

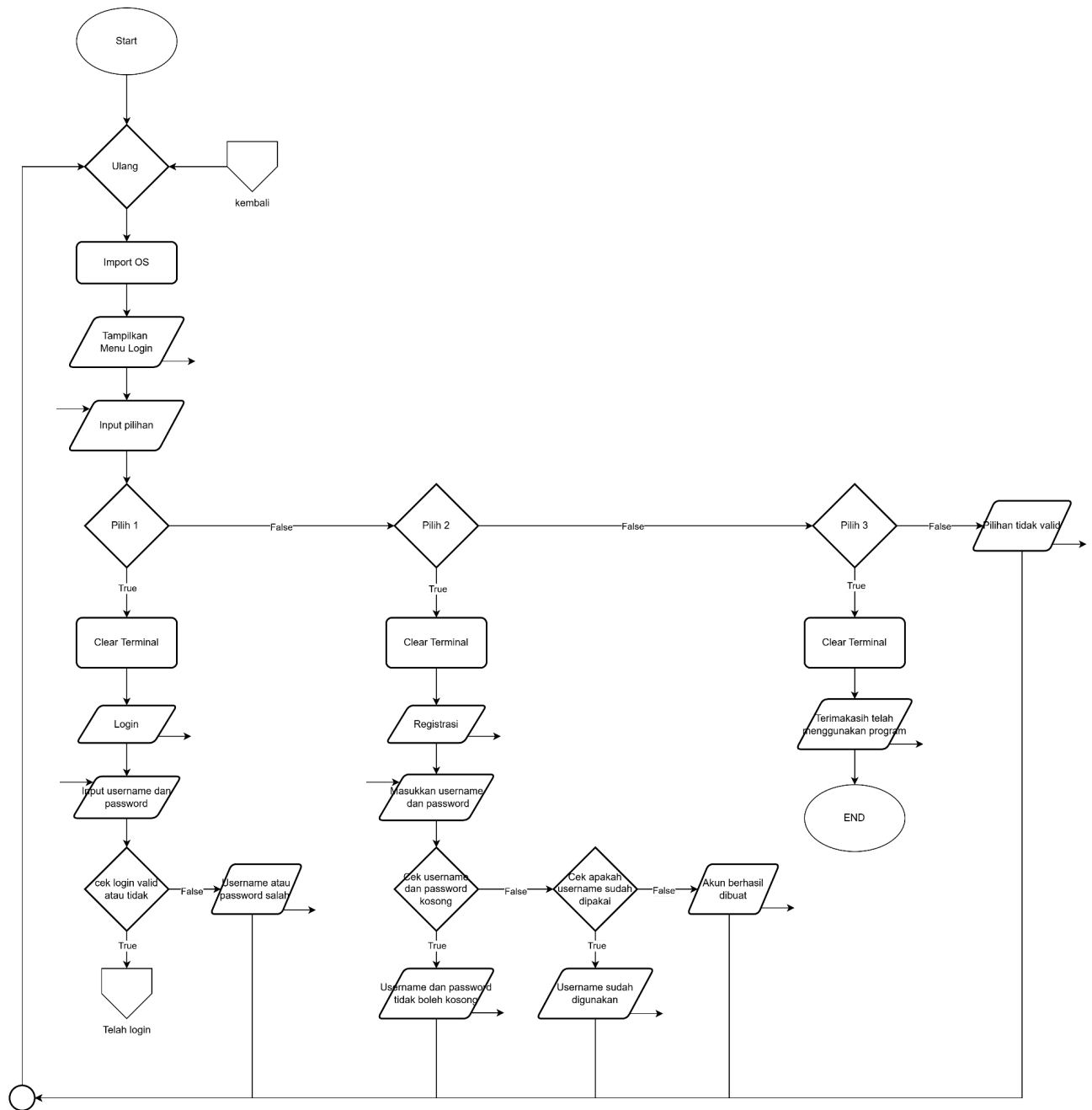
**LAPORAN PRAKTIKUM**  
**POSTTEST 7**  
**ALGORITMA PEMROGRAMAN DASAR**



