

BAB III

LANDASAN TEORI

3.1 Sistem

Definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu. (Herlambang dan Tanuwijaya, 2005:116)

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitar. (Herlambang dan Tanuwijaya, 2005:116)

3.2 Sistem Informasi

Data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang

digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya. (Herlambang dan Tanuwijaya, 2005:121)

3.3 Sistem Informasi Sumber Daya Manusia

Tiap perusahaan memiliki sistem untuk mengumpulkan dan memelihara data yang menjelaskan sumber daya manusia, mengubah data tersebut menjadi informasi, dan melaporkan informasi itu kepada pemakai. Sistem ini dinamakan sistem manajemen sumber daya manusia (*human resource information system*) atau HRIS. (McLeod, 2008:523)

Sistem Informasi Sumber Daya Manusia (SISDM/HRIS) merupakan sebuah bentuk interseksi/pertemuan antara bidang ilmu manajemen sumber daya manusia (MSDM) dan teknologi informasi. Sistem ini menghubungkan MSDM sebagai suatu disiplin yang utamanya mengaplikasikan bidang teknologi informasi ke dalam aktifitas-aktifitas MSDM seperti dalam hal perencanaan, dan menyusun sistem pemrosesan data dalam serangkaian langkah-langkah yang terstandarisasi dan terangkum dalam aplikasi perencanaan sumber daya perusahaan/*enterprise resource planning* (ERP). (McLeod, 2008:523)

Secara keseluruhan sistem ERP bertujuan mengintegrasikan informasi yang diperoleh dari aplikasi-aplikasi yang berbeda ke dalam satu sistem basisdata yang bersifat universal. Keterkaitan dari modul kalkulasi finansial dan modul MSDM melalui satu basisdata yang sama merupakan hal yang sangat penting yang membedakannya dengan bentuk aplikasi lain yang pernah dibuat sebelumnya,

menjadikan aplikasi ini lebih fleksibel namun juga lebih kaku dengan aturan-aturannya. (McLeod, 2008:543)

3.4 Koperasi Simpan Pinjam

Koperasi Simpan Pinjam adalah koperasi yang bergerak di bidang simpan dan pinjaman. Sedangkan pengertian koperasi simpan pinjam berdasarkan PSAK 27/Reformat 2007 adalah koperasi yang kegiatan atau jasa utamanya menyediakan jasa penyimpanan dan peminjaman untuk anggotanya.

Pengertian simpanan adalah dana yang dipercayakan oleh anggotanya kepada koperasi dalam bentuk simpanan pokok, simpanan wajib, dan tabungan. Sedangkan pinjaman adalah penyediaan uang kepada anggota berdasarkan kesepakatan pinjam meminjam, yang mewajibkan kepada peminjam melunasi hutangnya dalam jangka waktu tertentu, disertai dengan pembayaran sejumlah imbalan yang dapat berbentuk bunga atau bagi hasil. Pada dasarnya Koperasi Simpan Pinjam menjalankan fungsi yang hampir sama dengan bank, yaitu sebagai badan usaha yang melakukan penggalan atau mobilisasi dana dari masyarakat dan menyalurkannya kembali dalam bentuk kredit kepada warga masyarakat yang membutuhkan. Yang membedakannya adalah bahwa koperasi dimiliki secara bersama oleh anggotanya dengan hak dan kedudukan yang sama, dan hanya memberikan pelayanan kredit kepada anggotanya. Sedangkan bank dimiliki oleh sejumlah orang atau badan sebagai pemegang saham, memobilisasi dana dari masyarakat luas untuk menyimpan uang di bank tersebut, namun hanya menyalurkan dana yang terhimpun kepada warga masyarakat yang mampu memenuhi persyaratan teknis bank.

Prinsip-prinsip yang harus dimiliki oleh Koperasi Simpan Pinjam haruslah dijalankan dengan memperhatikan semangat dari prinsip dasar koperasi simpan pinjam rumusan Friedrich William Raiffeisen selaku pendiri pertama credit union pada pertengahan abad ke-19, yaitu :

1. Dana koperasi hanyalah diperoleh dari anggota-anggotanya saja.
2. Pinjaman juga hanya diberikan kepada anggota-anggotanya saja.
3. Jaminan terbaik bagi peminjam adalah watak si peminjaman itu sendiri.

