ANALISIS DAN SISTEM PERANGKAT LUNAK

KONTRAK PERKULIAHAN

1. Mata Kuliah Analisis dan Sistem Perangkat Lunak berbentuk **Mata Kuliah Berbasis Proyek.** Pada pertemuan 8 dilakukan penilaian proyek tahap awal dan pada pertemuan 16 dilakukan penilain proyek akhir.
   a. Pertemuan 1-6 : Perkuliahan Tatap Muka, Diskusi dan Penugasan
   b. Pertemuan 7 : Pengumpulan Proyek Awal
   b. Pertemuan 8 : Penilaian Proyek Awal
   c. Pertemuan 9-12 : Bimbingan dan Persiapan Presentasi
   d. Pertemuan 13-15 : Presentasi Proyek Akhir
   e. Pertemuan 16 : Entry Nilai Akhir

2. Bobot Penilaian

Nilai Akhir = (20% x Nilai Kehadiran) + (25% x Nilai Tugas) + (25% x Nilai Proyek Awal) + (30% x Nilai Proyek Akhir)

   a. Nilai Kehadiran : Berdasarkan jumlah kehadiran/presensi
   b. Nilai Tugas : Berdasarkan Tugas yang dikerjakan mahasiswa
   c. Nilai Proyek Awal : Berdasarkan Penilaian Proyek Awal
   d. Nilai Proyek Akhir : Berdasarkan Penilaian Proyek Akhir

KETENTUAN PROYEK

Ketentuan Umum:

1. Proyek dikerjakan secara **berkelompok** (3 – 5 Orang).
2. Laporan Proyek dibuat **sesuai dengan sistematika** yang ditentukan.
3. Laporan proyek berupa **Laporan Analisis dan Sistem Perangkat Lunak** dibuat menggunakan diagram UML dengan Metode Waterfall atau Metode yang lainnya.
4. Tentukan **Judul Project** berdasarkan tema Perangkat Lunak yang akan dianalisis dan dirancang. **Tema perangkat lunak** dapat berupa Perangkat Lunak Bisnis, Sains, Teknik, Games, IoT atau perangkat lunak lain. Tema tersebut harus mendapat persetujuan dosen dan konsisten dari pertemuan 1 sampai akhir.
5. Tentukan **Tempat Riset**, dapat berupa: Perusahaan, instansi pemerintah, Lembaga Pendidikan, Komunitas, Organisasi sosial kemasyarakatan, dan lain-lain. Kegiatan Riset dibuktikan dengan Surat Keterangan Riset dan Dokumentasi berupa Foto/Video

Ketentuan Proyek (Lanjutan)

A. Proyek Tahap Awal :

1. Bentuk : Analisis Kebutuhan Perangkat Lunak
2. Waktu :
   - Penugasan pada Pertemuan 2
   - Pengumpulan pada Pertemuan 7

B. Proyek Tahap Akhir :

1. Bentuk : Laporan Analisis dan Sistem Perangkat Lunak
2. Waktu :
   - Penugasan pada Pertemuan 9
   - Bimbingan dan persiapan Presentasi pada Pertemuan 9-12
   - Presentasi Proyek Akhir : Pertemuan 13-15

PERTEMUAN 1

METODOLOGI ANALISIS DAN SISTEM PERANGKAT LUNAK

MATERI PERTEMUAN I

1. Pengembangan Perangkat Lunak
2. *Software/System Development Life Cycle (SDLC)*
3. Model *Waterfall*
4. Metodologi Analisis Dan Sistem Perangkat Lunak Berorientasi Objek

I. PENGEMBANGAN PERANGKAT LUNAK

- **Perangkat lunak yang baik** adalah Perangkat lunak yang selalu sesuai dengan perubahan. Namun pada kenyataannya hal tersebut sulit terwujud dikarenakan dinamika yang terjadi. Oleh karenanya diperlukan pengembangan perangkat lunak.
- **Pengembangan perangkat lunak** dapat diartikan sebagai proses membuat suatu perangkat lunak baru untuk menggantikan perangkat lunak lama secara keseluruhan atau memperbaiki perangkat lunak yang telah ada
- **Metodologi pengembangan perangkat lunak** adalah suatu proses pengorganisasian kumpulan metode dan konvensi notasi yang telah didefinisikan untuk mengembangkan perangkat lunak. Secara prinsip bertujuan untuk membantu menghasilkan perangkat lunak yang berkualitas.

a. Alasan Pengembangan Perangkat Lunak

- ***Problem-Solving***

  Perangkat lunak lama tidak berfungsi sesuai dengan kebutuhan. Untuk itu analisis diperlukan untuk memperbaiki perangkat lunak sehingga dapat berfungsi sesuai dengan kebutuhan.

