

BAB 6 IMPLEMENTASI

Bab ini membahas implementasi dari perancangan perangkat lunak yang telah dibuat sebelumnya. Bab ini berisikan spesifikasi sistem, implementasi Algoritme, dan implementasi antarmuka.

6.1 Spesifikasi Sistem

Sistem Informasi Manajemen DIKLAT pendidik dan tenaga kependidikan ini adalah sebuah sistem informasi manajemen berbasis web yang dibuat untuk mempermudah proses-proses yang dilakukan dalam kegiatan DIKLAT. Untuk menjalankan sistem ini, membutuhkan spesifikasi komputer dan koneksi internet yang memadai. Berikut ini merupakan spesifikasi perangkat lunak yang dibutuhkan untuk mengakses sistem informasi manajemen DIKLAT:

1. Sistem operasi *microsoft windows 7, windows 8, windows 8.1, windows 10, dan Linux.*
2. Web browser *Google chrome versi 61.0.3163.100, dan Microsoft Edge versi 40.15063.674.0*

Selain perangkat lunak, terdapat syarat minimum spesifikasi perangkat keras yang dibutuhkan untuk dapat mengakses sistem informasi Manajemen DIKLAT:

1. *Processor Intel Core i3 2.4 Ghz*
2. *Ram 2 GB*
3. *VGA Intel HD Graphics*

6.2 Implementasi Algoritme

Implementasi Algoritme ini hanya menjelaskan proses pendaftaran peserta yang sebelumnya telah dijelaskan pada sub bab perancangan komponen. Proses pendaftaran ini meliputi emapt kelas yaitu kelas *c_peserta_daftar* sebagai *controller* yang mengatur alur logika proses dan kelas *m_peserta, m_peserta_berkas* dan *m_ruangDiklat* sebagai *model* yang mengatur pengolahan data pada proses pendaftaran.

6.2.1 Fungsi getData

Tabel 6.1 Implementasi Algoritme Fungsi getData

1	<code>function getData(\$nuptk) {</code>
2	<code> \$getDataJson = file_get_contents("http://" . \$_SERVER['HTTP_HOST'] . "/project/Dapodik/getData.php? username=Ednan&&password=12345&& nuptk=" . \$nuptk);</code>
3	<code> \$tmp = json_decode(\$getDataJson, true);</code>
4	<code> if(\$tmp != null){</code>
5	<code> \$data[] = \$tmp;</code>

```

6      $data[0]['label'] = $tmp['nuptk'];
7      echo json_encode($data);
8  }else{
9      echo json_encode($data);
10     }
11 }

```

Fungsi `getData` adalah fungsi yang digunakan untuk mendapatkan data peserta dari aplikasi DAPODIK dan kemudian disimpan dalam variabel tertentu untuk ditampilkan pada halaman pendaftaran. Fungsi ini diakses oleh `view` `daftar.php` ketika pengguna memasukkan NUPTK pada formulir pendaftaran. Fungsi ini terdapat pada kelas `c_peserta_daftar` dan dijabarkan pada Tabel 6.1.

6.2.2 Fungsi `submitData`

Tabel 6.2 Implementasi Algoritme `submitData`

```

1  function submitData() {
2      $this->dataSubmit = array(
3          "nuptk"=>$this->input->post('nuptk'),
4          "nip"=>$this->input->post('nip'),
5          "nama"=>$this->input->post('nama'),
6          "pangkat"=>$this->input->post('pangkat'),
7          "jabatan"=>$this->input->post('jabatan'),
8          "tempat_lahir"=>$this->input->post('tempat_lahir'),
9          "tanggal_lahir"=>$this->input->post('tanggal_lahir'),
10         "jenis_kelamin"=>$this->input->post('jenis_kelamin'),
11         "alamat"=>$this->input->post('alamat'),
12         "nomor_handphone"=>$this->input->post('nomor_handphone'),
13         "email"=>$this->input->post('email'),
14         "pendidikan_terakhir"=>$this->input->post('pendidikan_terakhir'),
15         "sekolah_asal"=>$this->input->post('sekolah_asal'),
16         "jenjang"=>$this->input->post('jenjang'),
17         "mata_pelajaran"=>strtoupper($this->input->
18         post('mata_pelajaran')),
19         "pengalaman_mengajar"=>$this->input->post('pengalaman_mengajar'),
20         "pengalaman_pelatihan"=>$this->input->post('pengalaman_pelatihan')
21     );
22     $config['upload_path']          = './uploads/foto';
23     $config['allowed_types']        = 'jpg|jpeg|png';
24     $config['max_size']              = 10240;
25     $this->load->library('upload', $config);
26     $this->upload->initialize($config);
27     if($this->upload->do_upload('foto')) {
28         $tempData = $this->upload->data();

