

# MODUL 1

## Sistem Informasi dan OOAD

---

### Daftar Isi

1.1	Pengantar Sistem Informasi.....	2
1.1.1	System Transformation .....	3
1.1.2	Karakteristik Sistem .....	3
1.1.3	Sistem dan real Word .....	3
1.1.4	Tipe Sistem Informasi .....	4
1.2	Permasalahan yang Timbul dari Pengembangan Sistem Informasi.....	5
1.3	Menghindari masalah Pengembangan Teknologi Informasi .....	7
1.3.1	Problem Solving.....	7
1.3.2	Project Life Cycle .....	8
	Waterfall.....	8
	TLC with Iteration .....	10
	Prototyping Life Cycle.....	10
	Spiral Model & Incremental Development.....	11
	Unified Software Development Process (USDP).....	12
	Keterlibatan user .....	14
	Agile Approaches.....	14
	Computer Aided Software Engineering (CASE) .....	14
1.4	Pengantar Analisis dan Desain Berbasis Obyek .....	15
1.5	Konsep Analisis dan Desain Berbasis Obyek.....	15
	Referensi .....	19

## 1.1 Pengantar Sistem Informasi

Sistem informasi memiliki peranan yang sangat penting bagi manusia sejak adanya manusia pertama yang mampu melakukan kegiatan yang terorganisir. Saat ini, aplikasi teknologi informasi telah membawa perubahan yang sangat besar pada lingkup dan sifat sistem informasi. Berikut ini kita akan mencoba untuk memahami apa yang dimaksud dengan sistem informasi dengan melihat contoh dari sebuah sistem informasi, elemen-elemen yang dimiliki oleh sistem informasi, karakteristiknya, dan sebagainya.

Salah satu bentuk Sistem informasi (SI) adalah *McGregor On-Line Retail Site*. Sistem informasi ini adalah berupa sebuah situs yang digunakan oleh sebuah perusahaan yang bernama McGregor untuk mengelola transaksi penjualan ritel produk-produk yang dijualnya. Sistem informasi ini berisi:

- *Online catalogue display and shopping cart (katalog online dan shopping chart)*
- *Back-office systems store stock details, orders, payment transactions, dsb (untuk mengelola persediaan, pemesanan, dan transaksi pembayaran).*
- *Communications link to credit-card processing centre (link komunikasi untuk pengolahan kartu credit)*
- *Robot warehouse control system*
- *Delivery scheduling (jadwal pengiriman)*

Elemen-elemen yang dimiliki sistem informasi meliputi :

- Aktivitas manusia yang memerlukan informasi
- Data yang tersimpan
- Metoda input untuk memasukkan data
- Proses yang merubah data menjadi informasi
- Metoda output untuk merepresentasikan informasi

### **Peranan Komputer**

Dalam sebuah sistem informasi, komputer melaksanakan tugas yang dilakukan oleh manusia dan teknologi lainnya, yaitu:

- *Storage (penyimpanan)* : signalman's memory / hard disk
- *Display(tampilan)* : Battle of Britain map / PC screen
- *Calculation (perhitungan)*: mental arithmetic / program
- *Communication (komunikasi)*: telephone line / LAN

Keuntungan menggunakan komputer:

- Kecepatannya yang tinggi, biayanya yang rendah dan keandalannya.

### 1.1.1 System Transformation

Input yang masuk dalam sebuah sistem akan diubah menjadi output melalui transformasi sistem. Semua transformasi sistem mengubah input menjadi output yang lebih berguna . Dalam sistem informasi input dan output bentuknya informasi. Informasi tersebut dapat berupa informasi tentang pemesanan barang, penjualan atau pembayaran. Transformasi bertujuan untuk membangun dan mengoperasikan sistem.

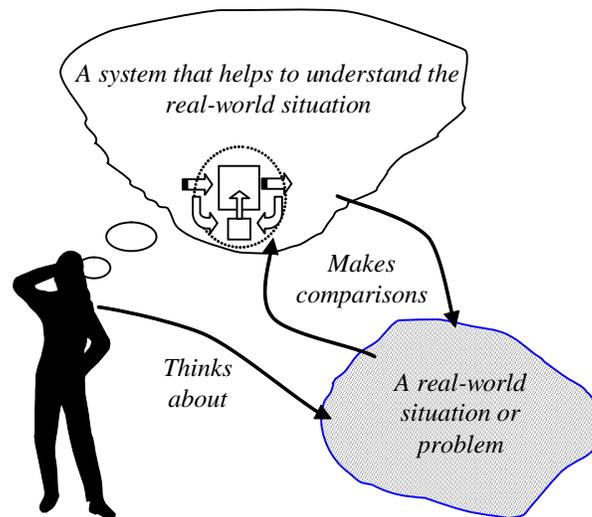
### 1.1.2 Karakteristik Sistem

Sistem informasi mirip dengan sistem. Setiap sistem memiliki :

