BAB 4

IMPLEMENTASI DAN PEMBAHASAN SISTEM

4.1 Implementasi Sistem

Berdasarkan rancangan sistem pada BAB III dengan berbagai fungsional dan ruang lingkup sistem yang telah di jelaskan pada bab-bab sebelumnya, maka pada BAB IV merupakan tahapan implementasi atau pengembangan rancangan dimana sistem siap digunakan dan di lakukan pengujian dalam keadaan yang sebenarnya dengan tahapan implementasi sebagai berikut :

4.1.1 Konfigurasi Database

Sebelumnya telah dilakukan rancangan basis data yang digunakan untuk menampung, mengelompokan, mengidentifikasi dan menampilkan data. sebelum sebelum melakukan pengembangan sistem perlu di lakukan dua konfigurasi dara pada framework CodeIgniter. yang pertama adalah melakukan ubahan pada base url pada gambar 4.1 , yang befungsi untuk mengakses sumber daya pada dokumen root/akar.

```
1. $config['base_url']     =     ((isset($_SERVER['HTTPS'])     &&
   $_SERVER['HTTPS'] == "on") ? "https" : "http");
2. $config['base_url'] .= "://" . $_SERVER['HTTP_HOST'];
3. $config['base_url']                                          .=
   str_replace(basename($_SERVER['SCRIPT_NAME']),              "",
   $_SERVER['SCRIPT_NAME']);
```

Gambar 4.1 Konfigurasi Base Url

SAMPLE DRAFT

membuat kerangka dasar program yaitu model pada pola desain MVC (Model, View Dan Controller). perlu di lakukan konfigurasi yaitu menguhubngkan CodeIgniter dengan basis data. Konfigurasi dapat ditemukan pada kode sumber CodeIgniter pada struktur folder application/config/databases.php , ubahan pada konfigurasi dapat dilihat pada Gambar 4.2

```
4.  $db['default'] = array(
5.          'dsn'  => '',
6.          'hostname' => 'localhost',
7.          'username' => 'root',//
8.          'password' => '',//
9.          'database' => 'zada_k13',//akademik
10.         'dbdriver' => 'mysqli',
11.         'dbprefix' => '',
12.         'pconnect' => FALSE,
13.         'db_debug' => (ENVIRONMENT !== 'production'),
14.         'cache_on' => FALSE,
15.         'cachedir' => '',
16.         'char_set' => 'utf8',
17.         'dbcollat' => 'utf8_general_ci',
18.         'swap_pre' => '',
19.         'encrypt' => FALSE,
20.         'compress' => FALSE,
21.         'stricton' => FALSE,
22.         'failover' => array(),
23.         'save_queries' => TRUE
24.);
```

Gambar 4.2 Konfigurasi Database

SAMPLE DRAFT

4.1.2 Login User

Gambar 4.3 Halaman Login

User akses pada sistem di bagi menjadi beberapa level akses, antara lain adalah Administrator, Guru dan Orangtua/Wali. setiap akun pengguna tersebut di indenfikasikan dengan menggunakan username dan password yang di simpan dalam databases. dan juga memiliki hak akses pada fitur-fitur sesuai dengan fungsionalnya. Adminsitrator memiliki hak akses tertinggi dalam sistem , Guru/Wali kelas Memiliki hak akses untuk melakukan input nilai sesuai dengan mata pelajaran yang di ampu sedangan Orangtua/Wali memiliki hak akses untuk melihat capaian belajar siswa. dalam implementasinya di bagai menjadi beberapa bagian antar alain adalah view pada Gambar 4.4 yang menampilkan formulir login kepada pengguna.

```
1.  <div class="col-lg-4 mx-auto">
2.  <div class="auto-form-wrapper ">
3.  <div class="text-center">
4.  <h2>E-RAPORT</h2>
5.  <h3>SD SANTA MARIA TIMIKA</h3>
6.  </div>
7.  <br>
8.  <?php echo $this->session->flashdata('notif'); ?>
9.  <form                action="<?php                echo
    base_url('login/aksi_login');?>" method="POST">
```

SAMPLE DRAFT

```
10. <div class="form-group">
11. <label class="label">Username</label>
12. <div class="input-group">
13. <input          type="text"          class="form-control"
    placeholder="Username" name="username"
14. id="username">
15. <div class="input-group-append">
16. <span class="input-group-text">
17. <i class="mdi mdi-check-circle-outline"></i>
18. </span>
19. </div>
20. </div>
21. </div>
22. <div class="form-group">
23. <label class="label">Password</label>
24. <div class="input-group">
25. <input        type="password"        class="form-control"
    placeholder="*********" name="password"
26. id="password">
27. <div class="input-group-append">
28. <span class="input-group-text">
29. <i class="mdi mdi-check-circle-outline"></i>
30. </span>
31. </div>
32. </div>
33. </div>
34. <div class="form-group">
35. <select name="level" class="form-control">
36. <option value="guru"> GURU </option>
37. <option value="admin"> ADMIN </option>
38. <option value="orangtua"> ORANGTUA/WALI </option>
39. </select>
40. </div>
41. <div class="form-group">
42. <button type="submit" class="btn btn-primary submit-btn
    btn-block">Login</button>
43. </div>
44. </form>
45. </div>
```

Gambar 4.4 Fungsi Login User

Pada gambar diatas menggunakan formulir login pada HTML5 yaitu memiliki struktur *form* dengan sebuah aksi menuju *controller* atau *route* *login* untuk menjalankan fungsi *aksi_login* menggunakan metode *post*. Metode *post* sendiri akan mengirimkan data pada aksi selanjutnya tanpa menampilkan pada *url* . untuk menampung input di dalam form di perulkan sebuah input, dalam login input yang di perulkan adalah username dan password dapat di lihat pada Gambar 4.4 pada baris 34 dan 46.

SAMPLE DRAFT

Untuk mendefenisikan level akun tersebut, di gunakan dropdown list sehingga user ketika mengakses dalam memilih baik login sebagai admin,guru atau orangtua/wali.

```
1. function aksi_login(){
2. $level = $this->input->post('level');
3. if($level=='admin'){
4. $username = $this->input->post('username');
5. $password = $this->input->post('password');
6. $where = array(
7. 'username' => $username,
8. 'password' => md5($password));
9. $cek = $this->m_login->cek_login('user_admin',$where)-
   >num_rows();
10. $row = $this->m_login->cek_login('user_admin',$where)->row();
11. if($cek > 0){
12. $data_session = array(
13. 'id' => $row->id_admin,
14. 'nama' => $username,
15. 'status' => "admin",
16. 'waktu' => date('d-m-Y H:i:s'));
17. $this->session->set_userdata($data_session);
18. redirect('admin');
19. }else{
20. $this->session->set_flashdata('notif', '<div class="alert alert-danger
    alert-dismissable"><button type="button" class="close" data-
    dismiss="alert" aria-hidden="true">&times;</button>Username
    atau Password tidak sah.</div>');
21. redirect('login');}
22. }
23. if($level=='guru'){
24. $username = $this->input->post('username');
25. $password = $this->input->post('password');
26. $where = array(
27. 'username' => $username,
28. 'password' => md5($password)
29. );
30.
31. $cek = $this->m_login->cek_login('data_guru',$where)-
    >num_rows();
32. $row = $this->m_login->cek_login('data_guru',$where)->row();
33. if($cek > 0){$data_session = array(
34. 'id' => $row->id_guru,
35. 'nama' => $row->nama_guru,
```