```

```

28     $this->dataSubmit['foto'] = $tempData['file_name'];
29 }else{
30     $rusak = $this->upload->display_errors();
31     $sub1 = substr($rusak,3,86);
32     $error = substr($sub1,0,82);
33     echo $error;
34     echo "
35     <script language='JavaScript'>
36         alert('$error');
37         document.location='../';
38     </script>
39     ";
40 }
41 $dataHasRuangan = $this->processData();
42 $error[0] = $this->m_peserta->inputData($this->dataSubmit);
43 $error[1] = $this->m_ruangDiklat->inputHasRuangan($dataHasRuangan);
44 if($error[0]['message'] == null){
45     $files = $_FILES;
46     $count = $_FILES['berkas']['name'];
47     for($b=0; $b<count($count); $b++){
48         $_FILES['berkas']['name'][$b] = $files['berkas']['name'][$b];
49         $_FILES['berkas']['type'][$b] = $files['berkas']['type'][$b];
50         $_FILES['berkas']['tmp_name'][$b] = $files['berkas']['tmp_name'][$b];
51         $_FILES['berkas']['error'][$b] = $files['berkas']['error'][$b];
52         $_FILES['berkas']['size'][$b] = $files['berkas']['size'][$b];
53         $config['upload_path'] = './uploads/berkas';
54         $config['allowed_types'] = 'pdf|doc|docx';
55         $config['max_size'] = 10240;
56         $this->load->library('upload', $config);
57         $this->upload->initialize($config);
58         if($this->upload->do_upload('berkas')){
59             $tempData = $this->upload->data();
60             $dataUpload['id_peserta'] = $this->dataSubmit['nuptk'];
61             $dataUpload['berkas'] = $tempData['file_name'];
62             $errorDbBerkas=$this->m_peserta_berkas->
63             inputData($dataUpload);
64         }else{
65             $rusak = $this->upload->display_errors();
66             $sub1 = substr($rusak,3,86);
67             $error = substr($sub1,0,82);
68             echo $error;

```

```

68         echo "
69             <script language='JavaScript'>
70                 alert('$error');
71             document.location='../';
72             </script>
73         ";
74     }
75 }
76 echo "
77     <script language='JavaScript'>
78         alert('Pendaftaran berhasil, username & password anda akan
79         dikirimkan melalui email jika telah diverifikasi oleh
80         panitia');
81         document.location='../peserta/login';
82     </script>
83     ";
84 }else{
85     echo "
86         <script language='JavaScript'>
87             alert('Pendaftaran gagal anda telah terdaftar');
88             document.location='../';
89         </script>
90     ";
91 }

```

Fungsi `submitData` adalah fungsi yang dijalankan oleh kelas *controller* `c_peserta_daftar` saat pengguna menekan tombol daftar pada halaman pendaftaran. Fungsi ini berfungsi untuk menampung data yang telah dimasukkan oleh peserta sekaligus menggunggah foto dan berkas yang diunggah oleh peserta. Fungsi ini dijabarkan pada Tabel 6.2

