

**LAPORAN PRAKTIKUM  
POSTTEST(8)  
ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**

**Nama (2509106043)**

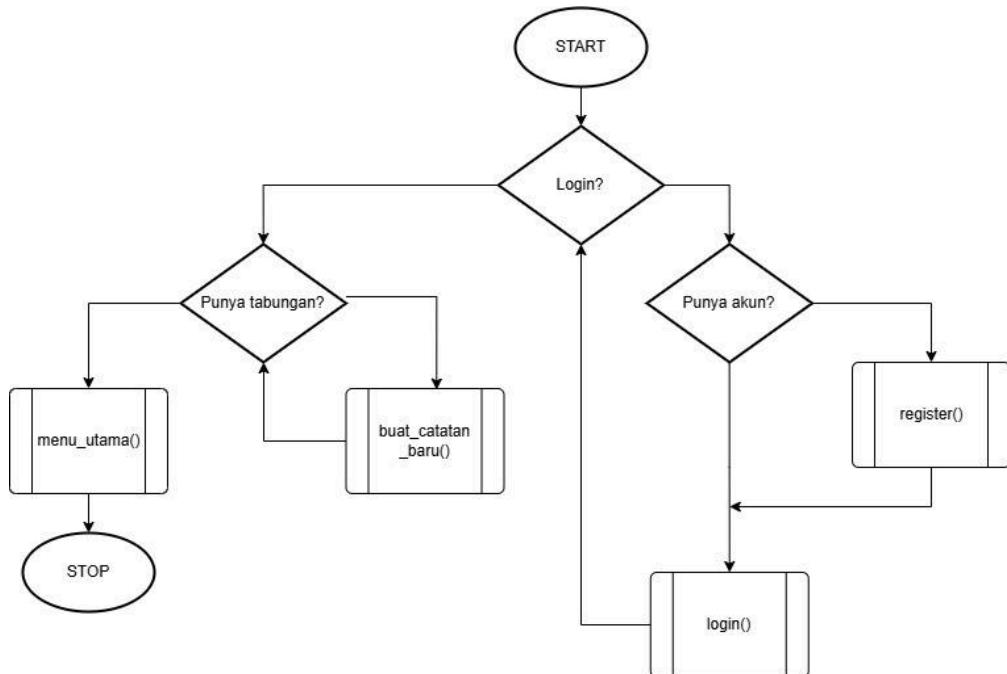
**Kelas (A2'25)**

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS MULAWARMAN  
SAMARINDA**

**2025**

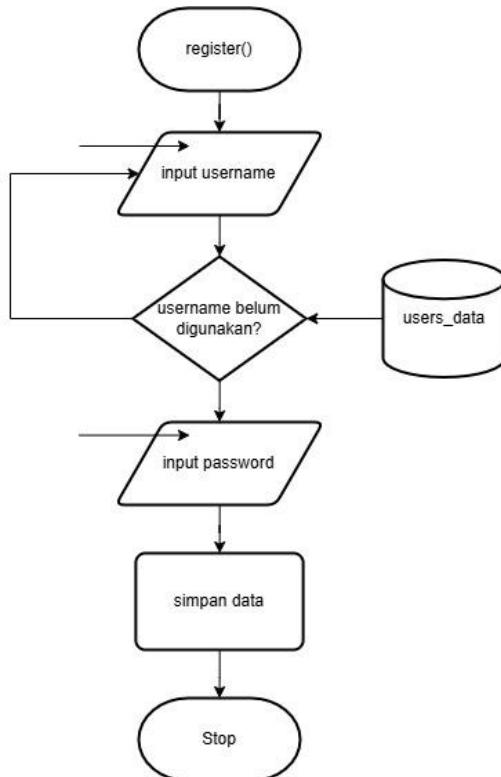
