

BAB II

LANDASAN TEORI

A. Perancangan

Menurut Kristanto (2008 : 61), Perancangan sistem adalah suatu fase dimana diperlukan suatu keahlian perancangan untuk elemen-elemen komputer yang akan menggunakan sistem yaitu pemilihan peralatan dan program komputer untuk sistem yang baru.

Adapun tujuan yang hendak dicapai dari tahap perancangan adalah sebagai berikut:

1. Untuk memenuhi kebutuhan pemakaian sistem (*user*).
2. Untuk memberikan gambaran yang jelas dan menghasilkan rancangan bangun yang lengkap kepada pemrograman komputer dan ahli-ahli teknik lainnya yang terlibat dalam pengembangan atau pembuatan sistem.

B. Pengertian Aplikasi

Terdapat beberapa teori yang mendefinisikan aplikasi di antaranya adalah :

1. Menurut KBBI, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi merupakan suatu program komputer

yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

2. Menurut Pranama (2012), Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti *system* perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.
3. Menurut Verman dkk (2009), Aplikasi adalah perangkat intruksi khusus dalam komputer yang di rancang agar kita menyelesaikan tugas-tugas tertentu.
4. Menurut Yuhefizar (2012), Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Jadi aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan.

C. Pengertian *Web*

Menurut Simarmata (2010:47), *Web* adalah sebuah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah *server web internet* yang disajikan dalam bentuk *hipertext*. *Web* dapat diakses oleh perangkat lunak *web client* yang disebut *browser*. *Browser* membaca halaman-halaman web yang tersimpan dalam

server web melalui *protocol* yang disebut HTTP (*Hypertext Transfer Protocol*).

D. Pengertian HTML

HTML (*Hypertext Markup Language*) dapat didefinisikan sebagai seperangkat kode khusus yang ditambahkan ke dalam teks dengan tujuan melakukan pemformatan dan menghubungkan informasi-informasi di dalamnya. HTML diciptakan berdasarkan pada *Standard Generalized Markup Language (SGML)*. Untuk menuliskan dokumen HTML dapat digunakan teks editor apapun. Dokumen HTML disimpan dengan ekstensi *.html.

Semua penulisan HTML diformat menggunakan perintah-perintah yang disebut *tag*. Sebuah *tag* adalah perintah yang ada diantara tanda kurung siku buka (<) dan tanda kurung siku tutup (>). Setiap pembuatan elemen selalu diawali dengan tag pembuka dan tidak selalu diakhiri dengan tag penutup. *Tag* penutup ini berfungsi memberitahu komputer sebagai akhir dari pembuatan elemen. Apabila *tag* memiliki penutup maka penutup ditulis dengan menambahkan karakter *slash (/)* di depan nama tag penutup, misalnya <p>Hello World</p>. Apabila *Tag* tidak memiliki penutup maka penutup ditulis dengan menambahkan karakter *slash (/)* setelah nama *tag*, misalnya
. Setiap *tag* kebanyakan juga memiliki *attribute* yang digunakan untuk pendefinisian identitas *tag*. Misal <input type="name">.

Struktur dasar pembuatan dokumen HTML adalah sebagai berikut ini :

```
<html>  
  
<head>  
  
<title>Judul Halaman</title>  
  
</head>  
  
<body>  
  
Isi halaman web  
  
</body>  
  
</html>
```

1. Perintah-Perintah Dasar HTML

Beberapa tag yang banyak digunakan dalam dokumen HTML antara lain :

- a. *Tag* untuk memformat teks.
 - 2) `` teks `` digunakan untuk mengubah teks menjadi tebal (*bold*).
 - 3) `<i>`teks`</i>` digunakan untuk mengubah teks menjadi cetak miring (*italic*).
 - 4) `<u>`teks`</u>` digunakan untuk menambahkan garis bawah pada teks (*underline*).
3. Tag untuk menghasilkan sebuah *dynamic link* adalah
`link `.

Teks yang akan dibuat menjadi sebuah link diletakkan diantara kedua tag.

4. *Tag* untuk menyisipkan sebuah gambar adalah

```

```

Penulisan lokasi gambar disertai nama ekstensi gambar tersebut.

Ukuran panjang gambar diletakkan di *attribute width* sedangkan ukuran tinggi diletakkan di *attribute height*.

5. *Tag* untuk menampilkan sebuah daftar (*list*) adalah

```
<ol>  
  
<li>daftar 1</li>  
  
<li>daftar 2</li>  
  
<li>daftar 3</li>  
  
</ol>
```

Tag diatas akan menghasilkan daftar bernomor. Untuk menghasilkan daftar dengan butir caranya dengan mengganti *tag ol* (*ordered list*) dengan *tag ul* (*unordered list*).

