

PERTEMUAN 16

(PEMELIHARAAN PERANGKAT LUNAK)

A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi ini mahasiswa diharapkan mampu untuk :

1. Memahami Pemeliharaan Perangkat Lunak.
2. Memahami masalah pada pemeliharaan perangkat lunak.

B. URAIAN MATERI

1. Pemeliharaan Perangkat Lunak

Pemeliharaan Pemrograman adalah keseluruhan program mengubah atau membuat program setelah diberikan kepada pembeli. Perubahan tersebut dapat berupa perubahan langsung untuk mengatasi kesalahan pengkodean atau perubahan ke arah tujuan yang lebih luas untuk mengatasi beberapa kesalahan rencana serta peningkatan penting untuk mengatasi kesalahan penentuan atau mewajibkan prasyarat baru.

Itu juga dapat diuraikan sebagai kombinasi dari berbagai aktivitas yang diselesaikan untuk mempertahankan kerangka kerja, atau bahkan memperbaikinya ke kondisi yang baik. Sekitar tahun 1970 tepatnya pada bulan April, sebuah istilah diklarifikasi untuk Inovasi Dukungan yang memasukkan makna yang lebih luas daripada makna Pemeliharaan di atas. Istilah ini disebut *theroteknologi*.

Sebagian besar kemajuan dalam item pemrograman biasanya memakan waktu sekitar 1 hingga 2 tahun, namun pada tahap pemeliharaan produk akan memakan waktu sekitar 5 hingga 10 tahun. Berikutnya adalah latihan yang terjadi selama tahap dukungan, khususnya:

- a. Menambah atau merombak dan memperbaiki item pemrograman
- b. Transformasi item dalam iklim mesin baru
- c. Memperbaiki masalah yang muncul Tujuan dari Pemeliharaan Sistem.

a. Tujuan dari Pemeliharaan Sistem :

- 1) Supaya dapat memperpanjang usia kegunaan asset pada sistem tersebut. Dalam hal ini biasanya menjadi sesuatu yang paling utama atau

penting karena kurangnya kekuatan modal dalam suatu Negara yang berkembang dalam melakukan perubahan. Sedangkan pada dinegara maju justru terkadang akan lebih menghasilkan untuk pergantian dari pada memeliharanya.

- 2) Demi menjamin adanya ketersediaan optimum peralatan.
- 3) Demi menjamin keamanan operasional pada semua peralatan yang nantinya dibutuhkan saat dalam keadaan yang darurat kapanpun itu.
- 4) Demi menjamin keselamatan orang yang memakai sarana tersebut.

1) Siklus Hidup Pemeliharaan Sistem (SMLC)

Berikut adalah Tahapan *Software Maintenance Life Cycle* (SMLC) :

- a) Mengerti dan memahami permintaan pada suatu pemeliharaan
- b) Dapat mentransformasikan permintaan pemeliharaan menjadi perubahan
- c) Menjabarkan perubahan
- d) Dapat mengembangkan perubahan
- e) Melakukan uji perubahan
- f) Melatih pengguna serta melakukan tes pada penerimaan
- g) Melakukan pengkonversian serta meluncurkan operasi
- h) Melakukan update pada dokumen
- i) Melaksanakan pemeriksaan pasca implementasi

2) Jenis Pemeliharaan Perangkat Lunak

Berikut adalah Pemeliharaan Software yang dapat digolongkan menjadi empat jenis, yaitu :

a) Pemeliharaan Korektif (*Corrective Maintenance*)

Pemeliharaan korektif merupakan sebuah bagian dari pemeliharaan sistem yang tidak terlalu tinggi nilainya dan justru lebih membebani, karena pada pemeliharaan ini mengkoreksi kesalahan-kesalahan yang ditemukan pada saat sistem sedang berjalan.

Pada umumnya pemeliharaan korektif ini meliputi kondisi yang penting atau bahkan bahaya yang memerlukan tindakan segera. Juga memiliki kemampuan untuk mengenali serta memperbaiki kesalahan atau kesalahan fungsi dengan tanggap dan cepat merupakan sesuatu yang sangatlah berharga bagi perusahaan.

b) Pemeliharaan Adaptif (*Adaptive Maintenance*)

Pemeliharaan adaptif biasanya dilaksanakan demi menyesuaikan perubahan pada lingkungan data ataupun pemrosesan juga memenuhi segala persyaratan pada pemakai baru.

