

# Teknik Informatika

Rabu, 23 Maret 2016

(OOAD & SSAD)

## MAKALAH Analisis Desain Berorientasi Objek dan Terstruktur (OOAD & SSAD)



Disusun oleh :

Annas Hendi Astomo	(145410187)
Daniel Christian Novianto	(145410023)
Siti Mutiah	(145410103)
Sulastri	(155410031)

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN  
KOMPUTER  
AKAKOM YOGYAKARTA  
2015/2016**

### KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga kami dapat menyelesaikan makalah berjudul Analisis Desain Berorientasi Objek dan Terstruktur (OOAD & SSAD) dengan tepat waktu.

Kami berterimakasih kepada Vebi Nova Lenti selaku dosen pembimbing mata kuliah Analisis Desain Berorientasi Objek yang telah memberikan tugas ini sehingga kami dapat menambah pengetahuan kami. Kami berharap makalah ini dapat bermanfaat bagi kita semua.

### Mengenai Saya

Unknown

[Lihat profil lengkapku](#)

### Arsip Blog

▼ 2016 (3)

► April (1)

▼ Maret (2)

[Analisis Desain Berorientasi Obyek](#)

(OOAD & SSAD)

Dalam proses penyusunan makalah inilah penulis berupaya untuk mengumpulkan buku-buku referensi dan berbagai tulisan di media massa yang bermutu dan benar benar dapat dipertanggung jawabkan kualitasnya.

Kepada semua pihak baik secara langsung maupun tidak langsung telah membantu demi kesempurnaan makalah ini kami mengucapkan rasa terima kasih. Tak ada gading yang tak retak, demikian pula dalam pembuatan makalah ini tentu saja banyak kekurangan, oleh sebab itu kritik dan saran sangat diharapkan untuk pembuatan makalah yang lebih baik di kemudian hari.

Yogyakarta,     Maret 2016

Penyusun

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Sudah diketahui bersama bahwa perkembangan teknologi saat ini sangatlah cepat. Manusia moderen saat ini dituntut untuk dapat bekerja dengan cepat, misalkan saja dulu pengisian formulir pendaftaran suatu kegiatan masih menggunakan tulisan tangan atau manual namun saat ini proses pengumpulan data dituntut untuk cepat, benar dan *realtime*. Sehingga tidak heran jika sistem-sistem informasi saat ini bermunculan. Sistem Informasi ini akan memudahkan *user* dalam proses administrasi maupun pelaporan. Dalam pembuatan sistem informasi ini tidak serta-merta *programmer* dapat menyelesaikannya, tentu saja terdapat proses-proses yang dilalui untuk membuat sistem tersebut mulai dari perencanaan, analisis, desain, pengembangan, *testing*, implementasi, pengopersian dan pemeliharaan. Pada makalah ini akan dibahas dua bagian proses pembuatan sistem informasi yaitu analisis dan desain. Analisis merupakan kegiatan menganalisa workflow sistem informasi yang sedang berjalan dan mengidentifikasi apakah workflow telah efisien dan sesuai standar tertentu. Desain adalah langkah yang sangat penting dalam pembuatan sistem informasi karena langkah ini menentukan pondasi sistem informasi. Kesalahan dalam desain dapat menimbulkan hambatan bahkan kegagalan proyek. Mengingat pentingnya analisis dan desain dalam pembuatan sistem informasi maka perlu diketahui lebih lanjut mengenai Analisis dan Desain Berorientasi Objek (OOAD) dan Analisis dan Desain Terstruktur (SSAD).

#### **B. Rumusan Masalah**

1. Apakah pengertian dan konsep dari OOAD (Object Oriented Analysis and Design) dan SSAD (Structured System Analisis Design)?
2. Apakah kelebihan OOAD dan SSAD?
3. Apakah kekurangan OOAD dan SSAD?
4. Apa sajakah isu-isu terkini mengenai OOAD dan SSAD?

