

BAB II

LANDASAN TEORI

2.1 Perawatan Mobil

Mobil adalah kendaraan darat yang digerakkan oleh tenaga mesin, beroda empat atau lebih (selalu genap), dengan menggunakan bahan bakar minyak seperti bensin atau solar untuk menghidupkan mesinnya.

Menurut Martin T. Teiseran (2003), konsep perawatan atau pemeliharaan mobil sudah dikenal sejak pertengahan abad 20. Kata pemeliharaan itu sendiri berasal dari bahasa Inggris "*Maintenance*". *Maintenance* itu sendiri berasal dari bahasa latin "*Manutentione*" yang berarti merawat dengan tangan. Definisi lain dari kamus yang sama adalah:

1. Perbuatan atau hasil dari penjagaan
2. Tolak ukur yang dibutuhkan untuk penjagaan atau membuat tetap suatu masalah atau situasi.
3. Perawatan teknik pada bagian yang penting, agar pengoperasian kendaraan atau mesin atau alat dapat teratur dan tetap.

Oleh karena itu dalam pengertian umum pemeliharaan adalah merawat, menjamin agar berfungsi. Dengan kata lain pemeliharaan adalah gabungan dari operasi kendaraan mobil yang bertujuan untuk mendapatkan efisiensi kendaraan yang maksimum dengan kemungkinan kerusakan yang rendah dan waktu perbaikan yang singkat. Fungsi pemeliharaan itu adalah:

1. Mempertahankan kondisi mobil secara maksimal baik tenaga dan kemampuan.
2. Mencegah terjadinya kerusakan yang fatal secara dini.

3. Meningkatkan usia pakai kendaraan.

Menurut Edi S. (2000:8), tujuan utama dari pembagian tingkat pemeliharaan ini adalah untuk membuat pekerjaan pemeliharaan lebih rasional sehingga lebih ekonomis serta rendah biaya pelaksanaannya. Setelah pembagian akan dilanjutkan dengan klasifikasi tingkat pemeliharaan mobil. Klasifikasi menjadi acuan dasar untuk setiap perawatan mobil khususnya bagian mesin mobil.

Klasifikasi tingkat pemeliharaan terdiri dari :

1. Pemeliharaan Dasar

Pemeliharaan dasar atau pemeliharaan harian pada intinya perawatan pertama yang harus dilakukan pengemudi ataupun teknisi khusus yang ditunjukkan untuk pekerjaan tersebut. Perawatan ini dilakukan secara menyeluruh sebelum atau sesudah kendaraan beroperasi.

Pada mobil pemeliharaan ini dilakukan saat mobil akan dioperasikan dengan cara mengecek kondisi ban, rem, lampu depan belakang, dan *wiper* atau pembersih air yang melekat pada kaca.

2. Pemeliharaan Berkala

Pemeliharaan periodik atau berkala, dalam bahasan lain berulang-ulang dan terprogram. Pemeliharaan *preventif* adalah rangkaian perencanaan pekerjaan melalui "Perencanaan Pemeliharaan". Dalam hal ini pihak pabrikan kendaraan telah menetapkan pekerjaan yang harus dilakukan agar diperoleh penampilan kendaraan yang selalu prima dan siap pakai. Termasuk dalam pemeliharaan ini pada kondisi mobil baru biasanya pemeliharaan dilakukan setiap 1.000 km

& 5.000 km dilakukan oleh pihak dealer, selanjutnya pemeliharaan dilakukan setiap 5.000 km, 10.000 km, 20.000 km, dan seterusnya.

3. Pemeliharaan Korektif

Pemeliharaan korektif adalah pemeliharaan perbaikan komponen mekanis, pergantian suku cadang yang rusak. Perbaikan ini tidak direncanakan terlebih dahulu kemudian dilakukan begitu diketahui adanya kerusakan serta proses perbaikan dalam waktu yang singkat, biaya rendah, dan kualitas yang baik. Pada mobil pemeliharaan ini termasuk kategori rumit, karena umumnya pemeliharaan korektif dapat diketahui ketika melakukan pemeliharaan dasar atau pemeliharaan *preventif*. Jika kelainan tersebut tidak ditemukan ketika pemeriksaan dasar dan *preventif*, maka kerusakan terjadi pada saat kendaraan beroperasi.

