

PERTEMUAN 1

System Development and Software Crisis



SUB POKOK BHS :

1. Sejarah Objek
2. Perlunya Pendekatan baru
3. Biaya dan Keuntungan Pendekatan Objek

1. Pendahuluan

- Pengembangan Sistem/Perangkat Lunak merupakan pekerjaan yang rumit
- Pada masa lalu pengembangan software selalu terlambat, mahal, kurang memenuhi harapan dan kebutuhan pengguna.
- Banyak software setelah dikembangkan tidak digunakan.
- Saat ini vendor ingin produk softwarentya segera diterima konsumen.

Sejarah Object Oriented

- Paradigma berorientasi objek mengambil bentuk dari konsep awal pendekatan pemrograman baru, sedangkan minat dalam desain dan metode analisis datang kemudian.
- Bahasa berorientasi objek pertama adalah Simula (Simulasi sistem real) yang dikembangkan pada tahun 1960 oleh para peneliti di Norwegia Computing Center.

Sejarah Object Oriented lanj...

- Pada tahun 1970, Alan Kay dan kelompok riset di Xerox PARC menciptakan komputer pribadi bernama Dynabook dan pertama murni berorientasi obyek bahasa pemrograman (OOPL) - Smalltalk, untuk pemrograman Dynabook.
- Pada 1980-an, Grady Booch menerbitkan sebuah makalah berjudul Desain Berorientasi Obyek yang terutama disajikan desain untuk bahasa pemrograman, Ada. Dalam edisi berikutnya, dia diperpanjang ide-idenya untuk sebuah metode desain berorientasi objek yang lengkap.
- Pada 1990-an, Coad menggabungkan ide-ide perilaku berorientasi objek metode.
- Inovasi penting lainnya adalah Teknik Object Modelling (OMT) oleh James Rumbaugh dan Object-Oriented Software Engineering (OOSE) oleh Ivar Jacobson.

Manfaat Object Model

Manfaat menggunakan model objek adalah:

- Membantu dalam pengembangan lebih cepat dari perangkat lunak.
- Sangat mudah untuk mempertahankan. Misalkan modul mengembangkan kesalahan, maka programmer dapat memperbaikinya modul tertentu, sedangkan bagian lain dari perangkat lunak yang masih berdiri dan berjalan.
- Mendukung upgrade yang relatif bebas repot.
- Hal ini memungkinkan penggunaan kembali benda-benda, desain, dan fungsi.
- Mengurangi risiko pengembangan, terutama dalam integrasi sistem yang kompleks.

Kelebihan Pemrograman Berorientasi Objek

- Salah satu kelebihan pemrograman berorientasi objek adalah penggunaan ulang kode-kode yang telah dibuat. Pewarisan adalah salah satu cara untuk menggunakan kode-kode yang telah dibuat sebelumnya.
- Karena sebuah metode hanya bisa mengakses dimana metode itu berada, pemrograman berorientasi objek mengurangi kesalahan yang terjadi.
- Dalam pemrograman berorientasi objek waktu yang diperlukan untuk merancang lebih lama dibandingkan dengan pembuatan program. Hal ini disebabkan oleh fokus menentukan kelas dan objek yang memerlukan perhatian lebih besar. Tetapi waktu yang diperlukan untuk menulis program lebih sedikit dibandingkan dengan program prosedural.
- Program berorientasi objek juga lebih mudah dipelihara serta dikembangkan.

Model berorientasi objek

- Sebuah model objek menangkap struktur statis dari sistem dengan menggambarkan objek dalam sistem, hubungan antara objek, serta atribut dan operasi yang merupakan karakteristik setiap kelas dan objek.
- Model Berorientasi objek lebih mendekati keadaan nyata, dan dilengkapi dengan penyajian grafis dari sistem yang sangat bermanfaat untuk komunikasi dengan user dan pembuatan dokumentasi struktur dari sistem.

Keunggulan Pendekatan berorientasi objek

Pendekatan berorientasi objek mempunyai keunggulan sebagai berikut :

- Pendekatan objek menuntun penggunaan ulang (*reuse*) komponen – komponen program sebelumnya. Penggunaan kembali menuntun pengembangan perangkat lunak yang lebih cepat dan berkualitas lebih tinggi.
- Perangkat lunak yang dikembangkan dengan berorientasi objek mempermudah pemeliharaannya karena strukturnya secara inheren sudah decouple.
- Sistem berorientasi objek lebih mudah diadaptasi dan diskala menjadi sistem lebih besar karena sistem – sistem lebih besar dibuat dengan merakit subsistem – subsistem yang dapat diguna ulang.

Definisi Perancangan Berorientasi Objek

- Perancangan berorientasi objek adalah suatu teknik atau cara pendekatan baru dalam melihat permasalahan dan sistem (sistem perangkat lunak, Sistem informasi, atau sistem lainnya).
- Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Pengertian “berorientasi objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya.

KELEBIHAN PERANCANGAN BERORIENTASI OBJEK

- Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalam pembangunan sistem.
- Dibandingkan dengan SSAD, waktu pengembangan, level organisasi, ketangguhan, dan penggunaan kembali (reuse) kode program lebih tinggi dibandingkan dengan metode OOAD (Sommerville, 2000).
- Tidak ada pemisahan antara fase desain dan analisis, sehingga meningkatkan komunikasi antara user dan developer dari awal hingga akhir pembangunan sistem.
- Analis dan programmer tidak dibatasi dengan batasan implementasi sistem, jadi desain dapat diformulasikan yang dapat dikonfirmasi dengan berbagai lingkungan eksekusi.

