

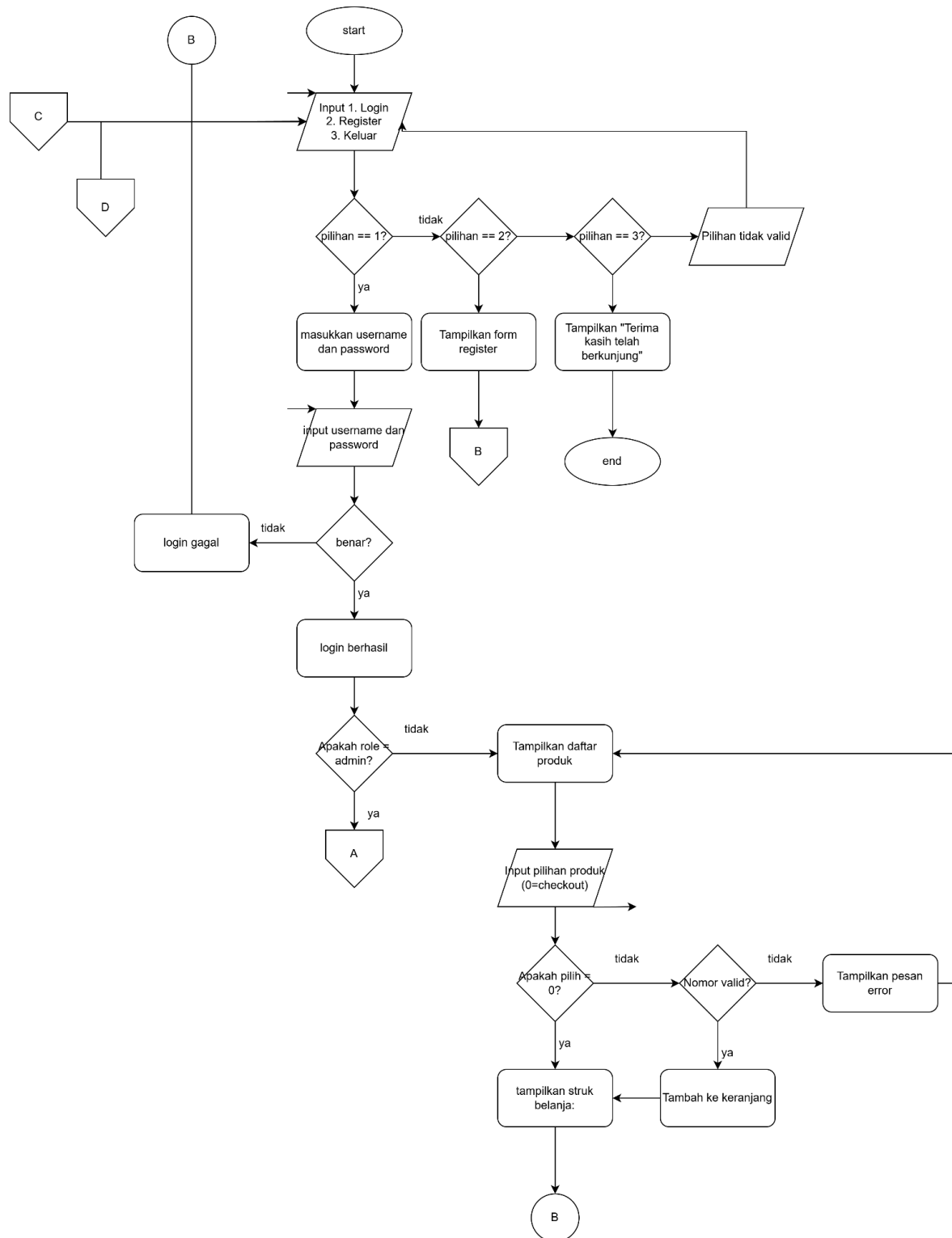
LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR



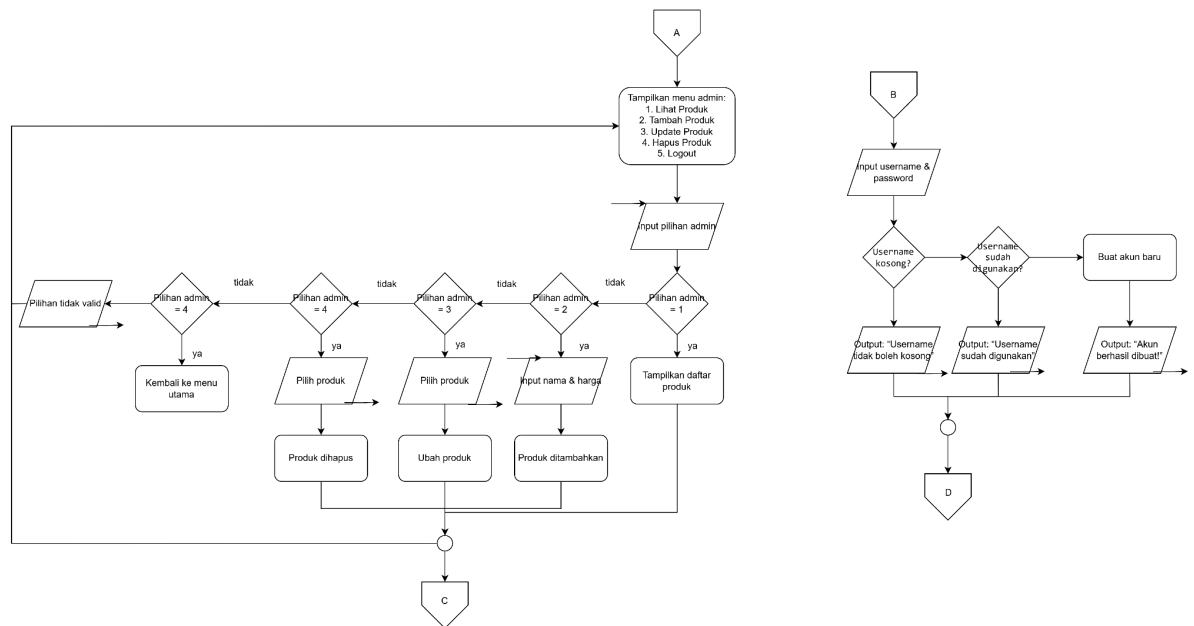
Disusun oleh:
Bintang Dhana Permadi (2509106014)
Kelas (A1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart Program toko baju brand



Gambar 1.2 Flowchart Program toko baju brand

- **Penjelasan singkat flowchart:**

Program “Toko Baju Brand” adalah program Python yang mensimulasikan sistem penjualan pakaian sederhana. Di dalamnya terdapat fitur login, register, serta menu admin dan menu user. Admin dapat menambah, mengubah, dan menghapus produk, sedangkan user dapat melihat produk dan melakukan pembelian.

2. Deskripsi Singkat Program

Program Toko Baju Brand adalah aplikasi toko pakaian sederhana berbasis terminal yang memiliki dua jenis pengguna: admin dan user. Admin dapat mengelola produk (melihat, menambah, mengubah, dan menghapus), sedangkan user dapat melihat daftar produk, menambahkannya ke keranjang, dan melakukan checkout. Program juga mendukung fitur login dan register, serta melakukan validasi input tanpa menggunakan `def()` atau `isdigit()`.

