

SYSTEM DEVELOPMENT LIFE CYCLE DENGAN BEBERAPA PENDEKATAN

I. Joko Dewanto

djoko.dewanto@indonusa.ac.id

Abstrak

It's better for the development project information system adapt System Development Life Cycle (SDLC) methodology in designing or establishing a system. This methodology is good to be implemented whether in a large or small case system. Many different and the requirements. The use of life cycle will make the work developer much more easier so that they will be able to produce a good system or software. The seven models that we are going discussed have their own advantages and disadvantages, therefore they are created with the hope that they will cover each other weakness.

Kata Kunci : SDLC, model pendekatan

PENDAHULUAN

Konsep pengembangan sistem informasi ternyata menjadi konsentration pembicaraan para kalangan analist di dalam menyikapi rancangan sistem informasi. Banyak model diterapkan tetapi model yang paling tradisional adalah SDLC, dengan membicarakan beberapa proses pengembangan sistem pada SDLC.

PEMBAHASAN

Pengertian dan Definisi System Development Life Cycle.

Pada awalnya pengembangan software (perangkat lunak) hanyalah sebuah pekerjaan membuat kode-kode program oleh *programmer* untuk memecahkan masalah, dan umumnya masih bersifat parsial. Saat ini system sudah sangat besar dan kompleks sehingga para perancang, analis, programmer juga user harus bekerja sama untuk membuat berjuta-juta baris kode program yang dapat menghasilkan solusi terbaik dan memiliki nilai Unggul. Untuk melakukan pekerjaan tersebut, diciptakan beberapa aplikasi pendekatan sistem untuk pengembangan

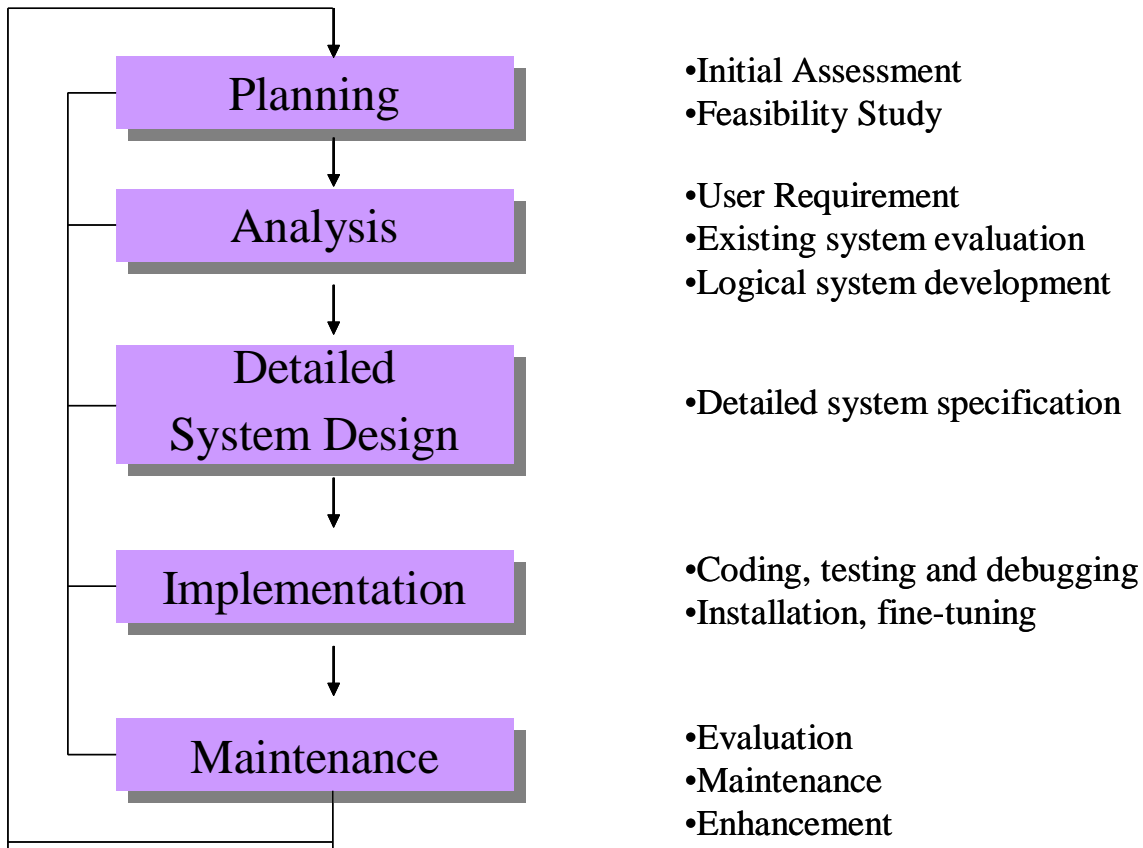
sistem informasi berbasis computer kemudian dikenal dengan System Development Life Cycle (SDLC).

