

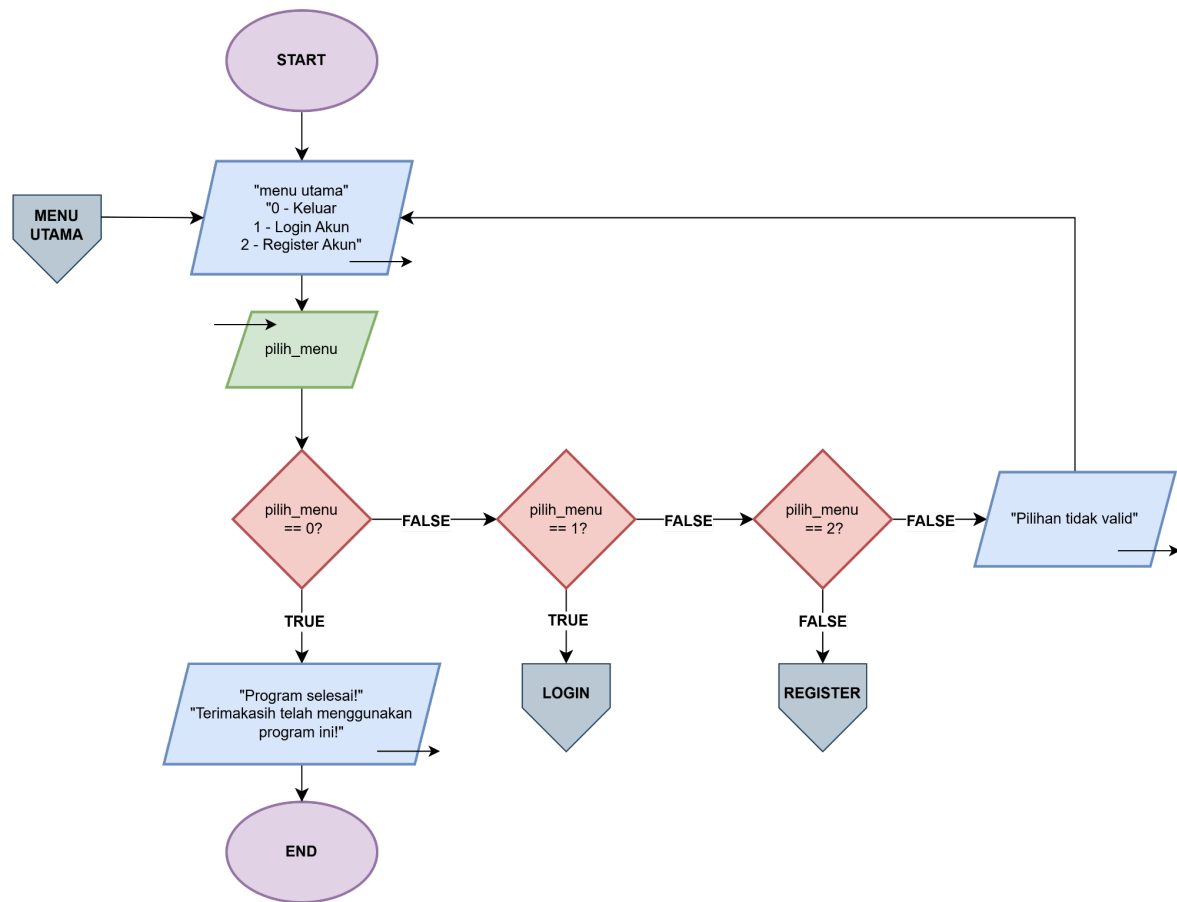
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN DASAR



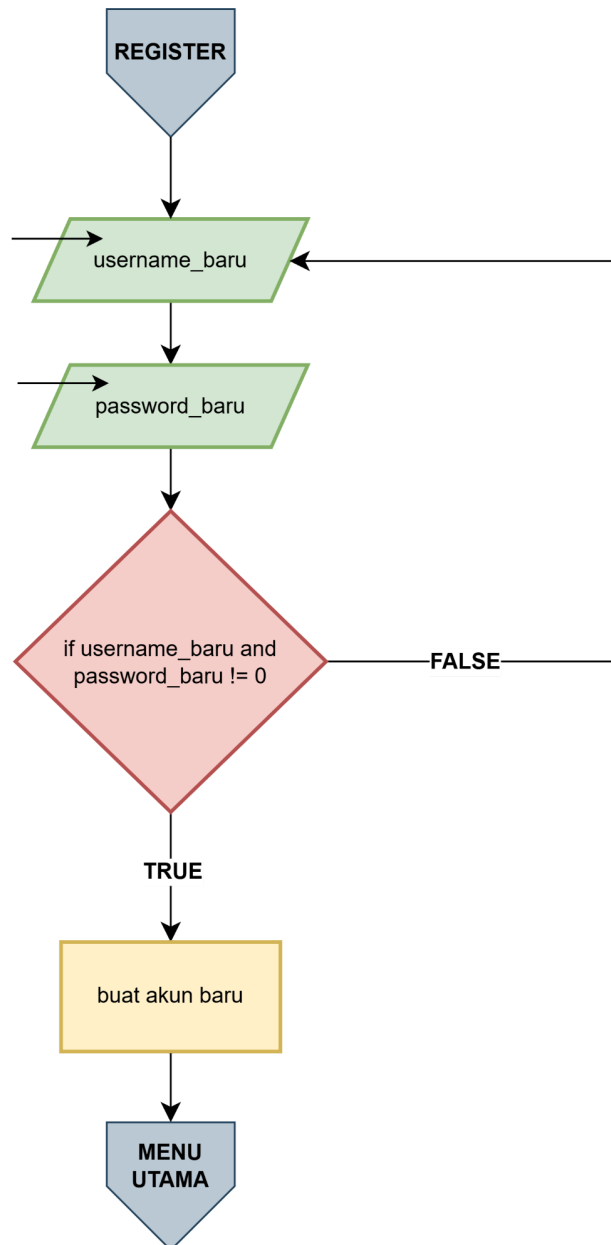
Disusun oleh:
Yoga Ananda Prasetya (2509106017)
Kelas (A1'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

