

**LAPORAN PRAKTIKUM**  
**POSTTEST 7**  
**ALGORITMA PEMROGRAMAN DASAR**

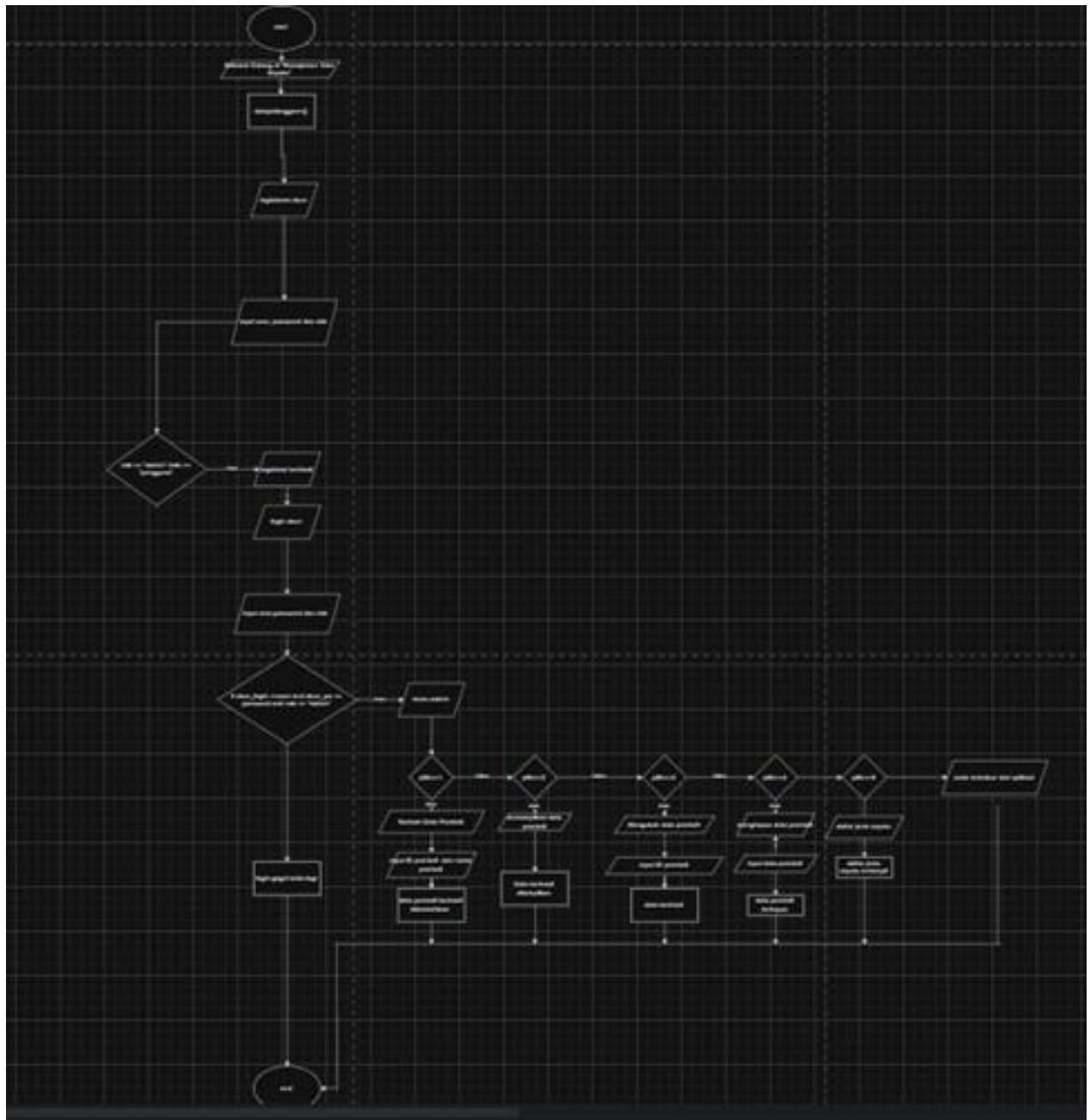


**Disusun oleh:**  
**Andhika Bima Saputra (2509106116)**  
**Kelas (C2 '25)**

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

**2025**

## 1. Flowchart



Gambar 1.1 Flowchart alur program

## 2. Deskripsi Singkat program

Program ini adalah aplikasi berbasis terminal untuk mengelola data pelanggan dan transaksi di sebuah toko sepatu