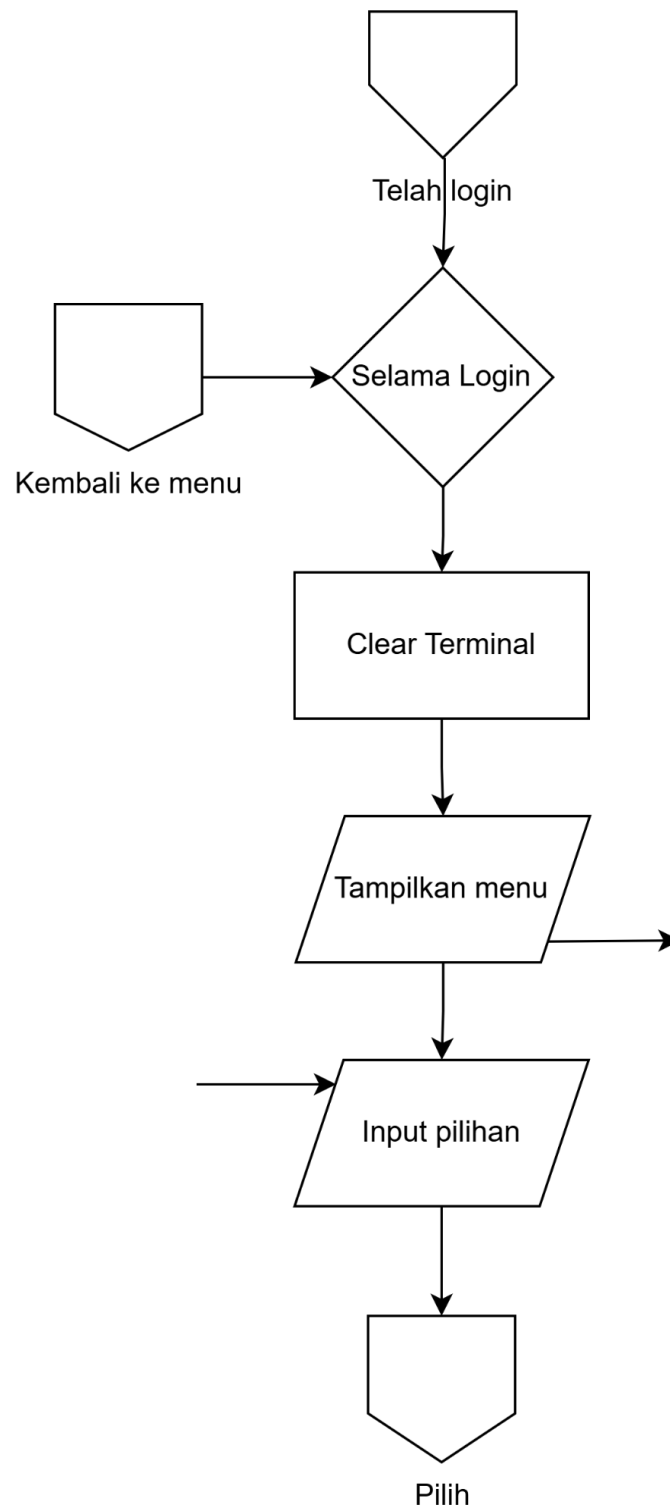
**Disusun oleh:**  
**Pirlo Syabila Hafuza (2509106008)**  
**Kelas (A1 '25)**

