

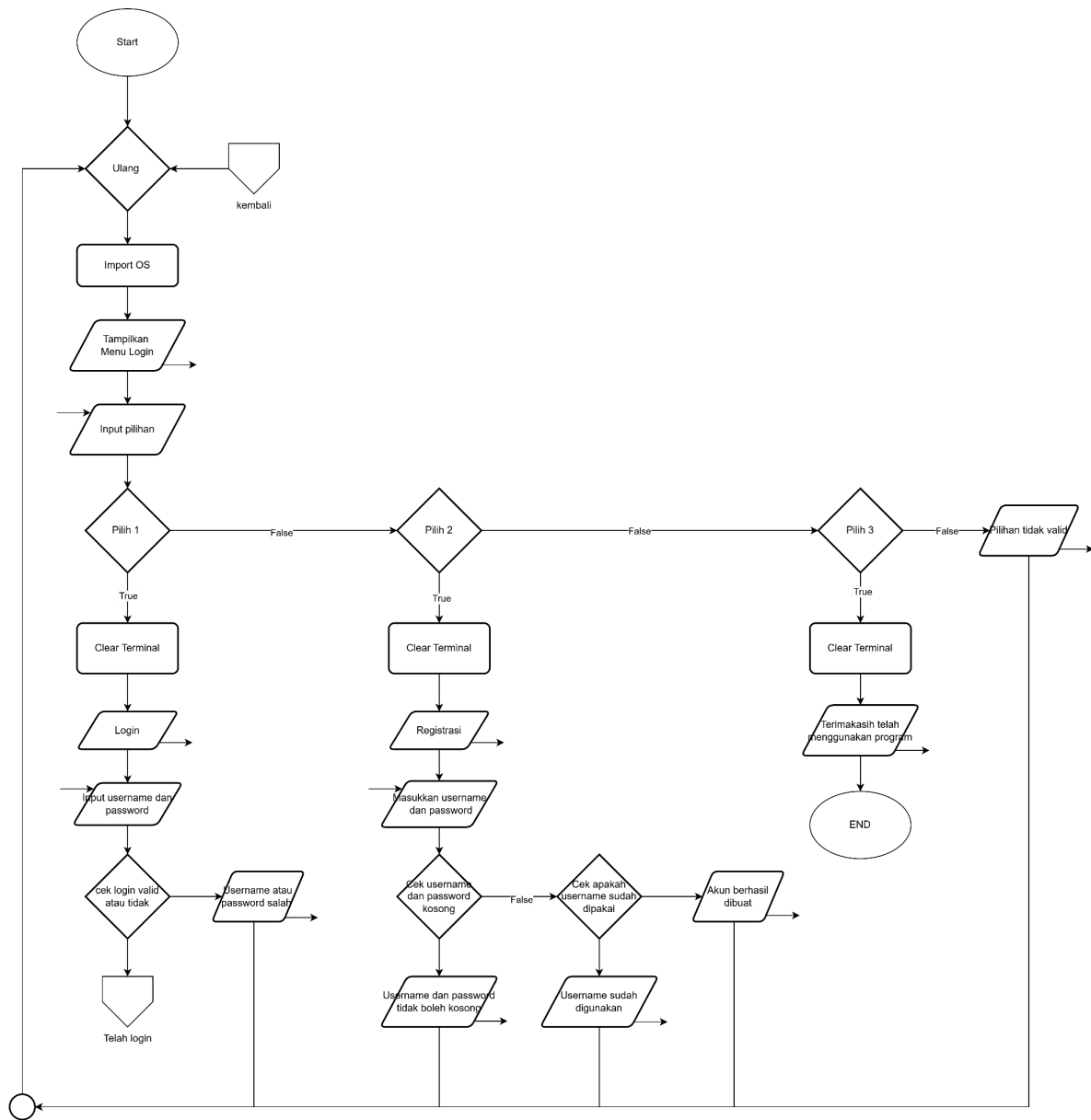
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN DASAR