SAMPLE DRAFT

```
36. 'status' => "guru",
37. 'waktu' => date('d-m-Y H:i:s')
38. );
39. $this->session->set_userdata($data_session);
40. redirect('guru?m=dashboard');
41. }else{
42. $this->session->set_flashdata('notif', '<div class="alert alert-danger
    alert-dismissable"><button type="button" class="close" data-
    dismiss="alert" aria-hidden="true">&times;</button>Username
    atau Password tidak sah. </div>');
43. redirect('login');}}
44. if($level=="orangtua"){
45. $username = $this->input->post('username');
46. $password = $this->input->post('password');
47. $where = array(
48. 'username' => $username,
49. 'password' => md5($password)
50. );
51. $cek = $this->m_login->cek_login('data_siswa',$where)-
    >num_rows();
52. $row = $this->m_login->cek_login('data_siswa',$where)->row();
53. if($cek > 0){
54. $data_session = array(
55. 'id' => $row->id_siswa,
56. 'nama' => $row->nama_siswa,
57. 'status' => "orang_tua",
58. 'waktu' => date('d-m-Y H:i:s')
59. );
60. $this->session->set_userdata($data_session);
61. redirect('ortu');
62. }else{this->session->set_flashdata('notif', '<div class="alert alert-
    danger alert-dismissable"><button type="button" class="close"
    data-dismiss="alert" aria-
    hidden="true">&times;</button>Username atau Password tidak
    sah. </div>');
63. redirect('login');}
```

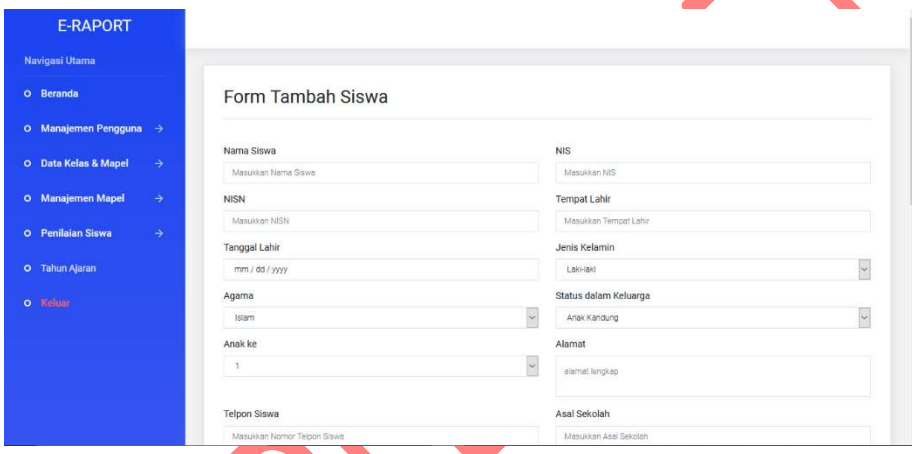
Gambar 4.5 Fungsi Aksi Lgogin

pada Gambar 4.5 merupakan fungsi login yang digunakan untuk melakukan pengkondisian user login baik sebagai admin,guru dan orangtua/wali , parameter pada dropdownliat yang di lempar melalui form ke fungsi tersebut dengan

SAMPLE DRAFT

menggunakan if dan else statment maka dapat di denfiniskan apakah user dan password yang di inputkan milik admin,guru atau orangtuawali. selanjutnya fungsi akan mengechek apakah usernaem dan password tersebut tersedia dalam database jika bernilai true , amakan akan di arahkan ke dashboard jika bernilai false akan di kembalikan ke halaman login.

4.1.3 Fungsional Manajemen Siswa



Gambar 4.6 Halaman Manajemen Siswa

Pada fungsional manajemen siswa, dapat di akses oleh administrator sebagai pemilik level akun tertinggi dalam sistem. fungsi-fungsi yang ada antara lain adalah admin dapat melakukan create,read,update dan delete pada data siswa. pada gambar 4.7 merupakan fungsi untuk melakukan simpan data siswa yang di input melalui formulir tambah data siswa.

```
1. public function tambah_siswa()
2.     {
3.         $data['kelas'] = $this->m_admin->get_kelas();
4.         $data['tahunajaran'] = $this->m_admin->m_get_aktif_tahun_ajaran();
5.         $data['content'] = "admin/siswa/v_tambah_siswa";
6.         $this->load->view('admin/index', $data);
```

SAMPLE

```
7.    }
```

Gambar 4.7 Fungsi tambah siswa

kode sumber di atas akan menampilkan formulir pendambaha data siswa baru , formulir akan menampung kebutuhan data yang di inputkan oleh user, untuk di kirim kedalam fungsi selanjutnya pada gambar 4.X yaitu mengirimkan data inputkan kedalam database.

```
1.  $post = $this->input->post();
2.  $data = array(
3.  'nama_siswa' => trim(ucwords(htmlentities($post['nama_siswa']))),
4.  'nis' => trim(htmlentities($post['nis'])),
5.  'nisn' => trim(htmlentities($post['nisn'])),
6.  'tempat_lahir' => trim(htmlentities($post['tempat_lahir'])),
7.  'tgl_lahir' => trim(htmlentities($post['tgl_lahir'])),
8.  'kelamin' => trim(htmlentities($post['kelamin'])),
9.  'agama' => trim(htmlentities($post['agama'])),
10. 'status_dlm_kel' => trim(htmlentities($post['status_dlm_kel'])),
11. 'anakke' => trim(htmlentities($post['anakke'])),
12. 'alamat_siswa' => trim(ucwords(htmlentities($post['alamat_siswa']))),
13. 'telpon_siswa' => trim(htmlentities($post['telpon_siswa'])),
14. 'asal_sekolah' => trim(htmlentities($post['asal_sekolah'])),
15. 'kelas' => trim(htmlentities($post['kelas'])),
16. 'diterima_tanggal' => trim(htmlentities($post['diterima_tgl'])),
17. 'nama_ayah' => trim(htmlentities($post['nama_ayah'])),
18. 'nama_ibu' => trim(htmlentities($post['nama_ibu'])),
19. 'alamat_ortu' => trim(htmlentities($post['alamat_ortu'])),
20. 'telpon_ortu' => trim(htmlentities($post['telpon_ortu'])),
21. 'kerja_ayah' => trim(htmlentities($post['kerja_ayah'])),
22. 'kerja_ibu' => trim(htmlentities($post['kerja_ibu'])),
23. 'nama_wali' => trim(htmlentities($post['nama_wali'])),
24. 'alamat_wali' => trim(htmlentities($post['alamat_wali'])),
25. 'telpon_wali' => trim(htmlentities($post['telpon_wali'])),
26. 'kerja_wali' => trim(htmlentities($post['kerja_wali'])),
27. 'tahun_ajaran' => trim(htmlentities($post['tahun_ajaran'])),
28. 'username' => trim(htmlentities($post['username'])),
29. 'password' => trim(md5(htmlentities($post['password']))),
30. 'status' => 1,
31. );
32. $this->m_admin->insert_dataTo($data, 'data_siswa');
33. $this->setmessage('Data Siswa berhasil ditambahkan :)', 'success');
```