Dinamik merupakan lingkungan tempat sistemnya melakukan pengoperasian, dengan demikian, sistem harus terus dapat merespon perubahan persyaratan pemakai. Contohnya, Undang-Undang Perpajakan yang baru mungkin membutuhkan sebuah perubahan pada kalkulasi pembayaran bersih. Pada umumnya pemeliharaan adaptif ini baik serta tak bisa dihindari.

c) Pemeliharaan Perfektif (*Perfective Maintenance*)

Pemeliharaan perfektif ini disebut juga pemeliharaan penyempurnaan yang mempertinggi cara kerja serta maintainabilitas (kemampuan untuk dipelihara). Tindakan ini seringkali memungkinkan sistem supaya dapat memenuhi segala persyaratan pemakai yang sebelumnya justru tidak dikenal.

Saat membuat suatu perubahan substansial pada modul apapun, petugas pemeliharaan akan memakai kesempatan tersebut untuk melakukan upgrade kode, melakukan pergantian pada cabang-cabang yang telah kadaluwarsa, melakukan perbaikan kecerobohan, serta melakukan pengembangan dokumentasi.

d) Pemeliharaan Preventif (*Preventive Maintenance*)

Pemeliharaan Preventif sendiri terdiri dari inspeksi periodik serta pemeriksaan sistem untuk mengungkap dan mengantisipasi segala masalah.

Pada pemeliharaan ini personil pemeliharaan sistem bekerja pada sistem ini, maka mereka sangat sering mendapatkan cacat-cacat (bukan kesalahan yang sebenarnya) yang mengisyaratkan sebuah permasalahan potensial. Sementara tidak membutuhkan tindakan segera, cacat tersebut jika tidak diperbaiki sedari awal, maka sangat jelas akan mempengaruhi sistem baik itu fungsi maupun kemampuan dalam memeliharanya pada waktu dekat.

b. Kebutuhan Dukungan Pemrograman

- 1) Kerangka kerja juga memerlukan pemeliharaan untuk alasan yang berbeda, termasuk:
- 2) Kerangka kerja harus ditingkatkan, umumnya karena perluasan kebutuhan klien.
- 3) Kerangka mengalami perubahan karena tuntutan baru dari klien atau klien.
- 4) Kerangka mengalami perubahan karena perubahan iklim luar (perubahan bisnis).
- 5) Kerangka tersebut memiliki kesalahan yang belum dibedakan sebelumnya, sehingga kesalahan dalam kerangka harus diperbaiki.

Biaya untuk mengikuti sistem sering diabaikan, meskipun biaya untuk mengikuti sistem sangat besar. Biaya pemeliharaan secara teratur lebih tinggi daripada biaya perbaikan, untuk alasan apa hal ini terjadi?, berikut adalah beberapa faktor yang menyebabkan biaya dukungan lebih tinggi, khususnya:

a) Stabilitas tim

Secara umum, grup pengembangan dan grup pendukung adalah individu yang berbeda karena grup pengembangan biasanya sekarang memiliki tugas lain untuk dihancurkan sehingga grup pemeliharaan tidak terlalu terbiasa dengan kerangka kerja yang dikembangkan. kelompok telah membuat.

b) Tanggung jawab kontrak

Kesepakatan yang diajukan untuk insinyur dan pemeliharaan biasanya diisolasi atau diberikan oleh berbagai organisasi dan bahkan tidak oleh organisasi desainer pertama, oleh karena itu tidak ada motivasi bagi desainer untuk membuat kerangka kerja yang tidak sulit untuk diubah.

c) Keahlian staff

Staf pendukung biasanya tidak terlatih dalam pemeliharaan pemrograman dan staf pendukung pada umumnya dianggap tidak memiliki kemampuan dukungan.

d) Usia dan struktur program

Proyek-proyek lama sebagian besar memiliki desain yang telah direndahkan oleh kesempatan, sehingga sulit untuk dipahami oleh kelompok pemeliharaan. Dalam dukungan pemrograman, beberapa masalah sering muncul, termasuk:

- (1) Tidak ada dokumentasi yang baik terhadap software yang dikembangkan.
- (2) Sulit untuk memahami program yang dibuat oleh orang lain atau pengembang.
- (3) Tidak adanya narasumber untuk menanyakan tentang software yang dibuat oleh pengembang.
- (4) Kesulitan pelacakan kesalahan pada proses pengembangan software.
- (5) Kebanyakan software yang dikembangkan dirancang tanpa adanya pemikiran untuk diubah.

