

BAB 1

A. Definisi

1. PHP (*PHP Hypertext Preprocessor*)

Sejarah awal PHP adalah pendekatan dari *Personal Home Page* (situs personal), dulu PHP masih berbentuk *script* yang berfungsi sebagai pengolah data form dari *web server-side* yang memiliki sifat *open source*, dan memiliki nama *Form Interpreter* (FI). PHP (*Hypertext Preprocessor*) memiliki sintak yang mirip dengan ASP, Java, bahasa C, Perl dan memiliki kelebihan fungsi yang mudah dipahami dan spesifik. Versi terkini dari PHP adalah PHP7. Kelebihan PHP yaitu:

- a. Bahasa pemrograman PHP tidak memerlukan *Compiler* dalam penggunaanya.
- b. Memiliki sifat *open source*.
- c. Banyak sekali aplikasi PHP yang gratis dan siap untuk digunakan seperti PrestaShop, WordPress, dll.
- d. Bisa membuat web menjadi dinamis.
- e. PHP memiliki banyak dukungan dari berbagai *web server* contohnya saja Apache.
- f. PHP memiliki keunggulan lebih cepat dibandingkan dengan Java dan ASP.
- g. MySQL merupakan paket aplikasi dengan PHP.

Kekurangan PHP yaitu:

- a. PHP memiliki kelemahan keamanan.
- b. Kode PHP bisa dibaca oleh semua orang.
- c. Biaya untuk *encoding* membutuhkan biaya yang sangat mahal

2. CodeIgniter

CodeIgniter adalah sebuah kerangka kerja untuk *web* yang dibuat dalam format PHP. Format yang dibuat dapat digunakan untuk membuat sistem aplikasi *web* yang kompleks. CodeIgniter dapat mempercepat proses pembuatan *web*, karena *class* dan *module* yang dibutuhkan sudah tersedia dan *programmer* hanya tinggal menggunakannya kembali pada aplikasi *web* yang akan dibuat.

CodeIgniter memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Tidak seperti beberapa *framework* PHP lainnya, dokumentasi untuk *framework* ini sangat lengkap, yang mencakup seluruh aspek dalam *framework*. CodeIgniter juga mampu berjalan pada lingkungan *Shared Hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat luar biasa. Dari segi pemrograman, CodeIgniter kompatibel dengan PHP4 dan PHP5, sehingga akan berjalan dengan baik pada *web host* yang banyak dipakai saat ini. CodeIgniter menggunakan pola desain *Model-View-Controller* (MVC), yang merupakan cara untuk mengatur aplikasi web ke dalam 3 bagian yang berbeda. Pada intinya CodeIgniter juga membuat penggunaan ekstensif dari pola desain Singleton. Maksudnya adalah cara untuk *me-load class* sehingga jika *class* itu di panggil dalam beberapa kali, kejadian yang sama pada *class* tersebut akan digunakan kembali. Hal ini

sangat berguna dalam koneksi *database*, karena kita hanya ingin menggunakan satu koneksi setiap kali *class* itu digunakan.

3. Javascript

JavaScript dibuat dan didesain selama sepuluh hari oleh Brandan Eich, seorang karyawan Netscape, pada bulan September 1995. Awalnya bahasa pemrograman ini disebut Mocha, kemudian diganti ke Mona, lalu LiveScript sebelum akhirnya resmi menyandang nama JavaScript. Versi pertama dari bahasa ini hanya terbatas di kalangan Netscape saja. Fungsionalitas yang ditawarkan pun terbatas. Namun, JavaScript terus dikembangkan oleh komunitas developer yang tak henti-hentinya mengerjakan bahasa pemrograman ini.

JavaScript adalah salah satu bahasa pemrograman yang paling banyak digunakan dalam kurun waktu dua puluh tahun ini. Bahkan JavaScript juga dikenal sebagai salah satu dari tiga bahasa pemrograman utama bagi web developer:

- HTML: Memungkinkan Anda untuk menambahkan konten ke halaman web.
- CSS: Menentukan *layout, style*, serta keselarasan halaman *website*.
- JavaScript: Menyempurnakan tampilan dan sistem halaman web.

JavaScript dapat dipelajari dengan cepat dan mudah serta digunakan untuk berbagai tujuan, mulai dari meningkatkan fungsionalitas *website* hingga mengaktifkan permainan (*games*) dan *software* berbasis web. Selain itu,

terdapat ribuan template dan aplikasi JavaScript yang bisa Anda gunakan secara gratis dan semuanya ini berkat beberapa situs, seperti Github.

4. JQuery

jQuery adalah library JavaScript yang populer. Bahasa pemrograman ini dibuat oleh John Resig, tepatnya pada tahun 2006, untuk memudahkan para *developer* dalam menggunakan dan menerapkan JavaScript di *website*. jQuery bukanlah bahasa pemrograman yang berdiri sendiri, melainkan bekerja sama dengan JavaScript. Dengan menggunakan jQuery, Anda bisa melakukan banyak hal.

Seperti yang kita ketahui, menulis kode bukanlah pekerjaan yang mudah dan terkadang menyulitkan, terlebih lagi kalau ada banyak string kode yang harus ditambahkan dan diaktifkan. Di sinilah jQuery memainkan perannya. Fungsi jQuery adalah meng-*compress* berbagai baris atau line kode ke dalam satu buah fungsi sehingga Anda tidak perlu menulis kembali semua baris kode hanya untuk menyelesaikan satu task. Salah satu alasan mengapa jQuery sangat populer dan banyak digunakan adalah kemampuan lintas platformnya. Secara otomatis, jQuery akan memperbaiki error serta punya fungsi yang sama seperti ketika dijalankan di browser, seperti Chrome, Firefox, Safari, MS-Edge, IE, Android, dan iOS.

Adanya jQuery juga memudahkan penggunaan Ajax. Ajax menganut sistem kerja yang asinkron, dan umumnya dimulai dari kode yang tersisa. Hal ini berarti kode yang ditulis dengan Ajax dapat berkomunikasi dengan server dan memperbarui kontennya tanpa harus me-*load* kembali halaman terlebih dulu. Sayangnya, tidak selamanya penggunaan Ajax lancar dan mulus-mulus saja. Setiap browser mengaktifkan Ajax Api dengan cara yang berbeda-beda.

Karena itulah, sebisa mungkin kode harus diatur agar bisa dijalankan di semua jenis browser.

5. Ajax

AJAX adalah sebuah singkatan dari *Asynchronous Javascript and XML* dan mengacu pada sekumpulan teknis pengembangan web (*web development*) yang memungkinkan aplikasi web untuk bekerja secara *asynchronous* (tidak langsung) – memproses setiap request (permintaan) yang datang ke *server* di sisi background.

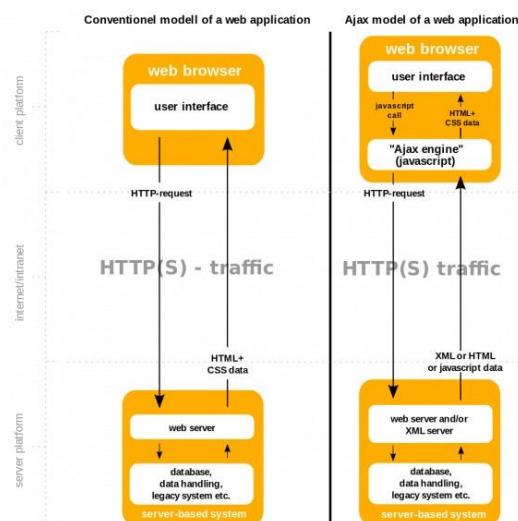
Layaknya HTML, XML atau *eXtensible Markup Language* adalah varian lain dari bahasa markup. Jika HTML dirancang untuk menampilkan data, maka XML dirancang untuk memuat dan membawa data. Baik JavaScript maupun XML bekerja secara *asynchronous* di dalam AJAX. Alhasil, aplikasi web yang menggunakan AJAX dapat mengirimkan dan menerima data dari *server* tanpa harus mereload keseluruhan halaman. Berikut adalah beberapa contoh dari penggunaan AJAX:

- Sistem *Voting* atau *Rating*
- *Chat Room*
- Notifikasi *Trending* di Twitter

AJAX bukanlah teknologi dan bukan pula bahasa pemrograman. Seperti yang telah dijelaskan sebelumnya, AJAX adalah sekumpulan teknik pengembangan web. Pada umumnya sistem ini terdiri atas:

- HTML/XHTML sebagai bahasa utama dan CSS untuk menampilkan data.
- *The Document Object Model (DOM)* untuk menampilkan data yang dinamis beserta interaksinya.
- XML untuk pertukaran data, sedangkan XSLT untuk manipulasi data. Sebagian besar *developer* mulai mengganti XML dengan JSON karena bentuknya yang mendekati JavaScript.
- Objek *XMLHttpRequest* untuk komunikasi tidak langsung (asynchronous).
- Bahasa pemrograman JavaScript untuk menyatukan semua teknologi ini.

Untuk memahami cara kerja AJAX secara keseluruhan, setidaknya Anda harus punya pemahaman teknis dasar terlebih dulu. Untungnya, prosedur umum dari cara kerja AJAX tidak begitu sulit. Lihat diagram dan tabel di bawah ini untuk perbandingannya.



Gambar 1 Diagram Alur AJAX

6. MySQL

MySQL adalah sebuah *database management system* (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. *Database management system* (DBMS) MySQL multi pengguna dan multi alur ini sudah dipakai lebih dari 6 juta pengguna di seluruh dunia.



Gambar 2 Logo MySQL

MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU General Public License (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (Relational Database Management System). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel. SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada relational database atau database yang terstruktur. Jadi MySQL adalah database management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server.

Penggunaan dari MySQL sendiri tentu memiliki kelebihan dan kekurangannya, berikut adalah kelebihan dari MySQL:

- Mendukung integrasi dengan bahasa pemrograman lain
- Tidak membutuhkan RAM besar
- Mendukung *multi user*
- Bersifat *opensource*
- Struktur tabel yang fleksibel
- Tipe data yang bervariasi

Selain itu kekurangan dari penggunaan MySQL sebagai berikut:

- Kurang cocok untuk aplikasi *mobile* dan *game*
- Sulit mengelola *database* yang besar
- *Technical support* yang kurang bagus

BAB 2

B. Metode Penyelesaian Aplikasi

1. Definisi Scrum

Scrum yang masih merupakan bagian dari metodologi *agile*. *Scrum* merupakan sebuah proses pengembangan perangkat lunak dengan konsep *agile* yang dikembangkan oleh Jeff Sutherland dan Ken Schwaber pada tahun 1996. Model *scrum* biasanya digunakan karena menerapkan proses dengan siklus pendek berulang, secara aktif melibatkan pengguna untuk membangun, memprioritaskan, dan memverifikasi kebutuhan.

Kelebihan *scrum* adalah dapat menghasilkan produk sesuai dengan keinginan pengguna dan cocok untuk pengembangan sistem dengan skala kecil serta banyak perubahannya. Di sisi lain, *scrum* merupakan metodologi yang paling sering digunakan oleh banyak perusahaan besar karena merupakan salah satu metodologi yang menggunakan konsep *agile* dengan keunggulannya yaitu lebih cepat dan lebih efektif dalam pengembangan perangkat lunak.

Berikut merupakan beberapa aktifitas yang dilakukan dalam penerapan metodologi *scrum*:

1. *Backlog* atau yang lebih dikenal *product backlog* adalah daftar kebutuhan atau fitur yang memberikan nilai bisnis kepada klien. *Product backlog* dapat bertambah seiring dengan kebutuhan penggunanya.
2. *Scrum* membagi sebuah *project* ke dalam sebuah perulangan yang disebut sebagai *sprint*. Sebuah *sprint* biasanya memiliki durasi

pengerjaan hingga 4 minggu, dimana tim pengembang fokus dalam mencapai target yang jelas dan telah ditentukan.

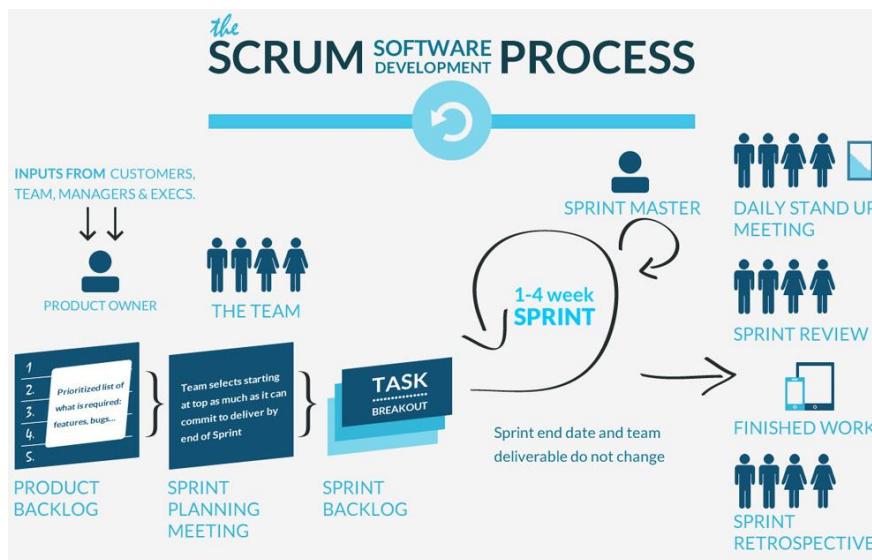
3. *Scrum meetings*, merupakan pertemuan rutin yang dilakukan perhari atau perminggu untuk mengevaluasi apa yang dikerjakan, hambatan, dan target penyelesaiannya. Dalam hal ini, penulis melakukan *scrum meetings* setiap minggu dengan membahas apa saja yang dikerjakan setiap minggunya serta hambatan apa yang dialami selama proses pengerjaan itu.
4. *Demo*, memberikan perubahan atau *update* kepada klien yang telah diimplementasikan agar dapat dilihat dan dievaluasi oleh klien.

Selain aktifitas yang telah disebutkan, ada beberapa aktor yang terlibat dalam pengembangan sistem menggunakan model *scrum*, yaitu:

1. *Product owner*, adalah orang yang bertanggung jawab atas keseluruhan sistem atau produk yang sedang dikembangkan
2. *Master scrum*, adalah orang yang mengeliminasi segala hambatan yang muncul dalam pengembangan proyeknya
3. *Team scrum*, adalah sekumpulan orang yang dapat terdiri dari *programmers*, *testers*, dan para ahli lainnya yang bertugas untuk mengembangkan produk dan memberikan kepuasan terhadap klien melalui produk yang dihasilkan.

2. Tahapan Metodologi Scrum

Untuk alur yang lebih jelas dari *Scrum*, maka lihatlah gambar 3 di bawah ini:



Gambar 3 Alur Scrum

Dengan diagram alur yang telah ditunjukkan pada gambar 3, alur tersebut dapat dibagi menjadi 3 tahapan utama yaitu tahap *pregame*, *game*, dan *post game*.

Berikut adalah tahapan – tahapan yang harus dilakukan:

1. Tahap *pregame* berfokus pada melakukan pengidentifikasi permasalahan yang terjadi, melakukan pengumpulan data yang relevan terkait dengan permasalahan yang terjadi serta melakukan perencanaan untuk membuat solusi dari masalah yang sedang dihadapi. Tahap *pregame* berisi aktifitas *scrum* seperti *backlog/product backlog*.

2. Tahap *game* dilakukan perancangan sesuai dengan perencanaan yang telah dibuat pada tahapan sebelumnya, biasanya pada tahapan ini berisi aktifitas *sprint* dan juga *scrum meetings*.
3. Sedangkan pada tahap *post game* dilakukan pengujian akhir guna melihat apakah sistem yang dikembangkan sudah memenuhi kebutuhan pengguna. Di dalam tahapan yang terakhir berisi aktifitas seperti *demo*.

Biasanya, untuk setiap *project* akan berbeda – beda *product backlog* dan *sprint* yang dilakukan tergantung bagaimana kompleksitas *project* itu atau faktor lainnya. Lalu untuk melakukan *scrum meetings* yang membahas kendala dan juga *milestone* dari *project* minimal satu minggu sekali yang seringkali harus menyesuaikan dengan jadwal dari *product owner*.

Untuk itu, sebelum dimulainya pengerjaan proyek perlu dilakukan pendefinisian yang jelas siapa saja *actor* yang terlibat dan juga *product backlog* apa saja yang akan dibuat. Selain itu *sprint* dari setiap *product backlog* juga perlu didefinisikan. Maka berikut adalah tabel yang mendefinisikan keseluruhan *actor*, *activity*, *product backlog*, dan juga *sprint* dari pengerjaan proyek.

Tabel 1 Tabel *Actor* dan *Activity Scrum*

<i>Activity and Actor</i>	<i>General</i>	
<i>Team Scrum</i>	M. Fauzan Aristyo	
	Muhamad Wahyunda	
	Dimas Haryoputra Maheswara	
	Riandaka Rizal	
<i>Scrum Master</i>	Riandaka Rizal	
<i>Product Owner</i>	M. Fauzan Aristyo	
	<i>Product Backlog</i>	<i>Sprint</i> (estimasi)
<i>Backlog</i>	<i>Create, update, delete, view</i> data peserta EPPM	1 minggu
	<i>Create, update, delete, view</i> data <i>workbook</i>	1 minggu
	<i>Create, update, delete, view</i> data <i>assessment</i> peserta EPPM	1 minggu
	Menu raport	3 hari
	Visualisasi data progres	4 hari
Total Estimasi Pengerjaan		4 minggu

BAB 3

C. Tutorial Pembuatan Aplikasi EPPM Go!

1. Inisiasi penggerjaan aplikasi

Untuk pembuatan aplikasi EPPM Go! ini akan dibagi menjadi 5 menu untuk admin dan juga 2 menu untuk *user*, menu yang ditujukan untuk admin adalah sebagai berikut:

- Menu Daftar Pekerja
- Menu *Workbook*
- Menu *Assessment*
- Menu Ruangkerja
- Menu Raport

Lalu untuk menu yang ditujukan untuk *user* adalah sebagai berikut:

- Menu *Workbook*
- Menu *Raport*

Untuk membuat aplikasi EPPM Go! Diperlukanlah sebuah logo sehingga aplikasi itu dapat dikenali oleh khalayak ramai, untuk logo harus dibuat dengan menarik dan atraktif yang dapat menarik perhatian dan minat orang untuk menggunakannya.



Gambar 4 Logo EPPM Go!

Ada tahapan-tahapan yang harus dilalui mulai dari inisiasi sampai dengan *testing*. Agar lebih terstruktur maka harus didefinisikan terlebih dahulu tahapannya, yaitu:

1. Download CodeIgniter berupa *zip file*
2. Inisiasi CodeIgniter dengan *unzip file*
3. Membuat *login*
4. Buat CRUD (*Create, Read, Update, Delete*) menu daftar pekerja.
5. Buat CRUD (*Create, Read, Update, Delete*) menu *workbook* baik untuk admin dan *user*.
6. Bagi fungsi yang bisa diakses oleh admin, dan batasi fungsi yang hanya bisa diakses oleh *user*
7. Buat CRUD (*Create, Read, Update, Delete*) menu *assessment* untuk admin.
8. Di dalam menu *assessment*, buat pembobotan nilai untuk penilaian yang akan dilakukan. 40% untuk nilai teori dan 60% untuk nilai praktik
9. Buat CRUD (*Create, Read, Update, Delete*) menu ruangkerja untuk admin.
10. Di dalam menu ruangkerja, buat fungsi *import* untuk mengimport *file excel* atau *csv* agar dapat efisien merubah data di dalam menu ruangkerja

11. Tampilkan hasil penilaian *assessment* di menu raport tanpa filter untuk admin dan filter sesuai dengan akun *user* jika digunakan oleh *user*.
12. Tambahkan fungsi *export* agar dapat diunduh di komputer masing masing.
13. Lalu pada halaman utama (*dashboard*) dari admin, tambahkan grafik untuk dapat memvisualisasikan data dari menu *workbook*.
14. Untuk grafik, gunakan ChartJS API agar dapat menggunakan grafik yang dinamis.

Pada halaman *user* atau pengguna, diperlukan sebuah gambar yang menarik agar mendapat kesan pertama dari pengguna itu sendiri berikut adalah gambar untuk halaman awal pengguna



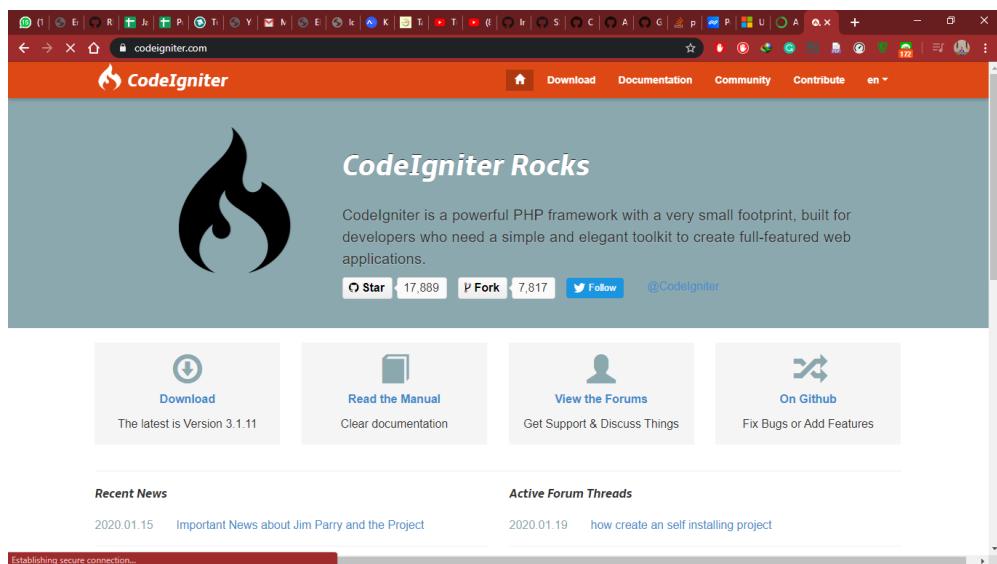
Gambar 5 Halaman Awal Pengguna

Dapat dilihat pada gambar 5, ada beberapa komponen dari gambar yang disesuaikan dengan kebiasaan dari pengguna sendiri. Hal ini merupakan suatu usaha untuk mendapatkan kesan pertama pengguna sendiri.

2. Pembuatan Aplikasi

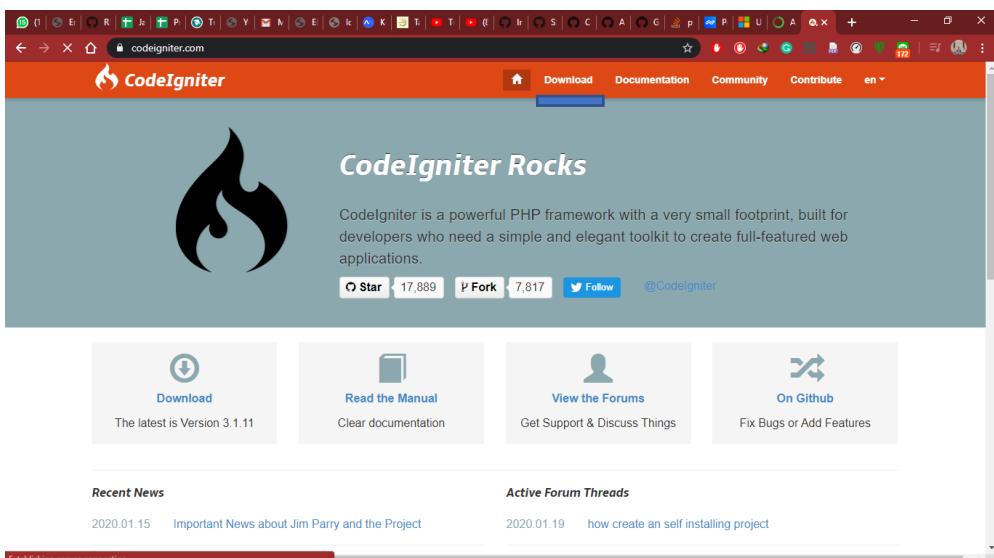
2.1. Inisiasi CodeIgniter

Untuk pembuatan aplikasi, maka harus disesuaikan dengan langkah yang telah diinisiasi sebelumnya. Maka hal pertama yang harus dilakukan adalah *download* CodeIgniter melalui *website* resminya yaitu www.codeigniter.com



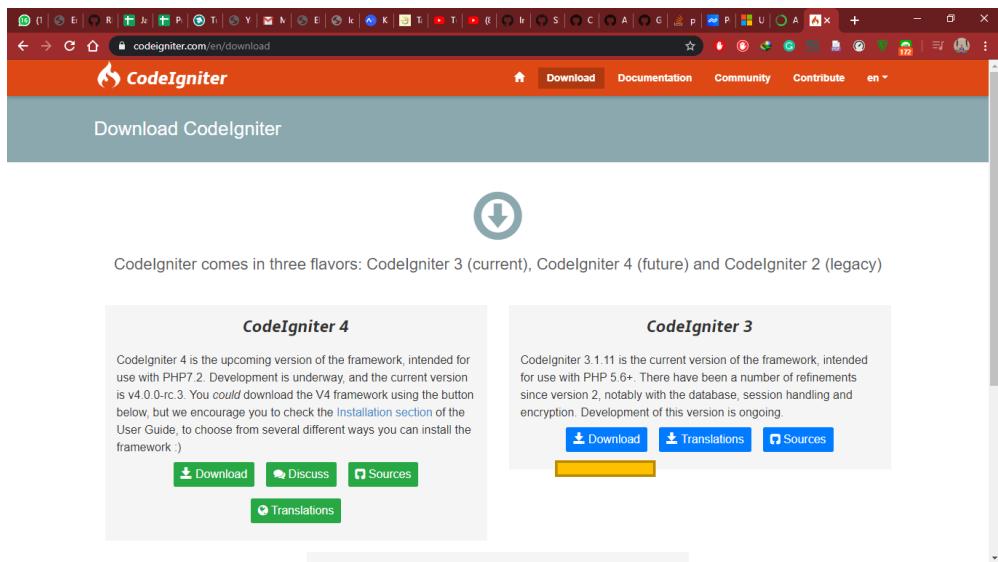
Gambar 6 Website Resmi CodeIgniter

Setelah membuka *website* CodeIgniter pilih menu *download* yang ada di atas menu bar.



