
BAB 1 PENDAHULUAN



Gambar 1.1. Tampilan desktop Microsoft Windows.

Coba kita perhatikan Gambar 1.1. di atas. Bagi pengguna komputer, gambar di atas merupakan tampilan yang sangat dikenal. Gambar ini merupakan tampilan desktop Sistem Operasi Microsoft Windows. Pada gambar tersebut, kita melihat sejumlah ikon-ikon tertentu. Apabila kita klik ganda pada satu ikon maka suatu perangkat lunak (software) tertentu akan terbuka dan dapat kita gunakan untuk menyelesaikan suatu tugas tertentu.

Pada masa sekarang, rasanya hampir semua bidang kehidupan tersentuh penggunaan perangkat lunak atau software. Beberapa perangkat lunak mungkin sudah terbiasa kita gunakan atau kita lihat seperti perangkat lunak untuk memainkan atau membuat musik, perangkat lunak untuk membantu kasir dalam penjualan barang, perangkat lunak untuk mengetik dokumen, dan lain-lain. Perangkat lunak ini merupakan hasil dari serangkaian proses atau kegiatan yang dikenal sebagai Rekayasa Perangkat Lunak. Apakah sebenarnya Rekayasa Perangkat Lunak itu? Bab ini akan memberi jawaban atas pertanyaan ini.

TUJUAN

Setelah mempelajari bab ini diharapkan kalian akan mampu :

- o Menjelaskan pengertian perangkat lunak, program, prosedur dan rekayasa perangkat lunak
- o Memahami tujuan rekayasa perangkat lunak
- o Memahami ruang lingkup rekayasa perangkat lunak
- o Memahami posisi bidang rekayasa perangkat lunak pada disiplin ilmu komputer dan keterkaitannya dengan bidang ilmu lain
- o Mengetahui perkembangan ilmu rekayasa perangkat lunak
- o Mengetahui profesi dan sertifikasi dalam bidang rekayasa perangkat lunak
- o Menjelaskan prinsip-prinsip pemecahan masalah dalam rekayasa perangkat lunak

1.1. PENGERTIAN REKAYASA PERANGKAT LUNAK

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah *Software Engineering*. Istilah *Software Engineering* mulai dipopulerkan tahun 1968 pada *Software Engineering Conference* yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (*software*) dan program komputer.

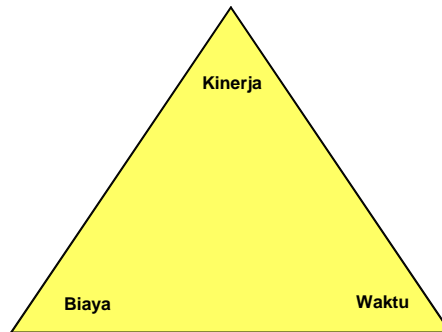
Perangkat lunak adalah *seluruh perintah yang digunakan untuk memproses informasi*. Perangkat lunak dapat berupa program atau prosedur. **Program** adalah kumpulan perintah yang dimengerti oleh komputer sedangkan **prosedur** adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999). Pengertian RPL sendiri adalah sebagai berikut:

Suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

Jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan "*semua aspek produksi*" pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

1.2. TUJUAN REKAYASA PERANGKAT LUNAK

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain. Mari kita perhatikan Gambar 1.2. berikut ini.



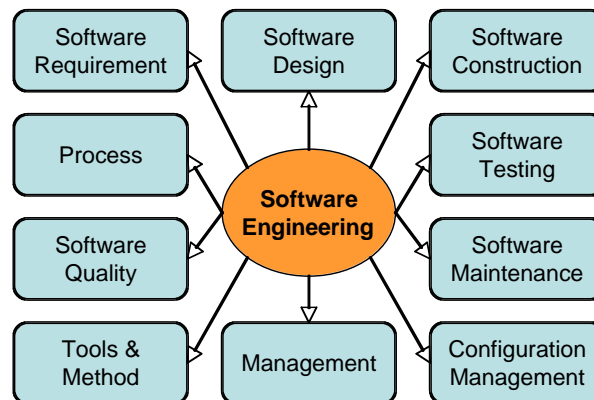
Gambar 1.2. Tujuan RPL.