Keperluan Pemeliharaan Perangkat Lunak

Sistem harus dipelihara karena hal-hal berikut ini, yaitu :

- (1) Biasanya sistem mempunyai kesalahan yang belum terdeteksi sejak dulu, maka kesalahan pada sistem tersebut haruslah segera diperbaiki
- (2) Mendapatkan beberapa perubahan pada sistem dikarenakan permintaan baru dari si pemakai sistem
- (3) Karena terjadinya perubahan lingkungan luar (perubahan bisnis), maka sistem haruslah diubah
- (4) Agar menjadi lebih baik, maka sistem sangat perlu untuk ditingkatkan lagi kedepannya

c. Manfaat Pemeliharaan Perangkat Lunak yaitu sebagai berikut :

- 1) Dipastikan kesesuaiannya dengan kebutuhan fungsionalitas teknis software
- 2) Seluruh kebutuhan pihak manajerial terkait budget yang akan dikeluarkan serta jadwal harus disesuaikan
- 3) Meningkatkan aktifitas pemeliharaan serta yang paling penting ialah mampu meningkatkan efesiensi software

Pada kasus biaya pemeliharaan sistem kebanyakan sering sekali diabaikan. Padahal pada kenyataannya biaya pemeliharaan sistem adalah biaya yang memerlukan budget cukup besar. Bahkan biaya pemeliharaan perangkat lunak selama kurang lebih 25 tahun belakangan ini terus menerus mengalami kenaikan. Biasanya kebanyakan perusahaan melakukan pembelanjaan sekitar 80% atau bahkan lebih dari anggaran sistem mereka terkait pemeliharaan perangkat lunak.

Lalu mengapa biaya pemeliharaan justru lebih tinggi dari pada biaya pengembangannya, berikut ini merupakan beberapa faktor yang menjadi penyebabnya :

- a) Kestabilan pada Tim, pada umumnya tim pengembang dan tim pemelihara merupakan orang yang tidak sama atau berbeda karena biasanya tim pengembang sudah menuju ke proyek baru lainnya yang mengakibatkan tim pemeliharanya terkadang tidak memahami pada sistem yang dikembangkan.
- b) Tanggung Jawab pada Kontrak, biasanya kontrak untuk si pengembang dan si pemelihara akan terpisah atau akan diberikan untuk perusahaan yang tidak sama atau berbeda yang bahkan bukan pengembang dari sistem aslinya, sehingga mengakibatkan tidak adanya insentif untuk si pengembang membuat sistem yang mudah untuk diubah.
- c) Keahlian Staff, kebanyakan staff pemelihara tidak memiliki pengalaman yang besar terkait dengan pemeliharaan software serta staff pemelihara sering kali dinilai untuk tidak membutuhkan keahlian yang mendalam pada bidang software.
- d) Usia serta Struktur Program, karena program yang sudah sangat lama atau tua maka biasanya strukturnya sudah mengalami degradasi dari perkembangan jaman sehingga mengakibatkan sangat sulitnya untuk dipahami oleh si pemelihara.

d. Hukum Lehman

Tindakan yang telah dirumuskan oleh Lehman dan Balady ialah mengenai tentang perawatan (*maintenance*) serta evolusi pada software, yaitu :

- 1) Transformasi yang berkelanjutan, yang dimana suatu software wajib mendapatkan perubahan supaya tetap bisa dipergunakan
- 2) Kerumitan atau kesulitan yang meningkat, yaitu jika suatu software semakin berevolusi maka akan semakin meningkat juga tingkat kerumitannya, kecuali kita memiliki tekad dan usaha untuk mengembangkannya
- 3) Perkembangan program yang besar
- 4) Kesetimbangan organisasi
- 5) Keamanan familiaritas
- 6) Pertumbuhan yang berkelanjutan, yaitu fungsi pada suatu software dapat terus bertumbuh supaya pengguna tetap merasa nyaman
- 7) Keunggulan yang menyusut, yaitu kualitas pada suatu software akan menyusut seiring berjalannya waktu, kecuali kita memiliki tekad serta upaya demi merawat serta beradaptasi pada lingkungan sekitarnya
- 8) Sistem umpan balik (*feedback system*).

