

**LAPORAN APLIKASI**  
**PEMPROGRAMAN BERBASIS OBJEK**  
**KELAS VIII**

T. A. Semester Ganjil 2024/2025



**5230411286**

**FIRMANTI ALHILMA S.**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**YOGYAKARTA**  
**2024**

## A. Perintah Program

### 1. Kelola Debitur

- Tampilkan Debitur Nama(public), KTP (Private), Limit Pinjaman (Protected)
- Cari Debitur berdasarkan nama >> (Nama, KTP, Limit Pinjaman)
- Tambah Debitur berdasar KTP (jika KTP sama = menampilkan validasi)

### 2. Kelola Pinjaman (Debitur)

- Tambah Pinjaman berdasarkan Nama (jika Nama tidak ada = menampilkan validasi gagal, jika pinjaman > Limit = menampilkan validasi gagal)
- Tampilkan pinjaman (Nama, Pinjaman, Bunga, Bulan, Angsuran(/bulan)

Nama angsuran :

Angsuran pokok = pinjaman \* bunga%

Angsuran Bulanan = angsuran pokok/bulan

Total Angsuran = Angsuran pokok =

Angsuran bulan

## B. Code Program

```
class Debitur:
    def __init__(self, nama, ktp, limit_pinjaman):
        self.nama = nama
        self.__ktp = ktp
        self._limit_pinjaman = limit_pinjaman

    def get_info(self):
        return (self.nama, self.__ktp, self._limit_pinjaman)

class Pinjaman:
    def __init__(self, debitur, pinjaman, bunga, bulan):
        self.debitur = debitur
        self.pinjaman = pinjaman
        self.bunga = bunga
        self.bulan = bulan

    def hitung_angsuran(self):
        angsuran_pokok = self.pinjaman * (self.bunga / 100)
        angsuran_bulanan = (angsuran_pokok + self.pinjaman) /
self.bulan
        total_angsuran = angsuran_bulanan * self.bulan
        return angsuran_pokok, angsuran_bulanan, total_angsuran

class ManajemenDebitur:
```

```

def __init__(self):
    self.debitur_list = []

def tambah_debitur(self, nama, ktp, limit_pinjaman):
    if any(deb._Debitur__ktp == ktp for deb in self.debitur_list):
        print("KTP sudah ada")
        return False

    new_debitur = Debitur(nama, ktp, limit_pinjaman)
    self.debitur_list.append(new_debitur)
    print("Debitur berhasil ditambahkan.")
    return True

def cari_debitur(self, nama):
    for deb in self.debitur_list:
        if deb.nama == nama:
            return deb.get_info()
    print("Debitur tidak ditemukan")
    return None

def tampilkan_semua_debitur(self):
    if not self.debitur_list:
        print("Tidak ada debitur yang terdaftar")
        return

    print("\n===== Daftar Debitur =====")
    for deb in self.debitur_list:
        info = deb.get_info()
        print(f>Nama: {info[0]}, KTP: {info[1]}, Limit Pinjaman: {info[2]})

class ManajemenPinjaman:
    def __init__(self):
        self.pinjaman_list = []

    def tambah_pinjaman(self, debitur_nama, pinjaman, bunga,
bulan, manajemen_debitur):
        debitur_info = manajemen_debitur.cari_debitur(debitur_nama)

        if debitur_info is None:
            print("Nama debitur tidak ditemukan")
            return False

        if pinjaman > debitur_info[2]:
            print("Pinjaman melebihi limit")

```

```

        return False

    new_pinjaman = Pinjaman(debitur_info[0], pinjaman, bunga,
                               bulan)
    self.pinjaman_list.append(new_pinjaman)

    print("Pinjaman berhasil ditambahkan.")
    return True

def tampilkan_pinjaman(self):
    if not self.pinjaman_list:
        print("Tidak ada pinjaman yang terdaftar.")
        return

    print("\n=== Daftar Pinjaman ===")
    for pinj in self.pinjaman_list:
        angsuran_pokok, angsuran_bulanan, total_angsuran =
        pinj.hitung_angsuran()
        print(f>Nama: {pinj.debitur}, Pinjaman: {pinj.pinjaman},
              Bunga: {pinj.bunga}, Bulan: {pinj.bulan}, "
              f'Angsuran Pokok: {angsuran_pokok:.2f}, Angsuran
              Bulanan: {angsuran_bulanan:.2f}, "
              f'Total Angsuran: {total_angsuran:.2f}')

    manajemen_debitur = ManajemenDebitur()
    manajemen_debitur.tambah_debitur("Alhilma", "123", 1230000)
    manajemen_debitur.tambah_debitur("Firman", "987", 8880000)
    manajemen_debitur.tambah_debitur("Afra", "456", 990000)
    manajemen_debitur.tambah_debitur("Salsa", "218", 1110000)
    manajemen_debitur.tambah_debitur("Bila", "151", 1110000)

    manajemen_pinjam = ManajemenPinjaman()

    while True:
        print("\n===== Aplikasi Admin Pinjol =====")
        print("1. Kelola Debitur")
        print("2. Kelola Pinjaman")
        print("3. Keluar")
        pilihan_menu_utama = input("Pilih opsi (1-3): ")

        if pilihan_menu_utama == '1':
            while True:
                print("\n===== Menu Debitur =====")
                print("1. Tampilkan Semua Debitur")
                print("2. Cari Debitur")
                print("3. Tambah Debitur")

