

BAB III

LANDASAN TEORI

3.1 Sistem

Sistem memiliki beberapa definisi yang berbeda-beda menurut pendapat beberapa ahli, definisi sistem adalah sebagai berikut:

1. Menurut Jogiyanto (1999:1), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Menurut Soeherman dan Pinontoan (2008:3), sistem dapat diartikan sebagai serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu.

3.2 Web Server

Abdul Kadir (2009) mengatakan bahwa *web server* adalah suatu *software* yang bertindak melayani pengaksesan aplikasi *web*. Contoh *software* yang berfungsi sebagai *web server* antara lain Apache, IIS, dan Xitami

Web browser seperti *explorer* atau navigator berkomunikasi melalui jaringan (termasuk jaringan internet) dengan *web server*, menggunakan HTTP browser akan mengirimkan *request* kepada *server* untuk meminta dokumen tertentu atau layanan yang disediakan oleh *server*. *Server* memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protokol HTTP.

3.3 Informasi

Informasi memiliki beberapa definisi yang berbeda-beda menurut pendapat beberapa ahli, definisi informasi adalah sebagai berikut:

1. Menurut McLeod dan Schell (2007:12), informasi adalah data yang telah diolah sehingga lebih bermakna.
2. Menurut Soeherman dan Pinontoan (2008:4), informasi merupakan hasil pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan keputusan.
3. Menurut Jogiyanto (1999:8), informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat yang tertentu. Didalam dunia bisnis, kejadian-kejadian nyata yang sering terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi.

3.4 Sistem Informasi

Sistem informasi mempunyai peranan yang penting dalam membantu menyediakan informasi untuk berbagai tingkatan manajemen. Sistem yang dibutuhkan adalah sistem yang memperlancar proses kegiatan yang sedang berjalan. Menurut Jogiyanto (1999:11), sistem informasi adalah suatu sistem didalam 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.

3.5 Sistem Informasi Penggajian

Definisi sistem informasi penggajian menurut Krismiaji (2005:25), sistem informasi penggajian adalah serangkaian aktivitas bisnis dan kegiatan pengolahan data yang terkait yang berhubungan dengan pengelolaan karyawan perusahaan secara efektif yang terdiri dari *input*, proses dan menghasilkan *output* berupa informasi tentang gaji yang bermanfaat bagi perusahaan. Proses penggajian di dalam PT. Mekar Asri Sidorukun dilakukan dengan menghitung jumlah presensi yang dilakukan oleh karyawan dan dikalikan dengan tunjangan. Ketika jumlah presensi yang dilakukan oleh karyawan kurang dari setengah dari jadwal kerja, maka gaji pokok yang di dapat akan di bagi dengan jumlah jadwal kerjanya, lalu akan dikalikan dengan jumlah karyawan melakukan presensi, dan akan dikurangi dengan data hutang dan potongan absensi yang di kalikan dengan jumlah absensi yang dilakukan karyawan lalu akan ditemukan total gaji.

3.6 Website

Menurut Hidayat (2010:2), *website* adalah suatu situs atau dapat diartikan sebagai kumpulan-kumpulan halaman yang digunakan untuk menampilkan informasi, gambar diam, atau gambar gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis atau dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* dengan halaman *web* lainnya disebut *Hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *Hypertext*.

3.7 *Hypertext Preprocessor (PHP)*

Menurut Menurut Anhar (2010:3), PHP adalah singkatan dari *Hypertext Preprocessor* yang merupakan bahasa pemrograman *web server-side* dan bersifat *open source*. PHP merupakan *scriPT.*, yang terintegrasi dengan HTML dan berada pada *server*. PHP digunakan untuk membuat halaman *website* lebih dinamis. Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan *scriPT.* PHP.

Syntax PHP mirip dengan bahasa C, Perl, Pascal dan Basic. PHP dapat dikembangkan sebagai *web* spesifik yang menyediakan fungsi-fungsi khusus yang membuat pengembangan suatu *web* dapat dilakukan dengan mudah. PHP juga menyediakan koneksi *database*, protokol, dan modul fungsi lainnya.

