

BAB II

LANDASAN TEORI

2.1. Akuntansi Biaya

Menurut Carter (2009:144), akuntansi biaya adalah proses mencatat, menggolongkan, meringkas, dan menyajikan biaya dengan cara tertentu dalam memproduksi dan menjual suatu barang atau jasa. Biaya merupakan pengorbanan sumber ekonomi, yang diukur dalam satuan uang, yang telah terjadi atau yang kemungkinan akan terjadi untuk tujuan tertentu. Dalam akuntansi biaya, penyajian informasi mengenai harga pokok produk dibagi menjadi dua metode, yaitu metode harga pokok pesanan (*job order cost method*) dan metode harga pokok proses (*process cost method*).

2.2. Metode Harga Pokok Pesanan (*Job Order Cost Method*)

Menurut Mulyadi (2009), sistem perhitungan biaya berdasarkan pesanan mengakumulasi biaya produksi untuk setiap pesanan yang terpisah. Rincian mengenai suatu pesanan dicatat dalam kartu harga pokok (*job order cost sheet*). Metode ini mengakumulasi biaya bahan baku langsung, tenaga kerja langsung, dan *overhead* yang dibebankan ke setiap pesanan. Rumus perhitungan biaya produksi:

$$= \quad + \quad +$$

Keterangan:

HP Produksi : Harga Pokok Produksi

BBBL : Biaya Bahan Baku Langsung

BTKL : Biaya Tenaga Kerja Langsung

BOP : Biaya *Overhead* Pabrik

2.2.1. Biaya Bahan Baku Langsung

Bahan baku langsung merupakan semua bahan baku yang membentuk bagian integral dari produk jadi dan dimasukkan secara eksplisit dalam perhitungan biaya produk. Contoh dari bahan baku langsung adalah kayu yang digunakan untuk membuat mebel dan minyak mentah yang digunakan untuk membuat bensin. Kemudahan penelusuran item bahan baku ke produk jadi merupakan pertimbangan utama dalam mengklasifikasikan biaya sebagai bahan baku langsung. Misalnya, jumlah paku di mebel merupakan bagian integral dari barang jadi, tetapi karena biaya dari paku yang diperlukan untuk setiap mebel tidak signifikan, maka paku diklasifikasikan sebagai bahan baku tidak langsung.

Menentukan biaya bahan baku meliputi kuantitas dan spesifikasi setiap bahan baku yang digunakan dalam membuat satu satuan produk dan harga setiap bahan baku tersebut. Rumus perhitungan yang digunakan adalah sebagai berikut:

$$= \times$$

Keterangan:

BBBL = Biaya Bahan Baku Langsung

KBBL = Kuantitas Bahan Baku Langsung

HBBL = Harga Bahan Baku Langsung tiap produk

2.2.2. Biaya Tenaga Kerja Langsung

Tenaga kerja langsung adalah tenaga kerja yang melakukan konversi bahan baku langsung menjadi produk jadi dan dapat dibedakan secara layak ke produk tertentu. Di pabrik yang sangat terotomasi, dua masalah sering muncul ketika usaha untuk mengidentifikasi tenaga kerja langsung sebagai elemen biaya yang terpisah dilakukan. Pertama, pekerja yang sama melakukan berbagai tugas. Mereka dapat bergantian mengerjakan tugas tenaga kerja langsung kemudian tugas tenaga kerja tidak langsung secara cepat dan sering, sehingga biaya tenaga kerja langsung dan tidak langsung menjadi sangat sulit bahkan tidak mungkin dipisahkan. Kedua, tenaga kerja langsung mungkin merupakan bagian yang tidak signifikan dari total biaya produksi, membuat hal tersebut menjadi sulit untuk menjustifikasi identifikasi biaya tenaga kerja langsung sebagai elemen biaya terpisah. Dalam *setting* dimana satu atau dua situasi tersebut ada, satu klasifikasi biaya konversi mencukupi, sehingga bahan baku langsung menjadi satu-satunya elemen biaya yang ditelusuri secara langsung ke produk.

