

**LAPORAN PRAKTIKUM  
POSTTEST 7  
ALGORITMA PEMROGRAMAN DASAR**



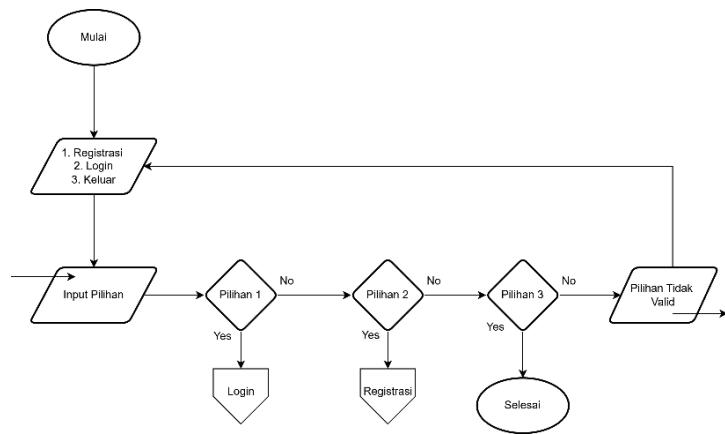
**Disusun oleh:**

**Aditya Fitriansyah (2509106002)**

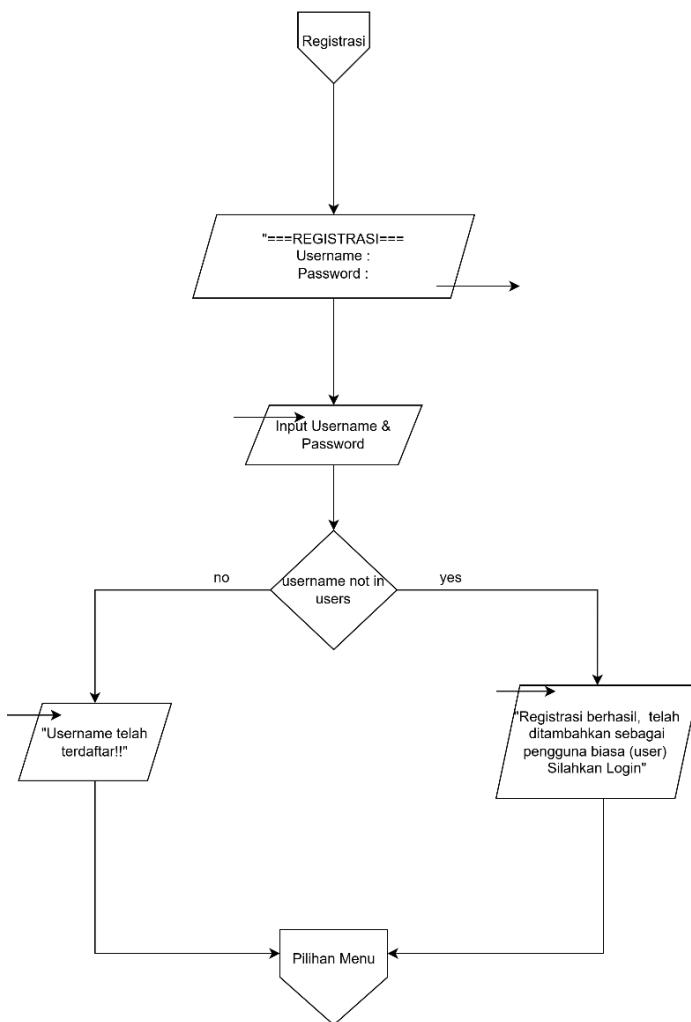
**Kelas A1'25**

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS MULAWARMAN  
SAMARINDA  
2025**