6.2.3 Fungsi `processData`

Tabel 6.3 Implementasi Algoritme `processData`

```

1 function processData(){
2     if($this->dataSubmit['jenjang'] == 'SD01'){
3         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
4         => 'SD01'));
5     }elseif($this->dataSubmit['jenjang'] == 'SD02'){
6         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
7         => 'SD02'));
8     }elseif($this->dataSubmit['jenjang'] == 'SD03'){
9         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
10        => 'SD03'));

```

```

8      }elseif($this->dataSubmit['jenjang'] == 'SD04'){
9          $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
=> 'SD04'));
10     }elseif($this->dataSubmit['jenjang'] == 'SD05'){
11         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
=> 'SD05'));
12     }elseif($this->dataSubmit['jenjang'] == 'SD06'){
13         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
=> 'SD06'));
14     }
15     elseif(substr($this->dataSubmit['jenjang'],0,3) == "SMP" ){
16         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
=> 'SMP_'. $this->dataSubmit['mata_pelajaran']));
17         if($idRuangan == null){
18             $error = $this->m_ruangDiklat->inputRuangan(array('id_ruang'
=> 'SMP_'.strtoupper($this->dataSubmit['mata_pelajaran']));
19             $idRuangan[0]['id_ruang'] = 'SMP_'.strtoupper($this-
>dataSubmit['mata_pelajaran']);
20         }
21     }
22     elseif(substr($this->dataSubmit['jenjang'],0,3) == "SMA" ){
23         $idRuangan = $this->m_ruangDiklat->getRuangan(array('id_ruang'
=> 'SMA_'. $this->dataSubmit['mata_pelajaran']));
24         if($idRuangan == null){
25             $error = $this->m_ruangDiklat->inputRuangan(array('id_ruang'
=> 'SMA_'.strtoupper($this->dataSubmit['mata_pelajaran']));
26             $idRuangan[0]['id_ruang'] = 'SMA_'.strtoupper($this-
>dataSubmit['mata_pelajaran']);
27         }
28     }
29     $dataHasRuangan = array(
30         "id_ruang" => $idRuangan[0]['id_ruang'],
31         "id_peserta" => $this->dataSubmit['nuptk']
32     );
33     if($this->dataSubmit['jenis_kelamin'] == "Pria"){
34         $mess = $this->m_mess->getMess("id_mess LIKE '%B%'");
35         $idMess = $mess[0]['id_mess'];
36     }elseif($this->dataSubmit['jenis_kelamin'] == "Wanita"){
37         $mess = $this->m_mess->getMess("id_mess LIKE '%A%'");
38         $idMess = $mess[0]['id_mess'];
39     }
40     $this->dataSubmit['mess'] = $idMess;
41     return $dataHasRuangan;
42 }

```

Fungsi `ProcessData` adalah fungsi yang dipanggil oleh fungsi `SubmitData` untuk melakukan pengolahan data peserta. Pengolahan tersebut meliputi penentuan ruangan diklat peserta dan penentuan lokasi *mess* peserta. Fungsi ini mengeluarkan id ruangan peserta yang telah ditentukan berdasarkan jenjang dan mata pelajaran yang dimasukkan oleh pengguna. Selain itu, ketika fungsi ini dijalankan juga menyimpan id *mess* pada data peserta sesuai hasil pengolahan yaitu sesuai jenis kelamin yang dimasukkan oleh pengguna. Fungsi ini dijabarkan pada Tabel 6.3.

6.2.4 Fungsi `inputData` (kelas `m_peserta`)

Tabel 6.4 Implementasi Fungsi `inputData` (kelas `m_peserta`)

1	<code>function inputData(\$data = array()){</code>
2	<code> \$this->db->insert('peserta', \$data);</code>
3	<code> return \$this->db->error();</code>
4	<code>}</code>

Fungsi `inputData` digunakan untuk memasukkan data peserta yang dimasukkan peserta pada proses pendaftaran. Data-data tersebut berupa NUPTK, NIP, nama, pangkat, jabatan, tempat lahir, tanggal lahir, jenis kelamin, alamat, nomor *handphone*, email, pendidikan terakhir, sekolah asal, jenjang, mata pelajaran yang diampu, pengalaman mengajar, dan pengalaman pelatihan. Fungsi ini dijabarkan pada Tabel 6.4

