

METODE PENGEMBANGAN SISTEM

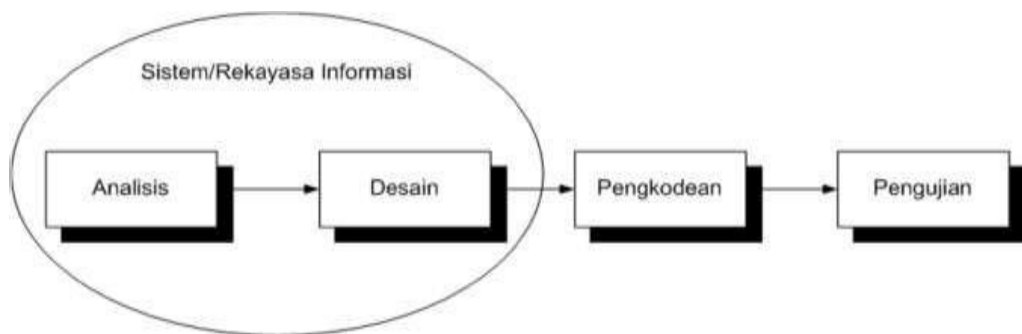
A. PENGERTIAN SDLC atau Software Development Life Cycle atau System Development Life Cycle

SDLC atau Software Development Life Cycle atau System Development Life Cycle adalah proses mengembangkan atau mengubah suatu system perangkat lunak dengan menggunakan model-model atau metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (Rosa dan Shalahuddin, 2015).

Beberapa Model Proses :

1. Model Air Terjun (Water Fall)

Menurut Rosa A.S dan M. Shalahuddin (2015:28-30) Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



Berikut langkah-langkah yang penting dalam model ini sebagai berikut:

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosuder pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program *computer* sesuai desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian focus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini



dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Kelebihan Waterfall :

- a. Keuntungan pengembangan dengan metode waterfall adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase satu per satu, sehingga meminimalisir kesalahan-kesalahan yang mungkin akan terjadi. Pengembangannya bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, troubleshooting, dan berakhir di operasi dan pemeliharaan.
- b. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
- c. Document pengembangan system sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase

berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

- d. Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

Kerugian Waterfall :

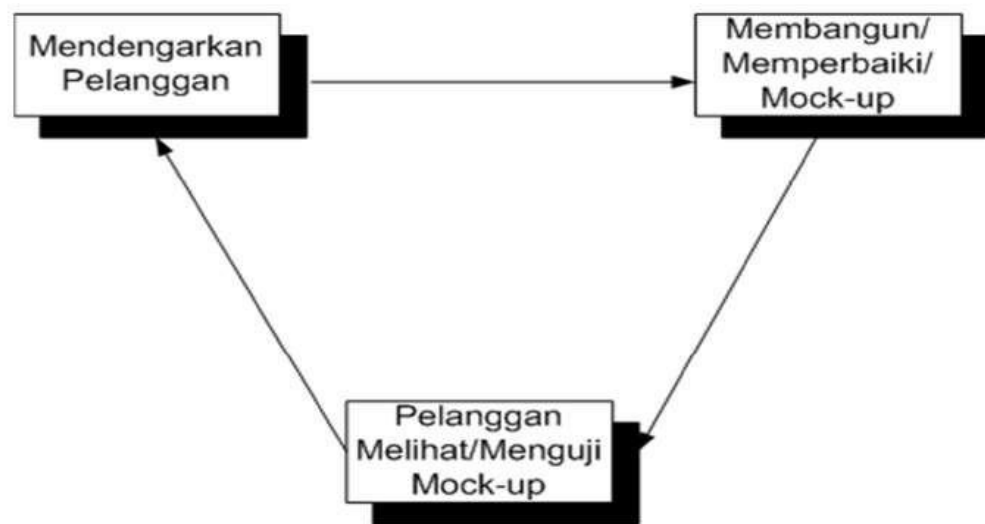
- a. Kerugian pembangunan menggunakan metode waterfall adalah tidak memungkinkan untuk banyak refleksi atau revisi jika terjadi kesalahan. Karna setelah aplikasi ini dalam tahap pengujian, sangat sulit untuk kembali dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep.
- b. Diperlukan majemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- c. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
- d. Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidak pastian pada saat awal pengembangan.
- e. Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
- f. Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.