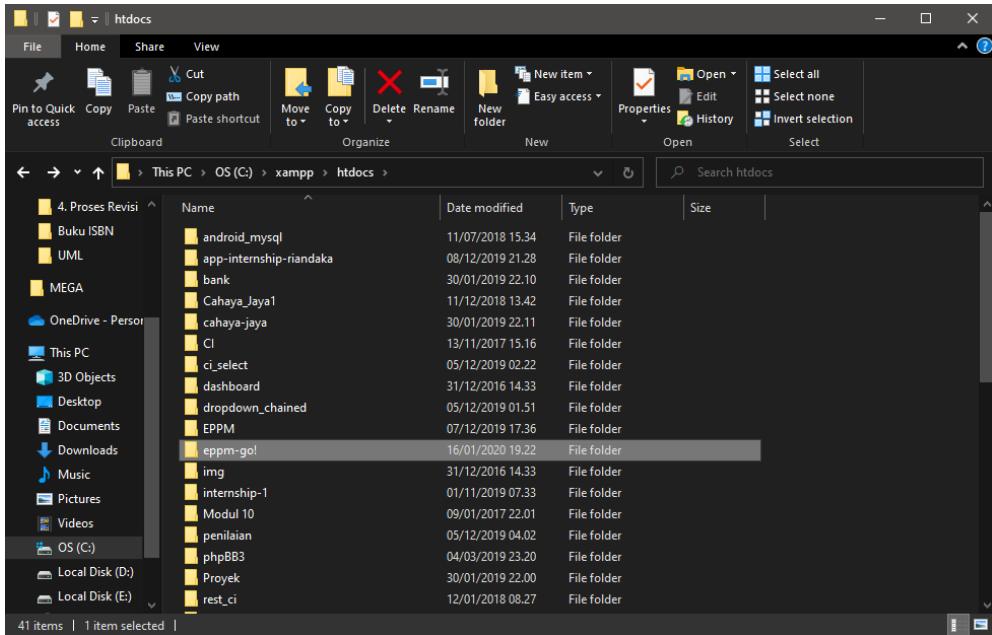
Gambar 7 Pilih Menu Download

Setelah berada di halaman *download* pilih python 3 untuk diunduh, karena saat ini python 3 adalah versi yang paling banyak digunakan oleh pengembang di dunia.



Gambar 8 CodeIgniter 3

Tunggulah beberapa saat untuk selesai mengunduh *file zip* dari CodeIgniter. Setelah itu *unzip file* CodeIgniter yang baru saja kita unduh ke dalam folder C://xampp/htdocs dan *rename* folder CodeIgniter menjadi eppm-go! Seperti yang ditunjukkan pada gambar 9.



Gambar 9 Folder eppm-go!

Setelah selesai membuat folder eppm-go! silakan lanjut ke langkah selanjutnya, yaitu membuat CRUD (*Create, Read, Update, Delete*) untuk menu daftar pekerja.

2.2. Login

Buat file dengan nama Login.php pada controller, nama file untuk controller selalu disarankan untuk berawalan huruf besar. Lalu buka file tersebut dan ketikkan seperti yang ditunjukkan pada gambar 10

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class Login extends CI_Controller {  
  
    public function __construct()  
    {  
        parent::__construct();  
        //Load library form validasi  
        $this->load->library('form_validation');  
        //Load model admin  
        $this->load->model('admin');  
    }  
}
```

Gambar 10 Controller Login.php

- Class Login extends CI_Controller = maksudnya adalah membuat *class Login* dengan memakai *library* CI_Controller
- Public function __construct() = maksudnya adalah menggunakan function construct
- \$this->load->library('form_validation') = maksudnya adalah *load library* form validasi
- \$this->load->mode('admin') = maksudnya adalah *load* model admin

Setelah itu kita lanjutkan kembali dengan mengetikkan sesuai dengan yang ditunjukkan oleh gambar 11

```

15 |     public function index()
16 |     {
17 |
18 |         if($this->admin->is_logged_in())
19 |         {
20 |             // jika memang session sudah terdaftar, maka redirect ke halaman dashboard
21 |             redirect("admin/dashboard");
22 |
23 |         // }elseif($this->user->is_logged_in()){
24 |
25 |             // redirect("user/dashboard");
26 |
27 |         }else{
28 |
29 |             //jika session belum terdaftar
30 |
31 |             //set form validation
32 |             $this->form_validation->set_rules('username', 'Username', 'required');
33 |             $this->form_validation->set_rules('password', 'Password', 'required');
34 |
35 |             //set message form validation
36 |             $this->form_validation->set_message('required', '<div class="alert alert-danger" style="margin-top: 3px">
37 |                 <div class="header"><b><i class="fa fa-exclamation-circle"></i> {field}</b> harus diisi</div></div>');
38 |
39 |             //cek validasi
40 |             if ($this->form_validation->run() == TRUE) {
41 |
42 |                 //get data dari FORM
43 |                 $username = $this->input->post("username", TRUE);
44 |                 $password = $this->input->post("password", TRUE);
45 |
46 |                 //checking data via model
47 |                 $checking = $this->admin->check_login('account', array('username' => $username), array('password' => $pass

```

Gambar 11 Controller Login.php

Seperti yang dilihat pada gambar 11, untuk function index akan mengecek terlebih dahulu apakah sudah ada *session* yang dibuat apa belum, jika sudah maka akan langsung *redirect* ke halaman *dashboard* admin. Jika belum maka akan membuat form validasi dengan *username* dan *password* yang harus diisi lalu akan dicek apakah data *username* dan *password* yang kita masukkan sudah ada atau belum,

```

19     //jika ditemukan, maka create session
20     if ($checking != FALSE) {
21         foreach ($checking as $apps) {
22
23             $session_data = array(
24                 'employee_id' => $apps->employee_id,
25                 'user_name' => $apps->username,
26                 'user_pass' => $apps->password,
27                 'user_email' => $apps->user_email,
28                 'user_nama' => $apps->employee_name,
29                 'directorate' => $apps->directorate,
30                 'role' => $apps->role
31             );
32             //set session userdata
33             // print_r($session_data); die;
34             $this->session->set_userdata($session_data);
35
36             //redirect berdasarkan Level user
37             if($this->session->userdata("role") == "admin"){
38                 redirect('admin/dashboard');
39             }else{
40                 redirect('user/dashboard');
41             }
42         }
43     }else{
44
45         $error = 'Username atau Password Anda salah. Silakan ulangi Kembali';
46         '<div class="alert alert-danger" style="margin-top: 3px">';
47         '<div class="header"><b><i class="fa fa-exclamation-circle"></i> ERROR</b> username atau password salah!';
48         $this->load->view('login', $error);
49     }
50 }
```

Gambar 12 Controller Login.php

Jika sudah ada maka akan langsung dibuat *session*-nya dan akan *redirect* ke halaman awal sesuai dengan hak aksesnya masing masing. Jika itu admin maka akan diarahkan ke halaman *dashboard* admin, sedangkan jika *user* maka akan diarahkan ke halaman awal untuk pengguna.

```

92     public function logout()
93     {
94         $this->session->sess_destroy();
95         redirect('index.php/login');
96     }
97 }
```

Gambar 13 Controller Login.php

Lalu setelah itu, kita membuat function logout() yang berguna untuk keluar dari aplikasi dengan cara menghapus atau menghancurkan *session* yang sedang berjalan dan mengarahkannya ke halaman login. Lanjut lagi buat kembali *file* dengan nama Admin.php pada folder model

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Admin extends CI_Model
5 {
6     //fungsi cek session Logged in
7     function is_logged_in()
8     {
9         return $this->session->userdata('id');
10    }
11 }
```

Gambar 14 Models Admin.php

Seperti yang dapat dilihat pada gambar 14, ada function is_logged_in() yang berguna untuk mengecek apakah *session* telah ada dan dibuat. Lalu hasil yang dikeluarkan dari function ini berupa userdata dengan membawa id dari user yang sedang *login*. Lalu dilanjutkan dengan function is_role() yang ditunjukkan oleh gambar 15, dimana akan mengecek *role* dari setiap pengguna yang masuk sehingga akan diberikan hak akses sesuai dengan level atau *role*-nya.

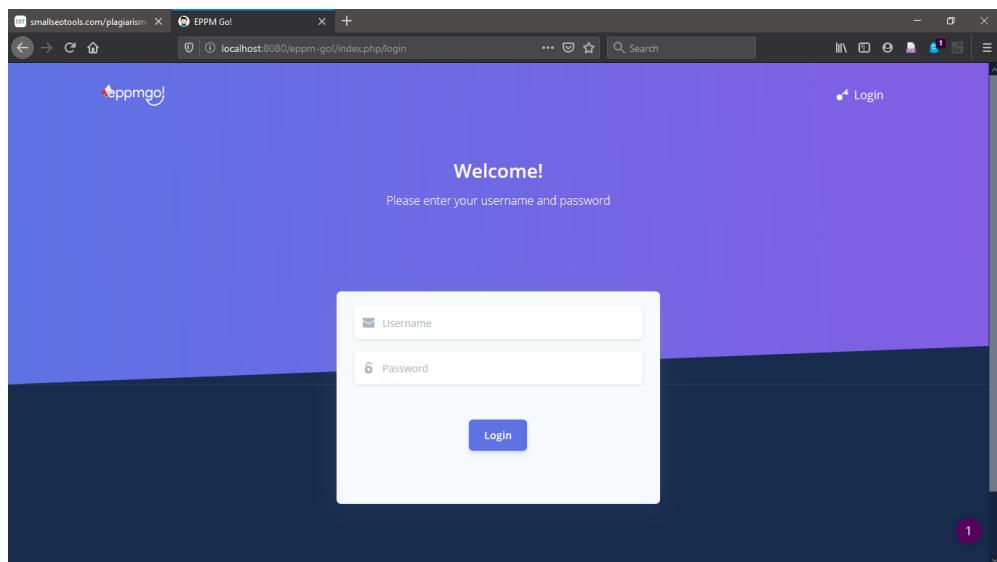
```
//fungsi cek level
function is_role()
{
    return $this->session->userdata('role');
```

Gambar 15 Models Admin.php

Lalu untuk *view*-nya silakan berkreasi sendiri menggunakan bootstrap pilihan anda sendiri. Disini saya menggunakan bootstrap dari argon dashboard. Pastikan pada tombol login sudah diisikan form methodnya seperti yang ditunjukkan pada gambar 16 berikut.

```
<form method="POST" action="=base_url();?&gt;index.php/login"&gt;</pre
```

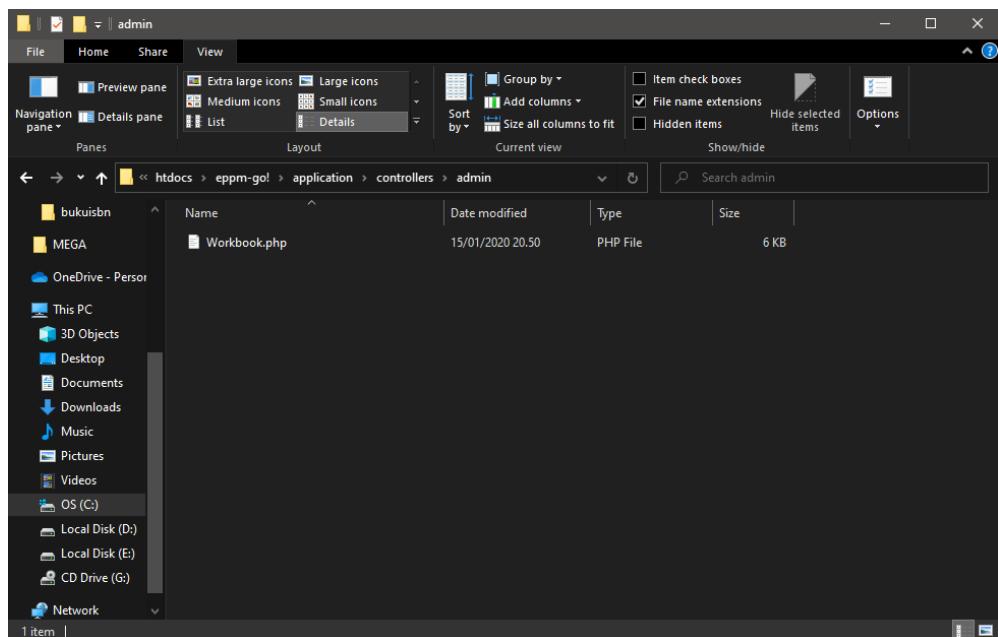
Gambar 16 Form Method Login



Gambar 17 Halaman Login

2.3. CRUD Menu Workbook

Untuk membuat CRUD menu daftar pekerja, mulai sekarang akan kita akan bekerja menggunakan HTML, PHP, dan juga *database*-nya. Untuk *database*-nya akan dijelaskan di lain bab. Pertama buatlah *file* bernama Workbook.php dan folder dengan nama admin, ini ditujukan untuk membedakan *file* untuk admin dan juga untuk *user*. Lalu tempatkan pada folder controller pada direktori eppm-go! Untuk penamaan *file* controller disarankan untuk selalu berawalan huruf kapital, agar penamaan *file* menjadi lebih rapi dan dapat dibedakan secara visual. Tetapi apabila lebih suka dibuat dengan berawalan huruf kecil maka sesuaikan saja dengan keinginan masing masing.



Gambar 18 Pembuatan Workbook.php

Setelah membuat file Workbook.php, maka selanjutnya yang akan kita lakukan adalah mulai untuk ngoding. Pada Workbook.php ketik kode berikut sesuai dengan yang ditunjukkan oleh gambar 19.

```
1 <?php
2 defined('BASEPATH') OR exit ('No direct script access allowed');
3
4 class Workbook extends CI_Controller
5 {
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->model("workbook_model"); //Load model workbook
10        $this->load->library('form_validation'); //Load library form validation
11        $this->load->helper(array('form', 'url'));
12        if($this->workbook_model->is_role() != "admin"){
13            redirect("index.php/login");
14        }
15    }
}
```

Gambar 19 Controller Workbook.php

Disini diawali dengan dengan me-*load* dari model workbook yang akan kita bahas selanjutnya, dan juga me-*load library* form validasi. Selain itu, pada controller Workbook.php akan terlebih dahulu mengecek *role* yang akan mengakses halaman *workbook* awal yang function-nya diambil dari model workbook, jika *role* yang masuk adalah admin maka akan langsung menuju dan mengakses halaman *workbook*. Sedangkan bila sebaliknya, maka akan langsung menuju halaman *login*.

Selanjutnya adalah membuat function index, yang di dalamnya terdapat sintaks yang berfungsi untuk me-*load* tampilan halaman *workbook* awal dengan mengambil data yang diinisiasi melalui method getAll() dari model *workbook_model* seperti yang ditunjukkan oleh gambar 20.

```
17 |     public function index()
18 |     {
19 |         $data["workbook"] = $this->workbook_model->getAll(); //ambil data dari model
20 |         $this->load->view("admin/workbook/list_workbook", $data); //load view data model ke workbook
21 |     }
22 | }
```

Gambar 20 Controller Workbook.php

Kemudian, lanjut dengan membuat function add yang berfungsi untuk me-*load* tampilan halaman form input progres *workbook* dengan mengambil data *stages* dari modul pelatihan EPPM dari model *workbook_model* seperti yang ditunjukkan oleh gambar 21.

```
23 |     public function add()
24 |     {
25 |         $data["workbook"] = $this->workbook_model->get_data_stage();
26 |         $this->load->view("admin/workbook/new_form", $data); //Load isi form workbook
27 |     }
28 | }
```

Gambar 21 Controller Workbook.php

Lalu dilanjutkan dengan membuat function aksi_add yang berfungsi untuk menyimpan data yang telah kita *input*-kan pada form *workbook* dengan menggunakan model *workbook_model* dan memberikan pesan apabila data telah sukses disimpan ke dalam *database* dan akan langsung mengarahkan ulang ke halaman *workbook* seperti yang ditunjukkan oleh gambar 22.

```
public function aksi_add()
{
    // print_r('tes'); die;
    $this->workbook_model->save(); //objek model redirect
    $this->session->set_flashdata('Sukses', 'Data Anda Berhasil Disimpan'); //pesan berhasil
    redirect('index.php/admin/workbook');
}
```

Gambar 22 Controller Workbook.php

Jika sudah selesai membuat function aksi_add, maka selanjutnya adalah membuat function baru di dalam file *Workbook.php* dengan nama function edit yang berguna untuk menampilkan halaman edit *workbook* yang data form *input* *workbook* diambil dari *database* dan dilempar melalui model *workbook_model* berdasarkan id yang dipilih seperti yang ditunjukkan oleh gambar 23 berikut.

```
public function edit($id=null)
{
    $data["workbook"] = $this->workbook_model->getById($id); //mengambil data berdasarkan id
    $data["status"] = ['Not Yet Started', 'On Progress', 'Acc. Done']; //mengambil data berdasarkan id
    if (!$data["workbook"]) show_404(); // jika tidak ada show error
    $this->load->view("admin/workbook/edit_form", $data); //Load edit form workbook
}
```

Gambar 23 Controller Workbook.php

Selanjutnya sama seperti function add, kita harus membuat function aksi_edit untuk melakukan proses edit pada aplikasi. Function aksi_edit berfungsi untuk memberikan perubahan pada form yang telah kita simpan sebelumnya. Function aksi_edit dimulai dengan menginisiasikan *config* atau pengaturan tentang folder mana yang menjadi tempat penyimpanan gambar dan mana saja ekstensi untuk *file* gambar yang didukung. Lalu setelah itu akan me-load *library upload* dengan menggunakan *config* yang telah diinisiasi sebelumnya. Jika tidak ada gambar yang diubah atau di-upload maka hanya ubah data yang dimasukkan nilai yang baru dan bila sukses akan langsung diarahkan kembali menuju halaman *workbook* awal.

```
19     public function aksi_edit()
20     {
21         $config['upload_path'] = './upload/workbook'; //path folder
22         $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp'; //type yang dapat diakses bisa anda sesuaikan
23
24
25         $this->load->library('upload', $config);
26         if ( ! $this->upload->do_upload('image')){
27             $id = $this->input->post('workbook_id');
28             $data = array (
29                 'workbook_id' => $this->input->post('workbook_id'),
30                 'stage_id' => $this->input->post('stage_id'),
31                 'modul_id' => $this->input->post('modul_id'),
32                 'employee_name' => $this->input->post('employee_name'),
33                 'progress' => $this->input->post('progress'),
34                 'status' => $this->input->post('status'),
35                 'image' => $this->input->post('old_image')
36             );
37
38             // print_r($data); die;
39             $this->workbook_model->update($id,$data);
40             $this->session->set_flashdata('success', 'Data Anda Berhasil Diupdate');
41             redirect('index.php/admin/workbook');
42         }else{
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
878
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
948
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
978
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1295
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1395
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1495
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1595
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1695
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1795
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2288
2289
2289
2290
2291
2292
2293
2294
2295
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2350
2351
235
```

tetapi bila ada gambar yang diubah atau gambar yang baru saja di-*upload* maka masukkan seluruh data yang baru saja diubah ke dalam *database* dan bila sukses akan langsung diarahkan kembali ke halaman awal *workbook*

```
else{
    $id = $this->input->post('workbook_id');
    $image = $this->upload->data();
    $data = array (
        'workbook_id' => $this->input->post('workbook_id'),
        'stage_id' => $this->input->post('stage_id'),
        'modul_id' => $this->input->post('modul_id'),
        'employee_name' => $this->input->post('employee_name'),
        'progress' => $this->input->post('progress'),
        'status' => $this->input->post('status'),
        'image' => $image['file_name']
    );
    // print_r($data); die;
    $this->workbook_model->update($id,$data);
    $this->session->set_flashdata('success', 'Data Anda Berhasil Diupdate');
    redirect('index.php/admin/workbook');
}
```

Gambar 25 Controller Workbook.php

Setelah membuat fungsi untuk *add* dan *edit*, maka kita juga harus membuat fungsi untuk menghapus data yang salah. Oleh karena itu, kita buat function *delete* yang berfungsi untuk menghapus data yang kita pilih atau sesuai dengan id-nya. Jika sudah maka akan diarahkan kembali ke halaman awal *workbook*.

```
public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->workbook_model->delete($id)){
        redirect('index.php/admin/workbook');
    }
}
```

Gambar 26 Controller Workbook.php

Setelah selesai membuat Workbook.php untuk admin, selanjutnya adalah kita harus membuat modelsnya yang berfungsi untuk melakukan manipulasi data yang kita inginkan. Silakan membuat *file* dengan nama Workbook_model.php di dalam folder models. Lanjut dengan mengetikkan *line* berikut sesuai dengan gambar 27.

```
1 <?php defined('BASEPATH') OR exit('No direct script access allowed');
2
3 class Workbook_model extends CI_Model
4 {
5     // private $_table = "workbook";
6
7     public $workbook_id;
8     public $stage_id;
9     public $modul_id;
10    public $employee_id;
11    public $employee_name;
12    public $status;
13    public $image = "default.jpg";
14 }
```

Gambar 27 Models Workbook_model.php

Pada *line* di atas, kita terlebih dahulu menginisiasikan variabel variabel yang akan digunakan, dengan membuat *class* Workbook_model dengan menggunakan *library* CI_Model. Jika sudah selesai, kita harus membuat aturan atau *rules* mana saja *field* yang harus diisi seperti yang ditunjukkan oleh gambar 28 berikut ini.

```
15     public function rules()
16     {
17         return [
18             ['field' => 'stage_id',
19              'rules' => 'required'],
20
21             ['field' => 'modul_id',
22              'rules' => 'required'],
23
24             ['field' => 'employee_id',
25              'rules' => 'required'],
26
27             ['field' => 'employee_name',
28              'rules' => 'required'],
29
30             ['field' => 'progress',
31              'rules' => 'required'],
32
33             ['field' => 'status',
34              'rules' => 'required'],
35
36             ['field' => 'image',
37              'rules' => 'required'],
38
39         ];
40     }
41
```

Gambar 28 Models Workbook_model.php

Lalu kita harus diatur pula hak akses, sehingga setiap pengguna hanya dapat menu yang sesuai dengan hak akses yang diberikan.

```
42     //fungsi cek level
43     function is_role()
44     {
45         return $this->session->userdata('role');
46     }
```

Gambar 29 Models Workbook_model.php

Jika sudah selesai mengatur hak aksesnya, maka langsung kita ambil datanya dari *database* di tabel *workbook* dengan membuat function getAll(). Lalu agar menampilkan data pada halaman *workbook* menjadi lengkap maka kita melakukan teknik *join* yang menggabungkan salah satu *field* di tabel *parent* ke dalam tabel *children*-nya sesuai dengan foreign key-nya.

```
48     public function getAll()
49     {
50         $this->db->select('workbook.*', 'account.employee_name as yaww');
51         $this->db->join('account', 'workbook.employee_id = account.employee_id');
52         $this->db->from('workbook');
53         return $this->db->get()->result();
54     }
```

Gambar 30 Models Workbook_model.php

Sama seperti function getAll(), kita membuat function getById() yang berfungsi untuk mengambil data berdasarkan id-nya pada tabel *workbook* di *database*. Dan kita juga melakukan teknik *join* yang menggabungkan field di tabel *parent* ke dalam tabel *children*-nya sesuai dengan foreign key-nya.

```
56     public function getById($id)
57     {
58         $this->db->select('workbook.*', 'account.employee_name as yaww');
59         $this->db->join('account', 'workbook.employee_id = account.employee_id');
60         $this->db->from('workbook');
61         $this->db->where('workbook_id', $id);
62         return $this->db->get()->row();
63     }
```

Gambar 31 Models Workbook_model.php

Lanjut lagi, kita buat function `get_data_stage()` yang berfungsi untuk mengambil data *stages* pada tabel *workbook* di kolom *stage*. Setelahnya kita buat function `save()` yang berfungsi untuk menyimpan *input*-an pada form dan memasukkan data tersebut ke dalam tabel *workbook* di dalam *database* seperti yang ditunjukkan pada gambar 32.

```
65 |     public function get_data_stage()
66 |     {
67 |         $query = $this->db->get('stage');
68 |         return $query;
69 |     }
70 |
71 |     public function save()
72 |     {
73 |         // $post = $this->input->post();
74 |         // print_r($post); die;
75 |         // $this->workbook_id = $post["workbook_id"];
76 |         $data = array (
77 |             "stage_id" => $this->input->post("stage_id"),
78 |             "modul_id" => $this->input->post("modul_id"),
79 |             "employee_id" => $this->input->post("employee_id"),
80 |             "employee_name" => $this->input->post("employee_name"),
81 |             "progress" => $this->input->post("progress"),
82 |             "status" => $this->input->post("status"),
83 |             "image" => $this->uploadImage()
84 |         );
85 |
86 |         $this->db->insert("workbook", $data);
87 |     }
88 | }
```

Gambar 32 Models Workbook_model.php

Jika sudah selesai, maka langkah selanjutnya adalah membuat function update, dimana data yang dimasukkan akan diupdate berdasarkan id yang sedang digunakan.

```
89 |     public function update($id, $data)
90 |     {
91 |         $this->db->where('workbook_id', $id);
92 |         $this->db->update("workbook", $data);
93 |     }
94 | }
```

Gambar 33 Models Workbook_model.php

Jika sudah maka langkah selanjutnya adalah membuat function delete yang akan menghapus data yang dipilih dan akan dihapus sesuai dengan id-nya. Selanjutnya adalah membuat function _uploadImage(). Lalu kita memberikan *config* tentang *upload* gambar. Disimpan di direktori mana, lalu jenis ekstensi apa saja yang dibolehkan, lalu besar *file size* yang diperbolehkan. Setelahnya baru *load library upload* dan menggunakan *config* yang baru saja diberikan. Jika sedang melakukan *upload* dengan nama *file* aslinya, maka replace nama *file* menjadi default.jpg.

