

LAPORAN PRAKTIKUM
POSTTEST 7
ALGORITMA PEMROGRAMAN DASAR

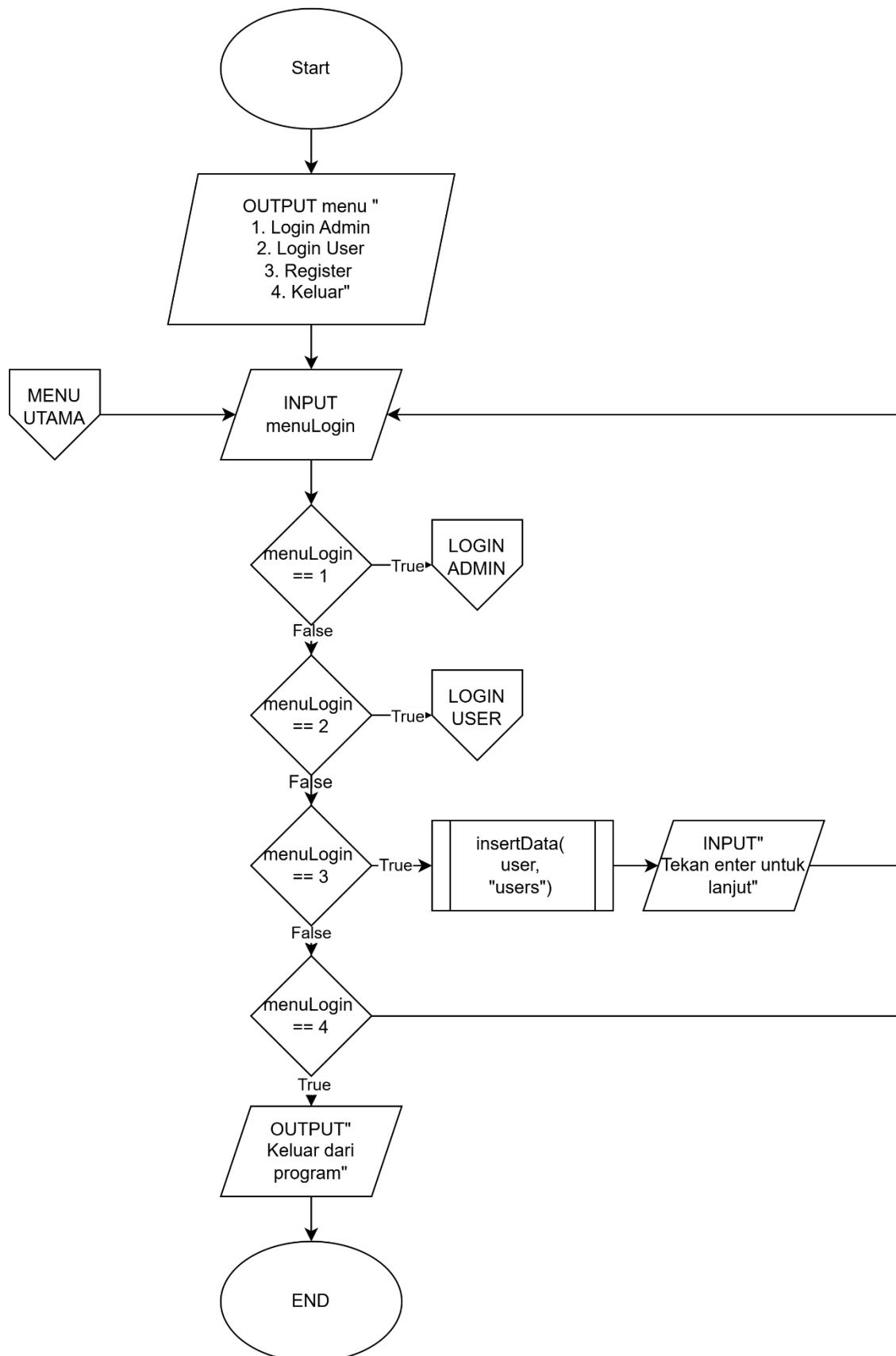