3.8 *Database Management System (DBMS)*

3.8.1 *Pengertian DBMS*

Menurut Connolly dan Begg (2010:66), *Database Management System* adalah sebuah sistem *software* yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke database.

3.8.2 *Keuntungan DBMS*

Keuntungan DBMS yang diuraikan Connolly dan Begg (2010:77), adalah sebagai berikut:

1. Mengontrol Pengulangan Data.

Pendekatan basis data melakukan percobaan untuk menghilangkan data yang redundansi dengan mengintegrasikan file-file sehingga data-data yang sama tidak disimpan di dalam basis data.

2. Data yang Konsisten.

Dengan menghilangkan atau mengontrol data yang redundansi, sudah mengurangi resiko data yang tidak konsisten. Apabila data yang disimpan hanya sekali dalam basis data, maka data yang diperbarui juga hanya sekali dan nilainya dapat digunakan oleh semua pemakai.

3. Tambahan Informasi.

Informasi yang lebih dari jumlah data yang sama operasional data memungkinkan organisasi untuk mendapatkan tambahan informasi dari data yang sama setelah data tersebut diintegrasikan.

4. Membagi Data.

Basis data tidak lagi hanya dimiliki oleh departemen atau bagian tertentu saja, tetapi basis data menjadi milik organisasi dan bisa diberikan kepada semua pemakai.

5. Meningkatkan Kesatuan Data.

Kesatuan data menunjukkan bahwa data yang disimpan adalah valid dan konsisten. Kesatuan mengizinkan DBA untuk menetapkan batasan kesatuan itu sendiri, dan DBMS menjalankan semua data tersebut.

6. Meningkatkan Keamanan.

Keamanan basis data melindungi basis data dari orang-orang yang tidak mempunyai hak akses terhadap basis data tersebut

7. Meningkatkan Standart.

Integrasi mengijinkan DBA untuk mendefinisikan dan DBMS untuk menjalankan kebutuhan-kebutuhan yang standart. Kebutuhan yang standart ini terdiri dari sebuah standar format data dari departemen, organisasi nasional, maupun internasional seperti fasilitas perubahan data, penamaan, dan perubahan prosedur.

8. Perbandingan Skala Ekonomi.

Penghematan biaya bisa dilakukan dengan menggabungkan semua data operasional suatu organisasi ke dalam satu basis data, dan membuat aplikasi yang bekerja dengan satu sumber data saja

9. Menyeimbangi Kebutuhan Masalah.

Database Administrator (DBA) akan membuat keputusan tentang rancangan dan penggunaan basis data secara operasional yang menyediakan kebutuhan yang terbaik untuk seluruh organisasi.

10. Meningkatkan Pengaksesan Data.

Banyak DBMS yang menyediakan bahasa *query* atau laporan penulis yang memungkinkan pemakai untuk bertanya tentang pertanyaan-pertanyaan dan memperoleh informasi yang dibutuhkan tanpa melibatkan *programmer* untuk mengambil informasi tersebut dalam basis data.

11. Meningkatkan Produktifitas.

DBMS dapat menyederhanakan pengembangan dari suatu aplikasi basis data sehingga dapat meningkatkan produktivitas *programmer* dan mengurangi waktu pengembangan.

12. Meningkatkan Pemeliharaan Melalui Data Independen.

DBMS memisahkan deskripsi basis data program aplikasi, sehingga program aplikasi tidak dapat mengubah basis data.

13. Meningkatkan Konkurensi.

Seringkali lebih dari satu pemakai sering mengakses *file* yang sama, dengan adanya DBMS kecepatan konkurensi basis data akan diatur

14. Meningkatkan layanan *backup* dan *recovery*.

Jika terjadi kesalahan, *backup* akan di *restore* dan pekerjaan setelah *backup* akan hilang. DBMS yang modern menyediakan fasilitas untuk meminimalis proses-proses yang hilang yang disebabkan karena kegagalan dalam memproses data.

