

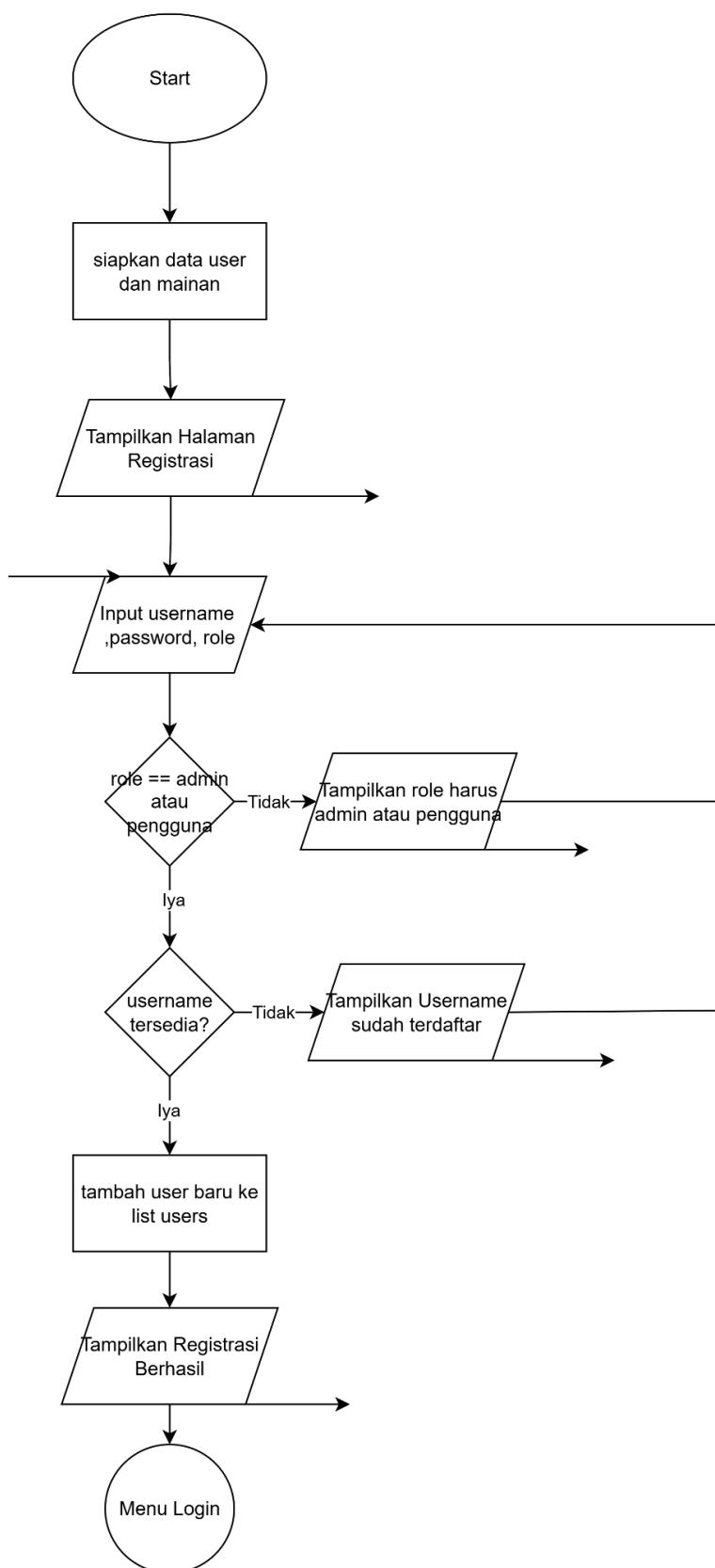
**LAPORAN PRAKTIKUM  
POSTTEST 8  
ALGORITMA PEMROGRAMAN DASAR**



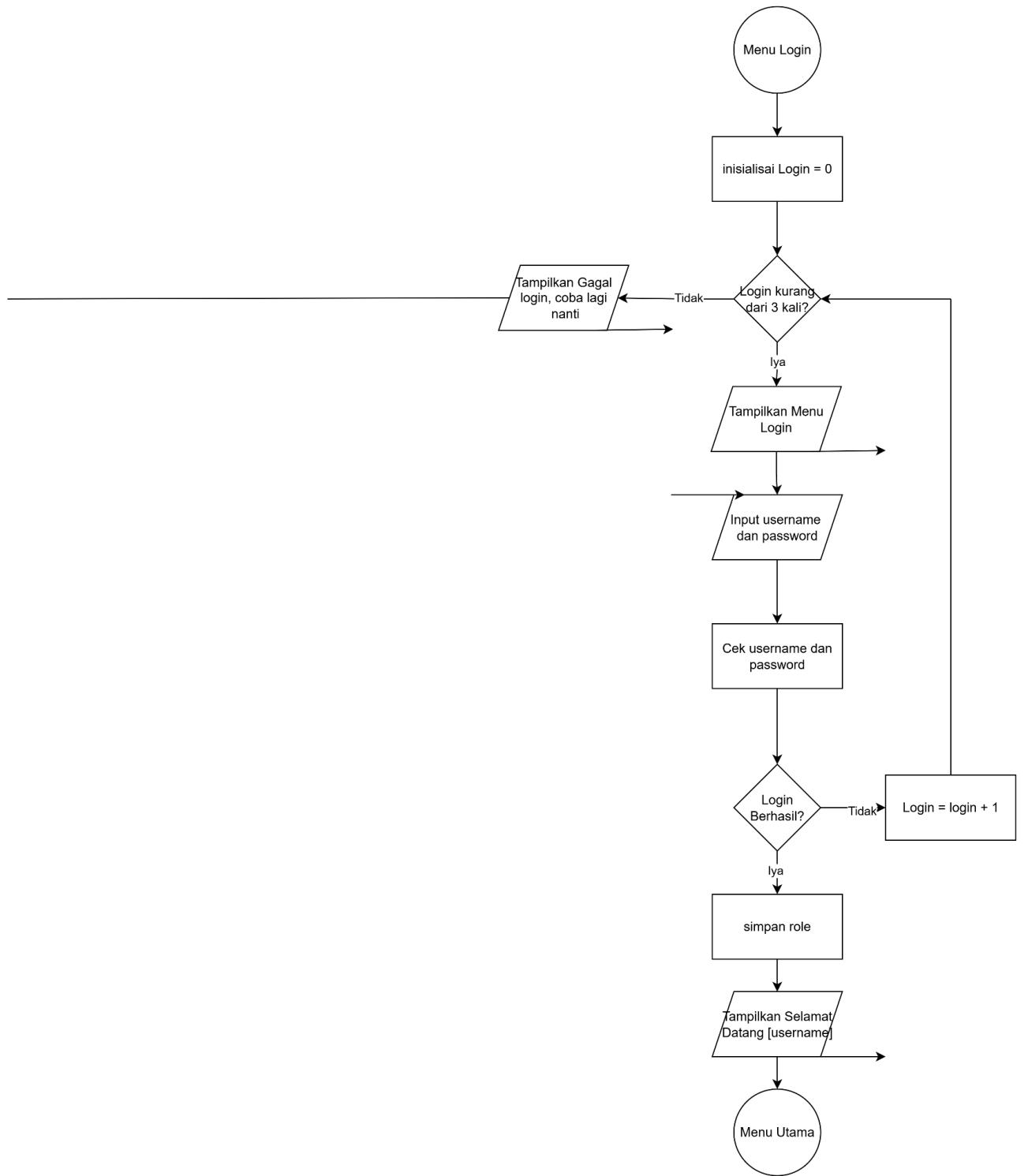
**Disusun oleh:**  
**Muhammad Fachri Akbar(2509106076)**  
**Kelas (B2 '25)**

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS MULAWARMAN  
SAMARINDA  
2025**