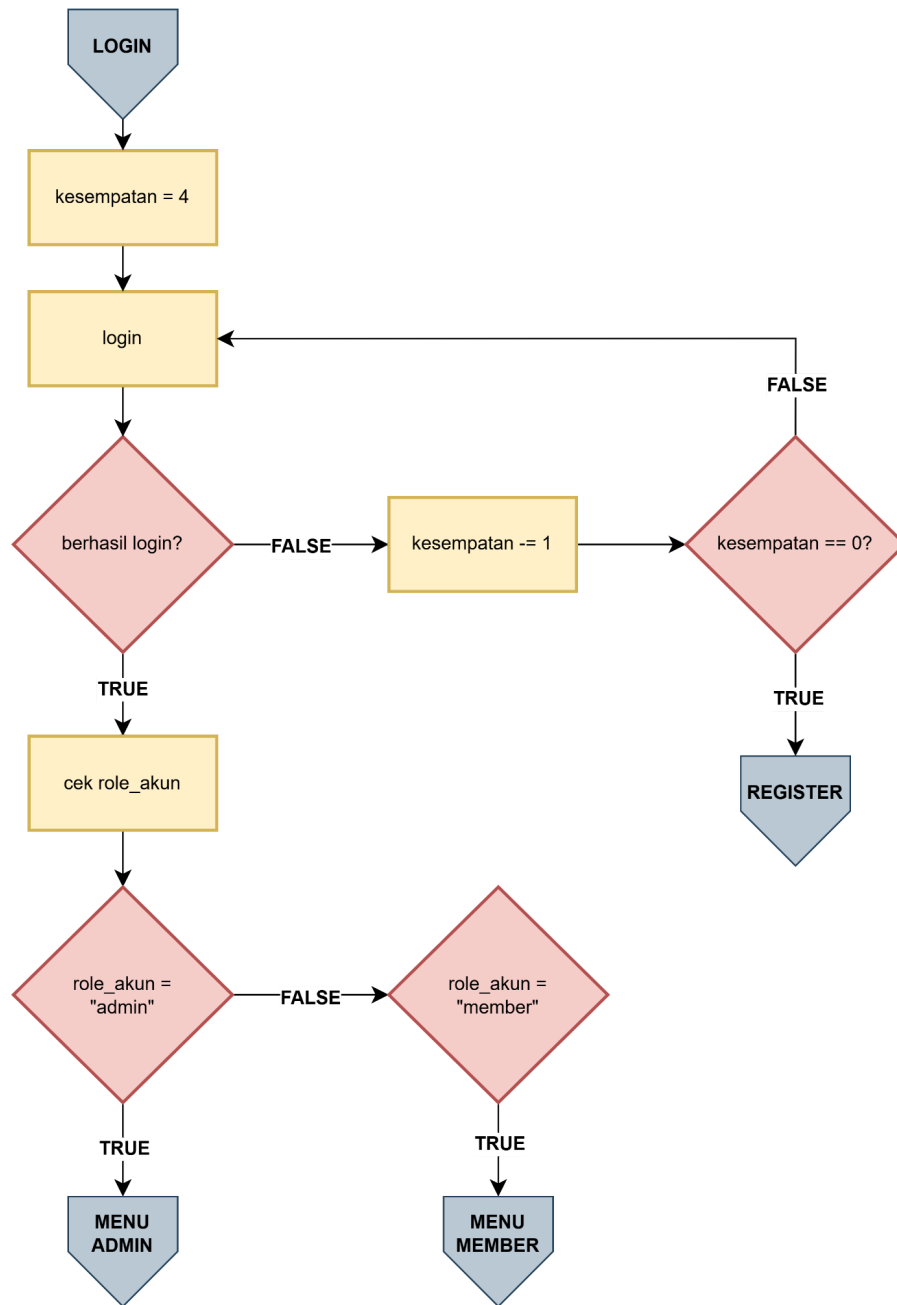
1. Flowchart



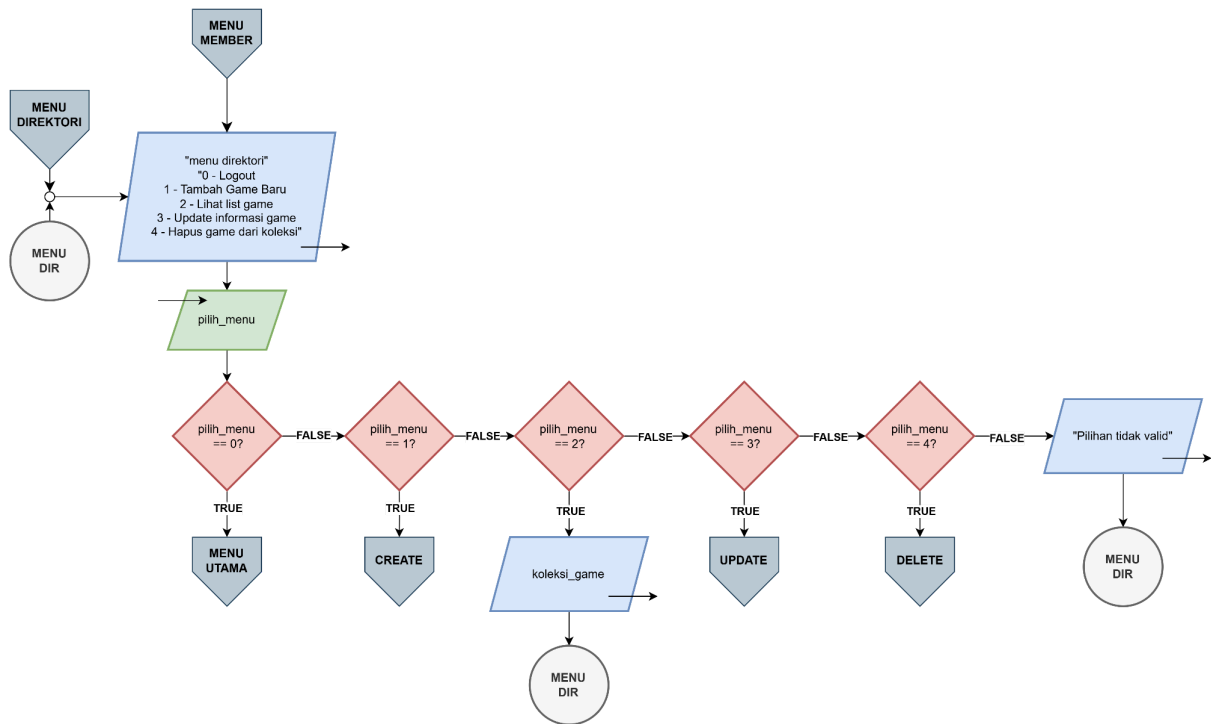
Gambar 1 Flowchart – Menu Awal



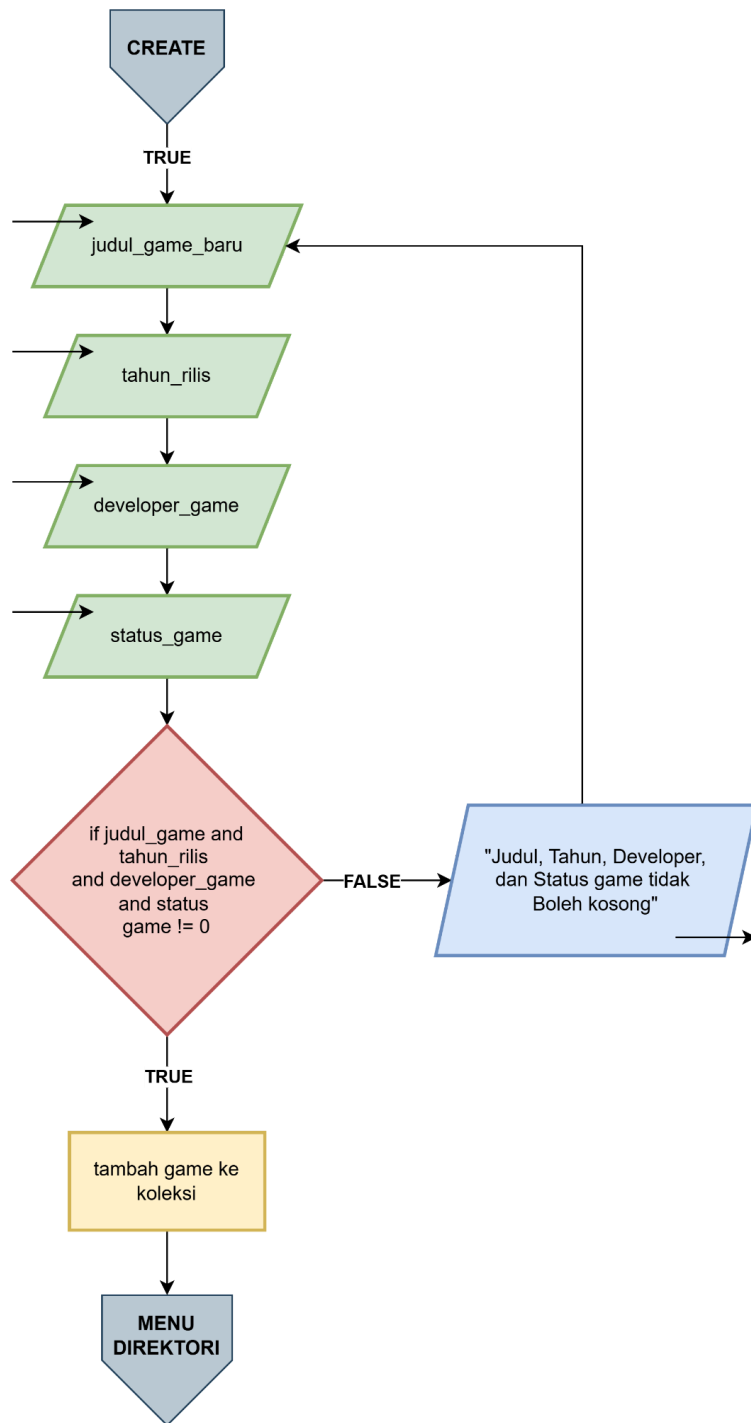
Gambar 2 Flowchart – Menu Register



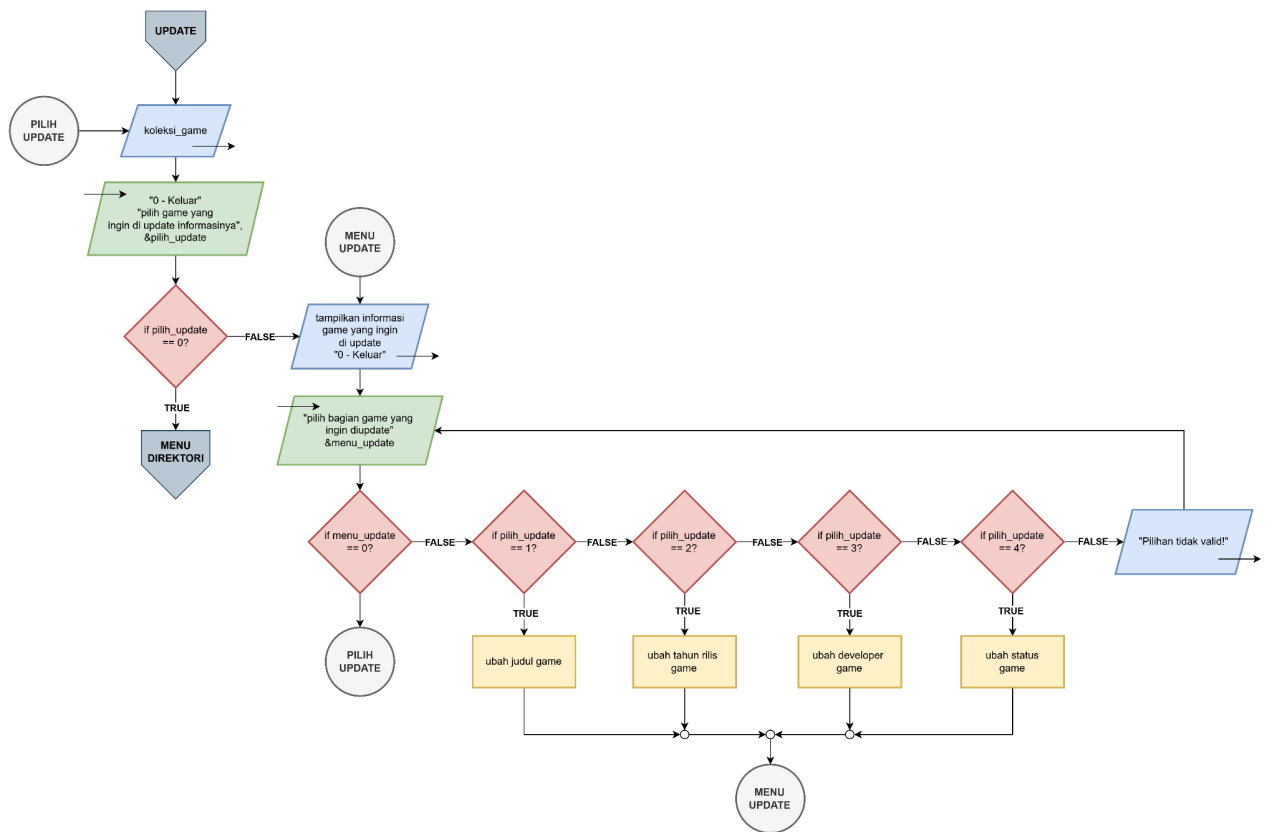
