**BAB V**

**IMPLEMENTASI DAN PEMBAHASAN SISTEM**

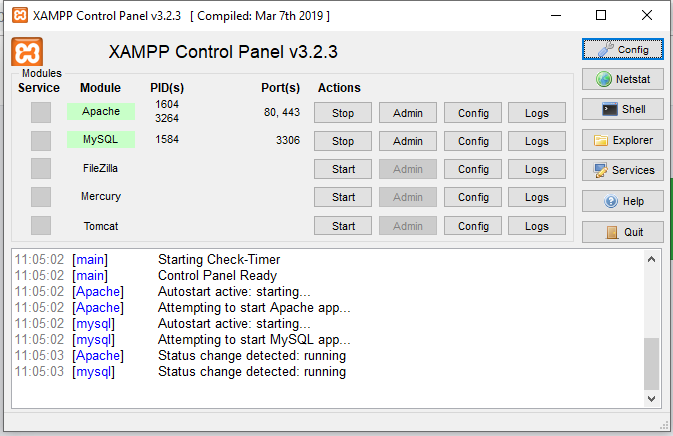
**5.1 Implementasi Sistem**

Sistem Ujian Online berbasis **website** yang mampu menampilkan soal-soal ujian sesuai kategori dan dapat melakukan penilaian hasil jawaban secara langsung, dengan demikian diharapkan dapat membantu peserta dalam mengikuti ujian online, Bahasa pemrograman yang akan digunakan dalam implementasi sistem adalah *PHP*, sedangkan untuk desain antarmuka akan menggunakan *HTML* dan *CSS* dan *MySQL* sebagai basis data.

**5.2 Pembahasan Sistem**

Berikut ini adalah pembahasan dan detail pemaparan dari masing – masing halaman web (*web pages*) dari Sistem Ujian Online berbasis website . Dalam bagian ini akan dipaparkan *screenshot* dari masing-masing halaman aplikasi web Sistem Ujian Online berbasis website beserta keterangan mengenai halaman tersebut.

Untuk dapat mengakses suatu basis data dengan *PHP* maka harus dibuat koneksi ke *localhost* komputer dengan menjalankan aplikasi *XAMPP* dan mengaktifkan *Module Apache* dan *MySQL,* dapat dilihat pada Gambar 5.1



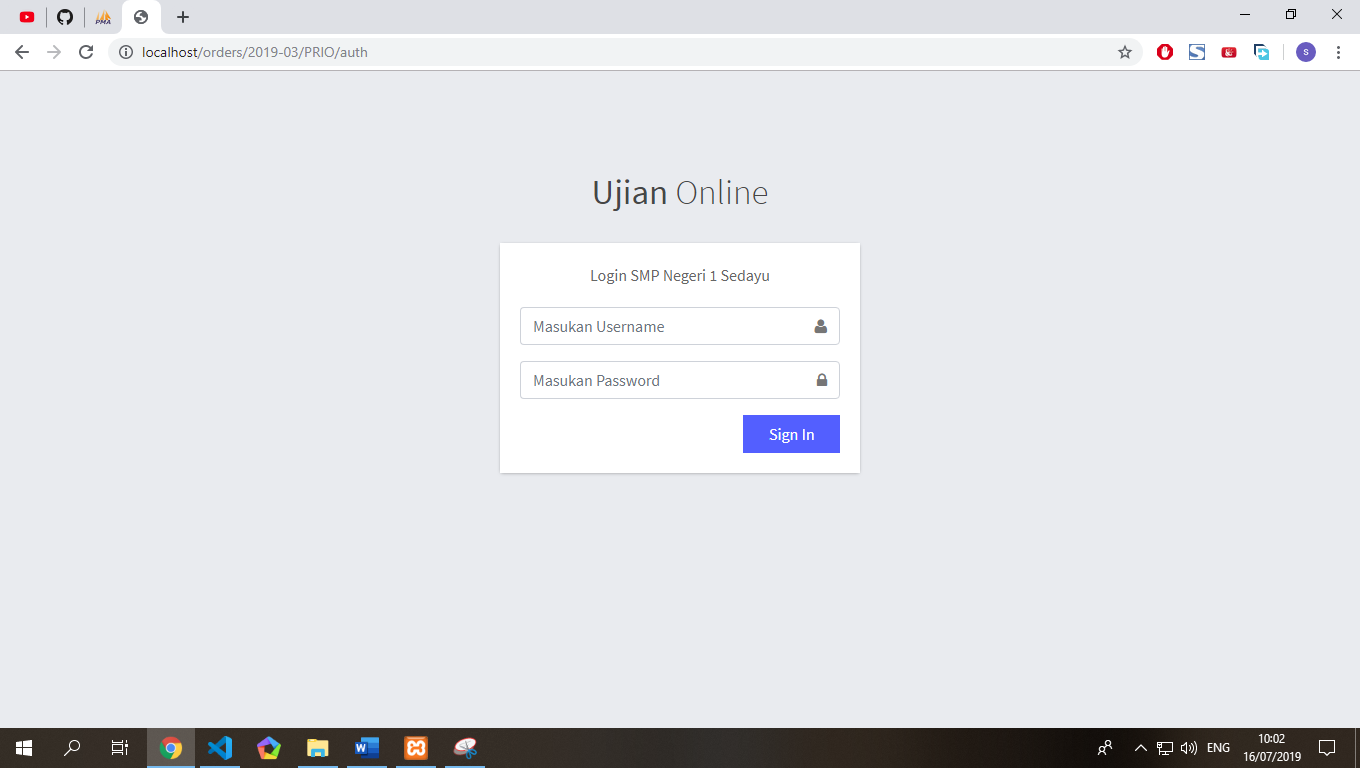
**Gambar 5.1** Aplikasi Web Server XAMPP

* + 1. **Halaman Admin**

Halaman admin merupakan halaman admin yang mempunyai hak akses penuh terhadap sistem, mulai dari input data admin,guru,siswa,kelas,pelajaran dan PBM. Berikut gambaran dari halaman Admin:

* + - 1. **Halaman Login User**

Halaman ini berisikan menu *login* yang dapat diakses oleh admin, untuk lebih jelas dapat dilihat pada Gambar 5.2



**Gambar 5.2** Halaman Login User

Ketika tombol “Sign In” di klik user akan diarah pada halaman dashboard user tetapi sebelum masuk halaman dashboard user akan diseleksi terlebih dahulu oleh proses login. *Script* proses login dapat dilihat pada Tabel5.1*.*

Tabel 5.1 Script Proses Login

function index(){

$this->load->view('login');

}

function process(){

$username = $this->input->post('username');

$password = md5($this->input->post('password'));

$where\_users = array(

'username' => $username,

'password' => $password

);

$cek\_users = $this

->m\_auth->check\_auth("users",$where\_users)->num\_rows();

// print\_r($cek\_users);

// die();

if ( $cek\_users > 0 ) {

# code...

$row = $this->m\_auth

->check\_auth("users",$where\_users)->row();

$data\_session = array(

'username' => $username,

'password' => $password,

'level' => $row->level,

'status' => 'login',

);

$this->session->set\_userdata($data\_session);

$url=[

'admin' => 'admin',

'guru' => 'guru',

'guru\_kep\_lab' => 'guru-kep-lab',

'siswa' => 'siswa',

];

redirect(base\_url( $url[$this

->session->userdata('level')] ));

}

else{

$this->session->set\_flashdata('msg',

'Maaf! Username atau Password anda salah!');

redirect(base\_url('auth'));

}

}

function logout(){

$this->session->sess\_destroy();

redirect(base\_url('auth'));

}

}

'Maaf! Username atau Password anda salah!');

redirect(base\_url('auth'));

}

}

function logout(){

$this->session->sess\_destroy();

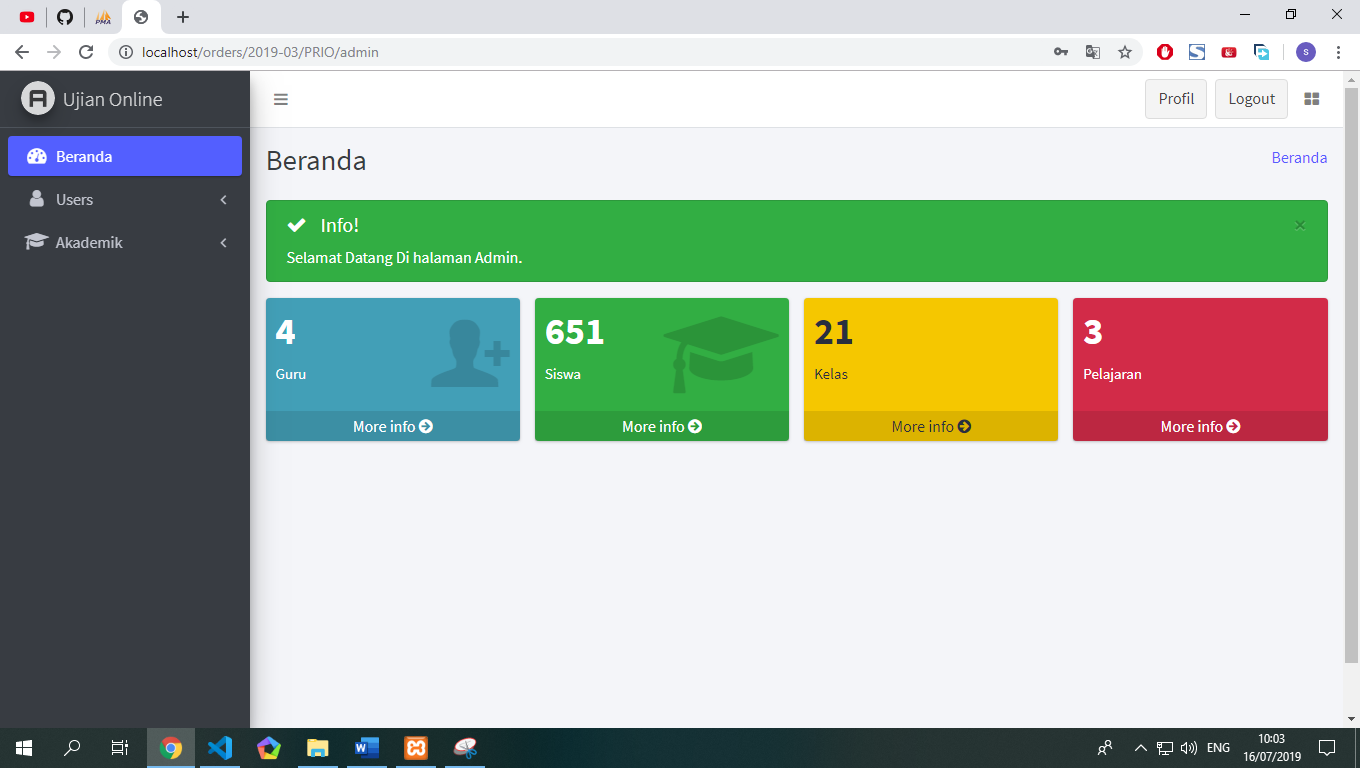
redirect(base\_url('auth'));

}

}

**2. Halaman Admin Beranda**

Halaman ini berisikan menu yang terdapat di form *admin* seperti data admin, guru, siswa, kelas, pelajaran dan PBM. untuk lebih jelas dapat dilihat pada Gambar 5.3 dan untuk scriptnya dapat dilihat pada Tabel 5.2.



**Gambar 5.3** Halaman Admin Beranda

Tabel 5.2. Script Halaman Admin Beranda

public function index()

{

$this->content['count\_guru']=

$this->m\_admin->beranda\_guru();

$this->content['count\_siswa']=

$this->m\_admin->beranda\_siswa();

$this->content['count\_kelas']=

$this->m\_admin->beranda\_kelas();

$this->content['count\_pelajaran']=

$this->m\_admin->beranda\_pelajaran();

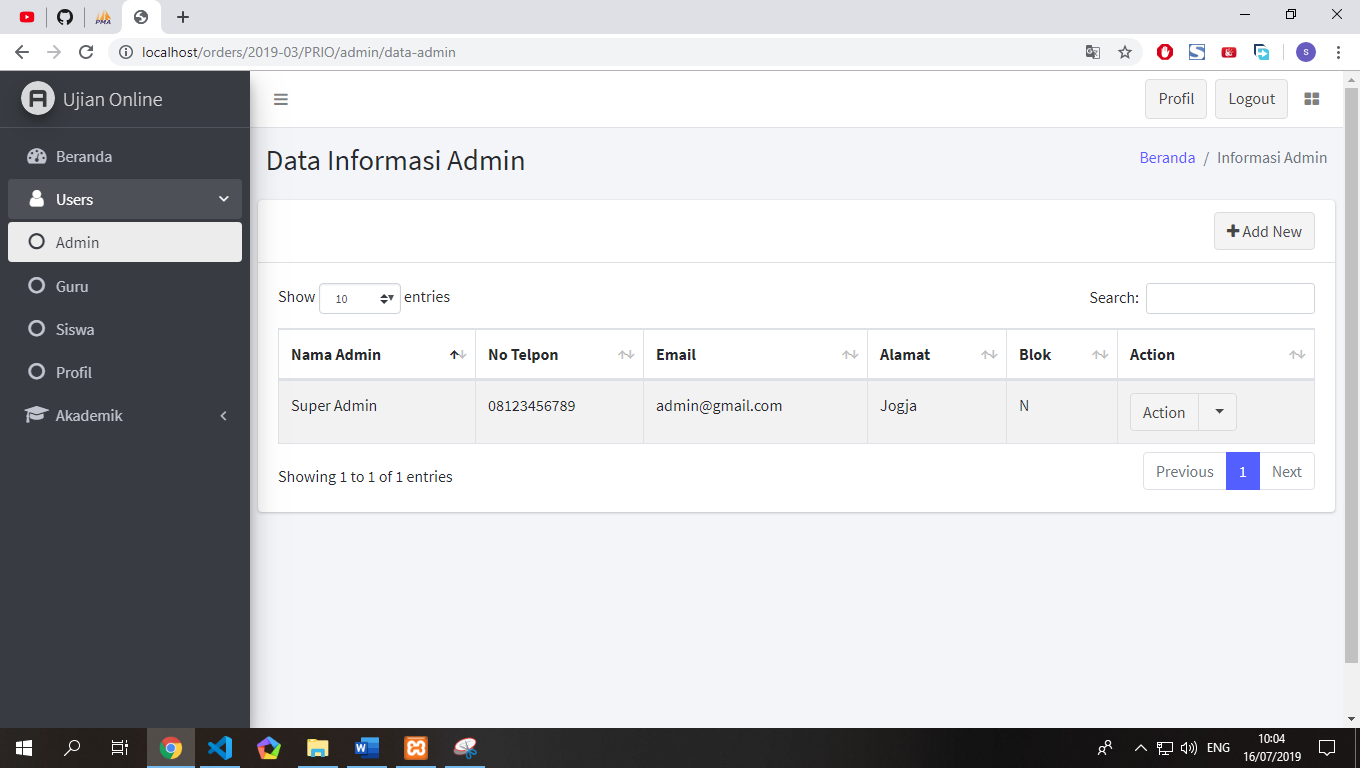
$this->view= 'admin/index';

$this->render\_pages();

}

**3. Halaman Admin Data Admin**

Halaman ini berisikan data admin. untuk lebih jelas dapat dilihat pada Gambar 5.4



**Gambar 5.4** Halaman Data Admin

Untuk mengetahui proses tampil data halaman admin lihat pada Tabel 5.3.

Tabel 5.3 Script Halaman Data Admin

public function data\_admin()

{

$this->content['rows']= $this->m\_admin->data\_admin();

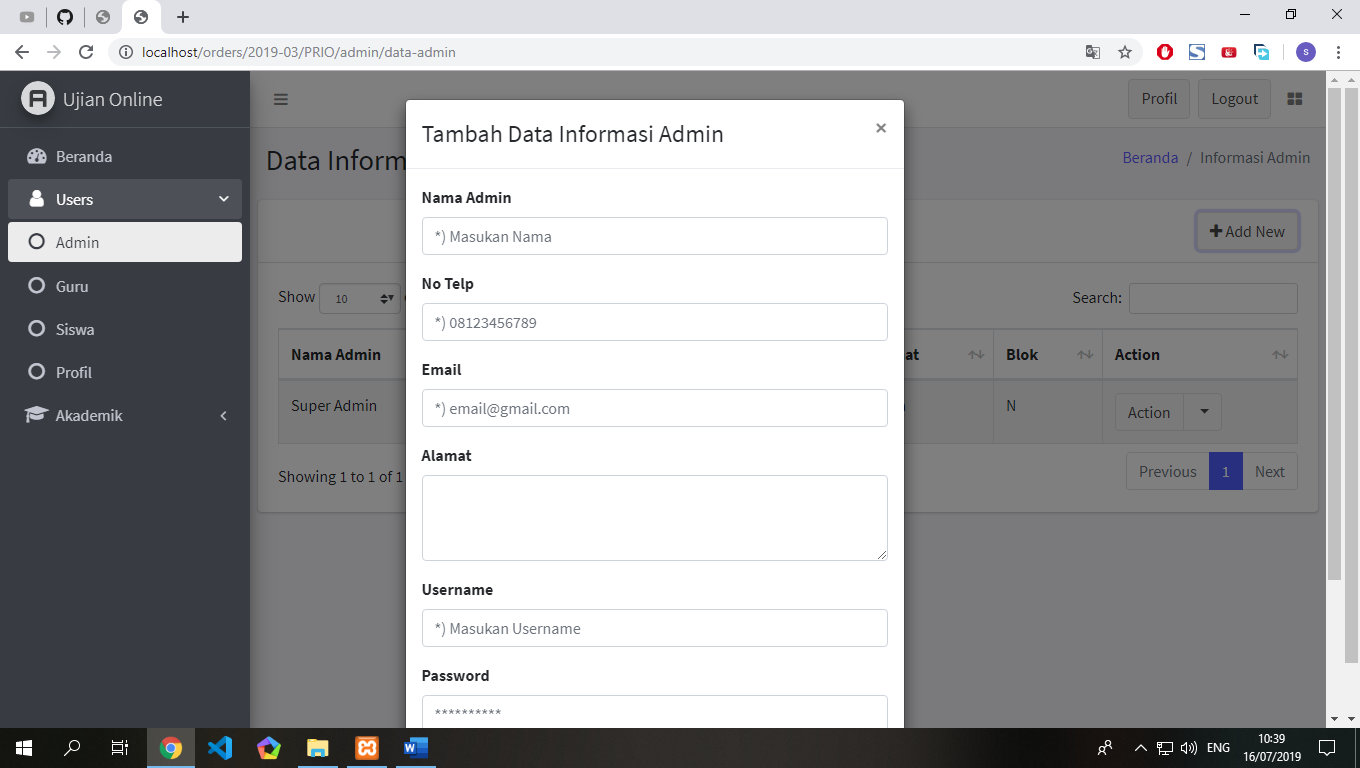
$this->view= 'admin/data\_admin';

$this->render\_pages();

}

**4. Halaman Admin Tambah Data Admin**

Halaman ini berfungsi untuk menampilkan informasi tambah data admin. Untuk lebih jelasnya dapat dilihat pada Gambar 5.5



**Gambar 5.5** Halaman Admin Tambah Data Admin

Untuk mengetahui proses simpan tambah data admin bisa dilihat pada Tabel 5.4.

Tabel 5.4 Script Proses Tambah Data Admin

public function data\_admin\_store()

{

$this->m\_admin->username= $this->input->post('username');

if ( $this->m\_admin->cek\_user() > 0 ) {

$this->msg= [

'stats'=> 0,

'msg'=> 'Maaf Username Sudah Digunakan',

];

} else {

$this->m\_admin->post= $this->input->post();

$this->m\_admin->post['level']= 'admin';

if ( $this->m\_admin->data\_admin\_store() ) {

$this->msg= [

'stats'=> 1,

'msg'=> 'Data Berhasil Ditambahkan',

];

} else {

$this->msg= [

'stats'=> 0,

'msg'=> 'Data Gagal Ditambahkan',

];

}

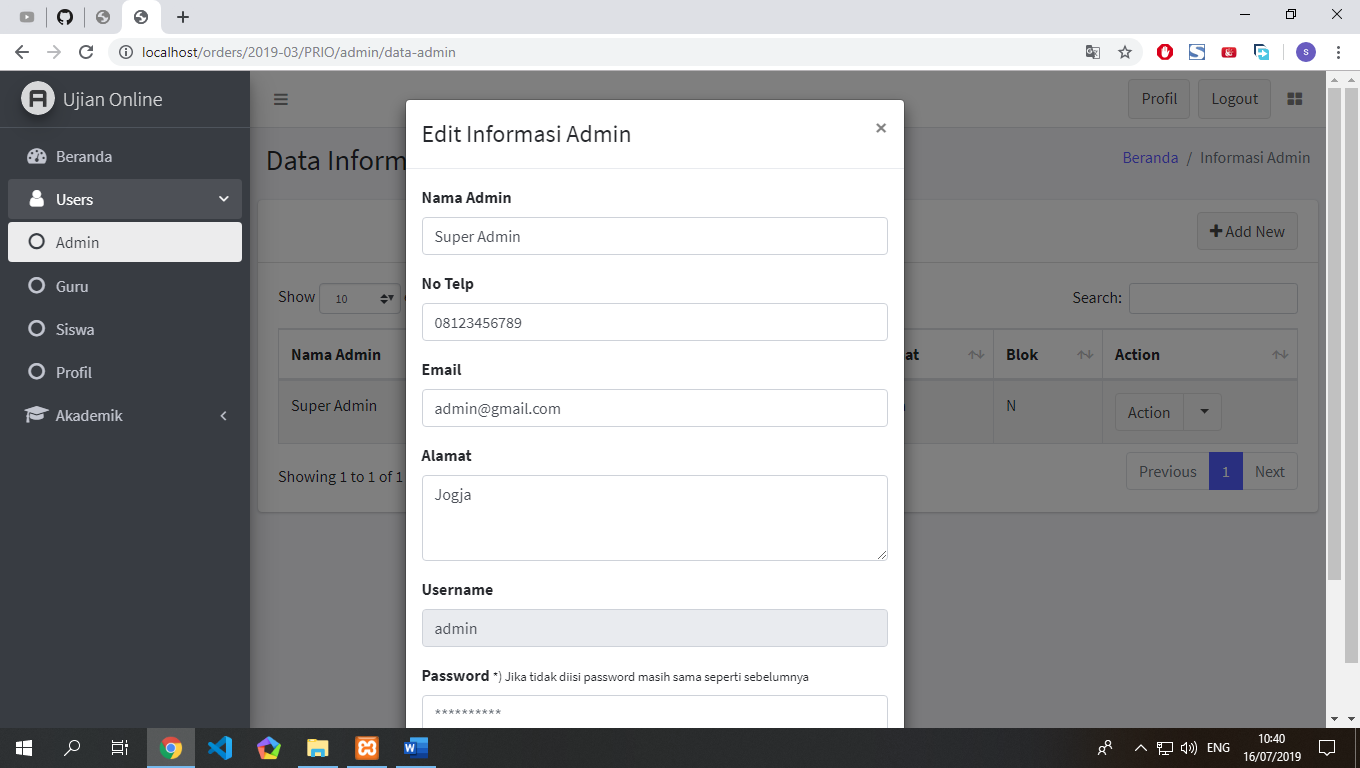
}

echo json\_encode($this->msg);

}

**5. Halaman Admin Edit Data Admin**

Halaman ini berfungsi untuk menampilkan informasi edit data admin. Untuk lebih jelasnya dapat dilihat pada Gambar 5.6



**Gambar 5.6** Halaman Admin Edit Data Admin

Untuk mengetahui proses update data admin dapat dilihat pada Tabel 5.5.

