

LAPORAN PRAKTIKUM POSTTEST

6

ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:

M Dirga Athayasa(2509106125)

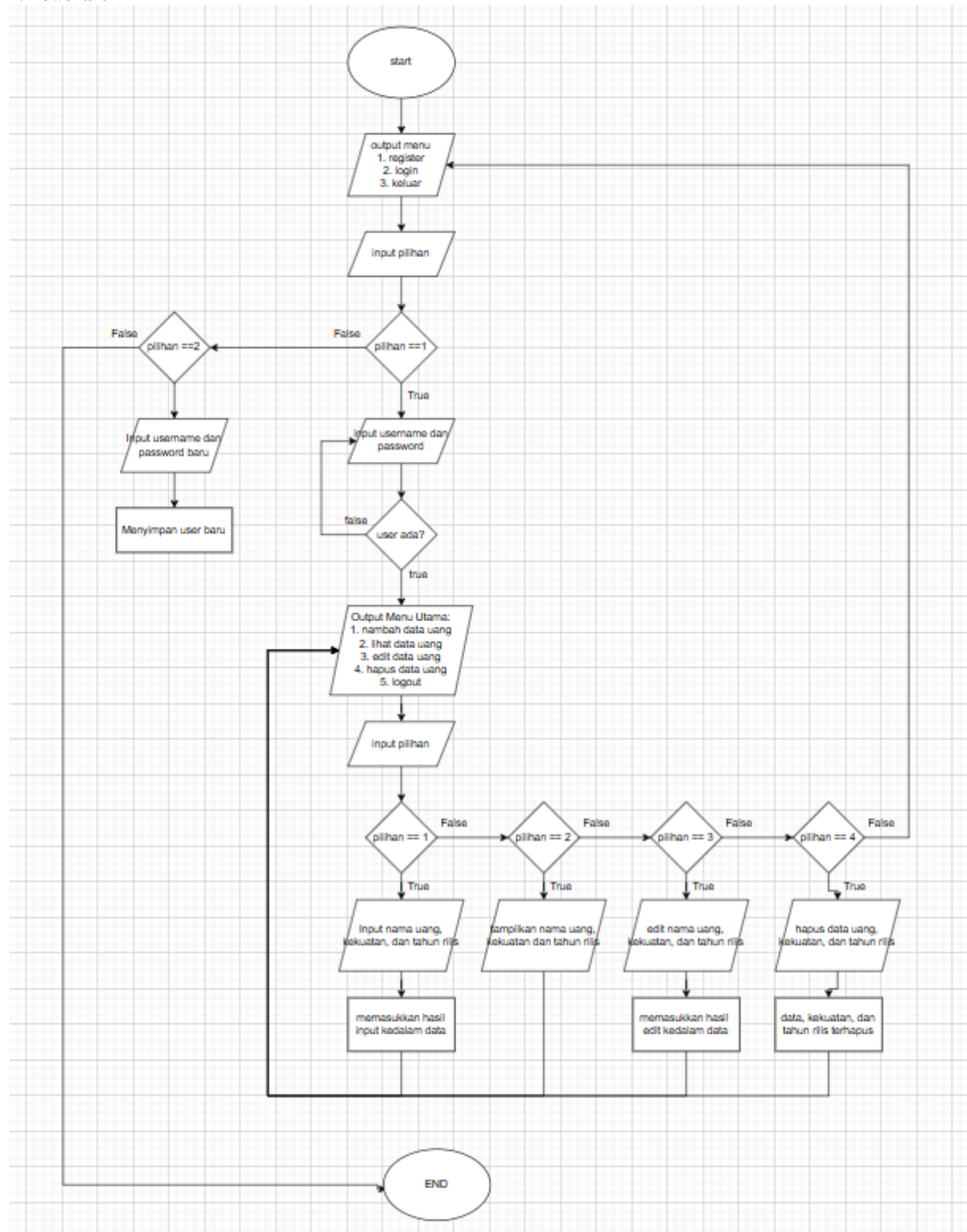
Kelas (C2 '25)

PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

1.Flowchart



2. Deskripsi Singkat program

Program ini memungkinkan pengguna untuk mendaftar (register), login, lalu mengelola data uang melalui beberapa fitur utama. Setelah login, pengguna dapat menambah, melihat, mengubah, dan menghapus data tentang jenis uang yang disimpan dalam program.

Fungsi Utama:

1. Register – Menambahkan akun baru ke daftar dataAkun.
2. Login – Memverifikasi username dan password agar bisa masuk ke menu keuangan.
3. Tambah Data Uang – Memasukkan nama, kekuatan, dan tahun rilis uang baru.
4. Tampilkan Data Uang – Menampilkan seluruh data uang yang tersimpan.
5. Ubah Data Uang – Mengedit data uang berdasarkan pilihan pengguna.
6. Hapus Data Uang – Menghapus data uang tertentu dari daftar.
7. Logout / Keluar – Kembali ke menu utama atau menutup program.

