

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Transformator (Trafo)**

Dalam operasi penyaluran tenaga listrik transformator dapat dikatakan jantung dari sistem transmisi dan distribusi. Dalam kondisi ini suatu transformator diharapkan dapat beroperasi secara maksimal. Mengingat kerja keras dari suatu transformator seperti itu, maka pemeliharaan juga dituntut sebaik mungkin. Oleh karena itu transformator harus dipelihara dengan menggunakan sistem dan peralatan yang benar, baik dan tepat. (Za'im, 2014:2)

##### **3.1.1 Transformator Daya**

Menurut Za'im (2014:2), Transformator daya diantaranya dilengkapi dengan transformator pentanahan yang berfungsi untuk mendapatkan titik netral dari transformator daya. Peralatan ini disebut *Neutral Current Transformer (NCT)*, perlengkapan lainnya adalah pentanahan transformator yang disebut *Neutral Grounding Resistance (NGR)*.

Berdasarkan tegangan operasinya, transformator dapat dibedakan menjadi transformator 500 / 150 kV dan 150/70 kV biasa disebut Interbus Transformer (IBT). Transformator 150/20 kV dan 70/20 kV disebut juga transformator distribusi. Titik netral transformator ditanahkan sesuai dengan kebutuhan untuk sistem pengamanan/proteksi. (Za'im, 2014:2)

#### **3.2 Aplikasi**

Menurut Supriyanto (2005:117) Aplikasi adalah program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.

Aplikasi adalah penggunaan dalam suatu komputer, instruksi dan pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* (Hartono, 1999:12).

Definisi aplikasi menurut Eka Noviansyah (2008:4) adalah penggunaan dan penerapan suatu konsep yang menjadi suatu pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang di buat untuk menolong manusia dalam melakukan tugas tertentu. Aplikasi software yang direncanakan untuk suatu tugas khusus dapat dibedakan menjadi dua jenis, yaitu :

1. Aplikasi software spesialis, program dengan dokumentasi tergabung yang di jalankan untuk menjalankan tugas tertentu.
2. Aplikasi software paket, suatu program dengan dokumentasi tergabung yang di rancang

### 3.3 Website (Web)

Menurut Yuhfizar (2009:2), Pengertian *website* adalah keseluruhan halaman- halaman *web* yang terdapat dalam sebuah domain yang mengandung informasi. Domain adalah sebuah nama unik yang dimiliki oleh sebuah perusahaan atau lembaga atau organisasi yang bisa diakses melalui internet. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan. Hubungan diantara satu halaman *web* dengan yang lainnya disebut *hyperlink*. Sedangkan teks yang menghubungkan media tersebut disebut *hypertext*.

Menurut Baridwan (1999 : 127), *Website* merupakan kumpulan berbagai halaman web yang ditulis dengan bahasa HTML yang kemudian bisa di lihat menggunakan *software* yang disebut *web browser*. Halaman *web* bisa berisi *file*

seperti gambar, *video*, dan sebagainya. Agar dapat diakses, halaman *web* harus diletakkan di *server web* untuk kemudian bisa diakses melalui peranti seperti internet, jaringan, dan sebagainya.

### 3.3.1 Jenis Website Berdasarkan Sifat

- a. *Website* statis : merupakan sebuah *website* yang konten atau isinya jarang berubah. Misalnya *website* profil organisasi.
- b. *Website* dinamis : merupakan sebuah *website* yang konten atau isinya selalu berubah- ubah setiap saat. Misalnya *website* berita.

### 3.3.2 Jenis Website Berdasarkan Tujuan

- a. *Personal web* : *website* yang berisi informasi tentang seseorang
- b. *Corporate web* : *website* yang dimiliki oleh sebuah perusahaan
- c. *Portal web* : *website* yang mempunyai banyak layanan, mulai dari layanan berita, email dan jasa- jasa lainnya. (Yuhefizar, 2009: 2-3).

## 3.4 Pemrograman Web

Menurut Rizky (2008:4), Pemrograman *web* adalah pemograman aplikasi berbasis *web*, Pemrograman *web* memiliki konsep yang berbeda dengan pemrograman aplikasi berbasis *desktop*. Perbedaan-perbedaan mendasar yang membedakan *web* dengan pemrograman *desktop* yaitu :

#### 1. *Stateless*

Dengan arsitektur yang cenderung bebas, maka sebuah aplikasi web hampir tidak memiliki batasan bagi para penggunanya.

#### 2. *Multi platform*

Sebuah aplikasi *web* harus mampu dijalankan dalam sebuah *browser* di lintas *platform* atau sistem operasi.

### 3. *Navigation* sentris

Secara umum akan membebaskan pengguna untuk mengerjakan sebuah proses tanpa harus melalui prosedur semestinya.

