

BAB III

LANDASAN TEORI

3.1. Sekolah

Secara umum sekolah merupakan sebuah lembaga pendidikan yang bersifat formal, nonformal dan informal yang didirikan oleh negara ataupun swasta yang dirancang untuk mengajari, mengelola dan mendidik peserta didik melalui bimbingan yang diberikan oleh tenaga pendidik. Untuk menjadi sebuah sekolah, ada beberapa sarana dan prasarana yang harus dipenuhi, seperti ruang belajar, perpustakaan, kantor dan lain sebagainya.

Kamus Besar Bahasa Indonesia mendefinisikan sekolah sebagai sebuah lembaga atau bangunan yang digunakan untuk kegiatan belajar mengajar serta menjadi tempat memberi dan menerima pelajaran sesuai dengan tingkatannya (sekolah dasar, sekolah lanjutan, dan sekolah tinggi).

3.2. Website

Website disusun dari halaman-halaman yang menggunakan teknologi *web* dan saling berkaitan satu sama lain. Suatu standar teknologi *web* saat ini sudah tersusun, meskipun penerapannya belum didukung oleh seluruh pengembang *web*. Standar ini disusun oleh suatu badan yaitu *World Wide Web Consortium (W3C)*. Standar ini dibutuhkan karena semakin banyaknya variasi dalam teknologi *web* sehingga terkadang satu sama lain tidak kompatibel.

Halaman *website* dapat digolongkan menjadi *website* statis dan *website* dinamis. Data dan informasi yang ada pada *website* statis tidak berubah-ubah. Dokumen *website* yang dikirim kepada *client* akan sama isinya dengan yang ada di *web server*. Sedangkan *website* dinamis, memiliki data dan informasi yang berbeda-beda tergantung input apa yang disampaikan *client*. Dokumen yang sampai di *client* akan berbeda dengan dokumen yang ada di *web server*. (B. Tujni dan Megawaty, 2017).

3.3. *Hypertext Markup Language (HTML)*

HTML adalah singkatan dari *Hypertext Markup Language* yaitu kumpulan kode bahasa pemrograman yang disusun untuk membuat suatu halaman *website* (A. Firman, H. F. Wowor dan X. Najoan, 2016).

3.4. *Hypertext Preprocessor (PHP)*

PHP atau yang memiliki kepanjangan *PHP Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. Di dalam PHP itu sendiri terdapat *script-script* HTML. HTML berfungsi sebagai pembangun dari kerangka layout *website* sedangkan PHP berfungsi sebagai prosesnya atau pada tahap pembuatannya pada sisi *server* sehingga dengan adanya PHP tersebut, sintaks *website* tidak terlihat pada sisi *client* (Supriyatna dan K. Nisa, 2015).

3.5. *XAMPP*

XAMPP adalah perangkat lunak yang menggabungkan empat aplikasi (Apache, *MySQL*, PHP dan Perl) dalam satu paket sehingga memudahkan pengguna dalam instalasi dan konfigurasi (B. Tujni dan Megawaty, 2017).

3.6. *Web Browser*

Web Browser merupakan perangkat lunak yang berfungsi sebagai media penjelajah suatu informasi dari situs tertentu menggunakan akses internet (A. Firman, H. F. Wowor dan X. Najoan, 2016).

3.7. *Bootstrap*

Bootstrap merupakan kerangka kerja atau template yang memudahkan dalam membangun tampilan web baik secara tata letak, huruf, *background*, elemen, komponen tampilan dan lain-lain (Siswanto dan Suwarni, 2011).

3.8. MySQL

MySQL adalah singkatan dari *My Structured Query Language*, merupakan *Relation Database Management System (RDBMS)* yang berfungsi sebagai tempat penyimpanan, modifikasi dan mengakses data atau informasi yang telah tersimpan di dalamnya (A. Oktaviani, D. Sarkawi dan A. Priadi, 2018).

