

TUGAS AKHIR

KONVERSI SHARIF JUDGE DARI CODEIGNITER 3 KE CODEIGNITER 4



Filipus

NPM: 6181901074

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023

FINAL PROJECT

**SHARIF JUDGE CONVERSION FROM CODEIGNITER 3 TO
CODEIGNITER 4**



Filipus

NPM: 6181901074

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2023**

ABSTRAK

SharIF Judge merupakan sebuah *online judge* yang dikustomisasi untuk keperluan tugas pada Universitas Katolik Parahyangan prodi Informatika. *SharIF Judge* pada awalnya bernama *Sharif Judge* yang merupakan sebuah perangkat lunak *open source*. Perangkat lunak *Sharif Judge* dibentuk oleh Mohammad Javad yang memiliki fungsi untuk menilai kode dengan beberapa bahasa seperti C, C++, Java, dan Python. *Sharif Judge* dibentuk menggunakan *framework CodeIgniter 3* yang merupakan *framework* berbasis PHP. *Sharif Judge* kemudian di *fork* dan dimodifikasi menjadi *SharIF Judge* dengan penambahan fungsi sesuai dengan kebutuhan Informatika UNPAR untuk mengumpulkan tugas dan ujian mahasiswa.

Perangkat lunak *SharIF Judge* pada saat ini menggunakan *CodeIgniter 3* yang merupakan sebuah *framework* ringan dan menyediakan banyak *library* untuk digunakan. Namun, *CodeIgniter 3* sudah memasuki fase *maintenance* sehingga tidak mendapat pembaharuan lebih lanjut dari pembentuknya. *CodeIgniter 3* juga pada akhirnya tidak dapat dipakai dan akan kehilangan dokumentasi dari situs *website* resminya. Oleh karena itu pada skripsi ini *SharIF Judge* akan dikonversi menuju *framework CodeIgniter 4*.

Konversi dilakukan dengan pemindahan *file* menuju direktori baru dan penulisan ulang fungsi-fungsi yang terdapat pada *SharIF Judge*. Selain itu, beberapa *library* dihapus dan digantikan karena sudah tidak terdapat pada *CodeIgniter 4*. *Library* yang sudah tidak ada diganti menggunakan fungsi baru yang terdapat pada *CodeIgniter 4* dan PHP. Konversi ini tidak mengubah fitur yang terdapat pada *SharIF Judge* sehingga dilakukan pengujian untuk setiap fitur agar berjalan sama dengan *SharIF Judge* versi *CodeIgniter 3*.

Kata-kata kunci: Konversi, Pembaharuan *CodeIgniter 3*, *CodeIgniter 3*, *CodeIgniter 4*, *SharIF Judge*, *Sharif Judge*

ABSTRACT

SharIF Judge is an online judge customized for tasks at the Parahyangan Catholic University, specifically for the Computer Science program. Initially named Sharif Judge, it was an open-source software developed by Mohammad Javad. The primary function of SharIF Judge is to evaluate code written in languages such as C, C++, Java, and Python. The software was built using CodeIgniter 3, a PHP-based framework. It was later forked and modified to become SharIF Judge, incorporating additional features to meet the specific needs of the Computer Science department at UNPAR (Parahyangan Catholic University) for collecting student assignments and exams.

The current version of SharIF Judge utilizes CodeIgniter 3, a lightweight framework that provides numerous libraries for development. However, as CodeIgniter 3 has entered the maintenance phase, it no longer receives updates from its creators. Furthermore, CodeIgniter 3 has become unusable and lost documentation from its official website. Therefore, in this thesis, SharIF Judge will be migrated to the CodeIgniter 4 framework.

The conversion involves transferring files to a new directory and rewriting functions found in SharIF Judge. Additionally, some libraries are removed and replaced due to their absence in CodeIgniter 4. The missing libraries are substituted with new functions available in CodeIgniter 4 and PHP. This conversion does not alter the features of SharIF Judge, and thorough testing is conducted for each feature to ensure it functions the same as SharIF Judge version CodeIgniter 3.

Keywords: Conversion, CodeIgniter 3 Upgrade, CodeIgniter 3, CodeIgniter 4, SharIF Judge, Sharif Judge

DAFTAR ISI

| | |
|---|-------------|
| DAFTAR ISI | ix |
| DAFTAR GAMBAR | xiii |
| 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Tujuan | 3 |
| 1.4 Batasan Masalah | 3 |
| 1.5 Metodologi | 3 |
| 1.6 Sistematika Pembahasan | 3 |
| 2 LANDASAN TEORI | 5 |
| 2.1 CodeIgniter 3 [1] | 5 |
| 2.1.1 <i>Model-View-Controller</i> | 6 |
| 2.1.2 <i>CodeIgniter URLs</i> | 8 |
| 2.1.3 <i>Helpers</i> | 9 |
| 2.1.4 <i>Libraries</i> | 9 |
| 2.1.5 <i>Database</i> | 14 |
| 2.1.6 <i>URI Routing</i> | 15 |
| 2.1.7 <i>Error Handling</i> | 16 |
| 2.1.8 <i>Auto-loading</i> | 16 |
| 2.2 SharIF Judge [2] | 17 |
| 2.2.1 Struktur Aplikasi | 17 |
| 2.2.2 Instalasi | 17 |
| 2.2.3 <i>Clean URLs</i> | 18 |
| 2.2.4 Users | 19 |
| 2.2.5 Menambah <i>Assignment</i> | 20 |
| 2.2.6 <i>Sample Assignment</i> | 23 |
| 2.2.7 <i>Test Structure</i> | 26 |
| 2.2.8 Deteksi Kecurangan | 28 |
| 2.2.9 Keamanan | 29 |
| 2.2.10 <i>Sandboxing</i> | 31 |
| 2.2.11 <i>Shield</i> | 31 |
| 2.3 CodeIgniter 4[3] | 33 |
| 2.3.1 <i>Models-Views-Controllers</i> | 34 |
| 2.3.2 <i>Autoloading Files</i> | 37 |
| 2.3.3 <i>Configuration</i> | 38 |
| 2.3.4 <i>CodeIgniter URLs</i> | 39 |
| 2.3.5 <i>Error Handling</i> | 40 |
| 2.3.6 <i>URI Routing</i> | 40 |
| 2.3.7 <i>Database</i> | 41 |
| 2.3.8 <i>Library</i> | 43 |

| | | |
|----------|--|------------|
| 2.3.9 | <i>Helpers</i> | 48 |
| 2.4 | Konversi CodeIgniter 3 ke CodeIgniter 4[3] | 48 |
| 2.4.1 | Struktur Aplikasi | 49 |
| 2.4.2 | <i>Routing</i> | 49 |
| 2.4.3 | <i>Model, View, dan Controller</i> | 49 |
| 2.4.4 | <i>Class Loading</i> | 50 |
| 2.4.5 | <i>Configuration</i> | 50 |
| 2.4.6 | <i>Database</i> | 50 |
| 2.4.7 | <i>Migrations</i> | 51 |
| 2.4.8 | <i>Routing</i> | 51 |
| 2.4.9 | <i>Libraries</i> | 51 |
| 2.4.10 | <i>Helpers</i> | 53 |
| 2.4.11 | <i>Framework</i> | 53 |
| 3 | ANALISIS | 55 |
| 3.1 | Analisis Sistem Kini | 55 |
| 3.1.1 | <i>Model</i> | 55 |
| 3.1.2 | <i>View</i> | 60 |
| 3.1.3 | <i>Controller</i> | 68 |
| 3.1.4 | <i>Assets</i> | 74 |
| 3.1.5 | <i>Config</i> | 75 |
| 3.1.6 | <i>Libraries</i> | 76 |
| 3.2 | Analisis Sistem Usulan | 80 |
| 3.2.1 | Persiapan <i>CodeIgniter 4</i> | 80 |
| 3.2.2 | Struktur Aplikasi | 80 |
| 3.2.3 | <i>Routing</i> | 82 |
| 3.2.4 | <i>Model, View, and Controller</i> | 82 |
| 3.2.5 | <i>Libraries</i> | 85 |
| 3.2.6 | <i>Configuration</i> | 89 |
| 3.2.7 | <i>Database</i> | 90 |
| 3.2.8 | <i>Error Handling</i> | 90 |
| 3.2.9 | <i>Helpers</i> | 90 |
| 4 | PERANCANGAN | 93 |
| 4.1 | Instalasi <i>CodeIgniter 4</i> | 93 |
| 4.2 | Perubahan Struktur Aplikasi | 93 |
| 4.2.1 | <i>app/Config</i> | 93 |
| 4.2.2 | <i>Controllers</i> | 100 |
| 4.2.3 | <i>Filters</i> | 102 |
| 4.2.4 | <i>Helpers</i> | 105 |
| 4.2.5 | <i>Libraries</i> | 105 |
| 4.2.6 | <i>Models</i> | 113 |
| 4.2.7 | <i>View</i> | 114 |
| 4.2.8 | <i>public</i> | 116 |
| 4.2.9 | <i>restricted</i> | 117 |
| 5 | IMPLEMENTASI DAN PENGUJIAN | 119 |
| 5.1 | Lingkungan Implementasi dan Pengujian | 119 |
| 5.2 | Implementasi | 120 |
| 5.2.1 | Kode Program | 120 |
| 5.2.2 | Basis Data | 120 |
| 5.3 | Pengujian Fungsional | 121 |