Langkah terakhir dalam membuat workbook_model.php adalah membuat function _deleteImage() yang akan dihapus sesuai dengan id yang dipilih, lalu hapus data yang ada di dalam tabel workbook di *database* seperti yang ditunjukkan oleh gambar 34 di bawah ini.

```

public function delete($id)
{
    $this->_deleteImage($id);
    return $this->db->delete("workbook", array("workbook_id" => $id));
}

private function _uploadImage()
{
    $config['upload_path']          = './upload/workbook/';
    $config['allowed_types']        = 'gif|jpg|png';
    $config['file_name']            = $this->workbook_id;
    $config['overwrite']            = true;
    $config['max_size']             = 10240; // 10MB
    // $config['max_width']          = 1024;
    // $config['max_height']         = 768;

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

    if ($this->upload->do_upload('image')) {
        return $this->upload->data("file_name");
    }

    return "default.jpg";
}

private function _deleteImage($id)
{
    $workbook = $this->getId($id);
    if ($workbook->image != "default.jpg") {
        $filename = explode(".", $workbook->image)[0];
        return array_map('unlink', glob(FCPATH."upload/workbook/$filename.*"));
    }
}

```

Gambar 34 Models Workbook_model.php

Jika models dan controller telah kita buat, maka langkah selanjutnya adalah membuat view. Untuk view silakan mengunduh *bootstrap* sesuai dengan keinginan masing masing, disini saya masih menggunakan argon *dashboard bootstrap* dikarenakan *bootstrap* yang saya gunakan memiliki keseluruhan halaman yang dibutuhkan. Sehingga hanya diperlukan sedikit modifikasi dari argon *dashboard*-nya.

Buatlah folder admin di dalam folder views, lalu jika sudah membuat folder admin selanjutnya buat kembali sebuah folder dengan nama workbook. Nah pada folder workbook kita selanjutnya membuat list_workbook.php. Mengapa banyak sekali membuat folder pada folder views? Ini dimaksudkan untuk merapikan direktori kerja kita, selain itu dalam penggerjaan aplikasi akan menjadi lebih efisien. Sehingga direktori dari beberapa folder yang telah dibuat akan terlihat menjadi seperti ini C://xampp/htdocs/eppm-go!/application/views/admin/workbook.

Ingat! Hanya ubah beberapa komponen di dalam *bootstrap* sesuai dengan yang kita perlukan. Jangan mengubah semua komponen, karena akan berakibat *layout* yang telah diatur menjadi berantakan. Lalu pada file list_workbook.php jangan lupa tambahkan *link* atau URL (*Uniform Resource Locator*) menuju menu menu yang telah atau akan dibuat. *Link* akan ada di setiap halaman yang kita buat, jadi pastikan jangan lupa menaruh *link* pada tiap tiap halaman yang dibuat Seperti yang ditunjukkan oleh gambar 35 di bawah ini

```

<ul class="navbar-nav">
- 

```

Gambar 35 Link/URL Menu

Setelah membuat *link* atau URL selanjutnya kita akan membuat tabel untuk menampilkan data yang kita punya. Pertama buat terlebih dahulu kolom untuk nama *field*-nya.

```

<table class="table align-items-center table-flush">
    <thead class="thead-light">
        <tr>
            <th scope="col">ID Stage</th>
            <th scope="col">ID Modul</th>
            <th scope="col">Nama Pekerja</th>
            <th scope="col">Progress</th>
            <th scope="col">Status</th>
            <th scope="col">Lembar Pengesahan</th>
            <th scope="col"></th>
        </tr>
    </thead>
    <tbody>

```

Gambar 36 Inisiasi Tabel Workbook

Setelah membuat kolom untuk nama *field*-nya maka kita selanjutnya harus membuat kolom untuk menempatkan atau mewadahi data yang akan ditampilkan. Agar memudahkan pembuatan tabel, kita akan menggunakan DataTables yang berguna untuk membuat tabel secara instan hanya dengan menggunakan sintaks jQuery yang masih merupakan *library* untuk Javascript.

```
<tbody>
    <?php foreach ($workbook as $workbook): ?>
    <tr>
        <td width="150">
            <?php echo $workbook->stage_id ?>
        </td>
        <td>
            <?php echo $workbook->modul_id ?>
        </td>
        <td>
            <?php echo $workbook->yaww ?>
        </td>
        <td>
            <?php echo $workbook->progress ?>
        </td>
        <td>
            <?php echo $workbook->status ?>
        </td>
        <td>
            
        </td>
        <td width="250">
            <a href="<?php echo site_url('admin/workbook/edit/'.$workbook->workbook_id)?>">
                <i class="fas fa-edit"></i>Update</a>
            <a onclick="deleteConfirm('<?php echo site_url('admin/workbook/delete/'.$workbook->workbook_id) ?>')"
               href="#" class="btn btn-small text-danger"><i class="fas fa-trash"></i>Hapus</a>
            </td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
```

Gambar 37 Views list_workbook.php

Disini kita menggunakan perulangan *foreach* yang mengambil data dari *models* *workbook* dan diinisiasikan sebagai variabel \$workbook. Lalu memanggil *field* yang diperlukan ke dalam tabelnya. Karena ada perulangan *foreach* maka setiap variabel yang diinisiasikan akan diulang secara terus

menerus hingga semua datanya terpanggil. Lalu kita membuat sebuah tombol untuk melakukan perubahan atau *update* dengan mengambil id dari data yang kita pilih, begitupun dengan tombol hapus sama seperti tombol *update*. Dan jangan lupa kita akhiri perulangan *foreach*-nya.

Setelah itu, untuk sintaks jQuery dari DataTables kita taruh dibawah setelah sintaks pembuatan tabel. Jangan lupa untuk menyisipkan tag <script> karena masih satu bagian dari JavaScript. Selain itu, karena kita menggunakan *resources* dari luar ada beberapa cara yang bisa digunakan, yaitu dengan cara:

1. Mengunduh *library*-nya dan disimpan pada folder *assets* yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini akan menggunakan *local resource* dari PC atau laptop kita.
2. Menyalin *link CDN* (*Content Delivery Network*) yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini hanya memerlukan akses internet. *Link CDN* DataTables untuk CSS adalah //cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css dan *link CDN* DataTables untuk JavaScript adalah //cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js

```
<script>$(document).ready( function () {
|   $('table').DataTable();
}
);
</script>
```

Gambar 38 Sintaks DataTables

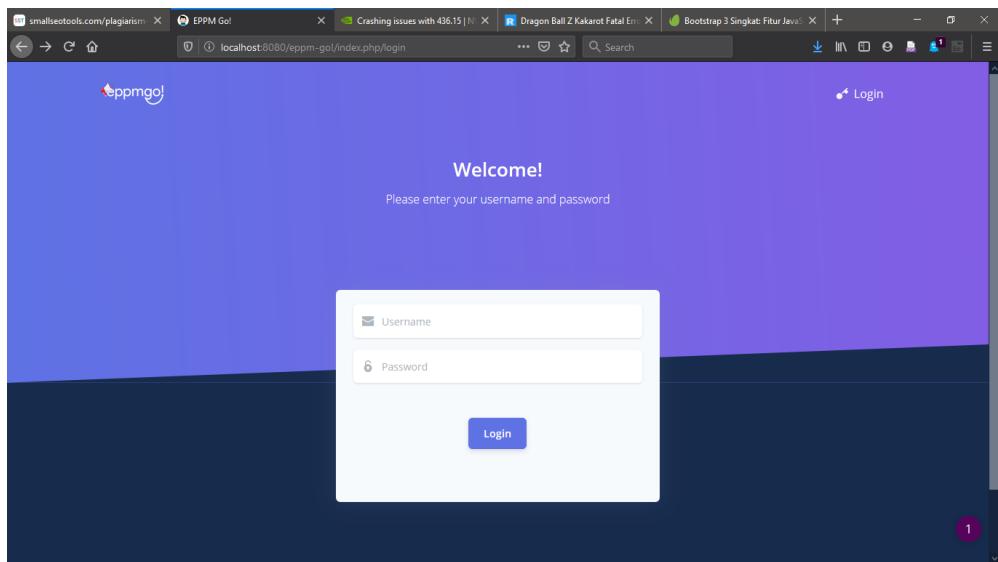
Lalu kita akan menggunakan modals yang juga masih bagian dari JavaScript, yang akan digunakan untuk tombol hapus. Modals adalah sebuah kotak dialog kecil yang biasanya muncul di atas halaman yang sedang kita buka. Sehingga apabila tombol hapus tidak sengaja ditekan maka akan menampilkan dialog konfirmasi untuk menghapus sebuah data, seperti yang ditunjukkan oleh gambar 39 berikut.

```
<script>
function deleteConfirm(url){
    $('#btn-delete').attr('href', url);
    $('#deleteModal').modal();
}
</script>
```

Gambar 39 Modal Tombol Hapus

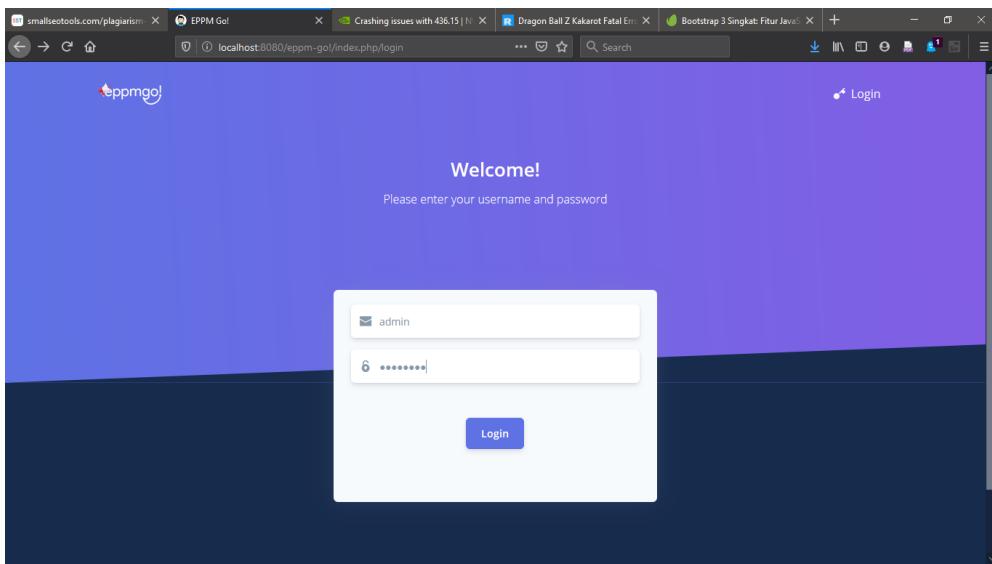
Jika controller, models, dan views telah kita buat maka langkah selanjutnya adalah mencoba untuk memanggil halaman web yang baru saja kita buat. Apakah terdapat error saat dibuka atau kan sudah sesuai dengan yang kita inginkan? Untuk itu silakan coba membuka *browser* atau peramban sesuai dengan preferensi kita. Dan coba ketikkan localhost:8080/eppm-go!/index.php/login.

Untuk pengaturan localhost masing masing pengguna pasti akan berbeda beda, dikarenakan *port* yang digunakan untuk mengakses *localhost*-nya. Karena saya menggunakan *port* 8080 untuk mengakses *localhost* makanya ada tambahan nama *port* di samping *localhost*.



Gambar 40 Halaman Login

Berhasil masuk ke halaman *login*, maka kita harus mengisi *username* dan *password* agar bisa mengakses halaman berikutnya. Karena halaman *workbook* yang baru saja kita buat adalah halaman untuk admin maka kita akan menggunakan *username* admin dan juga *password* admin123. Karena admin yang mengakses maka semua menu dan fitur fitur yang disediakan dapat diakses oleh admin. Sedangkan untuk pengguna akan mendapatkan limitasi terhadap menu dan fitur yang dapat digunakan.



Gambar 41 Halaman Login

A screenshot of a web browser showing the EPPM Go! Workbook list page. The left sidebar has a 'Workbook' section selected. The main area is titled 'Workbook' and contains a table with several rows of data. Each row represents a workbook entry with columns for 'PROGRESS', 'STATUS', and 'LEMBAR PENGESAHAN'. To the right of each row are 'Update' and 'Hapus' buttons. The table includes entries for 'Sudah mengerjakan modul 6.1 tinggal ttd coach', 'Sudah selesai semuanyaaaaa', 'Izwan Aristyo', 'Sudah di acc', 'Sudah selesai nih workbook modul ini', 'Sudah selesai', 'Sudah selesai untuk Level of Owner Integration', and 'asd'. Each row also has a small profile picture icon.

Gambar 42 Halaman List Workbook

Setelah halaman *workbook* selesai dibuat, maka selanjutnya adalah membuat form *update* untuk dapat mengubah beberapa data yang kita masukkan sebelumnya. Untuk form *update* baru akan muncul setelah mengklik tombol *update*, maka langsung saja kita buat sebuah *file* dengan nama edit_form.php pada direktori yang sama yaitu C://xampp/htdocs/eppm-go!/application/views/admin/workbook.

```
<div class="card mb-3">
  <div class="card-header">
    <a href="=site_url('admin/workbook/')?><i class="fas fa-arrow-left"></i>
    Kembali</a>
  </div>
  <div class="card-body">

    <form action="=base_url();?&gt;index.php/admin/workbook/aksi_edit" method="post" enctype="multipart/form-data"&gt;

      &lt;input type="hidden" name="workbook_id" value="<?=echo $workbook-&gt;workbook_id?&gt;" /&gt;

      &lt;div class="form-group"&gt;
        &lt;label for="name"&gt;Stage&lt;/label&gt;
        &lt;input class="form-control" type="text" name="stage_id" placeholder="Stage" value="<?=echo $workbook-&gt;stage_id?&gt;" readonly&gt;
        &lt;div class="invalid-feedback"&gt;
          &lt;?php echo form_error('stage') ?&gt;
        &lt;/div&gt;
      &lt;/div&gt;

      &lt;div class="form-group"&gt;
        &lt;label for="name"&gt;Modul&lt;/label&gt;
        &lt;input class="form-control" type="text" name="modul_id" min="0" placeholder="Modul" value="<?=echo $workbook-&gt;modul_id?&gt;" readonly&gt;
        &lt;div class="invalid-feedback"&gt;
          &lt;?php echo form_error('modul_id') ?&gt;
        &lt;/div&gt;
      &lt;/div&gt;</pre
```

Gambar 43 Form edit_form.php

Sesuai dengan yang ditunjukkan oleh gambar 43, disini kita membuat sebuah form menggunakan sintaks bawaan yang disediakan oleh *bootstrap argon dashboard*. Pada tiap *textbox* pada form, kita memanggil kembali data data yang akan diubah sesuai dengan id yang dipilih, sama seperti yang kita lakukan sebelumnya pada halaman list_workbook.php

```

<div class="form-group">
    <label for="name">Nama Pekerja</label>
    <input class="form-control" type="text" name="employee_name" min="0" placeholder="Nama Pekerja" value=<?php echo $workbook->yaww ?>
        <?php echo form_error('employee_name') ?>
    <div class="invalid-feedback">
        <?php echo form_error('employee_name') ?>
    </div>
</div>

<div class="form-group">
    <label for="name">Progress*</label>
    <input class="form-control" type="text" name="progress" min="0" placeholder="Progress" value=<?php echo $workbook->progress ?>" required
        <?php echo form_error('progress') ?>
    <div class="invalid-feedback">
        <?php echo form_error('progress') ?>
    </div>
</div>

<div class="form-group">
    <label for="name">Status*</label>
    <select class="form-control" name="status">
        <?php foreach ( $status as $status ) : ?>
        <?php if ( $status == $workbook->status ) : ?>
            <option value=<?php echo $status; ?>><?php echo $status; ?></option>
        <?php else : ?>
            <option value=<?php echo $status; ?>><?php echo $status; ?></option>
        <?php endif; ?>
        <?php endforeach; ?>
    </select>
    <div class="invalid-feedback">
        <?php echo form_error('status') ?>
    </div>
</div>

```

Gambar 44 Form edit_form.php

Dapat dilihat pada gambar 43 sampai gambar 44, pada *textbox stage*, modul, nama pekerja, dan juga progres hanya memanggil data berdasarkan variabel yang telah kita definisikan sebelumnya pada models. Sedangkan untuk status, pada gambar dicoba untuk *foreach* untuk menampilkan data status akan dipilih dalam bentuk *combo box*. Selanjutnya untuk *textbox* lembar pengesahan sama saja dengan *textbox* lainnya hanya memanggil data berdasarkan variabel yang telah didefinisikan sebelumnya. Lalu ada tombol simpan untuk menyimpan perubahan perubahan yang dilakukan di form ini ke dalam *database*. Seperti yang ditunjukkan oleh gambar 45 berikut ini.

```
<div class="form-group">
    <label for="name">Lembar Pengesahan</label>
    <input class="form-control-file" type="file" name="image" />
    <input type="hidden" name="old_image" value="<?php echo $workbook->image ?>" />
    <div class="invalid-feedback">
        <?php echo form_error('image') ?>
    </div>
</div>
</div>

<input class="btn btn-success" type="submit" name="btn" value="Simpan" />
</form>
```

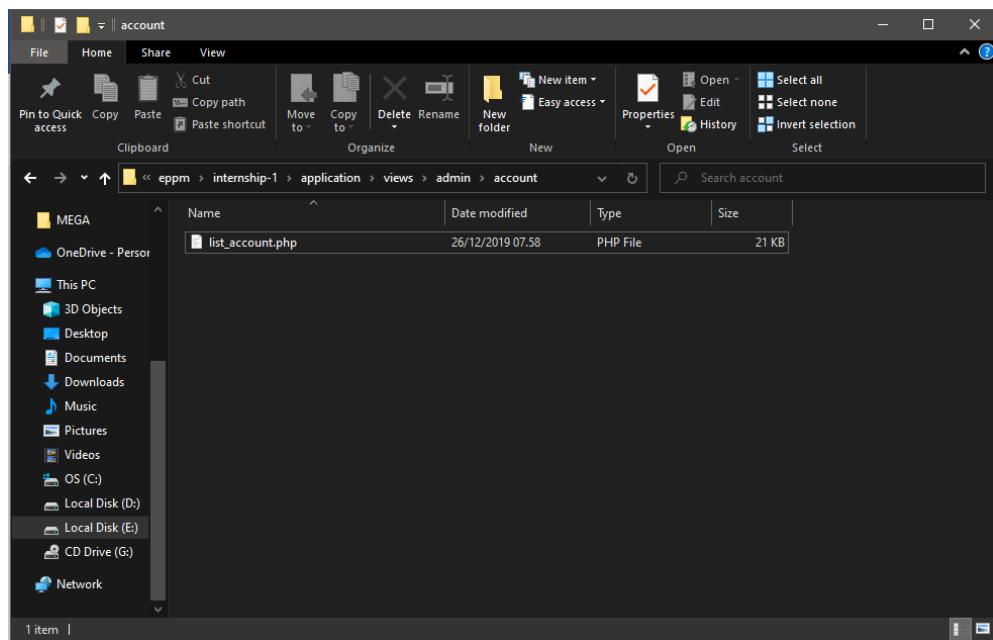
Gambar 45 Form edit_form.php

Dengan ini, selesailah sudah halaman *workbook* untuk admin. Mengapa tidak ada form untuk meng-*input* data *workbook* yang baru? Karena hal itu dilakukan untuk pengguna atau *user*. Jadi admin hanya untuk memantau apakah sudah melaksanakan *workbook* atau belum, nah untuk form tersebut baru akan kita buat pada tahapan *workbook* untuk *user* pada beberapa sub bab berikutnya. Lanjut terus ya bacanya!

2.4. CRUD Menu Daftar Pekerja

Untuk CRUD menu daftar pekerja, ini sebenarnya sama saja seperti CRUD menu *workbook*. Ada beberapa hal yang sama diulangi dan ada beberapa hal yang diganti menyesuaikan dengan kebutuhan pada menu yang akan dibuat. Maka langsung saja kita buat folder *account* pada direktori controller seperti berikut C://xampp/htdocs/eppm-go!/application/controller/admin/account.

Pada folder *account* buatlah sebuah *file* controller dengan nama Account.php lalu isikan di dalam *file* Account.php seperti yang ditunjukkan pada gambar 46 berikut ini.



Gambar 46 Controller Account.php

Seperti pada *file* controller Workbook.php maka kita terlebih dahulu membuat sebuah class dengan nama Account dengan menggunakan *library* CI_Controller. Lalu membuat function __construct() yang akan me-load model account_model dan juga me-load *library* form validasi. Karena sudah me-load model, maka kita gunakan function is_role() pada account_model yang akan mengecek apabila *role* yang masuk ke dalam aplikasi bukan admin maka akan diarahkan kembali ke halaman *login*.

Lalu setelah itu kita membuat function index() yang medefinisikan variabel \$data yang datanya diambil melalui function getAll() pada account_model. Jika sudah maka akan menampilkan halaman list_account dengan membawa data yang sudah didefinisikan pada variabel \$data sebelumnya. Seperti yang terlihat pada gambar 47 berikut ini.

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

class Account extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("account_model"); //Load model Account
        $this->load->library('form_validation'); //Load Library form validation
        if($this->account_model->is_role() != "admin"){
            redirect("index.php/login");
        }
    }

    public function index()
    {
        $data["account"] = $this->account_model->getAll(); //ambil data dari model
        $this->load->view("admin/account/list_account", $data); //load view data model ke account
    }
}
```

Gambar 47 Controller Account.php

Jika sudah maka kita lanjut untuk membuat function add() dengan mendefinisikan variabel \$account pada account_model dan juga variabel \$validation pada form validasi. Lalu variabel \$validation tersebut kita terapkan *rules*. Setelahnya kita tampilkan halaman tambah data atau form tambah data.

```
2 |     public function add()
3 |     {
4 |         $account = $this->account_model; //objek model
5 |         $validation = $this->form_validation; //objek form validation
6 |         $validation->set_rules($account->rules()); //terapkan rules
7 |
8 |         if ($validation->run()){
9 |             $account->save();
10 |             $this->session->set_flashdata('Sukses', 'Data Anda Berhasil Disimpan'); //pesan berhasil
11 |         }
12 |
13 |         $this->load->view("admin/account/new_account"); //load isi form account
14 |     }
```

Gambar 48 Controller Account.php

Setelahnya kita tambahkan function edit() yang akan mengambil datanya berdasarkan id yang dipilih, lalu akan ditampilkan kembali halaman form edit dengan data dari id yang sudah dipilih.

```
public function edit($id=null)
{
    if(!isset($id)) redirect('admin/account/list_account');

    $account = $this->account_model;
    $validation = $this->form_validation;
    $validation->set_rules($account->rules());

    if ($validation->run()){
        $account->update();
        $this->session->set_flashdata('success', 'Data Anda Berhasil Diupdate'); //pesan berhasil
    }

    $data["account"] = $account->getById($id); //mengambil data berdasarkan id
    if (!$data["account"]) show_404(); // jika tidak ada show error
    $this->load->view("admin/account/edit_account", $data); //Load edit form account
}
```

Gambar 49 Controller Account.php

Jangan lupa untuk membuat function delete() dengan mengambil data berdasarkan id yang kita pilih, sehingga hanya data yang dipilih saja yang terhapus. Jika sudah berhasil terhapus maka akan diarahkan kembali ke halaman daftar pekerja.

```
public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->account_model->delete($id)){
        redirect('index.php/admin/account');
    }
}
```

Gambar 50 Controller Account.php

Dengan ini, kita telah selesai membuat *file* controller Account.php. Maka langkah selanjutnya adalah kita haruss membuat *file* models account_model.php pada direktori models di localhost kita. Setelah selesai membuat *file* tersebut langsung saja kita isikan *file*-nya dengan isian seperti yang ditunjukkan pada gambar 51 berikut.

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

class Account_model extends CI_Model
{
    private $_table = "account";

    public $employee_id;
    public $directorate;
    public $employee_name;
    public $username;
    public $password;
    public $user_email;
    public $role;

    public function rules()
    {
        return [
            ['field' => 'employee_id',
            'label' => 'No. Pekerja',
            'rules' => 'required'],

            ['field' => 'employee_name',
            'label' => 'Nama Pekerja',
            'rules' => 'required'],

            ['field' => 'username',
            'label' => 'username',
            'rules' => 'required'],

            ['field' => 'password',
            'label' => 'password',
            'rules' => 'required']
        ];
    }
}
```

Gambar 51 Models Account_model.php

Seperti yang ditunjukkan gambar 51 di atas, terlebih dahulu kita harus mendefinisikan variabel yang akan digunakan untuk menu daftar pengguna. Jika sudah didefinisikan, maka lanjut dengan membuat function rules() dimana ini merupakan aturan aturan yang akan kita gunakan untuk menu ini. Dapat dilihat hanya ada 4 *field* yang diberi aturan tidak boleh kosong alias wajib diisi!.

