Tugas Praktikum Pertemuan 11

Anisa Intania Putri

2024071010

Code	Output
.env `CI_ENVIRONMENT = development`	

Routes.php
 	
app > Config > Routes.php
`<?php`
 	
`use CodeIgniter\Router\RouteCollection;`
 	
`/**`
`* @var RouteCollection $routes`
`*/`
 	
`$routes->get('/', 'Home::index');`
 	
`$routes->get('/register', 'Register::index');`
`$routes->post('/register/simpan', 'Register::simpan');`	

App.php
 	
app > Config > App.php
`<?php`
 	
`namespace Config;`
 	
`use CodeIgniter\Config\BaseConfig;`
 	
`class App extends BaseConfig`
`{`
`/**`
`* Base Site URL`
`* URL to your CodeIgniter root. Typically, this will be your b`
`* WITH a trailing slash:`
`* E.g., http://example.com/`
`*/`
`public string $baseURL = 'http://localhost/ci44';`
 	
`/**`
`* Allowed Hostnames in the Site URL other than the hostname in`
`* If you want to accept multiple Hostnames, set this.`
`* E.g.,`
`* When your site URL ($baseURL) is 'http://example.com/', and`
`* also accepts 'http://media.example.com/' and 'http://account`
`* ['media.example.com', 'accounts.example.com']`
`* @var list<string>`
`*/`
`public array $allowedHostnames = [];`	

Register.php	

app > Controllers > Register.php

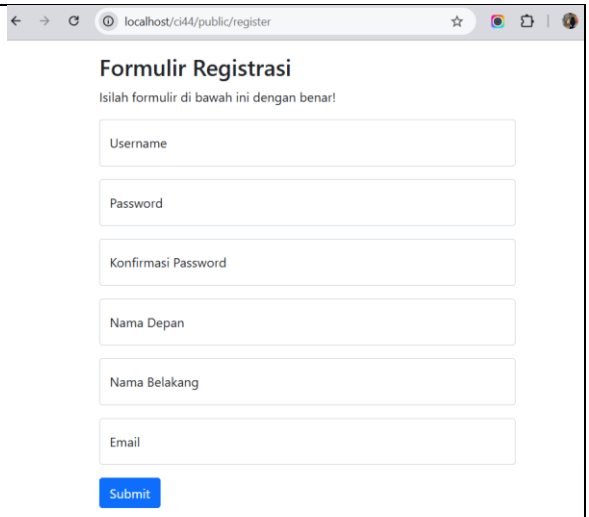
```php
<?php

namespace App\Controllers;

use CodeIgniter\Controller;
use CodeIgniter\HTTP\CLIRequest;
use CodeIgniter\HTTP\IncomingRequest;
use CodeIgniter\HTTP\RequestInterface;
use CodeIgniter\HTTP\ResponseInterface;
use Psr\Log\LoggerInterface;

class Register extends Controller
{
    public function index()
    {
        helper('form');
        echo view('register_view');
    }

    public function simpan()
    {
        helper(['form']);

        // RULES VALIDASI
        $rules = [
            'username' => [
                'rules' => 'required|is_unique[users.username]',
                'errors' => [
                    'required'  => 'Username tidak boleh kosong, silakan ulangi!',
                    'is_unique' => 'Username sudah terpakai, silakan ulangi!',
                ]
            ],

            'password' => [
                'rules' => 'required|min_length[6]|max_length[10]',
                'errors' => [
                    'required'   => 'Password tidak boleh kosong, silakan ulangi!',
```

```php
            'password' => [
                'rules' => 'required|min_length[6]|max_length[10]',
                'errors' => [
                    'required'   => 'Password tidak boleh kosong, silakan ulangi!',
                    'min_length' => 'Jumlah karakter password harus lebih dari 6 karakter, silakan ulangi!
                    'max_length' => 'Jumlah karakter password maksimal 10 karakter, silakan ulangi!',
                ]
            ],

            'password_conf' => [
                'rules' => 'required|matches[password]',
                'errors' => [
                    'required' => 'Konfirmasi password tidak boleh kosong!',
                    'matches'  => 'Password dan konfirmasi tidak sama, silakan ulangi!',
                ]
            ],

            'namadepan' => [
                'rules' => 'required',
                'errors' => [
                    'required' => 'Nama Depan tidak boleh kosong, silakan ulangi!',
                ]
            ],

            'namabelakang' => [
                'rules' => 'required',
                'errors' => [
                    'required' => 'Nama Belakang tidak boleh kosong, silakan ulangi!',
                ]
            ],
```

```php
            'email' => [
                'rules' => 'required|min_length[5]|valid_email',
                'errors' => [
                    'required'    => 'Email tidak boleh kosong, silakan ulangi!',
                    'min_length' => 'Masa sih email cuma 5 karakter? Silakan ulangi!',
                    'valid_email'=> 'Email tidak valid, silakan ulangi!',
                ]
            ],
        ];

        // PROSES VALIDASI
        if ($this->validate($rules)) {
            echo "Data telah tervalidasi!";
        } else {
            $data['validasi'] = $this->validator;
            echo view('register_view', $data);
        }
    }
}
?>
```

register_view.php

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>

<body>
    <div class="container mt-3">
        <h2>Formulir Registrasi</h2>
        <p>Isilah formulir di bawah ini dengan benar!</p>