| | |
|--|------------|
| 5.4 Pengujian Experimental | 128 |
| 5.4.1 <i>Ubuntu 22.04</i> | 128 |
| 5.4.2 <i>Ubuntu 23.04</i> | 129 |
| 5.4.3 <i>Ubuntu 23.10</i> | 130 |
| 6 KESIMPULAN DAN SARAN | 133 |
| 6.1 Kesimpulan | 133 |
| 6.2 Saran | 133 |
| DAFTAR REFERENSI | 135 |
| A KODE PROGRAM <i>Controllers</i> | 137 |
| B KODE PROGRAM <i>Models</i> | 157 |
| C KODE PROGRAM <i>Views</i> | 171 |
| D KODE PROGRAM LAINNYA | 183 |

DAFTAR GAMBAR

| | |
|--|----|
| 1.1 Tampilan halaman <i>SharIF Judge</i> | 1 |
| 1.2 Pemindahan struktur aplikasi menuju <i>CodeIgniter 4</i> | 2 |
| 2.1 <i>Flow Chart</i> Aplikasi <i>CodeIgniter 3</i> | 5 |
| 2.2 Tampilan halaman <i>SharIF Judge</i> untuk menambahkan <i>assignment</i> | 20 |
| 3.1 Tampilan Halaman <i>Dashboard</i> | 60 |
| 3.2 Tampilan Halaman <i>Profile</i> | 61 |
| 3.3 Tampilan Halaman <i>Settings</i> | 61 |
| 3.4 Tampilan Halaman <i>Users</i> | 62 |
| 3.5 Tampilan Halaman <i>Notifications</i> | 62 |
| 3.6 Tampilan Halaman <i>Assignments</i> | 63 |
| 3.7 Tampilan Halaman <i>Problems</i> | 63 |
| 3.8 Tampilan Halaman <i>Submit</i> | 64 |
| 3.9 Tampilan Halaman <i>Final Submission</i> | 64 |
| 3.10 Tampilan Halaman <i>All Submission</i> | 65 |
| 3.11 Tampilan Halaman <i>Scoreboard</i> | 65 |
| 3.12 Tampilan Halaman <i>Hall of Fame</i> | 66 |
| 3.13 Tampilan Halaman <i>24-hour Log</i> | 66 |
| 3.14 Tampilan Halaman <i>ReJudge</i> | 67 |
| 3.15 Tampilan Halaman <i>Submission Queue</i> | 67 |
| 3.16 Tampilan Halaman <i>Cheat Detection</i> | 68 |
| 3.17 Pemindahan struktur aplikasi menuju <i>CodeIgniter 4</i> | 81 |

1

BAB 1

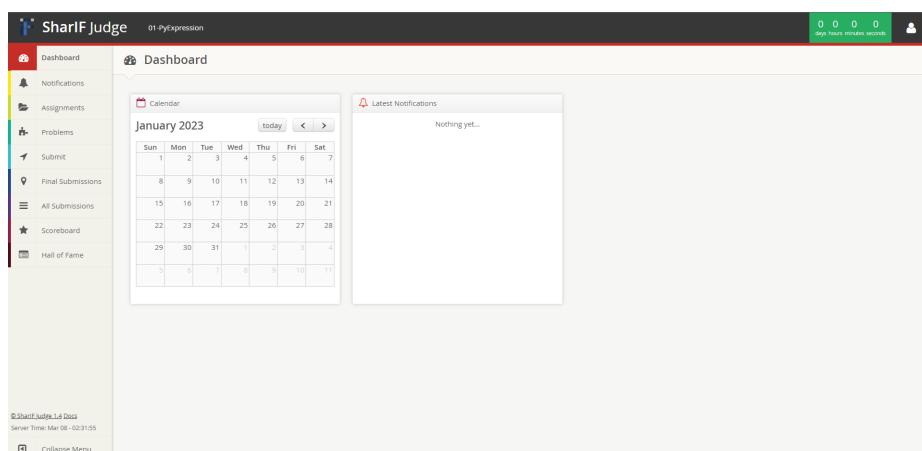
2

PENDAHULUAN

3 1.1 Latar Belakang

4 Tugas merupakan suatu bentuk pembelajaran dan penilaian yang diberikan oleh pengajar kepada
5 pelajar untuk membantu pelajar mendalami materi yang sudah diberikan[4]. Pembagian tugas
6 yang diberikan dapat dibagi menjadi 2 jenis yakni tugas individu dan tugas kelompok. Tugas
7 individu merupakan tugas yang hanya ditanggung oleh satu individu. Sedangkan tugas kelompok
8 merupakan tugas yang ditanggung oleh beberapa individu. Tugas yang telah diterima selanjutnya
9 akan dikumpulkan kepada pengajar dan diberikan penilaian berdasarkan tingkat ketepatan jawaban
10 dari tugas tersebut. Pengumpulan dan pengecekan tugas terutama *coding* secara manual memiliki
11 kekurangan dimana diperlukan banyak langkah dalam melakukan pengecekan dan pengiriman nilai.
12 Selain itu, pengecekan secara manual juga terdapat kesulitan dalam pengecekan yakni kekurangan
13 dalam pengecekan plagiat antara tugas pelajar. Oleh karena itu, dibutuhkan perangkat lunak untuk
14 melakukan pengecekan secara otomatis salah satunya adalah *Online Judge*.

15 *Online Judge* merupakan sebuah perangkat lunak yang dapat melakukan pengecekan program
16 sesuai dengan standar yang sudah diberikan. Perangkat lunak dapat menerima jawaban dari pelajar
17 dan melakukan pengecekan secara otomatis. Perangkat lunak kemudian memberikan keluaran
18 berupa nilai dari pelajar tersebut[5]. Salah satu perangkat lunak *Online Judge* terdapat pada
19 Universitas Katolik Parahyangan prodi Informatika bernama *SharIF Judge*. Gambar 1.1 menunjukkan
20 tampilan perangkat lunak *SharIF Judge* yang terletak pada Universitas Katolik Parahyangan.



