

LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR

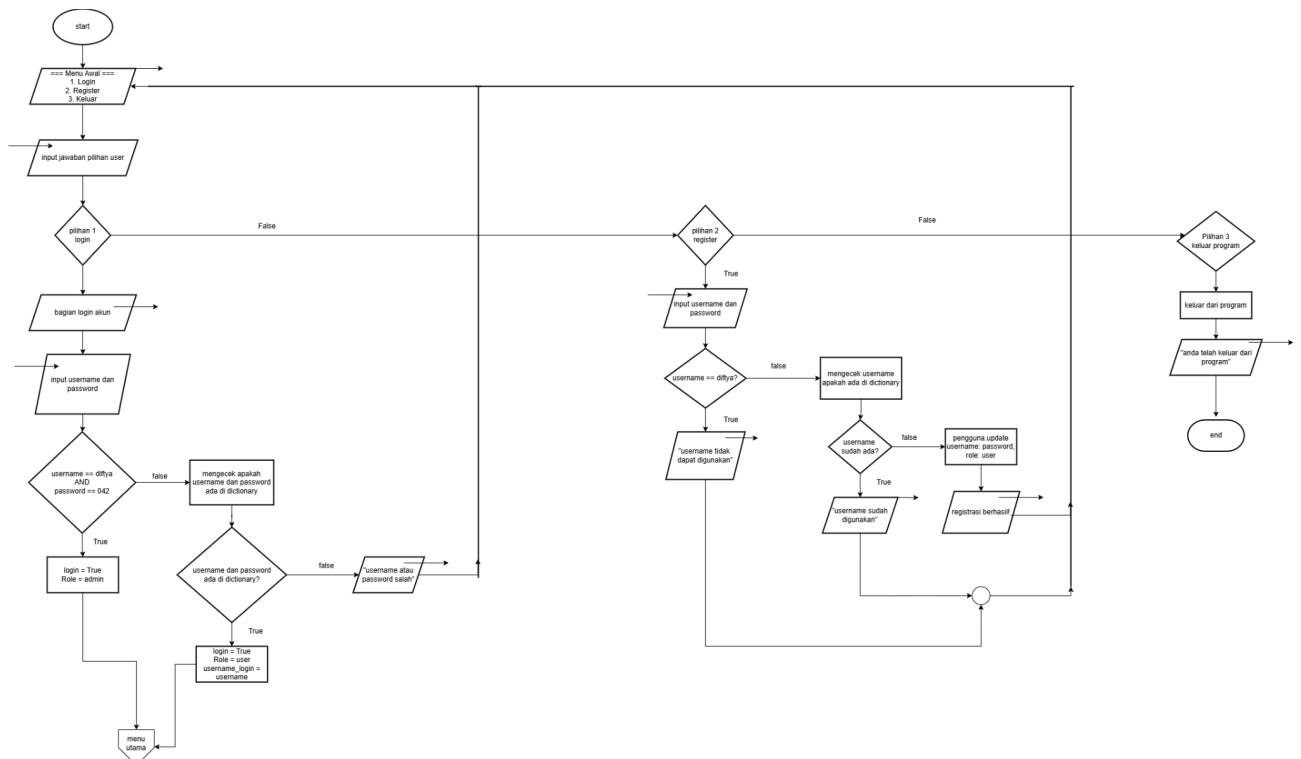


Disusun oleh:
Diftya Azzahra (2509106042)
Kelas (A2'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

