

LAPORAN PRAKTIKUM
POSTTEST(7)
ALGORITMA PEMROGRAMAN DASAR

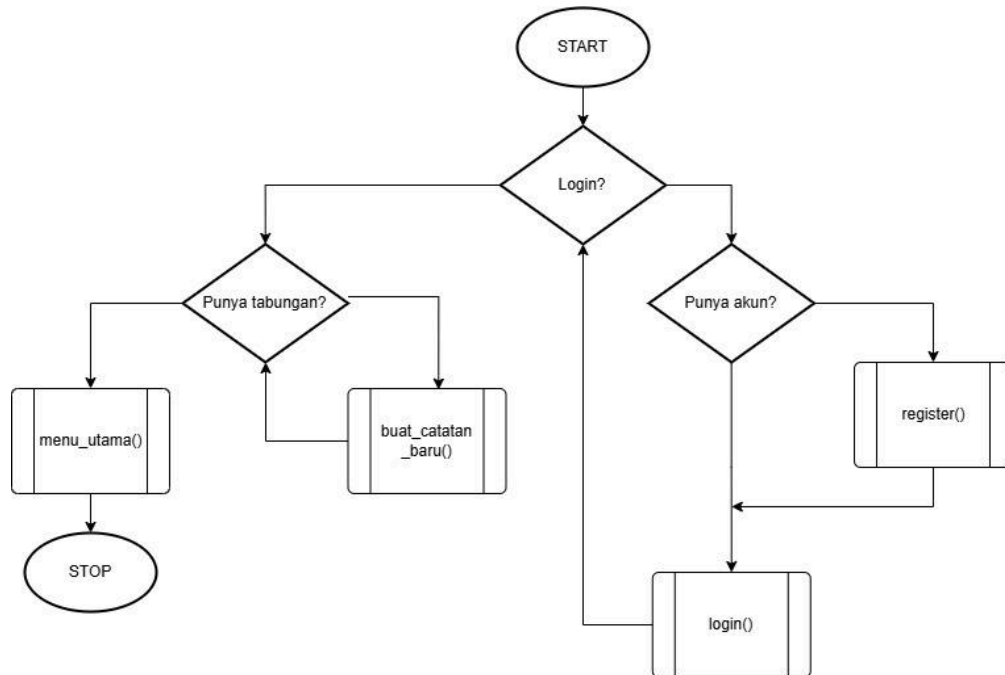


Disusun oleh:
Nama (2509106043)
Kelas (A2'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