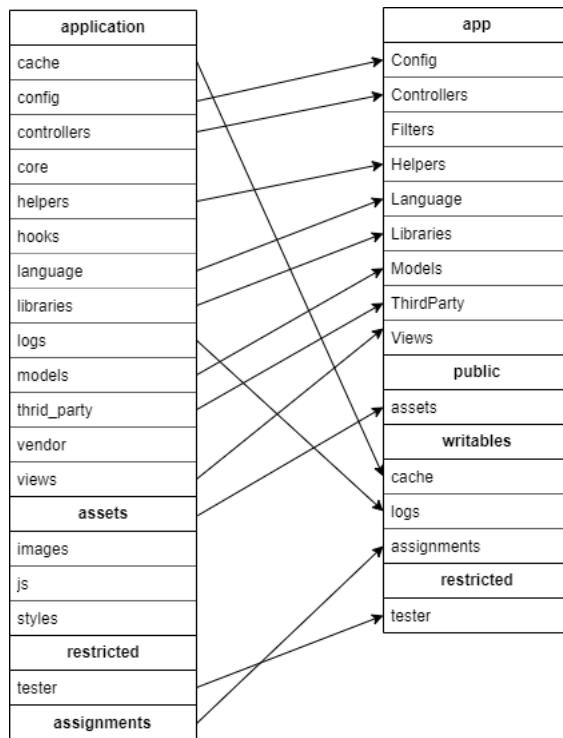
Gambar 1.1: Tampilan halaman *SharIF Judge*

21 *SharIF Judge* pada awalnya bernama *Sharif Judge* yang merupakan sebuah perangkat lunak

¹ *open source*. *Sharif Judge* berfungsi untuk menilai kode dengan beberapa bahasa seperti C, C++, Java, dan Python secara online. *Sharif Judge* pada awalnya dibentuk oleh Mohammad Javad menggunakan *framework CodeIgniter 3* yang merupakan *framework* berbasis PHP atau *Hypertext Preprocessor* [2]. *Sharif Judge* kemudian di *fork* dan dimodifikasi menjadi *SharIF Judge* dengan penambahan fungsi sesuai kebutuhan Informatika UNPAR untuk mengumpulkan tugas dan ujian mahasiswa[2].

⁷ *CodeIgniter 3* merupakan sebuah *framework opensource* yang bertujuan untuk mempermudah dalam pembangunan sebuah aplikasi *website* menggunakan PHP. *CodeIgniter 3* menggunakan struktur MVC yang membagi file menjadi 3 buah yaitu *Model*, *View*, dan *Controller*. Selain itu, *CodeIgniter 3* merupakan *framework* ringan dan menyediakan banyak *library* untuk digunakan oleh penggunanya[1]. Namun, *CodeIgniter 3* sudah memasuki fase *maintenance*¹ sehingga tidak mendapatkan pembaharuan lebih lanjut dari pembentuknya. *CodeIgniter 3* pada akhirnya tidak dapat dipakai dan akan kehilangan dokumentasi dari situs web resminya. Sehingga, perangkat lunak yang menggunakan *CodeIgniter 3* perlu dikonversi ke *framework* *CodeIgniter* dengan versi terbaru yakni *CodeIgniter 4*.

¹⁶ *CodeIgniter 4* merupakan versi terbaru dari *framework* *CodeIgniter* yang memiliki banyak perubahan fitur dari versi sebelumnya. *CodeIgniter 4* bisa dijalankan menggunakan versi PHP 7.4 atau lebih baru sedangkan *CodeIgniter 3* bisa dijalankan menggunakan versi PHP 5.6 atau lebih baru. *CodeIgniter 4* membagi file menggunakan struktur MVC, namun memiliki struktur direktori berbeda dengan versi sebelumnya[3]. Perubahan direktori ditunjukkan oleh Gambar 1.2.



Gambar 1.2: *Pemindahan struktur aplikasi menuju CodeIgniter 4*

²¹ Gambar 1.2 menunjukkan perubahan struktur yang terdapat pada *CodeIgniter 4*. Rincian perubahan dapat dilihat pada bab 3. Pada skripsi ini dilakukan konversi *SharIF Judge* dari

¹Pemberitahuan fase *maintenance* *CodeIgniter 3* <https://codeigniter.com/download>(19 Maret 2023)

1 *CodeIgniter 3* menjadi *CodeIgniter 4*.

2 **1.2 Rumusan Masalah**

- 3 • Bagaimana cara melakukan konversi SharIF Judge pada *CodeIgniter 3* menjadi *CodeIgniter*
4 4?

5 **1.3 Tujuan**

6 Tujuan dari skripsi ini adalah sebagai berikut:

- 7 • Melakukan konversi dengan megubah kode sesuai dengan standar *CodeIgniter 4*.

8 **1.4 Batasan Masalah**

9 Batasan masalah pada pembentukan skripsi ini adalah sebagai berikut:

- 10 • Autentifikasi RADIUS dan LDAP tidak akan diuji karena tidak memiliki *server* yang terintegrasi.
11 grasi.

12 **1.5 Metodologi**

13 Metodologi yang dilakukan dalam melakukan penelitian ini adalah sebagai berikut:

- 14 1. Melakukan analisis dan eksplorasi fungsi-fungsi perangkat lunak SharIF Judge.
15 2. Melakukan studi literatur kebutuhan konversi dari *CodeIgniter 3* menjadi *CodeIgniter 4*.
16 3. Melakukan konversi perangkat lunak dari *CodeIgniter 3* menjadi *CodeIgniter 4*.
17 4. Melakukan pengujian dan eksperimen terhadap perangkat lunak yang sudah di konversi.
18 5. Menyelesaikan pembentukan dokumen

19 **1.6 Sistematika Pembahasan**

20 Penelitian ini akan dibahas dalam enam bab yang masing-masing berisi:

21 1. **Bab 1:** Pendahuluan

22 Bab ini berisi latar belakang, rumusan masalah,tujuan, batasan masalah, metodologi, dan
23 sistematika pembahasan.

24 2. **Bab 2:** Landasan Teori

25 Bab ini berisi pembahasan dasar-dasar teori yang akan digunakan dalam melakukan konversi
26 SharIF Judge dari *CodeIgniter 3* ke *CodeIgniter 4*. Landasan Teori yang digunakan diantaranya
27 adalah SharIF Judge, *CodeIgniter 3*, *CodeIgniter 4*, dan Konversi *CodeIgniter 3* ke *CodeIgniter*
28 4.

29 3. **Bab 3:** Analisis

30 Bab ini berisi analisis *SharIF Judge* dan analisis kebutuhan konversi menuju *CodeIgniter 3*.

31 4. **Bab 4:** Perancangan

32 Bab ini berisi mengenai rancangan perangkat lunak yang akan dikonversi.

¹ 5. **Bab 5:** Implementasi dan Pengujian

² Bab ini berisi hasil implementasi dan pengujian yang telah dilakukan untuk melakukan
³ konversi *SharIF Judge* dari *CodeIgniter 3* ke *CodeIgniter 4*.

⁴ 6. **Bab 6:** Kesimpulan dan Saran

⁵ Bab ini berisi kesimpulan dari hasil konversi yang telah dilakukan dan saran-saran terhadap
⁶ perangkat lunak.

1

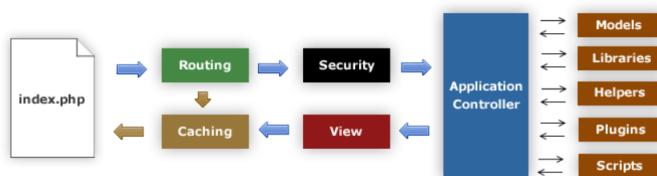
BAB 2

2

LANDASAN TEORI

3 2.1 CodeIgniter 3 [1]

4 CodeIgniter 3 merupakan sebuah *framework opensource* yang berfungsi untuk mempermudah
5 pengguna dalam membangun aplikasi *website* menggunakan bahasa PHP. *CodeIgniter 3* juga
6 merupakan *framework* ringan yang menggunakan struktur *Model-View-Controller*, dan menghasilkan
7 URL yang bersih. CodeIgniter 3 memiliki tujuan untuk membantu pengguna dalam membangun
8 aplikasi web lebih cepat dengan menyediakan beragam *library* dan tampilan dan *logic* yang simpel.
9 *Code Igniter 3* memiliki *flow chart* aplikasi yang ditunjukan pada Gambar 2.1.