## 1. Flowchart

### 1.1 Program



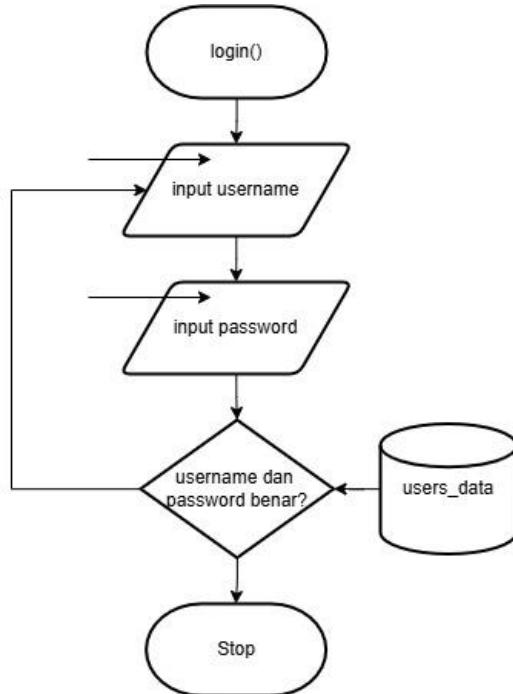
Gambar 1.1 Flowchart Program Pencatat Keuangan (1)

### 1.2 Register



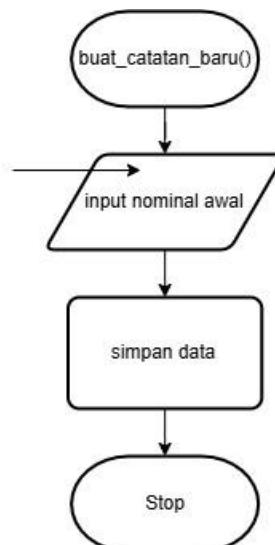
Gambar 1.2 Flowchart Program Pencatat Keuangan (2)

### 1.3 Login



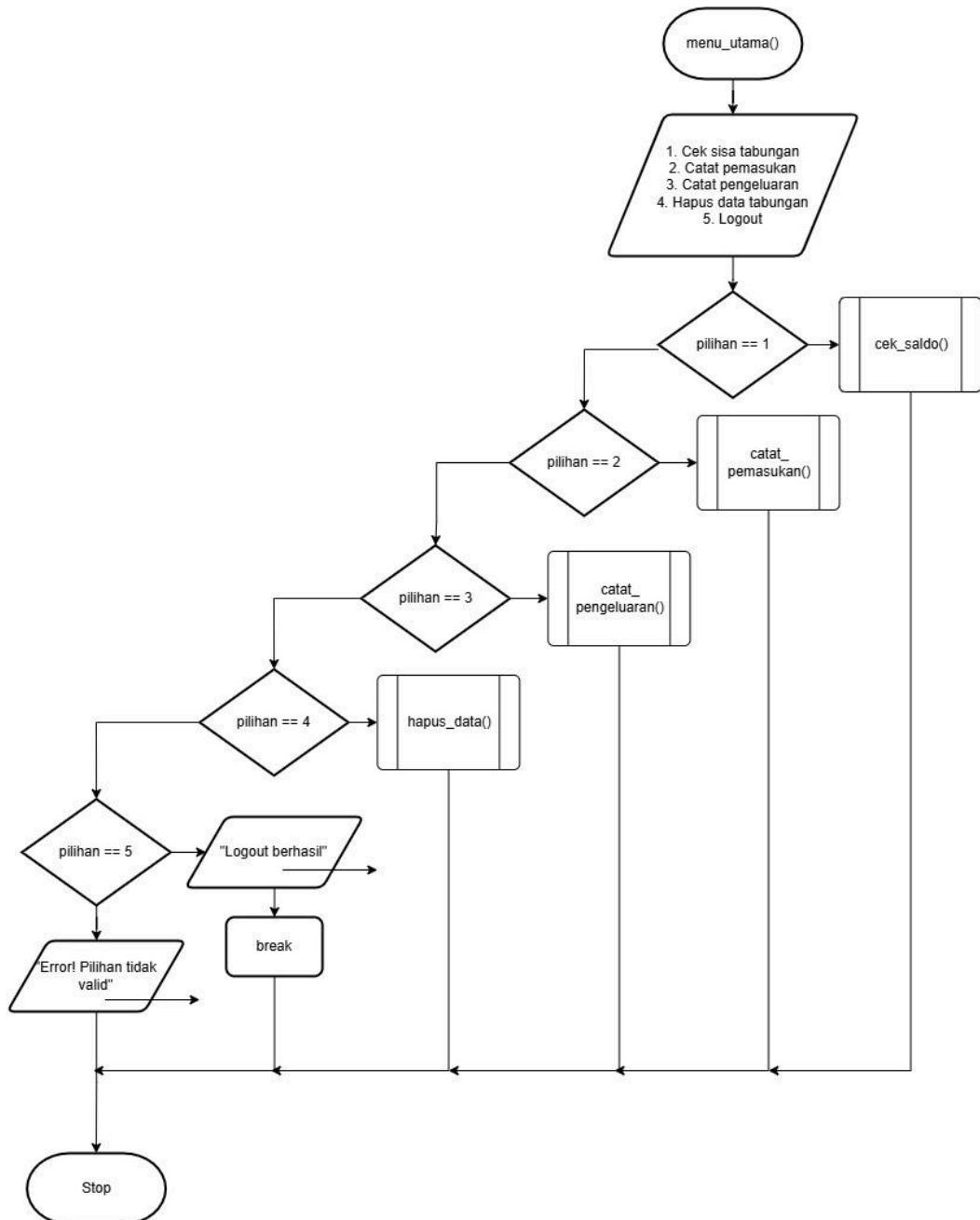
Gambar 1.3 Flowchart Program Pencatat Keuangan (3)

