

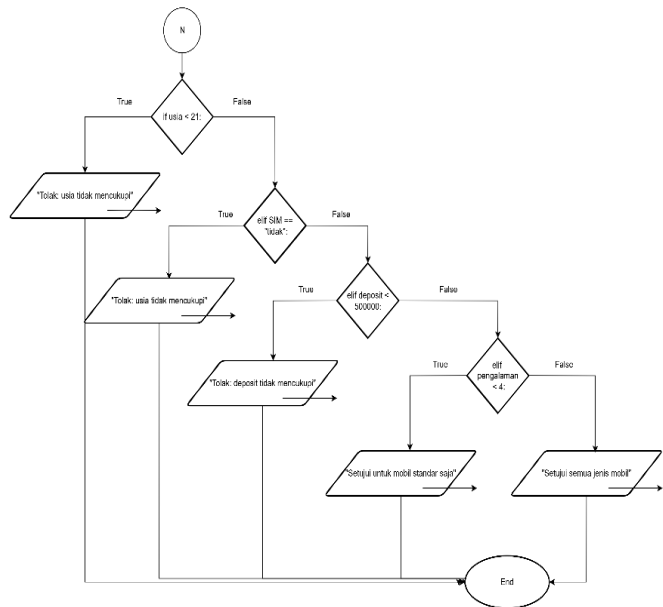
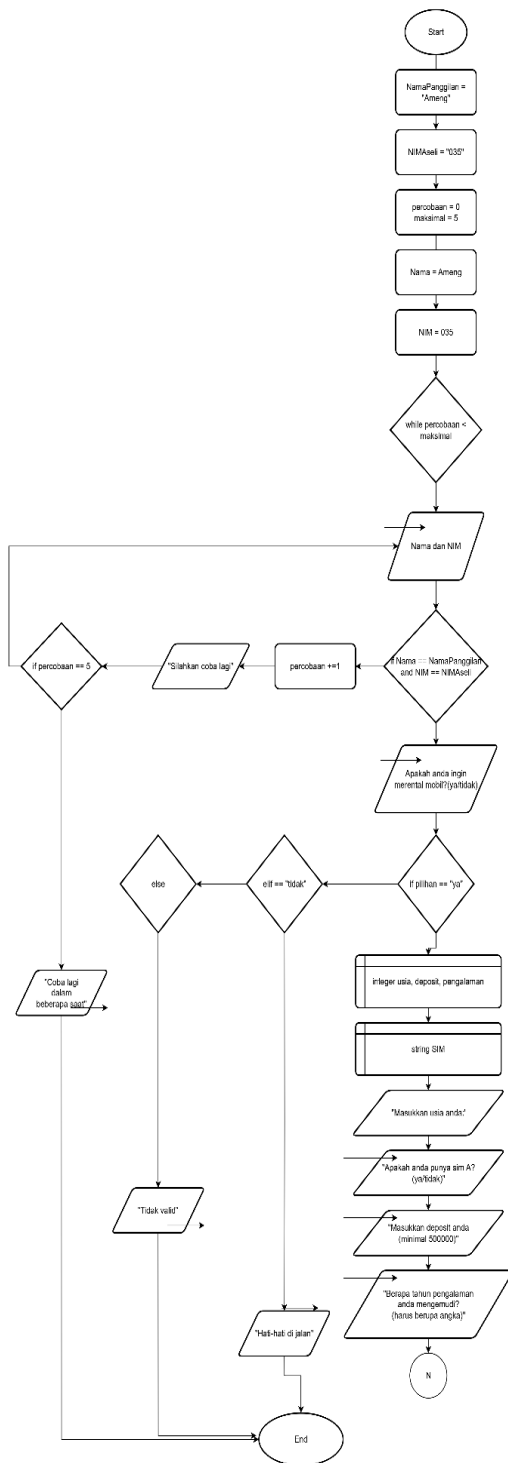
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Ammara Fadia FanySarahman (2509106035)**  
**Kelas (A'25)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

# 1. Flowchart



## 2. Deskripsi Singkat Program

Program ini dibuat untuk me-rental mobil dengan beberapa persyaratan yang harus dipenuhi, seperti usia minimal, adanya SIM, minimal deposit, dan lamanya penagalaman mengemudi. untuk menentukan mobil yang bisa dirental atau menolak jika ada beberapa persyaratan yang tidak terpenuhi. kita harus melakukan login terlebih dahulu lalu lalu memilih untuk merental mobil atau tidak.

## 3. Source Code

### A. Login

Ini digunakan untuk melakukan login tetapi memiliki batas tertentu untuk melakukan login.

#### Source Code:

```
percobaan = 0
maksimal = 5
while percobaan < maksimal:

    Nama = (input("Masukkan Nama Anda: "))
    NIM = (input("Masukkan NIM anda: "))
    if Nama == NamaPanggilan and NIM == NIMAseli:
        percobaan = maksimal
        print(f'Selamat datang {Nama}, silahkan ke tahap selanjutnya!')
        break

    else:
        percobaan += 1
        sisapercobaan = maksimal - percobaan
        if sisapercobaan > 0:
            print(f'Silahkan coba lagi")
        else:
            print("Coba lagi dalam beberapa saat")
            exit()
```

## B. Input data

Ini digunakan untuk menginput data yang nantinya data tersebut akan diproses di dalam kondisi.

