SOFTWARE MAINTENANCE

I. PENDAHULUAN

Sebuah perangkat lunak yang disimpan atau digunakan dalam jangka waktu yang cukup lama akan menimbulkan masalah apabila sering diabaikan atau tidak diperhatikan perkembangannya. Permasalahan yang akan timbul dari penggunaan perangkat lunak yang cukup lama adalah :

- Adanya kebutuhan baru. Sebagai contoh pada saat ini berkembangnya perangkat lunak berbasis Web.
- Untuk mengurangi kompleksitas, biaya , waktu pemasaran. Sebagai contoh pemanfaatan dari bahasa perkembangan bahsa pemrograman tingkat tinggi.
- Mengurangi cacat. Sebagai contoh dengan cara membuat standarisasi dari pengkodean yang tidak sesuai.

Untuk dapat menyelesaikan permasalahan yang timbul diatas, maka diperlukannya melakukan evolusi dari perangkat lunak tersebut.

Evolusi dari perangkat lunak itu sendiri meliputi :

1. Pembangunan Perangkat Lunak

   Yang termasuk dalam pembangunan perangkat lunak itu sendiri meliputi dari daur hidup perangkat lunak, yaitu : permintaan (*requirement*), spesifikasi, perancangan (*design*), testing dan sebagainya.
2. Perawatan Perangkat Lunak

   Merupakan proses-proses untuk memperpanjang waktu penggunaan sistem perangkat lunak yang ada, sehingga tetap dapat dipergunakan sebagaimana mestinya sesuai dengan baik.
3. Migrasi Perangkat Lunak

   Merupakan proses-proses yang memindahkan sistem yang ada ke sistem yang baru dikarenakan perkembangan dari kebutuhan perangkat lunak tersebut.

Untuk dapat menjaga kualitas dari perangkat lunak tersebut dalam melakukan evolusi perangkat lunak, maka dibuatlah *"Laws"* dari perangkat

lunak tersebut. Adapun *"Laws"* dari perangkat lunak tersebut menurut **Lehman** adalah :

1. Perubahan yang terus menerus secara kontinyu.
   Program yang berada pada lingkungan dunia nyata harus mengalami perubahan atau program tersebut menjadi tidak berguna pada lingkungan tersebut.
2. Meningkatnya kompleksitas.
   Seiring dengan berkembangnya sebuah program maka program tersebut menjadi lebih kompleks. Kemudian penambahan sumber daya juga dibutuhkan untuk memelihara dan menyederhanakan struktur programnya.
   Ada sebuah pernyataan yang mengatakan :
   "Most often overlooked risk in software engineering: As the system grows over time, it will become too complex or disjointed to understand or make work reliably." Deutsch (1998).
3. Aturan yang fundamental dari sebuah program evolusi.
   Pada saat membuat sebuah program evolusi telah ditentukan ukuran, metriks serta indikatornya.
4. Tetap dijaga stabilitas dari organisasinya.
   Sebagai contoh dengan menambahkan sumber daya (misal manusia) tetapi tidak mengubah produktivitas.
5. Tetap *familiarity*.
   Dimana pada saat membuat fungsi yang baru perbedaannya tidak jauh dari fungsi-fungsi yang sebelumnya.

II. PERAWATAN PERANGKAT LUNAK

Istilah perawatan perangkat lunak digunakan untuk menjabarkan aktivitas dari analis sistem (*software engineering*) yang terjadi pada saat hasil produk perangkat lunak sudah digunakan oleh pemakai (*user*)

Biasanya pengembangan produk perangkat lunak memerlukan waktu antara 1 sampai dengan 2 tahun, tetapi pada fase perawatan perangkat lunak menghabiskan 5 sampai dengan 10 tahun.

