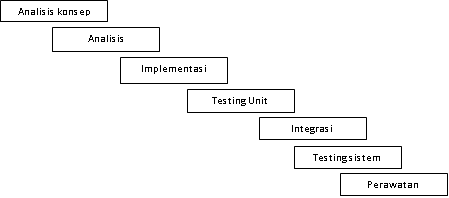
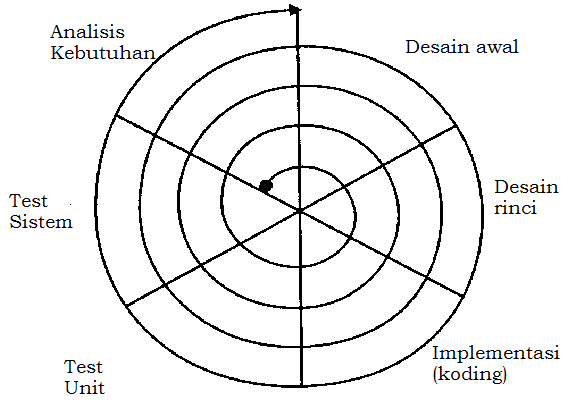
* *Dokumen adalah hal yang sangat penting dalam sebuah pelaksanaan proyek*
* *Dokumen dapat memberikan informasi yang sangat bermanfaat untuk dapat berkomunikasi baik dengan klien ataupun dengan insinyur/teknisi lain ketika akan dilakukan pengembangan lebih lanjut dari perangkat lunak tersebut*

1. *Waterfall*

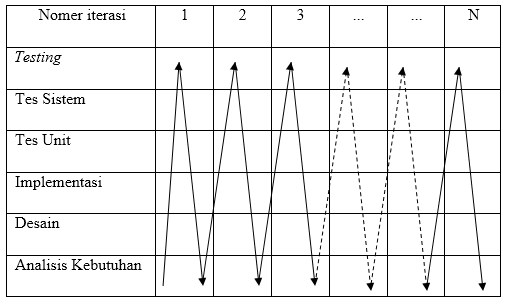
Model ini termasuk jenis model yang klasik di mana prosesnya dilakukan secara berurutan dimulai dari: analisis kebutuhan; desain; implementasi; proses integrasi; dan *testing.*



1. *Spiral*
   * Pada model ini proses siklus pengembangan dilakukan lebih dari satu kali siklus sesuai dengan kebutuhan dan analisis resiko;
   * Alasan lain dari model ini juga agar dapat menghasilkan bagian produk lebih awal sehingga dapat ditunjukkan ke klien dan mendapatkan *feedback* tentang bagian produk tersebut apakah sudah sesuai atau belum.



1. `*Incremental*
   1. Model ini hampir mirip dengan model *Spiral* di atas di mana proses pertama adalah menghasilkan produk dasar, kemudian diteruskan ke proses selanjutnya menjadi produk yang lebih baik atau sempurna;
   2. Perbedaannya dengan model *Spiral* adalah, pada model *Spiral* putaran produk ke-N+1 adalah merupakan produk hasil perbaikan dari produk ke-N yang belum sesuai dengan kebutuhan klien. Sedangkan pada model *Incremental* putaran produk ke-N+1 adalah menunjukan peningkatan atau perkembangan dari produk ke-N.



**Mengapa perlu standarisasi...???**

* Ketika sistem pendokumenannya tersebut tidak sesuai dengan standar maka tentunya ini akan membuat proses komunikasi eksternal yang tidak baik
* Sehingga dengan demikian sistem pendokumenan harus mengacu atau diterapkan pada standar pendokumenan yang ada

