

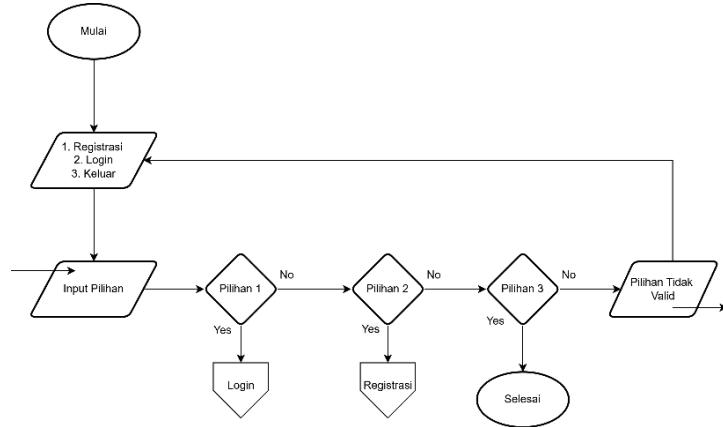
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN DASAR**



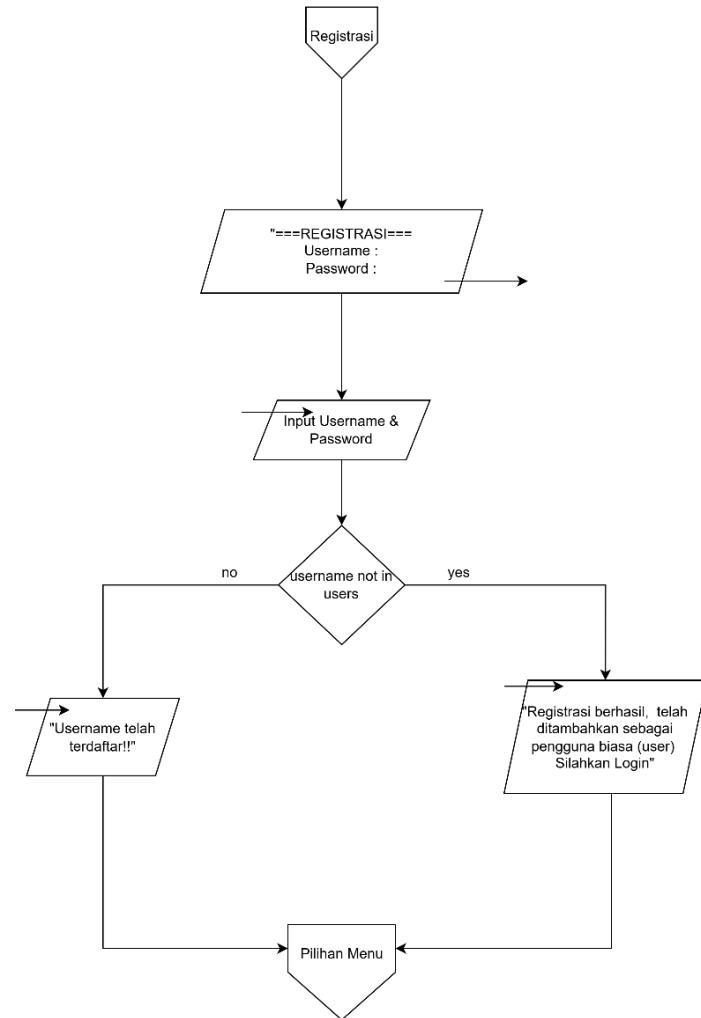
**Disusun oleh:**  
**Aditya Fitriansyah (2509106002)**  
**Kelas A1'25**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

