LAPORAN APLIKASI ADMIN PINJOL

PEMROGRAMAN BERORIENTASI OBJEK PRAKTIK

T. A. Semester Ganjil 2024/2025

5230411302 AYU KUSUMA SETYANINGSIH

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS & TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

YOGYAKARTA

2024

A. PENDAHULUAN

Aplikasi Admin Pinjol merupakan aplikasi berbasis teks yang digunakan untuk mengelola data debitur serta transaksi pinjaman. Aplikasi ini dibuat menggunakan bahasa pemrograman Python dan bertujuan untuk menyederhanakan proses administrasi pada sistem pinjaman online. Melalui aplikasi ini, pengguna dapat menambahkan, mencari, serta menampilkan daftar debitur, sekaligus mengelola pinjaman berdasarkan limit pinjaman yang dimiliki setiap debitur.

Sistem ini dilengkapi dengan fitur validasi untuk memastikan bahwa transaksi yang dilakukan sesuai dengan aturan yang berlaku, misalnya batas pinjaman tidak boleh melebihi limit yang telah ditentukan.

B. LANDASAN TEORI

Landasan teori aplikasi ini meliputi konsep pemrograman berbasis fungsi untuk memecah program menjadi bagian modular, serta penggunaan list dan dictionary untuk mengelola data debitur dan pinjaman. Validasi input diterapkan untuk memastikan data yang dimasukkan valid, seperti cek limit pinjaman dan KTP debitur. Percabangan dan perulangan digunakan untuk navigasi menu, sedangkan konsep perhitungan pinjaman melibatkan penambahan bunga pada pinjaman dan penghitungan angsuran bulanan.

C. IMPLEMENTASI

I. Struktur data

```
debitur_list = [
    {"nama": "Itsuki", "ktp": "234", "limit_pinjaman": 22000000},
    {"nama": "Ryuki", "ktp": "345", "limit_pinjaman": 1800000},
    {"nama": "Sakuya", "ktp": "147", "limit_pinjaman": 6200000},
    {"nama": "Yushi", "ktp": "246", "limit_pinjaman": 8000000},
    {"nama": "Sion", "ktp": "987", "limit_pinjaman": 2000000}
]
```

List ini menyimpan data debitur yang berupa dictionary. Setiap debitur memiliki atribut nama, ktp, dan limit_pinjaman

II. Menampilkan semua debitur

```
def tampilkan_semua_debitur():
```

```
    print("\n========== Daftar Debitur ==========")
    print(f"{'Nama Debitur':<15} {'No KTP':<10} {'Limit Pinjaman'}")
    print("-------------------------------------")
    for debitur in debitur_list:
        print(f"{debitur['nama']:<15}                    {debitur['ktp']:<10}
Rp.{debitur['limit_pinjaman']:,}")
    print("=============================================")
```

Fungsi ini menampilkan seluruh daftar debitur yang ada dalam list debitur_list, dengan format yang rapi menggunakan f-string.

III. Mencari debitur

```
def cari_debitur(nama):
    for debitur in debitur_list:
        if debitur['nama'].lower() == nama.lower():
            return debitur
    return None
```

Fungsi ini mencari debitur berdasarkan nama yang dimasukkan oleh pengguna. Jika ditemukan, fungsi akan mengembalikan data debitur, jika tidak, fungsi akan mengembalikan None

IV. Menambah debitur

```
def tambah_debitur():
    nama = input("Masukkan Nama Debitur: ")
    ktp = input("Masukkan No KTP Debitur: ")
    limit_pinjaman = int(input("Masukkan Limit Pinjaman: "))

    for debitur in debitur_list:
        if debitur["ktp"] == ktp:
            print("Debitur dengan KTP ini sudah ada. Gagal menambah debitur.")
            return

    debitur_list.append({"nama": nama, "ktp": ktp, "limit_pinjaman":
limit_pinjaman})
```

```
    print("Debitur berhasil ditambahkan.")
```

Fungsi ini memungkinkan pengguna untuk menambah debitur baru. Fungsi juga melakukan validasi apakah KTP debitur yang ingin ditambahkan sudah ada atau belum.