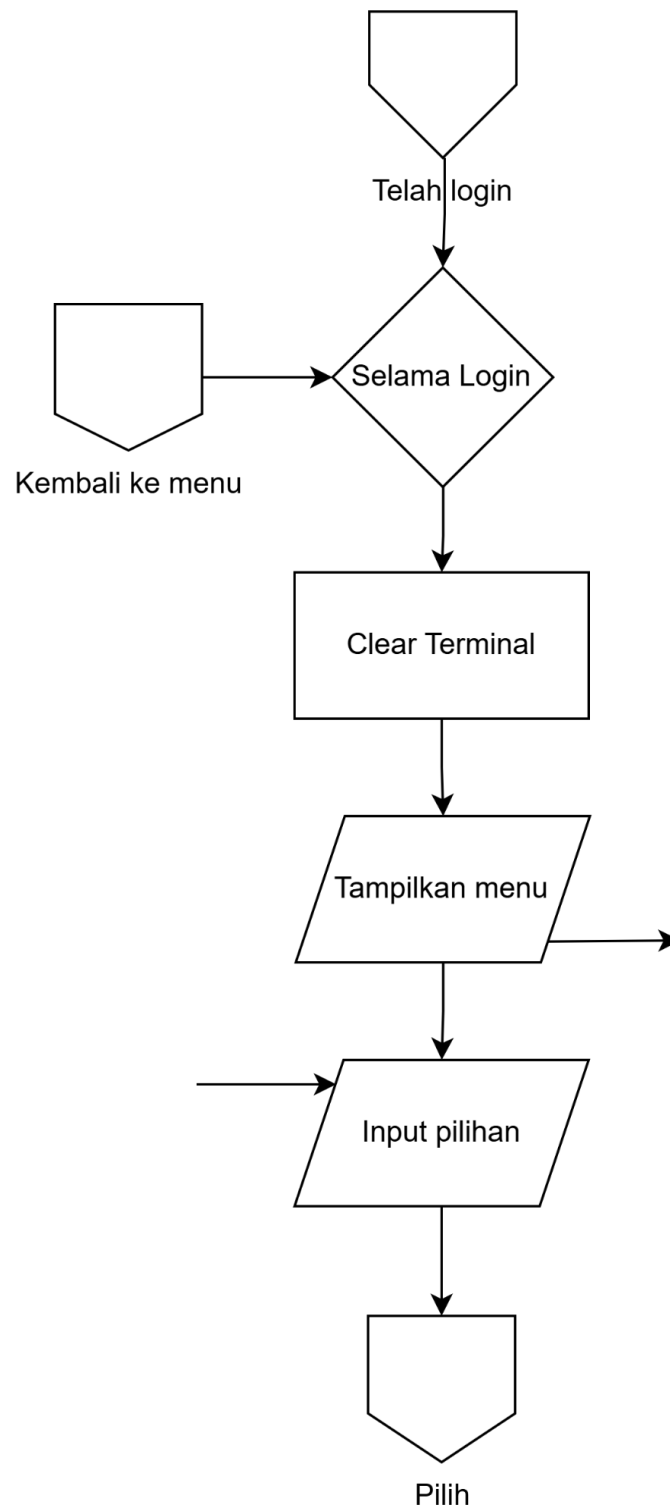
Disusun oleh:
Pirlo Syabila Hafuza (2509106008)
Kelas (A1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