Aktivitas yang tejadi pada fase pemeliharaan antara lain :

- Penambahan atau peningkatan atau perbaikan untuk produk perangkat lunak.
- Penambahan fungsi-fungsi baru.
- Perbaikan tampilan dan modus interaktif.
- Perbaharui dokumen eksternal.
- Perbaharui dokumen internal.
- Perbaharui karakteristik perfomasi dari system.
- Adaptasi produk dengan lingkungan mesin yang baru.
- Pemindahan perangkat lunak ke sistem yang berlainan.
- Modifikasi untuk dapat mempergunakan protokol atau disk drive tambahan.
- Perbaikan permasalah yang timbul.
- Pembenaran kesalahan yang timbul setelah produk perangkat lunak dipergunakan oleh user (pemakai).

Salah satu hal dari pelaksanaan evolusi perangkat lunak adalah perawatan. Perawatan perangkat lunak adalah melakukan modifikasi dari perangkat lunak yang telah jadi dan telah dikirimkan untuk memperbaiki kesalahan-kesalahan, untuk meningkatkan performansi atau hal lain yang berkaitan dengan perangkat lunak tersebut juga untuk melakukan adaptasi terhadap perubahan lingkungan dari penggunaan perangkat lunak tersebut. Hal tersebut berdasarkan diatas **ANSI/IEEE Std. 729-1983**. Aktivitas yang dilakukan pada saat perawatan perangkat lunak tersebut adalah :

1. Perawatan yang dilakukan untuk penyesuaian (*Adaptive Maintenance*)
   Melakukan pengawasan terhadap sistem yang akan berubah, seperti melakukan pertemuan untuk membahas mengenai permintaan dari kebutuhan baru.
2. Perawatan yang dilakukan untuk perbaikan (*Corrective Maintenance*)
   Melakukan pengawasan setiap saat sehingga sistem berjalan sesuai dengan fungsinya, seperti dengan cara membuat laporan dari kesalahan yang timbul.
3. Perawatan yang dilakukan untuk penyempurnaan (*Perfective Maintenance*)

Memperbaiki beberapa aspek agar sistem dapat meningkatkan kebutuhan yang diperlukan dimasa yang akan datang, seperti melakukan serangkaian tes.

4. Perawatan yang dilakukan untuk pencegahan (*Preventative Maintenance*)
   Melakukan perubahan sistem untuk menghindarkan kegagalan dimasa yang akan datang, seperti meningkatkan penanganan dari kesalahan.

Faktor-faktor yang ada dalam perawatan dibagi 2 yaitu

1. Faktor Staf
   - Menemukan orang dengan keterampilan yang tepat.
   - Mengkoordinasi staf.
   - Perancangan tidak lama dibutuhkan untuk memberikan masukan .
   - Programmer *maintenance* terlihat sebagai warga kelas dua.
   - Kemampuan komprehensif.
   - Prioritas bisnis dan manajemen mungkin diatas bidang teknis.
2. Faktor Teknik
   - Perawatan pengkodean dapat menambah kesalahan.
   - Memperkecil dokumentasi.
   - Sistem baru terlalu rumit bagi perangkat keras.
   - Model pemrograman.
   - Waktu yang terbatas untuk pengujian.
   - *Path* dinamis.
   - Ukuran dan kerumitan perangkat lunak .

Untuk menghadapi masalah perawatan perangkat lunak membutuhkan pemikiran yang berbeda dibandingkan pengembang, misalnya menyelesaikan :