- **Kebutuhan Baru**

  Adanya kebutuhan baru dalam organisasi atau lingkungan sehingga diperlukan adanya modifikasi atau tambahan sistem informasi untuk mendukung organisasi.

- **Mengimplementasikan ide atau teknologi baru**.

  Adanya ide baru atau teknologi baru yang memungkinkan untuk dimanfaatkan untuk keunggulan kompetitif perusahaan atau merupakan peluang bisnis.

b. Proses Pengembangan Perangkat Lunak

Proses pengembangan perangkat lunak adalah suatu proses dimana kebutuhan pemakai diterjemahkan menjadi produk perangkat lunak.

Proses ini mencakup aktivitas penerjemahan kebutuhan pemakai menjadi kebutuhan perangkat lunak, transformasi Analisis kebutuhan perangkat lunak menjadi desain, penerapan desain menjadi kode program, uji coba kode program, dan instalasi serta pemeriksaan kebenaran perangkat lunak untuk operasional.

b. Proses Pengembangan Perangkat Lunak (Lanjutan)

Pada pembangunan atau pengembangan perangkat lunak, yang perlu diperhatikan:

1. Menentukan **APA** persoalan yang harus dipecahkan (Kebutuhan Perangkat Lunak yang akan dicarikan solusi).
2. Mendefinisikan **BAGAIMANA** perangkat lunak dibuat, mencakup arsitektur perangkat lunaknya, antar muka internal, algoritma, dan sebagainya.
3. **Penerapan** (penulisan program) dan **pengujian** unit-unit program.
4. **Integrasi** dan **Validasi** perangkat lunak melalui pengujian modul-modul program dan pengujian sistem.

c. Tujuan Pengembangan Perangkat Lunak

Pengembangan perangkat lunak bertujuan untuk menghasilkan perangkat lunak yang berkualitas.

Perangkat lunak dapat dikatakan sebagai **perangkat lunak yang berkualitas apabila** :

- Perangkat lunak tersebut memenuhi keinginan pemesan atau pihak yang menggunakannya (*user*).
- Keinginan *user* tersebut meliputi beberapa aspek, antara lain fitur dan antarmuka.
- Perangkat lunak tersebut berfungsi dan dapat diimplementasikan dalam jangka waktu yang relatif lama.
- Mudah dimodifikasi untuk memenuhi kebutuhan yang berkembang.
- Mudah digunakan.
- Dapat mengubah atau membangun sesuatu dengan lebih baik.

C. Tujuan Pengembangan Perangkat Lunak (Lanjutan)

Perangkat lunak dikatakan gagal apabila :

- *User* tidak puas terhadap performansi perangkat lunak.
- Memiliki banyak kesalahan.
- Bila perangkat lunak tersebut sulit untuk dimodifikasi untuk kebutuhan yang berkembang.
- Bila perangkat lunak tersebut sulit untuk dioperasikan.
- Menghasilkan sesuatu yang tidak dikehendaki.

II. Software/Systems Development Life Cycle (SDLC)

SDLC atau ***Software Development Life Cycle*** atau ***System Development Life Cycle*** adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model atau metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (Rosa dan Shalahuddin, 2013).

Siklus hidup pengembangan sistem (SDLC) adalah proses memahami bagaimana sistem informasi (IS) dapat mendukung kebutuhan bisnis dengan merancang sistem, membangunnya, dan menyampaikan ke pengguna. (Dennis et al., 2015)