SDLC merupakan suatu urutan dari beberapa proses secara bertahap didalam merancang dan mengembangkan sistem yang dikenal juga dengan nama Information System Development atau juga Application Development.

Lalu, kenapa didalam pengembangan sistem saat ini mengenal prinsip siklus (cycle) ? Hal ini disebabkan pada saat ini system akan selalu memerlukan perubahan , seperti perkembangan zaman, teknologi, pelanggan dan lainnya, sehingga siklus atau putaran tahapan pengembangan system akan dimulai dari awal lagi.

Tahapan siklus hidup sistem secara umum

Pada saat ini perkembangan proyek sistem informasi dilihat dari segi kompleksitas dan kapasitasnya, saat ini banyak methodology atau pendekatan yang dikembangkan dari tahapan siklus secara umum. Sebagai gambaran dapat dilihat pada bagan dibawah ini :



Gambar 1
Tahapan Pengembangan Sstem

a. *Perencanaan*

Tahapan ini biasanya dilaksanakan untuk memberikan jawaban dari pertanyaan apakah proyek tersebut layak untuk dikerjakan dari sisi teknis, operasional dan juga biaya. Untuk itu biasanya dikerjakan beberapa aktifitas yang harus dikerjakan, sebagai berikut :

- Mengenal dan memastikan masalah
- Menentukan objektif
- Mengidentifikasi ruang lingkup sistem
- Mengatur dan menjalankan studi kelayakan system

- Menyiapkan proposal system yang akan dikerjakan
- Membangun mekanisme pengawasan

b. *Analisa*

Tahapan analisa mencoba menganalisa secara makro analisa

- Menganalisa kebutuhan informasi dari end-user, lingkungan perusahaan dan juga jika ada sistem yang sedang digunakan
- Mengorganisir tim proyek
- Menentukan jenis studi sistem yang akan digunakan

- Menentukan kriteria performa sistem
- Menentukan kriteria keamanan sistem
- Menentukan system *backup* dan *recovery*
- Menentukan aturan integritas data
- Menyiapkan proposal rancangan sistem

sebenarnya dan dilakukan pemeliharaan agar system selalu mampu menyesuaikan diri dengan perubahan-perubahan atau pengembangan dalam lingkungan. Aktifitas yang dikerjakan adalah :

- Menggunakan sistem baru
- Evaluasi sistem baru
- Memelihara sistem seperti memperbaiki kesalahan-kesalahan, menjaga sistem terupdate, juga mengembangkan sistem

c. *Perancangan*

- Identifikasi sistem alternative
- Evaluasi konfigurasi sistem alternative
- Menyiapkan bagan organisasi
- Menyiapkan diagram data
- Menyiapkan kamus data
- Menyiapkan bentuk formulir tercetak
- Menyiapkan diagram struktur, menyiapkan model objek rasional
- Menyiapkan spesifikasi kelas

Pendekatan siklus hidup sistem

Waterfall Model

Model Waterfall adalah merupakan model yang paling populer diantara model pendekatan yang lain didalam SDLC dan bekerja secara linier dan berurutan di lain pihak model ini merupakan model yang paling awal terbentuk. Pendekatan ini mengikuti suatu paradigma yang dikenal dengan *documentation driver paradigm*, harapannya adalah bahwa disetiap proses pengembangan sistem, setiap langkah yang dilakukan akan dikomentasi atau ditulis dalam bentuk spesifikasi dalam laporan yang detail.

