# BAB IV

# IMPLEMENTASI

Implementasi sistem merupakan tahapan setelah dilakukan analisa dan perancangan yang sudah dibuat. Dalam tahap implementasi sistem ini, akan membahas hal-hal yang berkaitan secara langsung dengan perangkat lunak.

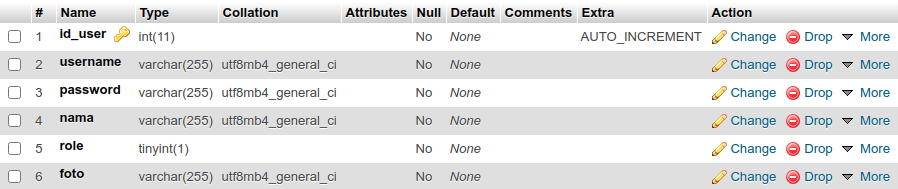
* 1. **Kebutuhan *Hardware* dan *Software***

1. *Hardware* yang diperlukan untuk menjalankan program, yaitu : *Personal Computer :*
2. Processor Intel Core i3@2.27GHz
3. RAM 2.00 GB
4. Mouse
5. Keyboard
6. Printer
7. *Software* yang diperlukan untuk menjalankan program yaitu :
8. Sistem Operasi *Windows* 7
9. Bahasa Pemrograman PHP
10. *FrameworkCI*
11. *Database MySQL*
12. *Rational Rose Enterprise*
    1. ***Database***

*Database* yang digunakan penulis untuk menyimpan data adalah *MySQL*. *Database* diberi nama “elearning” yang berisi beberapa tabeldidalamnya, diantaranya tabel admin, tabelsiswa, tabel guru, tabel mapel, tabel materi, tabel soal, tabel ujian.

1. Tabel User

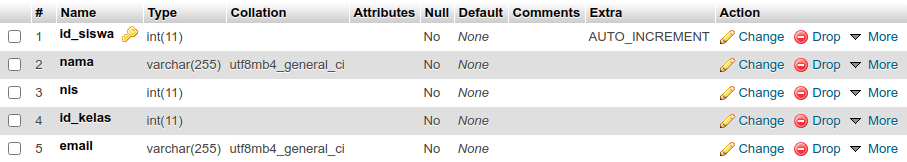
Pada Gambar 4.1 mencatat struktur database dari tabel user, yang digunakan untuk mengelola akses login dari admin, siswa, dan guru.



Gambar 4.1. Implementasi Tabel Admin

1. Tabel Siswa

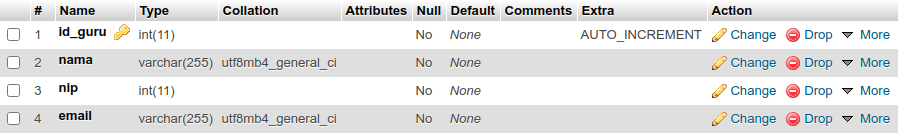
Pada Gambar 4.2 mencatat struktur database dari tabel siswa.



Gambar 4.2. Implementasi Tabel Siswa

1. Tabel Guru

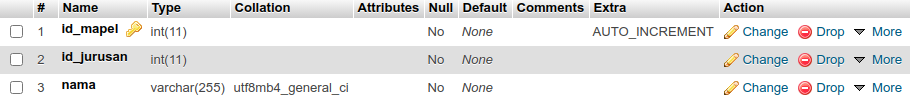
Pada Gambar 4.3 mencatat struktur database dari tabel guru.



Gambar 4.3. Implementasi Tabel Guru

1. Tabel Mapel

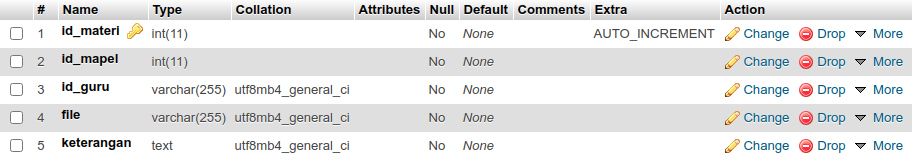
Pada Gambar 4.4 mencatat struktur database dari tabel mapel.



Gambar 4.4. Implementasi Tabel Mapel

1. Tabel Materi

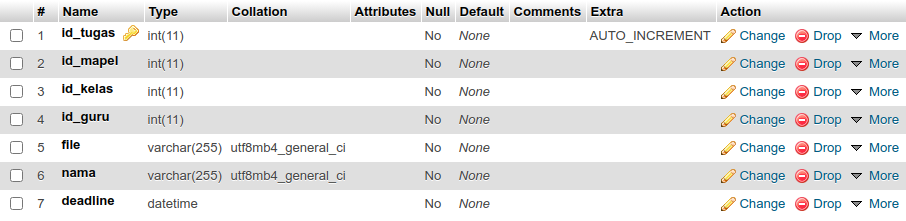
Pada Gambar 4.5 mencatat struktur database dari tabel materi.



Gambar 4.5. Implementasi Tabel Materi

1. Tabel Tugas

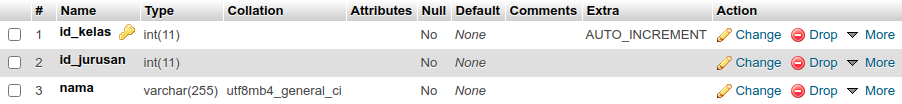
Pada Gambar 4.6 mencatat struktur database dari tabel tugas.



Gambar 4.6. Implementasi Tabel Tugas

1. Tabel Kelas

Pada Gambar 4.7 mencatat struktur database dari tabel kelas.



Gambar 4.7. Implementasi Tabel Kelas

1. Tabel Jurusan

Pada Gambar 4.8 mencatat struktur database dari tabel jurusan.

