

BAB II

LANDASAN TEORI

2.1. Perancangan

Menurut (Mulyadi, 2007) perancangan adalah suatu fase yang diawali dengan evaluasi atas alternative rancangan system yang diikuti dengan penyiapan spesifikasi rancangan yang berorientasi kepada pemakai tertentu dan diakhiri dengan pengajuan rancangan pada manajemen puncak.

Menurut (Jogiyanto, 2005) desain sistem dapat didefinisikan sebagai penggambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah dari satu kesatuan yang utuh dan berfungsi.

2.2. Aplikasi

Aplikasi merupakan sekumpulan perintah program yang dibayar untuk melakukan pekerjaan-pekerjaan tertentu (khusus). Dalam pembuatan aplikasi *file text* dibutuhkan beberapa komponen seperti *label*, *textbox*, dan beberapa *command*, dimana aplikasi *file text* merupakan sebuah program yang dapat menyimpan *text* atau tulisan ke dalam *extension* .txt. Pada pembuatan aplikasi tentu tersedia menu untuk melakukan *print form*, dimana *print form* sendiri merupakan komponen untuk mencetak *form* ke *file*, *preview* atau *printer*. Komponen tersebut terdapat pada *ToolBox Printing* (Hendrayudi, 2009).

2.3. Harga Jual

Penentuan harga jual merupakan sebuah keputusan sangat penting bagi perusahaan, Karena hal ini sangat berpengaruh terhadap keberhasilan perusahaan dalam mencapai tujuannya. Harga jual tinggi akan membuat masyarakat tidak membeli atau mengurangi jumlah pembelian produk, sehingga perusahaan tidak akan memperoleh pendapatan dan laba yang cukup. Sebaliknya, harga jual yang terlalu rendah akan membuat perusahaan tidak mencapai laba usaha yang direncanakan. Karena itu, menetapkan harga jual produk pada harga yang tepat sangatlah penting agar tujuan perusahaan secara umum dapat tercapai (Rudianto, 2013).

Harga jual adalah besarnya harga yang akan dibebankan kepada konsumen yang diperoleh atau dihitung dari biaya produksi ditambah biaya nonproduksi dan laba yang diharapkan (Mulyadi, 2005).

Harga Jual = Harga Pokok + (Persentase *Markup* x Harga Pokok) ... Rumus 1

Contoh perhitungan :

$$ROI = \frac{\text{Laba}}{\text{Investasi}}$$

$$\frac{3}{5} \times 100\% = \frac{\text{Laba}}{\text{Rp 5.000.000}}$$

$$\text{Laba Yang Diharapkan} = 60\% \times \text{Rp 30.000.000}$$

$$= \text{Rp 18.000.000}$$

$$\% \text{markup} = \frac{\text{Target ROI} + \text{Biaya Tetap} + \text{Biaya Non Produksi Variabel}}{(\text{Volume dalam unit}) \times (\text{Biaya produksi non variabel per unit})}$$

$$= \frac{(60\% \times \text{Rp 30.000.000}) + \text{Rp 16.000.000} + \text{Rp 4.000.000}}{100 \times \text{Rp 5.000}}$$

$$= \frac{\text{Rp } 38.000.000}{\text{Rp } 500.000}$$

$$= 76\%$$

2.4. Laba yang Diharapkan (Target ROI)

Return On Investment (ROI) merupakan salah satu dari rasio profabilitas dimana rasio ini menunjukkan kemampuan perusahaan memperoleh laba dalam hubungannya dengan penjualan, total aktiva maupun modal sendiri. Rasio profabilitas ini sangat diperhatikan oleh calon maupun pemegang saham karena akan berkaitan dengan harga saham deviden yang akan diterimanya.

Pada proses ini, melalui informasi rerata aktiva operasional menjadi *input* dari laba yang diharapkan (target ROI). Isi dari rerata aktiva operasional adalah hasil penjumlahan dari aktiva lancar dan aktiva tidak lancar yang nantinya akan dihitung dengan hasil perhitungan ROI untuk menentukan laba yang diharapkan. Untuk menentukan nilai ROI, perusahaan harus menargetkan berapa tahun investasi yang digunakan untuk aktiva operasional akan kembali. Berikut rumus yang digunakan menurut (Garrison, Noreen, & Brewer, 2007) dan (Sugiri, 2009):

a. Aktiva operasional = aktiva lancar + aktiva tidak lancar.....Rumus 2

Keterangan:

Aktiva lancar = hasil penjumlahan kas, piutang, dan persediaan.