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

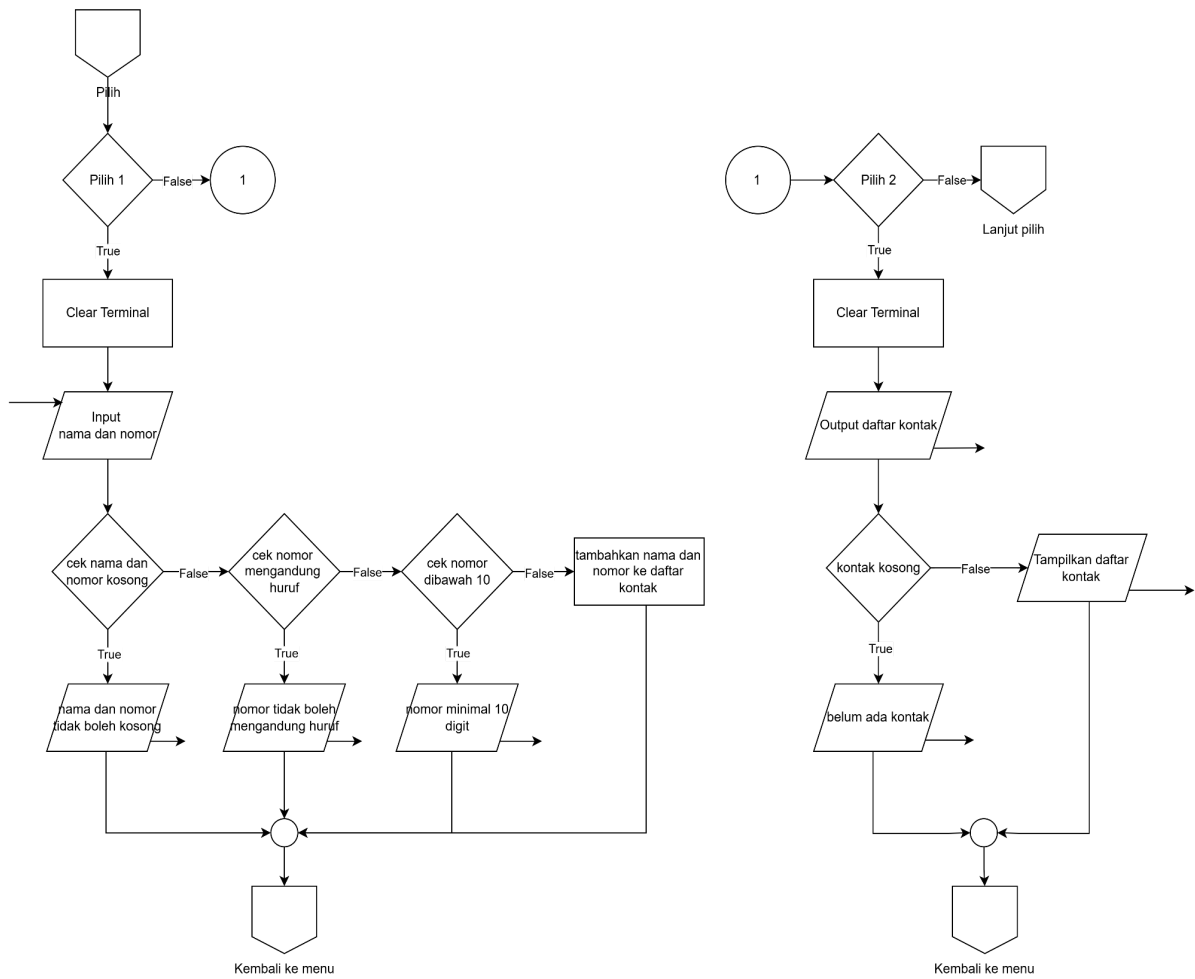
## 1. Flowchart



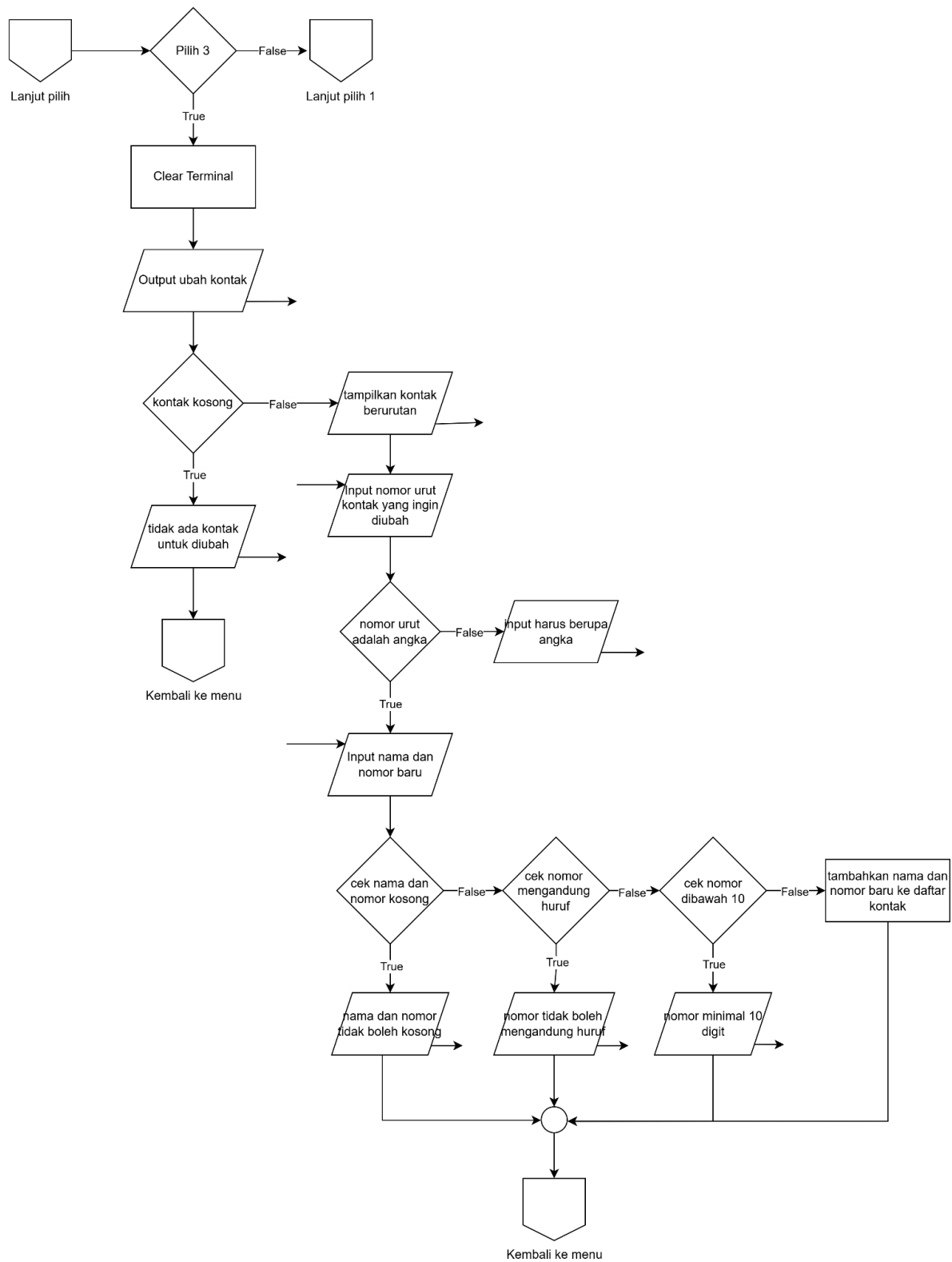
Gambar 1.1 Flowchart



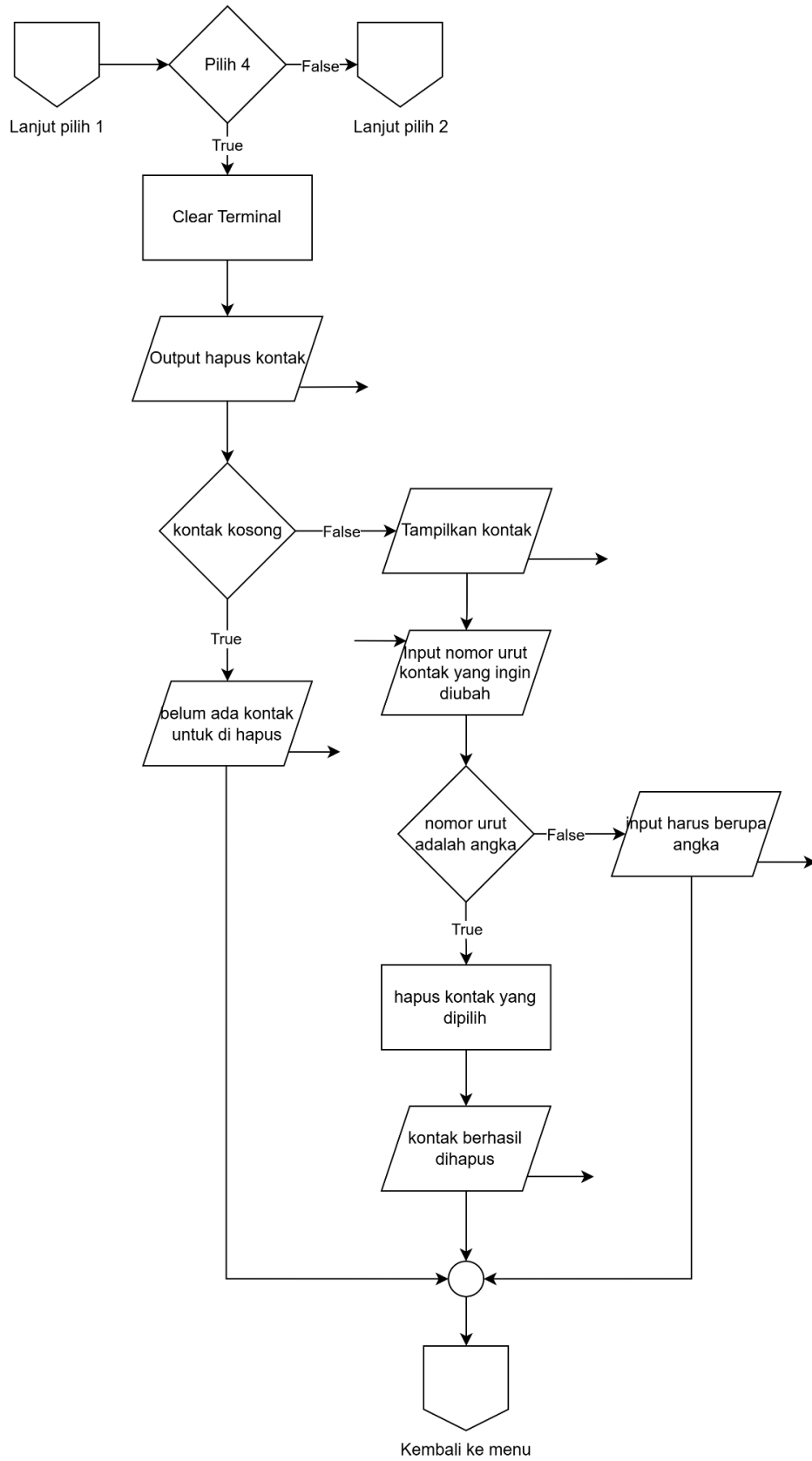
Gambar 1.2 Flowchart



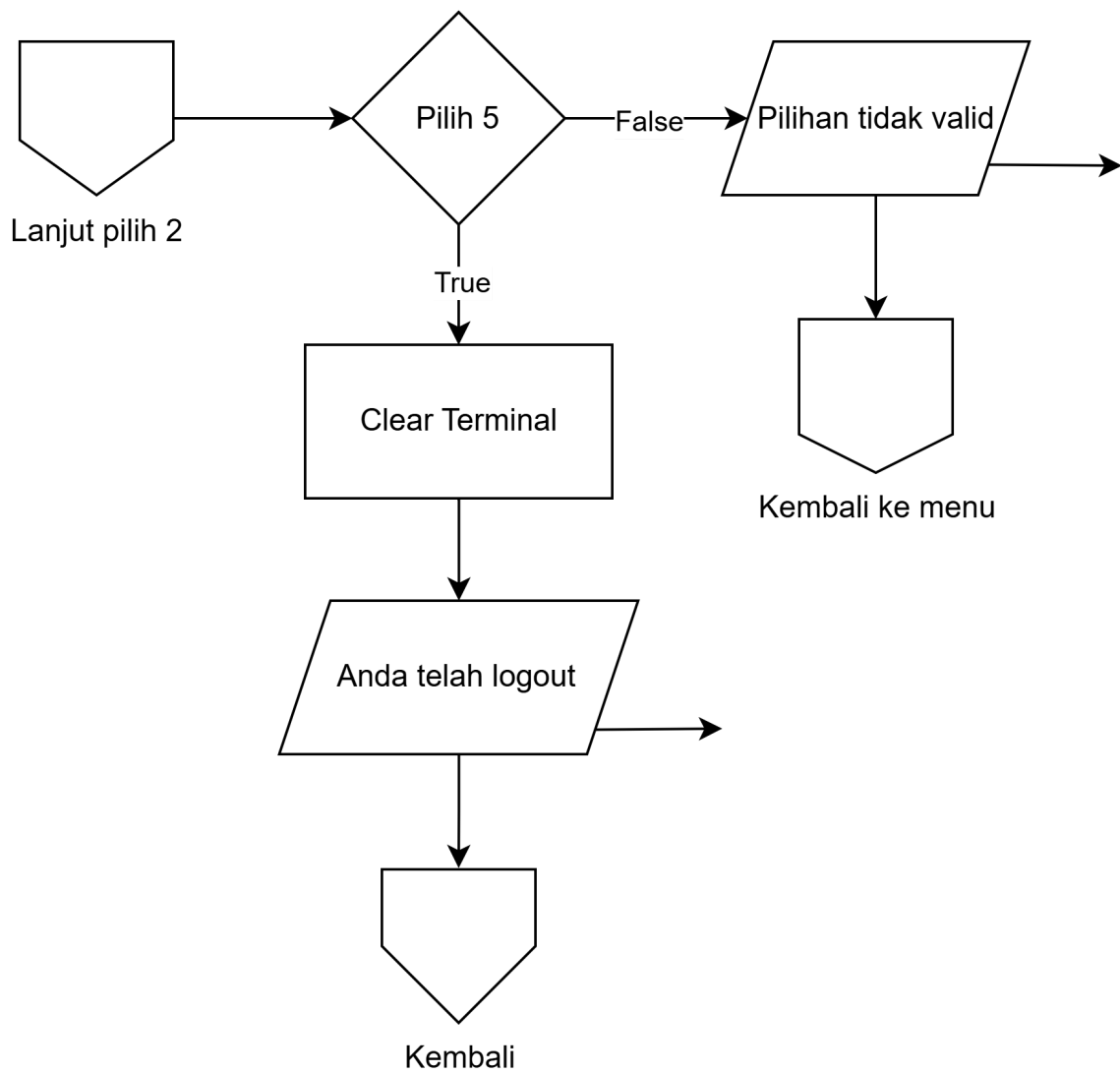
Gambar 1.3 Flowchart



Gambar 1.4 Flowchart



Gambar 1.5 Flowchart



*Gambar 1.6 Flowchart*

## 2. Deskripsi Singkat Program

Program belanja ini dibuat untuk membantu pengguna mengelola kontak mereka. Dengan program ini pengguna perlu melakukan login atau registrasi sebelum bisa mengakses fitur setelah pengguna login mereka dapat menambahkan kontak, melihat kontak yang sudah mereka dapat, mengubah data kontak jika pengguna mau dan menghapus data kontak yang sudah tidak dipakai lagi. Program ini juga menyediakan menu utama yang memandu pengguna dalam melakukan operasi dasar.