Prinsip Koperasi Simpan Pinjam ala Friedrich William Raiffeis tersebut mencerminkan bahwa Koperasi Simpan Pinjam haruslah dibangun atas usaha dan semangat swadaya dari anggotanya melalui usaha simpan pinjam berdasarkan kerjasama dan saling percaya. Oleh sebab itu pada seluruh anggota Koperasi Simpan Pinjam haruslah ada suatu kesadaran dan tekad yang kuat untuk membangun Koperasi Simpan Pinjam secara swadaya, dimana mereka adalah anggota yang sekaligus pemilik serta pengguna jasa dari Koperasi Simpan Pinjam, dengan cara :

1. Tekad untuk tidak tergantung kepada bantuan modal dari siapapun, termasuk dari pemerintah.
2. Hanya menyimpan (menabung) uang di Koperasi Simpan Pinjam, setiap kali mempunyai kelebihan uang dari kebutuhan sehari-hari, langsung ditabung di Koperasi Simpan Pinjam.

3.5 DataBase

Database adalah suatu sistem menyusun dan mengelola *record-record* menggunakan computer untuk menyimpan atau merekam serta memelihara data

operasional lengkap dengan sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan (Linda,2004:1). *Database* dapat dinyatakan sebagai suatu sistem yang memiliki karakteristik seperti berikut :

- a. Merupakan suatu kumpulan interaksi data yang disimpan bersama dan tanpa mengganggu satu sama lain atau membentuk duplikat data.
- b. Kumpulan data di dalam *database* dapat digunakan oleh sebuah program secara optimal .
- c. Penambahan data baru, modifikasi dan pengambilan kembali dari data dapat dilakukan dengan mudah dan terorganisasi.

Dalam arsitektur *database* terdapat tiga tingkatan yang saling mendukung.

Di bawah ini adalah penjelasan yaitu :

- a. *Internal level* yaitu tingkatan yang basis datanya secara fisik ditulis atau disimpan di media *storage* dan *level* yang berkaitan.
- b. *External level* disebut juga individual *user view*, yaitu tingkat yang basis datanya dapat didasarkan kebutuhan masing-masing aplikasi di user atau level yang berkaitan dengan pemakai.
- c. *Conceptual level* disebut juga *community user view*, yaitu tingkat *user view* dari aplikasi yang berbeda digabungkan sehingga menggunakan basis data secara keseluruhan dengan menyembunyikan penyimpanan data secara fisik yang merupakan penghubung dari *internal level* dan *external level*

Seluruh operasi yang dilakukan pada *database* didasarkan atas table-table dan hubungannya. Dalam model relasional dikenal antara lain table, *record*, *field*, indeks, *query* penjelasannya seperti dibawah ini :

- a. Table atau *entity* dalam model relasional digunakan untuk mendukung antar muka komunikasi antara pemakai dengan professional computer.
- b. *Record* atau baris atau dalam istilah model relational yang formal disebut tuple adalah kumpulan data yang terdiri dari satu atau lebih.
- c. *Field* atau kolom atau dalam istilah model relasional yang formal disebut dengan *attribute* adalah sekumpulan data yang mempunyai atau menyimpan fakta yang sama atau sejenis untuk setiap baris pada tabel.
- d. Indeks merupakan tipe dari suatu table tertentu yang berisi nilai-nilai *field* kunci atau *field*.
- e. *Query* merupakan sekumpulan perintah *Structure Query Language* (SQL) yang dirancang untuk memanggil kelompok record tertentu dari satu tabel atau lebih untuk melakukan operasi pada tabel.

3.6 Microsoft Visual Basic 2005

Microsoft Visual Basic 2005 adalah bahasa pemrograman yang digunakan untuk membuat aplikasi *windows* yang berbasis grafis (*GUI – Graphical User Interface*). Untuk mendesain tampilan yang diinginkan, hanya perlu meletakkan objek-objek grafis ke lembar (*form*) yang sudah tersedia pada *Visual Basic* dan

selanjutnya hanya perlu memikirkan struktur dan logika data dari program utama. (Yuswanto, 2006:1)

Selain itu *visual basic* juga merupakan sarana (*tools*) untuk menghasilkan program-program aplikasi berbasis *windows*. Beberapa kemampuan atau manfaat dari *visual basic* adalah bisa membuat program aplikasi berbasis *windows*, dan juga dapat membuat objek-objek pembantu program seperti kontrol *active*, *file*, *help*, aplikasi internet, dan sebagainya. (Yuswanto, 2006:2)