Berikut adalah flowchart yang saya buat

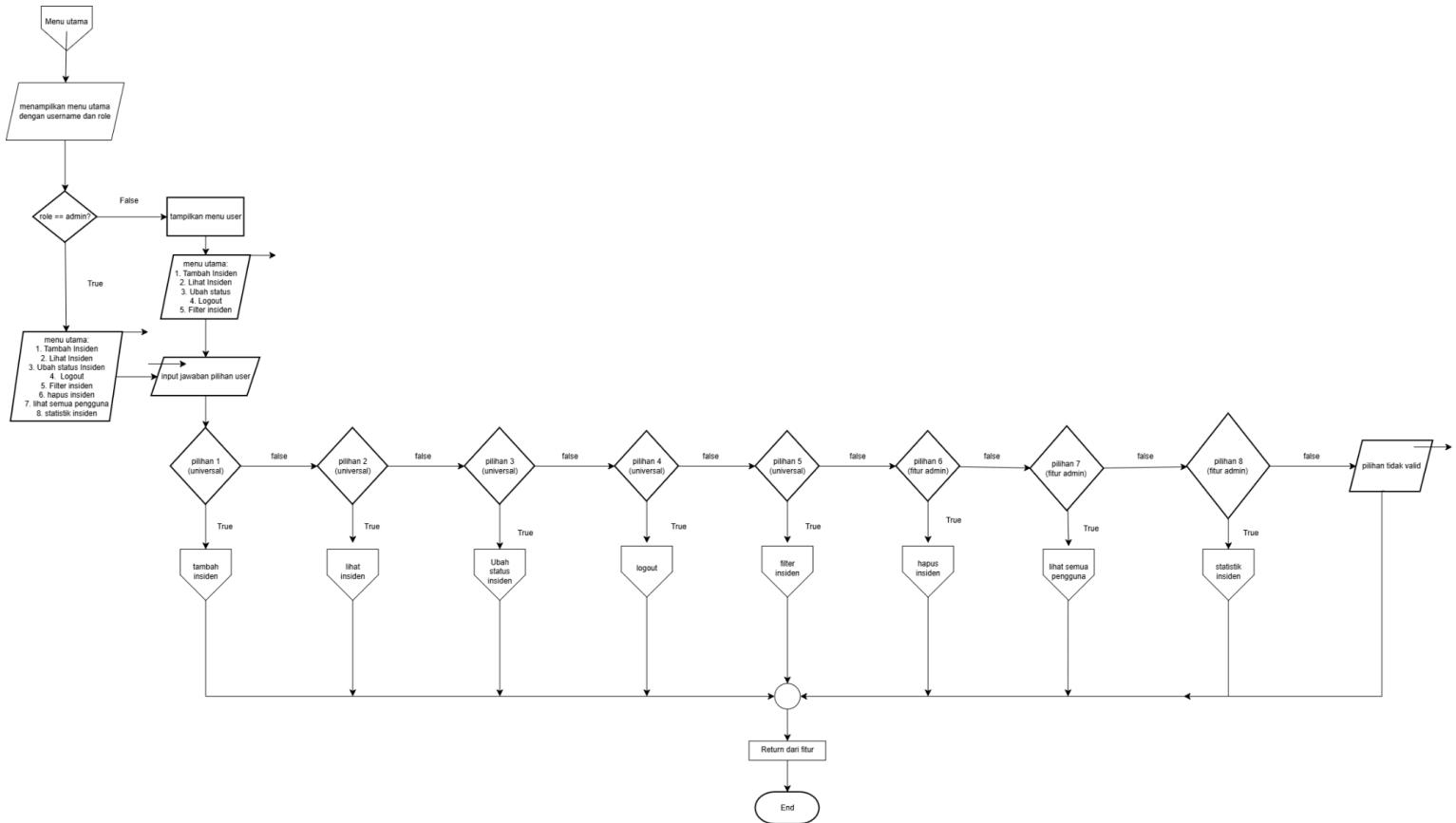


Gambar 1.1 Menu awal

1. **Start (Main):** Program dimulai
2. **Output menampilkan menu awal:** Di menu awal, sistem akan menampilkan menu yang menampilkan tiga pilihan yaitu Login, Register, dan Keluar.
3. **Input jawaban user:** Setelah menu ditampilkan, sistem **meminta input nomor pilihan menu** dari user (1 sampai 3).
4. **Decision:** Pada bagian ini, program akan menerima input berupa pilihan menu. Jika user memilih **1** maka akan masuk ke fitur login, jika memilih **2** maka user akan masuk ke fitur register, jika memilih menu ke **3** user akan keluar dari program. Jika user memilih pilihan di luar angka dan pilihan yang disediakan, sistem akan menampilkan pesan "**menu tidak valid**".
5. **Alur Pilihan 1 - Login**
6. Jika user memilih login, sistem akan masuk ke **pilihan 1 (login)**. Jika **true**, program akan menampilkan **bagian login** dan kemudian meminta **input username dan password**. Setelah data diinput, sistem melakukan pengecekan **username == diffty AND password == 042** untuk memvalidasi apakah penginput merupakan admin. Jika **true**, maka sistem akan mengatur **login = True dan Role = admin** kemudian

mengarah ke **menu utama**. Jika **false**, sistem akan melakukan pengecekan kedua yaitu username dan password ada di dictionary atau tidak. Untuk mengecek apakah user tersebut terdaftar sebagai user biasa.

7. Kalau masih tidak ada, muncul pesan "username atau password salah" lalu balik ke menu awal. Jika ada, login berhasil sebagai user biasa dan masuk ke menu utama.



Gambar 1.2 Menu Utama

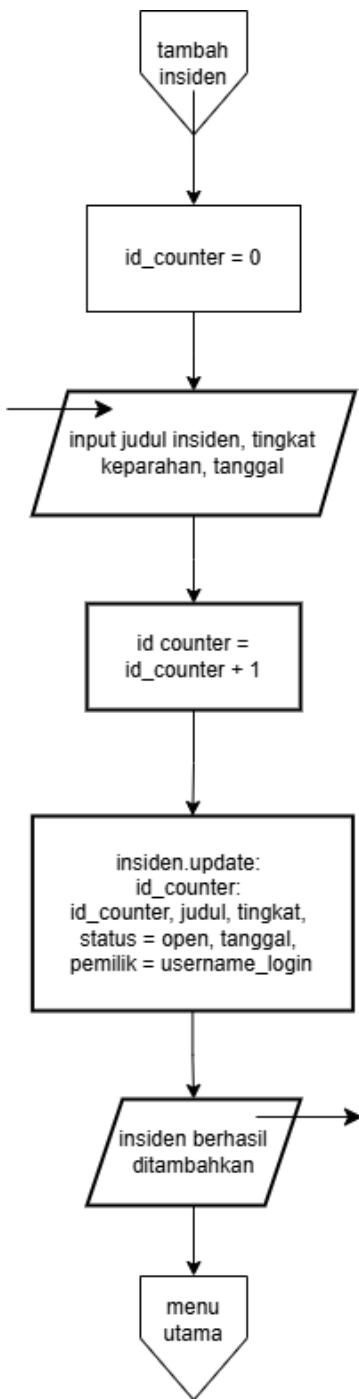
8. Alur Pilihan 2 - Register

9. Dari menu utama user memasukkan pilihan 2 (Register), maka flowchart akan masuk ke cabang pilihan.
10. Program akan meminta input berupa username dan password. Lalu sistem akan mengecek apakah username sama dengan admin. Hal ini dibuat agar user biasa tidak dapat memakai username admin. Jika **true** maka akan menampilkan pesan username tidak dapat dipakai.
11. Jika **false** maka sistem akan mengecek apakah username yang diinput oleh user terdapat pada dictionary **pengguna**. Jika **true** maka akan memunculkan pesan

username telah terdaftar. Namun jika **false**, maka username artinya username belum dipakai dan pengguna berhasil mendaftar.

12. Alur pilihan 3 - Keluar

13. Jika pengguna memilih pilihan 3 pada menu utama, maka program akan mengeluarkan output berupa pesan “**Anda telah keluar dari program**”. Lalu **break** (tidak terlihat pada flowchart) akan dijalankan agar pengguna otomatis keluar dari program.
14. Setelah login berhasil, user masuk ke **menu utama** (isi menu berbeda-beda) tergantung role.
15. Di menu utama, menampilkan daftar pilihan berbeda sesuai role. User biasa dapat 5 menu: Tambah Insiden, Lihat Insiden, Ubah Status, Logout, dan Filter Insiden. Admin dapat 8 menu, ditambah 3 menu khusus yaitu Hapus Insiden, Lihat Semua Pengguna, dan Statistik Status Insiden. User diminta **menginput** pilihan menu.
16. **Pilihan 1** kalau true masuk **Tambah insiden**. **Pilihan 2** kalau true masuk **Lihat insiden**. **Pilihan 3** kalau true masuk **Ubah status insiden**. **Pilihan 4** kalau true masuk **logout**. **Pilihan 5** kalau true masuk **filter insiden**. **Pilihan 6** (admin) kalau true masuk **hapus insiden**. **Pilihan 7** (admin) kalau true masuk **Lihat semua pengguna**. **Pilihan 8** kalau true masuk **Ubah insiden**.
17. Jika semua pilihan false, berarti user input nomor yang salah, jadi masuk ke **pilihan tidak valid**.