6.2.5 Fungsi `inputData` (kelas `m_peserta_berkas`)

Tabel 6.5 Implementasi Fungsi `inputData` (kelas `m_peserta_berkas`)

1	<code>function inputData(\$data = array()){</code>
2	<code> \$this->db->insert('peserta_berkas', \$data);</code>
3	<code> return \$this->db->error();</code>
4	<code>}</code>

Fungsi `inputData` digunakan untuk memasukkan data berkas-berkas peserta yang dimasukkan dalam proses pendaftaran. Data berkas-berkas peserta yang akan dimasukkan dalam fungsi ini hanyalah string lokasi penyimpanan dimana berkas-berkas tersebut disimpan di dalam sistem. Fungsi ini dijabarkan pada Tabel 6.5.

6.2.6 Fungsi inputHasRuangan

Tabel 6.6 Implementasi Fungsi inputHasRuangan

1	<code>function inputHasRuangan(\$data = array()){</code>
2	<code> \$this->db->insert('peserta_has_rdiklat',\$data);</code>
3	<code> return \$this->db->error();</code>
4	<code>}</code>

Fungsi `inputHasRuangan` terdapat pada kelas `m_ruangDiklat` yang digunakan untuk memasukkan hasil pengolahan data penentuan ruangan diklat peserta pada *database*. Data yang dimasukkan pada fungsi ini adalah id data peserta yang telah dimasukkan pada *database* dan id ruangan diklat yang telah ditentukan akan ditempati oleh peserta. Algoritme fungsi ini dijelaskan pada Tabel 6.6.

6.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk menjabarkan antarmuka sistem informasi manajemen DIKLAT yang telah dikembangkan. Implementasi antarmuka yang akan dijabarkan hanya yang terdapat pada proses pendaftaran.

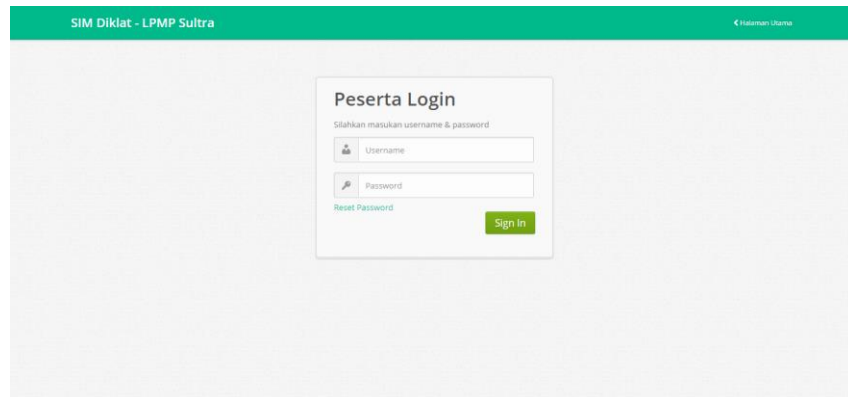
6.3.1 Halaman Daftar

The screenshot shows the registration page for SIM Diklat - LPMP Sultra. The page has a green header with the title "SIM Diklat - LPMP Sultra" and a link "Sudah mendaftar? Silahkan login disini". The main content area is titled "Pendaftaran" and contains a registration form. The form has two columns of fields. The left column includes: NUPTK (21313765...), NIP (21313765...), Nama (Ednan...), Pangkat (Penata Muda...), Jabatan (Guru SMP...), Tempat Lahir (Kendari...), Tanggal Lahir (mm/dd/yyyy), Jenis Kelamin (radio buttons for Laki-Laki and Perempuan), Alamat (Jl.Cempaka...), and Sekolah Asal (SDN 17...). The right column includes: Email (ednan@xxxx.com), Nomor Handphone (08482xxxx...), Pendidikan Terakhir (dropdown menu), Mengajar Pada Jenjang (dropdown menu), Mengajar Pada Mata Pelajaran (dropdown menu), Pengalaman Mengajar (text area), Pengalaman Pelatihan (text area), Upload Berkas Kelengkapan (text input with Browse... button), and Upload Foto (text input with Browse... button). At the bottom of the form are "Daftar" and "Reset" buttons. The footer contains "Bantuan" and "Tentang kami" sections with contact information and social media links.