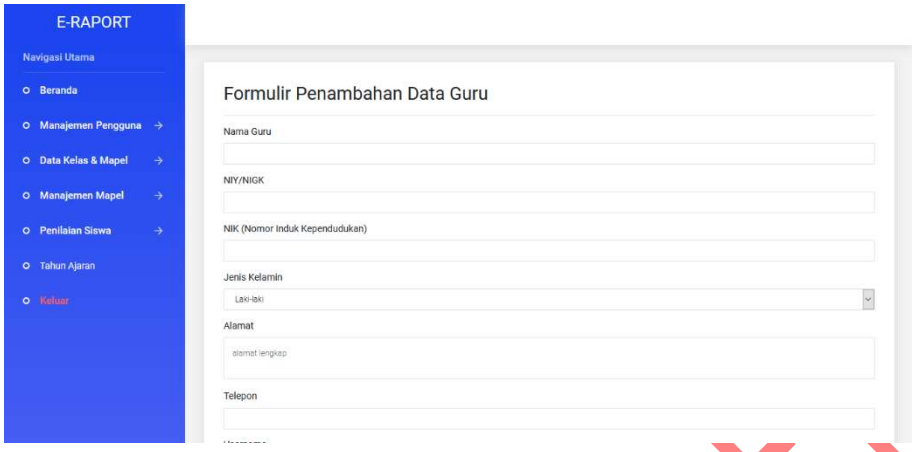
SAMPLE DRAFT

```
34. redirect('admin/data_siswa');
```

fungsi pada baris kode di atas adalah untuk menampung inputan dari form sebelumnya melalu parameter yang di simpan dalam sebuah array data, setelah itu dalam fungsi akan memanggil model untuk melakukan penyimpanan ke data, jika berhasil maka akan di arahkan ke halaman data siswa.

SAMPLE DRAFT

4.1.4 Fungsional Manajemen Guru



Gambar 4.8 Halaman Manajemen guru

Pada fungsional manajemen guru, dapat di akses oleh administrator sebagai pemilik level akun tertinggi dalam sistem. Fungsi-fungsi yang ada antara lain adalah admin dapat melakukan create,read,update dan delete pada data guru. Gambar 4.7 merupakan fungsi untuk menampilkan formulir halaman penambahan data guru.

```
1. public function form_tambah_guru()
2. {
3. $data['content'] = "admin/v_data_guru_tambah";
4. $this->load->view('admin/index', $data);
5. }
```

Gambar 4.9 Fungsi Tambah Guru

Kemudian pada gambar 4.9 merupakan fungsi untuk menyimpan data guru ke dalam database melalui model dari hasil data yang di kirim melalui formulir tambah guru. Pada function ini terdapat fungsi bawaan darii codeigniter untuk melakukan validasi data yang di isikan pada form tambah guru, jika pada saat function di jalankan tetapi ada data yang gagal di validasi makan function akan mengembalikan ke tampilan form tambah data guru dengan menampilkan indikator

SAMPLE DRAFT

kesalahan yang harus dilengkapi. Jika seluruh data yang di kirim berhasil telah di validasi maka data akan di simpan di database.

```
1. public function tambah_guru()
2. {
3.        $this->form_validation->set_rules('nama_guru',       'Nama
   Guru', 'required');
4.        $this->form_validation->set_rules('niy',       'NIY/NIGK',
   'required');
5.        $this->form_validation->set_rules('nik',            'NIK',
   'required|numeric');
6.        $this->form_validation->set_rules('kelamin',   'Kelamin',
   'required');
7.        $this->form_validation->set_rules('alamat',     'Alamat',
   'required');
8.        $this->form_validation->set_rules('telpon',     'Telepon',
   'required');
9.        $this->form_validation->set_rules('username',  'Username',
   'required');
10.       $this->form_validation->set_rules('password',  'Password',
   'required');
11.       $this->form_validation->set_rules('repassword',
   'Konfirmasi Password', 'required|matches[password]');
12.
13.       if ($this->form_validation->run() == false) {
14.       $this->setmessage(validation_errors(), 'warning');
15.       $this->data_guru();
16.       } else {
17.       $post = $this->input->post();
18.       $data = array(
19.             'nama_guru'                                    =>
   trim(ucwords(htmlentities($post['nama_guru']))),
20.             'nip' => trim(htmlentities($post['niy'])),
21.             'kelamin' => trim(htmlentities($post['kelamin'])),
22.             'alamat_guru'                                  =>
   trim(ucwords(htmlentities($post['alamat']))),
23.             'telpon_guru'                                  =>
   trim(htmlentities($post['telpon'])),
24.             'username' => trim(htmlentities($post['username'])),
25.             'nik'                                          =>
   trim(strtoupper(htmlentities($post['nik']))),
26.             'password'                                     =>
   trim(md5(htmlentities($post['password']))),
27.       );
28.       $this->m_admin->insert_dataTo($data, 'data_guru');
29.       $this->setmessage('Data  Guru  berhasil  ditambahkan  :)',
   'success');
30.       redirect('admin/data_guru');
31.       }
32.}
```

Gambar 4.10 Fungsi Tampil Halaman Penjadwalan

Pada gambar 4.10 merupakan fungsi untuk menampilkan halaman penjadwalan mata pelajaran pada guru. Data yang diambil dan kemudian

SAMPLE DRAFT

ditampilkan pada halaman ini adalah guru, mata pelajaran, kelas, semester aktif, dan tahun ajaran yang aktif.

```
1. public function form_tambah_jadwalguru()
2. {
3. $id_admin = $this->session->userdata('id');
4. $where = array('id_admin' => $id_admin);
5. $data['user']  =  $this->m_admin->select_dataWhere($where,
   'user_admin');
6. $data['guru']              =              $this->m_admin-
   >select_table_orderby('nama_guru ASC', 'data_guru');
7. $data['mapel']             =              $this->m_admin-
   >select_table_orderby('nama_pelajaran                ASC',
   'setup_pelajaran');
8. $data['kelas']             =              $this->m_admin-
   >select_table_orderby('nama_kelas ASC', 'setup_kelas');
9. $data['tahun']             =              $this->m_admin-
   >select_table_orderby('tahun DESC', 'setup_tahun');
10.      $data['semester']          =          $this->m_admin-
   >select_table_orderby('id_semester                   ASC',
   'setup_semester');
11.      $data['pd_tahun'] = $setup_tahun->tahun;
12.      $where = array(
13.      'jadwal.id_kelas' => $id_kelas,
14.      'jadwal.id_semester' => $id_semester,
15.      'jadwal.id_tahun' => $setup_tahun->id_tahun,
16.      );
17.      $data['jdwl_guru']          =          $this->m_admin-
   >jadwal_guru_mengajar($where);
18.      $data['jml_data'] = $data['jdwl_guru']->num_rows();
19.      $siteurl = site_url('admin/jadwal_guru/');
20.      $rows = $data['jml_data'];
21.      $perpage = $per_hlm;
22.      $urisegment = 3;
23.      $Mfunction = "jadwal_guru_wp";
24.      $key = array(
25.      'jadwal.id_kelas' => $id_kelas,
26.      'jadwal.id_semester' => $id_semester,
27.      'jadwal.id_tahun' => $setup_tahun->id_tahun,
28.      );
29.      $model = "m_admin";
30.      $type = "ada_where";
31.      include "pagination_config.php";
32.      $data['content']                                    =
   "admin/penjadwalan/v_form_jadwal_guru";
33.      $this->load->view('admin/index', $data);
34.      }
```

Gambar 4.11 Fungsi Tambah Data Jadwal

Pada gambar 4.11 merupakan function untuk menyimpan data yang telah dikirim dari form input data penjadwalan guru. Pada function ini juga terdapat