Setiap memproduksi pesanan pelanggan, pesanan mempunyai daftar kegiatan produksi untuk menunjang penjadwalan serta alat dalam menciptakan suatu produk. Dengan kata lain harus mempunyai suatu daftar kegiatan produksi atau *bill of operational*. Daftar kegiatan produksi ini mempunyai rumus untuk menentukan jam kerja yang dibutuhkan beserta tarif tenaga kerja langsung dalam setiap kali produksi pesanan pelanggan sebagai berikut:

=

Keterangan:

JKD = Jam kerja dibutuhkan

JP = Jumlah Pesanan

TJK = Total Jam Kerja

Untuk menentukan tarif tenaga kerja langsung dalam memproduksi pesanan pelanggan digunakan rumus sebagai berikut:

= /

Keterangan:

TTKL = Tarif Tenaga Kerja Langsung

TPH = Tarif per hari

JK = Jam Kerja

Untuk menentukan biaya tenaga kerja langsung, ditunjukkan dengan rumus perhitungan sebagai berikut:

$$BTKL = JKD \times TTKL$$

Keterangan:

BTKL = Biaya Tenaga Kerja Langsung

JKD = Jam Kerja Dibutuhkan

TTKL = Tarif Tenaga Kerja Langsung

2.2.3. Biaya Overhead Pabrik

Overhead pabrik terdiri atas semua biaya manufaktur yang tidak ditelusuri secara langsung ke *output* tertentu. Overhead pabrik biasanya memasukan semua

biaya manufaktur kecuali bahan baku langsung dan tenaga kerja langsung. Menurut Mulyadi (2009:193-207) Dalam tahap menyusun anggaran biaya *overhead* pabrik, tingkat kegiatan (kapasitas) digunakan sebagai dasar penaksiran biaya *overhead* pabrik. Salah satu kapasitas yang digunakan adalah kapasitas normal (*normal capacity*), yaitu kemampuan perusahaan untuk memproduksi dan menjual produknya dalam jangka panjang. Tahap selanjutnya, ada berbagai macam dasar pembebanan biaya *overhead* pabrik, salah satunya adalah jam mesin. Dasar ini digunakan jika biaya *overhead* pabrik mempunyai hubungan erat dengan waktu penggunaan mesin. Rumus perhitungan dasar pembebanan biaya *overhead* pabrik adalah sebagai berikut:

= +

Untuk mencari taksiran biaya *overhead* pabrik, harus dibedakan terlebih dahulu antara biaya *overhead* pabrik tetap dengan biaya *overhead* pabrik variable

1) Biaya Overhead Pabrik Tetap

Biaya *overhead* pabrik tetap pada UMKM Burhan Foam, terdiri dari biaya tenaga kerja tidak langsung, dan biaya pemakaian mesin. Berikut merupakan rumus-rumus yang digunakan dalam menghitung biaya *overhead* tetap diantaranya:

a) $BTKTL = (UTKTL \div KP) \times JTKL$

Keterangan:

BTKTL : Biaya Tenaga Kerja Tidak Langsung

UTKTL : Biaya Tenaga Kerja Tidak Langsung

KP : Kapasitas Produksi (bulan)

JTKL : Jam Tenaga Kerja Langsung

$$b) \text{ BPM} = \text{TPMJ} \times \text{JJM}$$

Keterangan:

BPM : Biaya Penyusutan Mesin

TPMJ : Tarif Penyusutan Mesin per Jam

JJM : Jumlah Jam Mesin

2) Biaya Overhead Pabrik Variabel

Biaya *overhead* pabrik variabel pada UMKM Burhan Foam, terdiri dari biaya bahan penolong, dan biaya listrik. Berikut merupakan rumus-rumus yang digunakan dalam menghitung biaya *overhead* variabel diantaranya:

$$a) \text{ BBP} = \sum (\text{JB} \times \text{HBP} \times \text{JP})$$

Keterangan:

BBP : Biaya Bahan Penolong

JB : Jumlah BOM

HBP : Harga Bahan Penolong

JP : Jumlah Pesanan

$$b) \text{ BL} = \sum (\text{DLM} \times \text{WPM} \times \text{TDL})$$

Keterangan:

BL : Biaya Listrik

WPM : Waktu Pemakaian Mesin

DLM : Daya Listrik Mesin

TDL : Tarif Dasar Listrik

2.3. Landasan Teori Berdasarkan Ilmu Terkait

Pada bab ini akan dijelaskan tentang landasan teori berdasarkan ilmu yang terkait yang akan digunakan untuk pembuatan rancang bangun aplikasi penentuan harga pokok produksi pada UMKM Burhan Foam. Terkait dalam pembuatan aplikasi ada beberapa teori yang terkait dengan ilmu yang nantinya akan dibuatkan program ada diantaranya, *System Development Life Cycle (SDLC)*, Bagan Alir, *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, *Visual Basic .Net*.

2.3.1. Analisis dan Desain Sistem Informasi

Menurut *Kendall (2003)*, Informasi ibarat yang mengalir di dalam tubuh suatu organisasi sehingga informasi ini sangat penting di dalam organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya berakhir. Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Tahap analisis sistem dilakukan setelah tahap perencanaan sistem (*system planning*) dan sebelum tahap desain sistem (*design system*). Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya.

a. Konsep dasar sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau

elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut: “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”. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Prosedur (*procedure*) didefinisikan oleh Richard F. Neuschel dalam buku Jugiyanto HM sebagai berikut: “Suatu prosedur adalah suatu urutan-urutan operasi klerikal (tulis menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.” Lebih lanjut Jerry FitzGerald dalam buku Jugiyanto HM mendefinisikan prosedur sebagai berikut: “Suatu prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa yang harus dikerjakan, siapa yang mengerjakannya, kapan dikerjakan dan bagaimana mengerjakannya.”

b. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), keluaran (*output*), pengolah (*process*) dan sasaran (*objective*) atau tujuan (*goal*).

1) Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-

sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut dengan *supra system*.

2) Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3) Lingkungan Luar Sistem

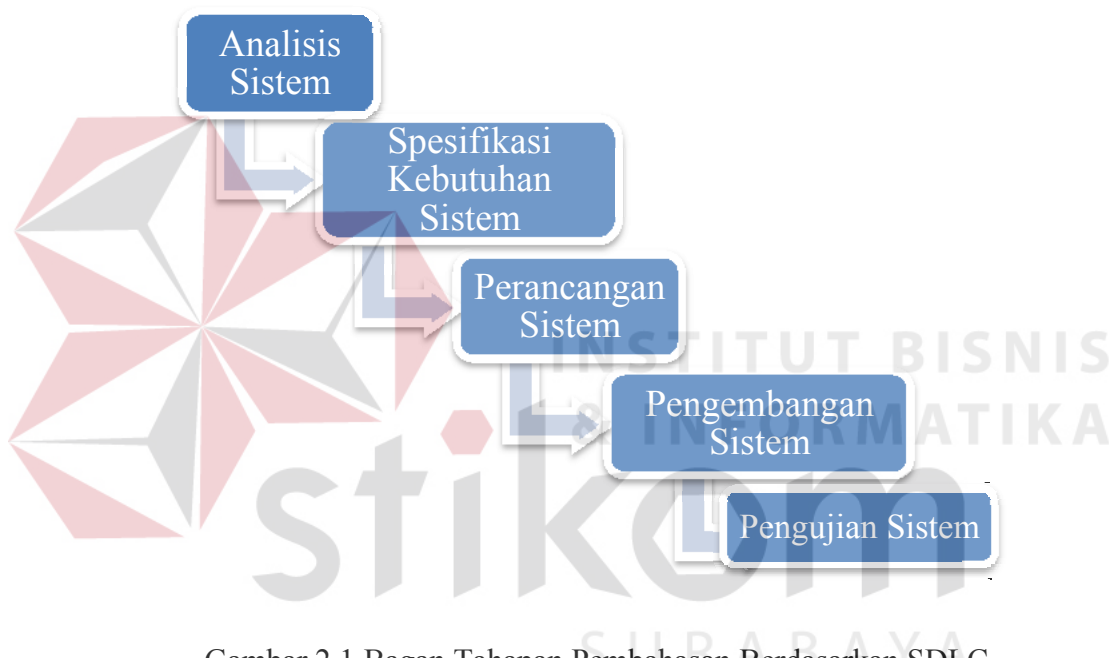
Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4) Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.3.2. *System Development Life Cycle (SDLC)*

