BAB III

LANDASAN TEORI

3.1 Aplikasi

Menurut Rizky (2009:32), aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft *Word*, *Microsoft Excel*. Menurut Kadir (2005), perangkat lunak aplikasi *(application software)* merupabkan program yang biasa dipakai oleh pemakai untuk melakukan tugas-tugas yang spesifik; misalnya untuk membuat dokumen, memanipulasi foto, atu membuat laporan keuangan.

Dari pengertian di atas, dapat disimpulkan bahwa aplikasi merupakan perangkat lunak uang difungsikan sebagai alat bantu untuk mengerjakan hal-hal tertentu yang dapat memudahkan jalannya proses bisnis atau kegiatan suatu perusahaan ataupun organisasi.

3.2 Pengertian Distribusi

Menurut Kotler (2009:106), saluran distribusi adalah sekelompok organisasi yang saling bergantung dan terlibat dalam proses pembuatan produk atau jasa yang disediakan untuk digunakan atau dikonsumsi.

Menurut Basu (2008:285), saluran distribusi merupakan suatu struktur organisasi dalam perusahaan dan luar perusahaan yang terdiri atas agen, dealer, pedagang besar, dan pengecer, melalui sebuah komoditi, produk, atau jasa dipasarkan.

Untuk memperlancar arus barang atau jasa dari produsen ke konsumen, salah satu faktor yang penting adalah memilih saluran secara tepat saluran distribusi yang dipergunakan untuk menyampaikan barang atau jasa dari produsen ke konsumen. Dari kedua definisi di atas dapat ditarik kesimpulan bahwa saluran distribusi merupakan seperangkat alur yang diikuti produk atau jasa setelah produksi berakhir dalam pembelian dan digunakan oleh pengguna akhir.

3.3 *Retail*

Menurut Hendri (2005:7), *retail* adalah kegiatan usaha menjual barang atau jasa kepada perorangan untuk keperluan diri sendiri, keluarga atau rumah tangga. Menurut Levy (2009:48), *retailing* merupakan serangkaian kegiatan usaha yang memberikan nilai tambah pada produk dan jasa yang dijual kepada pelanggan untuk penggunaan pribadi atau keluarga.

Gambar 3.1 Proses Penjualan Barang (Cristina, 2006:5)

3.4 *Retailing Mix* (Bauran Eceran)

Menurut Kotler (2012), *retailing mix* adalah sekelompok perlengkapan pemasaran yang digunakan perusahaan untuk mencapai sasaran pemasarannya dalam pasar sasaran, bauran eceran meliputi semua tindakan yang dapat dilakukan perusahaan untuk mempengaruhi permintaan akan produknya itu sendiri dan semua tindakan yang mungkin dilakukan oleh perusahaan itu dapat disimpulkan sebagai

satu kelompok variabel yang diantaranya adalah produk, lokasi, harga, dan promosi.

Munir (2010), memberikan definisi bahwa *retailing mix* adalah: *"This mix is a firm particular combination of this factor: store location, operating procedures, good service offered, pricing tactics, store atmosphere, customer service, and promotion methods"*. Yang artinya bahwa bauran ini gabungan dari pada faktor-faktor sebagai berikut: lokasi toko, prosedur operasi, pelayanan yang ditawarkan, strategi penetapan harga, suasana toko, pelayanan konsumen, dan metode promosi.

3.5 Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna bagi penerimanya yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan (Jogiyanto, 2005). Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi (Kadir, 2009).

3.6 Diagram Alir Dokumen *(Document Flowchart)*

Menurut Jogiyanto (2005: 795), *document flowchart* adalah diagram alir dokumen atau bisa disebut juga sebagai bagan alir formulir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Menurut James (2007:75), bagan 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 (baik klerikal maupun fisikal) yang dilakukan dalam departemen tersebut.

Document flowchart memiliki ketentuan-ketentuan yang harus diperhatikan dalam proses pembuatannya. Termasuk notasi-notasi yang terdapat di di dalamnya. Simbol-simbol yang yang digunakan dalam pembuatan *document flowchart* dijelaskan pada tabel berikut (Jogiyanto, 2005).