SAMPLE DRAFT

fungsi bawaan dari codeigniter untuk melakukan validasi form dari data yang dikirim form penjadwalan guru. Jika ada data yang gagal di validasi maka function ini akan mengembalikan ke halaman form penjadwalan guru, kemudian jika telah berhasil di validasi maka data akan di kirimkan ke database dengan mengirimkan data ke model m_admin dengan function insert_dataInto di model m_admin.

```
1. public function set_jadwal_guru()
2. {
3. $data = array(
4. 'id_guru'          =>          trim(htmlentities($this->input-
   >post('id_guru'))),
5. 'id_pelajaran'       =>       trim(htmlentities($this->input-
   >post('id_pelajaran'))),
6. 'id_kelas'         =>          trim(htmlentities($this->input-
   >post('id_kelas'))),
7. 'id_tahun'         =>          trim(htmlentities($this->input-
   >post('tahun'))),
8. 'id_semester'       =>       trim(htmlentities($this->input-
   >post('semester'))),
9. );
10.       $where = $data;
11.       $cek     =     $this->m_admin->select_dataWhere($where,
   'tbl_jadwal')->num_rows();
12.       if ($cek > 0) {
13.       $this->setmessage('Sytem menemukan data yang sama,
   proses tidak dapat dilanjutkan!', 'danger');
14.       redirect('admin/jadwal_guru?m=penjadwalan&sm=guru_men
   gajar');
15.       } else {
16.       $this->m_admin->insert_dataTo($data, 'tbl_jadwal');
17.       $this->setmessage('Data berhasil disimpan!',
   'success');
18.       redirect('admin/jadwal_guru?m=penjadwalan&sm=guru_men
   gajar');
19.       }
20.       }
```

Gambar 4.12 Fungsi Daftar guru

Pada gambar 4.12 merupakan function untuk menampilkan daftar guru yang telah melakukan penjadwalan berdasarkan mata pelajaran dan kelas. Untuk mendapatkan data function data_guru mengambil data dari model m_admin dengan

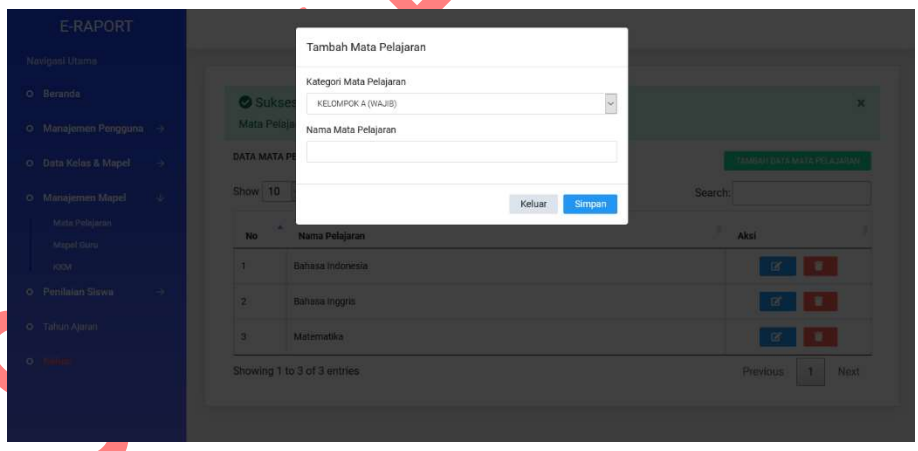
SAMPLE DRAFT

function select_dataWhere, dan data yang di dapat disimpan di dalam variable $data[‘dataguru’].

```
1. public function data_guru()
2. {
3. $id_admin = $this->session->userdata('id');
4. $where = array('id_admin' => $id_admin);
5. $data['user']      =      $this->m_admin->select_dataWhere($where,
   'user_admin');
6. $data['dataguru']                =                $this->m_admin-
   >select_table_orderby('nama_guru ASC', 'data_guru');
7. $siteurl = site_url('admin/data_guru/');
8. $rows = $data['dataguru']->num_rows();
9. $perpage = 10;
10. $urisegment = 3;
11. $Mfunction = "data_guru_wp";
12. $model = "m_admin";
13. $type = "tanpa_where";
14. include "pagination_config.php";
15. $data['content'] = "admin/guru/v_data_guru";
16. $this->load->view('admin/index', $data);
17. }
```

Gambar 4.13

4.1.5 Fungsional Manajemen Mata Pelajaran



Gambar 4.14 Halaman tambah mata pelajaran

Pada fungsional manajemen mata pelajaran, dapat diakses oleh administrator sebagai pemilik level akun tertinggi dalam sistem. Fungsi-fungsi yang ada antara lain adalah admin dapat melakukan create,read,update dan delete

DRAFT

pada data mata pelajaran. Gambar 4.15 merupakan fungsi untuk menampilkan seluruh data mata pelajaran.

```
1. public function setup_pelajaran()
2. {
3. $id_admin = $this->session->userdata('id');
4. $where = array('id_admin' => $id_admin);
5. $data['user']      =      $this->m_admin->select_dataWhere($where,
   'user_admin');
6. $data['datamapel']                =                $this->m_admin-
   >select_table_orderby('nama_pelajaran ASC', 'setup_pelajaran');
7. $data['kat_mapel']                =                $this->m_admin-
   >select_table_orderby('kategori_mapel                ASC',
   'tbl_kategori_mapel');
8. $data['content'] = "admin/mapel/v_setup_pelajaran";
9. $this->load->view('admin/index', $data);
10. }
```

Gambar 4.15 Fungsi Manajemen Kelas

Pada fungsi gambar 4.13 merupakan untuk menyimpan data mata pelajaran ke dalam database melalui model m_admin, tetapi sebelum data dikirim ke model, fungsi create_mapel melakukan validasi data terlebih dahulu, untuk data yang divalidasi adalah nama mata pelajaran yang tidak boleh kosong.

```
1. public function create_mapel()
2. {
3. $this->form_validation->set_rules('nama_pelajaran',   'Nama   Mata
   Pelajaran', 'required|max_length[50]');
4. if ($this->form_validation->run() == false) {
5. $this->session->set_flashdata('error_open',   '<div   class="alert
   alert-warning      alert-dismissable"><button      type="button"
   class="close"              data-dismiss="alert"              aria-
   hidden="true">&times;</button> <h4><i class="fa fa-warning"></i>
   Gagal</h4>');
6. $this->session->set_flashdata('validation_errors',
   validation_errors());
7. $this->session->set_flashdata('error_close', '</div>');
8. redirect('admin/setup_pelajaran');
9. } else {
10. $data = array(
11. 'id_kat_mapel' => $this->input->post('kat_mapel'),
12. 'nama_pelajaran' => $this->input->post('nama_pelajaran'),
13. );
14. $this->m_admin->insert_dataTo($data, 'setup_pelajaran');
15. $this->session->set_flashdata('sukses', '<div class="alert alert-
   success  alert-dismissable"><button  type="button"  class="close"
   data-dismiss="alert"  aria-hidden="true">&times;</button>  <h4><i
   class="fa  fa-check-circle"></i>  Sukses!</h4>  Mata  Pelajaran
   berhasil ditambahkan :) </div>');
16. redirect('admin/setup_pelajaran');
17. }
```

