Analisis dan Perancangan Berorientasi Objek

**Oleh :** Sintya Sukarta, ST.,MT

(Digunakan di lingkungan sendiri, sebagai buku ajar

mata kuliah Analisis dan Perancangan Berorientasi Objek)

******

**Fakultas Teknik dan Ilmu Komputer**

**Program Studi Manajemen Informatika**

**Universitas Komputer Indonesia**

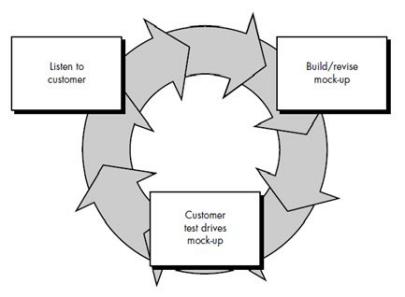
# Pertemuan 4

## Paradigma pengembangan system

pengembangan **[sistem informasi](http://www.warsidi.com/2016/02/definisi-sistem-informasi-tren-teknologi-informasi-terkini.html" \o "definisi sistem informasi)** yang paling tua adalah siklus hidup sistem (*systems life cycle*), yang mengharuskan tahap-tahap formal dalam pengembangan **[sistem informasi](http://www.warsidi.com/2016/02/definisi-sistem-informasi-tren-teknologi-informasi-terkini.html" \o "definisi sistem informasi)**. Tahap-tahap pengembangan sistem harus dilaksanakan secara urut dengan output-output yang telah ditentukan; setiap tahap mengharuskan persetujuan formal sebelum tahap selanjutnya dimulai. Metode siklus hidup sistem berguna untuk proyek-proyek besar yang memerlukan ketentuan dan spesifikasi formal dan pengendalian manajemen yang ketat pada setiap tahap pengembangan **[sistem informasi](http://www.warsidi.com/2016/02/definisi-sistem-informasi-tren-teknologi-informasi-terkini.html" \o "definisi sistem informasi)**, tetapi metode siklus hidup sistem ini sangat bertele-tele dan mahal.

### Paradigma prototype

Metode Prototype merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau waterfall development model.

[](https://murtri.files.wordpress.com/2014/08/3.jpg)

Dalam Model Prototype, prototype dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.

Perubahan dan presentasi prototype dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk dari perangkat lunak yang akan dikembangkan.

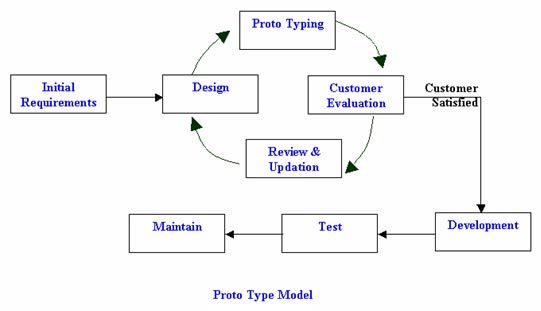
Teknik – teknik Prototyping Meliputi :

* Perancangan Model
* Perancangan Dialog
* Simulasi

Berikut adalah 4 langkah yang menjadi karakteristik dalam proses pengembangan pada metode prototype, yaitu :

* Pemilihan fungsi
* Penyusunan Sistem Informasi
* Evaluasi
* Penggunaan Selanjutnya

Metode ini menyajikan gambaran yang lengkap dari suatu sistem perangkat lunak, terdiri atas model kertas, model kerja dan program. Pihak pengembang akan melakukan identifikasi kebutuhan pemakai, menganalisa sistem dan melakukan studi kelayakan serta studi terhadap kebutuhan pemakai, meliputi model interface, teknik prosedural dan teknologi yang akan dimanfaatkan.

[](https://murtri.files.wordpress.com/2014/08/4.jpg)

Berikut adalah Tahapan – tahapan Proses Pengembangan dalam Model Prototype, yaitu :

* **Pengumpulan kebutuhan**

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.

* **Membangun prototyping**

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).

* **Evaluasi protoptyping**

Evaluasi ini dilakukan oleh pelanggan, apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan atau belum. Jika sudah sesuai, maka langkah selanjutnya akan diambil. Namun jika tidak, prototyping direvisi dengan mengulang langkah-langkah sebelumnya.

* **Mengkodekan sistem**

Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

* **Menguji sistem**

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, kemudian dilakukan proses Pengujian. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur, dll.