Gambar 3 Flowchart – Menu Login



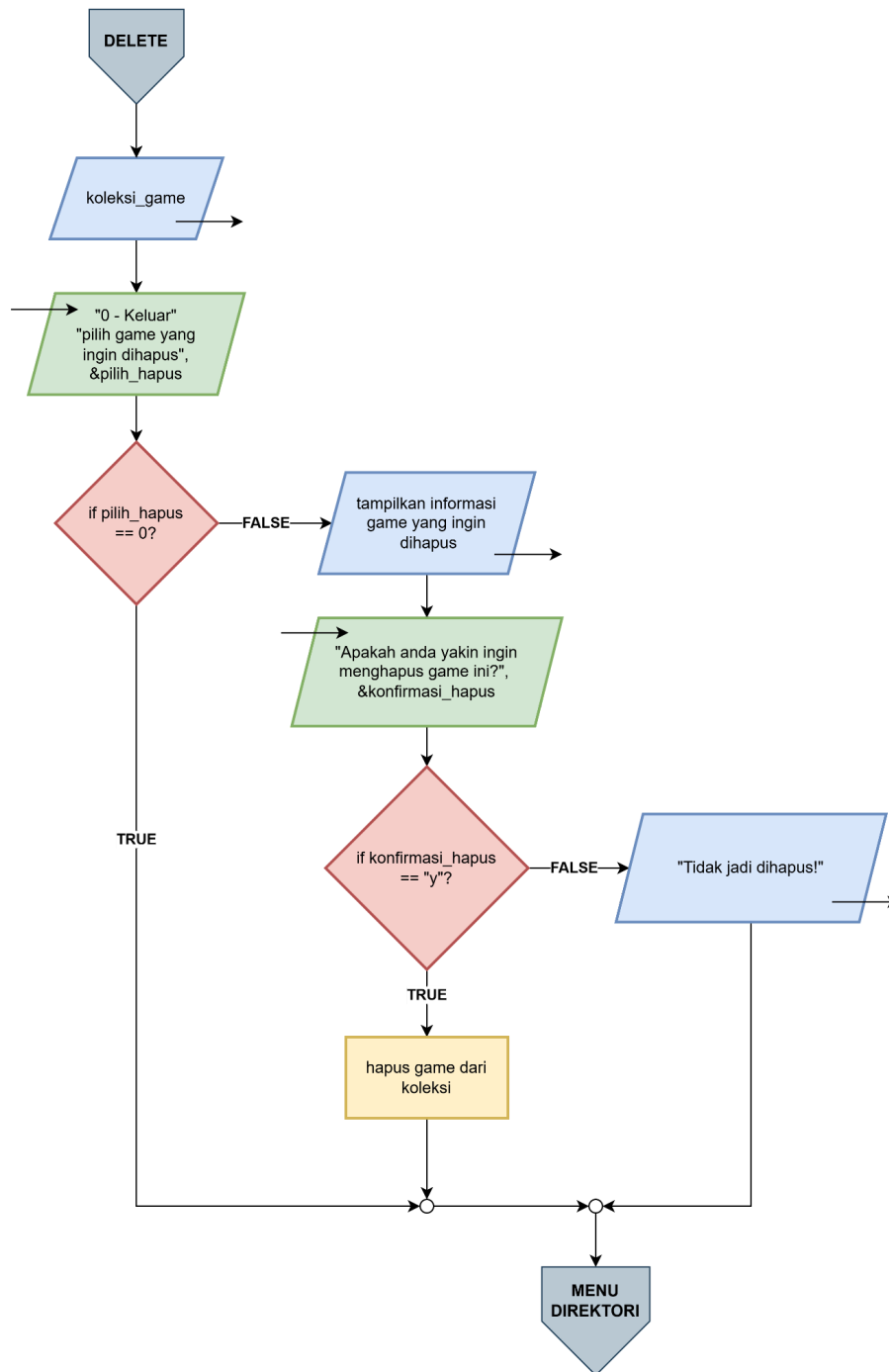
Gambar 4 Flowchart – Menu jika pengguna member



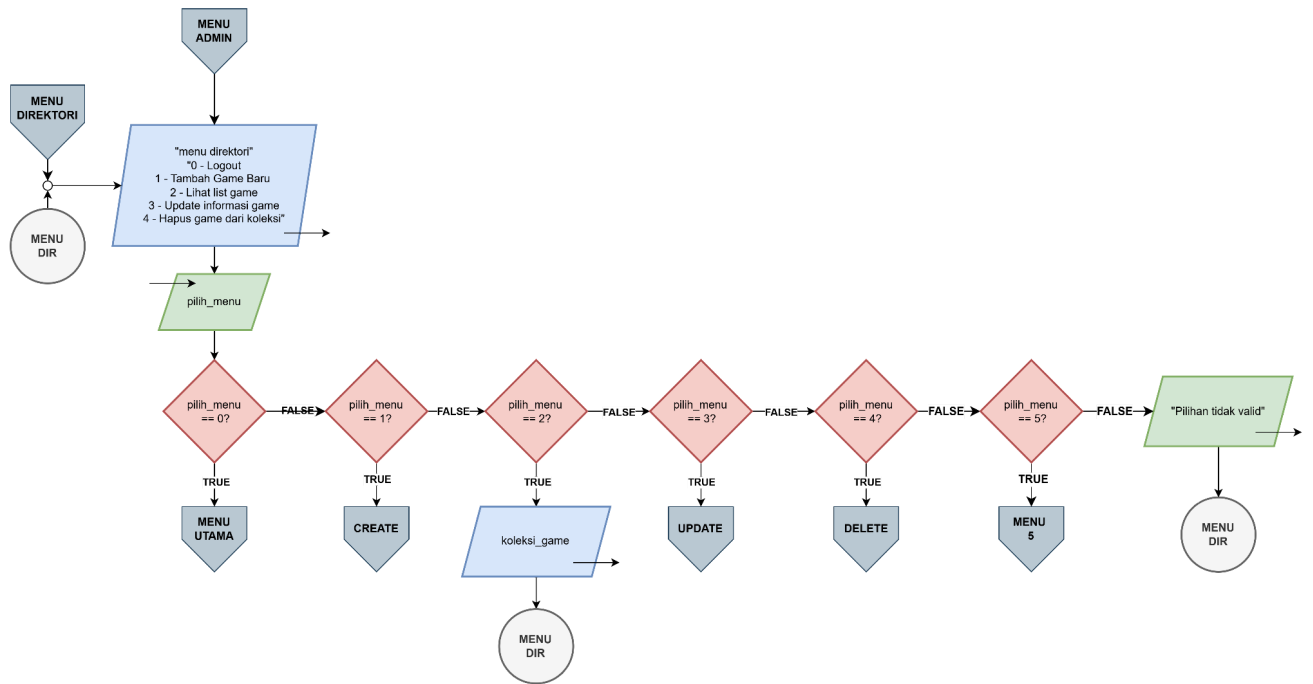
Gambar 5 Flowchart – Menambah game (Create)



