

BAB III

LANDASAN TEORI

Landasan teori digunakan untuk menyelesaikan masalah secara sistematis. Pada bab ini akan membahas landasan teori yang meliputi landasan teori mengenai hal – hal dari permasalahan yang ada dan landasan teori yang membahas tentang ilmu yang terkait dalam permasalahan tersebut.

3.1 Sumber Daya Manusia

Menurut Nawawi (2001) ada tiga pengertian Sumber daya manusia yaitu :

- Sumber daya manusia adalah manusia yang bekerja dilingkungan suatu organisasi (disebut juga personil, tenaga kerja, pekerja atau karyawan).
- Sumber daya manusia adalah potensi manusiawi sebagai penggerak organisasi dalam mewujudkan eksistensinya.
- Sumber daya manusia adalah potensi yang merupakan asset dan berfungsi sebagai modal (*non material/non finansial*) didalam organisasi bisnis, yang dapat mewujudkan menjadi potensi nyata (*real*) secara *fisik* dan non fisik dalam mewujudkan eksistensi organisasi.

3.1.1 Fungsi *Outsourcing*

- dapat menghemat biaya dan pengeluaran dalam hal pembiayaan sumber daya manusia (SDM) yang bekerja di perusahaan tersebut.
- menjembatani hubungan antara pekerja dan pengusaha.

3.1.2 Jenis *Outsourcing*

Menurut undang – undang nomer 13 (ketenagakerjaan) kata Menakertrans Muhaimin Iskandar di Kantor Menteri Koordinator Perekonomian Jakarta ada 5 pekerjaan yang diperbolehkan *outsourcing* yaitu :

1. **cleaning service**
2. **keamanan**
3. **transportasi**
4. **catering**
5. **jasa migas pertambangan**

3.2 *Outsourcing*

Outsourcing berasal dari kata *out* yang berarti keluar dan *source* yang berarti sumber. Menurut Undang – Undang nomer 13 tahun 2003 tentang ketenagakerjaan pasal 64, *outsourcing* adalah suatu perjanjian kerja yang dibuat antara pengusaha dengan tenaga kerja, dimana perusahaan tersebut dapat menyerahkan sebagian pelaksanaan pekerjaan kepada perusahaan lainnya melalui perjanjian pemborongan pekerjaan yang dibuat secara tertulis.

Dari pengertian-pengertian di atas maka dapat ditarik suatu definisi operasional mengenai *outsourcing* yaitu suatu bentuk perjanjian kerja antara perusahaan A sebagai pengguna jasa dengan perusahaan B sebagai penyedia jasa, dimana perusahaan A meminta kepada perusahaan B untuk menyediakan tenaga kerja yang diperlukan untuk bekerja di perusahaan A.

3.3 Aplikasi

Menurut Dhanta (2009:32), aplikasi (application) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas - tugas tertentu, misalnya *Microsoft Word, Microsoft Excel*. Sedangkan menurut Anisyah (2000:30), aplikasi adalah penerapan, penggunaan atau penambahan.

Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas - tugas tertentu seperti penerapan, penggunaan dan penambahan data.

3.4 Data

Menurut Dhanta (2009:136), data adalah kumpulan dari angka - angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.

Sedangkan menurut Anisyah (2000:96), data adalah keterangan yang benar dan nyata. Jadi dapat disimpulkan bahwa data adalah sekumpulan fakta yang benar dan nyata yang dapat berbentuk angka-angka maupun karakter yang belum memiliki arti.

3.5 Informasi

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian - kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan. (Jogiyanto HM, 1999: 692)

3.6 Analisis Sistem

Menurut (Jogiyanto HM 1990:129) analisa 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 – kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan dan pengembangannya.

3.7 Desain Sistem

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran yang jelas apa yang harus dikerjakan. Kemudian memikirkan bagaimana membentuk sistem tersebut. Menurut (Jogiyanto 1990:197) desain sistem dapat diartikan sebagai berikut :

- a. Tahap setelah analisis dari siklus pengembangan sistem.
- b. Pendefinisian dari kebutuhan – kebutuhan fungsional.
- c. Persiapan untuk rancang bangun implementasi.
- d. Menggambarkan bagaimana suatu sistem dibentuk.
- e. Berupa gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

- f. Menyangkut konfigurasi dari komponen – komponen perangkat lunak dan perangkat keras dari suatu sistem.

3.8 Teori Yang Terkait Dengan Penyelesaian Masalah

3.8.1 Microsoft Visual Basic. Net 2005

Visual Basic.Net 2005 merupakan bahasa yang benar – benar berorientasi objek dengan mendukung empat pilar utama dari *Object Oriented Programming* yaitu *Abstraction, Inheritance, Polymorphism* dan *Encapsulation*.