Tabel 5.5 Script Proses Update Data Admin

public function data\_admin\_update()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_admin\_update() ) {

$this->msg= [

'stats'=> 1,

'msg'=> 'Data Berhasil Diubah',

];

} else {

$this->msg= [

'stats'=> 0,

'msg'=> 'Data Gagal Diubah',

];

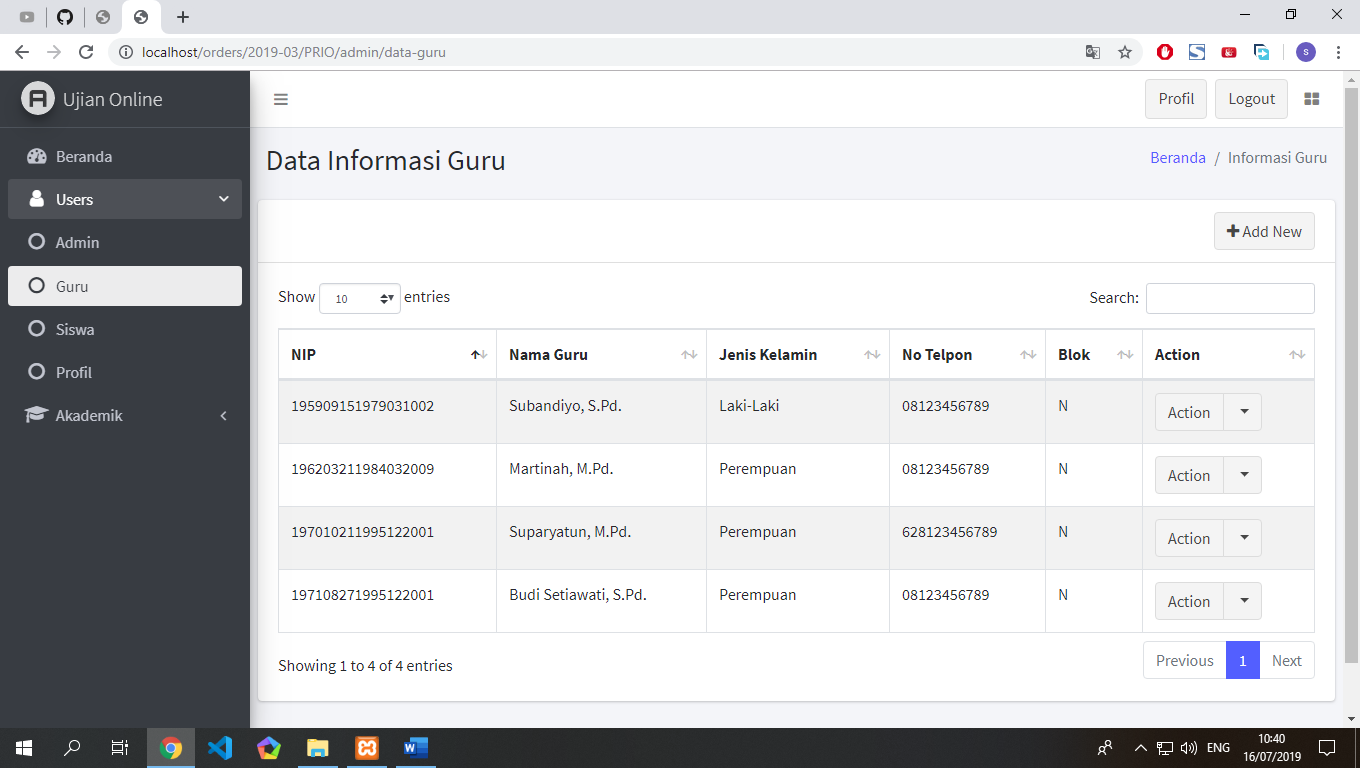
}

echo json\_encode($this->msg);

}

**6. Halaman Admin Data Guru**

Halaman ini berfungsi untuk menampilkan informasi data guru. Untuk lebih jelasnya dapat dilihat pada Gambar 5.7



**Gambar 5.7** Halaman Admin Data Guru

Untuk mengaetahui proses tampil data halaman admin data guru dapat dilihat pada Tabel 5.6.

Tabel 5.6 Script Data Halaman Admin Data Guru

public function data\_guru()

{

$this->content['rows']= $this->m\_admin->data\_guru();

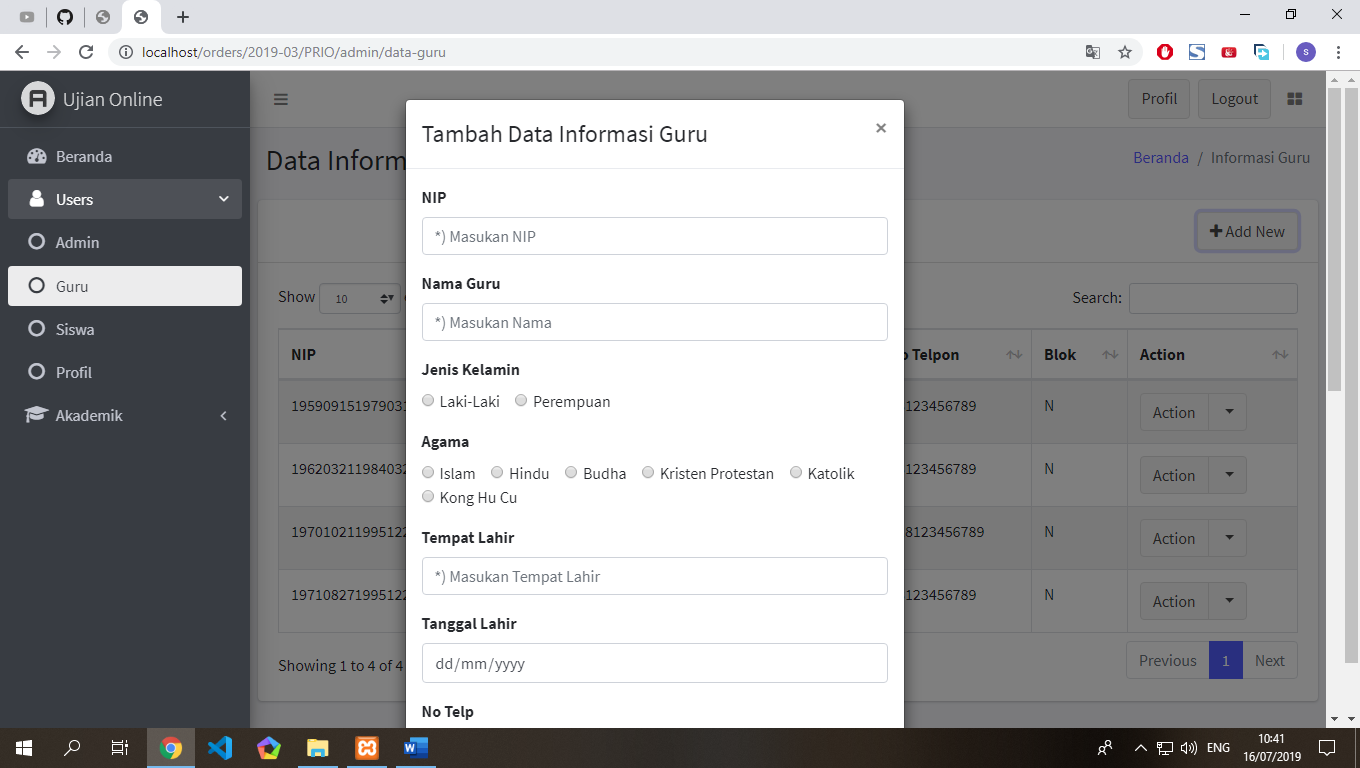
$this->view= 'admin/data\_guru';

$this->render\_pages();

}

**7. Halaman Admin Tambah Data Guru**

Halaman ini berfungsi untuk menampilkan informasi tambah data guru. Untuk lebih jelasnya dapat dilihat pada Gambar 5.8



**Gambar 5.8** Halaman Admin Tambah Data Guru

Untuk mengetahui proses simpan halaman admin tambah data guru dapat dilihat pada tabel 5.7.

Tabel 5.7 Script Proses Simpan Halaman Admin Data Guru

public function data\_guru\_store()

{

$this->m\_admin->username= $this->input->post('username');

if ( $this->m\_admin->cek\_user() > 0 ) {

$this->msg= [

'stats'=> 0,

'msg'=> 'Maaf Username Sudah Digunakan',

];

} else {

$config['upload\_path'] = 'src/guru/';

$config['allowed\_types'] = 'jpg|png';

// $config['max\_size'] = 1000000;

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('fupload'))

{

$this->msg= [

'stats'=>0,

'msg'=> $this->upload->display\_errors(),

];

}

else

{

$this->m\_admin->post= $this->input->post();

$this->m\_admin->post['level']= 'guru';

$this->m\_admin->post['gambar']=

$this->upload->data()['file\_name'];

if ( $this->m\_admin->data\_guru\_store() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=> 'Maaf Data Gagal Disimpan',

];

}

}

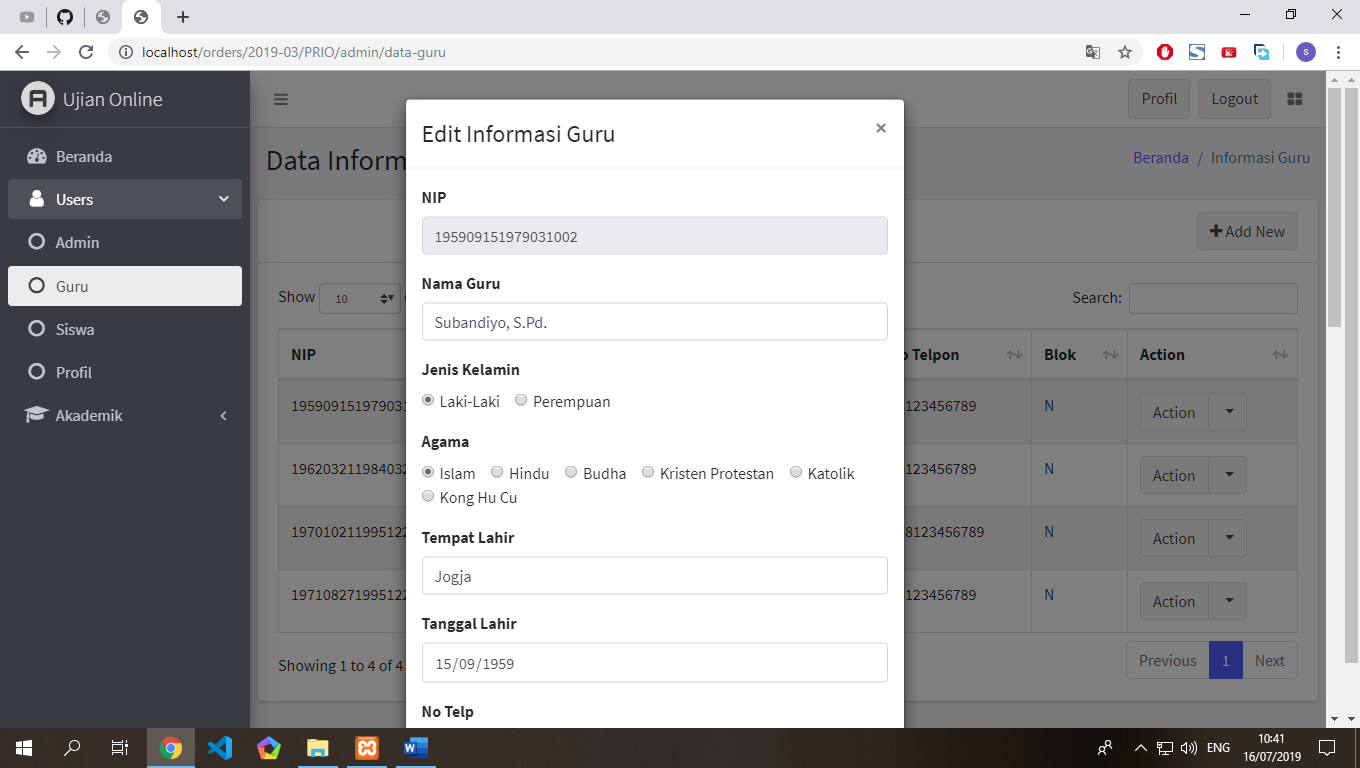
}

echo json\_encode($this->msg);

}

**8. Halaman Admin Edit Data Guru**

Halaman ini berfungsi untuk menampilkan informasi edit data guru. Untuk lebih jelasnya dapat dilihat pada Gambar 5.9



**Gambar 5.9** Halaman Admin Edit Data Guru

Untuk mengetahui proses update halaman admin edit data guru dapat dilihat pada Tabel 5.8.

Tabel 5.8 Script Proses Update Halaman Admin Edit Data Guru

public function data\_guru\_update()

{

$this->m\_admin->post= $this->input->post();

if ( empty($\_FILES['fupload']['tmp\_name']) ) {

# code...without upload file

if ( $this->m\_admin->data\_guru\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan'

];

} else {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Gagal Disimpan'

];

}

} else {

# code...with upload file

$this->m\_admin->username=

$this->input->post('username');

$row= $this->m\_admin->data\_guru\_edit();

$config['upload\_path'] = 'src/guru/';

$config['allowed\_types'] = 'jpg|png';

if ( file\_exists($config['upload\_path'].$row->gambar) ) {

unlink($config['upload\_path'].$row->gambar);

}

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('fupload'))

{

$this->msg= [

'stats'=>0,

'msg'=> $this->upload->display\_errors(),

];

}

else

{

$this->m\_admin->post['gambar']=

$this->upload->data()['file\_name'];

if ( $this->m\_admin->data\_guru\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=> 'Maaf Data Gagal Disimpan',

];

}

}

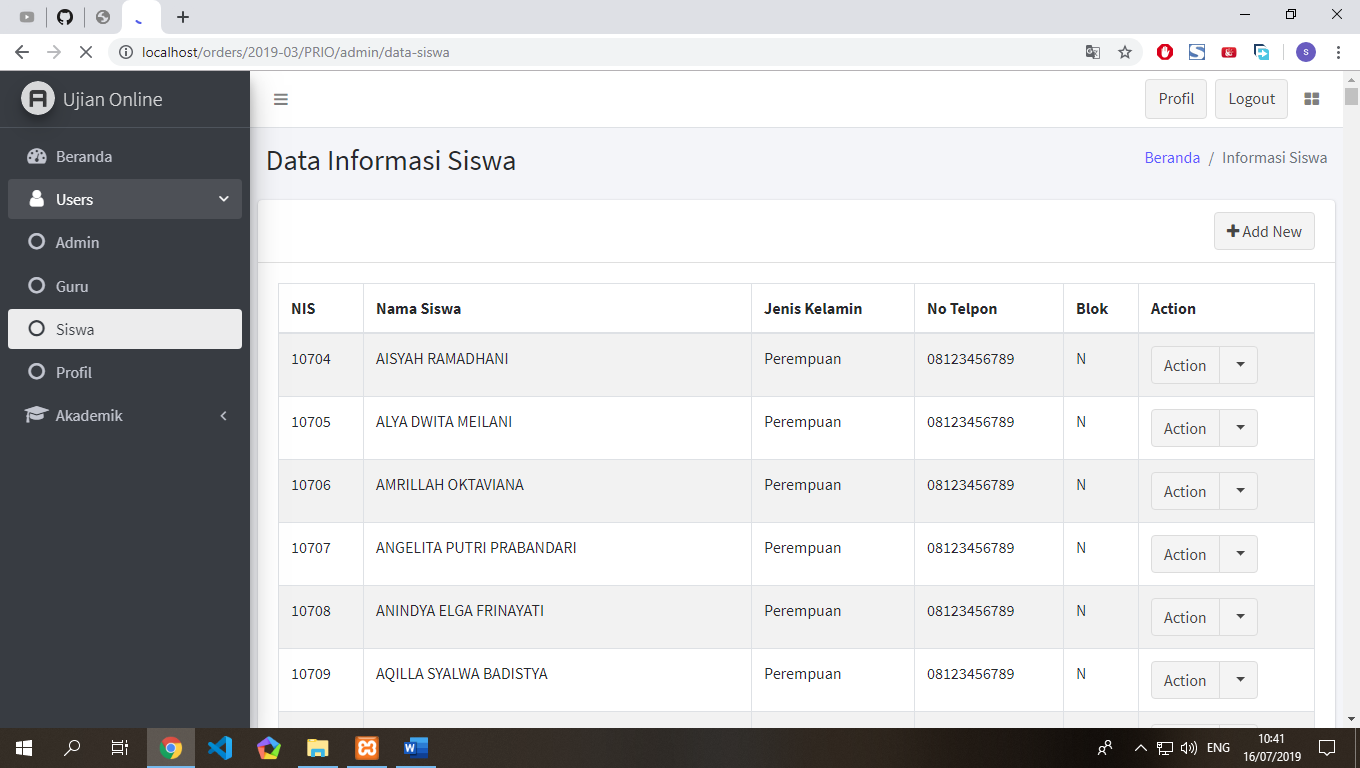
}

echo json\_encode($this->msg);

}

**9. Halaman Admin Data Siswa**

Halaman ini berfungsi untuk menampilkan informasi data siswa. Untuk lebih jelasnya dapat dilihat pada Gambar 5.10



**Gambar 5.10** Halaman Admin Data Siswa

Untuk mengetahui proses halaman admin data siswa dapat dilihat pada Tabel 5.9

Tabel 5.9 Script Proses Halaman Admin Data Siswa

public function data\_siswa()

{

$this->content['rows']= $this->m\_admin->data\_siswa();

$this->view= 'admin/data\_siswa';

$this->render\_pages();

}

public function data\_siswa()

{

switch ($this->session->userdata('level')) {

case 'admin':

# code...

return $this->db->query("

SELECT \*,

IF(siswa.jk='L','Laki-Laki','Perempuan') AS gender

FROM siswa

LEFT JOIN users

ON siswa.username=users.username

")->result\_object();

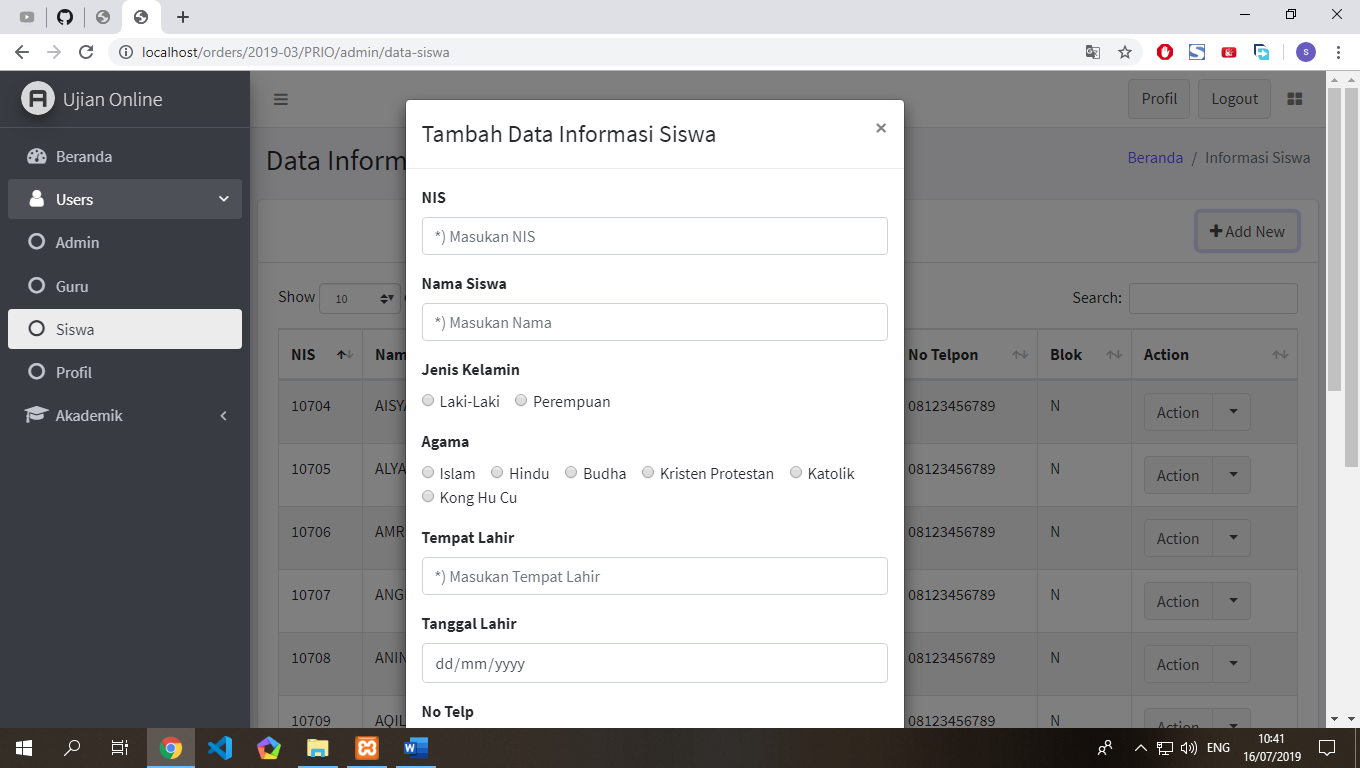
break;

}

}

**10. Halaman Admin Tambah Data Siswa**

Halaman ini berfungsi untuk menampilkan informasi tambah data siswa. Untuk lebih jelasnya dapat dilihat pada Gambar 5.11



**Gambar 5.11** Halaman Admin Tambah Data Siswa

Untuk mengetahui proses halaman admin tambah data siswa dapat dilihat pada Tabel 5.10

Tabel 5.10 Script Proses Halaman Admin Tambah Data Siswa

public function data\_siswa\_store()

{

$this->m\_admin->username= $this->input->post('username');

if ( $this->m\_admin->cek\_user() > 0 ) {

$this->msg= [

'stats'=> 0,

'msg'=> 'Maaf Username Sudah Digunakan',

];

} else {

$config['upload\_path'] = 'src/siswa/';

$config['allowed\_types'] = 'jpg|png';

// $config['max\_size'] = 1000000;

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('fupload'))

{

$this->msg= [

'stats'=>0,

'msg'=> $this->upload->display\_errors(),

];

}

else

{

$this->m\_admin->post= $this->input->post();

$this->m\_admin->post['level']= 'siswa';

$this->m\_admin->post['gambar']=

$this->upload->data()['file\_name'];

if ( $this->m\_admin->data\_siswa\_store() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=> 'Maaf Data Gagal Disimpan',

];

}

}

}

echo json\_encode($this->msg);

}

public function data\_siswa\_store()

{

$admin\_store= $this->db->insert('siswa', [

'nis' => $this->post['nis'],

'username' => $this->post['username'],

'nama' => $this->post['nama'],

'alamat' => $this->post['alamat'],

'tempat\_lahir' => $this->post['tempat\_lahir'],

'tgl\_lahir' => $this->post['tgl\_lahir'],

'agama' => $this->post['agama'],

'no\_telp' => $this->post['telp'],

'email' => $this->post['email'],

'gambar' => $this->post['gambar'],

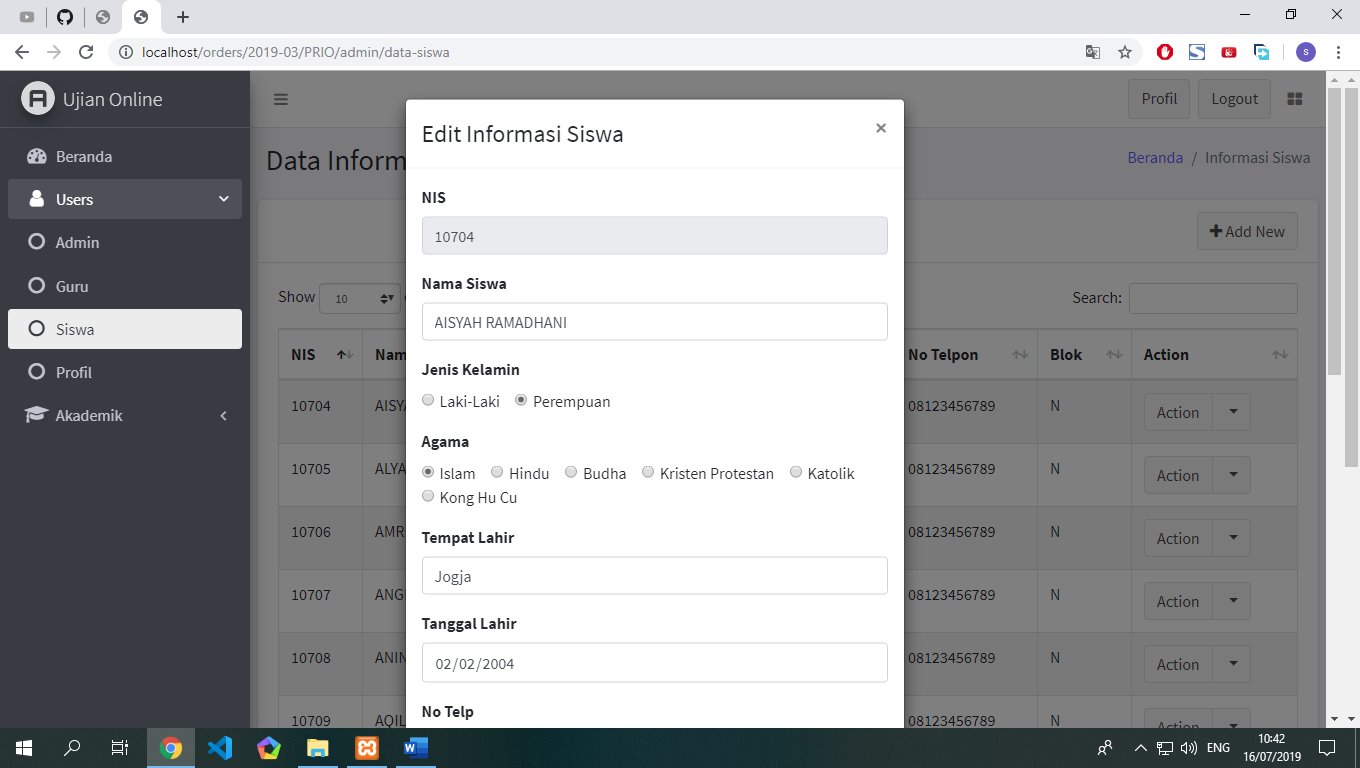
'jk' => $this->post['jk'],

]);

return TRUE; }

**11. Halaman Admin Edit Data Siswa**

Halaman ini berfungsi untuk menampilkan informasi edit data siswa. Untuk lebih jelasnya dapat dilihat pada Gambar 5.12



**Gambar 5.12** Halaman Admin Edit Data Siswa

Untuk mengetahui proses halaman admin edit data siswa dapat dilihat pada Tabel 5.11.

