

# PENGEMBANGAN SISTEM



Setelah mempelajari bab ini, anda diharapkan:

- Mengenal pendekatan sistem sebagai kerangka kerja dasar pemecahan segala jenis permasalahan.
- Mengetahui bagaimana cara menerapkan pendekatan sistem untuk memecahkan masalah-masalah sistem.

- Memahami bahwa siklus hidup pengembangan sistem (*system development life cycle* – SDLC) merupakan sebuah metodologi yang direkomendasikan untuk mengembangkan sistem.
- Mengenal pendekatan-pendekatan SDLC yang utama: siklus air terjun tradisional, prototyping, rapid application development, pengembangan berfase dan desain ulang proses bisnis.



- Mengetahui dasar-dasar proses pemodelan dengan diagram arus data (*data flow diagram*) dan kasus-kasus penggunaan (*use case*).
- Memahami bagaimana proyek-proyek pengembangan sistem dikelola dengan cara dari atas ke bawah.
- Mengenal proses-proses dasar mengestimasi biaya proyek.

# PENDEKATAN SISTEM

Pencarian asal muasal proses pemecahan masalah secara sistematis mengarah pada Jhon Dewey, seorang profesor ilmu filosofi di Columbia University tahun 1910.

Rangkaian pertimbangan yang terlibat dalam pemecahan sebuah kontroversi secara memadai :

- Mengenal kontrovesi
- Mempertimbangkan klaim-klaim alternative
- Membentuk satu pertimbangan

# JENIS-JEINIS PROTOTIPE

## **Prototipe evolusioner (*evolutionary prototype*)**

Terus-menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. Jadi, satu prototipe akan menjadi sistem aktual.

## **Prototipe persyaratan (*requirements prototype*)**

Dikembangkan sebagai satu cara untuk mendefinisikan persyaratan-persyaratan fungsional dari sistem baru ketika pengguna tidak mampu menggunakan dengan jelas apa yang mereka inginkan.

# PROTOTYPING

**Prototype** adalah satu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. Proses prototype ini disebut **prototyping**.



# PENGEMBANGAN PROTOTIPE EVOLUSIONER

Ada 3 Langkah Dalam Pembuatan Suatu Prototipe Evolusiner :

## 1. Mendefinisikan Kebutuhan Pengguna

Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.

## 2. Membuat Satu Prototipe

Pengembang mempergunakan satu alat prototyping atau lebih untuk membuat prototipe.



# PENGEMBANGAN PROTOTIPE EVOLUSIONER

## **3. Menentukan Apakah Prototype Dapat Diterima**

Pengembang mendemonstrasikan prototype kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan.

# PENGEMBANGAN PROTOTIPE PERSYARATAN

## Langkah-Langkah Yang Terlibat Dalam Pembuatan Sebuah Tipe Prototipe Persyaratan

### ✓ **Membuat kode sistem yang baru**

Pengembang menggunakan prototype sebagai dasar untuk pengkodean sistem baru.

### ✓ **Menguji sistem baru**

Pengembang menguji sistem

# PENGEMBANGAN PROTOTIPE PERSYARATAN

- ✓ **Menentukan apakah sistem yang baru dapat diterima**  
Pengguna memberitahukan kepada pengembang apakah sistem dapat diterima.
- ✓ **Membuat sistem baru menjadi sistem produksi**

# DAYA TARIK PROTOTYPING

1. Membaiknya komunikasi antara pengembang dan pengguna.
2. Pengembang dapat melakukan pekerjaan yang lebih baik dalam menentukan kebutuhan pengguna.
3. Pengguna memainkan peranan yang lebih aktif dalam pengembangan sistem.
4. Pengembang dan pengguna menghabiskan waktu dan usaha yang lebih sedikit dalam pengembangan sistem.
5. Implementasi menjadi jauh lebih mudah.



# POTENSI KESULITAN DARI PROTOTYPING

1. Terburu-buru dalam menyerahkan prototype dapat menyebabkan diambilnya jalan pintas dalam definisi masalah, evaluasi alternative dan dokumentasi. Dalam jalan pintas ini akan menciptakan usaha-usaha yang “cepat dan kotor”.
2. Pengguna dapat terlalu gembira dengan prototype yang diberikan yang mengarah pada ekspektasi yang tidak realitis sehubungan dengan sistem produksi nantinya.

