

BAB II

LANDASAN TEORI

2.1 Pembelian

Pembelian adalah usaha pengadaan barang-barang untuk perusahaan. Dalam perusahaan dagang pembelian dilakukan dengan dijual kembali tanpa mengadakan perubahan bentuk barang, sedangkan pada perusahaan manufaktur pembelian dilakukan dengan mengubah bentuk barang.

Pembelian (purchases) adalah harga pembelian (harga pokok) barang dagang yang diperoleh perusahaan selama periode tertentu (Aliminsyah dan Padji, 2003). Menurut Soemarso S.R (1994) kegiatan pembelian dalam perusahaan dagang adalah:

- a. Membeli barang secara tunai atau kredit.
- b. Membeli aktiva produksi untuk digunakan dalam kegiatan perusahaan.
- c. Membeli barang dan jasa lain sehubungan dengan kegiatan perusahaan.

Sehingga dapat diperoleh kesimpulan bahwa pembelian merupakan perkiraan yang digunakan untuk mencatat semua pembelian barang dagang dalam suatu periode tertentu.

2.2 Pemakaian

Menurut Eeng Ahman (2003) pemakaian atau konsumsi merupakan suatu kegiatan mengurangi atau menghabiskan nilai guna ekonomi suatu barang atau jasa baik secara sekaligus maupun secara berangsur-angsur.

Kegiatan konsumsi dilakukan dengan tujuan untuk:

- a. Mengurangi nilai guna suatu barang dan jasa secara bertahap.
- b. Menghabiskan atau mengurangi nilai guna suatu barang sekaligus.
- c. Memuaskan kebutuhan jasmani dan rohani.

2.3 Administrasi

Administrasi berdasarkan etimologis (asal kata) bersumber dari bahasa latin, yang terdiri *ad + ministrate*, yang secara operasional berarti melayani, membantu dan memenuhi. Dalam bahasa asalnya dari perkataan itu dapat terbentuk kata benda *administratio* dan kata sifat *administrativus*. Perkataan itu masuk ke dalam bahasa Inggris menjadi *administration* yang lebih banyak dikenal oleh para ilmuwan dan praktisi sekarang ini. Di bawah ini terdapat beberapa pendapat, arti atau definisi dari administrasi, yaitu:

A. Menurut Hendi Haryadi dalam bukunya *Administrasi Perkantoran untuk Manajer & Staf* (2009:1) mengatakan bahwa ada dua pengertian administrasi, yaitu dalam arti sempit dan administrasi dalam arti luas.

- a. Administrasi dalam arti sempit adalah kegiatan penyusunan dan pencatatan data dan informasi secara sistematis dengan tujuan untuk menyediakan keterangan serta memudahkan memperolehnya kembali secara keseluruhan dan dalam satu hubungan satu sama lain.
- b. Administrasi dalam arti luas adalah kegiatan kerja sama yang dilakukan sekelompok orang berdasarkan pembagian kerja sebagaimana ditentukan dalam struktur dengan mendayagunakan sumber daya untuk mencapai tujuan secara efektif dan efisien.

- B. Menurut Sondang P. Siagian dalam bukunya *Administrasi Pembangunan* (1947:2) mengatakan bahwa Administrasi adalah "Keseluruhan proses pelaksanaan daripada keputusan yang telah diambil dan pelaksanaan itu pada umumnya dilakukan oleh dua orang manusia atau lebih untuk mencapai tujuan yang telah ditentukan sebelumnya".
- C. Menurut J. Wayong dalam bukunya *Fungsi Administrasi Negara* terbitan tahun 1961, mengemukakan bahwa Administrasi adalah "Kegiatan yang dilakukan untuk mengendalikan suatu usaha (pemerintah) agar tujuan tercapai".

Administrasi seperti telah banyak disebutkan dalam uraian terdahulu adalah rangkaian kegiatan atau proses pengendalian suatu organisasi agar secara keseluruhan selalu terarah pada pencapaian tujuannya. Dengan demikian administrasi berarti rangkaian kegiatan atau proses pengendalian cara atau sistem kerja sama sejumlah orang agar berlangsung efektif dan efisien dalam mewujudkan tujuan bersama.

2.4 Desain Sistem

Setelah tahap analisis dan perancangan sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Lalu tahap selanjutnya adalah desain sistem.

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.

Menurut Jogiyanto (2005) 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*, *output*, *database*, teknologi, dan kontrol.

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.