Tabel 5.11 Script Halaman Admin Edit Data Siswa

public function data\_siswa\_update()

{

$this->m\_admin->post= $this->input->post();

if ( empty($\_FILES['fupload']['tmp\_name']) ) {

# code...without upload file

if ( $this->m\_admin->data\_siswa\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan'

];

} else {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Gagal Disimpan'

];

}

} else {

# code...with upload file

$this->m\_admin->username= $this->input->post('username');

$row= $this->m\_admin->data\_siswa\_edit();

$config['upload\_path'] = 'src/siswa/';

$config['allowed\_types'] = 'jpg|png';

if ( file\_exists($config['upload\_path'].$row->gambar) ) {

unlink($config['upload\_path'].$row->gambar);

}

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('fupload'))

{

$this->msg= [

'stats'=>0,

'msg'=> $this->upload->display\_errors(),

];

}

else

{

$this->m\_admin->post['gambar']= $this->upload->data()['file\_name'];

if ( $this->m\_admin->data\_siswa\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=> 'Maaf Data Gagal Disimpan',

];

}

}

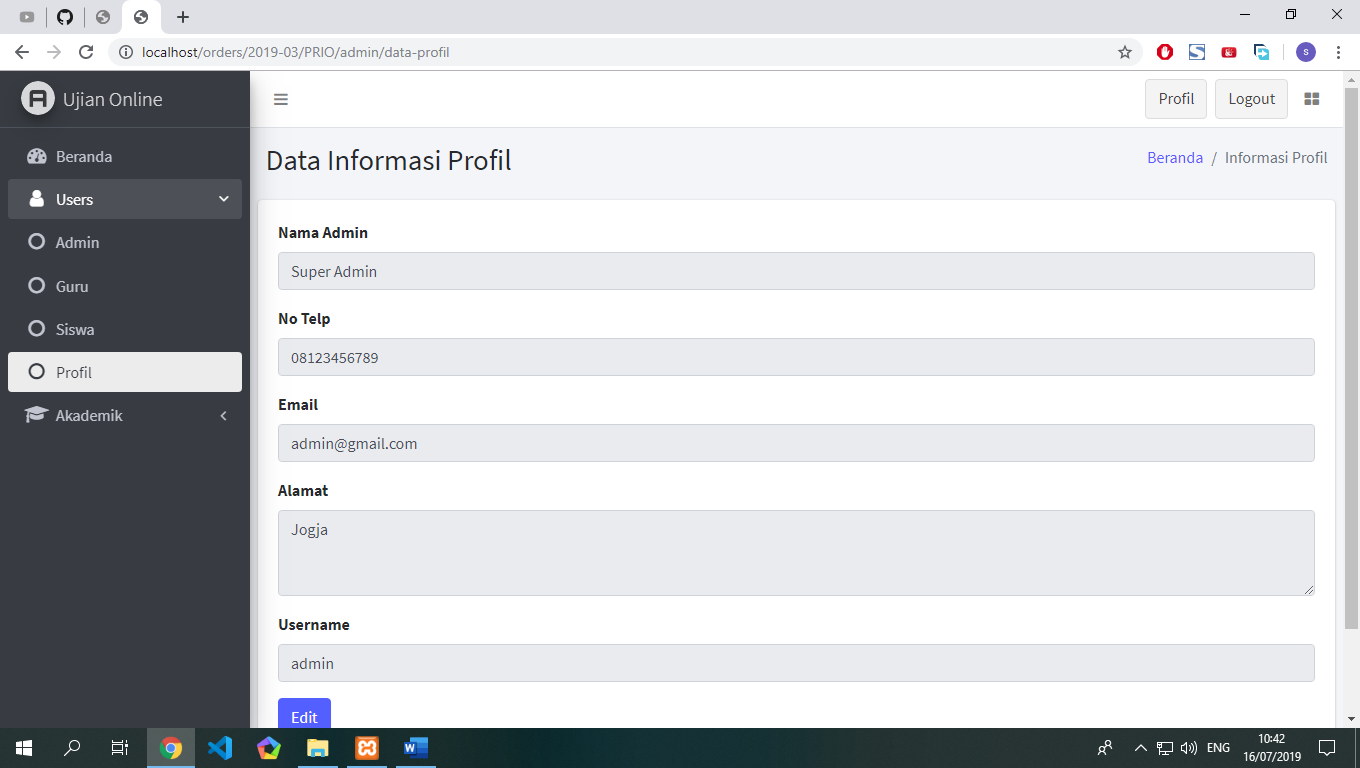
}

echo json\_encode($this->msg);

}

**12. Halaman Admin Data Profil**

Halaman ini berfungsi untuk menampilkan informasi edit data profil admin. Untuk lebih jelasnya dapat dilihat pada gambar 5.13



**Gambar 5.13** Halaman Admin Data Profil

Untuk mengetahui proses halaman admiin data profil dapat dilihat pada Tabel 5.12.

Tabel 5.12 Script Halaman Admin Data Profil

public function data\_admin\_update()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_admin\_update() ) {

$this->msg= [

'stats'=> 1,

'msg'=> 'Data Berhasil Diubah',

];

} else {

$this->msg= [

'stats'=> 0,

'msg'=> 'Data Gagal Diubah',

];

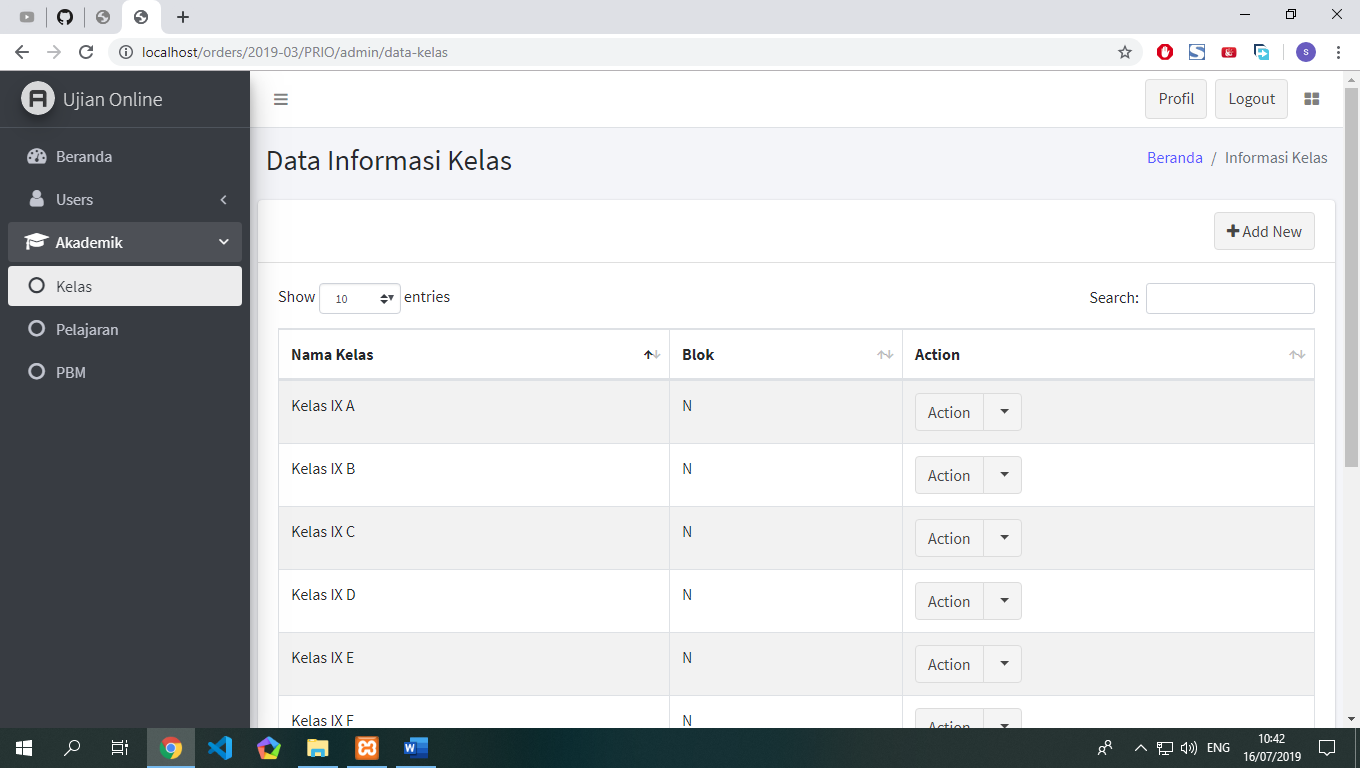
}

echo json\_encode($this->msg);

}

**13. Halaman Admin Data Kelas**

Halaman ini berfungsi untuk menampilkan informasi data kelas. Untuk lebih jelasnya dapat dilihat pada Gambar 5.14



**Gambar 5.14** Halaman Admin Data Kelas

Untuk mengetahui script proses halaman admin data kelas dapat dilihat pada Tabel 5.13.

Tabel 5.13 Script halaman Admin Data Kelas

public function data\_kelas()

{

$this->content['rows']= $this->m\_admin->data\_kelas();

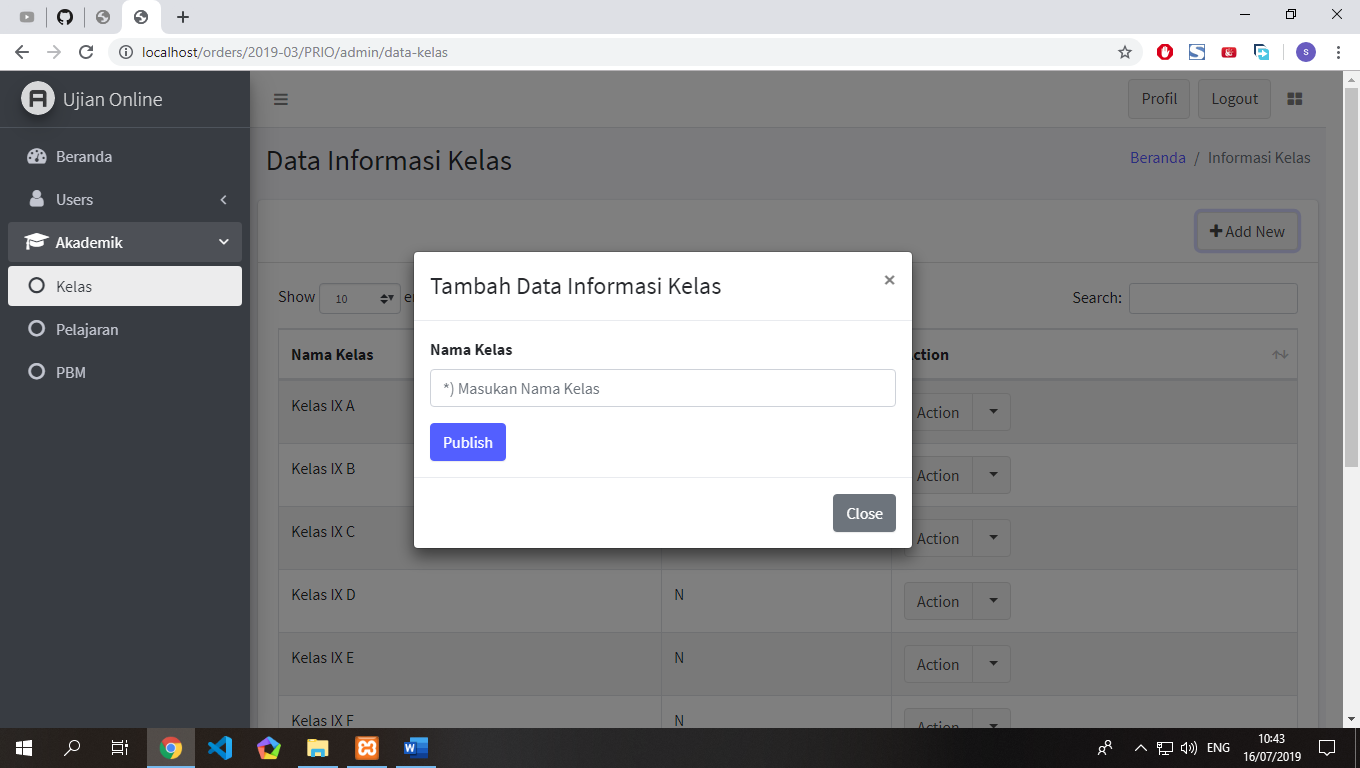
$this->view= 'admin/data\_kelas';

$this->render\_pages();

}

**14. Halaman Admin Tambah Data Kelas**

Halaman ini berfungsi untuk menampilkan informasi tambah data kelas. Untuk lebih jelasnya dapat dilihat pada Gambar 5.15



**Gambar 5.15** Halaman Admin Tambah Data Kelas

Untuk mengetahui script proses halaman admin tambah data kelas dapat dilihat pada Tabel 5.14.

Tabel 5.14 Script Halaman Admin Tambah Data Kelas

public function data\_kelas\_store()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_kelas\_store() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

} else {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

}

echo json\_encode($this->msg);

}

**15. Halaman Admin Edit Data Kelas**

Halaman ini berfungsi untuk menampilkan informasi tambah data siswa. Untuk lebih jelasnya dapat dilihat pada Gambar 5.16



**Gambar 5.16** Halaman Admin Edit Data Kelas

Untuk mengetahui script proses halaman admin edit data kelas dapat dilihat pada Tabel 5.15.

Tabel 5.15 Script Halaman Admin Edit Data Kelas

public function data\_kelas\_update()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_kelas\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Diubah',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=>'Data Gagal Diubah',

];

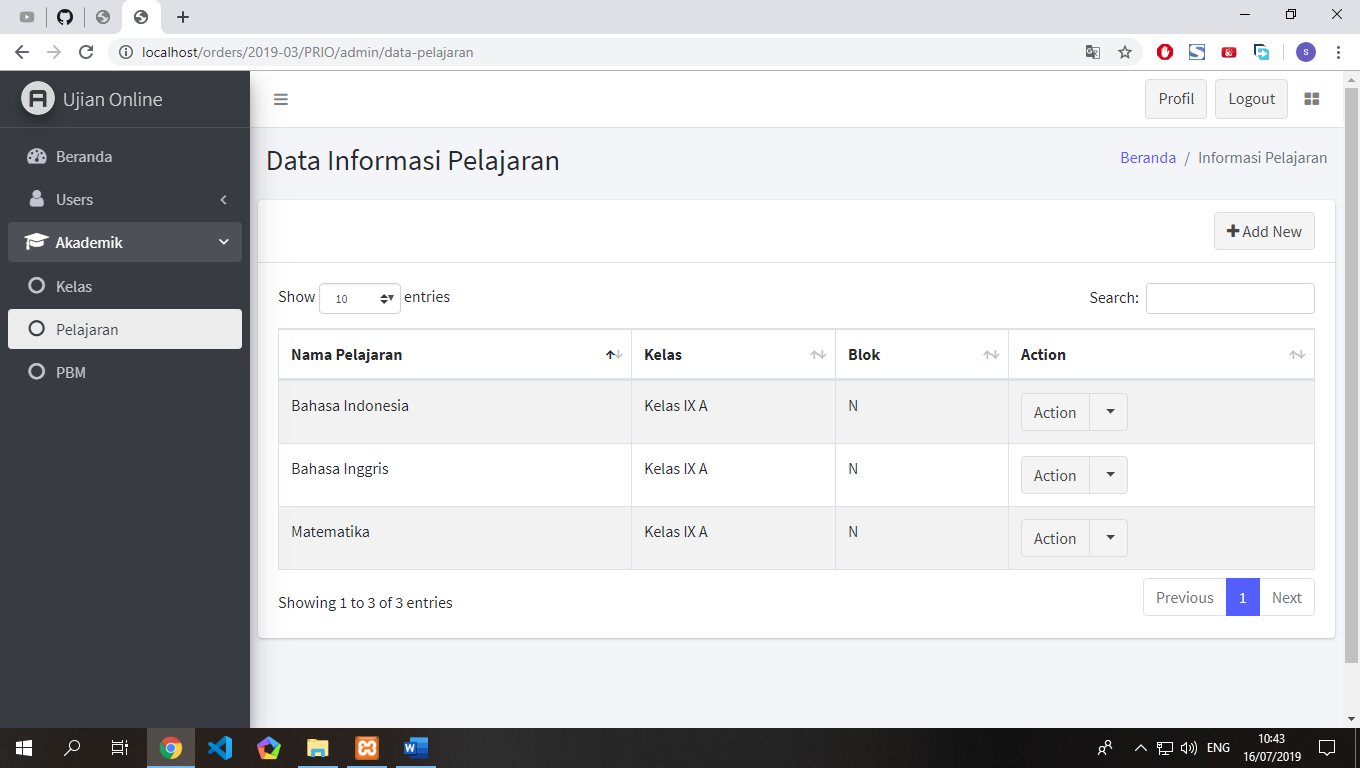
}

echo json\_encode($this->msg);

}

**16. Halaman Admin Data Pelajaran**

Halaman ini berfungsi untuk menampilkan informasi data pelajaran. Untuk lebih jelasnya dapat dilihat pada Gambar 5.17



**Gambar 5.17** Halaman Admin Data Pelajaran

Untuk mengetahui script proses halaman admin data pelajaran dapat dilihat pada Tabel 5.16.

Tabel 5.16 Script Halaman Admin Data Pelajaran

public function data\_pelajaran()

{

$this->content['rows']= $this->m\_admin->data\_pelajaran();

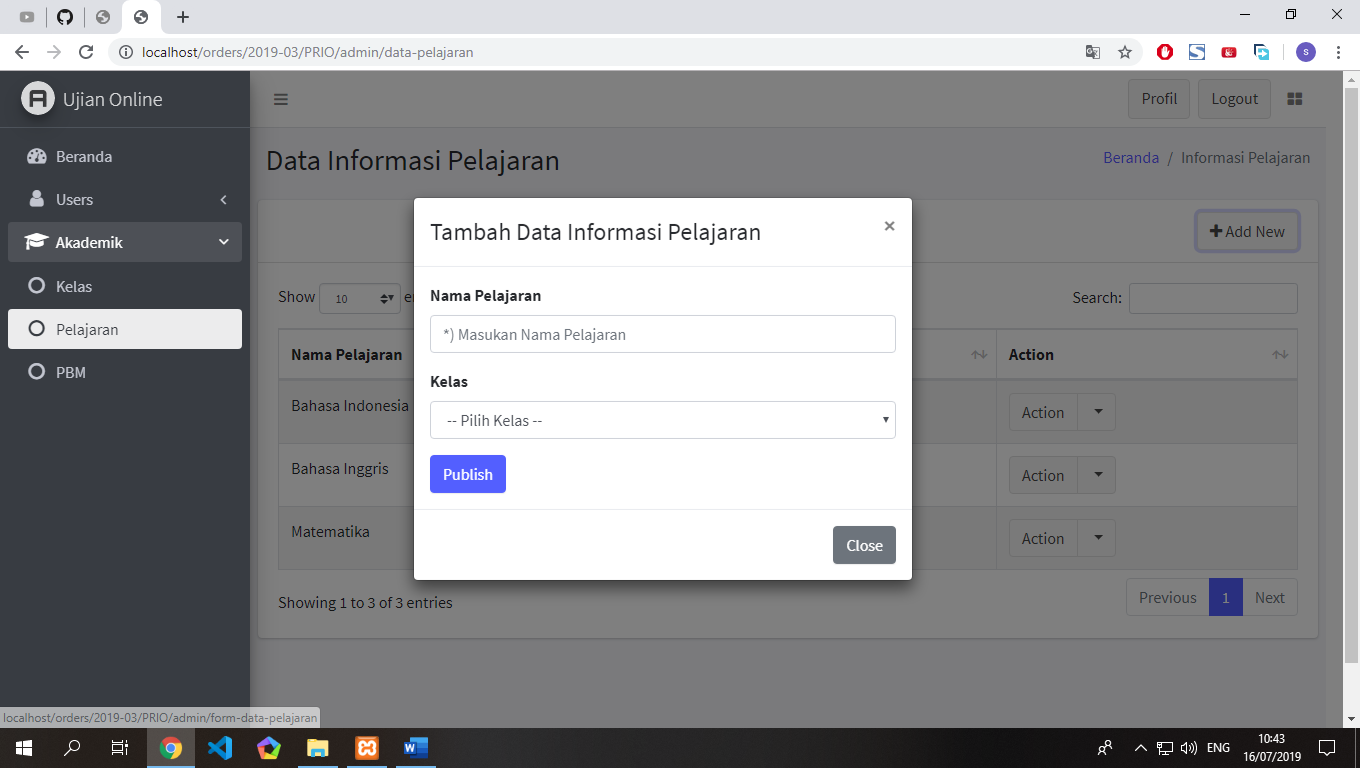
$this->view= 'admin/data\_pelajaran';

$this->render\_pages();

}

**17. Halaman Admin Tambah Data Pelajaran**

Halaman ini berfungsi untuk menampilkan informasi tambah data pelajaran. Untuk lebih jelasnya dapat dilihat pada Gambar 5.18



**Gambar 5.18** Halaman Admin Tambah Data Pelajaran

Untuk mengetahui script proses halaman admin tambah data pelajaran dapat dilihat pada Tabel 5.17.

Tabel 5.17 Script Halaman Admin Tambah Data Pelajaran

public function data\_pelajaran\_store()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_pelajaran\_store() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

} else {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

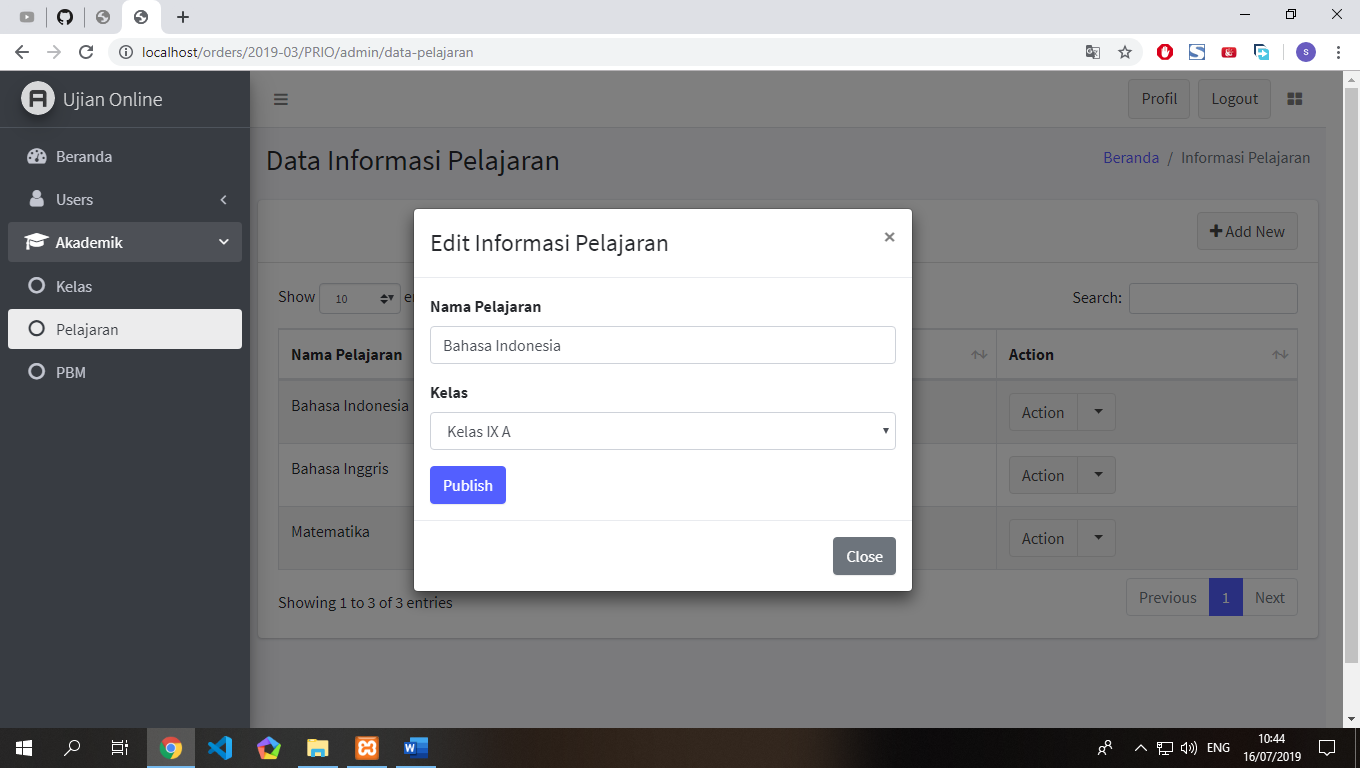
}

echo json\_encode($this->msg);

}

**18. Halaman Admin Edit Data Pelajaran**

Halaman ini berfungsi untuk menampilkan informasi edit data pelajaran. Untuk lebih jelasnya dapat dilihat pada Gambar 5.19



**Gambar 5.19** Halaman Admin Edit Data Pelajaran

Untuk mengetahui script proses halaman admin edit data pelajaran dapat dilihat pada Tabel 5.18.

