

**Overview**

Selama tiga dekade pertama dari era komputerisasi, tantangan utama adalah mengembangkan hardware komputer yang dapat mengurangi biaya pengolahan dan penyimpanan data.

Selama dekade tahun 1980-an, kemajuan pesat dari mikro elektronik menghasilkan kemampuan komputer yang lebih baik dengan tingkat biaya yang lebih rendah. Namun masalah sekarang berbeda. Tantangan utama adalah mengurangi biaya dan memperbaiki kualitas solusi berbasis komputer (solusi yang diimplementasikan dengan menggunakan software). Software merupakan faktor kunci dalam keberhasilan suatu usaha, software dapat membedakan satu perusahaan dari perusahaan saingannya.

Perancangan perangkat lunak adalah disiplin manajerial dan teknis yang berkaitan dengan pembuatan dan pemeliharaan produk perangkat lunak secara sistematis, termasuk pengembangan dan modifikasinya, yang dilakukan pada waktu yang tepat dan dengan mempertimbangkan faktor biaya.

Tujuan perancangan perangkat lunak adalah untuk:

- (1) memperbaiki kualitas produk perangkat lunak,*
- (2) meningkatkan produktifitas, dan*
- (3) memuaskan teknisi perangkat lunak, dsb.*

Dalam Rekayasa Perangkat Lunak dilakukan Proses Perangkat Lunak dengan menggunakan model Proses yang merupakan Daur Hidup Rekayasa Perangkat Lunak.

Model Proses ini terdiri dari beberapa karakteristik pendekatan proses. Dalam Proses pembangunan Perangkat Lunak perlu diketahui Biaya yang dikeluarkan.

Pengenalan Komputer**A. Definisi Komputer**

Komputer merupakan suatu perangkat elektronika yang memiliki kemampuan untuk menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis berdasarkan perangkat aturan tertentu.

Berdasarkan uraian di atas, maka dapat diuraikan bahwa setidaknya suatu komputer harus memenuhi kaidah-kaidah berikut ini:

- a. Komputer dapat melakukan pengolahan data,
- b. Komputer dapat memberikan/menghasilkan informasi,
- c. Komputer merupakan alat elektornik,
- d. Komputer dapat menerima input data (teks, angka, suara, signal, dll),
- e. Komputer menggunakan program yang tersimpan dalam memori komputer,
- f. Komputer bekerja secara otomatis,
- g. Komputer dapat menyimpan program dan data hasil olahannya.

B. Sistem Komputer

Sebuah sistem komputer tersusun atas tiga elemen yang saling terkait satu sama lainnya, yaitu:

1. **Hardware (Perangkat Keras),**

Hardware (Perangkat Keras) merupakan kumpulan segala piranti/komponen dari sebuah komputer yang sifatnya bisa dilihat secara kasat mata dan dapat diraba secara langsung.

Beberapa komponen perangkat keras mudah dikenali, seperti casing komputer, keyboard, dan monitor. Namun, ada banyak jenis komponen perangkat keras yang lainnya untuk membentuk sebuah komputer.



2. **Software (Perangkat Lunak)**

Software (Perangkat Lunak), merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer.

Software dapat berupa program atau aktifitas menjalankan suatu perintah atau instruksi melalui fasilitas interaksi pada komputer yang dibuat dengan menggunakan bahasa pemrograman oleh *programmer* yang selanjutnya dikompilasi dengan aplikasi kompilator sehingga menjadi sebuah kode yang nantinya akan dikenali oleh mesin *hardware*.



3. **Brainware (Perangkat Manusia)**

Brainware (Perangkat Manusia) merupakan orang yang terlibat dalam mengoperasikan atau memakai serta mengatur sistem di dalam perangkat komputer. Dapat diartikan juga sebagai perangkat intelektual yang mengoperasikan dan juga mengeksplorasi kemampuan dari perangkat keras maupun perangkat lunak.

Contoh dari *brainware* yaitu *programmer*, *operator*, serta orang yang sedang menggunakan perangkat komputer.