Aktiva tidak lancar = hasil penjumlahan yang meliputi pabrik dan peralatan, mesin, dan aktiva lain.

b. $ROI = \frac{\text{aktiva operasional}}{N}$Rumus 3

Keterangan:

ROI : *Return On Investment*

N : berapa tahun investasi kembali (maksimal umur mesin 5 tahun)

c. $\%ROI = \frac{ROI}{\text{aktiva operasional}} \times 100\%$Rumus 4

Keterangan:

$\%ROI$: Persentase *Return On Investment*

ROI : *Return On Investment*

d. Laba Yang Diharapkan = $\%ROI \times \text{aktiva operasional}$Rumus 5

Keterangan:

$\%ROI$: Persentase *Return On Investment*

Rerata aktiva operasional : total aktiva lancar dan aktiva tidak lancar

e. Laba Yang Diharapkan tiap Jam Kerja = $\frac{\text{Laba Yang Diharapkan}}{12 \times 26 \times 8 \times 18}$Rumus 6

Keterangan:

12 : 12 bulan dalam 1 tahun

8 : 8 jam kerja sehari

26 : 26 hari kerja selama 1 bulan

18 : 18 tenaga kerja

f. $LYD \text{ Produk} = LYD \text{ tiap Jam Kerja} \times LPD$Rumus 7

Keterangan:

LYD : Laba Yang Diharapkan

LPD : Lama Produk Diproduksi

2.5. Metode *Variable Costing*

Menurut (Sugiri, 2009) *Variable Costing* hanya terdiri atas biaya variabel yang diperlukan untuk memproduksi barang atau jasa. Elemen biaya produk

meliputi biaya bahan baku, biaya tenaga kerja langsung, dan biaya overhead pabrik variabel.

Harga Jual =

Biaya produksi variabel + Biaya non produksi + Markup Rumus 8

%Markup

$$= \frac{\text{Target ROI} + \text{Biaya Tetap} + \text{Biaya nonproduksi variabel}}{\text{Volume dalam unit} \times \text{Biaya produksi variabel per unit}} \dots \dots \dots \text{Rumus 9}$$

Biaya Administrasi dan Penjualan = PJTK x BP x LP Rumus 10

Keterangan:

PJKTK : Perbandingan Jam Tenaga Kerja

BP : Biaya Pemakaian

LP : Lama Pemakaian

$$\text{Perbandingan Jam Tenaga Kerja} = \frac{\text{WJTK}}{\text{Seluruh Jam Tenaga Kerja}} \dots \dots \dots \text{Rumus 11}$$

Keterangan:

Perbandingan Jam Tenaga Kerja : Hasil perbandingan waktu jam tenaga kerja yang dibutuhkan untuk memproduksi produk terhadap seluruh jam tenaga kerja.

WJTK : Waktu Jam Tenaga Kerja yang dibutuhkan untuk memproduksi produk yang akan diproduksi.

Seluruh Jam Tenaga Kerja : total seluruh jam tenaga kerja selama 1 bulan atau 26 hari.

2.6. Biaya Overhead Pabrik

Biaya *overhead* pabrik menurut Krismiaji (2002:19), yaitu biaya yang mencakup seluruh biaya produksi tidak langsung. Contoh biaya yang masuk

dalam kelompok biaya *overhead* pabrik adalah bahan baku tidak langsung, biaya tenaga kerja tidak langsung, reparasi dan pemeliharaan peralatan pabrik, dan biaya depresiasi dan asuransi fasilitas pabrik.

2.7. Software Development Cycle (SDLC)

Dibutuhkan beberapa metode dan arsitektur dalam perancangan, pengembangan, dan implementasi Sistem Informasi. *System Development Life Cycle* (SDLC) adalah keseluruhan proses dalam membangun sistem melalui beberapa langkah. Model yang cukup populer dan banyak digunakan adalah *waterfall*. Beberapa model lain SDLC misalnya *fountain*, *spiral*, *rapid prototyping*, *incremental*, *build fix*, dan *synchronize & stabilize*. Dengan siklus SDLC proses membangun sistem dibagi menjadi beberapa langkah dan masing-masing langkah dikerjakan oleh tim yang berbeda.

Dalam pengembangan sistem ini menggunakan proses *Waterfall Model*. Definisi yang diartikan oleh Roger S. Pressman (2015) dalam bukunya *Software Engineering A Practitioner's Approach Seventh Edition* dijelaskan bahwa:

“*Waterfall Model* adalah sebuah perancangan yang secara berurutan dan sering digunakan dalam proses pengembangan perangkat lunak.”

2.8. Testing

Menurut Romeo (2003), testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. *Testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi

dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Vaidasi adalah melihat kebenaran sistem apakah proses yang dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut tidak terjadi. *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 desain *White Box Testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi *Black Box Testing*.

2.9.1. *White Box Testing*

Menurut Romeo (2003), *white box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain procedural. Seringkali *white box testing* diasosiasikan dengan pengukuran cakupan tes yang mengukur persentase jalur-jalur dari tipe yang dipilih untuk dieksekusi oleh *test case*. *White box testing* dapat menjamin semua struktur internal data dapat di tes untuk memastikan validasinya.

Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test case* alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program. Alur logika suatu program dapat dipresentasikan dengan *flowgraph*.

2.9.2. *Black Box Testing*

Menurut Romeo (2003), *black box testing* dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal sistem atau komponen yang di tes juga disebut sebagai fungsional 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 kebutuhan pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.