```

```

print("4. Kembali")
pilihan_menu_debitur = input("Pilih opsi (1-4): ")

if pilihan_menu_debitur == '1':
    manajemen_debitur.tampilkan_semua_debitur()

elif pilihan_menu_debitur == '2':
    nama = input("Masukkan Nama Debitur yang dicari: ")
    info_debitur = manajemen_debitur.cari_debitur(nama)
    if info_debitur:
        print(f'Debitur Ditemukan: Nama: {info_debitur[0]},
KTP: {info_debitur[1]}, Limit Pinjaman: {info_debitur[2]}')

    elif pilihan_menu_debitur == '3':
        nama = input("Masukkan Nama Debitur: ")
        ktp = input("Masukkan KTP Debitur: ")
        limit_pinjaman = float(input("Masukkan Limit
Pinjaman: "))
        manajemen_debitur.tambah_debitur(nama, ktp,
limit_pinjaman)

    elif pilihan_menu_debitur == '4':
        break

    else:
        print("Pilihan tidak valid. Silakan coba lagi.")

elif pilihan_menu_utama == '2':
    while True:
        print("\n===== Menu Pinjaman =====")
        print("1. Tambah Pinjaman")
        print("2. Tampilkan Pinjaman")
        print("3. Kembali")

        pilihan_menu_pinjaman = input("Pilih opsi (1-3): ")

        if pilihan_menu_pinjaman == '1':
            debitur_nama = input("Masukkan Nama Debitur untuk
Tambah Pinjaman: ")
            pinjaman = float(input("Masukkan Jumlah Pinjaman:
"))
            bunga = float(input("Masukkan Bunga (%): "))
            bulan = int(input("Masukkan Lama Angsuran (bulan):
"))

```

```
        manajemen_pinjam.tambah_pinjaman(debitur_nama,
pinjaman, bunga, bulan, manajemen_debitur)

        elif pilihan_menu_pinjaman == '2':
            manajemen_pinjam.tampilkan_pinjaman()

        elif pilihan_menu_pinjaman == '3':
            break

        else:
            print("Pilihan tidak valid. Silakan coba lagi.")

    elif pilihan_menu_utama == '3':
        print("Terimakasih Telah menggunakan program ini")
        break

    else:
        print("Pilihan tidak valid. Silakan coba lagi.")
```

Dari program diatas bisa dideskripsikan sebagai berikut:

1. Class Debitur

Digunakan untuk merepresentasikan data debitur (nasabah) yang ingin meminjam uang. Ini mencakup informasi dasar seperti nama, nomor KTP, dan limit pinjaman yang diizinkan.

- a. `__init__(self, nama, ktp, limit_pinjaman)` : konstruktor yang menginisialisasi objek dengan nama, KTP, dan limit pinjaman.
- b. `get_info(self)` : digunakan untuk mengembalikan informasi debitur dalam bentuk tuple yang berisi nama, KTP, dan limit pinjaman. Ini memungkinkan akses data debitur secara terstruktur.

2. Class Pinjaman

Digunakan untuk merepresentasikan informasi tentang pinjaman yang diberikan kepada debitur. Ini mencakup detail seperti jumlah pinjaman, bunga, dan durasi pinjaman.

- a. `__init__(self, debitur, pinjaman, bunga, bulan)` : konstruktor yang menginisialisasi objek dengan debitur yang meminjam, jumlah pinjaman, bunga, dan lama angsuran dalam bulan.

- b. `Hitung_angsuran` : digunakan untuk menghitung angsuran berdasarkan jumlah pinjaman dan bunga. Ini mengembalikan tiga nilai yaitu angsuran pokok, angsuran bulanan, dan total angsuran.

### 3. Class ManajemenDebitur

Digunakan untuk mengelola daftar debitur. Ini termasuk menambah debitur baru, mencari debitur berdasarkan nama, dan menampilkan semua debitur yang terdaftar

- a. `__init__(self)` : konstruktor yang menginisialisasi list kosong untuk menyimpan objek debitur (`debitur_list`)
- b. `tambah_debitur(self, nama, ktp, limit_pinjaman)` : digunakan untuk menambahkan debitur baru kedalam daftar jika KTP belum ada di daftar. Jika KTP sudah ada, akan mencetak pesan kesalahan.
- c. `cari_debitur(self, nama)` : digunakan mencari debitur berdasarkan nama, jika ditemukan ia akan mengembalikan informasi debitur, jika tidak ditemukan akan mencetak pesan kesalahan.
- d. `Tampilan_semua_debitur(self)` : digunakan untuk menampilkan semua debitur yang terdaftar beserta informasi mereka, jika tidak ada debitur yang terdaftar, akan mencetak pesan bahwa tidak ada debitur

### 4. Class ManajemenPinjaman

Digunakan untuk mengelola daftar pinjaman yang diberikan kepada debitur. Ini termasuk menambah pinjaman baru dan menampilkan semua pinjaman yang terdaftar.

