# BAB V

# IMPLEMENTASI DAN PENGUJIAN SISTEM

## 5.1 Implementasi

Implementasi adalah proses penerapan rancangan program yang telah dibuat pada bab sebelumnya atau aplikasi dalam melaksanakan sistem informasi pemrograman yang telah dibuat, hasil dari tahapan implementasi ini adalah suatu sistem pengolahan data yang sudah dapat berjalan dengan baik. Dengan demikian dapat diketahui apakah perangkat lunak Pengembangan Sistem Informasi Skripsi dan Tugas Akhir Universitas Komputer Indonesia sesuai dengan tujuan yang diharapkan. Implementasi adalah proses penerapan rancangan program yang telah dibuat pada bab sebelumnya atau aplikasi dalam melaksanakan sistem informasi pemrograman yang telah dibuat, hasil dari tahapan implementasi ini adalah suatu sistem pengolahan data yang sudah dapat berjalan dengan baik. Dengan demikian dapat diketahui apakah perangkat lunak ini dapat menghasilkan Pengembangan Sistem Informasi Skripsi dan Tugas Akhir Universitas Komputer Indonesia sesuai dengan tujuan yang diharapkan.

### 5.1.1 Batasan Implementasi (optional)

Dalam rencana penerapan sistem yang terkomputerisi, yakni agar sistem siap dioperasikan, maka perlu diadakan kegiatan-kegiatan dari penerapannya. Adapun langkah-langkah yang perlu dilakukan untuk menerapkan sistem tersebut adalah pembuatan program, testing program, pelatihan serta terima dokumentasi, akan tetapi dalam penulisan tugas akhir ini

implementasi sistem yang dilakukan hanya sampai pada tahap pembuatan program

### 5.1.2 Implementasi Perangkat Lunak

Untuk mendukung kelancaran sistem informasi yang dirancang, maka sistem ini memerlukan perangkat lunak. Perangkat lunak digunakan untuk mendukung kinerja sistem operasi dan aplikasi database. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

1. Operating System : Microsoft Windows XP,Vista,7,8
2. Browser : Internet Explore, Mozilla Firefox, Opera, Google Chrome
3. Web Server : XAMPP versi > 1.4
4. Adobe Reader untuk mencetak

### 5.1.3 Implementasi Perangkat Keras

Untuk mendukung terlaksananya penerapan aplikasi PHP yang dirancang, maka sistem pengoperasian ini membutuhkan komputer sebagai server yang memudahkan dalam menjalankan program aplikasi dengan spesifikasi sebagai berikut :

**Server :**

a. Menggunakan minimal processor Intel Pentium dual core 2,8 GHz atau sekelasnya.
b. Menggunakan RAM minimal 1 Gb.
c. Tersedianya Hard Drive sebagai media penyimpanan, minimal 100 Gb
d. Monitor, Mouse, Keyboard sebagai peralatan antar muka.

**Client:**

a. Menggunakan minimal processor Intel Pentium Dual Core IV 2.0 GHz atau sekelasnya.
b. Menggunakan RAM minimal 1 Gb.
c. Tersedianya Hard Drive sebagai media penyimpanan, minimal 100 Gb
d. Monitor, Mouse, Keyboard sebagai peralatan antar muka.
e. Printer

**Konfigurasi jaringan (Networking):**

a. Switch HUB 8 port 10/100 Mbps i buah
b. Lan Card 3 buah
c. Kabel UTP
d. Conector Rj 45