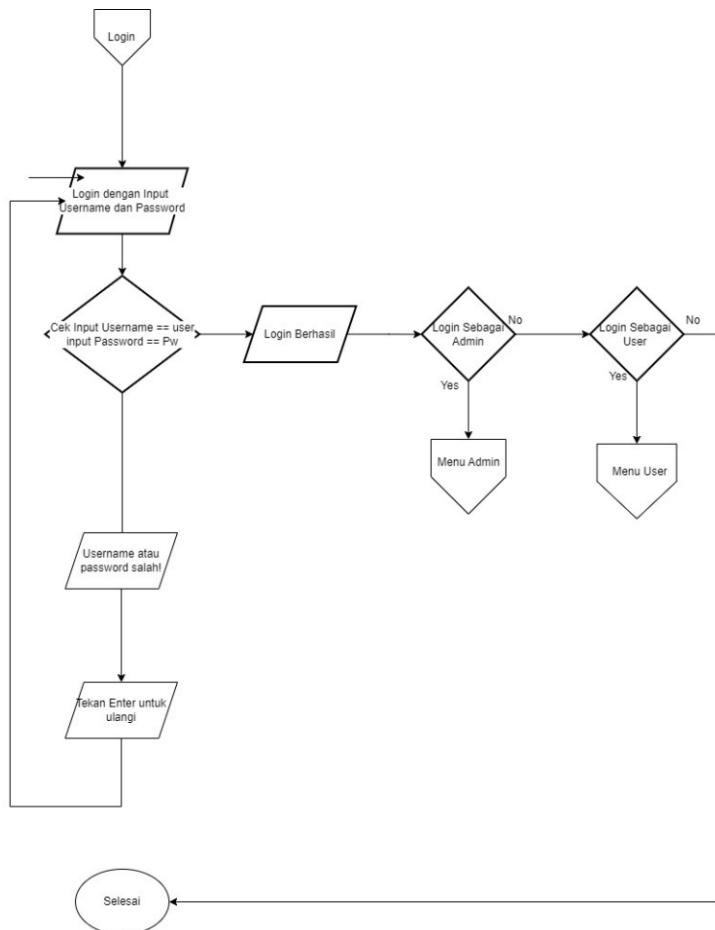
## 1. Flowchart



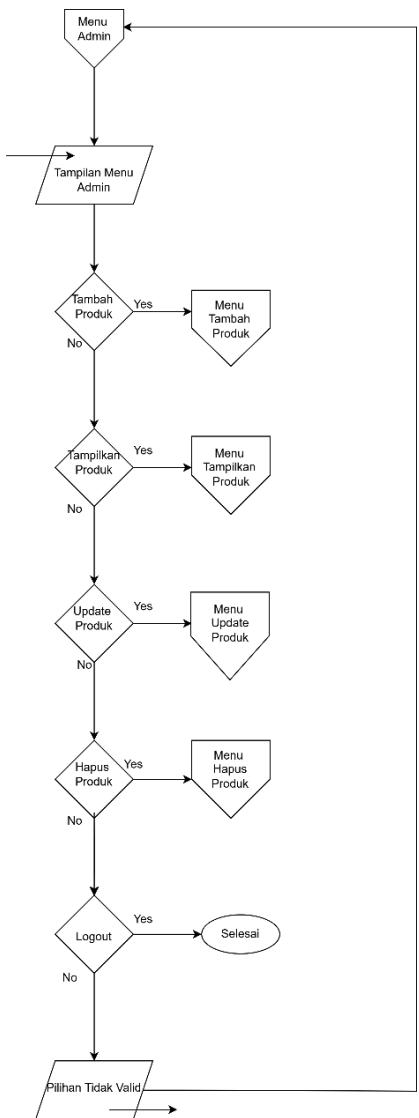
Flowchart 1.1 Pilihan Menu



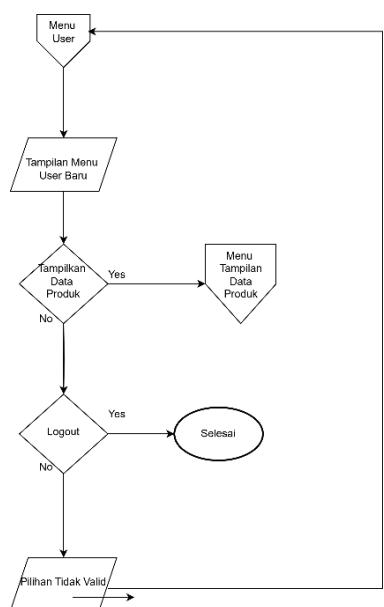
Flowchart 1.2 Registrasi



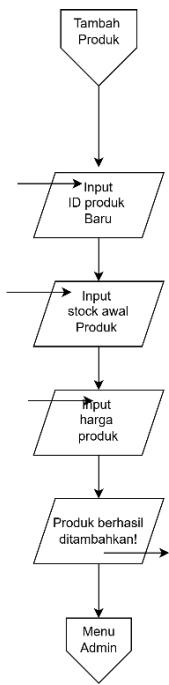
*Flowchart 1.3 Login*



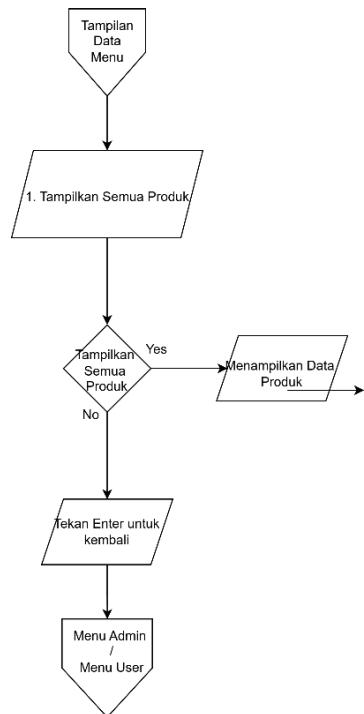
Flowchart 1.4 Menu Admin



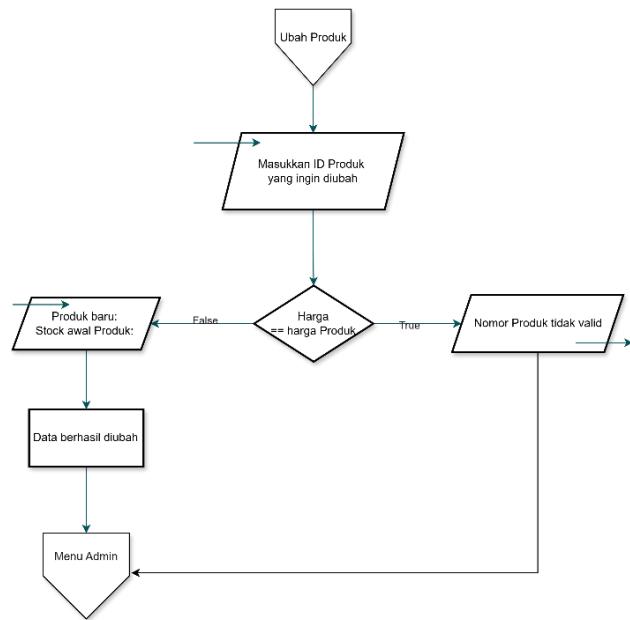
Flowchart 1.5 Menu User



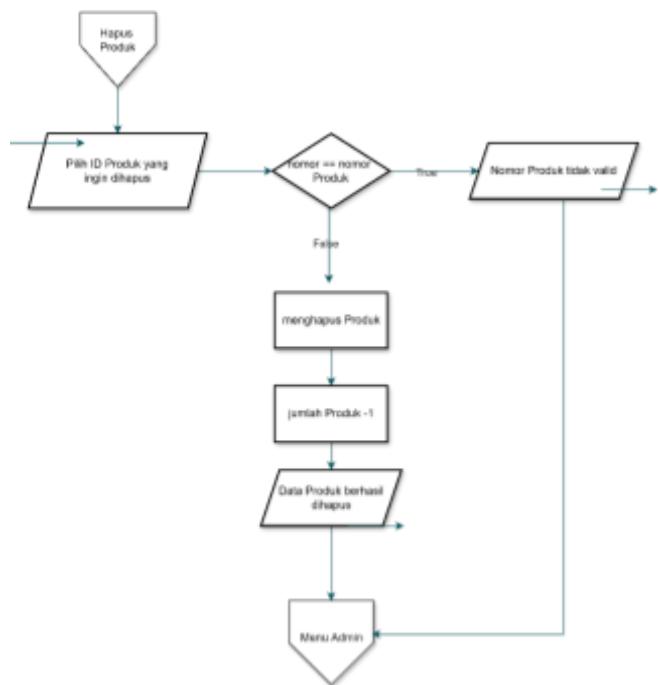
*Flowchart 1.6 Tambah Produk (Create)*



*Flowchart 1.7 Menampilkan Produk (Read)*



Flowchart 1.8 Mengubah Produk (Update)



Flowchart 1.9 Menghapus Produk (Delete)

- 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 menambah produk (Create)
  - Pengguna bisa melihat data produk (Read)
  - Pengguna bisa memgubah/memperbarui produk (Update)
  - Pengguna bisa menghapus produk (Delete)
- Jika memilih Tambah Produk (Create), Maka pengguna diminta untuk memasukkan ID produk berupa angka, Stock awal produk, dan Harga produk
- Jika memilih Lihat Produk (Read), Maka pengguna bisa melihat tampilan data produk
- Jika memilih Update Produk, Maka pengguna diarahkan untuk memilih produk mana yang akan diubah dengan memasukkan ID produk, lalu diubah dengan data yang baru
- Jika memilih Hapus produk (Delete), Maka pengguna diarahkan untuk memilih ID produk mana yang akan dihapus.
- Program diakhiri dengan menu Logout dan Keluar.