Dalam Visual Basic. Net 2005, setiap objek bisa memiliki coding yang berbeda tergantung pada even yang sudah disediakan pada setiap objeknya.

Dengan tersedianya berbagai komponen atau objek yang memiliki wadah inputan dalam suatu *form* membuat aplikasi lebih baik dan *user friendly*, sehingga aplikasi dapat dimanfaatkan semaksimal mungkin guna menghasilkan informasi yang cepat dan akurat. Selain itu, bahasa pemrograman ini memiliki *kompatibilitas* dengan berbagai macam aplikasi pendukung lainnya, baik itu yang mengarah kepada program yang berbasis *network, website* dan aplikasi lainnya.

3.8.2 Document Flow

Documennt Flow adalah bagan yang menunjukkan alur dalama program ataupun prosedur sistem secara fisik. Bagian alur digunakan terutama untuk alat bantu komunikasi dan dokumentasi. Bagian alur sistem digambar dengan menggunakan simbol – simbol antara lain sebagai berikut :

a. Document

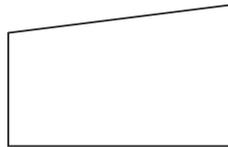
Sistem dokumen, menunjukkan dokumen input dan output baik untuk proses manual atau komputer.



b. Manual Input

Sistem ini berfungsi untuk menunjukkan pekerjaan yang masih dilakukan secara

manual.



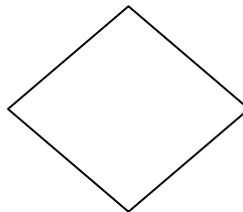
c. Proses

Digunakan untuk menunjukkan kegiatan dari operasi program komputer.



d. Decision

Digunakan untuk menggambarkan logika keputusan dengan nilai benar atau salah.



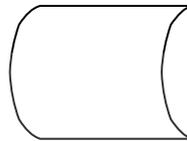
e. Flow Line

Digunakan untuk menggambarkan arah proses pengolahan data.



f. Stored Data

Digunakan untuk menyimpan data kedalam database.



3.9 Data Flow Diagram (DFD)

Pada tahap ini, penggunaan notasi dapat membantu komunikasi dengan pemakai sistem untuk memahami sistem tersebut secara logika. Diagram yang mengatasi notasi – notasi untuk menggambarkan alur dari data sistem ini dikenal dengan nama Diagram Arus Data (*Data Flow Diagram*). DFD berfungsi untuk menggambarkan proses aliran data yang terjadi di dalam sistem dari tingkat yang tertinggi sampai yang terendah, yang memungkinkan kita untuk melakukan dekomposisi, mempartisi atau membagi sistem ke dalam bagian yang lebih kecil dan sederhana.

Untuk memudahkan proses pembacaan DFD disusun berdasarkan tingkatan satu *level* dari atas kebawah, yaitu :

a. *Context Diagram*

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

b. Diagram Level 0

DFD memfokuskan pada aliran data dari dan ke dalam sistem serta memproses data tersebut. (Kendall, 2003:241)

Simbol – simbol dasar dalam DFD antara lain :

-Eksternal Entity

Suatu *eksternal entity* atau entitas merupakan orang, kelompok, departemen atau sistem lain di luar sistem yang dibuat yang bisa menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.

-Data Flow

Data Flow atau aliran data disimbolkan dengan tanda panah. *Data flow* menunjukkan arus data atau aliran yang menghubungkan dua proses atau *entitas* dengan proses.

-Proses

Suatu proses dimana beberapa tindakan atau sekelompok tindakan yang dijalankan.

-Data Source

Data Source adalah symbol yang digunakan untuk melambangkan proses penyimpanan data.

3.10 Entity Relational Diagram (ERD)

Entity Relational 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 intensitas beserta atribut relasinya dan mendokumentasikan kebutuhan – kebutuhan untuk sistem pemrosesan data. Sebuah ERD memiliki beberapa jenis model yaitu :

a. *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*. Sebuah CDM dapat diubah langsung menjadi PDM.

b. *Physical Data Model* (PDM)

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

ERD memiliki 4 jenis objek, yaitu :

- *Entity*

Sesuatu yang ada dan terdefiniskan bisa berupa nyata maupun abstrak yang dapat dibedakan satu dengan yang lainnya dan adanya hubungan saling ketergantungan.

- *Attribute*

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

- *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*).

-Relationship

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

- *One to One Relationship*

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

- *Many to Many Relationship*

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