Dari Gambar 1.2 dapat diartikan bahwa bidang rekayasa akan selalu berusaha menghasilkan output yang kinerjanya tinggi, biaya rendah dan waktu penyelesaian yang tepat. Secara lebih khusus kita dapat menyatakan tujuan RPL adalah :

- a. Memperoleh biaya produksi perangkat lunak yang rendah.
- b. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu.
- c. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis *platform*.
- d. Menghasilkan perangkat lunak yang biaya perawatannya rendah.

1.3. RUANG LINGKUP

Sesuai definisi yang telah disampaikan sebelumnya, maka ruang lingkup RPL dapat digambarkan sebagai berikut.



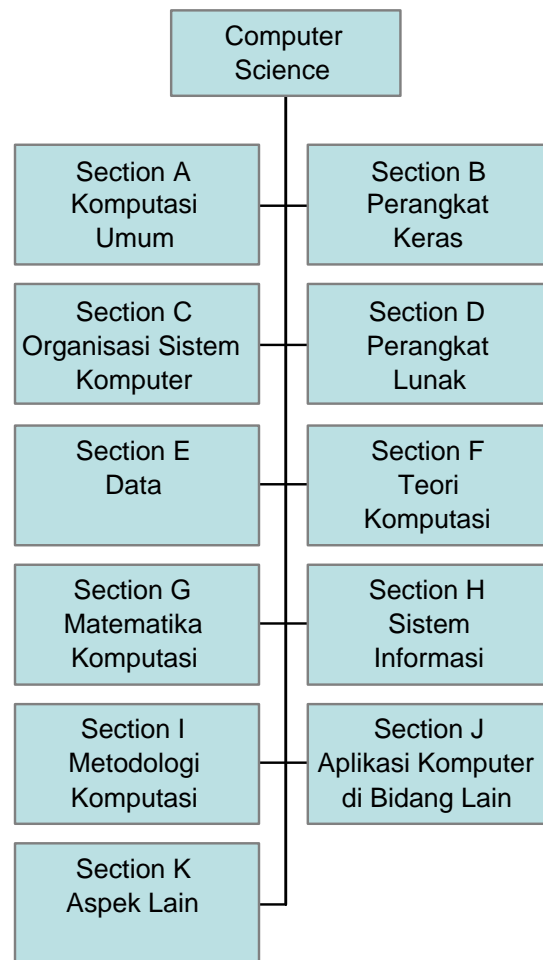
Gambar 1.3. Ruang lingkup RPL (Abran et.al., 2004).

- *Software requirements* berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak.
- *Software design* mencakup proses penentuan arsitektur, komponen, antarmuka, 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 keseluruhan perilaku perangkat lunak.
- *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 engineering process* berhubungan dengan definisi, implementasi, pengukuran, pengelolaan, perubahan dan perbaikan proses RPL.
- *Software quality* menitikberatkan pada kualitas dan daur hidup perangkat lunak.

