

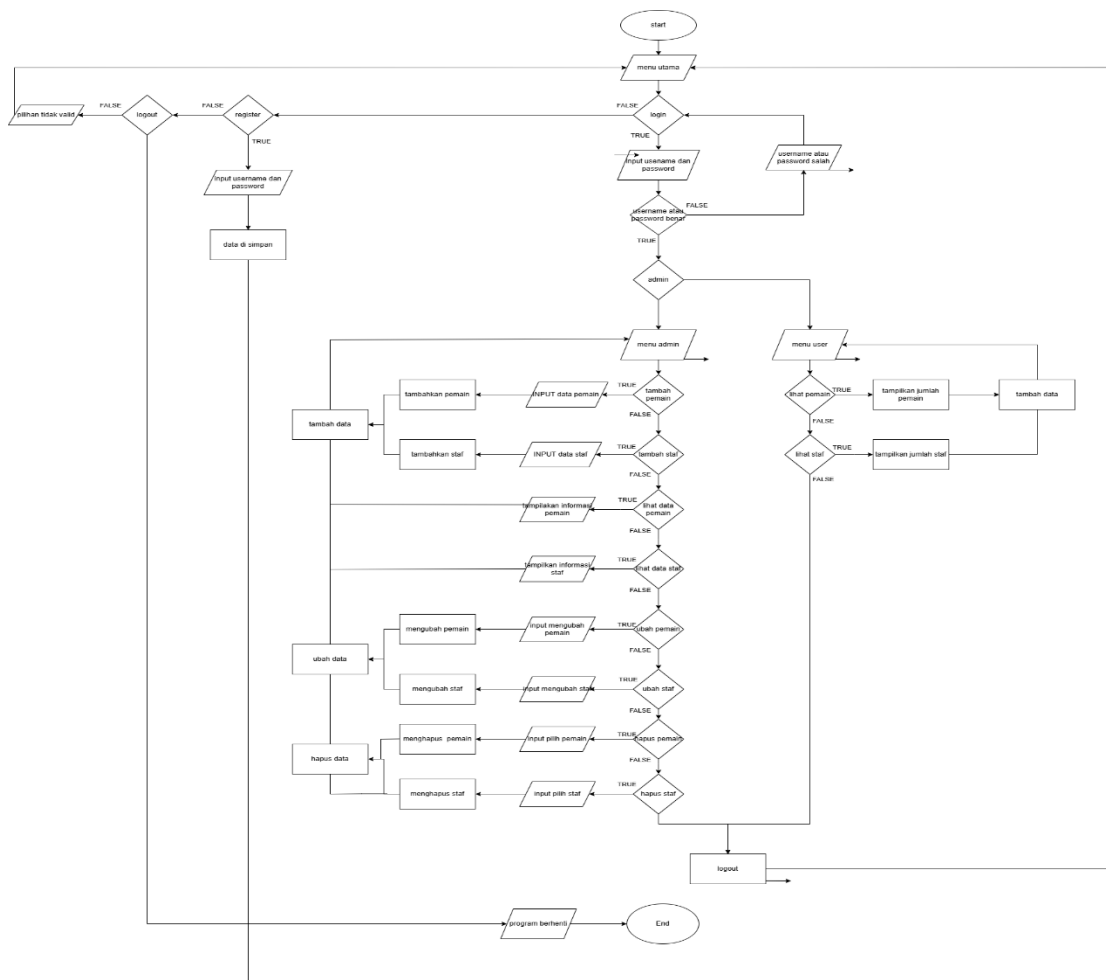
LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Michael Ariel V (2509106124)
INFORMATIKA C2 '25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Start

- Titik awal pengguna masuk ke sistem.

Login atau Register

- Pengguna memilih antara:
- **Login:** Masukkan username dan password.
- **Register:** Buat akun baru dengan username dan password.

Validasi Input

- Sistem mengecek apakah input sudah valid:
- Jika **tidak valid**, kembali ke input.
- Jika **valid**, lanjut ke proses berikutnya.

Verifikasi Akun

- Sistem mengecek apakah username dan password cocok dengan data yang tersimpan:
- Jika **tidak cocok**, kembali ke login.
- Jika **cocok**, lanjut ke menu utama.