### Source Code:

```
usia = int(input("Masukkan usia anda: "))
SIM = input("Apakah anda punya SIM A? (ya/tidak): ")
deposit = int(input("Masukkan deposit anda (minimal 500000): "))
pengalaman = int(input("Berapa tahun pengalaman anda mengemudi (harus berupa angka): "))
```

## C. Kondisi (if, elif, else)

Kondisi yang nantinya akan menampilkan data sesuai dengan persyaratan (jika menjawab ya). jika menjawab tidak, maka program selesai.

### Source Code:

```
pilihan = (input("\nApakah anda ingin merental mobil?(ya/tidak))
if pilihan == "ya":

    usia = int(input("Masukkan usia anda: "))
    SIM = input("Apakah anda punya SIM A? (ya/tidak): ")
    deposit = int(input("Masukkan deposit anda (minimal 500000):
    "))
    pengalaman = int(input("Berapa tahun pengalaman anda mengemudi
    (harus berupa angka): "))

    if usia < 21:
        print("Tolak: Usia tidak mencukupi")
    elif SIM == "tidak":
        print("Tolak: Tidak memiliki sim A")
    elif deposit < 500000:
        print("Tolak: Deposit tidak mencukupi")
    elif pengalaman < 4:
        print("Setujui untuk mobil standar saja")
    else:
        print("Setujui untuk semua jenis mobil")
```

```
elif pilihan == "tidak":  
    print("Hati-hati di jalan")  
else:  
    print("Tidak valid!")
```

#### 4. Hasil Output

```
Masukkan usia anda: 20  
Apakah anda punya SIM A? (ya/tidak): tidak  
Masukkan deposit anda (minimal 500000): 200000  
Berapa tahun pengalaman anda mengemudi (harus berupa angka): 2  
Tolak: Usia tidak mencukupi  
PS C:\Users\ASUS>
```

Gambar 4.1 Output pertama  
Tidak memenuhi persyaratan usia.

```
Masukkan usia anda: 22  
Apakah anda punya SIM A? (ya/tidak): tidak  
Masukkan deposit anda (minimal 500000): 200000  
Berapa tahun pengalaman anda mengemudi (harus berupa angka): 2  
Tolak: Tidak memiliki sim A  
PS C:\Users\ASUS>
```

Gambar 4.2 Output kedua  
Tidak memenuhi persyaratan SIM.

```
Masukkan usia anda: 22  
Apakah anda punya SIM A? (ya/tidak): ya  
Masukkan deposit anda (minimal 500000): 200000  
Berapa tahun pengalaman anda mengemudi (harus berupa angka): 2  
Tolak: Deposit tidak mencukupi  
PS C:\Users\ASUS>
```

Gambar 4.3 Output ketiga  
Tidak memenuhi persyaratan deposit.

```
Masukkan usia anda: 22
Apakah anda punya SIM A? (ya/tidak): ya
Masukkan deposit anda (minimal 500000): 500000
Berapa tahun pengalaman anda mengemudi (harus berupa angka): 2
Setujui untuk mobil standar saja
PS C:\Users\ASUS>
```

Gambar 4.4 Output keempat

Tidak memenuhi persyaratan untuk memilih semua jenis mobil.

```
Masukkan usia anda: 22
Apakah anda punya SIM A? (ya/tidak): ya
Masukkan deposit anda (minimal 500000): 500000
Berapa tahun pengalaman anda mengemudi (harus berupa angka): 4
Setujui untuk semua jenis mobil
PS C:\Users\ASUS>
```

Gambar 4.5 Output kelima

Memenuhi semua persyaratan.

```
Masukkan Nama Anda: Ameng
Masukkan NIM anda: 035
Selamat datang Ameng, silahkan ke tahap selanjutnya!

Apakah anda ingin merental mobil?(ya/tidak)ya
```

Gambar 4.6 Output keenam

Memenuhi semua persyaratan login.

```
Masukkan Nama Anda: Ara
Masukkan NIM anda: 039
Silahkan coba lagi
Masukkan Nama Anda:
```

Gambar 4.7 Output ketujuh

Tidak memenuhi semua persyaratan.

```
Masukkan Nama Anda: Ameng
Masukkan NIM anda: 035
Selamat datang Ameng, silahkan ke tahap selanjutnya!

Apakah anda ingin merental mobil?(ya/tidak)tidak
Hati-hati di jalan
```

Gambar 4.8 Output kedelapam  
Memenuhi semua persyaratan tapi memilih “tidak”.

## 5. Langkah-langkah GIT

```
(c) Microsoft Corporation. All rights reserved.
C:\Users\ASUS\Documents\praktikum-apd>git add .

C:\Users\ASUS\Documents\praktikum-apd>git commit -m "PT 4"
[main 980a86b] PT 4
 2 files changed, 137 insertions(+)
 create mode 100644 kelas/pertemuan-4/main.py
 create mode 100644 post-test/post-test-apd-4/2509106035-Ammara Fadia-PT-4

C:\Users\ASUS\Documents\praktikum-apd>git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.63 KiB | 555.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ammara fadia/praktikum-apd.git
   8969d19..980a86b  main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\ASUS\Documents\praktikum-apd>
```

Gambar 5.1  
GIT.

### 5.1 GIT Add

Add adalah perintah yang berguna untuk staging atau untuk menambahkan perubahan atau registrasi perubahan pada git proyek yang ada.

## **5.2 GIT Commit**

Commit adalah perintah yang digunakan untuk menyimpan perubahan pada repository git. Kita juga bisa mengembalikan file yang kita commit ke revisi sebelumnya.

## **5.3 GIT Push**

Push adalah perintah yang digunakan untuk mengunggah konten pada repositori lokal.