Gambar 2.1: *Flow Chart* Aplikasi *CodeIgniter 3*

10 Berikut merupakan pembagian *flow chart* aplikasi *CodeIgniter 3* seperti yang ditunjukan pada
11 Gambar 2.1:

- 12 1. *index.php* berfungsi sebagai *front controller* yang berguna untuk melakukan inisiasi
- 13 2. *Router* berfungsi untuk melakukan pemeriksaan dan menentukan penggunaan *HTTP Request*.
- 14 3. *Cache* berfungsi untuk mengirimkan *file cache* kepada *browser* secara langsung hanya jika
15 terdapat *cache*.
- 16 4. *Security* berfungsi untuk alat penyaringan setiap data dan *HTTP Request* yang masuk.
17 Penyaringan data tersebut dilakukan sebelum *controller* aplikasi dimuat agar aplikasi menjadi
18 lebih aman.
- 19 5. *Controller* befungsi sebagai alat untuk memuat *model*, *libraries*, dan sumber daya yang
20 dibutuhkan untuk menjalankan permintaan spesifik.
- 21 6. *View* dikirimkan menuju *browser* untuk dilihat oleh pengguna. Apabila *caching* dinyalakan,
22 maka *view* akan dilakukan *cached* terlebih dahulu sehingga permintaan selanjutnya dapat
23 diberikan.

2.1.1 Model-View-Controller

CodeIgniter 3 merupakan *framework* berbasis arsitektur *Model-View-Controller* atau yang selanjutnya akan disebut sebagai MVC. MVC merupakan sebuah pendekatan perangkat lunak yang memisahkan antara logika dengan presentasi atau tampilannya. Penggunaan struktur ini mengurangi penggunaan skrip pada halaman web karena tampilan terpisah dengan skrip PHP. Berikut merupakan penjelasan mengenai struktur MVC:

Model

Model berfungsi untuk mewakili struktur data perangkat lunak dalam mengambil, memasukan, dan memperbarui data pada *database*. Model biasanya digunakan pada *file controller* dan dapat dipanggil menggunakan kode sebagai berikut:

```
11     $this->load->model('model_name');
```

Sintaks diatas akan memuat *model* dengan nama *model_name*. Selanjutnya, pengguna dapat membentuk fungsi pada *file model* sesuai dengan kebutuhan. Kode 2.1 menunjukkan contoh *file Model CodeIgniter 3* pada direktori *application/models/*:

Kode 2.1: Contoh *model* pada *CodeIgniter 3*

```
15 class Blog_model extends CI_Model {
16
17     public $title;
18     public $content;
19     public $date;
20
21
22     public function get_last_ten_entries()
23     {
24         $query = $this->db->get('entries', 10);
25         return $query->result();
26     }
27
28     public function insert_entry()
29     {
30         $this->title    = $_POST['title']; // please read the below note
31         $this->content  = $_POST['content'];
32         $this->date     = time();
33
34         $this->db->insert('entries', $this);
35     }
36
37     public function update_entry()
38     {
39         $this->title    = $_POST['title'];
40         $this->content  = $_POST['content'];
41         $this->date     = time();
42
43         $this->db->update('entries', $this, array('id' => $_POST['id']));
44     }
45
46 }
```

Berikut merupakan penjelasan beberapa fungsi pada Kode 2.1:

- *get_last_ten_entries()* yang berfungsi untuk mengambil 10 data terakhir dari tabel *entries* menggunakan *query builder*.
- *insert_entry()* yang berfungsi untuk memasukan data *title*, *content*, dan *date* menuju tabel *entries*.
- *update_entry()* yang befungsi untuk memperbarui data *title*, *content*, dan *date* pada tabel *entries*.

1 *Views*

2 *View* berfungsi dalam menyajikan informasi kepada pengguna. *View* biasanya merupakan halaman
 3 web, namun pada *CodeIgniter 3* *View* dapat berupa pecahan halaman seperti *header*, *footer*, *sidebar*,
 4 dan lainnya. Pecahan halaman dapat dimasukan secara fleksibel apabila dibutuhkan.

Kode 2.2: Contoh *view* pada *CodeIgniter 3*

```

5 1 <?php
6 2 <html>
7 3 <head>
8 4     <title>My Blog</title>
9 5 </head>
10 6 <body>
11 7     <h1>Welcome to my Blog!</h1>
12 8 </body>
13 9 </html>
```

16 Kode 2.2 merupakan contoh *file view* *CodeIgniter 3* pada direktori **application/views/** yang
 17 berisi judul My Blog dan heading Welcome to my Blog!. *View* tidak dapat dipanggil secara
 18 langsung, namun pengguna dapat memanggil halaman yang sudah dibentuk pada *file controller*
 19 dengan cara sebagai berikut:

```
20         $this->load->view('name');
```

21 Sintaks diatas akan mengembalikan halaman *view* dengan nama **name** dan menampilkannya
 22 kepada pengguna.

23 *Controller*

24 *Controller* berfungsi sebagai perantara antara *Model*, *View*, dan sumber daya yang dibutuhkan
 25 untuk melakukan proses *HTTP Request* dan menjalankan halaman web. Penamaan *controller*
 26 biasanya digunakan sebagai *url* pada perangkat lunak pengguna. Contoh *controller* *CodeIgniter 3*
 27 pada direktori **application/controllers/** ditunjukan pada Kode 2.3:

Kode 2.3: Contoh *controller* pada *CodeIgniter 3*

```

28 1 <?php
29 2 class Blog extends CI_Controller {
30 3
31 4     public function index()
32 5     {
33 6         echo 'Hello_World!';
34 7     }
35 8
36 9     public function comments()
37 0     {
38 1         echo 'Look_at_this!';
39 2     }
40 3 }
```

43 Kode 2.3 berfungsi dalam mengembalikan *string* sesuai dengan fungsi *controller* yang dipanggil.
 44 Nama *controller* dan metode diatas akan dijadikan segmen pada *URL* seperti berikut:

```
45             example.com/index.php/blog/index/
```

46 Metode *index* akan secara otomatis dipanggil menjadi *URL* dan pengguna juga dapat memberi
 47 parameter untuk metode *controller* yang selanjutnya dapat diambil dari *URL*.

2.1.2 CodeIgniter URLs

CodeIgniter 3 menggunakan pendekatan *segment-based* dibandingkan menggunakan *query string* untuk membangun *URL* yang mempermudah mesin pencari dan pengguna. Berikut merupakan contoh *URL* pada CodeIgniter 3:

```
example.com/news/article/my_article
```

URL diatas dibentuk berdasarkan segmen *Controller* sebagai berikut :

```
example.com/class/function/ID
```

Segmen tersebut dibagi menjadi tiga buah sebagai berikut:

1. Segmen pertama merepresentasikan kelas *controller* yang dipanggil.
2. Segmen kedua merepresentasikan kelas fungsi atau metode yang digunakan.
3. Segmen ketiga dan segmen lainnya merepresentasikan *ID* dari variabel yang akan dipindahkan menuju *controller*.

Secara *default*, *URL* yang dihasilkan CodeIgniter 3 terdapat nama *file index.php* seperti contoh dibawah ini:

```
example.com/index.php/news/article/my_article
```

Pengguna dapat menghapus *file index.php* pada *url* menggunakan *file .htaccess* apabila *server Apache* pengguna menghidupkan *mod_rewrite*. Contoh *file .htaccess* menggunakan metode *negative* ditunjukan pada Kode 2.4:

Kode 2.4: Contoh *file .htaccess* pada halaman index.php

```
19 RewriteEngine On
20 RewriteCond %{REQUEST_FILENAME} !-f
21 RewriteCond %{REQUEST_FILENAME} !-d
22 RewriteRule ^(.*)$ index.php/$1 [L]
```

Aturan pada Kode 2.4 menyebabkan *HTTP Request* selain yang berasal dari direktori atau *file* diperlakukan sebagai sebuah permintaan pada *file index.php*. Selain itu, pengguna juga dapat menambahkan akhiran pada *URL* agar halaman pengguna dapat menampilkan halaman sesuai dengan tipe yang diinginkan. Kode 2.5 menunjukan contoh *URL* sebelum dan sesudah ditambahkan akhiran berupa *.html*:

Kode 2.5: Contoh URL sebelum dan sesudah ditambahkan akhiran

```
30 example.com/index.php/products/view/shoes
31 example.com/index.php/products/view/shoes.html
```

Kode 2.5 merupakan contoh URL sebelum dan sesudah ditambahkan akhiran. Pengguna juga dapat menyalakan fitur *query strings* dengan cara sebagai mengubah *file application/config.php* seperti berikut:

Kode 2.6: *File application/config.php*

```
38 $config['enable_query_strings'] = FALSE;
39 $config['controller_trigger'] = 'c';
40 $config['function_trigger'] = 'm';
```

1 Pengguna dapat mengubah `enable_query_strings` menjadi `TRUE`. Pengubahan fitur tersebut
 2 dapat memperbolehkan pengguna untuk menambahkan *query* pada *URL* yang dibentuk. Berikut
 3 merupakan contoh pengaksesan *URL* melalui *query strings*:

```
4 index.php?c=controller&m=method
```

5 Sintaks diatas membentuk URL agar dapat mengakses fungsi ataupun metode dari *controller*
 6 menggunakan *query* tanpa harus menggunakan *URL* biasa.