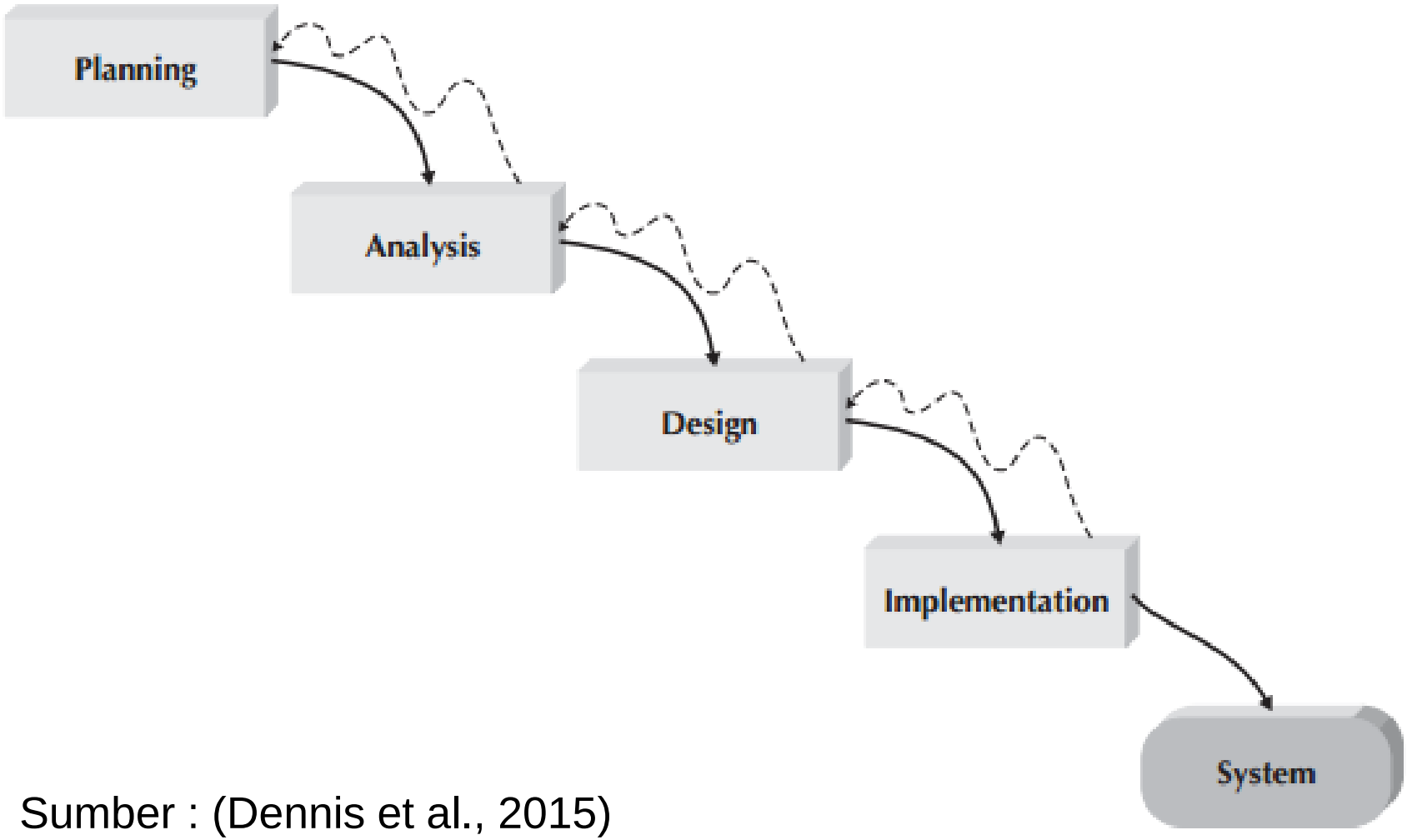
II. Software/Systems Development Life Cycle (SDLC) (Lanjutan)

Kata Kunci dalam ***Systems Development Life Cycle*** adalah Analis sistem, yang menganalisis situasi bisnis, mengidentifikasi peluang untuk perbaikan, dan mendesain sistem informasi untuk diimplementasikan. (Dennis et al., 2015)

Dalam pelaksanaannya, fase yang digunakan dapat berbeda-beda dan pada intinya empat fase umum yang ada termasuk dalam fase yang digunakan.

SDLC memiliki **fase secara umum** yaitu Planning, Analisis, Desain dan Implementasi (Dennis et al., 2015)

a. Fase Umum SDLC



Sumber : (Dennis et al., 2015)

a. Fase Umum SDLC (Lanjutan)

1. *Planning* (Perencanaan)

Fase perencanaan adalah proses fundamental untuk memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana tim proyek akan membangunnya

2. Analisis

Fase analisis menjawab pertanyaan tentang bagaimana menggunakan sistem, apa yang akan dilakukan sistem, dimana dan kapan akan digunakan.

3. Desain

Desain secara bertahap menentukan bagaimana sistem akan beroperasi, dalam hal perangkat keras, perangkat lunak, dan infrastruktur jaringan; antarmuka pengguna, formulir, dan laporan; dan program-program spesifik, database, dan file yang dibutuhkan.