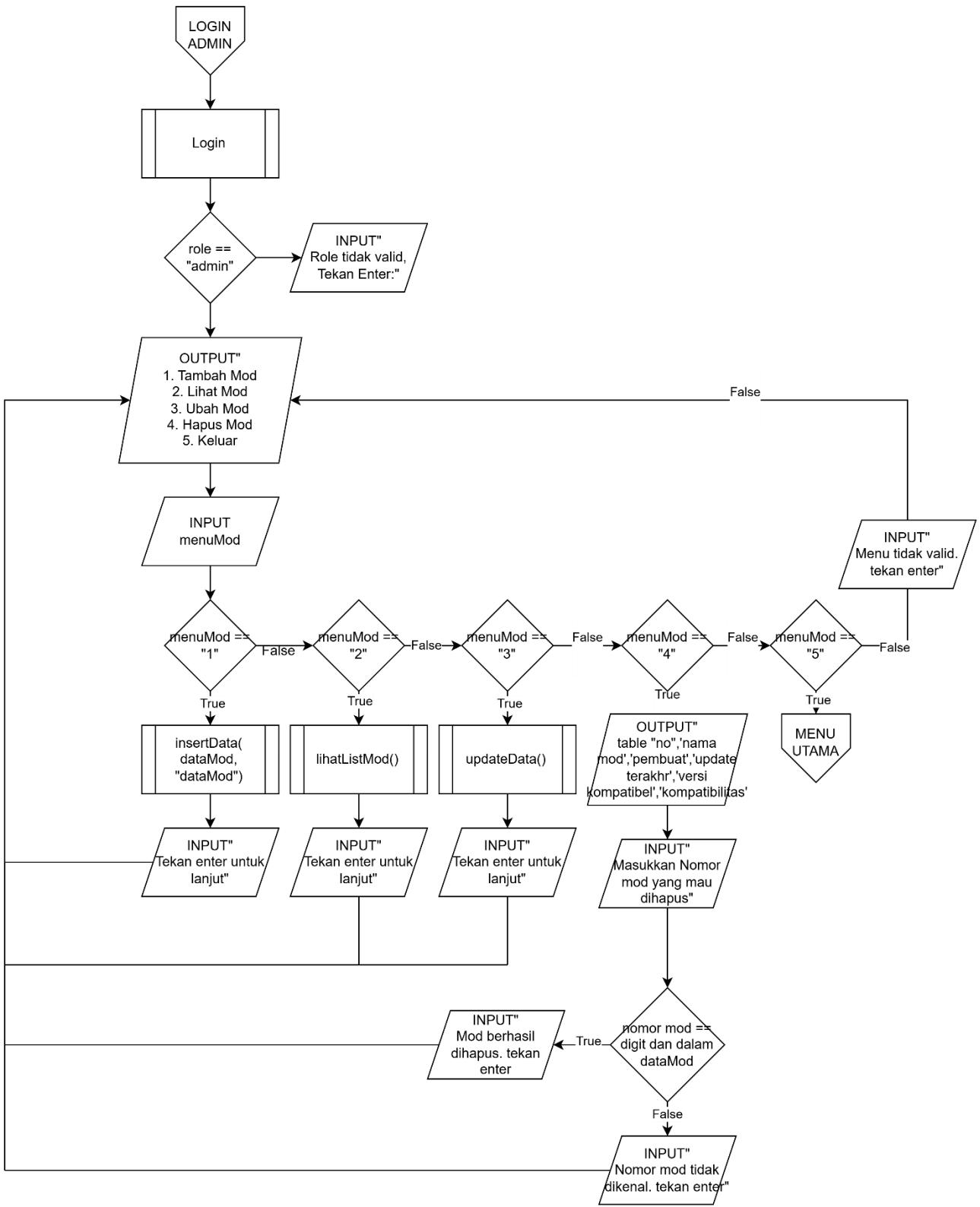
Disusun oleh:
Arif Abdurrahman Siddiq (2509106064)
Informatika (B1 '25)