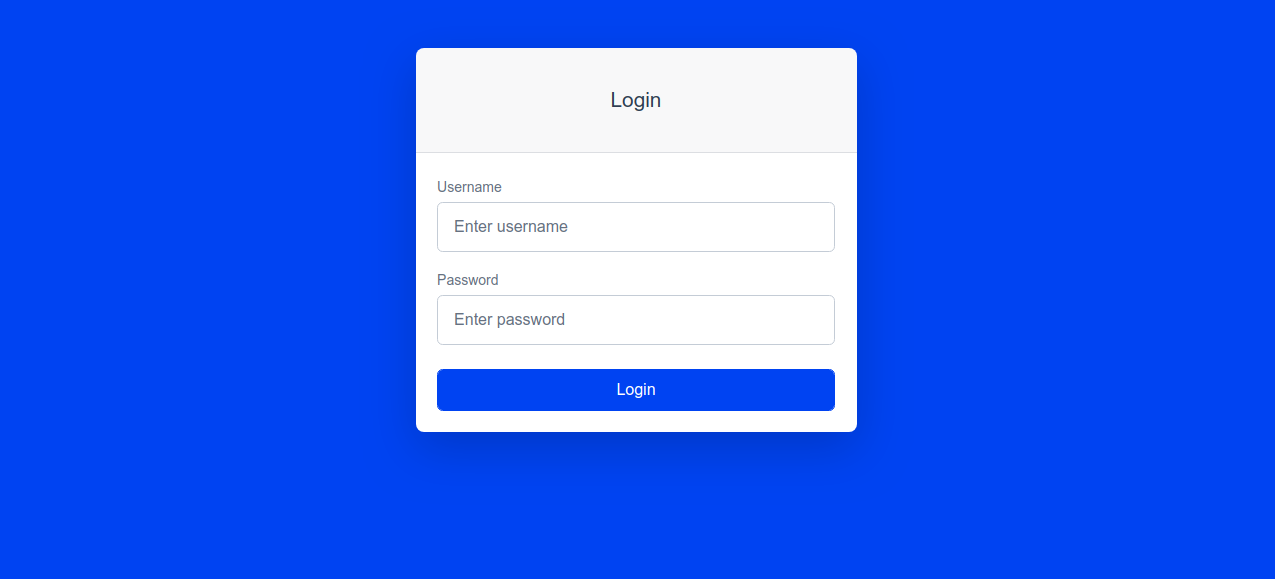


Gambar 4.8. Implementasi Tabel Jurusan

* 1. **Implementasi Program**

1. Tampilan Login Admin

Tampilan login admin berisi *form input username* dan *password* untuk masuk kedalam sistem tersebut. Untuk tampilan pengguna bisa dilihat di gambar 4.9.



Gambar 4.9. Implementasi Tampilan Login Admin

1. Tampilan Menu Utama Admin

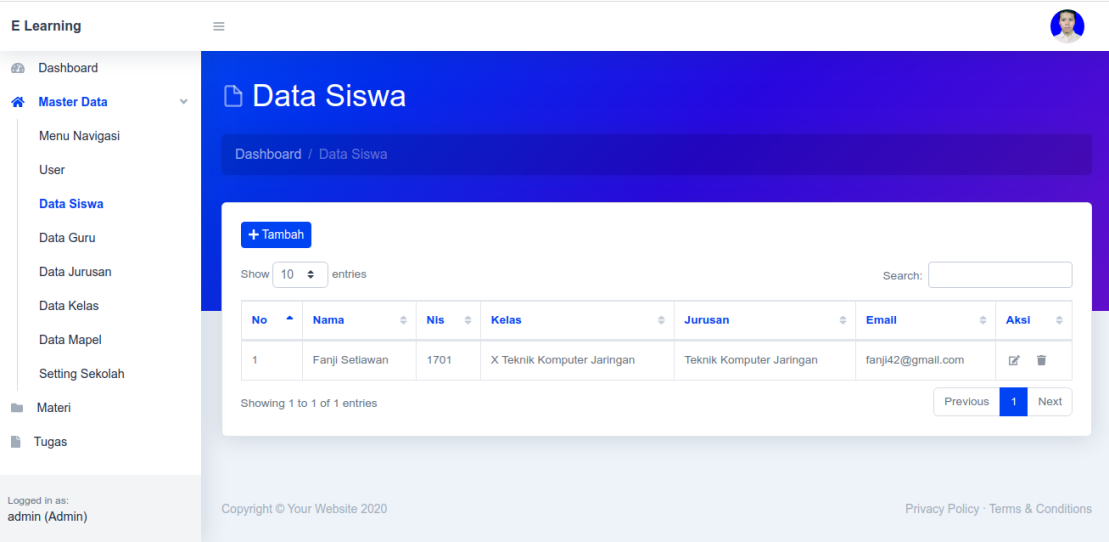
Tampilan menu utama admin berisi menu yang dapat diakses oleh admin, yaitu : Data Siswa, Data Guru, Data Mapel, Data Materi, Data Soal, Data Hasil Ujian, dan Setting Email. Untuk tampilan pengguna bisa dilihat di gambar 4.10.



Gambar 4.10. Implementasi Tampilan Menu Utama Admin

1. Tampilan Mengelola Data Siswa

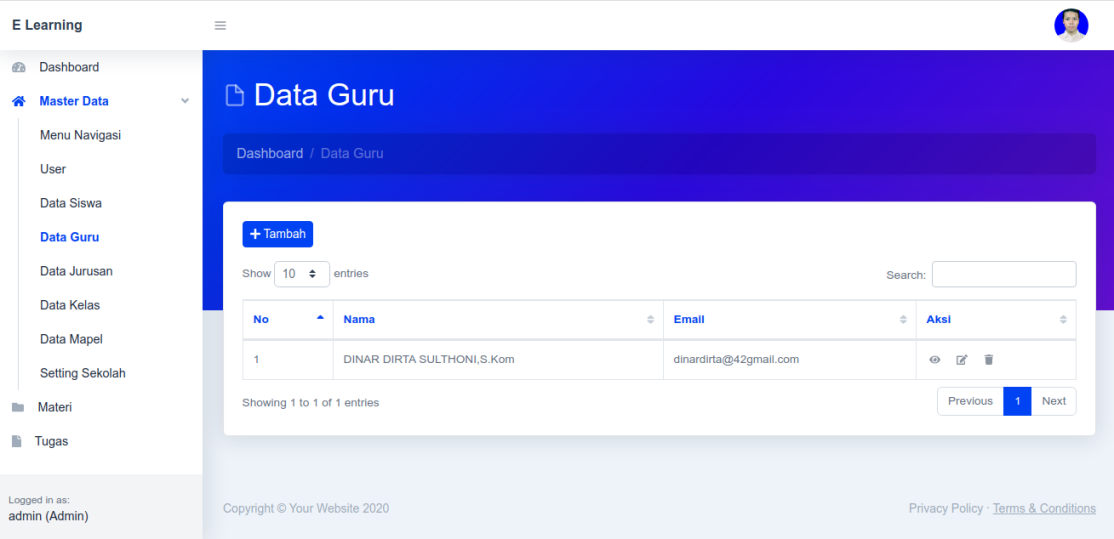
Tampilan Data Siswa pada admin digunakan untuk mengelola data siswa. Pada halaman Data Siswa terdapat fungsi tambah, edit, hapus, dan mata pelajaran. Fungsi tambah digunakan untuk menambahkan data siswa. Fungsi edit digunakan untuk mengupdate atau memperbarui data siswa yang sudah ada. Fungsi hapus digunakan untuk menghapus data siswa. Fungsi mata pelajaran digunakan untuk memilih kelas mata pelajaran yang dibutuhkan. Untuk tampilan pengguna bisa dilihat di gambar 4.11.