## 3. Source Code

### A. Menu Login dan Registrasi

```
import os

pengguna = {"Pirlo": "pirlo008"}
kontak_pengguna = {"Pirlo": {}}
pengguna_login = None

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

def menu_login():
    clear()
    print("=== SELAMAT DATANG ===")
    print("1. Login")
    print("2. Registrasi")
    print("3. Keluar")
    return input("Pilih (1-3): ")

def proses_login():
    clear()
    print("=== LOGIN ===")
    username = input("Username: ").strip()
    password = input("Password: ").strip()

    if username in pengguna and pengguna[username] == password:
        print("Login berhasil!")
        if username not in kontak_pengguna:
            kontak_pengguna[username] = {}
        return username
    else:
        print("Username atau password salah!")
        return None
```



```
def proses_registrasi():
    clear()
    print("=== REGISTRASI AKUN ===")
    username = input("Masukkan username: ").strip()
    password = input("Masukkan password: ").strip()

    if not username or not password:
        print("Username dan password tidak boleh kosong!")
        return False
    if username in pengguna:
        print("Username sudah digunakan!")
        return False

    pengguna[username] = password
    kontak_pengguna[username] = {}
    print("Akun berhasil dibuat! Silakan login.")
    return True
```

Di bagian ini pengguna akan diminta untuk login sebelum dapat masuk ke fitur utama, pengguna perlu memasukkan username dan password untuk login jika salah atau tidak valid sistem akan memberitahu. Pengguna dapat registrasi jika belum memiliki akun dengan memilih opsi registrasi dan setelahnya pengguna akan diminta memasukkan username dan password yang diinginkan sebelum mengkonfirmasi sistem akan mengecek apakah sudah ada akun yang memiliki username yang sama jika ada sistem akan memberitahu dan meminta pengguna untuk mengulang proses registrasi, jika semua proses sudah benar maka sistem akan memberi notifikasi bahwa akun telah berhasil dibuat.

## B. Menu Utama

```
def tampilkan_menu_utama(username):
    clear()
    print(f"=== MANAJEMEN KONTAK ({username}) ===")
    print("1. Tambah Kontak")
    print("2. Lihat Semua Kontak")
    print("3. Ubah Kontak")
    print("4. Hapus Kontak")
    print("5. Logout")
    return input("Pilih menu (1-5): ")
```

Setelah login sistem akan menunjukkan pengguna login sebagai siapa dan memberikan menu pilihan aksi atau fitur yang dapat di pilih oleh pengguna, pengguna dapat melakukan 5 aksi yang pertama menambah kontak, yang kedua melihat kontak yang ada atau sudah ditambahkan, yang ketiga mengubah data kontak yang diinginkan, keempat menghapus kontak dan yang terakhir berhenti atau keluar jika pengguna sudah tidak lagi ingin melakukan aksi atau sudah tidak ingin memakai program.

### C. Fitur Menambah Kontak

```
def tambah_kontak(kontak_dict):
    clear()
    print("=== TAMBAH KONTAK ===")
    nama = input("Masukkan nama: ").strip()
    no_hp = input("Masukkan nomor HP: ").strip()

    if nama == "":
        print("Nama tidak boleh kosong!")
    elif no_hp == "" or not no_hp.isdigit() or len(no_hp) < 10:
        print("Nomor HP harus angka dan minimal 10 digit!")
    else:
        if nama in kontak_dict:
            print(f"Peringatan: Kontak '{nama}' sudah ada dan akan ditimpa!")
            kontak_dict[nama] = no_hp
            print("Kontak berhasil ditambahkan!")
        input("Tekan Enter untuk kembali...")
```

Fitur ini digunakan saat pengguna memilih untuk menambahkan kontak pertama sistem akan membersihkan terminal lalu akan meminta pengguna untuk memasukkan nama dan nomor hp. Jika nama kosong maka sistem akan memberitahu bahwa nama tidak boleh kosong begitupun juga dengan nomor hp jika kosong sistem juga akan memberitahu bahwa nomor hp tidak boleh kosong, jika nama dan nomor hp tidak kosong selanjutnya sistem akan mengecek apakah nomor yang diinput menggunakan angka dan lebih dari 10 jika ya maka sistem akan menambahkan data input yang diberikan dan menyimpannya jika tidak sistem akan memberitahu bahwa nomor harus menggunakan digit (0-9) dan minimal 10 digit setelah. Sistem juga akan memberitahu jika ada kontak dengan nama yang sama dan akan menimpa data kontak sebelumnya setelah selesai maka pengguna akan dibawa kembali ke menu utama.

#### D. Fitur Melihat Daftar Kontak

```
def tampilkan_kontak(kontak_dict):
    if not kontak_dict:
        print("Belum ada kontak.")
    else:
        for i, (nama, nomor) in enumerate(kontak_dict.items(), 1):
            print(f"{i}. Nama: {nama}, No HP: {nomor}")

def lihat_kontak(kontak_dict):
    clear()
    print("=== DAFTAR KONTAK ===")
    tampilkan_kontak(kontak_dict)
    input("Tekan Enter untuk kembali...")
```

Fitur melihat daftar kontak jika pengguna memilih fitur ini maka sistem akan membersihkan terminal dan mengecek apakah ada kontak yang telah ditambahkan jika ada maka sistem menggunakan *for i* untuk mengakses nested dictionary atau kontak yang telah ditambahkan atau di simpan jika ternyata tidak ada data kontak didalam list maka sistem akan memberi output belum ada kontak.

