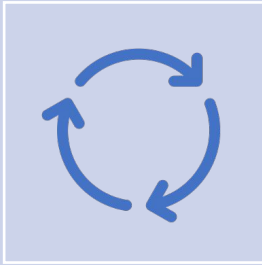


Object Oriented Analysis and Design (OOA-OOD)

Pertemuan 6

Kuliah Fundamen Pengembangan
Aplikasi

Software Model Process VS Software Development Paradigm

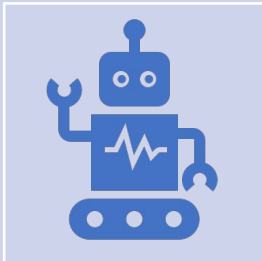


Software Model Process:

Urutan/tahapan pengembangan perangkat lunak

Contoh: *Waterfall, Prototyping*

Dikenal juga dengan ***SDLC (Software Development Life Cycle)***



Software Development Paradigm:

Cara pandang spesifik dalam upaya pencarian solusi pengembangan perangkat lunak

Contoh: *Object Oriented, Data Driven, Procedural*

Apa itu Object Oriented?

- Paradigma **Object Oriented** merupakan salah satu paradigma yang memfokuskan pada objek-objek di kehidupan nyata ketika mengembangkan sebuah sistem/program.
 - Hal ini seolah-olah objek-objek tersebut adalah yang menyusun sistem dengan segala interaksinya dengan objek lainnya.
- Tujuan dari paradigma ini yaitu untuk merepresentasikan dunia nyata (melalui objek-objeknya) ke dalam kode-kode program.
- Objek-objek ini memiliki **data** dan **perilaku**.
 - Misalnya objek **ORANG** memiliki **data** berupa nama, usia, tanggal lahir, dan sebagainya. Data-data ini akan diperlukan untuk diproses di dalam sistem
 - Objek **ORANG** memiliki **perilaku** seperti membaca, menulis, berlari, dan sebagainya. Perilaku-perilaku ini nantinya akan dijadikan sebagai fungsi yang dapat dikerjakan oleh sistem yang akan dibuat.

Kenapa Object Oriented?

- Kompleksitas perangkat lunak (kompleksitas masalah dan kompleksitas solusi)
 - Prinsip abstraksi dan dekomposisi pada OO bisa membantu menyelesaikan masalah kompleksitas pada pengembangan perangkat lunak.
- Menyelesaikan gap dunia nyata dan program komputer
 - Perangkat lunak merupakan gambaran interaksi objek-objek di dunia nyata dalam menyelesaikan masalah tertentu.
- Adaptable dan scalable (mempermudah pemeliharaan)
 - Perubahan kebutuhan itu hampir pasti terjadi, software harus menyesuaikan.

Comparing **Object-Oriented** vs **Procedural** Approaches

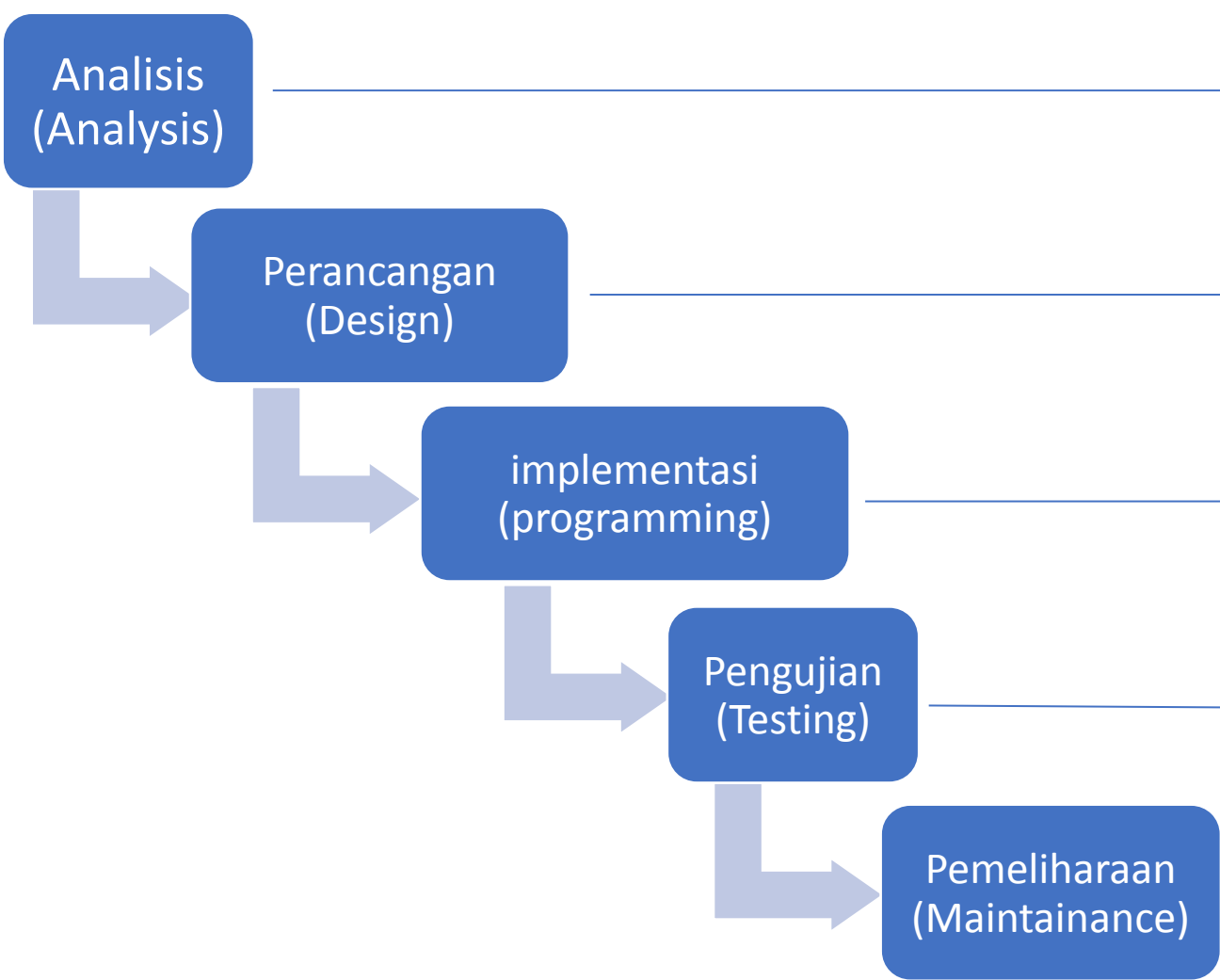
In the **procedural paradigm**:

- The problem is divided into smaller parts called procedures (or functions), and systems are built by combining procedures.
- These procedures share data by passing data between them or through global variables.
-

In the **object-oriented** paradigm:

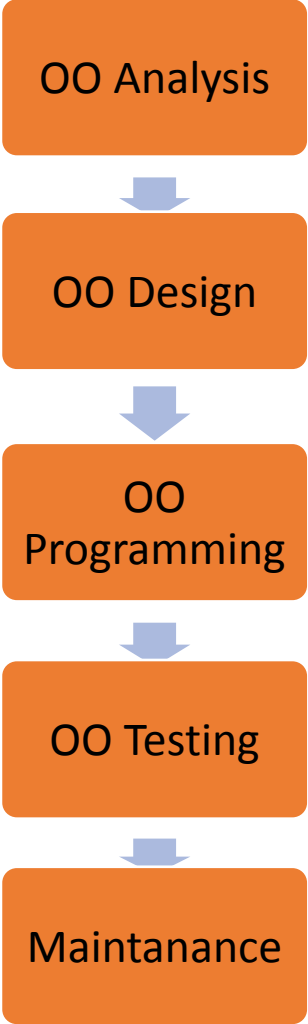
- The problem is divided into smaller parts called objects, and systems are built around objects.
- Objects are representations of things that exist in the real world that we wish to model in a computer system.
- Objects do not share data.