### 3. Source Code

```
import os
import time

# VARIABEL GLOBAL
DATA_PENGGUNA = {"andhika": "150407", "tamu": "12345"}
STOK_SEPATU = {
    "Sneakers Basic": 250000,
    "Running Pro": 400000,
    "Casual Street": 350000,
    "Formal Leather": 500000,
}

riwayat_transaksi = [] # Menyimpan semua transaksi
login_user = None      # Menyimpan user yang sedang login
```

Gambar 3.1 Source code Data dan variable global

```
# Prosedur tanpa return (menampilkan daftar sepatu)
def tampilkan_sepatu():
    print("\n=== DAFTAR SEPATU ===")
    for i, (nama, harga) in enumerate(STOK_SEPATU.items(), start=1):
        print(f"{i}. {nama} - Rp {harga:,}")
```

Gambar 3.2 Source Code Fitur tampilan (Prosedur tanpa parameter)

```
# Fungsi Register Akun Baru
def register():
    global DATA_PENGGUNA
    print("\n=== REGISTER AKUN BARU ===")
    username = input("Masukkan username baru: ")
    if username in DATA_PENGGUNA:
        print("Username sudah terdaftar!")
        return
    password = input("Masukkan password baru: ")
    DATA_PENGGUNA[username] = password
    print(f"Akun {username} berhasil dibuat!")
```

Gambar 3.3 Source Code Fitur Register akun baru

```

# Fungsi rekursif untuk login (mengulang jika gagal)
def login(percobaan=3):
    global login_user
    if percobaan == 0:
        print("Terlalu banyak percobaan! Keluar program...")
        return False

    print("\n=== LOGIN TOKO SEPATUKU ===")
    username = input("Masukkan Username: ")
    password = input("Masukkan Password: ")

    if username in DATA_PENGGUNA and DATA_PENGGUNA[username] == password:
        login_user = username
        print(f"\nLogin berhasil! Selamat datang, {login_user}.")
        return True
    else:
        print("Username atau password salah!")
        return login(percobaan - 1)

```

**Gambar 3.4 Source Code Fitur Login**

```

# Fungsi dengan parameter
def hitung_total(harga, jumlah):
    try:
        total = harga * jumlah
        return total
    except TypeError:
        print("Error: Input jumlah atau harga tidak valid!")
        return 0

# Fungsi dengan parameter (khusus admin untuk menambah sepatu)
def tambah_sepatu_admin(nama, harga):
    global STOK_SEPATU
    if nama in STOK_SEPATU:
        print("Sepatu sudah ada di daftar!")
    else:
        STOK_SEPATU[nama] = harga
        print(f"Sepatu {nama} berhasil ditambahkan dengan harga Rp {harga:,}.")

```

**Gambar 3.5 Source Code Fitur CRUD (fungsi dengan parameter)**

```
# Fungsi rekursif untuk login (mengulang jika gagal)
def login(percobaan=3):
    global login_user
    if percobaan == 0:
        print("Terlalu banyak percobaan! Keluar program...")
        return False
```

**Gambar 3.6 Source Code Fungsi rekursif**

```
# Fungsi tanpa parameter
def tampilkan_menu():
    print("\n=== MENU TOKO SEPATUKU ===")
    print("1. Lihat Daftar Sepatu")
    print("2. Beli Sepatu")
    print("3. Lihat Riwayat Transaksi")
    print("4. Logout")
```

