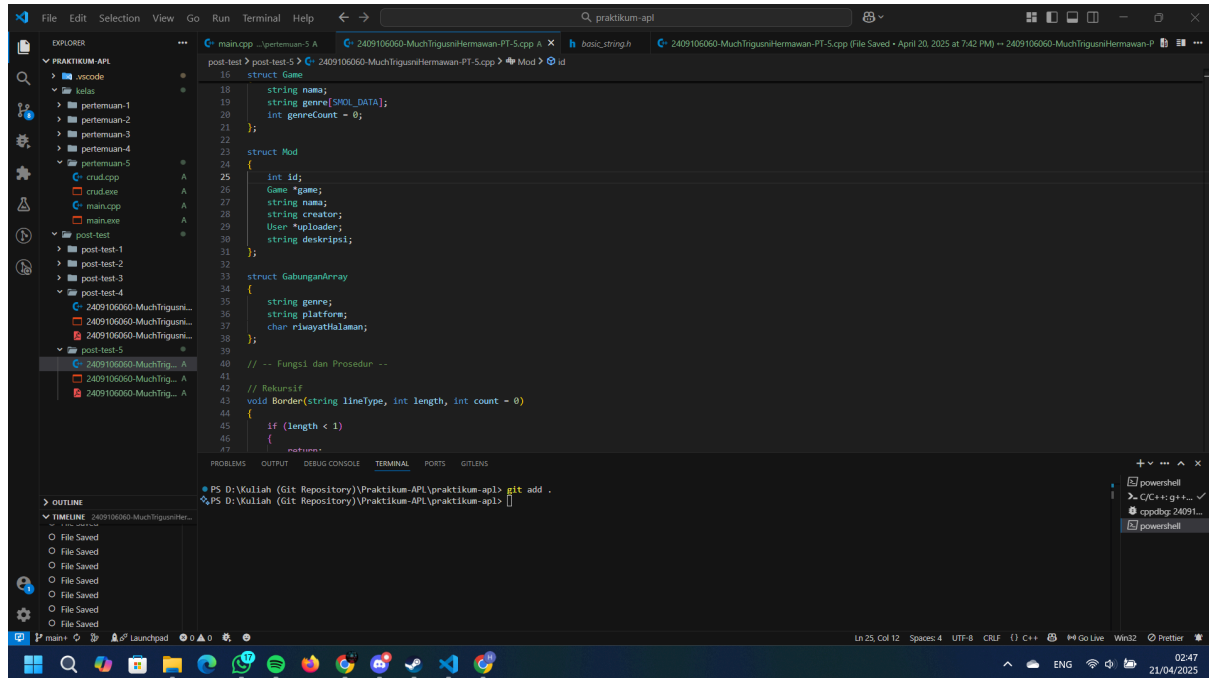
=====
Thorium Mod
=====
U > Edit | D > Hapus
=====
Creator       : DivermanSam
Uploader      : Trigusni
Game         : Terraria
=====
Deskripsi baru.
Q > Keluar | B > Kembali | L > Logout
Pilih Menu    : █

```

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

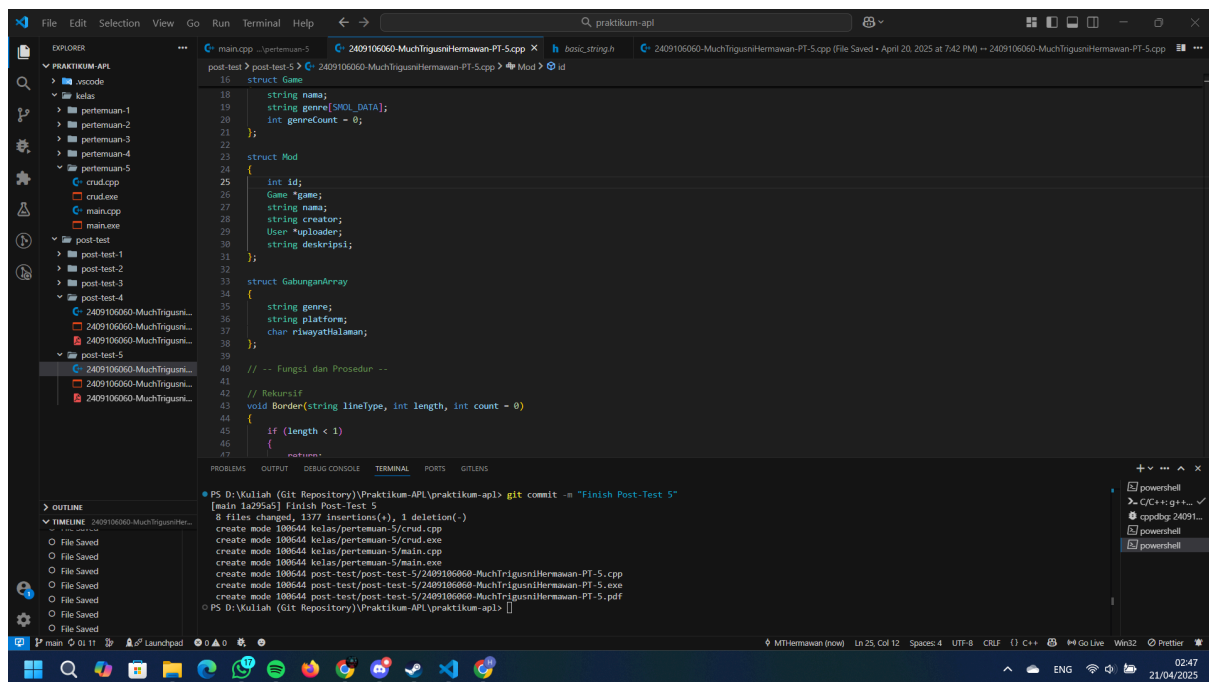
