

UTS

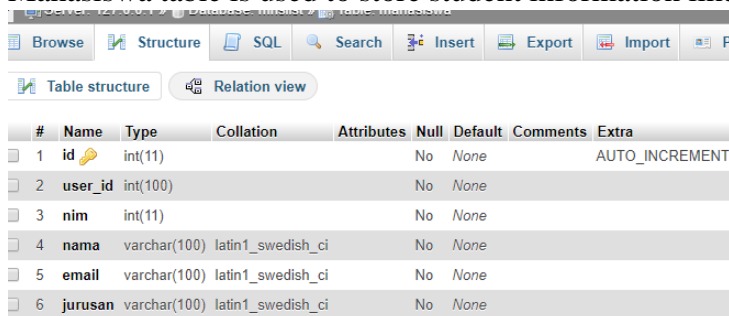
Advanced Web Programming

I. Database

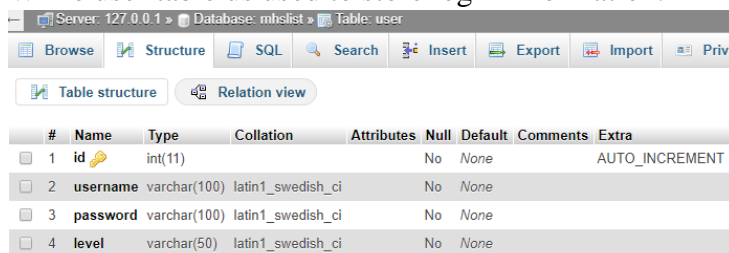
1. To set up the database you should turn on your xampp both apache and mysql services.
2. In this case you should make new database named mhslist, we provide the sql file named mhslist.sql.
3. On phpMyAdmin we have create mhslist database that contain mahasiswa table and user table.



4. Mahasiswa table is used to store student information like nim, nama, email, and jurusan.



5. While user table is used to store login information.



In the password field we use sha1 encryption that make it more secure.

II. Server Rest-API

1. We have 2 models, mahasiswa_model and user_model on rest-api/application/models directory. This php class contains CRUD methods that connected to the database.

Mahasiswa_model.php

```
class Mahasiswa_model extends CI_Model {
    public function getMahasiswa($id=null){
        if($id===null){
            return $this->db->get('mahasiswa')->result_array();
        }else{
```

```

        return $this->db->get_where('mahasiswa', ['user_id'=>$id])->result_array();
    }
}

public function deleteMahasiswa($id){
    $this->db->delete('mahasiswa', ['user_id'=>$id]);
    return $this->db->affected_rows();
}

public function createMahasiswa($data){
    $this->db->insert('mahasiswa', $data);
    return $this->db->affected_rows();
}

public function updateMahasiswa($data,$id){
    $this->db->update('mahasiswa', $data, ['user_id'=>$id]);
    return
    $this->db->affected_rows();
}
}

```

User_model.php

```

class User_model extends CI_Model{
    public function getUser($username=null){
        if($username===null){
            return $this->db->get('user')->result_array();
        }else{
            return $this->db->get_where('user', ['username'=>$username])->result_array();
        }
    }

    public function deleteUser($id){
        $this->db->delete('user', ['id'=>$id]);
        return $this->db->affected_rows();
    }

    public function createUser($data){
        $this->db->insert('user', $data);
        return $this->db->affected_rows();
    }

    public function updateUser($data,$username){
        $this->db->update('user', $data, ['username'=>$username]);
        return $this->db->affected_rows();
    }
}