7 2.1.3 *Helpers*

8 *Helpers* merupakan fungsi pada *CodeIgniter 3* yang mempermudah pengguna dalam membangun
 9 aplikasi web. Setiap file *helpers* terdiri dari banyak fungsi yang membantu sesuai kategori dan tidak
 10 ditulis dalam format *Object Oriented*. File *helpers* terdapat pada direktori `system/helpers` atau
 11 `application/helpers`. Pengguna dapat memakai fitur *helpers* dengan cara memuatnya seperti
 12 berikut:

```
13 $this->load->helper('name');
```

14 Pemanggilan *helper* tidak menggunakan ekteksi `.php` melainkan hanya menggunakan nama dari
 15 *helper* yang ingin digunakan. Pengguna dapat memanggil satu atau banyak *helper* pada metode
 16 *controller* ataupun *view* sesudah dimuat.

17 2.1.4 *Libraries*

18 *CodeIgniter 3* menyediakan *library* yang dapat dipakai pengguna untuk mempermudah pembentukan
 19 aplikasi web. *Library* merupakan kelas yang tersedia pada direktori `application/libraries` dan
 20 dapat ditambahkan, diperluas, dan digantikan.

Kode 2.7: Contoh kelas *library* pada *CodeIgniter 3*

```
21 <?php
22 defined('BASEPATH') OR exit('No direct script access allowed');
23
24 class Someclass {
25
26     public function some_method()
27     {
28     }
29 }
30 }
```

32 Kode 2.7 menunjukkan contoh file *library* pada *CodeIgniter 3*. Setiap pembentukan file *library*
 33 diperlukan huruf kapital dan harus sama dengan nama kelasnya. Kode 2.8 menunjukkan contoh
 34 pemanggilan file *library* pada file *controller*:

Kode 2.8: Contoh pemanggilan *library* pada file *controller*

```
35 $params = array('type' => 'large', 'color' => 'red');
36
37 $this->load->library('someclass', $params);
```

40 Kode 2.8 menunjukkan pemanggilan *library* `someclass` yang dilakukan melalui *controller* manapun.
 41 Kelas *library* juga dapat diberikan parameter sesuai dengan metode yang dibentuk. Selain
 42 membentuk *library*, *CodeIgniter 3* menyediakan berbagai *library* yang dapat digunakan oleh pengguna.
 43 Berikut merupakan *library* yang disediakan oleh *CodeIgniter 3*:

1 Kelas *Input*

2 Kelas *input* memiliki dua buah fungsi yakni melakukan praproses data masukan dan memberikan
 3 metode kepada beberapa *helper*. Penggunaan kelas *input* dapat dipanggil menggunakan sintaks
 4 berikut:

```
5           $something = $this->input->post('something');
```

6 Sintaks diatas akan mengambil data yang dikirim menggunakan metode *post* pada data bernama
 7 *something*.

8 JavaScript

9 Penggunaan kelas *javascript* dapat dipanggil pada konstruktor *controller* dengan melakukan inisiasi
 10 *library* seperti pada sintaks berikut:

```
11          $this->load->library('javascript');
```

12 Pengguna selanjutnya harus melakukan inisiasi *library* pada halaman *view tag <head>*. Kode
 13 [2.9](#) menunjukan penambahan *tag* pada halaman *view*.

Kode 2.9: Contoh penambahan *tag* pada *file view*

```
14 15 <?php echo $library_src;?>
16 2
17 3 <?php echo $script_head;?>
```

19 Sintaks *\$library_src* merupakan lokasi *library* yang akan dimuat sedangkan *\$script_head*
 20 merupakan lokasi untuk fungsi yang akan dijalankan. Selanjutnya *javascript* dapat diinisiasi
 21 pada *controller* menggunakan sintaks berikut:

```
22          $this->javascript
```

23 Selain menggunakan *javascript*, pengguna dapat memakai *jQuery* dengan menambahkan *jQuery*
 24 pada akhir inisiasi kelas *javascript* seperti berikut:

```
25          $this->load->library('javascript/jquery');
```

26 Pengguna dapat memakai berbagai fungsi *library jquery* menggunakan sintaks berikut:

```
27          $this->jquery
```

28 Kelas *Email*

29 *CodeIgniter 3* menyediakan kelas *email* dengan fitur sebagai berikut:

- 30 • Beberapa Protokol: *Mail*, *Sendmail*, dan *SMTP*
- 31 • Enkripsi TLS dan SSL untuk *SMTP*
- 32 • Beberapa Penerima
- 33 • CC dan BCCs
- 34 • HTML atau *email* teks biasa
- 35 • Lampiran

- 1 • Pembungkus kata
- 2 • Prioritas
- 3 • Mode BCC *Batch*, memisahkan daftar *email* besar menjadi beberapa BCC kecil.
- 4 • Alat *Debugging email*

5 Penggunaan *library email* dapat dikonfigurasikan pada *file config*. Pengguna dapat mengirim
6 *email* menggunakan fungsi-fungsi yang telah disediakan *library email*. Kode 2.10 menunjukkan contoh
7 pengiriman *email* dari *your@example.com* menuju *someone@example.com* melalui *controller*.

Kode 2.10: Contoh pengiriman *email* melalui *controller*

```

8 $this->load->library('email');
10 2
11 3 $this->email->from('your@example.com', 'Your_Name');
12 4 $this->email->to('someone@example.com');
13 5 $this->email->cc('another@another-example.com');
14 6 $this->email->bcc('them@their-example.com');
15 7
16 8 $this->email->subject('Email_Test');
17 9 $this->email->message('Testing_the_email_class.');
18 0
19 1 $this->email->send();

```

21 Kode 2.10 juga mengirim dua buah salinan yang akan di CC menuju *another@another-example.com*
22 dan BCC menuju *them@their-example.com* dengan subjek berupa *Email Test* dengan pesan
23 *Testing the email class*. Selain itu, pengguna juga dapat melakukan konfigurasi preferensi
24 *email* melalui dua puluh satu preferensi. Pengguna dapat melakukan konfigurasi secara otomatis
25 melalui *file config* atau melakukan konfigurasi secara manual. Kode 2.11 menunjukkan contoh
26 konfigurasi preferensi secara manual untuk mengirimkan *email*.

Kode 2.11: Contoh konfigurasi preferensi *library email* secara manual

```

27 1 $config['protocol'] = 'sendmail';
28 2 $config['mailpath'] = '/usr/sbin/sendmail';
29 3 $config['charset'] = 'iso-8859-1';
30 4 $config['wordwrap'] = TRUE;
31 5
32 6 $this->email->initialize($config);

```

35 Kode 2.11 menunjukkan contoh konfigurasi pengiriman *email* dengan protokol *sendmail*, tujuan
36 *usr/sbin/sendmail*, *character set* berjenis *iso-8859-1*, dan menyalakan fitur *wordwrap*. Selan-
37 jutnya konfigurasi dapat diinisialisasikan menggunakan sintaks *initialize*.