3.7 Diagram Alir Sistem *(System Flowchart)*

Menurut Krismiaji (2005:75), bagan alir sistem menggambarkan hubungan antara input, pemrosesan dan output sebuah sistem informasi akuntansi. Bagan alir sistem ini dimulai dengan identifikasi input yang masuk ke dalam sistem dan sumbernya.

Menurut Jogiyanto (2005), bagan alir program (*system flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Bagan ini menjelaskan urut-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Pembuatan *system flowchart* memiliki aturan dan ketentuan yang harus diikuti. Seperti halnya dalam pembuatan *document flowchart* sebelumnya, *system flowchart* memiliki notasi-notasi sebagai representasi dari proses kerja suatu sistem. Sebagian notasi dalam *system flowchart* memiliki kesamaan dengan notasi yang ada pada *document flow* seperti, terminator (*start/end*), dan notasi laporan. Selain kedua notasi tersebut terdapat perbedaan secara bentuk dan fungsinya.

System flowchart ini tidak digunakan untuk menggambarkan langkah-langkah pemecahalan masalah, namun hanya digunakan untuk menggambarkan prosedur pada sistem yang dibuat. Berikut simbol-simbol yang digunakan dalam pembuatan *system flowchart.*

3.8 Diagram Konteks *(Context Diagram)*

Menurut Jogiyanto (2005), diagram konteks merupakan diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks adalah level tertinggi dari *Data Flow Diagram* (DFD) yang menggambarkan seluruh inputan ke sistem atau output dari sistem. Pada proses inilah gambaran mengenai keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses dan tidak ada store di dalam diagram konteks.

Menurut Oetomo (2002), terdapat beberapa hal yang perlu diperhatikan dalam pembuatan diagram konteks, antara lain:

1. Kelompok pemakai, baik internal maupun eksternal perusahaan.
2. Identifikasi kejadian-kejadian yang mungkin terjadi dalam penggunaan sistem.
3. Arah anak panah yang menunjukkanaliran data.
4. Setiap proses digambarkan dalam bentuk yang sederhana dan mudah dipahami oleh pembuat sistem.

Proses dalam diagram konteks biasanya diberi nomor 0. Proses ini merupakan proses dari seluruh sistem dengan dunia luarnya.

3.9 *Data Flow Diagram* (DFD)

Data flow diagram adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam obyek kemudian melewati suatu proses yang mentransformasikan ke tujuan lain, yang ada pada objek lain.

Menurut Jogiyanto (2010:700) dalam bukunya yang berjudul Analisis & Disain, menjelaskan bahwa:"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. Data Flow Diagram juga digunakan pada metodologi pengembangan sistem yang terstruktur.

Dari kedua penjelasan di atas dapat disimpulkan bahwa, *Data Flow Diagram* (DFD) berfungsi untuk menggambarkan sistem baru ataupun sistem yang telah ada yang arus datanya akan dikembangkan secara terstruktur. Tingkatan *Data Flow Diagram* (DFD) terdiri dari:

1. Diagram Konteks (Context Diagram)

   Diagram konteks merupakan sebuah model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem.

2. Diagram Level 0

   Diagram level 0 merupakan diagram aliran data yang menggambarkan sebuah event konteks. Diagram ini menunjukkan interaksi antara input, output, dan data store pada setiap proses yang ada.

3. Diagram Level 1

   Diagram level 1 menggambarkan rincian dari diagram level 0. Setiap proses yang terdapat pada diagram level 0 akan dijelaskan pada proses ini.

3.10 *Entity Relationship Diagram* (ERD)

Menurut Ladjamudin (2005:142), *Entity Relationship Diargam* (ERD) dalam buku yang berjudul Analisis dan Desain Sistem Informasi adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak.

Menurut Fathansyah (2012:79), *Entity Relationship Diagram* (ERD) dalam buku yang berjudul Basis Data adalah model *Entity-Relationship* yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atributatribut yang mempresentasikan seluruh fakta dari dunia nyata yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Entity Relationship Diagram* (ERD).