### 1.4 Buat Catatan Baru



Gambar 1.4 Flowchart Program Pencatat Keuangan (4)

## 1.5 Menu Utama



Gambar 1.5 Flowchart Program Pencatat Keuangan (5)

## **2. Deskripsi Singkat Program**

Program Pencatat Keuangan adalah aplikasi sederhana yang dirancang untuk membantu pengguna dalam mencatat dan mengelola keuangan pribadi mereka. Program ini menggunakan struktur dictionary untuk menyimpan informasi saldo dan riwayat transaksi. Saat pertama kali dijalankan, pengguna akan diminta untuk membuat catatan baru dengan memasukkan nominal awal tabungan. Setelah catatan dibuat, pengguna dapat mengakses menu utama yang menyediakan berbagai fitur seperti mengecek sisa tabungan beserta riwayat transaksi, mencatat pemasukan dengan keterangan opsional, mencatat pengeluaran yang dilengkapi dengan peringatan jika nominal melebihi saldo, menghapus seluruh data catatan untuk memulai dari awal, dan keluar dari program.

Program ini dibangun dengan konsep loop yang memungkinkan pengguna untuk terus menggunakan berbagai fitur secara berulang tanpa harus keluar dan menjalankan ulang aplikasi. Setiap transaksi yang dicatat akan secara otomatis memperbarui saldo dan menambahkan riwayat transaksi ke dalam list. Program menggunakan validasi input untuk memastikan pengguna hanya memasukkan angka pada nominal dan memberikan feedback berupa pesan error jika input tidak sesuai. Data yang tersimpan bersifat sementara dan akan hilang ketika program ditutup karena tidak menggunakan penyimpanan file eksternal.