Gambar 1.3 Fitur tambah insiden

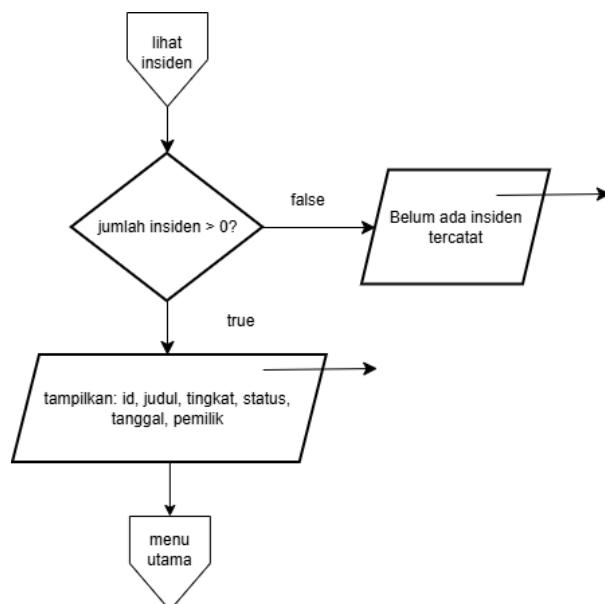
18. Dijalankan saat pengguna memilih fitur “tambah insiden”. pertama-tama dilakukan inisialisasi pada variabel **id_counter = 0**. Ini dilakukan agar program menghitung dari 1 (jika ada insiden baru yang masuk, id nya akan langsung 1 bukan 0).
19. Sistem akan meminta pengguna memasukkan data insiden seperti judul kejadian, tingkat keparahan, dan tanggal kejadian. Setelah semua input diterima, program membuat entri baru dengan menambahkan data ke daftar insiden menggunakan

```

id_counter += 1
    insiden.update({
        id_counter: {
            "id_counter" : id_counter,
            "judul" : judul,
            "tingkat" : tingkat,
            "status" : "open",
            "tanggal" : tanggal,
            "pemilik": username_login,
        }
    }

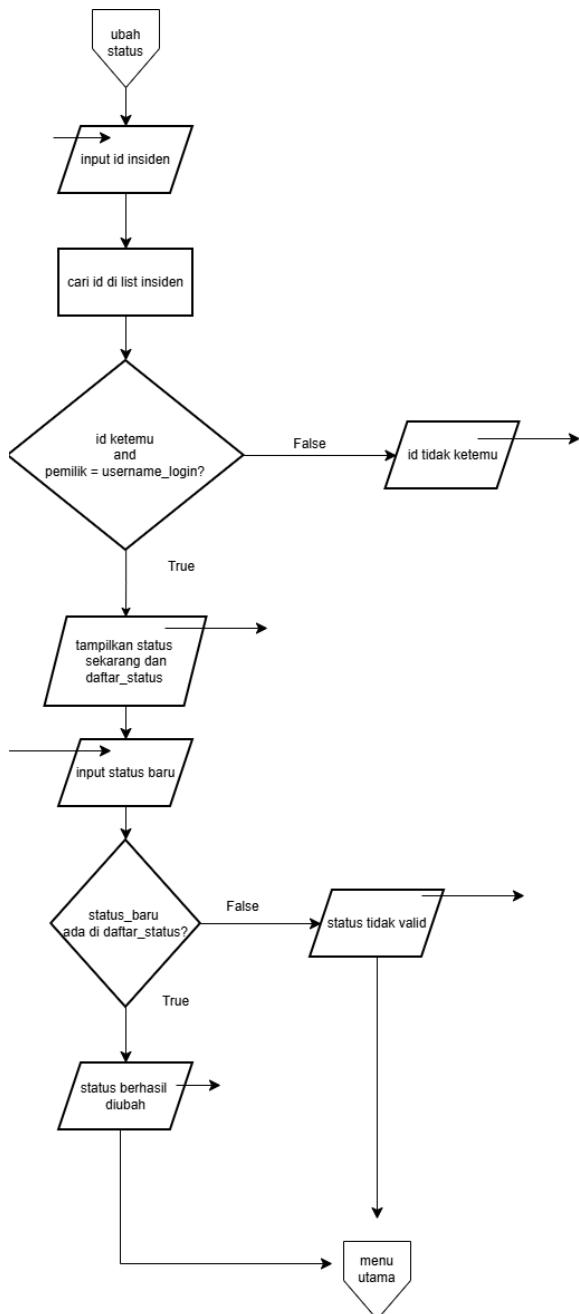
```

Note: Saya sengaja mencantumkan blok kode dictionary agar tidak terjadi miskonsepsi pada aslab yang pada akhirnya akan menyangkut flowchart. **id_counter +1** adalah penghitung id. Setiap kali pengguna menambahkan insiden baru, nilainya bertambah 1. Untuk **insiden.update**, insiden merupakan dictionary utama yang berisi semua data insiden. lalu **.update** merupakan cara untuk menambahkan data baru ke dictionary yang pada posttest sebelumnya merupakan **.append** karna tipe data masih berupa list. **id_counter** di dalam kurung kurawal pertama merupakan key utama di dictionary insiden. Nantinya, pada menu-menu lain pun akan menggunakan ID untuk mengidentifikasi sebuah insiden seperti pada fitur menghapus insiden.



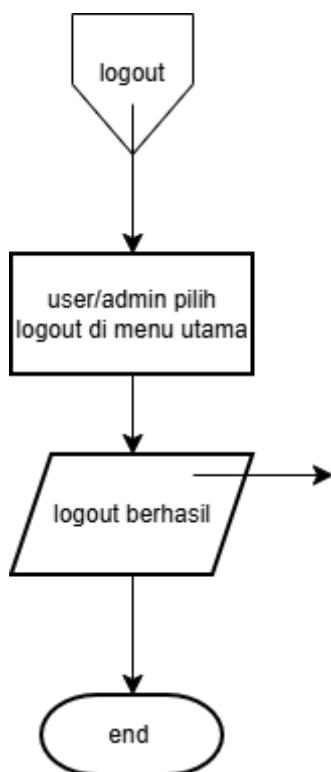
Gambar 1.4 Fitur lihat insiden

20. Ketika insiden berhasil ditambahkan. Maka sistem akan memberikan output berupa pesan “**Insiden berhasil ditambahkan**”.
21. Proses dimulai dengan pengecekan apakah ada insiden yang tersimpan di sistem. Kalau belum ada sistem akan langsung menampilkan pesan “**Belum ada insiden tercatat**”. Jika ada, sistem akan langsung menampilkan id insiden, judul, tingkat, status, tanggal terjadinya, dan pemilik atau penginput dari insiden tersebut.