SAMPLE DRAFT

```
18. }
```

Gambar 4.16 Fungsi Simpan Mata Pelajaran

Pada gambar 4.16 merupakan fungsi untuk menampilkan daftar mata pelajaran beserta dengan nilai KKMnya. Untuk data yang diambil dari database melalui model adalah mata pelajaran dan nilai kkm yang telah diisikan permata pelajaran. Dan di fitur KKM ini admin dapat menambahkan atau mengisikan nilai KKM baru dari mata pelajaran yang belum memiliki nilai KKM serta dapat mengupdate nilai KKM.

```
1.  public function setup_kkm()
2.  {
3.  $where = array('status_aktif' => 1);
4.  $setup_tahun = $this->m_admin->select_dataWhere($where, 'setup_tahun')-
    >row();
5.  $kategori_kls = $this->m_admin->select_table_orderby('kategori_kls ASC',
    'setup_kelas')->row('kategori_kls');
6.  $id_admin = $this->session->userdata('id');
7.  $where = array('id_admin' => $id_admin);
8.  $data['user'] = $this->m_admin->select_dataWhere($where, 'user_admin');
9.  $get_tbl_kkm  =  $this->m_admin->select_tbl_kkm($setup_tahun->id_tahun,
    $kategori_kls);
10. $data['pd_tahun'] = $setup_tahun->tahun;
11. $data['id_tahun'] = $setup_tahun->id_tahun;
12. $data['tahun_ajaran']   =   $this->m_admin->select_table_orderby('tahun
    ASC', 'setup_tahun');
13. $data['mapel']   =   $this->m_admin->select_table_orderby('nama_pelajaran
    ASC', 'setup_pelajaran');
14. if ($get_tbl_kkm->num_rows() > 0) {
15. $data['type_form'] = "update";
16. $data['tbl_kkm'] = $get_tbl_kkm;
17. $data['mapel_notset']                =                $this->m_admin-
    >select_mapel_notset_kkm($data['id_tahun'], $kategori_kls);
18. $data['kategori_kls'] = $kategori_kls;
19. } else {
20. $data['type_form'] = "input";
21. $data['tbl_kkm'] = $this->m_admin->get_list_mapel();
22. $data['kategori_kls'] = $kategori_kls;
23. }
24. $data['content'] = "admin/kkm/v_setup_kkm";
25. $this->load->view('admin/index', $data);
26. }
```

Gambar 4.17

Pada gambar 4.17 merupakan fungsi untuk menyimpan maupun mengupdate nilai dari KKM yang telah di input. Untuk mengetahui nilai KKM yang telah diinput atau belum function set_kkm akan mengecek terlebih dahulu apakah

SAMPLE DRAFT

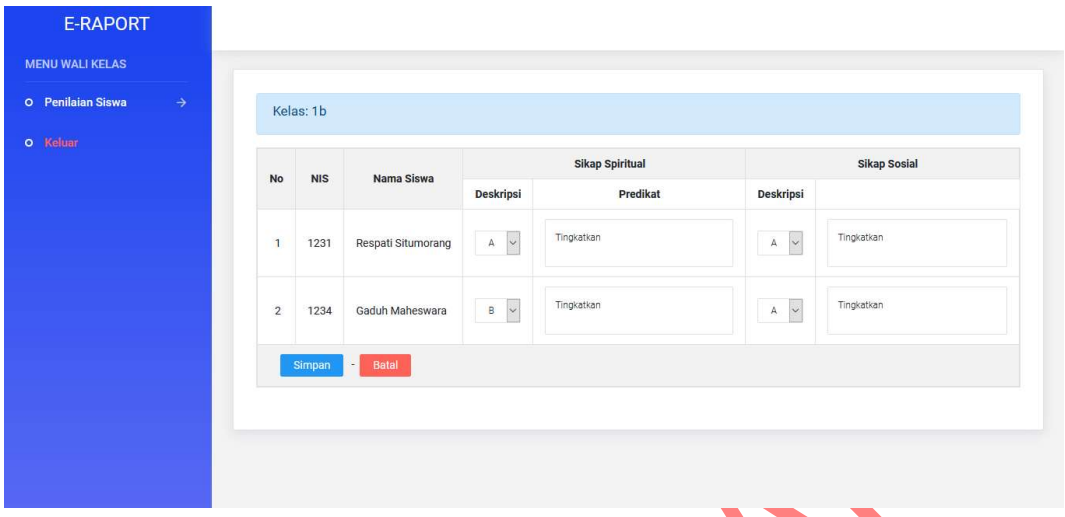
ada nilai KKM yang sudah terinput, jika sudah maka akan terbaca sebagai data update jika belum maka akan terbaca sebagai data baru.

```
1. public function set_kkm(){
2. $id_admin = $this->session->userdata('id');
3. $where = array('id_admin' => $id_admin);
4. $data['user']      =      $this->m_admin->select_dataWhere($where,
   'user_admin');
5. $data['tahun_ajaran']              =              $this->m_admin-
   >select_table_orderby('tahun ASC', 'setup_tahun');
6. $data['pd_tahun']  =  $this->m_admin->select_dataWhere('id_tahun='
   .    $this->input->post('idtahun')    .    '',    'setup_tahun')-
   >row('tahun');
7. $data['id_tahun'] = $this->input->post('idtahun');
8. $where = array(
9. 'id_tahun' => trim($this->input->post('idtahun')),
10. 'kategori_kls' => trim($this->input->post('kategori_kls')),
11. $cektabel  =  $this->m_admin->select_dataWhere($where,  'tbl_kkm')-
   >num_rows();
12. if ($cektabel > 0) {
13. $post = $this->input->post();
14. $data['tbl_kkm']                =              $this->m_admin-
   >select_tbl_kkm($post['idtahun']);
15. $data['mapel_notset']           =              $this->m_admin-
   >select_mapel_notset_kkm($post['idtahun']);
16. $data['type_form'] = "update";
17. } else {
18. $post = $this->input->post();
19. $data['tbl_kkm'] = $this->m_admin->get_list_mapel();
20. $data['type_form'] = "input";
21. $data['kategori_kls'] = trim($this->input->post('kategori_kls'));
22. $data['content'] = "admin/kkm/v_setup_kkm";
23. $this->load->view('admin/index', $data);
```

Gambar 4.18 Fungsi Nilai KKM

SAMPLE DRAFT

4.1.6 Fungsional Manajemen Nilai Raport



Gambar 4.19 Halaman Manajemen Nilai Rapot

Pada fungsional manajemen Raport, dapat diakses oleh administrator sebagai pemilik level akun tertinggi dalam sistem dan wali kelas sebagai pengisi nilai untuk murid dari setiap kelas. Fungsi-fungsi yang ada antara lain adalah admin dapat melakukan CRUD untuk modul nilai mata pelajaran yang di bagi menjadi 2 yaitu nilai keterampilan dan pengetauan, nilai sosial, nilai spiritual, ekstrakurikuler, absensi, tinggi dan berat, dan catatan wali kelas .

Gambar 4.16 merupakan fungsi untuk menampilkan seluruh data nilai sikap spiritual dan sikap sosial. Pada fungsi form nilai sikap ini memiliki beberapa fungsi yaitu menampilkan nama siswa beserta nilai spiritual dan sikap sosial. Dan fungsi untuk melakukan penambahan nilai serta update nilai. Untuk membedakan apakah data di input baru dan update terdapat pada script nomor 19 dan 23, untuk nilai yang masuk dalam kondisi input baru maka akan di kirimkan ke model m_guru dengan fungsi nilai_sikap1 dan untuk yang masuk dalam kondisi update akan di kirimkan ke model m_guru dengan fungsi nilai sikap_2.