Untuk bekerja dengan *visual basic 2005* untuk *windows*, *user* harus berada di lingkungan operasi *windows*. Secara umum lingkungan kerja *visual basic* terdiri dari lima buah *windows* yang dirancang sebagai aplikasi *multi windows*. Kelima *windows* tersebut adalah (Yuswanto, 2006:2):

1. *Form*

Berisi latar belakang program *windows* yang akan ditulis. Menggambar dan meletakkan item itu pada *form* sehingga pengguna program terbiasa melihat dan berinteraksi. *Form* akan mengendalikan program yang disunting.

2. *Toolbox*

Windows Toolbox berisi alat-alat (kontrol-kontrol), kontrol-kontrol inilah yang nantinya ditempatkan di *windows form*. Misalkan memilih kontrol *text* dan menempatkan kotak *text* pada *form*.

3. *Project*

Program *visual basic windows* yang sering berisi beberapa jenis *file* yang berbeda yang semuanya bekerja dalam satu kesatuan membentuk program tunggal yang

dijalankan. *Windows project* berisi daftar semua *file* yang digunakan. Aplikasi *visual basic* umumnya disebut dengan *project*.

4. *Properties*

Windows Properties menguraikan setiap elemen individual pada aplikasi yang dibuat. Sebagai contoh, ada *windows Properties* untuk *form project* sebab *form* berisi *property*, seperti warna ukuran dan lain-lain.

5. *Code*

Tidak seperti kebanyakan bahasa pemrograman yang lain, *user* tidak harus memilih banyak *code* karena bagian yang terlihat dari *visual basic* telah menyederhanakan *Code* itu. *Code* di dalam *windows code* adalah *source code*. Ketika *user* menjalankan program *visual basic* maka komputer menginterpretasikan sebagai *source code* kemudian komputer akan mengeksekusi instruksi di dalam *source code* tersebut.

3.7 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau DBMS (Database Management System) yang multithread, multi-user, dengan sekitar 6 jutainstalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (General Public License).

3.8 Analisis dan Perancangan Sistem

Analisis sistem dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya. Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi. Analisis dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi. (Kendall dan Kendall, 2003:7)

3.8.1 Data Flow Diagram (DFD)

Data Flow Diagram atau yang sering disebut *Bubble Chart* atau diagram, model proses, diagram alur kerja atau model fungsi adalah alat pembuatan model yang memungkinkan professional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alir data baik secara manual maupun terkomputerisasi. DFD merupakan alat pembuat model yang sering digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluaran. Untuk memudahkan proses pembacaan DFD, maka penggambaran DFD disusun berdasarkan tingkatan atau *level* dari atas ke bawah, yaitu :

1. *Context Diagram*

Merupakan diagram paling atas yang terdiri dari suatu proses dan menggambarkan ruang lingkup proses. Hal yang digambarkan dalam diagram konteks adalah hubungan terminator dengan sistem dan juga sistem dalam suatu proses. Sedangkan hal yang tidak digambarkan dalam *Context Diagram* adalah hubungan antar *terminator* dan *data source*

2. Diagram Zero (Level 0)

Merupakan diagram yang berbeda diantara diagram konteks dan diagram detail serta menggambarkan proses utama dari DFD. Hal yang digambarkan dalam *Diagram Zero* adalah proses utama dari sistem serta hubungan *entity*, proses, alur data dan *data source*.

3. Diagram Detai (Primitf)

Merupakan penguraian dalam proses yang ada dalam *Diagram Zero*. Diagram yang paling rendah dan tidak dapat diuraikan lagi.

Data Flow Diagram (DFD) memiliki empat komponen, yaitu :

1. Terminator

Terminator atau *External Entity* atau kesatuan luar mewakili entitas *external* yang berkomunikasi dengan sistem yang dikembangkan. Terminator merupakan kesatuan dilungkungan system yang dapat berupa orang atau system yang berada dilingkungan luar system yang memberikan inputan maupun yang menerima output dari system serta berupa bagian atau devisi diluar system yang

berkomunikasi dengan system. Terminator ini sering juga disebut entitas (external), sumber atau tujuan (source and sink).