Pada model ini memiliki beberapa kelemahan antara lain :

- Spesifikasi dalam perancangan sistem harus sudah pasti dan adanya model sistem yang berubah-ubah menjadi tidak stabil.
- Pekerjaan merancang dan pembuatan kode sering tidak konsisten dengan spesifikasi, misalnya adanya komponen sistem yang hilang dan kebutuhan pengembangan yang tidak diharapkan
- Performa system tidak dapat diuji sampai dengan proses coding benar-benar selesai dilakukan

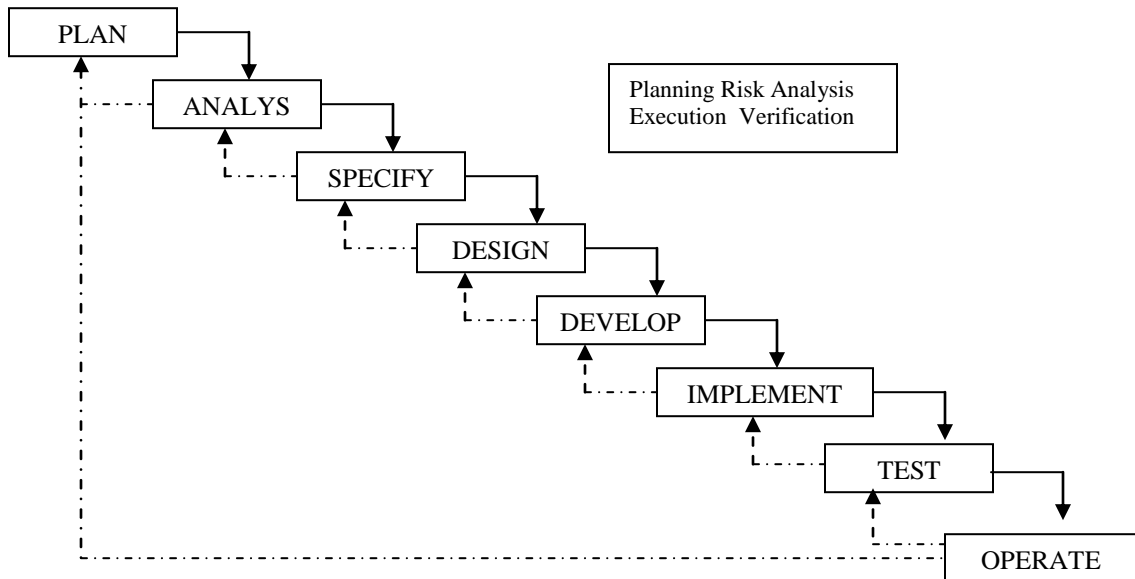
d. *Implementasi*

- Rancangan diterjemahkan ke dalam kode program
- Memenuhi sumber daya perangkat keras
- Menyiapkan sumber daya perangkat lunak
- Menyiapkan database
- Menyiapkan fasilitas fisik
- Melatih user
- Menyiapkan prososal implementasi

e. *Perawatan*

Penggunaan adalah tahap terakhir dalam SDLC. Disini sistem baru akan digunakan dalam lingkungan

Pada gambar 1 terlihat jelas bahwa proses tahapan akan berjalan secara berurutan. Namun antar masing-masing tahap bias dilakukan *review* sehingga jika ditemukan kesalahan memungkinkan untuk diperbaiki dahulu :

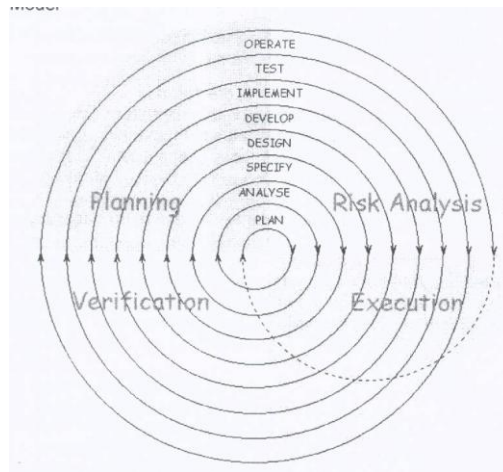


Gambar 2 Model *Waterfall*

Spiral Model

Pendekatan Model Spiral menyajikan pembuktian konsep dari tahapan pengembangan system yang selalu berputar. Pendekatan ini lebih akurat menggambarkan bagaimana tahapan-tahapan tersebut sebenarnya dilakukan dengan sangat tidak berurutan, Jika lebih diperhatikan model spiral merupakan

serangkaian putaran dari model waterfall dalam lingkup yang kecil dimana masing-masing tahap menghasilkan prototype awal yang memperesentasikan sebagian dari keseluruhan system. Model spiral juga memperhatikan kebutuhan untuk kembali ke awal dan memungkinkan pengulangan ke tahap awal sampai beberapa kali sejalan dengan perkembangan system