SAMPLE DRAFT

```
1. public function form_nilai_sikap()
2. {
3. $setup_tahun = $this->m_guru->get_ref_thn_aktif()->row();
4. $setup_semester = $this->m_guru-
   >select_dataWhere('status_semester=1','setup_semester')->row();
5. $data['thn_aktif'] = $setup_tahun->tahun;
6. $data['idtahun'] = $setup_tahun->id_tahun;
7. $data['semester_aktif'] = ucwords($setup_semester->semester);
8. $data['idsemester'] = $setup_semester->id_semester;
9. $id_guru = $this->session->userdata('id');
10. $where = array('id_guru' => $id_guru);
11. $data['user'] = $this->m_guru-
   >select_dataWhere($where,'data_guru');
12. include "include/id_wali_kelas.php";
13. $data['wali_id_wali'] = $wali_id_wali;
14. $post_idtahun = trim(htmlentities($this->input-
   >post('idtahun')));
15. $post_idkelas = trim(htmlentities($this->input-
   >post('idkelas')));
16. $post_semester = trim(htmlentities($this->input-
   >post('semester')));
17. $data['kelas'] = $this->m_guru-
   >select_dataWhere('id_kelas='.$post_idkelas.'','setup_kelas');
18. $data['id_kelas'] = $post_idkelas;
19. if(isset($_POST['input'])){
20. $data['nilai_sikap'] = $this->m_guru-
   >get_nilai_sikap1($post_idkelas,$post_idtahun);
21. $data['jumSis'] = $data['nilai_sikap']->num_rows();
22. $data['content'] = "guru/nilai_sikap/v_form_nilai_sikap";
23. }elseif(isset($_POST['update']))
24. {
25. $data['page_title'] = '<h1>Update Nilai Sikap Siswa
   '.$setup_tahun->tahun.' Semester
   '.$data['semester_aktif'].'</h1>';
26. $data['nilai_sikap'] = $this->m_guru-
   >get_nilai_sikap2($post_idkelas,$post_idtahun,$post_semester);
27. $data['jumSis'] = $data['nilai_sikap']->num_rows();
28. $data['content'] = "guru/nilai_sikap/v_form_up_nilai_sikap";
29. }
30. $this->load->view('guru/index',$data);
31. }
```

Gambar 4.20 Fungsi Nilai Sikap

Gambar 4.17 merupakan fungsi untuk menampilkan nilai pengetahuan dan keterampilan siswa per mata pelajaran. Kemudian di fungsi ini juga terdapat aksi untuk melakukan update nilai atau input nilai baru dari setiap siswa yang telah mengambil mata pelajaran. Untuk membedakan input baru dan update pada saat melakukan proses penyimpanan akan di cek terlebih dahulu apakah data nilai dari siswa ada di database jika tidak ada maka akan masuk ke dalam kondisi penyimpan

SAMPLE DRAFT

baru sedangkan jika ada maka akan masuk ke dalam penyimpanan update. Untuk penyimpanan data baru dan data update akan dikirim ke model m_guru dengan fungsi list_siswa_dikelas dengan parameter yang membedakan adalah $data['mode_form'] untuk data baru akan berisi input sedangkan untuk update akan berisi update.

Gambar 4.18 merupakan fungsi untuk mengelola data nilai ekstrakurikuler. Pada halaman yang mengelola ekstrakurikuler data yang diambil dan ditampilkan adalah user dan jenis kegiatan ekstrakurikuler. Kemudian jika telah menuliskan data - data nilai ekstrakurikuler data akan di simpan ke database melalui model m_guru dengan function lihat_nilai_ekstra dengan data yang dikirimkan pada parameternya adalah idwali, idsemester, idtahun, dan query untuk mengkondisikan agar data yang di update sesuai.

```
1. public function edit_nilai_ekstra()
2. {
3. $setup_tahun = $this->m_guru->get_ref_thn_aktif()->row();
4. $setup_semester = $this->m_guru-
   >select_dataWhere('status_semester=1','setup_semester')->row();
5. $data['thn_aktif'] = $setup_tahun->tahun;
6. $data['idtahun'] = $setup_tahun->id_tahun;
7. $data['semester_aktif'] = ucwords($setup_semester->semester);
8. $data['idsemester'] = $setup_semester->id_semester;
9. $id_guru = $this->session->userdata('id');
10. $where = array('id_guru' => $id_guru);
11. $data['user'] = $this->m_guru-
   >select_dataWhere($where,'data_guru');
12. $data['kegiatan_ekstra'] = $this->m_guru-
   >select_table_orderby('nama_ekstra ASC','ekstrakurikuler');
13. $get = $this->input->get();
14. if($get['set']=='update'){
15. $nis = $this->uri->segment(3);
16. $idkelas = $this->uri->segment(4);
17. $geturi5 = explode("-", $this->uri->segment(5));
18. $idwali = $geturi5['0'];
19. $idsemester = $geturi5['1'];
20. $idtahun = $geturi5['2'];
21. $where = "siswa.nis=$nis AND ekstra.id_kelas=$idkelas AND
   ekstra.id_wali=$idwali AND ekstra.id_semester=$idsemester AND
   ekstra.id_tahun=$idtahun";
22. $data['nilai_ekstra'] = $this->m_guru-
   >lihat_nilai_ekstra($where);
23. $data['type_form'] = "update";
```

SAMPLE DRAFT

```
24. $data['content'] = "guru/ekstra/v_form_in_nilai_ekstra";
25. }
26. $this->load->view('guru/index',$data);
27. }
```

Gambar 4.21 Fungsi Nilai Ektrakulikuler

Gambar 4.18 merupakan function untuk mengelola input nilai prestasi, dimana pada halaman input nilai prestasi mengambil data siswa dengan kelas yang dikelola oleh wali kelas yang melakukan login, terdapat 2 inputan yaitu nama prestasi yang diperoleh siswa dan keterangannya, setelah data input di submit maka data tersebut akan dikirimkan ke model m_guru dengan fungsi insert_dataTo dengan data yang dikirim kan adalah nis, id_kelas, id_wali, id_semester, id_tahun, jenis_kegiatan, dan keterangan. Setelah data berhasil di input maka akan dikembalikan ke halaman dari function input prestasi serta menampilkan data prestasi siswa yang telah diinputkan.

```
1.  function set_prestasi_siswa()
2.  {
3.  $this->form_validation->set_rules('jenis','Jenis
    Kegiatan','required');
4.  $this->form_validation->set_rules('ket','Keterangan','required');
5.  $post = $this->input->post();
6.  if($this->form_validation->run()==FALSE){
7.  $this->setmessage(validation_errors(),'warning');
8.  redirect('guru/input_prestasi?m=input_nilai&sm=prestasi');
9.  }else{
10. $data = array(
11. 'nis' => $post['nis'],
12. 'id_kelas' => $post['idkelas'],
13. 'id_wali' => $post['idwali'],
14. 'id_semester' => $post['idsemester'],
15. 'id_tahun' => $post['idtahun'],
16. 'jenis_kegiatan' => trim(strip_tags($post['jenis'])),
17. 'keterangan' => trim(strip_tags($post['ket']))
18. );
19. $this->m_guru->insert_dataTo($data,'tbl_prestasi_siswa');
20. $this->setmessage('Prestasi siswa berhasil disimpan!','success');
21. redirect('guru/input_prestasi?m=input_nilai&sm=prestasi');
22. }
23. }
```