Software Model Process

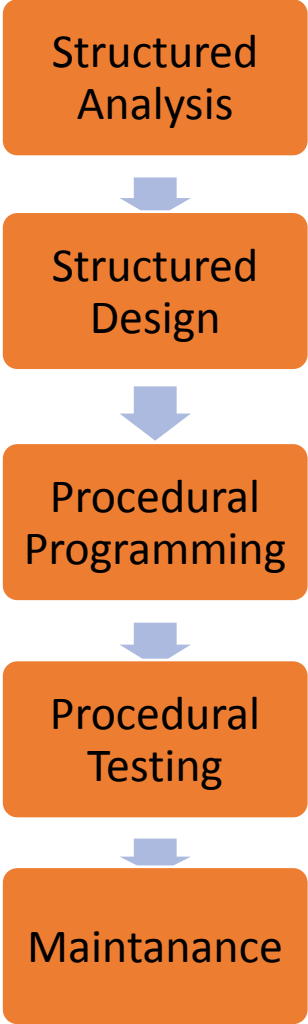


Software Development Paradigm

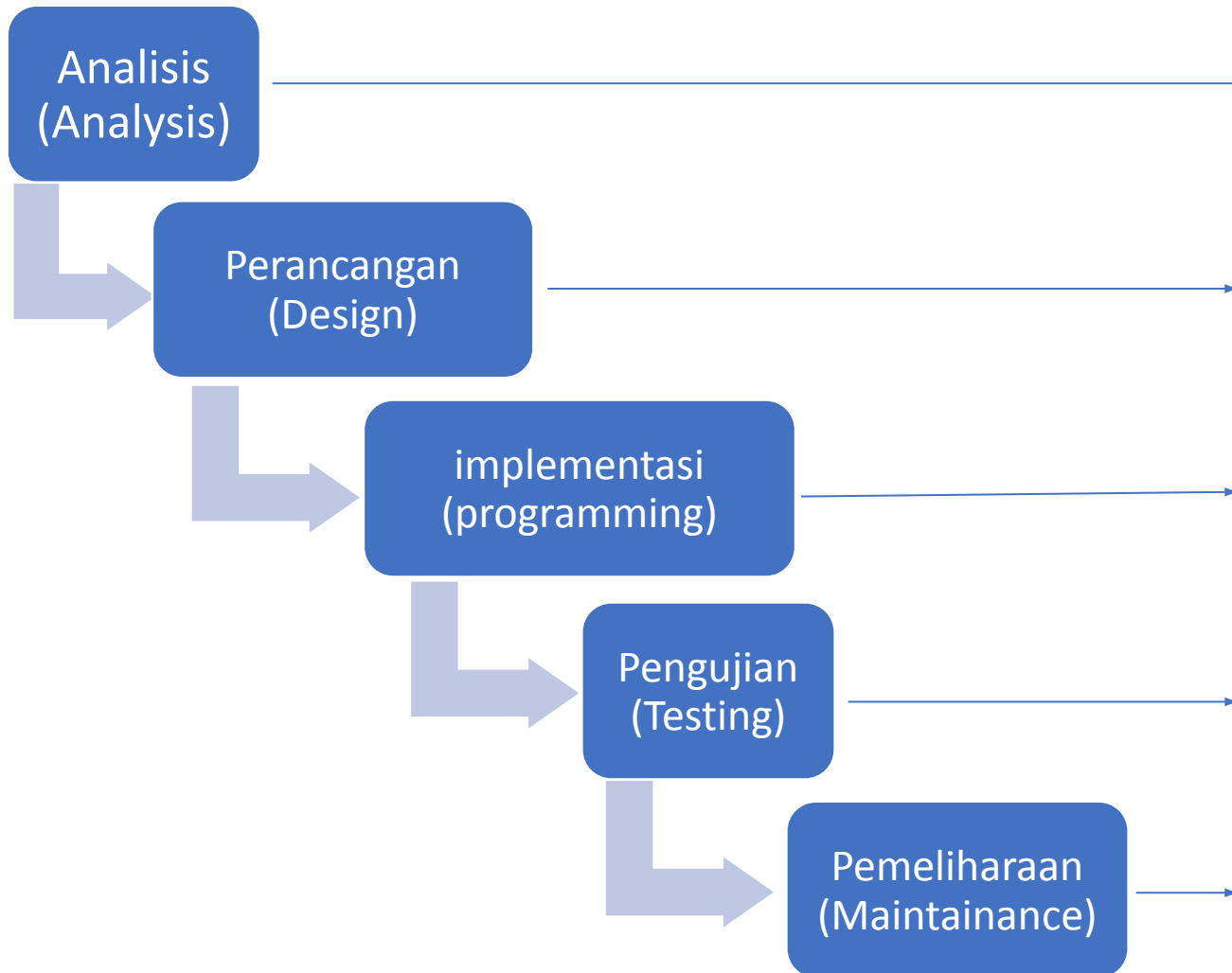
Object Oriented



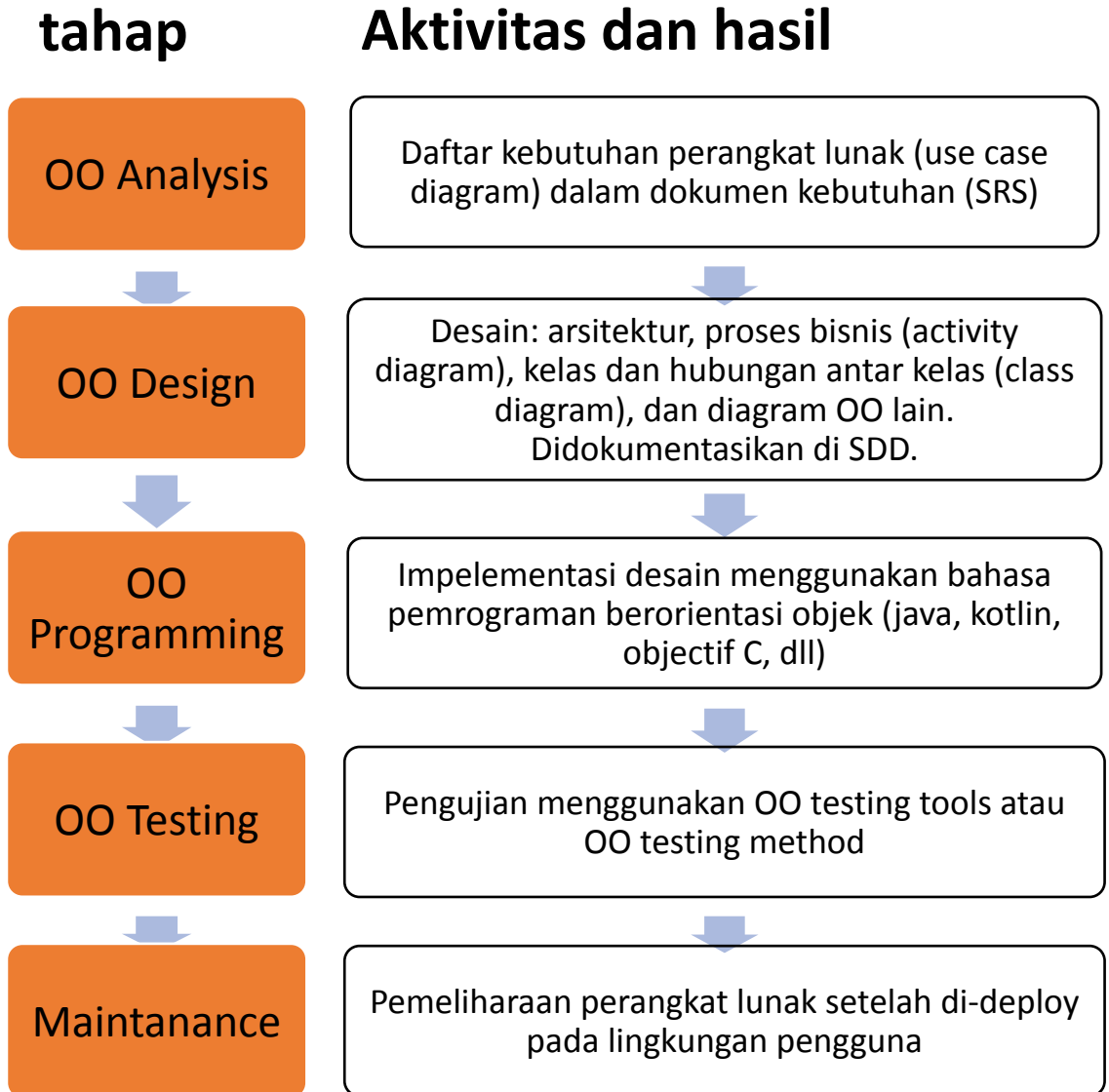
Procedural



Software Model Process



Object Oriented Paradigm



Paradigma OO dalam SDLC

OO Analysis

OO Design

OO Programming

Implementasi diagram kelas