38 Kelas *File Uploading*

39 Pengunggahan *file* terdapat empat urutan proses sebagai berikut:

- 40 1. Dibentuk sebuah form untuk pengguna memilih dan mengunggah *file*.
- 41 2. Setelah *file* diunggah, *file* akan dipindahkan menuju direktori yang dipilih.
- 42 3. Pada pengiriman dan pemindahan *file* dilakukan validasi sesuai dengan ketentuan yang ada.
- 43 4. Setelah *file* diterima akan dikeluarkan pesan berhasil.

44 Perangkat lunak akan memindahkan *file* yang sudah diunggah pada *form* menuju *controller*
45 untuk dilakukan validasi dan penyimpanan. *File* yang tidak berhasil dilakukan validasi akan
46 mengembalikan pesan dan tampilan *error*.

Kode 2.12: Contoh *controller* untuk melakukan validasi dan penyimpanan

```

47 1 <?php
48 2
49 2

```

```

13| class Upload extends CI_Controller {
14|
15|     public function __construct()
16|     {
17|         parent::__construct();
18|         $this->load->helper(array('form', 'url'));
19|
20|
21|     }
22|
23|     public function index()
24|     {
25|         $this->load->view('upload_form', array('error' => '' ));
26|
27|
28|     }
29|
30|     public function do_upload()
31|     {
32|         $config['upload_path']          = './uploads/';
33|         $config['allowed_types']        = 'gif|jpg|png';
34|         $config['max_size']             = 100;
35|         $config['max_width']            = 1024;
36|         $config['max_height']           = 768;
37|
38|         $this->load->library('upload', $config);
39|
40|         if ( ! $this->upload->do_upload('userfile'))
41|         {
42|             $error = array('error' => $this->upload->display_errors());
43|
44|             $this->load->view('upload_form', $error);
45|         }
46|         else
47|         {
48|             $data = array('upload_data' => $this->upload->data());
49|
50|             $this->load->view('upload_success', $data);
51|         }
52|
53|     }
54|
55| ?>
```

Kode 2.12 menunjukkan contoh kode dengan dua buah metode sebagai berikut:

1. `index()` yang digunakan untuk mengembalikan *view* bernama `upload_form`
2. `do_upload` yang digunakan untuk melakukan validasi berupa tipe, ukuran, lebar, dan panjang maksimal sebuah *file*. Metode ini juga mengembalikan *error* apabila pengunggahan *file* tidak berhasil dan menyimpan *file* pada direktori `./uploads/`.

Direktori penyimpanan dapat diubah sesuai dengan kebutuhan namun perlu dilakukan penrubahan izin direktori menjadi 777 agar dapat dibaca, ditulis, dan dieksekusi oleh seluruh pengguna.

47 Session

48 *Library session* memperbolehkan aplikasi untuk melihat jejak dari aktivitas pengguna. *CodeIgniter* menyediakan beberapa cara penyimpanan *session* yakni menggunakan *file*, *database*, *redis*, dan *memcached*. *Library* ini dapat diinisiasi menggunakan sintaks sebagai berikut:

```
51         $this->load->library('session');
```

52 Sintaks diatas akan melakukan inisiasi terhadap *library session* dan dapat digunakan menggunakan sintaks sebagai berikut:

```
54         $this->session
```

55 *Session* bekerja ketika halaman termuat dan akan melakukan pengecekan terhadap *session cookie* pengguna. Data ini akan disimpan dan diperbarui sesuai dengan data yang diterima. Pengguna dapat mengambil data dari *session* ini menggunakan sintaks sebagai berikut:

```
58         $this->session->userdata('item');
```

- 1 Sintaks diatas akan mengambil seluruh data *session* dengan nama *item*. Secara *default*, aplikasi
 2 akan menyimpan data *session* pada file namun pengguna dapat mengubah konfigurasi sesuai dengan
 3 kebutuhan. Kode 2.13 menunjukan konfigurasi untuk menyimpan data pada *database* dengan nama
 4 tabel *ci_sessions*.

Kode 2.13: Contoh konfigurasi untuk menyimpan data pada *database*

```
5 $config['sess_driver'] = 'database';
6 $config['sess_save_path'] = 'ci_sessions';
```

9 Kelas *URI*

- 10 Kelas *URI* menyediakan metode untuk mengambil informasi yang terdapat pada *URI strings*.
 11 Kelas ini diinisiasi secara otomatis oleh *CodeIgniter* sehingga tidak perlu dilakukan inisiasi lagi.
 12 Informasi dapat diambil menggunakan sintaks berikut:

```
13 $array = $this->uri->uri_to_assoc(3, $default);
```

- 14 Sintaks diatas akan mengambil informasi yang tersedia pada *URI segment* menjadi sebuah *associative array*. *Array* yang dihasilkan akan berisikan *key* dan *value* sesuai dengan penempatan *segment*.

16 Kelas *Zip Encoding*

- 17 Zip merupakan format sebuah *file* yang berguna untuk melakukan kompress terhadap *file* untuk
 18 mengurangi ukuran dan menjadikannya sebuah *file*. *CodeIgniter 3* menyediakan *library Zip Encoding*
 19 yang dapat digunakan untuk membangun arsip Zip yang dapat diunduh menuju *desktop* atau
 20 disimpan pada direktori. *Library* ini dapat diinsiasi dengan kode sebagai berikut:

```
21 $this->load->library('zip');
```

- 22 Setelah diinisiasi, pengguna dapat memanggil *library* tersebut menggunakan kode sebagai
 23 berikut:

```
24 $this->zip
```

- 25 Sintaks diatas akan memanggil *library Zip* pada sebuah *file*. Kode 2.14 menunjukan contoh
 26 penggunaan *library Zip Encoding* untuk melakukan penyimpanan *Zip file* pada direktori dan dapat
 27 mengunduh *file* menuju *desktop* pengguna.

Kode 2.14: Contoh penggunaan *library Zip Encoding*

```
28
29 1 $name = 'mydata1.txt';
30 2 $data = 'A_Data_String!';
31 3
32 4 $this->zip->add_data($name, $data);
33 5
34 6 // Write the zip file to a folder on your server. Name it "my_backup.zip"
35 7 $this->zip->archive('/path/to/directory/my_backup.zip');
36 8
37 9 // Download the file to your desktop. Name it "my_backup.zip"
38 0 $this->zip->download('my_backup.zip');
```

- 40 Selain menggunakan *library* yang sudah disediakan, pengguna dapat membangun dan memperluas
 41 *libraries* sendiri sesuai dengan kebutuhan. Kode 2.15 merupakan contoh *library* yang dibentuk oleh
 42 pengguna yang berfungsi untuk memanggil *helper* bernama URL.

Kode 2.15: Contoh *library* yang dibentuk

```

1 class Example_library {
2
3     protected $CI;
4
5     // We'll use a constructor, as you can't directly call a function
6     // from a property definition.
7     public function __construct()
8     {
9         // Assign the CodeIgniter super-object
10        $this->CI =& get_instance();
11    }
12
13    public function foo()
14    {
15        $this->CI->load->helper('url');
16        redirect();
17    }
18}
19

```

Pengguna dapat memanggil *library* tersebut seperti memanggil kelas *library* lainnya. Selain itu, pengguna juga dapat mengganti *library* yang sudah ada dengan *library* yang dibentuk pengguna. *Library* yang dibentuk harus memiliki nama kelas sama persis dengan nama *library* yang ingin digantikan.

2.1.5 Database

CodeIgniter 3 memiliki konfigurasi *database* yang menyimpan data-data terkait aturan *database*. Kode 2.16 menunjukkan contoh konfigurasi pada file *database* bernama **database_name** dengan **username root** tanpa sebuah **password**.