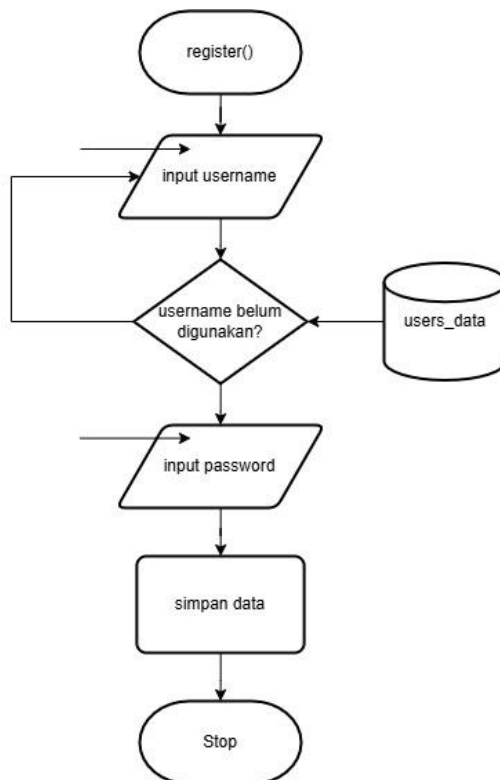
1. Flowchart

1.1 Program



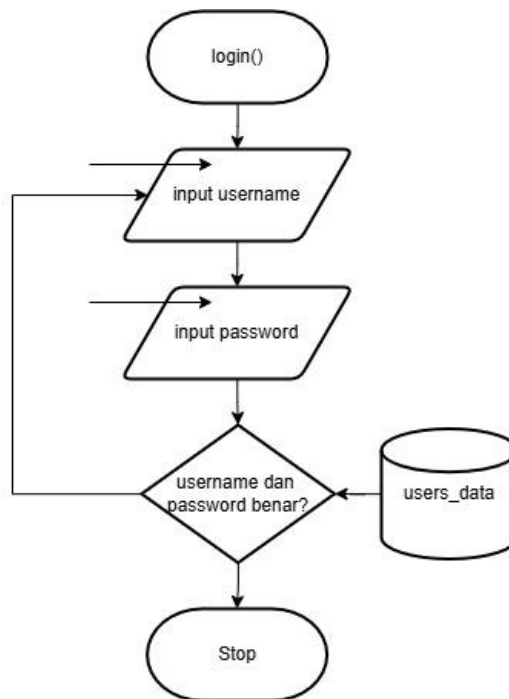
Gambar 1.1 Flowchart Program Pencatat Keuangan (1)

1.2 Register



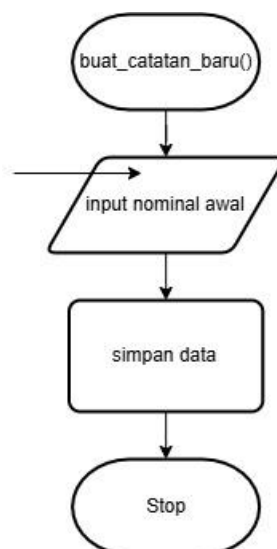
Gambar 1.2 Flowchart Program Pencatat Keuangan (2)

1.3 Login



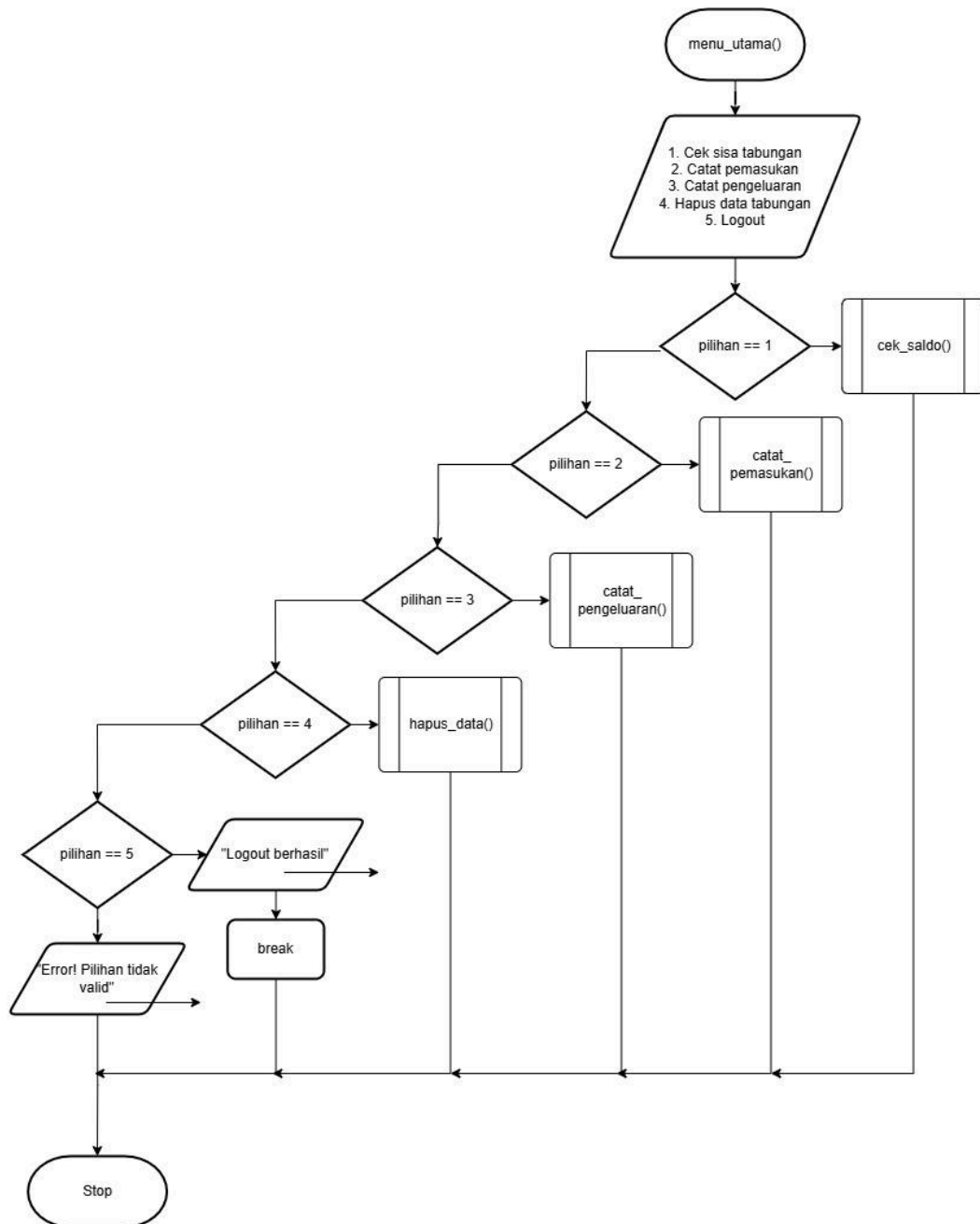
Gambar 1.3 Flowchart Program Pencatat Keuangan (3)