V. Menambah pinjaman

```
def tambah_pinjaman():
    nama = input("Masukkan Nama Debitur: ")
    debitur = cari_debitur(nama)

    if debitur:
        pinjaman = int(input("Masukkan Jumlah Pinjaman: "))

        if pinjaman > debitur['limit_pinjaman']:
            print("Pinjaman melebihi limit! Gagal menambahkan pinjaman.")
        else:
            bunga = float(input("Masukkan Bunga (%): "))
            bulan = int(input("Masukkan Jangka Waktu (bulan): "))
            angsuran = round((pinjaman + (pinjaman * bunga / 100)) / bulan)
            pinjaman_list.append({"nama": nama, "pinjaman": pinjaman, "bunga":
bunga, "bulan": bulan, "angsuran": angsuran})
            print("Pinjaman berhasil ditambahkan.")
    else:
        print("Debitur tidak ditemukan. Gagal menambahkan pinjaman.")
```

Fungsi ini digunakan untuk menambah pinjaman. Sistem melakukan pengecekan apakah pinjaman melebihi limit yang ditentukan dan menghitung angsuran bulanan dengan mempertimbangkan bunga.

VI. Menampilkan pinjaman

```
def tampilkan_pinjaman():
    print("\n========================= Daftar Pinjaman
=========================")
```

```
    print(f"{'Nama Debitur':<15} {'Pinjaman':<10} {'Bunga (%)':<10}
{'Bulan':<6} {'Angsuran / Bulan'}")
    print("-----------------------------------------------------------------")
    for pinjaman in pinjaman_list:
        print(f"{pinjaman['nama']:<15} Rp.{pinjaman['pinjaman']:<10}
{pinjaman['bunga']:<10} {pinjaman['bulan']:<6}
Rp.{pinjaman['angsuran']:,}")

print("================================================================
================")
```

Fungsi ini menampilkan daftar pinjaman yang telah ditambahkan, termasuk informasi debitur, jumlah pinjaman, bunga, jangka waktu, dan angsuran bulanan.

VII. Menu Utama

```
def menu_utama():
    while True:
        print("\n========== Aplikasi Admin Pinjol ==========")
        print("1. Kelola Debitur")
        print("2. Pinjaman")
        print("0. Keluar")
        pilihan = input("Masukkan Pilihan: ")

        if pilihan == "1":
            kelola_debitur()
        elif pilihan == "2":
            menu_pinjaman()
        elif pilihan == "0":
            print("Keluar dari program...")
            break
        else:
            print("Pilihan tidak valid, coba lagi.")
```

VIII. Menu Kelola Debitur

```
def kelola_debitur():
    while True:
        print("\n========== Kelola Debitur ==========")
        print("1. Tampilkan Semua Debitur")
        print("2. Cari Debitur")
        print("3. Tambah Debitur")
        print("0. Kembali")
        pilihan = input("Masukkan Pilihan Sub Menu: ")

        if pilihan == "1":
            tampilkan_semua_debitur()
        elif pilihan == "2":
            nama = input("Masukkan Nama Debitur: ")
            debitur = cari_debitur(nama)
            if debitur:
                print(f"\nNama: {debitur['nama']}\nNo KTP: {debitur['ktp']}\nLimit
Pinjaman: Rp.{debitur['limit_pinjaman']:,}")
            else:
                print("Debitur tidak ditemukan.")
        elif pilihan == "3":
            tambah_debitur()
        elif pilihan == "0":
            break
        else:
            print("Pilihan tidak valid, coba lagi.")
```

IX. Menu Pinjaman

```
def menu_pinjaman():
    while True:
        print("\n========== Menu Pinjaman ==========")
        print("1. Tambah Pinjaman")
        print("2. Tampilkan Pinjaman")
        print("0. Kembali")
        pilihan = input("Masukkan Pilihan Sub Menu: ")
```

```
        if pilihan == "1":
            tambah_pinjaman()
        elif pilihan == "2":
            tampilkan_pinjaman()
        elif pilihan == "0":
            break
        else:
            print("Pilihan tidak valid, coba lagi.")

menu_utama()
```

D. KESIMPULAN

Aplikasi ini berhasil mengimplementasikan sistem pengelolaan debitur dan pinjaman secara efektif menggunakan konsep-konsep dasar pemrograman seperti pemrograman berbasis fungsi, struktur data list dan dictionary, validasi input, serta perhitungan pinjaman dengan bunga. Dengan fitur yang memungkinkan pengelolaan data debitur dan penambahan pinjaman yang terintegrasi, aplikasi ini memberikan kemudahan dalam administrasi pinjaman dan membantu memastikan data yang dikelola tetap valid serta terstruktur dengan baik.