Kesimpulan

Program ini adalah contoh penerapan dasar struktur data list, perulangan (looping), dan percabangan (if-else) dalam Python untuk membuat sistem login dan CRUD (Create, Read, Update, Delete) sederhana.

3. Source Code

```

#      ===      VARIABEL      GLOBAL      ===
dataAkun      =      {
    "admin":      "123",
    "dirga":      "115"
}

dataUang      =      [
    {"nama":      "Rupiah",      "kekuatan":      "Lemah",      "tahun":      2020}
]

statusLogin = None # global untuk menyimpan siapa yang sedang login

#      ===      FUNGSI      TANPA      PARAMETER      ===
def      tampilkan_data():
    """Menampilkan      semua      data      uang"""
    if      not      dataUang:
        print("Belum      ada      data      uang.")
    else:
        for      i,      uang      in      enumerate(dataUang,      start=1):
            print(f"
            |      Data      uang      ke-{i}
            |      Nama      Uang      :      {uang['nama']}
            |      Kekuatan      Uang      :      {uang['kekuatan']}
            |      Tahun      Rilis      :      {uang['tahun']}
            |      -----")

#      ===      FUNGSI      DENGAN      PARAMETER      ===
def      tambah_data(nama,      kekuatan,      tahun):
    """Menambahkan      data      uang      baru"""
    dataUang.append({"nama":      nama,      "kekuatan":      kekuatan,      "tahun":
tahun})
    print(f"Data      uang      '{nama}'      berhasil      ditambahkan!")

#      ===      PROSEDUR      (tidak      mengembalikan      nilai)      ===
def      ubah_data():
    """Prosedur      untuk      mengubah      data"""

```

```

try:
    for i, uang in enumerate(dataUang, start=1):
        print(f"{i}. {uang['nama']}")
    pilihUang = int(input("Pilih nomor data yang ingin diupdate:
"))

    if 0 <= pilihUang < len(dataUang):
        nama_baru = input("Masukkan nama uang baru: ")
        kekuatan_baru = input("Masukkan kekuatan uang baru: ")
        tahun_baru = input("Masukkan tahun rilis baru: ")

        dataUang[pilihUang] = {
            "nama": nama_baru,
            "kekuatan": kekuatan_baru,
            "tahun": tahun_baru
        }
        print("Data berhasil diupdate!")
    else:
        print("Nomor data tidak valid!")
except ValueError:
    print("Input harus berupa angka!")

def hapus_data():
    """Prosedur untuk menghapus data"""
    try:
        for i, uang in enumerate(dataUang, start=1):
            print(f"{i}. {uang['nama']}")
        inputHapus = int(input("Pilih nomor data yang ingin dihapus:
"))

        if 0 <= inputHapus < len(dataUang):
            hapus = dataUang.pop(inputHapus)
            print(f"Data '{hapus['nama']}' berhasil dihapus.")
        else:
            print("Nomor data tidak valid!")
    except ValueError:
        print("Input harus berupa angka!")

# === FUNGSI REKURSIF ===
def menu_keuangan():
    """Fungsi rekursif untuk menampilkan menu keuangan berulang"""

```

```

    print("""
===          KEUANGAN          DUNIA          ===
-----
1.          TAMBAH          DATA          UANG
2.          TAMPILKAN          DATA          UANG
3.          UBAH          DATA          UANG
4.          HAPUS          DATA          UANG
5.                                     LOGOUT
=====
""")
    pilih          =          input("PILIH:          ")

    #          Variabel          lokal
    nama_uang          =          kekuatan_uang          =          tahun_rilis          =          ""

    if          pilih          ==          "1":
        print("==          TAMBAH          DATA          ==")
        nama_uang          =          input("Nama          Uang:          ")
        kekuatan_uang          =          input("Kekuatan Uang (kuat/sedang/lemah):
")
        tahun_rilis          =          input("Tahun          Rilis:          ")
        tambah_data(nama_uang,          kekuatan_uang,          tahun_rilis)

    elif          pilih          ==          "2":
        tampilkan_data()

    elif          pilih          ==          "3":
        ubah_data()

    elif          pilih          ==          "4":
        hapus_data()

    elif          pilih          ==          "5":
        print("Logout...\n")
        return          #          menghentikan          rekursi

    else:
        print("Input          salah!          hanya          bisa          (1-5)")

    #          Rekursif          kembali          ke          menu
    menu_keuangan()