4. Pemeliharaan Menyeluruh

Pemeliharaan ini termasuk pemeliharaan yang dilakukan secara menyeluruh (bongkar pasang atau *overhaul*) untuk unit (*Assy*), perlengkapan mekanis, *body*, sasis, dan lain-lain. Dengan tujuan rasional dan mempercepat proses perbaikan maka sebaiknya bengkel yang melaksanakan pemeliharaan ini harus menyediakan komponen cadangan dalam bentuk unit (*Assy*). Dengan cara ini komponen yang rusak dapat ditukar dengan cepat dan kendaraan dapat segera beroperasi kembali sehingga kendaraan tidak menunggu terlalu lama untuk perbaikan komponen yang rusak. Contoh komponen cadangan: Unit Mesin (*Alternator*), Unit Transmisi (*Motor Stater*), Unit Diferensial (*Steering GearBox*), dan lain-lain. Penentuan jadwal dan strategi perawatan atau

maintenance ada baiknya di rancang sesuai tipikal operasi dan kemampuan bengkel atau *workshop* dengan tidak mengurangi kualitas pekerjaannya.

2.2 Pelayanan Jasa Perawatan Mobil

Menurut Daryanto (2002:45), pelayanan jasa perawatan mobil terdiri dari berbagai pekerjaan dan tugas yang harus dilakukan untuk menjaga kegiatan perawatan dengan penanganan mobil berjalan aman dan efisien. Setiap bagian yang terlibat dengan fungsinya masing-masing saling berkomunikasi dengan intensif dalam pekerjaan pelayanan jasa perawatan mobil.

Beberapa tugas dan kewajiban yang dilakukan oleh staf perawatan mobil adalah:

- a. Pembuatan laporan.
- b. Menjaga tren grafik kehandalan sistem kerja mobil.
- c. Mengatur dan menjalankan daftar pustaka, serta data yang berhubungan dengan perawatan mobil.
- d. Mengeluarkan perintah kerja dan merilis formulir.
- e. Menjalankan tugas administrasi seperti dokumetasi dan pencetakan.
- f. Pembuatan laporan hasil perawatan mobil dan korespondensi.
- g. Mempertahankan mesin mobil dan catatan yang terkait agar mudah dilakukan perawatan.

Lingkungan kerja untuk pekerjaan pelayanan jasa perawatan mobil biasanya merupakan lingkungan kantor yang bersih dan nyaman. Tempat kerja bervariasi tergantung tempat mereka ditugaskan. Tugas mereka memerlukan kerjasama yang erat antar sesama pekerja pelayanan jasa perawatan mobil di tiap bagian yang berbeda-beda fungsinya.

Dokumen dan *form* yang ada pada kegiatan pelayanan jasa perawatan mobil Autofocus antara lain:

- a. *Checklist*, berisi *work order* dari *customer service*, di dalamnya terdiri dari beberapa *repair order* atau biasa disebut *subject*.
- b. Perintah Kerja Bengkel (PKB) adalah paket atau kumpulan pekerjaan yang harus dilakukan oleh *mekanik* dalam proyek perawatan mobil. PKB terdiri dari dua jenis kegiatan. Pertama adalah kegiatan dalam bidang jasa yang kedua adalah *sparepart* (suku cadang), tergantung sang pemilik atau penyeton mobil ingin menambah perbaikan pada mobilnya. Biaya yang diberikan bervariasi mulai dari asuransi terkait atau tanggungan pribadi. Setelah surat perintah keluar maka langsung diserahkan ke mekanik.
- c. Estimasi, adalah tugas dari divisi *estimator* yang bertujuan untuk menentukan harga suku cadang maupun jasa, yang akan diajukan ke pihak asuransi sebagai acuan harganya. Kemudian dari pihak asuransi yang mengeluarkan surat persetujuan.
- d. Penagihan, pembelian, dan gudang. Dalam penelitian ini maksud dari penagihan adalah, proses penentuan biaya yang sudah dibuat oleh *estimator* yang disetujui oleh pihak asuransi yang terkait. Selanjutnya pembelian adalah pemesanan barang ke pihak *supplier*.
- e. *Checkout*, merupakan proses yang dikerluarkan oleh bagian *customer service* sebagai tanda bahwa suatu proyek perawatan mobil telah selesai. *Checkout* dibuat dan dipertanggungjawabkan secara penuh oleh *customer service*. *Checkout* diberikan kepada *customer* saat proyek perawatan telah selesai.