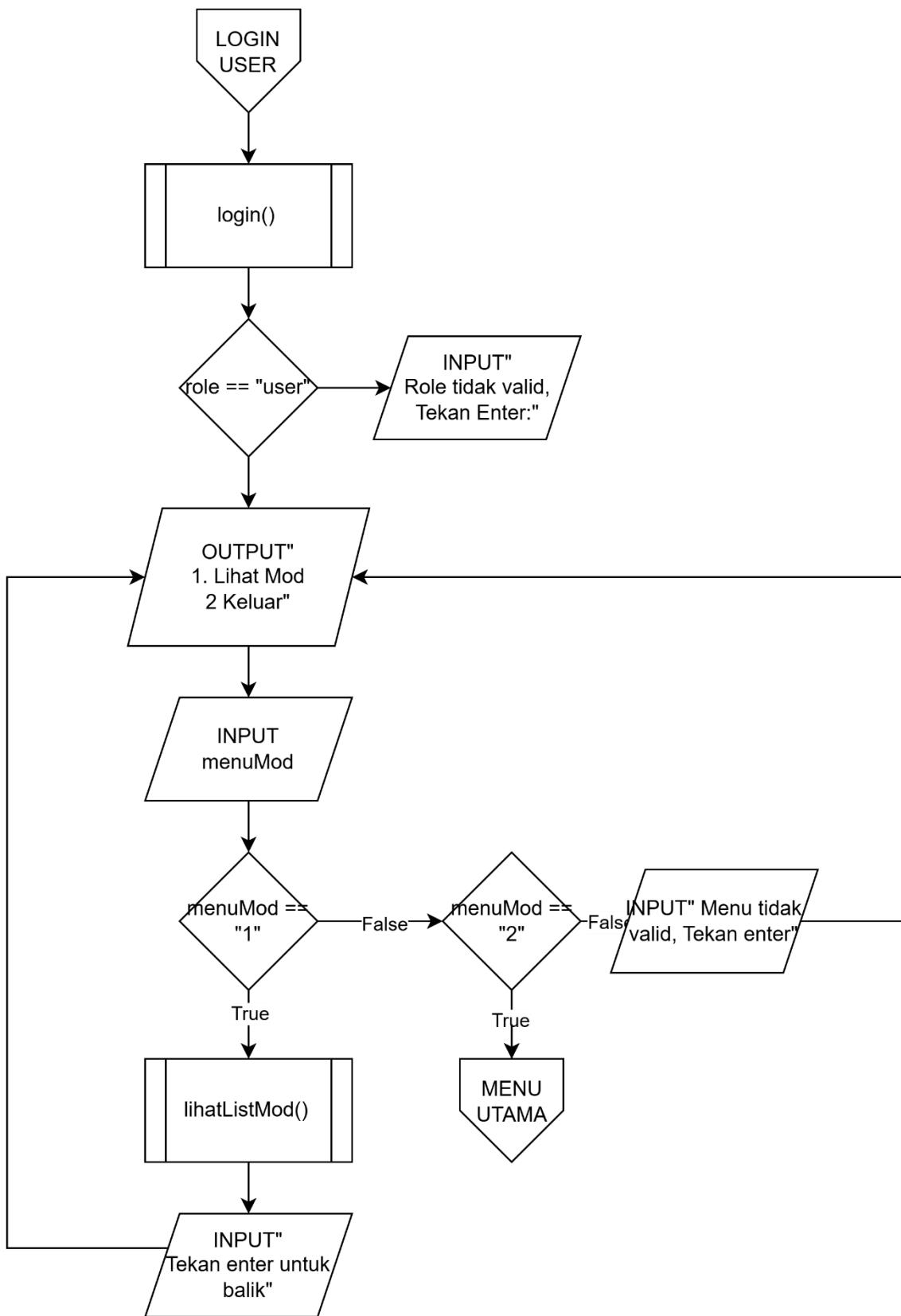
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