Gambar 6.1 Implementasi Antarmuka Halaman Daftar

Halaman pendaftaran adalah halaman yang pertama kali diakses oleh pengguna saat mengakses sistem. Halaman ini berfungsi untuk pengguna dapat melakukan pendaftaran dengan melengkapi data diri yang didapatkan dari aplikasi DADODIK. Implementasi antarmuka halaman daftar digambarkan pada Gambar 6.1.

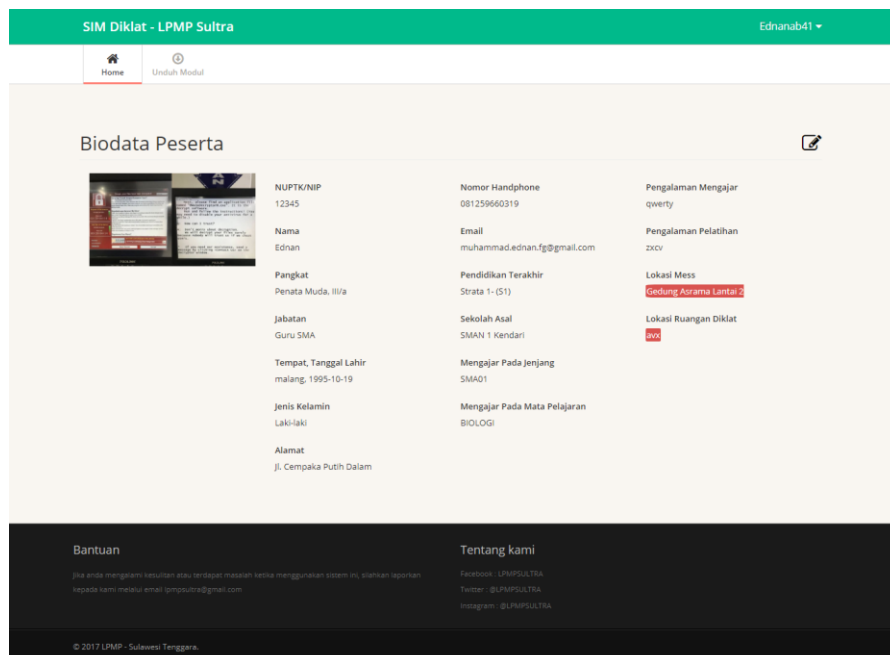
6.3.2 Halaman Login



Gambar 6.2 Implementasi Antarmuka Halaman Login

Halaman login digunakan oleh pengguna untuk dapat masuk ke dalam sistem. Setelah pengguna masuk ke dalam sistem, pengguna kemudian menjadi peserta dan dapat melakukan beberapa proses selanjutnya seperti, melihat data peserta, mengubah data peserta, melihat lokasi ruang DIKLAT, dan melihat lokasi *mess*. Implementasi antarmuka halaman ini digambarkan pada Gambar 6.2.