Jika sudah selesai, maka kita selanjutnya membuat function is_role() yang akan berfungsi untuk mengecek *role* yang masuk ke dalam aplikasi berdasarkan *session* yang aktif. Lalu kita juga membuat function getAll() yang berfungsi untuk mengambil semua data pada tabel account. Nah sama seperti function getAll() kita juga membuat function getById() yang berfungsi untuk mengambil data berdasarkan id-nya. Disini employee_id dijadikan *identifier* dari id yang akan diambil. Untuk lebih lengkapnya silakan lihat gambar 52 di bawah ini.

```
//fungsi cek level
function is_role()
{
    return $this->session->userdata('role');
}

public function getAll()
{
    return $this->db->get_where($this->_table, array('role' => 'user'))->result();
}

public function getById($id)
{
    return $this->db->get_where($this->_table, ["employee_id" => $id])->row();
}
```

Gambar 52 Models Account_model.php

Langkah selanjutnya adalah membuat function save() yang berfungsi untuk menyimpan data yang telah kita *input* kan ke dalam *database* pada tabel account. Begitupun dengan function edit() yang dibuat setelah function save() selesai, function edit() akan menampilkan data yang telah disimpan sebelumnya berdasarkan id yang dipilih. Lalu apabila telah mengubah data yang diperlukan maka akan langsung meng-*update* data tersebut sesuai dengan id yang dipilih.

Sama seperti function edit(), function delete() juga mengambil data berdasarkan id yang dipilih akan tetapi peruntukannya berbeda. Jika edit() untuk mengubah data, maka delete() untuk menghapus data yang kita pilih. Untuk lengkapnya dapat dilihat pada gambar 53 di bawah ini.

```
public function save()
{
    $post = $this->input->post();
    $this->employee_id = $post["employee_id"];
    $this->directorate = $post["directorate"];
    $this->employee_name = $post["employee_name"];
    $this->username = $post["username"];
    $this->password = $post["password"];
    $this->user_email = $post["user_email"];
    $this->role = $post["role"];
    $this->db->insert($this->_table, $this);
}

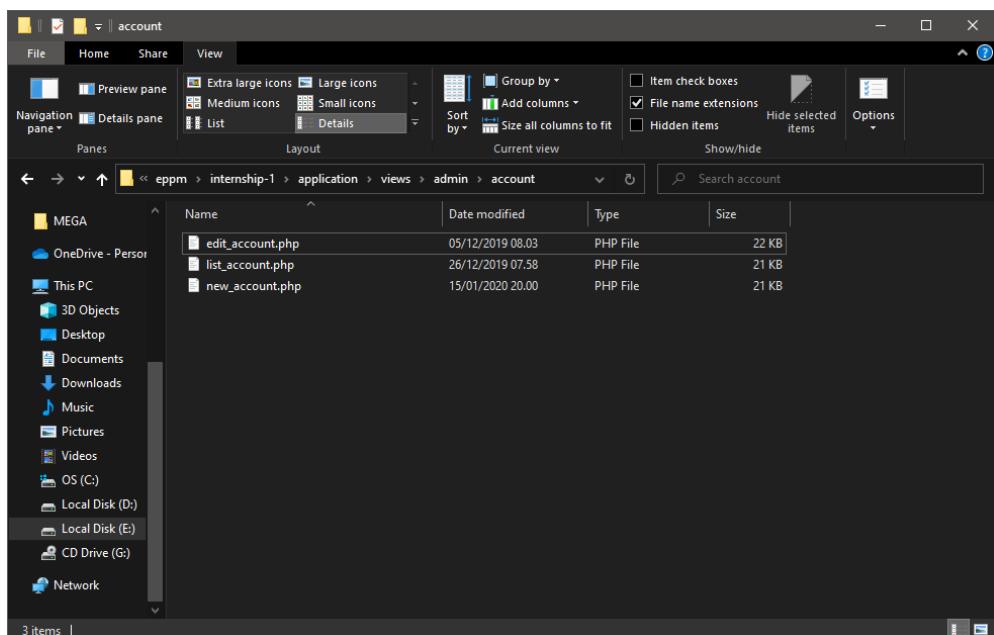
public function update()
{
    $post = $this->input->post();
    $this->employee_id = $post["employee_id"];
    $this->directorate = $post["directorate"];
    $this->employee_name = $post["employee_name"];
    $this->username = $post["username"];
    $this->password = $post["password"];
    $this->user_email = $post["user_email"];
    $this->role = $post["role"];
    $this->db->update($this->_table, $this, array('employee_id' => $post['employee_id']));
}

public function delete($id)
{
    return $this->db->delete($this->_table, array("employee_id" => $id));
}
```

Gambar 53 Models Account_model.php

Mantap! Selesai sudah tahapan-tahapan yang diperlukan untuk membuat file models Account_model.php. Nah bila temen-temen yang lain sudah maka bisa langsung dilanjutkan untuk membuat file views-nya yang berfungsi untuk menampilkan datanya ke dalam sebuah *user interface* yang lebih nyaman untuk dilihat oleh pengguna. Untuk itu buat folder account pada direktori views dan admin, dan buat kembali sebuah file dengan nama list_account.php, edit_account.php dan juga new_account.php.

Sekarang fokusnya akan kita alihkan hanya kepada file list_account.php yang berguna untuk menampilkan seluruh data yang ada pada database di dalam tabel account.



Gambar 54 File Account

Jangan lupa, untuk setiap menu dan halaman yang akan dibuat kita harus selalu menaruh *link* atau URL yang akan mengarah pada 5 menu utama yang akan dibuat seperti yang ditunjukkan pada gambar 55 di bawah ini.

```
<ul class="navbar-nav">
- <a class=" nav-link " href="<?php echo base_url(); ?>index.php/admin" ><i class="ni ni-tv-2 text-primary"></i> Das
| </a>
</li>
- <a class="nav-link " href="<?php echo base_url(); ?>admin/workbook">
| <i class="ni ni-bullet-list-67 text-red"></i> Workbook
</a>
</li>

```

Gambar 55 Views Link atau URL

Lanjut lagi, kita membuat sebuah tabel dengan terlebih dahulu membuat kolom untuk nama *field*-nya seperti yang ditunjukkan oleh gambar 56 di bawah ini.

```
<div class="row">
<div class="col">
<div class="card shadow">
<div class="card-header border-0">
<h3 class="mb-0">Daftar Peserta</h3>
<a href="<?php echo site_url('admin/account/add') ?>"><i class="fas fa-plus"></i> Tambah Baru</a>
</div>
<div class="table-responsive">
<table id="table" class="table align-items-center table-flush" >
<thead class="thead-light">
<tr>
<th scope="col">Nomor Pekerja</th>
<th scope="col">Direktorat</th>
<th scope="col">Nama Pekerja</th>
<th scope="col">Username</th>
<!-- <th scope="col">Password</th> -->
<th scope="col">Email</th>
<!-- <th scope="col">Role</th> -->
<th scope="col"></th>
</tr>
</thead>
```

Gambar 56 Views list_account.php

Setelahnya baru kita masukkan datanya dengan cara memanggil variabel yang telah didefinisikan sebelumnya pada bagian models. Lalu kita melakukan perulangan *foreach* untuk mengulang pemanggilan data sampai datanya telah terpanggil semua, sehingga tidak hanya satu data saja yang dipanggil.

Lalu jangan lupa buat juga tombol *update* dan juga *hapus*, dimana tombol *update* akan mengambil id dari data yang dipilih dan akan langsung diarahkan menuju form *update*. Sedangkan untuk tombol *hapus* akan mengambil id dari data yang dipilih untuk kita hapus. Oh iya pada menu daftar pekerja disini juga menggunakan DataTables, jadi cara penggunaannya masih sama seperti yang telah kita lakukan di menu workbook. Untuk lebih jelasnya lihat gambar 57 berikut ini.

```
<tbody>
<?php foreach ($account as $account): ?>
<tr>
    <td width="150">
        <?php echo $account->employee_id ?>
    </td>
    <td width="150">
        <?php echo $account->directorate ?>
    </td>
    <td>
        <?php echo $account->employee_name ?>
    </td>
    <td width="150">
        <?php echo $account->username ?>
    </td>
    <!-- <td width="150">
        <?php echo $account->password ?>
    </td> -->
    <td>
        <?php echo $account->user_email ?>
    </td>
    <!-- <td>
        <?php echo $account->role ?>
    </td> -->
    <td width="250">
        <a href=<?php echo site_url('admin/account/edit/'.$account->employee_id)?>">
            class="btn btn-small"><i class="fas fa-edit"></i>Update</a>
        <a onclick="deleteConfirm('<?php echo site_url('admin/account/delete/'.$account->employee_id) ?>')>
            href="#" class="btn btn-small text-danger"><i class="fas fa-trash"></i>Hapus</a>
        </td>
    </tr>
<?php endforeach; ?>
</tbody>
</table>
```

Gambar 57 Views list_account.php

Seperti yang sudah dijelaskan barusan, kita masih menggunakan DataTables jadi kita harus menggunakan sintaks jQuery untuk menggunakan *resources* dari DataTables. Setelah itu, untuk sintaks jQuery dari DataTables kita taruh dibawah setelah sintaks pembuatan tabel. Jangan lupa untuk menyisipkan tag <script> karena masih satu bagian dari JavaScript. Selain itu, karena kita menggunakan *resources* dari luar ada beberapa cara yang bisa digunakan, yaitu dengan cara:

1. Mengunduh *library*-nya dan disimpan pada folder *assets* yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini akan menggunakan *local resource* dari PC atau laptop kita.
2. Menyalin *link CDN (Content Delivery Network)* yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini hanya memerlukan akses internet. *Link CDN* DataTables untuk CSS adalah //cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css dan *link CDN* DataTables untuk JavaScript adalah //cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js

```
</div>
<script>$(document).ready( function () {
    $('table').DataTable();
} );
</script>
</div>
```

Gambar 58 Views list_account.php

Jangan lupa gunakan modal untuk konfirmasi sebelum dilakukannya penghapusan sebuah data, seperti yang ditunjukkan pada gambar 59 berikut ini.

```
<script>
function deleteConfirm(url){
    $('#btn-delete').attr('href', url);
    $('#deleteModal').modal();
}
</script>
```

Gambar 59 Views list_account.php

Karena untuk list sudah selesai, maka kita akan lanjutkan kembali ke halaman form tambah data atau form *input* karena sudah dibuat sebelumnya *file* new_account.php maka langsung saja diberi isian seperti yang akan ditunjukkan oleh gambar 60, dan 62 berikut.

```

<div class="card mb-3">
    <div class="card-header">
        <a href="=php echo site_url('admin/account/') ?&gt;"&gt;&lt;i class="fas fa-arrow-left"&gt;&lt;/i&gt; Back&lt;/a&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;
        &lt;form action="<?=php base_url('admin/account/add') ?&gt;" method="post" enctype="multipart/form-data"&gt;

            &lt;!-- &lt;input type="hidden" name="account_id" value="<?=php echo $account-&gt;employee_id?&gt;" /&gt; --&gt;

            &lt;div class="form-group"&gt;
                &lt;label for="name"&gt;No. Pekerja*&lt;/label&gt;
                &lt;input class="form-control" type="text" name="employee_id" placeholder="Nomor Pekerja" /&gt;
                &lt;div class="invalid-feedback"&gt;&lt;?php echo form_error('employee_id')?&gt;
            &lt;/div&gt;
        &lt;/div&gt;

        &lt;div class="form-group"&gt;
            &lt;label for="name"&gt;Direktorat&lt;/label&gt;
            &lt;input class="form-control" type="text" name="directorate" placeholder="Direktorat" /&gt;
            &lt;div class="invalid-feedback"&gt;&lt;?php echo form_error('directorate')?&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label for="name"&gt;Nama Pekerja*&lt;/label&gt;
        &lt;input class="form-control" type="text" name="employee_name" placeholder="Nama Pekerja" /&gt;
        &lt;div class="invalid-feedback"&gt;&lt;?php echo form_error('employee_name')?&gt;
    &lt;/div&gt;
&lt;/div&gt;
</pre

```

Gambar 60 Views new_account.php

```

<div class="form-group">
    <label for="name">Username*</label>
    <input class="form-control" type="text" name="username" placeholder="Username" />
    <div class="invalid-feedback"><?php echo form_error('username')?>
</div>
</div>

<div class="form-group">
    <label for="name">Password*</label>
    <input class="form-control" type="password" name="password" placeholder="Password" />
    <div class="invalid-feedback"><?php echo form_error('modul_id')?>
</div>
</div>

<div class="form-group">
    <label for="name">Email</label>
    <input class="form-control" type="text" name="user_email" placeholder="Status" />
    <div class="invalid-feedback"><?php echo form_error('user_email')?>
</div>
</div>

<div class="form-group">
    <label for="name">Role</label>
    <select class="form-control" name="role">
        <option value="admin">Admin</option>
        <option value="user">User</option>
    </select>
    <div class="invalid-feedback"><?php echo form_error('role')?>
</div>
</div>

```

Gambar 61 Views new_account.php

Dengan ini *file* new_account.php sudah selesai dibuat. Maka langkah selanjutnya adalah mengisi *file* edit_account.php yang sebelumnya telah kita buat. Langsung saja *file* tersebut kita isi dengan isian seperti yang ditunjukkan pada gambar 62 di bawah ini.



```
<div class="card mb-3">
  <div class="card-header">
    <a href="=php echo site_url('admin/account/') ?&gt;"&gt;&lt;i class="fas fa-arrow-left"&gt;&lt;/i&gt;Kembali&lt;/a&gt;
  &lt;/div&gt;
  &lt;div class="card-body"&gt;

    &lt;form action="<?=php base_url('admin/account/edit') ?&gt;" method="post" enctype="multipart/form-data"&gt;
      &lt;!-- &lt;input type="hidden" name="account_id" value="<?=php echo $account-&gt;employee_id?&gt;" /&gt; --&gt;

      &lt;div class="form-group"&gt;
        &lt;label for="name"&gt;No. Pekerja*&lt;/label&gt;
        &lt;input class="form-control" type="text" name="employee_id" placeholder="Nomor Pekerja" value="<?=php echo $account-&gt;employee_id?&gt;"&gt;
        &lt;div class="invalid-feedback"&gt;<?=php echo form_error('employee_id')?&gt;
      &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
      &lt;label for="name"&gt;Direktorat&lt;/label&gt;
      &lt;input class="form-control" type="text" name="directorate" placeholder="Direktorat" value="<?=php echo $account-&gt;directorate?&gt;"&gt;
      &lt;div class="invalid-feedback"&gt;<?=php echo form_error('directorate')?&gt;
    &lt;/div&gt;
  &lt;/div&gt;
</pre
```

Gambar 62 Views edit_account.php

Sesuai dengan gambar 62 di atas, kita membuat sebuah *textbox* sesuai dengan *field* yang kita butuhkan. Lalu ambil datanya dengan memanggil variabel yang telah didefinisikan pada models Account_model.php. Jangan lupa untuk memanggil datanya sesuai dengan id yang dipilih, maka panggil variabel employee_id pada models melalui inisiasi variabel \$account di views. Untuk *textbox* selanjutnya akan sama saja seperti yang kita lakukan untuk *textbox* No. Pekerja.

```

<div class="form-group">
    <label for="name">Nama Pekerja*</label>
    <input class="form-control" type="text" name="employee_name" placeholder="Nama Pekerja" value=<?php echo $account->employee_na
    <div class="invalid-feedback"><?php echo form_error('employee_name')?>
    </div>
</div>

<div class="form-group">
    <label for="name">Username*</label>
    <input class="form-control" type="text" name="username" placeholder="Username" value=<?php echo $account->username?>"/>
    <div class="invalid-feedback"><?php echo form_error('username')?>
    </div>
</div>

<div class="form-group">
    <label for="name">Password*</label>
    <input class="form-control" type="password" name="password" placeholder="Password" value=<?php echo $account->password?>"/>
    <div class="invalid-feedback"><?php echo form_error('password')?>
    </div>
</div>

<div class="form-group">
    <label for="name">Email</label>
    <input class="form-control" type="text" name="user_email" placeholder="Email" value=<?php echo $account->user_email?> "/>
    <div class="invalid-feedback"><?php echo form_error('user_email')?>
    </div>
</div>

```

Gambar 63 Views edit_account.php

Sama saja seperti yang telah kita lakukan, maka tinggal kita ikuti saja langkah – langkah di atas. Lalu apabila ada salah satu *textbox* yang harus diisi, karena sudah menerapkan *rules* pada models kita tinggal menggunakananya saja.

Jika *file* edit_account.php sudah selesai dibuat, maka selesai sudah tahapan yang harus dijalani untuk membuat menu daftar pekerja. Untuk itu kita langsung jalankan pada browser halaman yang baru saja kita buat, dengan memanggil *link* seperti ini localhost:8080/eppm-go!/admin/account. Lalu jangan lupa saat kita membuka *link* di atas, akan diarahkan pertama kali untuk melakukan *login* maka tinggal masukkan *username* dan *password* yang sama

untuk admin. Setelahnya kita langsung menuju halaman daftar pekerja yang akan terlihat seperti yang ditunjukkan pada gambar 64 di bawah ini

NAMA PEKERJA	USERNAME	EMAIL	Update	Hapus	
User	user	user@email.com	[Update]	[Hapus]	
pekerja	pekerja	pekerja@email.com	[Update]	[Hapus]	
Budi Jaya	budi	budijaya@gmail.com	[Update]	[Hapus]	
Rizal	rizal	rizal	[Update]	[Hapus]	
Riandaka Rizal	riandakar	riandakarizal@gmail.com	[Update]	[Hapus]	
Petrokimia	Bambang Purwito H.	bambangpurwi	bampur@pertamina.com	[Update]	[Hapus]
Petrokimia	Widodo Tri Rahardjo	widodotri	widodo.rahardjo@pertamina.com	[Update]	[Hapus]

Gambar 64 Halaman Daftar Pekerja

Saat kita mengklik menu daftar pekerja, maka yang akan pertama kali ditampilkan adalah tabel dari data yang telah kita masukkan sebelumnya. Lalu kita coba untuk menambahkan data baru ke dalam tabel tersebut, dengan mengklik tombol tambah baru di pojok kiri atas yang akan mengarahkan ke halaman form tambah baru atau form *input* seperti yang ditunjukkan pada gambar 65 di bawah ini.

The screenshot shows a web-based application interface for 'EPPM Go!' with a purple header and sidebar. The sidebar on the left contains links for Dashboard, Account (selected), Workbook, Assessment, Logout, Documentation, Getting started, Foundation, and Components. The main content area is titled 'Back' and displays a form for adding a new account. The form fields are: 'No. Pekerja*' (Nomor Pekerja), 'Direktorat' (Direktorat), 'Nama Pekerja*' (Nama Pekerja), 'Username*' (Username), 'Password*' (Password), and 'Email' (Status). A small purple circle with the number '1' is located in the bottom right corner of the sidebar.

Gambar 65 Halaman Form Input

Tinggal kita masukkan data pekerja yang akan ditambahkan, lalu jika sudah selesai mengisi semua *textbox*-nya maka tinggal klik tombol simpan pada akhir halaman atau di bawah halaman daftar pekerja seperti yang ditunjukkan pada gambar 66 di bawah ini.

The screenshot shows a web browser window with the URL `localhost:8080/eppm-go/admin/account/add`. On the left, there is a sidebar with the EPPM Go! logo and navigation links: Dashboard, Account, Workbook, Assessment, Logout, DOCUMENTATION, Getting started, Foundation, and Components. The main content area contains a form for adding a new account. The fields are: Email (with value "ema"), Password* (with value "*****"), Email (with value "ema@email.com"), and Role (with value "User" selected). Below the form is a green "Simpan" button. A note at the bottom says "*Harus Dilisi". At the very bottom of the page, there is a footer with the text "© 2019 PT. Pertamina (Persero)".

Gambar 66 Halaman Form Input

Maka hasilnya akan terlihat pada tabel di halaman awal daftar pekerja saat awal kita mengklik halaman daftar pekerja. Dan data yang baru saja kita tambahkan adalah seperti yang ditunjukkan pada gambar 67 di bawah ini. Seluruh data yang kita tambahkan pada setiap halaman bersifat *real-time* atau akan ter-*update* sesuai dengan waktu sekarang, sehingga ini merupakan salah satu keunggulan yang dipunyai oleh aplikasi yang sedang kita bangun atau *develop*. Membuat aplikasi seperti ini tidaklah sulit, hanya membutuhkan kesabaran dan ketekunan.

+ Tambah Baru				
Show 10 entries				
NOMOR PEKERJA	DIREKTORAT	NAMA PEKERJA	USERNAME	EMAIL
2	User	User	user	user@email.com
111223	tes	pekerja	pekerja	pekerja@email.com
111234	Internship FHCI	Budi Jaya	budi	budijaya@gmail.com
112234	Magang FHCI	Rizal	rizal	rizal
123456	Internship FHCI	Riandaka Rizal	riandakar	riandakarizal@gmail.com
232122	SDM	Ema Ainun Novia	ema	ema@email.com
705423	Megaprojek Pengolahan dan Petrokimia	Bambang Purwito H.	bambangpurwi	bampur@pertamina.com
706169	Megaprojek Pengolahan dan Petrokimia	Widodo Tri Rahardjo	widodotri	widodo.rahardjo@pertamina.com
706655	Megaprojek Pengolahan dan Petrokimia	Arie Wisianto	arie.wisianto	arie.wisianto@pertamina.com

Gambar 67 Halaman Daftar Pekerja

Lanjut kembali kita akan mencoba mengubah atau meng-*update* data dari pengguna Ema Ainun Novia, maka kita langsung klik saja tombol *update* yang tersedia di samping tabelnya. Setelah di klik kita akan diarahkan menuju halaman form *update* seperti yang ditunjukkan pada gambar 68 di bawah ini.

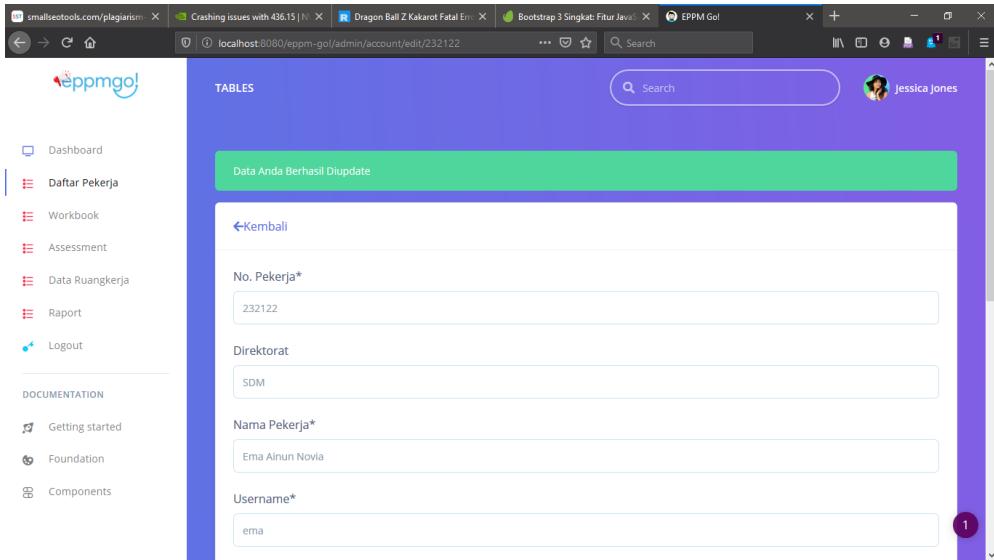
The screenshot shows a web-based application interface titled 'eppmgo!' with a blue header bar. The main content area is a form for updating an employee account. The form fields include:

- No. Pekerja*: 232122
- Direktorat: SDM
- Nama Pekerja*: Ema Ainun Novia
- Username*: ema
- Password*: (redacted)
- Email: ema@email.com
- Role: (dropdown menu with value '1' highlighted)

A vertical sidebar on the left contains navigation links such as Dashboard, Daftar Pekerja, Workbook, Assessment, Data Ruangkerja, Raport, Logout, Documentation, Getting started, Foundation, and Components.

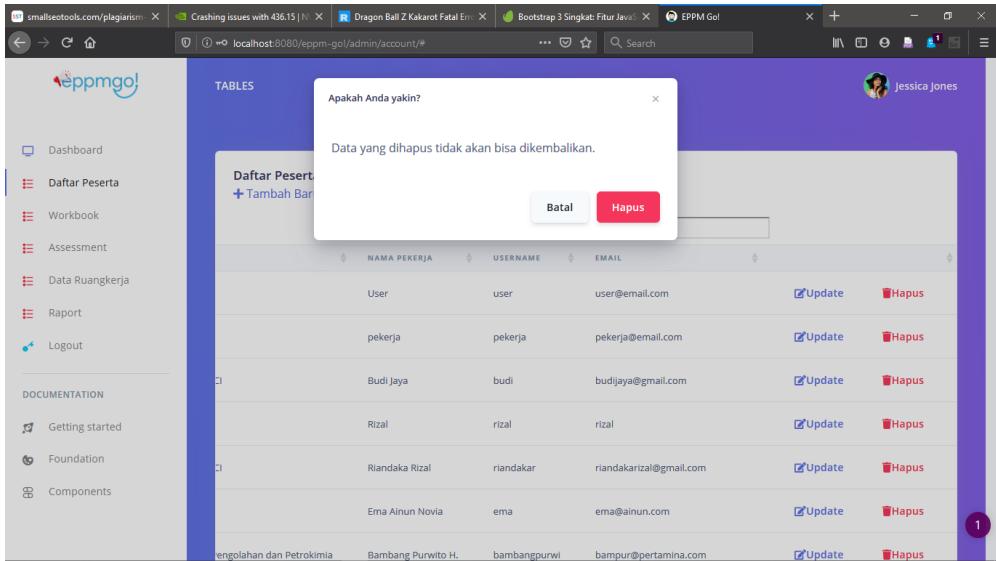
Gambar 68 Halaman Form Update

Setelah ada di halaman form *update*, maka kita tinggal mengubah data yang perlu diubah pada *textbox textbox* yang dirasa perlu. Setelah kita selesai mengubah beberapa *field* maka langsung disimpan saja hasil perubahannya dengan mengklik tombol simpan di bawah halaman form *update*, maka akan muncul notif kalau datanya telah berhasil kita *update* seperti yang ditunjukkan pada gambar 69 di bawah ini.