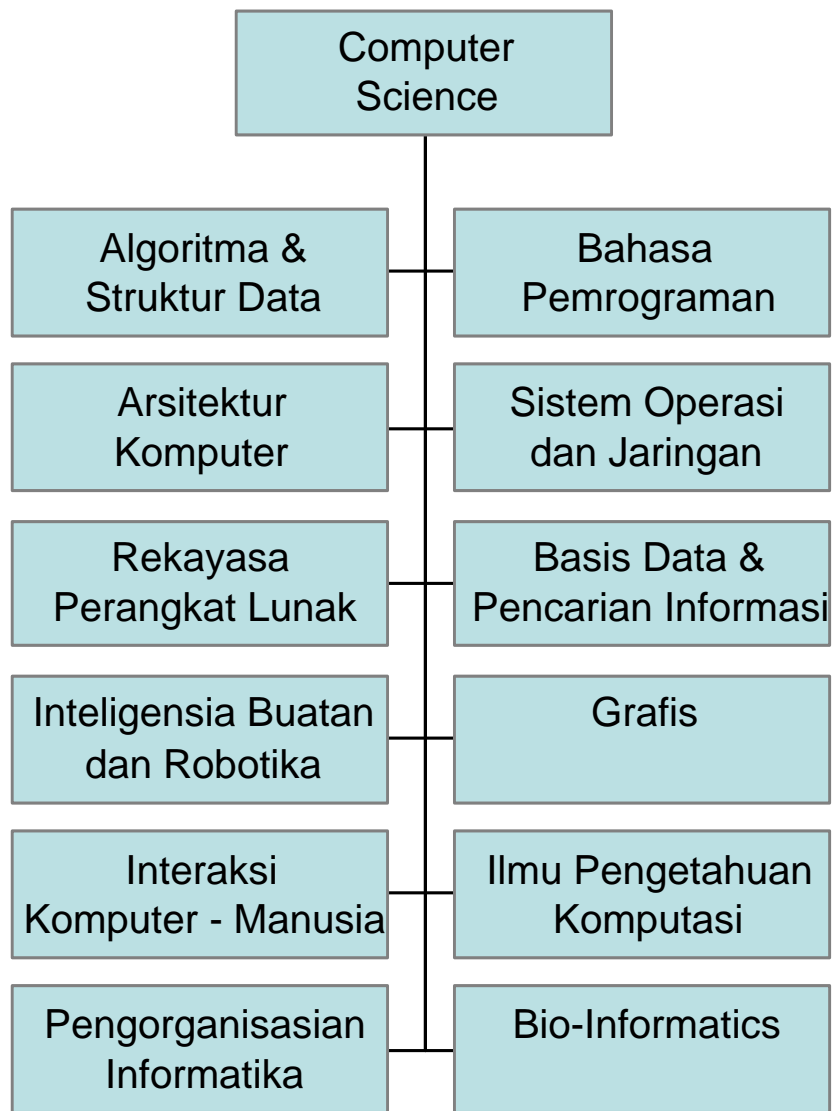
1.4. REKAYASA PERANGKAT LUNAK DAN DISIPLIN ILMU KOMPUTER

Disiplin ilmu komputer (*Computer Science*) lahir pada awal-awal tahun 1940-an yang merupakan integrasi dari teori algoritma, logika matematika dan ditemukannya cara penyimpanan program secara elektronik pada komputer. Sejak itu ilmu komputer mengalami perkembangan yang terus menerus sehingga cakupannya menjadi semakin meluas.

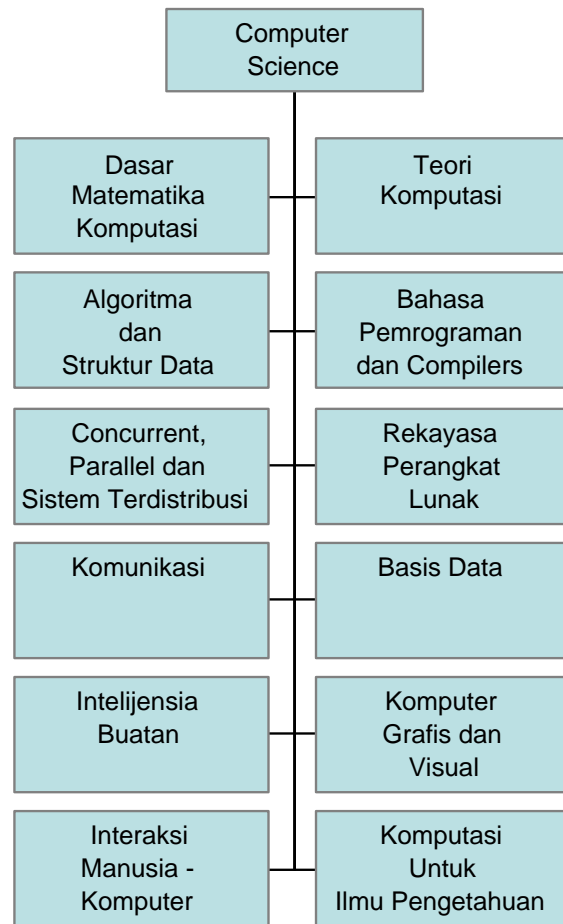
Cakupan pengetahuan dalam ilmu komputer seringkali didiskripsikan sebagai suatu studi sistematis pada proses-proses algoritma yang menjelaskan dan mentransformasikan informasi (Denning, 2000). Termasuk di sini adalah teori, analisis, disain, efisiensi, penerapan dan aplikasinya. Ada beberapa model pengelompokan sub-bidang ilmu dalam disiplin ilmu komputer seperti terlihat pada Gambar 1.4, 1.5 dan 1.6.