Pengenalan Rekayasa Perangkat Lunak

Untuk menghasilkan suatu perangkat lunak (*software*) dibutuhkan dukungan banyak aspek, proses, dan tindakan yang semuanya itu menyatu dalam suatu keilmuan yang disebut dengan Rekayasa Perangkat Lunak (*Software Engineering*). Secara sederhana target dari Rekayasa Perangkat Lunak adalah dalam rangka menghasilkan suatu *software* yang berkualitas.

A. Perangkat Lunak

Perangkat lunak saat ini telah menjadi kekuatan baru yang sangat menentukan dalam mendukung suatu aktifitas. Perangkat lunak menjadi mesin yang mengendalikan proses pengambilan keputusan di dalam dunia bisnis, berfungsi sebagai basis dari berbagai bentuk pelayanan serta penelitian keilmuan modern.

Saat ini perangkat lunak memiliki dua peran. Di satu sisi berfungsi sebagai sebuah produk, dan di sisi lain sebagai media yang mengantarkan sebuah produk.

Di masa lalu, perangkat lunak bersifat sederhana dan karenanya, pengembangan perangkat lunak merupakan kegiatan yang juga sederhana. Namun, seiring meningkatnya teknologi dan peradaban manusia, perangkat lunak menjadi lebih kompleks dan proyek perangkat lunak tumbuh menjadi lebih besar.

Jadi, apa yang dimaksud dengan perangkat lunak ? Perangkat lunak (*Software*) adalah :

- (1) Perintah/instruksi (program komputer) yang mana bila dieksekusi akan memberikan fungsi dan unjuk kerja seperti yang diinginkan.
- (2) Struktur data yang memungkinkan program memanipulasi data dan informasi secara proporsional.
- (3) Dokumen yang menggambarkan operasi dan kegunaan program.

B. Klasifikasi Perangkat Lunak

1. **Sistem Operasi (*operating system*)**, merupakan perangkat lunak yang berfungsi untuk mengoperasikan komputer serta menyediakan antarmuka dengan perangkat lunak lain atau dengan pengguna.

Contoh perangkat lunak Sistem Operasi :

- ☒ MS DOS,
- ☒ MS Windows (dengan berbagai generasi),
- ☒ Macintosh,
- ☒ UNIX (dengan berbagai versi),
- ☒ LINUX (dengan berbagai distribusi), *NetWare*, dan sebagainya.

Masing-masing sistem operasi memiliki perbedaan dalam lingkup / *platform* operasi, jumlah pemakai, metode interaksi pemakai, sifatnya (*open source* / *close source*) dan lisensi penggunaan.

2. **Program Utilitas (*Utility*)**, merupakan program khusus yang berfungsi sebagai perangkat pemeliharaan suatu sistem komputer, seperti anti virus, partisi hardisk, manajemen hardisk, dan sebagainya.

Contoh produk program utilitas : *Norton Utilities*, *PartitionMagic*, *McAfee*, dan sebagainya

3. **Program Aplikasi**, merupakan program yang dikembangkan untuk memenuhi kebutuhan yang spesifik.

Contoh : aplikasi akuntansi, aplikasi perbankan, aplikasi manufaktur, dan sebagainya.

4. **Paket Pemrograman**, merupakan program yang dikembangkan untuk kebutuhan umum,

Contoh: ☒ pengolah kata / text editor (misalnya: MS.Word, Notepad, Latex, dll),
☒ pengolah angka / lembar kerja (misalnya: MS Excel, dll),
☒ presentasi (Misalnya: MS PowerPoint, dll),
☒ desain grafis (Misalnya: 3D, CorelDraw, PhotoShop, dan sebagainya)

5. **Bahasa Pemrograman**, merupakan sebuah instruksi standar yang bertugas untuk memberikan instruksi kepada komputer. Sering disebut juga dengan bahasa komputer atau bahasa pemrograman komputer.