### 5.1.4 Implementasi Basis Data (Sintaks SQL)

Implementasi basis data yang digunakan memakai MySQL dan perancangan Database menggunakan PHPMyAdmin. Adapun implementasi perancangan basis data adalah sebagai berikut:

```
-- ----------------------------
-- Table structure for `gelombang`
-- ----------------------------
DROP TABLE IF EXISTS `gelombang`;
CREATE TABLE `gelombang` (
 `id` int(10) NOT NULL AUTO_INCREMENT,
 `periode_id` int(10) NOT NULL,
 `tgl_awal` datetime DEFAULT NULL,
 `tgl_akhir` datetime DEFAULT NULL,
 `status` tinyint(1) DEFAULT '0',
 PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;


-- ----------------------------
-- Table structure for `jadwal`
-- ----------------------------
DROP TABLE IF EXISTS `jadwal`;
CREATE TABLE `jadwal` (
 `id` int(10) NOT NULL AUTO_INCREMENT,
 `gelombang_id` int(10) NOT NULL,
 `daftar_id` int(10) DEFAULT NULL,
 `tgl_awal` datetime DEFAULT NULL,
 `tgl_akhir` datetime DEFAULT NULL,
 `room_id` int(10) DEFAULT NULL,
 `status` tinyint(1) NOT NULL DEFAULT '0',
 `class_name` varchar(15) NOT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=289 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `jenis_skripsi`
-- ----------------------------
DROP TABLE IF EXISTS `jenis_skripsi`;
CREATE TABLE `jenis_skripsi` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `jenis_skripsi` varchar(200) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `judul_skripsi`
-- ----------------------------
DROP TABLE IF EXISTS `judul_skripsi`;
CREATE TABLE `judul_skripsi` (
```

```
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `mahasiswa_id` int(11) NOT NULL DEFAULT '0',
 `judul_indo` varchar(200) DEFAULT NULL,
 `judul_eng` varchar(200) DEFAULT NULL,
 `des_indo` text,
 `des_eng` text,
 `jenis_skripsi_id` int(11) DEFAULT NULL,
 `tahun_ajaran_id` int(11) DEFAULT NULL,
 `dsn_pembimbing_id` int(11) DEFAULT NULL,
 `dsn_penguji_id_1` int(11) DEFAULT NULL,
 `dsn_penguji_id_2` int(11) DEFAULT NULL,
 `pending` tinyint(1) DEFAULT '0',
 PRIMARY KEY (`id`,`mahasiswa_id`)
) ENGINE=InnoDB AUTO_INCREMENT=376 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `jurusan`
-- ----------------------------
DROP TABLE IF EXISTS `jurusan`;
CREATE TABLE `jurusan` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `jurusan` varchar(200) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `mahasiswa`
-- ----------------------------
DROP TABLE IF EXISTS `mahasiswa`;
CREATE TABLE `mahasiswa` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `nim` int(8) NOT NULL,
 `nama` varchar(30) NOT NULL,
```

```
 `tempat_lahir` varchar(20) NOT NULL,
 `tanggal_lahir` date NOT NULL,
 `jenis_kelamin` char(1) NOT NULL,
 `alamat1` varchar(200) NOT NULL,
 `alamat2` varchar(200) NOT NULL,
 `email` varchar(50) NOT NULL,
 `telepon` varchar(15) NOT NULL,
 `jj` smallint(3) NOT NULL,
 `angkatan` year(4) NOT NULL,
 `sks` smallint(3) NOT NULL,
 `bapak` varchar(30) NOT NULL,
 `Ibu` varchar(50) NOT NULL,
 `alamat_ortu` varchar(200) NOT NULL,
 `telepon_ortu` varchar(15) NOT NULL,
 `pekerjaan_ortu` varchar(20) DEFAULT NULL,
 `foto` varchar(40) DEFAULT NULL,
 `wali_id` bigint(11) NOT NULL DEFAULT '0',
 `status` tinyint(1) DEFAULT '0',
 `password` varchar(200) DEFAULT NULL,
 `jurusan_id` int(11) NOT NULL,
 `access_token` varchar(300) DEFAULT NULL,
 `step` tinyint(1) DEFAULT NULL,
 `lulus` tinyint(1) NOT NULL,
 PRIMARY KEY (`id`,`nim`,`email`)
) ENGINE=MyISAM AUTO_INCREMENT=610 DEFAULT CHARSET=latin1;


-- ----------------------------
-- Table structure for `mahasiswa_persyaratan`
-- ----------------------------
DROP TABLE IF EXISTS `mahasiswa_persyaratan`;
```

```
CREATE TABLE `mahasiswa_persyaratan` (
 `id` int(50) NOT NULL AUTO_INCREMENT,
 `daftar_id` int(10) DEFAULT NULL,