Gambar 4.11. Implementasi Tampilan Data Siswa

1. Tampilan Mengelola Data Guru

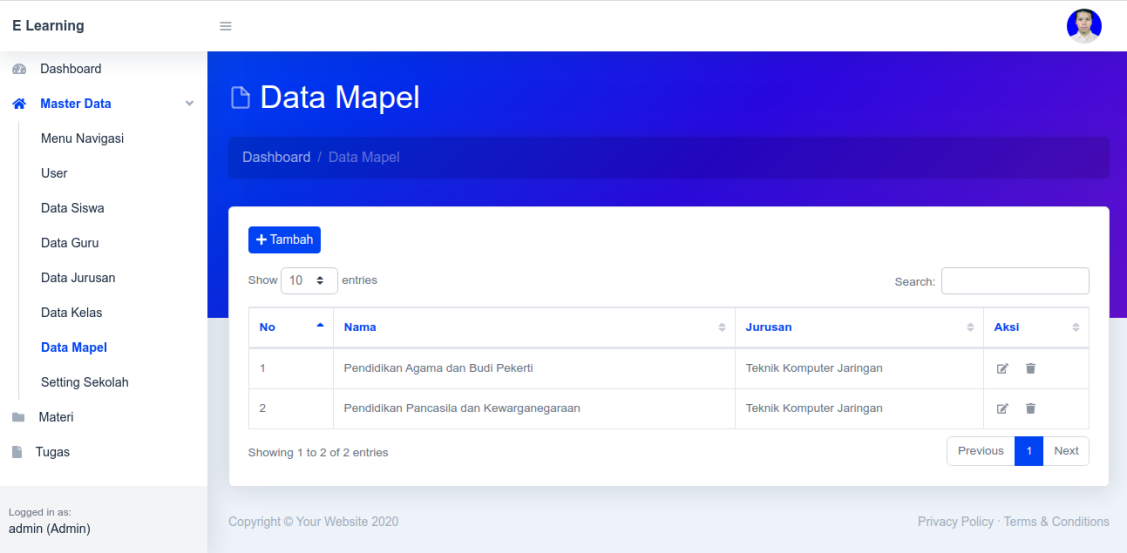
Tampilan Data Guru pada admin digunakan untuk mengelola data guru. Pada halaman Data Guru terdapat fungsi tambah, edit, hapus, dan mata pelajaran. Fungsi tambah digunakan untuk menambahkan data guru. Fungsi edit digunakan untuk mengupdate atau memperbarui data guru yang sudah ada. Fungsi hapus digunakan untuk menghapus data guru. Fungsi mata pelajaran digunakan untuk memilih kelas mata pelajaran yang dibutuhkan. Untuk tampilan pengguna bisa dilihat di gambar 4.12.



Gambar 4.12. Implementasi Tampilan Data Guru

1. Tampilan Mengelola Data Mapel

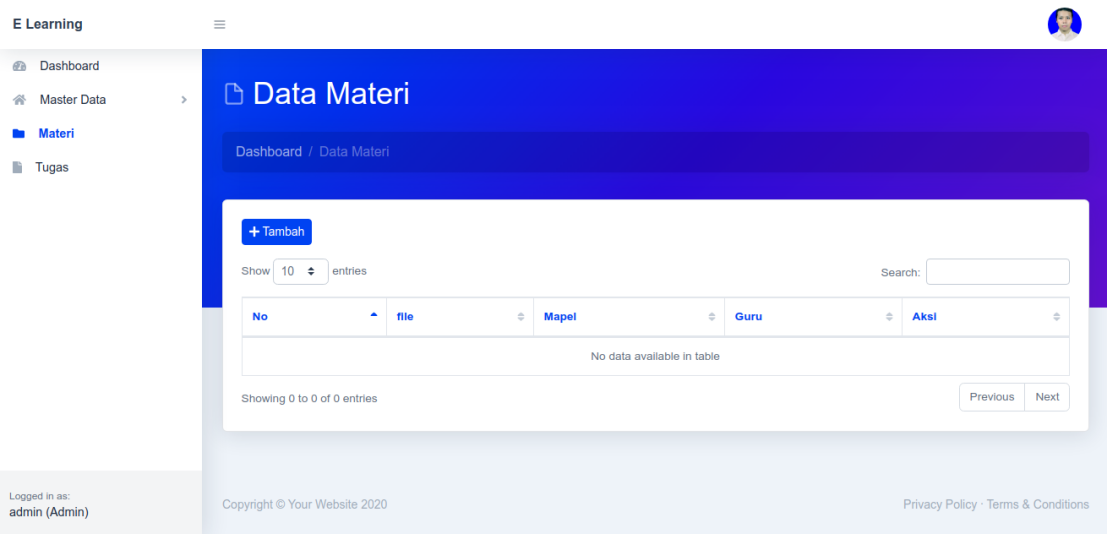
Tampilan Data Mapel pada admin digunakan untuk mengelola data mata pelajaran. Pada halaman Data Mapel terdapat fungsi tambah, edit, dan hapus. Fungsi tambah digunakan untuk menambahkan data mata pelajaran. Fungsi edit digunakan untuk mengupdate atau memperbarui data mata pelajaran yang sudah ada. Fungsi hapus digunakan untuk menghapus data mata pelajaran. Untuk tampilan pengguna bisa dilihat di gambar 4.13.



Gambar 4.13. Implementasi Tampilan Data Mapel

1. Tampilan Mengelola Data Materi

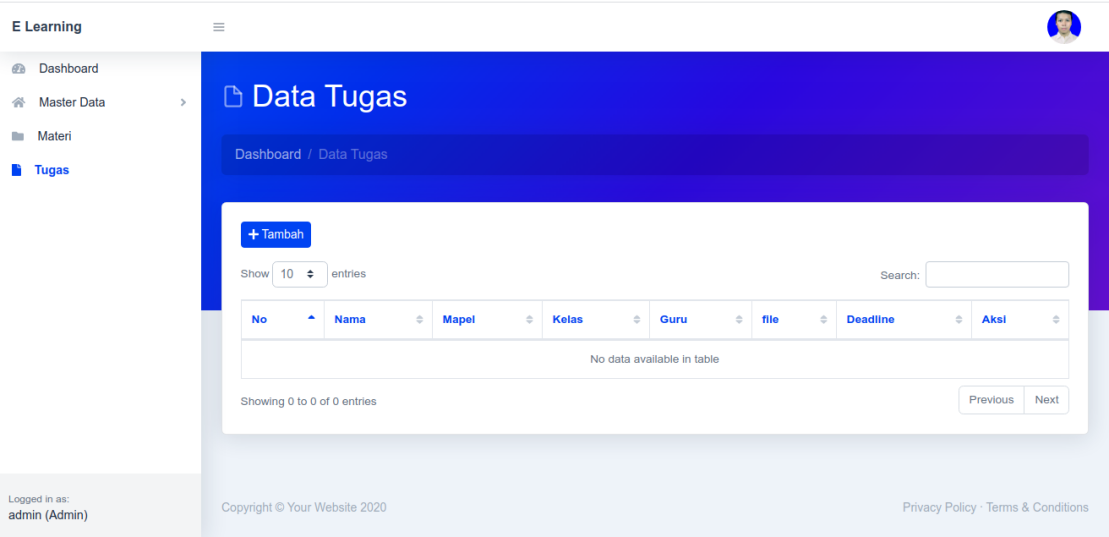
Tampilan Data Materi pada admin digunakan untuk mengelola data mata materi. Pada halaman Data Materi terdapat fungsi tambah, edit, dan hapus. Fungsi tambah digunakan untuk menambahkan data mata pelajaran. Fungsi edit digunakan untuk mengupdate atau memperbarui data materi yang sudah ada. Fungsi hapus digunakan untuk menghapus data materi. Untuk tampilan pengguna bisa dilihat di gambar 4.14.