6. *Tag* untuk membuat tabel adalah

```
<table>  
  
<tr>  
  
<td>kolom 1 baris 1</td>  
  
<td>kolom 2 baris 1</td>  
  
</tr>  
  
</table>
```

Tag diatas akan menghasilkan tabel 1 baris dengan 2 kolom

7. *Tag* untuk membuat form dengan input berupa teks adalah

```
<input name="#" type="text" value="#"/>
```

Attribute value berguna untuk menampilkan teks awal dalam form.

Attribute name berfungsi untuk sebagai *variable* untuk menampung teks pada *attribute value*

8. *Tag* untuk membuat *form* dengan *input* berupa pilihan

```
<select name="#">
  <option value="1">pilihan 1</option>
  <option value="2">pilihan 2</option>
</select>
```

Attribute name berfungsi sebagai *variable* untuk menampung opsi yang dipilih.

Attribute value berisi nilai yang mewakili teks pilihan.

9. *Tag* untuk membuat *form input* berupa teks yang panjang adalah

```
<textarea name="#" rows="#" cols="#">
  teks awal
</textarea>
```

Attribute name berfungsi sebagai *variable* untuk menampung teks yang di *input*.

Attribute rows berfungsi untuk menentukan maksimal baris yang akan ditampilkan saat pengisian.

Attribute cols berfungsi untuk menentukan maksimal panjang sebuah baris.

10. *Tag* untuk membuat *input* berupa kotak *check list*

`<input type="checkbox" value="1">` nama pilihan 1

`<input type="checkbox" value="2">` nama pilihan 2

Attribute value bersisi teks untuk mewakili nama pilihan.

11. Tag untuk membuat tombol adalah

`<input type="button" value="#">`

Attribute value berfungsi untuk menampilkan teks pada tombol

E. Pengertian PHP

Menurut Agus Saputra (2011, p.1), PHP atau yang memiliki kepanjangan *PHP Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, *web* akan sangat mudah di-*maintenance*.

PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam menjalankan PHP, wajib adanya *web server*. PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi windows maupun linux. PHP juga dibangun sebagai modul pada *web server apache* dan sebagai *binary* yang dapat berjalan sebagai *CGI (Common Gateway Interface)*.

Contoh dasar penulisan dokumen PHP.

```
<?php
    echo "Hello World";
?>
```

1. Beberapa kelebihan PHP sebagai Bahasa pemrograman web
 - a. PHP cepat kerjanya karena PHP adalah bahasa pemrograman yang berbasis bahasa C. Selain cepat PHP juga mudah digunakan.
 - b. PHP dapat berfungsi pada semua sistem operasi.
 - c. PHP dapat mengakses berbagai program (misalnya *LDAP*, *IMAP*, *mail server*, *Oracle*, *Informix*, *DB2*, dll). Hal ini karena PHP dilengkapi dengan berbagai fungsi yang dapat mengakses berbagai jenis program.
 - d. PHP terus berkembang dan disempurnakan. Karena sifatnya yang *open source*, maka banyak pengembang perangkat lunak yang ikut berpartisipasi mengembangkan dan menyempurnakan PHP.
 - e. Kemudahan mencari informasi mengenai PHP sebab banyak sekali pengembang dan pengguna PHP yang dapat menyediakan informasi yang diperlukan mengenai PHP (baik mengenai fungsi-fungsinya, maupun *error* yang terjadi pada PHP).

- f. PHP merupakan perangkat lunak yang gratis. PHP memiliki kelebihan sebagai perangkat lunak open source, yaitu tidak dipungut biaya saat menggunakannya.

Karena sifatnya yang fleksibel, perintah-perintah PHP dapat dituliskan atau disisipkan dalam sebuah dokumen HTML. Tidak seperti halaman web HTML biasa, baris-baris perintah PHP tidak dikirimkan kepada *client* oleh *server*, tetapi hanya dianalisa oleh modul PHP yang ada di *web server*. Tidak seperti kode-kode HTML, perintah PHP diinterpretasikan untuk kemudian dieksekusi. Setiap baris kode dalam PHP disebut sebagai *statement*. Semua *statement* dan perintah dalam PHP diinterpretasi dan dieksekusi secara berurutan (sekuensial).

2. Ada beberapa aturan dasar dalam penulisan perintah (*statement*) PHP, di antaranya adalah :
 - a. Setiap halaman web yang mengandung script PHP harus disimpan dengan ekstensi *.php.
 - b. Perintah (*statement*) PHP dituliskan di antara tanda <?php dan tanda ?> . Tanda awal dan akhir perintah PHP tersebut akan memberitahu interpreter bahwa kode-kode yang ada diantara kedua tanda tersebut adalah kode PHP dan harus dieksekusi.
 - c. Setiap *statement* harus diakhiri dengan tanda titik koma.
 - d. Semua variabel dalam PHP diawali dengan tanda \$, tidak peduli apakah variabel tersebut berupa tipe *string*, *integer*, *float*, atau *array*.