3.8.3 Kekurangan DBMS

Kekurangan dari DBMS menurut Connolly dan Begg (2010:80), adalah sebagai berikut:

1. Kompleks.

DBMS merupakan bagian dari perangkat lunak yang sangat kompleks. Kesalahan terhadap pengertian sistem akan mengakibatkan rancangan keputusan yang buruk pada suatu organisasi sehingga perancang basis data dan pengembang basis data, *Database Administrator* (DA), serta *end-user* perlu mengerti tentang keuntungan fungsional DBMS terlebih dahulu

2. Ukuran.

Kompleksitas dan banyaknya kegunaan dari DBMS menjadikannya perangkat lunak yang sangat besar, sehingga memerlukan tempat penyimpanan data yang besar dan membutuhkan memori yang cukup agar berjalan secara efisien.

3. Biaya.

Biaya yang dikeluarkan untuk DBMS sangat bervariasi, tergantung dari lingkungan dan kegunaan yang disediakan oleh DBMS tersebut.

4. Tambahan Biaya Untuk Perangkat Keras.

Kebutuhan tempat penyimpanan data untuk DBMS dan basis data mungkin mengharuskan pembelian tempat penyimpanan data tambahan.

5. Biaya Konversi.

Dalam suatu situasi tertentu, biaya untuk DBMS dan perangkat keras tambahan dapat menjadi tidak penting disbanding dengan biaya konversi dari aplikasi yang sudah ada agar dapat berjalan di DBMS dan perangkat keras yang baru.

6. Peforma.

Bagaimanapun, DBMS dibuat untuk menjadi lebih umum, agar dapat menangani banyak aplikasi daripada satu saja. Sehingga menghasilkan aplikasi yang tidak berjalan secepat yang diharapkan

7. Tingkat Kegagalan yang Lebih Besar.

Pemutusan dari sumber daya meningkatkan kerentanan sistem yang dikarenakan semua pemakai dan aplikasi bergantung pada ketersediaan dari DBMS, kegagalan dari salah satu komponen dapat membuat operasi berhenti.

3.9 SQL (*Structured Query Language*)

Menurut Connolly dan Begg (2002:111), SQL (*Structured Query Language*) adalah sebuah bahasa yang dirancang untuk menggunakan relasi untuk mengubah masukan (*Input*) menjadi keluaran (*Output*) yang dibutuhkan.

Komponen SQL

Menurut Connolly dan Begg (2002:111), SQL terdiri dari 2 komponen utama yaitu:

1. DDL (*Data Definition Language*)

Untuk mendefinisikan struktur database dan mengontrol akses ke data.

2. DML (*Data Manipulation Language*)

Untuk mendapatkan dan memperbarui data.

3.10 Alat Pengembangan Sistem

3.10.1 Bagian Alir Dokumen (*Document Flowchart*)

Menurut Krismiaji (2005:75) Bagian alir dokumen menggambarkan aliran dokumen dan informasi antar area pertanggung jawaban di dalam sebuah organisasi. Bagan alir ini menelusuri sebuah dokumen dari asalnya sampai dengan tujuannya. Tujuan digunakan dokumen tersebut, kapan tidak dipakai lagi dan hal-hal lain yang terjadi ketika dokumen tersebut mengalir melalui sebuah sistem.

Menurut James A Hall (2007:75) Bagian alir dokumen digunakan untuk menggambarkan elemen-elemen dari sebuah sistem manual, termasuk record-record akuntansi (dokumen, jurnal, buku besar, dan file), departemen organisasional yang terlibat dalam proses, dan kegiatan-kegiatan yang dilakukan dalam departemen tersebut. Berdasarkan dua definisi tersebut penulis dapat menarik simpulan bahwa bagan alir dokumen adalah suatu bagan yang menunjukkan aliran dokumen dari asalnya sampai dengan tujuannya.

3.10.2 Bagian Alir Sistem (*System Flowchart*)

Menurut Krismiaji (2005:75) Bagian alir sistem menggambarkan hubungan antara *input*, pemrosesan dan *output* sebuah sistem informasi akuntansi. Bagan alir sistem ini dimulai dengan identifikasi input yang masuk kedalam sistem sumbernya. Bagian alir sistem merupakan salah satu alat penting untuk menganalisa, mendesain dan mengevaluasi sistem.