3. Source Code

```
import os

users = {
    "admin": {"password": "admin123", "role": "admin"},
    "udin": {"password": "1234", "role": "user"}
}
```

```

produk = {
    1: {"nama": "Kaos Lacoste", "harga": 250000},
    2: {"nama": "Celana Rucas", "harga": 400000},
    3: {"nama": "Jaket H&M", "harga": 550000},
    4: {"nama": "Topi Nike", "harga": 150000},
    5: {"nama": "Sepatu Adidas", "harga": 750000}
}

keranjang_global = []

def clear_screen():
    """Membersihkan layar terminal."""
    os.system('cls' if os.name == 'nt' else 'clear')

def tampilkan_header():
    """Menampilkan header toko."""
    print("=== SELAMAT DATANG DI TOKO BAJU BRAND ===")

def tambah_produk(id_baru, nama, harga):
    """Menambahkan produk baru ke dalam dictionary produk."""
    global produk
    produk[id_baru] = {"nama": nama, "harga": harga}
    print(f"Produk '{nama}' berhasil ditambahkan!")

def hitung_total(keranjang):
    """Menghitung total harga dari semua produk di keranjang."""
    total = 0
    for item in keranjang:
        total += item["harga"]
    return total

def tampilkan_produk_recursive(keys, index=0):
    """Menampilkan produk secara rekursif."""
    if index >= len(keys):
        return
    key = keys[index]
    data = produk[key]

```

```

print(f"{key}. {data['nama']} - Rp{data['harga']:,}")
tampilkan_produk_recursive(keys, index + 1)

def daftar_produk():
    """Prosedur untuk menampilkan daftar produk."""
    clear_screen()
    print("=== DAFTAR PRODUK ===")
    if len(produk) == 0:
        print("Belum ada produk.")
    else:
        keys = list(produk.keys())
        tampilkan_produk_recursive(keys)
    input("\nTekan Enter untuk kembali...")

def struk_belanja(keranjang):
    """Menampilkan struk belanja (prosedur tanpa return)."""
    clear_screen()
    print("=== STRUK BELANJA ===")
    for item in keranjang:
        print(f"- {item['nama']} : Rp{item['harga']:,}")
    total = hitung_total(keranjang)
    print("-----")
    print(f"Total belanja : Rp{total:,}")
    print("Terima kasih telah berbelanja!\n")
    input("Tekan Enter untuk keluar...")

while True:
    clear_screen()
    tampilkan_header()
    print("1. Login")
    print("2. Register")
    print("3. Keluar")
    pilih = input("Pilih menu: ")

    if pilih == "1":
        clear_screen()
        print("=== LOGIN ===")
        username = input("Username: ").strip()
        password = input("Password: ").strip()

```

```

        if username in users and users[username]["password"] ==
password:
            role = users[username]["role"]
            print(f"Login berhasil sebagai {role.upper()}!")
            input("Tekan Enter untuk melanjutkan...")

            if role == "admin":
                while True:
                    clear_screen()
                    print("=== MENU ADMIN ===")
                    print("1. Lihat Produk")
                    print("2. Tambah Produk")
                    print("3. Update Produk")
                    print("4. Hapus Produk")
                    print("5. Logout")
                    menu_admin = input("Pilih menu: ")

                    try:
                        if menu_admin == "1":
                            daftar_produk()

                        elif menu_admin == "2":
                            clear_screen()
                            print("=== TAMBAH PRODUK ===")
                            nama = input("Masukkan nama produk:
").strip()

                            harga = input("Masukkan harga produk:
").strip()

                            if nama == "" or harga == "":
                                print("Nama dan harga tidak boleh
kosong!")

                                elif not harga.isdigit():
                                    print("Harga harus berupa angka!")
                                else:
                                    id_baru = max(produk.keys()) + 1
                                    tambah_produk(id_baru, nama,
int(harga))

                                    input("\nTekan Enter untuk
kembali...")