Bahasa pemrograman juga dapat dikatakan sebagai alat untuk menampung suatu kumpulan dari aturan sintaks dan semantik yang khususnya dipakai untuk mendefinisikan sebuah program yang ada di komputer.

Contoh bahasa pemrograman yang umum digunakan, yaitu : Java Script, PHP, HTML, C++ , Python, dan sebagainya.

C. Aplikasi Perangkat Lunak

Perangkat lunak dapat diaplikasikan ke berbagai situasi dan lingkungan dimana serangkaian langkah prosedural telah didefinisikan. Cakupan perangkat lunak berikut ini menunjukkan betapa luasnya potensi pengembangan aplikasi yang dapat dilakukan :

- ☞ **Perangkat Lunak Sistem**, Sekumpulan program yang ditulis untuk melayani program-program yang lain. Seperti kompiler, editor dan utilitas pengatur file.
- ☞ **Perangkat Lunak Real-Time**, Program-program yang memonitor, menganalisa, mengontrol kejadian di dunia nyata pada saat peristiwa itu terjadi (*real-time event*).
- ☞ **Perangkat Lunak Bisnis**, memproses informasi bisnis seperti *payroll*, *inventory*, *account payable*, dll.
- ☞ **Perangkat Lunak Teknik dan Ilmu Pengetahuan**, ditandai dengan penggunaan algoritma *number crunching*. Perangkat lunak ini memiliki cakupan aplikasi mulai dari aplikasi untuk sistem astronomi sampai vulkanologi, dari analisa otomotif sampai dinamika orbit pesawat ruang angkasa, dan dari biologi molekuler sampai pabrik yang sudah diotomatisasi.
- ☞ **Embedded Software**, produk pintar telah menjadi bagian yang umum bagi hampir semua level konsumen dan pasar industri karena dapat memberikan fungsi yang terbatas serta fungsi esoterik (misalnya, *keypad control* untuk sebuah *microwave*) atau memberikan kemampuan kontrol dan fungsi yang penting (contohnya fungsi digital dalam sebuah mobil seperti kontrol bahan bakar, penampilan panel di *dashboard*, sistem pengereman, dan sebagainya).
- ☞ **Perangkat Lunak Komputer Personal**, sesuai kebutuhan personal seperti pengolah kata, spreadsheet, multimedia, manajemen database, aplikasi keuangan bisnis, hanya merupakan beberapa contoh saja dari ratusan aplikasi yang ada.
- ☞ **Perangkat Lunak Kecerdasan Buatan**, perangkat lunak kecerdasan buatan (*Artificial Intelligent / AI*) menggunakan algoritma non-numeris untuk memecahkan masalah kompleks yang tidak dapat dilakukan perhitungan atau analisa secara langsung. Area kecerdasan buatan yang aktif adalah sistem pakar, disebut juga sistem berbasis pengetahuan. Sistem yang lain adalah Jaringan Syaraf Tiruan, *Voice and Image Recognition*, *game playing*, dan sebagainya.

D. Karakteristik Perangkat Lunak

Atribut perangkat lunak seharusnya memberikan pengguna kebutuhan fungsionalitas dan unjuk kerja yang dapat dipelihara. Dalam Buku *Software Engineering* _ Ian Sommerville, Perangkat Lunak mempunyai Karakteristik sebagai berikut:

- 1) *Maintanability*, perangkat lunak harus dapat memenuhi perubahan kebutuhan, dapat dipelihara.
- 2) *Dependability*, perangkat lunak harus dapat dipercaya.
- 3) *Efisiensi*, perangkat lunak harus efisien dalam penggunaan resource.
- 4) *Usability*, perangkat lunak harus dapat digunakan sesuai dengan yang direncanakan.

E. Proses Perangkat Lunak

Proses Perangkat Lunak merupakan sekumpulan aktifitas yang memiliki tujuan untuk pengembangan ataupun evolusi perangkat lunak.