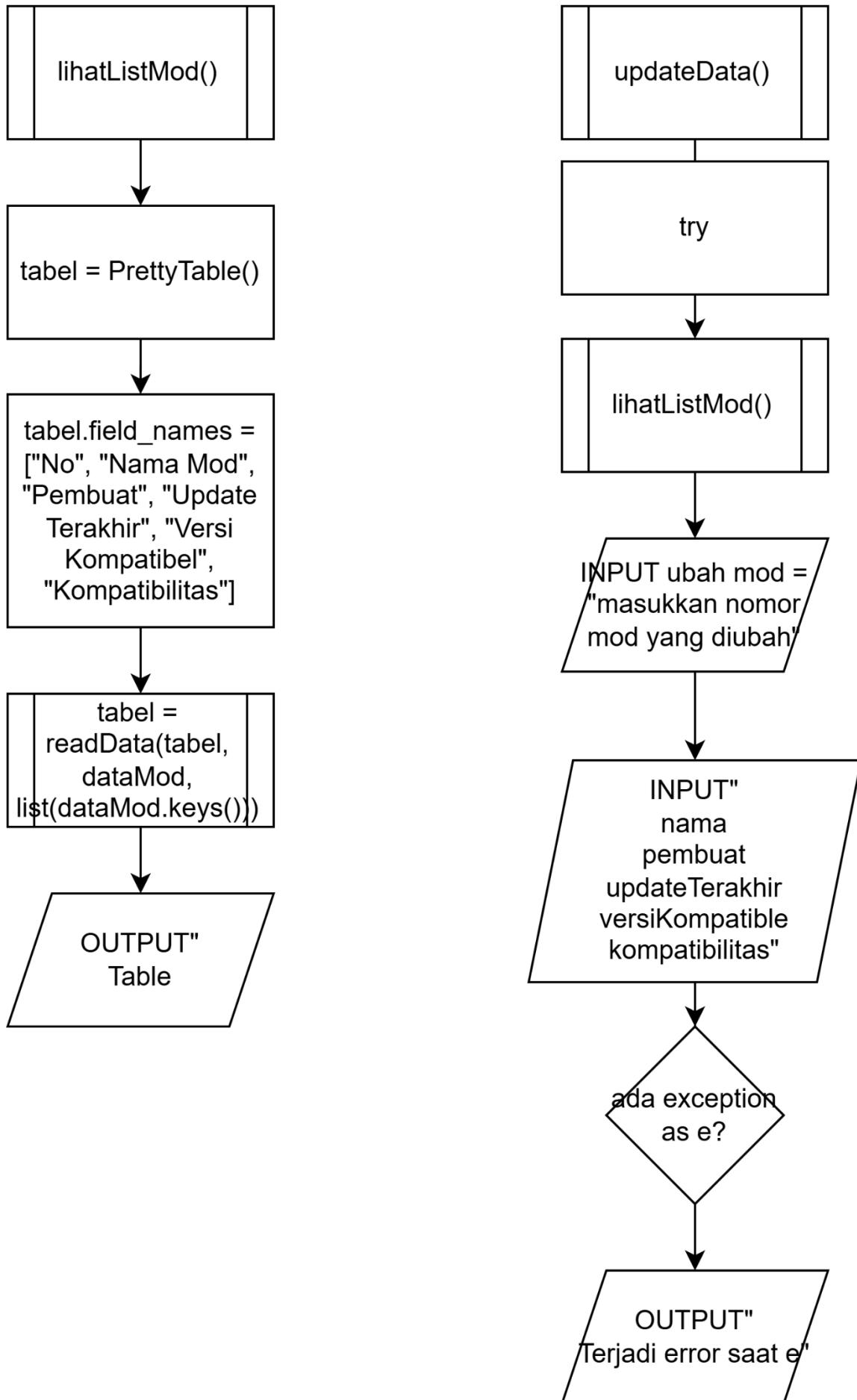
2025

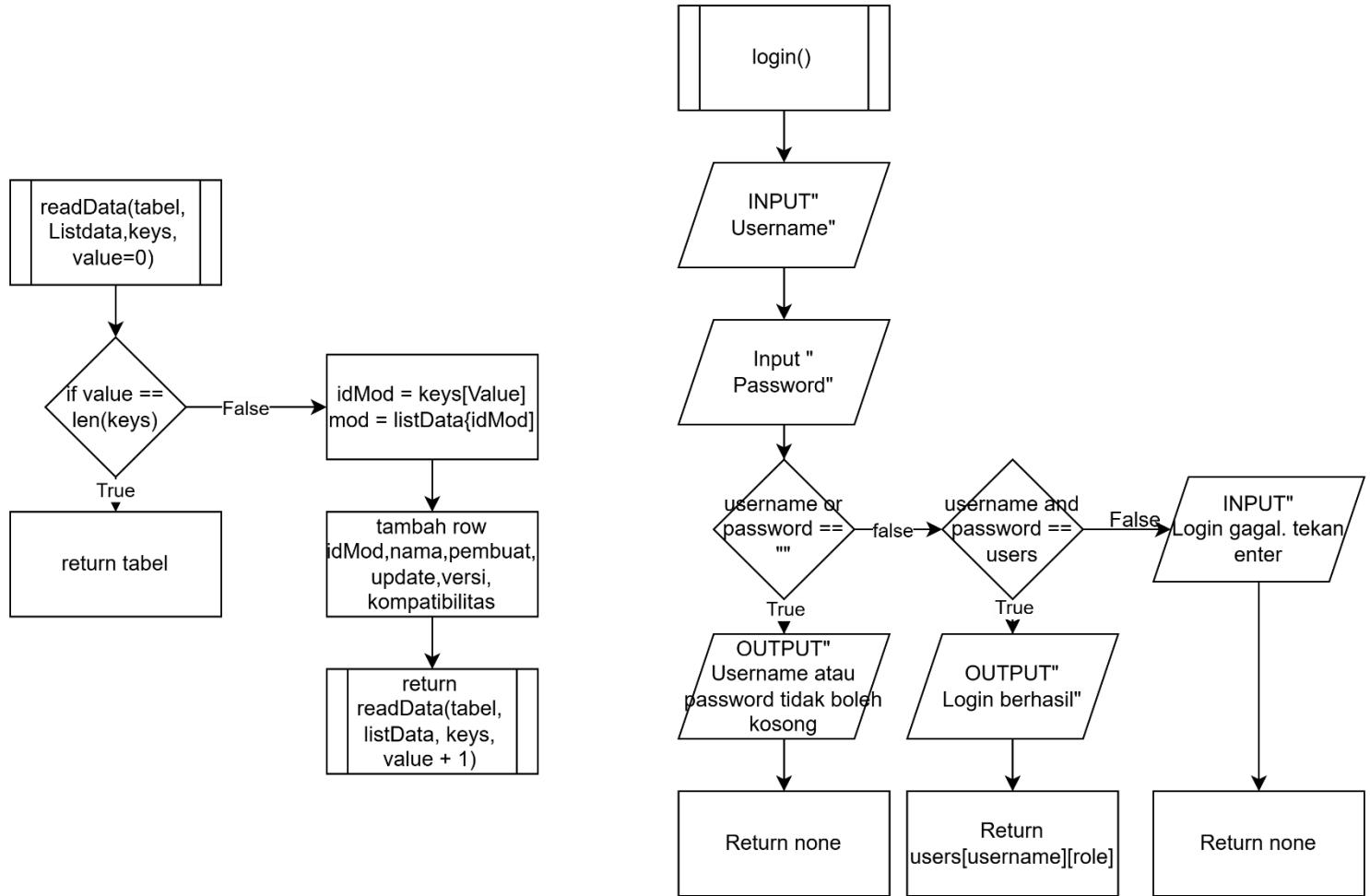
1. Flowchart

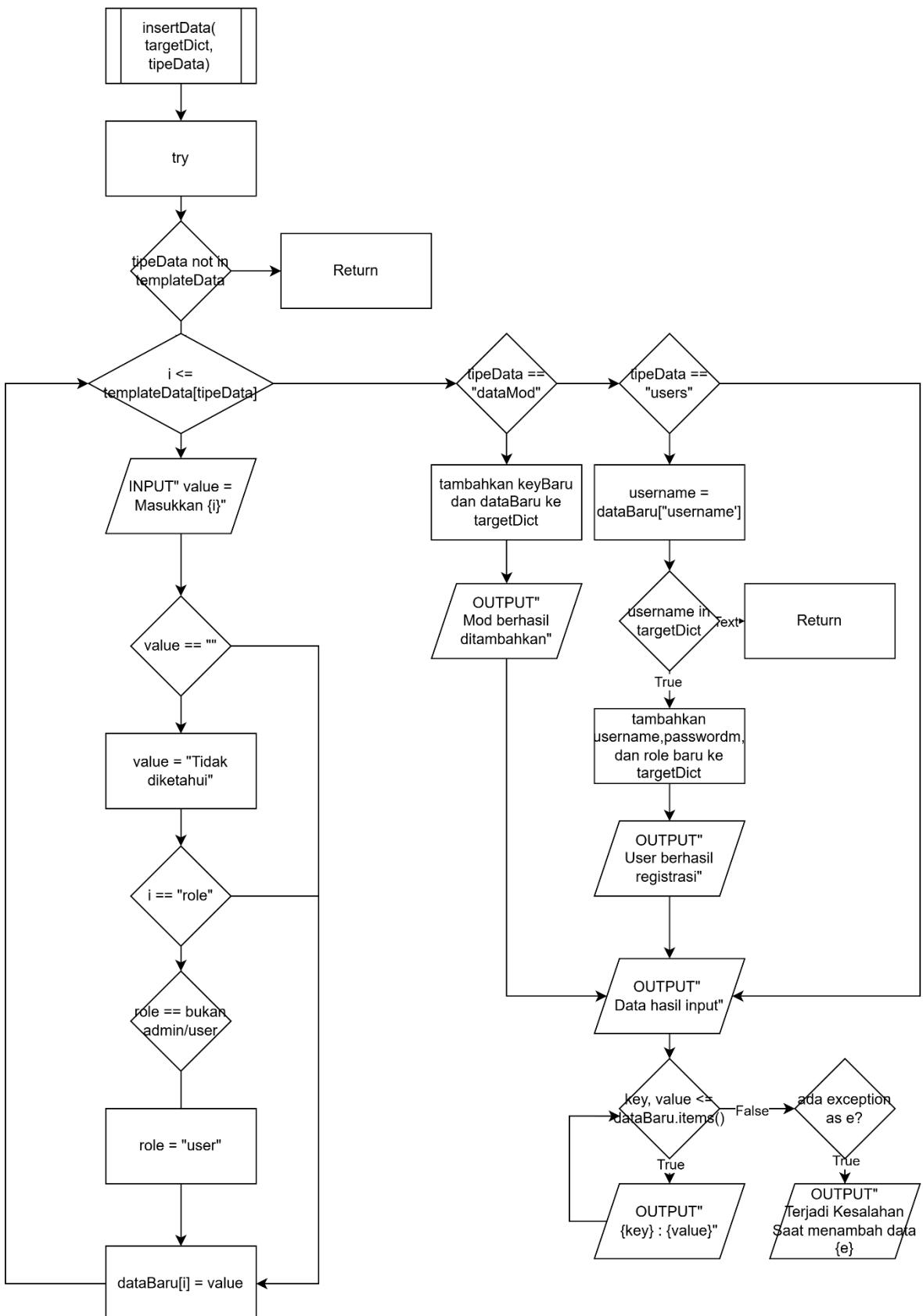












2. Deskripsi Singkat Program

Program dapat berfungsi sebagai catatan mod yang di download user buat sebuah game bernama Victoria 3. Victoria sendiri adalah sebuah game sandbox yang termasuk dalam kategori “Grand Strategy” karena gameplay yang berfokus pada pembangunan negara – negara yang berada di pertengahan Masa Victoria. Oleh karena itu fitur sandbox-nya memungkinkan developer untuk membuat banyak mod.