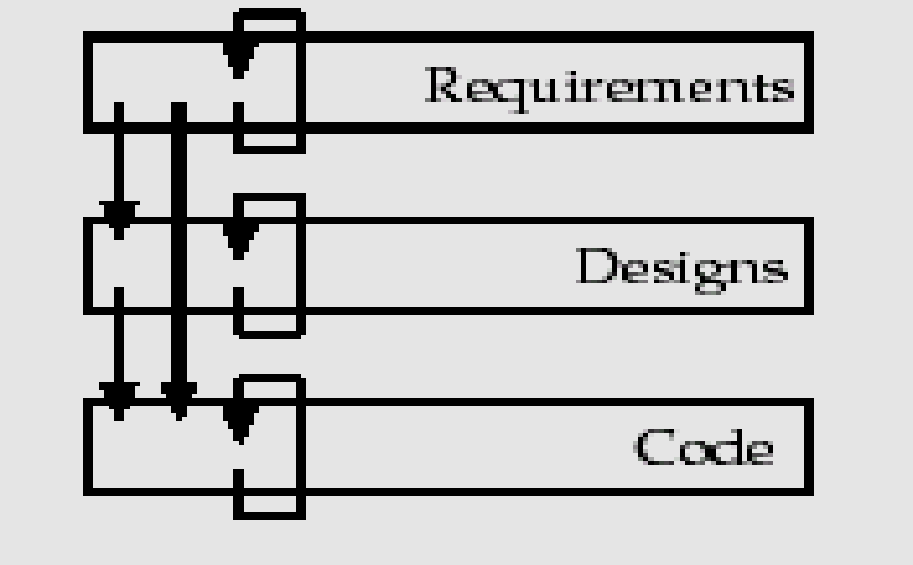
- Hilangkan kode program.
- Pencarian .

III. EVOLUTION STRATEGI

Dalam melakukan evolusi perangkat lunak ada beberapa strategi yang dapat digunakan, diantaranya :

1. Forward Engineering

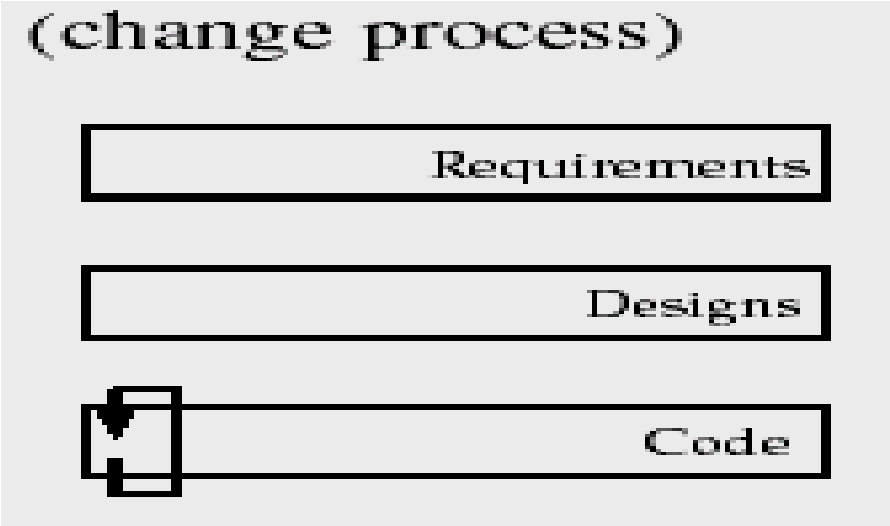
Forward Engineering adalah sebuah proses pengubahan dari abstraksi level yang paling tinggi(*Requirement*) dan logik ke level design sampai ke level fisik (C*ode*)dari sistem.