```

2. We also make 2 api that contain GET, POST, DELETE, and PUT request method on application/api directory.

Mahasiswa API

```
class Mahasiswa extends REST_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');

        $this->load->model('Mahasiswa_model','mahasiswa');
        $this->load->model('User_model','user');
    }

    public function index_get()
    {
        $id=$this->get('id');
        if($id==null){
            $mahasiswa=$this->mahasiswa->getMahasiswa();
        } else{
            $mahasiswa=$this->mahasiswa->getMahasiswa($id);
        }
        if($mahasiswa){
            $this->response([
                'status'=>true,
                'data'=>$mahasiswa
            ], REST_Controller::HTTP_OK);
        }else{
            $this->response([
                'status'=>false,
                'data'=>'id not found'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }

    public function index_delete(){
        $id=$this->delete('id');
        if($id==null){
            $this->response([
                'status'=>false,
                'message'=>'provide an id!'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }else{
            if($this->mahasiswa->deleteMahasiswa($id)>0){
                $this->response([
```

```

        'status'=>true,
        'id'=>$id,
        'message'=>'deleted'
    ], REST_Controller::HTTP_OK);
}else{
    $this->response([
        'status'=>false,
        'message'=>'id not found!'
    ], REST_Controller::HTTP_BAD_REQUEST);
}
}
}

public function index_post(){
    $data=[
        'user_id'=>$this->post('user_id'),
        'nim'=>$this->post('nim'),
        'nama'=>$this->post('nama'),
        'email'=>$this->post('email'),
        'jurusan'=>$this->post('jurusan')
    ];
    if($this->mahasiswa->createMahasiswa($data)>0){
        $this->response([
            'status'=>true,
            'message'=>'new mahasiswa has been created'
        ], REST_Controller::HTTP_CREATED);
    }else{
        $this->response([
            'status'=>false,
            'message'=>'failed to create new data'
        ], REST_Controller::HTTP_BAD_REQUESTD);
    }
}

public function index_put(){
    $id=$this->put('id');
    $data=[
        'nim'=>$this->put('nim'),
        'nama'=>$this->put('nama'),
        'email'=>$this->put('email'),
        'jurusan'=>$this->put('jurusan')
    ];
    if($this->mahasiswa->updateMahasiswa($data,$id)>0){
        $this->response([
            'status'=>true,

```

```

        'message'=>'data mahasiswa has been updated'
    ], REST_Controller::HTTP_OK);
    }else{
        $this->response([
            'status'=>false,
            'message'=>'failed to update data!'
        ],REST_Controller::HTTP_BAD_REQUEST);
    }
}
}
}

```

User API

```

class User extends REST_Controller{
    public function __construct($config = 'rest') {
        parent::__construct($config);
        $this->load->helper('url');
        $this->load->model('User_model','user');
    }
    public function index_get()
    {
        $username=$this->get('username');
        if($username==null){
            $user=$this->user->getUser();
        } else{
            $user=$this->user->getUser($username);
        }
        if($user){
            $this->response([
                'status'=>true,
                'data'=>$user
            ], REST_Controller::HTTP_OK);
        }else{
            $this->response([
                'status'=>false,
                'data'=>'id not found'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }

    public function index_delete(){
        $id=$this->delete('id');
        if($id===null){
            $this->response([
                'status'=>false,
                'message'=>'provide an id!'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
}

```

```

        ], REST_Controller::HTTP_BAD_REQUEST);
    }else{
        if($this->user->deleteUser($id)>0){
            $this->response([
                'status'=>true,
                'id'=>$id,
                'message'=>'deleted'
            ], REST_Controller::HTTP_OK);
        }else{
            $this->response([
                'status'=>false,
                'message'=>'id not found!'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
}

public function index_post(){
    $data=[
        'username'=> $this->post('username'),
        'password'=> sha1($this->post('password')),
        'level'=> $this->post('level')
    ];
    if($this->user->createUser($data)>0){
        $this->response([
            'status'=>true,
            'message'=>'User has been created'
        ], REST_Controller::HTTP_CREATED);
    }else{
        $this->response([
            'status'=>false,
            'message'=>'failed to create new data'
        ], REST_Controller::HTTP_BAD_REQUEST);
    }
}

public function index_put(){
    $username=$this->put('username');
    $data=[
        'password'=> sha1($this->put('password')),
        'level'=> $this->put('level')
    ];
    if($this->user->updateUser($data,$username)>0){
        $this->response([
            'status'=>true,
            'message'=>'data User has been updated'