Kode 2.16: Contoh konfigurasi *database*

```

29 $db['default'] = array(
30     'dsn'      => '',
31     'hostname' => 'localhost',
32     'username' => 'root',
33     'password' => '',
34     'database' => 'database_name',
35     'dbdriver' => 'mysqli',
36     'dbprefix' => '',
37     'pconnect' => TRUE,
38     'db_debug' => TRUE,
39     'cache_on' => FALSE,
40     'cachedir' => '',
41     'char_set' => 'utf8',
42     'dbcollat' => 'utf8_general_ci',
43     'swap_pre' => '',
44     'encrypt' => FALSE,
45     'compress' => FALSE,
46     'stricton' => FALSE,
47     'failover' => array()
48 );
49

```

CodeIgniter 3 juga menyediakan fitur *query* untuk menyimpan, memasukan, memperbarui, dan menghapus data pada *database* sesuai dengan konfigurasi *database* yang sudah diatur. Kode 2.17 menunjukkan contoh kode untuk melakukan *query* pada tabel **blogs** yang melakukan *join* dengan tabel **comments**. Pengguna dapat mengambil hasil dari *query* menjadi *object* atau *array*.

Kode 2.17: Contoh penggunaan *query*

```

55 $this->db->select('*');
56 $this->db->from('blogs');
57 $this->db->join('comments', 'comments.id=_blogs.id');
58 $query = $this->db->get();
59

```

1 Selain untuk melakukan *query*, *database* pada *CodeIgniter 3* juga dapat digunakan untuk mem-
 2 bangun, menghapus, dan mengubah *database* ataupun menambahkan kolom pada *table*. Penggunaan
 3 *database* untuk membangun, menghapus, atau mengubah *database* harus dilakukan inisiasi sebagai
 4 berikut:

```
5 $this->load->dbforge()
```

6 Setelah dilakukan inisiasi pengguna dapat membangun *database* menggunakan kelas **Forge**. Kode
 7 [2.18](#) menunjukan contoh untuk membangun *database* dengan nama **db_name**.

Kode 2.18: Contoh membangun *database* menggunakan *CodeIgniter 3*

```
8 1 $this->dbforge->create_database('db_name')
```

11 Selain itu, pengguna juga dapat menambahkan kolom dengan konfigurasinya. Kode [2.19](#)
 12 menunjukan contoh penambahan kolom sesuai dengan kebutuhan pengguna.

Kode 2.19: Contoh menambahkan kolom dengan konfigurasinya menggunakan *CodeIgniter 3*

```
13 1 $fields = array(
14 2     'blog_id' => array(
15 3         'type' => 'INT',
16 4         'constraint' => 5,
17 5         'unsigned' => TRUE,
18 6         'auto_increment' => TRUE
19 7     ),
20 8     'blog_title' => array(
21 9         'type' => 'VARCHAR',
22 0         'constraint' => '100',
23 1         'unique' => TRUE,
24 2     ),
25 3     'blog_author' => array(
26 4         'type' => 'VARCHAR',
27 5         'constraint' => '100',
28 6         'default' => 'King_of_Town',
29 7     ),
30 8     'blog_description' => array(
31 9         'type' => 'TEXT',
32 0         'null' => TRUE,
33 1     ),
34 2 );
35 3 $this->dbforge->add_field($fields)
36 4 $this->dbforge->create_table('table_name');
```

39 Kode [2.19](#) menunjukan contoh penggunaan *database* untuk menambahkan kolom sesuai dengan
 40 tipe, batas dari data yang disimpan, penambahan otomatis, dan isi *default* dari kolom pada tabel
 41 **table_name**.

42 2.1.6 *URI Routing*

43 *URL string* yang dihasilkan pada *CodeIgniter 3* biasanya menggunakan nama atau metode *controller*
 44 seperti pada berikut:

```
45 example.com/class/function/id/
```

46 Namun, pengguna dapat melakukan pemetaan ulang terhadap url yang dibentuk agar dapat
 47 memanggil metode dengan penambahanan *segment* yang diinginkan.

Kode 2.20: Contoh url yang sudah dimetakan

```
48 1 example.com/product/1/
49 2 example.com/product/2/
50 3 example.com/product/3/
51 4 example.com/product/4/
```

1 Kode 2.20 merupakan contoh url yang sudah dimetakan ulang. Pengguna dapat menambahkan
 2 kode pemetaan pada *file application/config/routes.php* yang terdapat *array* bernama **\$route**. Berikut
 3 merupakan beberapa cara melakukan pemetaan terhadap *url*:

4 **WildCards**

5 *Route wildcard* biasanya berisikan kode seperti berikut:

```
6 $route['product/:num'] = 'catalog/product_lookup';
```

7 *Route* diatas dibagi menjadi dua buah yakni:

8 1. Bagian segmen *URL*

9 Bagian pertama merupakan segmen pertama *url* yang akan tampil pada *url*. Bagian kedua
 10 merupakan segmen kedua dapat berisikan angka atau karakter.

11 2. Bagian kelas dan metode

12 Bagian kedua berisikan kelas dan metode dari *controller* yang akan digunakan pada *url*.

13 **Ekspresi Reguler**

14 Pengguna dapat memakai ekspresi reguler untuk melakukan pemetaan ulang *route*. Berikut
 15 merupakan contoh ekspresi reguler yang biasa digunakan:

```
16 $route['products/([a-z]+)/(\d+)'] = '$1/id_$2';
```

17 Ekspresi diatas menghasilkan URI *products/shirts/123* yang memanggil kelas *controller* dan
 18 metode *id_123*. Pengguna juga dapat mengambil segmen banyak seperti berikut:

```
19 $route['login/(.+)'] = 'auth/login/$1';
```

20 **2.1.7 Error Handling**

21 *CodeIgniter 3* menyediakan sebuah fungsi untuk melaporkan *error* pada aplikasi. Fungsi ini juga
 22 memungkinkan *error* dan *debugging message* disimpan menuju file *text*. Berikut merupakan contoh
 23 penggunaan fungsi ini:

```
24 $show_error($message, $status_code, $heading = 'An Error Was Encountered')
```

25 Sintaks diatas menerima parameter berupa pesan, kode *HTTP*, dan *heading* yang ditampilkan pa-
 26 da halaman *error*. Sintaks ini akan mengembalikan halaman *error* dengan nama *error_general.php*.

27 **2.1.8 Auto-loading**

28 *CodeIgniter 3* menyediakan sebuah fungsi untuk memuat berbagai kelas seperti *libraries*, *helpers*,
 29 dan *models*. Kelas dapat dimasukan pada *application/config/autoload.php* sesuai dengan
 30 petunjuk yang terdapat pada *file*. *File autoload* akan diinisiasi setiap aplikasi dijalankan
 31 sehingga pengguna tidak perlu memuat kelas tersebut berulang kali.

¹ 2.2 SharIF Judge [2]

SharIF Judge merupakan sebuah *Online Judge* percabangan dari *Sharif Judge* yang dibentuk oleh Mohammed Javad Naderi. *Sharif Judge* dibentuk menggunakan *CodeIgniter 3* dan dimodifikasi sesuai dengan kebutuhan di Informatika Universitas Katolik Parahyangan menjadi nama *SharIF Judge*. *SharIF Judge* dapat menilai kode berbahasa *C*, *C++*, *Java*, dan *Python* dengan mengunggah file ataupun mengetiknya secara langsung.

7 2.2.1 Struktur Aplikasi

⁸ Aplikasi *SharIF Judge* menggunakan struktur berbasis MVC yang membagi setiap file pada direktori. Kode 2.21 menunjukkan struktur aplikasi *SharIF Judge*.

Kode 2.21: Struktur aplikasi *SharIF Judge*

```
10
11 1   |-- application
12 2   |   |-- cache
13 3   |   |-- config
14 4   |   |-- controllers
15 5   |   |-- core
16 6   |   |-- helpers
17 7   |   |-- hooks
18 8   |   |-- language
19 9   |   |-- libraries
20 0   |   |-- logs
21 1   |   |-- models
22 2   |   |-- third_party
23 3   |   |-- vendor
24 4   |   |-- views
25 5   |-- assets
26 6   |   |-- images
27 7   |   |-- js
28 8   |   |-- styles
29 9   |-- restricted
30 0   |-- tester
31 1   |-- system
32 2   |-- assignments
```

34 2.2.2 Instalasi

SharIF Judge memerlukan beberapa persyaratan dan langkah dalam melakukan instalasi. Berikut merupakan persyaratan dan langkah-langkah melakukan instalasi *SharIF Judge*:

37 Persyaratan

38 SharIF Judge dapat dijalankan pada sistem operasi *Linux* dengan syarat sebagai berikut:

39 • Diperlukan *webserver* dengan versi PHP 5.3 atau lebih baru.