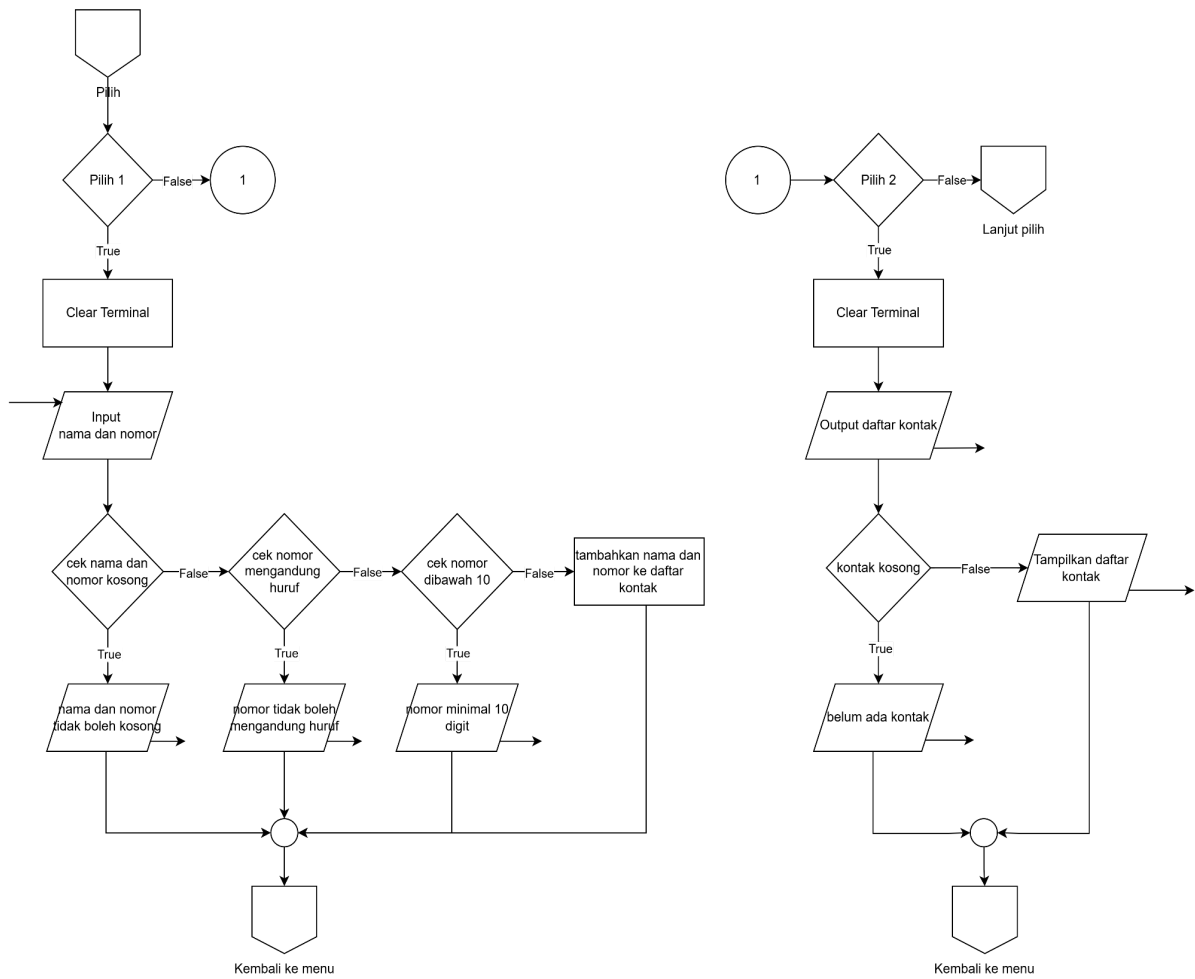
1. Flowchart



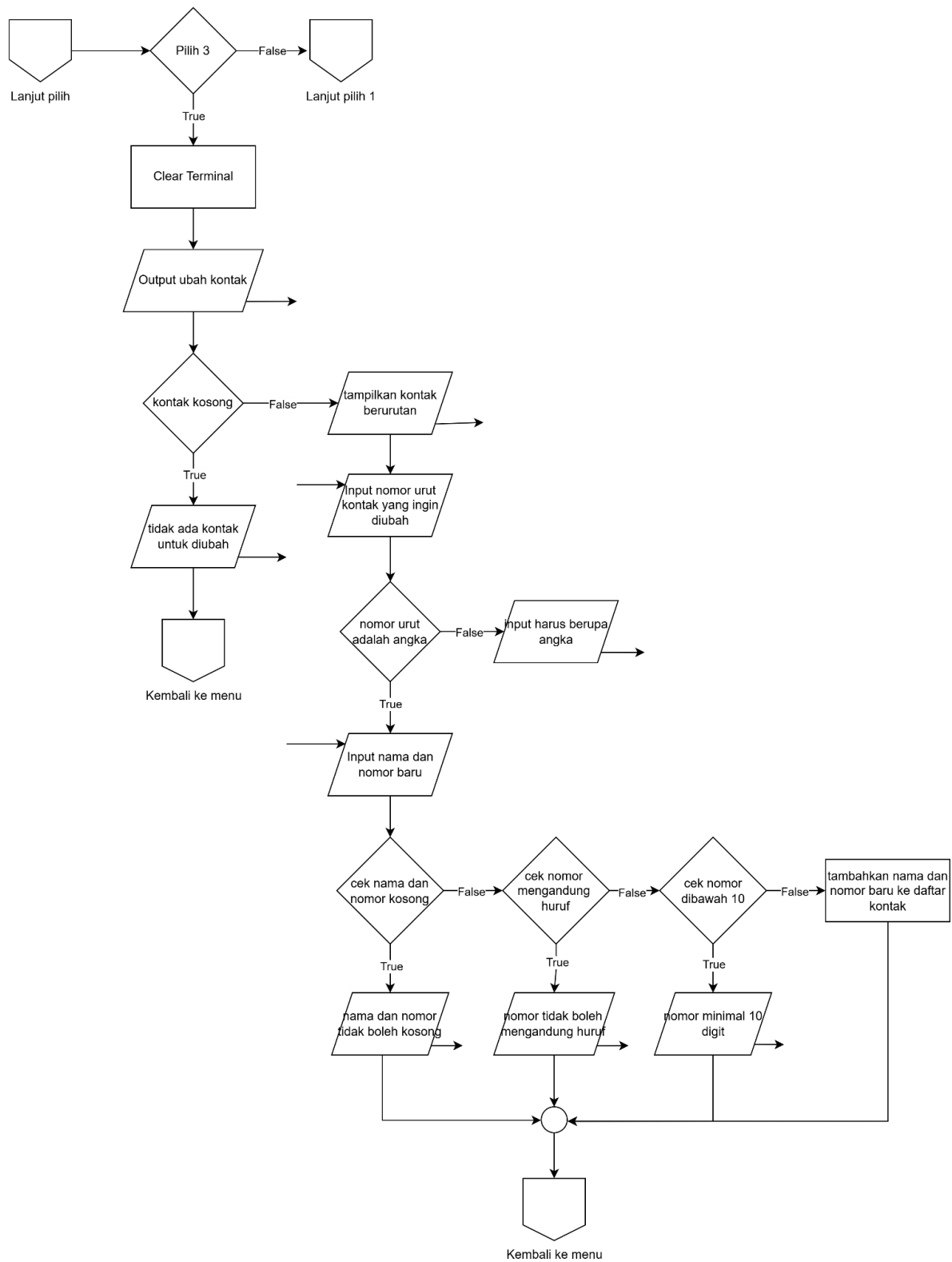
Gambar 1.1 Flowchart



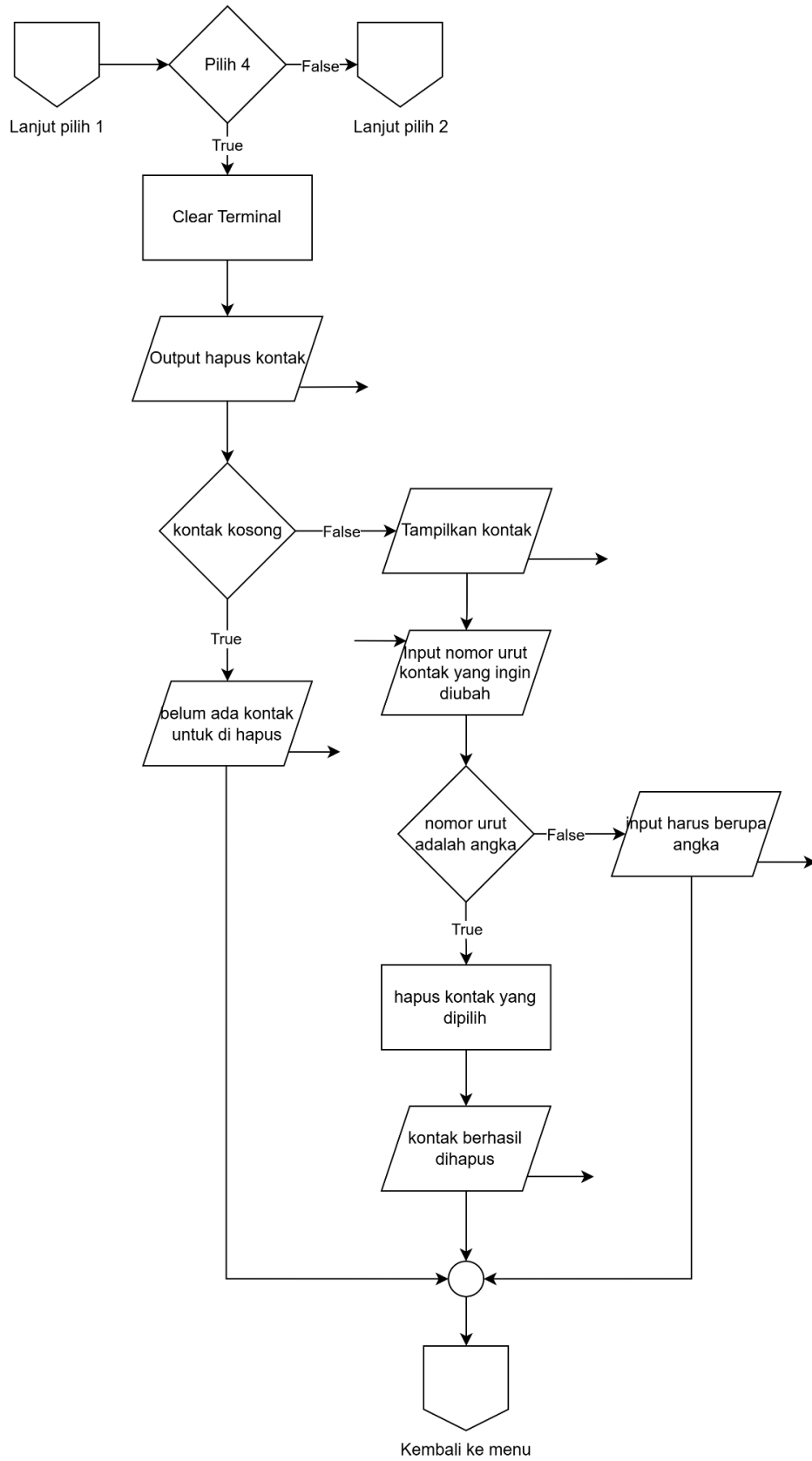
Gambar 1.2 Flowchart



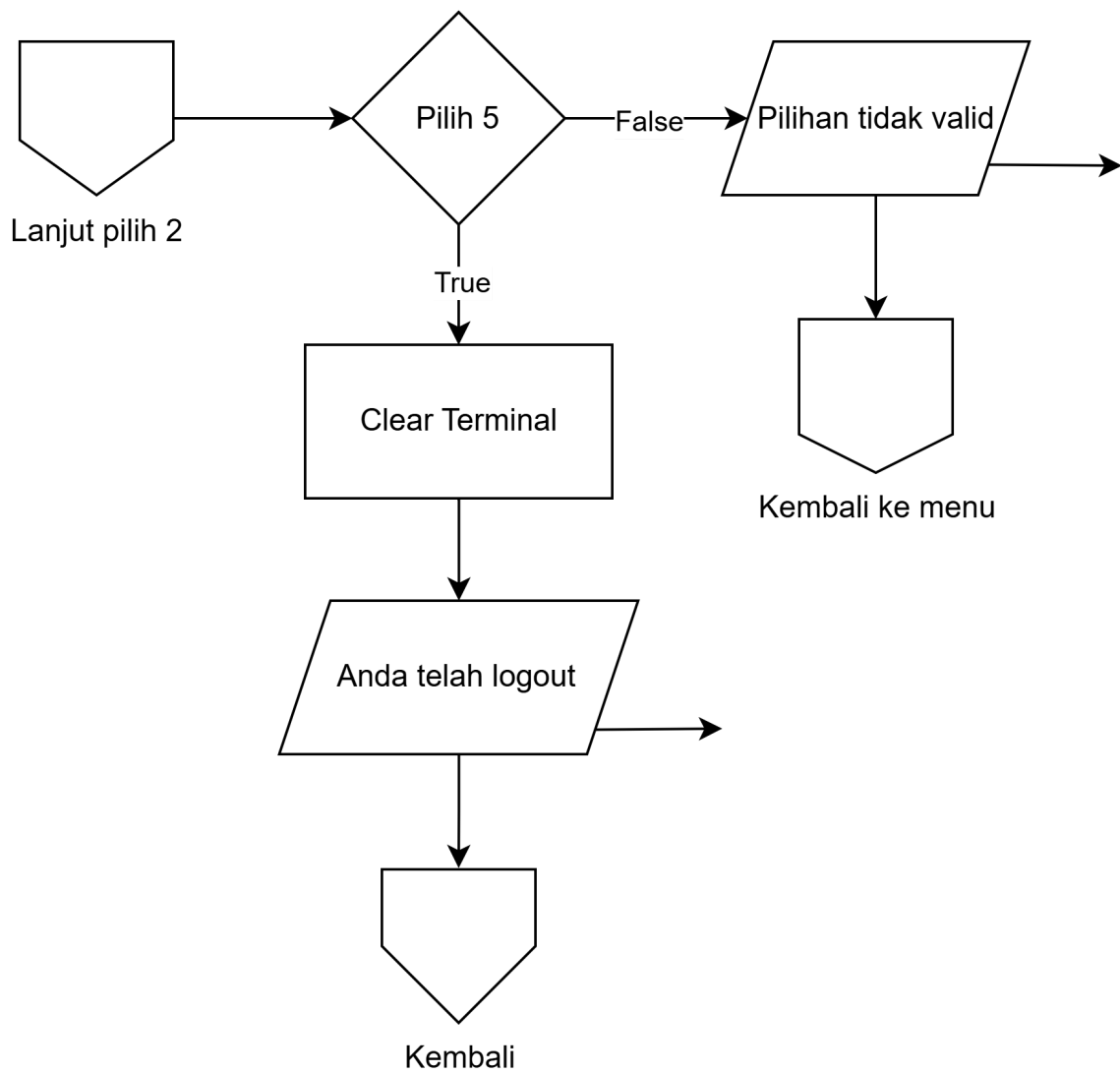
Gambar 1.3 Flowchart



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

```
pengguna = {"pirlo": "pirlo008"}

kontak_pengguna = {"pirlo": {}}

pengguna_login = None

while True:
    if pengguna_login is None:
        os.system('cls' if os.name == 'nt' else 'clear')
        print("=== SELAMAT DATANG ===")
        print("1. Login")