Menu Utama Berdasarkan Role

- *Sistem mengecek apakah pengguna adalah **admin** atau **user biasa**:*
- ***Admin** mendapat akses penuh ke fitur manajemen.*

Fitur Admin

Admin bisa:

- ***Tambah pemain***
- ***Tentukan jumlah pemain***
- ***Tambah staf***
- ***Ubah atau hapus data pemain/staf***
- ***Pilih data pemain yang akan digunakan:***
- *Gunakan data lama*
- *Gunakan data baru*

Fitur User

User hanya bisa:

- *Melihat daftar pemain*
- *Melihat daftar staf*

Logout atau Keluar

- *Pengguna bisa logout dan kembali ke menu awal.*
- *Jika memilih keluar, sistem menampilkan "**Program berhenti**" dan menuju **End**.*

2.Deskripsi Singkat Program

*Memfasilitasi pengelolaan data pemain dan staf dalam sebuah tim bola voli, serta menyediakan sistem login dan registrasi untuk membedakan hak akses antara **admin** dan **user biasa**.*

2. Source Code Source Code:

Auth.py

```
from utils import clear

users = [["admin", "admin123", "admin"]]

def register():
    clear()
    print("=== REGISTRASI USER BARU ===")
    username = input("Username: ")
    password = input("Password: ")

    for user in users:
        if user[0] == username:
            print("Username sudah digunakan!")
            return

    users.append([username, password, "user"])
    print("Registrasi berhasil! Silakan login.")

def login():
    clear()
    print("=== LOGIN ===")
    username = input("Username: ")
    password = input("Password: ")

    for user in users:
        if user[0] == username and user[1] == password:
            print(f"Login berhasil sebagai {user[2]}!")
            return user[2]

    print("Login gagal. Username atau password salah.")
    return None
```

Main.py

```
from auth import login, register
from pemain import tambah_pemain, tampilkan_pemain, ubah_pemain,
hapus_pemain
from staf import tambah_staf, tampilkan_staf, ubah_staf, hapus_staf

def menu_admin():
    while True:
        print("""
===== MENU ADMIN TIM VOLI =====
1. Tambah Pemain
2. Tambah Staf
3. Lihat Pemain
4. Lihat Staf
5. Ubah Pemain
6. Ubah Staf
""")
```

```

7. Hapus Pemain
8. Hapus Staf
9. Logout
=====
"""
    pilihan = input("Pilih menu: ")
    if pilihan == "1":
        tambah_pemain()
    elif pilihan == "2":
        tambah_staf()
    elif pilihan == "3":
        tampilkan_pemain()
    elif pilihan == "4":
        tampilkan_staf()
    elif pilihan == "5":
        ubah_pemain()
    elif pilihan == "6":
        ubah_staf()
    elif pilihan == "7":
        hapus_pemain()
    elif pilihan == "8":
        hapus_staf()
    elif pilihan == "9":
        break
    else:
        print("Pilihan tidak valid!")

def menu_user():
    while True:
        print("""
===== MENU USER TIM VOLI =====
1. Lihat Pemain
2. Lihat Staf
3. Logout
=====
""")
        pilihan = input("Pilih menu: ")
        if pilihan == "1":
            tampilkan_pemain()
        elif pilihan == "2":
            tampilkan_staf()
        elif pilihan == "3":
            break
        else:
            print("Pilihan tidak valid!")

while True:
    print("""
=== SELAMAT DATANG DI SISTEM TIM VOLI ===
1. Login
2. Register
0. Keluar
""")
    pilih = input("Pilih menu: ")

```

```

if pilih == "1":
    role = login()
    if role == "admin":
        menu_admin()
    elif role == "user":
        menu_user()
elif pilih == "2":
    register()
elif pilih == "0":
    print("Program selesai. Terima kasih!")
    break
else:
    print("Menu tidak valid!")

```

Pemain.py

```

from utils import clear, input_angka
from prettytable import PrettyTable

pemain = [
    ["bang gilbert/Lian", "Spiker", 7, "081234567890", "Inti"],
    ["mikeng", "Tosser", 3, "089876543210", "Inti"],
    ["asoy", "Libero", 31, "082233445566", "Inti"],
    ["erpang", "Tosser", 20, "082300942134", "Cadangan"]
]

def tambah_pemain():
    clear()
    print("--- Tambah Pemain Baru ---")
    nama = input("Nama pemain: ")
    posisi = input("Posisi (Tosser/Spiker/Libero): ")
    nomor = input_angka("Nomor punggung: ")
    if nomor is None:
        return
    kontak = input("Nomor kontak: ")
    status = input("Status (Inti/Cadangan): ")
    pemain.append([nama, posisi, nomor, kontak, status])
    print(f"Pemain {nama} berhasil ditambahkan!")

def tampilkan_pemain():
    clear()
    print("=== Daftar Pemain ===")
    if not pemain:
        print("Belum ada pemain.")
    else:
        table = PrettyTable(["Nama", "Posisi", "No", "Kontak", "Status"])

```

```

        for p in pemain:
            table.add_row(p)
        print(table)

def ubah_pemain():
    tampilkan_pemain()
    if not pemain:
        return
    nama_cari = input("\nMasukkan nama pemain yang ingin diubah: ")
    for p in pemain:
        if p[0].lower() == nama_cari.lower():
            print("Kosongkan untuk tidak mengubah.")
            nama = input(f>Nama ({p[0]}): ") or p[0]
            posisi = input(f>Posisi ({p[1]}): ") or p[1]
            nomor = input(f>No Punggung ({p[2]}): ")
            nomor = int(nomor) if nomor.isdigit() else p[2]
            kontak = input(f>Kontak ({p[3]}): ") or p[3]
            status = input(f>Status ({p[4]}): ") or p[4]
            p[:] = [nama, posisi, nomor, kontak, status]
            print("Data pemain berhasil diubah!")
            return
    print("Pemain tidak ditemukan!")

def hapus_pemain():
    tampilkan_pemain()
    if not pemain:
        return
    nama_cari = input("\nMasukkan nama pemain yang ingin dihapus: ")
    for p in pemain:
        if p[0].lower() == nama_cari.lower():
            pemain.remove(p)
            print(f>Pemain {p[0]} berhasil dihapus.")
            return
    print("Pemain tidak ditemukan!")

