



JURUSAN TEKNOLOGI INFORMASI

Pemrograman Berorientasi Objek  
**05. Pewarisan (Bagian-2)**

Yoppy Yunhasnawa, S.ST., M.Sc.

# Topik

1. Penerapan Pewarisan
2. Studi Kasus
3. Class Abstrak



# *Topik #1: Penerapan Pewarisan*

---

---

# 1. Penerapan Pewarisan

- Konsep pewarisan sangat-sangat membantu dalam penulisan program dimana kita **tidak perlu** mengetik ulang kode sumber yang sama berulang-ulang.
- Contoh kasus dimana pewarisan sangat bermanfaat:
  - Pada saat membuat *data class*.
    - *Data Class* → Class yang didalamnya hanya terdapat property dan getter setter-nya saja.
  - Pada saat ingin menambahkan fungsionalitas/fitur baru pada sistem yang **sudah jadi** sebelumnya.
- Dengan pewarisan kita tidak perlu khawatir kode yang sudah ada menjadi error.
  - Karena memang kode yang sudah ada tidak perlu kita apa-apakan selain mengubah *access modifier*-nya.
  - Ingat! Agar bisa diakses dari *child class*, maka *access modifier*-nya harus: **protected**.
- Bagi programmer yang baru bergabung juga akan lebih mudah untuk memahami alur/konsep yang ada pada aplikasi.

# 1. Penerapan Pewarisan

## Tips

- Untuk bisa menerapkan pewarisan pada suatu aplikasi diperlukan sedikit analisis tentang **generalisasi** dan **spesialisasi**.
- **Generalisasi** → Perhatikan property dan method yang bisa **diangkat ke** class induk (*superclass*).
  - Jadikan protected atau public sesuai kebutuhan.
  - Property jangan dijadikan public. (Ingat! Gunakan Getter)
- **Spesialisasi** → Perhatikan property dan method yang memang **tetap harus** berada di child class.
  - Tidak bisa terdapat di class lain.
  - Harus tetap private.
- Berikan nama yang representatif → Melambangkan hubungan inheritance antara class induk dan anak.
  - Contoh: Mahasiswa ← MahasiswaD3 & MahasiswaD4.

## *Topik #2: Studi Kasus Pewarisan*

---

---

# 1. Studi Kasus Pewarisan

Angsuran Ke-	Jumlah Bayar	Sisa Pinjaman
1	55000.0	495000.0
2	55000.0	440000.0
3	55000.0	385000.0
4	55000.0	330000.0

- Masih ingat aplikasi **Kreditor** yang kita buat?
  - Pada aplikasi tersebut terdapat class Pinjaman yang menjadi class utama untuk menghitung data-data pinjaman yang ditampilkan di layar.
- Tahukah Anda ada berbagai macam skema pinjaman?
  - Pinjaman Suku Bunga Tetap
  - Pinjaman Suku Bunga Mengambang
  - Pinjaman Suku Bunga Flat
  - **Pinjaman Suku Bunga Efektif** ← Pinjaman yang dinilai paling adil bagi Peminjam.
  - Pinjaman Suku Bunga Anuitas
- Bagaimana jika kita ingin menambahkan fitur suku bunga efektif pada program kita?

# 1. Studi Kasus Pewarisan Pinjaman Suku Bunga Efektif

- Pinjaman yang bunganya dihitung berdasarkan sisa pinjaman pada periode angsuran sebelumnya.
  - Semakin sedikit sisa pinjaman maka semakin sedikit bunga pada pokok angsuran.