Gambar 1.4. Klasifikasi disiplin ilmu komputer menurut ACM (1998).



Gambar 1.5. Klasifikasi disiplin ilmu komputer menurut Denning (2000).

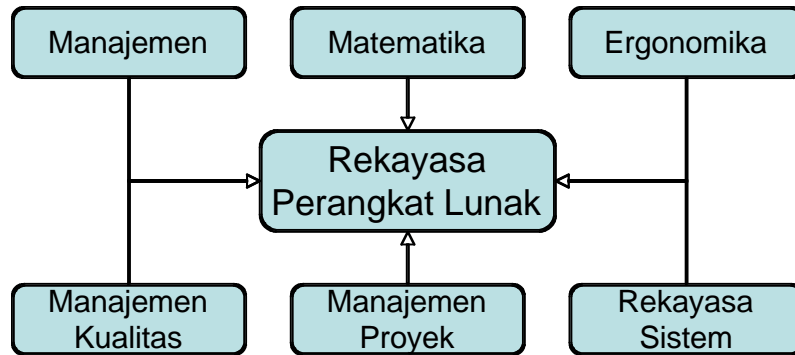


Gambar 1.6. Klasifikasi disiplin ilmu komputer menurut Wikipedia (2007).

Berdasarkan pengelompokan Denning (2000) dan Wikipedia (2007), RPL merupakan sub-bidang ilmu komputer yang setara dengan sub-bidang lainnya. Sedangkan menurut ACM (*Association for Computing Machinery*), RPL merupakan bagian dari Section D (Perangkat Lunak). Meskipun terlihat terpisah-pisah, namun dalam penerapannya, sub-bidang RPL selalu membutuhkan dukungan dari sub-bidang lain, terutama sub-bidang Algoritma dan Struktur Data, Bahasa Pemrograman, Basis Data, Sistem Operasi dan Jaringan, dan Sistem Informasi.

1.5. REKAYASA PERANGKAT LUNAK DAN DISIPLIN ILMU LAIN

Cakupan ruang lingkup yang cukup luas, membuat RPL sangat terkait dengan disiplin bidang ilmu lain. Tidak saja dengan sub-bidang dalam disiplin ilmu komputer namun dengan beberapa disiplin ilmu lain di luar ilmu komputer. Hubungan keterkaitan RPL dengan ilmu lain dapat dilihat pada Gambar 1.7.



Gambar 1.7. Keterkaitan RPL dengan bidang ilmu lain.

- Bidang ilmu manajemen meliputi akuntansi, finansial, pemasaran, manajemen operasi, ekonomi, analisis kuantitatif, manajemen sumber daya manusia, kebijakan dan strategi bisnis.
- Bidang ilmu matematika meliputi aljabar linier, kalkulus, peluang, statistik, analisis numerik dan matematika diskrit.
- Bidang ilmu manajemen proyek meliputi semua hal yang berkaitan dengan proyek, seperti ruang lingkup proyek, anggaran, tenaga kerja, kualitas, manajemen resiko, dan penjadwalan proyek.
- Bidang ilmu manajemen kualitas meliputi pengembangan sistem kualitas, manajemen resiko dan keandalan, perbaikan kualitas, dan metode-metode kuantitatif.
- Bidang ilmu ergonomika menyangkut hubungan (interaksi) antara manusia dengan komponen-komponen lain dalam sistem komputer.
- Bidang ilmu rekayasa sistem meliputi teori sistem, analisis biaya-keuntungan, pemodelan, simulasi, proses dan operasi bisnis.

1.6. PERKEMBANGAN REKAYASA PERANGKAT LUNAK

Meskipun baru dicetuskan pada tahun 1968, namun RPL telah memiliki sejarah yang cukup panjang. Gambar 1.8 menyajikan intisari perkembangan RPL. Dari sisi disiplin ilmu, RPL masih relatif muda dan akan terus berkembang. Arah perkembangan yang saat ini sedang dikembangkan antara lain meliputi :

Agile Software Development, Experimental Software Development, Model-Driven Software Development dan Software Product Lines.