```

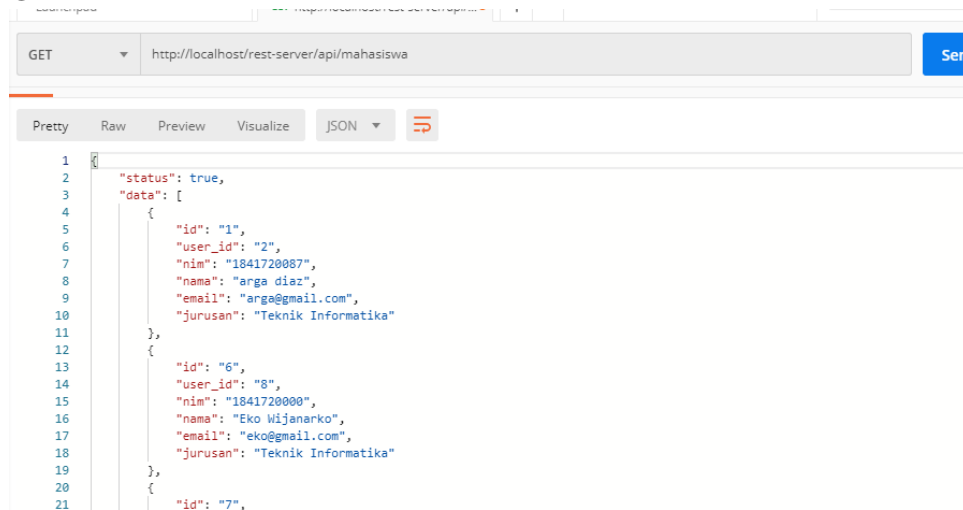
```

    ], REST_Controller::HTTP_OK);
}else{
    $this->response([
        'status'=>false,
        'message'=>'failed to update data!'
    ],REST_Controller::HTTP_BAD_REQUEST);
}
}
}
}

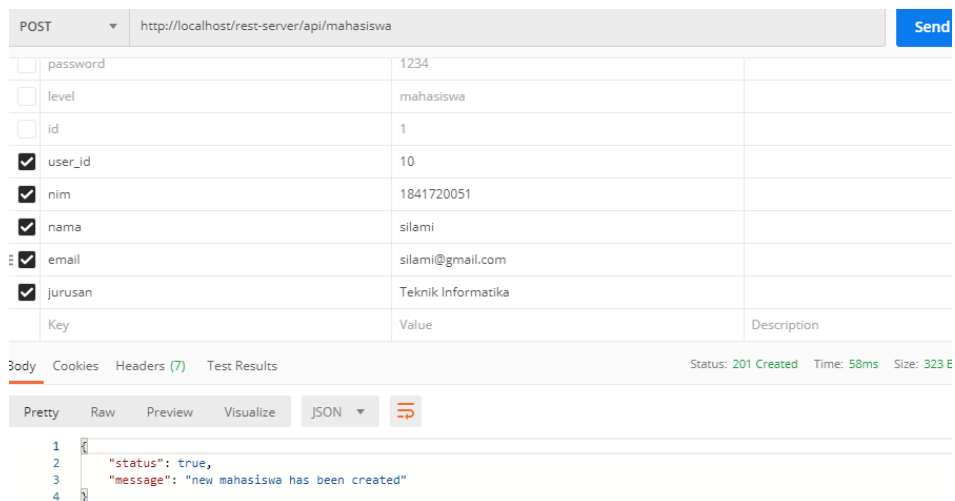
```

3. To test the API, first turn on your xampp both apache and mysql services. We use postman to try to send some request. For example, we use Mahasiswa database. Use 'http://localhost/rest-server/api/mahasiswa/' for the url prefix.

a) GET



b) POST – in post, delete, and put, use 'x-www-form-urlencoded' in body navbar below the url field.



Then Silami is saved with id number 8.

GET http://localhost/rest-server/api/mahasiswa?id=10

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	10	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 65ms Size: 40

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "data": [
4     {
5       "id": "8",
6       "user_id": "10",
7       "nim": "1841720051",
8       "nama": "silami",
9       "email": "silami@gmail.com",
10      "jurusan": "Teknik Informatika"
11     }
12   ]
13 }

```

c) DELETE - We try to delete silami's information

DELETE http://localhost/rest-server/api/mahasiswa

<input type="checkbox"/> jurusan	Teknik Informatika	
<input checked="" type="checkbox"/> id	8	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 58ms Size:

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "id": "8",
4   "message": "deleted"
5 }

```

Then the data no longer in the database.