**Gambar 3.7 Source Code Fitur Menu utama**

```

# PROGRAM UTAMA
def main():
    global login_user
    os.system('cls' if os.name == 'nt' else 'clear')

    if not login():
        return

    while True:
        tampilkan_menu()
        try:
            pilihan = int(input("Pilih menu (1-4): "))
        except ValueError:
            print("Input harus berupa angka!")
            continue

        if pilihan == 1:
            tampilkan_sepatu()

        elif pilihan == 2:
            tampilkan_sepatu()
            try:
                pilihan_sepatu = int(input("Pilih sepatu: "))
                nama_sepatu = input("Masukkan nama sepatu: ")
                harga = STOK[input("Masukkan harga: ")]
                jumlah = int(input("Masukkan jumlah: "))

                total = hitung_total(harga, jumlah)
                simpan_transaksi(nama_sepatu, jumlah, total)
                print(f"Total harga: Rp {total:,}")
                print("Transaksi berhasil!")

            except (ValueError, IndexError):
                print("Input tidak valid, silakan ulangi!")

        elif pilihan == 3:

```

(function) def input(  
prompt: object = "",  
/  
) -> str

Read a string from standard input. The trailing newline is stripped.  
The prompt string, if given, is printed to standard output without a trailing newline before reading input.  
If the user hits EOF (\*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.  
On \*nix systems, readline is used if available.

```

elif pilihan == 3:
    lihat_riwayat()

elif pilihan == 4:
    print(f"Terima kasih, {login_user}! Anda telah logout.\n")
    login_user = None
    break

else:
    print("Pilihan tidak tersedia!")

# Fitur tambahan untuk admin
if login_user == "admin":
    print("\n === MENU ADMIN ===")
    print("a. Tambah Sepatu Baru")
    print("b. Kembali ke Menu Utama")
    admin_choice = input("Pilih: ").lower()
    if admin_choice == "a":
        nama_baru = input("Masukkan nama sepatu baru: ")
        try:
            harga_baru = int(input("Masukkan harga sepatu: "))
            tambah_sepatu_admin(nama_baru, harga_baru)
        except ValueError:
            print("Harga harus berupa angka!")
    elif admin_choice == "b":
        continue

```

Gambar 3.8 Source Code Program utama

#### 4. Hasil Output

```

=== LOGIN TOKO SEPATUKU ===
Masukkan Username:
Masukkan Password: █

```

Gambar 4.1 Tampilan menu awal

```

=== LOGIN TOKO SEPATUKU ===
Masukkan Username: andhika
Masukkan Password: 150407

```

Gambar 4.2 Tampilan input username dan password

```
=== LOGIN TOKO SEPATUKU ===  
Masukkan Username: tamu  
Masukkan Password: 12345  
  
Login berhasil! Selamat datang, tamu.
```

Gambar 4.3 Tampilan login admin

```
Masukkan username pelanggan baru: hamm  
Masukkan password pelanggan baru: hamm1  
Akun Pelanggan Berhasil Ditambahkan
```

Gambar 4.4 Tampilan ingin menambah pelanggan

```
=== DAFTAR PELANGGAN ===  
1. Username: andhika | Password: 150407  
2. Username: tamu | Password: 12345
```

Gambar 4.5 Tampilan Data pelanggan yang sudah ditambahkan (read)

```
=== MENU DATA PELANGGAN ===  
1. Lihat Semua Pelanggan  
2. Tambah Pelanggan Baru  
3. Ubah Data Pelanggan  
4. Hapus Pelanggan  
5. Kembali ke Menu Utama  
Pilih menu (1-5): 3  
  
=== UBAH DATA PELANGGAN ===  
Masukkan username yang ingin diubah: andhika  
Masukkan password baru: 3334  
Password pelanggan 'andhika' berhasil diubah.
```

Gambar 4.6 Tampilan data yang mau di ubah

```
=== DAFTAR PELANGGAN ===  
1. Username: andhika | Password: 3334  
2. Username: tamu | Password: 12345
```