- Input dan output
- Tujuan ( berkaitan dengan transformasi)
- Batasan dan Lingkungan
- Subsistems dan interface
- Kontrol menggunakan *feedback* dan *feed-forward*
- Beberapa property lainnya

### 1.1.3 Sistem dan *Real Word* (dunia nyata)

Sebuah sistem bisa dalam bentuk *real*, bisa tidak. *Systems thinking (sistem berfikir)* digunakan untuk membantu memahami dan menganalisa permasalahan, yang terpenting adalah pemahaman . Kita bisa menganggap bahwa segala sesuatu sebagai suatu sistem.



**Gambar 1.1 Relasi antara Sistem dan Kenyataan**

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006.

#### 1.1.4 Tipe Sistem Informasi

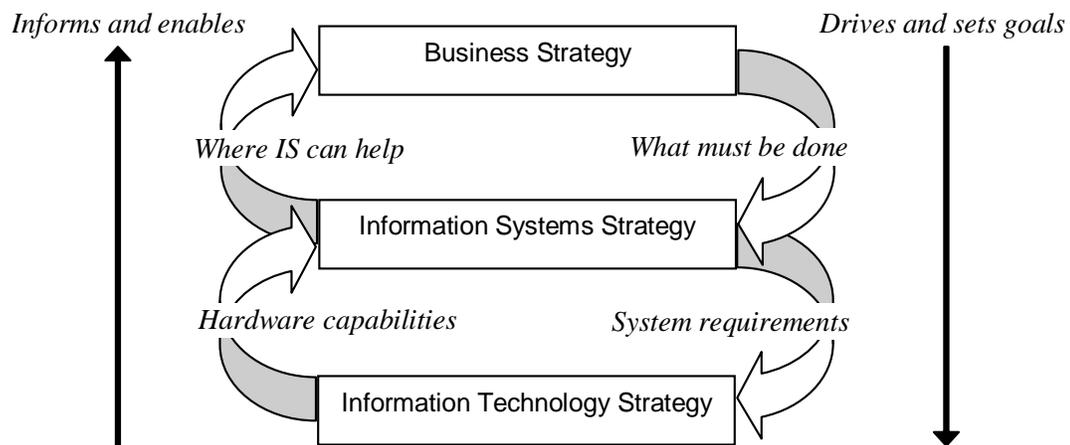
Sistem informasi digunakan untuk membantu pekerjaan manusia seperti :

- Mengambil dan menyimpan data
- Melakukan perhitungan
- Alat bantu komunikasi
- Mengontrol dan membuat rencana kerja
- Dan lain sebagainya ... ?

Sistem operasional membantu atau mengendalikan operasi bisnis. Sebuah sistem akuntansi mampu mengurangi kesalahan yang dilakukan manusia. Sistem informasi membantu manager untuk memutuskan sesuatu atau sebagai bahan komunikasi. Sistem penjadwalan dapat membantu untuk membuat keputusan bagaimana pengiriman barang dilakukan. *Real-time Control Systems* mengoperasikan perangkat fisik, biasanya yang berkaitan dengan keselamatan. Beberapa mobil memiliki *Engine Management System* untuk mengatur *fuel supply and ignition* (bahan bakar dan pengapian).

Kita bisa memandang sebuah organisasi sebagai sebuah sistem, biasanya dengan beberapa subsistem. Idealnya, setiap subsistem membantu sistem secara keseluruhan untuk memenuhi

tujuannya. Sistem informasi juga bisa berupa subsistem dan membantu untuk memenuhi tujuan orang dalam organisasi.



**Gambar 1.2. Relasi antara Bisnis, Sistem Informasi, dan Strategi Teknologi Informasi**

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006.

## 1.2 Permasalahan yang Timbul dari Pengembangan Sistem Informasi

Ada 3 tipe pelaku utama dalam mengembangkan sistem informasi, yaitu:

- Yang menikmati keuntungan dari output sistem baik langsung atau tidak langsung (*end-users*)
- Yang membayar dan berkuasa penuh atas pengembangan sistem (*owners or sponsors*)
- Yang membuat software (*developers*)

Sebuah proyek sistem informasi bisa saja gagal sebelum diserahkan. misalkan proyek dibatalkan sebelum dikembangkan atau sebuah proyek bisa saja gagal ketika dijalankan. Misalkan system mengalami kegagalan pada saat diimplementasikan bahkan sebuah sistem informasi bisa saja terus digunakan walaupun ada masalah yang timbul antara user, pemilik ataupun pengembang.