Gambar 4.14. Implementasi Tampilan Data Materi

1. Tampilan Mengelola Data Tugas

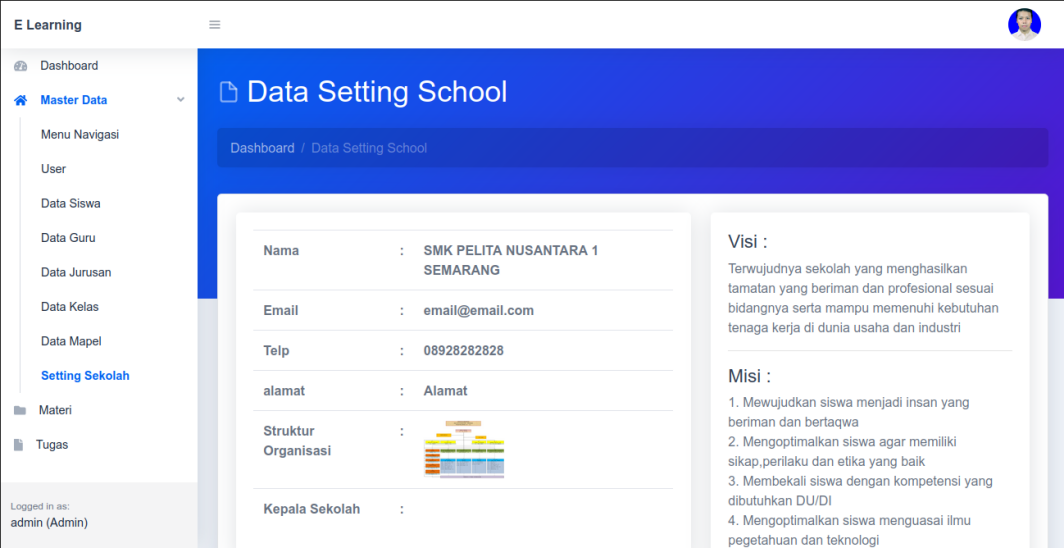
Tampilan Tugas pada admin digunakan untuk mengelola data tugas. Pada halaman Tugas terdapat fungsi tambah, edit, dan hapus, dan cetak. Fungsi tambah digunakan untuk menambahkan data tugas. Fungsi edit digunakan untuk mengupdate atau memperbarui data mata tugas yang sudah ada. Fungsi hapus digunakan untuk menghapus data mata tugas. Fungsi cetak digunakan untuk mencetak hasil laporan dari data tugas disesuaikan dengan mata pelajaran yang dipilih. Untuk tampilan pengguna bisa dilihat di gambar 4.15.



Gambar 4.15. Implementasi Tampilan Data Soal

1. Tampilan Setting Sekolah

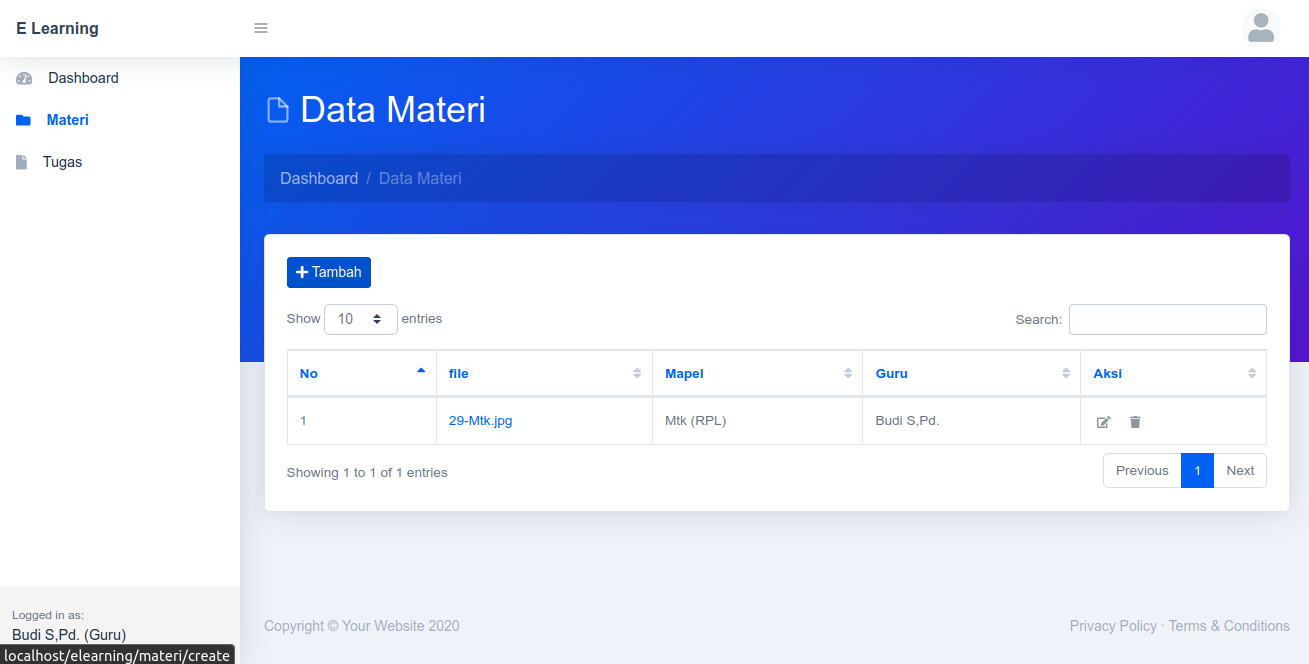
Tampilan Setting Sekolah pada admin digunakan untuk mengisi data-data identitas sekolah dan juga sekaligus untuk mensetting email yang digunaakan untuk notifikasi ke siswa. Untuk tampilan pengguna bisa dilihat di gambar 4.16.