 `persyaratan_id` int(10) DEFAULT NULL,
 `class_name` varchar(50) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=55 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `periode`
-- ----------------------------
DROP TABLE IF EXISTS `periode`;
CREATE TABLE `periode` (
 `id` int(10) NOT NULL AUTO_INCREMENT,
 `pendaftaran` varchar(50) NOT NULL,
 `tgl_awal` date NOT NULL,
 `tgl_akhir` date NOT NULL,
 `status` tinyint(1) DEFAULT NULL,
 `tahun_ajaran_id` int(10) NOT NULL,
 `nilai` int(11) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `persyaratan`
-- ----------------------------
DROP TABLE IF EXISTS `persyaratan`;
CREATE TABLE `persyaratan` (
 `id` int(50) NOT NULL AUTO_INCREMENT,
 `persyaratan` varchar(100) DEFAULT NULL,
 `step` int(11) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1;
```

```
-- ----------------------------
-- Table structure for `room`
-- ----------------------------
DROP TABLE IF EXISTS `room`;
CREATE TABLE `room` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `room_name` varchar(11) NOT NULL DEFAULT '0',
 `qouta` int(11) DEFAULT NULL,
 `status` tinyint(1) NOT NULL,
 `step_id` int(11) DEFAULT NULL,
 `staff_id` int(11) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `rule`
-- ----------------------------
DROP TABLE IF EXISTS `rule`;
CREATE TABLE `rule` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `rule` varchar(255) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;

-- ----------------------------
-- Table structure for `seminar`
-- ----------------------------
DROP TABLE IF EXISTS `seminar`;
CREATE TABLE `seminar` (
 `id` int(100) NOT NULL AUTO_INCREMENT,
 `periode_id` smallint(3) NOT NULL,
 `judul_skripsi_id` int(8) NOT NULL,
```

```
 `tgl_daftar` date NOT NULL,
 `status` tinyint(1) NOT NULL DEFAULT '0',
 `seminar_nilai_pembimbing` int(11) DEFAULT NULL,
 `seminar_nilai_penguji_1` int(11) DEFAULT NULL,
 `seminar_nilai_penguji_2` int(11) DEFAULT NULL,
 PRIMARY KEY (`id`),
 UNIQUE KEY `mhs` (`judul_skripsi_id`)
) ENGINE=InnoDB AUTO_INCREMENT=377 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `sidang`
-- ----------------------------
DROP TABLE IF EXISTS `sidang`;
CREATE TABLE `sidang` (
 `id` int(100) NOT NULL AUTO_INCREMENT,
 `periode_id` smallint(3) NOT NULL,
 `judul_skripsi_id` int(8) NOT NULL,
 `tgl_daftar` date NOT NULL,
 `status` tinyint(1) NOT NULL DEFAULT '0',
 `sidang_nilai_pembimbing` int(11) DEFAULT NULL,
 `sidang_nilai_penguji_1` int(11) DEFAULT NULL,
 `sidang_nilai_penguji_2` int(11) DEFAULT '0',
 `sub_penguji_id_1` int(11) DEFAULT NULL,
 `sub_penguji_id_2` int(11) DEFAULT '0',
 PRIMARY KEY (`id`),
 UNIQUE KEY `mhs` (`judul_skripsi_id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `staff`
-- ----------------------------
DROP TABLE IF EXISTS `staff`;
CREATE TABLE `staff` (
```

```
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `nip` bigint(11) NOT NULL,
 `nama` varchar(40) NOT NULL,
 `jenis_kelamin` char(1) NOT NULL,
 `tempat_lahir` varchar(20) NOT NULL,
 `tanggal_lahir` date NOT NULL,
 `alamat` varchar(200) NOT NULL,
 `email` varchar(50) NOT NULL,
 `telepon` varchar(15) NOT NULL,
 `foto` varchar(40) NOT NULL,
 `jj` smallint(3) NOT NULL DEFAULT '0',
 `status_staff_id` int(11) NOT NULL DEFAULT '0',
 `jurusan_id` int(11) NOT NULL,
 `username` varchar(20) NOT NULL,
 `password` varchar(200) DEFAULT NULL,
 `access_token` varchar(300) DEFAULT NULL,
 `acc_admin` int(1) DEFAULT '0',
 PRIMARY KEY (`id`,`nip`,`email`)
) ENGINE=InnoDB AUTO_INCREMENT=35 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `staff_rules`
-- ----------------------------
DROP TABLE IF EXISTS `staff_rules`;
CREATE TABLE `staff_rules` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `staff_id` int(11) DEFAULT NULL,