Kita bisa membagi permasalahan berdasarkan beberapa sudut pandang, yaitu:

- Sudut pandang *End User*
- Sudut pandang *Owner*

- Sudut Pandang *Developer*

### **Sudut pandang End User**

End-users adalah pihak yang secara langsung mengoperasikan software atau bahkan lebih dari itu, misalkan seorang manager yang menerima laporan. Kekhawatiran yang timbul dari user meliputi:

- Sebuah sistem dijanjikan tetapi tidak disampaikan
- Sebuah sistem yang sulit untuk digunakan
- Sebuah sistem yang tidak memenuhi kebutuhan penggunanya

### **Sudut pandang Owner**

Pemilik berkepentingan dalam memenuhi kebutuhan bisnis dan berapa dana yang akan dikeluarkan untuk membangun proyek pembuatan sistem informasi. Kekhawatiran yang timbul dari pihak owner, meliputi:

- Proyek-proyek yang menghabiskan anggaran mereka (mungkin tidak lagi memiliki keuntungan bersih)
- Sistem yang disampaikan terlambat
- Proyek dikelola dengan buruk
- Sistem yang dianggap tidak relevan oleh peristiwa

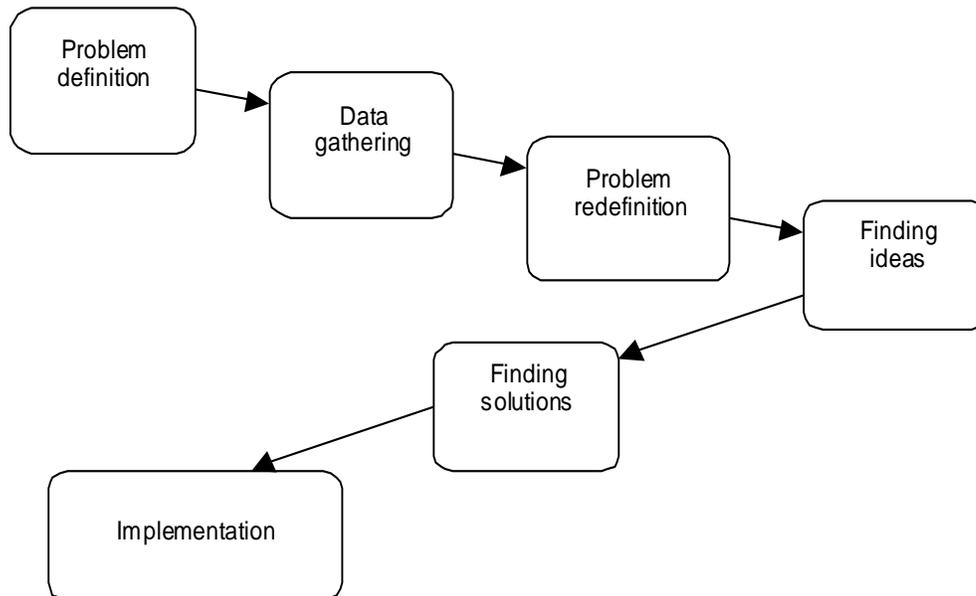
### **Sudut Pandang Developer**

Pengembang SI kadang dihadapkan pada beberapa masalah, antara lain anggaran dan waktu yang sering bertentangan untuk melakukan pekerjaan dengan benar, pengguna dan pemilik yang mungkin tidak tahu apa yang mereka inginkan, dan teknologi, pendekatan pengembangan, dan kebutuhan bisnis yang selalu berubah.

## 1.3 Menghindari masalah Pengembangan Teknologi Informasi

### 1.3.1 Problem Solving

Berikut ini adalah model pemecahan masalah secara umum yang diadopsi dari Hicks, 1991.



**Gambar 1.2 General problem solving model (adapted from Hicks, 1991).**

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006.

Fase utama dalam model pemecahan masalah secara umum adalah:

- Pendefinisian Masalah
- Pengumpulan data
- Redefinisi masalah , hal ini berfokus pada pemahaman akan masalah
- Pencarian ide, berkaitan dengan pemahaman tentang sifat dari masalah dan solusi yang mungkin
- Mencari solusi
- Implementasi

### 1.3.2 Project Life Cycle (Siklus Hidup Proyek)

Pekerjaan pengembangan sistem dikerjakan sebagai suatu proyek. Proyek dapat didefinisikan sebagai usaha sementara, temporer, dan bukan permanen, yang memiliki sasaran khusus dengan waktu pelaksanaan yang tegas.

Terdapat perbedaan antara pengembangan sistem dan pengembangan perangkat lunak adalah pengembangan sistem mencakup manusia, perangkat lunak, dan elemen perangkat keras. Sedangkan pengembangan perangkat lunak, terutama berkaitan dengan sistem perangkat lunak.