Gambar 4.16. Implementasi Tampilan Setting Email

1. Tampilan Mengelola Data Materi

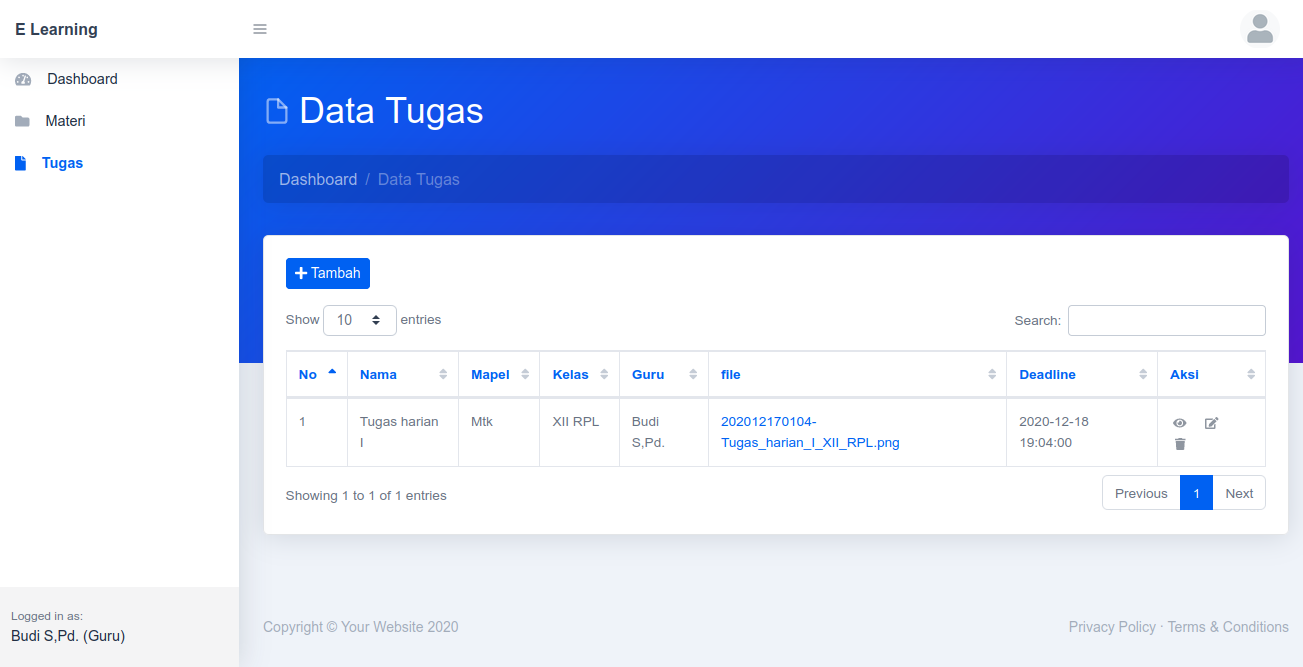
Tampilan Data Materi pada guru digunakan untuk mengelola data mata materi. Pada halaman Data Materi terdapat fungsi tambah, edit, dan hapus. Fungsi tambah digunakan untuk menambahkan data mata pelajaran. Fungsi edit digunakan untuk mengupdate atau memperbarui data materi yang sudah ada. Fungsi hapus digunakan untuk menghapus data materi. Untuk tampilan pengguna bisa dilihat di gambar 4.17.



Gambar 4.17. Implementasi Tampilan Data Materi

1. Tampilan Mengelola Data Tugas

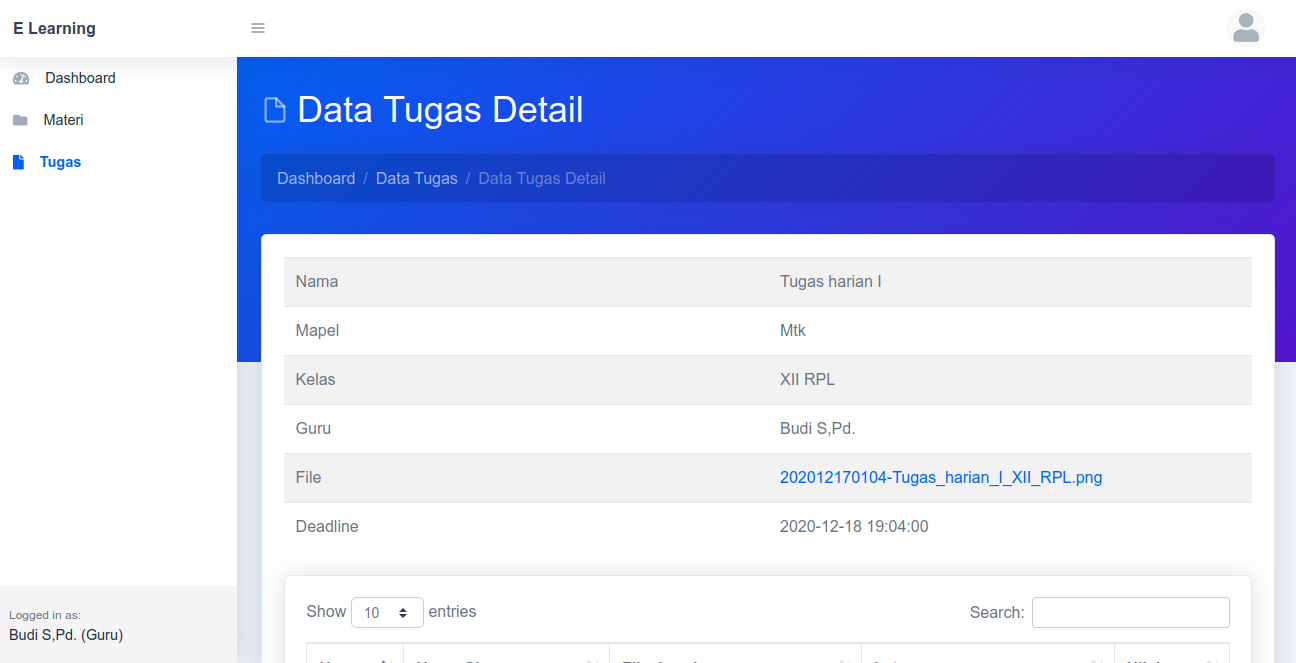
Tampilan tugas pada guru digunakan untuk mengelola data tugas. Pada halaman tugas terdapat fungsi tambah, edit, dan hapus, dan cetak. Fungsi tambah digunakan untuk menambahkan data tugas. Fungsi edit digunakan untuk mengupdate atau memperbarui data mata tugas yang sudah ada. Fungsi hapus digunakan untuk menghapus data mata tugas. Fungsi cetak digunakan untuk mencetak hasil laporan dari data tugas disesuaikan dengan mata pelajaran yang dipilih. Untuk tampilan pengguna bisa dilihat di gambar 4.18.



