

# **TUGAS INDIVIDU**

## **PEWARISAN DAN HAK AKSES**

### **PEMROGRAMAN BERORIENTASI OBJEK KELAS VIII**



**Dibuat Oleh:**

**Aji Kharisma Atmaja (5230411292)**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2024**

## 1. Class Debitur

```
class debitur:
    def __init__(self, nama, ktp, limit):
        self.nama = nama
        self.__ktp = ktp
        self._limit = limit
```

Dari class itu terdapat fungsi init yang memiliki parameter di dalamnya di antaranya ada nama,ktp,dan juga limit.Atribut kelas ini ada self.nama yang bersifat public dapat di akses dimana saja dan menyimpan parameter nama ke dalam atribut instance nama.Kemudian self.\_\_ktp yang bersifat privat tidak dapat diakses secara langsung dari luar kelas dan menyimpan nilai parameter ktp ke dalam atribut instance \_\_ktp.Terakhir ada self.\_limit yang bersifat protected dapat diakses dari dalam kelas dan subclass dan menyimpan nilai parameter limit ke dalam atribut instance \_limit.

## 2. Fungsi Display

```
def display(self):
    print(f>Nama Debitur: {self.nama} dengan No KTP: {self.__ktp}
    dengan limit pinjaman: {self._limit}")
```

Fungsi ini berfungsi untuk menampilkan atau format tampilan apa bila kita ingin print data debitur yang ada.

## 3. Fungsi Tambah Debitur

```
def tambah_debitur(self):
    self.__ktp = ktp
    self._limit = limit
    self.nama = nama
```

Fungsi ini berfungsi menambahkan data pada debitur dimana di dalamnya ada bagian untuk mengatur atribut masing-masing.

## 4. Fungsi Cari Debitur

```
def cari_debitur(nama_dicari, daftar_debitur):
    for debitur in daftar_debitur:
        if debitur.nama.lower() == nama_dicari.lower():
            return debitur
    return None
```

Fungsi ini berfungsi untuk mencari data debitur sesuai dengan nama yang kita inputkan,kemudian akan di cek apakah nama yang kita inputkan ada di daftar debitur,jika nama yang di cari sama dengan daftar debitur nama maka akan menampilkan debitur yang kita cari,disitu ada lower agar jika ada penulisan typo huruf besar nya masih dapat di running contohnya “Rahmad” sama dengan “rahmad”.

## 5. Class Pinjaman

```
class pinjaman:
    def __init__(self, nama, ktp, limit, pinjaman=0, bunga=0, bulan=0,
    angsuran=0):
        self.nama = nama
        self.__ktp = ktp
        self._limit = limit
        self.pinjaman = pinjaman
```

```

self.bunga = bunga
self.bulan = bulan
self.angsuran = angsuran

```

Pada class tersebut terdapat parameter yang sama dengan class debitur, namun ada beberapa tambahan yaitu pinjaman, bunga, bulan, dan angsuran. Disitu saya membuat jumlah defaultnya 0. Menyimpan jumlah pinjaman, bunga, bulan dan angsuran pada atribut masing-masing sesuai nama atributnya.

## 6. Fungsi Display

```

def display(self):
    angsuran_pokok = self.pinjaman * (self.bunga / 100)
    angsuran_bulanan = angsuran_pokok / self.bulan
    total_angsuran = angsuran_pokok + angsuran_bulanan
    print(f>Nama Debitur: {self.nama}")
    print(f"Pinjaman: {self.pinjaman}, Bunga: {self.bunga}%")
    print(f"Bulan: {self.bulan}, Angsuran Bulanan:
{angsuran_bulanan:.2f}")
    print(f>Total Angsuran: {total_angsuran:.2f}")

```

Fungsi ini berfungsi untuk menampilkan pinjaman dimana di dalamnya juga sudah ada rumus yang sudah ditentukan kemudian ada juga format dalam ngeprint jumlah pinjaman.