Pada proyek pengembangan sistem informasi terdapat dua fase penting, yaitu perencanaan sistem informasi strategis dan pemodelan bisnis. Pada perencanaan sistem informasi strategis, fokusnya bukan pada komputerasi tetapi pada identifikasi persyaratan/kebutuhan organisasi. Sistem informasi bekerja dalam konteks organisasi dan harus memenuhi kebutuhan organisasi saat ini serta memberikan dasar dari mana kebutuhan masa depan dapat diatasi. Contohnya, pada studi kasus tentang perusahaan Agate (Agate adalah sebuah perusahaan periklanan) keputusan strategis dapat dibuat untuk menargetkan perusahaan-perusahaan multinasional untuk kampanye iklan internasional. Ini memiliki konsekuensi untuk manajemen kampanye dan sistem informasi pendukungnya.

Pada pemodelan bisnis, untuk menentukan bagaimana suatu sistem informasi dapat mendukung aktifitas bisnis tertentu maka penting untuk memahami bagaimana aktifitas dilakukan dan bagaimana aktifitas ini memberikan sumbangan terhadap fungsi bisnis penting Agate dan harus dimodelkan untuk menentukan bagaimana aktifitas tersebut dilaksanakan sehingga menyediakan beberapa parameter untuk pengembangan sistem informasi selanjutnya.

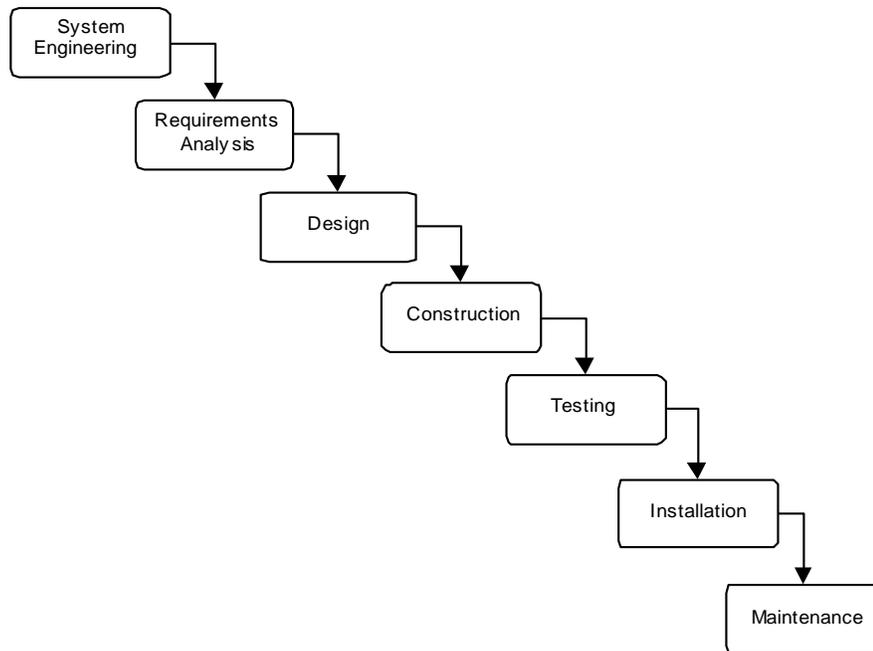
### 1.3.3 Metodologi untuk Siklus Hidup Pengembangan Sistem

Terdapat beberapa metodologi untuk pengembangan sistem informasi. Berikut ini akan dibahas metodologi waterfall, prototyping, iteratif dan incremental development serta Unified Software Development Process (USDP).

#### Waterfall

*The traditional life cycle* (TLC) siklus hidup tradisional untuk pengembangan sistem informasi, juga dikenal sebagai model siklus hidup air terjun. Disebut demikian karena kesulitan untuk kembali ke fase

sebelumnya. Model yang ditampilkan di sini adalah salah satu dari beberapa alternatif. Hasilnya ditampilkan untuk setiap tahap.