Gambar 4.18. Implementasi Tampilan Data Tugas

1. Tampilan Mengelola Hasil Tugas

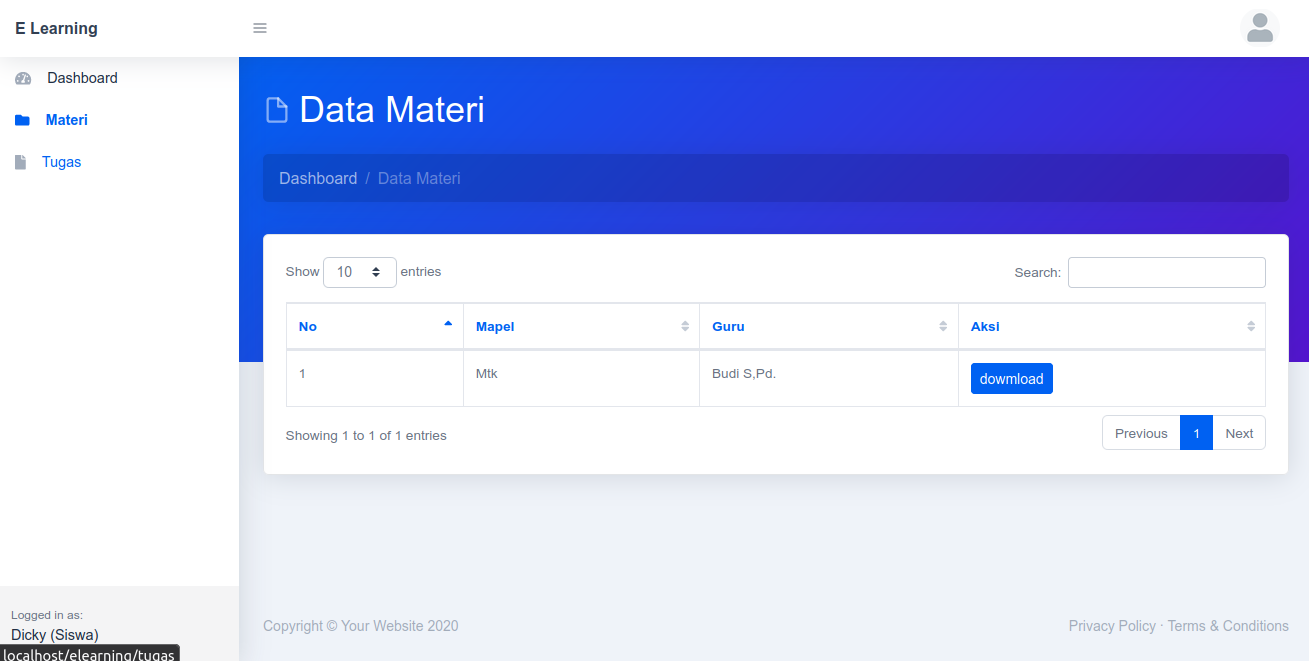
Tampilan menu hasil tugas pada guru digunakan untuk mengelola data hasil tugas. Pada halaman Hasil Tugas terdapat fungsi untuk menambahkan nilai. Fungsi menambahkan nilai ini digunakan untuk mencetak memberikan nilai kepada siswa yang sudah mengerjakan. Untuk tampilan pengguna bisa dilihat di gambar 4.19.



Gambar 4.19. Implementasi Tampilan Hasil Ujian

1. Tampilan Data Materi

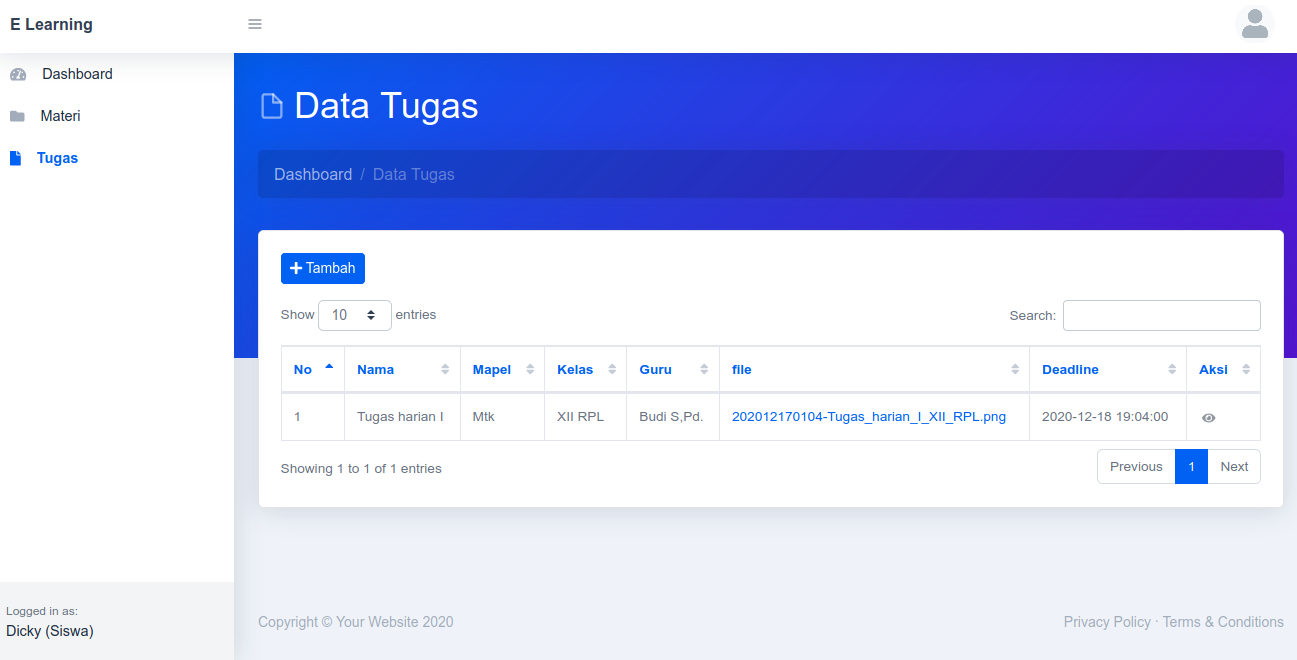
Tampilan Data Materi pada siswa digunakan untuk mengelola data mata materi. Pada halaman Data Materi terdapat fungsi download. Fungsi download digunakan untuk mengunduh materi dari guru. Untuk tampilan pengguna bisa dilihat di gambar 4.20.



Gambar 4.20. Implementasi Tampilan Data Materi

1. Tampilan Tugas

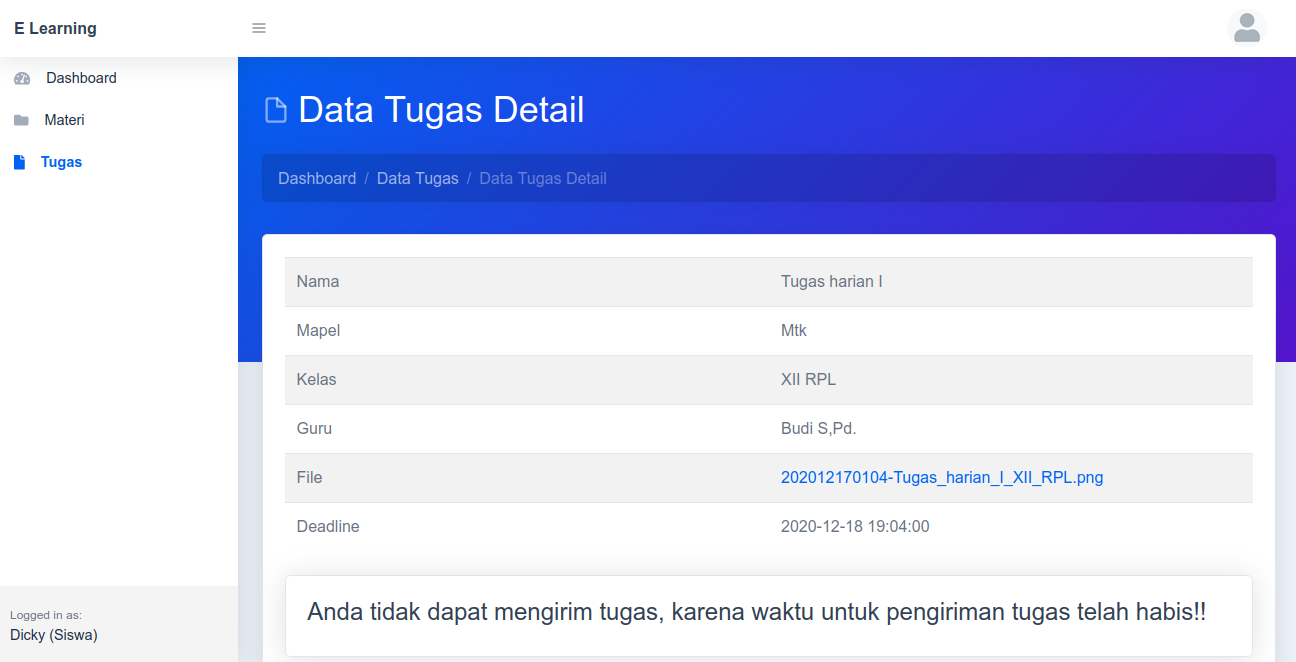
Tampilan tugas pada siswa berfungsi untuk mengerjakan tugas yang diberikan oleh guru. Siswa dapat mendownload tugas yang diberikan guru dengan mengklik link pada kolom file. Untuk tampilan pengguna bisa dilihat di gambar 4.21.