GET http://localhost/rest-server/api/mahasiswa?id=8

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	8	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 404 Not Found Time: 56ms Size: 305

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": false,
3   "data": "id not found"
4 }

```

d) PUT - We try to change silami's email.

PUT http://localhost/rest-server/api/mahasiswa

<input checked="" type="checkbox"/>	nim	1841720051	
<input checked="" type="checkbox"/>	nama	silami	
<input checked="" type="checkbox"/>	email	silami211@gmail.com	
<input checked="" type="checkbox"/>	jurusan	Teknik Informatika	
	Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 59ms Size: 3

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "message": "data mahasiswa has been updated"
4 }

```

The email record changed.

GET http://localhost/rest-server/api/mahasiswa?id=8

<input checked="" type="checkbox"/>	id	8	
	Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 58ms Size: 406

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "data": [
4     {
5       "id": "6",
6       "user_id": "8",
7       "nim": "1841720051",
8       "nama": "silami",
9       "email": "silami211@gmail.com",
10      "jurusan": "Teknik Informatika"
11     }
12   ]
13 }

```

III. Client Web-Based

1. First, we need to make the model of each table in database on application/models. Mahasiswa_model.php

```

class Mahasiswa_model extends CI_Model {
    public function getMahasiswa($id){
        return $this->db->get_where('mahasiswa', ['user_id'=>$id])->result_array();
    }
    public function getMahasiswaInfo($id){
        return $this->db->get_where('mahasiswa', ['id'=>$id])->result_array();
    }
}

```

Login_model.php

```

class login_model extends CI_Model {

    function login($username,$password){

```

```

        $this->db->select('id,username,password,level');
        $this->db->from('user');
        $this->db->where('username', $username);
        $this->db->where('password', sha1($password));
        $this->db->limit(1);

        $query=$this->db->get();
        if($query->num_rows()==1){
            return $query->result_array();
        }else{
            return false;
        }
    }
}

```

2. Second, we need to make the controller on application/controller.

Login.php

```

class login extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();
        $this->load->model('login_model');
    }

    public function index()
    {
        $this->session->sess_destroy();
        $data['title']='Login';

        $this->load->view('template/header_login',$data);
        $this->load->view('login/index',$data);
        $this->load->view('template/footer');
    }

    public function proses_login(){
        $username=htmlspecialchars($this->input->post('uname1'));
        $password=htmlspecialchars($this->input->post('pwd1'));

        $check= $this->login_model->login($username, $password);
        if($check){
            foreach($check as $chk){
                $this->session->set_userdata('id', $chk["id"]);
                $this->session->set_userdata('username', $chk["username"]);
            }
        }
    }
}

```

```

        $this->session->set_userdata('level',$chk["level"]);

        if($this->session->userdata('level')== "dosen"){
            redirect('mahasiswa/index');
        }
    }else{
        $data['pesan']="username dan password anda salah";
        $data['title']='login';
        $this->load->view('template/header_login',$data);
        $this->load->view('login/index',$data);
        $this->load->view('template/footer');
    }
}

public function logout(){
    $this->session->sess_destroy();

    redirect('login','refresh');
}
}

```

Mahasiswa.php

```

class mahasiswa extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('Mahasiswa_model', 'mahasiswa');
        $this->load->library('form_validation');

        if($this->session->userdata('level')!="dosen"){
            redirect('login','refresh');
        }
    }

    public function index()
    {
        $data['title']=ucfirst($this->session->userdata('username'));
        $key=$this->input->post('keyword');
        $data['mahasiswa']=$this->mahasiswa->getMahasiswa($this->session->userdata('id'));
        $this->load->view('template/header_datatables',$data);
        $this->load->view('mahasiswa/index',$data);
        $this->load->view('template/footer_datatables');
    }
}

```

```

        public function detail($id){
            $data['title']='Dosen';
            $data['mahasiswa']=$this->mahasiswa->getMahasiswaInfo($id);
            $this->load->view('template/header', $data);
            $this->load->view('mahasiswa/detail', $data);
            $this->load->view('template/footer');

        }
    }
}

```

3. Third make the view of login, List mahasiswa, and details.
Login (application/views/login/index.php)

```

<?=
form_open('login/proses_login');
?>

<div class="container">
    <div class="row py-5">
        <div class="col-md-12 min-vh-100 d-flex flex-column justify-content-center">
            <div class="row">
                <div class="col-lg-6 col-md-8 mx-auto">