Gambar 1.5 Ubah status insiden

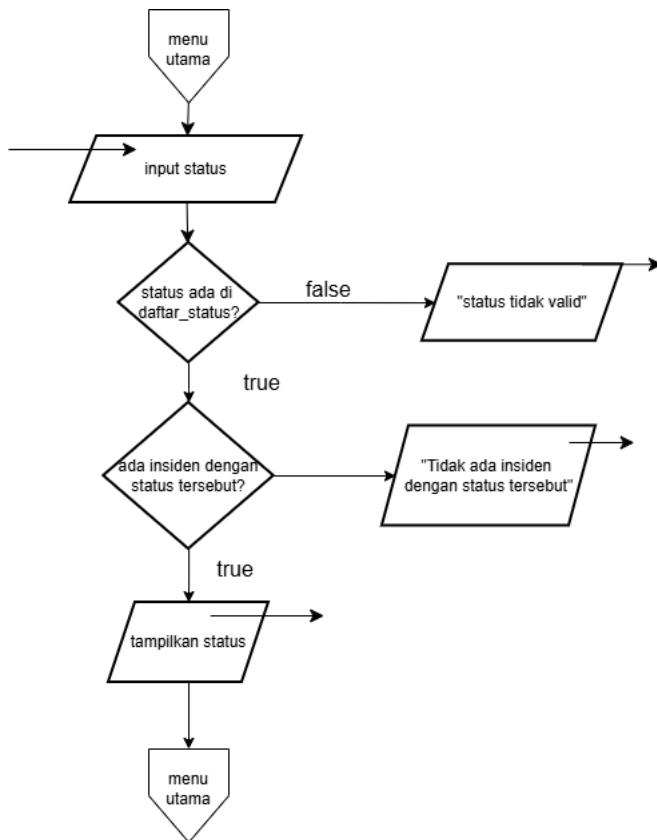
22. Dimulai ketika pengguna ingin **mengubah status insiden** dengan memilih menu nomor **3**. Pengguna perlu memasukkan **ID insiden** yang ingin diubah. Setelah itu, sistem mencari ID tersebut di daftar insiden. Kalau ID tidak ditemukan atau bukan milik pengguna yang sedang login, sistem akan menampilkan pesan “**ID tidak ketemu**” dan proses berhenti. Namun jika ID ditemukan sistem akan menampilkan **status insiden saat ini**, lalu meminta pengguna untuk memasukkan **status baru**. Sebelum pengguna memasukkan status baru, program terlebih dahulu memberikan pilihan status yaitu dengan memunculkan variabel **daftar_status** yang berisi pilihan status untuk dipilih. Setelah status baru dimasukkan, sistem memeriksa apakah status tersebut valid (tersedia di **daftar_status**). Jika status yang dimasukkan tidak valid, maka muncul pesan “**status tidak valid**” dan meminta user untuk memasukkan ulang status. Tapi jika statusnya valid, sistem akan mengganti status lama dengan status baru dan menampilkan pesan “**status berhasil diubah**”.



Gambar 1.6 Fitur Logout

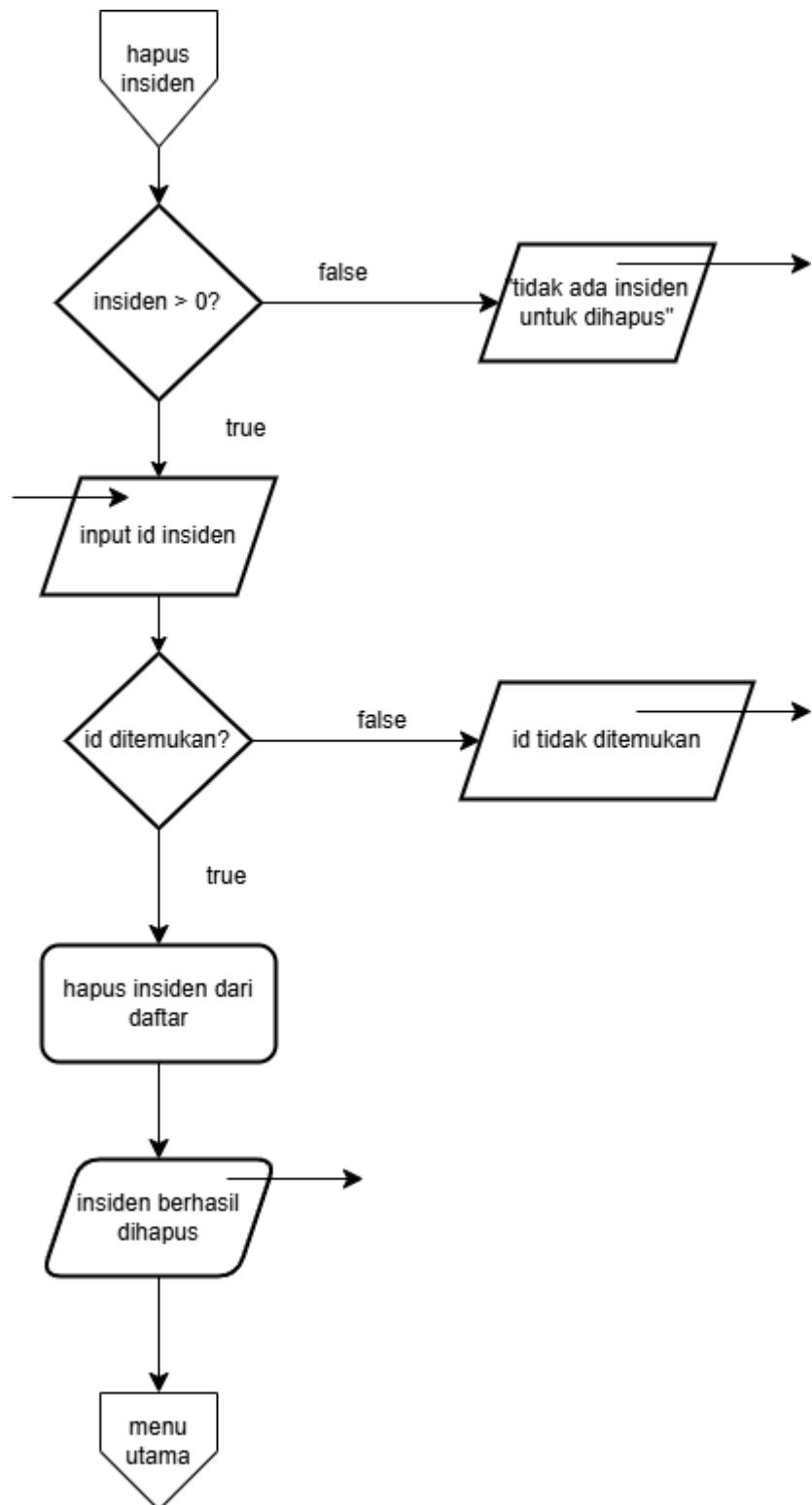
23. Berjalan ketika user memilih menu logout. Proses dimulai dari menu utama ketika user atau admin memilih opsi **logout**. Setelah pilihan itu dipilih, sistem langsung menjalankan proses logout dan menampilkan pesan “**Logout berhasil**”. Setelah itu,

alur berakhir di **end**, yang menandakan sesi pengguna sudah selesai dan mereka tidak lagi terhubung ke sistem.