2. Masalah Pemeliharaan Software

Berikut ini adalah contoh dari beberapa permasalahan yang sangat sering ada pada pemeliharaan software, yakni :

- a. Pada versi yang terdahulu akan mengalami sebuah kesulitan dalam melaksanakan pelacakan evolusi software
- b. Pada proses pengembangan software juga terjadinya sebuah kesulitan dalam melaksanakan pelacakan
- c. Cukup sulit untuk memahami program lain
- d. Sering kali tidak adanya dokumentasi yang cukup baik
- e. Narasumbernya tidak ada
- f. Kebanyakan tidak adanya pemikiran untuk dapat melakukan perubahan pada software yang telah dirancang.

a. Masalah Teknis

- 1) *Limited Understanding* atau pemahaman terbatas, yaitu biasanya insinyur perangkat lunak baru diharuskan bisa pandai dalam memahami aplikasi ataupun program yang telah dirancang pengembang lainnya, supaya

apabila terjadinya kesalahan maupun adanya peningkatan fungsi atau karakteristik maka bisa segera diperbaiki

- 2) *Testing*, yaitu dilakukannya pengujian terhadap software demi menemukan sebuah bug, hal ini adalah kegiatan yang dilaksanakan dalam perawatan software
- 3) *Impact Analysis*, yaitu melakukan Analisa pada pengaruh yang didapatkan pada saat penambahan fitur, karena jangan sampai ketika penambahan fitur justru membuat fungsinya tidak berfungsi atau bahkan membuat program error
- 4) *Maintainability*, yaitu dilakukannya perawatan terhadap software agar dapat menjaganya dari sebuah kerusakan, dan juga meng-update segala fiturnya agar tidak tertinggal dengan software baru yang bermunculan.

b. Masalah Manajemen

- 1) *Alignment with Organizational Objectives*, yaitu suatu perusahaan atau organisasi yang bergerak dibidang Teknik Informatika. Memberikan pekerjaan bagi pekerjaanya mengikuti peranannya masing-masing serta keperluan dari si pemelihara yang harus diikuti seperti kebutuhannya, akan tetapi disebabkan banyaknya pekerjaan yang dilaksanakan perproyek. Maka deadline rencana yang memiliki waktu tenggang kebanyakan menjadikan industry tersebut untuk tidak mengadakan pemeliharaan
- 2) Kepegawaian, yaitu merupakan pekerja pemelihara (pegawai dengan pekerjaannya melakukan perawatan pada software) pekerjaan ini seringkali dianggap enteng karena kalah terkenal dibandingkan developer. Sehingga jika mendapatkan seorang staff yang bisa melakukan pekerjaan sebagai maintener, sepatutnya diberikan penghasilan serta fasilitas yang lebih baik, agar tidak berhenti bekerja
- 3) *Process*, yaitu jalan dalam melaksanakan perawatan yang cukup banyak serta bahkan cukup rumit. Proses siklus hidup perangkat lunak merupakan suatu kegiatan, edukasi serta metode juga modifikasi dipakai demi merawat serta mengembangkan perangkat lunak. Dibutuhkannya beberapa aktifitas pada maintenance yang tidak didapatkan di software development

- 4) *Outsourcing*, yaitu pegawai diperusahaan yang mempunyai hanya sedikit kontrol pada perusahaan tersebut. Biasanya melalui perantara pihak ketiga serta bukan seorang pegawai tetap. Contoh pekerjaan outsourcing yaitu petugas call center.