#### **C. Tujuan**

1. Mengetahui pengertian OOAD dan SSAD.
2. Mengetahui kelebihan OOAD dan SSAD.
3. Mengetahui kekurangan OOAD dan SSAD.
4. Mengetahui isu-isu terkini mengenai OOAD dan SSAD.

## **BAB II**

### **PEMBAHASAN**

#### **A. Pengertian dan Konsep**

I. Analisis dan Desain Berorientasi Objek (Object Oriented Analysis and Design(OOAD)) adalah suatu pendekatan rekayasa perangkat lunak dari sebuah sistem yang terdiri dari sekelompok objek yang saling berinteraksi,

dan setiap objek itu mewakili beberapa entitas. Yang ditandai dengan adanya sebuah kelas, elemen data dan perilaku dari objek tersebut.

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

#### 1. Karakteristik Object

Ada 2 karakteristik dari objek :

##### a. Objek/Object

- Objek adalah benda secara fisik dan konseptual yang ada di sekitar kita. Sebuah objek memiliki keadaan sesaat yang disebut state.
- Objek dapat kongkrit, seperti halnya arsip dalam sistem, atau konseptual seperti kebijakan penjadwalan dalam multiprocessing pada sistem operasi.
- Dua objek dapat berbeda walaupun bila semua nilai atributnya identik.



Mobil	Singa
NoPeg	Nama
96001	Susan
96002	David
97001	Shila

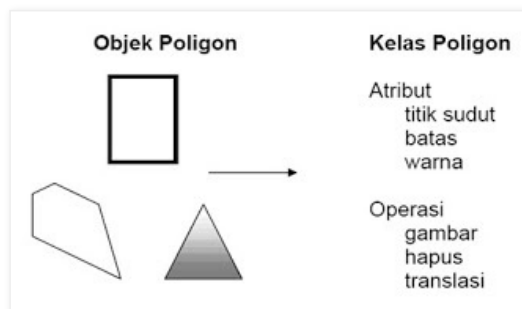
Tabel

Gambar 1. Macam-macam Objek

##### a. Kelas/Class

Kelas merupakan gambaran sekumpulan Objek yang terbagi dalam atribut, operasi, metode, hubungan, dan makna yang sama.

- Suatu kegiatan mengumpulkan data (atribut) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup.
- Kelas Objek merupakan wadah bagi Objek. Dapat digunakan untuk menciptakan Objek.
- Objek mewakili fakta/keterangan dari sebuah kelas.



Gambar 2. Kelas dan Objek

Istilah-istilah Objek

- Atribut: Data item yang menegaskan Objek.
- Operasi: Fungsi di dalam kelas yang dikombinasikan ke bentuk tingkah laku kelas.
- Metode: Pelaksanaan prosedur (badan dari kode yang mengeksekusi respon terhadap permintaan objek lain di dalam sistem).

#### 1. Karakteristik Metodologi Berorientasi Object

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama :

**a. Encapsulation (Pengapsulan)**

- Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
- Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.
- Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

**b. Inheritance (Pewarisan)**

- Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung.
- Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya.
- Inheritance mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki.
- Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.
- Kelas Objek dapat didefinisikan atribut dan service dari kelas Objek lainnya.
- Inheritance menggambarkan generalisasi sebuah kelas.

**c. Polymorphism (Polimorfisme)**

- Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.
- Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.
- Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama.
- Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Objek.

**I. Analisis dan Desain Terstruktur (Structured System Analysis and Design (SSAD))**

Metodologi yang umumnya digunakan dalam pembangunan sistem berbasis komputer dalam dunia bisnis dan industri saat ini adalah metode analisis dan desain terstruktur (Structured Analysis and Design / SSAD). Metode ini diperkenalkan pada tahun 1970, yang merupakan hasil turunan dari pemrograman terstruktur. Metode pengembangan dengan metode terstruktur ini terus diperbaiki sampai akhirnya dapat digunakan dalam dunia nyata.

Melalui SSAD, permasalahan yang kompleks di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan).

**Elemen Perancangan Terstruktur**

1. Modul merupakan sebuah instruksi atau sekumpulan instruksi program yang terdiri dari : input(masukan), output(keluaran), fungsi, mekanisme dan data internal. Contoh : Foxpro / Pascal (Procedure, function), COBOL (Program, section, paragraph), FORTRAN (subroutine).

