

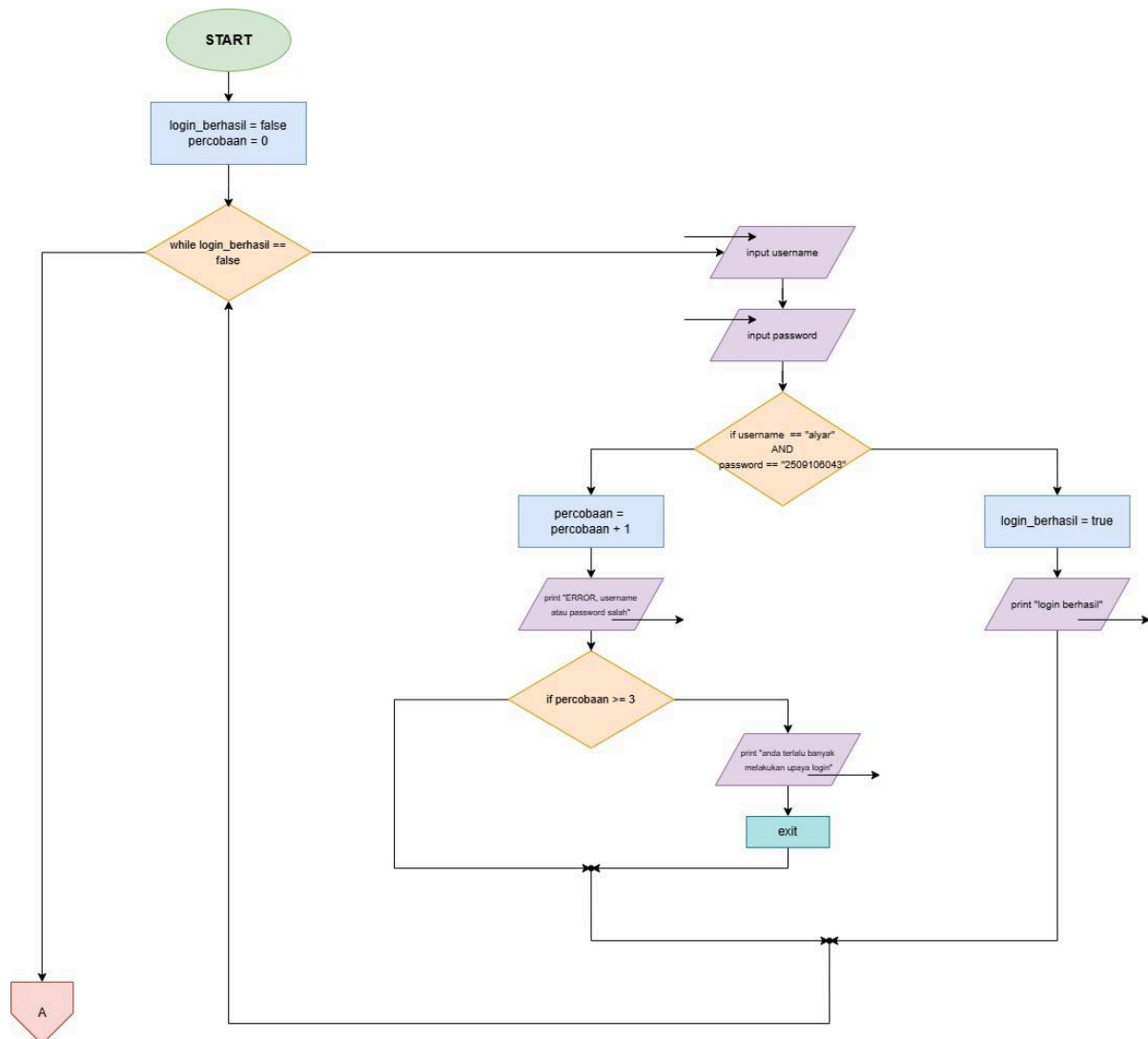
LAPORAN PRAKTIKUM
POSTTEST(6)
ALGORITMA PEMROGRAMAN DASAR



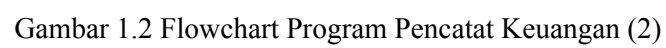
Disusun oleh:
Nama (2509106043)
Kelas (A2'25)