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

```
import os

users_data = {}
data_keuangan = {}
current_user = ""

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

def tampil_header(judul):
    print("=" * 50)
    print(f"    {judul}")
    print("=" * 50)

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

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

    username_baru = input("Masukkan username baru: ")

    if username_baru == "":
        print("ERROR! Username tidak boleh kosong!")
        input("Tekan Enter untuk kembali...")
        return False

    if username_baru in users_data:
        print("ERROR! Username sudah digunakan!")
        input("Tekan Enter untuk kembali...")
        return False
```

```

password_baru = input("Masukkan password: ")

if password_baru == "":
    print("ERROR! Password tidak boleh kosong!")
    input("Tekan Enter untuk kembali...")
    return False

konfirmasi_password = input("Konfirmasi password: ")

if password_baru != konfirmasi_password:
    print("ERROR! Password tidak cocok!")
    input("Tekan Enter untuk kembali...")
    return False

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

print()
print(f"Akun {username_baru} berhasil dibuat!")
input("Tekan Enter untuk login...")
return True

def login():
    global current_user

    percobaan = 0
    login_berhasil = False

    while login_berhasil == False:
        bersihkan_layar()
        tampil_header("LOGIN")
        print()
        print("Belum punya akun? Ketik 'register' pada username")
        print("Ingin keluar? Ketik 'exit' pada username")
        print()

        username = input("Username: ")

        if username.lower() == "register":
            if register():
                continue
            else:

```

```

        continue

    if username.lower() == "exit":
        print()
        print("Terima kasih! Program ditutup.")
        input("Tekan Enter untuk keluar...")
        exit()

    password = input("Password: ")

    try:
        if username in users_data and users_data[username] == password:
            print()
            print(f"Login berhasil! Selamat datang, {username}!")
            current_user = username
            input("Tekan Enter untuk melanjutkan...")
            login_berhasil = True
        else:
            percobaan = percobaan + 1
            print()
            print("ERROR! Username atau password salah.")

            if percobaan >= 3:
                print("Anda melakukan terlalu banyak upaya login. Program
ditutup.")

                input("Tekan Enter untuk keluar...")
                exit()
            else:
                input("Tekan Enter untuk mencoba lagi...")
    except Exception as e:
        print(f"ERROR! Terjadi kesalahan: {e}")
        input("Tekan Enter untuk mencoba lagi...")

def buat_catatan_baru():
    global data_keuangan

    bersihkan_layar()
    tampil_header("CATATAN BARU")
    print()
    print("Belum ada catatan keuangan.")
    print()
    print("Apakah Anda ingin membuat catatan baru?")