Tabel 5.18 Script Halaman Admin Edit Data Pelajaran

public function data\_pelajaran\_update()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_pelajaran\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Diubah',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=>'Data Gagal Diubah',

];

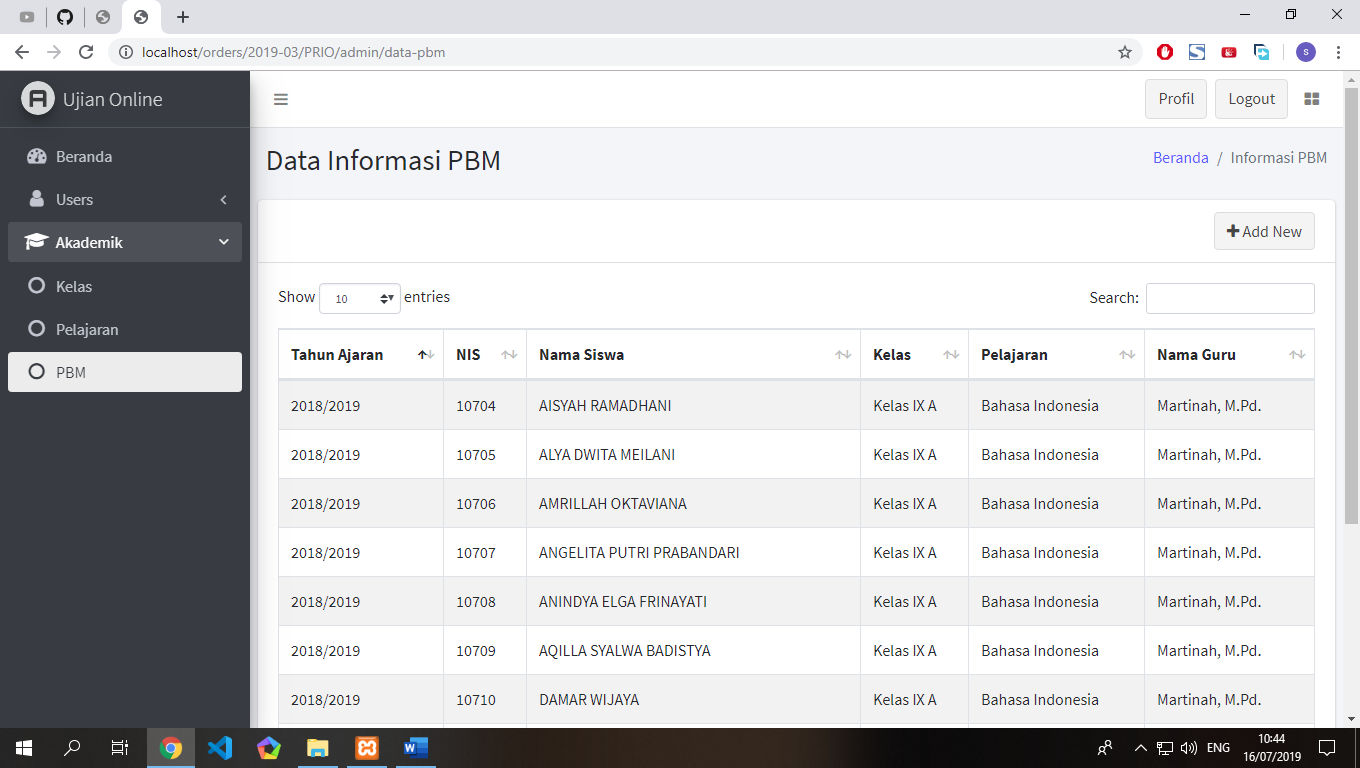
}

echo json\_encode($this->msg);

}

**19. Halaman Admin Data PBM**

Halaman ini berfungsi untuk menampilkan informasi data PBM(proses belajar mengajar). Untuk lebih jelasnya dapat dilihat pada Gambar 5.20



**Gambar 5.20** Halaman Admin Data PBM

Untuk mengetahui script proses halaman admin data PBM dapat dilihat pada Tabel 5.19.

Tabel 5.19 Script Halaman Admin Data PBM

public function data\_pbm()

{

$this->content['rows']= $this->m\_admin->data\_pbm();

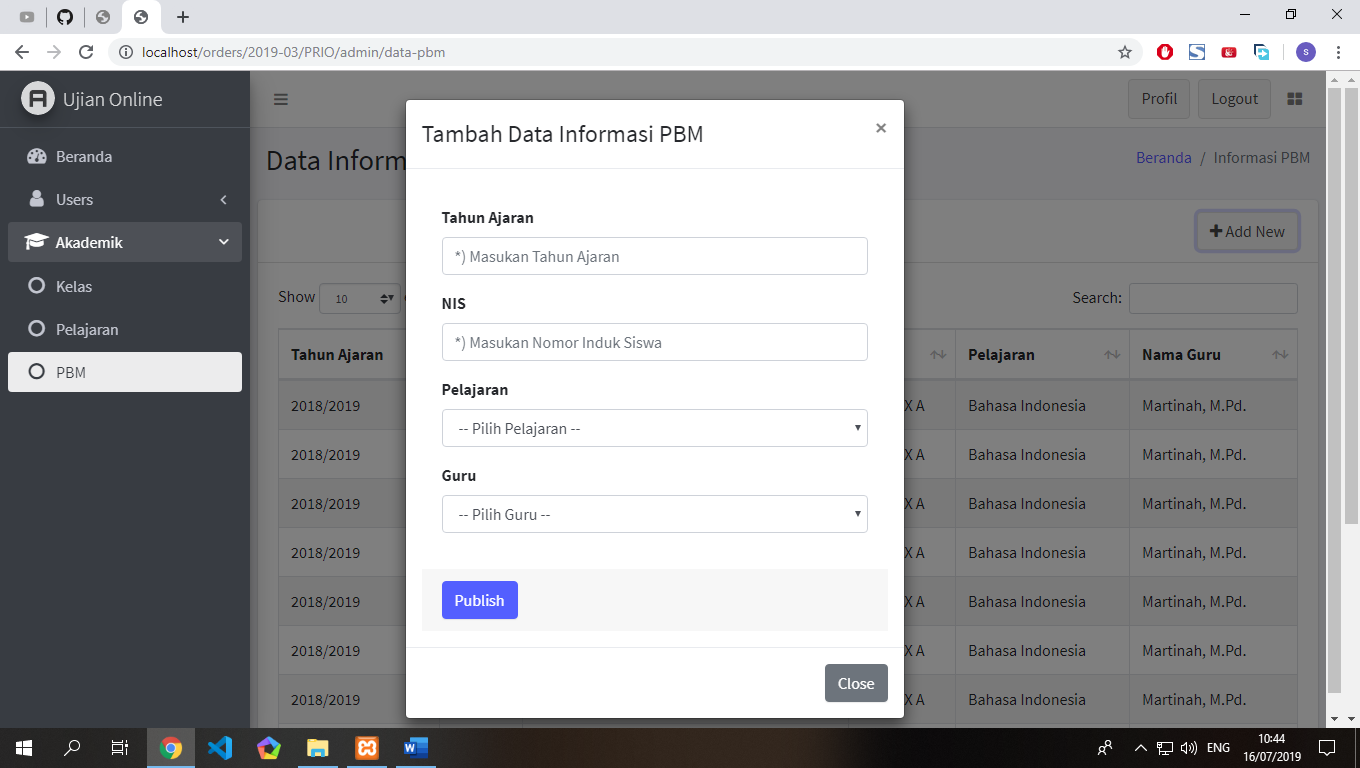
$this->view= 'admin/data\_pbm';

$this->render\_pages();

}

**20. Halaman Admin Tambah Data PBM**

Halaman ini berfungsi untuk menampilkan informasi tambah data siswa. Untuk lebih jelasnya dapat dilihat pada Gambar 5.21



**Gambar 5.21** Halaman Admin Tambah Data PBM

Untuk mengetahui script proses halaman admin tambah data PBM dapat dilihat pada Tabel 5.20.

Tabel 5.20 Script Halaman Admin Tambah Data PBM

public function data\_pbm\_store()

{

$this->m\_admin->post= $this->input->post();

if ( $this->m\_admin->data\_pbm\_cek\_nis()

->num\_rows() > 0 ) {

$this->m\_admin->post['nis']=

$this->m\_admin->data\_pbm\_cek\_nis()->row()->nis;

if ( $this->m\_admin->data\_pbm\_store() ) {

# code...

$this->msg=[

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

# code...

$this->msg=[

'stats'=>0,

'msg'=> 'Data Gagal Disimpan',

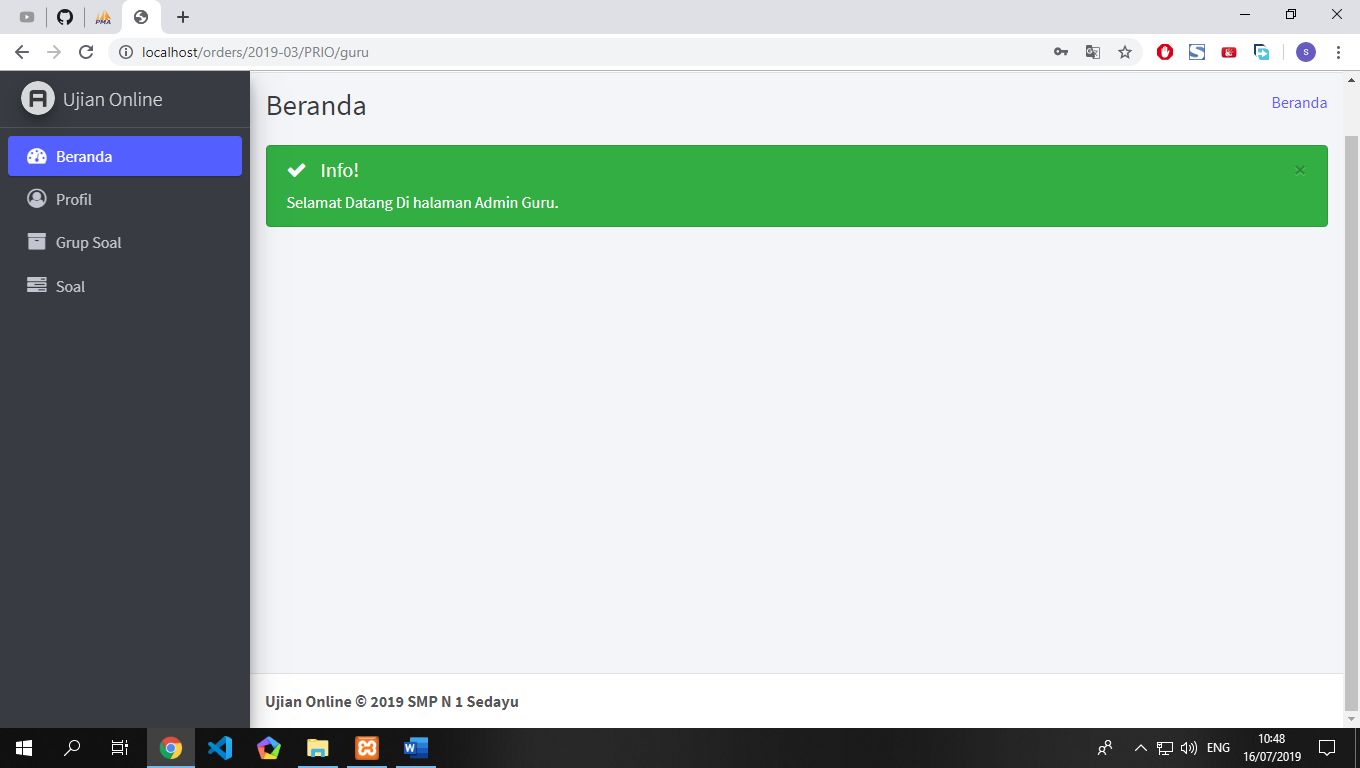
];

}

}

**21. Halaman Guru Beranda**

Halaman ini berisikan menu yang terdapat di halaman guru seperti data profil, grup soal dan soal. untuk lebih jelas dapat dilihat pada Gambar 5.22



**Gambar 5.22** Halaman Guru Beranda

Untuk mengetahui script proses Halaman Guru Beranda dapat dilihat pada Tabel 5.21.

Tabel 5.21 Script Halaman Guru Beranda

public function index()

{

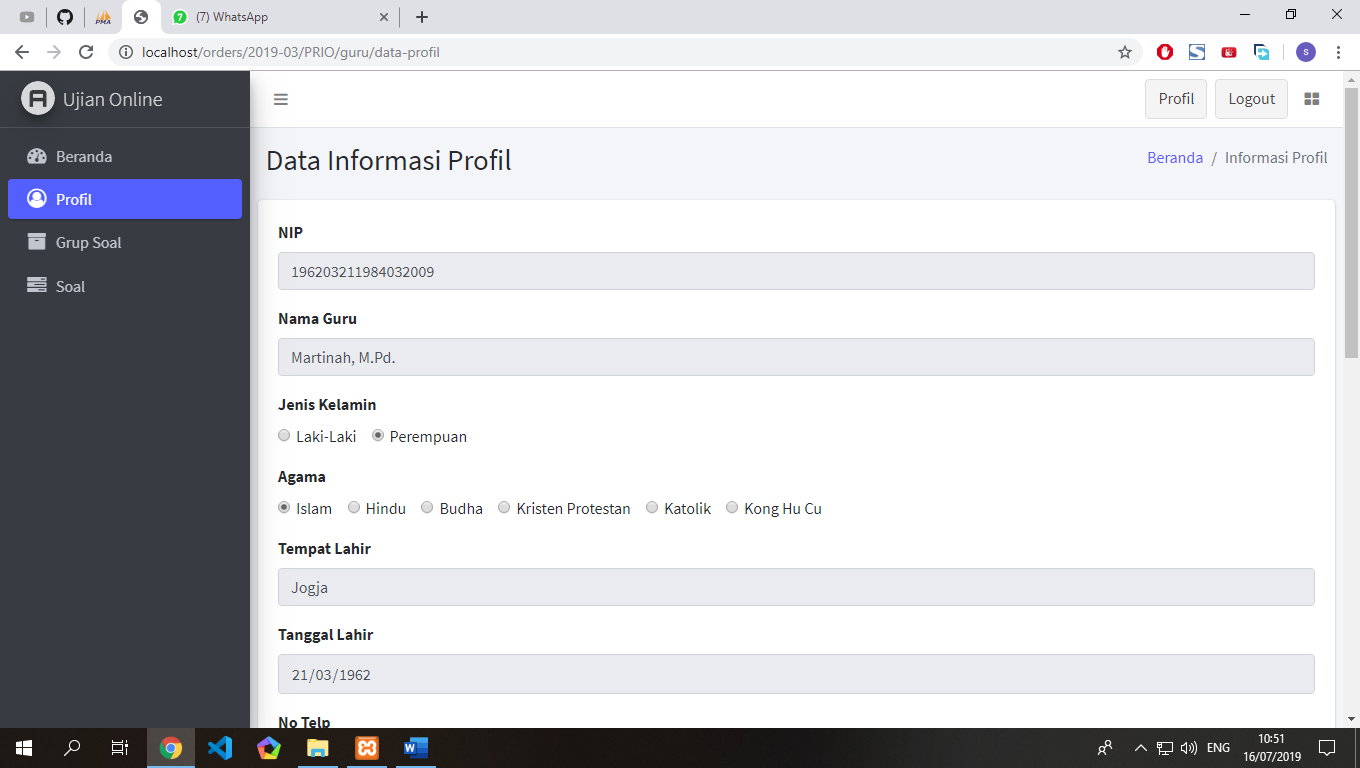
$this->view= 'guru/index';

$this->render\_pages();

}

**22. Halaman Guru Data Profil**

Halaman ini berfungsi untuk menampilkan informasi data profil guru. Untuk lebih jelasnya dapat dilihat pada Gambar 5.23



**Gambar 5.23** Halaman Guru Data Profil

Untuk mengetahui script proses Halaman Guru Data Profil dapat dilihat pada Tabel 5.22.

Tabel 5.22 Script Halaman Guru Data Profil

public function data\_guru\_update()

{

$this->m\_guru->post= $this->input->post();

if ( empty($\_FILES['fupload']['tmp\_name']) ) {

# code...without upload file

if ( $this->m\_guru->data\_guru\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan'

];

} else {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Gagal Disimpan'

];

}

} else {

# code...with upload file

$this->m\_guru->username= $this->input->post('username');

$row= $this->m\_guru->data\_guru\_edit();

$config['upload\_path'] = 'src/guru/';

$config['allowed\_types'] = 'jpg|png';

if ( file\_exists($config['upload\_path'].$row->gambar) ) {

unlink($config['upload\_path'].$row->gambar);

}

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('fupload'))

{

$this->msg= [

'stats'=>0,

'msg'=> $this->upload->display\_errors(),

];

}

else

{

$this->m\_guru->post['gambar']=

$this->upload->data()['file\_name'];

if ( $this->m\_guru->data\_guru\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=> 'Maaf Data Gagal Disimpan',

];

}

}

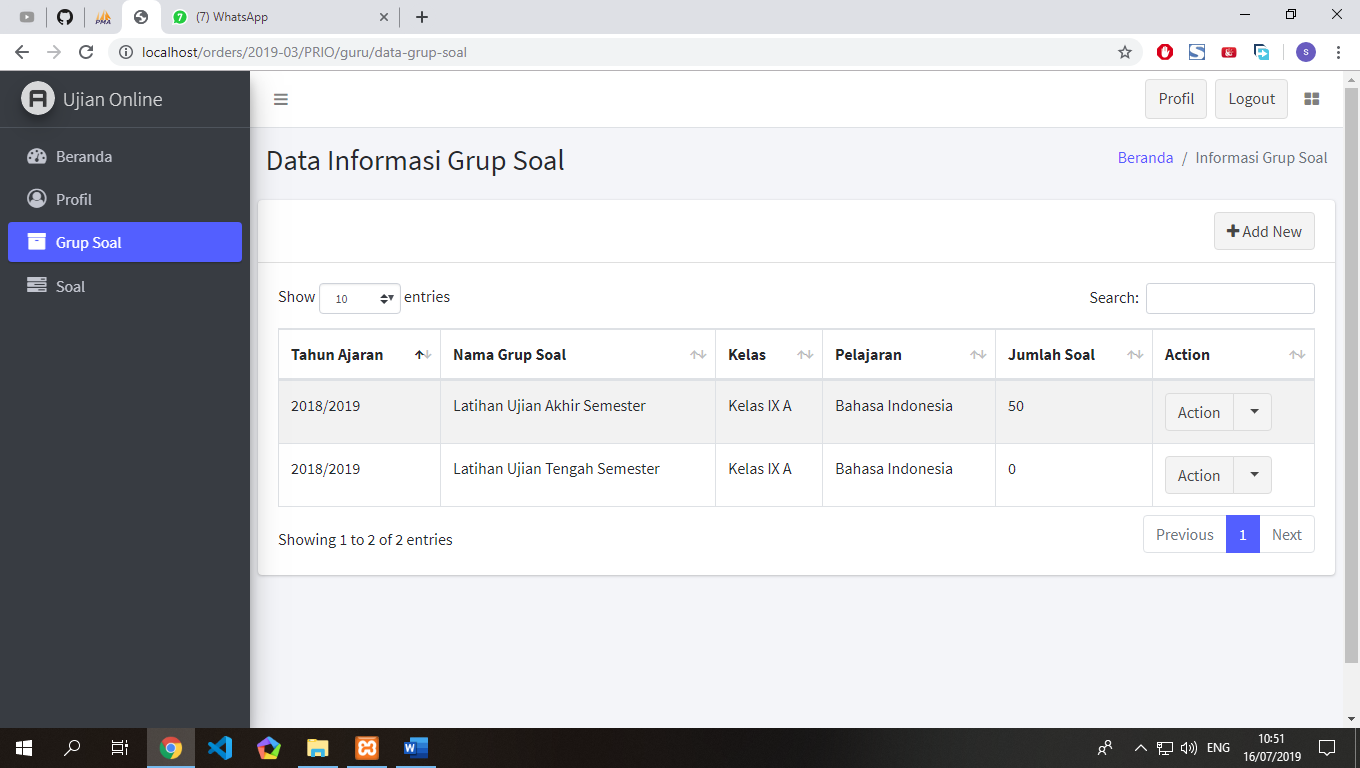
}

echo json\_encode($this->msg);

}

**23. Halaman Guru Data Grup Soal**

Halaman ini berfungsi untuk menampilkan informasi data grup soal. Untuk lebih jelasnya dapat dilihat pada Gambar 5.24



**Gambar 5.24** Halaman Guru Data Grup Soal

Untuk mengetahui script proses Halaman Guru Data Grup Soal dapat dilihat pada Tabel 5.23.

Tabel 5.23 Script Halaman Guru Data Grup Soal

public function data\_grup\_soal()

{

$this->view= 'guru/data\_grup\_soal';

$this->content['rows']= $this->m\_guru->data\_grup\_soal();

$this->render\_pages();

}

public function data\_grup\_soal()

{

return $this->db->query("

SELECT \*,

( (SELECT COUNT(id\_soal) FROM soal WHERE soal.id\_grup\_soal=grup\_soal.id\_grup\_soal) ) AS jumlah\_soal

FROM grup\_soal

LEFT JOIN pbm

ON grup\_soal.id\_pelajaran=pbm.id\_pelajaran

LEFT JOIN pelajaran

ON pbm.id\_pelajaran=pelajaran.id\_pelajaran

LEFT JOIN kelas

ON pelajaran.id\_kelas=kelas.id\_kelas

WHERE 1=1

AND pbm.nip='{$this->session->userdata('username')}'

GROUP BY grup\_soal.id\_grup\_soal

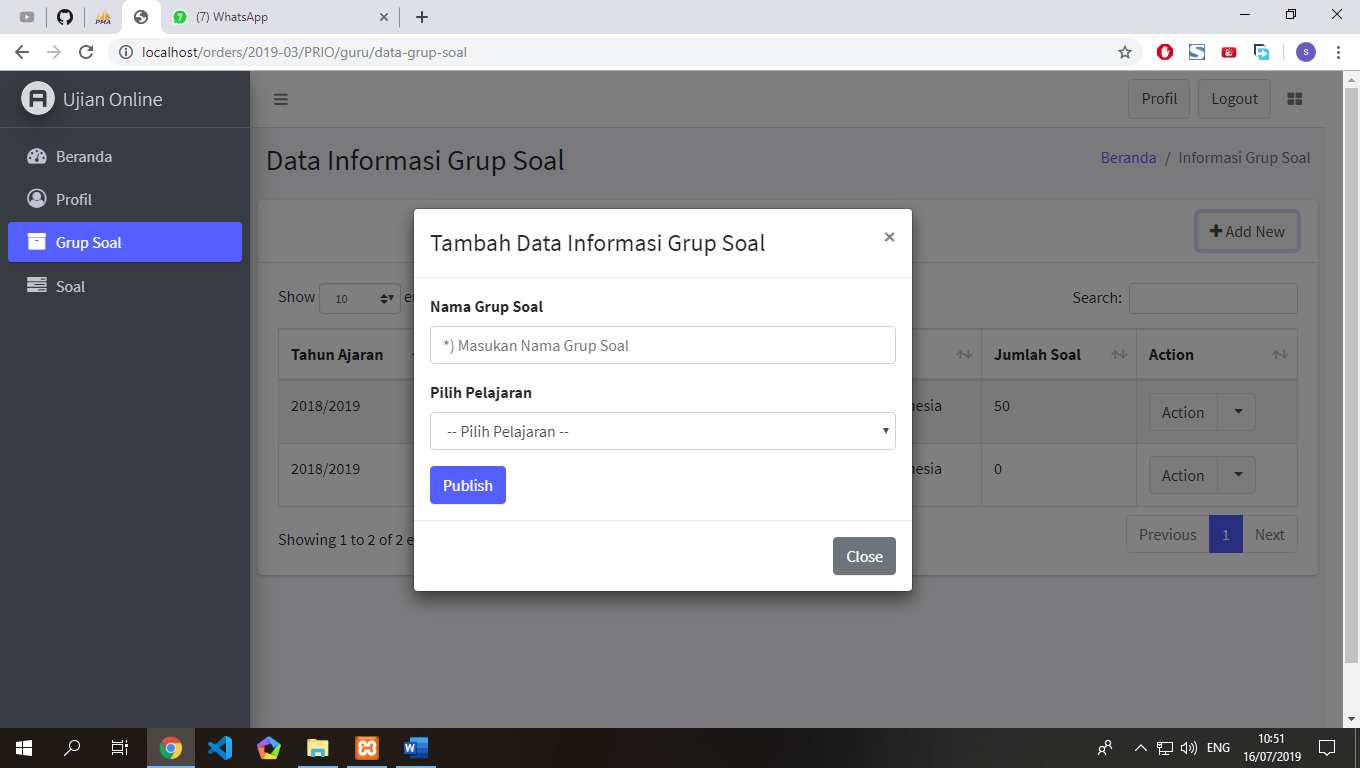
ORDER BY grup\_soal.nama\_grup\_soal ASC

")->result\_object();

}

**24. Halaman Guru Tambah Data Grup Soal**

Halaman ini berfungsi untuk menampilkan tambah informasi data grup soal. Untuk lebih jelasnya dapat dilihat pada gambar 5.25



**Gambar 5.25** Halaman Guru Tambah Data Grup Soal

Untuk mengetahui script proses Halaman Guru Tambah Data Grup Soal dapat dilihat pada Tabel 5.24.