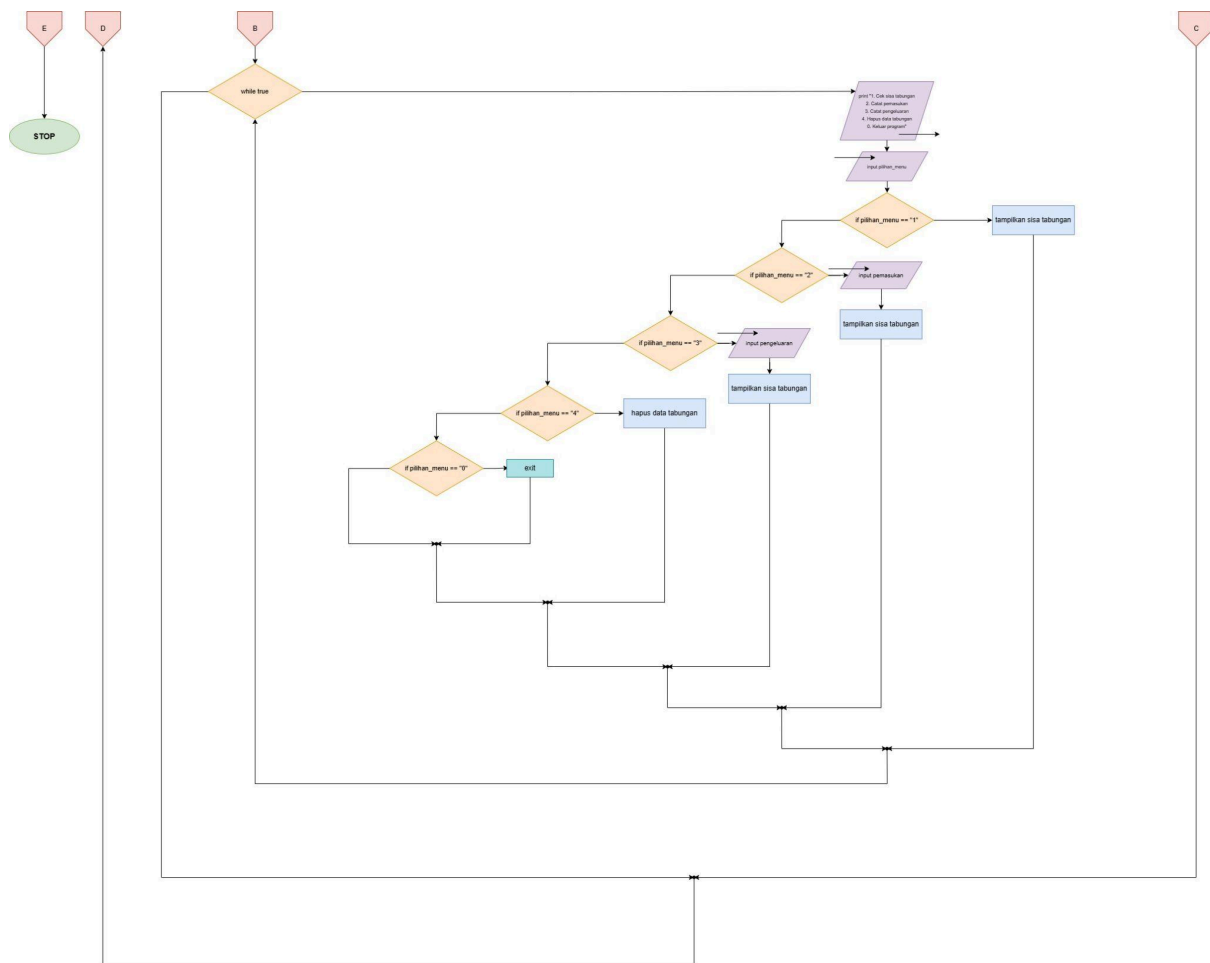
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart Program Pemcatat Keuangan (1)





Gambar 1.2 Flowchart Program Pencatat Keuangan (2)

Program dimulai dari START dengan melakukan inisialisasi dua variabel yaitu login_berhasil = false dan percobaan = 0. Setelah inisialisasi, program memasuki loop login dengan kondisi while login_berhasil == false yang akan terus berulang sampai user berhasil login. Di dalam loop login, program meminta user untuk menginput username dan password. Jika username yang diinput adalah "alyar" dan password adalah "2509106043", maka program akan mengubah login_berhasil = true, menampilkan pesan "login berhasil!" dan melanjutkan ke tahap berikutnya di connector A. Namun jika username atau password salah, program akan menambah variabel percobaan = percobaan + 1 dan menampilkan pesan error. Program kemudian mengecek apakah percobaan >= 3, jika sudah mencapai 3 kali salah maka program akan menampilkan pesan "anda terlalu banyak melakukan upaya login" dan keluar melalui exit. Jika percobaan masih kurang dari 3, program akan kembali ke loop login untuk memberi kesempatan mencoba lagi.