        print("2. Registrasi")
        print("3. Keluar")
        pilihan_awal = input("Pilih (1-3): ")

        if pilihan_awal == "1":
            os.system('cls' if os.name == 'nt' else 'clear')
            print("=== LOGIN ===")
            nama_pengguna = input("Username: ").strip()
            kata_sandi = input("Password: ").strip()

            if nama_pengguna in pengguna and pengguna[nama_pengguna] == kata_sandi:
                pengguna_login = nama_pengguna
                if nama_pengguna not in kontak_pengguna:
                    kontak_pengguna[nama_pengguna] = {}
                print("Login berhasil!")
                input("Tekan Enter untuk melanjutkan...")
            else:
                print("Username atau password salah!")
```



```

        input("Tekan Enter untuk kembali...")

    elif pilihan_awal == "2":
        os.system('cls' if os.name == 'nt' else 'clear')
        print("=== REGISTRASI AKUN ===")
        nama_pengguna = input("Masukkan username: ").strip()
        kata_sandi = input("Masukkan password: ").strip()

        if not nama_pengguna or not kata_sandi:
            print("Username dan password tidak boleh kosong!")
        elif nama_pengguna in pengguna:
            print("Username sudah digunakan!")
        else:
            pengguna[nama_pengguna] = kata_sandi
            kontak_pengguna[nama_pengguna] = {}
            print("Akun berhasil dibuat! Silakan login.")
            input("Tekan Enter untuk kembali...")

    elif pilihan_awal == "3":
        os.system('cls' if os.name == 'nt' else 'clear')
        print("Terima kasih! Sampai jumpa.")
        break

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