c. Proses Pemeliharaan

Adalah suatu proses yang dibutuhkan demi melaksanakan perawatan terhadap software, proses perawatannya sangat banyak, dimulai dari menentukan kebutuhannya, memodifikasinya, serta mereview kesesuaiannya, berikut ini ialah beberapa langkah dalam proses perawatan software, yakni :

Perawatan software dimulai dari melakukan penentuan permasalahan serta penguraian dalam transformasi. Kemudian melakukan analisa pada kesalahan serta menetapkan perubahan apa yang akan dibutuhkan untuk diganti, kemudian tahap berikutnya yaitu melaksanakan penerapan terhadap perubahan yang dikerjakan. Penerapan perubahan itu membutuhkan suatu komunikasi dari pengembang serta analis, selanjutnya sesudah dilaksanakannya penerapan yaitu melaksanakan evaluasi serta penerimaan tes pada penerapan tersebut dan ini merupakan langkah akhir.

Proses perawatan software ini dikerjakan secara berkelanjutan mengikuti kebutuhannya. Secara umum perawatan software dilaksanakan dengan mengimplementasikan fitur, kemudian melaksanakan perpindahan ke fitur baru. Dan yang terakhir jika perangkat lunak itu tak bisa lagi membantu keperluan dari si pemakai atau konsumen, berikut ini adalah tipe atau bentuk dalam metode perawatan perangkat lunak, yaitu :

- 1) *Quick-Fix Model*
- 2) *Spiral Model*
- 3) *Osborne's Model*
- 4) *Iterative-Enhancement Model*
- 5) *Full-Reuse Model*

1) Kegiatan-kegiatan yang ada pada Pemeliharaan

- a) Pemahaman Program, yaitu suatu kegiatan yang dibutuhkan demi mendapatkan sebuah ilmu atau wawasan umum mengenai software,

seperti mekanismenya seperti apa serta bagaimana komponen yang ada didalamnya saling berinteraksi

- b) Transisi, yaitu suatu proses penyampaian yang tertata serta terstruktur melalui cara yang benar dalam seluruh kegiatan dimana bertautan dengan software, dari si developer kepada si pemelihara
 - c) *Acceptance/Rejection* permohonan Pemeliharaan, yaitu permohonan pemeliharaan bisa saja tidak diterima oleh si pemelihara kemudian dipindahkan kepada si developer
 - d) Analisis Pengaruh, yaitu melaksanakannya metode analisis pengaruh pada modifikasi pada software
 - e) Perjanjian Tingkat-Layanan Pemeliharaan dan Lisensi Pemeliharaan serta Kontrak, yaitu kontrak serta persyaratan persetujuan.
- 2) Teknik Pemeliharaan
- a) Yang pertama ada Pemahaman Program, yaitu seorang pemrogram akan menggunakan sebanyak mungkin waktunya demi memahami aplikasi yang nantinya akan diganti dengan cara membaca, serta melakukan pengarsipan dengan singkat serta jelas bisa sangat amat meringankan pada saat proses pemahaman pada program
 - b) Yang kedua ada Rekayasa Balik, yaitu melakukan analisis software demi mengenali elemen-elemen pada software serta keterkaitan yang ada didalamnya demi menciptakan perwakilan atau cabang perangkat lunak pada wujud yang berbeda dan dalam tingkatan generalisasi atau penyamarataan dimana harus semakin baik dari sebelumnya, rekayasa balik ini juga memiliki karakter yang kurang aktif yaitu enggan menghasilkan software yang mutakhir atau mengubahnya, berikut ialah contoh dari rekayasa balik yaitu : pengarsipan ulang.
 - c) Yang ketiga ada Rekayasa Ulang, diartikan yaitu seperti penyelidikan serta modifikasi atau transformasi software melalui cara mengatur atau mengurutkannya lagi sebagian bahkan seluruhnya terkait mengenai software tersebut dengan tidak mengganti kegunaannya, kebanyakan dipakai demi mengganti software yang sudah “tua”, bukan untuk meningkatkan kinerja perangkat.

3) *Reengineering 2* : Keuntungan

- a) Tarif tidak terlalu besar bahkan dapat diminimalisir, pengeluaran rekayasa ulang sangat sering tidak terlalu banyak dibandingkan dengan pengeluaran *software development* yang baru.
- b) Mengurangi risiko, kemungkinan adanya permasalahan pengembangan, masalah staff dan masalah spesifikasi, menjadikannya pengembangan perangkat lunak baru memiliki yang risiko yang lebih tinggi.