## 2. Deskripsi Singkat Program

Program ini adalah sistem pengelolaan stok sederhana dengan fitur login, register, dan manajemen produk. Data pengguna dan produk disimpan dalam dictionary. Saat dijalankan, pengguna bisa register akun baru atau login. Jika login sebagai admin, dapat menambah, melihat, mengubah, dan menghapus produk. Jika login sebagai user, hanya bisa melihat daftar produk. Fungsi clear() membersihkan layar, dan tampilkan\_produk() menampilkan daftar produk dalam format tabel. Program berjalan berulang hingga pengguna memilih keluar.

## 3. Source Code

### A. Struktur Data Awal

Menyimpan data pengguna (login) dan produk dalam bentuk **dictionary** agar mudah diakses dan dimodifikasi.

- users : untuk autentikasi login dan peran (admin/user)
- produk : untuk daftar barang dengan atribut 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. Fungsi [tampilkan\\_produk\(\)](#)

Menampilkan daftar produk dengan format tabel rapi. Digunakan oleh admin dan user untuk melihat stok barang.

```
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. Register dan Login

- Register : menambah akun baru ke dictionary users
- Login : memverifikasi username dan password, lalu menentukan menu sesuai role

```
# REGISTER if pilihan ==
"2":    if username in
users:
    print("X Username sudah digunakan.")
else:
    users.setdefault(username, {"password": password, "role": "user"})

# LOGIN elif pilihan
== "1":
    user = users.get(username)    if user
and user["password"] == password:
    role = user["role"]
```

### D. Menu Admin (CRUD Produk)

Admin memiliki hak penuh untuk mengelola produk (CRUD):

- Create : tambah produk baru
- Read : lihat daftar produk
- Update : ubah data produk
- Delete : hapus produk dari daftar

```
# Tambah produk
produk.update({idp: {"nama": nama, "stok": int(stok), "harga": int(harga)}})

# Update produk
produk[idp]["nama"] = nama_baru or produk[idp]["nama"]
produk[idp]["stok"] = int(stok_baru)
produk[idp]["harga"] = int(harga_baru)

# Hapus produk
dihapus = produk.pop(idp)
```

## E. Menu User

User hanya dapat melihat daftar produk, tidak bisa mengubah data.

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

## F. Keluar Program

Mengakhiri program dengan pesan penutup.

```
elif pilihan == "3":  
    print("👋 Terima kasih telah menggunakan sistem ini!")      break
```

#### 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. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Logout
Pilih menu: █
```

Output 4.4 Menu Admin

```
==== TAMBAH PRODUK ===
ID Produk : 004
Nama Produk : Odol
Stok Awal : 25
Harga : 15000
 Produk berhasil ditambahkan.
```

Tekan Enter untuk kembali...█

Output 4.5 Tambah Produk (Create)

```
==== DAFTAR PRODUK ===
ID      Nama          Stok      Harga
-----
001    Sabun         15        3000
002    Shampoo       10        7000
003    Sikat Gigi   20        4000
004    Odol          25       15000
-----
```

Tekan Enter untuk kembali...█

Output 4.6 Lihat Produk (Read)

```
==== UPDATE PRODUK ===
ID      Nama          Stok      Harga
-----
001    Sabun         15        3000
002    Shampoo       10        7000
003    Sikat Gigi   20        4000
004    Odol          25       15000
-----
Masukkan ID produk yang ingin diupdate: 003
Nama baru: Parfum
Stok baru: 30
Harga baru: 24000
 Data produk berhasil diperbarui!
```

Tekan Enter untuk kembali...█

Output 4.7 Update 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 PENGGUNA ===
1. Lihat Produk
2. Logout
Pilih menu: []
```

*Output 4.1 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 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

### 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:adityaftriansyah/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