

**LAPORAN PRAKTIKUM**  
**POSTTEST 8**  
**ALGORITMA PEMROGRAMAN DASAR**

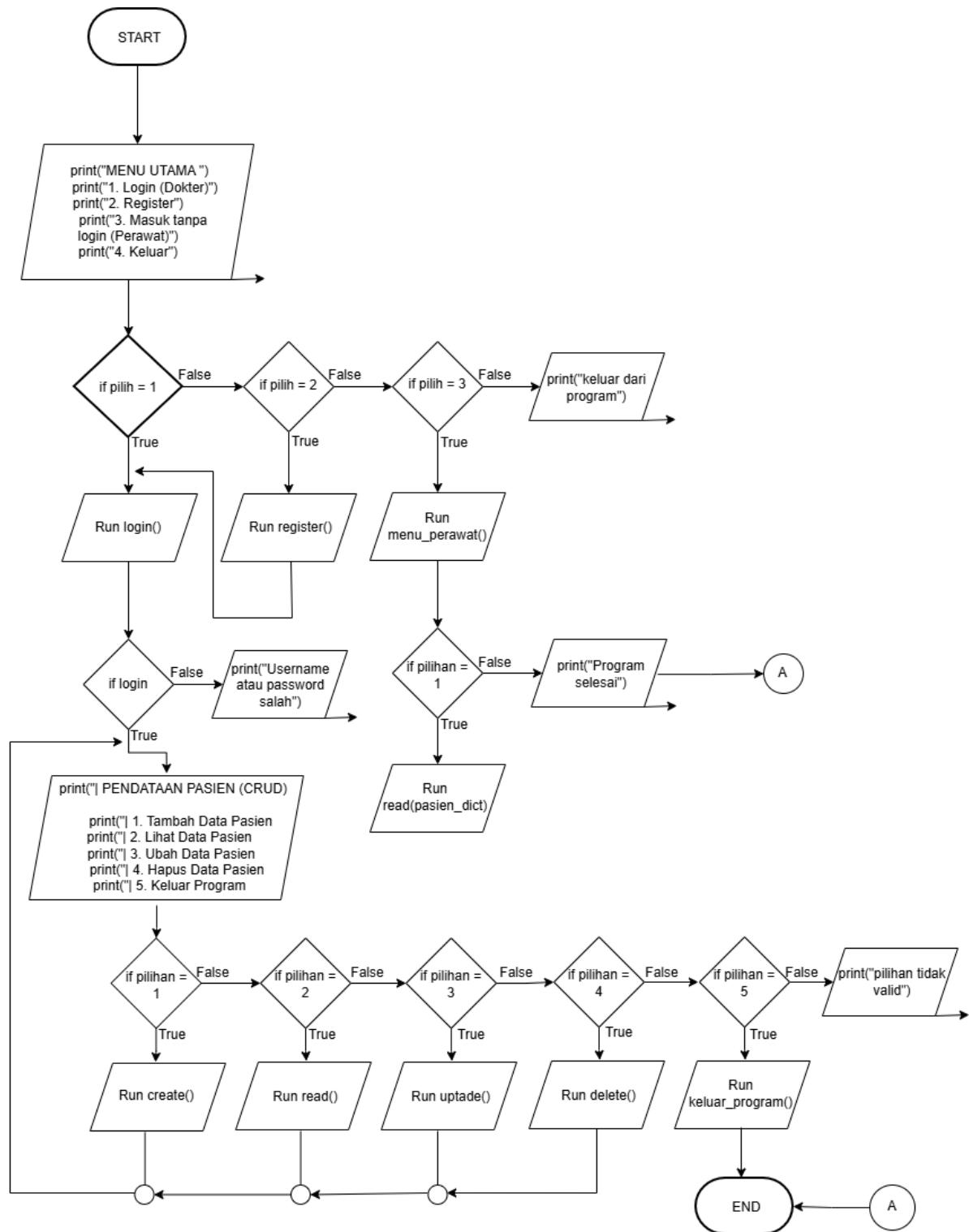


**Disusun oleh:**  
**Muhammad Fakih Nabal**  
**2509106009**  
**Kelas A1 '25**

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

**2025**

## 1. Flowchart



Pertama program akan menampilkan menu utama yang berisi login (opsi 1), register (opsi 2), masuk tanpa login (opsi 3), keluar (opsi 4). Jika user memilih opsi 1 maka, program akan menjalankan fungsi *login*, user bisa melakukan login jika sudah memiliki akun, jika tidak maka user tidak bisa melakukan login.

jika user memilih opsi ke 2 maka program akan menjalankan fungsi *register*, disini user dapat membuat akun baru di program pendataan pasien. Jika user memilih opsi ke 3 maka program akan menjalankan fungsi *perawat*, disini user hanya bisa melihat data pasien yang sudah dimasukkan oleh dokter. Dan jika user memilih opsi 4 maka program akan berhenti berjalan.