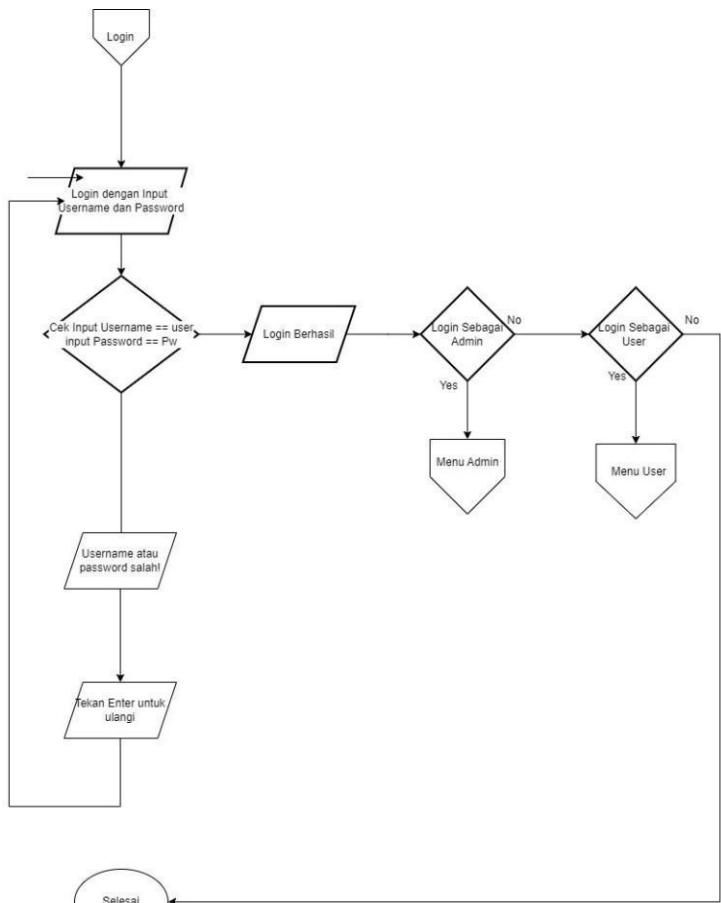
## 1. Flowchart



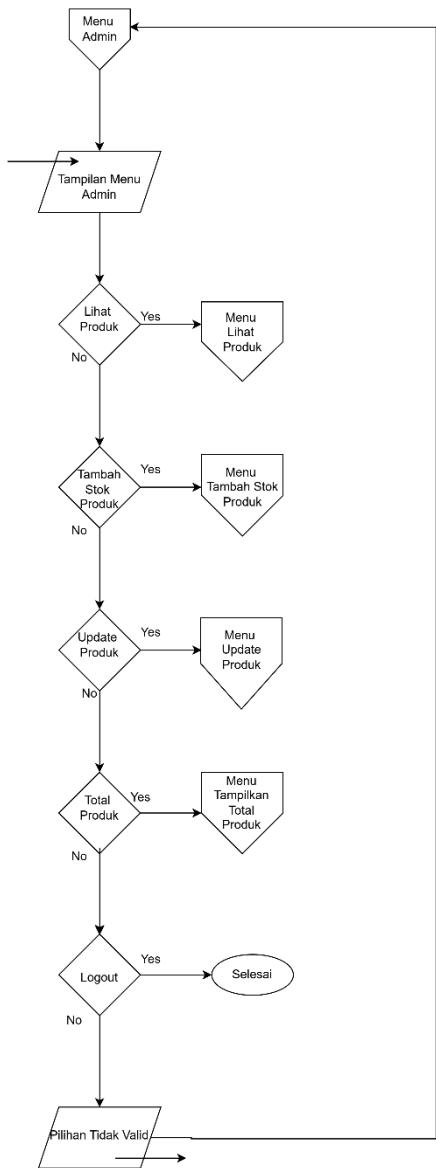
Flowchart 1.1 Pilihan Menu



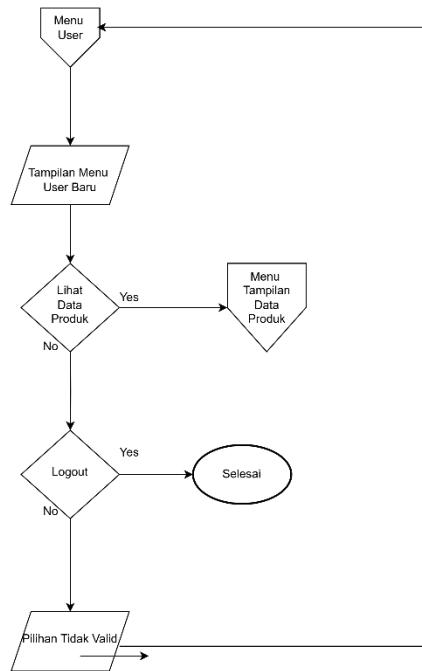
Flowchart 1.2 Registrasi



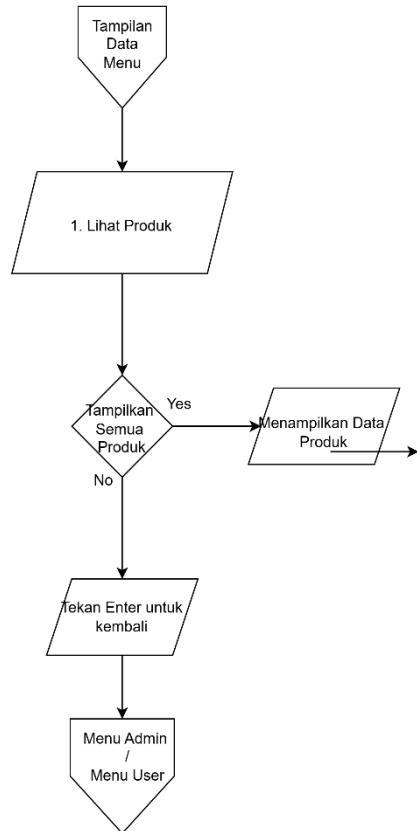
*Flowchart 1.3 Login*



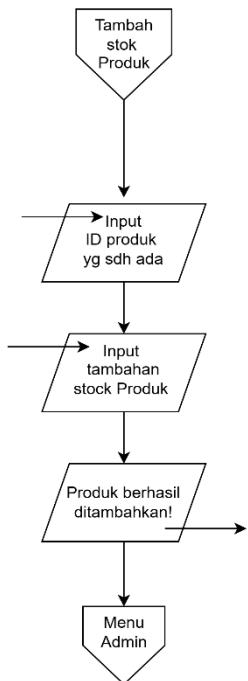
Flowchart 1.4 Menu Admin



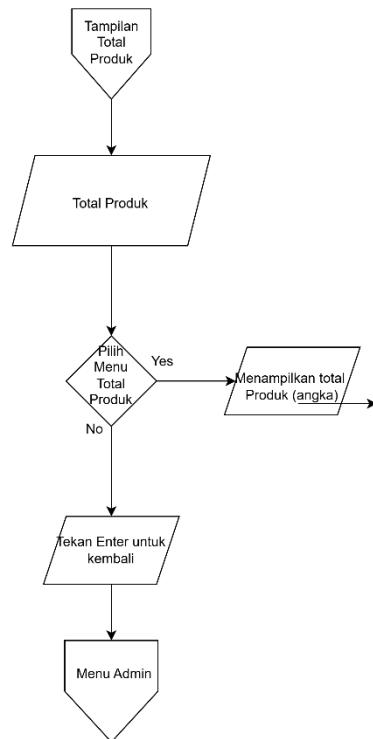
Flowchart 1.5 Menu User



Flowchart 1.6 Lihat Produk



Flowchart 1.7 Tambah Stok



- Program dimulai dengan menampilkan menu Registrasi/Daftar, Login, dan Keluar program
- Jika memilih menu regis, maka pengguna akan diarahkan mengisi Username dan Password
- Login sebagai User atau Admin dengan memasukkan Username dan Password yang telah ditentukan
- Jika Login sebagai user, Maka Pengguna hanya bisa Melihat tampilan/list produk lalu Logout
- Jika Login sebagai Admin, Maka :
  - Pengguna bisa melihat data produk
  - Pengguna bisa menambah stok produk
  - Pengguna bisa melihat berapa total produk
- Jika memilih **Lihat Produk**, Maka pengguna langsung bisa melihat list/data produk (ID, Nama, Stok, dan Harga)
- Jika memilih **Tambah Stok**, Maka pengguna akan diminta untuk memasukkan ID dan Jumlah tambah stok produk yang telah tersedia
- Jika memilih **Total Produk**, Maka pengguna bisa melihat berapa total produk yang tersedia (berupa angka)
- Program diakhiri dengan menu Logout dan Keluar.