# POTENSI KESULITAN DARI PROTOTYPING

3. Prototype evolusioner bisa jadi tidak terlalu efisien.
4. Antarmuka komputer manusia yang diberikan oleh beberapa alat prototyping tertentu kemungkinan tidak mencerminkan teknik-teknik desain yang baik

# PENGEMBANGAN APLIKASI LEBIH CEPAT

Satu metodologi yang memiliki tujuan yang sama dengan prototyping yaitu memberikan respons yang cepat atas kebutuhan pengguna namun dengan lingkup yang lebih luas adalah ***Rapid Application Development (RAD)*** atau ***pengembangan aplikasi cepat*** yang diperkenalkan oleh konsultan komputer dan penulis James Martin.

# PENGEMBANGAN APLIKASI LEBIH CEPAT

RAD adalah kumpulan strategi, metodologi dan alat terintegrasi yang terdapat di dalam suatu kerangka kerja yang disebut rekayasa informasi.

**Rekayasa informasi (*information engineering-IE*)** adalah nama yang diberikan Martin kepada keseluruhan pendekatan pengembangan sistemnya yang ia perlakukan sebagai suatu aktivitas perusahaan secara menyeluruh.



# UNSUR-UNSUR PENTING RAD

- 1. Manajemen** : Khususnya manajemen puncak. Hendaknya menjadi penguji coba (experimenter) yang suka melakukan hal-hal dengan cara baru atau pengadaptasi awal (early adapter).
- 2. Orang** : Daripada hanya memanfaatkan satu tim untuk melakukan seluruh aktivitas SDLC, RAD menyadari adanya efisiensi yang dapat dicapai melalui penggunaan tim-tim khusus.

- 3. Metodologi** : Metodologi dasar RAD adalah siklus hidup RAD.
- 4. Alat-alat** : Alat-alat RAD terutama terdiri atas bahasa-bahasa generasi keempat dan alat-alat rekayasa peranti lunak dengan bantuan komputer (computer-aided software engineering-CASE) yang memfasilitas prototyping dan penciptaan kode.

# PENGEMBANGAN BERFASE

Satu metodologi pengembangan sistem yang dewasa ini digunakan oleh banyak perusahaan adalah kombinasi dari SDLC tradisional, prototyping, dan RAD dengan mengambil fitur-fitur yang terbaik dari masing-masing metodologi.

# TAHAP-TAHAP PENGEMBANGAN BERFASE

## Investigasi Awal

Menganalisis dengan tujuan untuk mempelajari tentang organisasi dengan masalah sistemnya: mendefinisikan tujuan, hambatan , risiko, dan ruang lingkup sistem baru.



# Pemodelan Proses

Pemodelan proses pertama kali dilakukan dengan menggunakan diagram alur (*flowchart*). Diagram ini mengilustrasikan aliran data melalui sistem dan program.

International Organization for Standardization (ISO) menciptakan standar untuk bentuk-bentuk simbol flowchart, memastikan penggunaannya di seluruh dunia.

Ketika diagram arus data dengan empat simbolnya muncul pada akhir tahun 1980-an, minat akan penerapannya pun muncul dengan seketika.

Istilah **terminator** sering kali di pergunakan untuk menyatakan unsur-unsur lingkungan, karena menunjukkan titik-titik dimana sistem berakhir.

Suatu terminator dapat berupa:

- Orang, seperti seorang manajer, yang menerima laporan dari sistem
- Organisasi, seperti departemen lain dalam perusahaan atau perusahaan lain.
- Sistem lain yang memiliki antar muka dengan sistem.

# Proses

Proses adalah sesuatu yang mengubah input menjadi output. Proses dapat digambarkan dengan sebuah lingkaran, sebuah persegi panjang horizontal, atau sebuah persegi panjang tegak bersudut melingkar. Masing-masing simbol proses diidentifikasi dengan sebuah label.

# Arus Data

Arus data terdiri atas sekumpulan unsur-unsur data yang berhubungan secara logis (mulai dari satu unsur data tunggal hingga satu file atau lebih) yang bergerak dari satu titik atau proses ke titik atau proses yang lain.

Simbol panah digunakan untuk menggambarkan arus ini dan dapat digambar dengan menggunakan garis lurus maupun melingkar.



# Penyimpanan Data

Ketika kita perlu menyimpan data karena suatu alasan tertentu, maka kita akan menggunakan penyimpanan data. Dalam terminologi DFD, **penyimpanan data** adalah suatu gudang data.

DFD berikut mengilustrasikan sebuah sistem yang dapat dipergunakan oleh perusahaan untuk menghitung komisi bagi para agen penjualnya. Di sini, terminator digambarkan dengan kotak, proses dengan kotak tegak bersudut tumpul, arus data dengan garis lurus, dan penyimpanan data dengan kotak berujung terbuka.