```



```

print("1. Ya")
print("2. Tidak (Logout)")
print()

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

if pilihan_buat == "1":
    saldo_valid = False

    while saldo_valid == False:
        print()
        saldo_awal = input("Masukkan nominal awal tabungan: Rp ")

        try:
            valid, angka = validasi_angka(saldo_awal)

            if not valid:
                print("ERROR! Input harus berupa angka positif!")
                input("Tekan Enter untuk melanjutkan...")
                continue

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

            print()
            print("Catatan berhasil dibuat!")
            input("Tekan Enter untuk melanjutkan...")
            saldo_valid = True

        except Exception as e:
            print(f"ERROR! {e}")
            input("Tekan Enter untuk melanjutkan...")

elif pilihan_buat == "2":
    print()
    print("Logout...")
    return "logout"
else:
    print()
    print("ERROR! Pilihan tidak valid!")

```

```

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

    return "continue"

def cek_saldo():
    bersihkan_layar()
    tampil_header("CEK SISA TABUNGAN")
    print()

    try:
        saldo = data_keuangan[current_user]['saldo']
        print(f"Saldo saat ini: Rp {saldo}")
        print()
        print("Riwayat Transaksi (5 terakhir):")
        print("-" * 50)

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

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

        print()
        input("Tekan Enter untuk kembali...")
    except Exception as e:
        print(f"ERROR! Terjadi kesalahan: {e}")
        input("Tekan Enter untuk kembali...")

def catat_pemasukan():
    bersihkan_layar()
    tampil_header("CATAT PEMASUKAN")
    print()

    input_valid = False

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

        try:
            valid, angka = validasi_angka(nominal)

```

```

        if not valid:
            print("ERROR! Input harus berupa angka positif!")
            input("Tekan Enter untuk melanjutkan...")
            bersihkan_layar()
            tampil_header("CATAT PEMASUKAN")
            print()
            continue

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

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

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

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

except Exception as e:
    print(f"ERROR! {e}")
    input("Tekan Enter untuk melanjutkan...")
    bersihkan_layar()
    tampil_header("CATAT PEMASUKAN")
    print()

def catat_pengeluaran():
    bersihkan_layar()
    tampil_header("CATAT PENGELUARAN")
    print()

    try:
        saldo_sekarang = data_keuangan[current_user]['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 ")

    valid, angka = validasi_angka(nominal)

    if not valid:
        print("ERROR! Input harus berupa angka positif!")
        input("Tekan Enter untuk melanjutkan...")
        bersihkan_layar()
        tampil_header("CATAT PENGELUARAN")
        print()
        print(f"Saldo saat ini: Rp {saldo_sekarang}")
        print()
        continue

    if angka > 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
    else:
        input_valid = True

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

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

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

    print()

```

```

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

except Exception as e:
    print(f"ERROR! {e}")
    input("Tekan Enter untuk kembali...")

def hapus_data():
    bersihkan_layar()
    tampil_header("HAPUS DATA TABUNGAN")
    print()
    print("PERINGATAN: Data yang dihapus tidak dapat dikembalikan!")
    print()

    try:
        print(f"Saldo saat ini: Rp {data_keuangan[current_user]['saldo']}")
        print(f"Total transaksi: {len(data_keuangan[current_user]['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_keuangan[current_user] = {}

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