2.3 Sistem

Menurut Fitz Gerald dalam Jogiyanto (2005: 1), suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan Hall (2007: 6), mengatakan bahwa sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama. Banyak komponen yang dimaksud adalah sebuah sistem harus berisi lebih dari satu bagian.

Menurut Kristanto (2003: 2), sistem adalah kumpulan elemen – elemen dan bekerja sama untuk memproses masukan atau *input* yang ditunjukkan kepada sistem tersebut dan mengolah input tersebut sampai menghasilkan keluaran atau *output* yang diinginkan. Adapun penjelasan tentang elemen – elemen dari sistem adalah:

1. Tujuan, sistem dapat berupa tujuan usaha, kebutuhan pemecahan masalah, dan lain sebagainya.
2. Batasan, merupakan batasan – batasan yang ada dalam mencapai tujuan dari sistem, yang dapat berupa peraturan – peraturan, permasalahan yang dibahas, peralatan, personil, dan lain sebagainya.
3. Penghubung, penghubung merupakan media antara satu subsistem dengan subsistem lain sehingga *output* (keluaran) dari subsistem akan dapat menjadi *input* (masukan) bagi subsistem lain.
4. *Input* (masukan), merupakan bagian yang bertugas untuk menerima data masukan, frekuensi pemasukan data dan jenis pemasukan data.

5. Proses, merupakan bagian yang memproses masukan data menjadi informasi yang sesuai dengan keinginan penerima.
6. *Output* (keluaran) merupakan keluaran atau tujuan akhir dari sistem yang dapat berupa laporan, tabel atau grafik.

Tujuan umum dari suatu sistem adalah menghubungkan berbagai bagian dari sistem tersebut. Meskipun tiap bagian berfungsi secara *independen* dari yang lainnya, semua bagian tersebut melakukan tujuan yang sama. Jika komponen tertentu tidak memberikan kontribusinya pada tujuan bersama, maka komponen tersebut bukanlah bagian dari sistem.

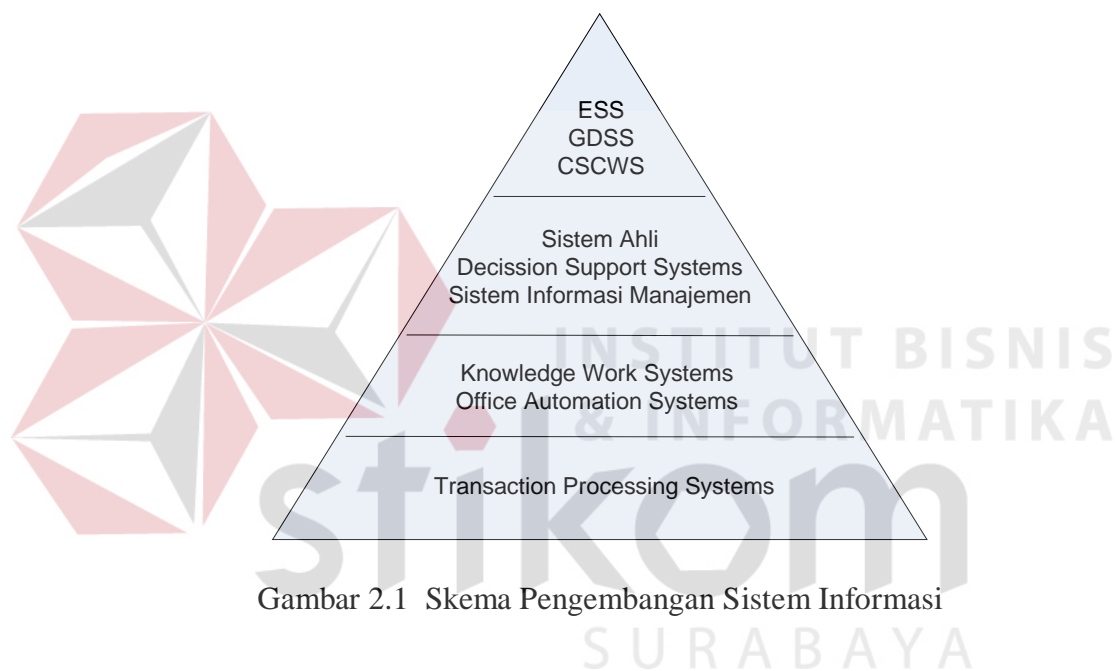
2.4 Sistem Informasi

Menurut Leitch dan Davis dalam Jogiyanto (2005:11), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sedangkan Gelinas, Oram dan Wiggins dalam Kadir (2003:11) mendefinisikan sistem informasi sebagai suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.