 `rule_id` int(11) DEFAULT NULL,
 `option_pembimbing` int(11) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=76 DEFAULT CHARSET=latin1;
-- ----------------------------
```

```
-- Table structure for `status_staff`
-- ----------------------------
DROP TABLE IF EXISTS `status_staff`;
CREATE TABLE `status_staff` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `status` varchar(200) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `step`
-- ----------------------------
DROP TABLE IF EXISTS `step`;
CREATE TABLE `step` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `step` varchar(200) DEFAULT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
-- ----------------------------
-- Table structure for `tahun_ajaran`
-- ----------------------------
DROP TABLE IF EXISTS `tahun_ajaran`;
CREATE TABLE `tahun_ajaran` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `tahun_awal` year(4) NOT NULL,
 `tahun_akhir` year(4) NOT NULL,
 `smt` tinyint(1) NOT NULL DEFAULT '0',
 `status` tinyint(1) NOT NULL DEFAULT '0',
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

### 5.1.5 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan membuat antar muka pada masing-masing form. Setiap halaman form yang dibuat akan dibentuk sebuah file yang berekstensi php. File-file tersebut selanjutnya dapat diakses dan akan menjadi penghubung antara pengguna dengan sistem.

#### *5.1.5.1* Implementasi Antar Muka Halaman Admin

Halaman Admin merupakan halaman yang digunakan dalam mengolah seluruh data aplikasi, dari pembuatan jadwal sampai konfigurasi data master. Halaman ini hanya dapat diakses oleh Admin. Berikut penjelasan menu-menu yang ada di halaman Mahasiswa :

**Tabel V.1** Implementasi Antar Muka Halaman Admin

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Penjadwalan Seminar | Membuat dan mengelola data penjadwlan seminar. | Process.php<br>Class Process |
| Penjadwalan Sidang | Membuat dan mengelola data penjadwlan sidang. | Process.php<br>Class Process |
| Laporan Nilai | Mengelola dan mencetak laporan nilai hasil seminar dan sidang mahasiswa. | Process.php<br>Class Process |
| Laporan Keuangan | Mengelola dan mencetak laporan keuangan dosen pembimbing dan penguji. | Process.php<br>Class Process |

### *5.1.5.2* Implementasi Antar Muka Halaman Mahasiswa

Halaman Mahasiswa merupakan halaman yang digunakan dalam mengolah data yang berkaitan dengan data-data pendaftaran dan pencetakan persyaratan seminar/sidang Mahasiswa. Halaman ini hanya dapat diakses oleh Mahasiswa. Berikut penjelasan menu-menu yang ada di halaman Mahasiswa :

**Tabel V.2** Implementasi Antar Muka Halaman Mahasiswa

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Dasboard Informasi | Mengetahui proses yang telah di lakukan | Step.php<br>Class Step |
| Pendaftaran Seminar | Melakukan pendaftaran pengisian dan penguduhan dokumen untuk syarat seminar | Step.php<br>Class Step |
| Pendaftaran Sidang | Melakukan pendaftaran pengisian dan penguduhan dokumen untuk syarat seminar | Step.php<br>Class Step |

### *5.1.5.3* Implementasi Antar Muka Halaman Pembimbing

Halaman Pembimbing merupakan halaman yang digunakan dalam mengolah waktu ketersedian pembimbing untuk jadwal bimbingan Mahasiswa. Halaman ini hanya dapat diakses oleh Pembimbing. Berikut penjelasan menu-menu yang ada di halaman Pembimbing :

**Tabel V.3** Implementasi Antar Muka Halaman Pembimbing

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Persetujuan Pengajuan Seminar Mahasiswa | Melakukan persetujuan pengajuan seminar mahasiswa | Process.php<br>Class Process |
| Persetujuan Pengajuan Sidang Mahasiswa | Melakukan persetujuan pengajuan sidang mahasiswa | Process.php<br>Class Process |
