

## **Honor.php**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Honor extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');

        $this->load->model('honor/honor_model');
    }

    public function generate() {
        if($this->input->get('remove')) {
            $honor_no = $this->input->get('remove');
            $this->honor_model->DelNoHonor($honor_no);
            redirect('honor/honor/generate');
        }
        $data['prodis'] = $this->honor_model->getProdi();
        $data['noHonors'] = $this->honor_model->getNoHonor();
        Template::render('honor/generate_honor_v', $data);
    }

    public function onGenerate() {
        if($_POST) {
            $honor = $this->honor_model->createInvoice($_POST);
            $data['honor_no'] = $honor['honor_no'];
        }
    }
}
```

```

        $data['honor_id'] = $honor['honor_id'];
        $data['honor_teoris'] = $this->honor_model->getHonorTeori($_POST);
        $data['honor_praktikum'] = $this->honor_model-
>getHonorPraktikum($_POST);
        $data['prodis'] = $this->honor_model->getProdi();
        $data['noHonors'] = $this->honor_model->getNoHonor();
        Template::render('honor/generate_honor_v', $data);
    }
}

public function insertHonor() {
    if($_POST) {
        if(isset($_POST['jadwal_teoris'])) {
            $this->honor_model->insertHonorTeori($_POST['jadwal_teoris']);
        }

        if(isset($_POST['jadwal_praktikum'])) {
            $this->honor_model-
>insertHonorPraktikum($_POST['jadwal_praktikum']);
        }

        redirect('honor/honor/generate');
    }
}

public function searchHonor() {
    if($_GET) {
        $data['honor_teoris'] = $this->honor_model-
>searchHonorTeori($_GET['search']);
    }
}

```

```

        $data['honor_praktikum'] = $this->honor_model-
>searchHonorPraktikum($_GET['search']);
        $data['prodis'] = $this->honor_model->getProdi();
        $data['noHonors'] = $this->honor_model->getNoHonor();
        Template::render('honor/generate_honor_v', $data);
    }
}
}

```

### **Kurikulum.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Kurikulum extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');

        if($_SESSION['role_nama'] != 'penjadwalan' && $_SESSION['role_nama']
!= 'super admin') {
            $this->session->set_flashdata('err', 'Anda tidak mempunyai akses ke view
Kurikulum. ');
            redirect('dashboard');
        }

        $this->load->model('kurikulum/kurikulum_model');
    }
}

```

```
public function index() {
    $data['prodis'] = $this->kurikulum_model->getProdis();

    Template::render('kurikulum/kurikulum_v', $data);
}

public function kurikulum_ajax() {
    $prodi = $this->input->post('prodi');
    $tahun = $this->input->post('thn');
    $data['prodis'] = $this->kurikulum_model->getProdis();
    $data['smt'] = $this->kurikulum_model-
>getkurikulum2($prodi,$tahun);
    $this->load->view('kurikulum/kurikulum_load_v', $data);
}

public function changeProdi() {
    if($_POST) {
        $data['query'] = [
            'kurikulum_year' => $this->input->post('kurikulum_year'),
            'prodi_id' => $this->input->post('prodi_id')
        ];
        $data['prodis'] = $this->kurikulum_model->getProdis();
        $data['kurikulums'] = $this->kurikulum_model-
>getKurikulums($data['query']);
        $this->load->view('kurikulum/kurikulum_load_v', $data);
    }
}
```

```

public function crud($id = NULL) {
    if($_POST) {
        $result = $this->kurikulum_model->crud($_POST);
        $this->session->set_flashdata('arr', $result);
        redirect('kurikulum/kurikulum');
    }

    $data['prodis'] = $this->kurikulum_model->getProdis();
    $data['kurikulum'] = $this->kurikulum_model->crud($id);
    $data['semesters'] = $this->kurikulum_model->getSemesters();
    Template::render('kurikulum/kurikulum_crud_v', $data);
}

public function delete($id) {
    $result = $this->kurikulum_model->delete($id);
    redirect('kurikulum/kurikulum');
}
}

```

### **Mail.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Mail extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');

        $this->load->model('mail/mail_model');
    }
}

```

```
}

public function index() {
    $data['inboxes'] = $this->mail_model->getInbox();
    Template::render('mail/inbox_new_v', $data);
}

public function compose() {
    if($_POST && $_FILES) {
        $url = $this->do_upload();
        $to = explode("-", $this->input->post('to'));
        $data = [
            'inbox_file' => $url,
            'to' => $to[0],
            'broadcast' => $this->input->post('broadcast'),
            'inbox_subjek' => $this->input->post('inbox_subjek'),
            'inbox_konten' => $this->input->post('inbox_konten')
        ];
        $result = $this->mail_model->crudInbox($data);
        $this->session->set_flashdata('err', $result);
    }

    $data['sends'] = $this->mail_model->getSend();
    $data['inboxes'] = $this->mail_model->getInbox();

    $data['trash'] = $this->mail_model->getTrash();

    //Template::render('mail/compose_v');
    Template::render('mail/compose_new_v',$data);
}
```

```
}

private function do_upload()
{
    $type = explode('.', $_FILES["attach_file"]["name"]);
    $type = $type[count($type)-1];
    $url = "./files/" . uniqid(rand()) . ".$type";
    if(is_uploaded_file($_FILES["attach_file"]["tmp_name"]))
    {

        if(move_uploaded_file($_FILES["attach_file"]["tmp_name"], $url))
            return $url;
        // return "";
    }
}

public function inbox() {
    $data['inboxes'] = $this->mail_model->getInbox();
    $data['trash'] = $this->mail_model->getTrash();
    $data['sends'] = $this->mail_model->getSend();

    Template::render('mail/inbox_new_v', $data);
}

public function ajaxInbox() {
    $result = $this->mail_model->getAjaxInbox();
    echo $result;
}
```

```
public function send() {
    $data['sends'] = $this->mail_model->getSend();
    $data['inboxes'] = $this->mail_model->getInbox();

    $data['trash'] = $this->mail_model->getTrash();
    Template::render('mail/send_new_v', $data);
}

public function crud($id = NULL) {
    if($_POST) {
        $result = $this->mail_model->crud($_POST);
        $this->session->set_flashdata('arr', $result);
        redirect('mail/mail');
    }

    $data['prodis'] = $this->mail_model->getProdis();
    $data['mail'] = $this->mail_model->crud($id);
    $data['semesters'] = $this->mail_model->getSemesters();
    Template::render('mail/mail_crud_v', $data);
}

    public function reply() {
        $data['receiver'] = $_GET['sender'];
        $data['sends'] = $this->mail_model->getSend();
        $data['inboxes'] = $this->mail_model->getInbox();
        $data['trash'] = $this->mail_model->getTrash();

        Template::render('mail/compose_new_v',$data);
    }

public function delete() {
```

```

        $data = array(
            'aktif' => 'Tidak'
        );
        $id = explode('-', $_GET['id']);
        $this->mail_model->trash($id[0], $data);
        redirect('mail/mail/inbox');
    }

    public function mail_list()
    {
        $term = $_GET['term'];
        $rows = $this->mail_model->cari($term);
        $json_array = array();
        foreach ($rows as $row)
            $json_array[] = ('.$row->pegawai_id.' - '.$row->nama_pegawai.') '.$row->inbox_subjek;

        echo json_encode($json_array);
    }

    public function read($id) {
        $data['sends'] = $this->mail_model->getSend();
        $data['inboxes'] = $this->mail_model->getInbox();
        $data['read'] = $this->mail_model->getRead($id);
        $data['trash'] = $this->mail_model->getTrash();
        Template::render('mail/read_new_v', $data);
    }

    public function read_send($id) {

```

```

    $data['sends'] = $this->mail_model->getSend();
    $data['inboxes'] = $this->mail_model->getInbox();
    $data['read'] = $this->mail_model->getReadSend($id);
    $data['trash'] = $this->mail_model->getTrash();
    Template::render('mail/send_read_new_v', $data);
}

public function trash() {
    $data['trash'] = $this->mail_model->getTrash();
    $data['inboxes'] = $this->mail_model->getInbox();
    $data['sends'] = $this->mail_model->getSend();
    Template::render('mail/trash_new_v', $data);
}

    public function totrash(){
        $ids = ( explode( ',', $this->input->get_post('ids') ));
        $this->mail_model->totrash($ids);
    }

    public function deltrash(){
        $ids = ( explode( ',', $this->input->get_post('ids') ));
        $this->mail_model->deltrash($ids);
    }
}
}

```

### **Materi.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class Materi extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');

        $this->load->model('materi/materi_model');
        $this->load->model('penjadwalan/jadwal_teor_i_model');
    }

    public function materi($id) {
        $data['id'] = $id;
        $data['jadwal_teor_i'] = $this->materi_model->getJadwalTeori($id);
        $data['materi'] = $this->materi_model->getMateri($id);
        Template::render('materi/materi_v', $data);
    }

    public function materi_teor_i($jadwal_teor_i_id, $pegawai_id,
$kurikulum_id,$sesi) {
        $data['jadwal_teor_i'] = $this->jadwal_teor_i_model-
>getSelectedJadwalTeori($jadwal_teor_i_id);
        $data['mahasiswa'] = $this->jadwal_teor_i_model-
>getMahasiswa($jadwal_teor_i_id);
        $data['details'] = $this->jadwal_teor_i_model->getDetailMateri($pegawai_id,
$kurikulum_id,$sesi);
        Template::render('materi/materi_teor_i_v', $data);
    }

    public function isiMateriTeori($jadwal_teor_i_id, $edit = NULL)
    {
        if($_POST) {

```

```

    $jadwal_teori_id = $this->uri->segment(4, 0);
    $this->materi_model->crudMateriTeori($jadwal_teori_id, $_POST);
    redirect('materi/materi/isiMateriTeori/'.$jadwal_teori_id);
}

if($edit)
{
    $data['edit'] = true;
}

$data['jadwal_teori'] = $this->jadwal_teori_model-
>getSelectedJadwalTeori($jadwal_teori_id);
    $data['mahasiswa'] = $this->jadwal_teori_model-
>getMahasiswa($jadwal_teori_id);
    Template::render('materi/isi_materi_teori_v',$data);
}

public function crud($id = NULL, $edit = NULL) {
    if($_POST) {
        $result = $this->materi_model->crud($_POST);
        $this->session->set_flashdata('arr', $result);
        redirect('materi/materi/materi/'.$_POST['jadwal_teori_id']);
    }

    if($edit) {
        $data['materi'] = $this->materi_model->crud($id, true);
    }

    $data['jadwal_teori_id'] = $id;
    Template::render('materi/materi_crud_v', $data);
}

public function delete($id, $jadwal_teori_id) {

```

```

$result = $this->materi_model->delete($id);
    redirect('materi/materi/materi/'. $jadwal_teorid);
}
}

```

### **Pegawai.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Pegawai extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');

        if($_SESSION['role_nama'] != 'sdm' && $_SESSION['role_nama'] !=
'super admin') {
            $this->session->set_flashdata('err', 'Anda tidak mempunyai akses ke view
pegawai. ');
            redirect('dashboard');
        }

        $this->load->model('pegawai/pegawai_model');
    }

    public function index() {
        $data['pegawais'] = $this->pegawai_model->getPegawais();
        Template::render('pegawai/pegawai_v', $data);
    }
}

```

```

}

public function crud($id = NULL) {
    if($_POST) {
        $result = $this->pegawai_model->crud($_POST);
        $this->session->set_flashdata('arr', $result);
        redirect('pegawai/pegawai');
    }
    $data['roles'] = $this->pegawai_model->getRole();
    $data['pegawai'] = $this->pegawai_model->crud($id);
    $data['prodis'] = $this->pegawai_model->getProdis();
    Template::render('pegawai/pegawai_crud_v', $data);
}

public function delete($id) {
    $result = $this->pegawai_model->delete($id);
    redirect('pegawai/pegawai');
}
}

```

### **Jadwal\_Praktikum.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Jadwal_Praktikum extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');
    }
}