Komponen dari sistem informasi adalah *hardware*, *software*, data, manusia dan prosedur. Kegiatan dari suatu sistem informasi mencakup kegiatan *input*, *proses*, *output*, penyimpanan dan *control*.

Sistem informasi dapat dikembangkan menjadi beberapa jenis, dengan tujuan yang berbeda-beda tergantung pada kebutuhan bisnis (Kendall, 2003:2).

Jenis-jenis sistem tersebut diantaranya adalah, *Transaction Processing Systems* (TPS), *Office Automation Systems* (OAS), *Knowledge Work Systems* (KWS), *Manajemen Information Systems* (MIS), *Decission Support Systems* (DSS), *Artificial Intelligent* (AI), *Computer Supported Collaborative Work Systems* (CSCWS), *Group Decission Support Systems* (GDSS) dan *Executive Support Systems* (ESS). Skema pengembangan sistem informasi dapat dilihat pada Gambar 2.1.



Gambar 2.1 Skema Pengembangan Sistem Informasi

Desain sistem adalah tahap setelah analisis dari siklus pengembangan sistem pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi, menggambarkan bagaimana suatu sistem dibentuk.

“Pada tahap desain secara umum, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan dengan pemakai sistem, bukan pemrogram. Komponen sistem informasi yang didesain adalah model, *input*, *ouput*, teknologi dan kontrol” (Jogiyanto, 2003:211).

Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system* dan *logical model*. Bagan alir sistem (*system flowchart*) merupakan alat yang tepat untuk menggambarkan *physical system*. Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya seperti simbol terminal, *harddisk*, dan laporan-laporan.

Logical model dari sistem informasi lebih menjelaskan kepada pemakai sistem bagaimana nantinya fungsi-fungsi pada sistem informasi secara logika akan bekerja. *Logical model* dapat digambarkan dengan diagram arus data (*data flow diagram*). Arus data pada *data flow diagram* dapat dijelaskan dengan kamus data atau *data dictionary*. Sketsa dari *physical system* dapat menjelaskan kepada pemakai sistem bagaimana nantinya sistem secara fisik akan diterapkan.

Maka dari itulah pada akhirnya *physical system* dan *logical model* sangat diperlukan di tahap desain sistem ini, karena sangat berguna untuk menjelaskan kepada pemakai, pemrogram dan ahli teknik yang terlibat tentang kerja sistem.

2.5 Rekayasa Perangkat Lunak

Menurut Yasin (2012:2), 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.

Pengertian Rekayasa Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dengan

pengertian ini jelaslah bahwa Rekayasa Perangkat Lunak 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.

2.5.1 Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan Rekayasa Perangkat Lunak tidak berbeda dengan bidang rekayasa yang lain. Hal ini dapat kita lihat pada Gambar 2.2 di bawah ini.



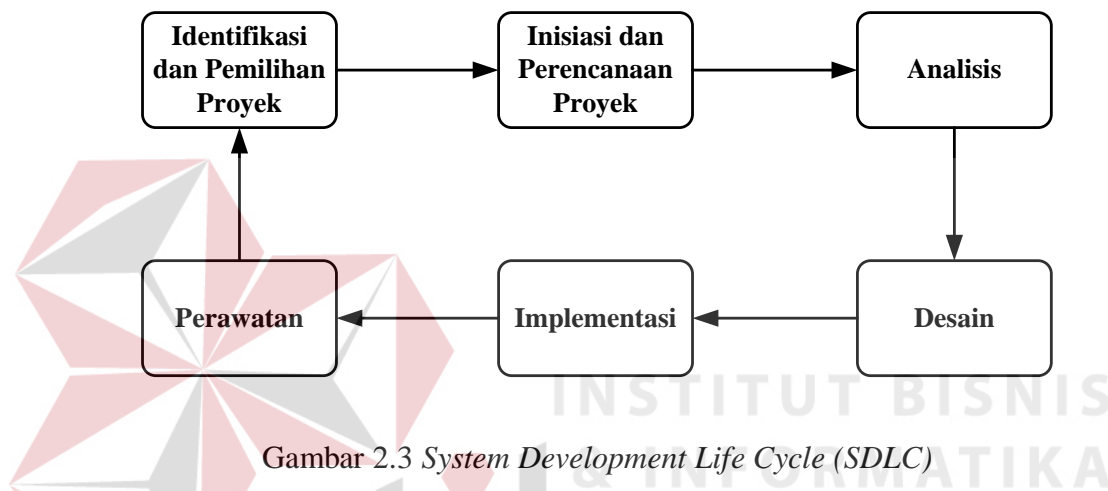
Gambar 2.2 Tujuan Rekayasa Perangkat Lunak

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

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal, dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah.