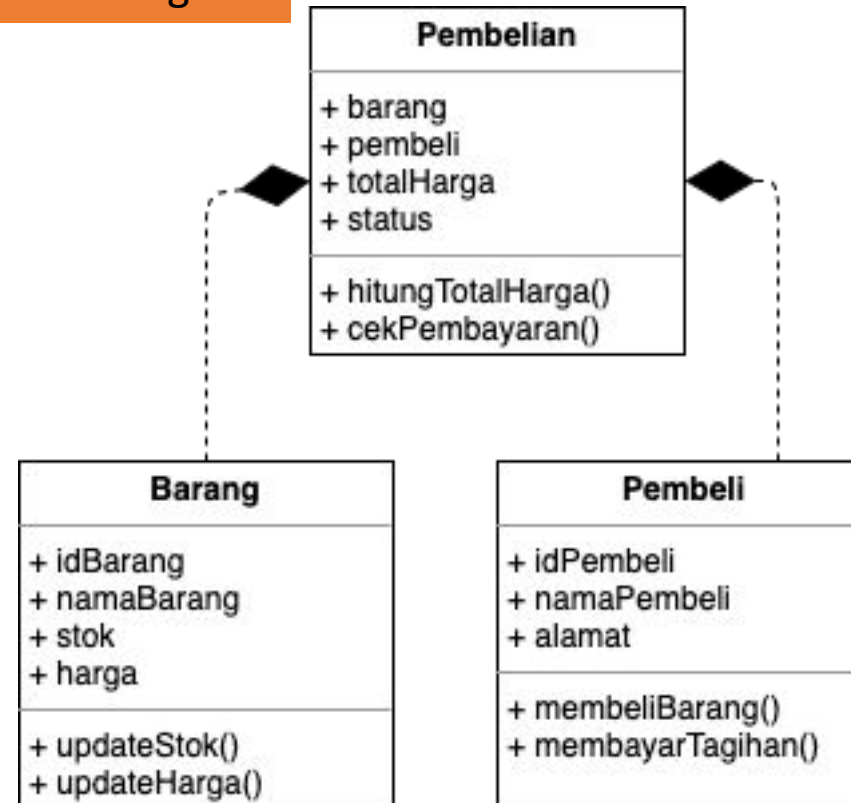
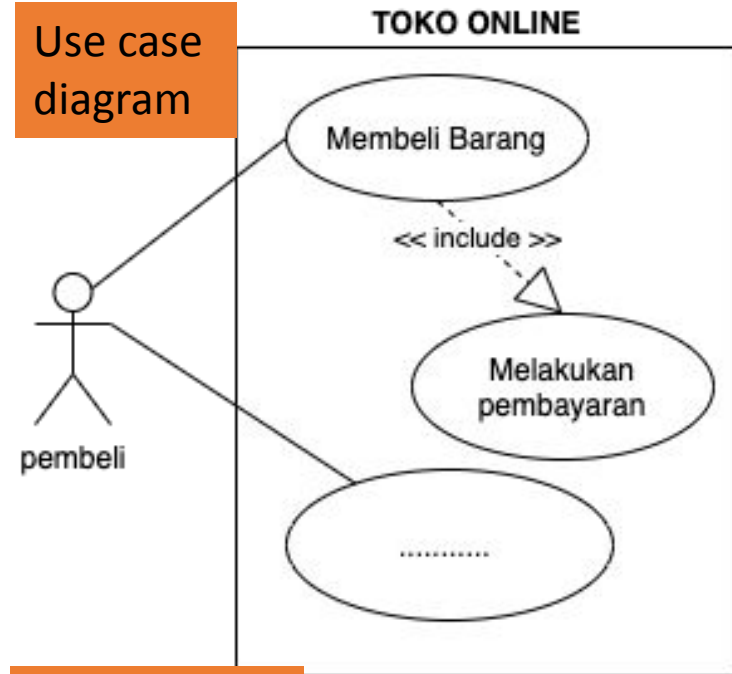
Use case diagram

class diagram

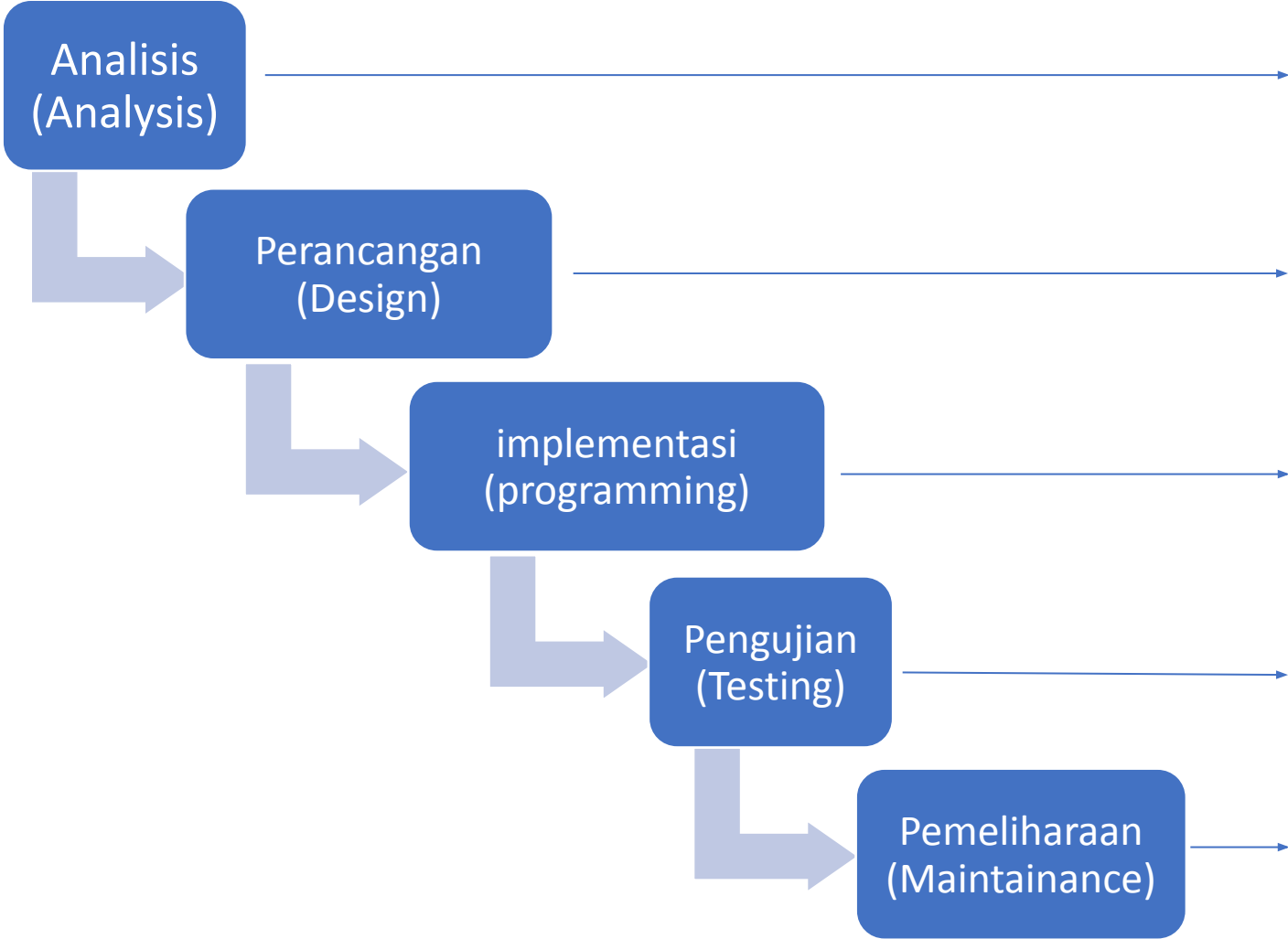
```
public class Pembeli {  
    String idPembeli;  
    String namaPembeli;  
    String alamat;  
  
    void membeliBarang () {  
        // .....  
    }  
  
    void membayarTagihan() {  
        // .....  
    }  
}
```