Gambar 1.7 Fitur filter status insiden

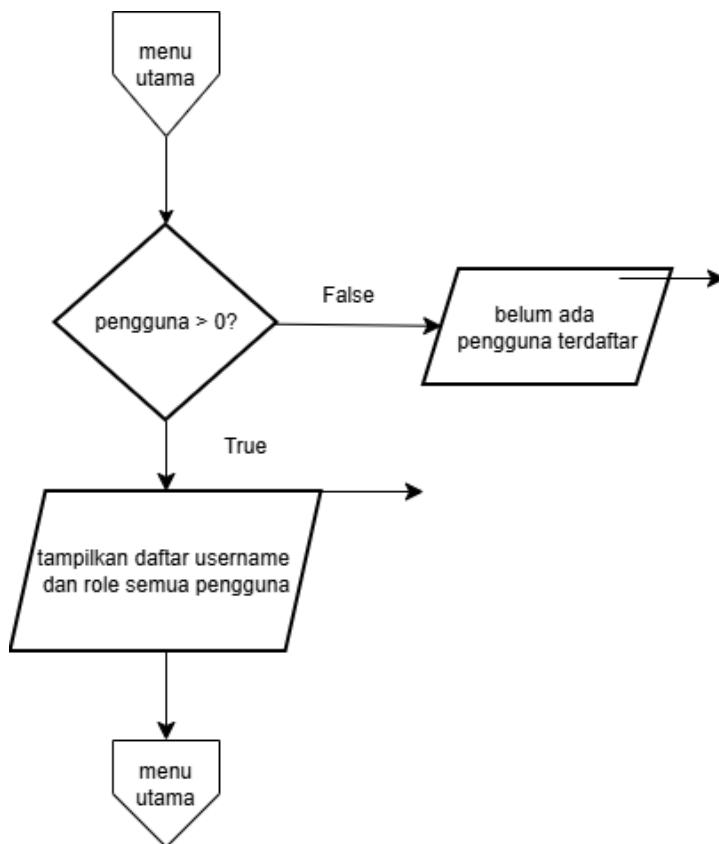
24. Setelah itu, masuk ke bagian untuk proses saat pengguna ingin memfilter daftar insiden berdasarkan status tertentu. Proses dimulai ketika user memilih nomor 5 di menu **utama**, lalu pengguna diminta untuk **memasukkan status yang ingin difilter**. Setelah itu, sistem akan memeriksa apakah status tersebut valid (tersedia di `daftar_status`). Jika statusnya tidak valid, sistem memberikan output berupa pesan “status tidak valid” dan meminta user untuk memasukkan ulang status. Namun, jika statusnya valid, sistem akan **menampilkan daftar insiden** yang sesuai dengan status tersebut.



Gambar 1.8 Fitur menghapus insiden (Khusus admin)

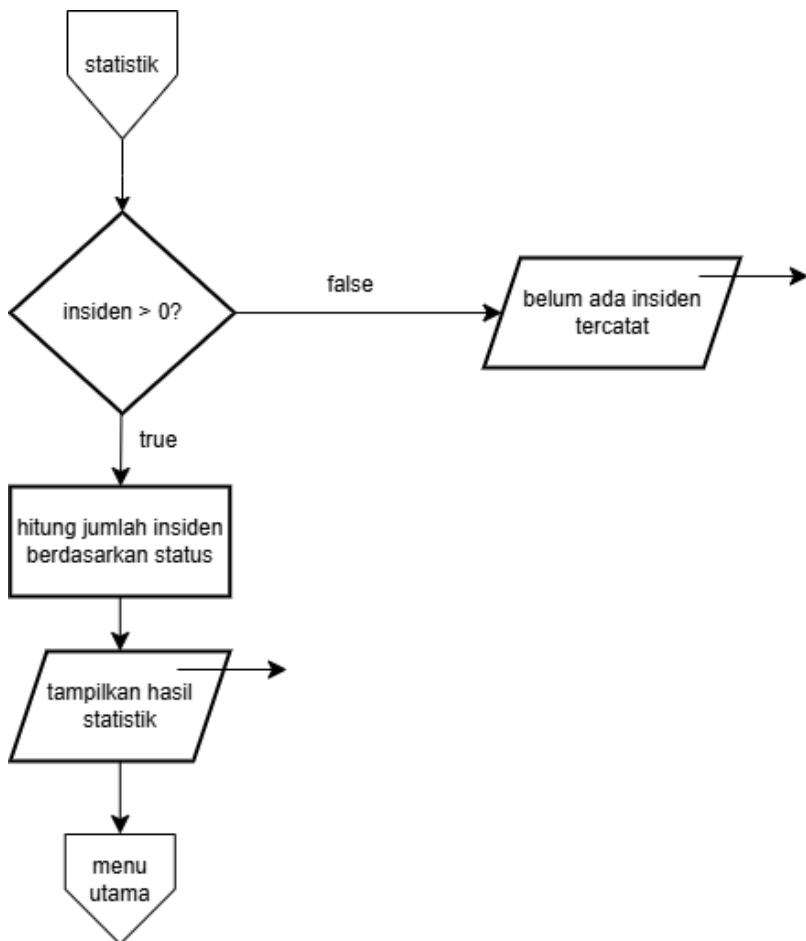
25. Bagian ini menjelaskan proses penghapusan insiden dari sistem. Proses dimulai saat admin memilih menu **hapus insiden**. Sistem terlebih dahulu mengecek apakah ada insiden yang tersimpan. Jika tidak ada, sistem langsung menampilkan pesan “**tidak**

ada insiden untuk dihapus”. Namun, jika ada, sistem meminta admin untuk memasukkan **ID insiden** yang ingin dihapus. Setelah itu, sistem mencari ID tersebut dalam daftar insiden. Jika ID tidak ditemukan, muncul pesan “**ID tidak ditemukan**”. Tapi jika ID nya ditemukan, sistem menghapus insiden dari daftar dan menampilkan pesan “**insiden berhasil dihapus**”. Setelah proses selesai, alur kembali ke **menu utama**.



Gambar 1.9 Fitur Lihat Semua Pengguna (Khusus admin)

26. Dimulai ketika admin memilih menu **lihat semua pengguna**. Setelah fitur ini dijalankan, sistem akan memeriksa apakah ada pengguna yang sudah terdaftar di dalam sistem. Jika tidak ada, maka sistem menampilkan pesan “**belum ada pengguna terdaftar**”. Namun, jika ada pengguna yang tersimpan, sistem akan **menampilkan daftar username beserta role** dari semua pengguna yang ada.



Gambar 1.10 Fitur Statistik (khusus admin)

27. Dimulai saat admin memilih menu **statistik** untuk melihat data insiden. Setelah fitur dijalankan, sistem akan memeriksa apakah ada insiden yang tersimpan di dalam data. Jika tidak ada, sistem langsung menampilkan pesan “**belum ada insiden tercatat**”. Namun, jika ada insiden tercatat, sistem akan **menghitung jumlah insiden berdasarkan statusnya**, misalnya berapa yang masih terbuka, sedang diproses, atau sudah selesai. Setelah perhitungan selesai, sistem akan menampilkan hasil statistik tersebut, lalu alur kembali ke menu utama.

2. Deskripsi Singkat Program

Program ini diibaratkan sebagai aplikasi di sebuah perusahaan keamanan IT. Jika terjadi suatu insiden seperti kebocoran data atau teretasnya suatu sistem maka pekerja biasa bisa menginput kejadian yang terjadi agar bisa dievaluasi lebih lanjut. Lalu admin adalah manager atau Team Lead yang dapat mengawasi serta memberi tahu apa saja yang harus

dilakukan. Username mewakili nama masing-masing pekerja. Username dibuat tidak boleh sama agar team lead tidak kebingungan pekerja mana yang menginput atau menemukan insiden tersebut.