Setelah berhasil login dan melewati connector A, program melakukan inisialisasi `data_catatan = {}` sebagai dictionary kosong kemudian masuk ke loop utama dengan kondisi `while true`. Di dalam loop utama, program pertama kali mengecek apakah `data_catatan = {}` masih kosong atau tidak. Jika masih kosong, program akan menampilkan output "apakah anda ingin membuat catatan baru: 0. Ya 1. Tidak (tutup program)" dan meminta user menginput pilihan melalui input `pilihan_buat`. Program kemudian mengecek kondisi `if pilihan_buat == "0"`. Jika user memilih 0, program akan meminta user untuk menginput saldo awal melalui input `saldo_awal` dan menyimpannya ke dalam `data_catatan` di connector C. Jika user memilih 1 pada pilihan buat catatan, program akan keluar melalui `exit`. Jika user menginput selain 0 atau 1, program akan menampilkan pesan "ERROR! pilihan tidak valid" dan kembali ke loop untuk meminta input ulang. Setelah saldo awal berhasil disimpan, program kembali ke loop utama melalui connector C yang mengarah ke B.

Dari connector B, karena `data_catatan` sudah tidak kosong lagi, program akan masuk ke bagian menu utama yang ditandai dengan connector D. Di menu utama, program memasuki loop `while true` dan menampilkan output menu dengan 5 pilihan: "1. Cek sisa tabungan, 2. Catat pemasukan, 3. Catat pengeluaran, 4. Hapus data tabungan, 0. Keluar program". Program kemudian meminta user menginput pilihan menu melalui input `pilihan_menu` dan melakukan pengecekan secara berurutan. Jika `pilihan_menu == "1"`, program akan menampilkan sisa tabungan dan kembali ke loop menu. Jika `pilihan_menu == "2"`, program akan meminta input pemasukan, menampilkan sisa tabungan yang sudah diupdate, dan kembali ke loop menu. Jika `pilihan_menu == "3"`, program akan meminta input pengeluaran, menampilkan sisa tabungan baru, dan kembali ke loop menu. Jika `pilihan_menu == "4"`, program akan menjalankan proses hapus data tabungan yang mengosongkan `data_catatan` dan kembali ke loop utama melalui connector D ke B untuk mengecek kondisi data kosong lagi. Jika `pilihan_menu == "0"`, program akan keluar dan menuju ke connector E yang langsung berakhir di STOP.

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 data list 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

```
while True:
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=" * 50)
    print("    PROGRAM PENCATAT KEUANGAN")
    print("=" * 50)
    print()
    print("MENU UTAMA:")
    print()
    print("1. Cek Sisa Tabungan")
    print("2. Catat Pemasukan")
    print("3. Catat Pengeluaran")
    print("4. Hapus Data Tabungan")
    print("0. Keluar Program")
    print()

    pilihan_menu = input("Ketik angka untuk memilih: ")

    # 1
    if pilihan_menu == "1":
        os.system('cls' if os.name == 'nt' else 'clear')
        print("=" * 50)
        print("    CEK SISA TABUNGAN")
        print("=" * 50)
        print()

        print(f"Saldo saat ini: Rp {data_catatan['saldo']}")
        print()
        print("Riwayat Transaksi:")
        print("-" * 50)

        jumlah_riwayat = len(data_catatan['riwayat'])
        mulai = 0
        if jumlah_riwayat > 5:
            mulai = jumlah_riwayat - 5

        for i in range(mulai, jumlah_riwayat):
            print(data_catatan['riwayat'][i])

        print()
        input("Tekan Enter untuk kembali...")
```

```

# 2
elif pilihan_menu == "2":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=" * 50)
    print("      CATAT PEMASUKAN")
    print("=" * 50)
    print()

    input_valid = False
    while input_valid == False:
        nominal = input("Masukkan nominal pemasukan: Rp ")

        if nominal == "":
            print("ERROR! Input tidak boleh kosong!")
            input("Tekan Enter untuk melanjutkan...")
            os.system('cls' if os.name == 'nt' else 'clear')
            print("=" * 50)
            print("      CATAT PEMASUKAN")
            print("=" * 50)
            print()
            continue

        cek_angka = True
        for angka in nominal:
            if angka not in "0123456789":
                cek_angka = False
                break

        if cek_angka == False:
            print("ERROR! Input harus berupa angka!")
            input("Tekan Enter untuk melanjutkan...")
            os.system('cls' if os.name == 'nt' else 'clear')
            print("=" * 50)
            print("      CATAT PEMASUKAN")
            print("=" * 50)
            print()
            continue

        nominal = int(nominal)
        input_valid = True