* **Evaluasi Sistem**

Pelanggan mengevaluasi apakah perangkat lunak yang sudah jadi sudah sesuai dengan yang diharapkan . Jika ya, maka proses akan dilanjutkan ke tahap selanjutnya, namun jika perangkat lunak yang sudah jadi tidak/belum sesuai dengan apa yang diharapkan, maka tahapan sebelumnya akan diulang.

* **Menggunakan sistem**

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Model Prototyping ini sangat sesuai diterapkan untuk kondisi yang beresiko tinggi di mana masalah-masalah tidak terstruktur dengan baik, terdapat fluktuasi kebutuhan pemakai yang berubah dari waktu ke waktu atau yang tidak terduga, bila interaksi dengan pemakai menjadi syarat mutlak dan waktu yang tersedia sangat terbatas sehingga butuh penyelesaian yang segera. Model ini juga dapat berjalan dengan maksimal pada situasi di mana sistem yang diharapkan adalah yang inovatif dan mutakhir sementara tahap penggunaan sistemnya relatif singkat.

Berikut merupakan Jenis – jenis dari Prototyping :

* **Feasibility prototyping**

digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk system informasi yang akan disusun.

* **Requirement prototyping**

digunakan untuk mengetahui kebutuhan aktivitas bisnis user.

* **Desain Prototyping**

digunakan untuk mendorong perancangan sistem informasi yang akan digunakan.

* **Implementation prototyping**

merupakan lanjutan dari rancangan prototype, prototype ini langsung disusun sebagai suatu sistem informasi yang akan digunakan.

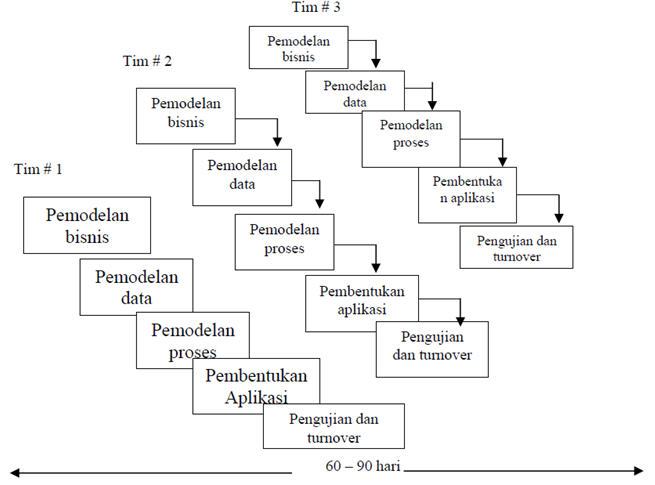
* **Contoh Penerapan Metode Prototype.**

Sebuah rumah sakit ingin membuat aplikasi sistem database untuk pendataan pasiennya. Seorang atau sekelompok programmer akan melakukan identifikasi mengenai apa saja yang dibutuhkan oleh pelanggan, dan bagaimana model kerja program tersebut. Kemudian dilakukan rancangan program yang diujikan kepada pelanggan. Hasil/penilaian dari pelanggan dievaluasi, dan analisis kebutuhan pemakai kembali di lakukan.

* **Kelebihan Model Prototype :**
* Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan.
* Penentuan kebutuhan lebih mudah diwujudkan.
* Mempersingkat waktu pengembangan produk perangkat lunak.
* Adanya komunikasi yang baik antara pengembang dan pelanggan.
* Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
* Lebih menghemat waktu dalam pengembangan sistem.
* Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya.
* **Kekurangan Model Prototype :**
* Proses analisis dan perancangan terlalu singkat.
* Biasanya kurang fleksibel dalam mengahadapi perubahan.
* Walaupun pemakai melihat berbagai perbaikan dari setiap versi prototype, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.
* Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien.

### Paradigma RAD (Rapid Application *Development*)

Rapid Aplication Development (RAD) adalah sebuah model proses perkembanganperangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

[](https://murtri.files.wordpress.com/2014/08/5.jpg)

Berikut adalah Tahapan – tahapan Proses Pengembangan dalam Model Rapid Application Development (RAD), yaitu :

* **Bussiness Modeling**

Fase ini untuk mencari aliran informasi yang dapat menjawab pertanyaan berikut:

* Informasi apa yang menegndalikan proses bisnis?
* Informasi apa yang dimunculkan?
* Di mana informasi digunakan ?
* Siapa yang memprosenya ?
* **Data Modeling**

Aliran informasi yang didefinisikan sebagai bagian dari fase bussiness modeling disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik (atribut) masing-masing objek diidentifikasi dan hubungan antar objek-objek tersebut didefinisikan.

* **Proses Modeling**

Aliran informasi yang didefinisikan di dalam fase data modeling ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.