```

```

        $this->load->model('penjadwalan/jadwal_praktikum_model');
    }

    public function praktikum($kurikulum_id,$sesi) {
        $data['id'] = $kurikulum_id;
        $data['sesi'] = $sesi;
        $data['jadwal_teoris'] = $this->jadwal_praktikum_model-
>getJadwalTeori($kurikulum_id,$sesi);
        $data['jadwal_praktikums'] = $this->jadwal_praktikum_model-
>getJadwalPraktikums($kurikulum_id,$sesi);
        $data['mahasiswa'] = $this->jadwal_praktikum_model-
>getMahasiswa($kurikulum_id);
        $data['pegawais'] = $this->jadwal_praktikum_model->getPegawai();
        Template::render('penjadwalan/jadwal_praktikum_v', $data);
    }

    public function crud($id = NULL, $sesi = NULL, $sedit = FALSE) {
    if($_POST) {
        $result = $this->jadwal_praktikum_model->crud($_POST);
        $this->session->set_flashdata('arr', $result);

        redirect('penjadwalan/jadwal_praktikum/praktikum/'.$_POST['kurikul
um_id'].'/'.$_POST['sesi']);
    }
    if($sedit) {
        $data['jadwal_praktikum'] = $this->jadwal_praktikum_model->crud($id);
        $data['jadwal_teoris'] = $this->jadwal_praktikum_model-
>getJadwalTeori($sedit);
    }
}

```

```

} else {
    $data['jadwal_teor'] = $this->jadwal_praktikum_model-
>getJadwalTeori($id,$sesi);
}

    $data['pegawais'] = $this->jadwal_praktikum_model->getPegawai();
    Template::render('penjadwalan/jadwal_praktikum_crud_v', $data);
}

    public function detail($kelompok, $pegawai_id, $kurikulum_id, $sesi)
    {
        $data['jadwal_teor'] = $this->jadwal_praktikum_model-
>getJadwalTeori($kurikulum_id, $sesi);
        $data['mahasiswa'] = $this->jadwal_praktikum_model-
>getMahasiswa($kurikulum_id);
        $data['details'] = $this->jadwal_praktikum_model->getDetail($kelompok,
$pegawai_id, $kurikulum_id,$sesi);
        Template::render('penjadwalan/detail_praktikum_v', $data);
    }

    public function editDetail($kelompok,$pegawai_id, $kurikulum_id,
$sesi) {
        if($_POST) {
            $this->jadwal_praktikum_model->crudDetail($_POST);

            redirect('penjadwalan/jadwal_praktikum/detail/'.$kelompok.'/'.$pegaw
ai_id.'/'.$kurikulum_id.'/'.$sesi);
        }
        $data['kelompok'] = $kelompok;
        $data['pegawai_id'] = $pegawai_id;
        $data['jadwal_teor_id'] = $kurikulum_id;

```

```

    $data['sesi'] = $sesi;
        $data['jadwal_teoris'] = $this->jadwal_praktikum_model-
>getJadwalTeori($kurikulum_id, $sesi);
    $data['pegawais'] = $this->jadwal_praktikum_model->getPegawai();
        $data['jadwal_praktikum'] = $this-
>jadwal_praktikum_model->getDetail($kelompok, $pegawai_id,
$kurikulum_id,$sesi);
    $this->load->view('penjadwalan/crud_detail_praktikum_v', $data);
    }
    public function delete($kelompok, $pegawai_id, $kurikulum_id, $sesi)
    {
        $result = $this->jadwal_praktikum_model->delete($kelompok,
$pegawai_id, $kurikulum_id, $sesi);
        redirect('penjadwalan/jadwal_praktikum/praktikum/'.$kurikulum_id.'/'.
$sesi);
    }
    public function editOneDetail($id = NULL) {
        if($_POST) {
            $this->jadwal_praktikum_model->editOne($_POST);

            redirect('penjadwalan/jadwal_praktikum/detail/'.$_POST['kelompok'].'
/'.$_POST['pegawai_id'].'/'.$_POST['kurikulum_id'].'/'.$_POST['sesi']);
        }

        $data['kelompok'] = $_GET['kelompok'];
        $data['pegawai_id'] = $_GET['pegawai_id'];
        $data['jadwal_teoris_id'] = $_GET['kurikulum_id'];
        $data['sesi'] = $_GET['sesi'];
        $data['jadwal_praktikum'] = $this->jadwal_praktikum_model-

```

```

>getOne($id);
    $this->load->view('penjadwalan/crud_one_praktikum_v', $data);
}

public function deleteOneDetail($id) {
    $this->jadwal_praktikum_model->deleteOne($id);

    redirect('penjadwalan/jadwal_praktikum/detail/'.$_GET['kelompok'].'/
$_GET['pegawai_id'].'/'.$_GET['kurikulum_id'].'/'.$_GET['sesi']);
}

// Materi
public function materi($kelompok,$pegawai_id,$kurikulum_id,$sesi) {
    $data['jadwal_teoris'] = $this->jadwal_praktikum_model-
>getJadwalTeori($kurikulum_id,$sesi);
    $data['mahasiswa'] = $this->jadwal_praktikum_model-
>getMahasiswa($kurikulum_id);
    $data['details'] = $this->jadwal_praktikum_model-
>getDetailMateri($kelompok,$pegawai_id,$kurikulum_id,$sesi);
    Template::render('penjadwalan/materi_praktikum_v', $data);
}

public function isimateri_praktikum($pegawai_id, $kurikulum_id, $sesi,
$jadwal_praktikum_id, $edit = NULL) {
    if($_POST) {
        $kel = $_GET['kel'];
        $peg = $_GET['peg'];
        $mk = $_GET['mk'];
        $sesinya = $_GET['sesi'];
        $id_jadwal_praktikum = $_GET['jadwal_praktikum'];
    }
}

```

```

        $pegawai_id = $this->uri->segment(4,0);
        $kurikulum_id = $this->uri->segment(5, 0);
        $this->jadwal_praktikum_model->crudMateri($pegawai_id,
$jadwal_praktikum_id, $_POST);

        redirect('penjadwalan/jadwal_praktikum/isimateri_praktikum/'.$peg.'/'.$
$mk.'/'.$sesinya.'/'.$id_jadwal_praktikum);
    }
    if($edit) {
        $data['edit'] = true;
    }
    $data['jadwal_teoris'] = $this->jadwal_praktikum_model-
>getJadwalTeori($kurikulum_id,$sesi);
    $data['mahasiswa'] = $this->jadwal_praktikum_model-
>getMahasiswa($kurikulum_id);
    $data['detail'] = $this->jadwal_praktikum_model-
>getDetail_isimateri($jadwal_praktikum_id);

        Template::render('penjadwalan/materi_praktikum_crud_v',
$data);
    }
}

```

### Jadwal\_teoris.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Jadwal_teoris extends CI_Controller

```

```
{  
    public function __construct() {  
        parent::__construct();  
        my_session('login');  
  
        if($_SESSION['role_nama'] != 'penjadwalan' && $_SESSION['role_nama']  
!= 'super admin' && $_SESSION['role_nama'] != 'prodi' &&  
$_SESSION['role_nama'] != 'pengajar') {  
            $this->session->set_flashdata('err', 'Anda tidak mempunyai akses ke view  
penjadwalan.');
```

            redirect('dashboard');

```
        }  
  
        $this->load->model('penjadwalan/jadwal_teor_i_model');  
    }  
    public function index() {  
        $data = [  
            'jadwal_teoris' => $this->jadwal_teor_i_model->get_jadwal_teor_i(),  
            'prodis' => $this->jadwal_teor_i_model->getProdi()  
        ];  
        Template::render('penjadwalan/jadwal_teor_i_v', $data);  
    }  
    public function crud($id = NULL) {  
        if($_SESSION['role_nama'] != 'penjadwalan' && $_SESSION['role_nama']  
!= 'super admin') {  
            $this->session->set_flashdata('err', 'Anda tidak mempunyai akses ke view  
penjadwalan.');
```

            redirect('penjadwalan/jadwal\_teor\_i');

```
        }  
    }  
}
```

```

if($_POST) {
    $result = $this->jadwal_teori_model->crud($_POST);
    $this->session->set_flashdata('arr', $result);
    redirect('penjadwalan/jadwal_teori');
}
$data['jadwal_teori'] = $this->jadwal_teori_model->crud($id);
$data['prodis'] = $this->jadwal_teori_model->getProdi();
    $data['kurikulums'] = $this->jadwal_teori_model-
>getKurikulums();
    $data['pegawais'] = $this->jadwal_teori_model->getPegawais();
    Template::render('penjadwalan/jadwal_teori_crud_v', $data);
}
//public function delete($pegawai_id, $kurikulum_id) {
    public function delete($prodi, $kurikulum_id, $pegawai_id, $sesi) {
        $result = $this->jadwal_teori_model->delete($prodi,
$kurikulum_id, $pegawai_id, $sesi);
        redirect('penjadwalan/jadwal_teori');
    }

    public function deleteOneDetail($id) {
        $this->jadwal_teori_model->deleteOne($id);
        redirect('penjadwalan/jadwal_teori');
    }

    public function detail($jadwal_teori_id, $pegawai_id, $kurikulum_id,$sesi) {
        $data['jadwal_teori'] = $this->jadwal_teori_model-
>getSelectedJadwalTeori($jadwal_teori_id);
        $data['mahasiswa'] = $this->jadwal_teori_model-
>getMahasiswa($jadwal_teori_id);
        $data['details'] = $this->jadwal_teori_model->getDetail($pegawai_id,

```

```

$kurikulum_id,$sesi);
    Template::render('penjadwalan/detail_teor_i_v', $data);
}
public function cariJadwal() {
    if($_POST)
    {
        $queries['query'] = [
            'prodi_id' => $this->input->post('prodi_id'),
            'periode' => $this->input->post('periode'),
            'basis' => $this->input->post('basis'),
            'tahun' => $this->input->post('tahun')
        ];
    }
    $data['jadwal_teoris'] = $this->jadwal_teor_i_model-
>getCariJadwal($queries['query']);
    $data['prodis'] = $this->jadwal_teor_i_model->getProdi();
    Template::render('penjadwalan/jadwal_teor_i_v', $data);

}
//public function editDetail($jadwal_teor_i_id) {
    public function editDetail($jadwal_teor_i_id, $pegawai_id,
$kurikulum_id,$sesi) {
        if($_POST)
        {
            $update = $this->jadwal_teor_i_model-
>crudSelectedJadwalTeori($_POST);
            if(!empty($update))
            {
                redirect('penjadwalan/jadwal_teor_i');
            }
        }
    }
}

```

```

    }
}
$data['jadwal_teoris'] = $this->jadwal_teoris_model-
>getSelectedJadwalTeori($jadwal_teoris_id);
$data['pegawais'] = $this->jadwal_teoris_model->getPegawais();
$data['prodis'] = $this->jadwal_teoris_model->getProdi();
$data['kurikulums'] = $this->jadwal_teoris_model->getKurikulums();
$this->load->view('penjadwalan/crud_detail_teoris_v', $data);
}
public function editPertemuanTeori($jadwal_teoris_id) {
    if($_POST)
    {
        $this->jadwal_teoris_model->editOne($_POST);
        redirect('penjadwalan/jadwal_teoris/detail/'.$_POST['jadwal_teoris_id'].'
/'.$_POST['pegawai_id'].'/'.$_POST['kurikulum_id'].'/'.$_POST['sesi']);
    }
    $data['jadwal_teoris'] = $this->jadwal_teoris_model-
>getSelectedJadwalTeori($jadwal_teoris_id);
    $this->load->view('penjadwalan/crud_pertemuan_teoris_v', $data);
}
public function changeMatkul($id = NULL) {
    if($id) {
        $kurikulum = $this->jadwal_teoris_model->changeMatkul($id);
        $option = "";
        foreach($kurikulum as $k) {
            $option .= "<option value='".$k->kurikulum_id.">{$k-
>kurikulum_nama}</option>";
        }
        echo $option;
    }
}

```