User dapat menambahkan (CREATE) sebuah mod ke dalam programnya dengan mengisi parameter tertentu seperti Nama Mod, Pembuat, Tanggal Update, Versi Kompatibel, dan Kompatibilitas melalui function insertData(). User juga dapat melihat (READ) mod yang mereka tambahkan ke dalam tabel melalui function lihatListData(). Selain itu User bisa mengubah (UPDATE) paremeter mod-nya masing-masing melalui function ubahData() dan juga menghapus (DELETE) mod yang telah di uninstall. Semua data yang di manipulasi akan disimpan dalam dictionary

3. Source Code

A. Variable Global

```
users = {
    "admin": {"password": "admin123", "role": "admin"},
    "arif": {"password": "064", "role": "user"}
}

dataMod = {
    1: {
        "nama": "Realistic Economy",
        "pembuat": "Victor Hugo",
        "update": "2024-08-12",
        "versi": "1.5.3",
        "kompatibilitas": "Kompatibel"
    },
    2: {
        "nama": "World Overhaul",
        "pembuat": "Marie Curie",
        "update": "2024-06-25",
        "versi": "1.5.2",
        "kompatibilitas": "Tidak Kompatibel"
    },
    3: {
        "nama": "Population Rework",
    }
}
```

```

        "pembuat": "Arthur Conan",
        "update": "2024-09-14",
        "versi": "1.5.3",
        "kompatibilitas": "Kompatibel"
    }
}

templateData = {
    "users": ["username", "password", "role"],
    "dataMod": ["nama", "pembuat", "update", "versi", "kompatibilitas"]
}

```

Fitur ini buat membuat variable global

B. Function Login

```

def login():
    username = input("\nUsername: ")
    password = input("Password: ")

    if username == "" or password == "":
        input("\nUsername atau Password tidak boleh kosong. Tekan Enter...")
        return None

    if username in users and users[username]["password"] == password:
        print("\nLogin berhasil!")
        return users[username]["role"]
    else:
        input("\nLogin gagal. Tekan enter...")
        return None

```

Fitur ini buat mendefinisikan function login

C. Function InsertData

```

def insertData(targetDict, tipeData):
    try:
        if tipeData not in templateData:
            print("Tipe Data tidak dikenal.")
            return

        dataBaru = []

```

```

print(f"\n== Tambah {tipeData.upper()} ==")

for i in templateData[tipeData]:
    value = input(f"Masukkan {i}").strip()

    if value == "":
        print(f"{i} tidak boleh kosong, diisi 'Tidak Diketahui'.")
        value = "Tidak Diketahui"

    elif i == "role":
        if value.lower() not in ["admin", "user"]:
            print("Role tidak valid, default: user")
            value = "user"

    dataBaru[i] = value

if tipeData == "dataMod":
    keyBaru = max(targetDict.keys(), default=0) + 1
    targetDict[keyBaru] = dataBaru
    print("\nMod berhasil ditambahkan")

elif tipeData == "users":
    username = dataBaru["username"]
    if username in targetDict:
        print("\nUsername sudah terdaftar.")
        return
    targetDict[username] = {
        "password": dataBaru["password"],
        "role": dataBaru["role"]
    }
    print("\nUser berhasil di registrasi")

    print("\nData Hasil Input:")
    for key, value in dataBaru.items():
        print(f"{key}: {value}")

except Exception as e:
    print(f"\nTerjadi kesalahan saat menambah data: {e}")

```

Fitur ini mendefinisikan function insertData

D. Function readData dan lihatListMod

```

#Function baca data 1 per satu rekursif
def readData(tabel, listData, keys, value=0):
    if value == len(keys):

```

```

        return tabel

    idMod = keys[value]
    mod = listData[idMod]

    tabel.add_row([idMod, mod["nama"], mod["pembuat"], mod["update"],
    mod["versi"], mod["kompatibilitas"]])

    return readData(tabel, listData, keys, value + 1)
#Function lihat data dari function rekursif readData prosedur
def lihatListMod():
    os.system('cls')
    print("== DAFTAR MOD ==")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Mod", "Pembuat", "Update
Terakhir", "Versi Kompatibel", "Kompatibilitas"]

    tabel = readData(tabel, dataMod, list(dataMod.keys()))
    print(tabel)

```

Fitur ini buat mendefinisikan function rekursif readData dan function lihatListMod