Gambar 4.21. Implementasi Tampilan Ujian

1. Tampilan Submit Tugas

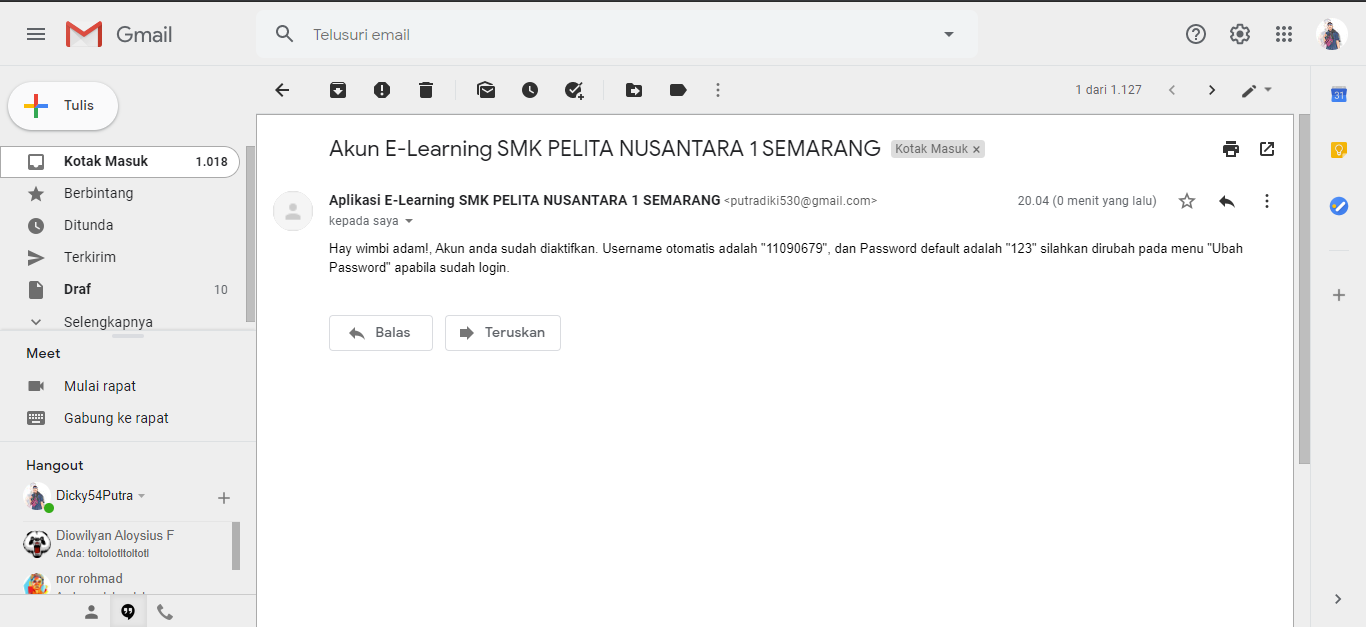
Tampilan submit tugas digunakan kepada siswa untuk mengumpulkan kerjaan yang sudah mereka kerjakan. Untuk tampilan pengguna bisa dilihat di gambar 4.22.



Gambar 4.22. Implementasi Tampilan Submit Tugas

1. Tampilan Halaman Notifikasi Email

Tampilan halaman notifikasi setelah admin mengaktifkan user siswa. Untuk tampilan pengguna bisa dilihat di gambar 4.23.



Gambar 4.23. Implementasi Tampilan Notifikasi Email

* 1. **Pengujian Sistem**

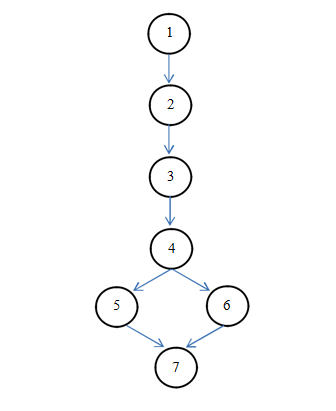
Penulis melakukan pengujian sistem setelah melakukan implementasi sistem. Metode pengujian yang dilakukan yaitu *whitebox* dan *blackbox*.*White-Box Testing* yaitu salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak. Jika model yang sudah dihasilkan berupa *output* yang tidak sesuai dengan yang diharapkan maka akan dikompilasi ulang dan dicek kembali kode-kode tersebut hingga sesuai dengan yang diharapkan. *Black-Box Testing* yaitu pengujian yang terfokus pada spesifikasi fungsional dari *software,* penguji dapat mendefinisikan sekumpulan kondisi input danmelakukan pengetesan pada spesifikasi fungsional program.