Gambar 3 Spiral Model

Keuntungan menggunakan pendekatan model spiral, yaitu :

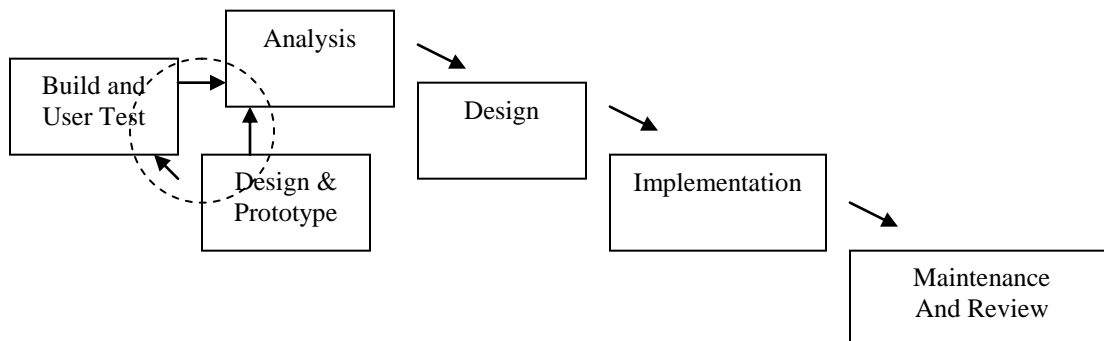
- Berfokus pada *reusable software*
- Memungkinkan evolusi siklus hidup, pertumbuhannya dan perubahan spesifikasi
- Berfokus kepada pendektasian awal dari adanya kesalahan
- Dapat digunakan untuk pengembangan system hardware dan software

skala yang lebih kecil, sedangkan model rapid prototype adalah prototype yang dapat diselesaikan lebih cepat.

Dalam model rapid prototype ini hal yang ditekankan adalah pembuatan protipe yang bentuk dan fungsinya serupa dengan produk yang diinginkan sehingga dapat melakukan pengujian terhadap kegunaannya. Prototipe adalah bagian yang paling penting dalam penentuan spesifikasi, dan sangat mungkin dibuat dengan menggunakan peralatan yang berbeda dengan peralatan yang digunakan untuk membuat produk aslinya. Biasanya prototype yang dibuat sejenia *Throwaway*, dan pada saat protoripe telah disetujui

Rapid Application Development (RAD) Model

Prototype merupakan model dari sesuatu system yang nyata dengan

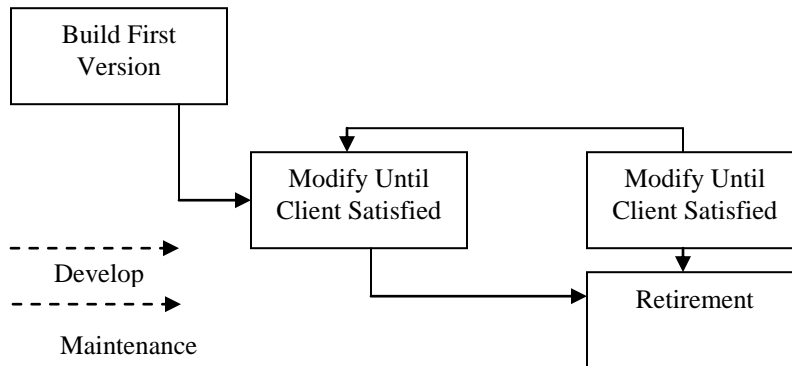


Gambar 4. Rapid Prototype

Build and Fix Model

Model *Build and Fix* merupakan metode yang paling kasar karena sistem atau produk yang akan dibuat tidak memiliki spesifikasi atau ide rancangan.