Tahun	Kejadian
1940an	Komputer pertama yang membolehkan pengguna menulis kode program langsung
1950an	Generasi awal interpreter dan bahasa macro Generasi pertama compiler
1960an	Generasi kedua compiler Komputer mainframe mulai dikomersialkan Pengembangan perangkat lunak pesanan Konsep Software Engineering mulai digunakan
1970an	Perangkat pengembang perangkat lunak Perangkat minicomputer komersial
1980an	Perangkat Komputer Personal (PC) komersial Peningkatan permintaan perangkat lunak
1990an	Pemrograman berorientasi obyek (OOP) Agile Process dan Extreme Programming Peningkatan drastis kapasitas memori Peningkatan penggunaan internet
2000an	Platform interpreter modern (Java, .Net, PHP, dll) Outsourcing

Gambar 1.8. Perkembangan RPL.

1.7. PROFESI DAN SERTIFIKASI

Profesi sebagai seorang *Software Engineer* mungkin masih terasa asing di telinga orang Indonesia. Sebagian besar orang Indonesia mungkin lebih familiar dengan sebutan **Ahli Teknologi Informasi, Analis Sistem Informasi, Programmer, Operator** atau sebutan profesi lainnya. Hal ini karena adanya kerancuan tentang istilah RPL seperti telah disebutkan di awal bab. Namun di negara-negara yang maju dalam bidang teknologi informasi, sebutan *Software Engineer* telah mulai banyak digunakan.

Sertifikasi kompetensi dalam bidang RPL, saat ini masih menjadi perdebatan di kalangan ahli dan penyedia perangkat lunak. Sebagian besar sertifikasi dalam industri perangkat lunak biasanya sangat spesifik untuk perangkat lunak tertentu. Sebagai contoh, perusahaan perangkat lunak seperti Redhat Linux Inc., Adobe Inc., Oracle, atau Microsoft, memberikan sertifikasi

kemampuan pada seseorang yang menguasai perangkat lunak yang diproduksinya.

ACM (*Association for Computing Machinery*) pernah menyelenggarakan sertifikasi untuk program *Software Engineer* pada tahun 1980an, namun dihentikan karena kurangnya peminat. IEEE (*Institute of Electrical and Electronics Engineers*) telah mengeluarkan lebih dari 500 sertifikat profesi perangkat lunak. Di Canada, telah dikeluarkan sebuah sertifikat legal untuk RPL yang disebut sebagai ISP (*Information Systems Profesional*).

Saat ini, sertifikasi untuk RPL di Indonesia juga belum tersedia, namun telah disusun **Standar Kompetensi Kerja Nasional Indonesia untuk Bidang Programmer Komputer**. Meskipun belum memenuhi cakupan bidang RPL secara keseluruhan, namun paling tidak dapat digunakan sebagai pendekatan sertifikasi bidang RPL.

1.8. REKAYASA PERANGKAT LUNAK DAN PEMECAHAN MASALAH

Secara konsep, rekayasa perangkat lunak memiliki kedekatan dengan prinsip-prinsip pemecahan masalah. Pemahaman tentang masalah, strategi dan proses pemecahan masalah, serta pendekatan sistem pada pemecahan masalah akan sangat membantu proses rekayasa perangkat lunak.

1.8.1. Masalah dan Gejala

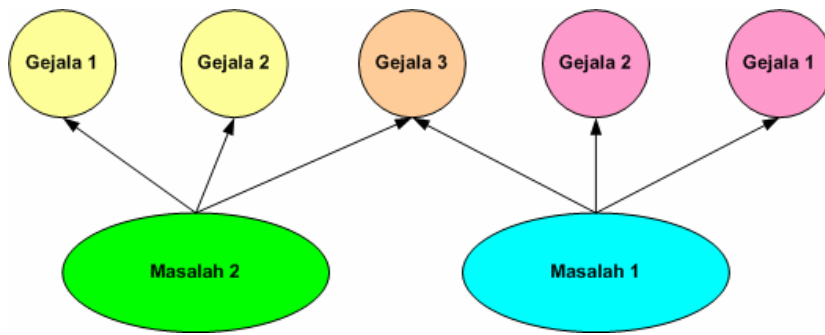
Masalah (problem) adalah *perbedaan antara kondisi yang terjadi dan kondisi yang diharapkan atau boleh juga diartikan sebagai perbedaan antara kondisi sekarang dengan tujuan yang diinginkan*. Sebagai contoh seorang siswa berharap memperoleh nilai di atas 80 untuk ujian mata pelajaran Pemrograman C++, namun pada kenyataannya dia hanya memperoleh nilai 60. Adanya perbedaan ini menunjukkan adanya masalah.