3. Source Code

Di sini saya memecah program menjadi beberapa bagian yaitu:

1. Menu Awal
2. Autentikasi Program
3. Menu User
4. Menu Admin
5. Fitur Utama

3.1 Menu Awal

Pada bagian ini, terdapat menu awal yang berisi menu pilihan untuk autentikasi login pada program. Terdapat login, register, serta keluar program. Setelah login berhasil, program akan mengarahkan pengguna ke menu sesuai perannya, jika user login sebagai admin maka program akan langsung mengarahkan ke **menu admin**. Namun, jika pengguna ingin masuk sebagai user biasa, diperlukan register terlebih dahulu. Setelah itu baru user dapat login dan mendapatkan role sebagai user (tanpa privilege lebih seperti admin). Ketika pengguna sebagai user biasa telah berhasil login, program baru akan otomatis mengarahkan user ke **menu user**. Selain itu, digunakan library inquirer karena membuat tampilan menu terlihat lebih menarik. Berikut adalah source code kodingan untuk menu awal.

```
import inquirer
from auth import login_menu, register_menu, keluar_sistem
from menuAdmin import menu_admin
from menuUser import menu_user
import os

def main():
    os.system('cls')
    while True:
        tampil_menu_awal = [
            inquirer.List(
                "menu_awal",
                message="Incident Respond Management System Program",
                choices=[
                    "1. Login",
```

```

        "2. Register",
        "3. Keluar"
    ]
)
]
jawaban = inquirer.prompt(tampil_menu_awal)
menu_awal = int(jawaban["menu_awal"].split(".")[0])

if menu_awal == 1:
    os.system('cls')
    login, username_login, role = login_menu()
    if login:
        if role == "admin":
            menu_admin(username_login, role)
        else:
            menu_user(username_login, role)
    elif menu_awal == 2:
        os.system('cls')
        register_menu()
    elif menu_awal == 3:
        os.system('cls')
        keluar_sistem()
        break

if __name__ == "__main__":
    main()

```

3.2 Fitur Utama

Setelah Login selesai, pengguna akan masuk ke menu utama sesuai dengan perannya masing-masing. Namun, semua fitur-fitur inti termasuk di dalam bagian ini. Mencakup tambah insiden, lihat insiden, ubah status insiden, statistik insiden, filter insiden dan hapus insiden serta lihat pengguna. Berikut adalah source code dari fitur utama.

```

from datetime import datetime
import os
from prettytable import PrettyTable

insiden = {}
pengguna = {}
daftar_status = {
    "open": "Insiden baru dilaporkan",
    "in progress": "Insiden sedang ditangani",
    "resolved": "Insiden berhasil diselesaikan",
}

```

```

        "closed": "Kasus telah ditutup",
    }

def clear():
    os.system('cls' if os.name == 'nt' else 'clear')

def tampil_daftar_status():
    print("Pilihan status yang tersedia:")
    for k, v in daftar_status.items():
        print(f"- {k} : {v}")

def keluar_program(nama_program):
    print(f"Anda telah keluar dari {nama_program}")

def tambah_insiden(username_login):
    global insiden
    clear()
    print("===== TAMBAH INSIDEN =====")
    judul = input("Judul insiden: ")
    tingkat = input("Tingkat keparahan: ")
    status = "open"
    tanggal = str(datetime.now())[:16]
    id_counter = len(insiden) + 1

    insiden[id_counter] = {
        "id_counter": id_counter,
        "judul": judul,
        "tingkat": tingkat,
        "status": status,
        "tanggal": tanggal,
        "pemilik": username_login,
    }
    print("Insiden berhasil ditambahkan!")
    input("Tekan Enter untuk melanjutkan")

def lihat_insiden(username_login, role):
    clear()
    print("===== LIHAT INSIDEN =====")
    if len(insiden) == 0:
        print("Belum ada insiden tercatat.")
    else:
        table = PrettyTable()
        table.field_names = ["ID", "Judul", "Tingkat", "Status", "Tanggal",
"Penulis"]
        ditemukan = False
        for _, data in insiden.items():
            if role == "admin" or data["pemilik"] == username_login:
                table.add_row([

```

```

        data["id_counter"],
        data["judul"],
        data["tingkat"],
        data["status"],
        data["tanggal"],
        data["pemilik"]
    ])
ditemukan = True
if ditemukan:
    print(table)
else:
    print("Tidak ada insiden yang dapat Anda lihat.")
input("Tekan Enter untuk melanjutkan program")

def ubah_status_insiden(username_login, role):
    global insiden
    clear()
    print("===== UBAH STATUS INSIDEN =====")

    while True:
        try:
            ubah_id = int(input("Masukkan ID insiden: "))
            break
        except ValueError:
            print("ValueError: input harus berupa angka!")
            input("Tekan Enter untuk lanjut")

    if ubah_id not in insiden:
        print("ID tidak ditemukan.")
        input("Tekan Enter untuk melanjutkan")
        return

    data = insiden[ubah_id]
    if data.get("pemilik") != username_login and role != "admin":
        print("Anda tidak memiliki izin untuk mengubah insiden ini.")
        input("Tekan Enter untuk melanjutkan")
        return

    print(f"Status sekarang: {data['status']}")
    tampil_daftar_status()

    while True:
        try:
            status_baru = input("Masukkan status baru: ")
            if status_baru not in daftar_status:
                raise ValueError
            break
        except ValueError:

```

```

        print("ValueError: Status tidak valid!")
        print("Pilihan status yang tersedia:")
        for k, v in daftar_status.items():
            print(f"- {k} : {v}")
        input("Tekan Enter untuk lanjut")

data["status"] = status_baru
print("Status berhasil diubah!")
input("Tekan Enter untuk melanjutkan")

def filter_insiden(username_login, role):
    clear()
    print("===== FILTER INSIDEN =====")
    tampil_daftar_status()
    while True:
        try:
            status_filter = input("Masukkan status untuk filter: ")
            if status_filter not in daftar_status:
                raise ValueError
            break
        except ValueError:
            print("Status tidak valid! Pilihan status: open, in progress,
resolved, closed")
            input("Tekan Enter untuk lanjut")

    ditemukan = False
    table = PrettyTable()
    table.field_names = ["ID", "Judul", "Tingkat", "Status", "Tanggal",
"Pemilik"]

    for id_insiden, data in insiden.items():
        if data["status"] == status_filter and (role == "admin" or
data["pemilik"] == username_login):
            table.add_row([
                data["id_counter"],
                data["judul"],
                data["tingkat"],
                data["status"],
                data["tanggal"],
                data["pemilik"]
            ])
            ditemukan = True

    if ditemukan:
        print(table)
    else:
        print("Tidak ada insiden dengan status tersebut.")
    input("Tekan Enter untuk lanjut")