```

```

elif menu_admin == "3":
    clear_screen()
    print("=== UBAH PRODUK ===")
    daftar_produk()
    index = input("\nMasukkan nomor produk
yang ingin diubah: ")

    if index.isdigit() and int(index) in
produk:
        index = int(index)
        nama_baru = input("Nama baru
(kosongkan jika tidak diubah): ").strip()
        harga_baru = input("Harga baru
(kosongkan jika tidak diubah): ").strip()
        if nama_baru != "":
            produk[index]["nama"] =
nama_baru
        if harga_baru != "" and
harga_baru.isdigit():
            produk[index]["harga"] =
int(harga_baru)
        print("Produk berhasil
diperbarui!")
    else:
        print("Nomor produk tidak
ditemukan!")
        input("\nTekan Enter untuk
kembali...")

elif menu_admin == "4":
    clear_screen()
    print("=== HAPUS PRODUK ===")
    daftar_produk()
    index = input("\nMasukkan nomor produk
yang ingin dihapus: ")

    if index.isdigit() and int(index) in
produk:
        hapus = produk.pop(int(index))
        print(f"Produk '{hapus['nama']}'
berhasil dihapus.")
    else:

```

```

                                print("Nomor produk tidak
ditemukan!")

                                input("\nTekan Enter untuk
kembali...")

                                elif menu_admin == "5":
                                    break
                                else:
                                    print("Pilihan tidak valid!")
                                    input("\nTekan Enter untuk
kembali...")

                                except Exception as e:
                                    print(f"Terjadi kesalahan: {e}")
                                    input("\nTekan Enter untuk kembali...")

                                else:
                                    keranjang_global.clear()
                                    while True:
                                        clear_screen()
                                        print(f"=== MENU PENGGUNA: {username.upper()}
===")

                                        keys = list(produk.keys())
                                        tampilkan_produk_recursive(keys)
                                        print("0. Checkout")

                                        pilih_user = input("\nPilih produk: ")

                                        try:
                                            if pilih_user == "0":
                                                struk_belanja(keranjang_global)
                                                break
                                            elif pilih_user.isdigit() and
int(pilih_user) in produk:
                                                item = produk[int(pilih_user)]
                                                keranjang_global.append(item)
                                                print(f"{item['nama']} berhasil
ditambahkan ke keranjang!")
                                            else:
                                                print("Input tidak valid!")
                                                input("\nTekan Enter untuk Lanjut...")
                                        except Exception as e:
                                            print(f"Terjadi kesalahan: {e}")

```

```

        input("\nTekan Enter untuk Lanjut...")

    else:
        print("Login gagal! Username atau password salah.")
        input("Tekan Enter untuk kembali...")

elif pilih == "2":
    clear_screen()
    print("=== REGISTER AKUN BARU ===")
    username = input("Buat username: ").strip()
    password = input("Buat password: ").strip()

    try:
        if username in users:
            print("Username sudah digunakan!")
        elif username == "" or password == "":
            print("Username dan password tidak boleh kosong!")
        else:
            users[username] = {"password": password, "role":
"user"}

            print("Akun berhasil dibuat! Silakan login.")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
        input("\nTekan Enter untuk kembali...")

elif pilih == "3":
    clear_screen()
    print("Terima kasih telah mengunjungi Toko Baju Brand!")
    break

else:
    print("Pilihan tidak valid!")
    input("Tekan Enter untuk kembali...")

```

Gambar 2.1 Source Code

4. Hasil Output

- Menu awal saat datang ke toko

```
=== SELAMAT DATANG DI TOKO BAJU BRAND ===  
1. Login  
2. Register  
3. Keluar  
Pilih menu:
```

Gambar 3.1 Output pilihan/opsi awal

- Tampilan login berhasil

```
=== LOGIN TOKO BAJU BRAND ===  
Username: udin  
Password: 1234  
  
Login berhasil sebagai USER!  
Tekan Enter untuk melanjutkan...
```

Gambar 3.2 Output login berhasil

- Daftar produk

```
=== MENU PENGGUNA: UDIN ===  
1. Kaos Lacoste - Rp250,000  
2. Celana Rucas - Rp400,000  
3. Jaket H&M - Rp550,000  
4. Topi Nike - Rp150,000  
5. Sepatu Adidas - Rp750,000  
0. Checkout  
  
Pilih produk: 
```