3.9. PHP MyAdmin

PHP *MyAdmin* merupakan aplikasi *open source* yang berfungsi untuk mengelola data pada *database* yang meliputi membuat *database*, membuat tabel dan proses DML (*Data Manipulation Language*) dengan berbasis GUI (*Graphical User Interface*) (A. Oktaviani, D. Sarkawi dan A. Priadi, 2018).

3.10. UML (Unified Modeling Language)

Unified Modeling Language (UML) merupakan permodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Ardhian Agung Yulianto, 2009).

BAB IV

HASIL DAN PEMBAHASAN

Berikut adalah pembahasan tahapan-tahapan yang digunakan untuk pengembangan *website* sekolah di Sekolah Menengah Atas Negeri 2 Cibinong.

4.1. Analisa Kebutuhan

4.1.1. Analisis Pengguna

Berdasarkan hasil wawancara dan observasi dengan pihak kerja praktik yang dilakukan pada tanggal 3 Desember 2019, pengguna sistem atau *website* ini terbagi menjadi tiga bagian atau hak akses yang berguna untuk mengelola *website* tersebut dan satu bagian visitor atau pengunjung biasa, untuk lebih jelasnya dapat dilihat pada tabel 4.1.

Tabel 4.1. Analisis Pengguna Sistem

No	Hak Akses Pengguna	Keterangan
1	Administrator	Mengelola semua data yang ada pada <i>website</i> tersebut
2	Admin Publish	Mengelola data berita atau artikel yang berkaitan dengan akademik maupun non akademik
3	Admin Siswa	Mengelola data berita atau artikel yang berkaitan dengan kemahasiswaan
4	Visitor	Menerima informasi melalui <i>website</i> tersebut (Pengunjung biasa)

4.2. Implementasi

Pada tahap implementasi ini terdapat modul-modul yang berfungsi untuk memenuhi kebutuhan *website*, modul-modul tersebut dibagi menjadi 2 bagian yaitu modul master dan modul artikel atau penggabungan beberapa modul menjadi satu.

4.2.1. Modul Master

4.2.1.1. Modul Mata Pelajaran

Berikut tahapan pembuatan modul mata pelajaran :

1. Buatlah sebuah file *controller* dengan nama *kat_mapel.php* dan simpan pada folder *application >> controllers*
2. Buat *class* dengan nama *Kat_mapel* dan masukkan *script* di bawah ini :

```
<?php
    defined('BASEPATH') OR exit('No direct script access allowed');

    class Kat_mapel extends CI_Controller {

        function __construct(){
            parent::__construct();
            //validasi jika user belum login
            if($this->session->userdata('masuk') != TRUE){
                echo "<script>alert('Waktu Anda Telah Habis,Silakan Login Terlebih Dahulu');
                    document.location='http://localhost/smavo/login'</script>";
            }
        }
    }
```

3. Buat *function* dengan nama *index* dan masukkan *script* di bawah ini :

```
public function index()
{
    $konten['css'] = '';
    $konten['js'] = '
        <script src="'.base_url().\'assets/js/jquery.dataTables.min.js\'"></script>
        <script src="'.base_url().\'assets/js/jquery.dataTables.bootstrap.min.js\'"></script>
        <script src="'.base_url().\'assets/js/dataTables.buttons.min.js\'"></script>
        <script src="'.base_url().\'assets/js/buttons.flash.min.js\'"></script>
        <script src="'.base_url().\'assets/js/buttons.html5.min.js\'"></script>
        <script src="'.base_url().\'assets/js/buttons.print.min.js\'"></script>
```

```

        <script src="'.base_url().'assets/js/butt
ons.colVis.min.js"></script>
        <script src="'.base_url().'assets/js/data
Tables.select.min.js"></script>
        <script src="'.base_url().'assets/plugins
/sweetalert2/sweetalert2.min.js"></script>
        <script src="'.base_url().'assets/js/tabl
e.js"></script>';
        $konten['konten']          = 'kat_mapel/view
_kat_mapel';
        $konten['judul']           = 'Data Master';
        $konten['sub_judul']       = 'Data Kategori
Mapel';
        $konten['data']            = $this->db-
>get('tb_mapel');
        $this->load->view('v_dashboard',$konten);
    }

