

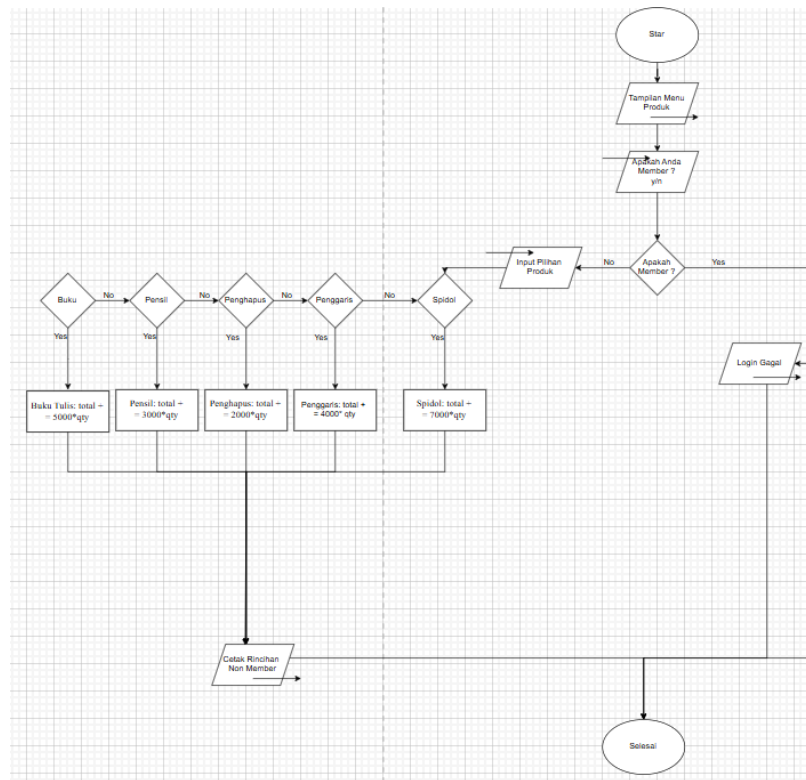
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN DASAR



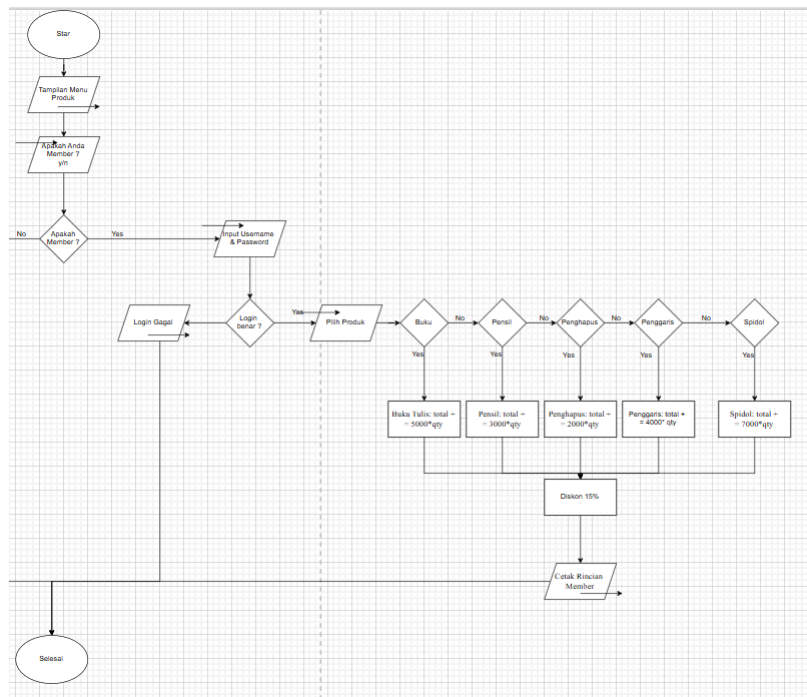
Disusun oleh:
Aditya Fitriansyah (2509106002)
Kelas A1'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Flowchart 1.1



Flowchart 1.2

2. Deskripsi Singkat Program

Program **Sistem Belanja Toko Sederhana** ini dibuat untuk menerapkan konsep **perulangan (looping)** dan **percabangan (if-else)** dalam bahasa pemrograman Python. Program ini mensimulasikan kegiatan belanja di toko, di mana pengguna dapat memilih produk, menentukan jumlah pembelian, dan melakukan checkout. Pada awal program, sistem akan menanyakan status keanggotaan pengguna. Jika pengguna memilih “y”, maka sistem akan meminta login dengan username dan password, dengan batas tiga kali percobaan. Jika login berhasil, pengguna menjadi **member** dan mendapatkan **diskon 15%** dari total belanja. Namun, jika gagal login tiga kali atau memilih “n”, pengguna dianggap **non-member** dan tidak memperoleh potongan harga.

