

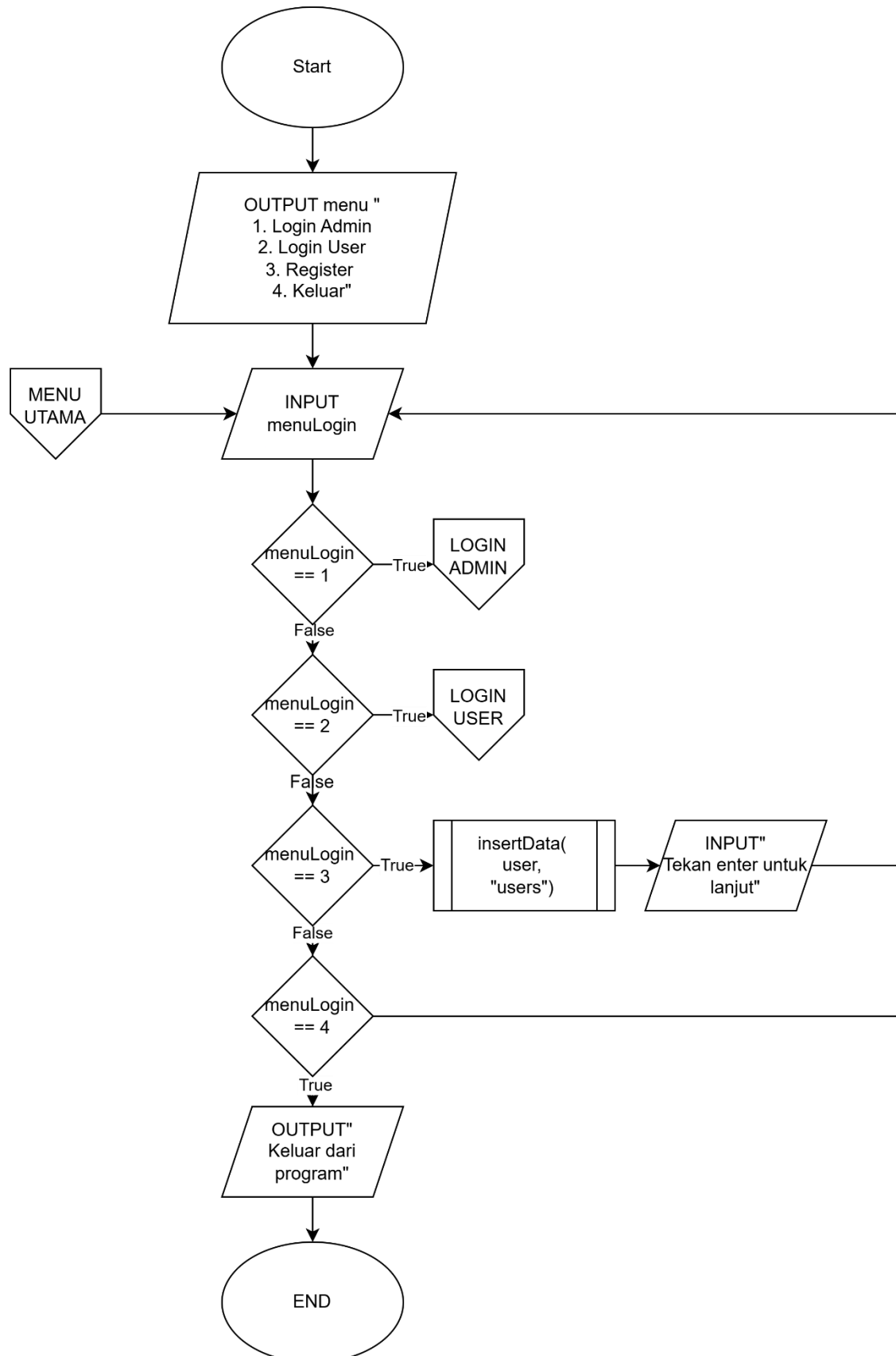
LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR

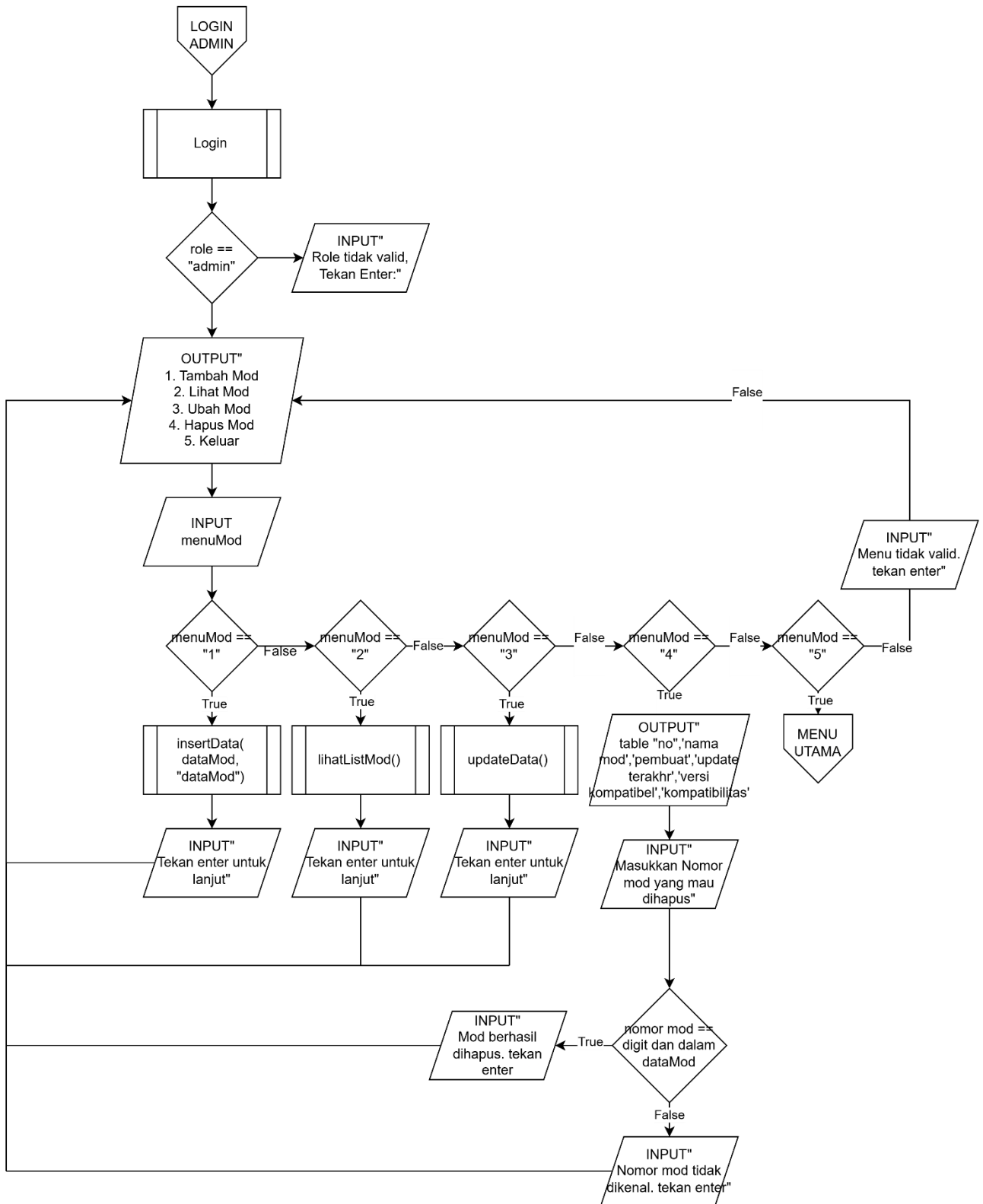


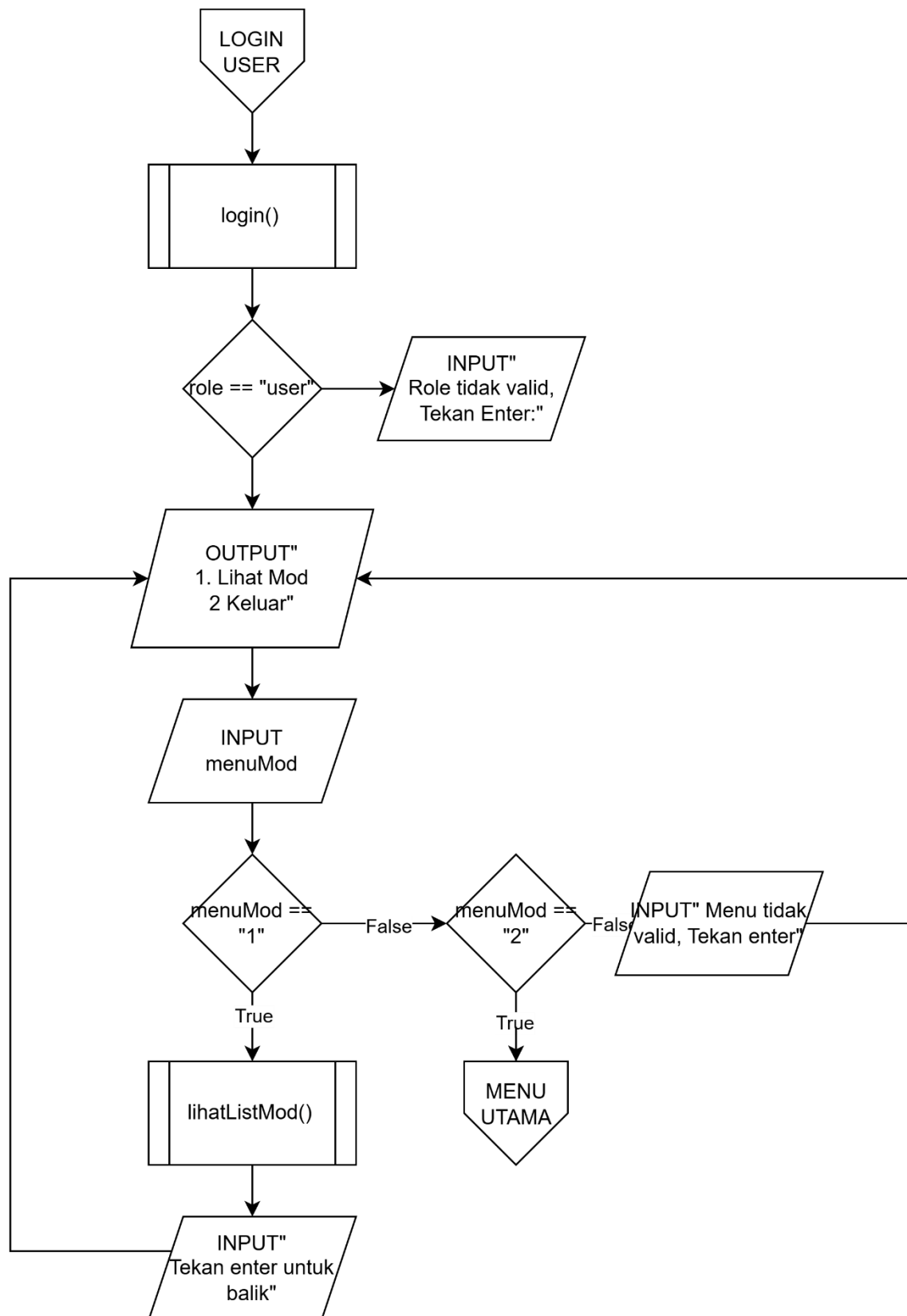
Disusun oleh:
Arif Abdurrahman Siddiq (2509106064)
Informatika (B1 '25)