Biasanya pendekatan ini akan digunakan pada pengembangan sistem dengan skala yang kecil, yang tidak memerlukan beribu-ribu baris kode program

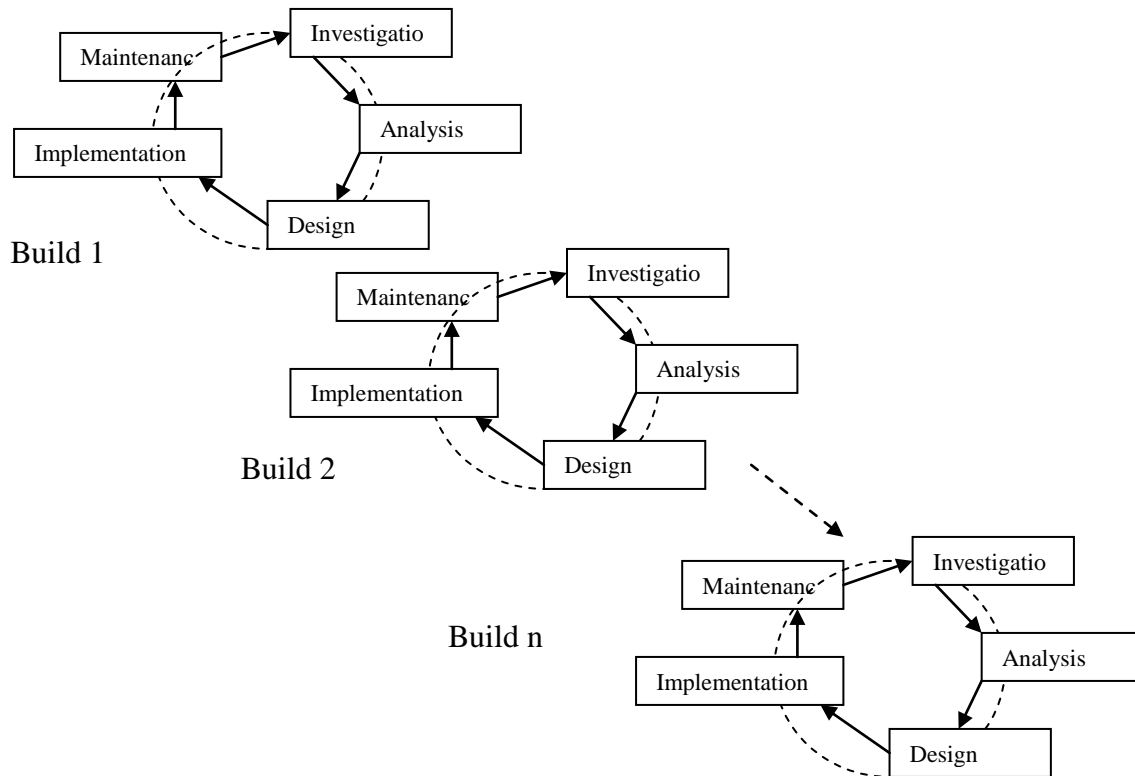


Gambar 5. Model *Build and Fix*

Joint Application Development (JAD) Model

Melalui model JAD, biasanya setiap software dikembangkan melalui serangkaian pembangunan yang selalu bertambah (increment), dan kadang sistem ini disebut sebagai sistem yang incremental model. Kelebihan dengan menggunakan model ini adalah bahwa

turunan dari model JAD adalah model build and fix. Selain itu kelebihan lainnya kesalahan dapat segera ditemukan di dalam spesifikasi user karena dalam tiap tahap pengembangannya feedback dari user selalu segera dilakukan perbaikan dan langsung diuji.