Pelanggan





Sistem primer adalah sebuah program komputer dan sistem sekunder adalah orang yang berinteraksi dengan program komputer. Dialog biasanya terdiri atas tindakan-tindakan yang diambil oleh para partisipan, seperti yang dilakukan oleh seorang operator entri data dan sistem komputer.



## Kapan Menggunakan Diagram Arus Data dan Kasus Penggunaan

Diagram arus data dan kasus penggunaan sering kali dibuat selama tahap-tahap investigasi awal dan analisis dari metodologi pengembangan berfase. DFD mengilustrasikan suatu tinjauan atas pemrosesan, dan kasus penggunaan memberikan detailnya. Biasanya dibutuhkan beberapa kasus penggunaan untuk mendukung satu diagram angka 0.

## Steering committee SIM menjalankan tiga fungsi utama:

- **Menciptakan Kebijakan** yang memastikan dukungan komputer untuk mencapai sasaran strategis perusahaan.
- **Melakukan Pengendalian Fiskal** dengan bertindak sebagai yang berwenang dalam memberikan persetujuan untuk seluruh permintaan akan pendanaan yang berhubungan dengan komputer.
- **Menyelesaikan Perselisihan** yang terjadi sehubungan dengan prioritas penggunaan komputer.

## Kepemimpinan Proyek

Steering committee SIM jarang ikut terlibat langsung dengan detail pekerjaan. Tanggung jawab jatuh ke tangan tim proyek. **Tim proyek** meliputi semua orang yang ikut berpartisipasi dalam pengembangan sistem informasi. Satu tim dapat memiliki anggota hingga 12 orang yang terdiri atas gabungan beberapa orang pengguna, spesialis informasi, dan auditor internal. Auditor akan memastikan bahwa desain sistem telah memenuhi beberapa persyaratan tertentu dilihat dari segi keakuratan, pengendalian, keamanan, dan auditabilitas.



# Input Pengestimasian Biaya

Sebuah *work breakdown structure* (WBS) mengidentifikasi aktivitas-aktivitas proyek yang akan membutuhkan sumber daya. Kebutuhan sumber daya (*resource requirement*) mencantumkan sumber daya tertentu yang akan dibutuhkan dan berapa jumlahnya.

- Tarif sumber daya (*resource rates*) adalah biaya per-unit untuk setiap jenis sumber daya.
- Estimasi durasi aktivitas (*activity duration estimates*) menyebutkan periode pekerjaan yang dibutuhkan untuk menyelesaikan aktivitas.
- Informasi historis (*historical information*) terdiri atas file-file dari data proyek masa lalu, basis data pengestimasian biaya komersial, dan pengetahuan tim proyek.



## Alat-alat dan Teknik Estimasi Biaya

Estimasi analogis (**analogous estimating**) menggunakan biaya actual proyek-proyek serupa yang telah dilakukan di masa lalu sebagai dasar untuk memproyeksikan biaya dari proyek yang sedang dipertimbangkan. Teknik ini digunakan ketika hanya terdapat sedikit informasi lain yang tersedia.

Estimasi dari bawah ke atas (**bottom-up estimating**) dimulai dengan detail, seperti aktivitas di dalam grafik Gantt, lalu mengalikannya dengan data biaya, seperti tarif per-jam untuk karyawan, untuk menghasilkan estimasi biaya proyek.

Alat-alat terkomputerisasi (**computerized tools**) dapat digunakan secara terpisah atau untuk menyederhanakan alat-alat yang baru saja diuraikan. Satu sumber bagi alat-alat terkomputerisasi adalah [WWW.CONSTRUX.COM](http://WWW.CONSTRUX.COM).

# Komponen-komponen Proses Estimasi Biaya

Input	Alat Dan Teknik	Output
Work breakdown structure	Estimasi analogis	Estimasi biaya
Kebutuhan sumber daya	Estimasi dari bawah ke atas	Detail-detail Pendukung
Tarif sumber daya Estimasi durasi aktivitas Informasi historis	Alat-alat terkomputerisasi	Rencana manajemen Biaya

# Simpulan

- Pendekatan sistem terdiri dari 3 fase :persiapan,definisi,dan solusi.
- Pendekatan SDLC terdiri dari lima tahap:
- Perencanaan,analisis,desain,dan implementasi.
- Data input digunakan tidak hanya dalam bentuk estimasi melainkan juga detail-detail pendukung seperti bagaimana asumsi dilakukan,asumsi-asumsi,dan varians biaya akan dikelola setelah proyek di jalankan.

**Terima Kasih**