### 3. Source Code

#### 3.1 Main

```
from utils import bersihkan_layar
from autentikasi import login
from menu import menu_utama
from transaksi import buat_catatan_baru

users_data = {}
data_keuangan = {}

def main():
    print("Memulai Program Pencatat Keuangan...")
    print("Loading...")

    while True:
        current_user = login(users_data, data_keuangan)

        if current_user is None:
            break

        while True:
            if current_user not in data_keuangan or
data_keuangan[current_user] == {}:
                result = buat_catatan_baru(current_user, data_keuangan)
                if result == "logout":
                    break
                else:
                    menu_utama(current_user, data_keuangan)
                    break

    print("\nProgram selesai. Terima kasih!")

if __name__ == "__main__":
    main()
```

### 3.2 Utils

```
import os
from prettytable import PrettyTable

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

def tampil_header(judul):
    table = PrettyTable()
    table.field_names = [judul.upper()]
    table.align = "c"
    table.max_width = 50
    print(table)

def validasi_angka(input_str):
    try:
        angka = int(input_str)
        if angka < 0:
            return False, 0
        return True, angka
    except ValueError:
        return False, 0

def tampil_pesan(pesan, tipe="INFO"):
    table = PrettyTable()
    table.field_names = [tipe]
    table.add_row([pesan])
    table.align = "l"
    print(table)

def input_dengan_prompt(prompt):
    print(f">> {prompt}", end="")
    return input()

def pause():
    input("\nTekan Enter untuk melanjutkan...")
```

### 3.3 Autentikasi

```
from utils import bersihkan_layar, tampil_header, tampil_pesan,
input_dengan_prompt, pause
from prettytable import PrettyTable

def register(users_data, data_keuangan):
    bersihkan_layar()
    tampil_header("REGISTER AKUN BARU")
    print()

    username_baru = input_dengan_prompt("Masukkan username baru: ")

    if username_baru == "":
        tampil_pesan("Username tidak boleh kosong!", "ERROR")
        pause()
        return False, None

    if username_baru in users_data:
        tampil_pesan("Username sudah digunakan!", "ERROR")
        pause()
        return False, None

    password_baru = input_dengan_prompt("Masukkan password: ")

    if password_baru == "":
        tampil_pesan("Password tidak boleh kosong!", "ERROR")
        pause()
        return False, None

    konfirmasi_password = input_dengan_prompt("Konfirmasi password: ")

    if password_baru != konfirmasi_password:
        tampil_pesan("Password tidak cocok!", "ERROR")
        pause()
        return False, None

    users_data[username_baru] = password_baru
    data_keuangan[username_baru] = {}

    tampil_pesan(f"Akun {username_baru} berhasil dibuat!", "SUKSES")
    pause()
```

```

        return True, username_baru

def login(users_data, data_keuangan):
    percobaan = 0

    while percobaan < 3:
        bersihkan_layar()
        tampil_header("LOGIN")

        # login
        table = PrettyTable()
        table.field_names = ["OPSI"]
        table.add_row(["Belum punya akun? Ketik 'register' pada username"])
        table.add_row(["Ingin keluar? Ketik 'exit' pada username"])
        table.align = "l"
        print(table)
        print()

        username = input_dengan_prompt("Username: ")

        if username.lower() == "register":
            success, new_user = register(users_data, data_keuangan)
            if success:
                continue
            else:
                continue

        if username.lower() == "exit":
            tampil_pesan("Terima kasih! Program ditutup.", "INFO")
            pause()
            return None

        password = input_dengan_prompt("Password: ")

        if username in users_data and users_data[username] == password:
            tampil_pesan(f"Login berhasil! Selamat datang, {username}!", "SUKSES")
            pause()
            return username
        else:
            percobaan += 1

```

```

        tampil_pesan(f"Username atau password salah. Percobaan
{percobaan}/3", "ERROR")

        if percobaan >= 3:
            tampil_pesan("Terlalu banyak percobaan login. Program
ditutup.", "ERROR")
            pause()
            return None
        else:
            pause()

    return None

```

### 3.4 Transaksi

```

from utils import bersihkan_layar, tampil_header, tampil_pesan,
input_dengan_prompt, pause, validasi_angka
from prettytable import PrettyTable

def buat_catatan_baru(current_user, data_keuangan):
    bersihkan_layar()
    tampil_header("CATATAN BARU")

    tampil_pesan("Belum ada catatan keuangan.", "INFO")
    print()

    # menu pilihan
    table = PrettyTable()
    table.field_names = ["NO", "PILIHAN"]
    table.add_row(["1", "Buat catatan baru"])
    table.add_row(["2", "Logout"])
    table.align["NO"] = "c"
    table.align["PILIHAN"] = "l"
    print(table)
    print()

    pilihan = input_dengan_prompt("Pilih menu (1-2): ")

    if pilihan == "1":
        while True:

```

```

        print()
        saldo_awal = input_dengan_prompt("Masukkan nominal awal tabungan:
Rp ")

        valid, angka = validasi_angka(saldo_awal)

        if not valid:
            tampil_pesan("Input harus berupa angka positif!", "ERROR")
            continue

        data_keuangan[current_user] = {
            'saldo': angka,
            'riwayat': [f"Saldo Awal: Rp {angka:,}"]
        }

        tampil_pesan("Catatan berhasil dibuat!", "SUKSES")
        pause()
        return "continue"

    elif pilihan == "2":
        tampil_pesan("Logout...", "INFO")
        return "logout"
    else:
        tampil_pesan("Pilihan tidak valid!", "ERROR")
        pause()
        return "continue"

def cek_saldo(current_user, data_keuangan):
    bersihkan_layar()
    tampil_header("CEK SISA TABUNGAN")

    try:
        saldo = data_keuangan[current_user]['saldo']

        # tampil saldo
        table_saldo = PrettyTable()
        table_saldo.field_names = ["INFORMASI", "NILAI"]
        table_saldo.add_row(["Saldo Saat Ini", f"Rp {saldo:,}"])
        table_saldo.align["INFORMASI"] = "l"
        table_saldo.align["NILAI"] = "r"
        print(table_saldo)
        print()

```

```

# tampil riwayat transaksi
print("RIWAYAT TRANSAKSI (5 Terakhir):")
table_riwayat = PrettyTable()
table_riwayat.field_names = ["NO", "TRANSAKSI"]
table_riwayat.align["NO"] = "c"
table_riwayat.align["TRANSAKSI"] = "l"

riwayat = data_keuangan[current_user]['riwayat']
jumlah = len(riwayat)
mulai = max(0, jumlah - 5)

for i in range(mulai, jumlah):
    table_riwayat.add_row([i + 1 - mulai, riwayat[i]])

print(table_riwayat)
pause()

except Exception as e:
    tampil_pesan(f"Terjadi kesalahan: {e}", "ERROR")
    pause()

def catat_pemasukan(current_user, data_keuangan):
    bersihkan_layar()
    tampil_header("CATAT PEMASUKAN")
    print()

    while True:
        nominal = input_dengan_prompt("Masukkan nominal pemasukan: Rp ")

        valid, angka = validasi_angka(nominal)

        if not valid:
            tampil_pesan("Input harus berupa angka positif!", "ERROR")
            print()
            continue

        keterangan = input_dengan_prompt("Keterangan (opsional): ")

        if keterangan == "":
            keterangan = "Pemasukan"

```

```

        data_keuangan[current_user]['saldo'] += angka
        data_keuangan[current_user]['riwayat'].append(
            f"+ Pemasukan: Rp {angka:,} ({keterangan})"
        )

        # tampilkan hasil
        table = PrettyTable()
        table.field_names = ["INFORMASI", "NILAI"]
        table.add_row(["Pemasukan", f"Rp {angka:,}"])
        table.add_row(["Keterangan", keterangan])
        table.add_row(["Saldo Baru", f"Rp
{data_keuangan[current_user]['saldo']:,}")]
        table.align["INFORMASI"] = "l"
        table.align["NILAI"] = "r"
        print()
        print(table)

        tampil_pesan("Pemasukan berhasil dicatat!", "SUKSES")
        pause()
        break

def catat_pengeluaran(current_user, data_keuangan):
    bersihkan_layar()
    tampil_header("CATAT PENGELOUARAN")

    try:
        saldo_sekarang = data_keuangan[current_user]['saldo']

        # tampil saldo skrg
        table_saldo = PrettyTable()
        table_saldo.field_names = ["INFORMASI", "NILAI"]
        table_saldo.add_row(["Saldo Saat Ini", f"Rp {saldo_sekarang:,}"])
        table_saldo.align["INFORMASI"] = "l"
        table_saldo.align["NILAI"] = "r"
        print(table_saldo)
        print()

        while True:
            nominal = input_dengan_prompt("Masukkan nominal pengeluaran: Rp ")
            valid, angka = validasi_angka(nominal)