```

    }
  }
}

```

### Report.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Report extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');
        $this->load->model('report/report_model');
    }
    public function honor() {
        $data['prodis'] = $this->report_model->getProdi();
        $data['noHonors'] = $this->report_model->getNoHonor();
        Template::render('report/report_honor_v', $data);
    }
    public function rekap() {
        if($_POST) {
            $data['pengajar'] = $this->report_model-
>getPengajar($_POST['noHonor']);
            $data['isi'] = $this->report_model-
>getLabRekap($_POST['noHonor']);
            $this->load->view('report/report_rekap_ok_v.php', $data);
        }
    }
    public function detailHonor() {

```

```

if($_GET) {
    $data['pengajar'] = $this->report_model-
>getPengajar($_GET['noHonor']);
    $this->load->view('report/report_detail_ok_v', $data);
}
}
}

```

### **Semester.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Semester extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');
        $this->load->model('semester/semester_model');
    }
    public function index() {
        $data['semesters'] = $this->semester_model->getSemesters();
        Template::render('semester/semester_v', $data);
    }

    public function crud($id = NULL) {
        if($_POST) {
            $result = $this->semester_model->crud($_POST);
            $this->session->set_flashdata('arr', $result);
            redirect('semester/semester');
        }
    }
}

```

```

    $data['semester'] = $this->semester_model->crud($id);
    Template::render('semester/semester_crud_v', $data);
}
public function delete($id) {
    $result = $this->semester_model->delete($id);
    redirect('semester/semester');
}
}

```

### **Slipgaji.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class SlipGaji extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');

        $this->load->model('slipgaji/slipgaji_model');
    }
    public function index() {
        $data['prodis'] = $this->slipgaji_model->getProdi();
        Template::render('slipgaji/slip_v', $data);
    }

    public function rekap() {
        if($_POST) {
            $no = explode('-',$_POST['noPengajar']);
            $bln = $this->input->post('bln_pengajuan');

```

```

    $thn = $this->input->post('thn_pengajuan');
    $data['isi'] = $this->slipgaji_model->getLabRekap($no[0],$bln,$thn);
    $data['pengajar'] = $this->slipgaji_model->getPengajar($no[0],$bln,$thn);
    $this->load->view('slipgaji/slip_rekap_v.php', $data);
}
}

public function detailGaji() {
    if($_GET) {
        $no = explode('-',$_GET['pengajar']);
        $bln = $_GET['bln'];
        $thn = $_GET['thn'];
        $data['isi'] = $this->slipgaji_model->getLabDetail($no[0],$bln,$thn);
        $data['pengajar'] = $this->slipgaji_model->getPengajar($no[0],$bln,$thn);
        $this->load->view('slipgaji/slip_detail_v', $data);
    }
}

    public function anggota_list()
{
    $term = $_GET['term'];
        $nama = $this->input->post('noPengajar',TRUE);
        $rows = $this->slipgaji_model->getAnggota($term);

    $json_array = array();
    foreach ($rows as $row)
    $json_array[]=$row->pegawai_id.'-'.$row->nama_pegawai;
    echo json_encode($json_array);
}
}
}

```

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class User extends CI_Controller
{
    public function __construct() {
        parent::__construct();
        my_session('login');
        $this->load->model('user/user_model');
    }
    public function index() {
        $msg = 'password';
        $data['isi'] = $this->encrypt->encode($msg);
        $hash =
'$2y$10$YCFsG6e1Yca568hBi2pZ0.3LDL5wjxct1N8w/oLR/jfHsiQwCqTS';
        $options = array('cost' => 11);
        $data['isi'] = password_needs_rehash($hash, PASSWORD_DEFAULT,
$options);
        $data['pegawais'] = $this->user_model->getPegawais();
        Template::render('user/user_list_v', $data);
    }
    public function crud($id = NULL) {
        if($_POST) {
            $result = $this->user_model->crud($_POST);
            $this->session->set_flashdata('arr', 'Password berhasil dirubah. ');
            redirect('user/user/profile');
        }
    }
    public function edit($id = NULL) {
        if($_POST) {

```

```

    $data = array(
        'pegawai_password' => $this->input->post('pegawai_password'),
    );
    $result = $this->user_model->edit($data);
    $this->session->set_flashdata('arr', 'Password berhasil dirubah. ');
    redirect('user/user/edit/'.$this->uri->segment(4));
}
    $data['pegawai'] = $this->user_model->detail($id);
    Template::render('user/user_edit_v', $data);
}
public function delete($id) {
    $result = $this->user_model->delete($id);
    redirect('user/user');
}
public function profile() {
    Template::render('user/user_crud_v');
}
}

```

### Dashboard.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Dashboard extends CI_Controller
{
    function __construct() {
        parent::__construct();
        my_session('login');
        if(!isset($_SESSION['id'])) {
            $this->session->set_flashdata('err', 'Wrong username or password. ');

```

```

        redirect('login');
    }
}

public function index() {
    Template::render('dashboard');
}

public function logout() {
    $this->session->unset_userdata(['id', 'name']);
    session_destroy();
    redirect('login');
}
}

```

### **Login.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Login extends CI_Controller
{
    function __construct() {
        parent::__construct();
        $this->load->model('login_model', 'login');
        if(isset($_SESSION['id']))
            redirect('dashboard');
    }
    public function index() {
        if($_POST) {
            $idpegawai = $this->input->post('pegawai_id');
            $password = $this->input->post('password');
            $result = $this->login->validate($idpegawai, $password);

```

```
if(!empty($result)){
    $this->session->set_userdata([
        'id' => $result->pegawai_id,
        'name' => $result->nama_pegawai,
        'role_id' => $result->role_id,
        'role_nama' => $result->role_nama
    ]);
    redirect('dashboard');
} else {
    $this->session->set_flashdata('err', 'Wrong username or password. ');
    redirect('login');
}
$this->load->view('login');
}
```

**Modul.php**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Modul extends CI_Controller
{
    function __construct() {
        parent::__construct();
    }
    public function index() {
        $this->load->view('modules/modul');
    }
}
```

```
}

```

### **Honor\_model.php**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Honor_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }

    public function getProdi() {
        $query = $this->db->get('prodi')->result();
        return $query;
    }

    public function getNoHonor() {
        $query = $this->db->query("select * from honor where
status='Aktif'")->result();
        return $query;
    }

    public function DelNoHonor($no) {
        $query = $this->db->query("update honor set status='Tidak' where
honor_no='$no'");
        $query = $this->db->query("update detail_honor set status='Tidak'
where honor_id=(select honor_id from honor where honor_no='$no')");
        return $query;
    }

    public function createInvoice($data) {
        $honor_no =

```

```

$data['prodi']..'/'.$data['month_pembayaran']..'/'.$data['tahun_pembayaran']./'.$sprintf("%03d", ($this->db->count_all('honor')+1));
    $arr = array(
        'honor_no' => $honor_no,
        'prodi_kode' => $data['prodi'],
        'honor_pembayaran' => $data['month_pembayaran'],
        'tahun_pembayaran' => $data['tahun_pengajuan'],
        'honor_pengajuan' => $data['month_pengajuan'],
        'tahun_pengajuan' => $data['tahun_pembayaran'],
        'honor_basis' => $data['basis'],
        'created_at' => date('Y-m-d')
    );
    $this->db->insert('honor', $arr);
    return array(
        'honor_no' => $honor_no,
        'honor_id' => $this->db->insert_id()
    );
}
public function getHonorTeori($data) {
    $this->db->select('prodi.prodi_id,
    prodi.prodi_kode,
    prodi_nama,
        jadwal_teor_i.jadwal_teor_i_id,
        jadwal_teor_i.pegawai_id,
        jadwal_teor_i.jadwal,
        jadwal_teor_i.basis,
        jadwal_teor_i.status_materi,
        jadwal_teor_i.periode,
        jadwal_teor_i.tahun,

```

```

        jadwal_teor_i.jenis,
        jadwal_teor_i.kurikulum_id,
        jadwal_teor_i.sesi,

jadwal_teor_i.pertemuan,
        kurikulum.kurikulum_nama,
        kurikulum.kurikulum_sks,
        pegawai.nama_pegawai,
        pegawai.rate');
    $this->db->join('jadwal_teor_i', 'jadwal_teor_i.prodi_id =
prodi.prodi_id', 'left');
    $this->db->join('kurikulum', 'jadwal_teor_i.kurikulum_id =
kurikulum.kurikulum_id');
    $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_teor_i.pegawai_id');
    $this->db->where('prodi.prodi_kode', $data['prodi']);
    $this->db->where('jadwal_teor_i.tahun', $data['tahun']);
    $this->db->where("MONTH(jadwal_teor_i.jadwal) =
".$data['month_pembayaran'], NULL, FALSE);
    $this->db->where("YEAR(jadwal_teor_i.jadwal) =
".$data['tahun_pembayaran'], NULL, FALSE);
    $this->db->where('jadwal_teor_i.basis', $data['basis']);
    $this->db->where('jadwal_teor_i.periode', $data['honor_periode']);
    $this->db->where('jadwal_teor_i.status_materi', 'selesai');
    $this->db->where("jadwal_teor_i.jadwal_teor_i_id NOT IN (SELECT
detail_honor.jadwal_id FROM detail_honor WHERE
detail_honor.honor_jadwal = 'jadwal_teor_i' and detail_honor.status='Aktif)",
NULL, FALSE);
    $this->db->order_by('jadwal_teor_i.pertemuan asc');

```

```

$query = $this->db->get('prodi')->result();
// xlog($data['month_pembayaran']);
// xlog($query);exit;
return $query;
}
public function getHonorPraktikum($data) {
    $this->db->select('jadwal_praktikum.jadwal_praktikum_id,
    jadwal_praktikum.jadwal,
    jadwal_praktikum.pegawai_id,
    jadwal_praktikum.kelompok,
    jadwal_teor_i.jadwal_teor_i_id,
    jadwal_teor_i.periode,
    jadwal_teor_i.tahun,
    jadwal_teor_i.jenis,
    jadwal_teor_i.basis,
    jadwal_praktikum.status_materi,
    jadwal_praktikum.pertemuan,
    prodi.prodi_id,
    prodi.prodi_kode,
    prodi_nama,
    jadwal_teor_i.kurikulum_id,
    kurikulum.kurikulum_nama,
    kurikulum.kurikulum_sks,
    jadwal_teor_i.sesi,
    pegawai.nama_pegawai,
    pegawai.rate');
    $this->db->join('jadwal_teor_i', 'jadwal_teor_i.kurikulum_id =
    jadwal_praktikum.jadwal_teor_i_id');
    $this->db->join('prodi', 'jadwal_teor_i.prodi_id = prodi.prodi_id');

```

```

        $this->db->join('kurikulum', 'jadwal_teorikurikulum_id =
kurikulum.kurikulum_id');
        $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_praktikum.pegawai_id');
        $this->db->where("MONTH(jadwal_praktikum.jadwal) =
".$data['month_pembayaran'], NULL, FALSE);
        $this->db->where("YEAR(jadwal_praktikum.jadwal) =
".$data['tahun_pembayaran'], NULL, FALSE);
            $this->db->where('prodi.prodi_kode', $data['prodi']);
            $this->db->where('jadwal_teoritahun', $data['tahun']);
            $this->db->where('jadwal_teoribasis', $data['basis']);
            $this->db->where('jadwal_teoriperiode', $data['honor_periode']);
            $this->db->where('jadwal_praktikum.status_materi', 'selesai');

        $this->db->where("jadwal_praktikum.jadwal_praktikum_id NOT
IN (SELECT detail_honor.jadwal_id FROM detail_honor WHERE
detail_honor.honor_jadwal = 'jadwal_praktikum' and
detail_honor.status='Aktif)", NULL, FALSE);