Gambar 6 Joint Application Development

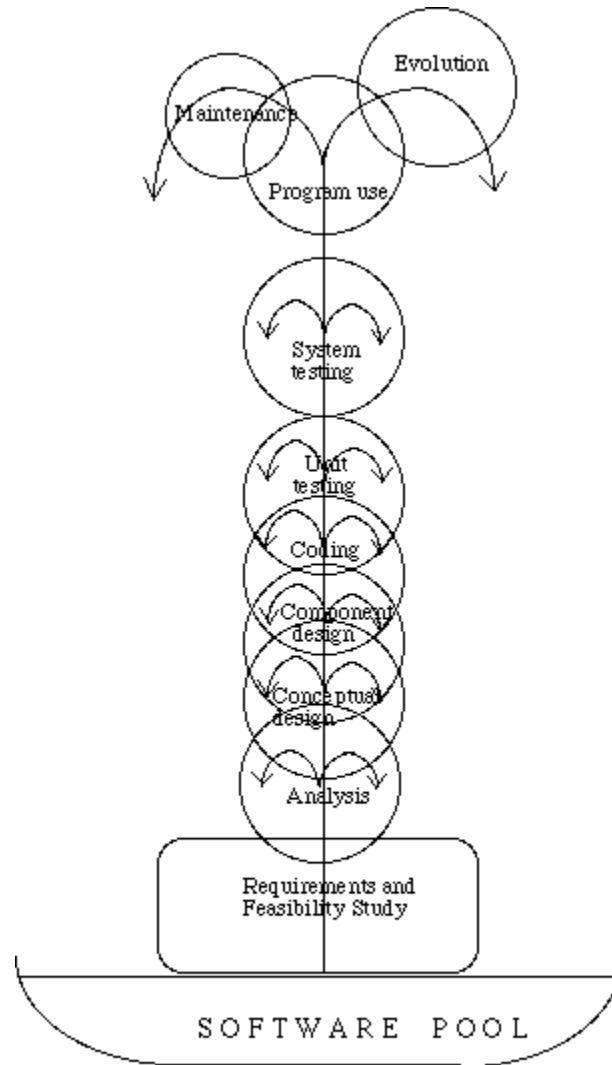
Synchronize and Stabilize Model

Model Synchrononous and Stablize merupakan jenis dari model incremental yang juga menggabungkan kelebihan dari model spiral dengan teknologi mengamati dan mengatur coding. Pendekatan ini memungkinkan beberapa tim bekerja secara paralel. Pada saat penyelesaian program biasanya setiap developer melakukan testing dalam berbagai metode dengan menstabilkan pengcodingan. Pada penyelesaian program ada beberapa versi alpha (pengujian di dalam pihak internal), betha (fitur yang sudah mulai lengkap) untuk kandidat biasanya mendekati versi final dan disebut *god master* untuk sistem manufaktur.

Fountain Model

Pendekatan dengan model fountain memungkinkan pekerjaan yang saling tumpang tindih dalam siklus pengembangan sistem tentunya dengan melihat pertimbangan yang aada dan disesuaikan dengan kebutuhan pengembangan. Beberapa tahapan dan pengulangan yang terjadi tersebut akan berjalan secara paralel. Model fountain ini merupakan siklus hidup berorientasi objek sehingga akan lebih fokus pada bagian-bagian keseluruhan sistem yang dikenal dengan nama subsistem.

Model ini dikenal dengan model air mancur dimana arus air akan naik keatas melalui tengah dan kemudian turun kembali. Lefel terjadinya pengembangan sistem secara berulang dengan level yang lebih tinggi



Real-World System

©B. Henderson-Sellers and J.M. Edwards, 1993

Gambar 7 Fountain Model

KESIMPULAN

Penerapan *System Development Life Cycle* oleh para developer dengan menggunakan berbagai pilihan model atau pendekatan harus memperhatikan berbagai macam pertimbangan baik dari segi sistemnya itu sendiri, user dan juga lingkungan. Dilihat dari bentuk model dengan perkembangan sistem informasi memungkinkan perkembangan model lain

yang sesuai dengan proyek implementasi pengembangan sistem yang sedang dijalankan. Namun semua model tetap selalu menggunakan model atau pendekatan yang terstruktur.

DAFTAR PUSTAKA

Jawadekar WS., *Management Information Systems*, second edition. New Delhi :

- Tata McGraw-Hill Publishing
Company Limited. 2002
<http://www.trinity.sa.edu.au/infosys/sdlc/intro.html>
- McLeod, Raymond., *Management Information System*, eight edition. New Jersey Prentice Hall. 2001
<http://www.ibun.edu.tr/comp2002/sysdec.doc>
- Royce WW. *Managing the Development of Large Software System*. San Fransisco : In Proc. WESTCON. 1970
<http://gates.comm.virginia.edu/rm2n/teaching/sdlc.htm>
<http://www.swt.edu/effective/upps/upps-04-02-03.html>