- a. `__init__(self)` : konstruktor yang menginisialisasi list kosong untuk menyimpan objek pinjaman (`pinjaman_list`)
- b. `Tambah_pinjaman(self, debitur_nama, pinjaman, bunga, bulan_manajemen_debitur)` : untuk menambahkan pinjaman baru jika nama debitur ditemukan dan jumlah pinjaman tidak melebihi limit. Jika salah satu kondisi tidak terpenuhi, akan mencetak pesan kesalahan.
- c. `Tampilkan_pinjaman(self)` : metode ini menampilkan semua Riwayat transaksi pinjaman beserta detailnya seperti angsuran pokok dan total angsuran.

## C. Hasil Running

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS C:\file semester 3\pbo praktik\perempat> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe C:/Users/LENOVO/AppData/Local/Programs/Python/Python38-64/Scripts/python.exe ik/perempat/bismilah.py
Debitur berhasil ditambahkan.
Debitur berhasil ditambahkan.
Debitur berhasil ditambahkan.
Debitur berhasil ditambahkan.
Debitur berhasil ditambahkan.

===== Aplikasi Admin Pinjol =====
1. Kelola Debitur
2. Kelola Pinjaman
3. Keluar
Pilih opsi (1-3): 1

===== Menu Debitur =====
1. Tampilkan Semua Debitur
2. Cari Debitur
3. Tambah Debitur
4. Kembali
Pilih opsi (1-4): 1

===== Daftar Debitur =====
Nama: Alhilma, KTP: 123, Limit Pinjaman: 1230000
Nama: Firman, KTP: 987, Limit Pinjaman: 8880000
Nama: Afra, KTP: 456, Limit Pinjaman: 990000
Nama: Salsa, KTP: 218, Limit Pinjaman: 1110000
Nama: Bila, KTP: 151, Limit Pinjaman: 1110000

===== Menu Debitur =====
1. Tampilkan Semua Debitur
2. Cari Debitur
3. Tambah Debitur
4. Kembali
Pilih opsi (1-4): 2
Masukkan Nama Debitur yang dicari: Salsa
Debitur Ditemukan: Nama: Salsa, KTP: 218, Limit Pinjaman: 1110000
```



```

===== Menu Debitur =====
1. Tampilkan Semua Debitur
2. Cari Debitur
3. Tambah Debitur
4. Kembali
Pilih opsi (1-4): 3
Masukkan Nama Debitur: Cici
Masukkan KTP Debitur: 710
Masukkan Limit Pinjaman: 200000
Debitur berhasil ditambahkan.

===== Menu Debitur =====
1. Tampilkan Semua Debitur
2. Cari Debitur
3. Tambah Debitur
4. Kembali
Pilih opsi (1-4): 4

===== Aplikasi Admin Pinjol =====
1. Kelola Debitur
2. Kelola Pinjaman
3. Keluar
Pilih opsi (1-3): 2

===== Menu Pinjaman =====
1. Tambah Pinjaman
2. Tampilkan Pinjaman
3. Kembali
Pilih opsi (1-3): 1
Masukkan Nama Debitur untuk Tambah Pinjaman: Cici
Masukkan Jumlah Pinjaman: 30000
Masukkan Bunga (%): 20
Masukkan Lama Angsuran (bulan): 12
Pinjaman berhasil ditambahkan.

```

```

===== Menu Pinjaman =====
1. Tambah Pinjaman
2. Tampilkan Pinjaman
3. Kembali
Pilih opsi (1-3): 2

=== Daftar Pinjaman ===
Nama: Cici, Pinjaman: 30000.0, Bunga: 20.0, Bulan: 12, Angsuran Pokok: 6000.00, Angsuran Bulanan: 3000.00, Total Angsuran: 36000.00

===== Menu Pinjaman =====
1. Tambah Pinjaman
2. Tampilkan Pinjaman
3. Kembali
Pilih opsi (1-3): 3

===== Aplikasi Admin Pinjol =====
1. Kelola Debitur
2. Kelola Pinjaman
3. Keluar
Pilih opsi (1-3): 1

===== Menu Debitur =====
1. Tampilkan Semua Debitur
2. Cari Debitur
3. Tambah Debitur
4. Kembali
Pilih opsi (1-4): 2
Masukkan Nama Debitur yang dicari: 123
Debitur tidak ditemukan

```