Menurut Cahyono (2013:2) *waterfall* ini didapat dari rekayasa lain yang saling terkait, model ini menwarkan pembuatan perangkat lunak secara lebih nyata yaitu sesuai dengan tahapan, analisis sistem, spesifikasi kebutuhan sistem, perancangan sistem, pengembangan sistem, pengujian sistem. Dapat dilihat gambar 3 seperti dibawah:



Gambar.2.1 Bagan Tahapan Pembahasan Berdasarkan SDLC

2.3.3. **Bagan Alir**

Menurut Jogiyanto (2003:795), Bagan alir merupakan bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Terdapat beberapa jenis bagan alir yang biasa digunakan yakni:

- a. Bagan Alir Sistem (*system flowchart*)

Bagan alir Sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Bagan alir sistem digambar dengan menggunakan symbol-simbol tertentu.

b. Bagan Alir Dokumen (*document flowchart*)

Bagan alir dokumen atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir ini menggunakan symbol-simbol yang sama seperti yang digunakan di dalam bagan alir sistem.

c. Bagan Alir Skematik (*scematic flowchart*)

Bagan alir skematik merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar computer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir.

d. Bagan Alir Program (*program flowchart*)

Bagan alir program merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari diverifikasi bagan alir sistem. Bagan alir program menggunakan simbol-simbol tertentu.

e. Bagan Alir Proses (*process flowchart*)

Bagan alir proses merupakan bagan alir yang banyak digunakan di teknik industry. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan

proses dalam suatu prosedur. Bagan alir proses menggunakan lima buah simbol tersendiri.

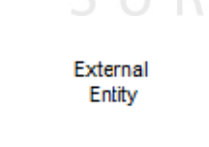
Berdasarkan penjelasan di atas, dapat ditarik kesimpulan bahwa bagan alir (*flowchart*) adalah suatu gambaran umum tentang sistem yang berjalan dan berfungsi sebagai alat bantu komunikasi serta untuk mendokumentasikan dan menyajikan kegiatan mulai dari manual, semi manual maupun komputerisasi.

2.3.4. *Data Flow Diagram (DFD)*

Menurut Indrajani (2011:11), *data flow diagram* (DFD) adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut. Istilah dalam bahasa Indonesianya adalah diagram aliran data. Dalam DFD ini terdapat empat komponen utama, yaitu:

a. *External Entity*

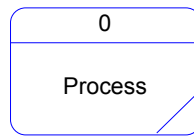
Entitas Eksternal mendefinisikan orang atau sebuah unit organisasi, sistem lain, atau organisasi yang berada di luar sistem proyek tetapi dapat mempengaruhi kinerja sistem.



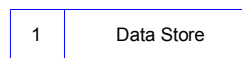
Gambar 2.2 *External Entity*

b. *Process*

Proses adalah penyelenggaraan kerja atau jawaban, datangnya aliran data atau kondisinya.

Gambar 2.3 *Process*c. *Data Stores*

Data Stores adalah penyimpanan data.