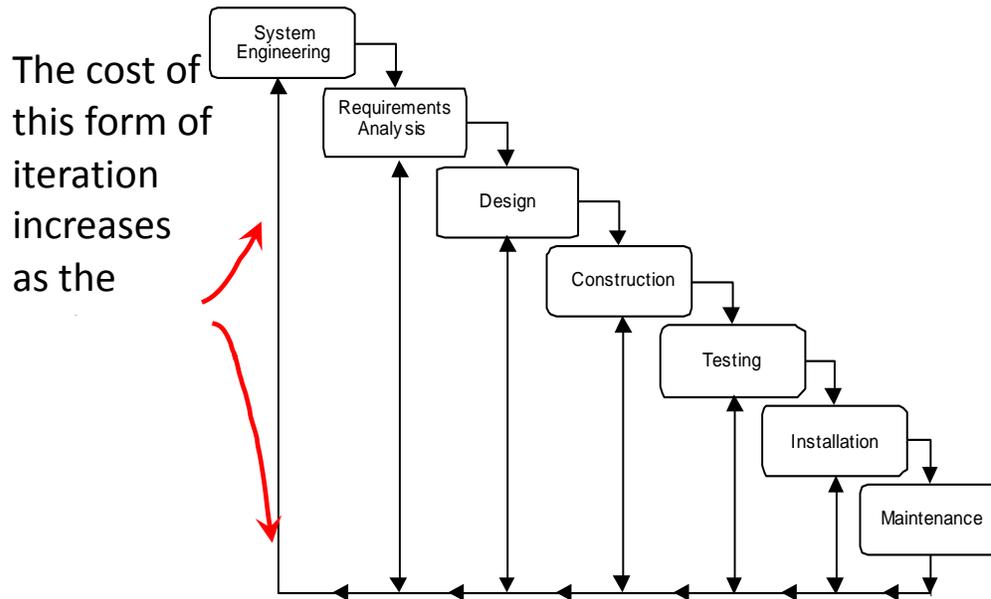
Gambar 1.4 Model Siklus Hidup Tradisional-Waterfall

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006

Masalah yang timbul dari TLC antara lain :

- Proyek-proyek yang sebenarnya jarang mengikuti siklus hidup ini
- Penyimpangan waktu antara rekayasa sistem dan instalasi akhir panjang
- Iterasi hampir tak dapat dilakukan dalam proyek-proyek sebenarnya dan akan menjadi mahal & bermasalah dengan TLC
- Responsif terhadap perubahan selama proyek sulit dikerjakan

## TLC with Iteration



Gambar 1.5 Model Siklus Hidup Tradisional-Waterfall dengan Feedback dan Loops

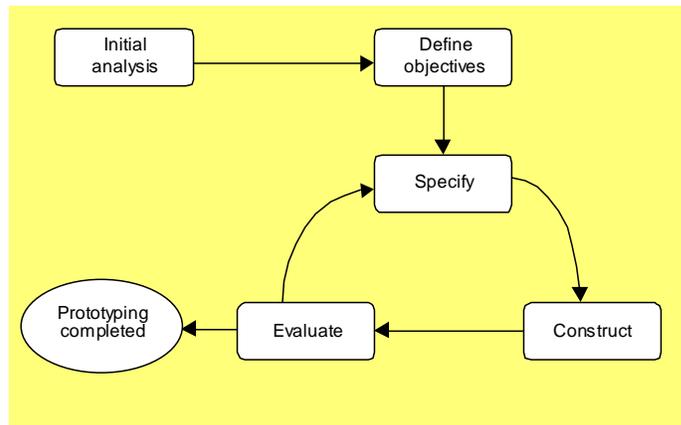
Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006

Kelebihan TLC :

- Tugas secara bertahap dapat diberikan kepada tim khusus.
- Kemajuan proyek dievaluasi pada akhir setiap tahap.
- Dapat digunakan untuk mengelola proyek dengan tingkat tinggi risiko.

## Prototyping Life Cycle

Model waterfall bertujuan untuk menyerahkan sistem yang sudah final sebagai produk jadi ke pengguna. Akan tetapi pendekatan ini memiliki kesulitan, yaitu pengguna hanya akan benar-benar memiliki pengalaman bagaimana sebuah sistem bekerja pada saat sudah diserahkan. Kadangkala pengguna kesulitan untuk membayangkan bagaimana kebutuhan mereka ditranslasikan ke dalam sistem kerja dan jadi masalah bahwa penerapan yang berbeda dapat dihasilkan dari persyaratan yang sama. Pendekatan prototyping mengatasi banyak potensi kesalahpahaman dan ambiguitas yang ada dalam persyaratan.



Gambar 1.6 Siklus Hidup Prototyping

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006

Kelebihan Prototyping :

- Demonstrasi awal fungsi sistem membantu mengidentifikasi kesalahpahaman antara pengembang dan klien
- Kebutuhan klien diidentifikasi
- Kesulitan dalam antarmuka dapat diidentifikasi
- Kelayakan dan kegunaan dari sistem dapat diuji meskipun prototipe tidak lengkap