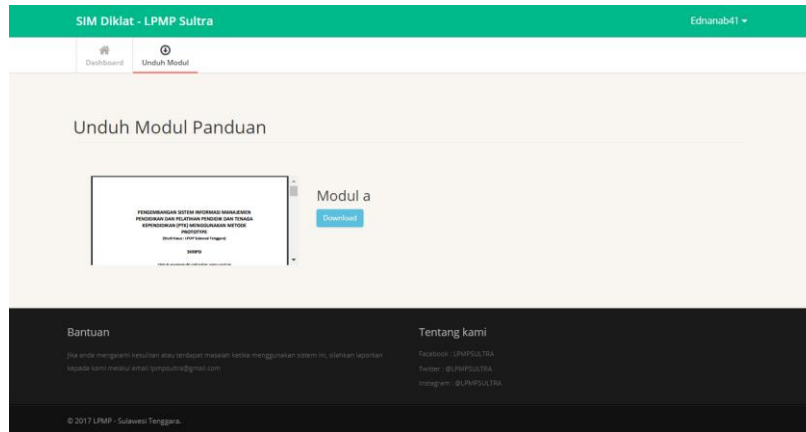
6.3.3 Halaman Home Peserta



Gambar 6.3 Implementasi Antarmuka Halaman Home Peserta

Halaman home peserta adalah halaman yang pertama kali diakses peserta ketika telah melakukan login. Halaman tersebut menampilkan data peserta, foto peserta dan tombol untuk mengubah data peserta. Selain data peserta, halaman ini juga menampilkan lokasi ruangan diklat peserta, dan lokasi *mess* peserta yang didapatkan dari pengolahan data peserta pada proses pendaftaran. Implementasi antarmuka halaman ini digambarkan pada Gambar 6.3.

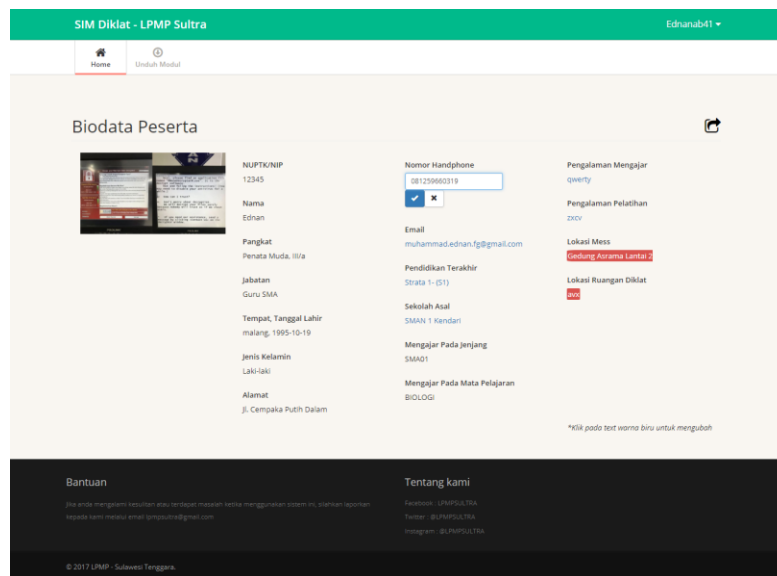
6.3.4 Halaman Unduh Modul Panduan



Gambar 6.4 Implementasi Antarmuka Halaman Unduh Modul Panduan

Halaman unduh modul panduan adalah halaman yang menampilkan *preview* dari kumpulan modul-modul panduan yang dapat di unduh oleh peserta. Dalam halaman tersebut terdapat tiga bagian utama, yaitu *preview* modul, nama modul yang akan diunduh, dan tombol *download* untuk mengunduh modul panduan. Perancangan antarmuka halaman ini digambarkan pada Gambar 5.12.