Tabel 5.24 Script Halaman Guru Tambah Data Grup Soal

public function data\_grup\_soal\_store()

{

$this->m\_guru->post= $this->input->post();

if ( $this->m\_guru->data\_grup\_soal\_store() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

} else {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

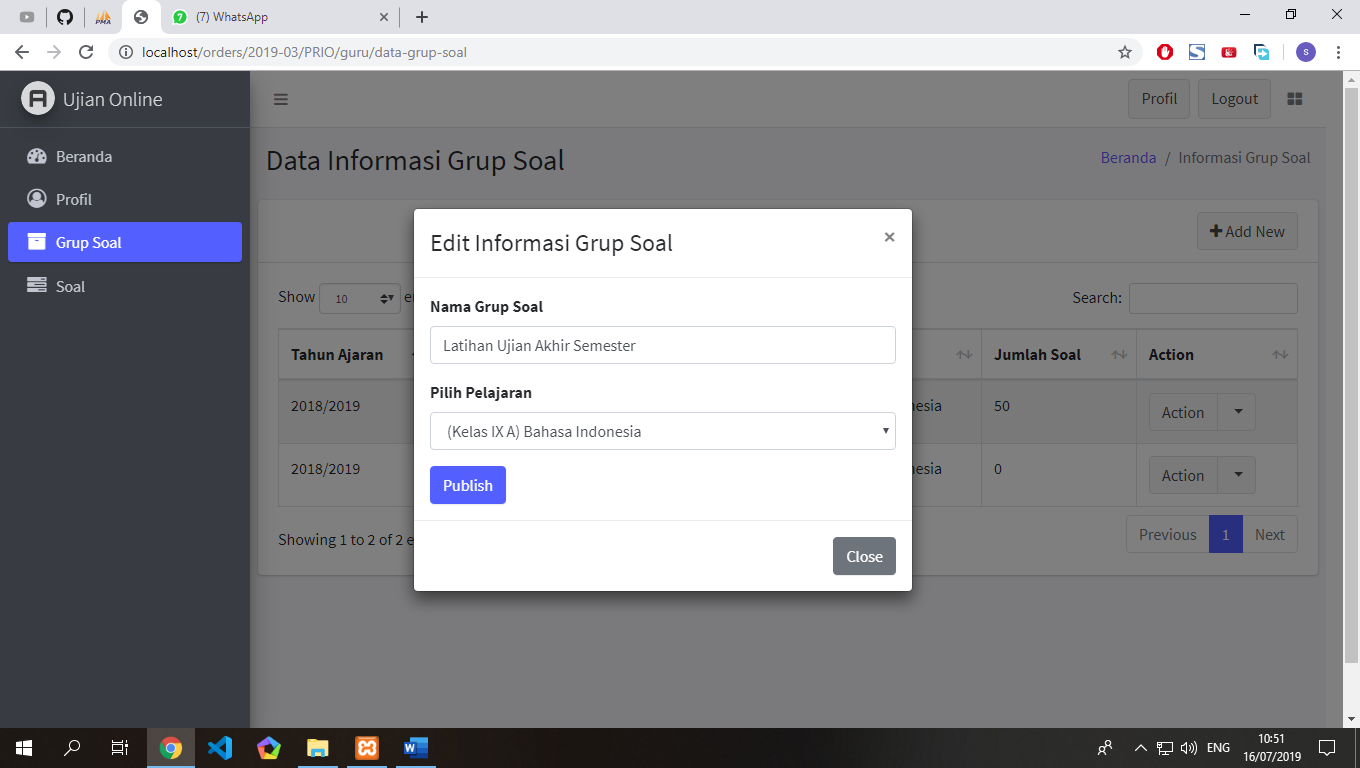
}

echo json\_encode($this->msg);

}

**25. Halaman Guru Edit Grup Soal**

Halaman ini berfungsi untuk menampilkan edit informasi data grup soal. Untuk lebih jelasnya dapat dilihat pada Gambar 5.26



**Gambar 5.26** Halaman Guru Edit Data Grup Soal

Untuk mengetahui script proses Halaman Guru Edit Data Grup Soal dapat dilihat pada Tabel 5.25.

Tabel 5.25 Script Halaman Guru Edit Data Grup Soal

public function update\_grup\_soal()

{

$this->m\_guru->post= $this->input->post();

if ( $this->m\_guru->update\_grup\_soal() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Diubah',

];

} else {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Diubah',

];

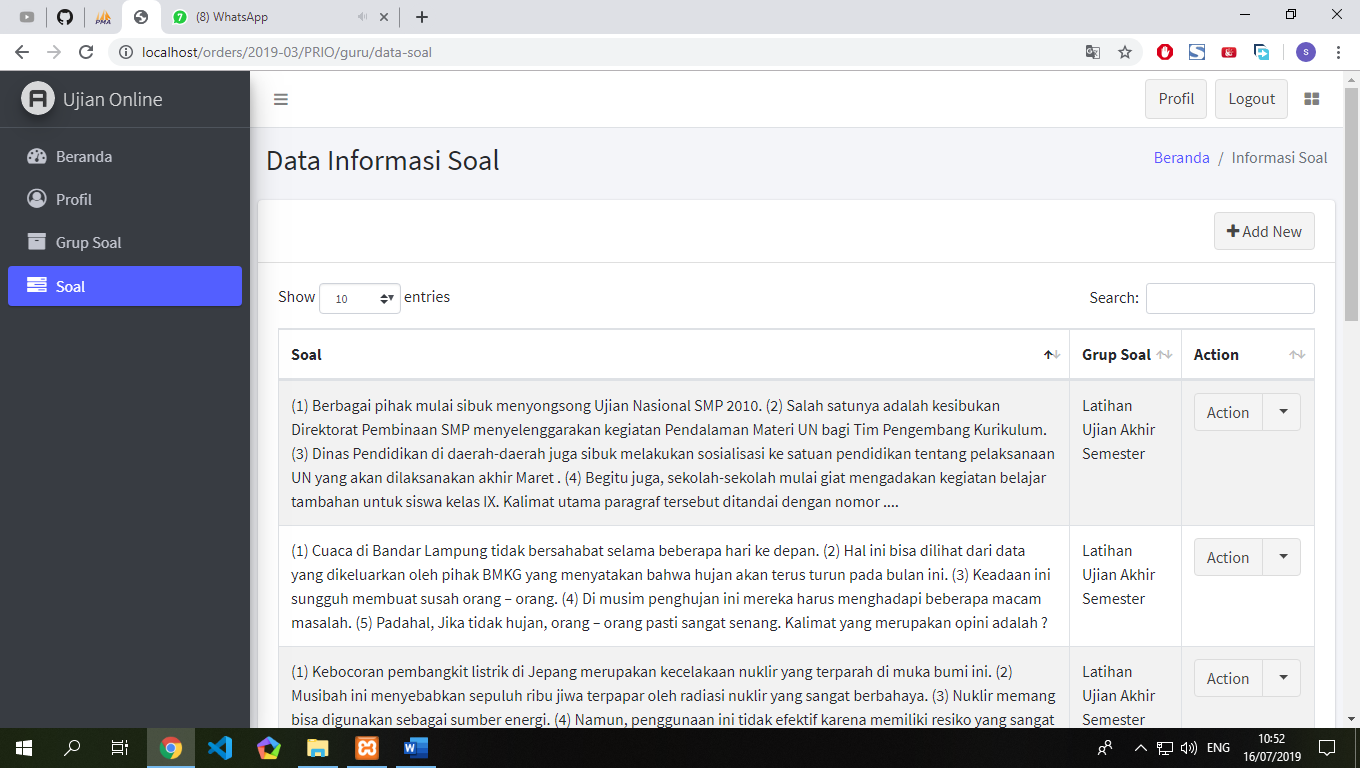
}

echo json\_encode($this->msg);

}

**26. Halaman Guru Data Soal**

Halaman ini berfungsi untuk menampilkan informasi data soal. Untuk lebih jelasnya dapat dilihat pada gambar 5.27



**Gambar 5.27** Halaman Guru Data Soal

Untuk mengetahui script proses Halaman Guru Data Soal dapat dilihat pada Tabel 5.26.

Tabel 5.26 Script Halaman Guru Data Soal

public function data\_soal()

{

$this->content['rows']= $this->m\_guru->data\_soal();

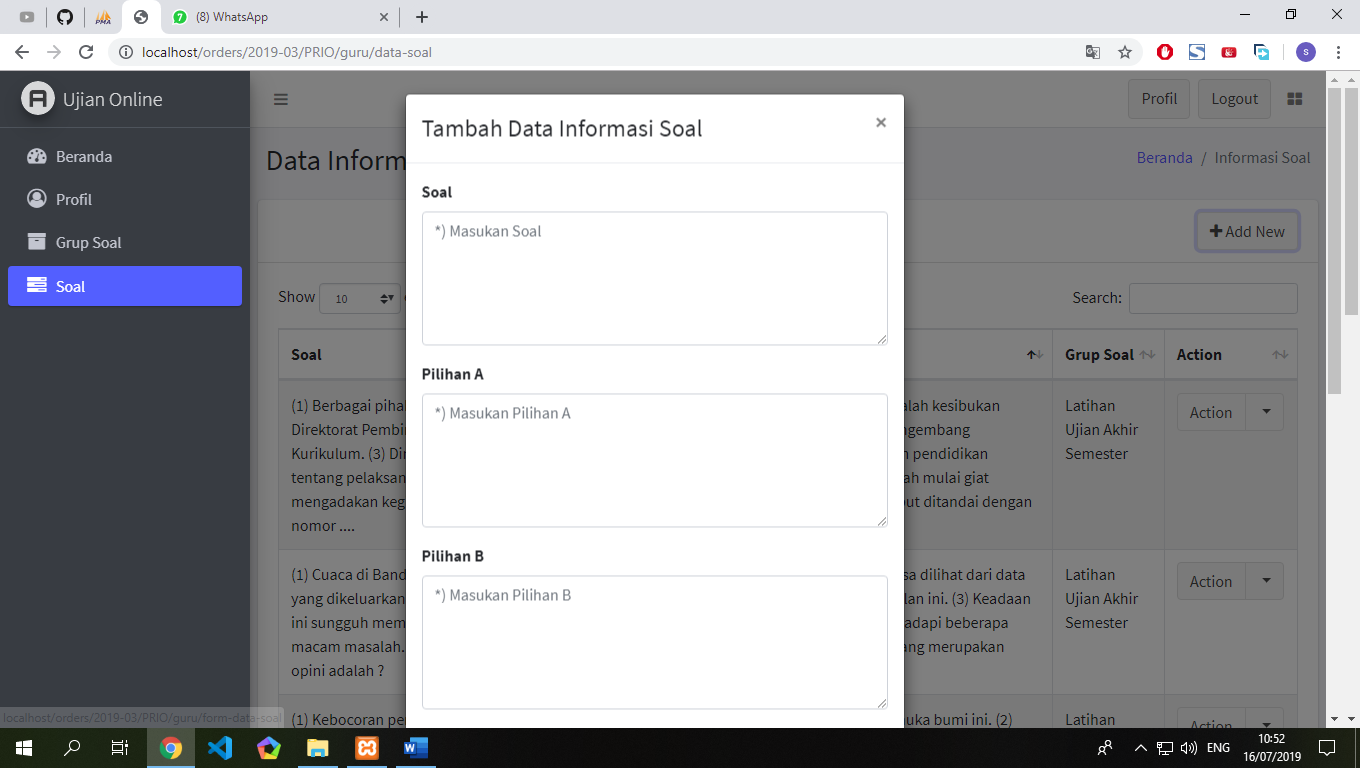
$this->view= 'guru/data\_soal';

$this->render\_pages();

}

**27. Halaman Guru Tambah Data Soal**

Halaman ini berfungsi untuk menampilkan tambah informasi data grup soal. Untuk lebih jelasnya dapat dilihat pada Gambar 5.28



**Gambar 5.28** Halaman Guru Tambah Data Soal

Untuk mengetahui script proses Halaman Guru Tambah Data Soal dapat dilihat pada Tabel 5.27.

Tabel 5.27 Script Halaman Guru Tambah Data Soal

public function data\_soal\_store()

{

$this->m\_guru->post= $this->input->post();

if ( $this->m\_guru->data\_soal\_store() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

} else {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Ditambahkan',

];

}

echo json\_encode($this->msg);

}

**28. Halaman Guru Edit Data Soal**

Halaman ini berfungsi untuk menampilkan edit informasi data grup soal. Untuk lebih jelasnya dapat dilihat pada gambar 5.29



**Gambar 5.29** Halaman Guru Edit Data Soal

Untuk mengetahui script proses Halaman Guru Tambah Data Soal dapat dilihat pada Tabel 5.28.

Tabel 5.28 Script Halaman Guru Edit Data Soal

public function data\_soal\_update()

{

$this->m\_guru->post= $this->input->post();

if ( $this->m\_guru->data\_soal\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Diubah',

];

} else {

$this->msg= [

'stats'=>1,

'msg'=>'Data Berhasil Diubah',

];

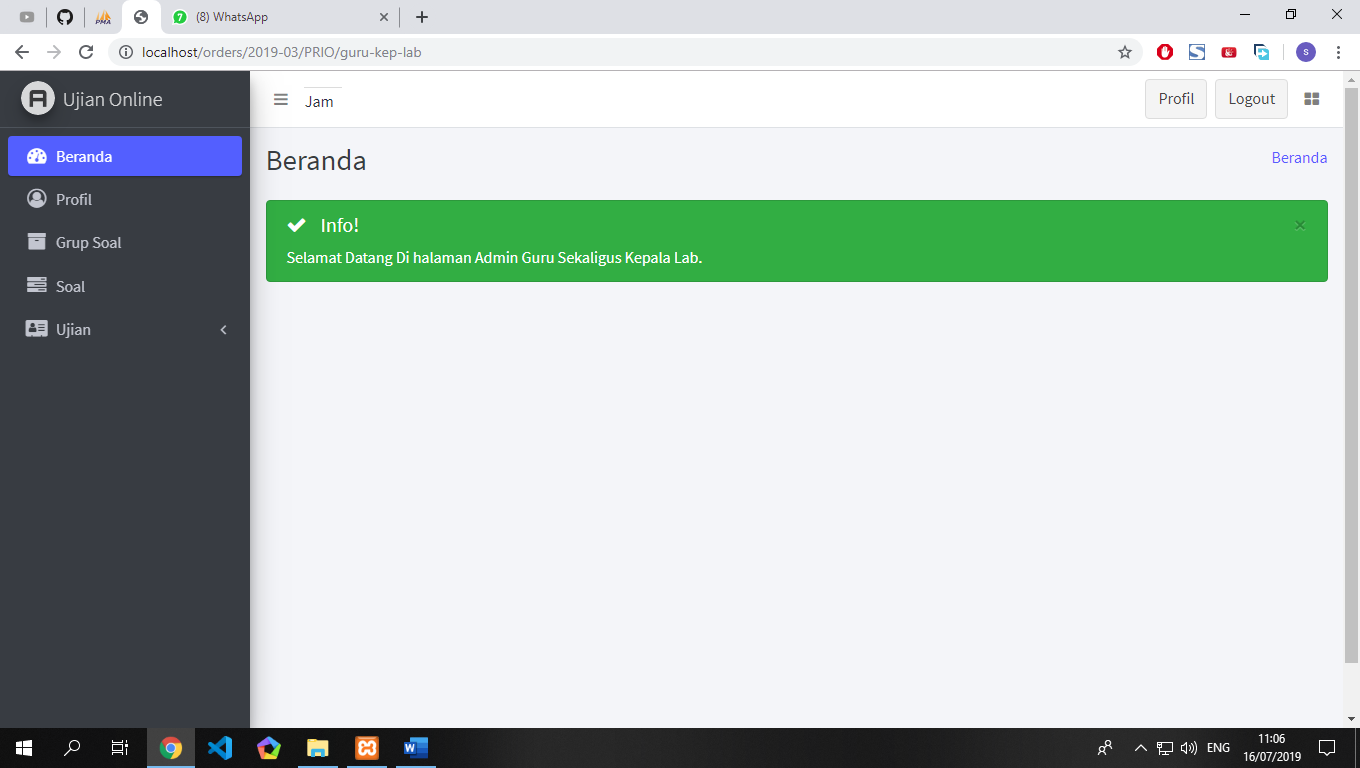
}

echo json\_encode($this->msg);

}

**29. Halaman Kepala Lab Beranda**

Halaman ini berisikan menu yang terdapat di halaman Kepala Lab sama dengan halaman guru hanya saja untuk kepala lab terdapat tambahan satu menu yang berfungsi untuk membuat jadwal ujian untuk lebih jelas dapat dilihat pada Gambar 5.30



**Gambar 5.29** Halaman Kepala Lab Halaman Beranda

Untuk mengetahui script proses Halaman Kepala Lab Halaman Beranda dapat dilihat pada Tabel 5.29.

Tabel 5.29 Script Halaman Kepala Lab Halaman Beranda

public function index()

{

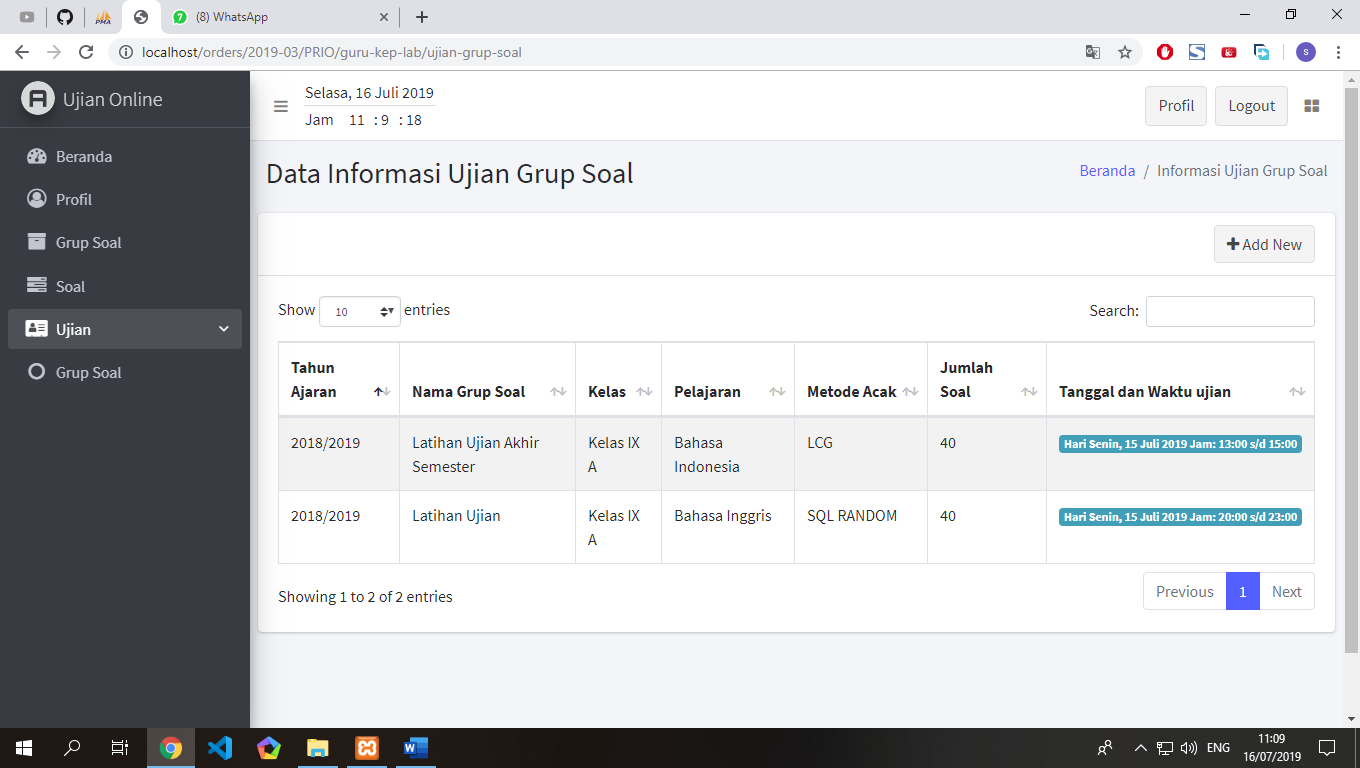
$this->view= 'guru\_kep\_lab/index';

$this->render\_pages();

}

**30. Halaman Kepala Lab Data Ujian Grup Soal**

Halaman ini berfungsi untuk membuat jadwal ujian berdasarkan grup soal yang sudah dibuat oleh guru-guru sesuai dengan mata pelajaran yang diampu grup soal yang bisa dibuatkan jadwal ujian hanya grup soal yang mempunyai soal lebih dari 40 jika. Untuk lebih jelasnya dapat dilihat pada Gambar 5.31



**Gambar 5.31** Halaman Kepala Lab Data Ujian Grup Soal

Untuk mengetahui script proses Halaman Kepala Lab Data Ujian Grup Soal dapat dilihat pada Tabel 5.30.