Gambar 69 Halaman Form Update

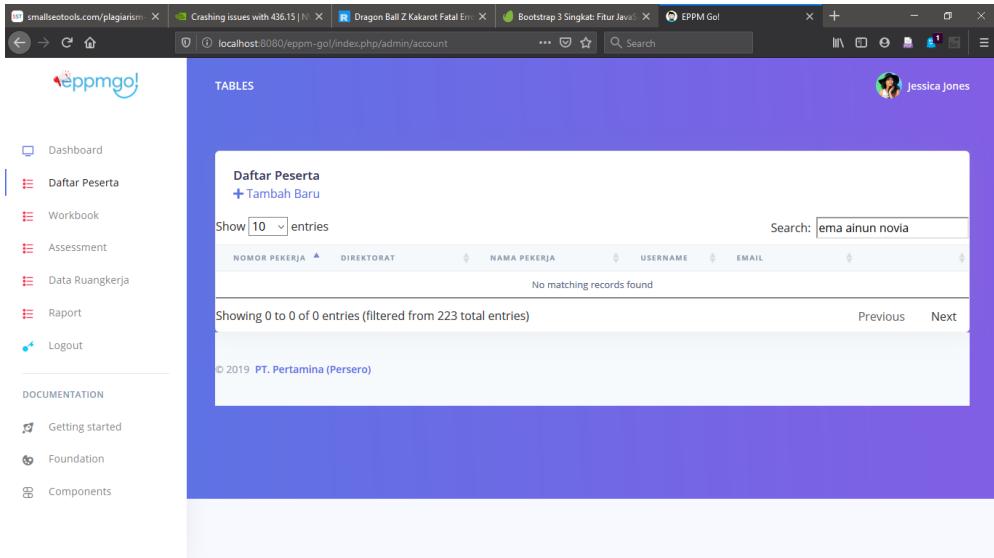
Lalu untuk menghapus data dari pengguna Ema Ainun Novia, tinggal kita klik tombol hapus di samping tombol *update*, maka akan muncul sebuah jendela kecil atau modal yang berguna untuk melakukan konfirmasi sebelum datanya dihapus. Sehingga kita tidak perlu takut atau kaget ketika tombol hapus tidak sengaja tertekan. Seperti yang ditunjukkan pada gambar 70 di bawah ini.



Gambar 70 Modal Tombol Hapus

Jika kita ingin benar benar menghapus data pengguna tersebut, maka langsung saja kita klik hapus. Jika tidak ingin, kita tinggal batalkan saja dengan menekan tombol batal. Karena kita ingin menghapus data pengguna tersebut maka kita tekan tombol hapus. Untuk memastikan bahwa datanya telah dihapus, kita akan mengujinya dengan mencari nama pengguna di kolom pencarian yang disediakan di atas tabel.

Karena data tersebut telah dihapus dari tabel dan juga *database*, maka tidak ditemukan hasil pencarian dari nama penggunanya. Seperti yang dapat kita lihat pada gambar 71 di bawah ini.



Gambar 71 Halaman Daftar Pekerja

Nah sangat mudahkan? Karena menu daftar pekerja telah selesai dibuat kita akan lanjut membuat untuk menu selanjutnya, yaitu menu *assessment* yang bertujuan untuk memberikan penilaian dari pelatihan yang sedang dijalani. Untuk menu *assessment* sendiri, dalam penilaianya akan ada pembobotan nilai sehingga kita juga harus mengatur fungsi matematikanya, agar hasil penilaianya sesuai dengan keinginan dan pembobotan nilai yang kita berikan

Untuk menu *assessment*, dalam memberikan sebuah penilaian pastinya diperlukan sebuah standar kelulusan sehingga hanya nilai yang memenuhi nilainya saja yang mendapat predikat lulus. Oleh karena itu, apabila rata rata nilai setelah dibobotkan lebih dari 70 maka akan mendapat predikat lulus. Bila nilai yang didapatkan setelah dibobotkan ternyata kurang dari 70 maka akan mendapat predikat tidak lulus.

2.4. CRUD Menu *Assessment*

Seperti nama untuk menu ini, *assessment* yaitu yang berarti penilaian. Jadi pada menu ini akan diberikan penilaian dari modul pelatihan yang telah diselesaikan, sehingga peserta pelatihan bisa mendapatkan nilai dari apa yang telah mereka kerjakan.

Sesuai dengan apa yang telah dijelaskan pada sub-bab sebelumnya, untuk menu *assessment* ini ada pembobotan nilai setiap kriteria penilaianya. Contoh untuk kriteria penilaian A bobotnya 10% dan nilai yang diberikan adalah 80. Jadi nilai untuk penilaian A adalah 0,8, sehingga berapapun nilai yang diperoleh untuk kriteria penilaian A akan dikalikan dengan bobot yang telah kita tentukan untuk penilaianya. Selain itu, ada juga standar kelulusan minimum yang telah ditetapkan yang menjadi acuan agar para peserta pelatihan dapat melampaui standar yang ditetapkan. Standar kelulusannya sendiri minimum rata – rata nilai 70. Bila nilai rata – rata yang diperoleh oleh peserta pelatihan lebih dari 70, maka peserta tersebut lulus. Dan sebaliknya apabila nilai rata – rata yang diperoleh peserta tersebut kurang dari 70, maka peserta tersebut dianggap tidak lulus pelatihannya. Coba lihat ilustrasi dari gambar 72 di bawah ini.

RATA - RATA	STATUS LULUS
78.3	Lulus
81.65	Lulus
83.95	Lulus
68.85	Tidak Lulus
77.5	Lulus
78.25	Lulus
84.4	Lulus
76.6	Lulus
79.15	Lulus
75.75	Lulus

Gambar 72 Ilustrasi Standar Kelulusan

Baiklah untuk mari kita bagi kriteria penilaian untuk *assessment* dan juga bobot penilaian dari kriteria penilaianya. Kriteria penilaianya dibagi menjadi 2 yaitu teori dan praktik

- a. Kriteria penilaian pertama adalah, definisi. Dapat menjelaskan definisi tentang suatu teori dengan bobot nilai 10%
- b. Kriteria penilaian kedua adalah, tujuan. Dapat menjelaskan tujuan dari suatu teori dengan bobot nilai 10%

- c. Kriteria penilaian ketiga adalah, pelaksanaan. Dapat menjelaskan pelaksanaan tentang teori tersebut di dunia nyata dengan bobot nilai 20%
- d. Kriteria penilaian keempat adalah, contoh praktik. Dapat menjelaskan contoh praktik dari teori yang diterapkan dengan bobot nilai 15%
- e. Kriteria penilaian kelima adalah, analisis. Dapat menjelaskan analisis dari suatu permasalahan yang dialami dengan bobot nilai 20%
- f. Kriteria penilaian keenam adalah, solusi. Dapat menjelaskan solusi dari suatu permasalahan yang dialami dengan bobot nilai 25%

Dengan ini bobot nilai untuk teori adalah 40% dan bobot nilai untuk praktik adalah 60%. Karena bobot nilai dan juga standar kelulusan telah kita definisikan, langsung saja kita membuat *file* Assessment.php pada direktori controller di dalam folder admin. Lalu kita isikan *file* Assessment.php tersebut dengan isian sebagai berikut.

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

class Assessment extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("assessment_model"); //Load model workbook
        $this->load->library('form_validation'); //Load Library form validation
        $this->load->helper(array('form', 'url'));
        if($this->assessment_model->is_role() != "admin"){
            redirect("index.php/login");
        }
    }
}
```

Gambar 73 Controller Assessment.php

Disini kita membuat sebuah class dengan nama Assessment dengan menggunakan *library* dari *CI_Controller*. Lalu kita gunakan function *__construct()* yang disediakan *CodeIgniter* dengan me-*load* models *assessment_model* dan juga *library* dari form validasi. Lalu jika ada pengguna yang melakukan *login* maka function *is_role()* yang di-*load* dari *assessment_model* mengecek apakah *role* pengguna tersebut adalah bukan admin, jika benar maka arahkan kembali ke halaman *login*.

```

public function index()
{
    $data["assessment"] = $this->assessment_model->getAll(); //ambil data dari model
    // print_r('tes'); die;
    $this->load->view("admin/assessment/list_assessment", $data); //load view data model ke workbook
}

public function get_stage()
{
    $data['data'] = $this->assessment_model->get_data_stage();
    $this->load->view('admin/assessment/new_assessment', $data);
}

public function get_modul()
{
    $id = $this->input->post('id');
    $data = $this->assessment_model->get_data_modul($id);
    echo json_encode($data);
}

public function add()
{
    $data["assessment"] = $this->assessment_model->get_data_stage();
    $this->load->view("admin/assessment/new_assessment", $data); //Load isi form workbook
}

```

Gambar 74 Controller Assessment.php

Setelah selesai membuat *line* di atas, maka kita lanjut untuk mengetik isian seperti yang ada di gambar 74 di atas. Buat function index() dengan mengambil data dari assessment_model menggunakan function getAll() lalu tampilkan halaman *assessment* dengan membawa data yang telah diambil dari assessment_model. Selanjutnya kita membuat function get_stage() dengan mengambil data kembali dari assessment_model melalui function get_data_stage() lalu tampilkan halaman form tambah baru *assessment* dengan membawa datanya dari assessment_model. Lalu kita kembali membuat function get_modul() dengan mengambil id dari *stage* dan ambil data modul dari assessment_model sesuai dengan id *stage* yang dipilih. Jangan lupa membuat function add() dengan isian yang sama seperti function get_stage() yang telah dijelaskan sebelumnya

```

public function aksi_add()
{
    // print_r('tes'); die;
    $this->session->set_flashdata('Sukses', 'Data Anda Berhasil Disimpan'); //pesan berhasil
    $this->assessment_model->save(); //objek model redirect
    redirect('index.php/admin/assessment');
}

public function edit($id=null)
{
    $data["assessment"] = $this->assessment_model->getById($id); //mengambil data berdasarkan id
    if (!$data["assessment"]) show_404(); // jika tidak ada show error
    $this->load->view("admin/assessment/edit_assessment", $data); //Load edit form workbook
}

```

Gambar 75 Controller Assessment.php

Setelah membuat function add() tambahkan kembali function aksi_add() dengan menggunakan function save() yang diambil melalui assessment_model. Lanjut kembali membuat function edit() dengan mengambil data berdasarkan id melalui assessment_model menggunakan function getById(). Lalu tampilkan halaman form edit *assessment* dengan membawa data yang sebelumnya telah diambil menggunakan function getById().

```

function aksi_edit()
{
    $id = $this->input->post('assessment_id');

    $nilai_definisi = $this->input->post('nilai_definisi');
    $nilai_definisi *= 0.1;
    $nilai_tujuan = $this->input->post('nilai_tujuan');
    $nilai_tujuan *= 0.1;
    $nilai_langkah = $this->input->post('nilai_langkah');
    $nilai_langkah *= 0.2;
    $nilai_contoh = $this->input->post('nilai_contoh');
    $nilai_contoh *= 0.15;
    $nilai_analisis = $this->input->post('nilai_analisis');
    $nilai_analisis *= 0.2;
    $nilai_solusi = $this->input->post('nilai_solusi');
    $nilai_solusi *= 0.25;
    $total_nilai = $nilai_definisi + $nilai_tujuan + $nilai_langkah + $nilai_contoh + $nilai_analisis + $nilai_solusi;
    $status_lulus = $nilai_definisi + $nilai_tujuan + $nilai_langkah + $nilai_contoh + $nilai_analisis + $nilai_solusi;

    if($total_nilai >= 70){
        $status_lulus = "Lulus";
    }else{
        $status_lulus = "Tidak Lulus";
    }
}

```

Gambar 76 Controller Assessment.php

Pada gambar 76 di atas, kita membuat function aksi_edit() kita mengambil id dari data yang kita pilih lalu mendefinisikan kriteria penilaian ke dalam variabelnya masing masing, jangan lupa untuk memberikan bobot nilai pada setiap variabel dari masing masing kriteria penilaian. Jika sudah maka keseluruhan nilannya akan kita jumlahkan, dan diberi sebuah pengkondisian bila nilai lebih dari sama dengan 70, maka mendapatkan predikat lulus. Bila tidak maka mendapat predikat tidak lulus. Pada *line* sintaks di atas, kita terlebih dahulu mendefinisikan variabel penilaian dan juga bobot nilainya jika sudah selesai maka tinggal menyimpan dan memasukkan datanya ke dalam *database* pada tabel assessment. Dan jangan lupa untuk membuat function delete() untuk menghapus data yang dipilih. Seperti yang ditunjukkan oleh gambar 77 berikut ini.

```

        }

        $data = array (
            'assessment_id' => $this->input->post('assessment_id'),
            'stage_id' => $this->input->post('stage_id'),
            'modul_id' => $this->input->post('modul_id'),
            'employee_id' => $this->input->post('employee_id'),
            'nilai_definisi' => $this->input->post('nilai_definisi'),
            'nilai_tujuan' => $this->input->post('nilai_tujuan'),
            'nilai_langkah' => $this->input->post('nilai_langkah'),
            'nilai_contoh' => $this->input->post('nilai_contoh'),
            'nilai_analisis' => $this->input->post('nilai_analisis'),
            'nilai_solusi' => $this->input->post('nilai_solusi'),
            'total_nilai' => $total_nilai,
            'status_lulus' => $status_lulus
        );

        // print_r($data); die;
        $this->assessment_model->update($id,$data);
        $this->session->set_flashdata('success', 'Data Anda Berhasil Diupdate');
        redirect('index.php/admin/assessment');

    }

    public function delete($id=null)
    {
        if (!isset($id)) show_404();

        if ($this->assessment_model->delete($id)){
            redirect('index.php/admin/assessment');
        }
    }
}

```

Gambar 77 Controller Assessment.php

Setelah menyelesaikan isian seperti di atas, maka selesai sudah langkah untuk membuat *file* controller Assessment.php dan akan dilanjutkan untuk membuat *file* models dengan nama Assessment_model.php pada direktori models di dalam folder admin.

Jika sudah dibuat, maka langsung saja kita mendefinisikan class Assessment_model dengan menggunakan *library* dari CI_Model lalu kita definisikan juga variabel variabel yang akan digunakan untuk menu *assessment*. Setelahnya buat kembali function is_role() yang berguna untuk mengetahui *role* dari setiap pengguna yang akan melakukan *login*. Lanjut lagi kita akan mengambil seluruh data dari tabel *assessment* pada *database* dengan membuat function getAll()

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

class Assessment_model extends CI_Model
{
    private $_table = "assessment";

    public $assessment_id;
    public $stage_id;
    public $modul_id;
    public $employee_id;
    public $nilai_definisi;
    public $nilai_tujuan;
    public $nilai_langkah;
    public $nilai_contoh;
    public $nilai_analisis;
    public $nilai_solusi;
    public $total_nilai;

    //fungsi cek Level
    function is_role()
    {
        return $this->session->userdata('role');
    }

    public function getAll()
    {
        $this->db->select('assessment.*', 'account.employee_name as yaww');
        $this->db->join('account', 'assessment.employee_id = account.employee_id');
        $this->db->from('assessment');
        return $this->db->get()->result();
        // print_r('tes'); die;
    }
}
```

Gambar 78 Models Assessment_model.php

Lanjut lagi kita membuat function getById() dengan mengambil hanya id yang kita pilih pada halaman menu *assessment*. Langkah selanjutnya adalah membuat function get_data_stage() yang berfungsi untuk mengambil data *stage* dari tabel *stage* pada *database*. Dan kita buat juga function get_data_modul() yang berfungsi untuk mengambil data modul berdasarkan *stage* yang kita pilih nanti di halaman menu *assessment*.

Dapat dilihat disini, bahwa kedua function get_data_stage() dan get_data_modul() ini saling berhubungan karena data modul baru akan diambil bila sebelumnya telah memilih *stage*-nya.

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

class Assessment_model extends CI_Model
{
    private $_table = "assessment";

    public $assessment_id;
    public $stage_id;
    public $modul_id;
    public $employee_id;
    public $nilai_definisi;
    public $nilai_tujuan;
    public $nilai_langkah;
    public $nilai_contoh;
    public $nilai_analisis;
    public $nilai_solusi;
    public $total_nilai;

    //fungsi cek level
    function is_role()
    {
        return $this->session->userdata('role');
    }

    public function getAll()
    {
        $this->db->select('assessment.*', 'account.employee_name as yaww');
        $this->db->join('account', 'assessment.employee_id = account.employee_id');
        $this->db->from('assessment');
        return $this->db->get()->result();
        // print_r('tes'); die;
    }
}
```

Gambar 79 Models Assessment_model.php

Nah kita teruskan untuk membuat function save() yang berfungsi untuk menyimpan data yang kita isikan pada form *input assessment* ke dalam *database*. Sama seperti function aksi_edit() di file controller Assessment.php terlebih dahulu kita definisikan variabel untuk setiap kriteria penilaian dan juga pembobotannya. Jika sudah diberikan pengkodisian bila nilai lebih dari sama dengan 70 akan mendapatkan predikat lulus, dan bila kurang dari 70 akan mendapat predikat tidak lulus.

```
public function save()
{
    $nilai_definisi = $this->input->post('nilai_definisi');
    $nilai_definisi *= 0.1;
    $nilai_tujuan = $this->input->post('nilai_tujuan');
    $nilai_tujuan *= 0.1;
    $nilai_langkah = $this->input->post('nilai_langkah');
    $nilai_langkah *= 0.2;
    $nilai_contoh = $this->input->post('nilai_contoh');
    $nilai_contoh *= 0.15;
    $nilai_analisis = $this->input->post('nilai_analisis');
    $nilai_analisis *= 0.2;
    $nilai_solusi = $this->input->post('nilai_solusi');
    $nilai_solusi *= 0.25;
    $total_nilai = $nilai_definisi + $nilai_tujuan + $nilai_langkah + $nilai_contoh + $nilai_analisis + $nilai_solusi;
    $status_lulus = $nilai_definisi + $nilai_tujuan + $nilai_langkah + $nilai_contoh + $nilai_analisis + $nilai_solusi;

    if($total_nilai >= 70){
        $status_lulus = "Lulus";
    }else{
        $status_lulus = "Tidak Lulus";
    }
}
```

Gambar 80 Models Assessment_model.php

Jika telah didefinisikan variabel variabelnya, langkah selanjutnya adalah membuat sintaks dengan isian yang akan memasukkan data data yang telah diisi pada form *input assessment* ke dalam *database*. Seperti yang ditunjukkan oleh gambar 81 di bawah ini.

```
$data = array (
    "stage_id" => $this->input->post("stage_id"),
    "modul_id" => $this->input->post("modul_id"),
    "employee_id" => $this->input->post("employee_id"),
    "nilai_definisi" => $this->input->post("nilai_definisi"),
    "nilai_tujuan" => $this->input->post("nilai_tujuan"),
    "nilai_langkah" => $this->input->post("nilai_langkah"),
    "nilai_contoh" => $this->input->post("nilai_contoh"),
    "nilai_analisis" => $this->input->post("nilai_analisis"),
    "nilai_solusi" => $this->input->post("nilai_solusi"),
    "total_nilai" => $total_nilai,
    "status_lulus" => $status_lulus
);

$this->db->insert("assessment", $data);
}
```

Gambar 81 Models Assessment_model.php

Masih semangat? Lanjut saja kita membuat function *update* yang berfungsi untuk mengubah suatu data yang kita inginkan dengan mengambil id dari data yang dipilih. Jika sudah selesai mengubah datanya, maka data yang baru saja diubah akan dimasukkan ke dalam *database*. Begitupun dengan function *delete()* yang akan mengambil id dari data yang ingin kita hapus, jika benar benar ingin dihapus maka data tersebut akan terhapus pada halaman menu *assessment* dan juga *database*-nya. Seperti yang ditunjukkan pada gambar 82 di bawah ini.

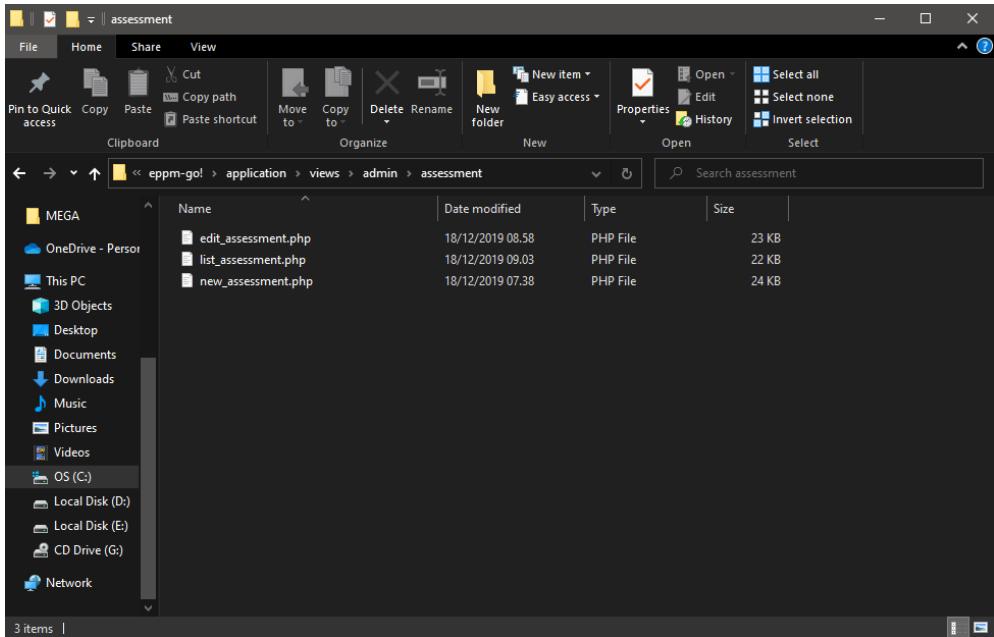
```
public function update($id, $data)
{
    $this->db->where('assessment_id',$id);
    $this->db->update("assessment", $data);
}

public function delete($id)
{
    // $this->_deleteImage($id);
    return $this->db->delete($this->_table, array("assessment_id" => $id));
}
```

Gambar 82 Models Assessment_model.php

Kalau function delete() sudah selesai kita buat, maka selesai sudah langkah langkah pembuatan dari models Assessment_model.php. Mari kita istirahat sejenak, coba renungkan setelah sampai sejauh ini apa saja yang telah teman teman dapatkan saat membaca tutorial ini? Apakah ada manfaatnya untuk pribadi teman teman semua? Ataukah sudah sejauh ini teman teman masih tidak mudeng dari apa yang sudah dibaca? Coba renungkan!

Baiklah dengan telah selesainya models Assessment_model.php maka langkah terakhir dalam membuat menu *assessment* adalah bagian views. Untuk itu silakan buat terlebih dahulu folder *assessment* dan langsung membuat *file* dengan nama list_assessment.php, new_assessment.php, dan juga edit_assessment.php di dalamnya pada direktori views di dalam folder admin.



Gambar 83 File Views Assessment

Jika sudah dibuat ketiga *file* tersebut, lanjut untuk mengisi *file* list_assessment.php. Kembali saya ingatkan, disini saya menggunakan *bootstrap argon dashboard* dan hanya memodifikasinya sesuai dengan kebutuhan pada tutorial kali ini. Kembali ke topik, kita buat sebuah tabel untuk menampung atau mewadahi data yang akan kita ambil dari *database* dan isikan sesuai dengan gambar 84 di bawah ini.

```
<!-- table -->
<div class="row">
  <div class="col">
    <div class="card shadow">
      <div class="card-header border-0">
        <h3 class="mb-0">Assessment</h3>
        <a href=<?php echo site_url('admin/assessment/add') ?>><i class="fas fa-plus"></i> Tambah Baru</a>
      </div>
      <div class="table-responsive">
        <table class="table align-items-center table-flush">
          <thead class="thead-light">
            <tr>
              <th scope="col">ID Stage</th>
              <th scope="col">ID Modul</th>
              <th scope="col">Nama Pekerja</th>
              <!-- <th scope="col">Nilai Definisi</th>
              <th scope="col">Nilai Tujuan</th>
              <th scope="col">Nilai Langkah</th>
              <th scope="col">Nilai Contoh</th>
              <th scope="col">Nilai Analisis</th>
              <th scope="col">Nilai Solusi</th> -->
              <th scope="col">Total Nilai</th>
              <th scope="col">Status Kelulusan</th>
              <th scope="col"></th>
            </tr>
          </thead>
```

Gambar 84 Views list_assessment.php

Dapat dilihat pada gambar 84, ada tabel yang di *comment* – *comment* membuat baris atau *line* yang dipilih tidak akan memengaruhi program – ini dikarenakan sebelumnya baik untuk menampilkan keseluruhan kolom tapi lebih baik lagi hanya menampilkan kolom yang dibutuhkan saja.

Kembali lagi kita menggunakan perulangan *foreach* untuk menampilkan datanya sehingga keseluruhan datanya dapat ditampilkan tidak hanya satu data saja yang ditampilkan, seperti yang ditunjukkan pada gambar 85 di bawah ini tentang penggunaan dari perulangan *foreach*.