| Ketersediaan Waktu Membimbing Seminar | Memilih slot jadwal bimbingan seminar | Process.php<br>Class Process |
| Ketersediaan Waktu Membimbing Sidang | Memilih slot jadwal bimbingan sidang | Process.php<br>Class Process |

#### *5.1.5.4* Implementasi Antar Muka Halaman Penguji

Halaman Penguji merupakan halaman yang digunakan dalam mengolah ketersedian waktu Penguji untuk jadwal seminar dan Mahasiswa. Halaman ini hanya dapat diakses oleh Penguji. Berikut penjelasan menu-menu yang ada di halaman Penguji :

**Tabel V.4** Implementasi Antar Muka Halaman Penguji

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Ketersediaan Waktu Menguji Seminar | Memilih slot jadwal yang sudah ada data mahasiswa seminar yang diajukan pembimbing mahasiwa | Process.php<br>Class Process |

### *5.1.5.5* Implementasi Antar Muka Halaman Sekretariat Jurusan

Halaman Sekjur merupakan halaman yang digunakan dalam mengolah kelengkapan persyaratan pengajuan seminar dan sidang Mahasiswa. Halaman ini hanya dapat diakses oleh Sekjur. Berikut penjelasan menu-menu yang ada di halaman Sekjur :

**Tabel V.5** Implementasi Antar Muka Halaman Sekjur

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Persyaratan Seminar | Mengecek perlengkapan persyaratan seminar | Process.php<br>Class Process |
| Persyaratan Sidang | Mengecek perlengkapan persyaratan sidang | Process.php<br>Class Process |

### 5.1.6 Implementasi Instalasi Program

Berikut ini adalah cara menginstall XAMPP:

i. Jalankan file **xampp-win32-1.4.12-installer.exe** (atau versi lainnya)

ii. Kemudian akan tampil pilihan untuk memilih bahasa ketika proses instalasi berjalan. Silakan pilih bahasa **Indonesian** atau **English**, kecuali anda menguasai bahasa lainnya. Pada contoh ini saya memilih bahasa Indonesian karena saya cinta bahasa Indonesia.




**Gambar 5.1** XAMPP Pemilihan Bahasa

iii. Proses instalasi akan dimulai. Klik **Maju** untuk memulainya.




**Gambar 5.2** XAMPP Setup

iv. Akan muncul lisensi software. Silahkan membacanya jika anda mau, tetapi saya lebih suka untuk tidak membacanya karena terlalu banyak. Klik **Saya Setuju** untuk melanjutkan.



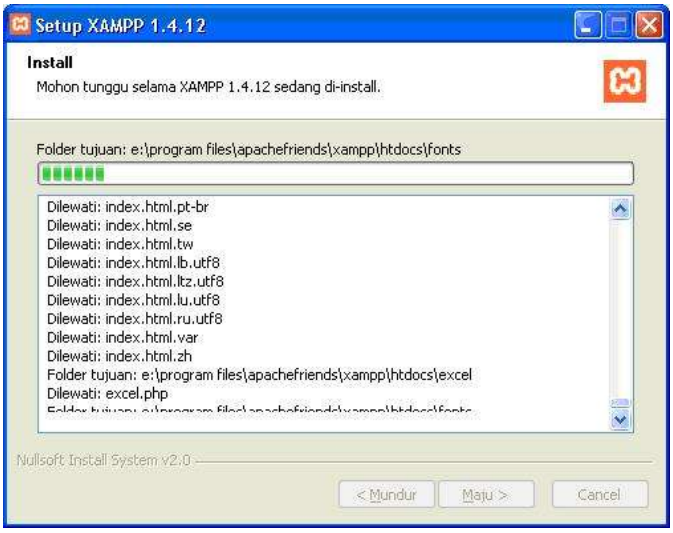

**Gambar 5.3** XAMPP Lisensi Software

v. Selanjutnya silakan anda **pilih lokasi install** untuk XAMPP. Kemudian klik **install**




**Gambar 5.4** XAMPP Pemilihan Lokasi Install

vi. Tunggu beberapa saat sampai proses instalasi selesai.



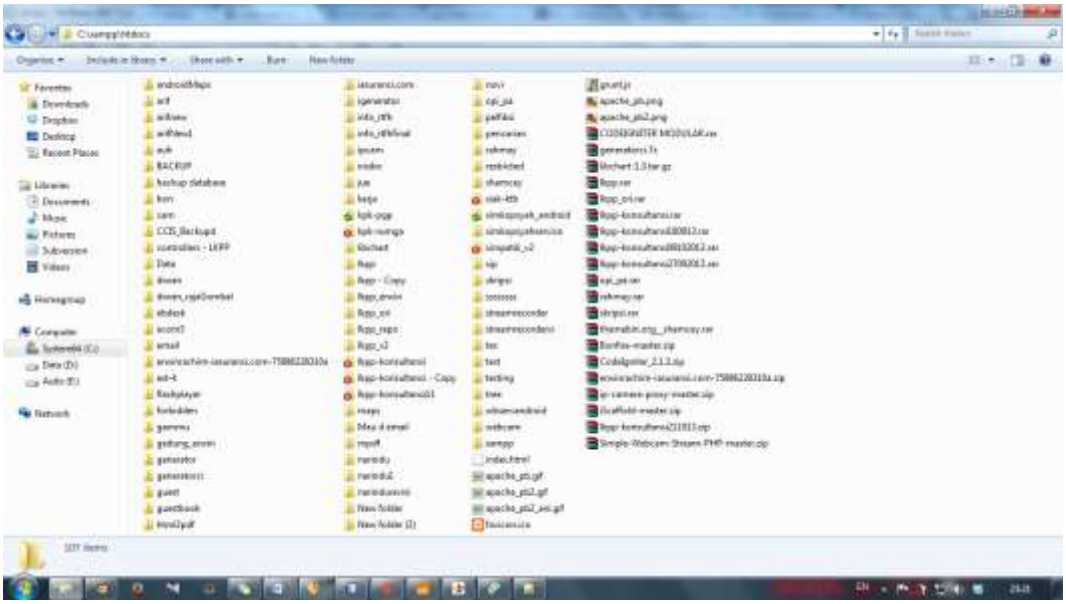

**Gambar 5.5** XAMPP Loading Install

vii. Instalasi Xampp selesai



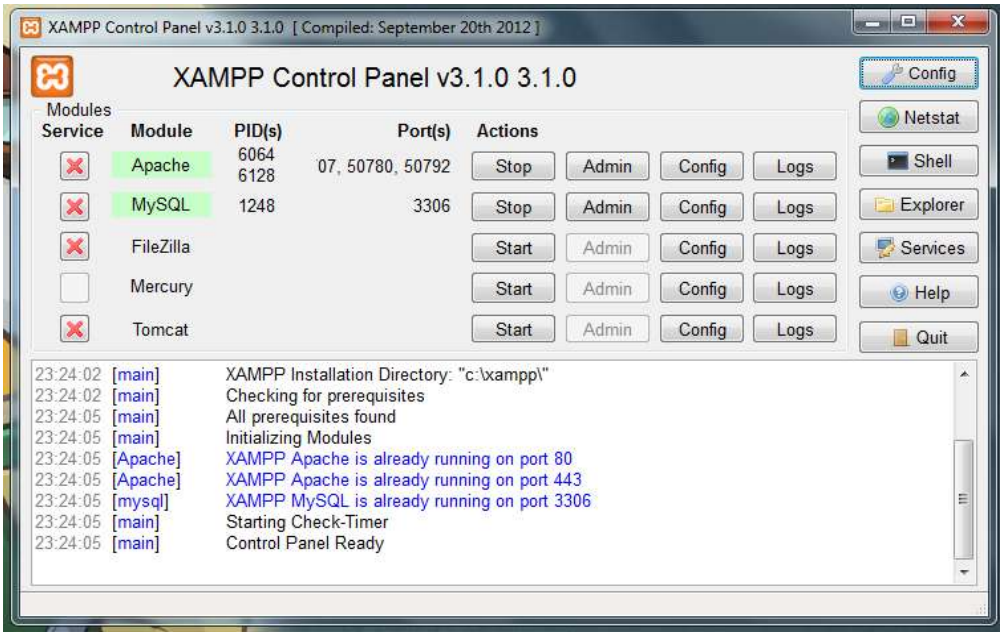

**Gambar 5.6** XAMPP Install Selesai

viii. Copy Aplikasi ke folder htdocs didalam folder XAMPP, bila location install di root C: maka copy ke C:\xampp\htdocs



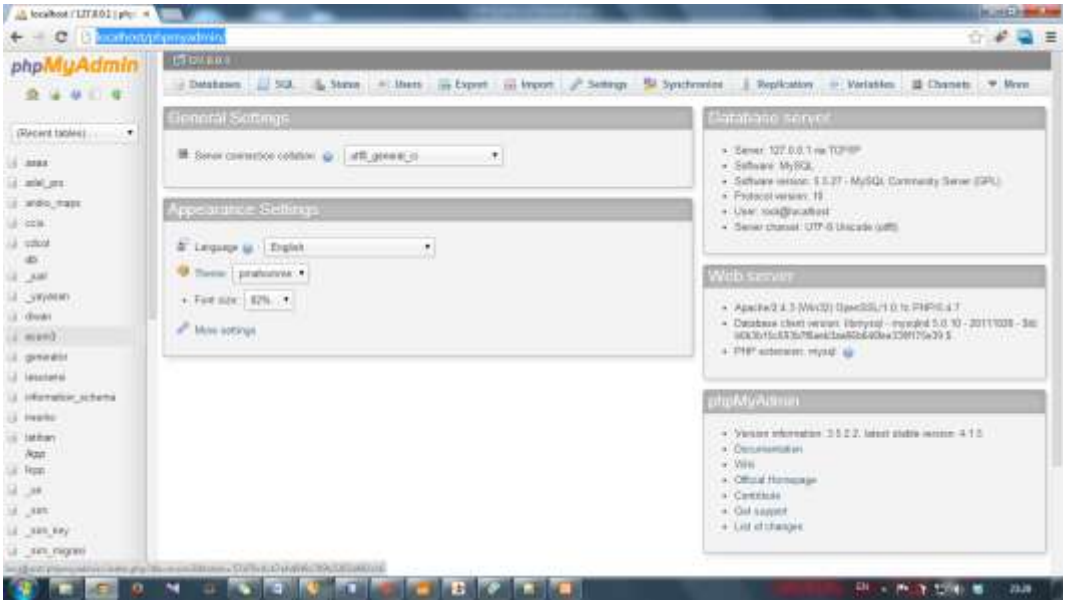
**Gambar V.1** Copy Aplikasi Ke Htdocs

ix. Jalankan Service Apache dan MySql di Window Control XAMPP



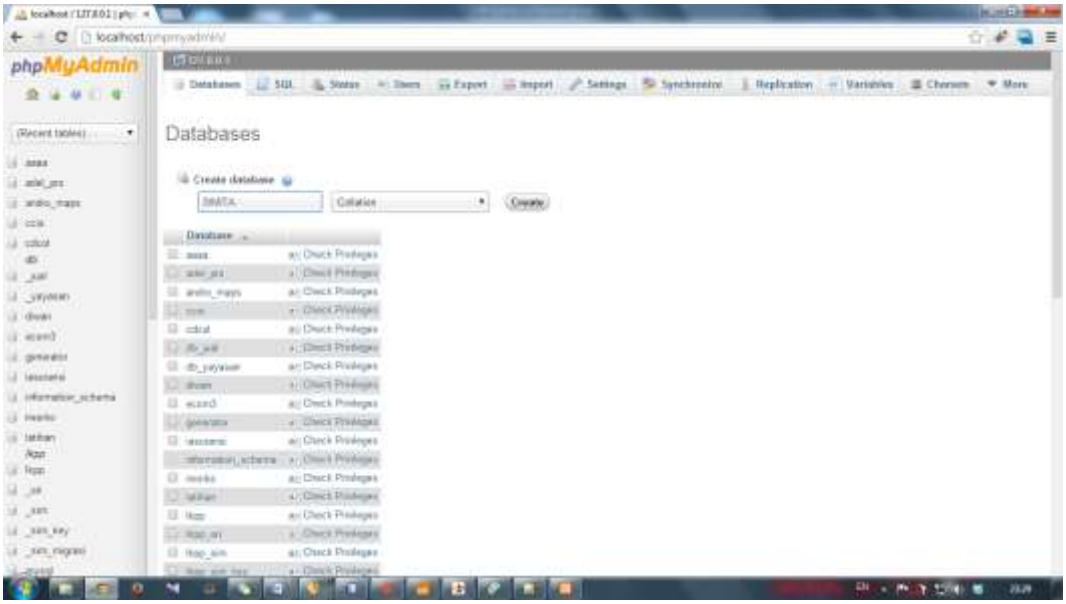

**Gambar V.2** Menjalankan Services Apache Dan MySQL

x. Buka URL http://localhost/phpmyadmin/ pada browser



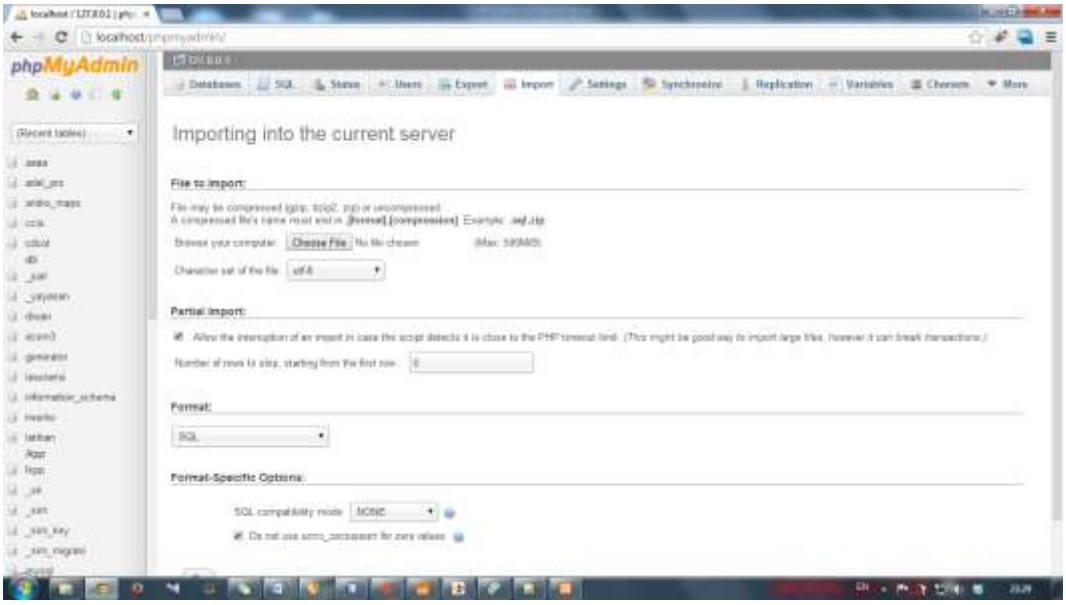

**Gambar V.3** Buka URL PHPMYADMIN

xi. Buat database baru dengan nama yang di inginkan, misalkan simita



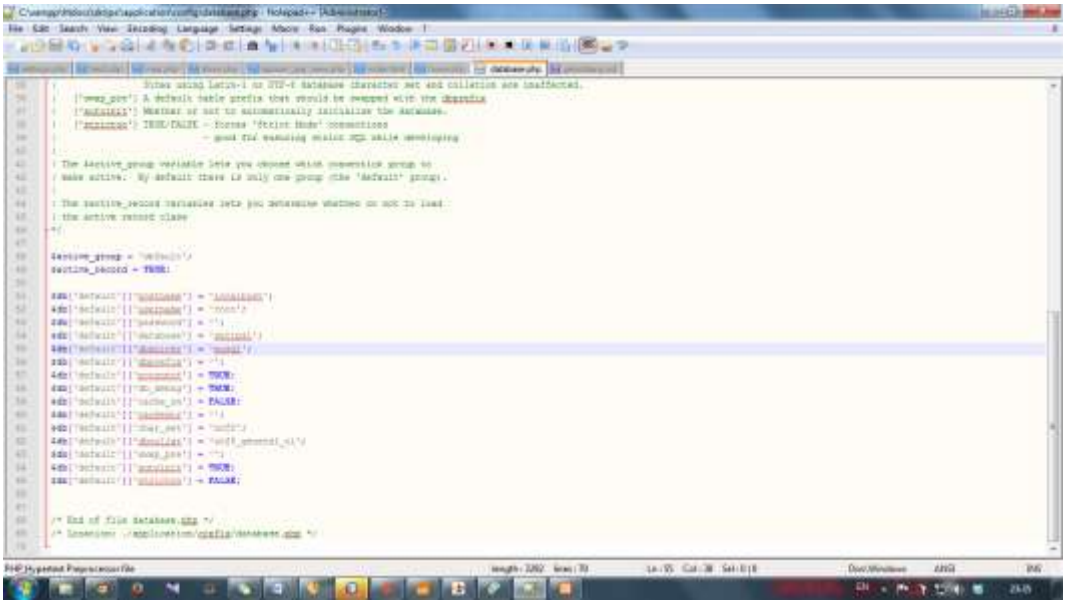

**Gambar V.4** Membuat Database

xii. Masuk ke menu *Import*, masukan file Dump SQL (Ekstensi .sql / sql.zip) setelah tekan tombol *GO*




**Gambar V.5** Import Data SQL

xiii. Setelah semua data di database sudah masuk. Maka konfigurasi file database.php di C:\xampp\htdocs\nama_aplikasi\application\config



**Gambar V.6** Konfigurasi Database

xiv. Setelah semua konfigurasi dan data selesai maka langkah terakhir adalah membuka aplikasi website di browser dengan mengetikan URL http://localhost/nama_aplikasi/