```

```

def hapus_insiden():
    global insiden
    clear()
    print("===== HAPUS INSIDEN =====")
    if len(insiden) == 0:
        print("Tidak ada insiden untuk dihapus.")
        input("Tekan Enter untuk melanjutkan program")
        return

    # Tampilkan daftar insiden terlebih dahulu
    table = PrettyTable()
    table.field_names = ["ID", "Judul", "Tingkat", "Status", "Tanggal",
    "Pemilik"]
    for _, data in insiden.items():
        table.add_row([
            data["id_counter"],
            data["judul"],
            data["tingkat"],
            data["status"],
            data["tanggal"],
            data["pemilik"]
        ])
    print(table)

try:
    hapus_id = int(input("\nMasukkan ID insiden yang ingin dihapus: "))
except ValueError:
    print("Input harus berupa angka!")
    input("Tekan Enter untuk lanjut")
    return

if hapus_id in insiden:
    del insiden[hapus_id]
    print("Insiden berhasil dihapus.")
else:
    print("ID tidak ketemu.")
input("Tekan Enter untuk melanjutkan program")

def lihat_pengguna():
    clear()
    print("===== DAFTAR PENGGUNA =====")
    if len(pengguna) == 0:
        print("Belum ada pengguna terdaftar.")
    else:
        table = PrettyTable()
        table.field_names = ["Username", "Role"]
        for username, data in pengguna.items():

```

```

        table.add_row([username, data.get("role", "user")])
    print(table)
    input("Tekan Enter untuk melanjutkan program")

def jumlah_insiden(data_list):
    if not data_list:
        return 0
    return 1 + jumlah_insiden(data_list[1:])

def statistik_insiden():
    clear()
    print("===== STATISTIK INSIDEN =====")
    if len(insiden) == 0:
        print("Belum ada insiden tercatat.")
    else:
        jumlah_status = {}
        for _, data in insiden.items():
            status = data.get("status", "unknown")
            jumlah_status[status] = jumlah_status.get(status, 0) + 1

        table = PrettyTable()
        table.field_names = ["Status", "Jumlah Insiden"]
        for status, jumlah in jumlah_status.items():
            table.add_row([status, jumlah])

        total = jumlah_insiden(list(insiden.items()))
        print(table)
        print(f"\nTotal seluruh insiden: {total}")
    input("Tekan Enter untuk lanjut")

```

3.3 Menu User

Setelah login sebagai user, menu utama pun akan otomatis tampil. Pada pengguna dengan role user biasa, terdapat beberapa fitur yaitu:

1. Tambah insiden
2. Lihat insiden
3. Ubah status insiden
4. Filter insiden
5. Logout

Lalu pada bagian ini akan berisi kodingan yang menjalankan tugas sebagai antarmuka pengguna yang mengatur interaksi antara user dan fitur utama sistem manajemen insiden. Berikut adalah source code dari menu user.

```

import inquirer
import os
from NIM_2509106042_DifftyaAzzahra_PT_8 import tambah_insiden, lihat_insiden,
ubah_status_insiden, filter_insiden

def menu_user(username_login, role):
    while True:
        os.system('cls')
        pertanyaan = [
            inquirer.List(
                'menu',
                message=f"Menu User ({username_login})",
                choices=[
                    "1. Tambah Insiden",
                    "2. Lihat Insiden",
                    "3. Ubah Status Insiden",
                    "4. Filter Insiden",
                    "5. Logout"
                ]
            )
        ]
        jawaban = inquirer.prompt(pertanyaan)
        pilih = int(jawaban['menu'].split('.')[0])

        if pilih == 1:
            tambah_insiden(username_login)
        elif pilih == 2:
            lihat_insiden(username_login, role)
        elif pilih == 3:
            ubah_status_insiden(username_login, role)
        elif pilih == 4:
            filter_insiden(username_login, role)
        elif pilih == 5:
            print("Logout berhasil.")
            input("Tekan Enter untuk lanjut")
            break

```

3.4 Menu Admin

Pada bagian ini, berfungsi sebagai menu yaitu navigasi utama untuk pengguna dengan peran admin dalam sistem. Hampir sama seperti menu user, berperan sebagai antarmuka pengguna yang mengatur interaksi namun admin memiliki hak askes lebih tinggi daripada user biasa. Pada menu admin, total terdapat 8 fitur yaitu:

1. Menambah insiden baru,
2. Melihat seluruh insiden,

3. Mengubah status insiden,
4. Melihat statistik insiden,
5. Melakukan filter insiden,
6. Menghapus insiden,
7. Melihat semua pengguna, dan
8. Logout dari sistem.

Berikut adalah source code dari menu admin.

```

import inquirer
import os
from NIM_2509106042_DifftyAzzahra_PT_8 import tambah_insiden, lihat_insiden,
ubah_status_insiden, statistik_insiden, filter_insiden, hapus_insiden,
lihat_pengguna

def menu_admin(username_login, role):
    while True:
        os.system('cls')
        pertanyaan = [
            inquirer.List(
                'menu',
                message=f"Menu Admin ({username_login})",
                choices=[
                    "1. Tambah Insiden",
                    "2. Lihat Insiden",
                    "3. Ubah Status Insiden",
                    "4. Statistik Insiden",
                    "5. Filter Insiden",
                    "6. Hapus Insiden",
                    "7. Lihat Semua Pengguna",
                    "8. Logout"
                ]
            )
        ]
        jawaban = inquirer.prompt(pertanyaan)
        pilih = int(jawaban['menu'].split('.')[0])
        os.system('cls')

        if pilih == 1:
            os.system('cls')
            tambah_insiden(username_login)
        elif pilih == 2:
            os.system('cls')
            lihat_insiden(username_login, role)
        elif pilih == 3:
            os.system('cls')
            ubah_status_insiden(username_login)
        elif pilih == 4:
            os.system('cls')
            statistik_insiden(username_login)
        elif pilih == 5:
            os.system('cls')
            filter_insiden(username_login)
        elif pilih == 6:
            os.system('cls')
            hapus_insiden(username_login)
        elif pilih == 7:
            os.system('cls')
            lihat_pengguna(username_login)
        elif pilih == 8:
            print("Logout berhasil!")
            break
        else:
            print("Pilihan tidak valid. Silakan coba lagi.")