```

```

        if not valid:
            tampil_pesan("Input harus berupa angka positif!", "ERROR")
            print()
            continue

        if angka > saldo_sekarang:
            tampil_pesan("PERINGATAN: Pengeluaran melebihi saldo!",
"WARNING")
            konfirmasi = input_dengan_prompt("Lanjutkan? (y/n): ")
            if konfirmasi.lower() != "y":
                tampil_pesan("Pengeluaran dibatalkan.", "INFO")
                pause()
                return

        keterangan = input_dengan_prompt("Keterangan (opsional): ")

        if keterangan == "":
            keterangan = "Pengeluaran"

        data_keuangan[current_user]['saldo'] -= angka
        data_keuangan[current_user]['riwayat'].append(
            f"- Pengeluaran: Rp {angka:,} ({keterangan})"
        )

        # tampil hasil
        table = PrettyTable()
        table.field_names = ["INFORMASI", "NILAI"]
        table.add_row(["Pengeluaran", f"Rp {angka:,}"])
        table.add_row(["Keterangan", keterangan])
        table.add_row(["Saldo Baru", f"Rp
{data_keuangan[current_user]['saldo']:,}")]
        table.align["INFORMASI"] = "l"
        table.align["NILAI"] = "r"
        print()
        print(table)

        tampil_pesan("Pengeluaran berhasil dicatat!", "SUKSES")
        pause()
        break

    except Exception as e:
        tampil_pesan(f"Terjadi kesalahan: {e}", "ERROR")

```

```

pause()

def hapus_data(current_user, data_keuangan):
    bersihkan_layar()
    tampil_header("HAPUS DATA TABUNGAN")

    tampil_pesan("PERINGATAN: Data yang dihapus tidak dapat dikembalikan!", "WARNING")
    print()

    try:
        # tampil informasi data
        table_info = PrettyTable()
        table_info.field_names = ["INFORMASI", "NILAI"]
        table_info.add_row(["Saldo Saat Ini", f"Rp {data_keuangan[current_user]['saldo']}"])
        table_info.add_row(["Total Transaksi", f"{len(data_keuangan[current_user]['riwayat'])} transaksi"])
        table_info.align["INFORMASI"] = "l"
        table_info.align["NILAI"] = "r"
        print(table_info)
        print()

        # menu konfirmasi
        table_menu = PrettyTable()
        table_menu.field_names = ["NO", "PILIHAN"]
        table_menu.add_row(["0", "Ya, Hapus Data"])
        table_menu.add_row(["1", "Tidak, Batalkan"])
        table_menu.align["NO"] = "c"
        table_menu.align["PILIHAN"] = "l"
        print(table_menu)
        print()

        pilihan = input_dengan_prompt("Pilih menu (0-1): ")

        if pilihan == "0":
            data_keuangan[current_user] = {}
            tampil_pesan("Data tabungan berhasil dihapus!", "SUKSES")
            pause()
            return True
        elif pilihan == "1":
            tampil_pesan("Penghapusan dibatalkan.", "INFO")

```

```

        pause()
        return False
    else:
        tampil_pesan("Pilihan tidak valid!", "ERROR")
        pause()
        return False

except Exception as e:
    tampil_pesan(f"Terjadi kesalahan: {e}", "ERROR")
    pause()
    return False