Daftar objek konseptual

1. Pembeli
2. Barang
3. Pembayaran
4.




Software Model Process



Procedural Paradigm

tahap	Aktivitas dan hasil
Procedural Analysis	Daftar kebutuhan perangkat lunak (use case diagram) dalam dokumen kebutuhan (SRS)
Procedural Design	Dekomposisi fungsi menjadi fungsi yang lebih kecil. Atau bisa dekomposisi poses menjadi proses yang lebih kecil
Procedural Programming	Implementasi dekomposisi proses membagi program besar ke dalam program yang lebih kecil (sub program)
Procedural Testing	Semua pengujian yang tidak spesifik untuk paradigma tertentu bisa dilakukan
Maintanance	Pemeliharaan perangkat lunak setelah di-deploy pada lingkungan pengguna



OOAD

(Object Oriented Analysis and Design)

OOAD adalah metode pengembangan sistem yang lebih menekankan pada **objek** dibandingkan dengan data atau proses.

Dalam Tahapan nya OOAD terbagi menjadi dua yaitu:

1. **OOA** (Object oriented analysis)
 2. **OOD** (Object Oriented Design).
- 
- 

Object Oriented Analysis (OOA)

- Object oriented analysis (OOA) merupakan metode analisis yang memeriksa *requirements* (yang harus dipenuhi oleh sistem) dari sudut pandang kelas – kelas dan objek – objek yang ditemui dalam ruang lingkup permasalahan.
- OOA mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi objek.

Object Oriented Analysis (OOA)

Tugas pada tahap analisis OO:

- Menjelaskan proses yang terjadi dalam sistem
- Identifikasi objek konseptual (objek dunia nyata)
- Menentukan atribut sebuah objek
- Menentukan kemampuan atau perilaku sebuah objek
- Menentukan hubungan antar objek




Objek



Definisi objek:

- a) *An object is an individual [thing] with a state and relationships to other objects (UML 2.5 specification)*
- b) *An entity with a well defined boundary and identity that encapsulates state and behavior (UML 14.2 specification)*

Objek adalah benda atau entitas yang mempunyai state, perilaku, dan hubungan dengan objek yang lain.



Objek

- Benda

- Tangible: memiliki bentuk dan fisik sehingga dapat diraba, dilihat, dirasakan, dll. Contoh: mobil, manusia, mahasiswa, hewan, gelas, dll.
- Intangible: tidak memiliki bentuk fisik sehingga tidak dapat diindera. Contoh: ide, perasaan, harapan, dll.



Apa saja yang tangible?

- Buku
- Meja
- Pegawai perpustakaan
- Mahasiswa
- dll

Apa saja yang intangible?

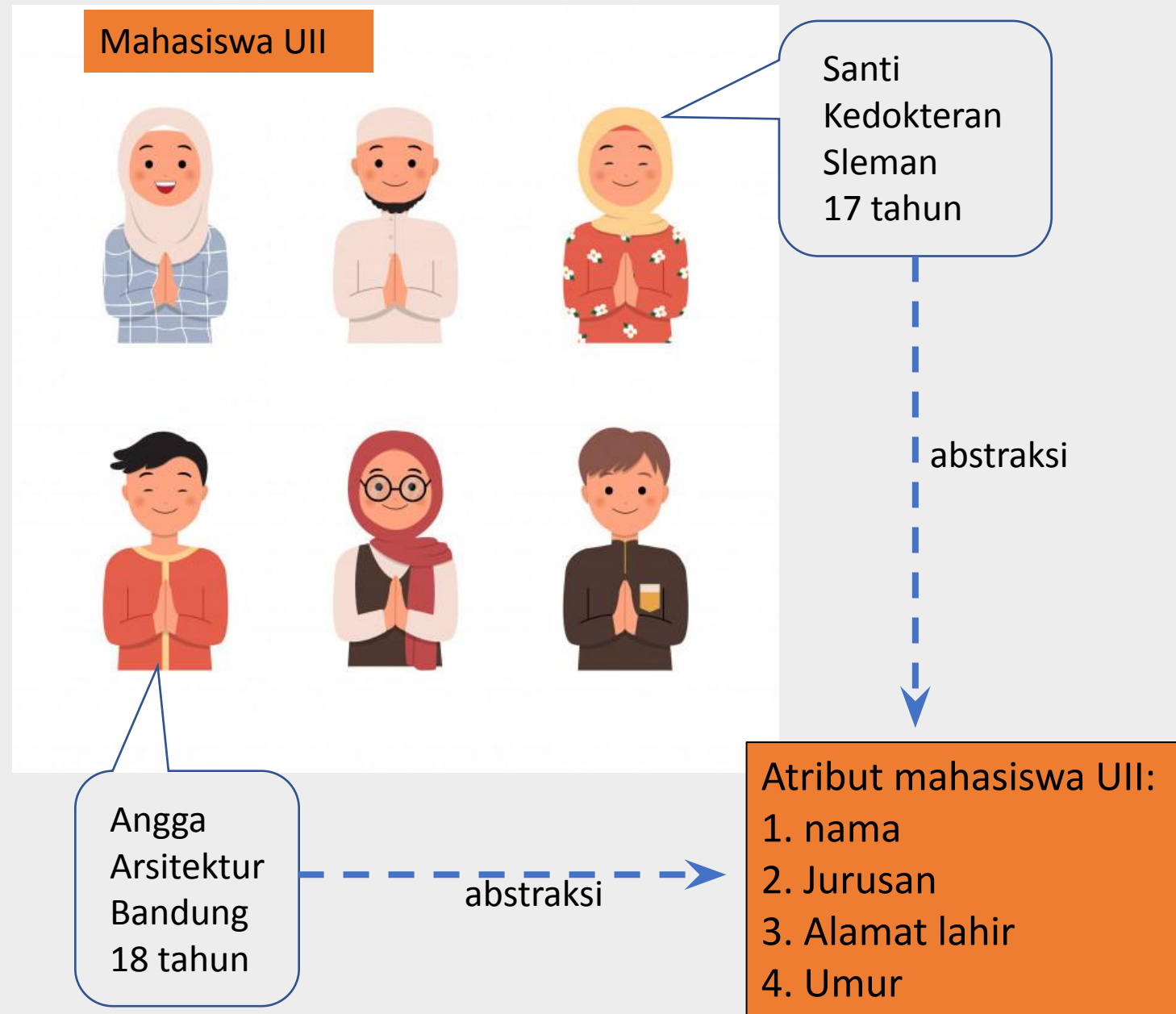
- Peminjaman buku
- Pengembalian buku
- Denda
- dll