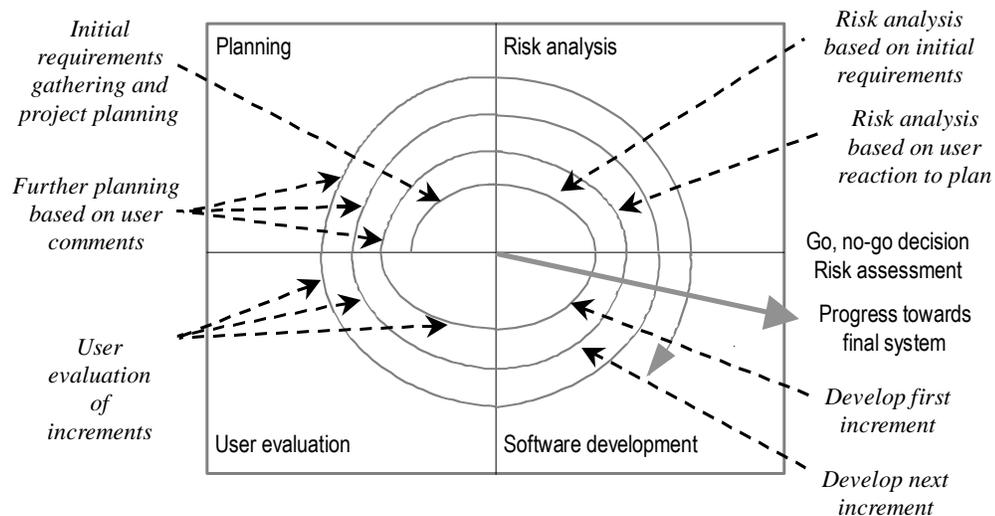
Masalah Prototyping

- Klien mungkin merasa prototipe sebagai bagian dari sistem final
- Prototipe dapat mengalihkan perhatian dari masalah fungsional ke masalah antarmuka
- Prototyping memerlukan keterlibatan pengguna yang signifikan
- Mengelola siklus hidup prototyping membutuhkan pengambilan keputusan yang hati-hati

### **Spiral Model & Incremental Development**

Sistem besar yang sukses bermula dari sistem-sistem kecil yang dikembangkan secara bertahap. Pendekatan incremental (bertahap) melaksanakan analisis awal lingkup masalah dan mengidentifikasi persyaratan utama. Persyaratan tersebut akan memberikan keuntungan kepada klien dipilih menjadi

fokus dari tahap pertama pengembangan dan penyerahan sistem. Instalasi dari tiap tahap menyediakan umpan balik untuk tim pengembangan dan menginformasikan pengembangan tahap selanjutnya.



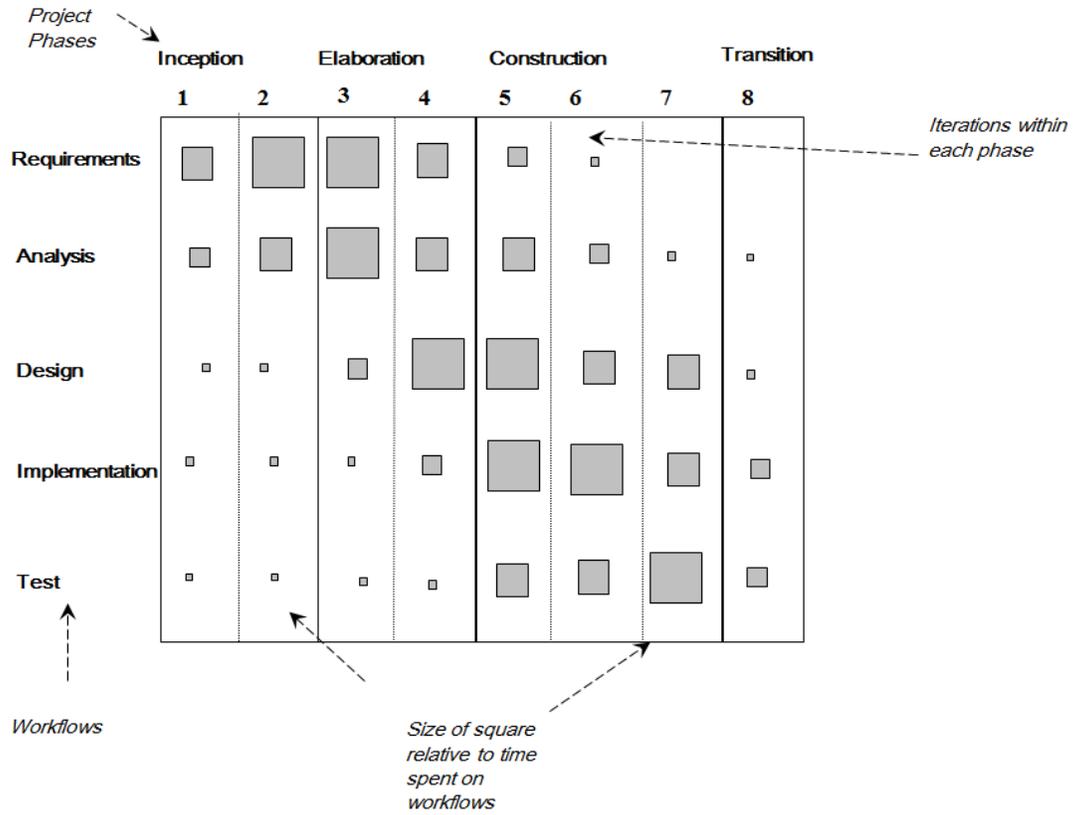
Gambar 1.7 Model Spiral untuk Penyerahan Bertahap