```
<tbody>
    <?php foreach ($assessment as $assessment): ?>
    <tr>
        <td width="150">
            <?php echo $assessment->stage_id ?>
        </td>
        <td>
            <?php echo $assessment->modul_id ?>
        </td>
        <td>
            <?php echo $assessment->yaww ?>
        </td>
        <!-- <td>
            <?php echo $assessment->nilai_definisi ?>
        </td>
        <td>
            <?php echo $assessment->nilai_tujuan ?>
        </td>
        <td>
            <?php echo $assessment->nilai_Langkah ?>
        </td>
        <td>
            <?php echo $assessment->nilai_contoh ?>
        </td>
        <td>
            <?php echo $assessment->nilai_analisis ?>
        </td>
        <td>
            <?php echo $assessment->nilai_solusi ?>
        </td> -->
        <td>
            <?php echo $assessment->total_nilai ?>
        </td>
```

Gambar 85 Views list_assessment.php

Lalu kita kembali membuat tombol untuk *update* dan *hapus* untuk mengubah data yang kita pilih dan menghapus data yang kita pilih, kita buat tombol tombol tersebut di samping datanya. Kembali saya ingatkan, untuk menampilkan tabel pada setiap halaman kita menggunakan DataTables.

```
<td width="250">
    <a href="=php echo site_url('admin/assessment/edit/'.$assessment-&gt;assessment_id)?" 
        class="btn btn-small"><i class="fas fa-edit"></i>Update</a>
    <a onclick="deleteConfirm('=php echo site_url('admin/assessment/delete/'.$assessment-&gt;assessment_id)?'" href="#" class="btn btn-small text-danger"><i class="fas fa-trash"></i>Hapus</a>
</td>
```

Gambar 86 Views list_assessment.php

Karena, masih menggunakan DataTables jadi kita harus menggunakan sintaks jQuery untuk menggunakan *resources* dari DataTables. Setelah itu, untuk sintaks jQuery dari DataTables kita taruh dibawah setelah sintaks pembuatan tabel. Jangan lupa untuk menyisipkan tag <script> karena masih satu bagian dari JavaScript. Selain itu, karena kita menggunakan *resources* dari luar ada beberapa cara yang bisa digunakan, yaitu dengan cara:

1. Mengunduh *library*-nya dan disimpan pada folder *assets* yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini akan menggunakan *local resource* dari PC atau laptop kita.

- Menyalin *link CDN* (*Content Delivery Network*) yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini hanya memerlukan akses internet. *Link CDN* DataTables untuk CSS adalah //cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css dan *link CDN* DataTables untuk JavaScript adalah //cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js

```
<script>$(document).ready( function () {
|   $('table').DataTable();
} );
</script>
```

Gambar 87 Views list_assessment.php

Untuk tombol hapus, kita harus selalu menggunakan konfirmasi sebelum dilakukannya penghapusan data. Ini berguna untuk mencegah dan meminimalisir terjadinya penghapusan data secara tidak sengaja, dan kita menggunakan modal untuk konfirmasinya seperti yang ditunjukkan gambar 88 di bawah ini

```
<script>
function deleteConfirm(url){
  $('#btn-delete').attr('href', url);
  $('#deleteModal').modal();
}
</script>
```

Gambar 88 Views list_assessment.php

Maka selesai sudah *file* list_assessment.php yang kita buat, langkah selanjutnya adalah membuat *file* new_assessment.php yang berfungsi untuk memberikan sebuah form penilaian untuk disimpan dan dimasukkan ke dalam *database*. Oke langsung saja isikan *file* tersebut dengan isian seperti yang ditunjukkan pada gambar 89 di bawah ini.

```
<div class="card mb-3">
    <div class="card-header">
        <a href="=php echo site_url('admin/assessment/') ?&gt;"&gt;&lt;i class="fas fa-arrow-left"&gt;&lt;/i&gt; Kembali&lt;/a&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;
        &lt;form action="<?=base_url();?&gt;index.php/admin/assessment/aksi_add" method="post" enctype="multipart/form-data"&gt;
            &lt;!-- &lt;input type="hidden" name="assessment_id" value="<?=php echo $assessment-&gt;assessment_id?&gt;" /&gt; --&gt;

            &lt;div class="form-group"&gt;
                &lt;label for="Stage"&gt;Stage*&lt;/label&gt;
                &lt;select class="form-control" name="stage_id" placeholder="Stage" id="stage_id" required&gt;
                    &lt;option&gt; - Pilih Stage -&lt;/option&gt;
                    &lt;?php
                        foreach ($assessment-&gt;result() as $baris) {
                            echo "&lt;option value='".$baris-&gt;stage_id."'&gt;".$baris-&gt;stage_name."&lt;/option&gt;";
                        }
                    &gt;
                &lt;div class="invalid-feedback"&gt;&lt;?php echo form_error('stage_id')?&gt;
                &lt;/div&gt;
                &lt;/select&gt;
            &lt;/div&gt;

            &lt;div class="form-group"&gt;
                &lt;label for="Modul"&gt;Modul*&lt;/label&gt;
                &lt;select class="form-control" name="modul_id" placeholder="Modul" id="modul_id" required&gt;
                    &lt;option value=""&gt; - Pilih Modul -&lt;/option&gt;
                    &lt;div class="invalid-feedback"&gt;&lt;?php echo form_error('modul_id')?&gt;
                    &lt;/div&gt;
                    &lt;/select&gt;
            &lt;/div&gt;
        &lt;/form&gt;
    &lt;/div&gt;
</pre
```

Gambar 89 Views new_assessment.php

Disini terdapat 2 *textbox* yang berbentuk form *dropdown*. Karena *field stage* dan juga modul ini saling berhubungan maka kita akan menggunakan AJAX untuk membuat *dropdown chained* atau sebuah form *dropdown* yang saling berkaitan.

```

<div class="form-group">
    <label for="No. Pekerja">No. Pekerja*</label>
    <input class="form-control" type="text" name="employee_id" placeholder="Nomor Pekerja" required>
    <div class="invalid-feedback"><?php echo form_error('employee_id')?>
    </div>
</div>

<div class="form-group">
    <label for="Nilai Definisi">Nilai Untuk Definisi</label>
    <input class="form-control" type="text" name="nilai_definisi" placeholder="Nilai Definisi" required>
    <div class="invalid-feedback"><?php echo form_error('nilai_definisi')?>
    </div>
</div>

<div class="form-group">
    <label for="Nilai Tujuan">Nilai Untuk Tujuan</label>
    <input class="form-control" type="text" name="nilai_tujuan" placeholder="Nilai Tujuan" required>
    <div class="invalid-feedback"><?php echo form_error('nilai_tujuan')?>
    </div>
</div>

<div class="form-group">
    <label for="nilai_langkah">Nilai Untuk Langkah-Langkah</label>
    <input class="form-control" type="text" name="nilai_langkah" placeholder="Nilai Langkah" required>
    <div class="invalid-feedback"><?php echo form_error('nilai_langkah')?>
    </div>
</div>

```

Gambar 90 Views new_assessment.php

Sama saja seperti pada menu *workbook* dan juga daftar pekerja yang sebelumnya telah dibuat, disini kita juga membuat sebuah *textbox* untuk dapat mengisi formnya. Dapat dilihat pada gambar 90 ini, kita sudah membuat *textbox* untuk *field* No. pekerja, dan juga kriteria – kriteria penilaian yang sudah dibahas sebelumnya.

```

<div class="form-group">
    <label for="nilai_contoh">Nilai Untuk Contoh Penerapan</label>
    <input class="form-control <?php echo form_error('nilai_contoh')? 'is-invalid': ''?>" type="text" name="nilai_contoh" placeholder="Nilai Contoh" required/>
    <div class="invalid-feedback"><?php echo form_error('nilai_contoh')?>
    </div>
</div>

<div class="form-group">
    <label for="nilai_analisis">Nilai Untuk Analisis</label>
    <input class="form-control <?php echo form_error('nilai_analisis')? 'is-invalid': ''?>" type="text" name="nilai_analisis" placeholder="Nilai Analisis" required/>
    <div class="invalid-feedback"><?php echo form_error('nilai_analisis')?>
    </div>
</div>

<div class="form-group">
    <label for="nilai_solusi">Nilai Untuk Solusi</label>
    <input class="form-control <?php echo form_error('nilai_solusi')? 'is-invalid': ''?>" type="text" name="nilai_solusi" placeholder="Nilai Solusi" required/>
    <div class="invalid-feedback"><?php echo form_error('nilai_solusi')?>
    </div>
</div>

<input class="btn btn-success" type="submit" name="btn" value="Simpan" />
</form>

```

Gambar 91 Views new_assessment.php

Jika semua *textbox* dari seluruh *field* yang dibutuhkan telah selesai dibuat, jangan lupa untuk memberikan tombol simpan untuk dapat menyimpan isian isian pada *textbox* yang akan diisi nanti ke dalam *database* kita. Sempat disinggung pada AJAX sebelumnya, karena *stage* dan modul itu saling berhubungan satu sama lain maka untuk memfasilitasi agar dapat membuat *dropdown chained* yaitu dengan AJAX. Sama seperti *JavaScript*, sintaks AJAX ditempatkan pada bagian bawah sebelum tag </body> dan kita diwajibkan untuk memberikan *link resources* CDN agar dapat menggunakan AJAX. *Link* CDN untuk AJAX sendiri ditempatkan di atas HTML pada bagian tag <head> dengan *link Content Delivery Network* berikut ini <https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js>.

Kalau lebih suka menggunakan AJAX dengan mengunduh *resources*-nya maka unduh terlebih dahulu dan taruh pada direktori *assets*. Berikut sintaks AJAX yang berada pada bawah tag </body> seperti yang ditunjukkan pada gambar 92 berikut ini.

```
<script> $(document).ready(function(){
    $('#stage_id').change(function(){
        // console.log('tes');
        var id=$(this).val();
        // console.log(id);
        $.ajax({
            url : "<?php echo base_url();?>index.php/admin/assessment/get_modul",
            method : "POST",
            data : {id: id},
            async : false,
            dataType : 'json',
            success: function(data){
                var html = '';
                var i;
                for(i=0; i<data.length; i++){
                    console.log(data)
                    html += '<option value='+data[i].modul_id+'>'+data[i].modul_name+'</option>'
                }
                $('#modul_id').html(html);
                //nu bener modul_id
            }
        });
    });
});</script>
```

Gambar 92 Sintaks AJAX

Seperti yang dapat kita lihat pada gambar 92 di atas, disinilah pembuktian bahwa untuk *field stage* dan modul itu saling berhubungan. Karena pada setiap *stage* memiliki modul yang berbeda – beda. Sehingga apabila kita memilih *stage* 1 maka kita juga hanya dapat memilih modul modul yang terdaftar pada *stage* 1.

Selesai sudah tahapan untuk membuat *file* new_assessment.php, tinggal satu langkah terakhir yaitu membuat *file* edit_assessment.php yang kurang lebih memiliki langkah yang sama. Sehingga ini akan lebih memudahkan kita dalam membuat *file* edit_assessment.php

```
<!-- Table -->
<div class="card mb-3">
  <div class="card-header">
    <a href="php echo site_url('admin/assessment/') ?&gt;"<i class="fas fa-arrow-left">
```

Gambar 93 Views edit_assessment.php

Dapat dilihat ini adalah langkah yang sama seperti menu menu yang sebelumnya telah kita buat. Tinggal membuat *textbox* untuk form dan panggil datanya sesuai dengan id yang dipilih sebelumnya. Mulai dari sini kita buat terlebih dahulu *textbox* sesuai dengan *field* yang dibutuhkan, seperti yang ditunjukkan pada gambar 94 sampai gambar 96 di bawah ini

```
<div class="form-group">
    <label for="name">No. Pekerja*</label>
    <input class="form-control <?php echo form_error('employee_id') ? 'is-invalid':'' ?>" type="text" name="employee_id" min="0" placeholder="No. Pekerja" value="<?php echo $assessment->employee_i
    <div class="invalid-feedback">
        <?php echo form_error('employee_id') ?>
    </div>
</div>

<div class="form-group">
    <label for="nilai_definisi">Nilai Untuk Definisi</label>
    <input class="form-control <?php echo form_error('nilai_definisi') ? 'is-invalid':'' ?>" type="text" name="nilai_definisi" min="0" placeholder="Nilai Definisi" value="<?php echo $assessment->nila
    <div class="invalid-feedback">
        <?php echo form_error('nilai_definisi') ?>
    </div>
</div>
```

Gambar 94 Views edit_assessment.php

```
<div class="form-group">
    <label for="name">No. Pekerja*</label>
    <input class="form-control <?php echo form_error('employee_id') ? 'is-invalid':'' ?>" type="text" name="employee_id" min="0" placeholder="No. Pekerja" value="<?php echo $assessment->employee_i
    <div class="invalid-feedback">
        <?php echo form_error('employee_id') ?>
    </div>
</div>

<div class="form-group">
    <label for="nilai_definisi">Nilai Untuk Definisi</label>
    <input class="form-control <?php echo form_error('nilai_definisi') ? 'is-invalid':'' ?>" type="text" name="nilai_definisi" min="0" placeholder="Nilai Definisi" value="<?php echo $assessment->nila
    <div class="invalid-feedback">
        <?php echo form_error('nilai_definisi') ?>
    </div>
</div>
```

Gambar 95 Views edit_assessment.php

```

        <div class="form-group">
    <label for="nilai_analisis">Nilai Untuk Analisis</label>
    <input class="form-control" type="text" name="nilai_analisis" min="0" placeholder="Nilai Analisis" value=<?php echo $assessment->nila
    <div class="invalid-feedback">
        <?php echo form_error('nilai_analisis') ?>
    </div>
    </div>
    <div class="form-group">
    <label for="nilai_solusi">Nilai Untuk Solusi</label>
    <input class="form-control" type="text" name="nilai_solusi" min="0" placeholder="Nilai Solusi" value=<?php echo $assessment->nila
    <div class="invalid-feedback">
        <?php echo form_error('nilai_solusi') ?>
    </div>
    </div>
    </div>

    <input class="btn btn-success" type="submit" name="btn" value="Simpan" />
</form>