Di opsi 1 jika user berhasil melakukan login maka program akan masuk ke mode dokter dan memberikan user akses penuh ke program CRUD dengan menampilkan halaman yang berisi pilihan dari 1-5, jika user memilih opsi 1 maka user bisa melakukan penambahan data pasien, jika memilih opsi 2 maka user bisa melihat semua data pasien yang ada, jika memilih opsi 3, user bisa mengubah atau mengedit data pasien yang ada, jika memilih opsi 4, user bisa menghapus data pasien, jika memilih opsi 5 maka program akan berhenti.

## **2. Deskripsi Singkat Program**

Tujuan dari program ini adalah untuk memudahkan para tenaga medis untuk mengatur pendataan pasien.

## **3. Source Code**



```
1  from fungsi import menu_dokter, menu_perawat, keluar_program
2  from user import menu_awal
3  from data import input_pasien
4  from pasien import create, read, update, delete
5
6  # input data pasien
7  pasien_dict = input_pasien()
8
9  # login (dokter / perawat)
10 mode = menu_awal()
11
12 if mode == "dokter":
13     while True:
14         menu_dokter()
15         pilihan = input("Pilih menu: ")
16
17         if pilihan == "1":
18             create(pasien_dict)
19         elif pilihan == "2":
20             read(pasien_dict)
21         elif pilihan == "3":
22             update(pasien_dict)
23         elif pilihan == "4":
24             delete(pasien_dict)
25         elif pilihan == "5":
26             keluar_program()
27             break
28         else:
29             print("Pilihan tidak valid")
30
31 elif mode == "perawat":
32     while True:
33         menu_perawat()
34         pilihan = input("Pilih menu: ")
35
36         if pilihan == "1":
37             read(pasien_dict)
38         elif pilihan == "2":
39             print("Program selesai")
40             break
41         else:
42             print("Pilihan tidak valid.")
43
```



```
1 def menu_awal():
2     while True:
3         print("==== MENU UTAMA ===")
4         print("1. Login (Dokter)")
5         print("2. Register")
6         print("3. Masuk tanpa login (Perawat)")
7         print("4. Keluar")
8
9         pilih = input("Pilih menu: ")
10        if pilih == "1":
11            if login():
12                return "dokter"
13        elif pilih == "2":
14            register()
15        elif pilih == "3":
16            print("Masuk sebagai perawat")
17            return "perawat"
18        elif pilih == "4":
19            print("Keluar dari program")
20            exit()
21        else:
22            print("Pilihan tidak valid")
23
```



```
1 def register():
2     users = input_user()
3     print("\n==== REGISTRASI AKUN BARU ===")
4     username = input("Masukkan username: ")
5
6     if username in users:
7         print("Username sudah ada")
8         return
9
10    password = input("Masukkan password: ")
11    users[username] = password
12    simpan_user(users)
13    print("Akun berhasil dibuat")
14
```



```
1 def login():
2     users = input_user()
3     print("\n==== LOGIN ====")
4     username = input("Username: ")
5     password = input("Password: ")
6
7     if username in users and users[username] == password:
8         print("Login berhasil")
9         return True
10    else:
11        print("Username atau password salah")
12        return False
```

```
● ● ●
```

```
1 def create(pasien_dict):
2     try:
3         nama = input("Nama Pasien : ")
4         tinggi = int(input("Tinggi Badan : "))
5         bb = int(input("Berat Badan : "))
6     except ValueError:
7         print("Tinggi dan berat badan harus berupa angka")
8         return
9
10    print("Masukkan penyakit satu per satu (ketik 'selesai' jika sudah):")
11    penyakit_list = []
12    while True:
13        penyakit = input("Penyakit: ")
14        if penyakit.lower() == "selesai":
15            break
16        penyakit_list.append(penyakit)
17
18    beratIdeal, status = hitung_status(tinggi, bb)
19    pasien_dict[nama] = {
20        "tinggi": tinggi,
21        "berat": bb,
22        "ideal": beratIdeal,
23        "status": status,
24        "penyakit": penyakit_list
25    }
26
27    simpan_pasien(pasien_dict)
28    print(f"Data pasien '{nama}' berhasil ditambahkan")
29
```



```
1 def read(pasien_dict):