Gambar 4.22 Prestasi Siswa

SAMPLE DRAFT

Gambar 4.19 merupakan fungsi untuk mengatur input nilai tinggi dan berat dari siswa. Di dalam fungsi ini terdapat aksi validasi yang mengatur agar data yang sifatnya tidak boleh kosong untuk terisi terlebih dahulu, kemudian jika lolos dalam proses validasi data maka data tersebut akan dikirimkan ke model dan simpan melalui function di model dengan nama m_store_berat_tinggi disertai dengan nilai yang dikirimkan melalui parameter, setelah itu maka akan dikembalikan ke function tinggi_berat beserta dengan notifikasi, begitu juga jika validasi data gagal maka akan dikembalikan ke function tinggi_badan.

```
1. public function store_berat_tinggi()
2. {
3. $this->form_validation->set_rules('nama_siswa', 'Nama Siswa',
   'required');
4. if ($this->form_validation->run() == false ) {
5. $this->session->set_flashdata('sukses', '<div class="alert alert-
   danger alert-dismissable"><button type="button" class="close"
   data-dismiss="alert" aria-hidden="true">&times;</button> <h4><i
   class="fa fa-check-circle"></i> Gagal</h4> Data siswa sudah di
   masukkan </div>');
6. redirect('admin/tambah_berat_tinggi');
7. } else {
8. $post = [
9. 'id_siswa' => $this->input->post('nama_siswa',true),
10. 'berat_badan' => $this->input->post('berat',true),
11. 'tinggi_badan' => $this->input->post('tinggi',true),
12. 'id_semester' => $this->input->post('semester',true),
13. 'id_tahun' => $this->input->post('tahun',true)
14. ];
15. $this->m_admin->m_store_berat_tinggi($post);
16. $this->session->set_flashdata('sukses', '<div class="alert alert-
   success alert-dismissable"><button type="button" class="close"
   data-dismiss="alert" aria-hidden="true">&times;</button> <h4><i
   class="fa fa-check-circle"></i> Sukses</h4> Menambah Data
   </div>');
17. redirect('admin/tinggi_berat');
18. }
19. }
```

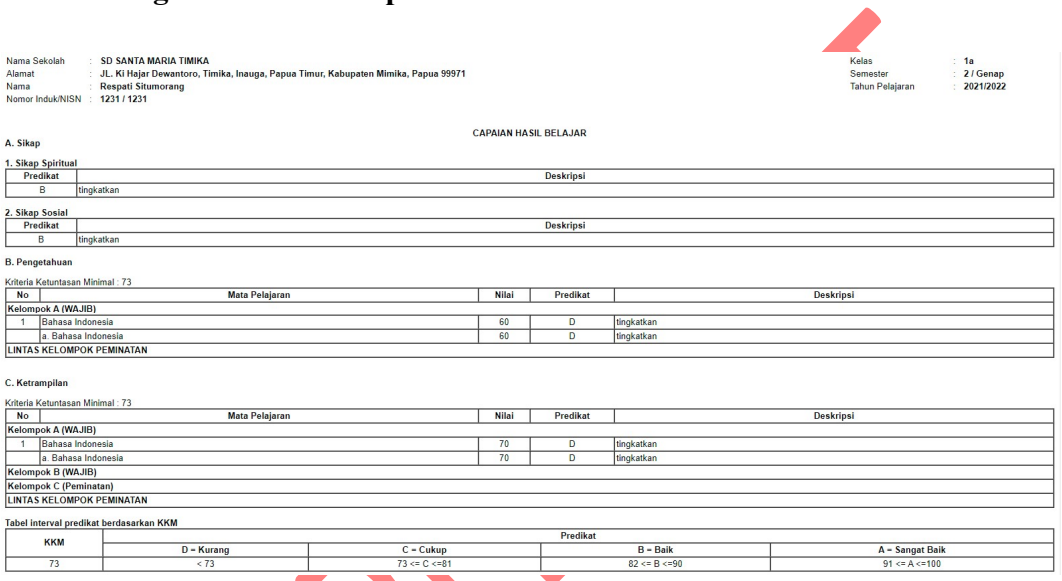
Gambar 4.23 Fungsi Berat Dan Tinggi

Gambar 4.x berfungsi untuk mengolah data kehadiran dari siswa, untuk data kehadiran yang dapat dikelola adalah sakit, izin, dan tanpa keterangan. Untuk data input kehadiran akan di simpan ke database melalui model m_guru dengan function

SAMPLE DRAFT

insert_batch dengan nilai parameter yang dikirimkan adalah nis, id_kelas, id_tahun, semester, sakit, izin, dan tnp_ket. Setelah proses data berhasil disimpan maka akan diarahkan ke function input_kehadian beserta notifikasi karena function inilah yang mengelola tampilan input halaman ketidak hadiran siswa.

4.1.7 Fungsional Cetak Raport Siswa



Nama Sekolah	: SD SANTA MARIA TIMIKA	Kelas	: 1a
Alamat	: JL. Ki Hajar Dewantoro, Timika, Inauga, Papua Timur, Kabupaten Mimika, Papua 99971	Semester	: 2 / Genap
Nama	: Respati Situmorang	Tahun Pelajaran	: 2021/2022
Nomor Induk/NISN	: 1231 / 1231		

CAPAIAN HASIL BELAJAR

A. Sikap

1. Sikap Spiritual

Predikat	Deskripsi
B	tingkatkan

2. Sikap Sosial

Predikat	Deskripsi
B	tingkatkan

B. Pengetahuan

Kriteria Ketuntasan Minimal : 73

No	Mata Pelajaran	Nilai	Predikat	Deskripsi
Kelompok A (WAJIB)				
1	Bahasa Indonesia	60	D	tingkatkan
	a. Bahasa Indonesia	60	D	tingkatkan
LINTAS KELOMPOK PEMINATAN				

C. Ketrampilan

Kriteria Ketuntasan Minimal : 73

No	Mata Pelajaran	Nilai	Predikat	Deskripsi
Kelompok A (WAJIB)				
1	Bahasa Indonesia	70	D	tingkatkan
	a. Bahasa Indonesia	70	D	tingkatkan
Kelompok B (WAJIB)				
Kelompok C (Peminatan)				
LINTAS KELOMPOK PEMINATAN				

Tabel interval predikat berdasarkan KKM

KKM	Predikat			
	D = Kurang	C = Cukup	B = Baik	A = Sangat Baik
73	< 73	73 <= C <=81	82 <= B <=90	91 <= A <=100

Gambar 4.24 Cetak Rapot Siswa

Pada fungsional Cetak Raport siswa, dapat menampilkan data nilai raport keseluruhan dari siswa, sesuai dengan semester dan tahun akademik yang aktif. Untuk data yang di ambil dari fungsi ini untuk menampilkan raport adalah siswa, tahun_ajaran, semester, kelas, nilai_sikap, nilai_rapor, deskripsi_nilai, nilai_ekstra, nilai_prestasi, kehadiran, catatan_wk, wali_kelas, tinggi_berat.

```
1. public function print_nilai_rapor()
2. {
3. $nis = $this->uri->segment(3);
4. $idkelas = $this->uri->segment(4);
5. $id_tahun = $this->uri->segment(5);
6. $semester = $this->input->get('semester');
7. $post = [
8. 'nis'          => $nis,
```