5. Langkah-Langkah Git

A. Git Add



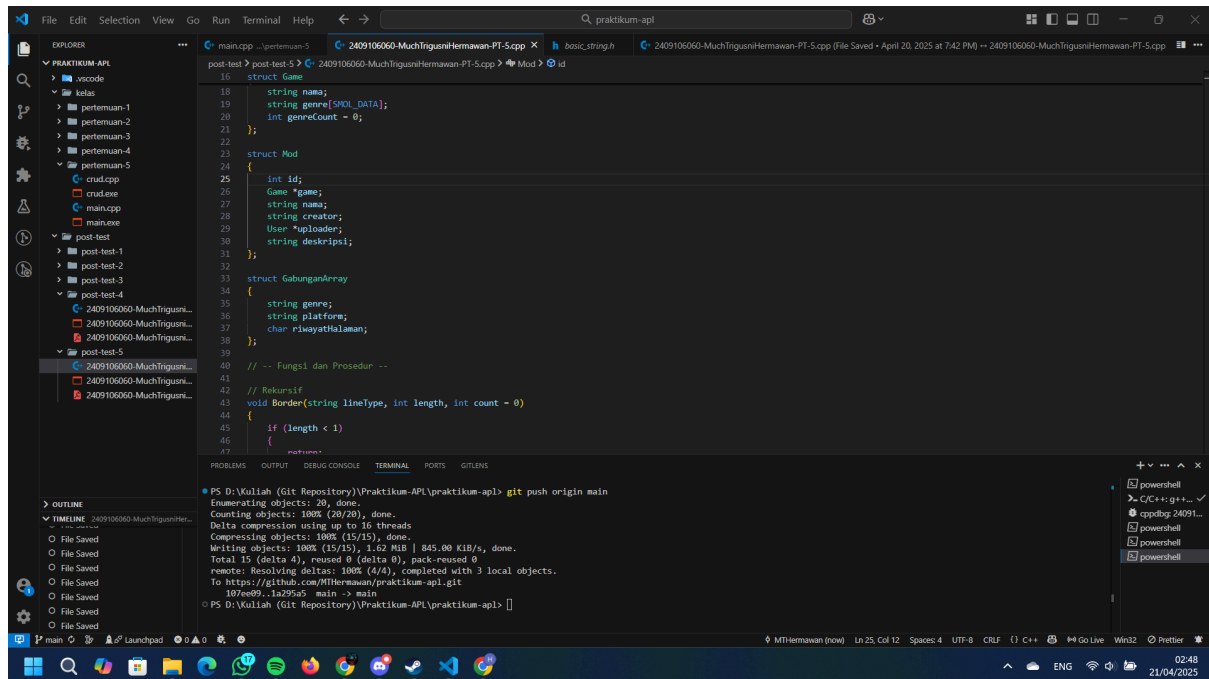
Menambahkan semua perubahan file saat ini ke commit berikutnya dengan “git add .”

B. Git Commit



Membuat checkpoint progres pada branch main dengan git commit dengan pesan “Finish Post-Test-5”.

C. Git Push



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