Tabel 5.30 Script Halaman Kepala Lab Data Ujian Grup Soal

public function ujian\_grup\_soal()

{

$this->view= 'guru\_kep\_lab/data\_ujian\_grup\_soal';

$this->content['rows']= $this->m\_kep\_lab->ujian\_grup\_soal();

$this->render\_pages();

}

public function ujian\_grup\_soal()

{

if( ! empty($this->post['id\_grup\_soal']) ){

return $this->db->query("

SELECT \*,

( (SELECT COUNT(id\_soal) FROM soal WHERE soal.id\_grup\_soal=grup\_soal.id\_grup\_soal) ) AS jumlah\_soal

FROM grup\_soal

LEFT JOIN pbm

ON grup\_soal.id\_pelajaran=pbm.id\_pelajaran

LEFT JOIN pelajaran

ON pbm.id\_pelajaran=pelajaran.id\_pelajaran

LEFT JOIN kelas

ON pelajaran.id\_kelas=kelas.id\_kelas

WHERE grup\_soal.id\_grup\_soal='{$this->post["id\_grup\_soal"]}'

")->result\_object();

} else {

return $this->db->query("

SELECT \*,

( (SELECT COUNT(id\_soal) FROM soal WHERE soal.id\_grup\_soal=grup\_soal.id\_grup\_soal) ) AS jumlah\_soal

FROM grup\_soal

LEFT JOIN pbm

ON grup\_soal.id\_pelajaran=pbm.id\_pelajaran

LEFT JOIN pelajaran

ON pbm.id\_pelajaran=pelajaran.id\_pelajaran

LEFT JOIN kelas

ON pelajaran.id\_kelas=kelas.id\_kelas

WHERE 1=1

GROUP BY grup\_soal.id\_grup\_soal

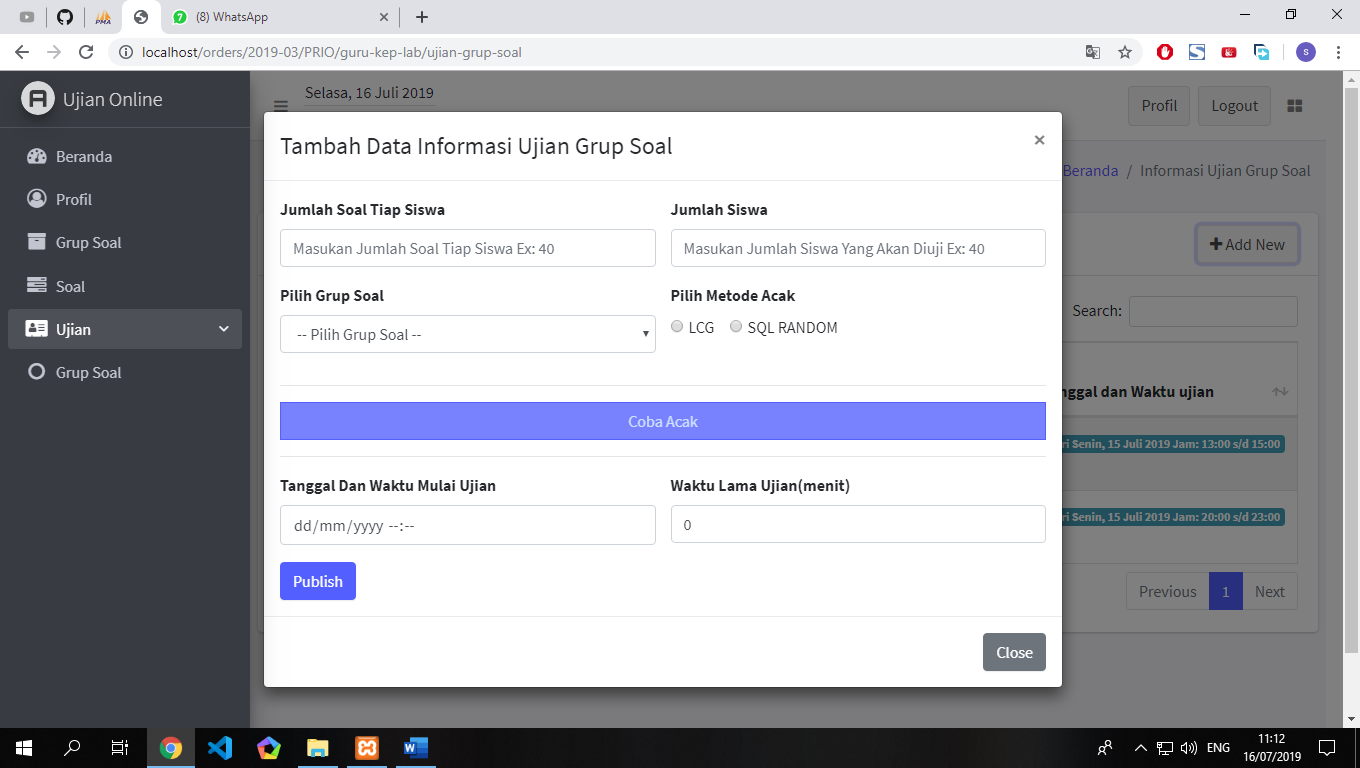
")->result\_object();

}

}

**31. Halaman Kepala Lab Tambah Data Ujian Grup Soal**

Halaman ini berfungsi untuk membuat jadwal ujian baru berdasarkan grup soal, grup soal hanya bisa dipakai satu kali ujian. Lebih jelasnya dapat dilihat pada gambar 5.32



**Gambar 5.32** Halaman Kepala Lab Tambah Data Ujian Grup Soal

Untuk mengetahui script proses Halaman Kepala Lab Tambah Data Ujian Grup Soal dapat dilihat pada Tabel 5.31.

Tabel 5.31 Script Halaman Kepala Lab Tambah Data Ujian Grup Soal

public function store\_ujian\_grup\_soal()

{

$this->m\_kep\_lab->post= $this->input->post();

if ( $this->m\_kep\_lab->store\_ujian\_grup\_soal() ) {

$this->msg= [

'stats'=>1,

'msg'=>'Informasi jadwal ujian berhasil dibuat',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=>'Informasi jadwal ujian gagal dibuat',

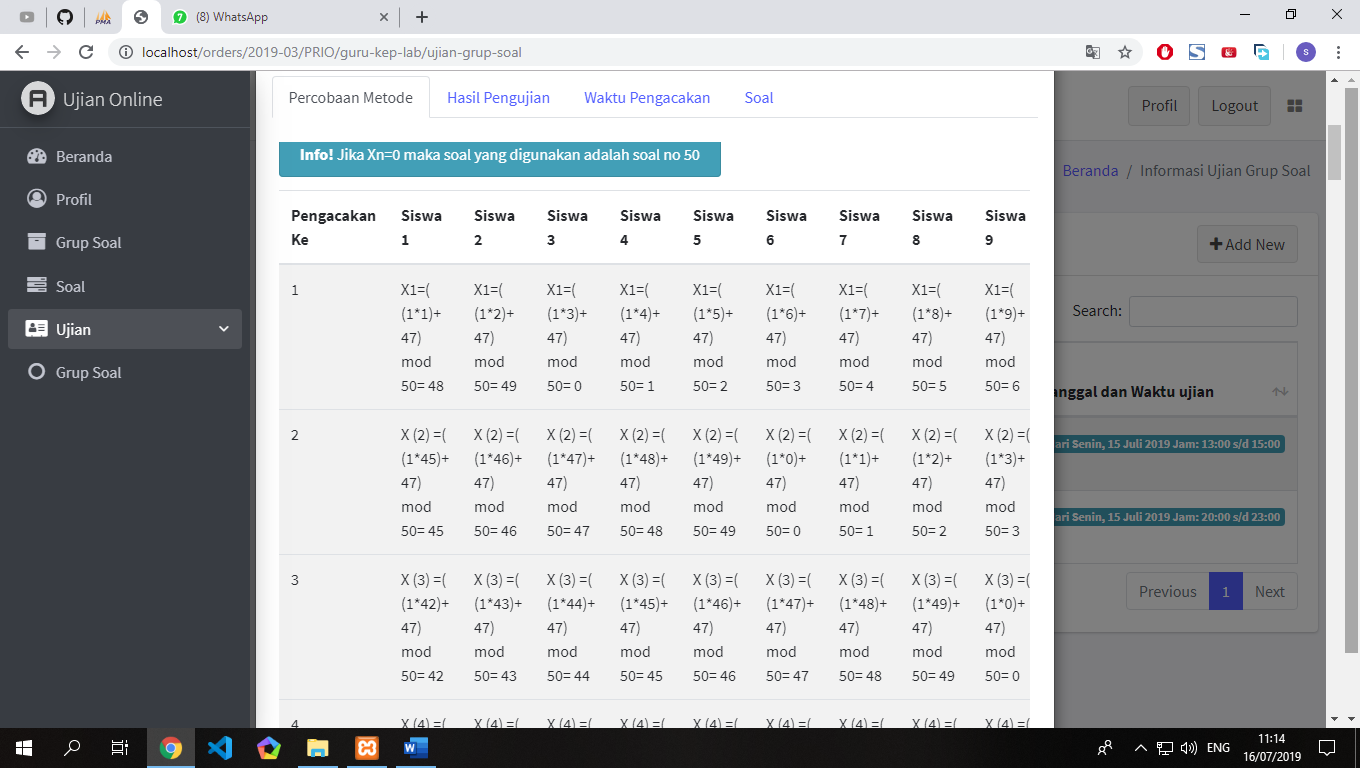
];

}

echo json\_encode($this->msg);

}

Setelah kepala lab menginputkan jumlah soal, jumlah siswa, pilih grup soal, dan metode acak kepala lab dapat mencoba pengujian dengan memilih menu coba acak. Hasilnya seperti Gambar (5.33-5.40).



**Gambar 5.33** Percobaan Menggunakan Metode LCG

Untuk mengetahui script proses Percobaan Menggunakan Metode LCG dapat dilihat pada Tabel 5.32.

Tabel 5.32 Script Percobaan Menggunakan Metode LCG

$lcg= [];

$xn= [];

$tr= '';

/\* generate th \*/

$th= '';

for ($s=1; $s <

($this->input->get('jumlah\_siswa')+1) ; $s++) {

$th.= "<th>Siswa $s</th>";

}

$\_lcg= [

'b'=> $this->input->get('bil\_prima'),

'm'=> $this->input->get('total\_soal')

];

$this->benchmark->mark('code\_start');

# mulai waktu pengacakan metode LCG

for ($i=1; $i <

($this->input->get('jumlah\_soal')+1) ; $i++) {

$tr.= '<tr>';

$tr.= "<td>$i</td>";

for ($j=1; $j <

($this->input->get('jumlah\_siswa')+1) ; $j++) {

if ( $i==1 ) {

$lcg[$i][$j]=

( (1\* $j )+ $\_lcg['b'] ) % $\_lcg['m'];

$xn[$i][$j]= $lcg[$i][$j];

$tr.= "<td>X".($i)."=( (1\*{$j})+

{$\_lcg['b']}) mod {$\_lcg['m']}=

{$lcg[$i][$j]} </td>";

}

else {

$xn[$i][$j]= $xn[($i-1)][$j];

$lcg[$i][$j]=

( (1\* $xn[$i][$j] )+ $\_lcg['b'] ) %

$\_lcg['m'];

$xn[$i][$j]= $lcg[$i][$j];

$tr.= "<td>X ($i) =( (1\*{$xn[$i][$j]})+ {$\_lcg['b']}) mod {$\_lcg['m']}= {$lcg[$i][$j]} </td>";

}

}

$tr.= '<tr>';

}

$this->benchmark->mark('code\_end');

# selesai waktu pengacakan metode LCG

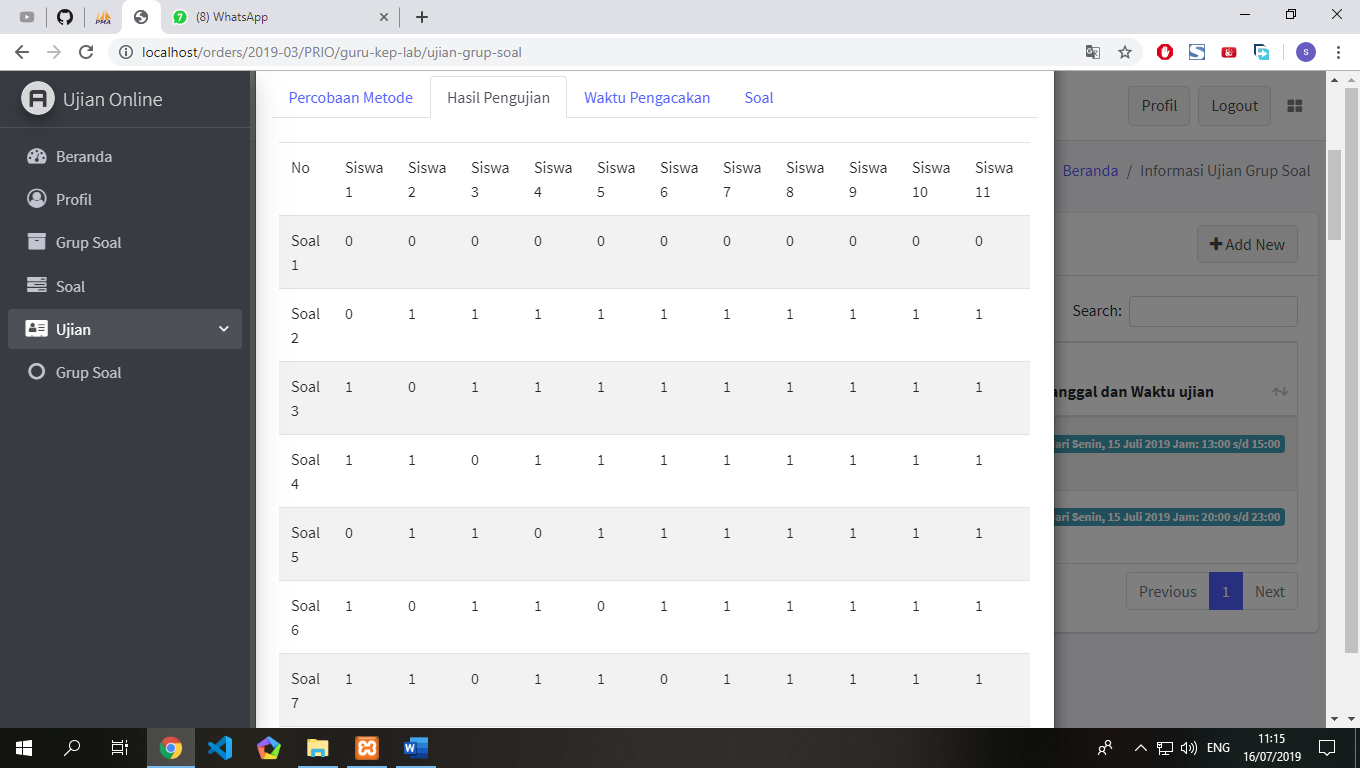
$\_get\_hasil\_pengujian= [

'jumlah\_siswa'=> ($this->input->get('jumlah\_siswa') +1),

'jumlah\_soal'=> ($this->input->get('total\_soal')),

'xn'=> $xn

];



**Gambar 5.34** Hasil Pengujian LCG

Untuk mengetahui script proses Hasil Pengujian LCG dapat dilihat pada Tabel 5.33.

Tabel 5.33 Script Hasil Pengujian LCG

public function get\_hasil\_pengujian($jumlah\_siswa,$jumlah\_soal,$xn)

{

$this->html='';

$th='<td>No</td>';# header table

for ($i=1; $i < $jumlah\_siswa ; $i++) {

$th .= '<td>Siswa '.$i.'</td>';

}

$tr\_array= [];

for ($j=1; $j < $jumlah\_soal+1 ; $j++) {

for ($i=1; $i < $jumlah\_siswa ; $i++) {

$tr\_array[$j][$i]=0;

}

}

$siswa\_array= [];

foreach ($xn as $key => $value) {

$sub=[];

foreach ($xn as $key\_sub => $value\_sub) {

// $sub[$key\_sub]= ( $value\_sub[$key]==0 )? 50 : $value\_sub[$key] ;

$sub[]= $value\_sub ;

}

$siswa\_array[$key]= [

'siswa'=>$key,

'soal'=>$sub,

];

}

# ganti data tr\_array dengan siswa array

foreach ($siswa\_array as $key => $value) {

foreach ($value['soal'] as $key\_sub => $value\_sub) {

if ( $value\_sub[$value['siswa']]==0 ) {

# if index soal sama dengan 0

$tr\_array[50][$value['siswa']]= +1;

} else {

# code...

$tr\_array[$value\_sub[$value['siswa']]+1 ][$value['siswa']]= +1;

}

}

}

# generate tr body table

$tr='';

foreach ($tr\_array as $key => $value) {

$tr .= '

<tr>

<td>Soal '.$key.'</td>

';

foreach ($value as $key\_sub => $value\_sub) {

$tr .= '<td>' .$value\_sub .'</td>';

}

$tr .= '</tr>';

}

return $this->html .= '

<div class="table-responsive">

<table class="table table-striped">

<thead>

<tr>

'.$th.'

</tr>

</thead>

<tbody>

'.$tr.'

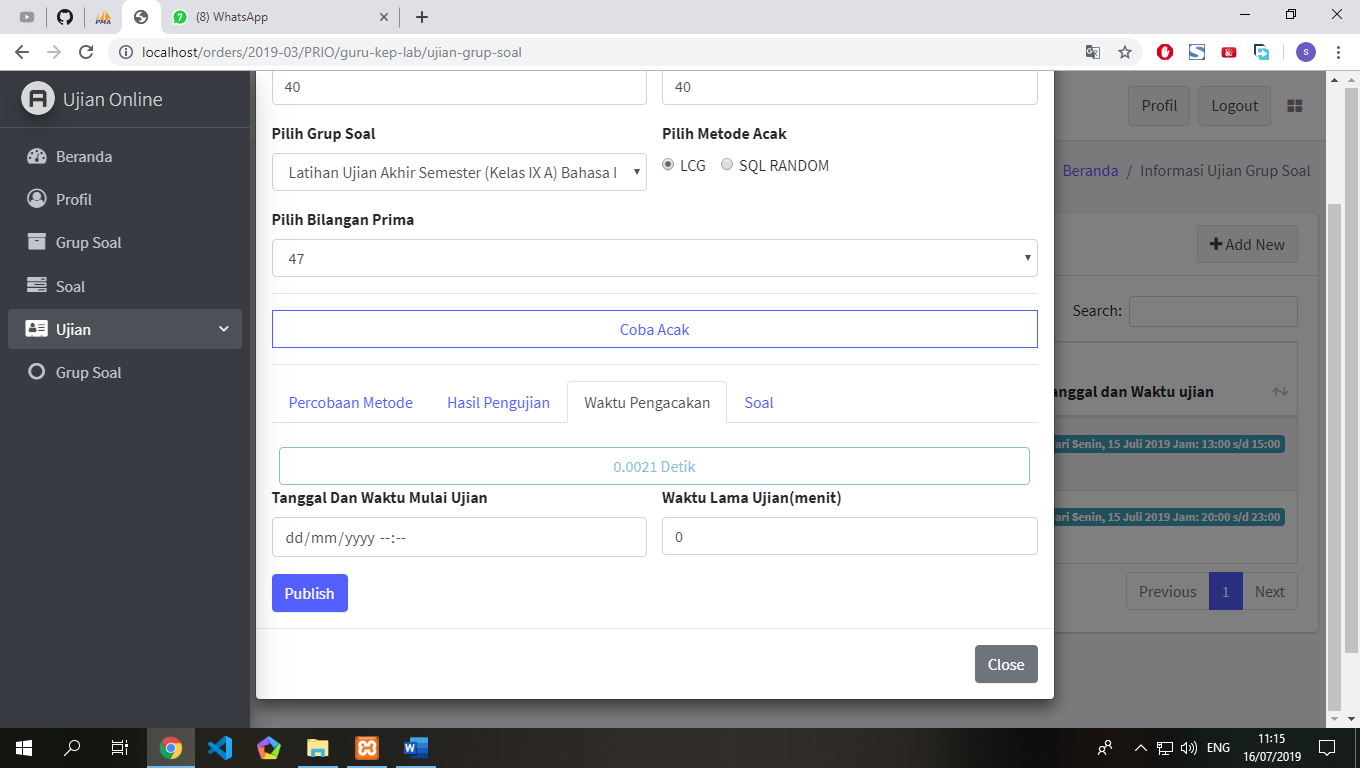
</tbody>

</table>

</div>

';

}

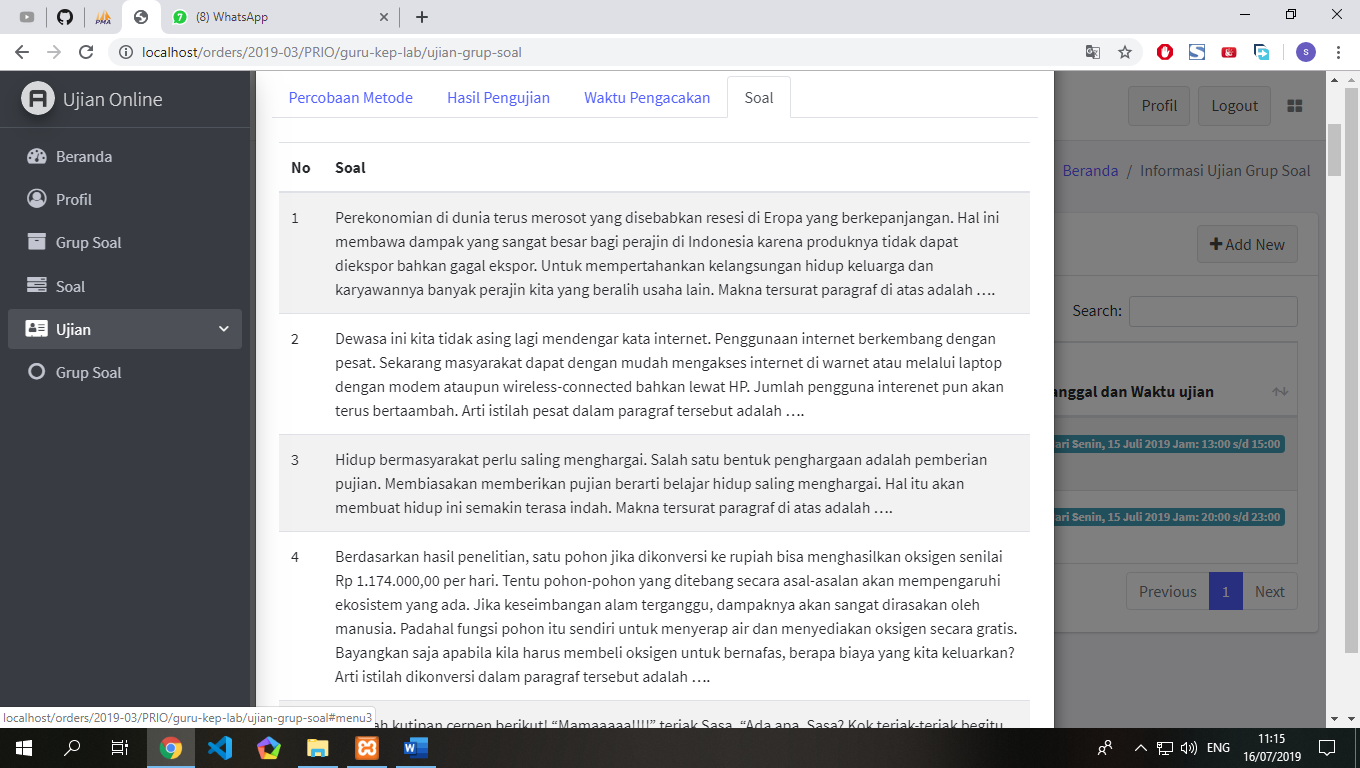


**Gambar 5.35** Waktu Pengacakan LCG

Untuk mengetahui script proses Waktu Pengacakan LCG dapat dilihat pada Tabel 5.34.

Tabel 5.34 Waktu Pengacakan LCG

$this->benchmark->elapsed\_time('code\_start','code\_end')



**Gambar 5.36** Soal LCG

Untuk mengetahui script proses Soal LCG dapat dilihat pada Tabel 5.35.

Tabel 5.35 Soal LCG

public function get\_soal($id\_grup\_soal)

{

$this->m\_kep\_lab->post['id\_grup\_soal']= $id\_grup\_soal;

$no=1;

$this->html='';

foreach ($this->m\_kep\_lab->get\_soal() as $key => $value) {

$this->html .= "

<tr>

<td>{$no}</td>

<td>{$value->soal}</td>

</tr>

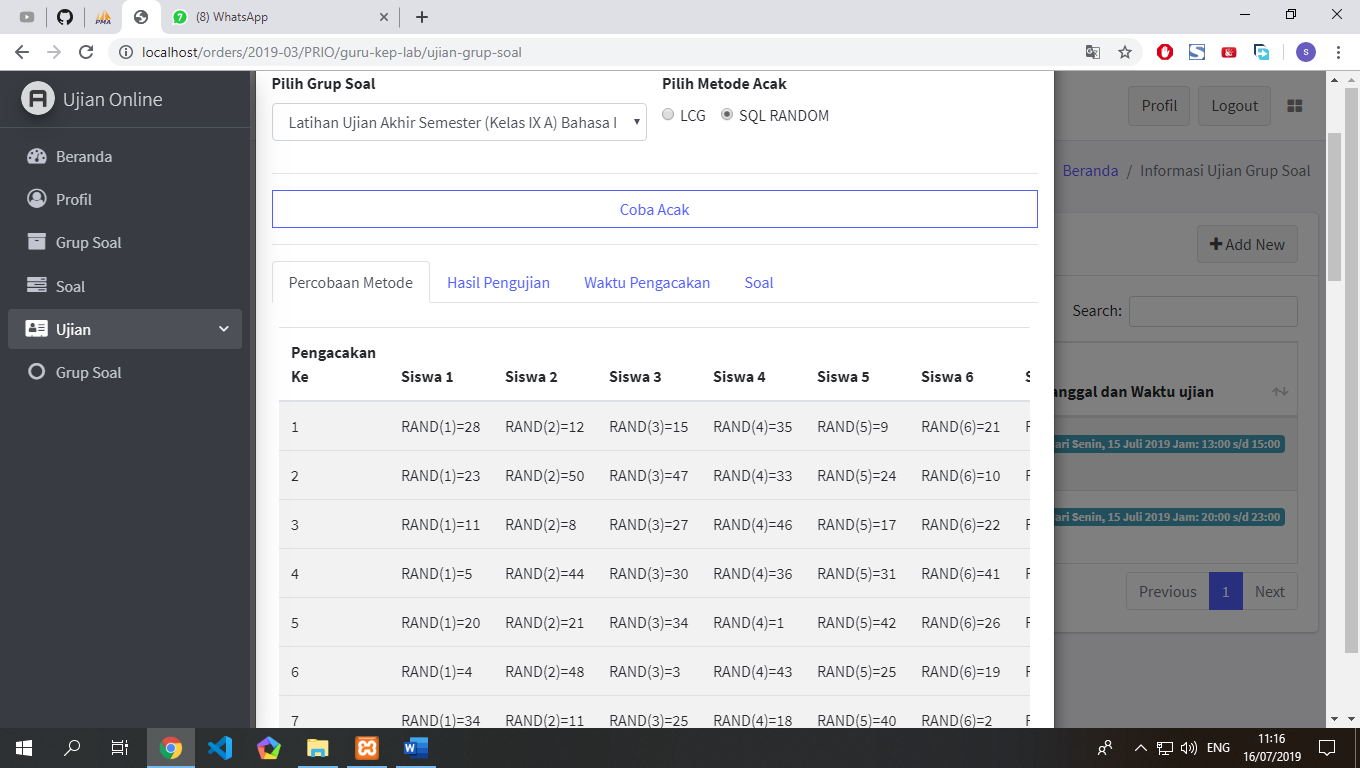
";

$no++;

}

return $this->html;

}



**Gambar 5.37** Percobaan Menggunakan Metode SQL Random

Untuk mengetahui script proses Percobaan Menggunakan Metode SQL Random dapat dilihat pada Tabel 5.36.