Menurut James A Hall (2009:83) Bagan alir sistem merupakan pemotretan aspek-aspek komputer dalam sebuah sistem. Berdasarkan dua definisi tersebut penulis dapat menarik simpulan bahwa alir sistem adalah suatu bagan yang menjelaskan urutan dari prosedur dalam sebuah sistem manual dan bagian alir sistem ini dimulai dengan *input* yang masuk kedalam sistem dan sumbernya.

3.10.3 Diagram Konteks

Menurut Al-Bahra (2005:64) Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Dari definisi tersebut dapat disimpulkan bahwa diagram konteks adalah diagram yang menggambarkan alur ruang lingkup dari suatu sistem dan terdiri dari dokumen-dokumen serta fungsi-fungsi terkait.

3.10.4 Diagram HIPO (*Hierarchy plus Input-Process-Output*)

Menurut Jogiyanto (2005:787) HIPO (*Hierarchy plus Input-Process-Output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM, tetapi saat ini HIPO juga banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO digunakan sebagai alat pengembangan

sistem dan teknik dokumentasi program. Penggunaan HIPO ini mempunyai sarana utama sebagai berikut:

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari program.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan statemen-statemen program yang digunakan.
3. Untuk menyediakan penjelasan yang jelas dari input yang harus digunakan dan output yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO
4. Untuk menyediakan output yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai.

3.10.5 Data Flow Diagram (DFD)

Menurut Jogyanto (2004:700) *data flow diagram* digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan.

Menurut Al-Bahra-(2005:64) *data flow diagram* merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Dari dua definisi tersebut dapat disimpulkan bahwa diagram arus data adalah model sistem yang digunakan untuk menjelaskan alur sistem namun lebih terperinci.

3.10.6 CDM (Conceptual Data Model)

CDM dipakai untuk menggambarkan secara detail struktur basis data dalam bentuk *logic*. Struktur ini independen terhadap semua *software* maupun struktur data

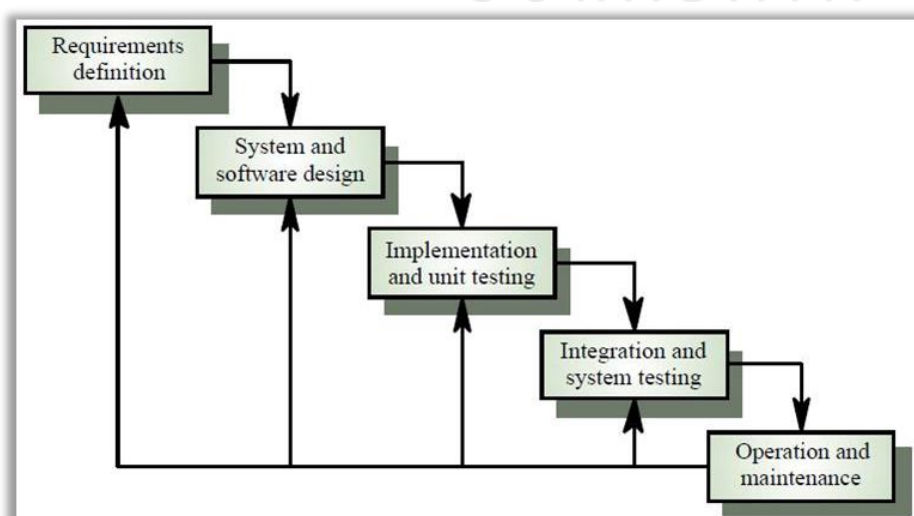
storage tertentu yang digunakan dalam aplikasi ini. CDM terdiri dari objek yang tidak diimplementasikan secara langsung kedalam basis data yang sesungguhnya.

3.10.7 PDM (*Physical Data Model*)

PDM merupakan gambaran secara detail basis data dalam bentuk fisik. Penggambaran rancangan PDM memperlihatkan struktur penyimpanan data yang benar pada basis data yang digunakan sesungguhnya.