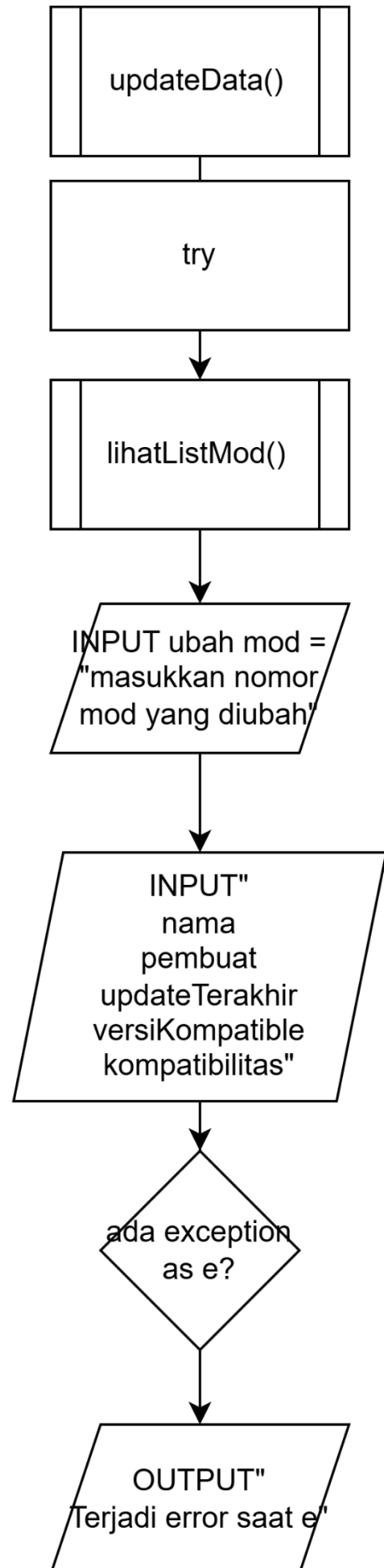
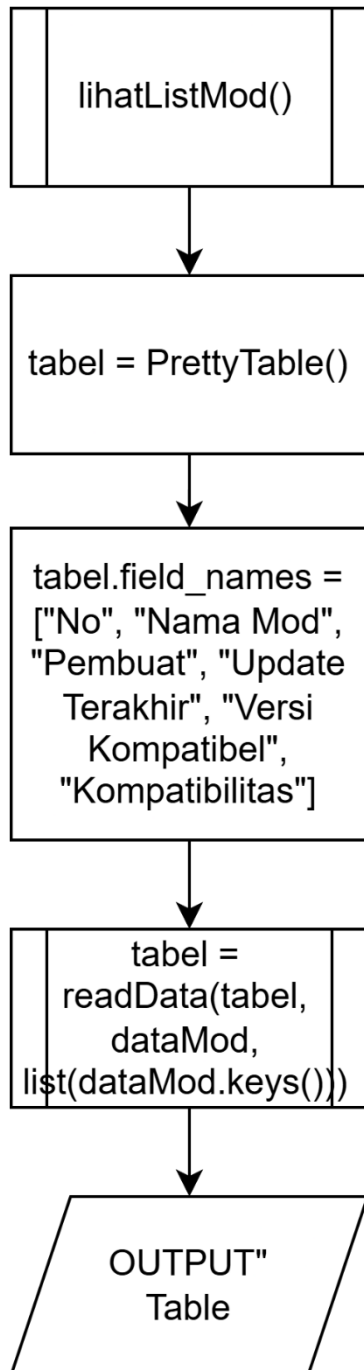
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