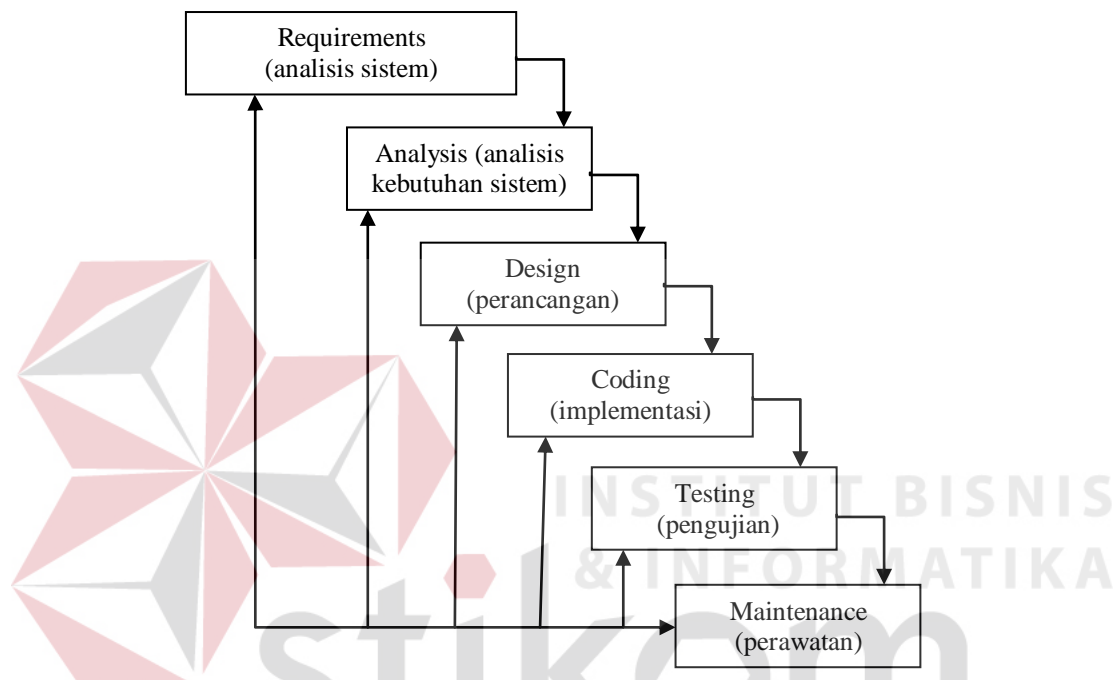
Logical model dari sistem informasi lebih menjelaskan kepada pemakai sistem bagaimana nantinya fungsi-fungsi pada sistem informasi secara logika akan bekerja. *Logical model* dapat digambarkan dengan diagram arus data (*data flow diagram*). Arus data pada *data flow diagram* dapat dijelaskan dengan kamus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

2.4.1 System Development Life Cycle (SDLC)

Pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Model yang digunakan biasa disebut dengan model *waterfall* atau disebut juga *classic life cycle*. Tahapan-tahapannya adalah *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian) dan *Maintenance* (perawatan) (Pressman, 1997).

Model eksplisit pertama dari proses pengembangan perangkat lunak, berasal dari proses-proses rekayasa yang lain. Model ini memungkinkan proses pengembangan lebih terlihat. Hal ini dikarenakan bentuknya yang bertingkat ke bawah dari satu fase ke fase lainnya, model ini dikenal dengan model *waterfall*, seperti terlihat pada Gambar 2.8.



. Gambar 2.1 *System Development Life Cycle (SDLC) Model Waterfall* (Pressman, 1997)

Penjelasan SDLC Model *Waterfall*, adalah sebagai berikut:

a. *Requirement (Analisis Sistem)*

Dalam merancang sebuah perangkat lunak, yang pertama harus dilakukan adalah membangun semua elemen sistem yang diperlukan. Sistem merupakan hal yang penting dalam membuat sebuah perangkat lunak, karena perangkat lunak harus berhubungan langsung dengan elemen lainnya seperti perangkat keras, basis data, dan manusia. Tahap ini didefinisikan sebagai sebuah tahap yang menghasilkan sebuah kondisi yang diperlukan oleh pengguna untuk

menyelesaikan permasalahan ataupun mencapai sebuah tujuan. Tahap ini bertujuan untuk mengumpulkan kebutuhan-kebutuhan pengguna dan kemudian mentransformasikan ke dalam sebuah deskripsi yang jelas dan lengkap.

b. *Analysis* (Analisis Kebutuhan Sistem)

Pada tahap ini dalam perancangan perangkat lunak, perlu mengetahui karakteristik dasar dari perangkat lunak yang akan dirancang, seperti fungsi, bentuk, dan tampilan dari perangkat lunak tersebut. Tahap analisis sistem ini bertujuan untuk menjabarkan segala sesuatu yang nantinya akan ditangani oleh perangkat lunak. Tahapan ini adalah tahapan pemodelan yang merupakan sebuah representasi *object* di dunia nyata.

c. *Design* (Perancangan)