## 7. Fungsi Tambah Pinjaman

```

def tambah_pinjaman(self):
    nama = input("Masukkan nama debitur: ")
    for debitur in tampil_debitur:
        if debitur.nama == nama:
            nominal_pinjaman = int(input("Masukkan nominal yang
ingin dipinjam: "))
            bunga = int(input("Masukkan bunga (%): "))
            bulan = int(input("Masukkan jumlah bulan: "))
            if nominal_pinjaman <= debitur._limit:
                pinjaman_baru = pinjaman(nama,
debitur._debitur__ktp, debitur._limit, nominal_pinjaman, bunga, bulan)
                list_pinjaman.append(pinjaman_baru)
                print("Pinjaman berhasil ditambahkan")
                return
            else:
                print("Pinjaman melebihi limit")
                return
        else:
            print("Debitur dengan nama tersebut tidak ditemukan")

```

Fungsi ini berfungsi untuk menambahkan pinjaman sesuai nama pada debitur, apabila nama tidak ada pada debitur maka akan ada peringatan nama tidak ditemukan, jika ada maka akan disuruh menginputkan yang harus di inputkan sesuai code di atas, namun disini ada validasi lagi apabila jumlah pinjaman lebih dari limit maka akan ada peringatan pinjaman melebihi limit.

## 8. Deskripsi Debitur dan Pinjaman

```
tampil_debitur=[
    debitur("Rahmad", 123, 10000000),
    debitur("Yoyo", 321, 10000000),
    debitur("Yiyi", 100, 10000000),
]

list_pinjaman = []
```

Code di atas saya membuat untuk daftar debitur nya ada default nya yaitu ada 3 yang di tamping dalam list tampil\_debitur sedangkan pinjaman saya kosongkan karena kita akan print pinjaman apabila sudah melakukan tambah pinjaman.

## 9. Menu utama

```
while True:
    print("Aplikasi Pinjaman Online")
    print("1. KELOLA DEBITUR")
    print("2. KELOLA PINJAMAN")
    print("3. KELUAR")
    pilihan = input("Pilih menu: ")
```

Itu merupakan menu utama pada program saya.

## 10. Sub Menu Debitur

```
if pilihan == "1":
    while True:
        print("KELOLA DEBITUR")
        print("1. TAMPIL DEBITUR")
        print("2. CARI DEBITUR")
        print("3. TAMBAH DEBITUR")
        print("4. KELUAR")
        pilihan2 = input("Pilih menu: ")
        if pilihan2 == "1":
            for debitur in tampil_debitur:
                debitur.display()
```

Itu merupakan menu-menu dalam sub menu kelola debitur, apabila memilih satu maka akan memanggil fungsi display pada class debitur untuk menampilkan data debitur yang ada.

### A. Sub Menu 2

```
elif pilihan2 == "2":
    nama_dicari = input("Masukkan nama: ")
    debitur = cari_debitur(nama_dicari,
tampil_debitur)
    if debitur:
        debitur.display()
    else:
        print("Debitur dengan nama tersebut tidak
ditemukan")
```

Pada sub menu ini kita akan mencari debitur sesuai nama yang kita inputkan kemudian akan di cek, jika ada maka akan ngeprint debitur sesuai nama yang kita inputkan jika tidak akan muncul validasi nama tidak di temukan.

### B. Sub Menu 3

```
elif pilihan2 == "3":
    ktp = int(input("Masukkan No KTP: "))
    if any(debitur._debitur__ktp == ktp for debitur in
tampil_debitur):
        print("Debitur dengan No KTP tersebut sudah
ada")
    else:
        nama = input("Masukkan nama: ")
        limit = int(input("Masukkan limit: "))
        debitur_baru = debitur(nama, ktp, limit)
        tampil_debitur.append(debitur_baru)
        print("Debitur berhasil ditambahkan")
```

Pada menu ini kita akan menambahkan debitur dengan inputkan ktp dahulu dan akan di cek apabila ktp yang kita inputkan sudah ada atau sama dengan ktp yang ada pada list tampil\_debitur maka akan muncul validasi ktp sudah ada, jika belum maka kita akan menginputkan nama, limit yang kemudian di tambahkan ke tampil\_debitur.