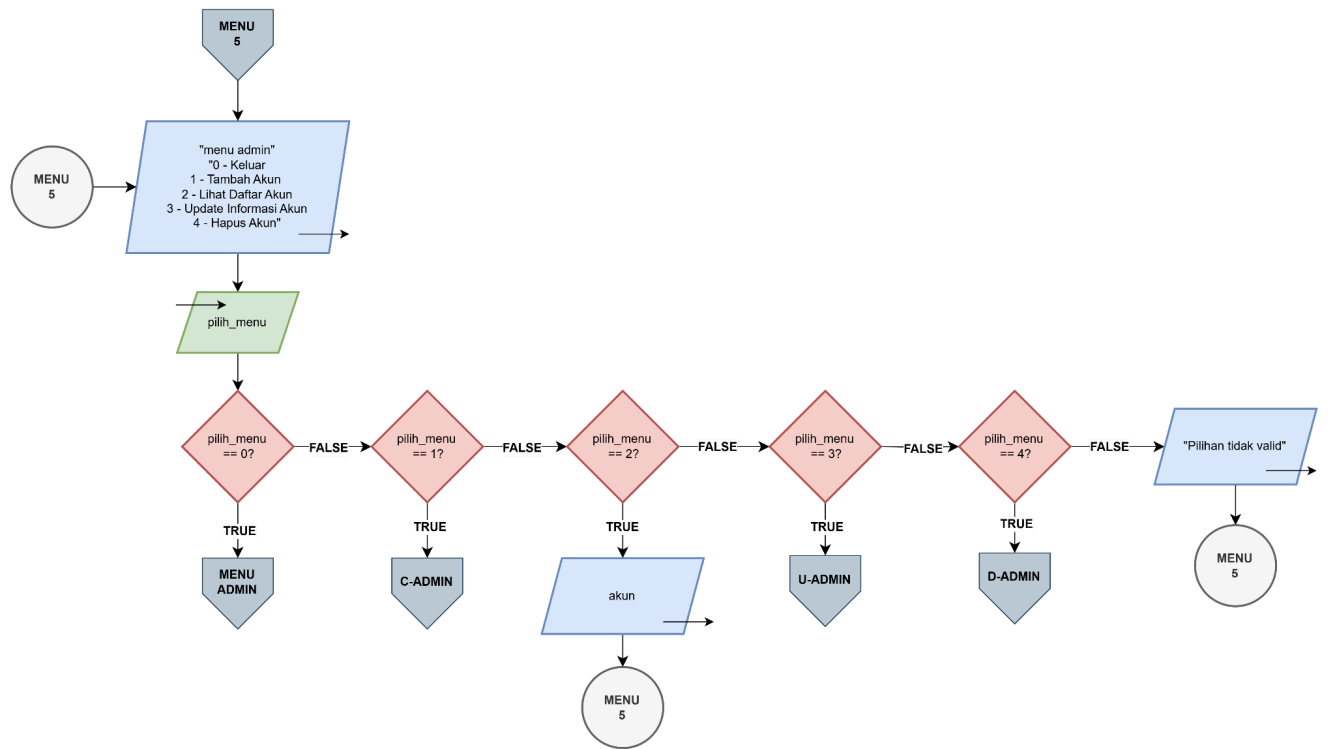
Gambar 6 Flowchart – Mengubah informasi game (Update)



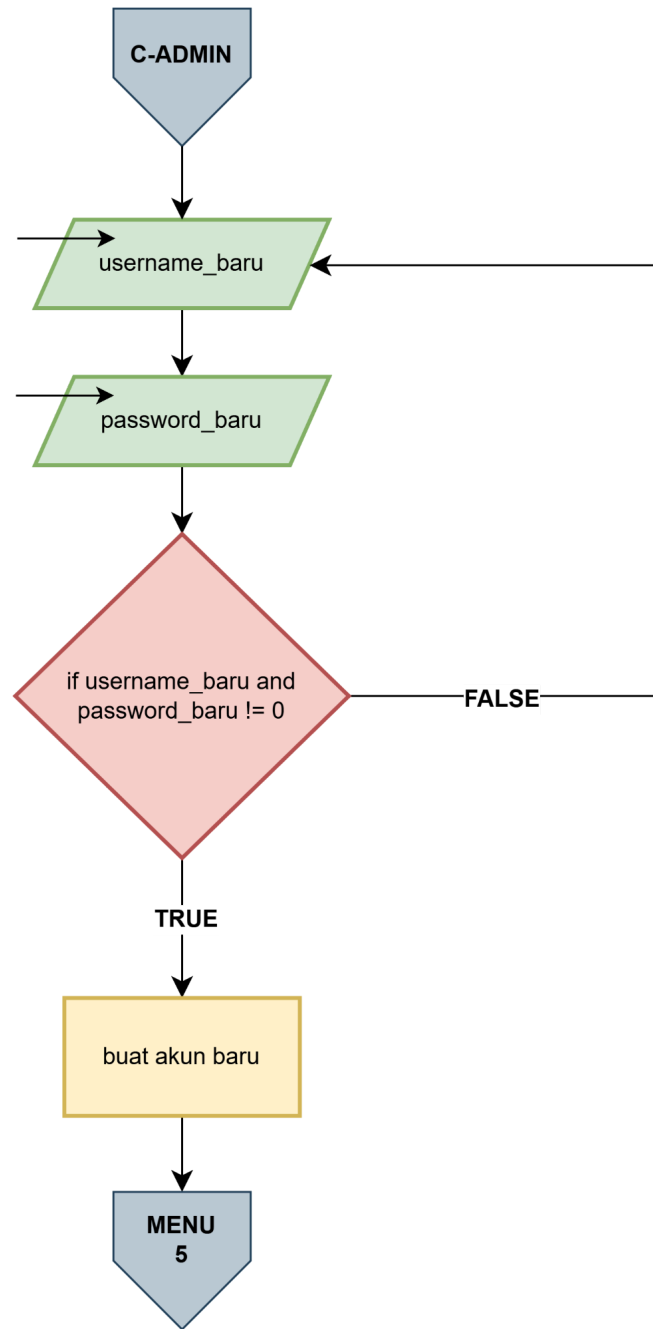
Gambar 7 – Menghapus game (Delete)