* **Aplication Generation**

Selain menggunakan bahasa pemrograman generasi ketiga, RAD juga memakai komponen program yang telah ada atau menciptakan komponen yang bisa dipakai lagi. Ala-alat bantu bisa dipakai untuk memfasilitasi konstruksi perangkat lunak.

* **Testing dan Turnover**

Karena proses RAD menekankan pada pemakaian kembali, banyak komponen program telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus diuji dan semua interface harus dilatih secara penuh.

* **Kelebihan Model RAD :**
* Lebih efektif dari Pengembangan Model waterfall/sequential linear dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan.
* Cocok untuk proyek yang memerlukan waktu yang singkat.
* Model RAD mengikuti tahap pengembangan sistem seperti pada umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien.
* **Kekurangan Model RAD :**
* Model RAD menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas rapid-fire yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek RAD akan gagal.
* Tidak semua aplikasi sesuai untuk RAD, bila system tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah.
* RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.
* Membutuhkan Tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar.
* Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan.

### Paradigma extreme programming

metode rekayasa perangkat lunak pendekatan berorientasi objek model *Extreme Programming* (XP). Paradigma pembangunan mencakup seperangkat aturan dan praktik yang terjadi dalam konteks kerangka empat kegiatan yaitu: perencanaan, desain, coding, dan pengujian. Keempat aktivitas inilah yang akan menghasilkan sebuah perangkat lunakyang didasari dengan konsep model *Extreme Programming*. (Pressman 2010).

1.      Perencanaan (*Planning*)

Dalam tahap ini dikumpulkan kebutuhan awal *user* atau dalam XP disebut *user stories.* Hal ini dibutuhkan agar pengembang mengerti bisnis konten, kebutuhan *output* sistem, dan fitur utama dari *software* yang dikembangkan. Tahapan ini untuk menganalisa kebutuhan dari sistem tersebut untuk dapat digunakan sesuai dengan *user requirement* atau *user stories.*

2.      Desain (*Design*)

Desain dari sistem pada penelitian ini digambarkan dengan model UML berupa *use case diagram, activity diagram,* dan *relation table.* Pembuatan desain pada XP tetap mengedepankan prinsip *Keep it Simple (KIS).* Desain disini merupakan representasi dari sistem guna mempermudah pengembang dalam membangun sistem*. Desain* ini dimaksudkan untuk mempermudah pengembangan sistem nantinya.

3.      *Coding*

Proses melakukan *coding system* (pengkodean perangkat lunak) oleh *Programmer/Software Engineer* sesuai dengan *planning* dan *design* yang telah dibuat sebelumnya.

*4.      Testing*

Tahap ini akan menggunakan *unit test* yang sebelumnya telah dibuat. Karena pembuatan dari unit test adalah pendekatan utama dari XP. Dalam melakukan pengujian, penulis menggunakan 2 teknik pengujian yaitu pengujian *white box.* Pada tahap pengujian *black box*, dilakukan pengujian setiap unit test, maksudnya melakukan pengujian integrasi antara input dan hasil output yang sesuai semestinya terjadi.

### Paradigma Inkremental

Model incremental menggabungkan elemen-elemen model sekuensial linier (diimplementasikan secara berulang) dengan filosofi prototype interatif. Model ini memakai urutan-urutan linier di dalam model yang membingungkan, seiring dengan laju waktu kalender. Setiap urutan linier menghasilkan pertambahan perangkat lunak yang kemudian dapat disampaikan kepada pengguna.

Pada saat model incremental (pertambahan) ini digunakan, pertambahan pertama sering merupakan produk inti (*core product)*, yaitu sebuah model pertambahan yang dipergunakan, tetapi beberapa muka tambahan (beberapa diketahui dan beberapa tidak) tetap tidak disampaikan. Produk inti tersebut dipergunakan oleh pelanggan (atau mengalami pengkajian detail). Sebagai hasil dari pemakaian dan/atau evaluasi maka dikembangkan rencana bagi pertambahan selanjutnya. Rencana tersebut menekankan modifikasi produk inti untuk secara lebih baik memenuhi kebutuhan para pelanggan dan penyampaian fitur serta fungsional tambahan. Proses ini mengikuti penyampaian setiap pertambahan sampai bisa menghasilkan produk yang lengkap.

Model proses incremental tersebut, seperti model prototype dan pendekatan-pendekatan evolusioner yang lain, bersifat iterative. Tetapi tidak seperti model prototype, model pertambahan berfokus pada penyampaian produk operasional dalam setiap pertambahannya. Pertambahan awal ada di versi *stripped down* dari produk akhir, tetapi memberikan kemampuan untuk melayani pemakai dan juga menyediakan platform untuk evaluasi oleh pemakai.