```

### 3.5 Menu

```

from utils import bersihkan_layar, tampil_header, tampil_pesan,
input_dengan_prompt, pause
from transaksi import cek_saldo, catat_pemasukan, catat_pengeluaran,
hapus_data
from prettytable import PrettyTable


def menu_utama(current_user, data_keuangan):
    while True:
        bersihkan_layar()
        tampil_header("PROGRAM PENCATAT KEUANGAN")

        # info user
        table_user = PrettyTable()
        table_user.field_names = ["USER AKTIF"]
        table_user.add_row([current_user])
        table_user.align = "c"
        print(table_user)
        print()

        # menu utama
        print("MENU UTAMA:")
        table_menu = PrettyTable()
        table_menu.field_names = ["NO", "MENU"]
        table_menu.add_row(["1", "Cek Sisa Tabungan"])
        table_menu.add_row(["2", "Catat Pemasukan"])
        table_menu.add_row(["3", "Catat Pengeluaran"])

```

```
table_menu.add_row([ "4", "Hapus Data Tabungan"])
table_menu.add_row([ "5", "Logout"])
table_menu.align["NO"] = "c"
table_menu.align["MENU"] = "l"
print(table_menu)
print()

pilihan = input_dengan_prompt("Pilih menu (1-5): ")

if pilihan == "1":
    cek_saldo(current_user, data_keuangan)
elif pilihan == "2":
    catat_pemasukan(current_user, data_keuangan)
elif pilihan == "3":
    catat_pengeluaran(current_user, data_keuangan)
elif pilihan == "4":
    if hapus_data(current_user, data_keuangan):
        break
elif pilihan == "5":
    tampil_pesan("Logout berhasil!", "SUKSES")
    pause()
    break
else:
    tampil_pesan("Pilihan tidak valid! Masukkan angka 1-5.", "ERROR")
    pause()
```

## 4. Hasil Output

### 1. Registrasi

```
+-----+
| REGISTER AKUN BARU |
+-----+
+-----+
>> Masukkan username baru: yaralyarrr
>> Masukkan password: 2509106043
>> Konfirmasi password: 2509106043
+-----+
| SUKSES |
+-----+
| Akun yaralyarrr berhasil dibuat! |
+-----+
Tekan Enter untuk melanjutkan...■
```

Gambar 4.1 Tampilan Registrasi Akun Baru

### 2. Login User

```
+-----+
| LOGIN |
+-----+
+-----+
+-----+
| OPSI |
+-----+
| Belum punya akun? Ketik 'register' pada username |
| Ingin keluar? Ketik 'exit' pada username |
+-----+
>> Username: yaralyarrr
>> Password: 2509106043
+-----+
| SUKSES |
+-----+
| Login berhasil! Selamat datang, yaralyarrr! |
+-----+
Tekan Enter untuk melanjutkan...■
```

Gambar 4.2 Tampilan Login User

### 3. Main Program

```
+-----+
| CATATAN BARU |
+-----+
+-----+
+-----+
| INFO |
+-----+
| Belum ada catatan keuangan. |
+-----+  
  
+-----+
| NO | PILIHAN |
+-----+
| 1 | Buat catatan baru |
| 2 | Logout |
+-----+  
  
>> Pilih menu (1-2): 1  
  
>> Masukkan nominal awal tabungan: Rp 55555555
+-----+
| SUKSES |
+-----+
| Catatan berhasil dibuat! |
+-----+  
  
Tekan Enter untuk melanjutkan...█
```

Gambar 4.3.1 Contoh Output (1)

```
+-----+
| PROGRAM PENCATAT KEUANGAN |
+-----+
+-----+
| USER AKTIF |
+-----+
| yaralyarrr |
+-----+  
  
MENU UTAMA:  
+-----+
| NO | MENU
+-----+
| 1 | Cek Sisa Tabungan
| 2 | Catat Pemasukan
| 3 | Catat Pengeluaran
| 4 | Hapus Data Tabungan
| 5 | Logout
+-----+  
  
>> Pilih menu (1-5): █
```