* + 1. **Pengujian** ***White-Box***

1. ListingProgram Siswa

|  |  |
| --- | --- |
| **MODEL**  public function m\_siswa()  {  $this->cek\_aktif();  //var def session  $a['sess\_level'] = $this->session->userdata('admin\_level');  $a['sess\_user'] = $this->session->userdata('admin\_user');  $a['sess\_konid'] = $this->session->userdata('admin\_konid');  //var def uri segment  $uri2 = mysql\_real\_escape\_string($this->uri->segment(2));  $uri3 = mysql\_real\_escape\_string($this->uri->segment(3));  $uri4 = mysql\_real\_escape\_string($this->uri->segment(4));  //var post from json  $p = json\_decode(file\_get\_contents('php://input'));  //return as json  2  $jeson = array(); | 1 |
| **CONTROLLER**  $a['data'] = $this->db->query("SELECT m\_siswa.\*,  (SELECT COUNT(id) FROM m\_admin WHERE level = 'siswa' AND kon\_id = m\_siswa.id) AS ada  FROM m\_siswa")->result();  if ($uri3 == "det") {  $a = $this->db->query("SELECT \* FROM m\_siswa WHERE id = '$uri4'")->row();  $this->j($a);  exit();  } else if ($uri3 == "simpan") {  $ket = "";  if ($p->id != 0) {  $this->db->query("UPDATE m\_siswa SET nama = '" . bersih($p, "nama") . "', nis = '" . bersih($p, "nis") . "', kelas = '" . bersih($p, "kelas") . "', email = '" . bersih($p, "email") . "' WHERE id = '" . bersih($p, "id") . "'");  $ket = "edit";  } else {  $ket = "tambah";  // var\_dump(bersih($p, "email"));  // die;  $this->db->query("INSERT INTO m\_siswa VALUES (null,  '" . bersih($p, "nama") . "',  '" . bersih($p, "nis") . "',  '" . bersih($p, "kelas") . "',  '" . bersih($p, "email") . "')");  }  $ret\_arr['status'] = "ok";  $ret\_arr['caption'] = $ket . " sukses";  $this->j($ret\_arr);  exit();  } else if ($uri3 == "hapus") {  $this->db->query("DELETE FROM m\_siswa WHERE id = '" . $uri4 . "'");  $this->db->query("DELETE FROM m\_admin WHERE level = 'siswa' AND kon\_id = '" . $uri4 . "'");  $ret\_arr['status'] = "ok";  $ret\_arr['caption'] = "hapus sukses";  $this->j($ret\_arr);  exit();  } else if ($uri3 == "user") {  6  $det\_user = $this->db->query("SELECT id, nis, email, nama FROM m\_siswa WHERE id = '$uri4'")->row();  $admin = $this->db->query("SELECT email, pass FROM m\_admin WHERE id = 1")->row();  // echo $admin->email;  // echo ' ' .$admin->pass;  // exit();  $data = array(  'username' => $det\_user->nis,  'password' => md5('123'),  'level' => 'siswa',  'kon\_id' => $det\_user->id  );  // var\_dump($data);  // die;  $this->db->insert('m\_admin', $data);  $ret\_arr['status'] = "ok";  $ret\_arr['caption'] = "tambah user sukses";  exit();  } | 5  4  3 |
| **Views**  $this->db->insert('m\_admin', $data);  $ret\_arr['status'] = "ok";  $ret\_arr['caption'] = "tambah user sukses";  $this->j($ret\_arr); | 7 |

1. *Flowgraph* Siswa



Gambar 4.22. *Flowgraph* Siswa

1. Kompleksitas Siklomatis

Pengukuran kuantitatif kompleksitaslogissuatu program pada *flowgraph* dapat diperoleh dengan perhitungan sebagai berikut:V(G)=E – N +2

Keterangan : E =Jumlah *Edge* dalam *flowgraph*

N =Jumlah *Node* dalam *flowgraph*

Sehingga kompleksitas siklomatis yang diperoleh adalah:

V(G)=E– N +2

V(G)=7-7 +2=2

1. Basis Set

BasisSetyangdihasilkan adalah 2 jalur,yaitu:

* 1. 1 – 2 – 3 – 4– 5 – 7 (data tidak valid, alamat belum diisi)
  2. 1 – 2 – 3 – 4– 6 – 7 (data valid, simpan)
     1. **Pengujian** ***Black-Box***

*Black-Box Testing* yaitu pengujian yang terfokus pada spesifikasifungsional dari *software,* penguji dapat mendefinisikan sekumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

1. Admin

Pada Tabel 4.1 mencatat skenario pengujian *black-box* admin.

Tabel 4.1. Pengujian *Black-Box* Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Uji Fungsi** | **Langkah Uji** | **Hasil Yang Diharapkan** | **Hasil Aktual** |
| 1. | Halaman Login Admin | Maskan username dan password untuk admin yang benar | Menampilkan halaman menu utama admin | berhasil |
| 2. | Halaman Menu Utrama Admin | Menekan menu dashboard admin | Menampilkan menu data siswa, data guru, data mapel, data materi, soal, hasil ujian, setting email | berhasil |
| 3. | Halaman Data Siswa Admin | Menekan menu data siswa admin | Menampilkan halaman data siswa admin | berhasil |
| 4. | Halaman Data Guru Admin | Menekan menu data guru admin | Menampilkan halaman data guru admin | berhasil |
| 5. | Halaman Data Mapel Admin | Menekan menu data mapel admin | Menampilkan halaman date mapel admin | berhasil |
| 6. | Halaman Data Materi Admin | Menekan menu data materi admin | Menampilkan halaman data materi admin | berhasil |
| 7. | Halaman Tugas Admin | Menekan menu tugas admin | Menampilkan halaman tugas admin | berhasil |
| 8. | Halaman Setting Emasil Admin | Menekan menu setting email admin | Menampilkan halaman setting email admin | berhasil |

1. Guru

Pada Tabel 4.2 mencatat skenario pengujian *black-box* guru.

Tabel 4.2. Pengujian *Black-Box* Guru

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Uji Fungsi** | **Langkah Uji** | **Hasil Yang Diharapkan** | **Hasil Aktual** |
| 1. | Halaman Login Guru | Maskan username dan password untuk guru yang benar | Menampilkan halaman menu utama guru | berhasil |
| 2. | Halaman Menu Utama Guru | Menekan menu dashboard siswa | Menampiplkan menu data materi, soal, ujian, hasil ujian. | berhasil |
| 3. | Halaman Data Materi | Menekan menu data materi | Menampilkan halaman data materi | berhasil |
| 4. | Halaman Tugas | Menekan menu tugas | Menampilkan menu halaman tugas | berhasil |

1. Siswa

Pada Tabel 4.3 mencatat skenario pengujian *black-box* siswa.

Tabel 4.3. Pengujian *Black-Box* Siswa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Uji Fungsi** | **Langkah Uji** | **Hasil Yang Diharapkan** | **Hasil Aktual** |
| 1. | Halaman Login Siswa | Maskan username dan password untuk siswa yang benar | Menampilkan halaman menu utama siswa | berhasil |
| 2. | Halaman Menu Utama Siswa | Menekan menu dashboard siswa | Menampilkan menu data materi, download materi | berhasil |
| 3. | Halaman Data Materi | Menekan menu data materi, download materi | Menampilkan halaman data meteri, materi berhasil didownload | berhasil |
| 4. | Halaman Tugas | Menekan menu tugas | Menampilkan halaman tugas | berhasil |

* 1. **Pemeliharaan Sistem**

*Maintenance* atau pemeliharaan sistem yaitu suatu upaya untukmenjaga, memperbaiki, mengembangan sistem yang ada. Berikut ini beberapa *maintenance* yang ada pada aplikasi e-learning.

1. Pengecekan kebenaran dan kecocokan database sistem terhadap data yang ditampilkan.
2. Perbaikan tampilan sistem pada web agar pengguna lebih mudah lagi untuk mengaplikasikan sistem.
3. Pengecekan keamanan program