4. Implementasi

Fase terakhir dalam SDLC adalah fase implementasi, di mana sistem sebenarnya dibangun (atau dibeli, dalam kasus desain software yang dikemas). Ini adalah fase yang biasanya mendapat perhatian paling besar, karena untuk kebanyakan sistem itu adalah yang terpanjang dan paling mahal bagian dari proses pengembangan.

b. Model SDLC

Menurut (Sukamto dan Shalahuddin, 2013), Model SDLC terdiri dari :

1. Model *Waterfall*
2. Model *Prototipe*
3. Model *Rapid Application Development* (RAD)
4. Model *Iteratif*
5. Model *Spiral*

Adapun model yang dibahas dalam matakuliah ini adalah model *Waterfall*

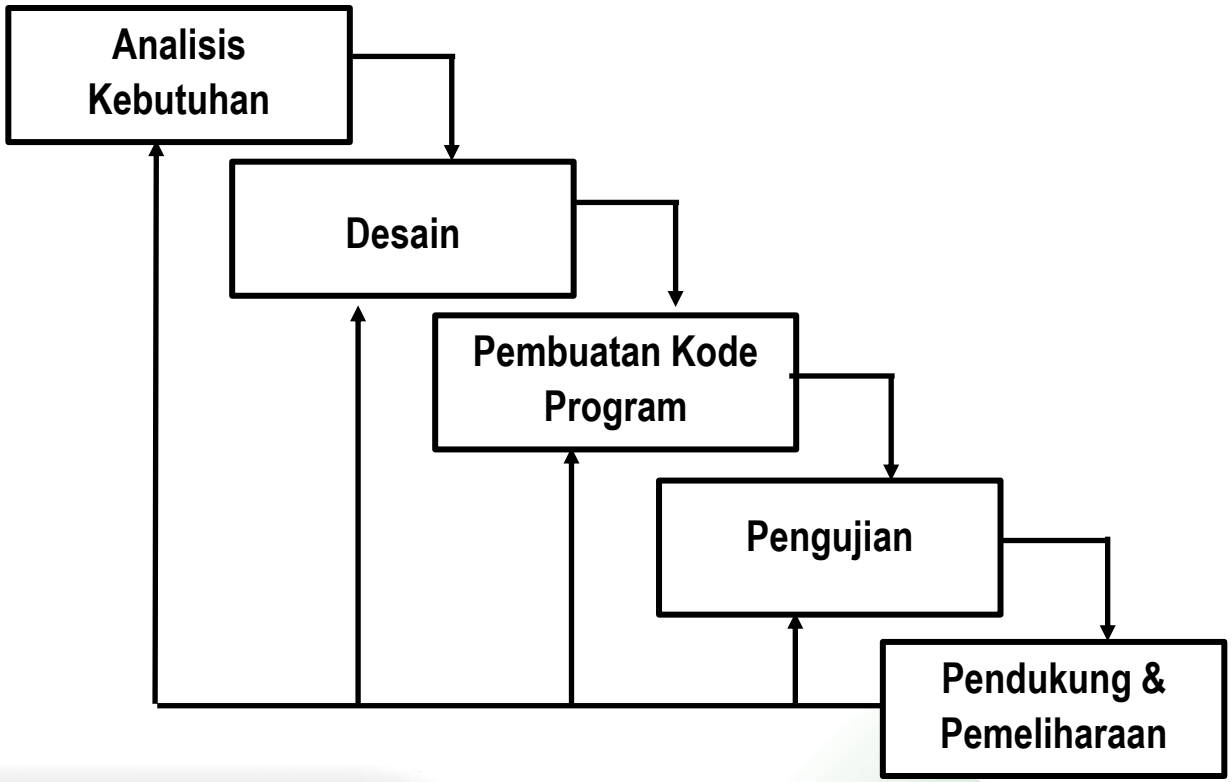
III. Model *Waterfall*

Model *waterfall* adalah model SDLC yang paling sederhana. Model ini cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-rubah (Sukamto dan Salahudin, 2013)

Metode ini disebut juga **metode air terjun**, karena dengan metode Waterfall para analis dan pengguna dapat melanjutkan fase – fase secara bertahap mulai dari fase atas ke fase paling bawah.

III. Model *Waterfall* (Lanjutan)

Model waterfall ini juga menyediakan pendekatan alur hidup perangkat lunak secara sequential atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung & pemeliharaan.