2.5.2 Metode Rekayasa Perangkat Lunak

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 2.3 berikut ini.



Gambar 2.3 *System Development Life Cycle (SDLC)*

Penjelasan dari Gambar 2.3 diatas adalah:

1. Kebutuhan terhadap definisi masalah yang jelas. *Input* utama dari setiap model pengembangan perangkat lunak adalah pendefinisian masalah yang jelas.
2. Tahapan-tahapan pengembangan yang teratur. Meskipun model-model pengembangan perangkat lunak memiliki pola yang berbeda-beda, biasanya model-model tersebut mengikuti pola umum *analysis – design – coding – testing – maintenance*.
3. *Stakeholder* berperan sangat penting, dapat berupa pengguna, pemilik, pengembang, pemrogram, dan orang-orang yang terlibat dalam rekayasa perangkat lunak tersebut.

4. Dokumentasi merupakan bagian penting karena masing-masing tahapan dalam model biasanya menghasilkan sejumlah tulisan, diagram, gambar, atau bentuk-bentuk lain yang harus didokumentasi dan merupakan bagian tak terpisahkan dari perangkat lunak yang dihasilkan.
5. Keluaran dari proses pengembangan perangkat lunak harus bernilai ekonomis. Efek dari penggunaan perangkat lunak yang telah dikembangkan haruslah memberi nilai tambah bagi organisasi.

Menurut Kendall (2007), *Systems Development Life Cycle* (SDLC) atau siklus hidup pengembangan sistem adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik. Siklus pengembangan sistem dibagi atas tujuh tahap, antara lain :

1. Mengidentifikasi masalah, peluang dan tujuan

Dalam tahap ini penganalisis menentukan dengan tepat masalah-masalah dalam bisnis mereka, mengukur peluang guna mencapai sisi kompetitif atau menyusun standar-standar industri, dan tujuan-tujuan yang harus dicapai.

2. Menentukan syarat-syarat informasi

Dalam tahap ini, penganalisis berusaha untuk memahami informasi apa yang dibutuhkan pemakai agar bisa ditampilkan dalam pekerjaan mereka. Orang-orang yang terlibat adalah penganalisis dan pemakai, manajer operasi dan pegawai operasional. Penganalisis sistem perlu tahu detail-detail dan fungsi-fungsi sistem yang ada yaitu: siapa, apa, dimana, kapan, dan bagaimana dari bisnis yang sedang dipelajari.

3. Menganalisis kebutuhan sistem

Dalam tahap ini, penganalisis menganalisis keputusan terstruktur yang dibuat. Penganalisis juga menyiapkan suatu proposal sistem yang berisikan ringkasan apa saja yang ditemukan, analisis biaya keuntungan alternatif yang tersedia serta rekomendasi atas apa saja yang harus dilakukan.

4. Merancang sistem yang direkomendasikan

Dalam tahap ini, penganalisis merancang *data-entry* sedemikian rupa sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Penganalisis juga merancang *file-file* basis data yang menyimpan data yang diperlukan oleh pembuat keputusan dan penganalisis bekerja sama dengan pemakai untuk merancang *output*. Terakhir penganalisis juga merancang prosedur-prosedur *back-up* dan kontrol untuk melindungi sistem dan data serta membuat paket-paket spesifikasi program bagi pemrogram.

5. Mengembangkan dan mendokumentasikan perangkat lunak

Dalam tahap ini, penganalisis bekerja sama dengan pemrogram mengembangkan suatu perangkat lunak awal yang diperlukan. Penganalisis juga bekerja sama dengan pemakai untuk mengembangkan dokumentasi perangkat lunak yang efektif, mencakup melakukan prosedur secara manual, bantuan *online* dan *website*.

6. Menguji dan mempertahankan sistem

Dalam tahap ini, sistem yang telah dibuat harus dilakukan pengujian terlebih dahulu. Sebagian pengujian dilakukan oleh pemrogram sendiri dan lainnya dilakukan oleh penganalisis sistem.