```

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

```
else:
    kontak = kontak_pengguna[pengguna_login]
    os.system('cls' if os.name == 'nt' else 'clear')
    print(f"=== MANAJEMEN KONTAK ({pengguna_login}) ===")
    print("1. Tambah Kontak")
    print("2. Lihat Semua Kontak")
    print("3. Ubah Kontak")
    print("4. Hapus Kontak")
    print("5. Logout")
    pilihan = 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

```
if pilihan == "1":
    os.system('cls' if os.name == 'nt' else '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 == "":
        print("Nomor HP tidak boleh kosong!")
    elif not no_hp.isdigit():
        print("Nomor HP hanya boleh berisi angka (0-9)!")
    elif len(no_hp) < 10:
        print("Nomor HP minimal 10 digit!")
    else:
        if nama in kontak:
            print(f"Peringatan: Kontak '{nama}' sudah ada dan akan ditimpa!")
        kontak[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

```
elif pilihan == "2":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== DAFTAR KONTAK ===")
    if not kontak:
        print("Belum ada kontak.")
    else:
        for i, (nama_kontak, nomor) in enumerate(kontak.items(), 1):
            print(f"{i}. Nama: {nama_kontak}, No HP: {nomor}")
        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

```
elif pilihan == "3":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== UBAH KONTAK ===")
    if not kontak:
        print("Tidak ada kontak untuk diubah.")
        input("Tekan Enter untuk kembali...")
    else:
        daftar_nama = list(kontak.keys())
        for i, nama_kontak in enumerate(daftar_nama, 1):
            print(f"{i}. Nama: {nama_kontak}, No HP: {kontak[nama_kontak]}")

        nomor_pilihan = input("Pilih nomor kontak yang ingin diubah: ")
        if not nomor_pilihan.isdigit():
            print("Input harus berupa angka!")
            input("Tekan Enter untuk kembali...")
        else:
            indeks = int(nomor_pilihan) - 1
            if not (0 <= indeks < len(daftar_nama)):
                print("Nomor kontak tidak valid!")
                input("Tekan Enter untuk kembali...")
            else:
                nama_lama = daftar_nama[indeks]
                print(f"Mengubah kontak: {nama_lama} ({kontak[nama_lama]}")
                nama_baru = input("Masukkan nama baru: ").strip()
                no_hp_baru = input("Masukkan nomor HP baru: ").strip()

                if nama_baru == "":
                    print("Nama tidak boleh kosong!")
                elif no_hp_baru == "":
                    print("Nomor HP tidak boleh kosong!")
                elif not no_hp_baru.isdigit():
                    print("Nomor HP hanya boleh berisi angka (0-9)!")
                elif len(no_hp_baru) < 10:
                    print("Nomor HP minimal 10 digit!")
                else:
                    del kontak[nama_lama]
                    kontak[nama_baru] = no_hp_baru
                    print("Kontak berhasil diubah!")
                    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

```
elif pilihan == "4":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== HAPUS KONTAK ===")
    if not kontak:
        print("Tidak ada kontak untuk dihapus.")
        input("Tekan Enter untuk kembali...")
    else:
        daftar_nama = list(kontak.keys())
        for i, nama_kontak in enumerate(daftar_nama, 1):
            print(f"{i}. Nama: {nama_kontak}, No HP: {kontak[nama_kontak]}")

        nomor_pilihan = input("Pilih nomor kontak yang ingin dihapus: ")
        if not nomor_pilihan.isdigit():
            print("Input harus berupa angka!")
            input("Tekan Enter untuk kembali...")
        else:
            indeks = int(nomor_pilihan) - 1
            if 0 <= indeks < len(daftar_nama):
                nama_hapus = daftar_nama[indeks]
                del kontak[nama_hapus]
                print("Kontak berhasil dihapus!")
            else:
                print("Nomor kontak tidak valid!")
                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.

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"  
On branch main  
Your branch is ahead of 'origin/main' by 1 commit.  
  (use "git push" to publish your local commits)  
  
nothing to commit, working tree clean  
PS C:\praktikum-apd>
```

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), 1.74 KiB | 446.00 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  
   8e07baf..dd59f25  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