III. Model *Waterfall* (Lanjutan)

1. Analisis Kebutuhan Perangkat Lunak

Pengumpulan kebutuhan untuk menspesifikasikan kebutuhan perangkat lunak sehingga dapat dipahami kebutuhan dari user.

Misal kebutuhan (*system requirement*) dari sistem penyewaan.

- Halaman *User yaitu* Pengunjung dapat melihat informasi website, Pengunjung melakukan registrasi dan menjadi member, Member dapat melihat daftar pemesanan lapangan, Member dapat *login* dengan *account* yang telah dibuat, Member dapat melakukan pemesanan secara *online,* Member dapat melakukan konfirmasi pembayaran, Member dapat mencetak bukti pemesanan
- Sedangan halaman Admin misalnya Admin dapat mengelola data member, Admin dapat mengelola harga lapangan, Admin dapat mengelola data galeri, Admin dapat mengelola data data berita, Admin dapat megelola data transaksi, Admin dapat mengelola data laporan

2. Desain

Desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean.

Contoh :

Unified Modeling Language (UML) untuk merancang dan mendokumentasikan sistem yang dibuat. Sedangkan untuk menggambarkan relasi antara objek dapat menggunakan *Entity Relationship Diagram* (ERD).

3. Pembuatan Kode Program

Hasil tahap ini adalahprogram komputer sesuari dengan desain yang telah dibuat pada tahap desain.

Contoh: bahasa pemrograman PHP dengan suport database menggunakan MySql .

III. Model *Waterfall* (Lanjutan)

4. Pengujian

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji sehingga keluaran yang dihasilkan sesuai dengan yang diinginkan.

contoh: *Black box testing* dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak

5. Pendukung atau Pemeliharaan

Dikarenakan adanya perubahan ketika sudah dikirimkan ke user. Perubahan dapat terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian.

Contoh: software maupun hardware yang sudah ditentukan baik pada saat pembuatan maupun operasi program.

III. Model *Waterfall* (Lanjutan)

Dua keunggulan utama dari waterfall :

1. Mengidentifikasi kebutuhan sistem sebelum program dibuat
2. Meminimalkan perubahan pada saat proyek berlangsung

Kelemahan utama dalam waterfall adalah Desain harus sepenuhnya ditentukan sebelum pemrograman dimulai.

IV. Metodologi Analisa dan Desain Perangkat Lunak Berorientasi Objek

Metodologi untuk melakukan analisis kebutuhan dan perancangan perangkat lunak dapat dikelompokkan berdasarkan pendekatan yang diambil pada saat melakukan aktivitas tersebut.

Metodologi adalah kesatuan metode, prosedur, konsep atau aturan yang digunakan. Sedangkan metode adalah suatu cara, Teknik yang sistematik untuk mengerjakan sesuatu.

Metodologi analisis dan sistem perangkat lunak dapat dilakukan melalui **dua pendekatan *(approach)***, yaitu:

1. Pendekatan Terstuktur *(Structural Approach)*
2. Berorientasi Objek *(Object Oriented)*

a. Pendekatan Terstruktur

- Orientasi pada Proses dan Data
- Alat yang digunakan : DFD *(Data Flow Diagram)*, ERD *(Entity Relationship Diagram)*, Bagan Terstruktur.
- Karakteristik Rancangan:
  - Modul disusun secara Hirarkis
  - Menggunakan Alur Kendali *(Top to Bottom / Bottom to Top)*
  - Repetisi dalam satu modul
- Konsep kendali standard (Urut, Seleksi, Repetisi)

b. Pendekatan Berorientasi Objek

- **Orientasi pada Obyek**, yaitu memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Pada
- Pendekatan ini, informasi dan proses yang dipunyai oleh suatu Objek “dienkapsulasi” (dibungkus) dalam satu kesatuan.
- **Alat yang digunakan** : contoh : UML (Use case diagram, activity diagram, sequence diagram, dll).
- **Tahapan:**
  - Mendeskripsikan Obyek, Kelas, Atribut dan Daftar operasi
  - Memodelkan relasi antara Obyek dan Kelas
  - Memodelkan Pewarisan

c. Metodologi Berorientasi Objek

- Merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan berorientasi objek secara sistematis.
- Setiap Objek mempunyai informasi-informasi atau atribut-atribut dan perilaku (*behavior*) sebagai suatu operasi pengaturnya.