## 2. Deskripsi Singkat Program

Program di atas bertujuan untuk mengelola data stok produk dalam sebuah sistem sederhana berbasis Python dengan fitur login dan register. Program ini memungkinkan pengguna untuk membuat akun baru (register) dan masuk ke sistem (login) dengan dua peran berbeda, yaitu **admin** dan **user**. Admin memiliki akses penuh untuk melihat daftar produk, menambah stok, dan menghitung total produk yang tersedia, sedangkan user hanya dapat melihat daftar produk. Selain itu, program ini juga menggunakan berbagai konsep pemrograman seperti fungsi dengan dan tanpa parameter, rekursi untuk konfirmasi keluar, serta penanganan kesalahan dengan try-except. Secara keseluruhan, program ini berfungsi sebagai simulasi sistem manajemen stok yang interaktif dan terstruktur, sekaligus melatih penerapan logika kontrol, validasi input, dan peran pengguna dalam aplikasi sederhana.

### 3. Source Code

#### A. Variabel Global

Berfungsi untuk Menyimpan data akun dan data produk dalam bentuk dictionary.

- users berisi username, password, dan peran (admin atau user).
- produk menyimpan ID produk, nama, stok, dan harga.

```
users = {
    "admin": {"password": "admin123", "role": "admin"},
    "user": {"password": "user123", "role": "user"}
}

produk = {
    "001": {"nama": "Sabun", "stok": 15, "harga": 3000},
    "002": {"nama": "Shampoo", "stok": 10, "harga": 7000},
    "003": {"nama": "Sikat Gigi", "stok": 20, "harga": 4000}
}
```

#### B. Menampilkan Produk

Berfungsi untuk Menampilkan daftar produk dalam format tabel rapi, digunakan oleh admin dan user.

```
def tampilkan_produk():
    print("{:<5} {:<15} {:<10} {:<10}".format("ID", "Nama", "Stok", "Harga"))
    for idp, data in produk.items():
        print("{:<5} {:<15} {:<10} {:<10}".format(idp, data["nama"], data["stok"],
data["harga"]))
```

### C. Menambah Stok (Fungsi dengan Parameter)

Fungsinya agar Admin dapat menambah stok produk berdasarkan ID produk.

- Parameter idp dan jumlah diterima dari input user.
- Jika ID valid, stok ditambah dan pesan sukses dikembalikan.

```
def tambah_stok(idp, jumlah):  
    if idp in produk:  
        produk[idp]["stok"] += jumlah  
        return f"✅ Stok {produk[idp]['nama']} bertambah {jumlah}."  
    else:  
        return "❌ Produk tidak ditemukan."
```

### D. Menghitung Total Produk (Fungsi tanpa Parameter)

Berfungsi untuk Menghitung jumlah total produk yang ada di sistem.

```
def total_produk():  
    total = len(produk)  
    return total
```

### E. Konfirmasi Keluar (Fungsi Rekursif)

Memastikan pengguna yakin ingin keluar. Jika input salah, fungsi memanggil dirinya sendiri (rekursif) untuk validasi ulang.

```
def konfirmasi_keluar():  
    pilihan = input("Yakin ingin keluar? (y/n): ").lower()  
    if pilihan == "y":  
        print("👋 Terima kasih telah menggunakan sistem ini!")  
        exit()  
    elif pilihan == "n":  
        return  
    else:
```

## F. Register Akun Baru

Membuat akun baru dengan validasi : Tidak boleh kosong, Tidak boleh duplikat, dan Password minimal 5 karakter

```
if pilihan == "2":  
    username = input("Masukkan username baru: ").strip()  
    password = input("Masukkan password baru: ").strip()  
  
    if not username or not password:  
        raise ValueError("X Username dan password tidak boleh kosong!")  
    elif username in users:  
        raise KeyError("X Username sudah digunakan!")  
    elif len(password) < 5:  
        raise ValueError("X Password minimal 5 karakter!")  
  
    users[username] = {"password": password, "role": "user"}
```