        $this->db->group_by('jadwal_praktikum.jadwal_praktikum_id');
        $this->db->order_by('jadwal_praktikum.pertemuan asc');
        $query = $this->db->get('jadwal_praktikum')->result();
        return $query;
    }
    public function insertHonorTeori($data) {
        $new_data = array();
        for($i=0; $i < count($data); $i++) {
            $new_data[$i] = explode(',', $data[$i]);
        }
        for($j=0; $j < count($new_data); $j++) {

```

```

        $arr[] = array(
            'honor_id' => $new_data[$j][0],
            'honor_jadwal' => 'jadwal_teoris',
            'jadwal_id' => $new_data[$j][1]
        );
    }
    $this->db->insert_batch('detail_honor', $arr);
}

public function insertHonorPraktikum($data) {
    $new_data = array();
    for($i=0; $i < count($data); $i++) {
        $new_data[$i] = explode(',', $data[$i]);
    }
    for($j=0; $j < count($new_data); $j++) {
        $arr[] = array(
            'honor_id' => $new_data[$j][0],
            'honor_jadwal' => 'jadwal_praktikum',
            'jadwal_id' => $new_data[$j][1]
        );
    }

    $this->db->insert_batch('detail_honor', $arr);
}

public function searchHonorTeori($search) {
    $this->db->select('detail_honor.detail_id,honor.honor_id,
        honor.honor_no,
        honor.prodi_kode,
        honor.honor_basis as basis,
        detail_honor.honor_jadwal,
        detail_honor.jadwal_id,

```



```
public function searchHonorPraktikum($search) {
    $this->db->select('detail_honor.detail_id,honor.honor_id,
        honor.honor_no,
        honor.prodi_kode,
        honor.honor_basis as basis,
        detail_honor.honor_jadwal,
        detail_honor.jadwal_id,
        jadwal_praktikum.jadwal_teorid_id,
        jadwal_praktikum.pegawai_id,
        jadwal_praktikum.jadwal,
        jadwal_praktikum.kelompok,
        jadwal_praktikum.pertemuan,
        jadwal_teorid.prodi_id,
        jadwal_teorid.kurikulum_id,
        jadwal_teorid.sesi,
        kurikulum.kurikulum_nama,
        kurikulum.kurikulum_sks,
        pegawai.nama_pegawai,
        pegawai.rate');

    $this->db->join('detail_honor', 'detail_honor.honor_id =
honor.honor_id');

    $this->db->join('jadwal_praktikum',
'jadwal_praktikum.jadwal_praktikum_id = detail_honor.jadwal_id');

    $this->db->join('jadwal_teorid', 'jadwal_teorid.kurikulum_id =
jadwal_praktikum.jadwal_teorid_id');

    $this->db->join('kurikulum', 'jadwal_teorid.kurikulum_id =
kurikulum.kurikulum_id');

    $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_praktikum.pegawai_id');
```

```

        $this->db->where('honor.honor_no', $search);
    $this->db->where('detail_honor.honor_jadwal', 'jadwal_praktikum');
        $this->db->where('jadwal_praktikum.status_materi', 'selesai');
        $this->db->group_by('detail_honor.detail_id');
        $this->db->order_by('jadwal_praktikum.pertemuan asc');
    $query = $this->db->get('honor')->result();
    return $query;
    }
function __destruct() {
    $this->db->close();
}
}

```

### **Kurikulum\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class kurikulum_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }

    public function getSemesters() {
        $query = $this->db->get('semester')->result();
        return $query;
    }

    public function getProdi() {
        $query = $this->db->get('prodi')->result();
    }
}

```

```

        return $query;
    }

    public function getkurikulum2($prodi,$tahun) {
        $sql = $this->db-
>where(array('prodi_id'=>$prodi,'kurikulum_year'=>$tahun))-
>group_by('semester_id')->get('kurikulum');
        return $sql;
    }

    public function getkurikulums($prodi, $tahun, $id) {
        $this->db->join('prodi', 'prodi.prodi_id = kurikulum.prodi_id');
        $this->db->join('semester', 'semester.semester_id =
kurikulum.semester_id');
        $this->db->where('prodi.prodi_id', $prodi);
        $this->db->where('kurikulum.kurikulum_year', $tahun);
        $this->db->where('kurikulum.semester_id',$id);
        $query = $this->db->get('kurikulum')->result();
        return $query;
    }

    public function crud($data = NULL) {
        if(!is_array($data)) {
            $query = $this->db->get_where('kurikulum', array('kurikulum_id' =>
$data))->row();
            return $query;
        }
        $kurikulum_id = trim($data['kurikulum_id']);
        $data = [
            'kurikulum_kode' => trim($data['kurikulum_kode']),
            'kurikulum_nama' => trim($data['kurikulum_nama']),
            'kurikulum_sks' => trim($data['kurikulum_sks']),

```

```

'kurikulum_teoris' => trim($data['kurikulum_teoris']),
'kurikulum_praktikum' => trim($data['kurikulum_praktikum']),
'semester_id' => trim($data['semester_id']),
'prodi_id' => trim($data['prodi_id']),
'kurikulum_year' => trim($data['kurikulum_year'])
];
if(!empty($kurikulum_id)) {
    $this->db->where('kurikulum_id', $kurikulum_id);
    $this->db->update('kurikulum', $data);
} else {
    $data['created_at'] = date('Y-m-d');
    $this->db->insert('kurikulum', $data);
}
}
}
public function delete($id) {
    $this->db->where('kurikulum_id', $id);
    $this->db->delete('kurikulum');
}
function __destruct() {
    $this->db->close();
}
}
}

```

```

Mail_model.php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Mail_model extends CI_Model
{
    function __construct() {
        parent::__construct();
    }
}

```

```

$this->load->database();
}

    public function getInbox() {
        $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.sender_id');
        $query = $this->db->order_by('inbox.created_at desc')->get_where('inbox',
array('receiver_id' => $_SESSION['id'],'aktif'=>'Ya'));
        return $query;
    }

    public function getAjaxInbox_old() {
        $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.sender_id');
        $query = $this->db->get_where('inbox', array('receiver_id' =>
$_SESSION['id']))->num_rows();
        return $query;
    }

    public function getAjaxInbox() {
        $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.sender_id');
        $query = $this->db->get_where('inbox', array('receiver_id' =>
$_SESSION['id'],'status' => 'Unread'))->num_rows();
        return $query;
    }

    public function crudInbox_old($data) {
        if(!empty($data['broadcast'])) {
            $this->db->where('pegawai_id !=', $_SESSION['id']);
            $this->db->where('role_id !=', 5);
            $pegawai = $this->db->get('pegawai')->result();
            foreach ($pegawai as $r) {
                $data = [
                    'inbox_file' => $data['inbox_file'],
                    'inbox_subjek' => $data['inbox_subjek'],

```

```

        'inbox_konten' => $data['inbox_konten'],
        receiver_id => $r->pegawai_id,
        'sender_id' => $_SESSION['id'],
        'created_at' => date('Y-m-d H:i:s')
    ];
    $this->db->insert('inbox', $data);
}
return "Sukses mengirim email";
}
foreach ($data['to'] as $to) {
    $arr[] = [
        'inbox_file' => $data['inbox_file'],
        'inbox_subjek' => $data['inbox_subjek'],
        'inbox_konten' => $data['inbox_konten'],
        'receiver_id' => $to,
        'sender_id' => $_SESSION['id'],
        'created_at' => date('Y-m-d H:i:s')
    ];
}
try {
    $this->db->insert_batch('inbox', $arr);
    return "Sukses mengirim email";
} catch (Exception $e) {
    return "Gagal kirim email {$e}";
}
}
public function crudInbox($data) {
    $arr[] = [
        'inbox_file' => $data['inbox_file'],

```

```

        'inbox_subjek' => $data['inbox_subjek'],
        'inbox_konten' => $data['inbox_konten'],
        'receiver_id' => $data['to'],
        'sender_id' => $_SESSION['id'],
        'created_at' => date('Y-m-d H:i:s')
    ];
    try {
        $this->db->insert_batch('inbox', $arr);
        return "Sukses mengirim email";
    } catch (Exception $e) {
        return "Gagal kirim email {$e}";
    }
}

public function getSend() {
    $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.receiver_id');
    $query = $this->db->get_where('inbox', array('sender_id' =>
$_SESSION['id'],'aktif'=>'Ya'));
    return $query;
}

public function crud($data = NULL) {
    if(!is_array($data)) {
        $query = $this->db->get_where('mail', array('mail_id' => $data))->row();
        return $query;
    }
    $mail_id = trim($data['mail_id']);
    $data = [
        'mail_kode' => trim($data['mail_kode']),
        'mail_nama' => trim($data['mail_nama']),
        'mail_sks' => trim($data['mail_sks']),
    ];
}

```

```

        'mail_teori' => trim($data['mail_teori']),
                'mail_praktikum' =>
trim($data['mail_praktikum']),
        'semester_id' => trim($data['semester_id']),
        'prodi_id' => trim($data['prodi_id']),
        'mail_year' => trim($data['mail_year'])
];
if(!empty($mail_id)) {
    $this->db->where('mail_id', $mail_id);
    $this->db->update('mail', $data);
} else {
    $data['created_at'] = date('Y-m-d');
    $this->db->insert('mail', $data);
}
}
public function delete($id) {
    $this->db->where('inbox_id', $id);
    $this->db->delete('inbox');
}
public function trash($id, $data) {
    $this->db->where('inbox_id', $id);
    $this->db->update('inbox', $data);
}
function cari($nama)
{
    $query = $this->db->query("select a.*,b.pegawai_id, b.nama_pegawai
from inbox a left join pegawai b on b.pegawai_id=a.sender_id or
b.pegawai_id=a.receiver_id
where pegawai_id like '%$nama%' or nama_pegawai like

```

```

'%nama%');
        return $query->result();
    }

    public function getTrash() {
        $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.sender_id');
        $query = $this->db->order_by('inbox.created_at desc')-
>where(array('receiver_id' => $_SESSION['id'],'aktif'=>'Tidak'))-
>or_where('sender_id',$_SESSION['id'])->get('inbox');
        return $query;
    }

    public function getRead($id) {
        $this->db->where('inbox_id', $id);
        $this->db->update('inbox', array('status'=>'Read'));
        $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.sender_id');
        $query = $this->db->order_by('inbox.created_at desc')->get_where('inbox',
array('inbox_id' => $id))->row();
        return $query;
    }

    public function getReadSend($id) {
        $this->db->join('pegawai', 'pegawai.pegawai_id = inbox.receiver_id');
        $query = $this->db->order_by('inbox.created_at desc')->get_where('inbox',
array('inbox_id' => $id))->row();
        return $query;
    }

    function totrash($ids){
        $ids = $ids;
        $count = 0;
        foreach ($ids as $id){
            $this->db->where('inbox_id', $id);

```

```

        $this->db->update('inbox', array('aktif'=>'Tidak'));
        $count = $count+1;
    }
}

function deltrash($sids){
    $sids = $sids;
    $count = 0;
    foreach ($sids as $sid){
        $this->db->where('inbox_id', $sid);
        $this->db->delete('inbox');
        $count = $count+1;
    }
}

function __destruct() {
    $this->db->close();
}
}

```

### **Materi\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Materi_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }

    public function getJadwalTeori($sid) {
        $this->db->join('kurikulum', 'kurikulum.kurikulum_id =

```