Aktifitas umum dalam semua proses perangkat lunak terdiri dari:

1. *Software Specification* – apa yang harus dilakukan oleh perangkat lunak dan batasan / kendala pengembangannya.
2. *Software Development* – proses memproduksi sistem perangkat lunak.
3. *Software Validation* – pengujian perangkat lunak terhadap keinginan pengguna.
4. *Software Evolution* – perubahan perangkat lunak berdasarkan perubahan keinginan.

Suatu proses model adalah suatu representasi abstrak suatu model. Proses model menampilkan suatu deskripsi suatu proses dari beberapa perspektif tertentu.

Proses Perangkat Lunak dapat dikatakan sebagai aktifitas yang saling terkait (koheren) untuk menspesifikasikan, merancang, implementasi dan pengujian sistem perangkat lunak.

F. Karakteristik Proses Perangkat Lunak

Karakteristik Proses Perangkat Lunak terdiri dari:

- *Understandability*, membuat proses secara eksplisit didefinisikan dan bagaimana sehingga mudah untuk mengerti definisi proses.
- *Visibility*, aktifitas proses menghasilkan hasil yang jelas sehingga tahapan proses yang dilakukan terlihat.
- *Supportability*, aktifitas proses dapat didukung atas *CASE tools*.
- *Acceptability*, penerimaan atas proses yang terdefinisi dan yang digunakan oleh *Engineer* selama pembangunan Produk Perangkat Lunak.
- *Reliability*, proses didesain dalam suatu metode untuk dihindarkan dari kesalahan.
- *Robustness*, proses dapat meneruskan dalam masalah yang tidak diharapkan terjadi.
- *Maintainability*, proses yang merefleksikan atas perubahan thd permintaan atau perbaikan proses yang diidentifikasi.
- *Rapidity*, bagaimana cepat dapat berjalan atas proses pengiriman atau implementasi sebuah sistem dari Spesifikasi yang ada sampai selesai.

G. Model Proses Perangkat Lunak

Proses adalah kerangka kerja untuk tugas yang diperlukan dalam membangun perangkat lunak yang berkualitas tinggi.

Model Proses adalah model yang memfokuskan pada seluruh proses di dalam sistem yang mentransformasikan data menjadi informasi. Model proses juga menunjukkan aliran data yang masuk dan keluar pada suatu proses.

Model Proses Perangkat Lunak merupakan suatu representasi proses perangkat lunak yang disederhanakan, dipresentasikan dari perspektif khusus.

Contoh perspektif proses:

- ☒ Perspektif Alur kerja (*workflow*) – barisan kegiatan
- ☒ Perspektif Alur Data (*Data flow*) – alur informasi
- ☒ Perspektif Peran/Aksi – siapa melakukan apa.

Ada 4 kegiatan/aktivitas pada **Proses Perangkat Lunak** :

❖ **Spesifikasi Perangkat Lunak :**

Fungsionalitas perangkat lunak dan batasan kemampuan operasinya harus didefinisikan.

❖ **Pengembangan Perangkat Lunak :**

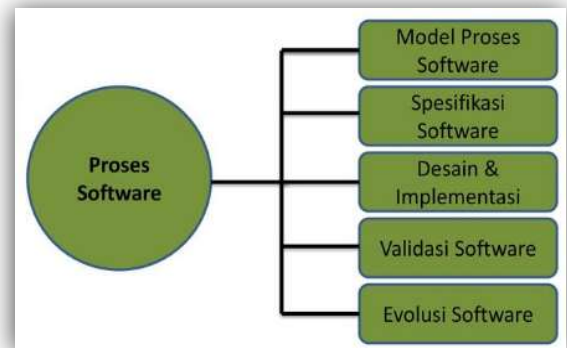
Perangkat lunak yang memenuhi spesifikasi harus di produksi.

❖ **Validasi Perangkat Lunak :**

Perangkat lunak harus divalidasi untuk menjamin bahwa perangkat lunak melakukan apa yang diinginkan oleh pelanggan.