```

```

        print("ERROR! Pilihan tidak valid!")
        input("Tekan Enter untuk melanjutkan...")
        return False
except Exception as e:
    print(f"ERROR! {e}")
    input("Tekan Enter untuk kembali...")
    return False

def menu_utama():
    while True:
        bersihkan_layar()
        tampil_header("PROGRAM PENCATAT KEUANGAN")
        print()
        print(f"User: {current_user}")
        print()
        print("MENU UTAMA:")
        print()
        print("1. Cek Sisa Tabungan")
        print("2. Catat Pemasukan")
        print("3. Catat Pengeluaran")
        print("4. Hapus Data Tabungan")
        print("5. Logout")
        print()

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

        if pilihan_menu == "1":
            cek_saldo()
        elif pilihan_menu == "2":
            catat_pemasukan()
        elif pilihan_menu == "3":
            catat_pengeluaran()
        elif pilihan_menu == "4":
            if hapus_data():
                break
        elif pilihan_menu == "5":
            print()
            print("Logout berhasil!")
            input("Tekan Enter...")
            break
        else:
            print()

```

```
        print("ERROR! Pilihan tidak valid! Masukkan angka 1-5.")
        input("Tekan Enter untuk melanjutkan...")

while True:
    login()

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

4. Hasil Output

1. Registrasi

```
=====
REGISTER AKUN BARU
=====

Masukkan username baru: yaralyar
Masukkan password: Choisoobin12~
Konfirmasi password: Choisoobin12~

Akun yaralyar berhasil dibuat!
Tekan Enter untuk login...█
```

Gambar 4.1 Tampilan Registrasi Akun Baru

2. Login User

```
=====
LOGIN
=====

Belum punya akun? Ketik 'register' pada username
Ingin keluar? Ketik 'exit' pada username

Username: yaralyar
Password: Choisoobin12~

Login berhasil! Selamat datang, yaralyar!
Tekan Enter untuk melanjutkan...█
```

Gambar 4.2 Tampilan Login User

3. Main Program


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

Belum ada catatan keuangan.

Apakah Anda ingin membuat catatan baru?
1. Ya
2. Tidak (Logout)

Ketik angka untuk memilih: █
```

Gambar 4.3.1 Contoh Output (1)

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

User: yaralyar

MENU UTAMA:

1. Cek Sisa Tabungan
2. Catat Pemasukan
3. Catat Pengeluaran
4. Hapus Data Tabungan
5. Logout

Ketik angka untuk memilih: █
```

Gambar 4.3.2 Contoh Output (2)

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

Saldo saat ini: Rp 500000

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

Tekan Enter untuk kembali...|
```

Gambar 4.3.3 Contoh Output (3)

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

Masukkan nominal pemasukan: Rp 50000
Keterangan (opsional):

Pemasukan berhasil dicatat!
Saldo baru: Rp 550000

Tekan Enter untuk kembali...|
```

Gambar 4.3.4 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 4.3.5 Contoh Output (5)

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

Saldo saat ini: Rp 500000

Masukkan nominal pengeluaran: Rp 600000

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

Gambar 4.3.6 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 4.3.7 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 661ee4e] second commit  
3 files changed, 545 insertions(+)  
rename A2-2025/{praktikum-6/praktikum-6.py => pertemuan-6/pertemuan-6.py} (100%)  
create mode 100644 A2-2025/pertemuan-7/pertemuan-7.py  
create mode 100644 post-test/post-test-apd-7/2509106043-ALYA-KAMILA-FAUZIYAH-PT-7.py  
PS C:\praktikum-apd> █
```

Gambar 5.2 Command Git Commit

5.5 GIT Push

```
PS C:\praktikum-apd> git push  
Enumerating objects: 13, done.  
Counting objects: 100% (13/13), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (8/8), done.  
Writing objects: 100% (10/10), 5.14 KiB | 5.14 MiB/s, done.  
Total 10 (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/yaralyar07/praktikum-apd.git  
14dde8c..661ee4e main -> main  
PS C:\praktikum-apd> █
```

Gambar 5.3 Command Git Push