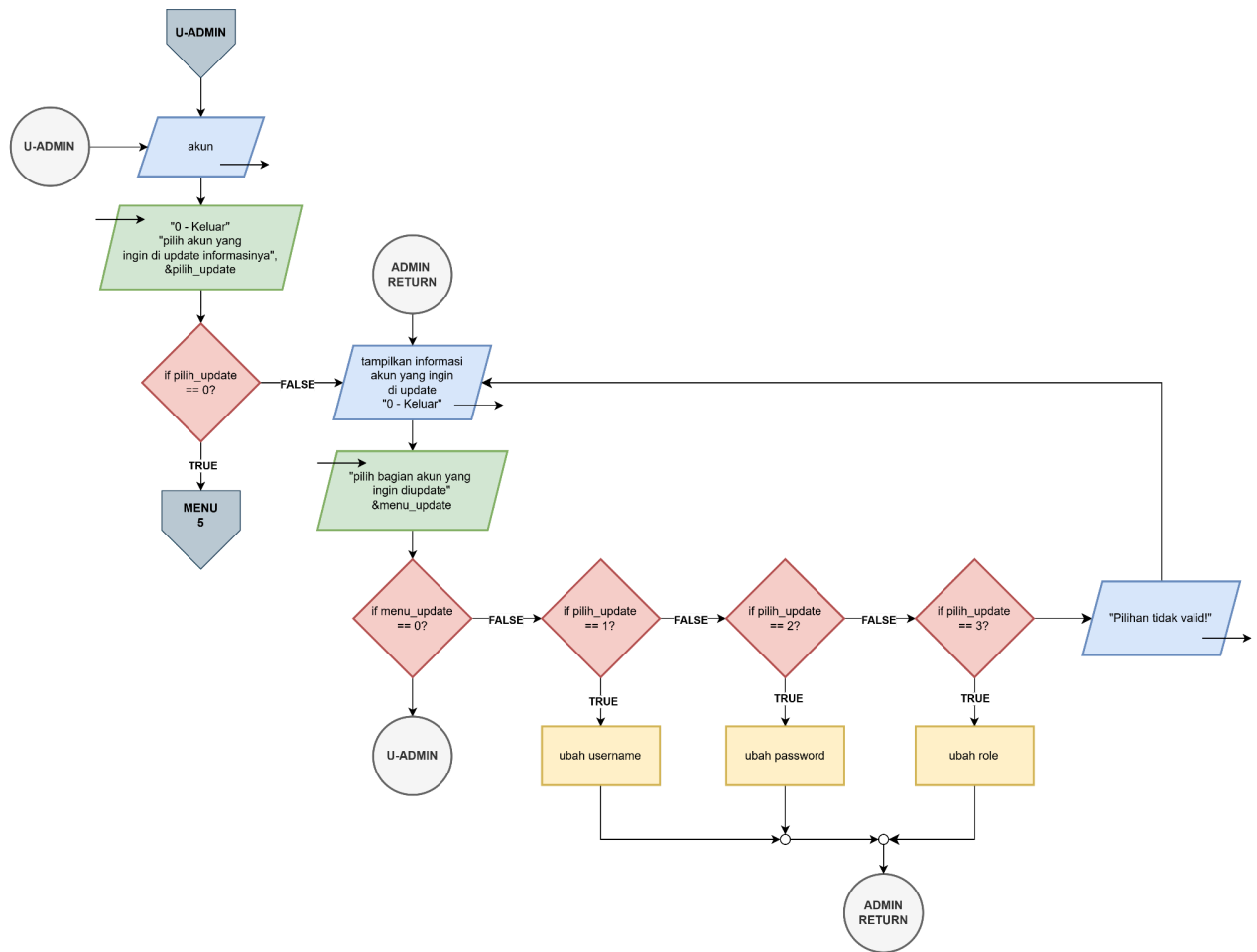
Gambar 8 – Tampilan menu jika pengguna adalah admin



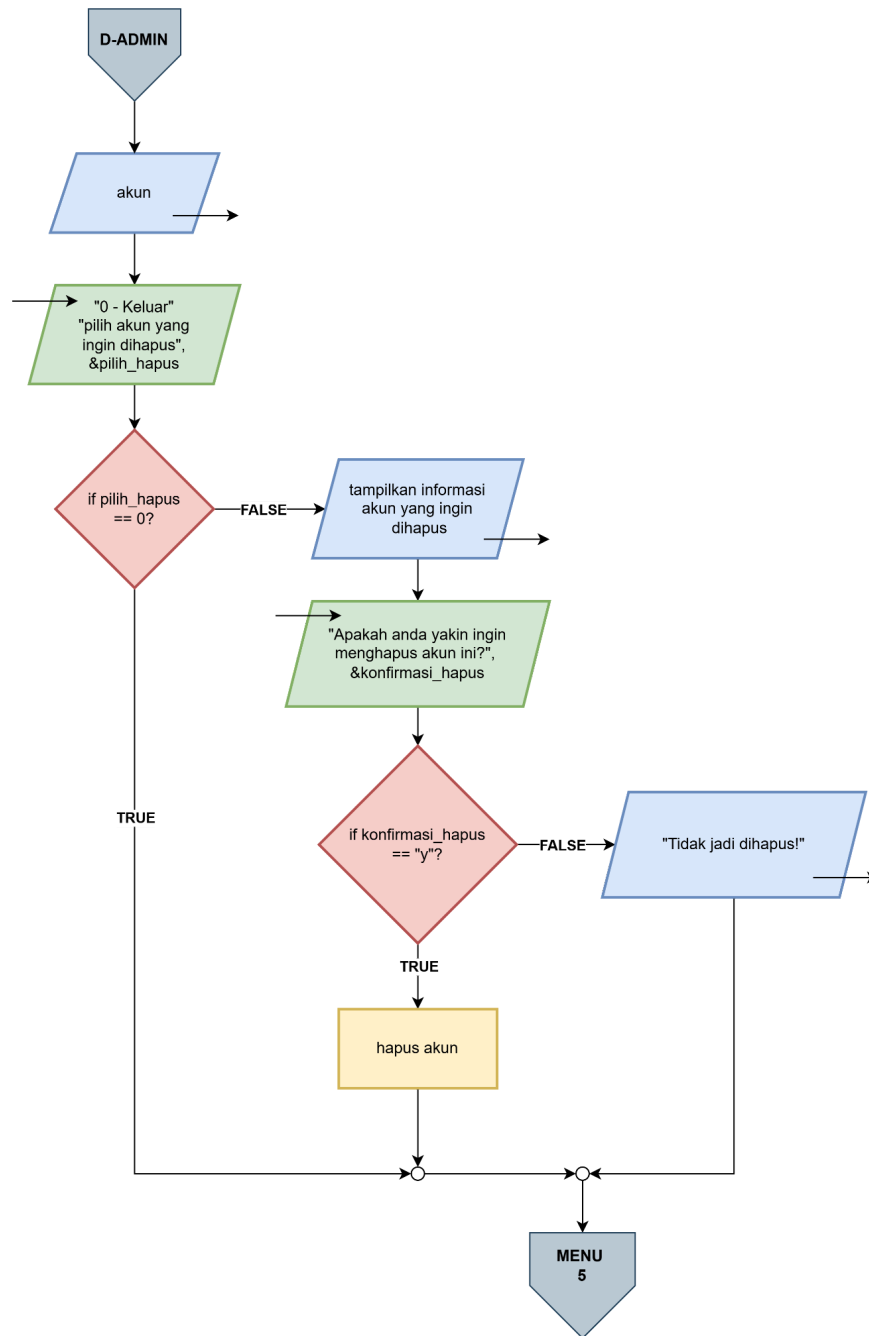
Gambar 9 – Menu Khusus admin



Gambar 10 – Menambah akun (Create)



Gambar 11 – Mengubah informasi akun (Update)



Gambar 12 – Menghapus akun (Delete)

2. Deskripsi Singkat Program

Program ini merupakan program yang menerapkan prinsip CRUD (*Create*, *Read*, *Update*, dan *Delete*), yang dimana prinsip CRUD ini diterapkan untuk membuat sebuah sistem direktori sederhana yang dapat menyimpan informasi game yang dimiliki oleh seseorang. Berikut ini merupakan implementasi dari setiap prinsip CRUD pada sistem direktori penyimpanan game.