Gambar 2.4 *Data Stores*d. *Data Flow*

Data flow mempresentasikan sebuah input data ke dalam sebuah proses atau output dari data atau informasi pada sebuah proses.

Gambar 2.5 *Data Flow*

2.3.5. *Entity Relationship Diagram (ERD)*

Menurut Simarmata (2010:96) Model Entity-Relationship (ER) mula-mula diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entitas*) dan hubungan (*relationship*). Komponen dasar model merupakan diagram *entity-relationship* yang digunakan untuk menyajikan objek data secara visual. Chen menulis bahwa model telah diperluas dan kini bias digunakan untuk merancang basis data. Kegunaan model ER dalam perancangan tersebut adalah:

- a. Mampu memetakan model relasional dengan baik. Pembangunan yang digunakan di dalam model ER dengan mudah diubah ke dalam table relasional
- b. Sederhana dan mudah dipahami hanya dengan sedikit pelatihan. Oleh karena itu, model bisa digunakan oleh perancangan basis data untuk mengkomunikasikan perancangan kepada pengguna akhir.
- c. Sebagai tambahan, model bisa digunakan sebagai suatu rencana perancangan oleh pengembangan basis data untuk menerapkan suatu model data dalam perangkat lunak manajemen basis data spesifik.

a. Entitas

Entitas (Entities) adalah objek data prinsip tentang informasi yang dikumpulkan. Entitas pada umumnya berupa konsep yang bisa dikenal, baik konkret maupun abstrak, seperti orang, tempat, benda, atau peristiwa yang memiliki keterkaitan terhadap basis data.

b. Atribut

Atribut menguraikan entitas dimana mereka dihubungkan. Kejadian dari suatu atribut tertentu adalah suatu nilai (*value*). Sebagai contoh, “Janner Simamarta” adalah satu nilai dari atribut Nama. Domain dari suatu atribut merupakan koleksi dari segala kemungkinan atas nilai-nilai atribut yang dimiliki.

2.3.6. Visual Basic .Net

Menurut Yuswanto (2007:23) *Visual Basic .Net* 2008 adalah salah satu bahasa pemrograman yang tergabung dalam *Microsoft Visual Studio* 2008. *Visual*

Studio 2008 dan *Microsoft .Net Framework 4.0* membantu *developer* menghasilkan performansi yang lebih baik dan menghasilkan aplikasi yang *scapable*.

2.3.7. Microsoft SQL Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar. Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai driver JDBC untuk bahasa pemrograman Java.

2.3.8. Testing Software

Menurut Romeo (2003), *Testing Software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan, untuk verifikasi apakah telah berlaku sebagaimana telah ditetapkan (menurut spesifikasinya), 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:

- 1) Apakah *software* telah siap digunakan ?
- 2) Apa saja resikonya ?
- 3) Apa saja kemampuannya ?
- 4) Apa saja keterbatsannya ?
- 5) Apa saja masalahnya ?
- 6) Apakah telah berlaku seperti yang diharapkan ?

2.3.9. 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:

- 1) Untuk melakukan *testing* kesesuaian suatu komponen terhadap spesifikasi (*Black Box Testing*)
- 2) Untuk melakukan *testing* kesesuaian suatu kompone terhadap desain (*White Box Testing*)

2.3.10. 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*.

Dengan adanya *black box testing*, perancang *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan fungsional pada suatu program. Kategori *error* yang akan diketahui melalui *black box testing* adalah sebagai berikut:

- 1) Fungsi yang hilang atau tidak 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

2.3.11. *White Box Testing*

White Box Testing merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan *output* yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian dilakukan *compile* ulang. Dengan menggunakan *white box* akan didapatkan kasus uji yang :

- 1) Menguji semua keputusan logika
- 2) Menguji seluruh *Loop* yang sesuai dengan batasannya
- 3) Menguji seluruh struktur data internal yang menjamin validitas