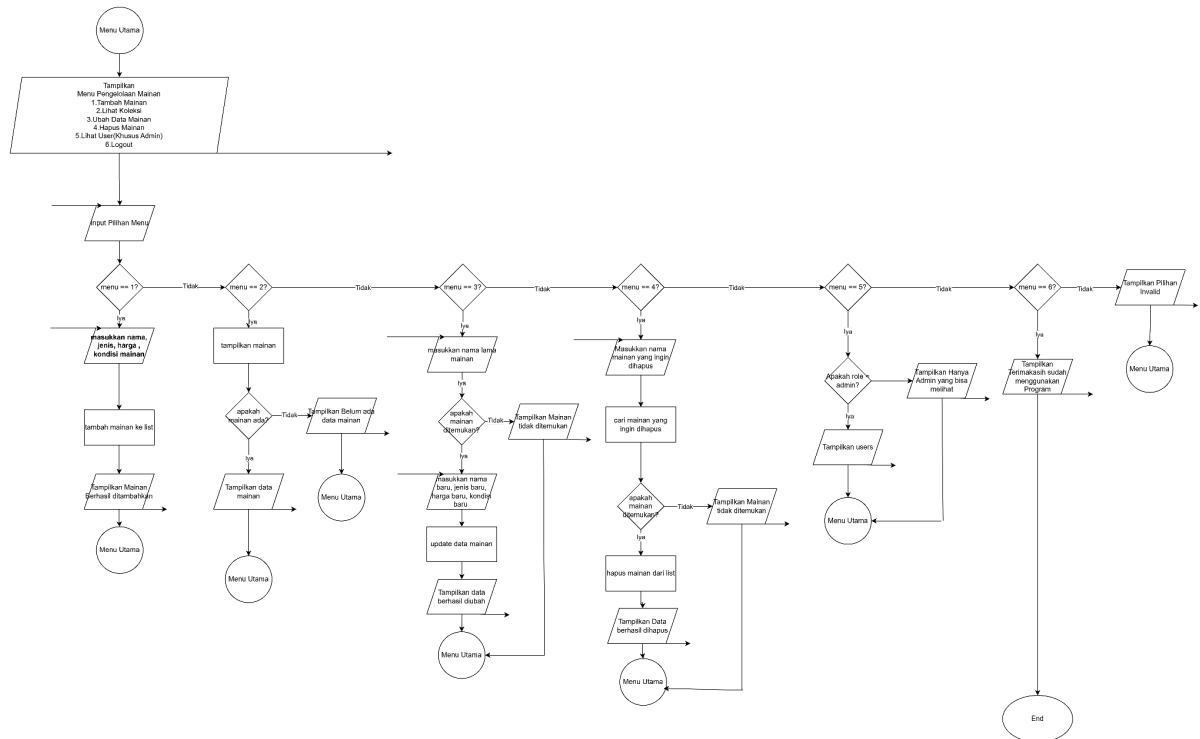
## 1. Flowchart



Gambar 1.1 Flowchart Halaman Registrasi



Gambar 1.2 Flowchart Halaman Login



Gambar 1.3 Halaman Menu Utama User

## 2. Deskripsi Singkat Program

Program ini adalah Sistem Manajemen Koleksi Mainan (LEGO & Action Figure) yang dirancang untuk membantu pengguna mengelola data koleksi mainan mereka secara digital. Program ini dibangun menggunakan bahasa pemrograman Python dengan konsep CRUD (Create, Read, Update, Delete) dan dilengkapi dengan sistem multiuser serta berbagai fitur keamanan dan validasi.

## 3. Source Code

### A.Fitur Tambah Mainan (Create)

Fitur ini berguna untuk menambahkan data mainan baru ke dalam koleksi. User akan diminta memasukkan nama, jenis, harga, dan kondisi mainan. Sistem akan otomatis memberikan nomor urut sebagai identitas mainan tersebut.

#### Source Code:

```

def tambah_mainan():
    print("==> Tambah Mainan ==>")
    nama = input("Nama: ")
    
```

```

jenis = input("Jenis: ")
harga = input_angka("Harga: ")
kondisi = input("Kondisi: ")

mainan[len(mainan) + 1] = {"nama": nama, "jenis": jenis, "harga": harga,
"kondisi": kondisi}
print("Mainan berhasil ditambahkan!\n")

```

## B. Fitur Lihat Koleksi(Read)

Fitur ini berguna untuk menampilkan seluruh data mainan yang tersimpan dalam sistem. Selain menampilkan detail setiap mainan (nomor, nama, jenis, harga, kondisi), fitur ini juga menghitung dan menampilkan total harga dari semua mainan menggunakan fungsi rekursif.

### Source Code:

```

def tampilan_mainan():
    if len(mainan) == 0:
        print("Belum ada data mainan\n")
    else:
        print("\n==== Koleksi Mainan ===")
        for i, data in mainan.items():
            print(f"{i}. {data['nama']} | {data['jenis']} | "
Rp{data['harga']:,} | {data['kondisi']}")
        print("")

```

## C.Fitur Ubah Mainan (Update)

Fitur ini berguna untuk mengubah data mainan yang sudah ada. User akan melihat daftar mainan terlebih dahulu, kemudian memilih nomor mainan yang ingin diubah. Setelah itu, user dapat memasukkan data baru (nama, jenis, harga, kondisi) untuk menggantikan data lama.