Gambar 5.1 Proses *Forward Engineering*

2. Restructuring

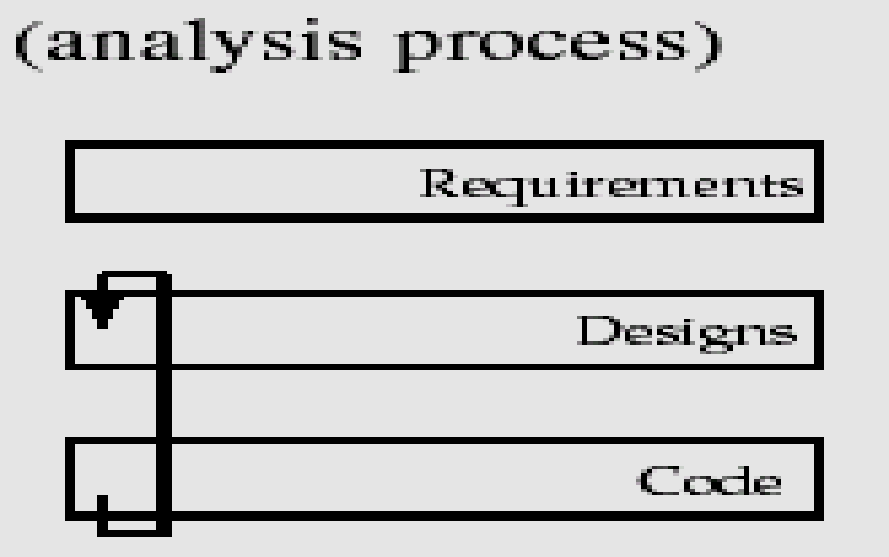
   Restructuring adalah merupakan proses perubahan perangkat lunak yang terjadi pada level phisik(*Code*).



Gambar 5.2 Proses *Restructuring*

3. Redocumenting

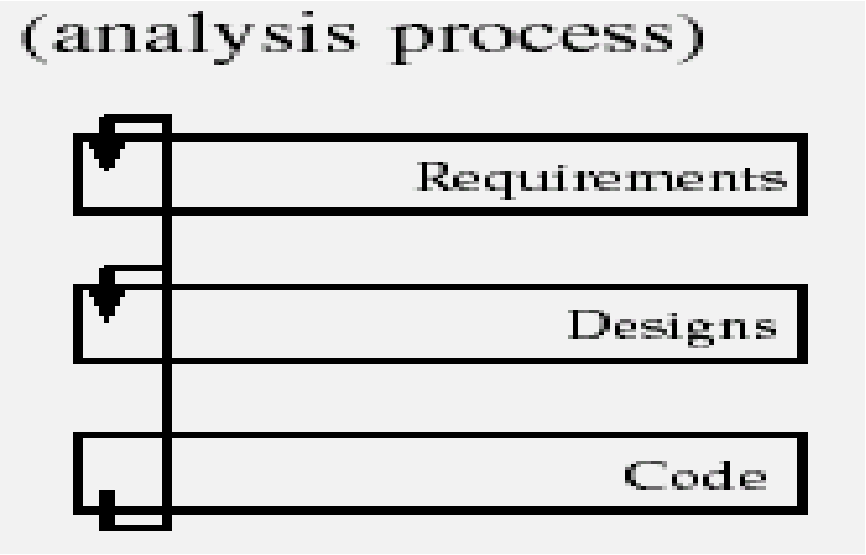
   Redocumenting adalah proses revisi terhadap dokumentasi system yang telah ada pada setiap level abstraksi.



Gambar 5.3 Proses *Redocumenting*

4. Reverse Engineering

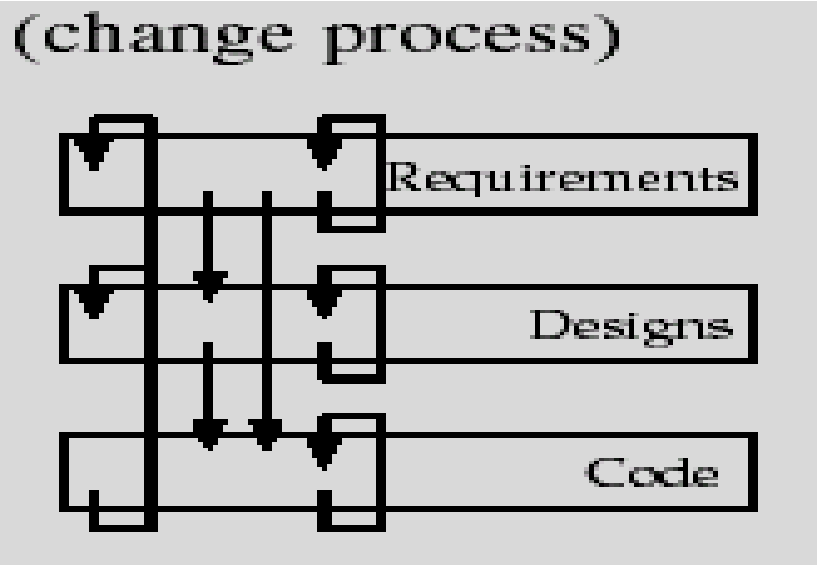
   Reverse Engineering adalah proses untuk mengindentifikasi sistem yang bermula dari level abstraksi yang paling rendah (misal *object code*), untuk menghasilkan spesifikasi formal.