Identifikasi objek

- Dekomposisi (objek)
 - Memecah kompleksitas sistem menjadi objek-objek yang terlibat pada domain masalah di lingkungan pengembangan perangkat lunak.
 - Objek bisa berupa:
 - Entitas eksternal yang terdefinisi dengan baik (eksplisit) yang berhubungan dengan sistem. **Misal: pengguna, pelanggan, dll.**
 - Kejadian yang muncul dari operasional sistem. **Misal: pengembalian buku, pembayaran tagihan, dll.**
 - Peran atau tanggung jawab yang dimainkan oleh entitas yang berhubungan dengan sistem. **Misal: mahasiswa, dosen, dll.**
 - Antarmuka atau layanan. **Misal: formulir pengembalian buku, autentifikasi pengguna, dll**
- Identifikasi objek dilakukan setelah mendapatkan kebutuhan dari pengguna (requirement gathering)
- Bisa didapatkan dari use case dan/atau scenario use case
- Hasil identifikasi objek akan masuk ke daftar kandidat objek.

Atribut objek

- Data/informasi yang menggambarkan/mencirikan objek secara utuh dan membedakan dengan objek yang lain.



****Abstraksi:** untuk menentukan atribut bisa dilakukan dengan menarik informasi yang umum dari informasi yang ada pada setiap objek.*

Perilaku objek

Menggambarkan kemampuan objek pada situasi dan ruang lingkup tertentu

- Mengeluarkan transkrip nilai
- Membayar gaji karyawan
- Memberikan fasilitas pendidikan
- Menerima mahasiswa baru
- dll



Rektorat UII

- Mengajar mata kuliah
- Membimbing skripsi
- Melakukan penelitian
- Memberikan nilai
- dll