Perkembangan pertambahan, khususnya berguna pada saat *staffing*, tidak bisa dilakukan dengan menggunakan implementasi lengkap oleh batasan waktu bisnis yang sudah disepakati untuk proyek tersebut. Jika produk inti diterima dengan baik, maka staf tambahan (bila dibutuhkan) bisa ditambahkan untuk mengimplementasi pertambahan selanjutnya. Sebagai tambahan, system mayor yang sedang pada masa perkembangan serta waktu penyampaiannya belum pasti, mungkin membutuhkan keberadaan perangkat keras yang baru. Bisa juga rencana tertentu dibuat untuk menghindari pemakaian perangkat lunak ini, sehingga memungkinkan fungsionalitas partial disampaikan kepada pemakai tanpa harus banyak tertunda.

### Penggunaan Agility

Kelincahan (*Agility*) dalam kinerja bisnis adalah kemampuan perusahaan untuk sejahtera dalam pasar global yang berubah cepat dan terus terfragmen unutk produk dan jasa berkualitas tinggi, berkinerja baik, dan disesuaikan dengan pelanggan. Perusahaan yang lincah dapat membuat laba dalam pasar dengan pilihan produk yang luas dan bermasa hidup pendek, dan dapat memproduksi pesanan secara individual dan dengan jumlah yang besar. Perusahaan tersebut mendukung penyesuaian massal (*mass customization*) dengan menawarkan produk individual sambil mempertahankan produksi dalam volume yang tinggi. Perusahaan yang lincah sangat bergantung pada tegnologi Internet untuk memadukan dan mengelola proses bisnis, sambil menyediakan daya pemrosesan informasi untuk melayani banyak pelanggan sebagai individual.

Ada empat strategi dasar yang harus diimplementasikan untu menjadi perusahaan yang lincah. Antara lain :

1. Pelanggan dari perusahaan yang lincah menganggap produk atau jasa sebagai solusi terhadap masalah individual mereka. Jadi, harga produk dapat ditentukan berdasarkan biaya produksinya.
2. Perusahaan yang lincah bekerja sama dengan pelanggan, pemasok dan perusahaan lain bahkan dengan pesaing. Hal ini memungkinkan perusahaan untuk memasarkan prosuk dengan cepat dan hemat, dimanapun sumber daya berada dan siapapun yang memlikinya.
3. Perusahaan yang lincah dapat bertahan ketika terjadi perubahan dan ketidakpastian. Perusahaan menggunakan struktur organisasi yang fleksibel sehingga sesuai dengan peluang pelanggan yang terus berubah dan berbeda-beda.

Akhirnya perusahaan yang lincah dapat meningkatkan dampak sumberdaya manusia dan pengetahuan yang mereka miliki. Dengan memelihara semangat wira usaha, perusahaan yang lincah dapat memberikan insentif yang tinggi bagi tanggung jawab, kemampuan beradaptasi, dan inovasi pegawai.

Cara lain untuk memikirkan mengenai kelincahan dalam bisnis. Kerangka kerja ini menekankan pada peran yang dapat dimainkan oleh pelanggan, mitra bisnis dan tegnologi informasi dalam mengembangkan dan mempertahankan kelincahan strategis perusahaan. Perhatikan bagaimana tegnologi informasi dapat memungkinkan perusahaan untuk mengembangkan hubungan dengan pelanggan dalam komunitas virtual yang membantu perusahaan untuk bermitra dengan pemasok, distributor, manufaktur kontrak dan pihak lainnya melalui portal kerja sama dan sistem rantai pasokan berbasis Web lainnya yang secara signifikan memperbaiki kelincahan perusahaan dalam melihat peluang bisnis yang inovatif.

**DAFTAR PUSTAKA**

*Kenneth H. Rossen, Discrete Mathematics and Its Application, 4th edition, 1999, Mc-Graw Hill International*

*Bernard Kolman dan Robert C. Busby, Discrete Mathematical Structures For Computer Science, 2nd edition, 1987, Prentice Hall*

*C. L. Liu, Elements of Discrete Mathematics, 1992, Gramedia Pustaka Utama*

*Rinaldi Munir, Matematika Diskrit, Edisi 2, 2003, Penerbit Informatika Bandung*

*Jong Jek Siang, Matematika Diskrit Dan Aplikasinya Pada Ilmu Komputer, 2002, Penerbit Andi Yogyakarta*