```

jadwal_teori.kurikulum_id');
        $this->db->join('prodi', 'prodi.prodi_id = jadwal_teori.prodi_id');
        $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_teori.pegawai_id');
        $query = $this->db->get_where('jadwal_teori', array('jadwal_teori_id'
=> $id))->row();
        return $query;
    }

    public function getMateri($jadwal_teori_id) {
        $query = $this->db->get_where('materi', array('jadwal_teori_id' =>
$jadwal_teori_id))->result();
        return $query;
    }

    public function crud($data = NULL, $edit = FALSE) {
        if($edit) {
            $query = $this->db->get_where('materi', array('materi_id' => $data))-
>row();
            return $query;
        }
        $materi_id = trim($data['materi_id']);
        $data = [
            'materi_isi' => trim($data['materi_isi']),
            'jadwal_teori_id' => trim($data['jadwal_teori_id']),
            'materi_status' => trim($data['materi_status'])
        ];
        if(!empty($materi_id)) {
            $this->db->where('materi_id', $materi_id);
            $this->db->update('materi', $data);
        } else {

```

```

        $data['created_at'] = date('Y-m-d');
        $this->db->insert('materi', $data);
    }
}

public function crudMateriTeori($jadwal_teorid, $data) {
    $data = array(
        'materi' => trim($data['materi']),
        'status_materi' => $data['status_materi'],
        'pegawai_id' => $data['pegawai_id']
    );
    $this->db->where('pegawai_id', $data['pegawai_id']);
    $this->db->where('jadwal_teorid', $jadwal_teorid);
    $this->db->update('jadwal_teorid', $data);
}

public function delete($id) {
    $this->db->where('materi_id', $id);
    $this->db->delete('materi');
}

function __destruct() {
    $this->db->close();
}
}

```

### **Pegawai\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Pegawai_model extends CI_Model
{
    private $_mongoDb;
    function __construct() {

```

```
parent::__construct();
$this->load->database();
}
public function getPegawais() {
    $this->db->join('role', 'role.role_id = pegawai.role_id', 'left');
    $this->db->join('prodi', 'prodi.prodi_id = pegawai.prodi_id', 'left');
    $query = $this->db->get('pegawai')->result();
    return $query;
}
public function getProdis() {
    $query = $this->db->get('prodi')->result();
    return $query;
}
public function getRole() {
    $query = $this->db->get('role')->result();
    return $query;
}
public function crud($data = NULL) {
    if(!is_array($data)) {
        $query = $this->db->get_where('pegawai', array('pegawai_id' => $data))-
>row();
        return $query;
    }
    $pegawai_id = trim($data['pegawai_id']);
    $tgl_lahir = Datetime::createFromFormat('d/m/Y', $data['tanggal_lahir']);
    $data = [
        'nama_pegawai' => trim($data['nama_pegawai']),
        'tempat_lahir' => trim($data['tempat_lahir']),
        'tgl_lahir' => $tgl_lahir->format('Y-m-d'),
```

```

'no_ktp' => trim($data['no_ktp']),
'pendidikan_terakhir' => trim($data['pendidikan_terakhir']),
'alamat' => trim($data['alamat']),
'jenis_kelamin' => trim($data['jenis_kelamin']),
'no_rekening' => trim($data['no_rekening']),
'agama' => trim($data['agama']),
'role_id' => trim($data['role_id']),
'unit' => trim($data['unit']),
'rate' => trim($data['rate']),
'pajak' => trim($data['pajak']),
'prodi_id' => ($data['prodi_id'] == "NULL") ? NULL : $data['prodi_id']
];
if(!empty($pegawai_id)) {
    $this->db->where('pegawai_id', $pegawai_id);
    $this->db->update('pegawai', $data);
} else {
    $data['pegawai_password'] = $tgl_lahir->format('dmy');
    $data['created_at'] = date('Y-m-d');
    $this->db->insert('pegawai', $data);
}
}
public function delete($id) {
    $this->db->where('pegawai_id', $id);
    $this->db->delete('pegawai');
}
function __destruct() {
    $this->db->close();
}
}

```

**Jadwal\_praktikum\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Jadwal_praktikum_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }

    public function getJadwalTeori($id,$sesi) {
        $this->db-
>select('a.*,c.prodi_nama,b.kurikulum_kode,b.kurikulum_nama,d.nama_pegawa
i');

        $this->db->join('kurikulum b', 'b.kurikulum_id = a.kurikulum_id');
        $this->db->join('prodi c', 'c.prodi_id = a.prodi_id');
        $this->db->join('pegawai d', 'd.pegawai_id = a.pegawai_id');
        $query = $this->db->get_where('jadwal_teor
array('b.kurikulum_id' => $id,'a.sesi'=>$sesi))->row();
        return $query;
    }

    public function
getDetailMateri($kelompok,$pegawai_id,$jadwal_teorid,$sesi) {
        $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_praktikum.pegawai_id');
        $this->db->where('jadwal_praktikum.pegawai_id', $pegawai_id);
        $this->db->where('jadwal_praktikum.jadwal_teorid', $jadwal_teorid);
        $this->db->where('jadwal_praktikum.kelompok', $kelompok);
        $this->db->where('jadwal_praktikum.sesi', $sesi);

```

```

$query = $this->db->get('jadwal_praktikum')->result();
return $query;
}

public function getDetail_isimateri($jadwal_praktikum_id) {
    $this->db->select('a.*,c.nama_pegawai,d.prodi_nama');
    $this->db->join('jadwal_teoris b', 'b.kurikulum_id = a.jadwal_teoris_id');
    $this->db->join('pegawais c', 'c.pegawai_id = a.pegawai_id');
    $this->db->join('prodis d', 'd.prodi_id = b.prodi_id');
    $this->db->where('a.jadwal_praktikum_id', $jadwal_praktikum_id);
    $query = $this->db->get('jadwal_praktikum a')->row();
    return $query;
}

public function getJadwalPraktikums($id,$sesi) {
    $this->db->select('
        jadwal_teoris.periode,
        jadwal_teoris.sesi,
        jadwal_teoris.basis,
        pegawai.*,
        kurikulum.*,
        prodi.*,
        jadwal_praktikum.*
    ');
    $this->db->join('jadwal_teoris', 'jadwal_teoris.kurikulum_id =
    jadwal_praktikum.jadwal_teoris_id and jadwal_teoris.sesi =
    jadwal_praktikum.sesi');
    $this->db->join('pegawais', 'pegawais.pegawai_id =
    jadwal_praktikum.pegawai_id');
    $this->db->join('kurikulums', 'kurikulums.kurikulum_id =
    jadwal_teoris.kurikulum_id');

```

```

        $this->db->join('prodi', 'prodi.prodi_id = jadwal_teoriprodi_id');
        $this->db->where('jadwal_praktikum.jadwal_teoriprodi_id', $id);
        $this->db->where('jadwal_praktikum.sesi', $sesi);
        $this->db->group_by('jadwal_praktikum.kelompok');
        $query = $this->db->get('jadwal_praktikum')->result();
        // xlog($query);exit;
        return $query;
    }

    public function getMahasiswa($id) {
        $this->db->join('jadwal_mahasiswa', 'jadwal_mahasiswa.mahasiswa_id =
mahasiswa.mahasiswa_id');
        $query = $this->db->get('mahasiswa')->num_rows();
        return $query;
    }

    public function getPegawai() {
        $this->db->where('role_id', 5);
        $query = $this->db->get('pegawai')->result();
        return $query;
    }
}

public function crud_old($data = NULL) {
    if(!is_array($data)) {
        $query = $this->db->get_where('jadwal_praktikum',
array('jadwal_praktikum_id' => $data))->row();
        return $query;
    }
    // Get pertemuan1
    $jadwal1 = Datetime::createFromFormat('d/m/Y', $data['jadwal1']);
    $date1 = $jadwal1->format('Y-m-d');

```

```

for($i = 0; $i < $data['pertemuan1']; $i++) {
    $arr1[] = array(
        'jadwal_teori_id' => 22, //trim($data['jadwal_teori_id']),
        'kapasitas' => trim($data['kapasitas']),
        'pegawai_id' => trim($data['pegawai_id1']),
        'jam_mulai' => trim($data['jam_mulai1']),
        'jam_selesai' => trim($data['jam_selesai1']),
        'materi' => ",
        'ruang' => trim($data['ruang1'])
    );
    if($i == 0) {
        $date1 = date('Y-m-d', strtotime("{$date1} +7 day"));
        $arr1[$i]['jadwal'] = $date1;
    } else {
        $date1 = date("Y-m-d", strtotime("{$date1} +7 day"));
        $arr1[$i]['jadwal'] = $date1;
    }
}
$this->db->insert_batch('jadwal_praktikum', $arr1);
}

public function crud($data = NULL) {
    if(!is_array($data)) {
        $query = $this->db->get_where('jadwal_praktikum',
array('jadwal_praktikum_id' => $data))->row();
        return $query;
    }
    // Get pertemuan1
    $jadwal1 = Datetime::createFromFormat('d/m/Y', $data['jadwal1']);
    $date1 = $jadwal1->format('Y-m-d');
    for($i = 0; $i < $data['pertemuan1']; $i++) {

```

```

    $arr1[] = array(
        'jadwal_teorid' => trim($data['kurikulum_id']),
        'kapasitas' => trim($data['kapasitas']),
        'kelompok' => trim($data['kelompok']),
        'pegawai_id' => trim($data['pegawai_id1']),
        'pertemuan' => $i+1,
        'sesi' => trim($data['sesi']),
        'jam_mulai' => trim($data['jam_mulai1']),
        'jam_selesai' => trim($data['jam_selesai1']),
        'materi' => "",
        'ruang' => trim($data['ruang1'])
    );
    if($i == 0) {
        $arr1[$i]['jadwal'] = $date1;
    } else {
        $date1 = date("Y-m-d", strtotime("{$date1} +7 day"));
        $arr1[$i]['jadwal'] = $date1;
    }
}
$this->db->insert_batch('jadwal_praktikum', $arr1);
}

public function getDetail($kelompok, $pegawai_id,$kurikulum_id,$sesi) {
    $this->db->select("a.*,c.nama_pegawai,b.kurikulum_id");
    $this->db->from('jadwal_praktikum a');
    $this->db->join('jadwal_teorid b',
'a.jadwal_teorid=b.kurikulum_id','inner');
    $this->db->join('pegawai c',
'a.pegawai_id=c.pegawai_id','inner');
    $this->db->where('a.jadwal_teorid', $kurikulum_id);
}

```

```

        $this->db->where('a.kelompok', $kelompok);
        $this->db->where('a.sesi', $sesi);
        $this->db->group_by('a.jadwal_praktikum_id');
        $query = $this->db->get()->result();
        return $query;
    }

    public function crudDetail($data) {
        $data = array(
            'kapasitas' => trim($data['kapasitas']),
            'pegawai_id' => trim($data['pegawai_id'])
        );
        $this->db->where('kelompok', $this->uri->segment(4));
        $this->db->where('pegawai_id', $this->uri->segment(5));
        $this->db->where('jadwal_teorid', $this->uri->segment(6));
        $this->db->where('sesi', $this->uri->segment(7));
        $this->db->update('jadwal_praktikum', $data);
    }

    public function delete($kelompok, $pegawai_id, $kurikulum_id, $sesi) {
        $this->db->where('kelompok', $kelompok);
        $this->db->where('pegawai_id', $pegawai_id);
        $this->db->where('jadwal_teorid', $kurikulum_id);
        $this->db->where('sesi', $sesi);
        $this->db->delete('jadwal_praktikum');
    }

    public function getOne($id) {
        $query = $this->db->get_where('jadwal_praktikum',
array('jadwal_praktikum_id' => $id))->row();
        return $query;
    }
}

```

```

public function editOne($data) {
    $date = Datetime::createFromFormat('d/m/Y', $data['jadwal']);
    $jadwal_praktikum_id = $data['jadwal_praktikum_id'];
    $data = array(
        'jadwal' => $date->format('Y-m-d'),
        'jam_mulai' => $data['jam_mulai'],
        'jam_selesai' => $data['jam_selesai'],
        'ruang' => trim($data['ruang'])
    );
    $this->db->where('jadwal_praktikum_id', $jadwal_praktikum_id);
    $this->db->update('jadwal_praktikum', $data);
}

public function deleteOne($id) {
    $this->db->where('jadwal_praktikum_id', $id);
    $this->db->delete('jadwal_praktikum');
}