Selanjutnya, sistem akan menampilkan daftar produk dan meminta pengguna memilih barang yang ingin dibeli. Proses ini dilakukan menggunakan **perulangan while**, sehingga pengguna dapat berbelanja beberapa kali hingga memilih menu checkout. Setelah checkout, program akan mencetak **struk belanja** yang menampilkan daftar produk, jumlah, total harga, dan diskon jika pengguna adalah member. Di akhir program, sistem menanyakan apakah pengguna ingin melakukan transaksi baru. Jika menjawab “y”, program akan mengulang dari awal, sedangkan jika “n”, program akan berhenti dan menampilkan pesan terima kasih. Program ini menunjukkan penerapan konsep dasar perulangan, kondisi, serta penggunaan fungsi **break** dan **continue**.

3. Source Code

- Mengimpor library `os` agar bisa menggunakan `os.system('cls'/'clear')` untuk membersihkan layar terminal.
- Menentukan username dan password member yang sah.

```
import os
username_db = "member"
password_db = "12345"
```

- Menggunakan `while True` agar program bisa **berulang terus** setiap kali pengguna ingin melakukan transaksi baru.

```
while True:
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== Selamat Datang di Toko Sederhana ===")
```

- Menanyakan apakah pengguna **member atau non-member**.
- Menyiapkan variabel awal untuk **keranjang belanja** dan **total harga**.
- `is_member` sebagai penanda status login.

```
status = input("Apakah Anda member? (y/n): ").lower()
keranjang = []
total = 0
is_member = False
```

- Jika user memilih “y”, sistem memberi **3 kali kesempatan login**.
- `.strip()` digunakan agar input kosong atau berisi spasi tidak diterima.
- Jika login benar → `is_member = True`, jika salah terus → dianggap non-member.

```
if status == 'y':
    kesempatan = 3
    while kesempatan > 0:
        username = input("Masukkan username: ").strip()
        password = input("Masukkan password: ").strip()
```

- Menampilkan **daftar produk dan harga** menggunakan `for loop`.
- Program terus berulang selama user belum memilih “6” (checkout).

```
while True:
    print("\n=== Menu Produk Toko ===")
    for i, (nama, harga) in produk.items():
        print(f"{i}. {nama:<12} - Rp{harga:,}")
    print("6. Checkout")
```

Fungsi utama program ini adalah untuk mensimulasikan proses belanja di toko secara interaktif menggunakan konsep perulangan (while) dan percabangan (if-else).

Program menanyakan status keanggotaan pengguna, memberikan fitur login dengan 3 kali kesempatan, menampilkan daftar produk, menghitung total harga dan diskon 15% bagi member, serta menampilkan struk belanja. Setelah transaksi selesai, pengguna dapat memilih untuk melakukan transaksi baru atau keluar dari program.