**Bunga =  $SP \times i \times (30/360)$**

SP = saldo pokok pinjaman bulan sebelumnya

i = suku bunga per tahun

30 = jumlah hari dalam 1 bulan

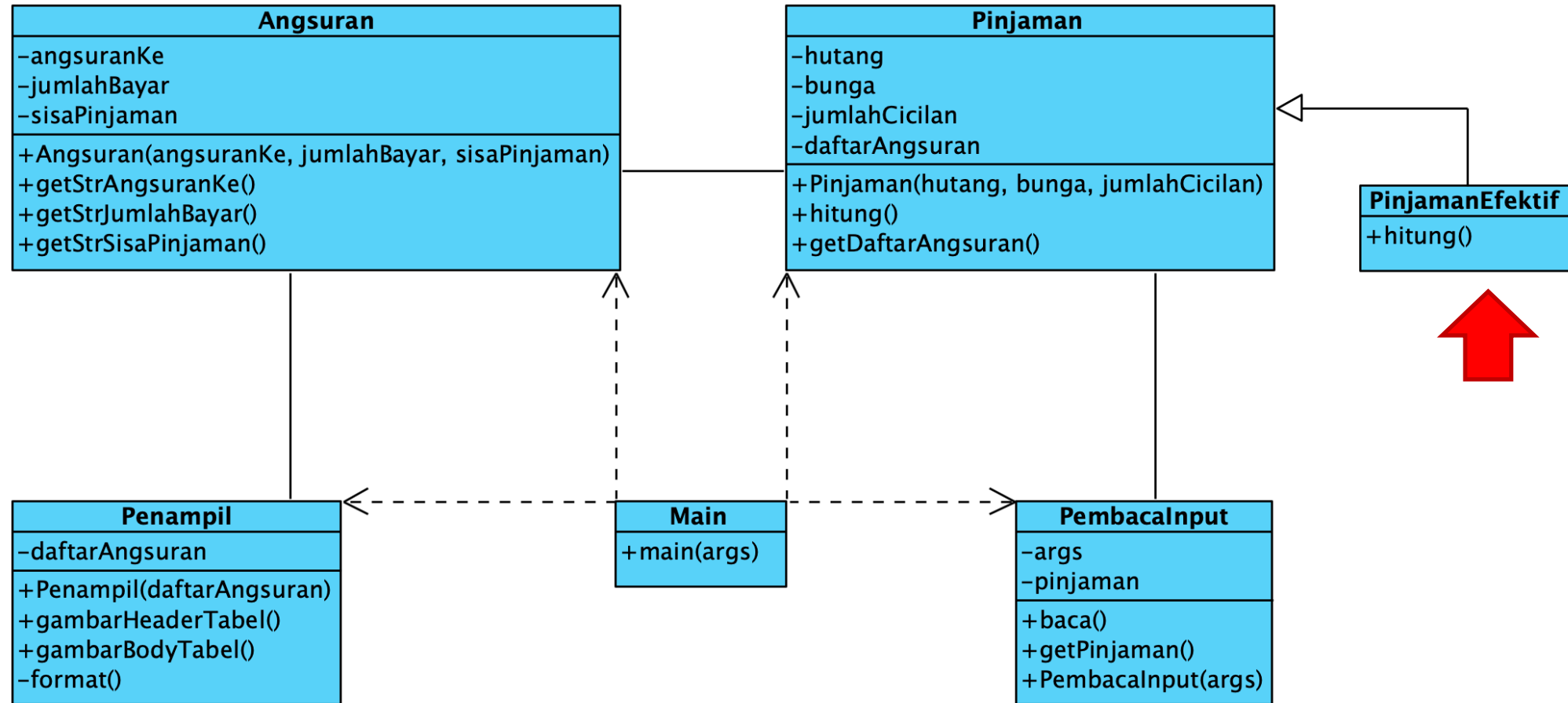
360 = jumlah hari dalam 1 tahun

Plafon	Rp15.000.000	Perhitungan Kredit Bunga Efektif
Angsuran	10 kali	
Bunga	10% per tahun	

Angsuran ke	Angsuran Pokok	Angsuran Bunga	Total Angsuran	Sisa Pokok Pinjaman
0				15.000.000
1	1.500.000	125.000	1.625.000	13.500.000
2	1.500.000	112.500	1.612.500	12.000.000
3	1.500.000	100.000	1.600.000	10.500.000
4	1.500.000	87.500	1.587.500	9.000.000
5	1.500.000	75.000	1.575.000	7.500.000
6	1.500.000	62.500	1.562.500	6.000.000
7	1.500.000	50.000	1.550.000	4.500.000
8	1.500.000	37.500	1.537.500	3.000.000
9	1.500.000	25.000	1.525.000	1.500.000
10	1.500.000	12.500	1.512.500	0
Jumlah	15.000.000	687.500	15.687.500	



## 2. Studi Kasus Pewarisan Class Diagram



## 2. Studi Kasus Pewarisan **Cobalah!**



- Cobalah dulu untuk membuat fitur tersebut!
  - Nanti jika tidak ada yang bisa akan saya tunjukkan caranya.

## 2. Studi Kasus Pewarisan Tips.



- Untuk bisa membuat suatu program yang benar untuk suatu bisnis, seorang programmer harus **mau mempelajari** terlebih dahulu **proses bisnis** riilnya.
  - Jangan membuat program terlebih dahulu apabila Anda belum paham perhitungan atau alur proses bisnis riilnya.
- Cobalah untuk membuat perhitungan manualnya terlebih dahulu.
  - Bisa menggunakan Excel.
  - Apabila Anda sudah paham, barulah untuk mulai membuat programnya.
- Penjelasan macam-macam pinjaman:
  - <https://sikapiuangmu.ojk.go.id/FrontEnd/CMS/Article/20649>