#### E. Fitur Mengubah Data Kontak

```
def ubah_kontak(kontak_dict):
    clear()
    print("=== UBAH KONTAK ===")
    if not kontak_dict:
        print("Tidak ada kontak untuk diubah.")
        input("Tekan Enter untuk kembali...")
        return

    tampilkan_kontak(kontak_dict)
    try:
        pilihan = int(input("Pilih nomor kontak yang ingin diubah: ")) - 1
        daftar_nama = list(kontak_dict.keys())
        if 0 <= pilihan < len(daftar_nama):
            nama_lama = daftar_nama[pilihan]
            nama_baru = input("Nama baru: ").strip()
            no_hp_baru = input("Nomor HP baru: ").strip()

            if nama_baru == "":
                print("Nama tidak boleh kosong!")
            elif no_hp_baru == "" or not no_hp_baru.isdigit() or len(no_hp_baru)
< 10:
```

```

        print("Nomor HP tidak valid!")
    else:
        del kontak_dict[nama_lama]
        kontak_dict[nama_baru] = no_hp_baru
        print("Kontak berhasil diubah!")
    else:
        print("Nomor kontak tidak valid!")
except ValueError:
    print("Input harus berupa angka!")
input("Tekan Enter untuk kembali...")

```

Fitur ubah data kontak jika pengguna memilih fitur ini maka sistem akan membersihkan terminal lalu mengecek apakah ada daftar kontak yang dapat di ubah jika ada sistem akan memberikan list kontak yang tersedia pengguna akan diminta untuk memilih kontak mana yang ingin di ubah input harus berupa angka jika tidak maka sistem akan memberitahu bahwa input harus berupa angka dan program di ulang. Jika sudah memilih maka pengguna diminta untuk memasukkan nama dan nomor yang baru sistem akan mengecek ulang nama dan nomor jika kosong maka akan diberitahu sistem bahwa tidak boleh kosong dan mengubah kontak gagal program pun diulang, begitu pula jika nomor tidak berupa angka dan dibawah 10 digit sistem akan memberitahu dan program pun diulang kembali. Jika benar dan tidak ada kesalahan input maka kontak lama akan diubah menjadi kontak baru dan di simpan.

## F. Fitur Hapus Data Kontak

```

def hapus_kontak(kontak_dict):
    clear()
    print("=== HAPUS KONTAK ===")
    if not kontak_dict:
        print("Tidak ada kontak untuk dihapus.")
        input("Tekan Enter untuk kembali...")
        return

    tampilkan_kontak(kontak_dict)
    try:
        pilihan = int(input("Pilih nomor kontak yang ingin dihapus: ")) - 1
        daftar_nama = list(kontak_dict.keys())
        if 0 <= pilihan < len(daftar_nama):
            nama_dihapus = daftar_nama[pilihan]
            del kontak_dict[nama_dihapus]

```

```

        print(f"Kontak '{nama_dihapus}' berhasil dihapus!")
    else:
        print("Nomor kontak tidak valid!")
except ValueError:
    print("Input harus berupa angka!")
input("Tekan Enter untuk kembali...")

```

Fitur hapus data kontak jika pengguna memilih fitur ini maka sistem akan membersihkan terminal lalu sistem akan mengecek apakah ada daftar atau data kontak yang dapat di hapus jika tidak sistem akan memberikan output bahwa tidak ada kontak yang dapat yang di hapus jika ada maka sistem akan mengakses list dan memberikan daftar kontak yang dapat dihapus pengguna akan diminta untuk memilih kontak mana yang ingin dihapus jika sudah memilih maka sistem akan langsung menghapus data kontak yang terpilih dan pengguna akan dibawa kembali ke menu utama.

## G. File Main

```

import Sistem as s

def main():
    while True:
        if s.pengguna_login is None:
            pilihan = s.menu_login()

            if pilihan == "1":
                s.pengguna_login = s.proses_login()
                input("Tekan Enter untuk kembali...")

            elif pilihan == "2":
                s.proses_registrasi()
                input("Tekan Enter untuk kembali...")

            elif pilihan == "3":
                s.clear()
                print("Terima kasih! Sampai jumpa.")
                break

            else:
                print("Pilihan tidak valid!")
                input("Tekan Enter untuk kembali...")

        else:
            pilihan = s.tampilkan_menu_utama(s.pengguna_login)

```

```

kontak = s.kontak_pengguna[s.pengguna_login]

if pilihan == "1":
    s.tambah_kontak(kontak)
elif pilihan == "2":
    s.lihat_kontak(kontak)
elif pilihan == "3":
    s.ubah_kontak(kontak)
elif pilihan == "4":
    s.hapus_kontak(kontak)
elif pilihan == "5":
    print("Anda telah logout.")
    s.pengguna_login = None
    input("Tekan Enter untuk kembali...")
else:
    print("Pilihan tidak valid!")
    input("Tekan Enter untuk kembali...")

if __name__ == "__main__":
    main()