Dengan kata lain, fungsi utamanya adalah mengatur alur belanja dari login, pemilihan produk, perhitungan harga, hingga checkout menggunakan dasar logika pemrograman yang sederhana dan efisien.

```
import os # Membersihkan layar terminal
# Data akun member
username_db = "member"
password_db = "12345"
# Daftar produk
produk = {
    1: ["Buku Tulis", 5000],
    2: ["Pensil", 3000],
    3: ["Penghapus", 2000],
    4: ["Penggaris", 4000],
    5: ["Spidol", 7000]
}

while True:
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== Selamat Datang di Toko Sederhana ===")
    status = input("Apakah Anda member? (y/n): ").lower()
    keranjang = []
    total = 0
    is_member = False

    # Proses Login jika member
    if status == 'y':
        kesempatan = 3
        while kesempatan > 0:
            username = input("Masukkan username: ").strip()
            password = input("Masukkan password: ").strip()
            if username == username_db and password == password_db:
                print("Login berhasil! Silakan belanja.")
                is_member = True
                break
            else:
                kesempatan -= 1
                print(f"Login gagal! Sisa percobaan: {kesempatan}")
        if not is_member:
            print("Gagal Login 3 kali, dianggap non-member.\n")

    # Menu belanja
    while True:
        print("\n=== Menu Produk Toko ===")
        for i, (nama, harga) in produk.items():
```

```

        print(f"{i}. {nama:<12} - Rp{harga:,}")
    print("6. Checkout")

    pilih = int(input("Pilih produk: "))
    if pilih == 6:
        break
    elif pilih in produk:
        qty = int(input(f"Masukkan jumlah {produk[pilih][0]}: "))
        subtotal = produk[pilih][1] * qty
        keranjang.append((produk[pilih][0], qty, subtotal))
        total += subtotal
        print(f"{produk[pilih][0]} ditambahkan. Total sementara:
Rp{total:,}")
    else:
        print("Produk tidak tersedia!")

# Cetak struk
os.system('cls' if os.name == 'nt' else 'clear')
print("==== STRUK BELANJA =====")
for item, qty, subtotal in keranjang:
    print(f"{item:<12} x{qty:<3} = Rp{subtotal:,}")
if is_member:
    diskon = total * 0.15
    total_bayar = total - diskon
    print(f"\nTotal: Rp{total:,}\nDiskon 15%: Rp{diskon:,}\nTotal Bayar:
Rp{total_bayar:,}")
else:
    print(f"\nTotal Bayar: Rp{total:,}")

# Ulangi transaksi
ulang = input("\nTransaksi baru? (y/n): ").lower()
if ulang != 'y':
    print("Terima kasih telah berbelanja!")
    break

```

4. Langkah-langkah GIT

4.1 GIT Init

Git Init adalah perintah Git yang digunakan untuk menginisialisasi repository Git baru dalam sebuah folder. Cukup ketik “git init” pada terminal VSCode

```

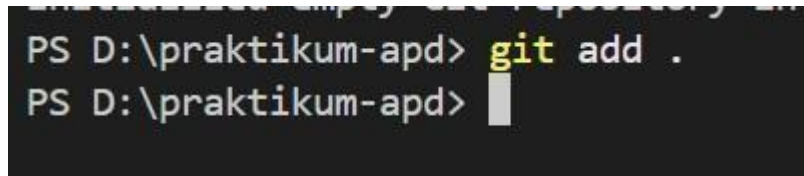
PS D:\praktikum-apd> git init
Initialized empty Git repository in D:/praktikum-apd/.git/

```

Gambar 5.1 Git Init

4.2 GIT Add

Git add adalah perintah Git untuk menambahkan semua perubahan file di folder kerja (working directory) ke staging area, agar siap dicommit ke dalam repository.

A screenshot of a Windows command prompt window. The background is black, and the text is white. The prompt shows the current directory as 'D:\praktikum-apd'. The command 'git add .' has been entered, and the cursor is at the end of the line. The word 'git' is highlighted in yellow.

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

Gambar 5.2 Git Add .

4.3 GIT Commit

Adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository. Setiap commit akan memiliki hash unik, pesan commit, dan menyimpan snapshot dari perubahan yang dilakukan.

```
PS D:\praktikum-apd> git commit -m "Finish Posttest 3"
[main 040802a] Finish Posttest 3
3 files changed, 84 insertions(+)
create mode 100644 post-test/post-test-apd-3/2509106002-AdityaFitriansyah-PT-3.pdf
create mode 100644 post-test/post-test-apd-3/2509106002-AdityaFitriansyah-PT-3.py
```

Gambar 5.3 Git Commit

4.4 GIT Remote

Adalah perintah Git yang digunakan untuk menghubungkan repository lokal dengan repository remote (misalnya di GitHub, GitLab, atau Bitbucket). Dengan perintah ini, kita bisa mengelola koneksi ke repository jarak jauh, memungkinkan push, pull, dan fetch dari repository tersebut.

```
PS D:\praktikum-apd> git remote add origin git@github.com:adityaftriansyah/praktikum-apd.git
```

Gambar 5.4 Git Remote

4.5 GIT Push

Adalah perintah Git yang digunakan untuk mengirim (mengunggah) perubahan dari repository lokal ke repository remote (seperti GitHub, GitLab, atau Bitbucket). Perintah ini memastikan bahwa perubahan yang sudah dikomit di lokal tersedia di repository jarak jauh sehingga bisa diakses oleh orang lain atau untuk Cadangan.

```
PS D:\praktikum-apd> git push -u origin main
```

Gambar 5.5 Git Push