3.11 *System Development Life Cycle (SDLC)*

Menurut Pressman (2001), model *System Development Life Cycle (SDLC)* ini biasa disebut juga dengan model *waterfall* atau disebut juga *classic life cycle*. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian), dan *Maintenance* (perawatan). Dan adapun tahapan SDLC sebagai berikut :



Gambar 3.1 Sistem *Development Life Cycle (SDLC)* Pressman (2001)

1. *Requirements* (Analisis Kebutuhan Sistem)

Pada tahap awal ini dilakukan analisa guna menggali kebutuhan yang dibutuhkan, yaitu kebutuhan data dan kebutuhan *user*. Kebutuhan sebenarnya dibedakan menjadi tiga jenis kebutuhan. Pertama tentang kebutuhan teknologi. Dari hal ini dilakukan analisa mengenai kebutuhan teknologi yang diperlukan dalam pengembangan suatu sistem seperti data penyimpanan informasi atau *database*. Kedua kebutuhan informasi, contohnya seperti informasi mengenai visi dan misi perusahaan, sejarah perusahaan, latar belakang perusahaan, dan yang ketiga yaitu kebutuhan *user*. Dalam hal ini dilakukan analisa terkait kebutuhan *user* dan kategori *user*. Dari analisa yang telah disebutkan diatas, terdapat satu hal yang tidak kalah penting dalam tahap analisa di metode SDLC, yaitu analisa biaya dan resiko. Dalam tahap ini diperhitungkan biaya yang akan dikeluarkan seperti biaya implementasi, *testing*, dan *maintenance*.

2. *Design* (Perancangan)

Selanjutnya, hasil analisa kebutuhan sistem tersebut akan dibuat sebuah *design database*, *DFD*, *ERD*, antarmuka pengguna atau *Graphical User Interface* dan jaringan yang dibutuhkan untuk sistem. Selain itu juga perlu merancang struktur data, arsitektur perangkat lunak, detil prosedur dan karakteristik tampilan yang akan disajikan. Proses ini menterjemahkan kebutuhan sistem ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum memulai tahap implementasi.

3. *Implementation* (Coding)

Rancangan yang telah dibuat dalam tahap sebelumnya akan diterjemahkan dalam bentuk bahasa yang dapat dibaca dan diterjemahkan oleh komputer untuk

diolah. Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi hasil perancangan sebelumnya ke dalam sebuah bahasa pemrograman yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunaanya.

4. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui kesesuaian sistem berjalan sesuai prosedur atau tidak dan memastikan sistem terhindar dari *error* yang terjadi. *Testing* juga dapat digunakan untuk memastikan ke-*valid*-an dalam proses *input*, sehingga dapat menghasilkan *output* yang sesuai. Pada tahap ini terdapat dua metode pengujian, yaitu metode *black-box* dan *white-box*. Pengujian dengan metode *black-box* menekankan pada fungsionalitas dari sebuah perangkat lunak tanpa harus mengetahui bagaimana struktur di dalam perangkat lunak tersebut. Perangkat lunak yang diuji menggunakan metode *black-box* dikatakan berhasil jika fungsi-fungsi yang ada telah memenuhi spesifikasi kebutuhan yang dibuat sebelumnya. Pengujian dengan metode *white-box* yaitu menguji struktur internal perangkat lunak dengan melakukan pengujian pada algoritma yang digunakan oleh perangkat lunak.

5. *Maintenance* (Perawatan)

Tahap terakhir dari metode SDLC ini adalah *Maintenance*. Pada tahap ini jika sistem sesuai dengan tujuan yang ditentukan dan dapat menyelesaikan masalah pada perusahaan, maka akan diberikan kepada pengguna. Setelah digunakan dalam periode tertentu pasti terdapat penyesuaian atau perubahan sesuai dengan keadaan yang diinginkan, sehingga membutuhkan perubahan terhadap sistem tersebut. Tahap ini dapat pula diartikan sebagai tahap penggunaan perangkat lunak yang disertai

dengan perawatan dan perbaikan. Perawatan dan perbaikan suatu perangkat lunak diperlukan, termasuk didalamnya adalah pengembangan, karena dalam praktiknya ketika perangkat lunak digunakan terkadang masih terdapat kekurangan atau penambahan fitur-fitur baru yang dirasa perlu.