2. Bagan terstruktur (Structured Chart) : Menggambarkan partisi sistem ke dalam : modul-modul, organisasi, dan komunikasi. Keuntungannya ; Menggunakan gambar, Dapat dipartisi, Fleksibel, Input sangat berguna pada implementasi, Membantu pemeliharaan (maintenance) dan modifikasi.

3. Strategi Perancangan : Mentransformasikan hasil analisis (DFD) menjadi Bagan Terstruktur, untuk diimplementasi. DFD memperlihatkan aliran data dan informasi dari sistem. Jika dalam suatu DFD aliran datanya ditentukan oleh suatu data item, misalnya 'T' yang mempunyai nilai/ karakteristik tertentu, kemudian nilai ini akan mempengaruhi atau menentukan arah aliran data (men-trigger arah), maka titik proses dimana terjadi percabangan arah aliran data tsb disebut titik pusat transaksi

4. Optimasi dari perancangan (Design Heuristic). Metodologi Perancangan Terstruktur

1. Metodologi pemecahan fungsional : Metodologi ini menekankan pada pemecahan sistem ke dalam subsistem-subsistem yang lebih kecil, sehingga akan lebih mudah untuk dipahami, dirancang, dan diterapkan.
2. Metodologi berorientasi data : Metodologi ini menekankan pada karakteristik data yang akan diproses.
3. Prescriptive methodologies : Metodologi ini merupakan metodologi yang dikembangkan oleh sistem house dan pabrik-pabrik perangkat lunak dan tersedia secara komersial dalam paket-paket program.

#### **A. Kelebihan dan Kekurangan**

##### **Kelebihan**

##### **- OOAD**

1. Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalam pembangunan system.
2. Dibandingkan dengan SSAD, waktu pengembangan, level organisasi, ketangguhan dan penggunaan kembali (re-use) kode program lebih tinggi dibandingkan dengan metode OOAD.
3. Tidak ada pemisahan antara fase desain dan analisi, sehingga meningkatkan komunikasi antara user dan developer dari awal hingga akhir pembangunan system.
4. Analis dan programmer tidak dibatasi dengan batasan implementasi system.
5. Relasi obyek dengan entitas (thing) umumnya dapat di-mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam system. Hal ini memudahkan dalam memahami desain.
6. Memungkinkan adanya perubahan dan kepercayaan diri yang tinggi terhadap kebenaran software yang membantu untuk mengurangi resiko pada pembangunan system yang kompleks.
7. Encapsulasi data dan method, memungkinkan penggunaan kembali pada proyek lain, hal ini akan memperjelas proses desain, pemrograman dan reduksi harga.
8. OOAD memungkinkan adanya standarisasi obyek yang akan memudahkan memahami desain dan mengurangi resiko pelaksanaan proyek.
9. Dekomposisi obyek, memungkinkan seorang analis untuk memecah masalah menjadi pecahan-pecahan masalah dan bagian-bagian yang di-manage secara terpisah. Kode program dapat dikerjakan bersama-sama. Metode ini memungkinkan pembangunan software dengan cepat, sehingga dapat segera masuk ke pasaran dan kompetitif. System yang dihasilkan sangat fleksibel dan mudah dalam memelihara.

##### **- SSAD**

1. Milestone diperlihatkan dengan jelas yang memudahkan dalam manajemen proyek
2. SSAD merupakan pendekatan visul, ini membuat metode ini mudah dimengerti oleh pengguna atau programmer
3. Penggunaan analisis grafis dan tool seperti DFD menjadikan SSAD bagus untuk digunakan
4. SSAD merupakan metode yang diketahui secara umum pada berbagai industri
5. SSAD sudah diterapkan begitu lama sehingga metode ini sudah matang dan layak untuk digunakan
6. SSAD memungkinkan untuk melakukan validasi antara berbagai kebutuhan
7. SSAD relative simple dan mudah dimengerti

##### **Kekurangan**

##### **- OOAD**

1. Pada awal desain OOAD, system mungkin akan sangat simple.
2. Pada OOAD lebih focus pada coding dibandingkan dengan SSAD.
3. Pada OOAD tidak menekankan pada kinerja team seperti pada SSAD.
4. Pada OOAD tidak mudah untuk mendefinisikan class dan obyek yang dibutuhkan system.
5. Seringkali pemrograman berorientasi obyek digunakan untuk melakukan analisis terhadap fungsional site, sementara metode OOAD tidak berbasis pada fungsional system.