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006

### Unified Software Development Process (USDP)

Metodologi ini mencerminkan penekanan pada siklus hidup iteratif dan bertahap. Model ini menangkap banyak unsur praktek terbaik. Tahapannya adalah:

- Inception, berkaitan dengan penentuan ruang lingkup dan tujuan proyek;
- Elaborasi, berfokus pada menangkap persyaratan dan menentukan struktur sistem;
- Konstruksi, tujuan utamanya adalah untuk membangun sistem perangkat lunak;
- Transisi, berkaitan dengan produk instalasi dan peluncuran



Gambar 1.8 Unified Software Development Process (USDP)

Sumber: Simon Bennet, Steve McRobb and Ray Farmer, Object Oriented Systems Analysis and Design Using UML, Edisi 3. ; McGraw Hill, 2006

### **Keterlibatan user**

Faktor penting untuk memaksimalkan kesuksesan adalah memastikan adanya keterlibatan pengguna yang efektif dan keberlanjutan di seluruh proyek. Pengguna dapat terlibat pada berbagai tingkat, yaitu:

- Sebagai bagian dari tim pengembangan (DSDM)
- Melalui pendekatan konsultatif
- Dalam berbagai pertemuan

### **Agile Approaches**

Pendekatan ini adalah pendekatan dimana terdapat perulangan secara ringan pada fase-fase perkembangannya. Pendekatan ini menerima kenyataan bahwa kebutuhan pengguna akan berubah selama perkembangan sistem informasi. Contoh dari pendekatan ini adalah XP (extreme programming)

### **Computer Aided Software Engineering (CASE)**

CASE tools biasanya menyediakan berbagai fitur termasuk untuk memeriksa kebenaran sintaksis, dukungan repositori, memeriksa konsistensi dan kelengkapan, dan navigasi untuk diagram terkait.

Fitur dari CASE tools antara lain:

- *layering;*
- *traceability;*
- *report generation;*
- *system simulation;*
- *performance analysis;*
- *code generation.*

## 1.4 Pengantar Analisis dan Desain Berbasis Obyek

Definisi analisis dan desain berbasis obyek adalah sebagai berikut:

- Analisa :
  - What
  - Memahami permasalahan bisnis, tidak tergantung pada solusi teknologi.
  - Lebih menekankan pada apa yang menjadi permasalahan
- Desain :
  - How
  - Memahami dan mendefinisikan solusi software yang merepresentasikan hasil analisa dan akan diimplementasikan dalam bentuk *code*
  - Tergantung pada solusi teknologi
- OOAD :
  - Pengembangan Software yang menggunakan pendekatan obyek/menekankan solusi yang berdasarkan obyek-obyek
  - Memahami permasalahan dan solusi logik dari sudut pandang obyek(benda, konsep, entitas)

## 1.5 Konsep Analisis dan Desain Berbasis Obyek

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek. Analisis berorientasi objek (OOA) adalah metode analisis yang memeriksa *requirement* (syarat/keperluan yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Sedangkan desain berorientasi objek (OOD) adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Terdapat beberapa konsep dasar dalam OOAD, yaitu :

### a. Objek (object)

**Objek** adalah benda secara fisik dan konseptual yang ada di sekitar kita. Beberapa contoh objek, misalnya hardware, software, dokumen, manusia, konsep, dan lainnya.

Untuk kepentingan pemodelan, misalnya seorang eksekutif akan melihat karyawan, gedung, divisi, dokumen, keuntungan perusahaan sebagai sebuah objek. Sedangkan seorang teknisi mobil, akan melihat ban, pintu, mesin, kecepatan tertentu dan banyaknya

bahan baker sebagai sebuah objek. Contoh lainnya adalah seorang software engineer akan memandang tumpukan, antrian intruksi, window, check box sebagai sebuah objek.

Sebuah objek mempunyai keadaan sesaat yang disebut state.

**State** dari sebuah objek adalah kondisi dari objek itu atau himpunan keadaan yang menggambarkan objek tersebut. Sebagai contoh, state dari rekening tabungan, dapat memuat saldo yang berjalan, state dari sebuah jam adalah catatan saat itu; sedangkan state dari sebuah bohlam lampu adalah suatu keadaan “nyala” atau “mati”. State dinyatakan dengan nilai dari atribut objeknya.