// materi

public function crudMateri($pegawai_id, $jadwal_praktikum_id, $data) {
    $data = array(
        'materi' => trim($data['materi']),
        'status_materi' => $data['status_materi']
    );
    $this->db->where('jadwal_praktikum_id', $jadwal_praktikum_id);
    $this->db->update('jadwal_praktikum', $data);
}

}

function __destruct() {
    $this->db->close();
}
}

```

```
}
```

### **Jadwal\_teor\_i\_model.php**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Jadwal_teor_i_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }
    public function getDetailMateri($pegawai_id, $kurikulum_id, $sesi) {
        $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_teor_i.pegawai_id');
        $this->db->where('jadwal_teor_i.pegawai_id', $pegawai_id);
        $this->db->where('jadwal_teor_i.kurikulum_id', $kurikulum_id);
        $this->db->where('jadwal_teor_i.sesi', $sesi);
        $query = $this->db->get('jadwal_teor_i')->result();
        return $query;
    }
    public function getDetail($pegawai_id, $kurikulum_id, $sesi) {
        $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_teor_i.pegawai_id');
        $this->db->where('jadwal_teor_i.pegawai_id', $pegawai_id);
        $this->db->where('jadwal_teor_i.kurikulum_id', $kurikulum_id);
        $this->db->where('jadwal_teor_i.sesi', $sesi);
        $query = $this->db->get('jadwal_teor_i')->result();
        return $query;
    }
}
```

```

public function get_jadwal_teorii() {
    $this->db->select('jadwal_teorii.*, kurikulum.*, prodi.prodi_id,
prodi.prodi_kode, prodi.prodi_nama, pegawai.pegawai_id, pegawai.role_id,
pegawai.nama_pegawai, pegawai.no_rekening, pegawai.rate, pegawai.pajak');
    $this->db->join('prodi', 'jadwal_teorii.prodi_id = prodi.prodi_id');
    $this->db->join('kurikulum', 'jadwal_teorii.kurikulum_id =
kurikulum.kurikulum_id');
    $this->db->join('pegawai', 'jadwal_teorii.pegawai_id = pegawai.pegawai_id');
    if($this->session->userdata('role_id')== 5){
        $this->db->where('jadwal_teorii.pegawai_id', $this->session-
>userdata('id'));
    }
    $this->db->group_by('jadwal_teorii.prodi_id, jadwal_teorii.kurikulum_id,
jadwal_teorii.pegawai_id, jadwal_teorii.sesi');
    $query = $this->db->get('jadwal_teorii')->result();
    return $query;
}

public function getAbsenMahasiswa($mahasiswa_id, $jadwal = NULL) {
    $this->db->where('mahasiswa_id', $mahasiswa_id);
    $this->db->where('jadwal', $jadwal);
    $query = $this->db->get('absen_teorii')->row_array();
    // xlog($query);?
    return $query;
}

public function getMahasiswa($id) {
    $this->db->join('jadwal_mahasiswa', 'jadwal_mahasiswa.mahasiswa_id =
mahasiswa.mahasiswa_id');
    $this->db->where('jadwal_mahasiswa.jadwal_teorii_id', $id);
    $query = $this->db->get('mahasiswa')->num_rows();
}

```

```

    return $query;
}

public function getJadwalTeori($id) {
    $this->db->join('kurikulum', 'kurikulum.kurikulum_id =
jadwal_teor_i.kurikulum_id');
    $this->db->join('prodi', 'prodi.prodi_id = jadwal_teor_i.prodi_id');
    $query = $this->db->get_where('jadwal_teor_i', array('jadwal_teor_i_id' =>
$id))->row();
    return $query;
}

public function getSelectedJadwalTeori($id) {
    $this->db->select('jadwal_teor_i.*, kurikulum.*, prodi.*,
jadwal_teor_i.prodi_id as jadwal_prodi_id, pegawai.*');
    $this->db->join('prodi', 'jadwal_teor_i.prodi_id = prodi.prodi_id');
    $this->db->join('kurikulum', 'jadwal_teor_i.kurikulum_id =
kurikulum.kurikulum_id');
    $this->db->join('pegawai', 'jadwal_teor_i.pegawai_id =
pegawai.pegawai_id');
    $query = $this->db->where('jadwal_teor_i.jadwal_teor_i_id', $id)-
>get('jadwal_teor_i')->row();
    return $query;
}
}

public function crudSelectedJadwalTeori($data)
{
    if(!empty($data))
    {
        $array = [
            'prodi_id' => $data['prodi_id'],

```

```

        'kurikulum_id' => $data['kurikulum_id'],
        'sesi' => $data['sesi'],
        'kapasitas' => $data['kapasitas'],
        'pegawai_id' => $data['pegawai_id']
    ];
    $this->db->where('pegawai_id', $this->uri->segment(5));
    $this->db->where('kurikulum_id', $this->uri->segment(6));
    $this->db->where('sesi', $this->uri->segment(7));
    return $this->db->update('jadwal_teoris', $array);
}
}
public function crud_old($data = NULL) {
    // xlog($data); exit;
    if(!is_array($data)) {
        $query = $this->db->get_where('jadwal_teoris', array('jadwal_teoris_id' =>
$data))->row();
        return $query;
    }
    $jadwal_teoris_id = trim($data['jadwal_teoris_id']);
    $jadwal1 = DateTime::createFromFormat('d/m/Y', $data['jadwal_1']);
    $date1 = $jadwal1->format('Y-m-d');
    for($i=0; $i<$data['pertemuan_1']; $i++)
    {
        $data1[] = array(
            'prodi_id' => trim($data['prodi_id']),
            'periode' => trim($data['periode']),
            'tahun' => trim($data['tahun']),
            'kurikulum_id' => trim($data['kurikulum_id']),
            'pegawai_id' => trim($data['pegawai_id_1']),

```

```

'sesi' => trim($data['sesi']),
'jadwal' => $jadwal1->format('Y-m-d'),
'basis' => trim($data['basis']),
'ruang' => trim($data['ruang_1']),
'jenis' => trim($data['jenis']),
'jam_mulai' => trim($data['jam_mulai_1']),
'jam_selesai' => trim($data['jam_selesai_1']),
'kapasitas' => trim($data['kapasitas'])
);
if($i == 0) {
    $data1[$i]['jadwal'] = $date1;
} else {
    $date1 = date( "Y-m-d", strtotime( "{ $date1 } +7 day" ));
    $data1[$i]['jadwal'] = $date1;
}
}
$this->db->insert_batch('jadwal_teoris', $data1);
}
public function crud($data = NULL) {
    // xlog($data); exit;
    if(!is_array($data)) {
        $query = $this->db->get_where('jadwal_teoris', array('jadwal_teoris_id' =>
$data))->row();
        return $query;
    }
    $jadwal_teoris_id = trim($data['jadwal_teoris_id']);

    $jadwal1 = DateTime::createFromFormat('d/m/Y', $data['jadwal_1']);
    $date1 = $jadwal1->format('Y-m-d');

```

```

for($i=0; $i<$data['pertemuan_1']; $i++)
{
    $data1[] = array(
        'prodi_id' => trim($data['prodi_id']),
        'periode' => trim($data['periode']),
        'tahun' => trim($data['tahun']),
        'kurikulum_id' => trim($data['kurikulum_id']),
        'pegawai_id' => trim($data['pegawai_id_1']),
        'sesi' => trim($data['sesi']),
        'pertemuan' => $i+1,
        'jadwal' => $jadwal1->format('Y-m-d'),
        'basis' => trim($data['basis']),
        'ruang' => trim($data['ruang_1']),
        'jenis' => trim($data['jenis']),
        'jam_mulai' => trim($data['jam_mulai_1']),
        'jam_selesai' => trim($data['jam_selesai_1']),
        'kapasitas' => trim($data['kapasitas'])
    );
    if($i == 0) {
        $data1[$i]['jadwal'] = $date1;
    } else {
        $date1 = date( "Y-m-d", strtotime( "{ $date1 } +7 day" ) );
        $data1[$i]['jadwal'] = $date1;
    }
}
$this->db->insert_batch('jadwal_teoris', $data1);
}

public function getkurikulums() {
$this->db->join('semester', 'semester.semester_id = kurikulum.semester_id');

```

```
$query = $this->db->get('kurikulum')->result();
return $query;
}
public function getPegawais() {
    $this->db->select('role.role_id');
    $role_id = $this->db->where('role_nama','pengajar')->get('role')->result();
    $this->db->select('pegawai.*');
    $this->db->where('role_id', $role_id[0]->role_id);
    $query = $this->db->get('pegawai')->result();
    return $query;
}
public function getProdi() {
$query = $this->db->get('prodi')->result();
return $query;
}
public function editOne_old($data) {
    $date = Datetime::createFromFormat('d/m/Y', $data['jadwal']);
    $jadwal_teorid = $data['jadwal_teorid'];
    $array = array(
        'jadwal' => $date->format('Y-m-d'),
        'jam_mulai' => $data['jam_mulai'],
        'jam_selesai' => $data['jam_selesai'],
    );
    $this->db->where('jadwal_teorid', $jadwal_teorid);
    $this->db->update('jadwal_teorid', $array);
}
public function editOne($data)
{
    if(!empty($data))
```

```

    {
        $date = Datetime::createFromFormat('d/m/Y', $data['jadwal']);
        $jadwal_teori_id = $data['jadwal_teori_id'];
        $array = [
            'jadwal' => $date->format('Y-m-d'),
            'jam_mulai' => $data['jam_mulai'],
            'jam_selesai' => $data['jam_selesai'],
            'ruang' => $data['ruang'],
        ];
        $this->db->where('jadwal_teori_id', $jadwal_teori_id);
        return $this->db->update('jadwal_teori', $array);
    }
}

public function deleteOne($id) {
    $this->db->where('jadwal_teori_id', $id);
    $this->db->delete('jadwal_teori');
}

public function delete($prodi, $kurikulum_id, $pegawai_id, $sesi) {
    $this->db->where('prodi_id', $prodi);
    $this->db->where('pegawai_id', $pegawai_id);
    $this->db->where('kurikulum_id', $kurikulum_id);
    $this->db->where('sesi', $sesi);
    $this->db->delete('jadwal_teori');
}

public function getCariJadwal($data)
{
    $this->db->select('jadwal_teori.*, kurikulum.*, prodi.*, pegawai.*');
    $this->db->join('prodi', 'jadwal_teori.prodi_id = prodi.prodi_id');
    $this->db->join('kurikulum', 'jadwal_teori.kurikulum_id =

```

```

    kurikulum.kurikulum_id');
    $this->db->join('pegawai', 'jadwal_teor_i.pegawai_id = pegawai.pegawai_id');
    $this->db->where('jadwal_teor_i.prodi_id', $data['prodi_id']);
    $this->db->where('jadwal_teor_i.periode', $data['periode']);
    $this->db->where('jadwal_teor_i.basis', $data['basis']);
    $this->db->where('jadwal_teor_i.tahun', $data['tahun']);
    if($this->session->userdata('role_id')==5){
    $this->db->where('jadwal_teor_i.pegawai_id', $this->session->userdata('id'));
    }
    $this->db->group_by('jadwal_teor_i.sesi');
    $query = $this->db->get('jadwal_teor_i')->result();
    return $query;
    }
    public function changeMatkul($id) {
        $query = $this->db->get_where('kurikulum', array('prodi_id' => $id))-
        >result();
        return $query;
    }
    function __destruct() {
        $this->db->close();
    }
}

```

### **Report\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Report_model extends CI_Model
{
    function __construct() {

```