4) *Migration* atau Migrasi

- a) Dilaksanakan apabila terjadinya transformasi lingkungan yang fungsinya demi mengendalikan *software*.
- b) Pemelihara diharuskan memahami aktifitas pada perosedur perpindahan

5) Migrasi 2 : Aktifitas Tambahan

- a) Pemberitahuan Tujuan atau Makdus, yaitu cetusan yang menyimpan suatu argumentasi dimana kawasan yang telah using tak mendapatkan bantuan lagi serta didukung melalui uraian daerah termutakhir serta tangga tersedianya
- b) Operasi Paralel, yaitu agar migrasi berjalan mulus dan lancer, maka disediakan lingkungan lama dan baru
- c) Pemberitahuan Komplit atau Selesai, yaitu laporan yang dimana segera diberikan kepada bidang yang tehubung, saat migrasi selesai
- d) Ulasan Pasca Operasi, yaitu evaluasi ketika sudah dilakukannya operasi paralel serta pengaruh metamorfosis melahirkan daerah mutakhir
- e) Arsip Data atau Informasi, yaitu penyimpanan informasi *software* yang telah usang.

6) Masa Pensiun

- a) Ketika perangkat lunak telah sampai pada akhir kegunaan, maka harus dipensiunkan

- b) Supaya dapat mendukung ketika dibuatnya suatu ketentuan purnakarya atau masa pensiun pada suatu perangkat lunak maka dilakukannya penelitian
- c) Analisis tersebut sangat perlu dituangkan kedalam berkas perencanaan pension, dimana meliputi syarat, dampaknya, penggantian, jadwal, serta effort.

7) *Software Maintenance Tools* atau Alat Pemeliharaan Perangkat Lunak

- a) *Program Slicers*, yaitu dapat melakukan pemilihan terhadap program atau aplikasi yang mendapatkan dampak transformasi
- b) Analisa Statis, yaitu memberikannya pengertian konvensional serta skema singkat yang ada dalam aplikasi atau program tersebut
- c) Analisa Dinamis, yaitu mendukung dalam melakukan pelacakan lintasan penyelesaian pada aplikasi atau program
- d) Analisa Aliran Data, yaitu melakukan dukungan untuk mencari sejumlah informasi
- e) Referensi Silang yaitu menciptakan indikator elemen pada aplikasi atau program
- f) Analisis Ketergantungan, yaitu mendukung melakukan analisis serta mengetahui segala korelasi atau kedekatan dari para elemen pada sebuah aplikasi atau program.

8) Kunci Masalah pada Pemeliharaan Perangkat Lunak

Terdapat empat masalah terkait dengan pemeliharaan perangkat lunak, yakni :

- a) Pemeliharaan dengan cara teknis
- b) Pemeliharaan secara manajerial
- c) Biaya estimasi perawatan (*Maintenance Cost Estimation*)
- d) Pengukuran perawatan (*Maintenance Measurement*)

9) Berikut ini adalah contoh dari beberapa permasalahan yang sangat sering ada pada pemeliharaan software, yakni :

- a) Pada versi yang terdahulu akan mengalami sebuah kesulitan dalam melaksanakan pelacakan evolusi software
- b) Pada proses pengembangan software juga terjadinya sebuah kesulitan dalam melaksanakan pelacakan
- c) Cukup sulit untuk memahami program lain
- d) Sering kali tidak adanya dokumentasi yang cukup baik
- e) Narasumbernya tidak ada
- f) Kebanyakan tidak adanya pemikiran untuk dapat melakukan perubahan pada software yang telah dirancang.