E. Function updateData

```

def updateData():
    try:
        os.system('cls')
        print("== UBAH MOD ==")
        lihatListMod()
        ubahMod = input("Masukkan nomor mod yang ingin diubah: ")
        if not ubahMod.isdigit() or int(ubahMod) not in dataMod:
            print("Nomor mod tidak valid.")
            return

        idMod = int(ubahMod)
        nama = input("Nama baru (kosongkan jika tidak diganti): ")
        pembuat = input("Pembuat baru (kosongkan jika tidak diganti):
")
        updateTerakhir = input("Update baru (YYYY-MM-DD): ")
        versiKompatibel = input("Versi baru (kosongkan jika tidak
diganti): ")
        kompatibilitas = input("Kompatibilitas (Y/T): ").upper()

        if nama:
            dataMod[idMod]["nama"] = nama
        if pembuat:
            dataMod[idMod]["pembuat"] = pembuat

```

```

if updateTerakhir:
    dataMod[idMod]["update"] = updateTerakhir
if versiKompatibel:
    dataMod[idMod]["versi"] = versiKompatibel
if kompatibilitas == "Y":
    dataMod[idMod]["kompatibilitas"] = "Kompatibel"
elif kompatibilitas == "T":
    dataMod[idMod]["kompatibilitas"] = "Tidak Kompatibel"
print("Mod berhasil diubah!")

except Exception as e:
    print(f"Terjadi error saat mengubah mod: {e}")

```

Fitur ini buat mendefinisikan function updateData

F. Loop Admin

```

while True:
    os.system('cls')
    print(f"== MOD MENU ({role.upper()}) ==")

    if role == "admin":
        print("1. Tambah Mod")
        print("2. Lihat List Mod")
        print("3. Ubah Mod")
        print("4. Hapus Mod")
        print("5. Keluar")
        menuMod = input("Pilih menu: ")

        if menuMod == "1":
            insertData(dataMod, "dataMod")
            input("\nTekan Enter untuk lanjut...")

        elif menuMod == "2":
            lihatListMod()
            input("\nTekan Enter untuk kembali ke menu...")

        elif menuMod == "3":
            updateData()
            input("\nMod berhasil diubah! Tekan Enter...")

        elif menuMod == "4":
            os.system('cls')
            print("== HAPUS MOD ==")
            tabel = PrettyTable()
            tabel.field_names = ["No", "Nama Mod", "Pembuat",
"Update Terakhir", "Versi Kompatibel", "Kompatibilitas"]
            for idMod, mod in dataMod.items():

```

```

        tabel.add_row([idMod, mod["nama"],
mod["pembuat"], mod["update"], mod["versi"], mod["kompatibilitas"]])
        print(tabel)

hapusMod = input("Masukkan nomor mod yang ingin
dihapus: ")
if hapusMod.isdigit() and int(hapusMod) in
dataMod:
    del dataMod[int(hapusMod)]
    input("Mod berhasil dihapus! Tekan Enter...")
else:
    input("Nomor mod tidak ditemukan. Tekan
Enter...")

elif menuMod == "5":
    break
else:
    input("Menu tidak valid. Tekan Enter...")

else:
    input("Role tidak valid. Tekan Enter...")
    break

```

Fitur ini buat akses admin untuk menambah,melihat,mengubah. Dan menghapus data

G. Loop User

```

while True:
    if role == "user":
        print("1. Lihat List Mod")
        print("2. Keluar")
        menuUser = input("Pilih menu: ")

        if menuUser == "1":
            lihatListMod()
            input("\nTekan Enter untuk kembali ke
menu...")

        elif menuUser == "2":
            break
        else:
            input("Menu tidak valid. Tekan Enter...")
    else:
        input("Role tidak valid. Tekan Enter...")
        break

```

Fitur ini buat user untuk melihat data

H. Register

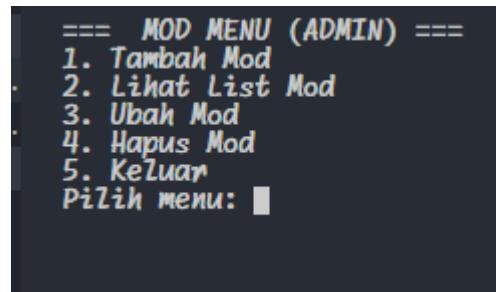
```
elif menuLogin == "3":  
    insertData(users, "users")  
    input("\nTekan Enter untuk lanjut...")
```

Fitur ini buat register

4. Hasil Output

```
== List Mod Victoria 3 ==  
1. Login Admin  
2. Login User  
3. Register  
4. Keluar  
Pilih menu: 1  
  
Username: admin  
Password: admin123
```

Gambar 4.1 Output Login Admin



Gambar 4.2 Menu Admin

```
==== MOD MENU (ADMIN) ====
1. Tambah Mod
2. Lihat List Mod
3. Ubah Mod
4. Hapus Mod
5. Keluar
Pilih menu: 1

==== Tambah DATAMOD ====
Masukkan nama: Tes
Masukkan pembuat: Tes
Masukkan update: TeS
Masukkan versi: Tes
Masukkan kompatibilitas: Tes

Mod berhasil ditambahkan

Data Hasil Input:
nama: Tes
pembuat: Tes
update: TeS
versi: Tes
kompatibilitas: Tes

Tekan Enter untuk Lanjut...■
```