Untuk membuat suatu perangkat lunak perlu dirancang struktur datanya, arsitektur perangkat lunak, detail prosedur dan karakteristik tampilan yang akan disajikan. Tahap perancangan perangkat lunak merupakan proses multi langkah dan berfokus pada beberapa atribut perangkat lunak yang berbeda, yaitu: struktur data, arsitektur perangkat lunak dan detail algoritma. Proses ini menterjemahkan kebutuhan ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum memulai tahap implementasi.

d. *Coding* (Implementasi)

Rancangan yang telah dibuat dalam tahap sebelumnya akan diterjemahkan ke dalam suatu bentuk atau bahasa yang dapat dibaca dan diterjemahkan oleh komputer untuk diolah. Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi apa yang telah dirancang

sebelumnya ke dalam sebuah bahasa yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

e. *Testing* (Pengujian)

Pengujian program dilakukan untuk mengetahui apabila terjadi kesalahan pada program yang telah dibuat. Dapat juga digunakan untuk memastikan apakah *input* proses dengan benar, sehingga dapat menghasilkan *output* yang sesuai.

Tahap ini terdapat 2 metode pengujian perangkat yang dapat digunakan, yaitu:

metode *black-box* dan *white-box*. Pengujian dengan metode *black-box* merupakan pengujian yang menekankan pada fungsionalitas dari sebuah perangkat lunak tanpa harus mengetahui bagaimana struktur di dalam perangkat lunak tersebut. Sebuah perangkat lunak yang diuji menggunakan metode *black-box* dikatakan berhasil jika fungsi-fungsi yang ada telah memenuhi spesifikasi kebutuhan yang telah dibuat sebelumnya. Pengujian dengan menggunakan metode *white-box* yaitu menguji struktur internal perangkat lunak dengan melakukan pengujian pada algoritma yang digunakan oleh perangkat lunak.

f. *Maintenance* (Perawatan)

Jika aplikasi tersebut telah sesuai, akan diberikan kepada pengguna dan terdapat penyesuaian atau perubahan sesuai dengan keadaan yang diinginkan, sehingga membutuhkan perubahan terhadap aplikasi 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

prakteknya ketika perangkat lunak digunakan terkadang masih terdapat kekurangan ataupun penambahan fitur-fitur baru yang dirasa perlu.

2.5 Testing Software

Menurut Romeo (2003) testing software adalah proses mengoperasikan software dalam suatu kondisi yang di kendalikan, untuk verifikasi apakah telah berlaku sebagaimana telah ditetapkan (menurut spesifikasi), mendeteksi *error*, dan validasi apakah spesifikasi yang telah ditetapkan sudah memenuhi keinginan atau kebutuhan dari pengguna yang sebenarnya. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk software, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem, apakah proses yang telah dilakukan adalah apa yang sebenarnya diinginkan atau dibutuhkan oleh *user*. Jadi, dapat disimpulkan bahwa testing merupakan tiap-tiap aktifitas pengumpulan informasi yang dibutuhkan untuk melakukan evaluasi atau mengukur suatu atribut dari software.

Testing software dilakukan untuk mendapatkan informasi *reliable* terhadap software dengan cara termudah dan paling efektif, antara lain:

- a. Apakah software telah siap digunakan?
- b. Apa saja resikonya?
- c. Apa saja kemampuannya?
- d. Apa saja keterbatasannya?
- e. Apa saja masalahnya?
- f. Apakah telah berlaku seperti yang diharapkan?

2.5.1 Test Case

Test case merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya.

Adapun kegunaan dari *test case* ini, adalah sebagai berikut:

- a. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi (Black Box Testing).
- b. Untuk melakukan testing kesesuaian suatu komponen terhadap desain (White Box Testing).

Menurut Ganesan (2011) tujuan dasar dari penulisan *test case* adalah untuk melakukan validasi cakupan *testing* dari sebuah aplikasi. *Test case* yang ditulis dengan baik dapat membuat siklus *testing* menjadi lebih efisien. Sebuah *test case* yang baik dapat dengan mudah menentukan apakah suatu fitur dari aplikasi bekerja dengan benar.

2.5.2 Black Box Testing

Black box testing, dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang ditest, 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*. Kategori *error* yang akan diketahui melalui *black box testing* adalah sebagai berikut:

- a. Fungsi yang hilang atau tidak benar.
- b. *Error* dari antar muka.
- c. *Error* dari struktur data atau akses eksternal database.
- d. *Error* dari kinerja atau tingkah laku.

- e. *Error* dari inisialisasi dan terminasi.

Test di desain untuk menjawab pertanyaan sebagai berikut:

- a. Bagaimana validasi fungsi yang akan dites?
- b. Bagaimana tingkah laku kinerja dari sistem yang akan dites?
- c. Kategori masukan apa saja yang bagus digunakan untuk *test case*?
- d. Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu?
- e. Bagaimana batasan suatu kategori masukan ditetapkan?
- f. Sistem mempunyai toleransi jenjang dan volume data apa saja?
- g. Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi dari sistem?