❖ **Evolusi Perangkat Lunak :**

Perangkat lunak harus berkembang untuk memenuhi kebutuhan pelanggan.



H. Mutu Perangkat Lunak

Terdapat 3 pihak (minimal) yang mempengaruhi mutu perangkat lunak yaitu:

✦ **Sponsor**

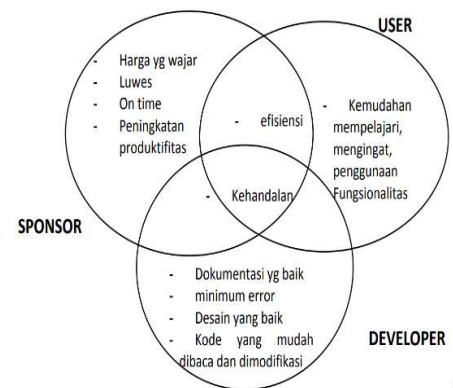
Seseorang atau organisasi yang membiayai/membayar selama pengembangan atau perantaraan sistem *software* dan biasanya mempunyai respon terhadap pengembangan sistem *software* itu sendiri dengan melibatkan perhitungan biaya yang optimal.

✦ **User**

Setiap orang yang secara langsung berinteraksi terhadap eksekusi *software*, yang secara langsung memberi input ke komputer dan menggunakan output dari komputer.

✦ **Developer**

Seseorang atau organisasi yang memberikan modifikasi dan memelihara terhadap *error* serta mengembangkan sistem *software* tersebut.



Gambar Sisi Pandang dari komponen kategori terhadap Mutu Perangkat Lunak

Masing-masing komponen kategori mempunyai sudut pandang tersendiri terhadap mutu suatu perangkat lunak, tapi kriteria tersebut tidak saling independen.

Ada beberapa faktor yang dapat kita gunakan untuk menentukan mutu/kualitas produk perangkat lunak dan game. Faktor-faktor tersebut seperti pada tabel berikut:



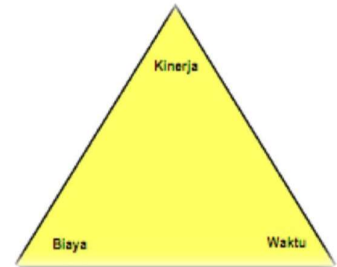
Faktor	Keterangan
Correctness	Kebenaran spesifikasi dan tujuan pembuatan produk.
Reliability	Pengukuran yg dilakukan untuk pemeriksaan tingkat kegagalan suatu produk.
Efficiency	Kemampuan sumber daya yg dibutuhkan sistem utk menjalankan fungsinya.
Integrity	Keamanan sistem perangkat lunak dan game.
Usability	Penggunaan produk perangkat lunak dan game.
Maintainability	Aktivitas untuk menemukan dan memperbaiki kesalahan produk.

Flexibility	Upaya yg dilakukan utk mengubah/memodifikasi program perangkat lunak dan game dalam pemeliharaan produk.
Testability	Pengujian produk utk memastikan program berjalan dengan baik.
Portability	Penyesuaian produk terhadap lingkungan. Seperti sistem operasi dan perangkat pengguna.
Reusability	Penggunaan kembali produk utk digunakan pada aplikasi lain.
Interoperability	Produk perangkat lunak dan game dapat dihubungkan dgn sistem lainnya.

I. Biaya Perangkat Lunak

Sekitar 60% untuk biaya pengembangan, 40% biaya pengujian. Untuk perangkat lunak berbasis pengguna (*custom*), biaya evolusi biasanya melebihi biaya pengembangan.

Biaya beragam tergantung pada tipe sistem yang akan dikembangkan dan kebutuhan sistem seperti unjuk kerja dan kehandalan system. Distribusi biaya bergantung pada model pengembangan yang digunakan.