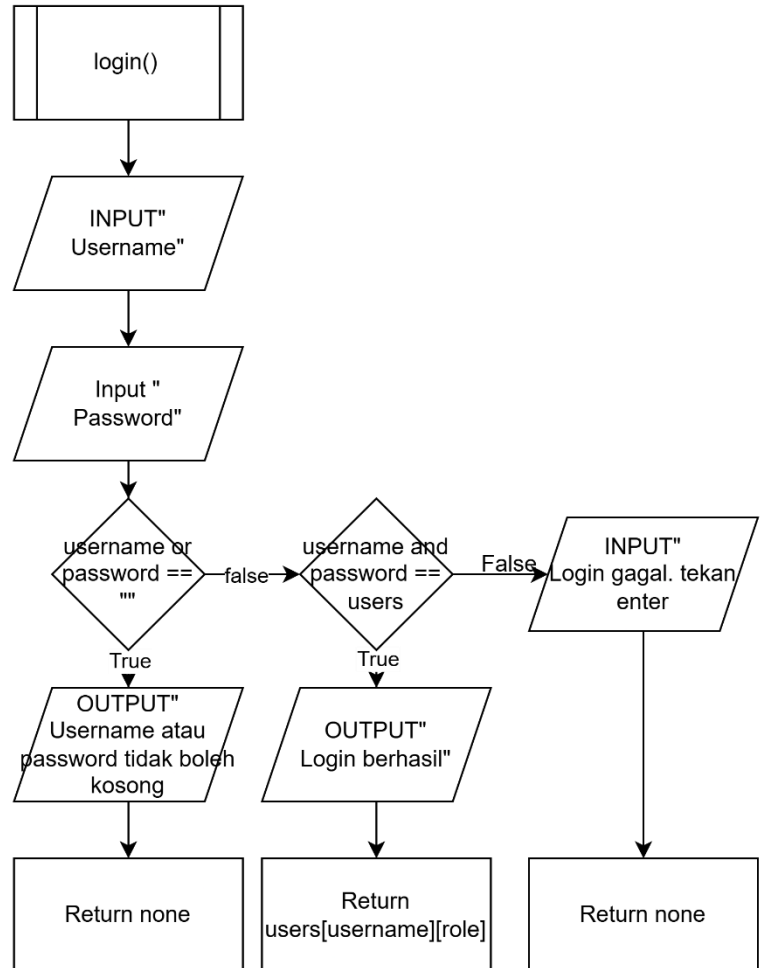
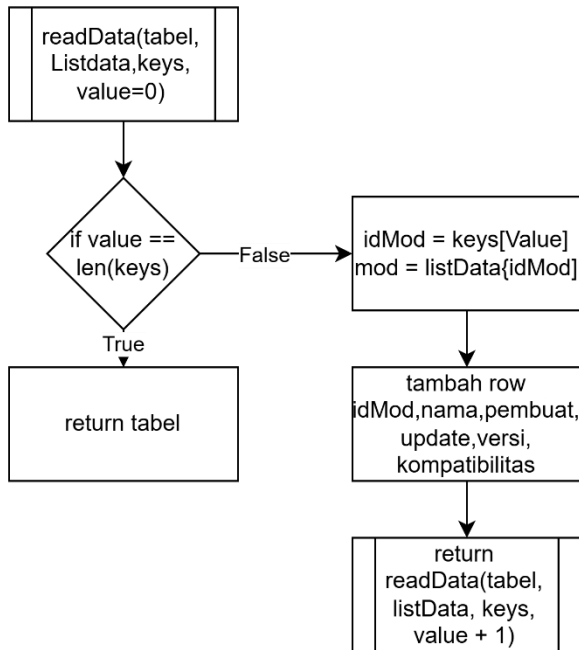
1. Flowchart