Tabel 5.36 Script Percobaan Menggunakan Metode SQL Random

$th= '';

for ($s=1; $s <

($this->input->get('jumlah\_siswa')+1) ; $s++) {

$th.= "<th>Siswa $s</th>";

}

$this->benchmark->mark('code\_start');

$tr= '';

$sql=[];

$xn=[];

$this->m\_kep\_lab->post["id\_grup\_soal"]=

$this->input->get('id\_grup\_soal');

$this->m\_kep\_lab->post["limit"]=

$this->input->get('jumlah\_soal');

for ($js=1; $js <

($this->input->get('jumlah\_siswa')+1) ; $js++) {

$this->m\_kep\_lab->post["seed"]= $js;

$no=1;

foreach ($this->m\_kep\_lab->metode\_sql() as

$key => $value) {

$sql[$no][$js]= [

'seed'=> $js,

'result'=> $value->id\_soal

];

$xn[$no][$js]= $value->id\_soal;

$no++;

}

}

// echo '<pre>';

// print\_r($sql);

// echo '</pre>';

for ($i=1; $i <

($this->input->get('jumlah\_soal')+1) ; $i++) {

$tr.= '<tr>';

$tr.= "<td>$i</td>";

for ($j=1; $j <

($this->input->get('jumlah\_siswa')+1) ; $j++) {

$tr .= "<td>RAND({$j})=

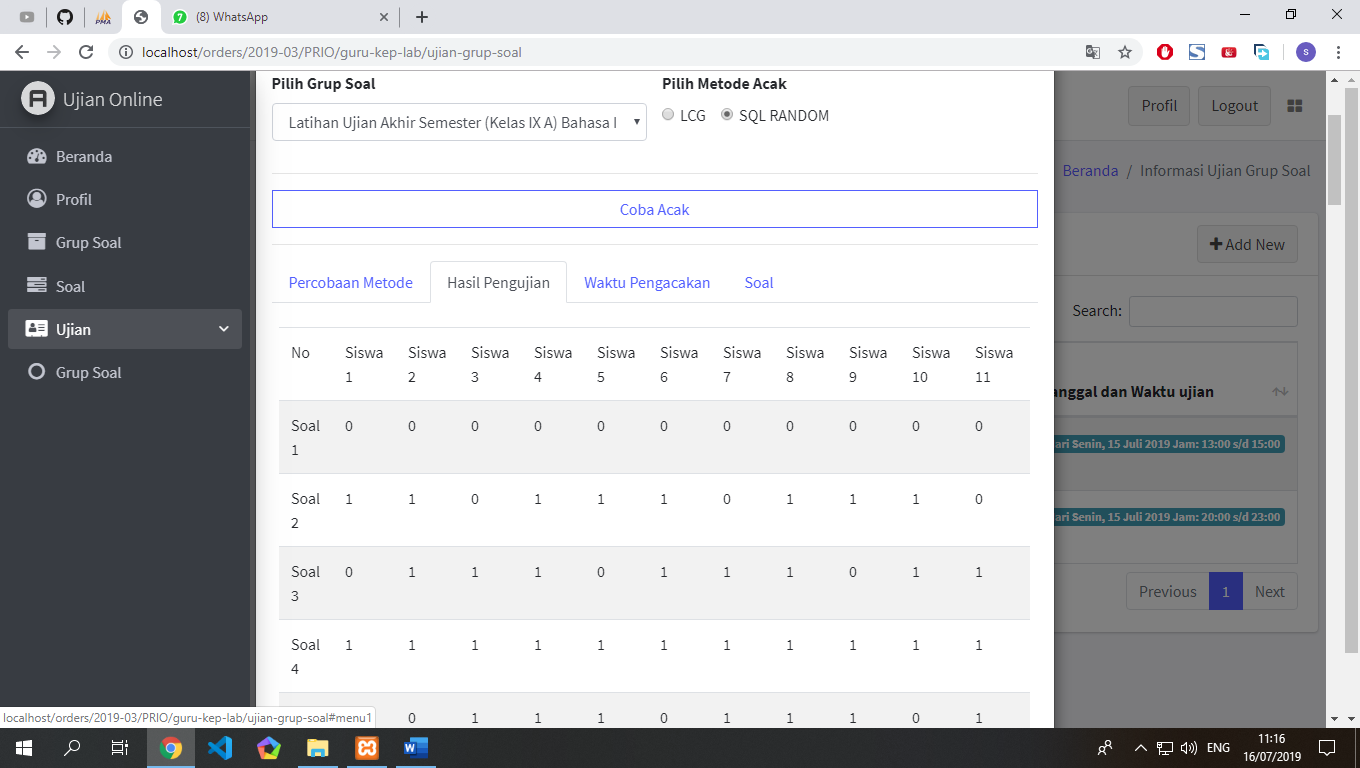
{$sql[$i][$j]["result"]}</td>";

}

$tr.= '<tr>';

}

$this->benchmark->mark('code\_end');



**Gambar 5.38** Hasil Pengujian Menggunakan Metode SQL Random

Untuk mengetahui script proses Hasil Pengujian Menggunakan Metode SQL Random dapat dilihat pada Tabel 5.37.

Tabel 5.37 Script Hasil Pengujian Menggunakan Metode SQL Random

public function

get\_hasil\_pengujian($jumlah\_siswa,$jumlah\_soal,$xn)

{

$this->html='';

$th='<td>No</td>';# header table

for ($i=1; $i < $jumlah\_siswa ; $i++) {

$th .= '<td>Siswa '.$i.'</td>';

}

$tr\_array= [];

for ($j=1; $j < $jumlah\_soal+1 ; $j++) {

for ($i=1; $i < $jumlah\_siswa ; $i++) {

$tr\_array[$j][$i]=0;

}

}

$siswa\_array= [];

foreach ($xn as $key => $value) {

$sub=[];

foreach ($xn as $key\_sub => $value\_sub) {

// $sub[$key\_sub]= ( $value\_sub[$key]==0 )? 50 : $value\_sub[$key] ;

$sub[]= $value\_sub ;

}

$siswa\_array[$key]= [

'siswa'=>$key,

'soal'=>$sub,

];

}

# ganti data tr\_array dengan siswa array

foreach ($siswa\_array as $key => $value) {

foreach ($value['soal'] as $key\_sub => $value\_sub) {

if ( $value\_sub[$value['siswa']]==0 ) {

# if index soal sama dengan 0

$tr\_array[50][$value['siswa']]= +1;

} else {

# code...

$tr\_array[$value\_sub[$value['siswa']]+1 ][$value['siswa']]= +1;

}

}

}

# generate tr body table

$tr='';

foreach ($tr\_array as $key => $value) {

$tr .= '

<tr>

<td>Soal '.$key.'</td>

';

foreach ($value as $key\_sub => $value\_sub) {

$tr .= '<td>' .$value\_sub .'</td>';

}

$tr .= '</tr>';

}

return $this->html .= '

<div class="table-responsive">

<table class="table table-striped">

<thead>

<tr>

'.$th.'

</tr>

</thead>

<tbody>

'.$tr.'

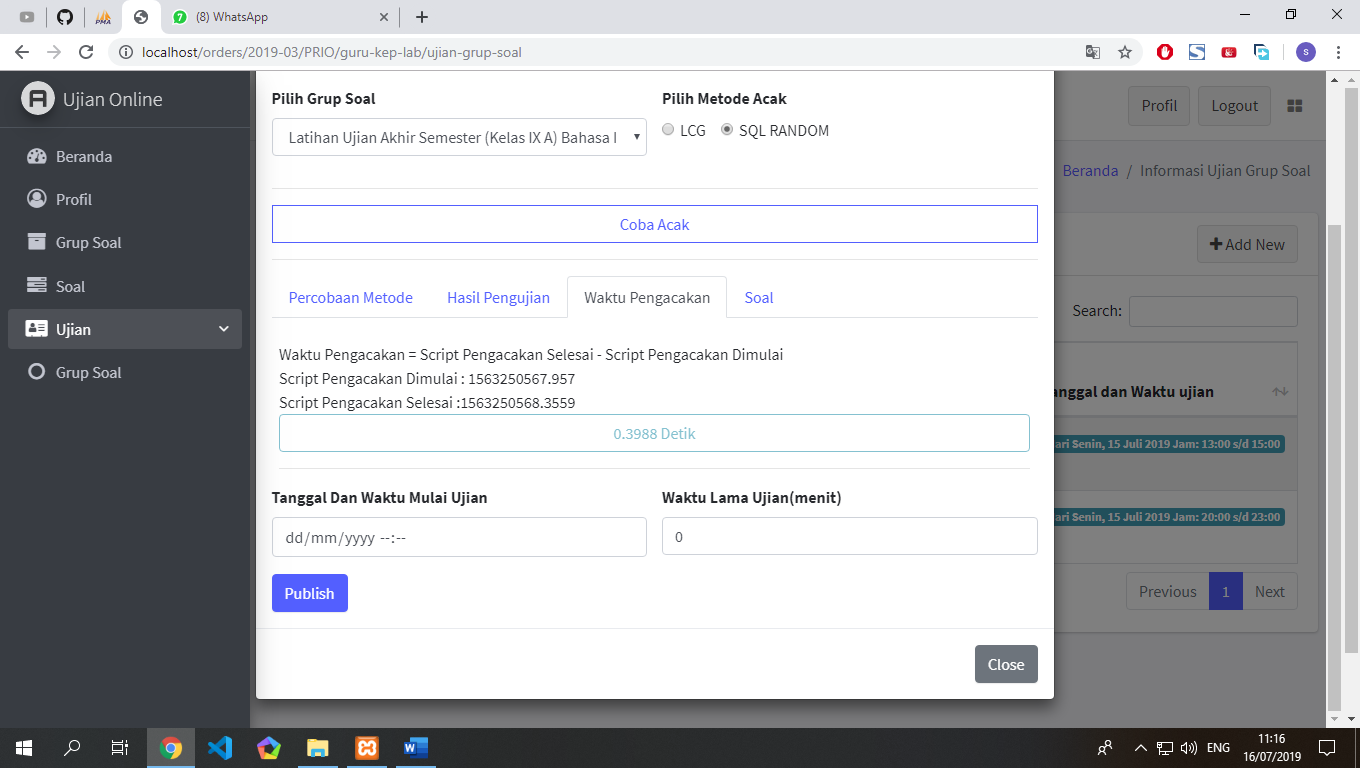
</tbody>

</table>

</div>

';

}



**Gambar 5.39** Waktu Pengacakan Menggunakan Metode SQL Random

Untuk mengetahui script proses Waktu Pengacakan Menggunakan Metode SQL Random dapat dilihat pada Tabel 5.38.

Tabel 5.38 Script Waktu Pengacakan Menggunakan Metode SQL Random

$this->benchmark->elapsed\_time('code\_start','code\_end')



**Gambar 5.40** Soal Menggunakan Metode SQL Random

Untuk mengetahui script proses Soal Menggunakan Metode SQL Random dapat dilihat pada Tabel 5.39.

Tabel 5.39 Script Soal Menggunakan Metode SQL Random

/\* mendapatkan soal berdasarkan grup soal \*/

public function get\_soal($id\_grup\_soal)

{

$this->m\_kep\_lab->post['id\_grup\_soal']= $id\_grup\_soal;

$no=1;

$this->html='';

foreach ($this->m\_kep\_lab->get\_soal() as $key => $value) {

$this->html .= "

<tr>

<td>{$no}</td>

<td>{$value->soal}</td>

</tr>

";

$no++;

}

return $this->html;

}

/\* mendapatkan soal \*/

public function get\_soal()

{

return $this->db->query("

SELECT \* FROM soal WHERE id\_grup\_soal='{$this->post["id\_grup\_soal"]}'

ORDER BY id\_soal ASC

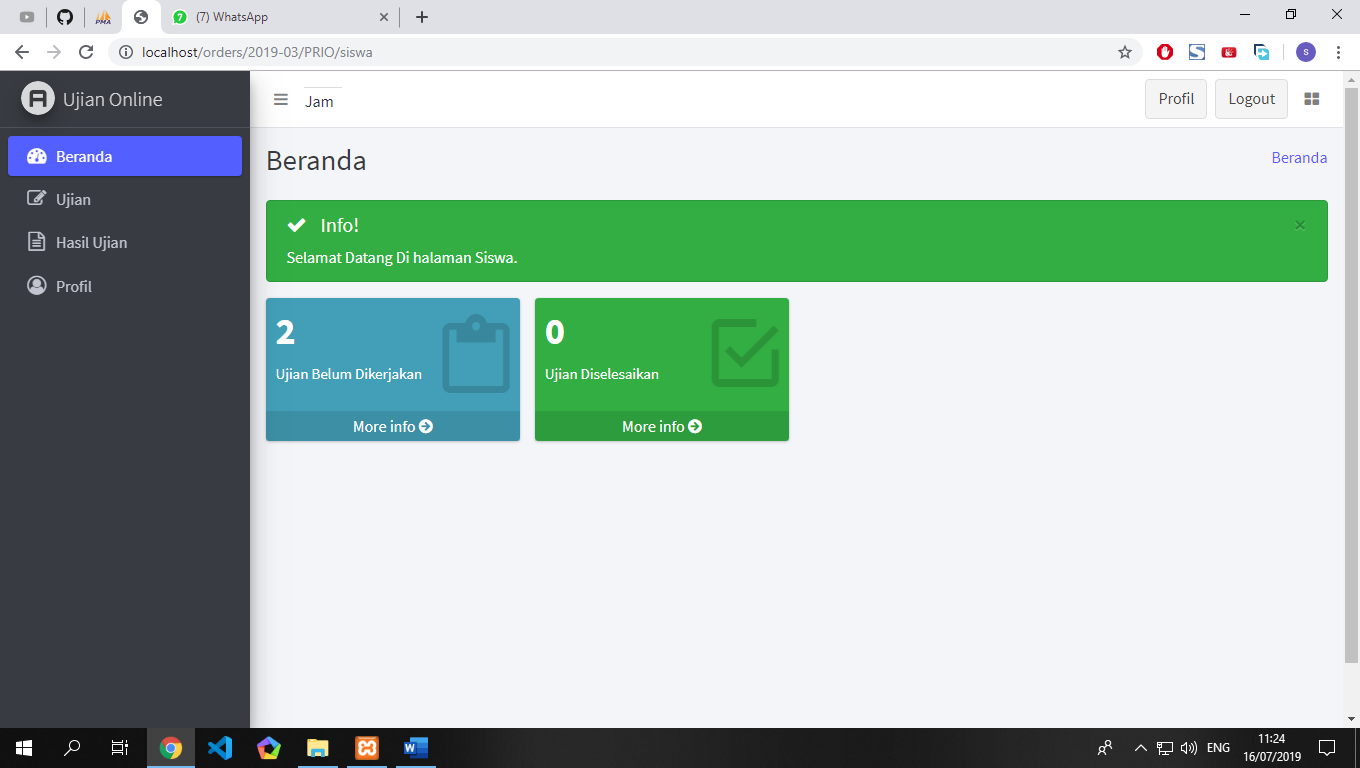
")->result\_object();

}

/\* end mendapatkan soal \*/

**31. Halaman Siswa Beranda**

Halaman ini menampilkan halaman beranda siswa pertama kali login menu yang dapat dipilih diantaranya: ujian, hasil ujian dan profil. Untuk lebih jelasnya dapat dilihat pada Gambar 5.41



**Gambar 5.41** Halaman Siswa Beranda

Untuk mengetahui script proses Halaman Siswa Beranda dapat dilihat pada Tabel 5.40.

Tabel 5.40 Script Halaman Siswa Beranda

/\* ==================== Start Beranda ==================== \*/

public function index()

{

$this->content['rows']= $this->m\_siswa->data\_ujian();

$this->view= 'siswa/index';

$this->render\_pages();

}

/\* ==================== End Beranda ==================== \*/

public function data\_ujian()

{

return $this->db->query("

SELECT \*,

(SELECT COUNT(\*) AS counts FROM hasil\_ujian WHERE hasil\_ujian.nis=pbm.nis AND hasil\_ujian.id\_grup\_soal=grup\_soal.id\_grup\_soal) AS counts

FROM pbm

INNER JOIN pelajaran

ON pbm.id\_pelajaran=pelajaran.id\_pelajaran

INNER JOIN grup\_soal

ON pbm.id\_pelajaran=grup\_soal.id\_pelajaran

WHERE 1=1

AND pbm.nis='".$this->session->userdata('username')."'

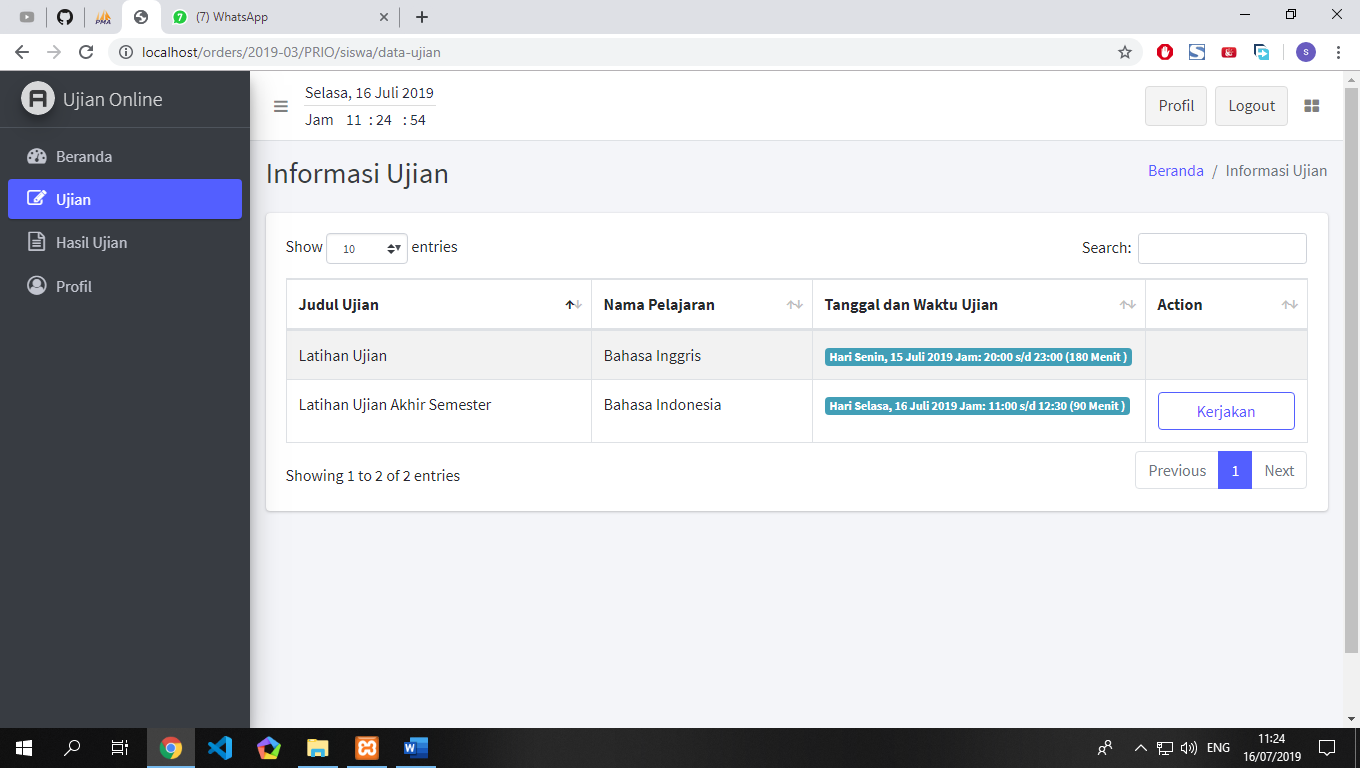
AND grup\_soal.metode\_acak!=''

")->result\_object();

}

**32. Halaman Siswa Data Ujian**

Halaman ini berfungsi untuk menampilkan informasi ujian yang belum dikerjakan, untuk mengikuti ujian tinggal pilih saja tombol kerjakan maka akan muncul popup proses ujian dengan ketentuan pada saat itu tanggal dan waktu sesuai dengan jadwal ujian. Lebih jelasnya dapat dilihat pada gambar 5.42



**Gambar 5.42** Halaman Siswa Data Ujian

Untuk mengetahui script proses Halaman Siswa Data Ujian dapat dilihat pada Tabel 5.41.

Tabel 5.41 Script Halaman Siswa Data Ujian

public function data\_ujian()

{

$this->content['rows']= $this->m\_siswa->data\_ujian();

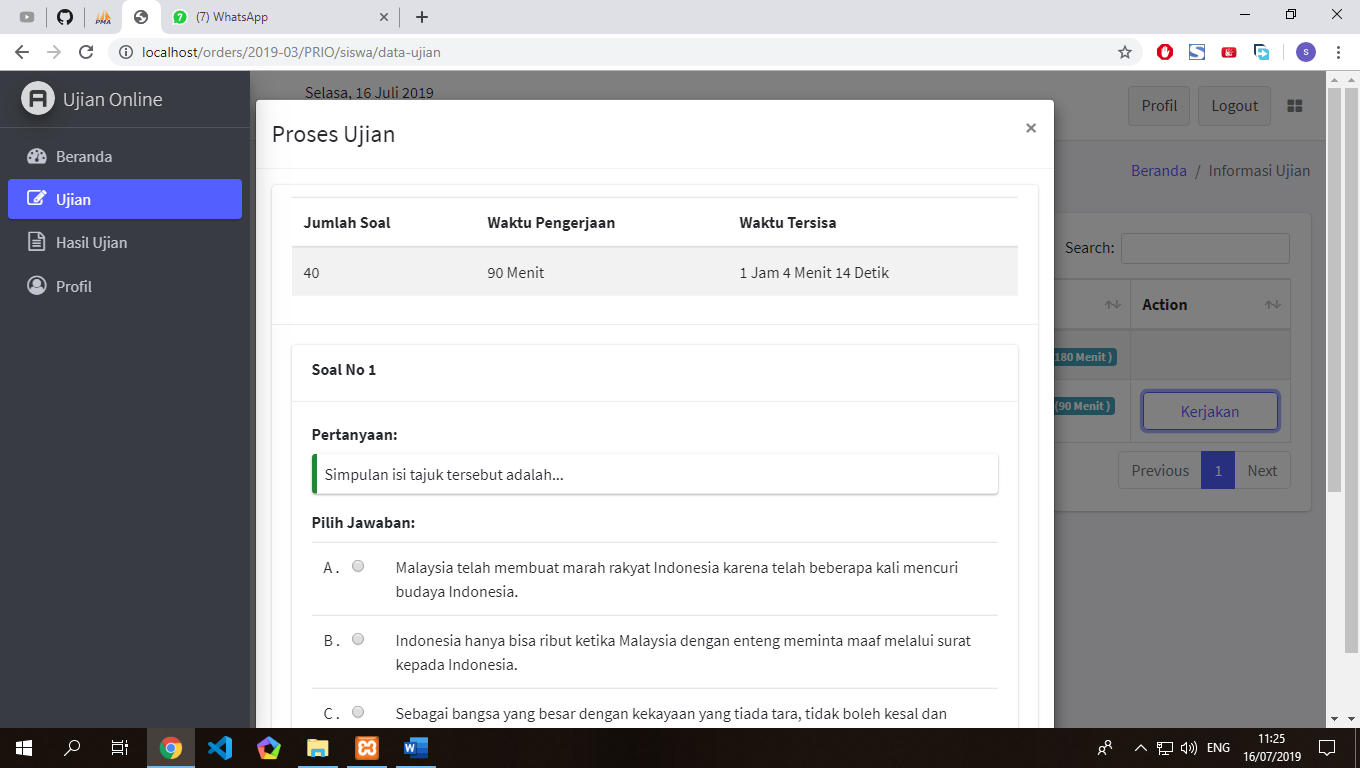
$this->view= 'siswa/ujian';

$this->render\_pages();

}

**33. Halaman Siswa Proses Ujian**

Halaman ini berfungsi untuk menampilkan proses ujian berlangsung. Lebih jelasnya dapat dilihat pada Gambar 5.43



**Gambar 5.43** Halaman Siswa Proses Ujian

Untuk mengetahui script proses Halaman Siswa Proses Ujian dapat dilihat pada Tabel 5.42.