J. Latar Belakang Rekayasa Perangkat Lunak

Faktor-faktor yang melatarbelakangi munculnya Rekayasa Perangkat Lunak:

- Ketidakmampuan organisasi memprediksi waktu, usaha dan biaya untuk membangun perangkat lunak,
- Perubahan rasio biaya perangkat keras terhadap harga perangkat lunak,
- Kemajuan pesat perangkat keras,
- Kemajuan dalam teknik-teknik pembuatan perangkat lunak,
- Tuntutan yang lebih tinggi terhadap jumlah perangkat lunak,
- Tuntutan yang lebih tinggi terhadap mutu perangkat lunak,
- Meningkatnya peran pemeliharaan (*maintenance*).

Pengembangan perangkat lunak sekarang mengharuskan kehadiran tim yang dapat mempersiapkan rencana dan desain secara terperinci, melakukan pengujian, mengembangkan antar muka pengguna yang intuitif, dan mengintegrasikan semua aktifitas ini ke dalam sistem. Pendekatan baru ini menyebabkan munculnya sebuah disiplin ilmu yang dikenal sebagai rekayasa perangkat lunak.

K. Rekayasa Perangkat Lunak

Perangkat lunak merupakan program-program komputer dan dokumentasi yang berkaitan. Produk perangkat lunak dibuat untuk pelanggan tertentu ataupun untuk pasar umum terdiri dari:

- *Generik* – dibuat untuk dijual ke suatu kumpulan pengguna yang berbeda.
- *Be spoke (custom)* – dibuat untuk suatu pengguna tunggal sesuai dengan spesifikasinya.

Rekayasa perangkat lunak berasal dari 2 kata yaitu:

- * *Software* (Perangkat Lunak)
- * *Engineering* (Rekayasa)

Perangkat Lunak (*Software*) adalah *source code* pada suatu program atau sistem. Perangkat lunak tidak hanya dokumentasi terhadap *source code* tapi juga dokumentasi terhadap sesuatu yang dibutuhkan selama pengembangan, instalasi, penggunaan dan pemeliharaan sebuah sistem.

Rekayasa (*Engineering*) adalah aplikasi terhadap pendekatan sistematis yang berdasar atas ilmu pengetahuan dan matematis serta aplikasi tentang produksi terhadap struktur, mesin, produk, proses atau sistem.

Rekayasa Perangkat Lunak merupakan suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu *communication*, *requirements capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), *design*, *coding*, *testing* sampai *maintenance* (pemeliharaan sistem) setelah digunakan.

Dari cara pandang lain, Rekayasa Perangkat Lunak adalah pendekatan sistematis untuk merekayasa perangkat lunak yang handal/bermutu, tepat waktu dan dengan biaya yang optimal.

Secara umum, Tujuan Rekayasa Perangkat Lunak (RPL) memiliki banyak kesamaan dengan disiplin ilmu rekayasa yang lain. Ilmu rekayasa akan selalu berusaha menghasilkan keluaran (output) yang performanya tinggi, waktu penyelesaian yang singkat dan hemat.

Secara lebih khusus dapat dinyatakan bahwa tujuan Rekayasa Perangkat Lunak adalah sebagai berikut:

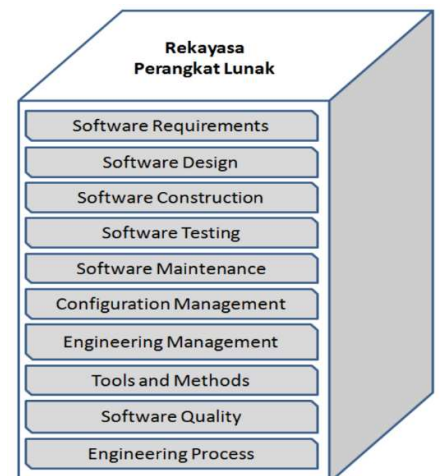
1. Menghasilkan perangkat lunak yang berunjuk kerja tinggi, andal serta tepat waktu.
2. Menghasilkan perangkat lunak yang memiliki kekuatan dalam menghadapi berbagai ancaman dan gangguan, baik yang bersumber dari internal maupun eksternal.
3. Menghasilkan perangkat lunak dengan biaya produksi yang rendah.
4. Menghasilkan perangkat lunak yang biaya pemeliharaan yang rendah.
5. Menghasilkan perangkat lunak yang bisa bekerja di berbagai macam platform.