2     print("==== DATA PASIEN ===")
3     if not pasien_dict:
4         print("Belum ada pasien.")
5     else:
6         nomor = 1
7         for nama in pasien_dict:
8             data = pasien_dict[nama]
9             print(f"\n[{nomor}] Nama: {nama}")
10            print("Tinggi : ", data["tinggi"])
11            print("Berat : ", data["berat"])
12            print("Ideal : ", data["ideal"])
13            print("Status : ", data["status"])
14            print("Penyakit:")
15            for p in data["penyakit"]:
16                print(" -", p)
17            nomor += 1
18        print("-----")
```

```
1      print("Belum ada pasien.")
2  else:
3      nomor = 1
4      for nama in pasien_dict:
5          data = pasien_dict[nama]
6          print(f"\n{n[1]} Nama: {nama}")
7          print("Tinggi : ", data["tinggi"])
8          print("Berat : ", data["berat"])
9          print("Ideal : ", data["ideal"])
10         print("Status : ", data["status"])
11         print("Penyakit:")
12         for p in data["penyakit"]:
13             print(" -", p)
14         nomor += 1
15     print("-----")
16
17
18
19 def update(pasien_dict):
20     if not pasien_dict:
21         print("Belum ada data pasien.")
22         return
23
24     print("== DAFTAR PASIEN ==")
25     nama_list = list(pasien_dict.keys())
26     nomor = 1
27     for nama in nama_list:
28         print(f"{nomor}. {nama}")
29         nomor += 1
30
31     try:
32         pilih_nomor = int(input("Pilih nomor pasien yang ingin diubah: ")) - 1
33         if pilih_nomor < 0 or pilih_nomor >= len(nama_list):
34             print("Nomor tidak valid.")
35             return
36
37         nama = nama_list[pilih_nomor]
38         data = pasien_dict[nama]
39
40         print(f"Mengubah data pasien: {nama}")
41         nama_baru = input("Nama baru : ") or nama
42         tinggi_baru = input("Tinggi baru : ")
43         berat_baru = input("Berat baru : ")
44
45         if tinggi_baru:
46             tinggi_baru = int(tinggi_baru)
47         else:
48             tinggi_baru = data["tinggi"]
49
50         if berat_baru:
51             berat_baru = int(berat_baru)
52         else:
53             berat_baru = data["berat"]
54
55         print("Masukkan penyakit baru satu per satu (ketik 'selesai' jika tidak ingin menambah):")
56         penyakit_baru = []
57         while True:
58             p = input("Penyakit: ")
59             if p.lower() == "selesai":
60                 break
61             penyakit_baru.append(p)
62         if not penyakit_baru:
63             penyakit_baru = data["penyakit"]
64
65         beratIdeal, status = hitung_status(tinggi_baru, berat_baru)
66         del pasien_dict[nama]
67         pasien_dict[nama_baru] = {
68             "tinggi": tinggi_baru,
69             "berat": berat_baru,
70             "ideal": beratIdeal,
71             "status": status,
72             "penyakit": penyakit_baru
73         }
74
75         simpan_pasien(pasien_dict)
76         print("Data pasien berhasil diperbarui")
77
```

```
1  def delete(pasien_dict):
2      if not pasien_dict:
3          print("Belum ada data pasien.")
4          return
5
6      print("== DAFTAR PASIEN ==")
7      nama_list = list(pasien_dict.keys())
8      nomor = 1
9      for nama in nama_list:
10         print(f"{nomor}. {nama}")
11         nomor += 1
12
13     try:
14         pilih_nomor = int(input("Pilih nomor pasien yang ingin dihapus: ")) - 1
15         if pilih_nomor < 0 or pilih_nomor >= len(nama_list):
16             print("Nomor tidak valid")
17             return
18
19         nama = nama_list[pilih_nomor]
20         del pasien_dict[nama]
21         delete(pasien_dict)
22         print(f"Data pasien '{nama}' berhasil dihapus")
23
24
25     except ValueError:
26         print("Input tidak valid, masukkan angka")
```

```
● ● ●  
1 elif mode == "perawat":  
2     while True:  
3         menu_perawat()  
4         pilihan = input("Pilih menu: ")  
5  
6         if pilihan == "1":  
7             read(pasien_dict)  
8         elif pilihan == "2":  
9             print("Program selesai")  
10        break  
11    else:  
12        print("Pilihan tidak valid.")  
13
```