- e. Untuk mengisi sebuah variabel dengan data bertipe string kita harus menuliskannya dalam tanda kutip buka dan tanda kutip tutup, misalnya
`$nama = "Tedy";`
- f. Komentar (keterangan yang tidak dimunculkan pada halaman web) dalam PHP dituliskan di antara tanda `/*` dan `*/`. Untuk komentar yang panjangnya hanya 1 baris dapat juga diawali dengan tanda `//`.
- g. Untuk menampilkan isi dari variabel atau menampilkan kalimat dalam PHP digunakan perintah `echo "isi kalimat";` atau `echo "$nama_variabel";`

F. Pengertian *Javascript*

Menurut Giffin (2012), *Javascript* merupakan sebuah bahasa pemrograman populer yang di dukung oleh semua *web browser* dan *web tools* lainnya yang memungkinkan fungsi-fungsi interaktif yang bisa ditambahkan ke dalam halaman web yang statis. Dengan *javascript* dapat membuat interaksi pengguna dan aplikasi lebih interaktif. *Javascript* berkembang dari bahasa *netscape live script*. *Javascript* tertanam kedalam halaman web karenanya hanya bisa di eksekusi dengan mengambil sebuah halaman dari semua situs *web*.

1. Fungsi *Javascript*

Secara fungsional, *JavaScript* digunakan untuk menyediakan akses *script* pada objek yang dibenamkan (*embedded*). Contoh

sederhana dari penggunaan *javascript* adalah membuka halaman *pop-up*, fungsi validasi pada form sebelum data dikirimkan ke *server*, merubah image kursor ketika melewati objek tertentu, dan lain lain.

2. Kelebihan *Javascript*

- a. *Javascript* bekerja pada sisi *browser* sehingga *Javascript* dapat merespon perintah pengguna dengan cepat dan membuat halaman *web* menjadi lebih responsif.
- b. *Javascript* dapat melakukan apa yang tidak bisa dilakukan oleh HTML, PHP, dan CSS seperti menangani hal-hal yang membutuhkan respon cepat terhadap aksi dari user.
- c. Tidak membutuhkan *resource memory* besar.
- d. Dinamis.
- e. *Multi platform*, bisa dijalankan di banyak system operasi.

3. Kekurangan *Javascript*

- a. *Script* yang digunakan pada javascript tidak terenkripsi. Ketika javascript diintegrasikan kedalam *web*, maka *script javascript* dapat dilihat layaknya kode HTML.
- b. Bukan untuk pengembangan aplikasi *stand alone*
Javascript bergantung pada aplikasi lain untuk menjalankannya.

G. Basis Data

Basis data terdiri atas dua kata, yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, *symbol*, teks, gambar, bunyi atau kombinasinya. Basis data memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data.

1. Definisi Basis Data

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/Tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik. Fathansyah(2002 : 2)

2. Tujuan Basis Data

Tujuan utama dalam pengelolaan data dalam sebuah basis data adalah agar dapat memperoleh atau menemukan kembali data yang dicari dengan mudah dan cepat disamping itu pemanfaatan basis data untuk pengelolaan data, memiliki sejumlah tujuan antara lain :

- a. Kecepatan dan kemudahan Pemanfaatan basis data memungkinkan dapat untuk menyimpan data atau melakukan perubahan terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.
- b. Efisiensi ruang penyimpanan karena keterkaitan yang erat antar kelompok data dalam sebuah basis data, maka pengulangan data pasti akan selalu ada. Banyaknya pengulangan ini tentu akan memperbesar ruang penyimpanan yang harus disediakan. Dengan basis data, efisiensi ruang penyimpanan dapat dilakukan, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi antar kelompok data yang saling berhubungan.
- c. Keakuratan pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.
- d. Ketersediaan Pertumbuhan data baik dari sisi jumlah maupun jenisnya, akan semakin membutuhkan ruang penyimpanan yang besar. Padahal

tidak semua data itu selalu digunakan atau dibutuhkan. Oleh karena itu dapat dilakukan pengelompokan data utama atau data transaksi.

H. Pengertian *QR Code*

Menurut Soon (2008) *QR Code* adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso *Wave*, yaitu sebuah divisi *Denso Corporation* yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh alat pemindai. QR merupakan singkatan dari *quick response* atau respon cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan *barcode*, yang hanya menyimpan informasi secara horizontal, *QR Code* mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis *QR Code* dapat menampung informasi yang lebih banyak daripada *barcode*.