```

parent::__construct();
$this->load->database();
}
public function getProdi() {
$query = $this->db->get('prodi')->result();
return $query;
}
public function getPengajar($no) {
$query = $this->db->query("select
pegawai.*,honor.honor_pembayaran,honor.honor_basis as
basis,jadwal_teor_i.periode,
        jadwal_teor_i.tahun,honor.honor_no, prodi.prodi_nama
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_teor_i on jadwal_teor_i.jadwal_teor_i_id =
detail_honor.jadwal_id
        join pegawai on pegawai.pegawai_id = jadwal_teor_i.pegawai_id
        join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
        where honor.honor_no='$no'
        and detail_honor.honor_jadwal='jadwal_teor_i' and honor.status='aktif'
and detail_honor.status='aktif'

        union
        select pegawai.*,honor.honor_pembayaran,honor.honor_basis as
basis,jadwal_teor_i.periode, jadwal_teor_i.tahun,honor.honor_no,
prodi.prodi_nama  from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_praktikum on jadwal_praktikum.jadwal_praktikum_id =
detail_honor.jadwal_id

```

```

        join jadwal_teor_i on jadwal_teor_i.kurikulum_id =
jadwal_praktikum.jadwal_teor_i_id
        join pegawai on pegawai.pegawai_id = jadwal_praktikum.pegawai_id
        join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
        where honor.honor_no = '$no'
        and detail_honor.honor_jadwal='jadwal_praktikum' and
honor.status='aktif' and detail_honor.status='aktif'
        group by jadwal_praktikum.jadwal_praktikum_id");
        return $query;
    }

    public function getLabDetail($no,$id) {
        $query = $this->db->query("select
detail_honor.jadwal_id,jadwal_teor_i.prodi_id,
kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,
kurikulum.kurikulum_teor_i,kurikulum.kurikulum_praktikum,
        jadwal_teor_i.pegawai_id, jadwal_teor_i.sesi, jadwal_teor_i.jadwal,
jadwal_teor_i.periode, '1' as kelompok, pegawai.rate*kurikulum_teor_i as honor,
jadwal_teor_i.tahun, pegawai.nama_pegawai, pegawai.rate, pegawai.pajak,
prodi.prodi_nama, 'Teori' as aktifitas, jadwal_teor_i.pertemuan
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
join jadwal_teor_i on jadwal_teor_i.jadwal_teor_i_id = detail_honor.jadwal_id
        join pegawai on pegawai.pegawai_id = jadwal_teor_i.pegawai_id
        join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
        join kurikulum on kurikulum.kurikulum_id =
jadwal_teor_i.kurikulum_id
        where honor.honor_no='$no' and pegawai.pegawai_id = '$id'
        and detail_honor.honor_jadwal= 'jadwal_teor_i' and honor.status='aktif'
and detail_honor.status='aktif'

```

```

union
select detail_honor.jadwal_id,jadwal_teoriprodi_id,
kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,
kurikulum.kurikulum_teorikurikulum.kurikulum_praktikum,
jadwal_praktikum.pegawai_id, jadwal_teorisesi,
jadwal_praktikum.jadwal, jadwal_teoriperiode, jadwal_praktikum.kelompok,
pegawai.rate*kurikulum_praktikum as honor,
jadwal_teoritahun, pegawai.nama_pegawai, pegawai.rate,
pegawai.pajak, prodi.prodi_nama,'Praktikum' as aktifitas,
jadwal_praktikum.pertemuan
from honor
join detail_honor on detail_honor.honor_id = honor.honor_id
join jadwal_praktikum on jadwal_praktikum.jadwal_praktikum_id =
detail_honor.jadwal_id
join jadwal_teorion jadwal_teorikurikulum_id =
jadwal_praktikum.jadwal_teorid
join pegawai on pegawai.pegawai_id = jadwal_praktikum.pegawai_id
join prodi on jadwal_teoriprodi_id = prodi.prodi_id
join kurikulum on kurikulum.kurikulum_id =
jadwal_teorikurikulum_id
where honor.honor_no='$no' and pegawai.pegawai_id ='$id'
and detail_honor.honor_jadwal='jadwal_praktikum' and
honor.status='aktif' and detail_honor.status='aktif'
group by jadwal_praktikum.jadwal_praktikum_id)->result();
return $query;
}
public function getLabRekap($no) {
$query = $this->db->query("select
pegawai_id,nama_pegawai,sum(honor) as honor,

```

```

pajak,kurikulum_teor_i,kurikulum_praktikum from (select
detail_honor.jadwal_id,jadwal_teor_i.prodi_id, kurikulum.kurikulum_nama,
kurikulum.kurikulum_sks, kurikulum.kurikulum_teor_i,
kurikulum.kurikulum_praktikum, jadwal_teor_i.pegawai_id, jadwal_teor_i.sesi,
jadwal_teor_i.jadwal, jadwal_teor_i.periode, pegawai.rate*kurikulum_teor_i as
honor, jadwal_teor_i.tahun, pegawai.nama_pegawai, pegawai.rate, pegawai.pajak,
prodi.prodi_nama, 'Teori' as aktifitas
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_teor_i on jadwal_teor_i.jadwal_teor_i_id = detail_honor.jadwal_id
        join pegawai on pegawai.pegawai_id = jadwal_teor_i.pegawai_id
        join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
        join kurikulum on kurikulum.kurikulum_id = jadwal_teor_i.kurikulum_id
        where honor.honor_no='$no' and detail_honor.honor_jadwal=
'jadwal_teor_i' and honor.status='aktif' and detail_honor.status='aktif'
        union
        select detail_honor.jadwal_id,jadwal_teor_i.prodi_id,
kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,kurikulum.kurikulum_teor_i,
kurikulum.kurikulum_praktikum,
        jadwal_praktikum.pegawai_id, jadwal_teor_i.sesi,
jadwal_praktikum.jadwal,
jadwal_teor_i.periode,pegawai.rate*kurikulum_praktikum as honor,
        jadwal_teor_i.tahun, pegawai.nama_pegawai, pegawai.rate,
pegawai.pajak, prodi.prodi_nama, 'Praktikum' as aktifitas
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_praktikum on jadwal_praktikum.jadwal_praktikum_id =
detail_honor.jadwal_id
        join jadwal_teor_i on jadwal_teor_i.kurikulum_id =

```

```

jadwal_praktikum.jadwal_teorid
    join pegawai on pegawai.pegawaiid =
jadwal_praktikum.pegawaiid
    join prodi on jadwal_teorid.prodid = prodi.prodid
    join kurikulum on kurikulum.kurikulumid =
jadwal_teorid.kurikulumid
    where honor.honor_no='$no'
    and detail_honor.honor_jadwal='jadwal_praktikum' and
honor.status='aktif' and detail_honor.status='aktif'
    group by jadwal_praktikum.jadwal_praktikumid a group by
pegawaiid")->result();
        return $query;
    }
    public function getNoHonor() {
        $query = $this->db->where('status','Aktif')->get('honor')->result();
        return $query;
    }
    public function getRekapTeori($data) {
        $this->db->select('honor.honorid,
            honor.honor_no,
            honor.honor_pembayaran,
            honor.prodi_kode,
            honor.honor_basis as basis,
            detail_honor.jadwalid,
            jadwal_teorid.prodid,
            jadwal_teorid.kurikulumid,
            jadwal_teorid.pegawaiid,
            jadwal_teorid.sesi,
            jadwal_teorid.jadwal,

```

```

        jadwal_teor_i.periode,
        jadwal_teor_i.tahun,
        pegawai.nama_pegawai,
        pegawai.rate,
        pegawai.pajak,
        prodi.prodi_nama');
    $this->db->join('detail_honor', 'detail_honor.honor_id =
honor.honor_id');
    $this->db->join('jadwal_teor_i', 'jadwal_teor_i.jadwal_teor_i_id =
detail_honor.jadwal_id');
    $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_teor_i.pegawai_id');
    $this->db->join('prodi', 'jadwal_teor_i.prodi_id = prodi.prodi_id');
    $this->db->where('honor.honor_no', $data);
    $this->db->where('detail_honor.honor_jadwal', 'jadwal_teor_i');
    $query = $this->db->get('honor')->result();
    return $query;
}
public function getRekapPraktikum($data) {
    $this->db->select('honor.honor_id,
        honor.honor_no,
        honor.honor_pembayaran,
        honor.prodi_kode,
        honor.honor_basis as basis,
        detail_honor.honor_jadwal,
        detail_honor.jadwal_id,
        jadwal_praktikum.jadwal_teor_i_id,
        jadwal_praktikum.pegawai_id,
        jadwal_praktikum.jadwal,

```

```

        jadwal_teoriprodi_id,
        jadwal_teorikurikulum_id,
        jadwal_teorisesi,
        jadwal_teoriperiode,
        jadwal_teoritahun,
        pegawai.nama_pegawai,
        pegawai.rate,
        pegawai.pajak,
        prodi.prodi_nama');
    $this->db->join('detail_honor', 'detail_honor.honor_id =
honor.honor_id');
    $this->db->join('jadwal_praktikum',
'jadwal_praktikum.jadwal_praktikum_id = detail_honor.jadwal_id');
    $this->db->join('jadwal_teoriprodi', 'jadwal_teoriprodi.jadwal_teoriprodi_id =
jadwal_praktikum.jadwal_teoriprodi_id');
    $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_praktikum.pegawai_id');
    $this->db->join('prodi', 'jadwal_teoriprodi.prodi_id = prodi.prodi_id');
    $this->db->where('honor.honor_no', $data);
    $this->db->where('detail_honor.honor_jadwal', 'jadwal_praktikum');
    $query = $this->db->get('honor')->result();
    return $query;
}
public function getDetailTeori($data) {
    $this->db->select('honor.honor_id,
        honor.honor_no,
        honor.honor_pembayaran,
        honor.prodi_kode,
        honor.honor_basis as basis,

```

```

        detail_honor.jadwal_id,
        jadwal_teor_i.prodi_id,
        jadwal_teor_i.kurikulum_id,
        jadwal_teor_i.pegawai_id,
        jadwal_teor_i.sesi,
        jadwal_teor_i.jadwal,
        jadwal_teor_i.periode,
        jadwal_teor_i.tahun,
        pegawai.nama_pegawai,
        kurikulum.kurikulum_nama,
        kurikulum.kurikulum_sks,
        pegawai.rate,
        pegawai.no_rekening,
        prodi.prodi_nama');

    $this->db->join('detail_honor', 'detail_honor.honor_id =
honor.honor_id');

    $this->db->join('jadwal_teor_i', 'jadwal_teor_i.jadwal_teor_i_id =
detail_honor.jadwal_id');

    $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_teor_i.pegawai_id');

    $this->db->join('prodi', 'jadwal_teor_i.prodi_id = prodi.prodi_id');

    $this->db->join('kurikulum', 'jadwal_teor_i.kurikulum_id =
kurikulum.kurikulum_id');

    $this->db->where('honor.honor_no', $data);
    $this->db->where('detail_honor.honor_jadwal', 'jadwal_teor_i');

    $query = $this->db->get('honor')->result();
    return $query;
}

public function getDetailPraktikum($data) {

```

```
$this->db->select('honor.honor_id,  
                honor.honor_no,  
                honor.honor_pembayaran,  
                honor.prodi_kode,  
                honor.honor_basis as basis,  
                detail_honor.honor_jadwal,  
                detail_honor.jadwal_id,  
                jadwal_praktikum.jadwal_teorid_id,  
                jadwal_praktikum.pegawai_id,  
                jadwal_praktikum.jadwal,  
                jadwal_teorid.prodid_id,  
                jadwal_teorid.kurikulum_id,  
                jadwal_teorid.periode,  
                jadwal_teorid.sesi,  
                jadwal_teorid.tahun,  
                kurikulum.kurikulum_nama,  
                kurikulum.kurikulum_sks,  
                pegawai.nama_pegawai,  
                pegawai.rate,  
                pegawai.no_rekening,  
                prodi.prodi_nama');  
  
$this->db->join('detail_honor', 'detail_honor.honor_id =  
honor.honor_id');  
  
