	REKAYASA PERANGKAT LUNAK		
	SMK MUHAMMADIYAH 1 YOGYAKARTA		
	KELAS : XI	SEMESTER : 2	WAKTU : 6 JP

A. Capaian Pembelajaran

Memahami pengertian, sejarah, komponen dan pengembangan aplikasi perangkat bergerak menggunakan IDE, bahasa pemrograman, basis data dan *framework* serta mampu mendokumentasikan dengan baik.

B. Tujuan Pembelajaran

Peserta didik mampu memahami konsep struktur dan mampu menerapkan metode create pada framework codeigniter

C. Alat Dan Bahan

4. Personal Computer
5. Visual Studio Code
6. Xampp

D. Teori Singkat

CodeIgniter adalah sebuah kerangka aplikasi web PHP yang digunakan untuk mengembangkan situs web dan aplikasi web dinamis. Ini adalah sebuah kerangka sumber terbuka yang menyediakan seperangkat alat yang sederhana dan elegan untuk membangun aplikasi web PHP berfitur lengkap.

CodeIgniter dibuat untuk memudahkan bagi para pengembang untuk membangun situs web dinamis dengan menyediakan seperangkat alat yang sederhana dan teratur untuk mengelola tugas-tugas umum seperti routing, akses database, dan manajemen sesi. Ini mengikuti arsitektur Model-View-Controller (MVC), yang memisahkan aplikasi menjadi tiga komponen: model, yang menangani data dan akses database, tampilan, yang menangani presentasi data, dan kontroler, yang menangani masukan pengguna dan mengelola interaksi antara model dan tampilan.

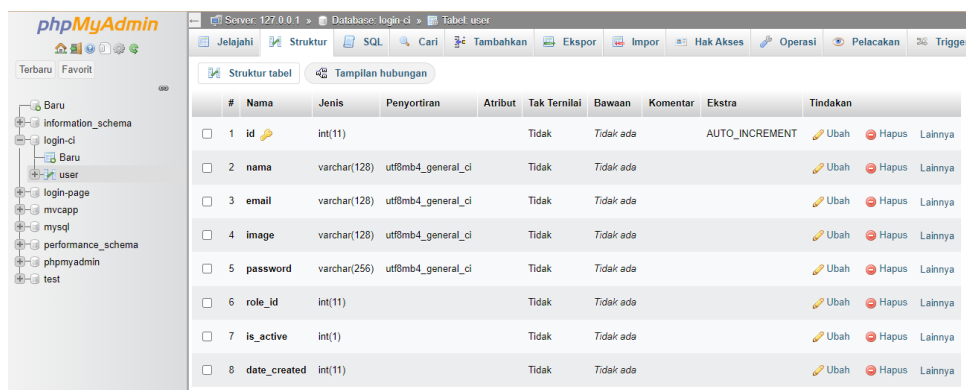
Salah satu keuntungan utama dari CodeIgniter adalah desainnya yang ringan dan minimalis, yang membuatnya cepat dan mudah untuk diinstal dan memulai. Ini juga

memiliki komunitas besar dan aktif pengguna, yang menyediakan sejumlah sumber daya dan dukungan bagi pengembang.

Secara keseluruhan, CodeIgniter adalah kerangka yang kuat dan fleksibel yang dapat digunakan untuk membangun berbagai aplikasi web, dari situs sederhana hingga aplikasi kompleks dan data-intensif.

E. Langkah kerja

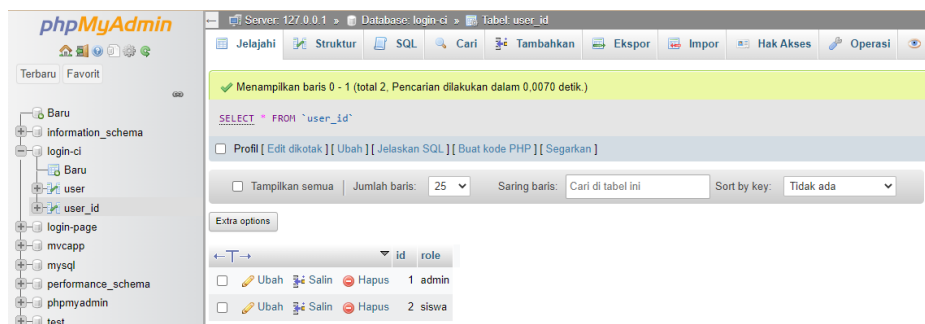
1. Buka kembali lembar kerja yang sudah dibuat pada pertemuan sebelumnya.
2. Buatlah database baru menggunakan phpmyadmin
3. Kemudian buat tabel baru dengan nama user