```

Ini adalah file utama dimana semua function sebelumnya akan di import dan dijalankan sesuai dengan pilihan pengguna di main. Fitur ini bekerja dengan mengecek apakah ada pengguna yang login jika tidak sistem akan memanggil function login agar pengguna dapat login terlebih dahulu kemudian setelah selesai sistem akan memanggil function menu utama di akhir saya memberikan “if \_\_name\_\_ == “\_\_main\_\_”” agar sistem menjalankan function main dengan benar.

#### 4. Hasil Output

```
=== SELAMAT DATANG ===  
1. Login  
2. Registrasi  
3. Keluar  
Pilih (1-3): █
```

Gambar 4.1 Output Menu Login

```
=== LOGIN ===  
Username: Pirlo  
Password: pirlo008  
Login berhasil!  
Tekan Enter untuk melanjutkan...█
```

Gambar 4.2 Output Jika Pengguna Login

```
=== REGISTRASI AKUN ===  
Masukkan username: Yoga  
Masukkan password: 017  
Akun berhasil dibuat! Silakan login.  
Tekan Enter untuk kembali...█
```

Gambar 4.3 Output Jika Pengguna Berhasil Registrasi

```
=== MANAJEMEN KONTAK (Pirlo) ===  
1. Tambah Kontak  
2. Lihat Semua Kontak  
3. Ubah Kontak  
4. Hapus Kontak  
5. Logout  
Pilih menu (1-5): █
```

Gambar 4.4 Output Menu Utama

```
=== TAMBAH KONTAK ===  
Masukkan nama: Pirlo  
Masukkan nomor HP: 0855908766  
Kontak berhasil ditambahkan!  
Tekan Enter untuk kembali...
```

Gambar 4.5 Output Jika Pengguna Menambahkan Data Kontak

```
=== DAFTAR KONTAK ===  
1. Nama: Pirlo, No HP: 0855908766  
Tekan Enter untuk kembali...
```

Gambar 4.6 Output Daftar Kontak

```
=== UBAH KONTAK ===  
1. Nama: Pirlo, No HP: 0855908766  
Pilih nomor kontak yang ingin diubah: 1  
Mengubah kontak: Pirlo (0855908766)  
Masukkan nama baru: Dapupu  
Masukkan nomor HP baru: 08997645537  
Kontak berhasil diubah!  
Tekan Enter untuk kembali...
```

Gambar 4.7 Output Jika Pengguna Mengubah Data Kontak

```
=== DAFTAR KONTAK ===  
1. Nama: Dapupu, No HP: 08997645537  
Tekan Enter untuk kembali...
```

Gambar 4.8 Output Setelah Pengguna Mengubah Data Kontak

```
=== HAPUS KONTAK ===  
1. Nama: Dapupu, No HP: 08997645537  
Pilih nomor kontak yang ingin dihapus: 1  
Kontak berhasil dihapus!  
Tekan Enter untuk kembali...
```

Gambar 4.9 Output Jika Pengguna Menghapus Data Kontak



```
=== DAFTAR KONTAK ===  
Belum ada kontak.  
Tekan Enter untuk kembali...
```

Gambar 4.10 Output Setelah Pengguna Menghapus Data Kontak

```
Terima kasih! Sampai jumpa.  
PS C:\praktikum-apd>
```

Gambar 4.11 Output Saat Pengguna Selesai Menggunakan Program

## 5. Langkah-langkah GIT

### 5.1 GIT Add

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

Fungsi ini menandai atau menambahkan semua file yang ada di folder bisa juga menggunakan (`git add namafile`) jika ingin menambahkan file spesifik.

### 5.2 GIT Commit

```
PS C:\praktikum-apd> git commit -m "push file"  
[main 2cc6db7] push file  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 post-test/post-test-apd-7/2509106008-PirloSyabilaHafuza-PT-7.pdf
```

Fungsi ini digunakan untuk *commit* atau menyimpan perubahan yang dilakukan untuk pesan commit sendiri tidak terpaku apapun dan terserah ingin diisi apa saja asal bisa dipahami.

### 5.3 GIT Push

```
PS C:\praktikum-apd> git push -u origin main  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 2 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (5/5), 1.03 MiB | 558.00 KiB/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/Altairstry/praktikum-apd.git  
dc7ad2e..2cc6db7 main -> main  
branch 'main' set up to track 'origin/main'.
```

Fungsi ini digunakan untuk mengupload atau menambahkan file ke repository yang awalnya di folder komputer ke github langsung, jika push berhasil biasanya akan langsung dibawa ke halaman repository/github nya langsung.