7. Mengimplementasikan dan mengevaluasi sistem

Tahap ini merupakan tahap terakhir yang melibatkan pelatihan bagi pemakai untuk pengendalian sistem. Pelatihan dilakukan oleh vendor, namun kesalahan pelatihan merupakan tanggung jawab penganalisis sistem. Proses ini mencakup pengubahan *file-file* dari format lama ke format baru atau membangun suatu basis data, menginstall peralatan, dan membawa sistem baru untuk diproduksi.

2.5.3 Tahapan Rekayasa Perangkat Lunak

Meskipun dalam pendekatan berbeda-beda, namun model-model pendekatan memiliki kesamaan, yaitu menggunakan pola tahapan *analysis – design – coding (construction) – testing – maintenance*.

1. Analisis Sistem adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi komponen-komponennya dengan tujuan mempelajari seberapa bagus komponen-komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka.
2. Model Proses adalah model yang menunjukkan aliran data yang masuk dan keluar pada suatu proses. Biasanya model ini digambarkan dalam *Data Flow Diagram / DFD*.
3. Desain Perangkat Lunak adalah tugas, tahapan atau aktivitas yang difokuskan pada spesifikasi detil dari solusi berbasis komputer (Whitten et al, 2004).
4. Konstruksi adalah tahapan menerjemahkan hasil desain logis dan fisik ke dalam kode-kode program komputer.
5. Pengujian sistem melibatkan semua kelompok pengguna yang telah direncanakan pada tahap sebelumnya. Pengujian tingkat penerimaan

terhadap perangkat lunak akan berakhir ketika dirasa semua kelompok pengguna menyatakan bisa menerima perangkat lunak tersebut berdasarkan kriteria-kriteria yang telah ditetapkan.

Perawatan dan Konfigurasi ketika sebuah perangkat lunak telah dianggap layak untuk dijalankan, maka tahapan baru menjadi muncul yaitu perawatan perangkat lunak.

2.6 Analisa dan Perancangan Sistem

Menurut Kristanto (2003:5), analisa sistem adalah seseorang yang mempunyai kemampuan untuk menganalisa sebuah sistem yang meliputi mempelajari masalah yang timbul dan menentukan kebutuhan pemakai sistem. Untuk mencapai tujuan dari suatu sistem yang dibuat, dibutuhkan tiga perangkat atau alat yang dapat meningkatkan kinerja dari sebuah sistem sehingga tujuan dari sistem tersebut dapat dicapai. Tiga perangkat tersebut meliputi: perangkat keras, perangkat lunak, dan perangkat manusia. Perangkat keras dapat berupa komputer, sedangkan perangkat lunak adalah program. Sedangkan perangkat manusia dapat berupa manajer, analisis sistem, programmer, dan sebagainya. Ketiga unsur tersebut bersama-sama membangun sistem yang efisien untuk mengatasi masalah yang dihadapi pemakai sistem.

Menurut Jogiyanto (2001:129) “Analisis Sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi, dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya”. Berdasarkan penjelasan di atas, analisa sistem adalah sebuah tahap

yang paling penting dalam suatu pemrograman karena tahap ini akan mengevaluasi permasalahan yang ada dan kendala-kendala yang dihadapi. Tahap analisis sistem dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem atau perancangan sistem.

Menurut Kendall (2003:7), analisa sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi.

Analisa dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

2.7 Konsep Basis Data

2.7.1 Database

Menurut Yuswanto (2005:2), *database* merupakan sekumpulan data yang berisi informasi yang saling berhubungan. Pengertian ini sangat berbeda antara *database Relasional* dan *NonRelasional*. Pada *database NonRelasional*, sebuah *database* hanya merupakan sebuah *file*.

Menurut Marlinda (2004:1), *database* adalah suatu susunan atau kumpulan data operasional lengkap dari suatu organisasi atau perusahaan yang

diorganisir atau dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu *redundansi* dan *inkonsistensi* data, kesulitan pengaksesan data, isolasi data untuk standarisasi, *multiple user* (banyak pemakai), masalah keamanan (*security*), masalah integrasi (kesatuan), dan masalah data *independence* (kebebasan data).