Jadi hal yang perlu digaris bawahi, bahwa Rekayasa Perangkat Lunak (RPL) merupakan serangkaian proses yang amat panjang untuk membuat atau menciptakan suatu perangkat lunak yang berkualitas, bukan merupakan cabang ilmu komputer yang mempelajari tentang *technical coding*.

L. Ruang Lingkup Rekayasa Perangkat Lunak

Apabila dilihat dari persepsi wilayah cakupannya, RPL memiliki wilayah operasional yang dapat dilihat pada Gambar berikut ini.

- ☑ **Software Requirement**, merupakan kegiatan yang dilakukan untuk mengidentifikasi dan menganalisis kebutuhan perangkat lunak. Hasil akhir tahapan ini adalah spesifikasi dan model perangkat lunak.
- ☑ **Software Design**, adalah tahapan perancangan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak.
- ☑ **Software Construction**, berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan.
- ☑ **Software Testing**, meliputi pengujian pada kinerja perangkat lunak secara keseluruhan.



- ☑ **Software Maintenance**, mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
- ☑ **Software Configuration Management**, berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
- ☑ **Software Engineering Management**, berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak.
- ☑ **Software Engineering Tools and Methods**, mencakup kajian teoritis tentang alat bantu dan metode RPL.
- ☑ **Software Quality**, menitik beratkan pada kualitas dan daur hidup perangkat lunak.
- ☑ **Software Engineering Process**, berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL.

M. Lapisan dalam Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak dapat dipandang sebagai teknologi berlapis (*Software layer*) seperti yang tersusun sebagai berikut:

1. A Quality Focus (*fokus pada kualitas*), Ketika anda sedang membangun aplikasi, maka fokus kualitas menjadi hal pertama yang harus anda bangun. Anda harus memikirkan tentang kualitas *software* yang mau anda bangun, siapa yang akan dijadikan sasaran, untuk siapa aplikasi tersebut dibangun. Untuk itulah, fokus kualitas akan membantu programmer untuk mengetahui level aplikasi yang akan dibangun.



2. Process, Sesudah fokus pada kualitas dibangun, maka pemrogram juga harus mengetahui cara memproses yang harus dilakukan berkaitan dengan fokus kualitas perangkat lunak yang diinginkan. Lapisan Proses memberikan pengetahuan teknis untuk mengembangkan perangkat lunak. Lapisan ini mencakup beragam tugas yang mencakup analisis kebutuhan, desain, pengkodean, pengujian, dan fase pemeliharaan pengembangan perangkat lunak.

3. Methods, Berfungsi sebagai panduan teknis untuk menciptakan *software*. Dengan memakai metode, maka *programmer* juga akan melakukan langkah serta tindakan sesuai dengan metode tersebut. Metode yang akan dipakai juga harus sesuai dengan *software* yang ingin dibangun serta tujuan dari dibuatnya perangkat lunak tersebut.

Lapisan Metode mencakup banyak hal seperti *communication*, *requirements analysis*, *design modeling*, *program construction*, *testing*, dan *support*.

4. Tools, Memberikan dukungan terkomputerisasi atau semi-terkomputerisasi untuk lapisan proses dan metode. Terkadang alat diintegrasikan sedemikian rupa sehingga *tool* lain dapat menggunakan informasi yang dibuat oleh satu *tool*. *Software engineering tools* mendukung fungsi kerja dari *process layer* dan *methods layer*. Apabila *tools* diintegrasikan sehingga informasi yang dihasilkan oleh sebuah *tool* dapat digunakan oleh *tool* lainnya, terciptalah sebuah sistem untuk mendukung process pengembangan *software*. Sistem ini disebut *computer-aided software engineering (CASE)*.