The screenshot shows the phpMyAdmin interface with the 'user' table selected. The table structure is as follows:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	nama	varchar(128)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	email	varchar(128)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	image	varchar(128)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	password	varchar(256)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	role_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
7	is_active	int(1)			Tidak	Tidak ada			Ubah Hapus Lainnya
8	date_created	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya

4. Buat juga tabel baru dengan nama user_id



The screenshot shows the phpMyAdmin interface with the 'user_id' table selected. The table contains two rows of data:

id	role
1	admin
2	siswa

5. Buka file registrasi.php, kemudian tambahkan script pada tag form, menjadi seperti gambar dibawah. Pastikan tombolnya memiliki type submit.

```
25 </div>
26 <div class="card-body">
27 <form method="post" action="{base_url('auth/register')}">
28 <div class="row mb-3">
```

6. Buka controller Auth, kemudian tambahkan script pada class Auth. Seperti gambar dibawah.

```
4 class Auth extends CI_Controller
5 {
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->library('form_validation');
10    }
11
12    public function index()
13    {
14        $this->load->view('login');
15    }
16 }
```

7. Kemudian tambahkan script seperti dibawah untuk membuat validasi terhadap data yang dimasukan pada method register.

```
17 public function register()
18 {
19     if ($this->form_validation->run() == false) {
20         $data['tittle'] = 'User Registration';
21         $this->load->view('register');
22     } else {
23         echo "data berhasil disimpan";
24     }
25 }
```

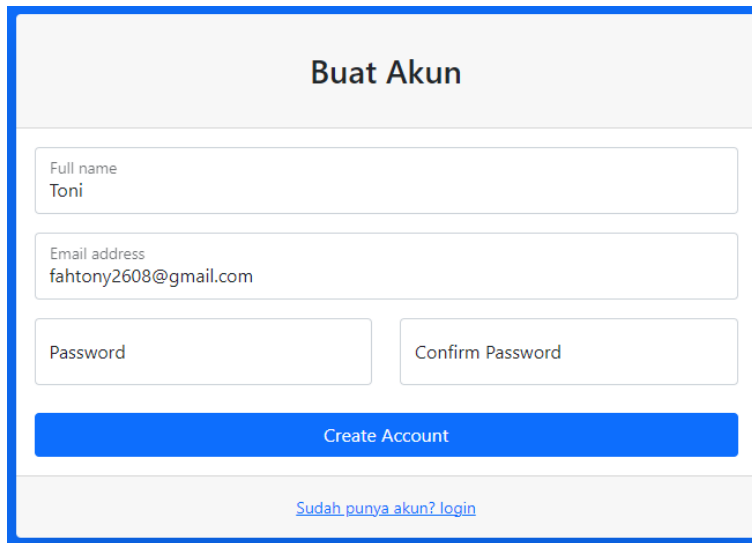
8. Tambahkan script pada method register untuk membuat aturan terhadap data yang akan ditambahkan, buatlah seperti gambar dibawah.

```
17 public function register()
18 {
19     $this->form_validation->set_rules('name', 'Name', 'required|trim');
20     $this->form_validation->set_rules('email', 'Email', 'required|trim|valid_email|is_unique[user.email]', [
21         'is_unique' => 'This email has already registered!'
22     ]);
23
24     if ($this->form_validation->run() == false) {
25         $data['tittle'] = 'User Registration';
26         $this->load->view('register');
27     } else {
28         echo "data berhasil disimpan";
29     }
30 }
31 }
```

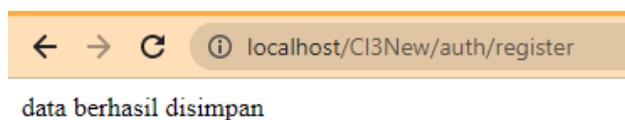
9. Buka file register.php pada folder view, kemudian tambahkan atribut name pada form input. Tambahkan semua nama pada form input dan sesuaikan dengan form_validation pada method register.

```
<form method="post" action="{>= base_url('auth/register') ?>">
  <div class="row mb-3">
    <div class="col-md-12">
      <div class="form-floating mb-3 mb-md-0">
        <input class="form-control" name="name" id="inputFirstName" type="text" placeholder="Enter your full name" />
        <label for="inputFirstName">Full name</label>
      </div>
    </div>
  </div>
  <div class="form-floating mb-3">
    <input class="form-control" id="inputEmail" name="email" type="email" placeholder="name@example.com" />
    <label for="inputEmail">Email address</label>
  </div>
</form>
```