### 4. *Based on page*

Sebuah aplikasi web juga mengandalkan nilai estetika selain nilai dari proses yang dilakukan itu sendiri.

### 5. *Just browser*

Secara umum, sebuah aplikasi *web* hanya membutuhkan sebuah *browser* tanpa perlu instalasi *runtime* ataupun aplikasi lain.

### 6. *Less secure*

Salah satu hal yang menjadi isu klasik dalam aplikasi *web* adalah masalah keamanan yang jauh lebih rentan dibandingkan dengan aplikasi *desktop*.

## 3.5 *Hypertext Preprocessor* (PHP)

Menurut Welling & Laura (2001:2), PHP adalah *server side scripting environment* yang dapat digunakan untuk membuat dan menjalankan aplikasi-aplikasi di web server menjadi lebih interaktif dan *programmable*. Dengan PHP, aplikasi-aplikasi yang ada di *web server* benar-benar akan dijalankan di *web server* tanpa mengharuskan adanya tambahan atau syarat tertentu untuk sisi *client* (*web browser*). PHP biasanya dijadikan sebagai modul dalam suatu *web server* agar bisa mengesekusi file-file PHP yang tersedia di *web server*. PHP dapat berjalan di hampir seluruh *platform*, *open source* dan ber lisensi GNU *Public License* (GPL).

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. Pada Juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman lain (Welling & Laura, 2001:2), antara lain :

1. Bahasa Pemrograman PHP adalah sebuah script yang tidak melakukan sebuah kompilasi dalam penggunaannya
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena mempunyai banyak referensi.
4. Dalam sisi pengembangan lebih mudah karena banyak merilis dan developer yang membantu dalam pengembangan.

### **3.6 Analisis Sistem dan Perancangan Sistem**

Menurut Kendall (2003:7), Analisis dan Perancangan Sistem berupaya menganalisis *input* data atau aliran data secara sistematis, memproses atau mentransformasikan data, menyimpan data, dan menghasilkan *output* informasi dalam konteks bisnis khusus. Selanjutnya, analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang bisa dicapai melalui penggunaan sistem informasi terkomputerisasi.

#### **3.6.1 Bagan Alir Dokumen**

Menurut Kedall (2003:40) Model Pertama adalah diagram aliran data *context-level* (disebut juga *environmetal model*), diagram aliran data menfokuskan

pada aliran data dari dan ke dalam sistem dan sekaligus memproses data-data tersebut. Komponen-komponen dasar dari setiap program komputer ini bisa digambarkan secara mendetail dan digunakan untuk menganalisis keakuratan dan kompetisi sistem.

### **3.6.2 Basis Data**

Menurut Kedall (2003:128), basis data adalah database management system (BDMS) yang memperbolehkan pembuatan modifikasi dan pembaharuan basis data, mendapatkan kembali data dan membangkitkan laporan. Menurut Kedall & Kendall, tujuan basis data yang efektif antara lain :

1. Memastikan bahwa data dapat dipakai di antara pemakai untuk berbagai aplikasi.
2. Memelihara data baik keakuratan maupun kekonsistennannya.
3. Memastikan bahwa semua data yang diperlukan untuk aplikasi sekarang dan yang akan datang akan disediakan dengan cepat.
4. Membolehkan basis data untuk berkembang dan kebutuhan pemakai untuk berkembang.
5. Membolehkan pemakai untuk membangun pandangan personalnya tentang data tanpa memperhatikan cara data disimpan secara fisik.

### **3.6.3 Data Flow Diagram**

Data Flow Diagram (DFD) adalah diagram yang digunakan untuk menggambarkan sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau data tersebut disimpan (Hartono, 1999:700).

Data Flow Diagram (DFD) sering digunakan untuk menggambarkan suatu sistem yang sudah ada atau baru yang telah di kembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (Jogiyanto, 1989:700).

Keuntungan menggunakan DFD adalah memudahkan pemakai yang kurang menguasai bidang atribut untuk mengerti sistem yang akan dikerjakan atau dikembangkan. Adapun atribut-atribut yang akan digunakan adalah sebagai berikut, (Hartono, 1999:700):

1. *External Entity*

Merupakan asal atau tujuan data yang memisahkan sistem dengan lingkungan luarnya. *External Entity* bisa berupa orang, organisasi atau sistem lain yang merupakan *input* atau *output* suatu sistem.

2. *Data Flow*

*Data Flow* berupa anak panah, dimana semua arus data dihasilkan dari suatu proses atau menuju suatu proses.

3. Proses

Merupakan sebuah proses yang biasanya berupa kata kerja serta memiliki *input* dan *output*, yang bisa berjumlah satu atau lebih *input data* dan menghasilkan satu atau lebih *output data*.

4. *Data Store*

*Data Store* digunakan untuk penyimpanan data atau file, arsip atau catatan manual pada saat sebelum atau sesudah proses lebih lanjut.