6. OOAD merupakan jenis manajemen proyek yang tergolong baru, yang berbeda dengan metode analisis dengan metode terstruktur. Konsekuensinya adalah team developer butuh waktu yang lebih lama untuk berpindah ke OOAD karena mereka sudah menggunakan SSAD dalam waktu yang lama.
7. Metodologi pengembangan system dengan OOAD menggunakan konsep re-use. Re-use merupakan salah satu keuntungan utama yang menjadi alasan digunakannya OOAD. Namun demikian, tanpa prosedur yang eksplisit terhadap re-use, akan sangat sulit untuk menerapkan konsep ini pada skala besar.

- **SSAD**

1. SSAD berorientasi utama pada proses, sehingga mengabaikan kebutuhan non-fungsional
2. Sedikit sekali manajemen langsung terkait dengan SSAD
3. Prinsip dasar SSAD merupakan pengembangan non-iterative (waterfall), akan tetapi kebutuhan akan berubah pada setiap proses
4. Interaksi antara analisis atau pengguna tidak komprehensif, karena system telah didefinisikan dari awal, sehingga tidak adaptif terhadap perubahan
5. Selain dengan menggunakan desain logic dan DFD, tidak cukup tool yang digunakan untuk mengkomunikasikan dengan pengguna, sehingga sangat sulit bagi pengguna untuk melakukan evaluasi.
6. Pada SSAD sulit sekali untuk memutuskan ketika ingin menghentikan dekomposisi dan memulai membuat system
7. SSAD tidak selalu memenuhi kebutuhan pengguna
8. SSAD tidak dapat memenuhi kebutuhan terkait bahasa pemrograman berorientasi obyek, karena metode ini memang didesain untuk mendukung bahasa pemrograman terstruktur, tidak berorientasi pada obyek.

### **BAB III PENUTUP**

#### **KESIMPULAN**

1. OOAD adalah suatu pendekatan dari sebuah sistem yang berorientasi pada objek, semua paradigma atau fungsi dibungkus dalam objek.
2. SSAD merupakan pendekatan sistem yang semua objeknya dapat berinteraksi dengan objek lainnya.

#### **DAFTAR PUSTAKA**

1. Mahrizal Ayah Anam, 7 Langkah Membangun Sistem Informasi, [http://www.kompasiana.com/mahrizal/7-langkah-membangun-sistem-informasi\\_55101304a33311ca39ba7eb0](http://www.kompasiana.com/mahrizal/7-langkah-membangun-sistem-informasi_55101304a33311ca39ba7eb0) , 23Maret 2016 13:20
2. Fatkhur Rohman, Kelebihan Kekurangan Model Terstruktur dan Model Berorientasi Obyek, <http://ruhtaf12.blogspot.co.id/2012/01/kelebihan-kekurangan-model-terstruktur.html> , 22 Maret 2016 10:00
3. Tubagus Putra Kencana, Pengenalan OOAD (Object Oriented Analysis and Design), <http://www.tubaguskencana.com/2014/03/pengenalan-ooad-object-oriented.html>, 23 Maret 2016 13:43
4. <http://catatankuliah.blogspot.co.id/>

Diposting oleh [Unknown](#) di 00.46

---

Tidak ada komentar:

## Posting Komentar

Masukkan komentar Anda...

Beri komentar sebagai:

Google Accour ▼

Publikasikan

Pratinjau

[Posting Lebih Baru](#)

[Beranda](#)

Langganan: [Posting Komentar \(Atom\)](#)