Gambar 4.3 Output 1. Tambah Mod

==== DAFTAR MOD ===					
No	Nama Mod	Pembuat	Update Terakhir	Versi Kompatibel	Kompatibilitas
1	Realistic Economy	Victor Hugo	2024-08-12	1.5.3	Kompatibel
2	World Overhaul	Marie Curie	2024-06-25	1.5.2	Tidak Kompatibel
3	Population Rework	Arthur Conan	2024-09-14	1.5.3	Kompatibel
4	Tes	Tes	TeS	Tes	Tes

Tekan Enter untuk kembali ke menu...■

Gambar 4.4 Output 2. Lihat List Mod

```

== DAFTAR MOD ==
+-----+
| No | Nama Mod      | Pembuat    | Update Terakhir | Versi Kompatibel | Kompatibilitas |
+-----+
| 1  | Realistic Economy | Victor Hugo | 2024-08-12       | 1.5.3           | Kompatibel      |
| 2  | World Overhaul   | Marie Curie  | 2024-06-25       | 1.5.2           | Tidak Kompatibel |
| 3  | Population Rework | Arthur Conan  | 2024-09-14       | 1.5.3           | Kompatibel      |
| 4  | Tes               | Tes          | Tes              | Tes             | Tes             |
+-----+
Masukkan nomor mod yang ingin diubah: 4
Nama baru (kosongkan jika tidak diganti): tes
Pembuat baru (kosongkan jika tidak diganti): tes
Update baru (YYYY-MM-DD): tes
Versi baru (kosongkan jika tidak diganti): tes
Kompatibilitas (Y/T): t
Mod berhasil diubah!
Mod berhasil diubah! Tekan Enter...■

```

Gambar 4.5 Output 3. Update Mod

```

== HAPUS MOD ==
+-----+
| No | Nama Mod      | Pembuat    | Update Terakhir | Versi Kompatibel | Kompatibilitas |
+-----+
| 1  | Realistic Economy | Victor Hugo | 2024-08-12       | 1.5.3           | Kompatibel      |
| 2  | World Overhaul   | Marie Curie  | 2024-06-25       | 1.5.2           | Tidak Kompatibel |
| 3  | Population Rework | Arthur Conan  | 2024-09-14       | 1.5.3           | Kompatibel      |
| 4  | tes               | tes          | tes              | tes             | Tidak Kompatibel |
+-----+
Masukkan nomor mod yang ingin dihapus: 4
Mod berhasil dihapus! Tekan Enter...■

```

Gambar 4.6 Output 4. Hapus Mod

```

== List Mod Victoria 3 ==
1. Login Admin
2. Login User
3. Register
4. Keluar
Pilih menu: 2

Username: arif
Password: 064■

```

Gambar 4.7 Output . Login User

```

1. Lihat List Mod
2. Keluar
Pilih menu: ■

```

Gambar 4.8 Output . Menu User

```

    === List Mod Victoria 3 ===
    1. Login Admin
    2. Login User
    3. Register
    4. Keluar
    Pilih menu: 3

    === Tambah USERS ===
    Masukkan username: tes
    Masukkan password: tes
    Masukkan role: user

    User berhasil di registrasi

    Data Hasil Input:
    username: tes
    password: tes
    role: user

    Tekan Enter untuk Lanjut...■

```

Gambar 4.9 Output . Register

5. Langkah-langkah GIT

5.1 GIT Add

```
PS E:\Python\praktikum-apd\post-test> git add .
```

Menyimpan semua perubahan yang terjadi di folder

5.2 GIT Commit

```
PS E:\Python\praktikum-apd\post-test> git commit -m "PT 7"
[main 21390c9] PT 7
 3 files changed, 314 insertions(+), 44 deletions(-)
 create mode 100644 post-test/post-test-apd-7/2509106064-ARIF ABDURRAHMAN SIDDIQ-PT-7.pdf
 create mode 100644 post-test/post-test-apd-7/2509106064-ARIF ABDURRAHMAN SIDDIQ-PT-7.py

```

Menambah perubahan ke repo lokal dengan pesan “Post 5 kumpul” ke branch m

5.3 GIT Push

```
PS E:\Python\praktikum-apd\post-test> git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.89 MiB | 879.00 KiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/TromBoLoN/praktikum-apd
 cb1f2a7..21390c9 main -> main
PS E:\Python\praktikum-apd\post-test> ■
```

Mendorong semua perubahan yang disimpan repo lokal ke repo online