## G. Login Sistem

Mengecek kecocokan username dan password dari users. Jika valid, login berhasil dan sistem menyesuaikan menu berdasarkan peran (role).

```
username = input("Username : ").strip()  
password = input("Password : ").strip()  
user = users.get(username)  
if not user:  
    raise KeyError("X Username tidak ditemukan!")  
elif user["password"] != password:  
    raise ValueError("X Password salah!")
```

## H. Menu Admin

Admin dapat :

- Melihat daftar produk (tampilkan\_produk())
- Menambah stok (tambah\_stok())
- Melihat total produk (total\_produk())

```
print("1. Lihat Produk")  
print("2. Tambah Stok")  
print("3. Total Produk")  
print("4. Logout")
```

## I. Menu User

User hanya bisa melihat produk tanpa mengubah data.

```
print("1. Lihat Produk")
print("2. Logout")
```

## 4. Hasil Output

```
==== SISTEM PENGELOLAAN STOK ====
1. Login
2. Register
3. Keluar
Pilih menu: █
```

*Output 4.1 Pilihan menu*

```
==== REGISTER AKUN BARU ===
Masukkan username baru : Aditya
Masukkan password baru : 002
 Akun berhasil dibuat!

Tekan Enter untuk kembali...█
```

*Output 4.2 Register*

```
==== LOGIN SISTEM ===
Username : admin
Password : admin123
 Login berhasil sebagai ADMIN!

Tekan Enter untuk melanjutkan...█
```

*Output 4.3 Login (Admin)*

```
==== MENU ADMIN ====
1. Lihat Produk
2. Tambah Stok
3. Total Produk
4. Logout
Pilih menu: 1
```

Output 4.4 Menu Admin

```
==== MENU ADMIN ====
1. Lihat Produk
2. Tambah Stok
3. Total Produk
4. Logout
Pilih menu: 1
ID      Nama          Stok       Harga
-----
001    Sabun           15        3000
002    Shampoo          10        7000
003    Sikat Gigi       20        4000
-----
Tekan Enter...■
```

Output 4.5 Lihat Produk

```
==== MENU ADMIN ====
1. Lihat Produk
2. Tambah Stok
3. Total Produk
4. Logout
Pilih menu: 2
ID      Nama          Stok       Harga
-----
001    Sabun           15        3000
002    Shampoo          10        7000
003    Sikat Gigi       20        4000
-----
Masukkan ID produk: 002
Jumlah tambah stok: 54
📢 🎉 Stok Shampoo bertambah 54.

Tekan Enter...■
```

Output 4.6 Tambah Stok Produk

```
==== MENU ADMIN ====
1. Lihat Produk
2. Tambah Stok
3. Total Produk
4. Logout
Pilih menu: 3
📦 Total produk: 3

Tekan Enter...█
```

Output 4.7 Melihat Total Produk

```
==== HAPUS PRODUK ===
ID      Nama        Stok      Harga
-----
001    Sabun        15       3000
002    Shampoo       10       7000
003    Parfum        30      24000
004    Odol          25      15000
-----
Masukkan ID produk yang ingin dihapus: 002
✓ Produk berhasil dihapus!

Tekan Enter untuk kembali...█
```

Output 4.8 Hapus Produk (Delete)

```
==== LOGIN SISTEM ====
Username : user
Password : user123
✓ Login berhasil sebagai USER!

Tekan Enter untuk melanjutkan...█
```

Output 4.9 Login (User)

```
==== MENU USER ====
1. Lihat Produk
2. Logout
Pilih menu: █
```

Output 4.10 Tampilan Menu (User)

## 5. GIT

### 5.1 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

### 5.2 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.

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

Gambar 5.2 Git Add .

### 5.3 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 Post Test"
[main e8ca52a] Finish Post Test
 4 files changed, 213 insertions(+)
 create mode 100644 kelas/pertemuan-6.py
 create mode 100644 post-test/post-test-apd-6/2509106002-AdityaFitriansyah-PT-6.pdf
 create mode 100644 post-test/post-test-apd-6/2509106002-AdityaFitriansyah-PT-6.py
```

Gambar 5.3 Git Commit

## 5.4 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:adityafriansyah/praktikum-apd.git
```

Gambar 5.4 Git Remote

## 5.5 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