**Gambar V.7** Buka Aplikasi Di Browser

### 5.1.7 Penggunaan Program

Penggunaan program merupakan langkah-langkah dalam menggunakan dan menjalankan suatu sistem informasi yang dilakukan oleh pengguna (user). Langkah-langkah penggunaan program sebagai berikut :

### *5.1.7.1* Penggunaan Halaman Mahasiswa

Pengunaan program atau aplikasi yang dilakukan oleh akses user Mahasiswa sebagai berikut:

1. **Form Menu Login Mahasiswa**

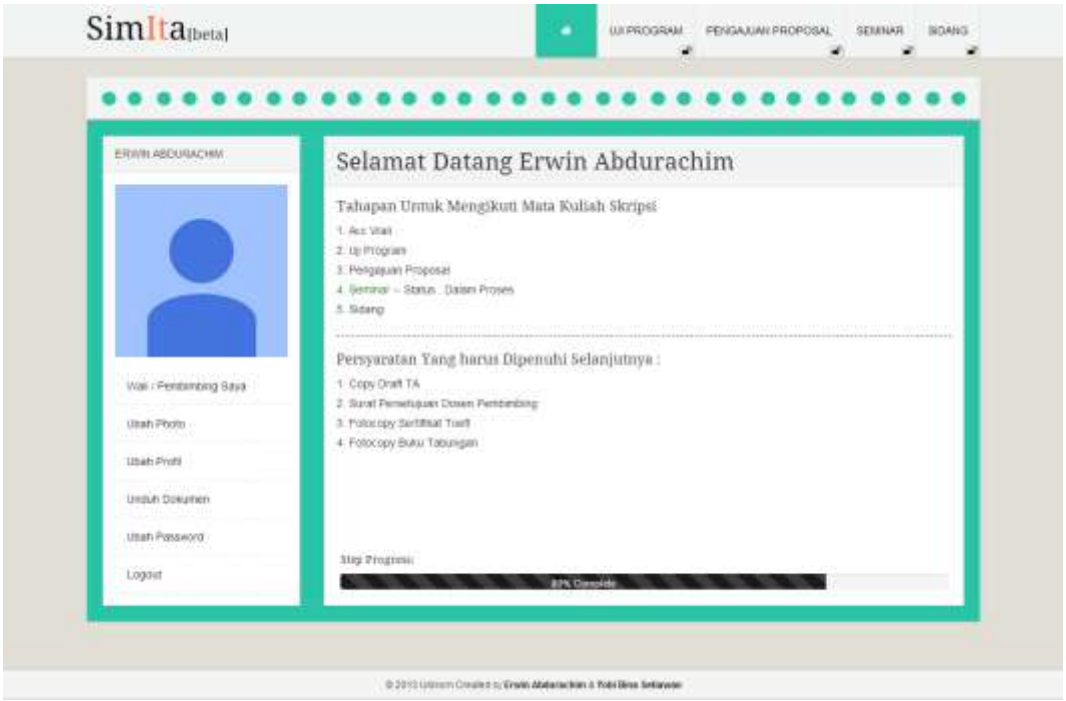
Sebelum masuk pada pilihan menu utama, user diharuskan mengisi dahulu username dan password. Pilihan menu yang dapat diakses atau digunakan disesuaikan dengan kebutuhan user dan diatur berdasarkan hak akses user, yang dikelola dan dibawah tanggung jawab Adiministrator (Admin).




**Gambar V.8** Form Login Mahasiswa

2. **Dashboard**

Halaman Dashboard ini terdapat pada menu awal pada saat berhasil melakukan login. Pada halaman ini, mahasiswa pengetahui proses atau tahapan yang telah di lakukan dan akan dilakukan. Berikut ini tampilan dari form dashboard:




**Gambar V.9** Halaman Dashboard Info Tahapan Mahasiswa

### 3. Form Pengajuan Pendaftaran Seminar

Pada form ini berfungsi untuk melakukan pengajuan pendaftaran dan untuk mengetahui info persyaratan pendaftaran Seminar yang dilakukan oleh mahasiswa. Berikut tampilan dari Form Pengajuan Seminar :



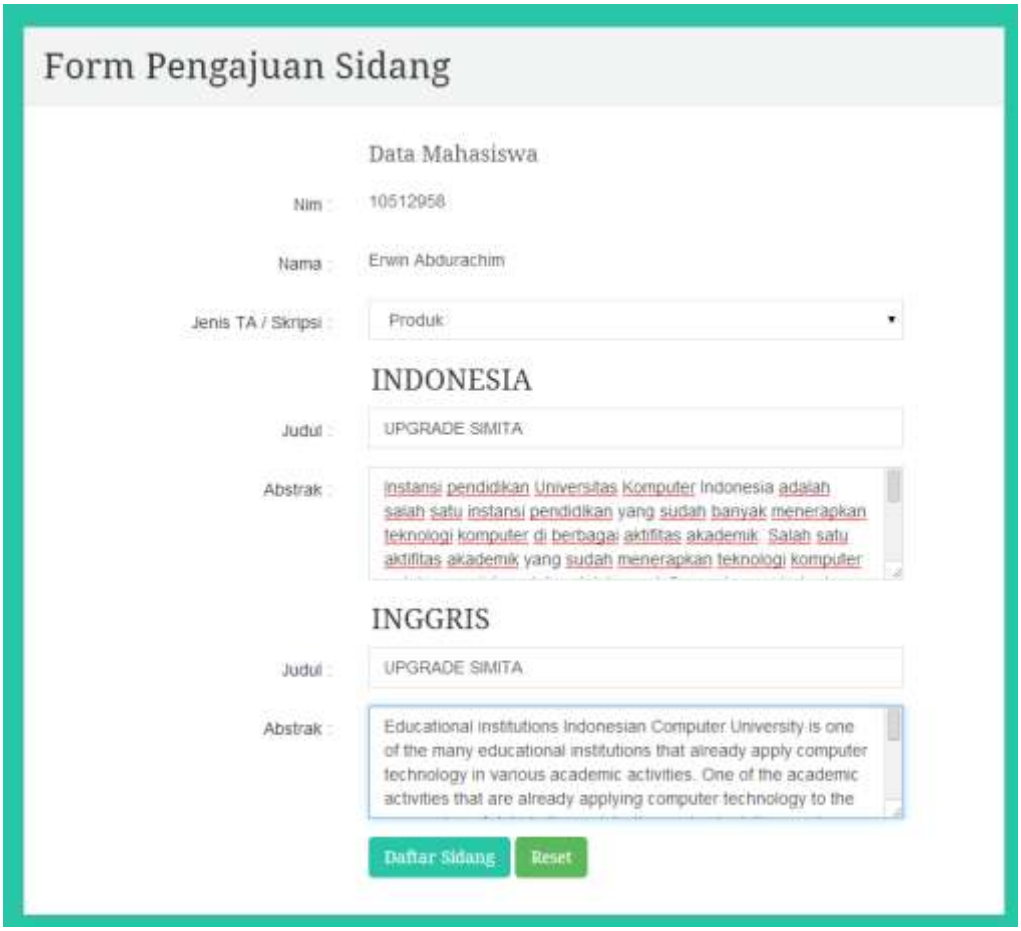

**Gambar V.10** Form Pengajuan Pendaftaran Seminar

Pada tampilan Form Pengajuan Seminar terdapat beberapa tombol yang disediakan untuk pengolahan data, antara lain :

a. Pengajuan Seminar, digunakan untuk menyimpan data
b. Reset, digunakan untuk mengosongkan data

**4. Form Pengajuan Pendaftaran Sidang**

Sama halnya dengan pengajuan seminar pada form ini berfungsi untuk melakukan pengajuan pendaftaran dan untuk mengetahui info persyaratan pendaftaran Sidang yang dilakukan oleh mahasiswa. Berikut tampilan dari Form Pengajuan Sidang :



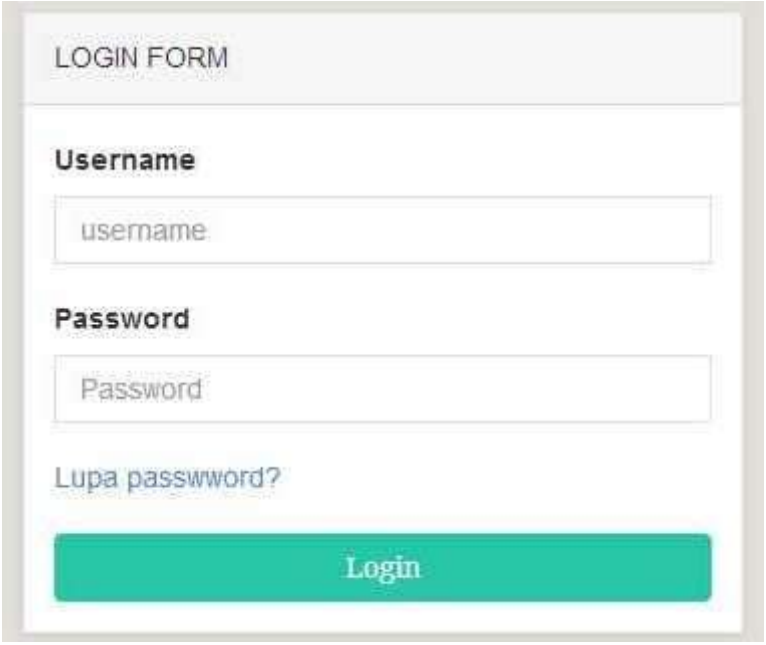

**Gambar V.11** Form Pengajuan Pendaftaran Sidang

Pada tampilan Form Pengajuan Seminar terdapat beberapa tombol yang disediakan untuk pengolahan data, antara lain :

a. Pengajuan Seminar, digunakan untuk menyimpan data
b. Reset, digunakan untuk mengosongkan data

### *5.1.7.2* Penggunaan Halaman Staff

Untuk masuk ke halaman staff maka user staff harus mengisi dahulu form login staff yang tampilannya sebagai berikut:




**Gambar V.12** Form Login Staff

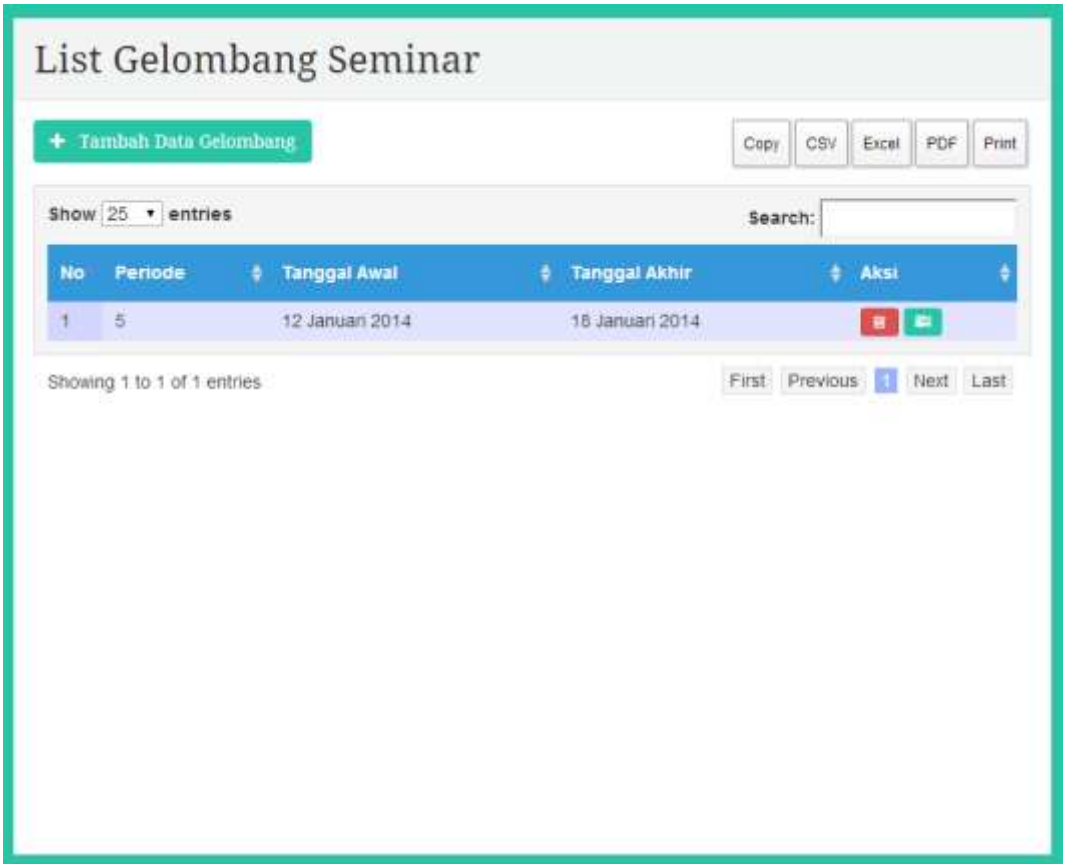
Setelah melakukan login pengunaan program atau aplikasi yang dilakukan oleh akses user Staff dibagi menjadi 4 hak akses yaitu sebagai berikut:

1. **Admin**

   a. **Penjadwalan**

      Untuk penjadwalan tahapann penjadwalan seminar dan sidang sama. Halaman tahapan penjadwalan di bagi menjadi sebagai berikut:

### i. List Gelombang

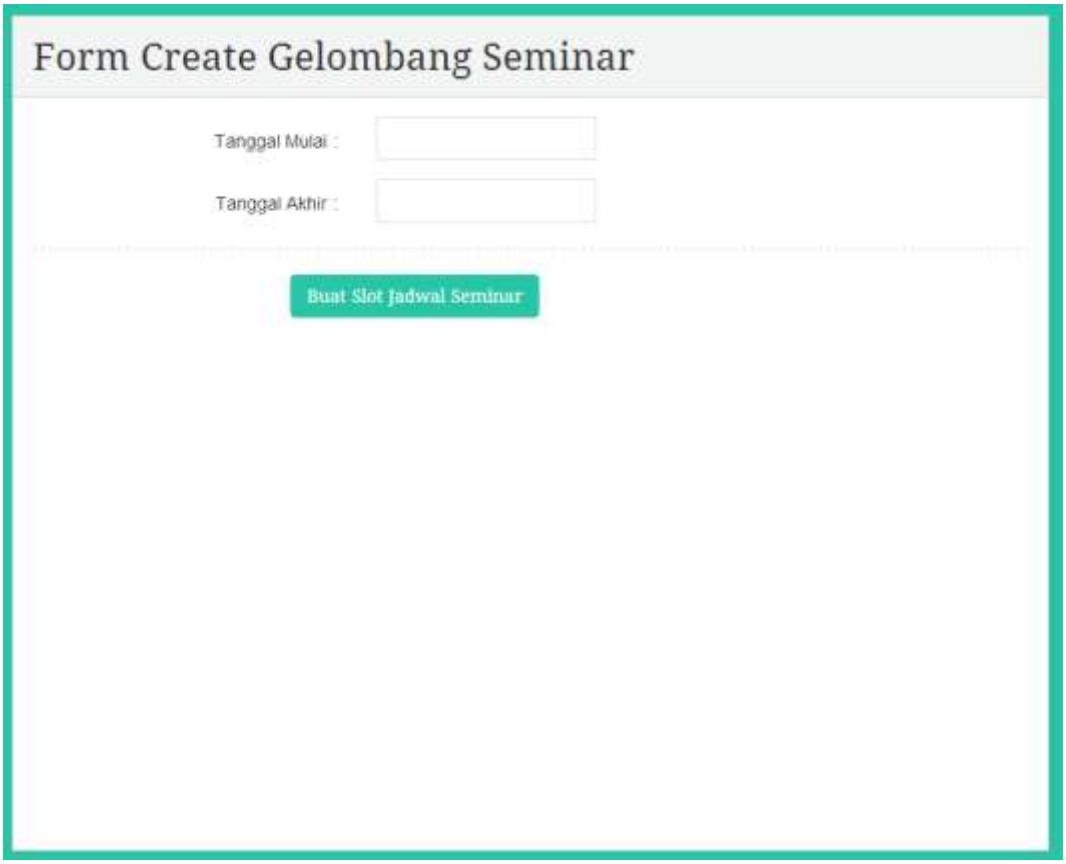



**Gambar V.13** Halaman List Gelombang jadwal

Halaman ini berfungsi sebagai pengelolaan gelombang jadwal yang telah di buat. Terdapat beberapa tombol dengan fungsi sebagai berikut:

a. Tambah data gelombang, akan masuk ke form gelombang untuk membuat data gelombang yang baru
b. Tombol delete gelombang, akan mendelete data geleombang peserta slot jadwal yang terisi.
c. Detail gelombang, akan masuk ke halaman detail gelombang yaitu berisi slot jadwlan yang sudah terisi

### ii. Form Gelombang

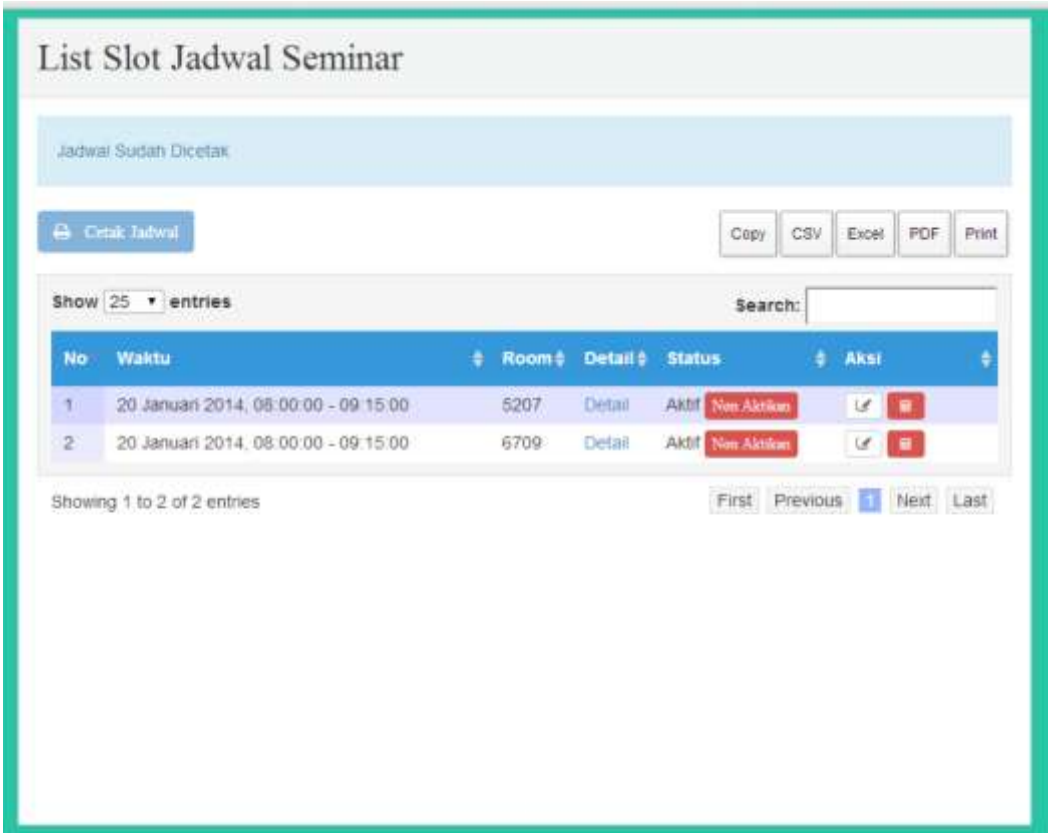



**Gambar V.14** From Gelombang

Form ini berfungsi sebagai input data gelombang jadwal. Terdapat beberapa tombol dengan fungsi sebagai berikut:

a. Tombol Buat slot jadwal, berfungsi sebagai pembuatan data slot jadwal berdasarkan data gelombang yang diinputkan

### iii. Detail Jadwal

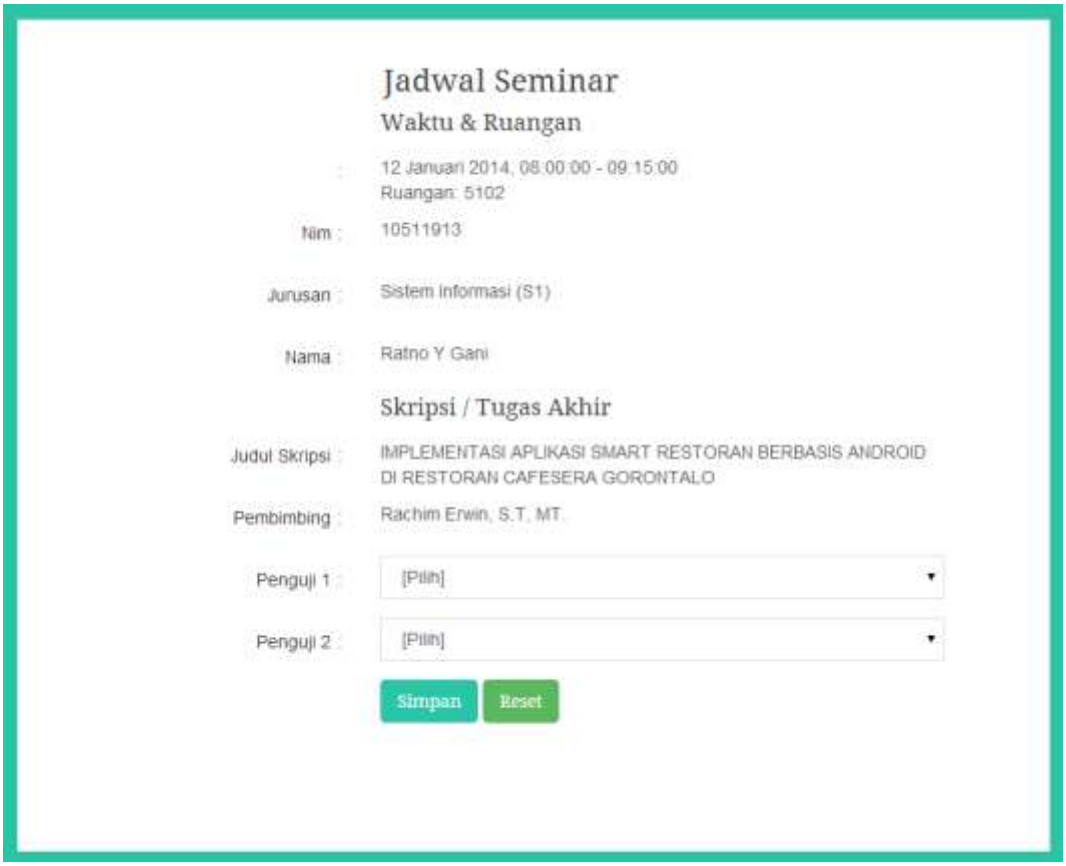