2. Model Prototipe

Digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Model prototipe dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat, dan protoptipe merupakan program yang belum jadi.

Gambar Model Prototipe

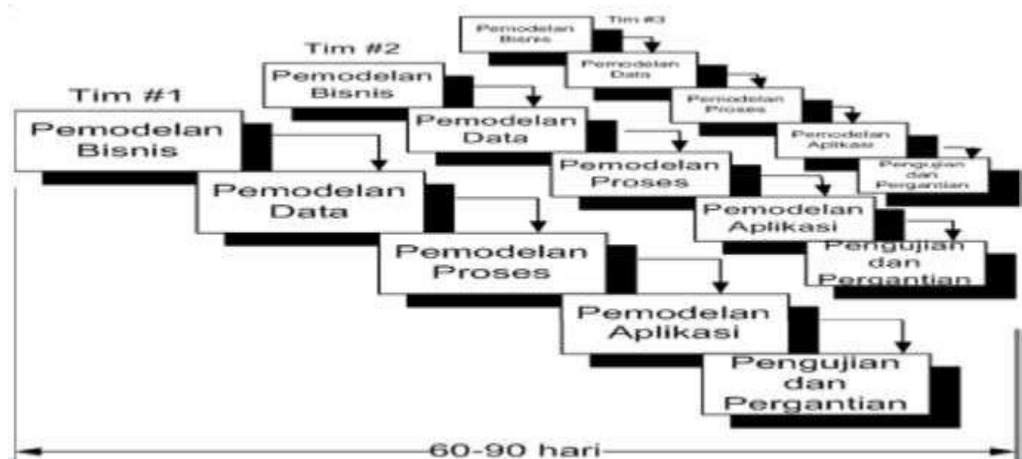


Mock-up adalah sesuatu yang digunakan sebagai model desain untuk mengajar, demonstrasi, evaluasi desain, promosi atau keperluan lain yang mampu menyediakan atau mendemonstrasikan sebagian besar fungsi perangkat lunak dan memungkinkan pengujian desain perangkat lunak.

3. Model Rapid Application Development (RAD)

Model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model RAD merupakan

adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak



Gambar Rapid Application Development (RAD)

a. Pemodelan Sistem

Untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait proses bisnis, informasi apa saja yang harus dibuat, siapa yang harus membuat informasi tersebut, bagaimana alur informasi tersebut, proses apa saja yang terkait informasi tersebut.

b. Pemodelan Data

Memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atribut beserta relasinya dengan data yang lain.

c. Pemodelan Proses

Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data.

d. Pembuatan Aplikasi

Implementasi proses dan data menjadi program.

e. Pengujian dan Pergantian

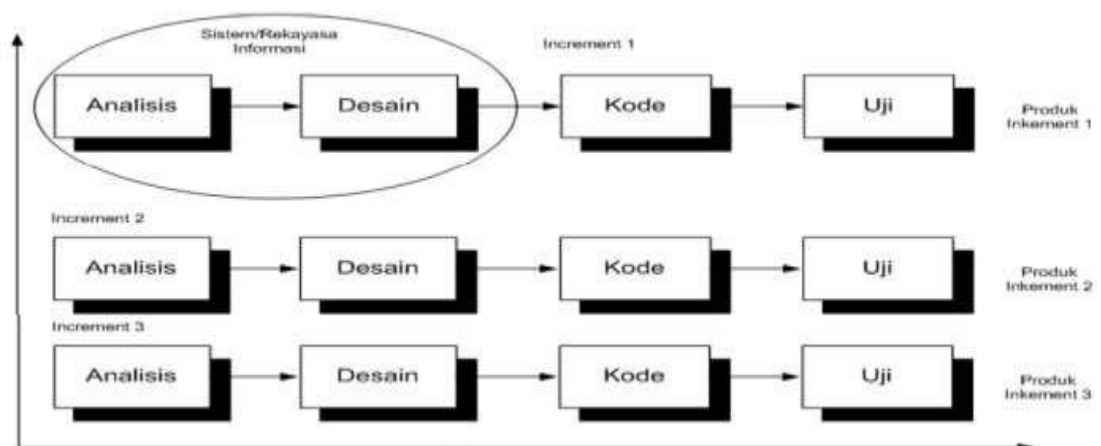
Menguji komponen yang sudah dibuat.