PRINSIP CRUD		IMPLEMENTASI PRINSIP CRUD
<i>Create</i>	:	<p>Pengguna yang terdaftar sebagai member dapat menambahkan game yang diinginkan ke dalam direktori. Disini ada 4 data yang harus dimasukkan pengguna ketika ingin menambah game baru ke direktori. 4 Data tersebut yaitu sebagai berikut :</p> <ol style="list-style-type: none"> 1. Nama game 2. Tahun rilis game 3. Pengembang/<i>Developer</i> game 4. Status game (sudah tamat atau belum)
<i>Read</i>	:	<p>Pengguna dapat melihat daftar game yang sudah ditambahkan ke direktori. Pengguna dapat melihat keempat data yang telah dimasukkan pada proses <i>create</i>.</p>
<i>Update</i>	:	<p>Pengguna dapat mengganti data game yang ada di direktori jika seandainya pengguna salah memasukkannya ketika sedang dalam proses <i>create</i>. Program juga dilengkapi fitur memilih bagian data mana yang akan diganti, sehingga pengguna tidak perlu repot memasukkan keempat data tersebut jikalau hanya satu data saja yang bermasalah.</p>
<i>Delete</i>	:	<p>Pengguna dapat menghapus game yang terdapat di dalam direktori. Proses <i>Delete</i> berlaku secara permanen, artinya data game akan hilang selamanya jika pengguna memilih untuk menghapusnya. Namun, program dilengkapi mekanisme pengaman yang memastikan kembali apakah pengguna ingin menghapus game yang dipilih.</p>

Dalam program terdapat pembagian *role* atau wewenang, yaitu *admin* dan *member*. Untuk *member* sendiri hanya memiliki hak untuk menambah (*Create*), melihat (*Read*), mengubah (*Update*), dan menghapus (*Delete*) game yang ada di direktori. Sedangkan *admin* memiliki wewenang serta hak yang lebih banyak, seperti membuat akun baru (diluar dari register akun), melihat data akun yang aktif, mengubah informasi akun (seperti password,

username atau bahkan role), serta menghapus akun. Jadi *admin* memiliki dua implementasi CRUD. Implementasi pertama adalah untuk menambah, melihat, mengubah dan menghapus koleksi game yang ada di direktorinya, sedangkan implementasi kedua adalah untuk menambah, melihat, mengubah serta menghapus akun yang terdaftar pada direktori.

3. Source Code

3.1. Fitur Utama

3.1.1. Import Library dan Deklarasi Dictionary

Program ini diawali dengan *import* library serta deklarasi dictionary yang akan digunakan. Library yang digunakan adalah *os* dan *time*, keduanya berperan pada proses membersihkan terminal agar hasil *output* terlihat bersih dan rapi. Hanya ada satu dictionary yang digunakan untuk program ini, yaitu dictionary yang digunakan untuk menyimpan data-data akun pengguna.

Source Code:

```
# Import library yang diperlukan
import os
import time

# Deklarasi Dictionary yang diperlukan
akun = {
    "acc1": {
        "username": "Chryse",
        "password": "79-Au",
        "role": "admin",
        "game": {
            "g1": {"nama": "Armored Core Verdict Day", "tahun": "2007", "dev": "Fromsoftware", "status": "Tamat"},
            "g2": {"nama": "Armored Core 4: For Answer", "tahun": "2008", "dev": "Fromsoftware", "status": "Belum tamat"}
        }
    },
    "acc2": {
        "username": "Dapupu",
        "password": "Ketua_Nibung",
        "role": "member",
        "game": {
            "g1": {"nama": "MudRunner", "tahun": "2017", "dev": "Saber Interactive", "status": "Belum Tamat"},

```

```
    }  
  }  
}
```

3.1.2. Menu Awal

Sebelum menggunakan program, pengguna harus login terlebih dahulu. Disini program juga memberikan menu register akun yang dapat digunakan pengguna untuk membuat akun terlebih dahulu jika tidak punya akun.

Source Code:

```
# Tampilan Header Program  
print("=" * 80)  
print("{:^80}".format("PROGRAM DIREKTORI GAME"))  
print("=" * 80)  
  
# Tampilan Menu  
print("1 - Login Akun")  
print("2 - Register Akun")  
print("=" * 80)  
  
# Tampilan Opsi Tambahan  
print("0 - Akhiri Program")  
print("=" * 80)  
  
# Input Pilihan Pengguna  
print("Silakan pilih menu yang anda inginkan:")  
aksi = input("> ")  
print("=" * 80)
```

3.1.3. Fitur Membersihkan Terminal

Fitur ini tersebar di seluruh kode program. Fitur ini berfungsi untuk membersihkan hasil output yang ada di terminal. Tujuannya adalah Terminal menjadi rapi dan memuat informasi yang benar-benar diperlukan oleh pengguna. Fitur ini hanya terdiri dari dua baris kode, yaitu `time.sleep()` dan `os.system()`. Kode `time.sleep()` berfungsi untuk menjeda program sesuai dengan waktu yang telah ditentukan sebelumnya, sedangkan kode `os.system()` berfungsi untuk menghapus semua output yang terdapat di terminal.

Tidak seperti di program sebelum, fitur ini ditambah satu baris kode dengan tujuan agar fitur ini bisa berjalan di *Operating System* yang berbeda.

Source Code:


```
time.sleep(1)
os.system("clear")
os.system("cls")
```

3.2. Fitur yang dimiliki oleh Admin dan User

3.2.1. Menu Direktori

Bagian ini merupakan penghubung dari setiap proses yang ada pada CRUD. Di bagian menu utama ini pengguna dapat memilih menu yang diinginkan.

Source Code:

```
# ... (Ada kode dibagian sini)

# Bagian Display Menu
print("=" * 74)
print(f"Selamat datang kembali, {username_login}")
print(f"Anda login sebagai: {role_anda}")
print("=" * 74)

# Display Menu
print("1 - Tambah Game Baru")
print("2 - Lihat List Game")
print("3 - Update Informasi Game")
print("4 - Hapus Game dari Direktori")
print("=" * 74)

# Display Opsi Tambahan
print("0 - Logout")
print("5 - Menu Admin")
print("=" * 74)

# Bagian Input menu yang diinginkan
print("Masukkan Menu yang Anda Inginkan :")
print("=" * 74)
pilih_menu = input("> ")

# ... (Ada Kode di bagian sini)
```

3.2.2. Fitur menambahkan game baru ke dalam direktori (Create)

Seperti yang sudah dijelaskan sebelumnya, pengguna dapat menambahkan game baru ke dalam direktori. Di bagian ini terdapat sebuah mekanisme yang memastikan agar data

yang dimasukkan oleh pengguna tidak kosong. Berikut merupakan cuplikan singkat dari fitur ini beserta mekanisme pengaman agar data yang dimasukkan tidak kosong.

Source Code:

```
# ... (Ada Kode dibagian sini)

# Bagian Input game beserta beberapa elemen lainnya
game_baru = input(f"{'Masukkan Judul Game':<35}: ")
tahun_rilis = input(f"{'Masukkan Tahun Rilis Game':<35}: ")
developer_game = input(f"{'Masukkan Developer Game':<35}: ")
status_game = input(f"{'Masukkan Status Game':<35}: ")
print("=" * 74)

os.system('clear')
os.system('cls')

if game_baru and tahun_rilis and developer_game and status_game != 0: # Jika
    semua data terisi, program lanjut ke bagian ini
    # Bagian ini menampilkan informasi terkait game baru yang akan dimasukkan ke
    direktori
    print("=" * 74)
    print(f"{'INFORMASI GAME BARU':^74}")
    print("=" * 74)
    print(f"{'Judul Game':<30} {'Tahun Rilis':<15} {'Developer':<20}
{'Status':<10}")
    print(f"{'game_baru':<30} {'tahun_rilis':<15} {'developer_game':<20}
{'status_game':<10}")

    print("=" * 74)
    input("Tekan 'Enter' untuk kembali...")
    print("=" * 74)

    os.system('clear')
    os.system('cls')

    id_game_baru = f"g{len(akun[id_anda]['game']) + 1}" # Membuat ID baru untuk
    game yang baru ditambahkan

    # Menambahkan game ke dalam direktori
    akun[id_anda]['game'][id_game_baru] = {
        "nama": game_baru,
        "tahun": tahun_rilis,
        "dev": developer_game,
        "status": status_game
    }

else: # Bagian ini akan dijalankan jika ada informasi game yang kosong
    print("=" * 74)
```

```

print("Tidak Boleh ada yang kosong!")
print("=" * 74)

time.sleep(1)
os.system('clear')
os.system('cls')

# ... (Ada Kode dibagian sini)

```

3.2.3. Fitur melihat game yang terdapat pada direktori (Read)

Fitur ini cukup sederhana. Pengguna dapat melihat game apa saja yang terdapat di direktori.

Source code:

```

# ... (Ada kode dibagian sini)

# Bagian Display menu
print("=" * 90)
print(f"{'KOLEKSI GAME':^90}")
print("=" * 90)
print(f"{'ID':<5} {'Judul Game':<30} {'Tahun Rilis':<15} {'Developer':<20} {'Status':<10}")
for gid, game in akun[id_anda]['game'].items():
    print(f"{'gid':<5} {'game['nama']':<30} {'game['tahun']':<15} {'game['dev']':<20} {'game['status']':<10}")

print("=" * 90)
input("Tekan 'Enter' untuk kembali ke menu utama...")
print("=" * 90)

# ... (Ada kode dibagian sini)

```

3.2.4. Fitur Mengganti data game yang diinginkan (Update)

Bagian ini memungkinkan pengguna untuk mengganti data game yang ada di direktori jika seandainya pengguna melakukan kesalahan pada saat proses *Create*. Sama seperti proses *Create*, bagian ini dilengkapi dengan mekanisme yang dapat memeriksa apakah pengguna memasukkan data kosong atau tidak. Berikut ini merupakan cuplikan kode yang berisi fitur tersebut.