### Source Code:

```

def ubah_mainan():
    tampilan_mainan()

```

```
try:
    i = int(input("Masukkan nomor mainan: "))
    if i not in mainan:
        raise Exception("Mainan tidak ditemukan")
    mainan[i]["nama"] = input("Nama baru: ")
    mainan[i]["jenis"] = input("Jenis baru: ")
    mainan[i]["harga"] = input_angka("Harga baru: ")
    mainan[i]["kondisi"] = input("Kondisi baru: ")
    print("Berhasil diubah!\n")
except Exception as e:
    print(e, "\n")
```

#### D.Fitur Hapus Mainan (Delete)

Fitur ini berguna untuk menghapus data mainan dari sistem. User akan melihat daftar mainan, kemudian memasukkan nomor mainan yang ingin dihapus. Sistem akan langsung menghapus data tersebut dari koleksi.

#### Source Code:

```
def hapus_mainan():
    tampilan_mainan()
    try:
        i = int(input("Masukkan nomor mainan: "))
        if i not in mainan:
            raise Exception("Mainan tidak ditemukan")
        del mainan[i]
        print("Berhasil dihapus!\n")
    except Exception as e:
        print(e, "\n")
```

## E.Fungsi Hitung Total Rekursif

Fungsi ini berguna untuk menghitung total harga dari semua mainan dalam koleksi menggunakan metode rekursif. Fungsi akan menjumlahkan harga satu per satu dari list key mainan hingga mencapai akhir list.

### Source Code:

```
def hitung_total_rekursif(keys, index=0):
    if index >= len(keys):
        return 0
    return mainan[keys[index]]["harga"] + hitung_total_rekursif(keys, index
+ 1)
```

#### 4.Hasil Program

```
==== Tambah Mainan ====
Nama: Bang Pernan
Jenis: Action Figure
Harga: 500000
Kondisi: Baru
Mainan berhasil ditambahkan!
```

Gambar 4.1 Hasil Fitur Tambah Data

```
==== Koleksi Mainan ====
1. LEGO Star Wars Millennium Falcon | LEGO | Rp1,500,000 | Baru
2. Iron Man Mark 85 | Action Figure | Rp350,000 | Bekas
3. Bang Pernan | Action Figure | Rp500,000 | Baru

Total Harga Semua Mainan: Rp2,350,000
```

Gambar 4.2 Hasil Fitur Lihat Koleksi

```
==== Koleksi Mainan ====
1. LEGO Star Wars Millennium Falcon | LEGO | Rp1,500,000 | Baru
2. Iron Man Mark 85 | Action Figure | Rp350,000 | Bekas
3. Bang Pernan | Action Figure | Rp500,000 | Baru

Masukkan nomor mainan: 2
Nama baru: Beruang
Jenis baru: LEGO
Harga baru: 3000000
Kondisi baru: Baru
Berhasil diubah!
```

Gambar 4.3 Hasil Ubah Mainan

```
==== Koleksi Mainan ====
1. LEGO Star Wars Millennium Falcon | LEGO | Rp1,500,000 | Baru
2. Beruang | LEGO | Rp3,000,000 | Baru
3. Bang Pernan | Action Figure | Rp500,000 | Baru

Masukkan nomor mainan: 3
Berhasil dihapus!
```

Gambar 4.4 Hasil Fitur Hapus Mainan

## 5. Langkah-langkah GIT

### 5.1 GIT Add

Berfungsi untuk menyimpan semua perubahan yang telah dilakukan oleh user, baik penambahan file, penghapusan, atau perubahan isi file ke dalam staging area sebelum dilakukan commit.

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

Gambar 5.1 GIT Add

### 5.2 GIT Commit

Berfungsi untuk menyimpan perubahan yang ada di staging area ke dalam local repository beserta pesan atau keterangan mengenai perubahan tersebut.

```
PS C:\Users\User\Documents\praktikum-apd> git commit -m "last pt"
[main 0cdd510] last pt
 3 files changed, 32 insertions(+)
 create mode 100644 kelas/Pertemuan 8.py
 create mode 100644 post-test/post-test-apd-8/2509106076-MuhammadFachriAkbar-PT-8.pdf
 create mode 100644 post-test/post-test-apd-8/2509106076-MuhammadFachriAkbar-PT-8.py
```

Gambar 5.2 GIT Commit

### 5.3 GIT Push

Berfungsi untuk mengirim dan menerapkan perubahan (commit) yang ada di local repository ke dalam remote repository seperti GitHub.

```
PS C:\Users\User\Documents\praktikum-apd> git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 691.45 KiB | 8.86 MiB/s, done.
```

Gambar 5.3 GIT Push