4. Model Iteratif

Mengkombinasikan proses-proses pada model air terjun dan iteratif pada model prototipe yang menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya.

Model ini cocok untuk pengembang dengan turnover staf yang tinggi. Model Inkremental dibuat untuk mengatasi kelemahan model waterfall yang tidak mengakomodasi iterasi dan mengatasi kelemahan dari metode prototipe yang memiliki proses terlalu pendek.

Model Iteratif

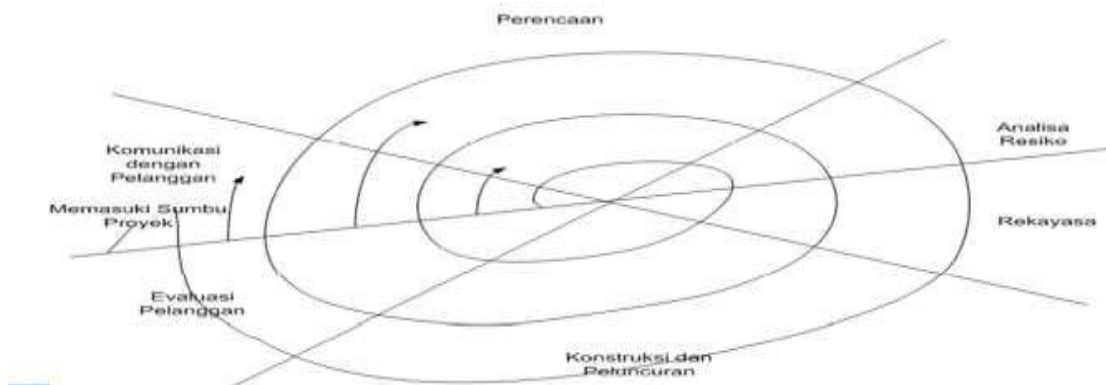


5. Model Spiral

Memasangkan iteratif pada model prototipe dengan kontrol dan aspek sistematis yang diambil dari model air terjun yang menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya.

Model Spiral dibagi menjadi beberapa kerangka aktivitas atau disebut juga wilayah kerja (task region). Cocok digunakan untuk mengembangkan aplikasi dengan skala besar tetapi target waktu dan biaya tidak terlalu tinggi.

Gambar Model Spiral



- Komunikasi dengan Pelanggan (customer Communication), Untuk membangun komunikasi yang efektif antara pengembang (developer) dan pelanggan (customer)
- Perencanaan (Planning), Untuk mendefinisikan sumber daya, waktu, dan informasi yang terkait dengan proyek
- Analisa Resiko (Risk Analysis), Diperlukan untuk memperkirakan resiko dari segi teknis maupun manajemen.
- Rekayasa, Diperlukan untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak (dapat juga berupa prototipe).
- Konstruksi dan Peluncuran (Construction and release), Dibutuhkan untuk mengonstruksi, menguji, melakukan instalasi, dan menyediakan dukungan terhadap user.

- f. Evaluasi Pelanggan (customer Evaluation), Untuk mendapatkan umpan balik berdasarkan evaluasi representasi perangkat lunak yang dihasilkan dari proses rekayasa dan diimplementasikan pada tahap instalasi.