### C. Sub Menu 4

```
elif pilihan2 == "4":
    break
else:
    print("Pilihan tidak ada")
```

Pada menu ini apabila memilih no 4 akan kembali ke menu utama apabila ada kesalahan input akan muncul pilihan tidak ada

## 11. Sub Menu Kelola Pinjaman

```
elif pilihan == "2":
    while True:
        print("KELOLA PINJAMAN")
        print("1. TAMBAH PINJAMAN")
        print("2. TAMPIL PINJAMAN")
        print("3. KELUAR")
        pilih = input("Pilih menu: ")
        if pilih == "1":
            pinjaman_obj = pinjaman("", 0, 0, 0)
            pinjaman_obj.tambah_pinjaman()
```

Pada menu ini ada beberapa menu dalam menu kelola pinjaman dan yang pertama tambah pinjaman. Disini saya membuat objek baru pinjaman\_obj. Dengan menggunakan kode ini, Saya membuat sebuah objek dari kelas pinjaman dan kemudian memanggil metode untuk menambah data pinjaman ke dalam objek tersebut. Namun, karena Saya menginisialisasi objek dengan nilai kosong, Saya perlu memasukkan data yang valid saat menjalankan metode tambah\_pinjaman().

### A. Sub Menu 2

```
elif pilih == "2":
    if not list_pinjaman:
        print("Belum ada pinjaman yang ditambahkan.")
```

```

else:
    for pinjaman in list_pinjaman:
        pinjaman.display()

```

Pada menu ini akan menampilkan data pinjaman, apabila belum ada data yang kita inputkan maka akan muncul belum ada pinjaman yang ditambahkan jika sudah maka akan memanggil display pinjaman.

#### B. Sub Menu 3

```

elif pilih == "3":
    break
else:
    print("Pilihan tidak ada")

```

Pada menu ini akan kembali ke menu utama dan apabila inputan kita salah maka akan muncul pilihan tidak ada.

#### 12. Menu Terakhir

```

elif pilihan == "3":
    print("TERIMA KASIH TELAH MENGGUNAKAN PROGRAM, TAPI JANGAN SAMPAI  
PINJOL BENERAN YAA!!!")
    break
else:
    print("Pilihan tidak ada")

```

Pada menu ini akan muncul ucapan terimakasih dan akan keluar dari program.

### HASIL RUNNING:

#### 1. Tampil Debitur

```

Aplikasi Pinjaman Online
1. KELOLA DEBITUR
2. KELOLA PINJAMAN
3. KELUAR
Pilih menu: 1
KELOLA DEBITUR
1. TAMPIL DEBITUR
2. CARI DEBITUR
3. TAMBAH DEBITUR
4. KELUAR
Pilih menu: 1
Nama Debitur: Rahmad dengan No KTP: 123 dengan limit pinjaman: 10000000
Nama Debitur: Yoyo dengan No KTP: 321 dengan limit pinjaman: 10000000
Nama Debitur: Yiyi dengan No KTP: 100 dengan limit pinjaman: 10000000
KELOLA DEBITUR
1. TAMPIL DEBITUR
2. CARI DEBITUR
3. TAMBAH DEBITUR
4. KELUAR
Pilih menu: █

```

#### 2. Cari Debitur

```

KELOLA DEBITUR
1. TAMPIL DEBITUR
2. CARI DEBITUR
3. TAMBAH DEBITUR
4. KELUAR
Pilih menu: 2
Masukkan nama: Aji
Debitur dengan nama tersebut tidak ditemukan
KELOLA DEBITUR
1. TAMPIL DEBITUR
2. CARI DEBITUR
3. TAMBAH DEBITUR
4. KELUAR
Pilih menu: 2
Masukkan nama: rahmad
Nama Debitur: Rahmad dengan No KTP: 123 dengan limit pinjaman: 10000000

```

### 3. Tambah Debitur

```

KELOLA DEBITUR
1. TAMPIL DEBITUR
2. CARI DEBITUR
3. TAMBAH DEBITUR
4. KELUAR
Pilih menu: 3
Masukkan No KTP: 001
Masukkan nama: aji
Masukkan limit: 230000000
Debitur berhasil ditambahkan
KELOLA DEBITUR
1. TAMPIL DEBITUR
2. CARI DEBITUR
3. TAMBAH DEBITUR
4. KELUAR
Pilih menu: 1
Nama Debitur: Rahmad dengan No KTP: 123 dengan limit pinjaman: 10000000
Nama Debitur: Yoyo dengan No KTP: 321 dengan limit pinjaman: 10000000
Nama Debitur: Yiyi dengan No KTP: 100 dengan limit pinjaman: 10000000
Nama Debitur: aji dengan No KTP: 1 dengan limit pinjaman: 230000000