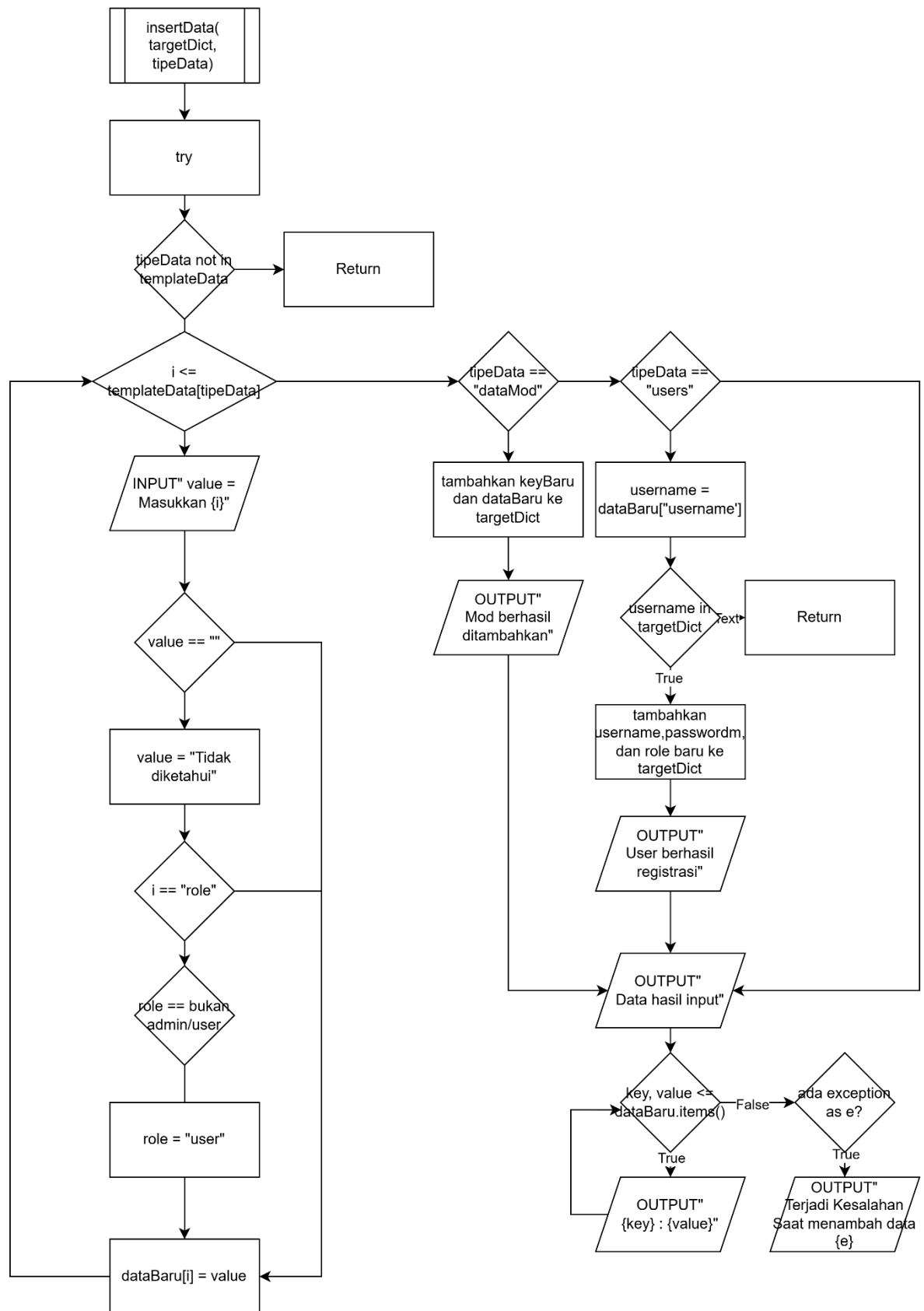












2. Deskripsi Singkat Program

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.