**Gambar V.15** Halaman List Slot Jadwal

Halaman ini berfungsi sebagai pengelolaan slot jadwal yang telah di buat dan pengelolaan ketersediaan waktu pengujian oleh admin. Terdapat beberapa tombol dengan fungsi sebagai berikut:

a. Tombol detail, berfungsi memperlihatkan detail jadwal yang telah terisi
b. Tombol active/non aktif, berfungsi mengaktifkan/menonaktifkan slot jadwal.
c. Tombol edit, berfungsi sebagai link untuk menuju form edit jadwal
d. Tombol delete, berfungsi untuk mengkosongkan slot jadwal yang telah terisi.

**iv. Form Jadwal**




**Gambar V.16** From Edit Jadwal

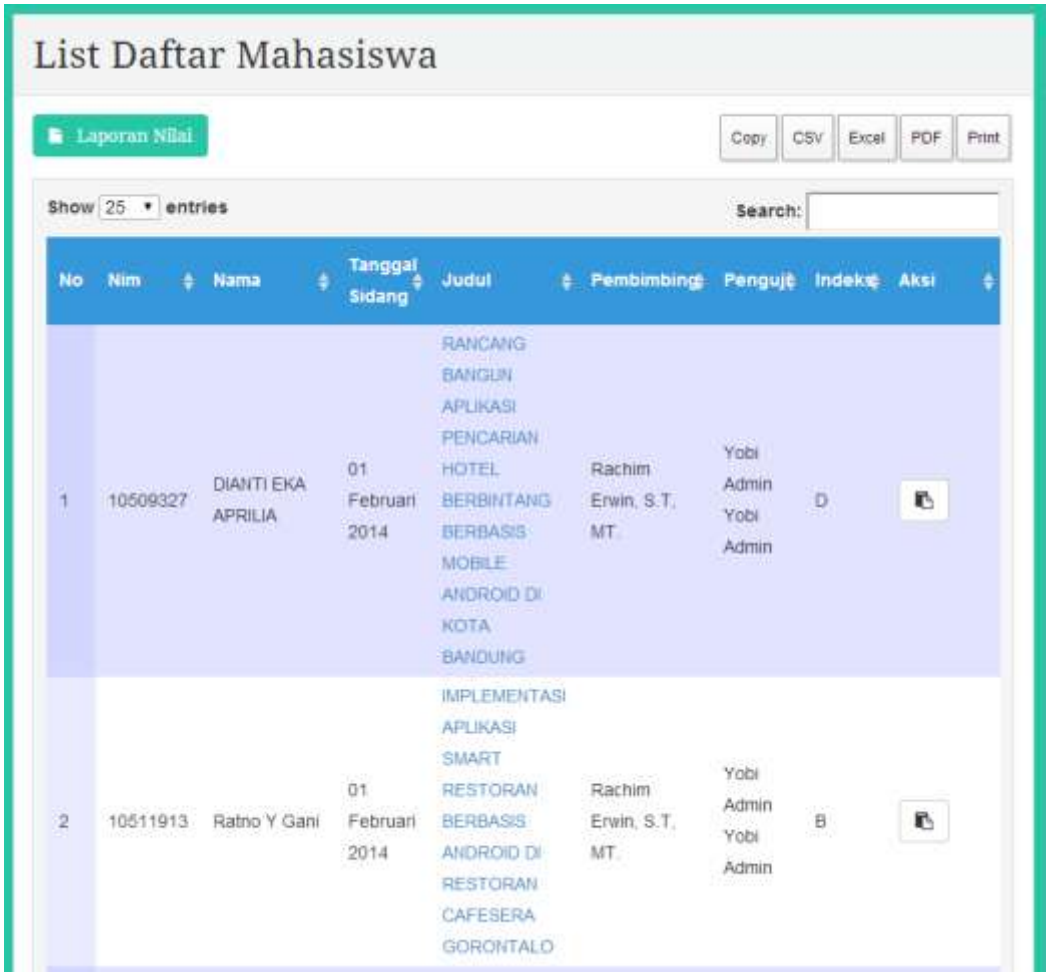
Form ini berfungsi sebagai input data ketersedian jadwal. Terdapat beberapa tombol dengan fungsi sebagai berikut:

a. Simpan, berfungsi untuk memasukan ketersediaan jadwal menguji dosen pembimbing dengan bantuan admin.
b. Reset, mengkosongkan inputan penguji 1 atau pun penguji 2.

**e. Laporan Nilai**

Halaman Laporan nilai berfungsi sebagai pengelolahan data nilai seminar dan sidang mahasiswa. Halaman laporan nilai sebagai berikut:

### i. List Laporan




**Gambar V.17** Halaman Laporan Nilai

Halaman ini berfungsi sebagai pengolah data mahasiswa yang sudah mengikuti seminar dan sidang. Ada beberapa tombol yang ada di halaman ini, yaitu

a. Tombol cetak, berfungsi untuk masuk ke form cetak nilai.

b. Tombol input nilai, berfungsi untuk masuk ke form input nilai

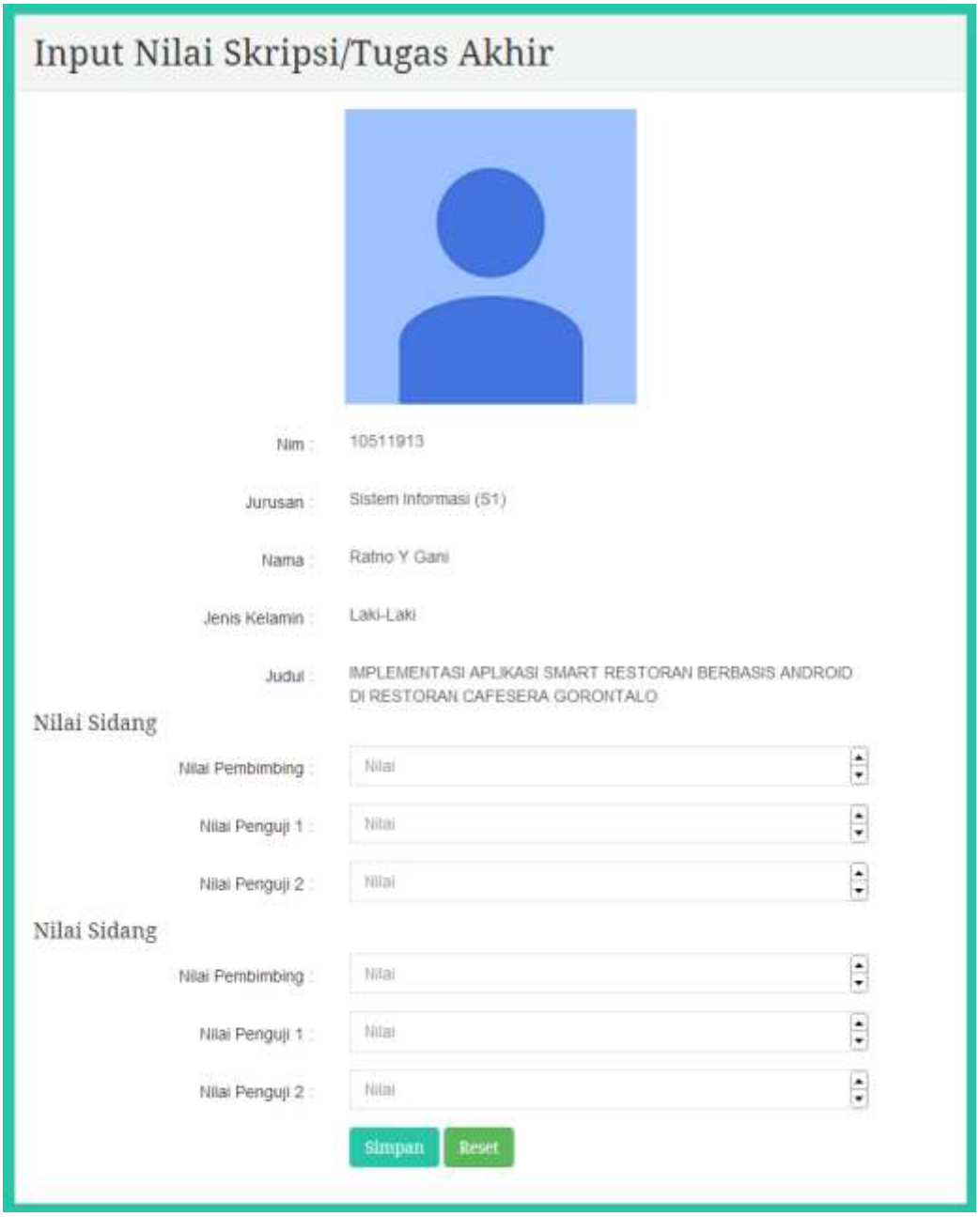
### ii. Form Cetak Laporan nilai




**Gambar V.18** Form Cetak Laporan Nilai

Form cetak nilai berfungsi untuk mencetak nilai skripsi mahasiswa berdasarkan tahun ajaran yang dipilih.

### iii. Form input nilai



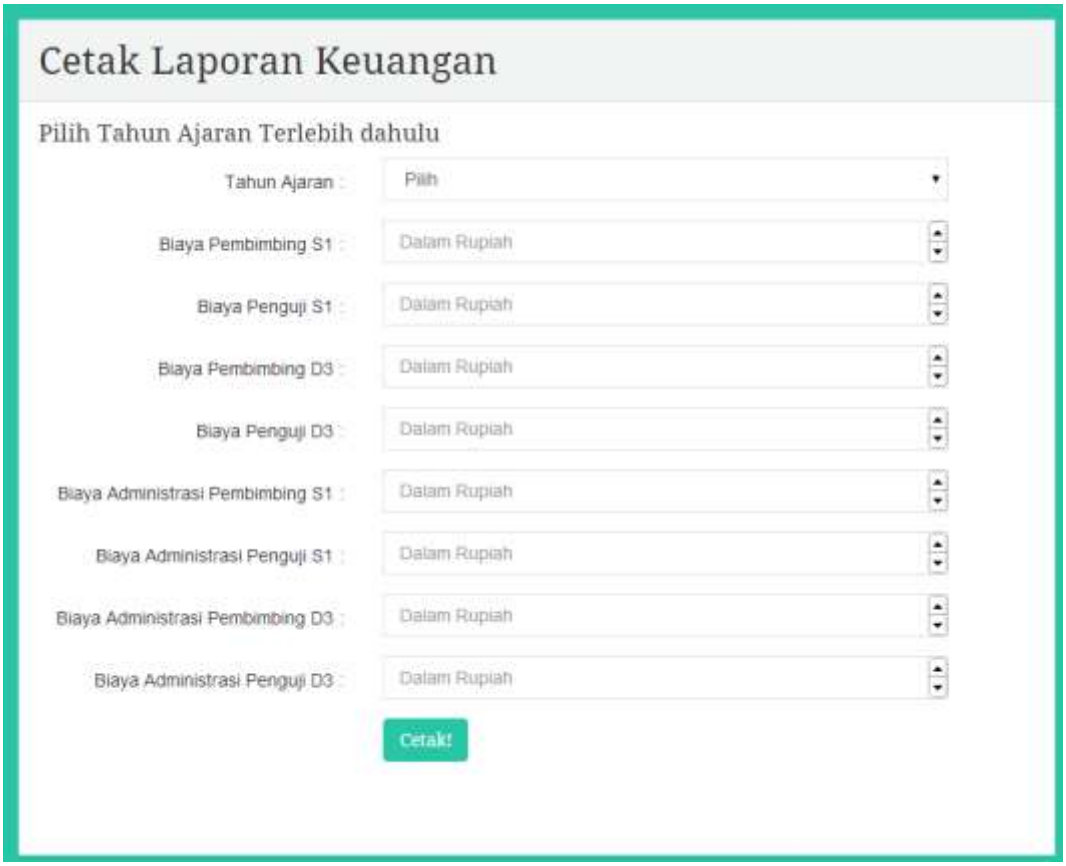

**Gambar V.19** Form Input Nilai skripsi/Tugas Akir

Halaman ini berfungsi sebagai peng-inputan data nilai mahasiswa yang sudah mengikuti seminar dan sidang. Ada beberapa tombol yang ada di halaman ini, yaitu:

a. Tombol simpan, berfungsi untuk menyimpan data nilai mahasiswa kedalam database.

b. Tombol reset, berfungsi untuk mengkosongkan field

**f. Laporan Keuangan**

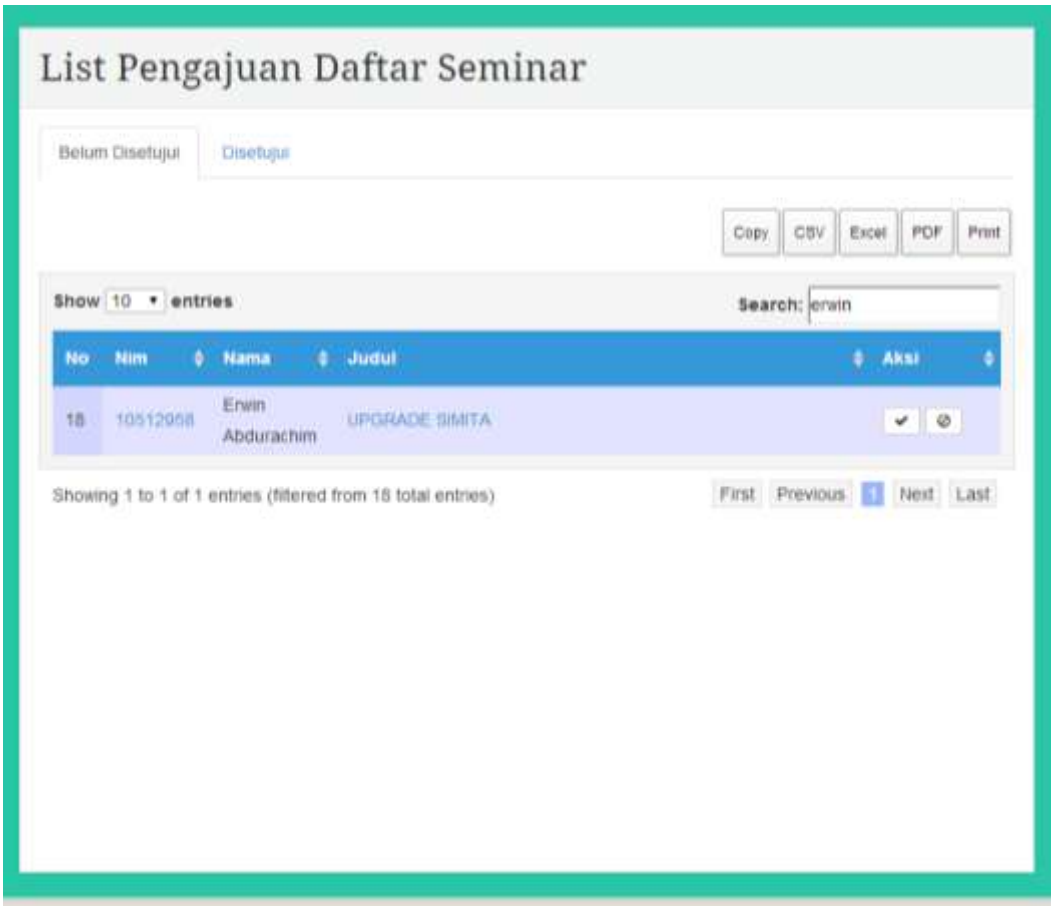



**Gambar V.20** Form Cetak Laporan Keuangan

Form cetak keuangan berfungsi untuk mencetak laporan keuangan skripsi per tahun ajaran dimana admin memasukan nilai biaya-biaya yang dihitung setiap bimbingan dan pengujian Skripsi/Tugas Akhir.

## 2. Dosen Pembimbing

### a. Persetujuan Seminar/Sidang




**Gambar V.21** Halaman Persetujuan Seminar/Sidang

Halaman ini berfungsi sebagai pemberian persetujuan kepada mahasiswa bimbingan oleh dosen pembimbing masing-masih. Ada beberapa tombol yang ada di halaman ini, yaitu:

a. Tombol tab disetujui, untuk melihat daftar mahasiswa yang sudah di terima.

b. Tombol terima, untuk menerima persetujuan pengajuan pendaftran mahasiswa dan melanjutkan mahasiswa kedalam tahap pengecekan persyaratan.

c. Tombol *Pending*, untuk merubah status mahasiswa menjadi pending.

**b. Ketersediaan Waktu Membimbing Seminar/Sidang**

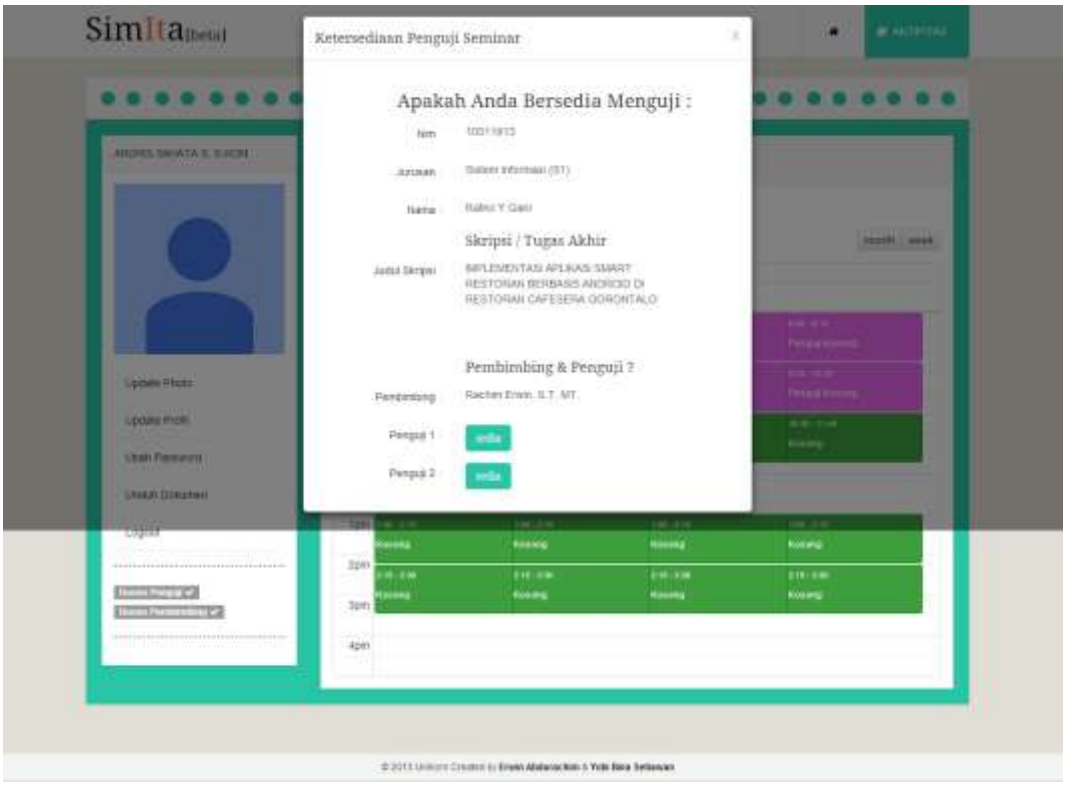



**Gambar V.22** Form Ketersediaan Waktu Membimbing

Form ini berfungsi sebagai pengisian waktu membimbing seminar/sidang mahasiswa bimbingan oleh masing-masin. Saat kuota untuk pengisian waktu membimbing/menguji tombol input tidak bisa melakukan action.

### 3. Dosen Penguji

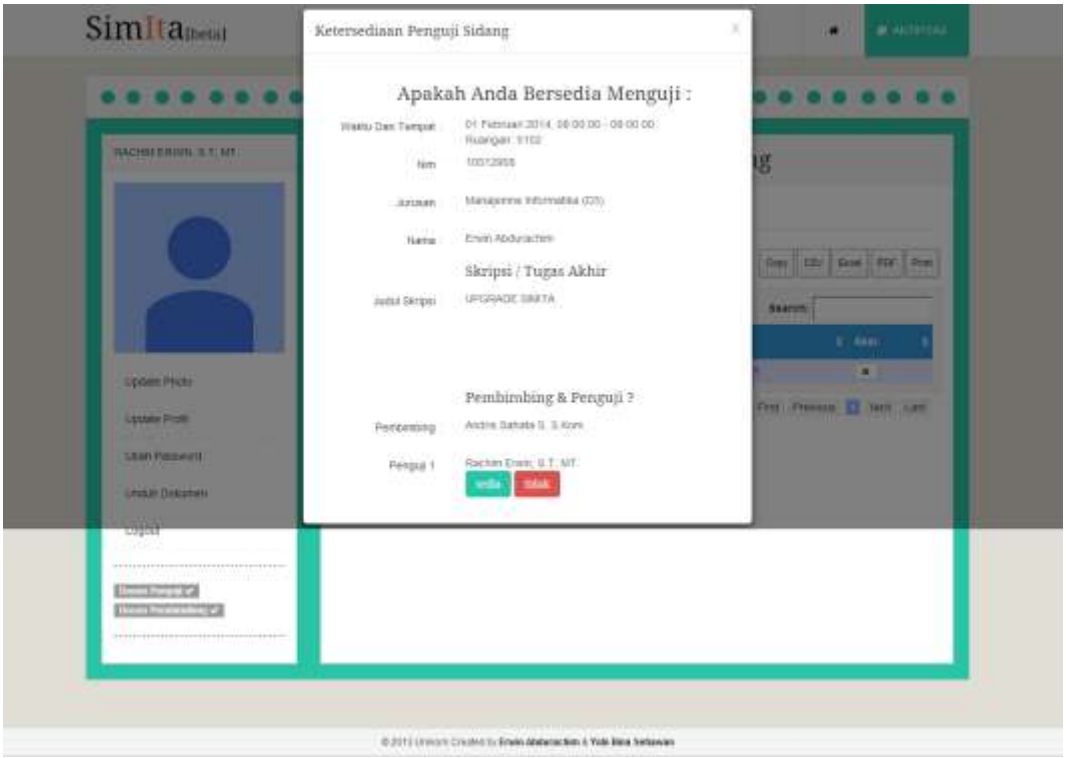
#### a. Ketersediaan Waktu Menguji Seminar




**Gambar V.23** Form ketersediaan menguji seminar

Form ini akan mencul setelah dosen pembimbing mengajukan anak bimbingannya pada slot yang tersedia. Setelah menekan kalender yang sudah terisi dosen pembimbing maka dosen penguji akan ditampilkan form seperti gambar 5.13, penguji tinggal memilih akan bersedia menjadi dosen penguji 1 atau 2 untuk mahasisa S1 atau menjadi penguji 1 saja untuk mahasiswa d3.