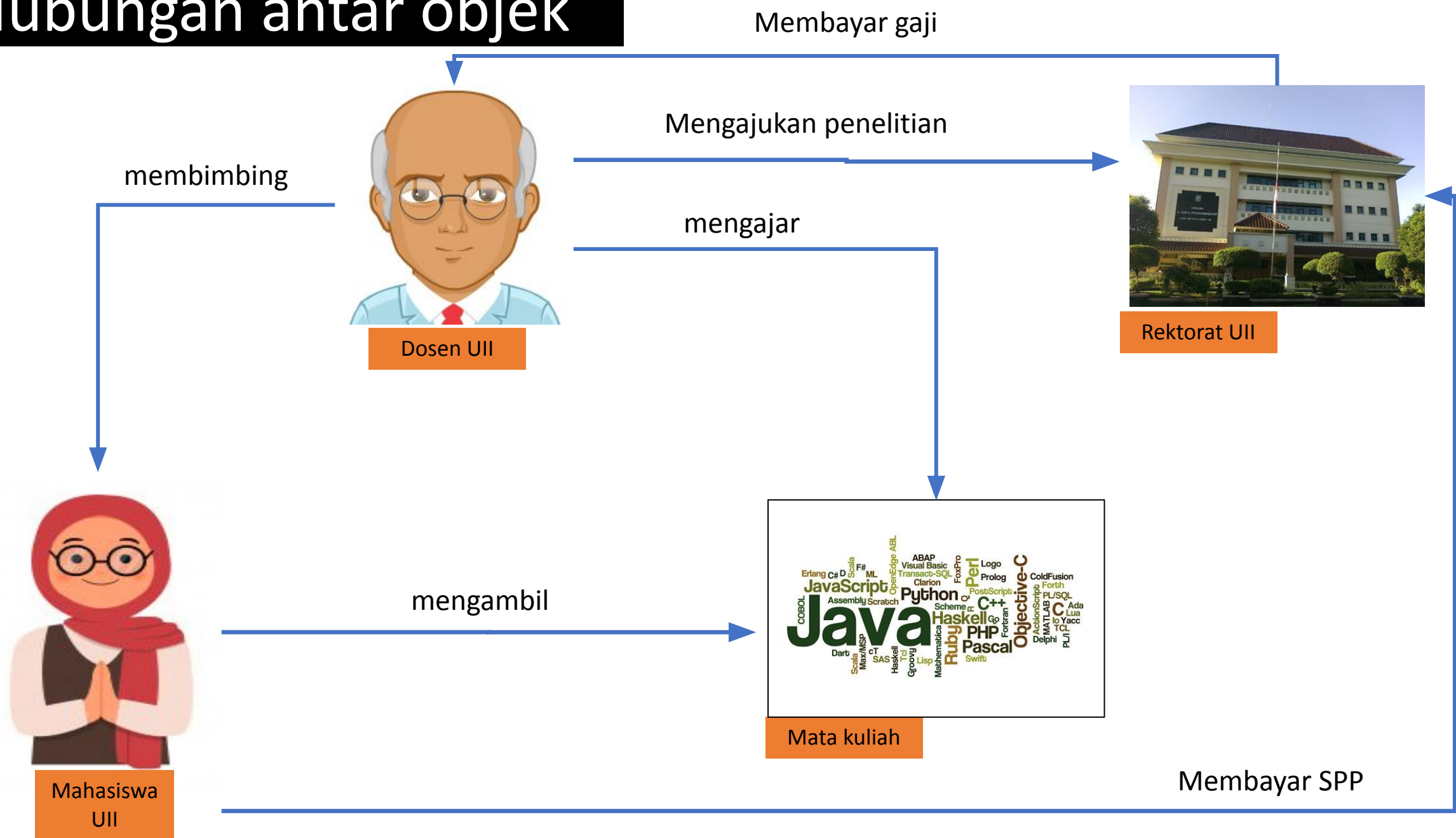
Dosen UII



Mahasiswa
UII

- Keyin semester
- Bimbingan skripsi
- Mengumpulkan tugas
- Membayar SPP
- dll

Hubungan antar objek



Menuliskan hasil analisis OO pada dokumen SRS

A.4 Template of SRS Section 3 organized by object

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communications interfaces

3.2 Classes/Objects

3.2.1 Class/Object 1

3.2.1.1 Attributes (direct or inherited)

3.2.1.1.1 Attribute 1

-

-

-

3.2.1.1.*n* Attribute *n*

3.2.1.2 Functions (services, methods, direct or inherited)

3.2.1.2.1 Functional requirement 1.1

-

-

-

3.2.1.2.*m* Functional requirement 1.*m*

3.2.1.3 Messages (communications received or sent)

Latihan1

Tentukan atribut dan perilaku dari objek-objek berikut:

1. Mobil
2. Counter teroris
3. Remote TV
4. virtual keyboard smartphone

Latihan 2

- a) Identifikasi kandidat objek yang terlibat pada saat mahasiswa melakukan keyin di setiap semester!
- b) Setelah mendapatkan kandidat objek, tentukan perilaku masing-masing objek dan gambarkan hubungan antar objeknya!



OO Design

Object Oriented Design (OOD)

- Proses mengubah requirement/persyaratan yang didapat pada proses OOA menjadi spesifikasi implementasi.
 - Perancang harus menamai objek, mendefinisikan perilaku, dan secara formal menentukan objek mana yang dapat mengaktifkan perilaku tertentu pada objek lain.
- OOD memungkinkan *software engineer* untuk mengetahui objek-objek yang dihasilkan oleh tiap **class** dan hubungan antar objek.
- Selain itu, OOD menggambarkan bagaimana hubungan antar objek bisa dilakukan, bagaimana behavior dari objek diimplementasikan dan bagaimana komunikasi antar objek diimplementasikan.

Tugas pada tahap desain OO

1. Mengidentifikasi kelas
2. Menentukan atribut dan method serta menentukan rancangan implementasinya
3. Menentukan hubungan antar kelas dan merancang implementasi hubungan tersebut
4. Menentukan interaksi antar objek (instan kelas)

Identifikasi kelas (class)

Apa itu kelas?

- Deskripsi dari sekumpulan objek dengan ciri dan perilaku yang sama.
- Cetak biru dari objek
- Representasi objek konseptual (tahap analisis) dan deskripsi detailnya dalam fase perancangan perangkat lunak.

*Jika pada **tahap analisis** sudah mendapatkan kandidat objek beserta atribut, perilaku dan hubungannya, bisa **dilanjutkan** dengan memilih objek yang akan menjadi kelas dan merinci lebih detail mengenai atribut, perilaku, dan hubungan tersebut sehingga bisa lebih mudah diimplementasikan pada tahap konstruksi (pemrograman).

Objek konseptual pada tahap analisis



Mahasiswa UII

Atribut mahasiswa UII:

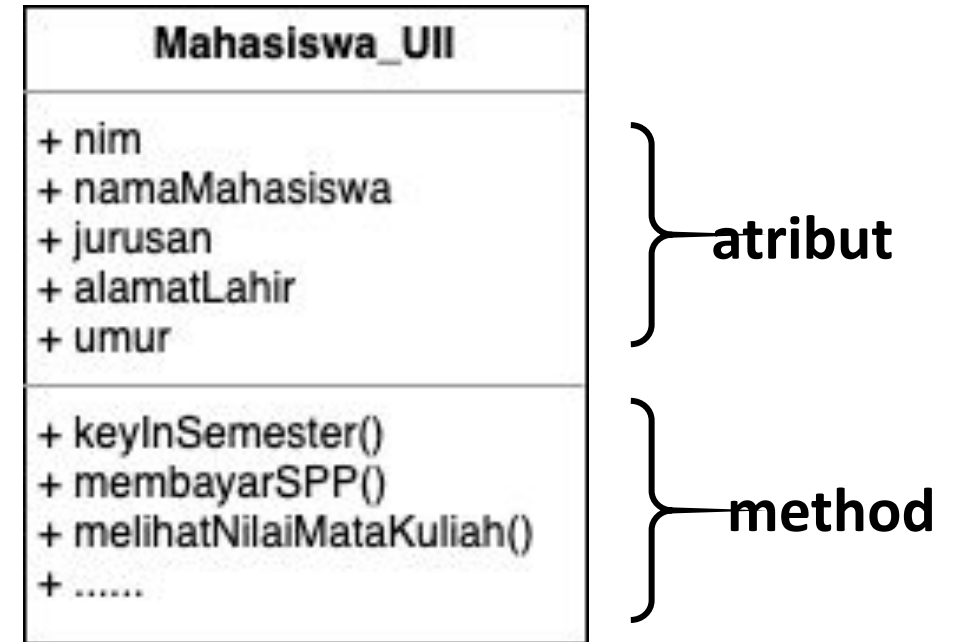
1. nama
2. Jurusan
3. Alamat lahir
4. Umur

Perilaku mahasiswa UII:

- Keyin semester
- Bimbingan skripsi
- Mengumpulkan tugas
- Membayar SPP
- dll



Kelas pada tahap rancangan



* Pada tahap perencanaan kita bisa melakukan *refinement* pada hasil analisis seperti **menambahkan** atribut dan method yang dibutuhkan dan **menghilangkan** atribut dan method yang kurang relevan dengan implementasi

Rancangan implementasi atribut dan method pada kelas

Mahasiswa_UII
+ nim + namaMahasiswa + jurusan + alamatLahir + umur
+ keyInSemester() + membayarSPP() + melihatNilaiMataKuliah() +



MahasiswaUII
+ nim: String + namaMahasiswa: String + jurusan: String + alamatLahir: String + umur: String
+ keyInSemester (daftarMataKuliah, semester) + membayarSPP (totalSPP, jumlahAngsuran, kaliAngsuran) + lihatNilaiMataKuliah (semester) + dll