2. Proses

Proses sering dikenal dengan *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input ke output, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu inputan atau lebih menjadi output dari sistem. Dilambangkan dengan lingkaran atau empat panjang tegak dengan sudut tumpul. Proses diberi nama untuk menerangkan proses atau kegiatan apa yang sedang atau kegiatan yang akan dilaksanakan. Dan setiap proses harus diberi penjelasan lengkap sebagai berikut :

a. Identifikasi Proses

Umumnya berupa angka yang menunjukkan nomor dari proses dan ditulis pada bagian atas symbol

b. Nama Proses

Menunjukkan apa yang sedang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya. Nama proses diletakkan dibawah identifikasi proses.

3. Data Store

Data Store digunakan sebagai sarana untuk pengumpulan data. *Data Store* disimbolkan dengan dua garis horizontal yang parallel dimana tertutup pada satu ujungnya atau dua garis horizontal. Suatu nama perlu diberikan pada Data Store menunjukkan nama dari filenya. *Data Source* biasanya berkaitan dengan penyimpanan *file* atau *database* yang dilakukan secara komputerisasi. *Data Store*

dihubungkan dengan alur data hanya pada komponen proses pengertian sebagai berikut :

- a. Alur data dari store yang berarti sebagai pengaksesan data untuk suatu proses.
- b. Alur data ke proses berarti meng-update data seperti menambah data, mengurangi data maupu mengubah data.

4. Alur Data

Alur Data dapat digambarkan dengan anak panah yang menuju ke dalam proses maupun keluar proses. Alur data digunakan untuk menerangkan perpindahan data atau informasi dari suatu bagian ke bagian lainnya.

5. Syarat-syarat sebuah PDF :

- a. Pemberian nama untuk setiap komponen DFD
- b. Pemberian nomor pada proses DFD
- c. Penggambaran DFD serapi mungkin
- d. Menghindari pembuatan DFD yang rumit
- e. Memastikan DFD dibangun secara konsisten

3.8.2 Entity Relationship Diagram (ERD)

Entity relationship diagram adalah suatu bentuk perencanaan *database* secara konsep fisik yang nantinya akan dipakai sebagai kerangka kerja dan pedoman dari struktur penyimpanan data. ERD digunakan untuk menggambarkan model hubungan data dalam sistem, dimana di dalamnya terdapat hubungan entitas beserta

atribut relasinya dan mendokumentasikan kebutuhab-kebutuhan untuk sistem pemrosesan data. ERD memiliki beberapa jenis model, yaitu :

Gambar 3.1 Jenis ERD

NO	JENIS ERD	KETERANAGAN
1.	Conceptual Data Model (CDM)	Merupakan model <i>universal</i> dan dapat menggambarkan semua struktur logic database (DBMS), dan tidak bergantung dari software atau pertimbangan struktur data storage. Sebuah CDM dapat diubah langsung menjadi PDM
2.	Physical Data Model (PDM)	Merupakan model ERD yang mengacu pada pemilihan <i>software</i> DBMS yang spesifik. Hal ini seringkali berbeda secara signifikan dikarenakan oleh struktur tipe <i>database</i> yang bervariasi, dari model schema, tipe data penyimpanan dsb

ERD memiliki 4 jenis obyek, yaitu :

1. *Entity*

Sesuatu yang ada dan terdefiniskan bias berupa nyata maupun abstrak yang dapat dibedakan satu dengan lainnya dan adanya hubungan saling ketergantungan ada 2 macam tipe *entity*, yaitu :

a. *Strong Entity*

Strong Entity merupakan tipe *entity* yang mempunyai *key attribute* untuk setiap individu yang ada didalamnya

b. *Weak Entity*

Strong Entity merupakan *entity* yang tidak memiliki *key attribute*, oleh karena itu *weak entity* harus dihubungkan dengan *strong entity* untuk menggunkan *attribute* kunci secara bersama-sama.

2. *Attribute*

Setiap *entity* memiliki beberapa *attribute*, yang merupakan ciri atau karakteristik dari *entity* tersebut. *Attribute* sering disebut juga data elemen atau data field.

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 sebagai elemen data kunci (*key*).

4. *Relationship*

Relationship menggambarkan hubungan yang terjadi antar *entity* yang mewujudkan pemetaan antar *entity*. Bentuk *relationship* yaitu :

a. *One to One Relationship*

Hubungan satu entity dengan satu entity yang lain

b. *One to Many Relationship*

Hubungan antar *entity* satu dengan *entity* yang lainnya adalah satu berbanding banyak.