3.11.1 Kebutuhan Perangkat Lunak

Dalam IEEE *Komputer Society* (2004), kebutuhan perangkat lunak merupakan piranti yang harus dipamerkan untuk memecahkan beberapa masalah yang ada. Kebutuhan perangkat lunak bertujuan untuk menjalankan secara otomatis bagian dari tugas seseorang untuk mendukung proses bisnis dari sebuah organisasi, memperbaiki kekurangan dari perangkat lunak yang ada, mengontrol perangkat dari beberapa masalah yang dapat dijadikan solusi perangkat lunak. Terdapat empat tahapan dalam kebutuhan perangkat lunak yaitu sebagai berikut:

1. Elisitasi

Tahapan ini merupakan tahapan awal dalam membangun pemahaman tentang perangkat lunak yang diperlukan untuk memecahkan masalah. Tahapan kegiatan elisitasi digunakan untuk mengidentifikasi darimana asal kebutuhan perangkat lunak dan bagaimana cara mendapatkannya.

2. Analisis

Tahap ini menjelaskan tentang konflik antar kebutuhan, menemukan batas-batas perangkat lunak dan bagaimana interaksi dengan lingkungan sekitar, serta menguraikan kebutuhan sistem untuk mendapatkan kebutuhan perangkat lunak.

3. Spesifikasi

Pada tahap ini dilakukan pembuatan dokumen perencanaan perangkat lunak (solusi aplikasi), dan mengajukan solusi aplikasi kepada pihak *Stake Holder* yang terkait.

4. Validasi

Tahapan ini digunakan untuk memastikan bahwa kebutuhan perangkat lunak telah sesuai dengan kebutuhan pengguna.

3.11.2 Konstruksi Perangkat Lunak

Dalam IEEE *Komputer Society* (2004), tahapan konstruksi perangkat lunak digunakan untuk melakukan konversi hasil desain ke sistem informasi yang lengkap melalui tahapan pengkodean termasuk bagaimana membuat basis data dan menyiapkan prosedur pengujian, mempersiapkan *file* pengujian, kompilasi pengkodean, memperbaiki dan membersihkan program serta melakukan peninjauan pengujian.

3.11.3 Uji Coba Perangkat Lunak

Menurut IEEE *Komputer Society* (2004), uji coba perangkat lunak terdiri dari verifikasi dinamis yang menyediakan perilaku sebuah perangkat lunak yang diwakili oleh beberapa contoh kasus uji coba. Kasus uji coba tersebut dilakukan dengan memberikan masukan kepada perangkat lunak agar muncul reaksi sesuai yang diharapkan, maupun sebaliknya.

Dalam melakukan uji coba perangkat lunak, ada lima hal yang harus diperhatikan. Pertama adalah fundamental dari uji coba perangkat lunak yang

menjelaskan tentang *terminology* dari uji coba terkait, kunci masalah dari uji coba, dan hubungan uji coba tersebut dengan aktifitas lainnya di dalam perangkat lunak. Kedua adalah tingkatan dari uji coba yang di dalamnya menjelaskan tentang target dari uji coba dan tujuan dari uji coba tersebut. Hal ketiga yang harus diperhatikan dalam teknik dari uji coba meliputi uji coba berdasarkan intuisi dan pengalaman *tester*, diikuti oleh teknik berdasarkan spesifikasi, teknik berdasarkan kode, teknik berdasarkan kesalahan, teknik berdasarkan penggunaan, dan teknik berdasarkan *relative* ketergantungan dari aplikasi tersebut. Keempat bahwa pengukuran dikelompokkan menjadi dua yang berhubungan dengan evaluasi ketika uji coba dilakukan dan ketika uji coba telah selesai dilakukan. Kelima adalah proses uji coba itu sendiri yang berisi tentang pertimbangan praktis dan aktivitas uji coba.