10. Uji coba masukan data pada form, seperti gambar dibawah.



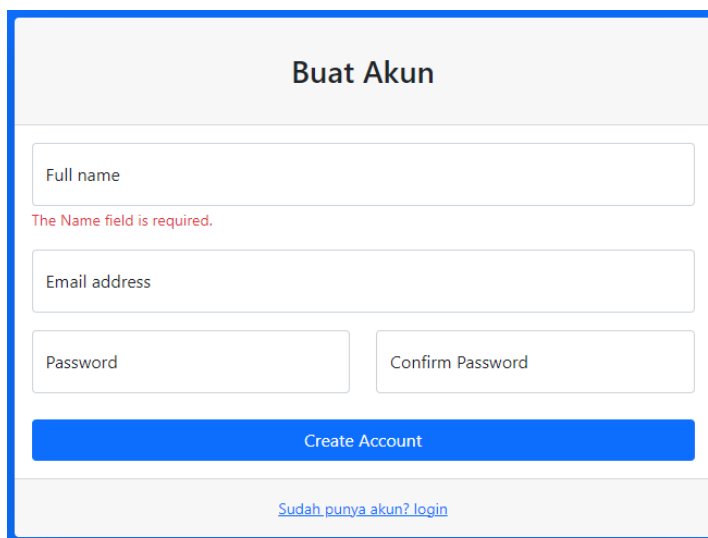
11. Kemudian jalankan, jika tamlilan seperti gambar dibawah, artinya form_validation berhasil ditambahkan.



12. Buka kembali file register pada folder view, kemudian tambahkan script pada input apabila data yang dimasukan tidak memenuhi aturan yang sudah dibuat.

```
<div class="col-md-12">
  <div class="form-floating mb-3 mb-md-0">
    <input class="form-control" name="name" id="inputFirstName" type="text" placeholder="Enter your full name" />
    <? = form_error('name', '<small class="text-danger pl-3">', '</small>'); ?>
    <label for="inputFirstName">Full name</label>
  </div>
</div>
```

13. Maka hasilnya akan seperti gambar dibawah, apabila form kosong. Karena pada aturan kita buat form nama tidak boleh kosong.



14. Buat hal yang sama pada semua form input yang ada pada file register.php
15. Perbaiki pada script input, agar setiap data yang dimasukkan pada form tidak hilang apabila terdapat inputan terdapat error. Pada form input bisa ditambahkan script seperti dibawah.

```
<div class="row mb-3">
  <div class="col-md-12">
    <div class="form-floating mb-3 mb-md-0">
      <input class="form-control" name="name" id="inputFirstName" type="text" placeholder="Enter
your full name" value="<?= set_value('name'); ?>" />
      <?= form_error('name', '<small class="text-danger pl-3">', '</small>'); ?>
      <label for=" inputFirstName">Full name</label>
    </div>
  </div>
</div>
<div class="form-floating mb-3">
  <input class="form-control" id="inputEmail" name="email" type="email" placeholder="name@example.
com" value="<?= set_value('email'); ?>" />
  <?= form_error('email', '<small class="text-danger pl-3">', '</small>'); ?>
  <label for=" inputEmail">Email address</label>
</div>
```

16. Maka hasilnya akan seperti gambar dibawah. Terlihat data pada fom input nilai tidak sesuai dengan aturan yang dibuat, maka error muncul pada form email dan data pada form nama tidak hilang.