Gambar 3.3 Output pilihan daftar produk

- Struk hasil transaksi

```
=== STRUK BELANJA ===  
- Kaos Lacoste : Rp250,000  
-----  
Total Belanja : Rp250,000  
Terima kasih sudah berbelanja!  
  
Tekan Enter untuk keluar...
```

Gambar 3.4 Output Struk hasil transaksi

- Pembeli membuat akun baru

```
=== REGISTER AKUN BARU ===  
Buat username: asep  
Buat password: 27  
Akun berhasil dibuat! Silakan login.  
  
Tekan Enter untuk kembali...
```

Gambar 3.5 Output membuat akun baru

- Login akun baru

```
=== LOGIN TOKO BAJU BRAND ===  
Username: asep  
Password: 27  
  
Login berhasil sebagai USER!  
Tekan Enter untuk melanjutkan...
```

Gambar 3.6 Output login akun baru

- Struk belanja akun baru

```
=== STRUK BELANJA ===  
- Topi Nike : Rp150,000  
- Sepatu Adidas : Rp750,000  
-----  
Total Belanja : Rp900,000  
Terima kasih sudah berbelanja!  
  
Tekan Enter untuk keluar...
```

Gambar 3.7 Output struk belanja akun baru

- Tampilan login gagal

```
=== LOGIN TOKO BAJU BRAND ===  
Username: asepa  
Password: 12  
  
Login gagal! Username atau password salah.  
Tekan Enter untuk kembali...
```

Gambar 3.8 Output struk belanja akun baru

- Login admin

```
=== LOGIN ===  
Username: admin  
Password: admin123  
Login berhasil sebagai ADMIN!  
Tekan Enter untuk melanjutkan...
```

Gambar 3.9 Output login akun admin

- Menu Login akun admin

```
=== MENU ADMIN TOKO BAJU BRAND ===  
1. Lihat Produk  
2. Tambah Produk  
3. Update Produk  
4. Hapus Produk  
5. Logout  
Pilih menu: 
```

Gambar 3.10 Output menu login akun admin

5. Langkah-langkah GIT

5.1 GIT Add

```
PS D:\kuliah\SEMESTER 1\APD\praktikum_apd> git add .
```

Gambar 4.1 Git add

Setelah membuat atau mengubah file, Kita perlu memberitahu Git file mana yang akan dimasukkan ke dalam riwayat versi berikutnya. Perintah git add ini biasanya digunakan untuk menambahkan perubahan dari satu atau lebih file ke area staging (area persiapan). Area ini berfungsi sebagai "keranjang" tempat Kita mengumpulkan semua perubahan yang ingin disimpan dalam satu commit.

5.2 GIT Commit

```
PS D:\kuliah\SEMESTER 1\APD\praktikum_apd> git commit -m "upload"
[main 47e16d9] upload
3 files changed, 372 insertions(+), 6 deletions(-)
create mode 100644 kelas/pertemuan7.py
create mode 100644 post-test/post-test-apd-7/2509106014-Bintang Dhana Permadi-PT-7.py
```

Gambar 4.2 Git Commit

Perintah ini menambahkan semua file yang ada di dalam folder ke staging area. Staging area adalah tempat sementara sebelum file dikomit ke dalam repository.

5.3 GIT Push

```
PS D:\kuliah\SEMESTER 1\APD\praktikum_apd> git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 3.53 KiB | 1.76 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Bintang-2727/praktikum-apd.git
b1bc66e..47e16d9 main -> main
```

Gambar 4.3 Git Push

Setelah commit Kita siap, perintah git push digunakan untuk mengunggah commit dari repository lokal Anda ke repository jarak jauh (misalnya, GitHub). Ini membuat perubahan yang Anda simpan di komputer pribadi kita bisa dilihat dan diakses oleh orang lain yang berkolaborasi dalam proyek yang sama. Perintah ini biasanya diikuti dengan nama remote dan nama branch, seperti: git push origin main.