```

```
os.system('cls')
ubah_status_insiden(username_login, role)
elif pilih == 4:
    os.system('cls')
    statistik_insiden()
elif pilih == 5:
    os.system('cls')
    filter_insiden(username_login, role)
elif pilih == 6:
    os.system('cls')
    hapus_insiden()
elif pilih == 7:
    os.system('cls')
    lihat_pengguna()
elif pilih == 8:
    os.system('cls')
    print("Logout berhasil.")
    input("Tekan Enter untuk lanjut...")
break
```

4. Hasil Output

```
[?] Incident Respond Management System Program:
> 1. Login
  2. Register
  3. Keluar
```

Gambar 4.1 Menu awal

```
===== LOGIN =====
Username: diftya
Password: 042
Login Berhasil (Admin)
Tekan Enter untuk lanjut
```

Gambar 4.2 Login berhasil (Admin)

```
===== LOGIN =====
Username: yaya
Password: biskuit yaya
Username atau password salah.
Tekan Enter untuk lanjut
```

Gambar 4.3 Login Gagal

```
===== REGISTER =====
Masukkan username anda: yaya
Masukkan password: biskuit yaya
Registrasi berhasil.
Tekan Enter untuk lanjut
```

Gambar 4.4 Registrasi Berhasil

```
===== REGISTER =====
Masukkan username anda: yaya
Masukkan password: biskuit yaya
Username telah terdaftar.
Tekan Enter untuk lanjut
```

Gambar 4.5 Registrasi Gagal (Username telah terdaftar)

```
===== REGISTER =====
Masukkan username anda: diftya
Masukkan password: 123
Username ini tidak bisa dipakai.
Tekan Enter untuk lanjut
```

Gambar 4.6 Registrasi Gagal (Username admin)

```
===== LOGIN =====
Username: serigala jika
Password: hawimau hawimau
Login Berhasil (User)
Tekan Enter untuk lanjut
```

Gambar 4.7 Login Berhasil (User)

```
[?] Menu Admin (diftya):
> 1. Tambah Insiden
  2. Lihat Insiden
  3. Ubah Status Insiden
  4. Statistik Insiden
  5. Filter Insiden
  6. Hapus Insiden
  7. Lihat Semua Pengguna
  8. Logout
```

Gambar 4.8 Menu Admin

```
===== TAMBAH INSIDEN =====
Judul insiden: diretasnya aplikasi A
Tingkat keparahan: high
Insiden berhasil ditambahkan!
Tekan Enter untuk melanjutkan
```

Gambar 4.9 Fitur Tambah Insiden

```
===== LIHAT INSIDEN =====
+---+---+---+---+---+
| ID | Judul | Tingkat | Status | Tanggal      | Pemilik |
+---+---+---+---+---+
| 1  |        | open   | 2025-11-05 04:02 | diftya    |
+---+---+---+---+---+
Tekan Enter untuk melanjutkan program
```

Gambar 4.10 Fitur Lihat Insiden

```
===== UBAH STATUS INSIDEN =====
Masukkan ID insiden: 1
Status sekarang: open
Pilihan status yang tersedia:
- open : Insiden baru dilaporkan
- in progress : Insiden sedang ditangani
- resolved : Insiden berhasil diselesaikan
- closed : Kasus telah ditutup
Masukkan status baru: resolved
Status berhasil diubah!
Tekan Enter untuk melanjutkan
```

Gambar 4.11 Ubah Status Insiden

```
===== STATISTIK INSIDEN =====
+-----+-----+
| Status | Jumlah Insiden |
+-----+-----+
| resolved | 1 |
| open | 1 |
+-----+-----+
Total seluruh insiden: 2
Tekan Enter untuk lanjut
```

Gambar 4.12 Statistik Insiden

```
===== FILTER INSIDEN =====
Pilihan status yang tersedia:
- open : Insiden baru dilaporkan
- in progress : Insiden sedang ditangani
- resolved : Insiden berhasil diselesaikan
- closed : Kasus telah ditutup
Masukkan status untuk filter: open
+-----+-----+-----+-----+-----+
| ID | Judul | Tingkat | Status | Tanggal | Pemilik |
+-----+-----+-----+-----+-----+
| 2 | terdeteksi adanya IDOR | Medium | open | 2025-11-05 04:05 | diftya |
+-----+-----+-----+-----+-----+
Tekan Enter untuk lanjut
```

Gambar 4.13 Filter Insiden

```
===== HAPUS INSIDEN =====
+-----+-----+-----+-----+
| ID | Judul | Tingkat | Status | Tanggal | Pemilik |
+-----+-----+-----+-----+
| 3 | kebocoran data pada server | critical | open | 2025-11-05 04:11 | diftya |
+-----+-----+-----+-----+
Masukkan ID insiden yang ingin dihapus: 3
Insiden berhasil dihapus.
Tekan Enter untuk melanjutkan program
```

Gambar 4.14 Hapus insiden

```
===== DAFTAR PENGGUNA =====
+-----+-----+
| Username | Role |
+-----+-----+
| yaya | user |
| serigala jika | user |
+-----+-----+
Tekan Enter untuk melanjutkan program
```

Gambar 4.15 Lihat Semua Pengguna

```
Logout berhasil.
Tekan Enter untuk lanjut...■
```

Gambar 4.16 Logout

5. Langkah-langkah GIT

5.1 GIT Add

```
PS C:\Users\ASUS\Documents\Kuliah\Praktikum APD\post-test\post-test-apd-8> git add .
PS C:\Users\ASUS\Documents\Kuliah\Praktikum APD\post-test\post-test-apd-8>
```

5.2 GIT Commit

```
PS C:\Users\ASUS\Documents\Kuliah\Praktikum APD\post-test\post-test-apd-8> git commit -m "ngantuk"
[main d18b8a1] ngantuk
 4 files changed, 9 insertions(+), 9 deletions(-)
```

5.3 GIT Remote

```
PS C:\Users\ASUS\Documents\Kuliah\Praktikumm-APD\post-test\post-test-apd-8> git remote -v
origin  https://github.com/HikaruYui/Praktikumm-APD.git (fetch)
origin  https://github.com/HikaruYui/Praktikumm-APD.git (push)
```

5.4 GIT Push

```
PS C:\Users\ASUS\Documents\Kuliah\Praktikumm-APD\post-test\post-test-apd-8> git push origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 2.19 KiB | 204.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
To https://github.com/HikaruYui/Praktikumm-APD.git
 a7301ea..d18b8a1  main -> main
```

5.4 Github

The screenshot shows a GitHub repository interface for 'Praktikumm-APD / post-test / post-test-apd-8'. At the top, there's a header with 'Add file' and a three-dot menu icon. Below the header, a user profile for 'HikaruYui' is shown, along with a commit message 'ngantuk' and a timestamp 'd18b8a1 - 3 minutes ago'. A 'History' button is also present. The main area displays a table of files with their last commit details:

Name	Last commit message	Last commit date
..		
pycache	ngantuk	3 minutes ago
NIM_2509106042_DiftyaAzzahra_PT_8.py	ngantuk	3 minutes ago
auth.py	ngantuk	3 minutes ago
menuAdmin.py	urutan run dari menu awal dulu	2 hours ago
menuAwal.py	urutan run dari menu awal dulu	2 hours ago
menuUser.py	urutan run dari menu awal dulu	2 hours ago