```

```

#          ===          PROGRAM          UTAMA          ===
while True:
    print("""
    -----
    - 1. Register -
    - 2. Login -
    - 3. KELUAR -
    -----
    """)
    inputMenuUtama = input("Pilih (1-3): ")

    if inputMenuUtama == "1":
        print("=== REGISTRASI ===")
        registUsername = input("Input username : ")
        registPassword = input("Input password : ")

        if registUsername in dataAkun:
            print("Username sudah terdaftar!")
        else:
            dataAkun[registUsername] = registPassword
            print("Berhasil register!")

    elif inputMenuUtama == "2":
        print("=== LOGIN ===")
        inputUsername = input("Username anda : ")
        inputPassword = input("Password anda : ")

        if inputUsername in dataAkun and dataAkun[inputUsername] == inputPassword:
            statusLogin = inputUsername
            print(f"\nBerhasil Login sebagai '{statusLogin}'!")

            # Multiuser: admin punya akses penuh
            if statusLogin == "admin":
                print("Selamat datang, ADMIN.")
                menu_keuangan()
            else:
                print("Selamat datang, Pengguna biasa.")
                # Pengguna hanya bisa lihat dan tambah data
                while True:
                    print("""
                    === MENU          PENGGUNA          ===
                    -----

```

```

1.          TAMBAH          DATA          UANG
2.          LIHAT          DATA          UANG
3.          LOGOUT
=====
""" )
        pilih = input("PILIH: ")

        if pilih == "1":
            nama = input("Nama Uang: ")
            kekuatan = input("Kekuatan Uang: ")
            tahun = input("Tahun: ")
            tambah_data(nama, kekuatan, tahun)

        elif pilih == "2":
            tampilkan_data()

        elif pilih == "3":
            print("Logout...")
            break
        else:
            print("Input salah!")

    else:
        print("Username atau password salah!")

elif inputMenuUtama == "3":
    print("Program Berhenti.")
    break

else:
    print("Input salah! hanya bisa (1-3)")

```


4. Hasil Output

```
==== KEUANGAN DUNIA ====
-----
1. TAMBAH DATA UANG
2. TAMPILKAN DATA UANG
3. UBAH DATA UANG
4. HAPUS DATA UANG
5. LOGOUT
=====
```

Gambar 1.1 Hasil output menu utama

```
PILIH: 1
== TAMBAH DATA ==
Nama Uang: dolar
Kekuatan Uang (kuat/sedang/lemah): kuat
Tahun Rilis: 2000
Data uang 'dolar' berhasil ditambahkan!
```

Gambar 1.2 Menampilkan CRUD Create

```
-----  
| Data uang ke-1  
| Nama Uang      : Rupiah  
| Kekuatan Uang  : Lemah  
| Tahun Rilis    : 2020  
-----  
  
-----  
| Data uang ke-2  
| Nama Uang      : dolar  
| Kekuatan Uang  : kuat  
| Tahun Rilis    : 2000  
-----
```

Gambar 1.3 Menampilkan CRUD Read

```
PILIH: 3  
== UPDATE DATA UANG ==  
1. Rupiah  
2. dolar  
Pilih nomor data yang ingin diupdate: 1  
Masukkan nama uang baru: ringgit  
Masukkan kekuatan uang baru: sedang  
Masukkan tahun rilis baru: 2015  
Data berhasil diupdate!
```

Gambar 1.4 Menampilkan CRUD Update

```
PILIH: 4  
== HAPUS DATA ==  
1. ringgit  
2. dolar  
Pilih nomor data yang ingin dihapus: 1  
Data 'ringgit' berhasil dihapus.
```

Gambar 1.5 Menampilkan CRUD Delete

5. Langkah Git

5.2 GIT Add

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

Gambar 1.6 git add .

Git add adalah perintah Git yang digunakan untuk menambahkan perubahan ke area staging sebelum melakukan commit

5.3 GIT Commit

```
PS D:\praktikum-apd> git commit -m "upload pt7"
[main 895f9dc] upload pt7
1 file changed, 181 insertions(+)
create mode 100644 POSTTEST-7/posttest7.py
```

Gambar 1.7 git commit -m ""

Git commit adalah perintah Git untuk menyimpan perubahan yang telah di-stage ke dalam riwayat versi proyek. Ini menciptakan snapshot permanen dari file yang ditandai, sehingga kamu bisa melacak dan mengembalikan versi sebelumnya jika diperlukan.

5.5 GIT Push

```
fatal: a branch named 'main' already exists
PS D:\praktikum-apd> git push --set-upstream origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 3.40 KiB | 3.40 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/dirgaathayasa24-coder/praktikum-apd.git
 d227019..666375d main -> main
branch 'main' set up to track 'origin/main'.
```

gambar 1.8 git push

Git push adalah perintah Git yang digunakan untuk mengirim commit dari repositori lokal ke repositori remote seperti GitHub, GitLab, atau Bitbucket. Ini memungkinkan orang lain melihat dan mengakses perubahan yang telah kamu buat.