Berdasarkan definisi di atas dapat disimpulkan bahwa diagram relasi entitas adalah model jaringan yang menggunakan susunan data yang disimpan secara abstrak. *Entity Relation Diagram* mempunyai 2 jenis model dan 4 jenis objek, berikut penjelasannya:

Jenis model *Entity Relation Diagram* (ERD), sebagai berikut:

1. *Conceptual Data Model* (CDM)

Merupakan model yang universal dan dapat menggambarkan semua struktur *logic database* (DBMS), dan tidak bergantung dari *software* atau pertimbangan struktur data *storage.* Sebuaj CDM dapat diubah langsung menjadi *Physical Data Model* (PDM).

2. *Physical Data Model* (PDM)

Merupakan model ERD yang mengacu pada pemilihan *software* yang spesifik. Hal ini sering kali berbeda dikarenakan oleh struktur *database* yang bervariasi, mulai dari model *schema*,tipe data penyimpanan dan sebagainya.

Jenis objek *Entity Relation Diagram* (ERD), sebagai berikut:

1. *Entity*

Sesuatu yang ada dan terdefinisikan dapat berupa nyata ataupun abstrak yang dapat dibedakan satu dengan yang lainnya dan adanya hubungan saling ketergantungan.

2. *Attribute*

Setiap *entity* memiliki beberapa *attribute*, yang merupakan ciri atau karakteristik dari *entity* tersebut. *Attribute* disebut juga dengan data elemen atau data *field*. Berikut adalah macam-macam attribute, antara lain:

a. *Simple Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya *entity* mahasiswa yang *attribute*-nya NIM.

b. *Composite Attribute*

Composite Attribute adalah *attribute* yang memiliki dua nilai harga, misalnya nama besar (nama keluarga) dan nama kecil (nama asli).

c. *Single Value Attribute*

Attribute ini hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya umur (tanggal lahir).

d. *Multi Value Attribute*

Multi Value Attribute adalah *attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD, SMP, SMA).

e. *Null Value Attribute*

Null Value Attribute adalah *attribute* tidak memiliki nilai harga, misalnya *entity* tukang becak dengan *attribute*-nya pendidikan (tanpa memiliki ijazah).

3. *Key*

Beberapa elemen data memiliki sifat, dengan mengetahui nilai yang telah diberikan oleh sebagian elemen data dari *entity* tertentu, dapat diidentifikasi nilai-nilai yang terkandung dalam elemen-elemen data lain ada *entity* yang sama. Elemen penentu tersebut adalah sebagau elemen data kunci *(key).*

4. *Relationship*

Relationship menggambarkan hubungan yang terjadi antara *entity* yang mewujudkan pemetaan antar *entity*. Berikut bentuk *relationship*:

a. *One to One Relationship*

Hubungan satu *entity* dengan *entity* yang lain.

b. *Many to Many Relationship*

Hubungan antara *entity* datu dengan *entity* yang lainnya adalah satu berbanding banyak.

3.11 *PHP Hypertext Preprocessor*

PHP Hypertext Prepocessor (PHP) adalah bahasa pemrograman *scripting* sisi server (*server-side*), bahasa pemrograman yang digunakan oleh *server web* untuk menghasilkan dokumen *Hypertext Markup Language* (HTML) secara *on-the-fly* (Sidik, 2005: 323).

Menurut Kadir (2008: 2), PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu penampilan berdasarkan permintaan

terkini. Misalnya, bisa menampilkan isi *database* ke halaman web. Pada prinsip PHP mempunyai fungsi yang sama dengan skrip-skrip seperti *Active Server Page* (ASP), Cold Fusion, atau Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, Skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

3.12 MySQL

MySQL merupakan *software* sistem manajemen database *(Database Management System - DBMS)* yang sangat popular di kalangan pemrogram *web*, terutama di lingkungan Linux dengan menggunakan *script* PHP dan Perl. *Software database* ini kini telah tersedia juga pada *platform* sistem operasi Windows (98/ME atau pun NT/2000/XP). *Database* MySQL, merupakan *database* yang menjanjikan sebagai alternatif pilihan database yang dapat digunakan untuk sistem *database* personal atau organisasi kita. Oracle sebagai database baru telah membuat *kit* (modul) untuk memudahkan proses migrasi dari MySQL ke dalam Oracle, hal ini dapat menunjukkan bahwa Oracle telah memperhitungkan database MySQL sebagai database alternatif masa depan. (Sidik, 2005: 1).