Atribut (data) :
Pedal, Ban, Rante, Rem

Methode (function/behaviour) :
Maju, Mengerem, Pindah gigi

d. Karakteristik Berorientasi Objek

1. **Pembungkusan (*Encapsulation*)**

   Pembungkusan berfungsi untuk melindungi suatu objek dari dunia luar, sehingga seseorang tidak akan mampu merusak objek yang terbungkus. Objek yang terbungkus dalam suatu kelas baik data maupun fungsinya tidak bisa terlihat apalagi dirubah pada saat objek digunakan.

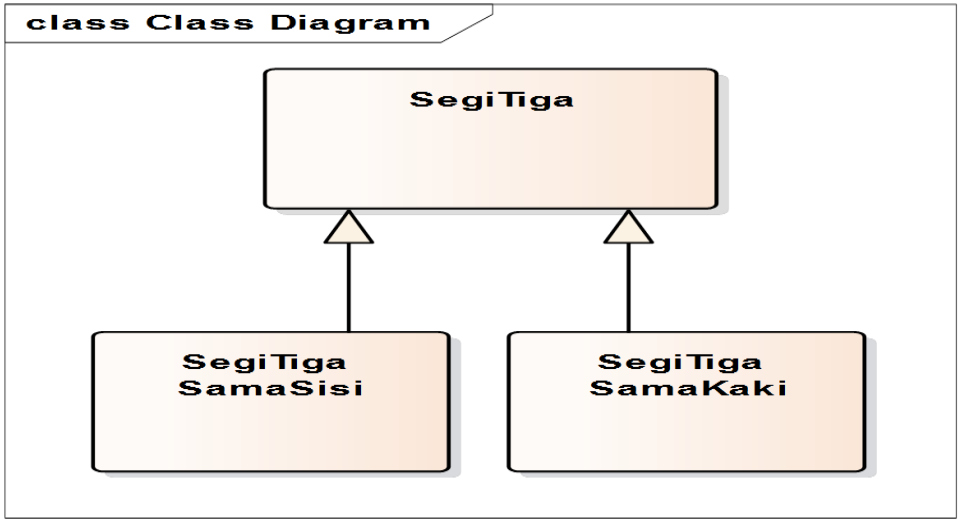
   Dalam Java data dibungkus dengan modifier *private* agar tidak bisa diakses secara langsung dari luar class.

2. Pewarisan (*Inheritance*)

Class dapat menurunkan metode-metode dan properti-properti yang dimilikinya pada class lain. Class yang mewarisi metode dan properti dari objek lain dinamakan class turunan. Class turunan ini mampu mengembangkan metode sendiri.

d. Karakteristik Berorientasi Objek (Lanjutan)

Contoh **Pewarisan (*Inheritance*)**



Dalam gambar diatas, dari **Kelas SegiTiga**, dapat dibuatkan **dua kelas yang baru** yaitu **kelas SegiTiga SamaSisi** dan **SegiTiga SamaKaki**, yang merupakan **turunan dari kelas SegiTiga**. Dalam bahasa pemrograman Java, biasanya dalam kelas turunan terdapat ***keyword extends***.

Penerapan dalam Java

```
public class SamaKaki extends SegiTiga
{
    public SamaKaki(){
    }
    public int hitungLuas(){
        return 0;
    }
}
public class SamaSisi extends SegiTiga
{
    public SamaSisi(){
    }
    public int hitungLuas(){
        return 0;
    }
}
```

3. Keanekaragaman (*Polymorphism*)

Polymorphism dapat diartikan sebagai kemampuan suatu bahasa pemrograman untuk memiliki fungsi-fungsi atau metode yang bernama sama tetapi berbeda dalam parameter dan implementasi kodenya (*overloading*).

Kelas turunan dapat menggunakan fungsi yang ada pada kelas pewarisnya dan dapat mengimplementasikan kode yang berbeda dari fungsi pewarisnya ini dinamakan *overriding*.

d. Karakteristik Berorientasi Objek (Lanjutan)

Pada contoh kelas SegiTiga sebelumnya, dapat ditambahkan fungsi dengan nama yang sama, misalnya hitungLuas, namun dengan penambahan parameter alas dan tinggi.

```
public class SegiTiga{
        public int hitungLuas(){
                int luas = 0;
                luas = (this.getAlas() * this.getTinggi()) / 2;
                return luas;
    }
        public int hitungLuas (int alas, int tinggi){
                int luas = 0;
                luas = (alas * tinggi) / 2;
                return luas;
        }
}
```

Sumber : (Akil, 2018)