```

4. Buatlah sebuah file *model* dengan nama `model_kat_mapel.php` dan simpan pada folder *application >> models*
5. Masukkan *script* pada `model_kat_mapel`, di dalam model tersebut terdapat *function* `getdata`, `getinsert`, `getupdate`, `getdelete` dan kode.

```

<?php
    defined('BASEPATH') OR exit('No direct script acc
ess allowed');

    class Model_kat_mapel extends CI_model {

        public function getdata($key) // memanggil da
ta mapel
        {

            $this->db->where('id_mapel',$key);
            $hasil = $this->db->get('tb_mapel');
            return $hasil;
        }

        public function getupdate($key,$data) // meng
update (mengubah) data mapel
        {

            $this->db->where('id_mapel',$key);
            $this->db->update('tb_mapel',$data);
        }
    }

```

```

    }

    public function getinsert($data) // memasukka
n data mapel
    {
        $this->db->insert('tb_mapel',$data);

    }

    public function getdelete($key) // menghapus
data mapel
    {
        $this->db->where('id_mapel',$key);
        $this->db->delete('tb_mapel');
    }

    public function kode(){
        $this->db-
>select('RIGHT(tb_mapel.id_mapel,2) as id_mapel', FAL
SE);
        $this->db->order_by('id_mapel','DESC');
        $this->db->limit(1);
        $query = $this->db-
>get('tb_mapel'); //cek dulu apakah ada sudah ada ko
de di tabel.
        if($query->num_rows() <> 0){
            //cek kode jika telah tersedia
            $data = $query->row();
            $kode = intval($data->id_mapel) + 1;
        }
        else{
            $kode = 1; //cek jika kode belum terd
apat pada table
        }
        // $tgl=date('dmY');
        $batas = str_pad($kode, 3, "0", STR_PAD
_LEFT);
        $kodetampil = "MP".$batas; //format ko
de
        return $kodetampil;
    }
}

```

6. Setelah membuat file *controller* dan *models* di atas, selanjutnya akan dilanjutkan untuk pembuatan fungsi *insert*, *update*, *delete* dan *view*.

- a. *Insert* Mata Pelajaran
- b. *Edit* Mata Pelajaran
- c. *Delete* Mata Pelajaran
- d. *View* Mata Pelajaran

4.2.1.2. Modul Guru

Berikut tahapan pembuatan modul guru :

- a. *Insert* Guru
- b. *Edit* Guru
- c. *Delete* Guru
- d. *View* Guru

4.2.1.3. Modul Jabatan

Berikut tahapan pembuatan modul jabatan :

- a. *Insert* Jabatan
- b. *Edit* Jabatan
- c. *Delete* Jabatan
- d. *View* Jabatan

4.2.1.4. Modul Organisasi

Berikut tahapan pembuatan modul organisasi :

- a. *Insert* Organisasi
- b. *Edit* Organisasi
- c. *Delete* Organisasi
- d. *View* Organisasi

4.2.1.5. Modul Statistik Pengguna

Berikut tahapan pembuatan statistik pengguna :

- a. *Insert* Statistik Pengguna
- b. *Edit* Statistik Pengguna
- c. *Delete* Statistik Pengguna
- d. *View* Statistik Pengguna

4.2.1.6.Modul Administrator

Berikut tahapan pembuatan administrator :

- a. *Insert* Administrator
- b. *Edit* Administrator
- c. *Delete* Administrator
- d. *View* Administrator

4.2.1.7.Modul Kategori Berita

Berikut tahapan pembuatan kategori berita :

- a. *Insert* Kategori Berita
- b. *Edit* Kategori Berita
- c. *Delete* Kategori Berita
- d. *View* Kategori Berita

4.2.2. Modul Artikel

4.2.2.1.Modul Profil

4.2.2.2.Modul Kesiswaan

4.2.2.3.Modul Berita