```

### 4. Kelola Pinjaman Tambah Pinjaman

```

KELOLA PINJAMAN
1. TAMBAH PINJAMAN
2. TAMPIL PINJAMAN
3. KELUAR
Pilih menu: 1
Masukkan nama debitur: wowo
Debitur dengan nama tersebut tidak ditemukan
KELOLA PINJAMAN
1. TAMBAH PINJAMAN
2. TAMPIL PINJAMAN
3. KELUAR
Pilih menu: 1
Masukkan nama debitur: Rahmad
Masukkan nominal yang ingin dipinjam: 1987000000000
Pinjaman melebihi limit
KELOLA PINJAMAN
1. TAMBAH PINJAMAN
2. TAMPIL PINJAMAN
3. KELUAR
Pilih menu: 1
Masukkan nama debitur: Rahmad
Masukkan nominal yang ingin dipinjam: 10000
Masukkan bunga (%): 10
Masukkan jumlah bulan: 5
Pinjaman berhasil ditambahkan

```

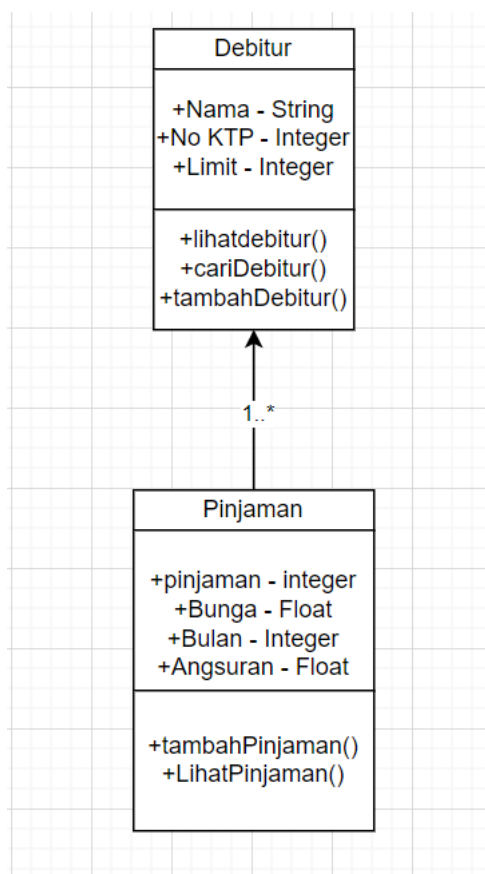
## 5. Tampil Angsuran

```
KELOLA PINJAMAN
1. TAMBAH PINJAMAN
2. TAMPIL PINJAMAN
3. KELUAR
Pilih menu: 2
Nama Debitur: Rahmad
Pinjaman: 10000, Bunga: 10%
Bulan: 5, Angsuran Bulanan: 200.00
Total Angsuran: 1200.00
```

## 6. Keluar Program

```
Aplikasi Pinjaman Online
1. KELOLA DEBITUR
2. KELOLA PINJAMAN
3. KELUAR
Pilih menu: 3
TERIMA KASIH TELAH MENGGUNAKAN PROGRAM, TAPI JANGAN SAMPAI PINJOL BENERAN YAA!!!
PS C:\Users\User\Desktop\PBO Praktikum>
```

## UML NYA:



Pada uml itu kelas debitur ada atribut nama dengan tipe data string, No KTP dan Limit dengan tipe data integer. Pada kelas debitur kita dapat melakukan beberapa aktivitas/proses yaitu lihat debitur, cari debitur, dan tambah debitur. Ada tanda panah dari kelas pinjaman ke debitur serta ada `1..*` itu merupakan garis solid dengan berlian kosong di ujung kelas Debitur menunjukkan bahwa setiap pinjaman "dimiliki oleh" satu debitur. Arah panah dari kelas



Pinjaman ke kelas Debitur menunjukkan arah ketergantungan. Multiplicity "1..\*" pada sisi kelas Debitur menunjukkan bahwa seorang debitur dapat memiliki satu atau lebih pinjaman. Kemudian pada kelas pinjaman ada beberapa atribut, karena ada atribut yang sama dengan kelas debitur dan karena saling ketergantungan maka tidak usah semua ditulis hanya pinjaman dengan tipe data integer, bunga dengan tipe data float, bulan dengan tipe data integer, dan angsuran dengan tipe data float saja yang di tulis. Pada kelas pinjaman juga dapat melakukan beberapa aktivitas/proses yaitu tambah pinjaman dan lihat pinjaman.