2.7.2 Sistem Basis Data

Menurut Marlinda (2004:1), sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara dan operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi optimal yang diperlukan pemakai untuk proses mengambil keputusan. Pada sebuah sistem basis data terdapat komponen-komponen utama yaitu Perangkat Keras (*Hardware*), Sistem Operasi (*Operating System*), Basis Data (*Database*), Sistem (Aplikasi atau Perangkat Lunak) Pengelola Basis Data (DBMS), Pemakai (*User*), dan Aplikasi (Perangkat Lunak) lain (bersifat opsional).

Keuntungan sistem basis data adalah:

1. Mengurangi kerangkapan data, yaitu data yang sama disimpan dalam berkas data yang berbeda-beda sehingga *update* dilakukan berulang-ulang.
2. Mencegah ketidakkonsistenan.
3. Keamanan data dapat terjaga, yaitu data dapat dilindungi dari pemakai yang tidak berwenang.

4. Integritas dapat dipertahankan.
5. Data dapat dipergunakan bersama-sama.
6. Menyediakan *recovery*.
7. Memudahkan penerapan standarisasi.
8. Data bersifat mandiri (*data independence*).
9. Keterpaduan data terjaga, memelihara keterpaduan data berarti data harus akurat. Hal ini sangat erat hubungannya dengan pengontrolan kerangkapan data dan pemeliharaan keselarasan data.

Kerugian sistem basis data adalah:

1. Diperlukan tempat penyimpanan yang besar.
2. Diperlukan tenaga yang terampil dalam mengolah data.
3. Perangkat lunaknya mahal.

Kerusakan sistem basis data dapat mempengaruhi departemen yang terkait.

2.8 Testing dan Implementasi Sistem

Menurut Standar ANSI/IEEE 1059, *testing* adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects/error/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Menurut Romeo (2003:3), *testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk:

1. Verifikasi. Apakah telah berlaku sebagaimana yang ditetapkan (menurut spesifikasi)
2. Mendeteksi kesalahan.

3. Validasi. Apakah spesifikasi yang ditetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya?

Menurut Romeo, *Test Case* merupakan tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya (2003:33). Metode *testing* yang digunakan adalah *Black Box Testing*.

Menurut Romeo (2003:52), metode uji coba *black box* memfokuskan pada keperluan fungsional dari *software*. Karena itu uji coba *black box* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang melatih seluruh syarat-syarat fungsional suatu program. Uji coba *black box* bukan merupakan alternatif dari uji coba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*.

Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database external*.
4. Kesalahan *performa*.
5. Kesalahan inisialisasi dan terminasi.

Tidak seperti metode *white box* yang dilaksanakan diawal proses, uji coba *black box* diaplikasikan di beberapa tahapan berikutnya. Karena uji coba *black box* dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Uji coba didesain untuk dapat menjawab pertanyaan berikut: (Romeo,2003:52)

1. Bagaimana validitas fungsionalnya diuji ?
2. Jenis input seperti apa yang menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan-batasan kelas data diisolasi ?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem ?
6. Apa akibat yang timbul dari kombinasi spesifik data pada operasi sistem ?

Dengan mengaplikasikan uji coba *black box*, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. Kasus uji yang berkurang, jika jumlahnya lebih dari satu, maka jumlah dari uji kasus tambahan harus didesain untuk mencapai uji coba yang cukup beralasan
2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, daripada kesalahan yang terhubung hanya dengan suatu uji coba yang spesifik.

Black box testing menurut Romeo (2003:62), dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*. Dengan adanya *black box testing*, perencana *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.

Kategori *error* yang diketahui melalui *black box testing* adalah:

1. Fungsi yang hilang atau tak benar.
2. *Error* dari antar-muka.
3. *Error* dari struktur data atau akses *eksternal database*.
4. *Error* dari kinerja atau tingkah laku.
5. *Error* dari inisialisasi dan terminasi.

2.9 Perancangan Antar Muka

Pada tahap ini, dilakukan perancangan antar muka masukan atau keluaran yang akan digunakan. Hal ini dilakukan dengan menggunakan perangkat lunak *Microsoft Visio 2010*.

a. Interaksi Manusia dan Komputer

Menurut Rizky (2006:4), Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-faktor utama dalam lingkungan interaksinya. Deskripsi lain dari IMK adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif. Dikatakan juga bahwa sebuah desain antar muka yang ideal adalah yang mampu memberikan kepuasan terhadap manusia sebagai pengguna dengan faktor kapabilitas serta keterbatasan yang terdapat dalam sistem.

Pada implementasinya, IMK dipengaruhi berbagai macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan, antar muka, kendala, dan produktifitas.