Gambar 1.9. Profesi dokter.

Seringkali kita kesulitan membedakan antara gejala dan masalah. **Gejala** adalah *tanda/petunjuk terjadinya suatu masalah*. Perhatikan seorang yang berprofesi sebagai dokter pada Gambar 1.9. Seorang dokter dalam usaha mengobati penyakit pasien selalu bertanya dulu tentang gejala-gejala yang dirasakan pasien kemudian menyimpulkan bahwa pasien menderita penyakit tertentu dan menentukan obat yang tepat. Pusing, demam, batuk, dan pilek merupakan gejala atau tanda dari penyakit flu. Apabila dokter hanya memberi obat sakit kepala, maka penyakit flu tidak akan sembuh. Satu masalah mungkin memiliki satu gejala tetapi mungkin juga

lebih (perhatikan Gambar 1.10).

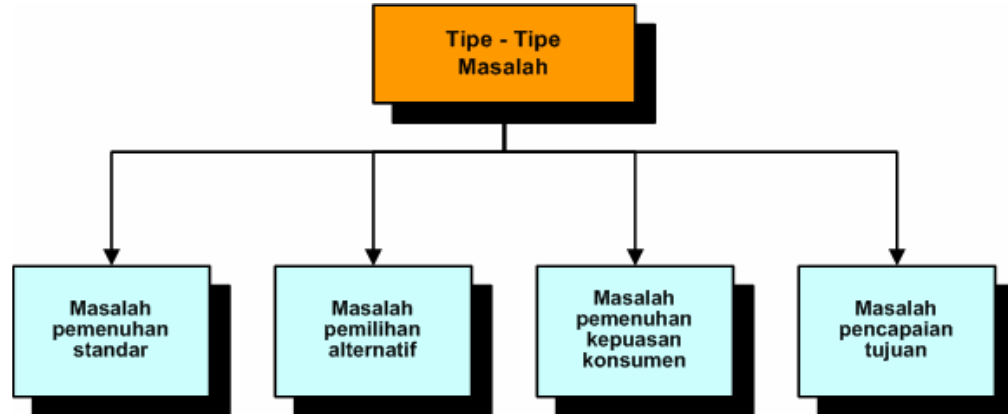


Gambar 1.10. Gejala dan masalah.

Mungkin kita bertanya-tanya apa hubungan masalah dan gejala dengan RPL. Seperti telah disampaikan di awal bab, perangkat lunak yang merupakan hasil dari RPL merupakan alat bantu yang digunakan untuk menyelesaikan tugas / masalah tertentu. Apabila kita tidak mengetahui dengan benar masalahnya mustahil kita dapat menentukan bagaimana menyelesaikannya. Dan, untuk mengetahui dengan baik masalah, maka pengetahuan tentang gejala dari masalah menjadi sangat penting.

1.8.2. Tipe-tipe Masalah

Masalah dapat dikelompokkan seperti pada Gambar 1.11.



Gambar 1.11. Tipe-tipe masalah (Deek et al, 2005).

- Masalah pemenuhan standar

Tipe masalah dalam kelompok ini adalah masalah-masalah yang berhubungan dengan pencapaian standar yang telah ditentukan dalam

sebuah organisasi. Biasanya tujuan seperti ini berlaku dalam jangka yang relative panjang.

- Masalah pemilihan alternative

Masalah dalam kelompok ini berhubungan dengan bagaimana memilih solusi terbaik dari berbagai alternative berdasarkan kriteria-kriteria tertentu. Permasalahan ini seringkali kita jumpai dalam kehidupan sehari-hari, seperti bagaimana memilih sekolah yang tepat, memilih lokasi tempat tinggal, memilih bidang pekerjaan. Masing-masing alternatif dan kriteria memiliki bobot yang telah disepakati.