KELEBIHAN PERANCANGAN BERORIENTASI OBJEK

- Relasi obyek dengan entitas (thing) umumnya dapat di mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam sistem. Hal ini memudahkan dalam mehami desain (Sommerville, 2000).
- Memungkinkan adanya perubahan dan kepercayaan diri yang tinggi terhadap kebenaran software yang membantu untuk mengurangi resiko pada pembangunan sistem yang kompleks (Booch, 2007).
- Encapsliation data dan method, memungkinkan penggunaan kembali pada proyek lain, hal ini akan memperingan proses desain, pemrograman dan reduksi harga.

KELEBIHAN PERANCANGAN BERORIENTASI OBJEK

- OOAD memungkinkan adanya standarisasi obyek yang akan memudahkan memahami desain dan mengurangi resiko pelaksanaan proyek.
- Dekomposisi obyek, memungkinkan seorang analis untuk memecah masalah menjadi pecahan-pecahan masalah dan bagian-bagian yang dimanage secara terpisah. Kode program dapat dikerjakan bersama-sama. Metode ini memungkinkan pembangunan software dengan cepat, sehingga dapat segera masuk ke pasaran dan kompetitif. Sistem yang dihasilkan sangat fleksibel dan mudah dalam memelihara.

Perbedaan OOAD dengan SSAD

- Pada awal desain OOAD, sistem mungkin akan sangat simple.
- Pada OOAD lebih fockus pada coding dibandingkan dengan SSAD.
- Pada OOAD tidak menekankan pada kinerja team seperti pada SSAD.
- Pada OOAD tidak mudah untuk mendefinisikan class dan obyek yang dibutuhkan sistem.
- Sering kali pemrogramam berorientasi obyek digunakan untuk melakukan anlisis terhadap fungsional sistem, sementara metode OOAD tidak berbasis pada fungsional sistem.

Perbedaan OOAD dengan SSAD

- 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 (Hantos, 2005).
- Metodologi pengembangan sistem dengan OOAD menggunakan konsep reuse. Reuse merupakan salah satu keuntungan utama yang menjadi alasan digunakannya OOAD. Namun demikian, tanpa prosedur yang emplisit terhadap reuse, akan sangat sliit untuk menerapkan konsep ini pada skala besar (Hantos, 2005).

1. Pendahuluan

- Reliability

Software perlu mempunyai aspek realibility, yaitu aspek :

- Flexibility → kemampuan sistem menangani suatu transaksi atau event/kejadian yang tidak biasa, tidak terduga
- Resilence → Kemampuan sistem beradaptasi ketika dilakukannya maintenance. Dalam artian ketika maintenance dilakukan dan terdapat perubahan, maka tidak terjadi/timbul masalah baru.
- Quality → Membangun sistem/software dengan baik, benar, tepat waktu, tidak crash, mengerjakan task dengan benar.

1. Pendahuluan

Metode pemrograman terstruktur dirasa kurang memadai lagi

Maka :

Perlu ada metode baru yang lebih sesuai dalam pengembangan sistem/perangkat lunak

1. Pendahuluan

Costs dan Benefits Objek

- Benefit Object
 - System Stability
 - Maintainability
 - Reusable software components
 - Reality-based systems
 - Data accessibility
 - User involvement and ownership

Benefits Objek

System Stability

Resilience to change → sebuah program atau sistem informasi setelah diinstal dan running, sesuai dengan perjalanan waktu dapat mengalami maintenance atau modifikasi sesuai kebutuhan user. Nah ketika proses tersebut dilakukan, sistem dikatakan resilience to change jika modifikasi tersebut tidak menimbulkan masalah baru pada sistem yang telah dibangun, dengan waktu yang singkat dan biaya yang sedikit.

Resilience dan stability ini dapat kita dapatkan dari objek. karena sistem benar-benar dirancang untuk mendukung bisnis user yang berdasarkan pemahaman dasar akan kebutuhan data pada bisnis user daripada hanya sekedar pemenuhan laporan atau query ad hoc

Benefits Objek

Maintainability

Metode sebelum objek cenderung dibuat berdasarkan kebutuhan laporan dan kebutuhan sekarang, sehingga ketika terjadi maintenance menjadi lebih sukar.

Metode berorientasi objek menghasilkan sistem yang lebih siap untuk proses maintenance dan peningkatan kualitas.

Benefits Objek

Reusable software components

Hasil analisa rekayasa perangkat lunak dan kode program dapat digunakan ulang. Reusable software components ini dapat dilakukan oleh adanya feature inheritance dan polimorphism. Contohnya pengembangan library untuk object classes pada JavaBean yang dapat digunakan ulang.

Benefits Objek

Reality-based systems

Memberikan gambaran yang lebih akurat terhadap operasi bisnis user dan kebutuhan informasinya. Sehingga nantinya sistem yang telah jadi akan lebih cocok, sesuai dengan kebutuhan aktual user untuk menjalankan bisnisnya.