```



```

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

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

        data_catatan['saldo'] = data_catatan['saldo'] + nominal
        data_catatan['riwayat'].append(f"+ Pemasukan: Rp {nominal}
({keterangan})")

        print()
        print("Pemasukan berhasil dicatat!")
        print(f"Saldo baru: Rp {data_catatan['saldo']}")
        print()
        input("Tekan Enter untuk kembali...")

# 3
elif pilihan_menu == "3":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=" * 50)
    print("      CATAT PENGELUARAN")
    print("=" * 50)
    print()

    saldo_sekarang = data_catatan['saldo']

    print(f"Saldo saat ini: Rp {saldo_sekarang}")
    print()

    input_valid = False
    batal = False

    while input_valid == False and batal == False:
        nominal = input("Masukkan nominal pengeluaran: Rp ")

        if nominal == "":
            print("ERROR! Input tidak boleh kosong!")
            input("Tekan Enter untuk melanjutkan...")
            os.system('cls' if os.name == 'nt' else 'clear')
            print("=" * 50)
            print("      CATAT PENGELUARAN")
            print("=" * 50)
            print()

```

```

        print(f"Saldo saat ini: Rp {saldo_sekarang}")
        print()
        continue

cek_angka = True
for angka in nominal:
    if angka not in "0123456789":
        cek_angka = False
        break

if cek_angka == False:
    print("ERROR! Input harus berupa angka!")
    input("Tekan Enter untuk melanjutkan...")
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=" * 50)
    print("      CATAT PENGELUARAN")
    print("=" * 50)
    print()
    print(f"Saldo saat ini: Rp {saldo_sekarang}")
    print()
    continue

nominal = int(nominal)

if nominal > saldo_sekarang:
    print()
    print("PERINGATAN: Pengeluaran melebihi saldo!")
    konfirmasi = input("Lanjutkan? (y/n): ")
    if konfirmasi != "y":
        print("Pengeluaran dibatalkan.")
        input("Tekan Enter untuk kembali...")
        batal = True
    else:
        input_valid = True

if batal == False:
    keterangan = input("Keterangan (opsional): ")

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

data_catatan['saldo'] = data_catatan['saldo'] - nominal

```

```

        data_catatan['riwayat'].append(f"- Pengeluaran: Rp
{nominal} ({keterangan})")

    print()
    print("Pengeluaran berhasil dicatat!")
    print(f"Saldo baru: Rp {data_catatan['saldo']}")
    print()
    input("Tekan Enter untuk kembali...")

# 4
elif pilihan_menu == "4":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=" * 50)
    print("      HAPUS DATA TABUNGAN")
    print("=" * 50)
    print()
    print("PERINGATAN: Data yang dihapus tidak dapat
dikembalikan!")
    print()

    print(f"Saldo saat ini: Rp {data_catatan['saldo']}")
    print(f"Total transaksi: {len(data_catatan['riwayat'])}
transaksi")

    print()
    print("Apakah Anda yakin ingin menghapus data ini?")
    print("0. Ya, Hapus")
    print("1. Tidak, Batalkan")
    print()

    pilihan_hapus = input("Ketik angka untuk memilih: ")

    if pilihan_hapus == "0":
        data_catatan.clear()

        print()
        print("Data tabungan berhasil dihapus!")
        input("Tekan Enter untuk melanjutkan...")
        break
    elif pilihan_hapus == "1":
        print()
        print("Penghapusan dibatalkan.")
        input("Tekan Enter untuk kembali...")