SAMPLE

```
9.  'idkelas'     => $idkelas,
10. 'id_tahun'     => $id_tahun,
11. 'semester'     => $semester
12. ];
13. $data['sekolah'] = $this->m_cetak->get_data_sekolah();
14. $data['siswa']                  =                  $this->m_cetak-
    >select_dataWhere('nis='.$nis.'','data_siswa');
15. $data['tahun_ajaran']               =              $this->m_cetak-
    >select_dataWhere('id_tahun='.$id_tahun.'','setup_tahun')-
    >row();
16. $data['semester']                 =                $this->m_cetak-
    >select_dataWhere('id_semester='.$semester.'','setup_semester')-
    >row();
17. $data['kelas']                  =                  $this->m_cetak-
    >select_dataWhere('id_kelas='.$idkelas.'','setup_kelas')->row();
18. $kategori_kls = $data['kelas']->kategori_kls;
19. $data['nilai_sikap']                =                $this->m_cetak-
    >get_nilai_sikap($nis,$idkelas,$id_tahun,$semester);
20. $data['nilai_rapor']                =                $this->m_cetak-
    >get_nilai_rapor($nis,$idkelas,$id_tahun,$semester,$kategori_kls
    );
21. $data['deskripsi_nilai']            =                $this->m_cetak-
    >get_deskripsi_nilai($nis,$idkelas,$id_tahun,$semester);
22. $data['nilai_ekstra']               =                $this->m_cetak-
    >get_nilai_ekstra($nis,$idkelas,$id_tahun,$semester);
23. $data['nilai_prestasi']             =                $this->m_cetak-
    >get_nilai_prestasi($nis,$idkelas,$id_tahun,$semester);
24. $data['kehadiran']                  =                $this->m_cetak-
    >get_kehadiran_siswa($nis,$idkelas,$id_tahun,$semester);
25. $data['catatan_wk']                 =                $this->m_cetak-
    >get_cttnwk($nis,$idkelas,$id_tahun,$semester);
26. $data['wali_kelas']                 =                $this->m_cetak-
    >get_wk_saat_ini($id_tahun,$idkelas)->row();
27. $data['tinggi_berat'] = $this->m_cetak->get_tinggi_berat($post);
28. $this->load->view('admin/cetak_rapor/v_hal_nilai_rapor',$data);
29. }
```

Gambar 4.25 Fungsi Nilai Rapot

Gambar 4.21 merupakan tampilan untuk nilai raport serta pengkategorian nilai di raport. Di sistem ini untuk melakukan pengkategorian nilai raport dilakukan di bagian view cetak nilai raport.

SAMPLE DRAFT

```
1. <?php
2. $s13 = 'a';
3. foreach ($nilai_rapor->result() as $subrownilai) {
4. ?>
5. <tr>
6. <td align="center" valign="middle"></td>
7. <td   align="left"   valign="middle"><?php   echo   "$s13.   "   .
   ucwords($subrownilai->nama_pelajaran); ?></td>
8. <?php $s13 = chr(ord($s13) + 1); ?>
9. <td   align="center"   valign="middle"><?php   echo   $subrownilai-
   >nilai_ketrampilan; ?></td>
10. <td align="center" valign="middle">
11. <?php if ($subrownilai->nilai_ketrampilan < 73) {
12. echo "D";
13. } elseif ($subrownilai->nilai_ketrampilan >= 73 && $subrownilai-
   >nilai_ketrampilan <= 81) {
14. echo "C";
15. } elseif ($subrownilai->nilai_ketrampilan >= 82 && $subrownilai-
   >nilai_ketrampilan <= 90) {
16. echo "B";
17. } elseif ($subrownilai->nilai_ketrampilan >= 91 && $subrownilai-
   >nilai_ketrampilan <= 100) {
18. echo "A";
19. }
20. ?></td>
21. <td valign="middle"><?php echo $subrownilai->ketrampilan; ?></td>
22. </tr>
23. <?php}?>
```

Gambar 4.26 Tampil Nila Rapot

4.2 Pembahasan Sistem

Pada penelitian ini melakukan uji coba sistem dengan meggunakan metode blackbox testing, yaitu pengujian dengan menggunakan aplikasi secara langsung untuk mengetahui fungsional-fungsional dalam sistem dapat berjalan dengan baik sesuai kebutuhan dan di gunakan untuk identifikasi kesalahan (bug) pada sistem. hasil pengujian dapat dlihat pata tabel 4.1

Tabel 4.1 Tabel Pengujian Fungsional

No	Aktifitas Pengujian	Hasil Yang Di Harapkan	Kesimpulan
1.	Melakukan Login Pengguna (Admin,Guru dan Orangtua/wali)	Login sistem multi level	Valid

SAMPLE DRAFT

No	Aktifitas Pengujian	Hasil Yang Di Harapkan	Kesimpulan
2.	CRUD Data Siswa	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Siswa.	valid
3.	CRUD Data Guru	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Guru	valid
4.	CRUD Data Kelas	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Kelas.	valid
5.	CRUD Data Mata Pelajaran & KKM	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Mata Pelajaran dan Nilai KKM.	valid
6.	CRUD Tahun Ajaran	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Tahun Ajaran. dan mampu digunakan untuk menonaktifkan TA tertentu	valid
7.	Penilaian Sikap	Sistem mampu melakukan penilaian SIkap Siswa Berdasarkan Kelas	valid
8.	Penilaian Pengetahuan Dan Ketrampilan	Sistem mampu melakukan penilaian Pengetahuan Dan Ketrampilan Berdasarkan Kelas	valid
9.	CRUD Prestasi Siswa	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Kelas.	valid
10.	CRUD Tinggi Dan Berat	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Tinggi Dan Berat.	valid
11.	CRUD Ekstrakulikuler Siswa	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Extrakulikuler.	valid
12.	CRUD Kondisi Fisik	Sistem mampu melakukan menampilkan, penambahan, ubahan, hapus data Kondisi Fisik.	valid
13.	Cetak Rapot	Sistem mampu mencetak rapot	valid

SAMPLE DRAFT

Berdasarkan data pada tabel 4.1 maka dapat di simpulkan bahwa sistem mampu mengakomodir kebutuhan pencapaian hasil belajar siswa ke orangtua.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasaran perancangan dan implementasi pada bab-bab sebelumnya pada pada penelitian ini dapat di ambil beberapa kesimpulan sebagai berikut :

1. Penelitian berhasil membangun sebuah aplikasi pelaporan capaian hasil belajar siswa
2. Aplikasi dapat berjalan sesuai kebutuhan dan rancangan fungsional yang ada.
3. Penggunaan framework Codeingiter sebagai dasar pembangunan aplikasi memudahkan dalam organisasi kode dan juga troubleshooting karena dukungan dokumentasi dan komunitas yang luas dan lengkap
4. MySQL sebagai DBMS memiliki kinerja yang baik dalam mengakomodir kebutuhan data.

5.2 Saran

1. Menggunakan rancangan arsitekrur yang lebih moderen yaitu memisahkan backend dan frontend , dan komunikasi data atara keduanya menggunakan web service

SAMPLE DRAFT

2. Memanfaatkan teknologi frontend seperti ReactJS , Vue atau teknologi antarmuka yang bersifat client rendering sehingga data akan terbaca lebih cepat guna meningkatkan pengalaman pengguna
3. menggunakan JSON/XML sebagai pertukaran data yang lebih ringan
4. Menambahkan fitur pemberitahuan melalui whatsapp ataupun email.
5. Menggunakan nomor telephone sebagi login pengguna, sehingga tidak mudah lupa dan meningkatkan dari sisi keamanan.

SAMPLE DRAFT