- Masalah pemenuhan kepuasan konsumen

Pada organisasi-organisasi yang bersifat profit (mencari keuntungan), masalah-masalah pada kelompok ini merupakan tipe yang seringkali muncul. Konsumen memiliki berbagai macam keinginan yang satu sama lain berbeda. Memenuhi seluruh keinginan konsumen sangat tidak mungkin dan sangat memberatkan sebuah organisasi. Oleh karena itu perlu dicari pemecahan yang sama-sama menguntungkan, baik bagi konsumen maupun organisasi tersebut.

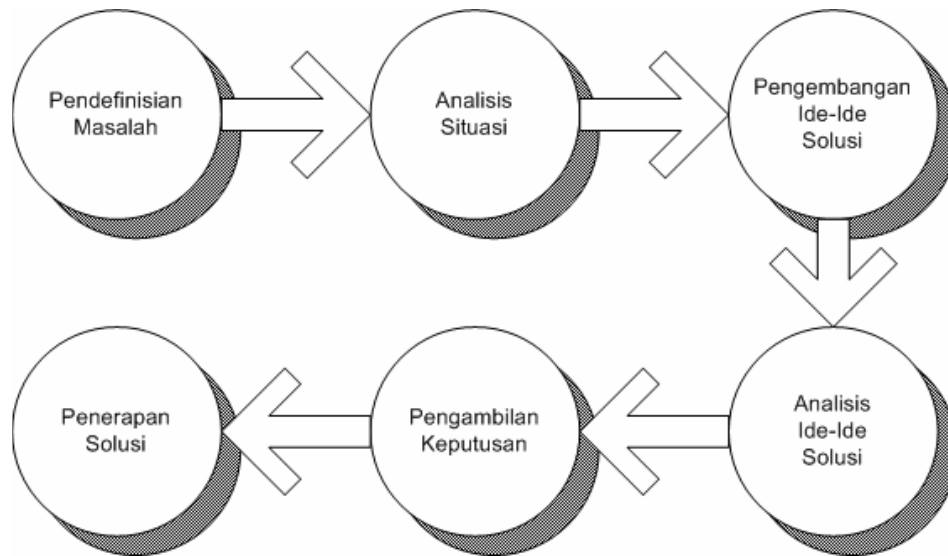
- Masalah pencapaian tujuan

Tipe ini mirip dengan tipe pertama (masalah pemenuhan standar). Yang berbeda adalah, pada tipe ini tujuan yang ingin dicapai dapat berubah-ubah dan bersifat jangka pendek.

1.8.3. Pemecahan Masalah

Pemecahan masalah adalah *sebuah proses dimana suatu situasi diamati kemudian bila ditemukan ada masalah dibuat penyelesaiannya dengan cara menentukan masalah, mengurangi atau menghilangkan masalah atau mencegah masalah tersebut terjadi*. Ada banyak urutan proses pemecahan masalah yang diajarkan oleh para ahli, salah satunya seperti terlihat pada Gambar 1.12.

Pada gambar 1.12 terlihat serangkaian tahapan proses yang berbeda yang dapat digunakan dalam berbagai tingkatan, tergantung dari tipe dan sifat masalahnya. Masalah yang berbeda membutuhkan penggunaan cara yang berbeda, bahkan mungkin urutan yang berbeda. Tahapan kritis dari proses pemecahan masalah adalah **Pendefinisian Masalah**. Apabila masalah tidak cukup jelas didefinisikan maka tahapan-tahapan berikut sulit untuk dijalankan. Bahkan apabila dipaksakan, kemungkinan besar penyelesaian yang tepat tidak akan diperoleh.



Gambar 1.12. Proses pemecahan masalah (diadopsi dari Deek et al, 2005)

Secara umum proses pemecahan masalah dapat dilakukan dengan empat tahapan utama yaitu :

- Memahami dan mendefinisikan masalah

Bagian ini merupakan bagian yang sangat penting karena menjadi awal dari seluruh proses pemecahan masalah. Tujuan pada bagian ini adalah memahami masalah dengan baik dan menghilangkan bagian-bagian yang dirasa kurang penting.