**Penjelasan – penjelasan**

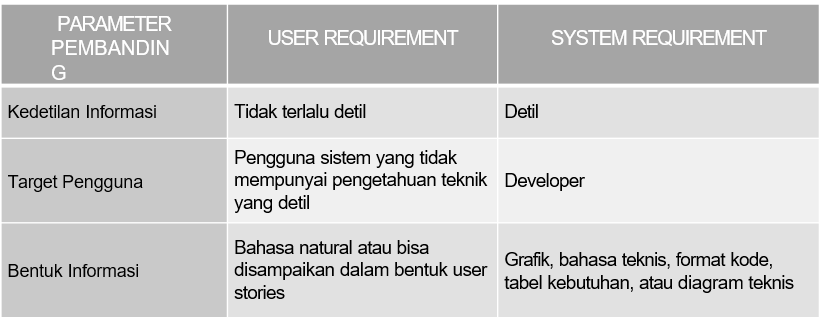
1. Apa itu rekayasa kebutuhan?

Proses pembentukan layanan-layanan (fungsional) yang dibutuhkan oleh customer dari sebuah perangkat lunak dan batasan-batasan di mana sistem tersebut beroperasi dan dikembangkan.

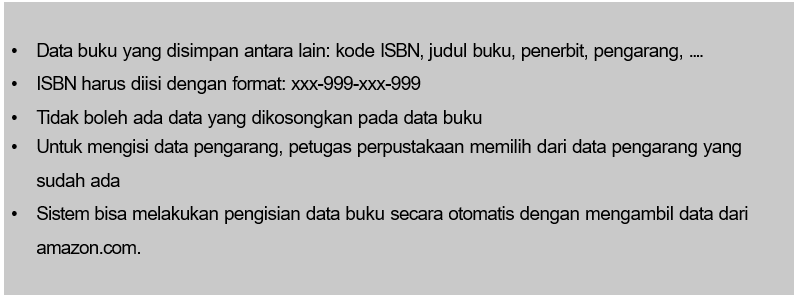
1. Kebutuhan pada software dimulai dari domain masalah yang dikenal sebagai user requirement

Apa yang user harapkan untuk bisa dilakukan pada suatu perangkat lunak, biasanya dituliskan dalam bentuk User Requirement Document

1. Dari User Requirement, kebutuhan akan didetailkan ke dalam bentuk solusi domain System Requirement



1. System Requirement Spesification/Requirement Spesification



1. Dokumen
   * + SDD adalah dokumen yang menjelaskan arsitektur dan desain secara rinci dari sebuah aplikasi. Biasanya, ditampilkan dalam bentuk diagram seperti model obyek dan diagram alur.
     + SPMP Dokumen ini adalah dokumen yang digunakan pada tahap perencanaan awal pengerjaan proyek PL.
     + SRS Dokumen ini adalah dokumen yang digunakan pada tahap pengumpulan/analisis kebutuhan-kebutuhan.

* *Data Flow Diagram*

Digunakan untuk menggambarkan aliran data dari sebuah proses/sistem

* Context diagram adalah DFD ruang lingkup dari sistem yang menunjukkan batas-batas sistem, external entitiy yang berinteraksi dengan sistem dan aliran data utama antara external entity dengan sistem.

Context diagram menggambarkan keseluruhan sistem.

|  |
| --- |
| * UML adalah standar dunia yang dibuat   oleh *Object Management Group (OMG),*  sebuah badan yang bertugas  mengeluarkan standar-standar teknologi  *objectoriented* dan *software component*   * UML adalah sebuah bahasa standard   untuk pengembangan sebuah software  yang dapat menyampaikan bagaimana  membuat dan membentuk model-model |

**Kebutuhan Fungsional**

* Yang dimaksud dengan kebutuhan fungsional adalah kebutuhan-kebutuhan yang harus ada pada aplikasi
* Contoh: “sebuah aplikasi harus dapat menghitung nilai IPK”; dan lain-lain
* Sedangkan “aplikasi harus dapat menghitung IPK paling lambat 1 menit”, itu bukan merupakan kebutuhan fungsional

**Kebutuhan Non-Fungsional**

* Yang termasuk kebutuhan non-fungsional adalah:
  + Performa
  + Kehandalan dan Ketersediaan
  + Penanganan kesalahan (*error handling*)
  + Kebutuhan Antarmuka
  + Batasan-batasan sistem