Multi-penggunaan ini biasanya disebut sebagai *Computer-Aided Software Engineering (CASE)*. *CASE* menggabungkan database perangkat lunak, perangkat keras, dan rekayasa perangkat lunak untuk membuat rekayasa perangkat lunak yang analog dengan *Computer-Aided Design (CAD)* untuk perangkat keras. *CASE* membantu dalam pengembangan aplikasi termasuk analisis, desain, pembuatan kode, dan *debugging* dan pengujian.

Berbagai permasalahan dalam Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah pendekatan sistematis untuk pengembangan, operasi, pemeliharaan, dan penghentian penggunaan suatu perangkat lunak. Ada beberapa masalah mendasar yang dihadapi rekayasa perangkat lunak, antara lain:

A. Masalah Ruang Lingkup

Masalah mendasar dari rekayasa perangkat lunak adalah masalah ruang lingkup. Pengembangan sistem yang sangat besar membutuhkan serangkaian metode yang sangat berbeda dibandingkan dengan mengembangkan sistem yang kecil.

Dengan kata lain, metode yang digunakan untuk mengembangkan sistem kecil umumnya tidak sama dengan sistem yang besar. Satu rangkaian metode yang berbeda harus digunakan untuk mengembangkan perangkat lunak besar. Setiap proyek besar melibatkan penggunaan teknologi dan manajemen proyek.

Untuk proyek perangkat lunak, dengan teknologi yang dimaksud adalah metode, prosedur dan alat yang digunakan. Dalam proyek kecil, metode informal untuk pengembangan dan manajemen dapat digunakan. Namun, untuk proyek besar, keduanya harus jauh lebih formal.

Saat berurusan dengan proyek perangkat lunak kecil, persyaratan teknologi rendah dan persyaratan manajemen proyek juga rendah. Namun, ketika skala berubah menjadi sistem besar, untuk menyelesaikan masalah seperti itu dengan benar, penting bagi kita untuk bergerak ke dua arah yaitu metode yang digunakan untuk pembangunan dan manajemen proyek untuk proyek pengembangan juga harus lebih formal.

B. Biaya, Jadwal dan Kualitas

Biaya pengembangan sistem adalah biaya sumber daya yang digunakan untuk sistem, yang dalam sisi pandang perangkat lunak, adalah tenaga kerja, perangkat keras, perangkat lunak, dan sumber daya pendukung lainnya. Secara umum, komponen tenaga kerja bersifat dominan, karena pengembangan perangkat lunak sebagian besar padat karya dan biaya sistem komputasi sekarang cukup rendah.

Oleh karena itu, biaya proyek perangkat lunak diukur dalam hal orang/bulan, yaitu biaya dianggap sebagai jumlah total orang/bulan yang dihabiskan dalam proyek. Jadwal adalah faktor penting dalam banyak proyek. Trend bisnis menentukan bahwa waktu untuk memasarkan suatu produk harus dikurangi; artinya, waktu siklus dari konsep ke *delivery* harus kecil.

Setiap bisnis dengan persyaratan seperti itu juga akan mensyaratkan bahwa waktu siklus untuk membangun perangkat lunak yang dibutuhkan oleh bisnis menjadi lebih kecil. Salah satu faktor utama yang mendorong aspek produksi adalah kualitas.

Kualitas produk perangkat lunak memiliki tiga dimensi, yaitu:

- ☒ Operasi Produk
- ☒ Transisi Produk
- ☒ Revisi Produk

C. Masalah Konsistensi

Meskipun telah memiliki kualitas tinggi, biaya rendah dan waktu siklus kecil adalah tujuan utama dari setiap proyek, namun untuk organisasi ada tujuan lain yang ingin dicapai yaitu konsistensi. Sebuah organisasi yang terlibat dalam pengembangan perangkat lunak tidak hanya menginginkan biaya rendah dan kualitas tinggi untuk suatu proyek, tetapi menginginkannya secara konsisten.