Programnya dipecah jadi 4 filenya. File data.py untuk menampung data di dictionarynya, file auth.py membuat fungsi login dan register sekaligus mengambil data dari data.py, file crud_mod.py untuk menyimpan fungsi CRUD modnya, dan file main.py buat tampilan modnya

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) parameter 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. data.py

```
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 menunjukkan isi dari file data.py untuk menampung data

B. auth.py

```

from data import users, templateData

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

    if not username or not 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"]

    input("\nLogin gagal. Tekan Enter...")
    return None

def register():
    dataBaru = {}

```

```

print("\n=== Registrasi User ===")

for i in templateData["users"]:
    value = input(f"Masukkan {i}: ").strip()
    if not value:
        print(f"{i} tidak boleh kosong, diisi 'Tidak Diketahui'.")
        value = "Tidak Diketahui"

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

    dataBaru[i] = value

username = dataBaru["username"]
if username in users:
    print("\nUsername sudah terdaftar.")
    return

users[username] = {
    "password": dataBaru["password"],
    "role": dataBaru["role"]
}
print("\nUser berhasil diregistrasi.")

```

Fitur ini buat menunjukan isi dari file auth.py yang dipakai untuk mendefinisikan function login dan register

C. Function InsertData

```

import os
from prettytable import PrettyTable
from data import dataMod, templateData
from datetime import date

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

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)

def lihatListMod():
    clear()
    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)

def insertData():
    try:
        dataBaru = {}
        print("\n=== Tambah MOD ===")

        for i in templateData["dataMod"]:
            value = input(f"Masukkan {i}: ").strip()
            if not value:
                print(f"{i} tidak boleh kosong, diisi 'Tidak
Diketahui'.")
                value = "Tidak Diketahui"
            dataBaru[i] = value

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

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

from datetime import date

def updateData():
    try:
        clear()
        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): ")
    ")
    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 versiKompatibel:
        dataMod[idMod]["versi"] = versiKompatibel
    if kompatibilitas == "Y":
        dataMod[idMod]["kompatibilitas"] = "Kompatibel"
    elif kompatibilitas == "T":
        dataMod[idMod]["kompatibilitas"] = "Tidak Kompatibel"

    dataMod[idMod]["update"] = str(date.today())

    print(f"Mod berhasil diubah! (Update otomatis:
{dataMod[idMod]['update']})")

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

def deleteData():
    clear()
    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...")

```

Fitur ini buat menunjukan isi dari file crud_mod yang dimana digunakan untuk mendefinisikan function insertData untuk menambahkan data, updateData untuk memperbarui data, readData untuk membaca list mod satu persatu, lihatListMod untuk memasukkannya dalam prettyTable, dan delete data untuk menghapus data

D. menu.py

```
import os
from auth import login, register
from crud_mod import LihatListMod, insertData, updateData,
deleteData, clear

while True:
    clear()
    print("=== List Mod Victoria 3 ===")
    print("1. Login Admin")
    print("2. Login User")
    print("3. Register")
    print("4. Keluar")
    menuLogin = input("Pilih menu: ")

    if menuLogin == "1":
        role = login()
        if not role:
            continue

        if role != "admin":
            input("Anda bukan admin. Tekan Enter...")
            continue

        while True:
            clear()
            print(f"=== MOD MENU ({role.upper()}) ===")
            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()
                input("\nTekan Enter untuk lanjut...")

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

```

        elif menuMod == "3":
            updateData()
            input("\nTekan Enter untuk Lanjut...")

        elif menuMod == "4":
            deleteData()

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

    elif menuLogin == "2":
        role = login()
        if not role or role != "user":
            input("Role tidak valid. Tekan Enter...")
            continue

    while True:
        clear()
        print("=== USER MENU ===")
        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...")

    elif menuLogin == "3":
        register()
        input("\nTekan Enter untuk Lanjut...")

    elif menuLogin == "4":
        clear()
        print("Keluar dari program...")
        break

    else:
        input("Menu tidak valid. Tekan Enter...")

```

Fitur ini buat menunjukkan menu.py yang berisi tampilan dari program di termin

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

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

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

Masukkan nomor mod yang ingin diubah: 3
 Nama baru (kosongkan jika tidak diganti): a
 Pembuat baru (kosongkan jika tidak diganti): a
 Versi baru (kosongkan jika tidak diganti): a
 Kompatibilitas (Y/T): a
 Mod berhasil diubah! (Update otomatis: 2025-11-05)

Tekan Enter untuk lanjut...

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