Source code:

```

# ... (Ada kode dibagian sini)

# Display menu
print("=" * 90)
print(f"{'MENU UPDATE':^90}")
print("=" * 90)
print(f"{'ID':<5} {'Judul Game':<30} {'Tahun Rilis':<15} {'Developer':<20} {'Status':<10}")
for gid, game in akun[id_anda]['game'].items():
    print(f"{'gid':<5} {'game['nama']':<30} {'game['tahun']':<15} {'game['dev']':<20} {'game['status']':<10}")

print("=" * 90)
print(f"{'0':<5} Keluar")
print("=" * 90)

# Input ID game yang ingin diupdate
print("Pilih ID Game yang Ingin Anda Update :")
print("=" * 90)
pilih_update = input("> ")

# ... (Ada kode dibagian sini)

```

3.2.5. Fitur Menghapus Game yang Sudah tidak diinginkan lagi (Delete)

Melalui fitur ini pengguna dapat menghapus game yang tidak diinginkan lagi dari direktori. Sifat dari proses *delete* ini adalah permanen, yang artinya data game akan terhapus selamanya jika pengguna memilih untuk menghapus data game tersebut. Oleh karena itu, program dilengkapi dengan sebuah pertanyaan yang bertujuan untuk memastikan kembali apakah pengguna yakin ingin menghapus game dari direktori.

Source Code:

```

# ... (Ada kode dibagian sini)

# Menampilkan game yang ingin dihapus
print("=" * 90)
print(f"{'GAME YANG INGIN DIHAPUS':^90}")
print("=" * 90)
print(f"{'ID':<5} {'Judul Game':<30} {'Tahun Rilis':<15} {'Developer':<20} {'Status':<10}")
print(f"{'gid':<5} {'game['nama']':<30} {'game['tahun']':<15} {'game['dev']':<20} {'game['status']':<10}")

# Bagian konfirmasi apakah pengguna yakin ingin menghapus game
print("=" * 90)
print("Apakah anda yakin ingin menghapus game ini? (y/n):")

```

```

print("=" * 90)
konfirmasi_hapus = input("> ")
print("=" * 90)

if konfirmasi_hapus == "y": # Bagian ini akan menghapus game yang sudah dipilih
    secara permanen
    del akun[id_anda]['game'][pilih_hapus] # Menghapus game

    # Bagian ini merupakan bagian reordering ID game yang tersisa
    games_lama = akun[id_anda]['game']
    games_baru = {}
    nomor = 1

    # Bagian ini berfungsi untuk membuat ulang ID game setelah ada satu game
    yang dihapus
    for _, data in games_lama.items():
        new_id = f"g{nomor}"
        games_baru[new_id] = data
        nomor += 1

    # Menyatukan perubahan yang ada pada proses sebelumnya
    akun[id_anda]['game'] = games_baru
    selesai_delete = True # Memberikan tanda kalau proses delete sudah selesai

    os.system('clear')
    os.system('cls')
    break
else: # Akan berjalan jika pengguna tidak jadi menghapus game
    print("Tidak jadi hapus!")
    os.system('clear')
    os.system('cls')
    break

# ... (Ada kode dibagian sini)

```

3.3. Fitur yang hanya dimiliki oleh Admin

3.3.1. Menu Khusus Admin

Menu ini hanya dapat diakses jika akun memiliki role sebagai admin. Menu ini menampilkan beberapa fungsi yang dapat digunakan oleh admin untuk mengatur akun yang terdaftar di dictionary.

Source Code:

```

# ... (Ada kode dibagian sini)

```

```

# Display menu
print("=" * 50)
print(f"{'MENU ADMIN':^50}")
print("=" * 50)

print("1 - Tambahkan Akun")
print("2 - Lihat Akun")
print("3 - Update Info Akun")
print("4 - Hapus Akun")
print("=" * 50)

print("0 - Keluar")
print("=" * 50)

print("Pilih Menu yang Anda Inginkan :")
print("=" * 50)
menu_admin = input("> ")

# ... (Ada kode dibagian sini)

```

3.3.2. Fitur menambahkan akun baru (Create)

Admin dapat menambahkan akun baru tanpa harus lewat menu register yang ada di menu utama.

Source Code:

```

# ... (Ada kode dibagian sini)

# Membuat akun baru melalui admin
print("=" * 50)
print(f"{'MEMBUAT AKUN BARU':^50}")
print("=" * 50)

username_baru = input("Masukkan Username : ")
password_baru = input("Masukkan Password : ")

# ... (Ada kode dibagian sini)

```

3.3.3. Fitur melihat akun yang terdaftar di direktori (Read)

Fitur ini cukup sederhana sama dengan fitur Read yang ada pada koleksi game. Disini admin dapat melihat informasi akun yang terdaftar di direktori, mulai dari username, password serta role akun.

Source code:

```

# ... (Ada kode dibagian sini)

# Menampilkan semua informasi aku yang ada pada dictionary
print("=" * 60)
print(f"{'AKUN-AKUN YANG TERDAFTAR':^60}")
print("=" * 60)

print(f"{'ID':<5} {'Nama Akun':<20} {'Password':<20} {'Role':<10}")
for id_acc, info_akun in akun.items():
    print(f"{'str(id_acc)':<5} {'info_akun['username']':<20} {'info_akun['password']':<20} {'info_akun['role']':<10}")

print("=" * 60)
input("Tekan 'Enter' untuk kembali ke Menu Admin... ")
print("=" * 60)

# ... (Ada kode dibagian sini)

```

3.3.4. Fitur Mengganti informasi akun yang diinginkan (Update)

Bagian ini memungkinkan admin untuk mengganti informasi yang terdapat pada akun, mulai dari username, password serta role akun.

Source code:

```

# ... (Ada kode dibagian sini)

# Bagian ini berfungsi untuk mengganti Username, Password, dan juga role dari
# akun-akun yang ada
# Prinsip kerjanya kurang lebih sama dengan proses Update yang ada pada menu
"Update Informasi Game"
print("=" * 70)
print(f"{'UPDATE INFORMASI AKUN':^70}")
print("=" * 70)

print(f"{'ID':<5} {'Nama Akun':<25} {'Password':<25} {'Role':<10}")
for id_acc, info_akun in akun.items():
    print(f"{'str(id_acc)':<5} {'info_akun['username']':<25} {'info_akun['password']':<25} {'info_akun['role']':<10}")

print("=" * 70)
print("0 - Keluar")
print("=" * 70)
print("Masukkan ID Pengguna yang Ingin Anda Ubah:")
admin_update = input("> ")
print("=" * 70)

# ... (Ada kode dibagian sini)

```

3.3.5. Fitur Menghapus akun (Delete)

Melalui fitur ini pengguna dapat menghapus akun yang tidak diinginkan lagi. Sama seperti fitur Delete pada koleksi game, sifat dari proses *delete* ini adalah permanen, yang artinya data game akan terhapus selamanya jika pengguna memilih untuk menghapus data game tersebut. Oleh karena itu, program dilengkapi dengan sebuah pertanyaan yang bertujuan untuk memastikan kembali apakah pengguna yakin ingin menghapus game dari direktori.

Source Code:


```
print("Tidak jadi hapus!")
os.system('clear')
os.system('cls')
break

# ... (Ada kode dibagian sini)
```

4. Hasil Output

```
=====
                        PROGRAM DIREKTORI GAME
=====
1 - Login Akun
2 - Register Akun
=====
0 - Akhiri Program
=====
Silakan pilih menu yang anda inginkan:
> 
```

Gambar 4.1 Tampilan menu awal program

```
=====
                        MENU LOGIN
=====
Masukkan Username : Chryse
Masukkan Password : 79-Au 
```

Gambar 4.2 Tampilan menu login

```
=====
                        KONFIRMASI AKUN BARU
=====
Username           : User3
Password           : 7890
=====
Apakah informasi akun sudah benar? (y/n)
=====
> 
```

Gambar 4.3 Tampilan menu register

```

=====
Selamat datang kembali, Dapupu
Anda login sebagai: member
=====
1 - Tambah Game Baru
2 - Lihat List Game
3 - Update Informasi Game
4 - Hapus Game dari Direktori
=====
0 - Logout
=====
Masukkan Menu yang Anda Inginkan :
=====
> 

```

Gambar 4.4 Tampilan menu bagi admin

```

=====
                        MENAMBAHKAN GAME KE DIREKTORI
=====
Masukkan Judul Game           : tes
Masukkan Tahun Rilis Game     : tes
Masukkan Developer Game       : tes
Masukkan Status Game          : tes 

```

Gambar 4.5 Tampilan menu tambah game (Create) ke direktori

```

=====
                        KOLEKSI GAME
=====
ID      Judul Game           Tahun Rilis   Developer     Status
g1      MudRunner            2017         Saber Interactive  Belum Tamat
g2      tes                  tes          tes            tes
=====
Tekan 'Enter' untuk kembali ke menu utama...