*Jika pada tahap analisis kita menggunakan abstraksi yang tinggi untuk memudahkan pihak yang berkepentingan membaca hasil analisis, pada tahap rancangan kita sudah harus memikirkan **implementasi** rancangan pada tahap **konstruksi**. Karena itu perlu membuat rancangan **yang lebih teknis** dan dekat dengan konstruksi aplikasi menggunakan teknologi (bahasa pemrograman yang akan dipakai)

Menentukan implementasi hubungan antar kelas

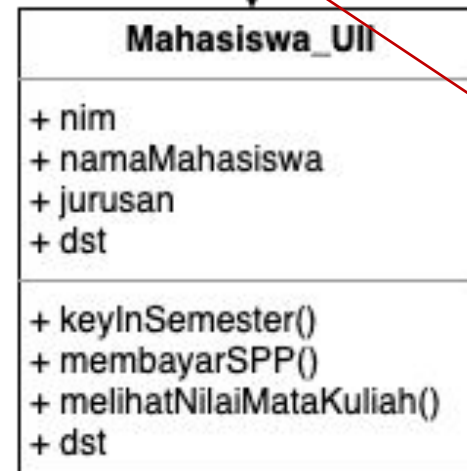
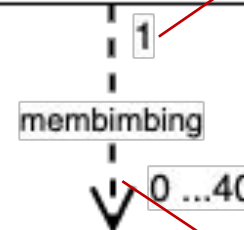
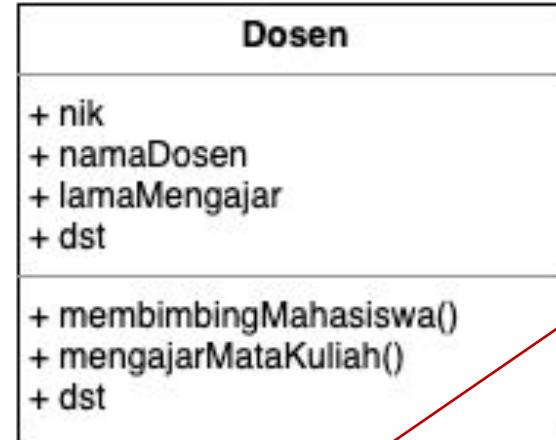


Dosen UII

Membimbing
akademik



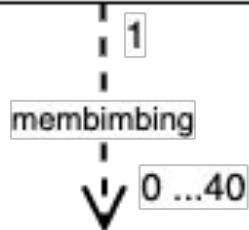
Mahasiswa
UII



Menggambarkan Kardinalitas hubungan antar kelas (1 dosen bisa membimbing 0 sampai dengan 40 mahasiswa)

Tanda “- - - >” menggambarkan bentuk hubungan antar kelasnya *dependency* yang artinya objek dari kelas mahasiswa akan digunakan sebagai parameter dari method objek kelas dosen UII

Contoh implementasi dalam program



```
package oodesign;

public class DosenUII {
    String nik, namaDosen; //dst

    void membimbingMahasiswa(MahasiswaUII []
                               mahasiswaYangDiBimbing) {

    }
}
```

```
package oodesign;

public class MahasiswaUII {

    String nim, namaMahasiswa; //dan seterusnya

    void membayarSPP() {
        //isi method
    }

    // dan seterusnya
}
```

Kesimpulan

tahap OO Analysis

aktivitas

Menentukan objek konseptual
(Daftar kandidat objek)
beserta atribut, **perilaku**,
dan hubungan **antar objek**

Contoh hasil



Mahasiswa UII

Atribut mahasiswa UII:

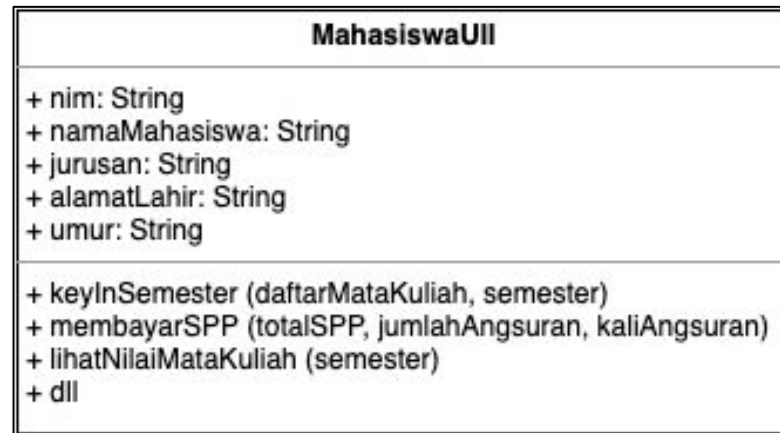
1. nama
2. Jurusan
3. Alamat lahir
4. Umur

Perilaku mahasiswa UII:

- Keyin semester
- Bimbingan skripsi
- Mengumpulkan tugas
- Membayar SPP
- dll

OO Design

Menentukan kelas dari kandidat objek
beserta **implementasi** atribut, **method**, dan hubungan **antar kelas**



OO Programming

Implementasi dalam bahasa pemrograman object oriented (misal: java)

```
package oodesign;  
  
public class MahasiswaUII {  
    String nim, namaMahasiswa; //dan seterusnya  
  
    void membayarSPP() {  
        //isi method  
    }  
  
    // dan seterusnya  
}
```

Kesimpulan 2: Perbedaan istilah objek

Objek konseptual

Kelas sebagai cetak biru objek di program

Objek sebagai instan kelas di memori



Mahasiswa UII

MahasiswaUII
+ nim: String + namaMahasiswa: String + jurusan: String + alamatLahir: String + umur: String
+ keyInSemester (daftarMataKuliah, semester) + membayarSPP (totalSPP, jumlahAngsuran, kaliAngsuran) + lihatNilaiMataKuliah (semester) + dll

```
public static void main(String[] args) {  
    MahasiswaUII mahasiswaFPA = new MahasiswaUII();  
    mahasiswaFPA.nim = "2010511";  
    mahasiswaFPA.namaMahasiswa = "budi";  
    mahasiswaFPA.membayarSPP();  
}
```

*dibahas pada pertemuan selanjutnya