**b. Ketersediaan Waktu Menguji Sidang**

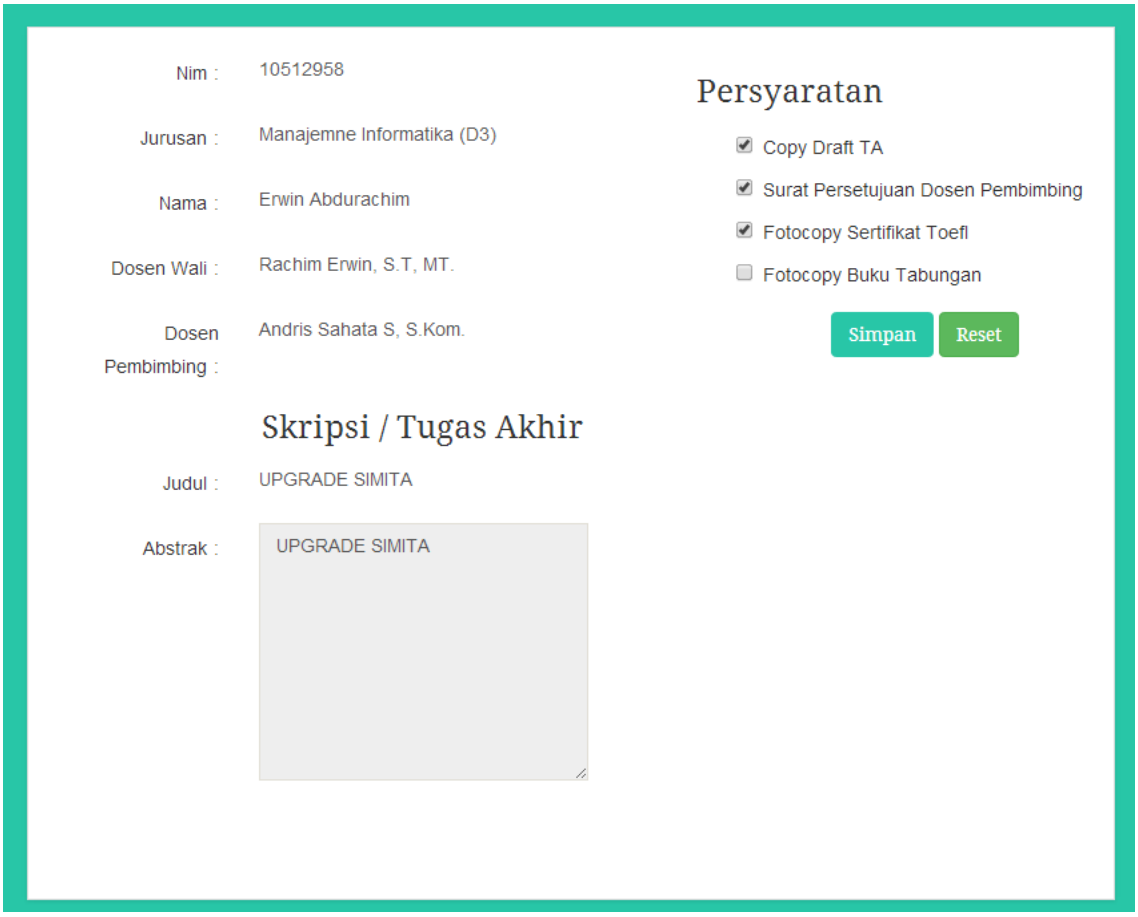



**Gambar V.24** Form ketersediaan menguji seminar

Halaman daftar masiswa ini akan mencul secara otomasti pada halaman dosen penguji setelah dosen pembimbing mengisi ketersediaan waktu memimbing sidang pada mahasiswa bimbingannya yang sebelumnya telah diuji pada saat seminar. Pada pilihan di atas diperlihatkan bahwa untuk pemiihan ada tombol sedia dan tidak bila menekan tombol sedia maka penguji brsedia menguji kembali pada saat sidang yang jadwalnya sudah diajukan pembimbing. Bila pada taham ini pembimbing tidak bersedia maka admin akan memilihkan dosen alternatif untuk menguji mahasiswa yang telah diajukan jadwal sidangnya oleh pembimbing.

**4. Sekretariat Jurusan**

**a. Persyaratan**




**Gambar V.25** Form Pengecekan Persyaratan

Form ini hanya ada pada user yang mempunyai hak akses secretariat jurusan. Fungsi form ini untuk pengecekan persyaratan yang telah di berikan mahasiswa yang telah mengajukan pendaftaran seminar atupun sidang. Bila persyaratan belum lengkap maka dosen pembimbing tidak bisa mengajukan jadwal untuk membimbing seminar/sidang mahasiswa.

## 5.2 Pengujian

Pengujian merupakan bagian yang tidak kalah pentingnya dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangakat lunak. Tujuan dari pengujian ini

adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu merepresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

### 5.2.1 Rencana Pengujian

Proses rencana pengujian meliputi input/output. Proses pengujian input/output adalah mencoba program dengan memasukan data ke dalam form form masukan yang telah disediakan. Pada tahap ini merupakan kelanjutan dari tahap implementasi yaitu melakukan pengujian- pengujian terhadap aplikasi yang telah dibangun. Adapun hal-hal yang akan diuji melalui teknik pengujian black box adalah sebagai berikut :

### 5.2.2 Kasus dan Hasil Pengujian

Berikut adalah hasil dari pengujian yang telah dilakukan :

1. Pengecekan Login

**Tabel 5.4** Pengujian Login

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Username:<br>Password: | Tercantum pada textbox username dan password | Dapat mengisi login sesuai yang diharapkan | [ x ] Diterima<br>[ ] Ditolak |
| Klik tombol submit | Login berhasil sesuai dengan hak aksesnya/ berdasarkan hak aksesnya | Tombol login dapat sesuai dengan yang diharapkan | [ x ] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| **Data masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| UserName :<br>Password : | Tidak dapat login | User tidak dapat login | [ x ] Diterima<br>[ ] Ditolak |
| Klik tombol submit | Tidak dapat masuk ke form menu utama sesuai dengan hak aksesnya | User tidak dapat login | [ x ] Diterima<br>[ ] Ditolak |

**Tabel V.6** Pengujian Inputan

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Data Pendaftaran | Semua Data Terisi | Dapat mengisi data sesuai yang diharapkan | [ x ] Diterima<br>[ ] Ditolak |
| Klik tombol submit | Data Pendaftaran Masuk Kedatabase Untuk Di Proses | Tombol Simpan dapat sesuai dengan yang diharapkan | [ x ] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| **Data masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Data Pendaftaran | Data Tidak Lengkap | Muncul Peringatan | [ x ] Diterima<br>[ ] Ditolak |
| Klik tombol submit | Data Tidak Masuk Kedalam database | Data tidak masuk | [ x ] Diterima<br>[ ] Ditolak |

### 5.2.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji sample diatas dapat ditarik kesimpulan bahwa perangkat lunak berjalan dengan baik dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan instansi pendidikan UNIKOM.