10) Beberapa Fakta Mengenai Pemeliharaan Perangkat Lunak :

- 1) Biaya yang harus disediakan untuk pemeliharaan suatu program ialah $\geq 40\%$ dari biaya yang dikeluarkan untuk mengembangkan program tersebut
- 2) Mengganti suatu kecacatan dan memperbaikinya akan memberikan kesempatan (20% s/d 50%) kecacatan lainnya yang bisa saja muncul (Fred Brooks, The Mythical Man-Month)
- 3) Software yang telah berhasil atau sukses juga pasti harus melakukan pemeliharaan.

d. Keperluan Pemeliharaan Perangkat Lunak

Sistem juga memerlukan pemeliharaan karena berbagai hal, diantaranya yaitu:

- 1) Sistem Perlu ditingkatkan, biasanya karena faktor peningkatan kebutuhan pengguna.
- 2) Sistem mengalami perubahan-perubahan karena permintaan baru dari pengguna atau *user*.
- 3) Sistem mengalami perubahan karena perubahan lingkungan luar (perubahan bisnis).
- 4) Sistem memiliki kesalahan yang sebelumnya belum terdeteksi, sehingga kesalahan-kesalahan pada sistem perlu diperbaiki.

Biaya pemeliharaan sistem sering kali diabaikan, padahal biaya pemeliharaan sistem memerlukan biaya yang cukup besar. Sering kali biaya pemeliharaan lebih tinggi dari biaya pengembangan, kenapa hal ini bisa terjadi?, berikut adalah beberapa faktor yang menyebabkan biaya pemeliharaan lebih tinggi, yaitu:

a) Stabilitas tim

Pada umumnya tim pengembang dan tim pemeliharaan adalah orang yang berbeda karena tim pengembang biasanya sudah mempunyai proyek baru untuk dikerjakan sehingga tim pemeliharaan tidak terlalu paham dengan sistem yang telah tim pengembang buat.

b) Tanggung jawab kontrak

Kontrak yang diajukan bagi pengembang dan pemeliharaan biasanya terpisah atau diberikan oleh perusahaan yang berbeda bahkan bukan oleh perusahaan pengembang aslinya, alibatnya tidak ada insentif bagi pengembang untuk membuat sistem yang mudah diubah.

c) Keahlian staff

Staff bagian pemeliharaan biasanya tidak berpengalaman dalam pemeliharaan software dan staff pemeliharaan biasanya dianggap tidak memerlukan keahlian dalam bidang pemeliharaan.

Umur dan struktur program

Program yang sudah tua biasanya strukturnya sudah terdegradasi oleh perkembangan zaman sehingga sangat sulit dipahami oleh tim pemeliharaan. Dalam pemeliharaan software juga sering terjadi beberapa permasalahan yang sering muncul, diantaranya yaitu:

- 5) Tidak ada dokumentasi yang baik terhadap software yang dikembangkan.
- 6) Sulit untuk memahami program yang dibuat oleh orang lain atau pengembang.
- 7) Tidak adanya narasumber untuk menanyakan tentang software yang dibuat oleh pengembang.

- 8) Kesulitan pelacakan kesalahan pada proses pengembangan software.
- 9) Kebanyakan software yang dikembangkan dirancang tanpa adanya pemikiran untuk diubah.

C. SOAL LATIHAN/TUGAS

1. Jelaskan perbedaan antara pemeliharaan adaptif dan perfektif ?
2. Berapa lama waktu yang dibutuhkan untuk kemajuan item pemograman ?
3. Apa yang dimaksud dengan heroteknologi ?
4. Sebutkan Manfaat Pemeliharaan Perangkat Lunak ?
5. Jelaskan proses perawatan software yang benar ?

D. REFERENSI

1. Pressman, R. S. (2015). Software Engineering. A Practitioner's Approach (8th ed.). New York: McGraw-Hill Education.
Sommerville, I. (2011). Software Engineering (9th ed.). Boston: Addison-Wesley.

GLOSARIUM

Siklus Hidup Pemeliharaan Sistem (SMLC) Mengerti dan memahami permintaan pada suatu pemeliharaan sistem mentransformasikan permintaan pemeliharaan menjadi pengubahan menjabarkan perubahan dapat mengembangkan perubahan.

Heroteknologi sebuah istilah diklarifikasi untuk Inovasi Dukungan yang memasukkan makna yang lebih luas daripada makna Pemeliharaan di atas

Impact Analysis, yaitu melakukan Analisa pada pengaruh yang didapatkan pada saat penambahan fitur