```

Gambar 4.6 Tampilan menu lihat (Read) daftar game

```
=====
                                MENU UPDATE
=====
ID      Judul Game      Tahun Rilis      Developer      Status
g1      MudRunner      2017             Saber Interactive  Belum Tamat
g2      tes             tes              tes              tes
=====
0      Keluar
=====
Pilih ID Game yang Ingin Anda Update :
=====
> 
```

Gambar 4.7 Tampilan menu ubah (Update) game

```
=====
                                MENU HAPUS
=====
ID      Judul Game      Tahun Rilis      Developer      Status
g1      MudRunner      2017             Saber Interactive  Belum Tamat
g2      tes             tes              tes              tes
=====
0      Keluar
=====
Masukkan ID Game yang ingin Anda hapus :
=====
> 
```

Gambar 4.8 Tampilan menu jika pengguna ingin menghapus game

```
=====
Selamat datang kembali, Chryse
Anda login sebagai: admin
=====
1 - Tambah Game Baru
2 - Lihat List Game
3 - Update Informasi Game
4 - Hapus Game dari Direktori
=====
0 - Logout
5 - Menu Admin
=====
Masukkan Menu yang Anda Inginkan :
=====
> 
```

Gambar 4.9 Tampilan menu jika pengguna adalah admin

```

=====
                        MENU ADMIN
=====
1 - Tambahkan Akun
2 - Lihat Akun
3 - Update Info Akun
4 - Hapus Akun
=====
0 - Keluar
=====
Pilih Menu yang Anda Inginkan :
=====
> 

```

Gambar 4.10 Tampilan dari Menu Admin

```

=====
                        KONFIRMASI AKUN BARU
=====
Username           : user5
Password           : 12345
=====
Apakah informasi akun sudah benar? (y/n)
=====
> 

```

Gambar 4.11 Tampilan menu tambah akun (Read) bagi admin

```

=====
                        AKUN-AKUN YANG TERDAFTAR
=====
ID   Nama Akun      Password      Role
acc1 Chryse          79-Au        admin
acc2 Dapupu          Ketua_Nibung  member
acc3 User3         7890         member
acc4 user5         12345        member
=====
Tekan 'Enter' untuk kembali ke Menu Admin... 

```

Gambar 4.12 Tampilan daftar akun (Read)

```
=====
                        UPDATE INFORMASI AKUN
=====
ID      Nama Akun      Password      Role
acc1    Chryse         79-Au        admin
acc2    Dapupu         Ketua_Nibung  member
acc3    User3          7890         member
acc4    user5          12345        member
=====
0 - Keluar
=====
Masukkan ID Pengguna yang Ingin Anda Ubah:
> 
```

Gambar 4.13 Tampilan mengubah informasi (Update) akun

```
=====
                        HAPUS AKUN
=====
ID      Nama Akun      Password      Role
acc1    Chryse         79-Au        admin
acc2    Dapupu         Ketua_Nibung  member
acc3    User3          7890         member
acc4    user5          12345        member
=====
0      Kembali
=====
Masukkan ID akun yang ingin dihapus :
=====
> 
```

Gambar 4.14 Tampilan menghapus (Delete) akun

```
Program Selesai!
Terimakasih telah menggunakan program ini!
```

Gambar 4.15 Tampilan jika program telah diakhiri

5. Langkah-langkah GIT

5.1 GIT Add

```
PS E:\Git\praktikum-apd\post-test\post-test-apd-6> git add 2509106017-Yoga-Ananda-Prasetya-PT-6.py
PS E:\Git\praktikum-apd\post-test\post-test-apd-6> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   2509106017-Yoga-Ananda-Prasetya-PT-6.py
```

Gambar 5.1 Git Add

“Git add” disini berfungsi untuk memasukkan file yang sudah kita kerjakan ke dalam staging area. Staging area disini berfungsi sebagai tempat singgah sementara file-file yang ingin kita commit di git nantinya. “Git status” disini hanya berfungsi untuk memeriksa apakah file yang ingin di-commit sudah ada di staging area atau belum.

5.2 GIT Commit

```
PS E:\Git\praktikum-apd\post-test\post-test-apd-6> git commit -m "Persiapan pengumpulan folder posttest"
[main 3058566] Persiapan pengumpulan folder posttest
1 file changed, 1 insertion(+)
```

Gambar 5.2 Git Commit

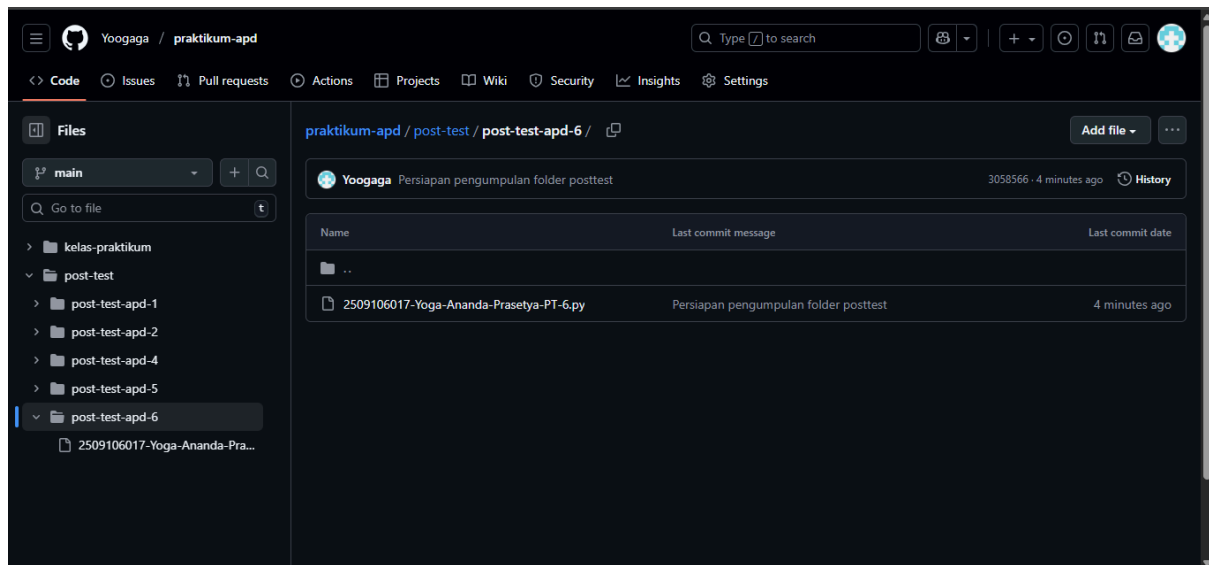
“Git Commit” disini berfungsi untuk menyimpan riwayat dari kode yang telah kita kerjakan ke dalam repository yang ada di dalam komputer kita (repository lokal). Disini kita juga dapat meninggalkan pesan singkat untuk mengingatkan kita atau orang lain terkait apa yang sudah kita kerjakan di kode tersebut.

5.3 GIT Push

```
PS E:\Git\praktikum-apd\post-test\post-test-apd-6> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 497 bytes | 2.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote: This repository moved. Please use the new location:
remote:   https://github.com/Yoogaga/praktikum-apd.git
To https://github.com/Yoogaga/belajargit.git
   5716344..3058566  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 Git Push

“Git Push” disini berfungsi untuk mengunggah file-file yang sudah di-commit di repository lokal menuju ke repository yang ada di Github. Tujuannya adalah agar file-file yang sudah dikerjakan dapat diakses secara online dan dapat diakses oleh siapapun selama pengaturan repository-nya tidak di-private.



Gambar 5.4 Tampilan pada Github