**Atribut** adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain dan identitas. Perubahan state dicerminkan oleh perilaku (behaviour) objek tersebut.

**Behaviour** atau perilaku sebuah objek mendefinisikan bagaimana sebuah objek bertindak(beraksi) dan memberi reaksi. Behaviour ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan oleh objek itu sendiri. Behaviour dari sebuah objek dicerminkan oleh interface, service dan method dari objek tersebut.

**Interface** adalah pintu untuk mengakses service dari objek.

**Service** adalah fungsi yang dapat dikerjakan oleh sebuah objek.

**Method** adalah mekanisme internal objek yang mencerminkan perilaku(behaviour) objek tersebut.

#### b. Kelas (Class)

Class adalah definisi umum (pola, template, atau cetak biru) dari himpunan objek yang sejenis. Kelas menetapkan spesifikasi perilaku (behaviour) dan atribut-atribut dari objek tersebut. Class adalah abstraksi dari entitas dalam dunia nyata. Sedangkan objek adalah contoh (“instances”) dari sebuah kelas.

Misalnya, atribut dari kelas binatang adalah berkaki empat dan mempunyai ekor. Perilakunya adalah makan dan tidur. Sedangkan contoh (instance) untuk kelas binatang ini adalah kucing, gajah, dan kuda.

c. Kotak Hitam (Black Boxes)

Sebuah objek adalah **kotak hitam (black-boxes)**. Konsep ini menjadi dasar untuk implementasi objek. Dalam operasi OO, hanya para developer (programmer, desainer, analis) yang dapat memahami detail dari proses-proses yang ada didalam kotak hitam tersebut, sedangkan para pemakai (user) tidak perlu mengetahui apa yang dilakukan, tetapi yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka.

**Encapsulation**, proses menyembunyikan detail implementasi sebuah objek. Satu-satunya jalan untuk mengakses data objek tersebut adalah melalui interface. Interface melindungi internal state sebuah objek dari “campur tangan” pihak luar. Oleh karena itu objek digambarkan sebagai sebuah kotak hitam yang menerima dan mengirim pesan-pesan (messages). Dalam OOP kotak hitam tersebut berisi kode (instruksi yang dipahami computer) dan data (informasi dimana instruksi tersebut beroperasi dengannya). Dalam OOP kode dan data disatukan dalam sebuah “benda” yang tersembunyi isinya yaitu objek. Pengguna objek tidak perlu mengetahui isi dalam kotak tersebut; untuk berkomunikasi dengan objek, diperlukan pesan(message).

**Message** adalah permintaan agar objek menerima (receive) untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (sender). Contohnya, satu objek orang mengirim kepada objek bola lampu sebuah pesan message untuk menyalakan melalui saklar. Objek bola lampu memiliki perilaku yang akan mengubah keadaannya(state) dari padam menjadi menyala. Objek lampu menyalakan dirinya dan menunjukkan kepada objek orang tersebut bahwa state barunya adalah menyala.

d. Asosiasi dan Agregasi

**Asosiasi** adalah hubungan yang mempunyai makna antara sejumlah objek. Asosiasi digambarkan dengan sebuah garis penghubung di antara objeknya. Contoh :

Asosiasi antara objek mobil dengan seseorang. Mobil dapat dimiliki oleh satu atau beberapa orang, sedangkan seseorang dapat mempunyai nol, satu atau banyak mobil

Asosiasi antara karyawan dengan unit-kerja. Seorang karyawan bekerja di satu unit-kerja. Sedangkan sebuah unit-kerja dapat memiliki beberapa orang karyawan

**Agregasi** adalah bentuk khusus sebuah asosiasi yang menggambarkan seluruh bagian pada satu objek merupakan bagian dari objek yang lain (◊). Contoh :

Kopling dan piston adalah bagian dari mesin. Sedangkan mesin, roda, body adalah merupakan bagian dari sebuah mobil.

Tanggal, bulan dan tahun adalah bagian dari tanggal-lahir. Sedangkan tanggal-lahir, nama, alamat, jenis kelamin adalah bagian dari identitas Seseorang.

## Referensi

1. Simon Bennet, Steve McRobb and Ray Farmer, *Object Oriented Systems Analysis and Design Using UML*, Edisi 3. ; McGraw Hill, 2006. (SB)
2. Edward V. Berard, *Origins Objects Oriented Technology*,,  
[http://www.toa.com/pub/net\\_articles/origin\\_articles.txt](http://www.toa.com/pub/net_articles/origin_articles.txt)
3. Edward V. Berard, *Origins Objects Oriented Technology*,,  
<http://www.toa.com/pub/oobasics/oobasics.html>