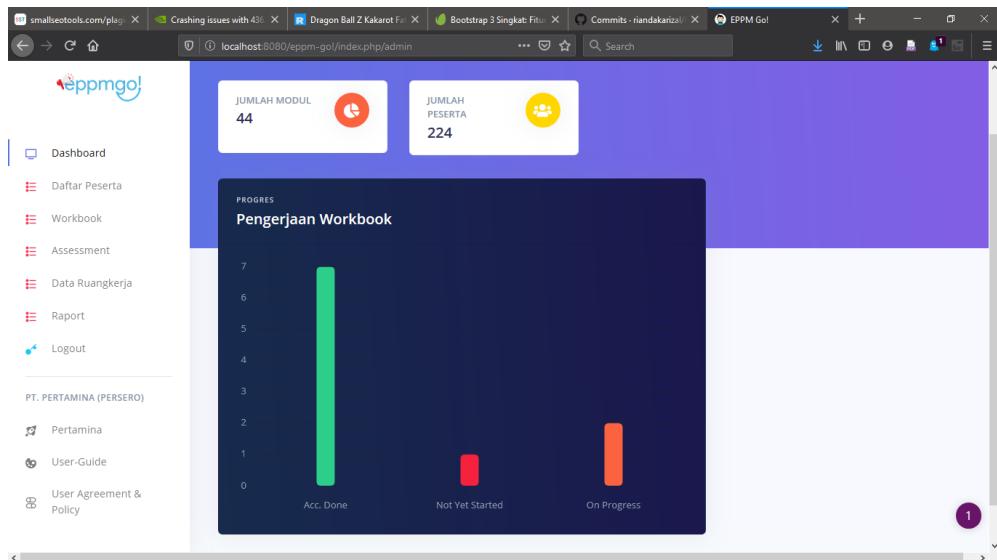
```

Gambar 96 Views edit_assessment.php

Dengan ini selesailah sudah keseluruhan proses untuk membuat menu *assessment* yaitu controller, models, dan juga views-nya. Karena itu marilah kita coba menu *assessment* ini dengan menggunakan *browser* atau peramban yang menjadi preferensi kita masing – masing.

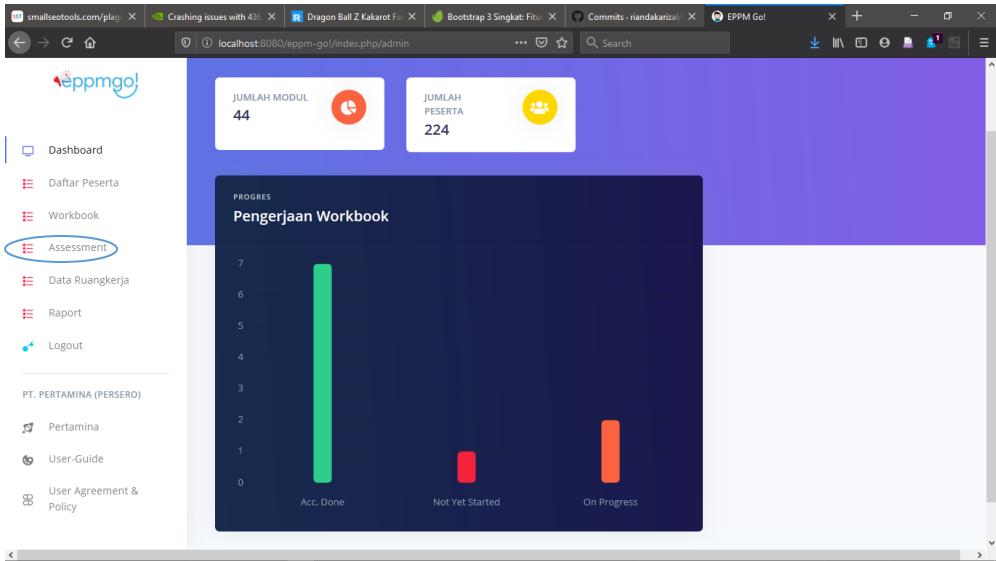
Bukalah *browser*-nya lalu ketikkan `localhost:8080/eppm-go!/admin/assessment` yang akan menuju langsung ke menu *assessment*. Tetapi akan diarahkan ke dalam halaman *login* untuk terlebih dahulu melakukan *login* dengan *username* dan *password* untuk admin. Kenapa hal ini terjadi? Karena bila suatu pengguna walaupun admin tidak diatur penggunaan *session* maka hal ini akan membuat aplikasi *website* yang kita sudah rancang dan kembangkan akan tampak tidak elok, dan terlihat seperti memiliki cacat.

Lanjut bila sudah masuk ke dalam aplikasi EPPM Go! maka akan terlihat seperti yang ditunjukkan pada gambar 97 di bawah ini.



Gambar 97 Halaman Awal Admin

Jika sudah seperti di atas, maka langsung kita memilih menu *assessment* karena kita ingin mencoba apakah sudah berhasil yang baru saja kita buat sebelumnya? Pilih menu *assessment* pada menu bar yang terdapat di samping layar laptop atau PC kalian



Gambar 98 Memilih Menu Assessment

The screenshot shows the Assessment list page. The sidebar includes the following items:

- Dashboard
- Daftar Peserta
- Workbook
- Assessment**
- Data Ruangkerja
- Raport
- Logout

Below the sidebar is a documentation section with links for Getting started, Foundation, and Components.

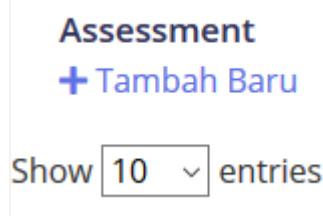
The main content area displays a table titled "Assessment" with the following data:

ID STAGE	ID MODUL	NAMA PEKERJA	TOTAL NILAI	STATUS KELULUSAN	Update	Hapus
1	2.1	admin	78.3	Lulus	Update	Hapus
1	2.1	Abdul Karim Amarullah	81.65	Lulus	Update	Hapus
1	2.2	Riandaka Rizal	83.95	Lulus	Update	Hapus
1	2.2	Riandaka Rizal	68.85	Tidak Lulus	Update	Hapus
1	3.1	admin	77.5	Lulus	Update	Hapus
1	2.1	Riandaka Rizal	78.25	Lulus	Update	Hapus
1	1.1	Muhammad Fauzan Aristyo	84.4	Lulus	Update	Hapus
1	1.1	Riandaka Rizal	76.6	Lulus	Update	Hapus
1	6.3	Muhammad Fauzan Aristyo	70.15	Tidak Lulus	Update	Hapus

Gambar 99 Halaman Menu Assessment

Seperti yang dapat dilihat pada gambar 97, untuk halaman awal dari menu *assessment* sudah berhasil berjalan tanpa ada kendala. Datanya dapat terpanggil semua dan juga standar kelulusan yang sudah diterapkan pada sintaks models sebelumnya juga sudah berjalan. Sekali lagi dapat dilihat, bahwa nilai rata – rata menjadi acuan dari status kelulusan masing masing peserta. Apabila nilai rata – rata lebih dari 70 maka lulus, dan jika yang terjadi adalah sebaliknya nilai rata – rata kurang dari 70 maka tidak lulus.

Mari mencoba untuk menambahkan suatu penilaian baru untuk satu pengguna, untuk menambahkan penilaian pada peserta dapat dilakukan dengan cara menekan atau mengklik tombol tambah baru pada pojok kiri atas dari halaman menu *assessment* seperti yang ditunjukkan pada gambar 100 di bawah ini.



Gambar 100 Tombol Tambah Baru

Dengan menekan tombol tersebut, kita akan diarahkan menuju halaman form tambah baru *assessment* yang berguna untuk memasukkan data data penilaian pada form yang telah disediakan. Seperti yang ditunjukkan pada gambar 101 di bawah ini.

The screenshot shows a web-based application interface for managing assessments. On the left, there's a sidebar with a blue header containing the logo 'eppmgo!' and several menu items: Dashboard, Daftar Pekerja, Workbook, Assessment, Data Ruangkerja, Report, Logout, DOCUMENTATION, Getting started, Foundation, and Components. The main content area has a white background with a purple border. At the top, there's a back button labeled 'Kembali'. Below it, there are four input fields with dropdown menus: 'Stage*' (with placeholder '- Pilih Stage -'), 'Modul*' (with placeholder '- Pilih Modul -'), 'No. Pekerja*' (with placeholder 'Nomor Pekerja'), and three numerical input fields: 'Nilai Untuk Definisi' (placeholder 'Nilai Definisi'), 'Nilai Untuk Tujuan' (placeholder 'Nilai Tujuan'), and 'Nilai Untuk Langkah-Langkah' (placeholder 'Nilai Langkah'). A small purple circle with the number '1' is located at the bottom right corner of the main form area.

Gambar 101 Halaman Form Tambah Baru

Nah disini, kita dapat membuktikan apakah *dropdown* dari *field stage* dan modul memang berhubungan apa tidak? Klik saja pada *stage* dan pilih *stage* 3 maka modul yang dapat dipilih adalah modul yang hanya terdapat pada *stage* 3 saja. Seperti yang dapat kita lihat pada gambar 102 berikut ini.

[Kembali](#)

Stage*

Stage 3

Modul*

Plan Maintenance and Compliance

Plan Maintenance and Compliance

Resource Loading and Leveling
Critical Path Optimization
Design MS Tracking and Review
Lookback Reviews
Review Architecture
Problem Solving and Root Cause Analysis
Strategic Procurement Involvement
Commercial Optimization Toolkit
Commercial Negotiation
Contract Administration Team
Change Order Management
Claims Management
Integrated Ramp Up Planning

The screenshot shows a user interface for selecting a stage and a module. The 'Stage*' field is populated with 'Stage 3'. The 'Modul*' field has a dropdown menu open, showing a list of items. The item 'Plan Maintenance and Compliance' is highlighted with a blue background, indicating it is the selected option. A scroll bar is visible on the right side of the dropdown menu.

Gambar 102 Dropdown Stage dan Modul

Lanjut kita mengisi form tersebut sesuai dengan penilaian yang ingin diberikan, jika sudah maka tinggal menyimpannya saja dengan menekan tombol simpan pada bagian bawah dari halaman ini. Lalu data yang baru saja kita masukkan dan simpan sudah dapat dilihat pada halaman awal menu *assessment*. Seperti yang dapat dilihat pada gambar 103 untuk tombol simpan dan juga gambar 104 untuk data yang sudah tersimpan.

Nilai Untuk Definisi
60

Nilai Untuk Tujuan
65

Nilai Untuk Langkah-Langkah
77

Nilai Untuk Contoh Penerapan
82

Nilai Untuk Analisis
70

Nilai Untuk Solusi
69

Simpan

Gambar 103 Tombol Simpan Halaman Assessment

ID STAGE	ID MODUL	NAMA PEKERJA	TOTAL NILAI	STATUS KELULUSAN	Update	Hapus
3	1.7	Rizal	71.45	Lulus		

Gambar 104 Data yang Sudah Disimpan

Lalu bagaiman jika sebelumnya pada pengisian form di halaman form tambah baru penilaian ada nilai yang salah? Tenang saja, kita dapat merubahnya dengan menekan tombol *update* yang telah disediakan pada bagian samping kanan tabel. Klik tombol tersebut maka kita akan diarahkan menuju halaman form *update*

Bila sudah selesai mengubah data yang dikiranya perlu perbaikan, maka langsung saja disimpan perubahan tersebut dengan menekan kembali tombol simpan yang juga disediakan pada bagian bawah halaman form *update* ini. Dan ternyata setelah ada data yang diubah status lulus dari peserta yang bernama rizal menjadi tidak lulus. Seperti yang ditunjukkan pada gambar 105 dan juga gambar 106 berikut ini.

The screenshot shows a web browser window with the URL `localhost:8080/eppm-go/admin/assessment/edit/17`. The page displays an assessment form for a participant named Rizal. The form contains six input fields with the following values:

Kriteria	Nilai
Nilai Untuk Tujuan	60
Nilai Untuk Langkah	65
Nilai Untuk Contoh Penerapan	77
Nilai Untuk Analisis	82
Nilai Untuk Solusi	70

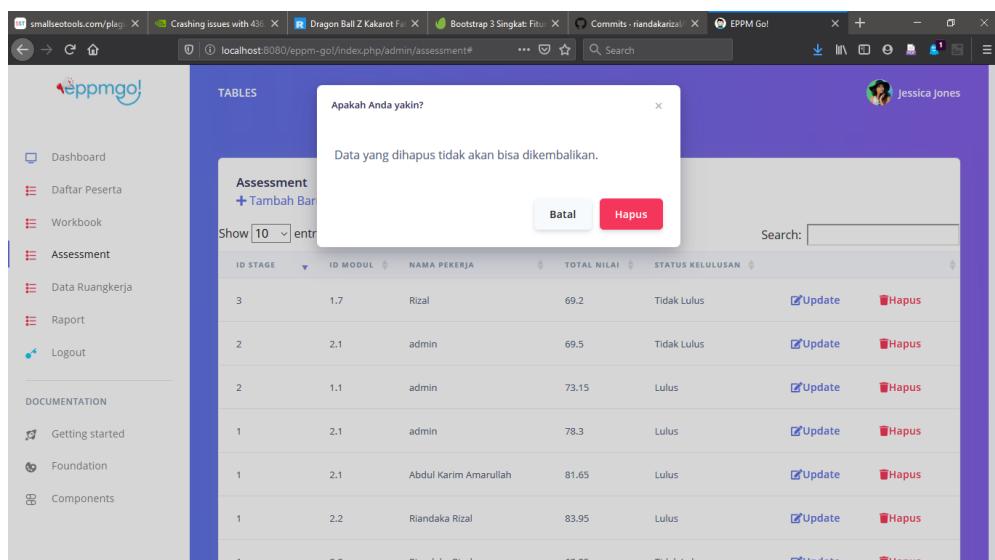
At the bottom of the form is a large green button labeled "Simpan".

Gambar 105 Halaman Form Update

ID STAGE	ID MODUL	NAMA PEKERJA	TOTAL NILAI	STATUS KELULUSAN	Actions
3	1.7	Rizal	69.2	Tidak Lulus	Update Hapus

Gambar 106 Perubahan Data Penilaian

Lalu apabila ternyata peserta atas nama rizal belum saatnya dilakukan *assessment* maka tinggal dihapus saja data dari peserta dengan nama rizal yang diberikan penilaian untuk modul 1.7 pada *stage* 3 dengan menekan tombol hapus berwarna merah di samping tombol *update*. Lalu apakah anda yakin ingin menghapus data ini? Apakah benar bahwa ini bukanlah suatu kesalahan? Maka akan keluar suatu jendela konfirmasi untuk menghapus data yang baru saja kita pilih seperti yang ditunjukkan pada gambar 107 berikut ini.



Gambar 107 Jendela Konfirmasi Hapus Data

Tekan saja tombol hapus untuk benar benar menghapus data ini dari aplikasi *website* dan juga *database*. Dan selesai, data sudah terhapus. Untuk menunjukkan bahwa datanya sudah benar benar terhapus, yaitu cari saja modul 1.7 pada kolom pencarian yang sudah disediakan. Dan hasilnya ditunjukkan pada gambar 108 berikut ini

Show	10	entries	Search:	1.7
ID STAGE	▲	ID MODUL	▼	NAMA PEKERJA
TOTAL NILAI	▼	STATUS KELULUSAN	▼	
No matching records found				
Showing 0 to 0 of 0 entries (filtered from 13 total entries)	Previous	Next		

Gambar 108 Data Sudah Terhapus

Dan ya, data sudah benar benar terhapus karena hasil pencarian dari modul 1.7 menunjukkan tidak ada data yang sesuai dengan *keyword* pada kolom pencarian. Dengan ini membuktikan bahwa untuk halaman menu *assessment* sudah dapat berfungsi dengan baik, karena tidak ditemukannya ada kesalahan ataupun *error* yang terjadi selama melakukan proses CRUD.

Selesai sudah pembuatan menu *assessment*, maka langkah selanjutnya yang harus dilakukan adalah membuat menu ruangkerja dimana ini akan meng-*import* data data ruangkerja. Karena pelatihan ini bekerja sama dengan ruangkerja, yang akan memberikan hasil penilaian *e-learning* dalam bentuk *file spreadsheet* atau yang biasa dikenal *file excel*. Kenapa diberikan fitur *import* pada menu ini? Karena data yang di-*update* sangat banyak dari ruangkerja perharinya, sehingga sangat tidak efisien apabila kita harus meng-*update* data yang berubah satu satunya.

Maka untuk memudahkan dan juga membuat proses *update* data ini menjadi lebih efisien, disediakanlah fitur *import*.

2.5. CRUD Menu Ruangkerja

Menu selanjutnya yang akan dibuat adalah menu ruangkerja. Dimana data data yang diterima admin akan dimasukkan ke dalam aplikasi *website* EPPM Go! ini. Kenapa harus ada menu ini? Karena pelatihan yang dijalankan bekerja sama dengan ruangkerja untuk modul pelatihan *e-learning* jadi para peserta diharuskan lulus terlebih dahulu dari pelatihan ruangkerja ini. Sebelum kita menuju penjelasan *tutorial* maka ada baiknya dijelaskan terlebih dahulu apa itu ruangkerja

Ruangkerja merupakan sebuah *platform* yang dikeluarkan oleh ruangguru yang bertujuan untuk meningkatkan kemampuan para pekerja di perusahaan tertentu. Dengan adanya pelatihan dari ruangkerja, maka perusahaan dapat menghemat banyak sekali *resources* yang dikeluarkan untuk pelatihan konvensional pada umumnya.



Gambar 109 Ruangkerja

Selain itu, ruangkerja memiliki keunggulan berupa beberapa fitur yang sangat menarik untuk digunakan karena ruangkerja memahami beragam kebutuhan terkait proses pelatihan di era modern ini. Berikut adalah fitur-fitur dari ruangkerja yang secara khusus disesuaikan untuk pengguna:

- a. *Mobile Learning*, pelatihannya dikemas untuk dapat diakses melalui *smartphone*
- b. *Learning Journey*, memiliki pendekatan belajar yang terstruktur dan sistematis
- c. *Powerful User Interface*, UI/UX yang menarik agar peserta nyaman
- d. *Collaboration*, adanya forum untuk diskusi yang membuka ruang untuk kolaborasi
- e. *Leaderboard*, adanya peringkat untuk menambah semangat belajar
- f. *Notifications*, informasi penugasan pelatihan atau *reminder* akan dikirim secara otomatis melalui email atau *push notification*
- g. *Certificate*, peserta pelatihan akan mendapat sertifikat bila telah menyelesaikan pelatihan dengan baik

Selain untuk pengguna, pastinya setiap pelatihan memiliki administrator yang bertugas untuk melakukan proses pengawasan dari pelatihan yang sedang dijalankan. Maka ruangkerja pun memfasilitasi fitur-fitur yang diberikan untuk administrator dalam proses pengawasan:

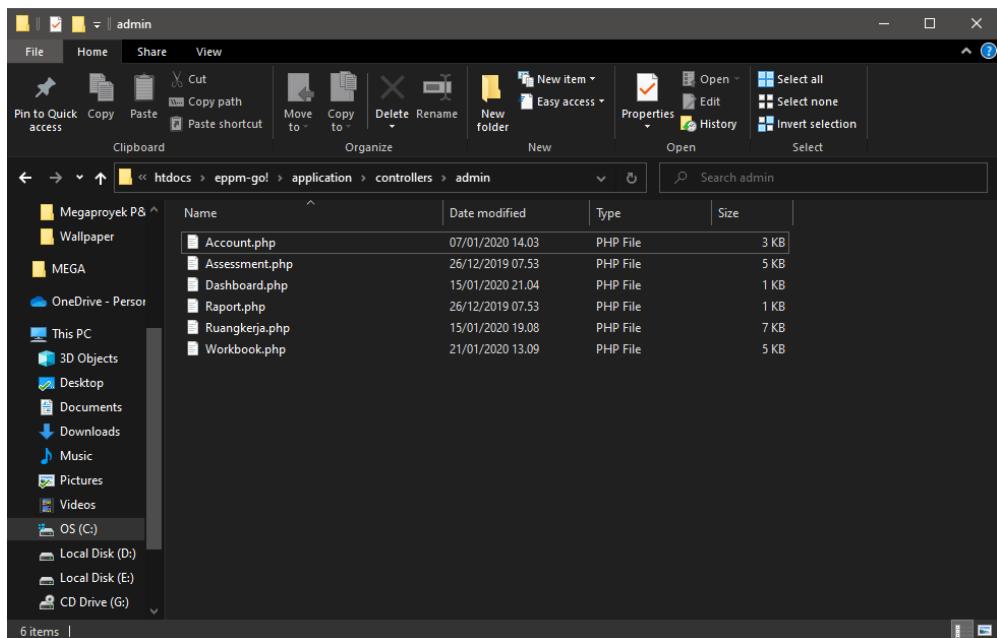
- a. *Monitoring Dashboard*, pemantauan pelatihan yang bersifat menyeluruh dan *seamless*.
- b. *Create Course*, dapat membuat kelas pelatihan yang sesuai dengan kebutuhan perusahaan
- c. *User Grouping & Enrollment*, adanya fitur untuk *enrollment* peserta pelatihan secara masal

Selain fitur-fitur yang ditawarkan, keunggulan dari ruangkerja adalah layanan yang mengcover keseluruhan proses pelatihan dari awal hingga pelatihan itu selesai, maka tak heran banyak perusahaan multinasional yang menggunakan jasa dari ruang kerja. Berikut adalah layanan yang ditawarkan oleh ruangkerja:

- a. *Pre-Onboarding*, yang meliputi analisis kebutuhan, penetapan tujuan, dan juga *strategy mapping*
- b. *Learning Design*, yang meliputi kebutuhan konseptual, kurasi konten, dan juga *storyboarding*
- c. *Content Development*, yang meliputi produksi video dan juga proses penyuntingan dan animasi
- d. *Implementation*, yang meliputi uji penerimaan pengguna, dan juga *go live*
- e. *Change Management Support*, yang meliputi *training of trainer*, *customer success support*, dan juga *data analysis*
- f. *Technical Support*, yang meliputi layanan pengguna 24 jam seminggu, dan juga perbaikan operasional

Setelah mendapatkan penjelasan mengenai apa itu ruangkerja, alangkah baiknya kita melanjutkan materi kita. Yaitu membuat CRUD menu ruangkerja dengan fitur *import* data csv.

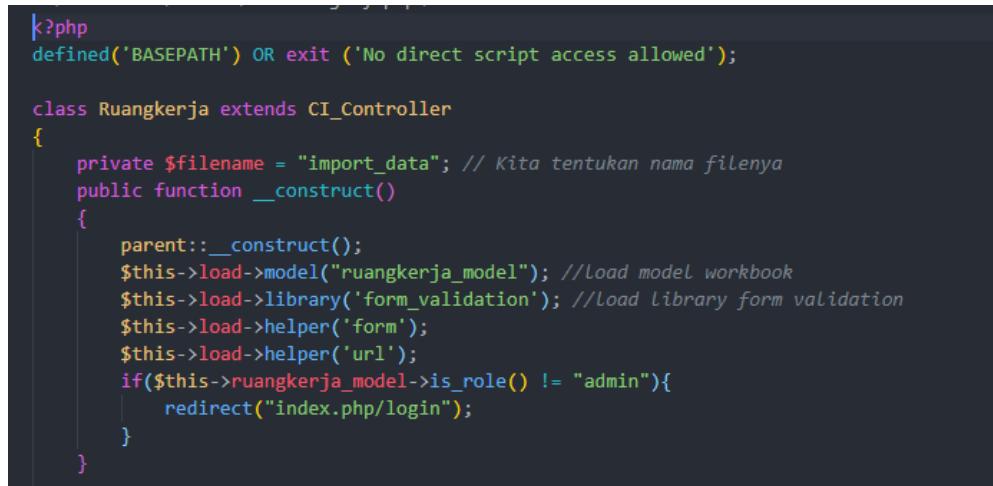
Langkah pertama adalah kita harus membuat *file* Ruangkerja.php, di dalam folder admin pada direktori *controllers* seperti yang ditunjukkan pada gambar 110 berikut ini.



Gambar 110 File Ruangkerja

Di dalam *file* Ruangkerja.php kita akan mengisikannya dengan sintak sintak untuk *controllers* yang berfungsi untuk menghubungkan *models* dan juga *views*. Selain itu, *controllers* berfungsi untuk melakukan pemrosesan dari data data yang dilemparkan oleh *models*.

Selanjutnya langsung saja kita isikan *file* Ruangkerja.php dengan isian yang ditunjukkan oleh gambar 111 di bawah ini.



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

class Ruangkerja extends CI_Controller
{
    private $filename = "import_data"; // Kita tentukan nama filenya
    public function __construct()
    {
        parent::__construct();
        $this->load->model("ruangkerja_model"); //Load model workbook
        $this->load->library('form_validation'); //Load Library form validation
        $this->load->helper('form');
        $this->load->helper('url');
        if($this->ruangkerja_model->is_role() != "admin"){
            redirect("index.php/login");
        }
    }
}
```

Gambar 111 Controllers Ruangkerja.php

Pada gambar 111 berikut, kita terlebih dahulu membuat sebuah *class* dengan nama Ruangkerja dengan menggunakan *library* dari CI_Controller. Selanjutnya adalah kita menginisiasikan sebuah variabel yang secara privat, yang artinya hanya akan digunakan oleh variabel itu sendiri saja. Berikan nama dari variabel tersebut \$filename. Variabel ini berfungsi untuk menentukan nama *file* dari csv dan excel yang akan kita *import* nanti. Kemudian kita membuat sebuah *function __construct()* yang di dalam *function* itu kita me-*load* models ruangkerja_model, me-*load* library form validasi dan juga *helper*-nya. Lalu kita bila ada pengguna yang akan menggunakan menu ruangkerja maka dicek terlebih dahulu *role* dari pengguna tersebut. Jika *role* yang masuk bukan admin maka akan diarahkan kembali menuju *login*.

```
public function index()
{
    $data["ruangkerja"] = $this->ruangkerja_model->getAll(); //ambil data dari model
    // print_r('tes'); die;
    $this->load->view("admin/ruangkerja/list_ruangkerja", $data); //Load view data model ke workbook
}
```

Gambar 112 Controllers Ruangkerja.php

Langkah selanjutnya adalah memberi isian Ruangkerja.php seperti yang ditunjukkan pada gambar 112 di atas. Pertama kita harus membuat sebuah *function index()* di dalamnya inisiasikan variabel yang berfungsi untuk mengambil semua data dari tabel ruangkerja di *database*. Lalu tampilkan data tersebut pada halaman *list_ruangkerja*. Setelah membuat *function index()* maka langkah selanjutnya yang harus ditempuh adalah kita membuat *function edit()* yang di dalamnya inisiasikan variabel yang berguna untuk mengambil data berdasarkan id yang dipilih, lalu tampilkan data dari id yang dipilih pada halaman *edit_ruangkerja*.

Selanjutnya adalah membuat *function aksi_edit()* dengan menyimpan data yang baru saja kita ubah dan simpan ke dalam *database*. Pertama inisiasikan variabel \$id yang akan menyimpan data dari id-nya. Lalu kita inisiasikan kembali variabel \$data dalam bentuk *array* yang akan menyimpan data yang banyak ke dalam satu variabel saja sebelum disimpan ke dalam *database*. Setelah itu masukkan perubahan yang terjadi ke dalam *database* dan bila datanya berhasil disimpan maka akan diarahkan kembali ke halaman ruangkerja. Seperti yang ditunjukkan pada gambar 113 berikut ini.

```

public function edit($id=null)
{
    $data["ruangkerja"] = $this->ruangkerja_model->getById($id); //mengambil data berdasarkan id
    if (!$data["ruangkerja"]) show_404();// jika tidak ada show error
    $this->load->view("admin/ruangkerja/edit_ruangkerja", $data); //load edit form workbook
}

public function aksi_edit()
{
    $id = $this->input->post('ruangkerja_id');

    $data = array (
        "ruangkerja_id" => $this->input->post("ruangkerja_id"),
        "employee_id" => $this->input->post("employee_id"),
        "employee_name" => $this->input->post("employee_name"),
        "user_email" => $this->input->post("user_email"),
        "start_time" => $this->input->post("start_time"),
        "posttest_score" => $this->input->post("posttest_score"),
        "status" => $this->input->post("status"),
        "deadline_course" => $this->input->post("deadline_course"),
        "clear_time" => $this->input->post("clear_time"),
        "directorate" => $this->input->post("directorate")
    );
    // print_r($data); die;
    $this->ruangkerja_model->update($id,$data);
    $this->session->set_flashdata('success', 'Data Anda Berhasil Diupdate');
    redirect('index.php/admin/ruangkerja');
}

```

Gambar 113 Controllers Ruangkerja.php

Setelah kita membuat *function* seperti yang ditunjukkan oleh gambar 113 di atas, maka untuk melengkapi lagi *file* Controllers yang sedang kta buat tambahkan *function* untuk *delete* dan juga *import file excel* dan *csv*. Seperti yang ditunjukkan oleh gambar 114 dan juga gambar 115 di bawah ini

```

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->ruangkerja->delete($id)){
        redirect('index.php/admin/ruangkerja');
    }
}

```

Gambar 114 Controllers Ruangkerja.php

Pada gambar 114, kita membuat *function* *delete()* yang berfungsi untuk menghapus data dari *database* dan juga aplikasi *website* dengan mengambil id dari data yang dipilih.

```

public function formImport(){
    $data = array(); // Buat variabel $data sebagai array

    if(isset($_POST['preview'])){ // Jika user menekan tombol Preview pada form
        // Lakukan upload file dengan memanggil function upload yang ada di ModelAdmin.php
        $upload = $this->ruangkerja_model->upload_file($this->filename);

        if($upload['result'] == "success"){ // Jika proses upload sukses
            // Load plugin PHPExcel nya
            include APPPATH.'third_party/PHPExcel/PHPExcel.php';

            $excelreader = new PHPExcel_Reader_Excel2007();
            $loadexcel = $excelreader->load('excel/'.$this->filename.'.xlsx'); // Load file yang tadi diupload
            $sheet = $loadexcel->getActiveSheet()->toArray(null, true, true ,true);

            // Masukan variabel $sheet ke dalam array data yang nantinya akan dikirim ke file form.php
            // Variabel $sheet tersebut berisi data-data yang sudah diinput di dalam excel yang sudah diupload
            $data['sheet'] = $sheet;
        }else{ // Jika proses upload gagal
            $data['upload_error'] = $upload['error']; // Ambil pesan error uploadnya untuk dikirim ke form
        }
    }
    $this->load->view('admin/ruangkerja/import_ruangkerja', $data);
}

```

Gambar 115 Controllers Ruangkerja.php

Lalu selanjutnya adalah membuat *function* formImport() lalu membuat variabel \$data dimana datanya akan diubah menjadi sebuah *array*. Lalu jika pengguna menekan tombol *import* lakukan *upload file* dengan memanggil *function upload* pada models. Jika proses *upload* selesai maka akan me-*load plugin* dari excelnya. Jika berhasil, maka tampilkan data data tersebut dengan menampilkan halaman import_ruangkerja.

Selanjutnya adalah membuat *function* import() yang akan berfungsi untuk memasukkan data dari kolom dan baris yang ada di excel yang akan kita lakukan *import*. Untuk lebih jelasnya silakan perhatikan gambar 116 di bawah ini.

```
public function formImport(){
    $data = array(); // Buat variabel $data sebagai array

    if(isset($_POST['preview'])){ // Jika user menekan tombol Preview pada form
        // Lakukan upload file dengan memanggil function upload yang ada di ModelAdmin.php
        $upload = $this->ruangkerja_model->upload_file($this->filename);

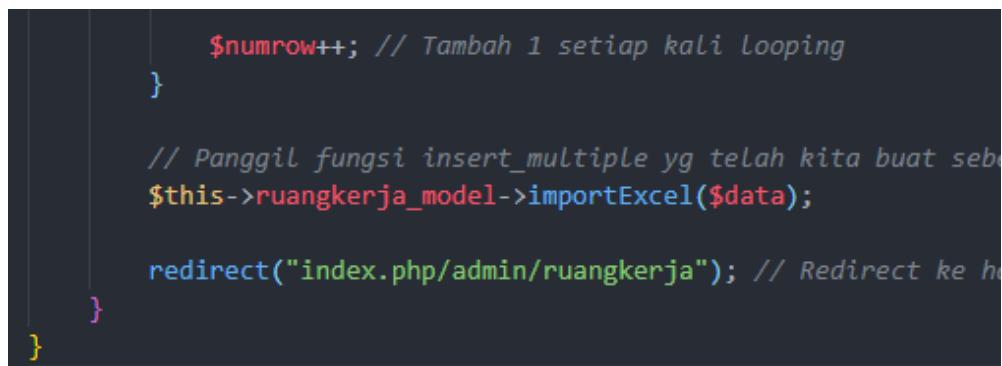
        if($upload['result'] == "success"){ // Jika proses upload sukses
            // Load plugin PHPExcel nya
            include APPPATH.'third_party/PHPExcel/PHPExcel.php';

            $excelreader = new PHPExcel_Reader_Excel2007();
            $loadexcel = $excelreader->load('excel/'.$this->filename.'.xlsx'); // Load file yang tadi diupload
            $sheet = $loadexcel->getActiveSheet()->toArray(null, true, true ,true);

            // Masukan variabel $sheet ke dalam array data yang nantinya akan di kirim ke file form.php
            // Variabel $sheet tersebut berisi data-data yang sudah diinput di dalam excel yang sudah diupload
            $data['sheet'] = $sheet;
        }else{ // Jika proses upload gagal
            $data['upload_error'] = $upload['error']; // Ambil pesan error uploadnya untuk dikirim ke form
        }
    }
    $this->load->view('admin/ruangkerja/import_ruangkerja', $data);
}
```

Gambar 116 Controllers Ruangkerja.php

Setelahnya kita akan lakukan *looping* dimana apabila ada banyak data baru yang ingin diimport maka akan dilakukan *looping* untuk memasukkan data tersebut satu persatu ke dalam *database*. Jika berhasil maka akan diarahkan kembali ke menu ruangkerja. Seperti yang ditunjukkan pada gambar 117 berikut ini.



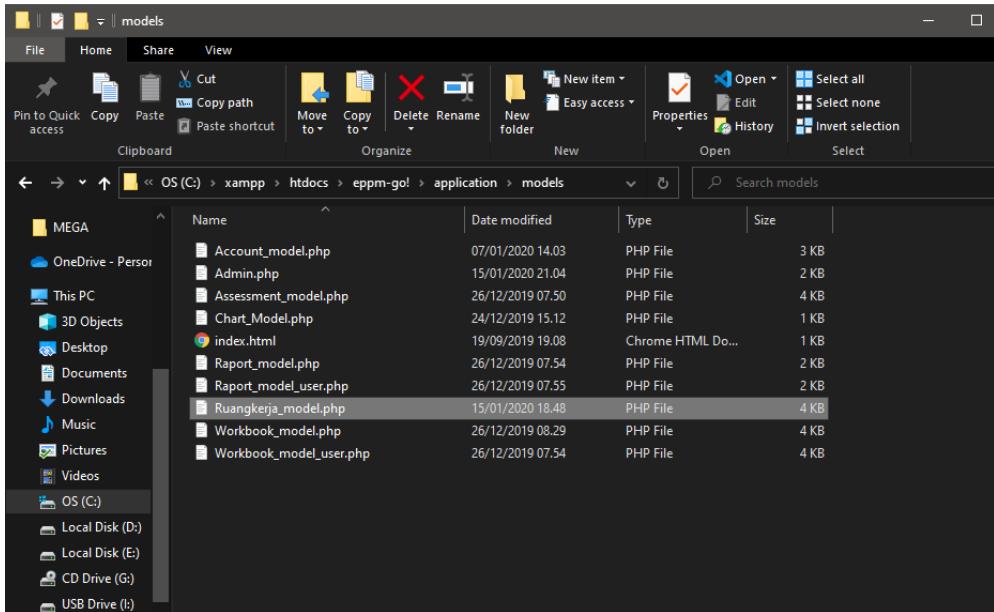
```
        $numrow++; // Tambah 1 setiap kali looping
    }

    // Panggil fungsi insert_multiple yg telah kita buat sebelumnya
    $this->ruangkerja_model->importExcel($data);

    redirect("index.php/admin/ruangkerja"); // Redirect ke halaman awal
}
}
```

Gambar 117 Controllers Ruangkerja.php

Dengan ini, selesai sudah tahapan tahapan dan komponen yang harus dibuat untuk menu ruangkerja. Maka langkah selanjutnya untuk menyempurnakan menu ruangkerja ini adalah membuat ruangkerja_model yang berfungsi untuk melakukan manipulasi datanya. Langsung saja buat terlebih dahulu sebuah *file* dengan nama ruangkerja_model.php pada folder admin di direktori models dari *localhost* kita. Seperti yang ditunjukkan oleh gambar 118 di bawah ini.



