* PENGERTIAN PERANGKAT LUNAK

Perangkat lunak umumnya digunakan untuk mengontrol perangkat keras yang biasa disebut sebagai device driver, melakukan proses penghitungan, berinteraksi dengan perangkat lunak yang lebih mendasar lainnya, seperti sistem operasi dan bahasa pemrograman.

* Pemodelan Perangkat Lunak

Adalah Disiplin ilmu untuk mempelajari bentuk-bentuk pemodelan perangkat lunak yang digunakan sebagai bagian dari tahapan pengembangan perangkat lunak secara terstruktur dan berorientasi objek. Pemodelan dalam suatu rekayasa perangkat lunak merupakan suatu hal yang dilakukan di tahapan awal.

* Menurut Grady Booch, James Rumbaugh dan Ivar Jacobson Prinsip dari Pemodelan adalah:

1. Memilih model apa yang di gunakan, bagaimana masalahnya dan bagaimana juga dengan solusinya.
2. Setiap Model dapat dinyatakan dalam tingkatan yang berbeda
3. Model yang terbaik adalah yang berhubungan dengan realitas.
4. Tidak pernah ada model tunggal yang cukup baik, setiap system yang baik memilik serangkaian model kecil yang independen.

* REKAYASA PERANGKAT LUNAK

**Rekayasa Perangkat Lunak (RPL)** Adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi kebutuhan pangguna, desain, pengkodean, pengujian, sampai pemeliharaan sistem setelah digunakan.

Pada rekayasa perangkat lunak, banyak model yang telah dikembangkan untuk membantu proses pengembangan perangkat lunak. Model-model ini pada umumnya mengacu pada model proses pengembangan sistem yang disebut System Development Life Cycle (SDLC) seperti terlihat pada gambar dibawah ini :

**Aktivitas Waterfall Model**

* Requirements analysis and definition : mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang hasrus dipenuhi oleh program yang akan dibangun.
* System and software design : desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
* Implementation and unit testing : desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji.
* Integration and system testing : penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing)
* Operation and maintenance : mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

## **Tahapan-Tahapan Prototyping dan Kelebihannya**

Tahapan-tahapan dalam Prototyping adalah sebagai berikut:

1. **Pengumpulan kebutuhan**  
   Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. **Membangun prototyping**  
   Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).
3. **Evaluasi protoptyping**  
   Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototyping direvisi dengan mengulang langkah 1, 2 , dan 3.
4. **Mengkodekan sistem**  
   Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. **Menguji sistem**  
   Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.
6. **Evaluasi Sistem**  
   Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.
7. **Menggunakan sistem**  
   Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

## D. Model Rapid Aplication Development (RAD)

**Rapid Aplication Development (RAD)** adalah sebuah model proses perkembangan software sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan kontruksi berbasis komponen. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60 sampai 90 hari). Karena dipakai terutama pada aplikasi sistem konstruksi, pendekatan RAD melingkupi fase – fase sebagai berikut :

#### 1. Bussiness modeling

Aliran informasi di antara fungsi – fungsi bisnis dimodelkan dengan suatu cara untuk menjawab pertanyaan – pertanyaan berikut : informasi apa yang mengendalikan proses bisnis? Informasi apa yang di munculkan? Siapa yang memunculkanya? Ke mana informasi itu pergi? Siapa yang memprosesnya?

#### 2. Data modeling

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

#### 3. Prosess 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.

#### 4. Aplication generation

RAD mengasumsikan pemakaian teknik generasi ke empat. Selain menciptakan perangkat lunak dengan menggunakan bahasa pemrograman generasi ketiga yang konvensional, RAD lebih banyak memproses kerja untuk memkai lagi komponen program yang ada ( pada saat memungkinkan) atau menciptakan komponen yang bisa dipakai lagi (bila perlu). Pada semua kasus, alat – alat bantu otomatis dipakai untuk memfasilitasi konstruksi perangkat lunak.

#### 5. Testing and turnover

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

**Model Spiral / Model Boehm**

Tahap-tahap model ini dapat dijelaskan secara ringkas sebagai berikut :

* **Tahap Liason:**pada tahap ini dibangun komunikasi yang baik dengan calon pengguna/pemakai.
* **Tahap Planning (perencanaan):**pada tahap ini ditentukan sumber-sumber informasi, batas waktu dan informasi-informasi yang dapat menjelaskan proyek.
* **Tahap Analisis Resiko:**mendefinisikan resiko, menentukan apa saja yang menjadi resiko baik teknis maupun manajemen.
* **Tahap Rekayasa (engineering):**pembuatan prototipe.
* **Tahap Konstruksi dan Pelepasan (release):**pada tahap ini dilakukan pembangunan perangkat lunak yang dimaksud, diuji, diinstal dan diberikan sokongan-sokongan tambahan untuk keberhasilan proyek.
* **Tahap Evaluasi:**Pelanggan/pemakai/pengguna biasanya memberikan masukan berdasarkan hasil yang didapat dari tahap engineering dan instalasi.