## *Topik #3: Classs Abstrak*

---

---

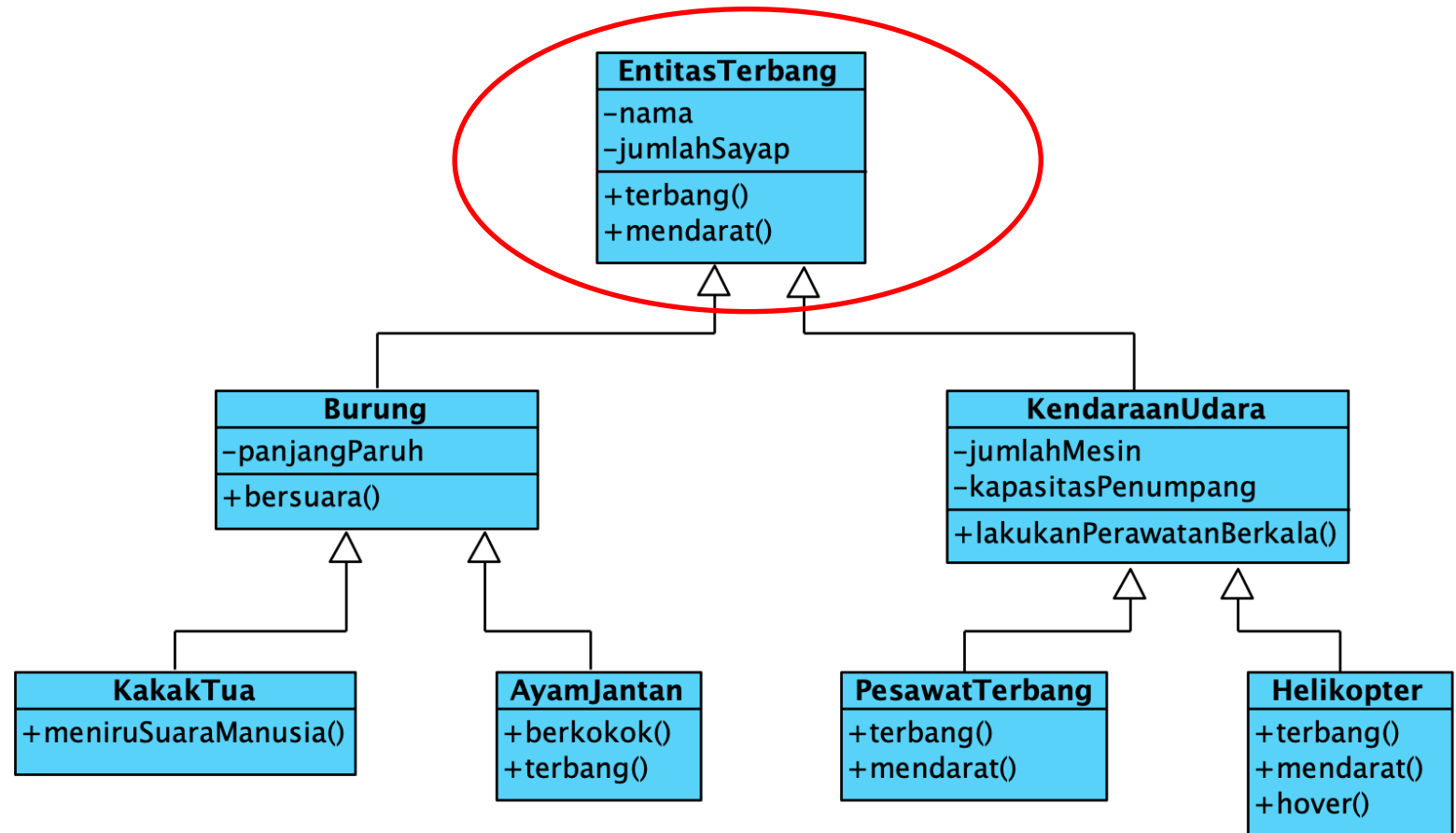
# 1. Class Abstrak

- Bahasa asli → *Abstract Class*
- Adalah class yang tidak dapat di-**instansiasi**.
  - Tidak bisa dijadikan objek dengan keyword **new**.
- Tujuan → Untuk menggambarkan sesuatu yang secara konseptual adalah abstrak.
  - Abstrak → Tidak bisa ada tapi bisa dibayangkan.
- Karakteristik:
  - Dibuat dengan keyword **abstract**.
  - Boleh punya property.
  - Boleh punya method.
  - Ada method yang hanya terdiri dari deklarasinya saja.
- Perhatikan contoh sebelumnya..

### 3. Class Abstrak

## Ilustrasi

- Apa itu **Entitas Terbang**?
- Pernahkah Anda melihat Entitas Terbang?
- Entitas Terbang itu bisa dibayangkan tetapi tidak ada wujudnya.
- Bagaimana dengan Burung?
- Bagaimana dengan Kakak Tua, Ayam Jantan, Pesawat Terbang, dan Helikopter?



## 4. Class Abstrak

### Method Abstrak

- Pada suatu class abstrak harus terdapat setidaknya satu **method abstrak**.
- Method abstrak:
  - Hanya terdiri dari deklarasi method dan tidak terdapat body.
  - Harus di-override oleh **semua** class yang merupakan turunan dari suatu class abstraknya.

# Pertanyaan?





*Terima Kasih*

# Tugas Latihan



- Ubahlah implementasi EntitasTerbang dari Latihan sebelumnya menjadi class abstrak!

# References



1. <https://sikapiuangmu.ojk.go.id/FrontEnd/CMS/Article/20649>