        <?php if (isset($validasi)): ?>
            <div class="alert alert-danger">
                <?= $validasi->listErrors() ?>
            </div>
        <?php endif; ?>

        <form action="<?= base_url() ?>/register/simpan" method="post">

            <div class="form-floating mb-3 mt-3">
                <input type="text" class="form-control" id="username" placeholder="Masukkan username" name="username" value="<?= set_value('username') ?>">
                <label for="username">Username</label>
            </div>

            <div class="form-floating mt-3 mb-3">
                <input type="password" class="form-control" id="password" placeholder="Masukkan password" name="password">
                <label for="password">Password</label>
            </div>

            <div class="form-floating mt-3 mb-3">
                <input type="password" class="form-control" id="cpassword" placeholder="Masukkan konfirmasi password" name="cpassword">
                <label for="cpassword">Konfirmasi Password</label>
```



Database.php

app > Config > Database.php

```
11  {
20      public string $defaultGroup = 'default';
21
22      /**
23       * The default database connection.
24       *
25       * @var array<string, mixed>
26       */
27      public array $default = [
28          'DSN'          => '',
29          'hostname'     => 'localhost',
30          'username'     => 'root',
31          'password'     => '',
32          'database'     => 'upj',
33          'DBDriver'     => 'MySQLi',
34          'DBPrefix'     => '',
35          'pConnect'     => false,
36          'DBDebug'      => true,
37          'charset'      => 'utf8mb4',
38          'DBCollat'     => 'utf8mb4_general_ci',
39          'swapPre'      => '',
40          'encrypt'      => false,
41          'compress'     => false,
42          'strictOn'     => false,
43          'failover'     => [],
44          'port'         => 8111,
45          'numberNative' => false,
46          'foundRows'    => false,
47          'dateFormat'   => [
48              'date'     => 'Y-m-d',
49              'datetime' => 'Y-m-d H:i:s',
50              'time'     => 'H:i:s',
51          ],
52      ];
```

UserModel.php

app > Models > UserModel.php

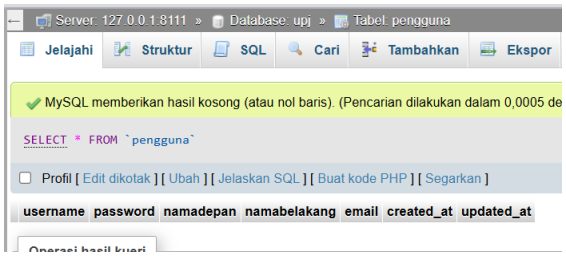
```
1   <?php namespace App\Models;
2   use CodeIgniter\Database\ConnectionInterface;
3   use CodeIgniter\Model;
4
5   class UserModel extends Model
6   {
7       protected $table = 'users';
8
9       protected $allowedFields = ['name', 'email', 'contact_no', 'created_a
10  }
```

PenggunaModel.php

app > Models > PenggunaModel.php

```php
<?php namespace App\Models;
use CodeIgniter\Database\ConnectionInterface;
use CodeIgniter\Model;

class UserModel extends Model
{
    protected $table = 'pengguna';

    protected $allowedFields = ['username', 'password', 'namadepan', 'namabelakang', 'email', 'created_at', 'updated_at'];
}
```

2025-11-19-094828_Users.php

app > Database > Migrations > 2025-11-19-094828_Users.php

```php
<?php

namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;

class Users extends Migration
{
    public function up()
    {
        /*
            - username varchar(20) primary key not null
            - password varchar(200) not null
            - namadepan varchar(20) not null
            - namabelakang varchar(20)
            - email varchar(20)
        */

        $this->forge->addField([
            'username' => [
                'type'       => 'VARCHAR',
                'constraint' => 20,
            ],
            'password' => [
                'type'       => 'VARCHAR',
                'constraint' => 200,
            ],
            'namadepan' => [
                'type'       => 'VARCHAR',
                'constraint' => 20,
            ],
            'namabelakang' => [
                'type'       => 'VARCHAR',
                'constraint' => 20,
                'null'       => true,
            ],
            'email' => [
                'type'       => 'VARCHAR',
                'constraint' => 20,
            ],
            'created_at' => [
                'type' => 'DATETIME',
                'null' => true,
            ],
            'updated_at' => [
                'type' => 'DATETIME',
                'null' => true,
            ],
        ]);

        // Primary key = username
        $this->forge->addKey('username', true);

        // Buat tabel
        $this->forge->createTable('pengguna');
    }

    public function down()
    {
    }
}
```