Gambar 118 Ruangkerja_model.php

Jika sudah dibuat *file*-nya langsung kita berikan isian untuk *file* ruangkerja_model.php tersebut, dengan membuat sebuah *class* terlebih dahulu dengan nama Ruangkerja_model dengan menggunakan *library* dari CI_Model.

Lanjut kita menginisiasikan terlebih dahulu variabel variabel yang akan kita gunakan, jika sudah buatlah *function* dengan nama getAll() yang akan mengambil seluruh data di dalam *database* seperti yang dapat kita lihat pada gambar 119 berikut ini

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

class Ruangkerja_model extends CI_Model
{
    private $_table = "ruangkerja";

    public $ruangkerja_id;
    public $employee_id;
    public $employee_name;
    public $user_email;
    public $start_time;
    public $posttest_score;
    public $status;
    public $deadline_course;
    public $clear_time;
    public $directorate;

    public function getAll()
    {
        return $this->db->get($this->_table)->result();
        // print_r('tes'); die;
    }
}
```

Gambar 119 Models Ruangkerja_model.php

Jika sudah selesai, maka langsung kita lanjutkan ke tahapan selanjutnya dalam pembuatan Ruangkerja_model.php seperti yang ditunjukkan pada gambar 120 di bawah ini, dimana kita akan membuat *function* upload_file() agar dapat mengimport file excel yang akan digunakan untuk menu ruangkerja ini.

```

// Fungsi untuk melakukan proses upload file
public function upload_file($filename){
    $this->load->library('upload'); // Load Librari upload

    $config['upload_path'] = './excel/';
    $config['allowed_types'] = 'xlsx';
    $config['max_size'] = '2048';
    $config['overwrite'] = true;
    $config['file_name'] = $filename;

    $this->upload->initialize($config); // Load konfigurasi uploadnya
    if($this->upload->do_upload('file')){ // Lakukan upload dan Cek jika proses upload berhasil
        // Jika berhasil :
        $return = array('result' => 'success', 'file' => $this->upload->data(), 'error' => '');
        return $return;
    }else{
        // Jika gagal :
        $return = array('result' => 'failed', 'file' => '', 'error' => $this->upload->display_errors());
        return $return;
    }
}

```

Gambar 120 Models Ruangkerja_model.php

Setelah membuat *function*-nya maka kita *load* terlebih dahulu *library upload* dan membuat konfigurasi untuk disimpan dimana *file*-nya lalu ekstensi apa yang diperbolehkan untuk *diupload*, ukuran maksimum *file* yang boleh *diupload* lalu apakah bisa *dioverwrite* dan nama *file*-nya.

Setelah membuat konfigurasinya, maka gunakan konfigurasi tersebut. Lalu lakukan *upload* dan cek apabila proses *upload* berhasil berikan pesan berhasil dan bila gagal maka berikan juga pesan gagal dengan memunculkan notifikasi galat.

```

public function autoIncrementRuangkerja(){
    return $this->db->query("ALTER TABLE ruangkerja AUTO_INCREMENT =1;");
}

public function importExcel($data)
{
    return $this->db->insert_batch('ruangkerja', $data);
}

public function getById($id)
{
    return $this->db->get_where($this->_table, ["ruangkerja_id" => $id])->row();
}

public function get_data_stage()
{
    $query = $this->db->get('stage');
    return $query;
}

```

Gambar 121 Models Ruangkerja_model.php

Jika sudah selesai membuat *function* upload_file() maka kita membuat *function* autoIncrementRuangkerja() yang mana berfungsi untuk menambahkan atau merubah id dari ruangkerja secara otomatis. Seperti yang dapat dilihat pada gambar 121 terdapat *query* untuk menambahkan id dari ruangkerja +1.

Lanjut langkah selanjutnya adalah membuat *function* importExcel() yang berfungsi untuk melakukan *import excel* ke dalam tabel ruangkerja di dalam *database*. Data data yang ada pada excel akan dimasukkan ke dalam *database*. Lalu lanjut kita membuat *function* getById yang akan mengambil id berdasarkan data yang dipilih nanti, dimana id-nya akan diambil melalui tabel ruangkerja di *database*. Tak lupa kita membuat *function* get_data_stage() untuk mengambil data *stage* pada tabel *stage* di dalam *database*.

```
public function update($id, $data)
{
    $this->db->where('ruangkerja_id', $id);
    $this->db->update("ruangkerja", $data);
}

public function delete($id)
{
    // $this->_deleteImage($id);
    return $this->db->delete($this->_table, array("ruangkerja_id" => $id));
}

//fungsi cek Level
function is_role()
{
    return $this->session->userdata('role');
}

}
```

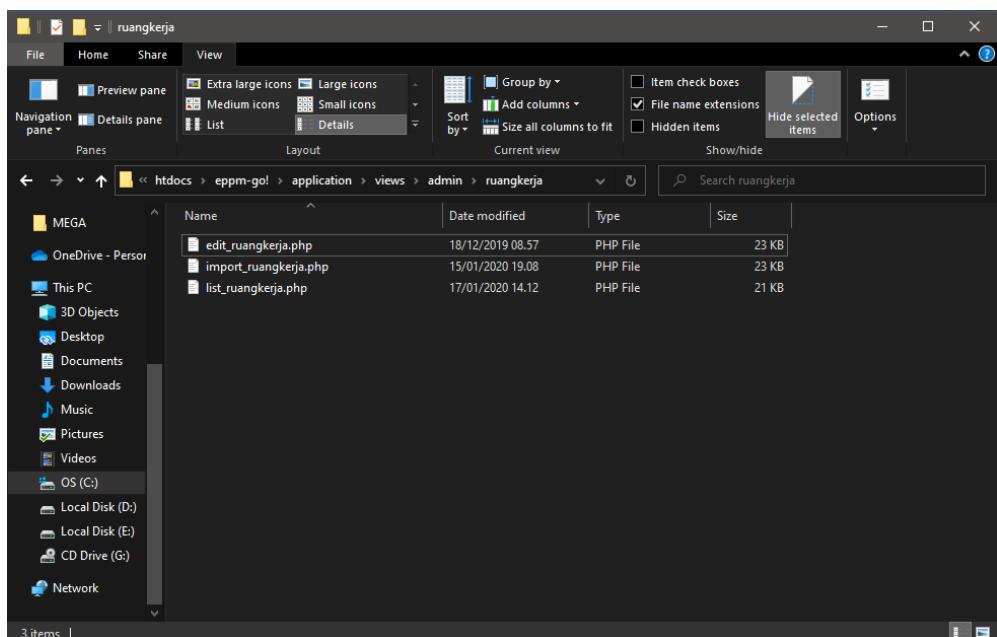
Gambar 122 Models Ruangkerja_model.php

Sekarang kita akan membuat *function* yang berfungsi untuk menyimpan perubahan data yang akan kita masukkan ke dalam *database*. Yaitu kita membuat *function* update() yang akan meyimpan segala perubahan data dengan mengambil id sebagai tanda pengenal dan juga meyimpan perubah dari setiap *field*.

Lalu kita membuat *function* delete() yang akan menghapus data dari *database* dengan mengambil id dari data yang kita pilih lalu akan menghapus data tersebut. Tak lupa kita juga membuat *function* untuk mengecek *role* yang masuk ke dalam sistem, sehingga akan mengecek *role* dari *session* yang sedang aktif.

Untuk lebih jelasnya maka dapat melihat beberapa baris sintak yang ditunjukkan oleh gambar 122 di atas.

Dengan ini selesai sudah langkah langkah yang diperlukan untuk membuat models Ruangkerja_model.php. Jadi tinggal satu langkah lagi sebelum menu ruangkerja dapat kita coba, yaitu membuat views atau tampilannya. Maka langsung saja kita membuat folder ruangkerja pada folder admin di direktori views. Ini adalah *path* direktorinya C:\xampp\htdocs\eppm-go!\application\views\admin\ruangkerja. Lalu di dalamnya buatlah 3 *file* dengan nama list_ruangkerja.php, edit_ruangkerja.php, dan juga import_ruangkerja.php seperti yang ditunjukkan oleh gambar 123 di bawah ini.



Gambar 123 File Ruangkerja

Lalu setelah dibuat ketiga *file* tersebut, langsung saja kita ke sintak inti dari *file* list_ruangkerja.php seperti yang akan ditunjukkan oleh gambar 124 berikut ini, dimana kita akan membuat tabel yang akan menampung atau mewadahi data data yang akan ditampilkan.

```
<!-- Table -->


### Ruangkerja

Import



| ID Ruangkerja | Nomor Pekerja | Nama Pekerja | Email | Start Time | Post Test Score | Status | Deadline Course | Clear Time | Direktorat | </th> |
|---------------|---------------|--------------|-------|------------|-----------------|--------|-----------------|------------|------------|-------|
|---------------|---------------|--------------|-------|------------|-----------------|--------|-----------------|------------|------------|-------|


```

Gambar 124 Views list_ruangkerja.php

Setelah selesai dibuat tabelnya, saatnya kita panggil datanya sesuai dengan kolom dari *field* setiap datanya. Sama seperti yang sudah dilakukan pada menu menu sebelumnya, maka kita juga membuat perulangan dengan metode *foreach* untuk menampilkan datanya sampai habis

Untuk itu, lakukan *foreach* dengan datanya dengan mengambil data tersebut dari tabel ruangkerja melalui inisiasi variabel \$ruangkerja pada models sebelumnya. Untuk lebih detailnya maka dapat dilihat pada gambar 125 berikut ini.

```
<tbody>
    <?php foreach ($ruangkerja as $ruangkerja): ?>
    <tr>
        <td width="150">
            <?php echo $ruangkerja->ruangkerja_id ?>
        </td>
        <td>
            <?php echo $ruangkerja->employee_id ?>
        </td>
        <td>
            <?php echo $ruangkerja->employee_name ?>
        </td>
        <td>
            <?php echo $ruangkerja->user_email ?>
        </td>
        <td>
            <?php echo $ruangkerja->start_time ?>
        </td>
        <td>
            <?php echo $ruangkerja->posttest_score ?>
        </td>
        <td>
            <?php echo $ruangkerja->status ?>
        </td>
        <td>
            <?php echo $ruangkerja->deadline_course ?>
        </td>
        <td>
            <?php echo $ruangkerja->clear_time ?>
        </td>
        <td>
            <?php echo $ruangkerja->directorate ?>
        </td>
```

Gambar 125 Views list_ruangkerja.php

Lalu jangan lupa, kita juga harus memberika tombol *update* dan juga hapus jikalau sewaktu waktu ada data yang perlu diubah atau dihapus. Lalu jika sudah selesai akhiri perulangan *foreach*-nya seperti yang ditunjukkan pada gambar 126 berikut ini.

```
<td width="250">
    <a href="php echo site_url('admin/ruangkerja/edit/'.$ruangkerja-&gt;ruangkerja_id);?"&gt;
        class="btn btn-small"&gt;&lt;i class="fas fa-edit"&gt;&lt;/i&gt;Update&lt;/a&gt;
    &lt;a onclick="deleteConfirm('<?php echo site_url('admin/ruangkerja/delete/'.$ruangkerja-&gt;ruangkerja_id);?'" href="#" class="btn btn-small text-danger"&gt;&lt;i class="fas fa-trash"&gt;&lt;/i&gt;Hapus&lt;/a&gt;
&lt;/td&gt;
&lt;/tr&gt;
&lt;?php endforeach; ?&gt;</pre
```

Gambar 126 Views list_ruangkerja.php

Karena, masih menggunakan DataTables jadi kita harus menggunakan sintaks jQuery untuk menggunakan *resources* dari DataTables. Setelah itu, untuk sintaks jQuery dari DataTables kita taruh dibawah setelah sintaks pembuatan tabel. Jangan lupa untuk menyisipkan tag <script> karena masih satu bagian dari JavaScript. Selain itu, karena kita menggunakan *resources* dari luar ada beberapa cara yang bisa digunakan, yaitu dengan cara:

1. Mengunduh *library*-nya dan disimpan pada folder *assets* yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini akan menggunakan *local resource* dari PC atau laptop kita.

2. Menyalin *link CDN* (*Content Delivery Network*) yang akan dipanggil di atas untuk CSS di bagian tag *head* dan di bawah untuk JavaScript di bagian tag *footer*. Kekurangan cara ini hanya memerlukan akses internet. *Link CDN* DataTables untuk CSS adalah //cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css dan *link CDN* DataTables untuk JavaScript adalah //cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js

```
<script>$(document).ready( function () {
|   $('table').DataTable();
} );
</script>
```

Gambar 127 Views list_ruangkerja.php

Untuk tombol hapus, kita harus selalu menggunakan konfirmasi sebelum dilakukannya penghapusan data. Ini berguna untuk mencegah dan meminimalisir terjadinya penghapusan data secara tidak sengaja, dan kita menggunakan modal untuk konfirmasinya seperti yang ditunjukkan gambar 128 di bawah ini

```
<script>
function deleteConfirm(url){
  $('#btn-delete').attr('href', url);
  $('#deleteModal').modal();
}
</script>
```

Gambar 128 Views list_ruangkerja.php

Maka kita coba terlebih dahulu, apakah tabelnya berhasil dibuat? Apakah datanya juga terpanggil? Maka kita buka *browser* dan ketikkan url `localhost:8080/eppm-go!/login`. Setelah melakukan login maka langsung menuju menu ruangkerja

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'localhost:8080/eppm-go!/admin/ruangkerja'. The page content is a table titled 'Ruangkerja' with the following data:

ID RUANGKERJA	NOMOR PEKERJA	NAMA PEKERJA	EMAIL	START TIME	POST TEST
1	755303	Abdul Karim Amarullah	abdul.amarullah@pertamina.com	2019-06-13 02:21:30	100
2	718027	Abdullah	abdullah2708@pertamina.com	0000-00-00 00:00:00	0
3	750787	Achmad Hattary	achmad.hattary@pertamina.com	0000-00-00 00:00:00	0
4	10026617	Achmad Riza Pamula	achmad.pamula@pertamina.com	0000-00-00 00:00:00	0
5	755304	Achmad Syafiudin	achmad.syafiudin@pertamina.com	2019-06-12 05:56:43	73
6	714163	Ali Dwiyono Haritsyah	ad_haritsyah@pertamina.com	0000-00-00 00:00:00	0

Gambar 129 Views Menu Ruangkerja

Dapat dilihat pada gambar 129 di atas, tabel dan datanya berhasil dibuat dan dipanggil ke dalam menu ruangkerja. Maka hal selanjutnya adalah kita harus membuat form *update* yang akan muncul bila menekan tombol *update* dan juga membuat form *import* yang akan muncul bila menekan tombol *import*.

Maka kita lanjut ke *file* selanjutnya, yaitu edit_ruangkerja.php dimana di dalam *file* ini kita akan membuat form *update* untuk merubah data yang ingin kita rubah. Langsung ke inti pembahasan, maka kita akan membuat form untuk setiap *field* dari ruangkerja. Seperti yang ditunjukkan pada gambar 130 di bawah ini.

```
<!-- Table -->


<div class="card-header">
    <a href="php echo site_url('admin/ruangkerja/') ?&gt;"<i class="fas fa-arrow-left"></i>
    Kembali</a>
  </div>
  <div class="card-body">

    <form action="php echo base_url();?&gt;index.php/admin/ruangkerja/aksi_edit" method="post" enctype="multipart/fo

    &lt;input type="hidden" name="ruangkerja_id" value="<?php echo $ruangkerja-&gt;ruangkerja_id?&gt;" /&gt;

    &lt;div class="form-group"&gt;
      &lt;label for="name"&gt;No. Pekerja&lt;/label&gt;
      &lt;input class="form-control &lt;?php echo form_error('employee_id') ? 'is-invalid':'?&gt;" type="text" name="employee_id" min="0" placeholder="No. Pekerja" value="<?php echo $ruangkerja-&gt;employee_i
      &lt;div class="invalid-feedback"&gt;
        &lt;?php echo form_error('employee_id') ?&gt;
      &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
      &lt;label for="employee_name"&gt;Nama Pekerja&lt;/label&gt;
      &lt;input class="form-control &lt;?php echo form_error('employee_name') ? 'is-invalid':'?&gt;" type="text" name="employee_name" min="0" placeholder="Nama Pekerja" value="<?php echo $ruangkerja-&gt;employe
      &lt;div class="invalid-feedback"&gt;
        &lt;?php echo form_error('employee_name') ?&gt;
      &lt;/div&gt;
    &lt;/div&gt;
  &lt;/div&gt;</pre



Gambar 130 Views edit_ruangkerja.php



Dapat dilihat pada gambar 130 di atas, kita membuat terlebih dahulu textbox agar dapat menampung datanya. Lalu jika textbox sudah terbuat maka akan langsung dipanggil datanya untuk dapat diubah di dalam textbox.



125


```

```

<div class="form-group">
    <label for="user_email">Email</label>
    <input class="form-control <?php echo form_error('user_email') ? 'is-invalid':'' ?>" type="text" name="user_email" min="0" placeholder="Email" value="<?php echo $ruangkerja->user_email ?>" />
    <div class="invalid-feedback">
        <?php echo form_error('user_email') ?>
    </div>
</div>

<div class="form-group">
    <label for="start_time">Start Time</label>
    <input class="form-control <?php echo form_error('start_time') ? 'is-invalid':'' ?>" type="date" name="start_time" min="0" placeholder="Start Time" value="<?php echo $ruangkerja->start_time ?>" />
    <div class="invalid-feedback">
        <?php echo form_error('start_time') ?>
    </div>
</div>

<div class="form-group">
    <label for="posttest_score">Post Test Score</label>
    <input class="form-control <?php echo form_error('posttest_score') ? 'is-invalid':'' ?>" type="text" name="posttest_score" min="0" placeholder="Post Test Score" value="<?php echo $ruangkerja->posttest_score ?>" />
    <div class="invalid-feedback">
        <?php echo form_error('posttest_score') ?>
    </div>
</div>

<div class="form-group">
    <label for="status">Status</label>
    <input class="form-control <?php echo form_error('status') ? 'is-invalid':'' ?>" type="text" name="status" min="0" placeholder="Status" value="<?php echo $ruangkerja->status ?>" required/>
    <div class="invalid-feedback">
        <?php echo form_error('status') ?>
    </div>

```

Gambar 131 Views edit_ruangkera.php

Sama seperti yang kita lakukan sebelumnya, pada gambar 131 di atas masih merupakan kelanjutan dari langkah yang sedang kita lakukan pada gambar 130 sebelumnya.

```

<div class="form-group">
    <label for="deadline_course">Deadline Course</label>
    <input class="form-control <?php echo form_error('deadline_course') ? 'is-invalid':'' ?>" type="date" name="deadline_course" min="0" placeholder="Deadline Course" value="<?php echo $ruangkerja->de
    <div class="invalid-feedback">
        <?php echo form_error('deadline_course') ?>
    </div>
</div>

<div class="form-group">
    <label for="clear_time">Clear Time</label>
    <input class="form-control <?php echo form_error('clear_time') ? 'is-invalid':'' ?>" type="date" name="clear_time" min="0" placeholder="Clear Time" value="<?php echo $ruangkerja->clear_time ?
    <div class="invalid-feedback">
        <?php echo form_error('clear_time') ?>
    </div>
</div>

<div class="form-group">
    <label for="directorat">Direktorat</label>
    <input class="form-control <?php echo form_error('directorat') ? 'is-invalid':'' ?>" type="text" name="directorat" min="0" placeholder="Direktorat" value="<?php echo $ruangkerja->directorat
    <div class="invalid-feedback">
        <?php echo form_error('directorat') ?>
    </div>
</div>

</div>

<input class="btn btn-success" type="submit" name="btn" value="Simpan" />
</form>

```

Gambar 132 Views edit_ruangkerja.php

Setelah semua *textbox* untuk setiap *field* dari tabel ruangkerja dibuat, maka jangan lupa untuk membuat tombol simpan dimana yang akan berfungsi untuk menyimpan perubahan yang baru saja kita lakukan pada form *update* ini. Untuk itu selesaikan sudah tahapan yang diperlukan untuk pembuatan form *update* maka kita akan lakukan ujicoba ala ala dengan mengetik url localhost:8080/eppm-go!/login pada *browser* dan lakukan *login*. Setelah itu masuk ke menu ruangkerja dan pilih satu data untuk dilakukan *update* seperti yang ditunjukkan oleh gambar 133 berikut ini.

The screenshot shows a web-based application interface for managing employee data. On the left, there is a sidebar with a blue header containing the logo 'eppmgo!' and several menu items: Dashboard, Daftar Peserta, Workbook, Assessment, Data Ruangkerja, Raport, Logout, DOCUMENTATION, Getting started, Foundation, and Components. The main content area has a purple header with the text 'TABLES' and a back arrow labeled 'Kembali'. Below the header, there is a form with the following fields:

- No. Pekerja*: Input field containing '718027'.
- Nama Pekerja: Input field containing 'Abdullah'.
- Email: Input field containing 'abdullah2708@pertamina.com'.
- Start Time: Input field containing 'mm / dd / yyyy'.
- Post Test Score: Input field containing an empty value.

Gambar 133 Views Form Update

Dapat dilihat, kita sudah membuka form *update* dan kita akan coba lakukan perubahan data dan akan kita coba simpan

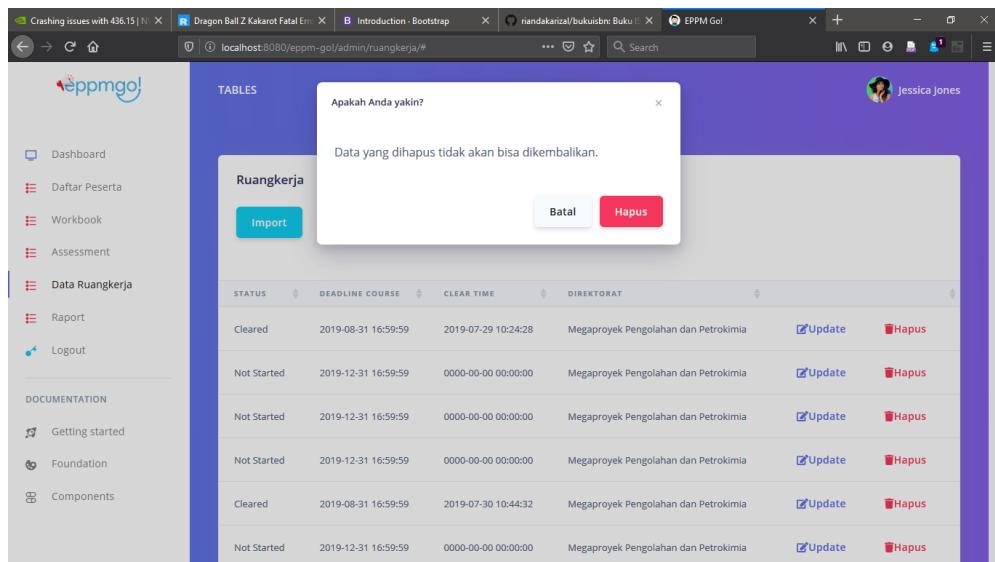
The screenshot shows a web-based application interface for managing course data. On the left, there is a sidebar with a blue header containing the logo 'eppmgo!' and several menu items: Dashboard, Daftar Peserta, Workbook, Assessment, Data Ruangkerja, Raport, Logout, DOCUMENTATION, Getting started, Foundation, and Components. The main content area has a purple header with the text 'Post Test Score' and a back arrow labeled 'Kembali'. Below the header, there is a form with the following fields:

- Status: Input field containing 'Not Started'.
- Deadline Course: Input field containing 'mm / dd / yyyy'.
- Clear Time: Input field containing 'mm / dd / yyyy'.
- Direktorat: Input field containing 'Megaprojek Pengolahan dan Petrokimia'.

A large green button at the bottom right of the form is labeled 'Simpan'.

Gambar 134 Views Form Update

Dan datanya berhasil kita simpan ke dalam *database*. Kita coba juga hapus salah satu data dengan menekan tombol hapus seperti yang ditunjukkan pada gambar 135 berikut ini.



Gambar 135 Views Tombol Hapus

Dan akan muncul notifikasi untuk mengkonfirmasi apakah datanya benar benar akan dihapus? Jika iya maka hapus saja datanya, tetapi karena ini merupakan data yang penting, jadi tidak akan ada data dari ruangkerja yang akan dihapus.

Setelah selesai membuat form *update*, maka yang akan kita lakukan selanjutnya adalah membuat form *import* maka langsung saja kita buka *file* dari import_ruangkerja.php yang sebelumnya telah kita buat, dan langsung saja ke bagian inti dari import_ruangkerja.php seperti yang ditunjukkan pada gambar 136 berikut ini.

```
<!-- Table -->
<div class="row">
  <div class="col">
    <div class="card shadow">
      <div class="card-header border-0">
        <h3 class="mb-0">Ruang Kerja</h3>
        <br>
        <form method="post" action=<?php echo base_url("index.php/admin/ruangkerja/formImport"); ?>" enctype="multipart/form-data">
          <p><label>Select Excel File</label>
          <input type="file" name="file" required /></p>
          <input type="submit" name="preview" value="Preview" class="btn btn-info" />
        </form>

      <?php
      if(isset($_POST['preview'])){ // Jika user menekan tombol Preview pada form
        if(isset($upload_error)){ // Jika proses upload gagal
          echo "<div style='color: red;'>".$upload_error."</div>"; // Muncul pesan error upload
          die; // stop skrip
        }
      }

      // Buat sebuah tag form untuk proses import data ke database
      echo "<form method='post' action='".$base_url("index.php/admin/ruangkerja/import")."'>";
```

Gambar 136 Views import_ruangkera.php