```

```
        else:
            print()
            print("ERROR! Pilihan tidak valid! Masukkan angka 0 atau
1.")

            input("Tekan Enter untuk melanjutkan...")

# 0
elif pilihan_menu == "0":
    print()
    print("Terima kasih!")
    exit()

else:
    print()
    print("ERROR! Pilihan tidak valid! Masukkan angka 0-4.")
    input("Tekan Enter untuk melanjutkan...")
```

4. Hasil Output

1. Login user

```
=====
                        SELAMAT DATANG
=====

Username: alyar
Password: 2509106043

Login berhasil!
Tekan Enter untuk melanjutkan...█
```

Gambar 4.1.1 Tampilan Login User (1)

```
=====
                        SELAMAT DATANG
=====

Username: alyar
Password: 44444

ERROR! Username atau password salah.
Tekan Enter untuk mencoba lagi...█
```

Gambar 4.1.2 Tampilan Login User (2)

2. Program utama

```
=====
CATATAN BARU
=====

Belum ada catatan keuangan.

Apakah Anda ingin membuat catatan baru?
0. Ya
1. Tidak (Tutup Program)

Ketik angka untuk memilih: 0

Masukkan nominal awal tabungan: Rp 500000

Catatan berhasil dibuat!
Tekan Enter untuk melanjutkan...█
```

Gambar 4.2.1 Contoh Output (1)

```
=====
PROGRAM PENCATAT KEUANGAN
=====

MENU UTAMA:

1. Cek Sisa Tabungan
2. Catat Pemasukan
3. Catat Pengeluaran
4. Hapus Data Tabungan
0. Keluar Program

Ketik angka untuk memilih: █
```

Gambar 4.2.2 Contoh Output (2)

```
=====
CEK SISA TABUNGAN
=====

Saldo saat ini: Rp 500000

Riwayat Transaksi:
-----
Saldo Awal: Rp 500000

Tekan Enter untuk kembali...█
```

Gambar 4.2.3 Contoh Output (3)

```
=====
CATAT PEMASUKAN
=====

Masukkan nominal pemasukan: Rp 50000
Keterangan (opsional):

Pemasukan berhasil dicatat!
Saldo baru: Rp 550000

Tekan Enter untuk kembali...█
```

Gambar 4.2.4 Contoh Output (4)

```
=====
CATAT PENGELUARAN
=====

Saldo saat ini: Rp 550000

Masukkan nominal pengeluaran: Rp 50000
Keterangan (opsional):

Pengeluaran berhasil dicatat!
Saldo baru: Rp 500000

Tekan Enter untuk kembali...█
```

Gambar 4.2.5 Contoh Output (5)

```
=====
CATAT PENGELUARAN
=====

Saldo saat ini: Rp 500000

Masukkan nominal pengeluaran: Rp 600000

PERINGATAN: Pengeluaran melebihi saldo!
Lanjutkan? (y/n): █
```

Gambar 4.2.6 Contoh Output (6)

```
=====
HAPUS DATA TABUNGAN
=====

PERINGATAN: Data yang dihapus tidak dapat dikembalikan!

Saldo saat ini: Rp 500000
Total transaksi: 3 transaksi

Apakah Anda yakin ingin menghapus data ini?
0. Ya, Hapus
1. Tidak, Batalkan

Ketik angka untuk memilih: 0

Data tabungan berhasil dihapus!
Tekan Enter untuk melanjutkan...█
```

Gambar 4.2.7 Contoh Output (7)

5. Langkah-langkah GIT

5.2 GIT Add

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

Gambar 5.1 Command Git Add

5.3 GIT Commit

```
PS C:\praktikum-apd> git commit -m "second commit"  
[main 571be4b] second commit  
3 files changed, 423 insertions(+), 4 deletions(-)  
create mode 100644 A2-2025/praktikum-6/praktikum-6.py  
create mode 100644 post-test/post-test-apd-6/2509106043-ALYA-KAMILA-FAUZIYAH-PT-6.py  
PS C:\praktikum-apd>
```

Gambar 5.2 Command Git Commit

5.5 GIT Push

```
PS C:\praktikum-apd> git push  
Enumerating objects: 15, done.  
Counting objects: 100% (15/15), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (8/8), done.  
Writing objects: 100% (10/10), 3.58 KiB | 611.00 KiB/s, done.  
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.  
To https://github.com/yaralyar07/praktikum-apd.git  
3b8f132..571be4b main -> main  
PS C:\praktikum-apd>
```

Gambar 5.3 Command Git Push