Gambar 5.4 Proses *Reverse Engineering*

5. Reengineering

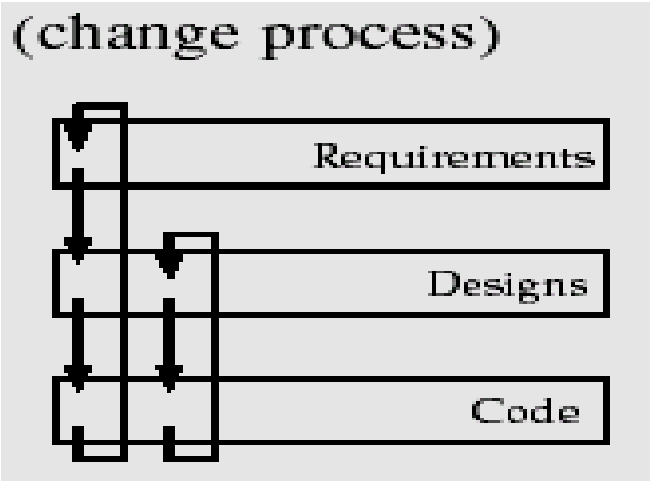
   Reengineering adalah proses untuk mengindentifikasi sistem yang bermula dari level abstraksi yang paling rendah (misal *object code*), untuk menghasilkan spesifikasi formal sehingga terbentuk *source code* baru.



Gambar 5.5 Proses *Reengineering*

6. *Roundtrip Engineering*

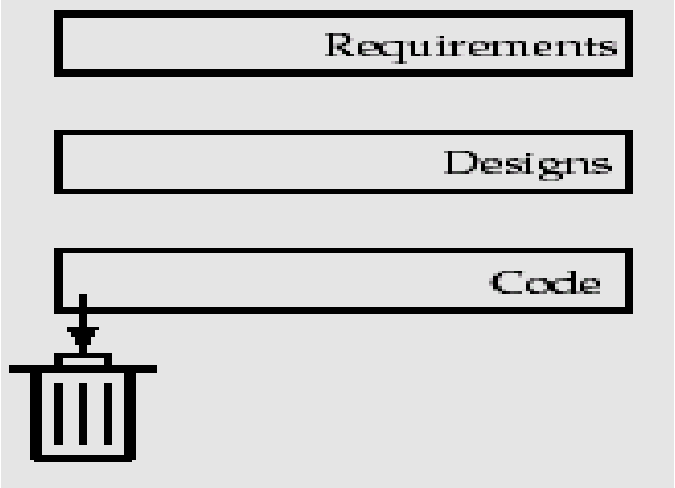
   Roundtrip Engineering merupakan proses untuk menjaga sinkronisasi antara *requirements, designs,* dan *code*.



Gambar 5.6 Proses Roundtrip Engineering

7. Retirement

   Retirement adalah proses dimana sebuah perangkat lunak secara keseluruhan sudah tidak dipergunakan kembali (dipensiunkan).



Gambar 5.7 Proses *Retirement*

Setiap strategi evolusi dapat dilakukan otomatisasi, dimana dapat dilakukan pada level fisik (*Code*). Proses tersebut terjadi pada bagian source code dengan mekanisme tertentu.