                    <!-- form card login -->
                    <div class="card rounded shadow shadow-sm">
                        <div class="card-header">
                            <h3 class="mb-0">Login</h3>
                            <div class='alert alert-info mt-3 role='alert'>
                                <?php
                                    if(isset($pesan)){
                                        echo $pesan;
                                    }else{
                                        echo "Masukkan username dan password anda";
                                    }
                                ?>
                            </div>
                        </div>
                        <div class="card-body">
                            <form class="form" role="form" autocomplete="off" id="formLogin" novalidate="" method="POST">

                                <div class="form-group">
                                    <label for="uname1">Username</label>
                                    <input type="text" class="form-control form-control-lg rounded-0" name="uname1" id="uname1" required="">

                                    <div class="invalid-feedback">Oops, you missed this one.</div>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>

        <div class="form-group">
            <label>Password</label>
            <input type="password" class="form-control form-control-lg rounded-
0" id="pwd1" name="pwd1" required="" autocomplete="new-password">
            <div class="invalid-feedback">Enter your password too!</div>
        </div>

        <button type="submit" class="btn btn-success btn-lg float-
right" id="btnLogin">Login</button>
    </form>
</div>

<!--/card-block-->
</div>

<!-- /form card login -->
</div>

</div>

<!--/row-->
</div>

<!--/col-->
</div>

<!--/row-->
</div>
<!--/container-->
<?=
form_close();
?>

```

List Mahasiswa (application/views/mahasiswa/index.php)

```

<div class="container">
    <div class="row mt-3">
        <div class="col-md-6">
            <div class="list-mahasiswa">
                <div class="col-md-12">
                    <h1 style="text-align: center; margin-bottom:30px">List Mahasiswa</h1>
                </div>
                <table class="table table-striped table-bordered" id="list_mhs">
                    <thead>
                        <tr>
                            <th>No</th>
                            <th>Nama</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php
                            $no=1;

```

```

        foreach($mahasiswa as $mhs){?>
            <tr>
                <td><?=$no++?></td>
                <td><?=$mhs['nama']?><a class="badge badge-primary float-
right" href="<?=base_url()?>mahasiswa/detail/<?=$mhs["id"]?>">Detail</a></td>
            </tr>
        <?php }?>
    </tbody>
</table>
</div>
</div>
</div>
</div>

```

4. Then to access the website, turn on your xampp, both of apache and mysql services.
5. Access the web using url below:
'<http://localhost/Project/>' or you can use '<http://localhost/Project/login>'
6. In the client website, we provide login as an authentication. Every time the user access login page, the session will reset so they can't go back to the previous page.

Login

Masukkan username dan password anda

Username

Password

Login

After login the user can see all of their student so, each teacher has its student. For example we login as 'arga' with password 1234.

Login

Masukkan username dan password anda

Username

Password

Login

Note that the password in database is encrypted, so notice the password based on this report. The encryption itself are using sha1 method.

7. It will appear 2 students brian and ilham. Note that the teacher has their own students. The layout is using data tables.

The screenshot shows a web browser at `localhost/Project/mahasiswa/index`. The user is logged in as 'Arga'. The page title is 'List Mahasiswa'. Below the title, there is a 'Show 10 entries' dropdown and a 'Search:' input field. A table lists two students:

No	Nama	
1	brian	Detail
2	ilham	Detail

At the bottom, it says 'Showing 1 to 2 of 2 entries' with 'Previous', '1', and 'Next' navigation links.

8. Then we will try to login with different user. Username 'silami' and password '12345'.

The screenshot shows the same 'List Mahasiswa' page but for user 'Silami'. The table now lists two different students:

No	Nama	
1	eko	Detail
2	prima	Detail

The navigation and search elements are identical to the previous screenshot.

Of course he has students named eko and prima.

9. On page detail is used to show the personal information of each student. For example, we click on eko's detail.

The screenshot shows the 'Detail Data Mahasiswa' page for student 'eko'. The page has a header with 'Dosen', 'Home', and 'Logout' links. The main content area displays the following information:

Eko
Email : eko@gmail.com
Nim : 1841720051
Jurusan : Teknik Informatika

At the bottom, there is a blue button labeled 'Kembali'.