Gambar 4.7 Tampilan data setelah diubah (update)



```
=== HAPUS DATA PELANGGAN ===  
Masukkan ID pelanggan: 1  
Data pelanggan berhasil dihapus!  
  
Tekan Enter untuk kembali...  
█
```

Gambar 4.8 Tampilan data yang mau di hapus

```
=== MENU DATA PELANGGAN ===  
1. Lihat Semua Pelanggan  
2. Tambah Pelanggan Baru  
3. Ubah Data Pelanggan  
4. Hapus Pelanggan  
5. Kembali ke Menu Utama  
Pilih menu (1-5): 4  
  
=== HAPUS DATA PELANGGAN ===  
Masukkan username yang ingin dihapus: tamu  
Yakin ingin menghapus tamu? (y/n): y  
Pelanggan 'tamu' berhasil dihapus.
```

Gambar 4.9 Tampilan data yang sudah di hapus

```
=== DAFTAR SEPATU ===  
1. Sneakers Basic - Rp 250,000  
2. Running Pro - Rp 400,000  
3. Casual Street - Rp 350,000  
4. Formal Leather - Rp 500,000
```

Gambar 4.10 Tampilan paket (nomor 5)

```
=== MENU TOKO SEPATUKU ===  
1. Lihat Daftar Sepatu  
2. Beli Sepatu  
3. Lihat Riwayat Transaksi  
4. Logout  
Pilih menu (1-4): █
```

Gambar 4.11 Tampilan login sebagai pelanggan biasa

```
=== REGISTER AKUN BARU ===  
Masukkan username: oppaa  
Masukkan password: oppa1  
Akun Berhasil Ditambahkan  
Akun oppaa berhasil dibuat! Silakan login untuk melanjutkan.
```

Gambar 4.12 Tampilan Register

```
=== SELAMAT DATANG DI TOKO SEPATUKU ===  
1. Login  
2. Register  
3. Keluar  
Pilih menu (1-3): 3  
Terima kasih! Program selesai.
```

Gambar 4.13 Tampilan Keluar

## 5. Langkah-langkah GIT

### 5.1 GIT Add

```
PS C:\partikum-apd\pratikum-apd> git push -u origin main
```

Gambar 5.1 Langkah Git

Perintah git add . digunakan untuk menambahkan semua perubahan file yang ada di dalam folder proyek ke dalam staging area Git. Staging area adalah tempat sementara di mana perubahan file disiapkan sebelum benar-benar disimpan ke dalam riwayat repository melalui perintah git commit.

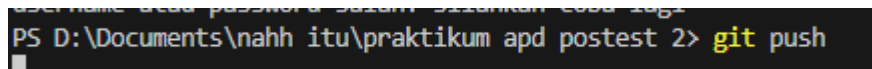
### 5.2 GIT Commit

```
PS D:\Documents\nahh itu\praktikum apd posttest 2> git commit
```

Gambar 5.2 Langkah Git

Commit dalam Git dapat diibaratkan seperti menyimpan catatan atau rekaman atas perubahan yang telah dilakukan pada proyek. Git commit berfungsi untuk menyimpan (merekam) snapshot atau perubahan pada kode atau file di repository Git. Saat kamu melakukan commit, Git akan menyimpan semua perubahan yang sudah kamu staging (dimasukkan ke area staging) dalam sebuah commit object

## 5.3 GIT Push



```
PS D:\Documents\nahh itu\praktikum apd posttest 2> git push
```

Gambar 5.3 Langkah Git

Git push adalah perintah yang digunakan untuk mengirimkan perubahan atau hasil kerja dari komputer Anda (repository lokal) ke penyimpanan Git yang ada di internet atau server (repository remote), seperti GitHub atau GitLab. Setelah Anda melakukan perubahan dan menyimpannya secara lokal, perintah ini berfungsi agar perubahan tersebut dapat tersimpan secara online dan dapat diakses oleh orang lain yang memiliki akses ke repository tersebut.