Tabel 5.42 Script Halaman Siswa Proses Ujian

public function proses\_ujian()

{

if ( $this->session->has\_userdata('proses-ujian') ) {

# get last\_soal

$index\_soal=

$this->session->userdata('proses-ujian')['last\_soal'];

if ( ($index\_soal <

count( $this->session->userdata('proses-ujian')['soal']) &&

!empty($this->input->post('jawaban')) ) ) {

# update soal->jawaban

$this->session->userdata('proses-ujian')['soal']

[$index\_soal]['jawaban']

$\_SESSION['proses-ujian']['soal']

[$index\_soal]->jawaban= $this->input->post('jawaban');

# update $this->session

->userdata('proses-ujian')['last\_soal']

$\_SESSION['proses-ujian']['last\_soal']=

$index\_soal +1;

} else {

// $this->session->unset\_userdata('proses-ujian');

}

} else {

$this->m\_siswa->post['id\_grup\_soal']=

$this->uri->segment(3);

$rows= $this->metode\_acak();

$data\_session = array(

'id\_grup\_soal' => $this->uri->segment(3),

'metode' => $rows['metode'],

'start\_ujian' => 'waktu start',

'end\_ujian' => 'waktu berakhir',

'last\_soal'=>0,

'soal'=> $rows['results'],

);

$this->session

->set\_userdata('proses-ujian',$data\_session);

}

$this->html= '

<div class="card">

<div class="card-header">

<div class="table-responsive-lg">

<table class="table table-striped">

<thead>

<tr>

<th>Jumlah Soal</th>

<th>Waktu Pengerjaan</th>

<th>Waktu Tersisa</th>

</tr>

</thead>

<tbody>

<tr>

<td>'.count( $this

->session->userdata('proses-ujian')['soal'] ).'</td>

<td>'.$this

->session->userdata('proses-ujian')['metode']

->bminutes.' Menit</td>

<td id="countDownUjian" data-tanggal="'.$this

->session->userdata('proses-ujian')['metode']->bdaytime.'" data-waktu="'.$this->session->userdata('proses-ujian')['metode']->bminutes.'"></td>

</tr>

</tbody>

</table>

</div>

</div>

<div class="card-body">

';

if ( ($this->session->userdata('proses-ujian')

['last\_soal'] < count( $this->session

->userdata('proses-ujian')['soal'])&& empty($this

->input->get('timeout') ) ) ) {

$soal= $this->session

->userdata('proses-ujian')['soal'][ $this

->session->userdata('proses-ujian')['last\_soal'] ];

$this->html .= '

<div class="card">

<form method="post" class="next-soal"

action="'.base\_url('siswa/proses-ujian').'">

<div class="card-header"> <label>Soal No '.($this->session->userdata('proses-ujian')['last\_soal'] +1).'</label></div>

<div class="card-body">

<div class="form-group">

<label for="pertanyaan">Pertanyaan:</label>

<div class="callout callout-success">

<p>'.$soal->soal.'</p>

</div>

</div>

<label for="pilihJawaban">Pilih Jawaban:</label>

<table class="table">

<tbody>

<tr>

<td style="width:3rem">A .</td>

<td style="width:1rem">

<input type="radio" class="form-check-input" name="jawaban" value="A" required="">

</td>

<td>'.$soal->a.'</td>

</tr>

<tr>

<td>B .</td>

<td>

<input type="radio" class="form-check-input" name="jawaban" value="B" required="">

</td>

<td>'.$soal->b.'</td>

</tr>

<tr>

<td>C .</td>

<td>

<input type="radio" class="form-check-input" name="jawaban" value="C" required="">

</td>

<td>'.$soal->c.'</td>

</tr>

<tr>

<td>D .</td>

<td>

<input type="radio" class="form-check-input" name="jawaban" value="D" required="">

</td>

<td>'.$soal->d.'</td>

</tr>

</tbody>

</table>

</div>

<div class="card-footer">

<button type="submit" class="btn btn-block btn-outline-primary">Klik Disini Untuk Melihat Soal Selanjutnya</button>

</div>

</form>

</div>

';

} else {

/\* create data for insert batch \*/

$this->m\_siswa->post['data\_jawaban']= [];

$benar= 0;

foreach ($this->session->userdata('proses-ujian')['soal'] as $key => $value) {

$keterangan= ($value->jawaban==''

? 'Tidak Terjawab'

: ($value->jawaban==$value->kunci

? 'Benar'

: 'Salah'

)

);

$benar += ($keterangan=='Benar' ? 1 : 0);

$this->m\_siswa->post['data\_jawaban'][]= [

'soal\_id'=>$value->id\_soal,

'nis'=>$this->session->userdata('username'),

'jawaban\_soal'=> ($value->jawaban != '' ? $value->jawaban : '-' ),

'keterangan'=>$keterangan

];

}

$this->m\_siswa->post['data\_hasil\_ujian']= [

'nis'=>$this->session->userdata('username'),

'id\_grup\_soal'=>$this->session->userdata('proses-ujian')['id\_grup\_soal'],

'nilai'=> ($benar/$this->session->userdata('proses-ujian')['metode']->jumlah\_soal)\*100,

];

$this->m\_siswa->store\_data\_jawaban();

$this->m\_siswa->store\_data\_hasil\_ujian();

$this->session->unset\_userdata('proses-ujian');

$this->html .= '

<div class="callout callout-info text-center">

<h5>Ujian Selesai</h5>

<p>\*Terima Kasih\*</p>

</div>

';

}

$this->html .= '

</div>

</div>

';

echo $this->html;

// $this->m\_siswa->post['id\_grup\_soal']= $this->uri->segment(3);

// echo '<pre>';

// print\_r( $\_SESSION['proses-ujian'] );

// print\_r( $this->m\_siswa );

// echo '</pre>';

// echo '<pre>';

// print\_r($\_SESSION['proses-ujian']['soal'][$index\_soal]['jawaban']);

// print\_r($this->input->post());

// print\_r($this->session->userdata('proses-ujian'));

// print\_r($this->session->userdata('proses-ujian')['soal'][ $this->session->userdata('proses-ujian')['last\_soal'] ]);

// print\_r($this->session->has\_userdata('proses-ujian'));

// echo '</pre>';

}

protected function metode\_acak()

{

$data = [];

// $this->m\_siswa->post['id\_grup\_soal']= $this->session->userdata('proses-ujian')['id\_grup\_soal'];

$row\_grup\_soal= $this->m\_siswa->data\_grup\_soal();

$data['metode']= json\_decode($row\_grup\_soal->metode\_acak);

// $data['metode']= $this->m\_siswa->data\_grup\_soal();

if ( $data['metode']->metode == 'LCG' ) {

/\* get no urut dalam kelas berdasarkan nis\*/

$\_no= null;

foreach ($this->m\_siswa->data\_siswa\_satu\_kelas($row\_grup\_soal->id\_pelajaran) as $key => $value) {

if ( $value->nis==$this->session->userdata('username') )

$\_no= $key+1;

}

/\* get data soal \*/

$row\_soal= [];

foreach ($this->m\_siswa->data\_soal\_lcg() as $key => $value) {

$value->kunci= $value->jawaban;

$value->jawaban= '';

$row\_soal[$key+1]= $value;

}

/\* generate soal dengan LCG \*/

$soal=[];

$xn=0;

for ($i=0; $i < $data['metode']->jumlah\_soal ; $i++) {

if ( $i==0 ) {

# code...

$xn= ( (1 \*$\_no) +$data['metode']->bil\_prima ) % $data['metode']->total\_soal;

$soal[]= $row\_soal[($xn==0? $data['metode']->total\_soal : $xn )];

} else {

$xn= ( (1 \*$xn) +$data['metode']->bil\_prima ) % $data['metode']->total\_soal;

$soal[]= $row\_soal[($xn==0? $data['metode']->total\_soal : $xn )];

}

}

$data['results'] = $soal;

} else { # metode SQL Random

/\* get no urut dalam kelas berdasarkan nis\*/

$\_no= null;

foreach ($this->m\_siswa->data\_siswa\_satu\_kelas($row\_grup\_soal->id\_pelajaran) as $key => $value) {

if ( $value->nis==$this->session->userdata('username') )

$\_no= $key+1;

}

$this->m\_siswa->post['random\_parameter']= $\_no;

$this->m\_siswa->post['jumlah\_soal']= $data['metode']->jumlah\_soal;

/\* get data soal \*/

$row\_soal= [];

foreach ($this->m\_siswa->data\_soal\_sql() as $key => $value) {

$value->kunci= $value->jawaban;

$value->jawaban= '';

// $row\_soal[$key+1]= $value;

$row\_soal[$key]= $value;

}

$data['results'] = $row\_soal;

}

return $data;

}

public function cek\_proses\_ujian()

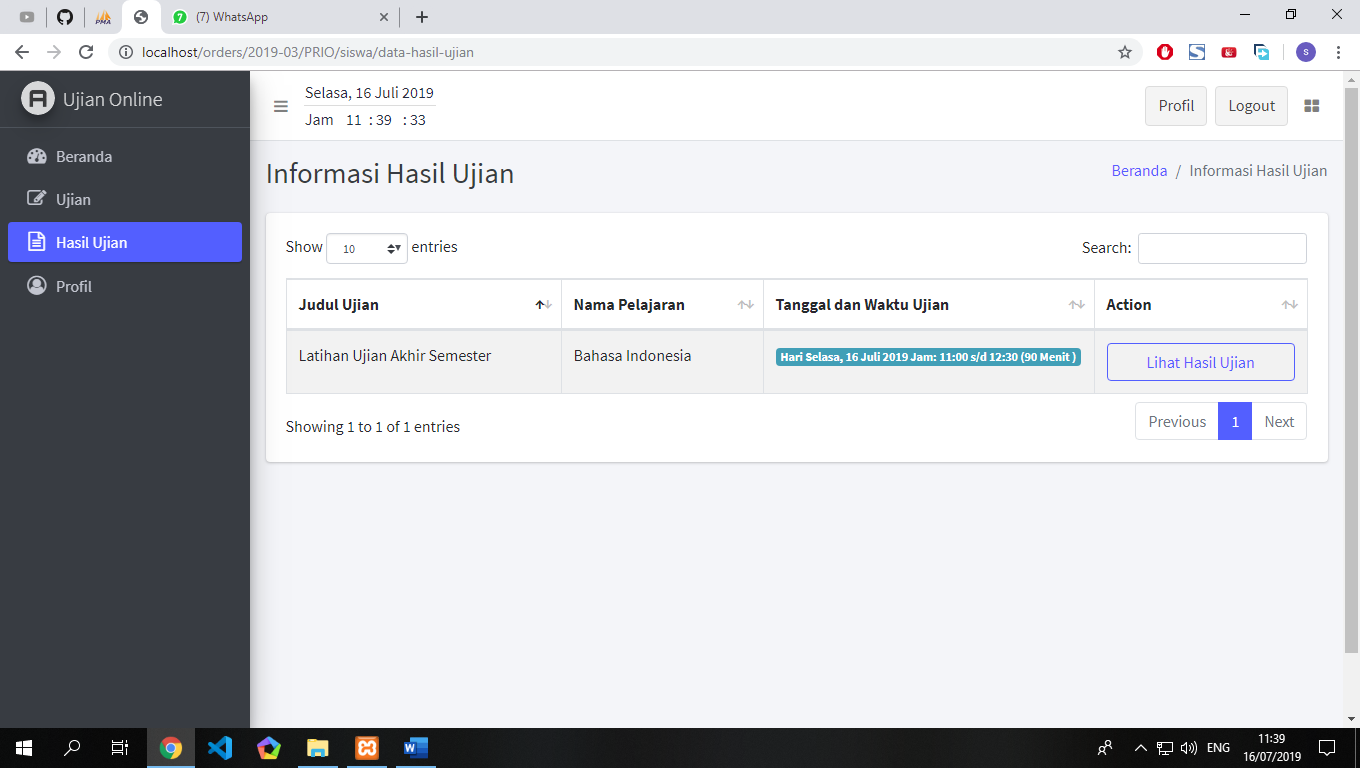
{

echo $this->session->has\_userdata('proses-ujian')? 1 : 0;

}

**34. Halaman Siswa Data Hasil Ujian**

Halaman ini berfungsi untuk menampilkan informasi data hasil ujian atau ujian yang sudah pernah diselesaikan. Untuk lebih jelasnya dapat dilihat pada Gambar 5.44



**Gambar 5.44** Halaman Siswa Hasil Ujian

Untuk mengetahui script proses Halaman Siswa Hasil Ujian dapat dilihat pada Tabel 5.43.

Tabel 5.43 Script Halaman Siswa Hasil Ujian

public function data\_hasil\_ujian()

{

$this->content['rows']= $this->m\_siswa->data\_ujian();

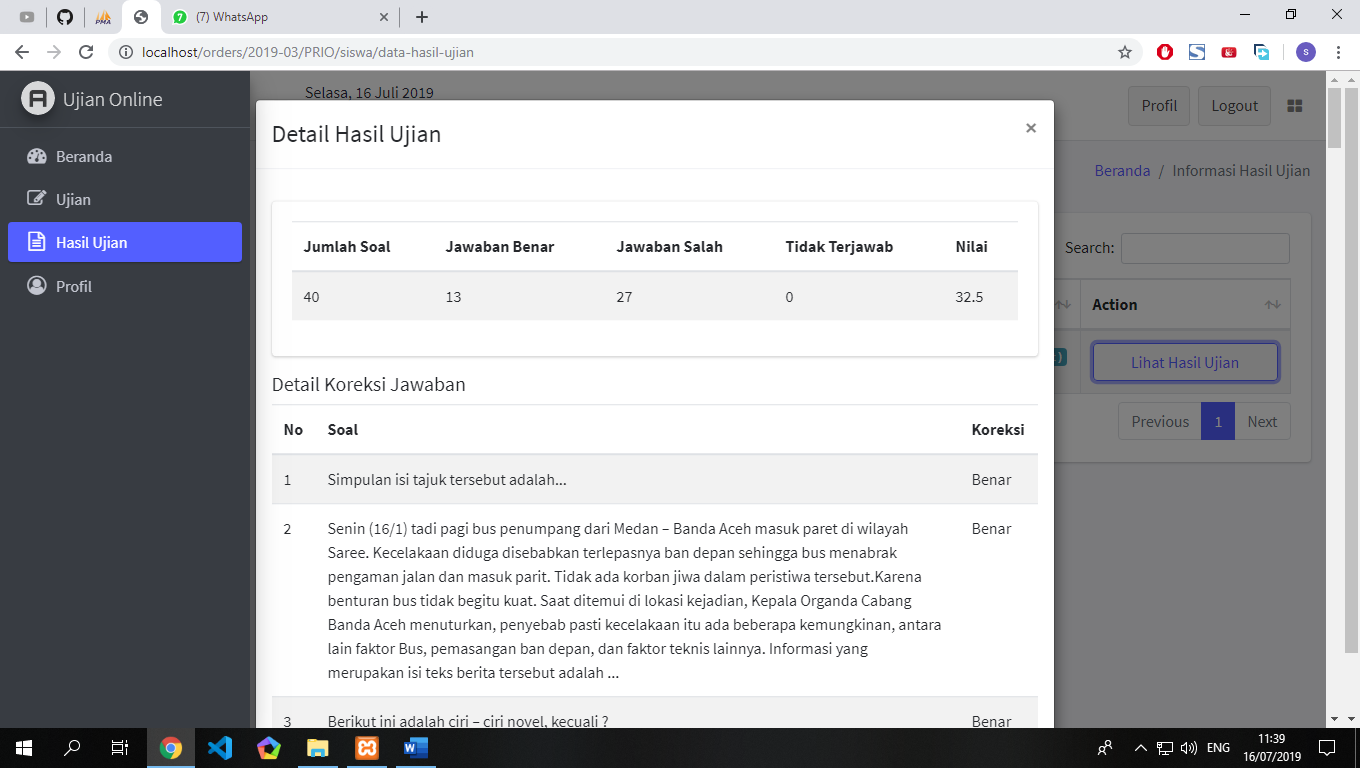
$this->view= 'siswa/hasil\_ujian';

$this->render\_pages();

}

**35. Halaman Siswa Detail Data Hasil Ujian**

Halaman ini berfungsi untuk menampilkan detail informasi hasil ujian. Untuk lebih jelasnya dapat dilihat pada Gambar 5.45



**Gambar 5.45** Halaman Siswa Detail Hasil Ujian

Untuk mengetahui script proses Halaman Siswa Detail Hasil Ujian dapat dilihat pada Tabel 5.44.

Tabel 5.44 Script Halaman Siswa Detail Hasil Ujian

public function detail\_hasil\_ujian()

{

$this->m\_siswa->post['id\_grup\_soal']= $this->uri->segment(3);

$jawaban = $this->m\_siswa->data\_soal\_hasil\_ujian();

$jumlah\_soal = count($jawaban);

$jawaban\_benar = 0;

$jawaban\_salah = 0;

$tidak\_terjawab = 0;

$no = 1;

$tbody = '';

foreach ($jawaban as $key => $value) {

switch ($value->keterangan) {

case 'Benar':

$jawaban\_benar ++;

break;

case 'Salah':

$jawaban\_salah ++;

break;

case 'Tidak Terjawab':

$tidak\_terjawab ++;

break;

default:

# code...

break;

}

$tbody .= "

<tr>

<td>{$no}</td>

<td>{$value->soal}</td>

<td>{$value->keterangan}</td>

</tr>

";

$no++;

}

echo '<pre>';

// print\_r($jawaban\_salah);

echo '</pre>';

$this->html= '

<div class="card">

<div class="card-body">

<div class="table-responsive-lg">

<table class="table table-striped">

<thead>

<tr>

<th>Jumlah Soal</th>

<th>Jawaban Benar</th>

<th>Jawaban Salah</th>

<th>Tidak Terjawab</th>

<th>Nilai</th>

</tr>

</thead>

<tbody>

<tr>

<td>'.$jumlah\_soal.'</td>

<td>'.$jawaban\_benar.'</td>

<td>'.$jawaban\_salah.'</td>

<td>'.$tidak\_terjawab.'</td>

<td>'.( ($jawaban\_benar/$jumlah\_soal)\*100 ).'</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

<div class="table-responsive-lg">

<h5>Detail Koreksi Jawaban</h5>

<table class="table table-striped">

<thead>

<tr>

<th>No</th>

<th>Soal</th>

<th>Koreksi</th>

</tr>

</thead>

<tbody>

'.$tbody.'

</tbody>

</table>

</div>

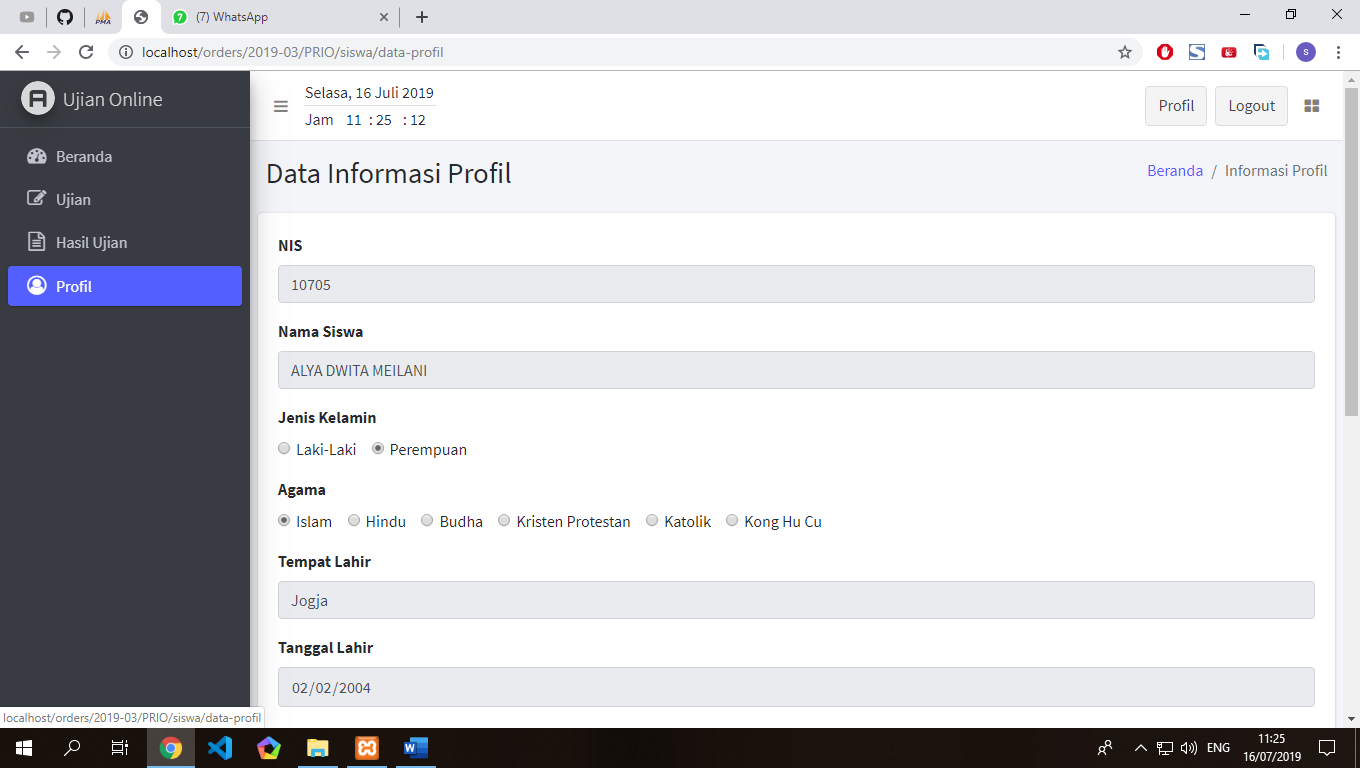
';

echo $this->html;

}

**34. Halaman Siswa Data Profil**

Halaman ini berfungsi untuk menampilkan informasi data profil siswa yang sedang login saat ini. Untuk lebih jelasnya dapat dilihat pada gambar 5.46



**Gambar 5.46 Halaman Siswa Data Profil**

Untuk mengetahui script proses Halaman Siswa Data Profil dapat dilihat pada Tabel 5.45.

Tabel 5.45 Script Halaman Siswa Data Profil

public function data\_siswa\_update()

{

$this->m\_siswa->post= $this->input->post();

if ( empty($\_FILES['fupload']['tmp\_name']) ) {

# code...without upload file

if ( $this->m\_siswa->data\_siswa\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan'

];

} else {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Gagal Disimpan'

];

}

} else {

# code...with upload file

$this->m\_siswa->username= $this->input->post('username');

$row= $this->m\_siswa->data\_siswa\_edit();

$config['upload\_path'] = 'src/siswa/';

$config['allowed\_types'] = 'jpg|png';

if ( file\_exists($config['upload\_path'].$row->gambar) ) {

unlink($config['upload\_path'].$row->gambar);

}

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('fupload'))

{

$this->msg= [

'stats'=>0,

'msg'=> $this->upload->display\_errors(),

];

}

else

{

$this->m\_siswa->post['gambar']= $this->upload->data()['file\_name'];

if ( $this->m\_siswa->data\_siswa\_update() ) {

$this->msg= [

'stats'=>1,

'msg'=> 'Data Berhasil Disimpan',

];

} else {

$this->msg= [

'stats'=>0,

'msg'=> 'Maaf Data Gagal Disimpan',

];

}

}

}

echo json\_encode($this->msg);

}

Dari Gambar (33-40) dapat disimpulkan bahwa percobaan metode berhasil karena tidak terjadi pengulangan pada soal yang sama pada satu sesi.

Dari Gambar 5.43 Proses Ujian, dapat disimpulkan bahwa hasil percobaan metode sama dengan hasil pengacakan waktu ujian berlangsung.