```

staf.py

```

from utils import clear
from prettytable import PrettyTable

staf = [
    ["Pak kiel", "Pelatih", "082112223333"],
    ["Bu meimei", "Manajer", "081355557777"],
    ["Pak attar", "Medis", "081282864935"]
]

def tambah_staf():
    clear()
    print("--- Tambah Staf Baru ---")
    nama = input("Nama staf: ")

```

```

jabatan = input("Jabatan: ")
kontak = input("Nomor kontak: ")
staf.append([nama, jabatan, kontak])
print(f"Staf {nama} berhasil ditambahkan!")

def tampilkan_staf():
    clear()
    print("=== Daftar Staf ===")
    if not staf:
        print("Belum ada staf.")
    else:
        table = PrettyTable(["Nama", "Jabatan", "Kontak"])
        for s in staf:
            table.add_row(s)
        print(table)

def ubah_staf():
    tampilkan_staf()
    if not staf:
        return
    nama_cari = input("\nMasukkan nama staf yang ingin diubah: ")
    for s in staf:
        if s[0].lower() == nama_cari.lower():
            print("Kosongkan untuk tidak mengubah.")
            nama = input(f>Nama ({s[0]}): ") or s[0]
            jabatan = input(f>Jabatan ({s[1]}): ") or s[1]
            kontak = input(f>Kontak ({s[2]}): ") or s[2]
            s[:] = [nama, jabatan, kontak]
            print("Data staf berhasil diubah!")
            return
    print("Staf tidak ditemukan!")

def hapus_staf():
    tampilkan_staf()
    if not staf:
        return
    nama_cari = input("\nMasukkan nama staf yang ingin dihapus: ")
    for s in staf:
        if s[0].lower() == nama_cari.lower():
            staf.remove(s)
            print(f"Staf {s[0]} berhasil dihapus.")
            return
    print("Staf tidak ditemukan!")

```


utils.py

```
import os

def clear():
    os.system("cls" if os.name == "nt" else "clear")

def input_angka(prompt):
    angka = input(prompt)
    if not angka.isdigit():
        print("Input harus berupa angka!")
        return None
    return int(angka)
```

4. Hasil Output

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

===== MENU ADMIN TIM VOLI =====
1. Tambah Pemain
2. Tambah Staf
3. Lihat Pemain
4. Lihat Staf
5. Ubah Pemain
6. Ubah Staf
7. Hapus Pemain
8. Hapus Staf
9. Logout
=====

Pilih menu: █
```

```
=== LOGIN ===
Username: michael
Password: michael123
Login berhasil sebagai user!

===== MENU USER TIM VOLI =====
1. Lihat Pemain
2. Lihat Staf
3. Logout
=====

Pilih menu: █
```

5. Langkah-langkah GIT

5.1 GIT Add

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

jangan lupa pakai spasi dan dot (.)

5.2 GIT Commit

```
PS C:\praktikum-apd> git commit -m "PTS8"
[main 6b5f4b5] PTS8
10 files changed, 310 insertions(+)
create mode 100644 kelas/Pertemuan-8.py
create mode 100644 kelas/pertemuan-7/autentikasi.py
create mode 100644 kelas/pertemuan-7/create.py
create mode 100644 kelas/pertemuan-7/main.py
create mode 100644 post-test/post-test-apd-7/2509106124_Michael Ariel Valentino PT 7APD.pdf
create mode 100644 post-test/post-test-apd-8/2509106124-Michael Ariel Valentino/auth.py
create mode 100644 post-test/post-test-apd-8/2509106124-Michael Ariel Valentino/main.py
create mode 100644 post-test/post-test-apd-8/2509106124-Michael Ariel Valentino/pemain.py
create mode 100644 post-test/post-test-apd-8/2509106124-Michael Ariel Valentino/staf.py
create mode 100644 post-test/post-test-apd-8/2509106124-Michael Ariel Valentino/Utils.py
```

5.3 GIT Push

```
PS C:\praktikum-apd> git push origin main
Enumerating objects: 22, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 16 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (16/16), 4.15 KiB | 472.00 KiB/s, done.
Total 16 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/ikelm520-boop/pratikum-apd/
 88b597f..7a48182 main -> main
```