- Membuat rencana untuk pemecahan masalah

Pada bagian ini ada dua kegiatan penting yaitu :

- a) mencari berbagai cara penyelesaian yang mungkin diterapkan
- b) membuat rencana pemecahan masalah

Penyelesaian suatu masalah biasanya tidak hanya satu tapi mungkin bisa beberapa macam. Sebagai ilustrasi, apabila kita berada di kota Surabaya dan ingin pergi ke Jakarta, maka banyak cara yang mungkin bisa dilakukan, misalnya kita bisa menempuh dengan angkutan darat, laut atau udara. Dengan angkutan darat kita bisa menggunakan kereta api, bus atau angkutan yang lain. Jalurnya pun kita bisa lewat jalur utara, tengah atau selatan. Jadi banyak sekali cara penyelesaian yang bisa kita kembangkan. Masing-masing mempunyai karakteristik sendiri-sendiri. Dari sekian banyak penyelesaian ini kita harus memilih satu yang berdasarkan persyaratan tertentu merupakan cara yang paling baik untuk menyelesaikan permasalahan. Setelah terpilih, maka kita dapat membuat rencana kasar (*outline*) penyelesaian masalah dan membagi masalah dalam bagian-bagian

yang lebih kecil. Rencana kasar (*outline*) penyelesaian masalah hanya berisi tahapan-tahapan utama penyelesaian masalah.

- Merancang dan menerapkan rencana untuk memperoleh cara penyelesaian
Pada bagian ini rencana kasar penyelesaian masalah diperbaiki dan diperjelas dengan pembagian dan urutan rinci yang harus ditempuh dalam penyelesaian masalah.
- Memeriksa dan menyampaikan hasil dari pemecahan masalah
Bagian ini bertujuan untuk memeriksa apakah akurasi (ketepatan) hasil dari cara yang dipilih telah memenuhi tujuan yang diinginkan. Selain itu juga untuk melihat bagaimana daya guna dari cara yang dipilih yang dipilih.

1.9. RINGKASAN

- **Perangkat lunak** adalah seluruh perintah yang digunakan untuk memproses informasi
 - **Program** adalah kumpulan perintah yang dimengerti oleh komputer
 - **Prosedur** adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi
- RPL adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.
- Tujuan RPL adalah menghasilkan perangkat lunak dengan kinerja tinggi, tepat waktu, berbiaya rendah, dan multiplatform.
- RPL merupakan sub bidang ilmu komputer yang dalam penerapannya membutuhkan dukungan baik dari sub bidang ilmu komputer lainnya maupun bidang-bidang ilmu lain.
- Sertifikasi untuk bidang RPL belum tersedia, namun mengacu pada bidang Programmer
- **Masalah (problem)** adalah perbedaan antara kondisi yang terjadi dan kondisi yang diharapkan dan **Gejala** adalah tanda/petunjuk terjadinya suatu masalah.
- Tipe-tipe masalah :
 - Masalah pemenuhan standar
 - Masalah pemilihan alternatif
 - Masalah pemenuhan kepuasan konsumen
 - Masalah pencapaian tujuan

- **Pemecahan masalah** adalah sebuah proses dimana suatu situasi diamati kemudian bila ditemukan ada masalah dibuat penyelesaiannya dengan cara menentukan masalah, mengurangi atau menghilangkan masalah atau mencegah masalah tersebut terjadi.
- Tahapan utama pemecahan masalah :
 - Memahami dan mendefinisikan masalah
 - Membuat rencana untuk pemecahan masalah
 - Merancang dan menerapkan rencana untuk memperoleh cara penyelesaian
 - Memeriksa dan menyampaikan hasil dari pemecahan masalah

1.10. SOAL-SOAL LATIHAN

1. Sebutkan pengertian dari perangkat lunak dan rekayasa perangkat lunak.
2. Apakah perbedaan antara program komputer dan prosedur?
3. Sebutkan lima sub bidang ilmu komputer berdasarkan pengelompokan Denning.
4. Sebutkan lima bidang ilmu lain yang erat kaitannya dengan rekayasa perangkat lunak.
5. Apakah gejala dan masalah itu?