#### 4. Hasil Output

```
==== MENU UTAMA ====  
1. Login (Dokter)  
2. Register  
3. Masuk tanpa login (Perawat)  
4. Keluar
```

```
==== REGISTRASI AKUN BARU ====  
Masukkan username: abdi  
Masukkan password: 007  
Akun berhasil dibuat
```

```
==== LOGIN ====
Username: fakih
Password: 009
Login berhasil
```

```
=====
|          PENDATAAN PASIEN (CRUD)          |
=====
| 1. Tambah Data Pasien
| 2. Lihat Data Pasien
| 3. Ubah Data Pasien
| 4. Hapus Data Pasien
| 5. Keluar Program
=====
Pilih menu: 1
nama Pasien    : budi
tinggi Badan   : 173
berat Badan    : 68
masukkan penyakit satu per satu ketik 'selesai' jika sudah
Penyakit: pusing
Penyakit: pilek
Penyakit: batuk
Penyakit: selesai
data pasien 'budi' berhasil ditambahkan
```

#### Hasil 4

```
-----
|          DATA PASIEN          |
-----

[1] budi
Tinggi Badan : 173
Berat Badan  : 68
Berat Ideal   : 73
Status        : Berat badan ideal
Penyakit      :
  - pusing
  - pilek
  - batuk
-----
```

```
pilih menu: 3
DAFTAR PASTEN:
1. nama
masukkan nomor pasien yang ingin diubah: 1
masukkan data baru (kosongkan jika tidak ingin diubah):
Nama baru : wahyu
Tinggi baru :
Berat baru :
Masukkan penyakit baru satu per satu (ketik 'selesai' jika tidak ingin menambah):
Penyakit: selesai
data pasien berhasil diperbarui
```

```
pilih menu: 4
DAFTAR PASIEN:
1. wahyu
masukkan nomor pasien yang ingin dihapus: 1
data pasien 'wahyu' berhasil dihapus
```

```
=====
|          PENDATAAN PASIEN (CRUD)          |
=====
| 1. Tambah Data Pasien
| 2. Lihat Data Pasien
| 3. Ubah Data Pasien
| 4. Hapus Data Pasien
| 5. Keluar Program
=====
Pilih menu: 5
Pendataan pasien selesai
```

## 5. Langkah-langkah GIT

### 5.1 GIT Init

```
PS C:\Users\Hasmiyah\.vscode\praktikum-apd\post-test\post-test-apd-2> git init
Initialized empty Git repository in C:/Users/Hasmiyah/.vscode/praktikum-apd/post-test/post-test-apd-2/.git/
PS C:\Users\Hasmiyah\.vscode\praktikum-apd\post-test\post-test-apd-2>
```

Digunakan untuk menginisiasi repository Git ke lokal.

### 5.2 GIT Add

```
C:\Users\Hasmiyah\.vscode\praktikum-apd\post-test>git add .
```

Digunakan untuk menambahkan perubahan file ke staging area.

### 5.3 GIT Commit

```
C:\Users\Hasmiyah\.vscode\praktikum-apd\post-test>git commit -m "posttest 6"
[main 880f3b9] posttest 6
 4 files changed, 171 insertions(+), 38 deletions(-)
 delete mode 100644 post-test/post-test-apd-2/A1'25/pertemuan-3/pertemuan_3.py
 create mode 100644 post-test/posttest_apd_5/tempCodeRunnerFile.py
 create mode 100644 post-test/posttest_apd_6/2509106009_Muhammad_Fakih_Nabal_PT_6.py
```

Digunakan untuk menyimpan perubahan file yang sudah dimasukkan ke staging area.

### 5.4 GIT Remote

```
C:\Users\Hasmiyah\.vscode\praktikum-apd\post-test>git remote add origin https://github.com/fakih004/praktikum-apd.git
error: remote origin already exists.
```

Digunakan untuk menghubungkan repository lokal ke repository lain.

### 5.5 GIT Push

```
PS C:\Users\Hasmiyah\.vscode\praktikum-apd\post-test\post-test-apd-2> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 390.74 KiB | 14.47 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/fakih004/praktikum-apd.git
  6ce701c..860c336  main -> main
```

Digunakan untuk mengirim commit repository lokal ke repository remote.