17. Buka kembali file controller auth, tambahkan kembali aturan untuk email dan password.

```
17 public function register()
18 {
19     $this->form_validation->set_rules('name', 'Name', 'required|trim');
20     $this->form_validation->set_rules('email', 'Email', 'required|trim|valid_email|is_unique[user.email]', [
21         'is_unique' => 'This email has already registered!'
22     ]);
23     $this->form_validation->set_rules('password1', 'Password', 'required|trim|min_length[3]|matches[password2]', [
24         'matches' => 'Password dont match!',
25         'min_length' => 'Password too short!'
26     ]);
27     $this->form_validation->set_rules('password2', 'Password', 'required|trim|matches[password1]');
```

18. Kemudian tambahkan pada baris else di method register, seperti gambar dibawah.

```
31     if ($this->form_validation->run() == false) {
32         $data['title'] = 'User Registration';
33         $this->load->view('register');
34     } else {
35         $email = $this->input->post('email', true);
36         $data = [
37             'name' => htmlspecialchars($this->input->post('name', true)),
38             'email' => htmlspecialchars($email),
39             'image' => 'default.jpg',
40             'password' => password_hash($this->input->post('password1'), PASSWORD_DEFAULT),
41             'role_id' => 2,
42             'is_active' => 0,
43             'date_created' => time()
44         ];
45
46         $this->db->insert('user', $data);
47
48         $this->session->set_flashdata('message', '<div class="alert alert-success" role="alert">Congratulation! Your account has been
49         created. Please activated
50         </div>');
51         redirect('auth');
52     }
```

19. Kemudian setting database agar data pada sistem yang kita bangun dapat tersimpan di database.

20. Buka file database.php pada folder config, kemudian setting beberapa baris.

```
76     $db['default'] = array([
77         'dsn' => '',
78         'hostname' => 'localhost',
79         'username' => 'root',
80         'password' => '',
81         'database' => 'login-ci',
82         'dbdriver' => 'mysqli',
83         'dbprefix' => '',
```

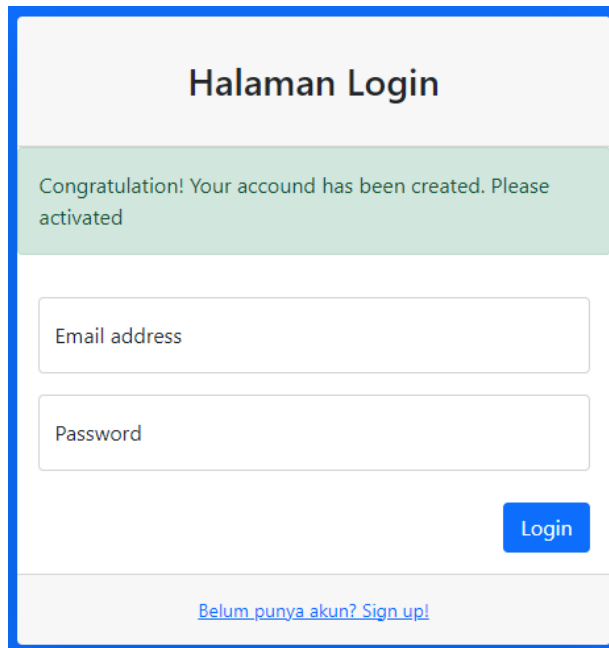
21. Buka kembali fileautoload.php pada folder configtambahkan database pada baris libraries.

```
60     */
61     $autoload['libraries'] = array('email', 'session', 'database');
62
```

22. Buka kembali file login.php pada folder views untuk menambahkan script yang bertujuan untuk memberikan pemberitahuan berhasil atau tidaknya data masuk ke database.

```
25         </div>
26         <?=$this->session->flashdata('message'); ?>
27         <div class="card-body">
```

23. Coba tambahkan data melalui form yang telah dibuat, jika berhasil maka akan redirect ke halaman login, dan akan keluar tampilan pemberitahuan bahwa data berhasil ditambahkan.



Halaman Login

Congratulation! Your account has been created. Please activated

Email address

Password

Login

[Belum punya akun? Sign up!](#)

24. Cek database pada tabel user, jika berhasil maka data akan tampil pada database.

	id	nama	email	image	password	role_id	is_active	date_created
<input type="checkbox"/> Ubah <input type="text"/> Salin <input type="text"/> Hapus	1		fahtony2608@gmail.com	default.jpg	\$2y\$10\$zn2dLUmoedDbXYNjzuBmB.IGx94qPgMohi0.nvmtSkc...	2	0	1676950169

F. Tugas

1. Identifikasi error yang didapat.
2. Cari solusi terhadap error tersebut.
3. Buat laporan dengan menggunakan format laporan yang pernah dibuat.