$this->db->join('jadwal_praktikum',  
'jadwal_praktikum.jadwal_praktikum_id = detail_honor.jadwal_id');  
  
$this->db->join('jadwal_teorid', 'jadwal_teorid.jadwal_teorid_id =  
jadwal_praktikum.jadwal_teorid_id');  
  
$this->db->join('kurikulum', 'jadwal_teorid.kurikulum_id =  
kurikulum.kurikulum_id');
```

```

        $this->db->join('pegawai', 'pegawai.pegawai_id =
jadwal_praktikum.pegawai_id');

        $this->db->where('honor.honor_no', $data);

        $this->db->where('detail_honor.honor_jadwal', 'jadwal_praktikum');

        $this->db->join('prodi', 'jadwal_teoriprodi_id = prodi.prodi_id');

        $query = $this->db->get('honor')->result();

        return $query;
    }

    public function __destruct() {
        $this->db->close();
    }
}

```

### **Semester\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Semester_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }

    public function getSemesters() {
        $query = $this->db->get('semester')->result();
        return $query;
    }

    public function crud($data = NULL) {
        if(!is_array($data)) {
            $query = $this->db->get_where('semester', array('semester_id' =>

```

```

$data))->row();
    return $query;
}
$semester_id = trim($data['semester_id']);
$data['semester_nama'] = url_title(trim($data['semester_nama']));
if(!empty($semester_id)) {
    $this->db->where('semester_id', $semester_id);
    $this->db->update('semester', $data);
} else {
    $data['created_at'] = date('Y-m-d');
    $this->db->insert('semester', $data);
}
}
public function delete($id) {
    $this->db->where('semester_id', $id);
    $this->db->delete('semester');
}
function __destruct() {
    $this->db->close();
}
}

```

### **Slipgaji\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Slipgaji_model extends CI_Model
{
    function __construct() {
        parent::__construct();
    }
}

```

```

$this->load->database();
}
public function getProdi() {
    $query = $this->db->get('prodi')->result();
    return $query;
}

public function getPengajar($no,$bln,$thn) {
    $query = $this->db->query("select
pegawai.*,jadwal_teor_i.periode,jadwal_teor_i.tahun,
honor.honor_pembayaran,honor.tahun_pembayaran, honor.tahun_pengajuan,
honor.honor_pengajuan from pegawai
join jadwal_teor_i on pegawai.pegawai_id = jadwal_teor_i.pegawai_id
join detail_honor on jadwal_teor_i.jadwal_teor_i_id = detail_honor.jadwal_id
join honor on detail_honor.honor_id = honor.honor_id
join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
where pegawai.pegawai_id ='$no' and honor.honor_pengajuan = '$bln' and
honor.tahun_pengajuan = '$thn' and detail_honor.honor_jadwal= 'jadwal_teor_i'
and honor.status='aktif' and detail_honor.status='aktif'

union

select pegawai.*,jadwal_teor_i.periode,jadwal_teor_i.tahun,
honor.honor_pembayaran, honor.tahun_pembayaran, honor.tahun_pengajuan,
honor.honor_pengajuan from pegawai
join jadwal_praktikum on pegawai.pegawai_id =
jadwal_praktikum.pegawai_id
join jadwal_teor_i on jadwal_teor_i.kurikulum_id =
jadwal_praktikum.jadwal_teor_i_id
join detail_honor on jadwal_praktikum.jadwal_praktikum_id =
detail_honor.jadwal_id
join honor on detail_honor.honor_id = honor.honor_id

```

```

        join prodi on jadwal_teoriprodi_id = prodi.prodi_id
        where pegawai.pegawai_id = '$no' and honor.honor_pengajuan = '$bln'
and honor.tahun_pengajuan = '$thn' and detail_honor.honor_jadwal=
'jadwal_praktikum' and honor.status='aktif' and detail_honor.status='aktif'
        group by jadwal_praktikum.jadwal_praktikum_id");
        return $query;
    }

    public function getLabDetail($id,$bln,$thn) {
        $query = $this->db->query("select
detail_honor.jadwal_id,jadwal_teoriprodi_id,
kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,
pegawai.rate*kurikulum_teoriprodi_id as honor, jadwal_teoriprodi_id,
jadwal_teoriprodi_id, jadwal_teoriprodi_id, jadwal_teoriprodi_id, honor.honor_no, '1'
as kelompok, kurikulum.kurikulum_teoriprodi_id, kurikulum.kurikulum_praktikum,
jadwal_teoriprodi_id, pegawai.nama_pegawai, pegawai.rate, pegawai.pajak,
prodi.prodi_nama, 'Teori' as aktifitas, jadwal_teoriprodi_id.pertemuan
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_teoriprodi_id on jadwal_teoriprodi_id.jadwal_teoriprodi_id =
detail_honor.jadwal_id
        join pegawai on pegawai.pegawai_id = jadwal_teoriprodi_id.pegawai_id
        join prodi on jadwal_teoriprodi_id.prodi_id = prodi.prodi_id
        join kurikulum on kurikulum.kurikulum_id =
jadwal_teoriprodi_id.kurikulum_id
        where pegawai.pegawai_id = '$id' and honor.honor_pengajuan = '$bln'
and honor.tahun_pengajuan = '$thn' and detail_honor.honor_jadwal=
'jadwal_teoriprodi_id' and honor.status='aktif' and detail_honor.status='aktif'
        union
        select detail_honor.jadwal_id,jadwal_teoriprodi_id,

```

```

kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,
pegawai.rate*kurikulum_praktikum as honor, jadwal_praktikum.pegawai_id,
jadwal_teoris.sesi, jadwal_praktikum.jadwal,
jadwal_teoris.periode,honor.honor_no,jadwal_praktikum.kelompok ,
kurikulum.kurikulum_teoris,kurikulum.kurikulum_praktikum,
        jadwal_teoris.tahun, pegawai.nama_pegawai, pegawai.rate,
pegawai.pajak, prodi.prodi_nama,'Praktikum' as aktifitas,
jadwal_praktikum.pertemuan
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_praktikum on jadwal_praktikum.jadwal_praktikum_id =
detail_honor.jadwal_id
        join jadwal_teoris on jadwal_teoris.kurikulum_id =
jadwal_praktikum.jadwal_teoris_id
        join pegawai on pegawai.pegawai_id = jadwal_praktikum.pegawai_id
        join prodi on jadwal_teoris.prodi_id = prodi.prodi_id
        join kurikulum on kurikulum.kurikulum_id =
jadwal_teoris.kurikulum_id
        where pegawai.pegawai_id ='$id' and honor.honor_pengajuan = '$bln'
and honor.tahun_pengajuan = '$thn' and
detail_honor.honor_jadwal='jadwal_praktikum' and honor.status='aktif' and
detail_honor.status='aktif'
        group by jadwal_praktikum.jadwal_praktikum_id")->result();
return $query;
}

public function getLabRekap($no,$bln,$thn) {
    $query = $this->db->query("select pegawai_id,
nama_pegawai,sum(honor) as honor, pajak, prodi_nama, basis, honor_no
        from (select detail_honor.jadwal_id,jadwal_teoris.prodi_id,

```

```

kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,
        jadwal_teor_i.pegawai_id, jadwal_teor_i.sesi, jadwal_teor_i.jadwal,
jadwal_teor_i.periode,honor.honor_basis as basis, pegawai.rate*kurikulum_teor_i
as honor,
        jadwal_teor_i.tahun, pegawai.nama_pegawai, pegawai.rate,
pegawai.pajak, prodi.prodi_nama, 'Teori' as aktifitas,honor.honor_no
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_teor_i on jadwal_teor_i.jadwal_teor_i_id =
detail_honor.jadwal_id
        join pegawai on pegawai.pegawai_id = jadwal_teor_i.pegawai_id
        join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
        join kurikulum on kurikulum.kurikulum_id =
jadwal_teor_i.kurikulum_id
        where pegawai.pegawai_id='$no' and honor.honor_pengajuan =
'$bln' and honor.tahun_pengajuan = '$thn'
        and detail_honor.honor_jadwal= 'jadwal_teor_i' and
honor.status='aktif' and detail_honor.status='aktif'
        union
        select detail_honor.jadwal_id,jadwal_teor_i.prodi_id,
kurikulum.kurikulum_nama,kurikulum.kurikulum_sks,
        jadwal_praktikum.pegawai_id, jadwal_teor_i.sesi,
jadwal_praktikum.jadwal, jadwal_teor_i.periode,honor.honor_basis as basis,
pegawai.rate*kurikulum_praktikum as honor,
        jadwal_teor_i.tahun, pegawai.nama_pegawai, pegawai.rate,
pegawai.pajak, prodi.prodi_nama,'Praktikum' as aktifitas,honor.honor_no
        from honor
        join detail_honor on detail_honor.honor_id = honor.honor_id
        join jadwal_praktikum on jadwal_praktikum.jadwal_praktikum_id =

```

```

detail_honor.jadwal_id
        join jadwal_teor_i on jadwal_teor_i.kurikulum_id =
jadwal_praktikum.jadwal_teor_i_id
        join pegawai on pegawai.pegawai_id =
jadwal_praktikum.pegawai_id
        join prodi on jadwal_teor_i.prodi_id = prodi.prodi_id
        join kurikulum on kurikulum.kurikulum_id =
jadwal_teor_i.kurikulum_id
        where pegawai.pegawai_id='$no' and honor.honor_pengajuan =
'$bIn' and honor.tahun_pengajuan = '$thn'
        and detail_honor.honor_jadwal='jadwal_praktikum' and
honor.status='aktif' and detail_honor.status='aktif'
        group by jadwal_praktikum.jadwal_praktikum_id) a group by
honor_no, pegawai_id")->result();
        return $query;
    }
    function getAnggota($nama)
    {
        $query = $this->db->query("select
pegawai_id,nama_pegawai

        from pegawai where pegawai_id like '%$nama%' or nama_pegawai
like '%$nama%'");
        return $query->result();
    }
    public function __destruct() {
        $this->db->close();
    }
}

```

**User\_model.php**

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class User_model extends CI_Model
{
    private $_mongoDb;
    function __construct() {
        parent::__construct();
        $this->load->database();
    }
    public function getPegawais() {
        $this->db->join('role', 'role.role_id = pegawai.role_id', 'left');
        $this->db->join('prodi', 'prodi.prodi_id = pegawai.prodi_id', 'left');
        $query = $this->db->get('pegawai')->result();
        return $query;
    }
    public function detail($id) {
        $this->db->join('role', 'role.role_id = pegawai.role_id', 'left');
        $this->db->join('prodi', 'prodi.prodi_id = pegawai.prodi_id', 'left');
        $this->db->where('pegawai_id', $id);
        $query = $this->db->get('pegawai')->row();
        return $query;
    }
    public function crud($data) {
        $data['pegawai_password'] = $data['pegawai_password'];
        $this->db->where('pegawai_id', $this->session->userdata('id'));
        $this->db->update('pegawai', $data);
    }
}
```

```

public function edit($data) {
    $this->db->where('pegawai_id', $this->uri->segment(4));
    $this->db->update('pegawai', $data);
}
function __destruct() {
    $this->db->close();
}
}

```

### **Login\_model.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Login_model extends CI_Model
{
    function __construct() {
        parent::__construct();
        $this->load->database();
    }
    public function login($data) {
        $this->db->join('role', 'role.role_id = pegawai.role_id');
        $query = $this->db->get_where('pegawai', array('pegawai_id' =>
trim($data['pegawai_id']))->row();
        return $query;
    }
    function validate($id, $password)
    {
        $this->db->join('role', 'role.role_id = pegawai.role_id');
        $this->db->where('pegawai_id', $id);
        $this->db->where('pegawai_password', $password);
    }
}

```

```
        $query = $this->db->get('pegawai')->row();
        return $query;
    }
    public function __destruct() {
        $this->db->close();
    }
}
```