Gambar 4.3.2 Contoh Output (2)

```
+-----+
| CEK SISA TABUNGAN |
+-----+
+-----+
| INFORMASI | NILAI |
+-----+
| Saldo Saat Ini | Rp 55,555,555 |
+-----+-----+  
  
RIWAYAT TRANSAKSI (5 Terakhir):
+-----+
| NO | TRANSAKSI |
+-----+
| 1 | Saldo Awal: Rp 55,555,555 |
+-----+  
  
Tekan Enter untuk melanjutkan... █
```

Gambar 4.3.3 Contoh Output (3)

```
+-----+
| CATAT PEMASUKAN |
+-----+
>> Masukkan nominal pemasukan: Rp 505050
>> Keterangan (opsional):

+-----+-----+
| INFORMASI |      NILAI |
+-----+-----+
| Pemasukan |    Rp 505,050 |
| Keterangan |      Pemasukan |
| Saldo Baru | Rp 56,060,605 |
+-----+-----+
+-----+-----+
| SUKSES |
+-----+
| Pemasukan berhasil dicatat! |
+-----+-----+
Tekan Enter untuk melanjutkan...|
```

Gambar 4.3.4 Contoh Output (4)

```
+-----+
| CATAT PENGELOUARAN |
+-----+
+-----+-----+
| INFORMASI | NILAI |
+-----+-----+
| Saldo Saat Ini | Rp 56,060,605 |
+-----+-----+  
  
>> Masukkan nominal pengeluaran: Rp 404040  
>> Keterangan (opsional):  
  
+-----+-----+
| INFORMASI | NILAI |
+-----+-----+
| Pengeluaran | Rp 404,040 |
| Keterangan | Pengeluaran |
| Saldo Baru | Rp 55,656,565 |
+-----+-----+
+-----+
| SUKSES |
+-----+
| Pengeluaran berhasil dicatat! |
+-----+  
  
Tekan Enter untuk melanjutkan... █
```

Gambar 4.3.5 Contoh Output (5)

```
+-----+
| HAPUS DATA TABUNGAN |
+-----+
+-----+
| WARNING |
+-----+
| PERINGATAN: Data yang dihapus tidak dapat dikembalikan! |
+-----+  
  
+-----+-----+
| INFORMASI | NILAI |
+-----+-----+
| Saldo Saat Ini | Rp 55,656,565 |
| Total Transaksi | 3 transaksi |
+-----+-----+  
  
+-----+
| NO | PILIHAN |
+-----+
| 0 | Ya, Hapus Data |
| 1 | Tidak, Batalkan |
+-----+  
  
>> Pilih menu (0-1): 0  
+-----+
| SUKSES |
+-----+
| Data tabungan berhasil dihapus! |
+-----+  
  
Tekan Enter untuk melanjutkan... █
```

Gambar 4.3.6 Contoh Output (6)

## 5. Langkah-langkah GIT

### 5.2 GIT Add

```
PS C:\praktikum-apd> git add .
```

Gambar 5.1 Command Git Add

### 5.3 GIT Commit

```
PS C:\praktikum-apd> git commit -m "second commit"
[main a354c18] second commit
14 files changed, 699 insertions(+), 8 deletions(-)
create mode 100644 A2-2025/pertemuan-8/code.py
create mode 100644 A2-2025/pertemuan-8/pertemuan-8.py
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/__pycache__/autentikasi.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/__pycache__/auth.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/__pycache__/menu.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/__pycache__/transaksi.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/__pycache__/utils.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/autentikasi.py
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/main.py
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/menu.py
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/transaksi.py
create mode 100644 post-test/post-test-apd-8/post-test-apd-8/utils.py
```

Gambar 5.2 Command Git Commit

### 5.5 GIT Push

```
PS C:\praktikum-apd> git push
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 16 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (24/24), 15.73 KiB | 2.25 MiB/s, done.
Total 24 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
To https://github.com/yaralyar07/praktikum-apd.git
  ea53d59..a354c18  main -> main
PS C:\praktikum-apd>
```

Gambar 5.3 Command Git Push