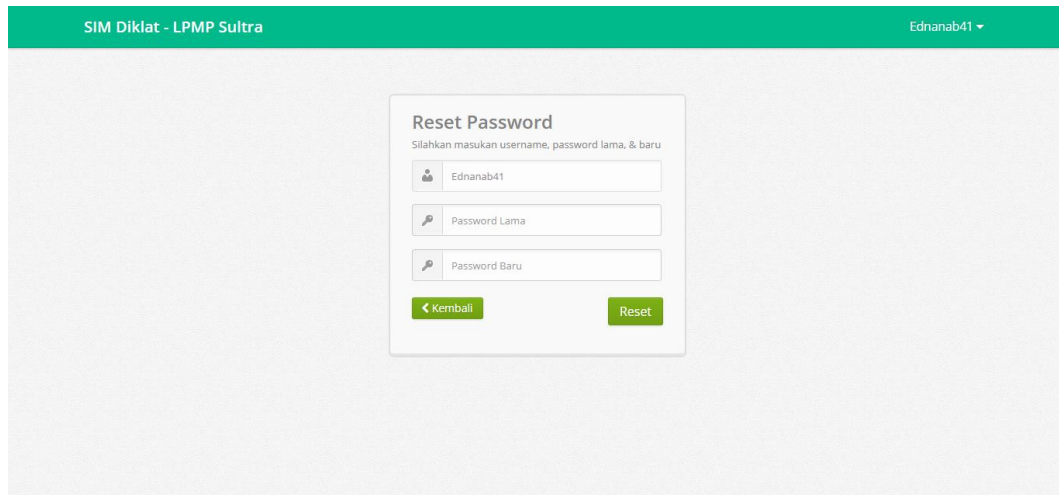
6.3.5 Halaman Edit Data Peserta



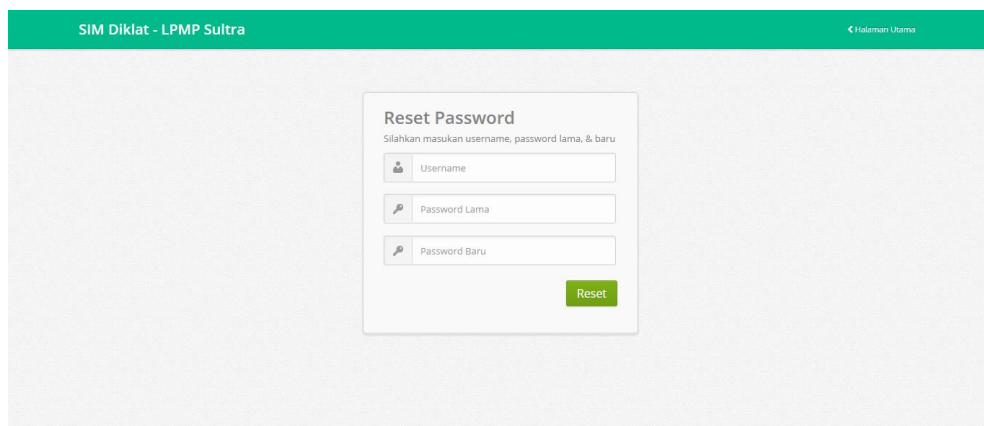
Gambar 6.5 Implementasi Antarmuka Halaman Edit Data Peserta

Halaman edit data peserta adalah halaman yang digunakan peserta untuk mengubah data yang telah dimasukkan dalam proses pendaftaran. Halaman ini disediakan oleh sistem untuk mencegah terjadinya kesalahan data peserta yang dimasukkan dalam proses pendaftaran. Data peserta yang dapat diubah adalah nomor *handphone*, *email*, pendidikan terakhir, sekolah asal, pengalaman mengajar, dan pengalaman pelatihan. Implementasi antarmuka pada halaman ini dijelaskan pada Gambar 6.5.

6.3.6 Halaman Reset Password



Gambar 6.6 Implementasi Antarmuka *Reset Password*



Gambar 6.7 Implementasi Antarmuka *Reset Password* (lanjutan)

Halaman *reset password* digunakan oleh peserta untuk mengubah *password* yang didapatkan oleh sistem melalui *email*. *Password* dibuat oleh sistem secara acak sehingga sulit untuk diingat oleh peserta, sehingga halaman ini dibuat untuk memudahkan peserta mengingat *password* yang mereka buat sendiri. Halaman ini terbagi dua kondisi, pertama adalah pengguna telah melakukan login, sehingga *username* yang diminta tidak perlu dimasukkan lagi oleh pengguna, dan yang kedua adalah pengguna belum melakukan login, dimana *username* pengguna masih kosong dan wajib untuk diisi. Implementasi antarmuka pada halaman ini digambarkan pada Gambar 6.6 dan Gambar 6.7.