40 • Pengguna dapat menjalankan PHP pada *command line*. Pada *Ubuntu* diperlukan instalasi
41 paket *php5-cli*.

42 • *MySQL database* dengan ekstensi *Mysqli* untuk PHP atau *PostgreSql database*.

43 • PHP harus memiliki akses untuk menjalankan perintah melalui fungsi *shell_exec*. Kode 2.22
44 menunjukkan sintaks untuk melakukan pengetesan fungsi PHP pada *shell*.

Kode 2.22: Kode untuk melakukah pengetesan fungsi

```
45
46    1 echo shell_exec("php -v");
```

- *Tools* untuk melakukan kompilasi dan menjalankan kode yang dikumpulkan (*gcc*, *g++*, *javac*, *java*, *python2*, *python3*).
- *Perl* disarankan untuk diinstalasi untuk alasan ketepatan waktu, batas memori, dan memaksimalkan batas ukuran pada hasil kode yang dikirim.

5 Instalasi

1. Mengunduh versi terakhir dari *SharIF Judge* dan melakukan *unpack* pada direktori *public html*.
2. Memindahkan *folder system* dan *application* diluar direktori *public* dan mengubah *path* pada *index.php*(Opsional). Kode 2.23 menunjukan contoh path pada file *index.php*.

Kode 2.23: Contoh *path* pada halaman *index.php*

```
10
11 $system_path = '/home/mohammad/secret/system';
12 application_folder = '/home/mohammad/secret/application';
```

14. Membangun *database MySql* atau *PostgreSQL* untuk *SharIF Judge*. Jangan melakukan instalasi paket koneksi *database* apapun untuk *C*, *C++*, *Java*, atau *Python*.
15. Mengatur koneksi *database* pada file *application/config/database.example.php* dan menyimpannya dengan nama *database.php*. Pengguna dapat menggunakan awalan untuk nama tabel. Kode 2.24 menunjukan contoh pengaturan koneksi untuk nama, *host*, dan prefix pada *database*.

Kode 2.24: Contoh pengaturan koneksi untuk *database*

```
20
21 /* Enter database connection settings here: */
22 'dbdriver' => 'postgre', // database driver (mysqli, postgre)
23 'hostname' => 'localhost', // database host
24 'username' => '', // database username
25 'password' => '', // database password
26 'database' => '', // database name
27 'dbprefix' => 'shj_', // table prefix
```

29. Mengatur *RADIUS server* dan *mail server* pada file *application/config/secrets.example.php* dan menyimpannya dengan nama *secrets.php*.
30. Mengatur *application/cache/Twig* agar dapat ditulis oleh PHP.
31. Membuka halaman utama *SharIF Judge* pada *web browser* dan mengikuti proses instalasi.
32. Melakukan *Log in* dengan akun admin.
33. Memindahkan direktori **tester** dan **assignments** diluar direktori publik dan mengatur kedua direktori agar dapat ditulis oleh PHP. Selanjutnya Menyimpan *path* kedua direktori pada halaman *Settings*. Direktori **assignments** digunakan untuk menyimpan *file-file* yang diunggah agar tidak dapat diakses publik.

38 2.2.3 Clean URLs

39. Secara *default*, *index.php* merupakan bagian dari seluruh *urls* pada *SharIF judge*. Kode 2.25 menunjukan contoh dari URL *SharIF Judge*.

Kode 2.25: Contoh URL *SharIF Judge*

```
41
42 \verb|http://example.mjnaderi.ir/index.php/dashboard|
43 \verb|http://example.mjnaderi.ir/index.php/users/add|
```

- 1 Pengguna dapat menghapus `index.php` pada URL dan mendapatkan URL yang baik apabila
 2 sistem pengguna mendukung *URL rewriting*. Kode 2.26 menunjukkan hasil URL yang telah ditulis
 3 ulang.

Kode 2.26: Contoh hasil URL *SharIF Judge* yang telah ditulis ulang

```
4 \verb|http://example.mjnaderi.ir/dashboard|
5 1
6 2
7 3 \verb|http://example.mjnaderi.ir/users/add|
```

9 Cara Mengaktifkan *Clean URLs*

- 10 Pengguna dapat mengaktifkan *clean url* dengan cara sebagai berikut:
- 11 • Mengganti nama file `.htaccess2` pada direktori utama menjadi `.htaccess`.
 - 12 • Mengganti `$config['index_page'] = 'index.php';` menjadi `$config['index_page'] = '';`
 13 pada file `application\config\config.php`.

14 2.2.4 Users

- 15 Pada perangkat lunak SharIF Judge, pengguna dibagi menjadi 4 buah. Keempat pengguna tersebut
 16 adalah *Admins*, *Head Instructors*, *Instructors*, dan *Students*. Tabel 2.1 menunjukkan pembagian
 17 tingkat setiap pengguna. Setiap pengguna memiliki akses untuk aksi yang berbeda berdasarkan
 18 tingkatnya. Tabel 2.2 merupakan aksi yang dapat dilakukan oleh setiap pengguna.

Tabel 2.1: *Tabel tingkat pengguna*

| User Role | User Level |
|------------------------|------------|
| <i>Admin</i> | 3 |
| <i>Head Instructor</i> | 2 |
| <i>Instructor</i> | 1 |
| <i>Student</i> | 0 |

Tabel 2.2: *Tabel izin aksi setiap pengguna*

| Aksi | Admin | Head Instructor | Instructor | Student |
|---|-------|-----------------|------------|---------|
| Mengubah <i>Settings</i> | ✓ | ✗ | ✗ | ✗ |
| Menambah/Menghapus Pengguna | ✓ | ✗ | ✗ | ✗ |
| Mengubah Peran Pengguna | ✓ | ✗ | ✗ | ✗ |
| Menambah/Menghapus/Mengubah <i>Assignment</i> | ✓ | ✓ | ✗ | ✗ |
| Mengunduh <i>Test</i> | ✓ | ✓ | ✗ | ✗ |
| Menambah/Menghapus/Mengubah Notifikasi | ✓ | ✓ | ✗ | ✗ |
| <i>Rejudge</i> | ✓ | ✓ | ✗ | ✗ |
| Melihat/ <i>Pause</i> /Melanjutkan/ <i>Submission Queue</i> | ✓ | ✓ | ✗ | ✗ |
| Mendeteksi Kode yang Mirip | ✓ | ✓ | ✗ | ✗ |
| Melihat Semua Kode | ✓ | ✓ | ✓ | ✗ |
| Mengunduh Kode Final | ✓ | ✓ | ✓ | ✗ |
| Memilih <i>Assignment</i> | ✓ | ✓ | ✓ | ✓ |
| <i>Submit</i> | ✓ | ✓ | ✓ | ✓ |

1 Menambahkan Pengguna

- ² Admin dapat menambahkan pengguna melalui bagian *Add User* pada halaman *Users*. Admin harus
³ mengisi setiap informasi dimana baris yang diawali # merupakan komen dan setiap baris lainnya
⁴ mewakili pengguna. Kode 2.27 menunjukkan sintaks untuk menambah pengguna.

Kode 2.27: Sintaks untuk menambahkan pengguna

```
5 | USERNAME,EMAIL,DISPLAY-NAME,PASSWORD,ROLE
6 |
7 |
8 | * Usernames may contain lowercase letters or numbers and must be between 3 and 20 characters in length.
9 | * Passwords must be between 6 and 30 characters in length.
10| * You can use RANDOM[n] for password to generate random n-digit password.
11| * ROLE must be one of these: 'admin', 'head_instructor', 'instructor', 'student'
```

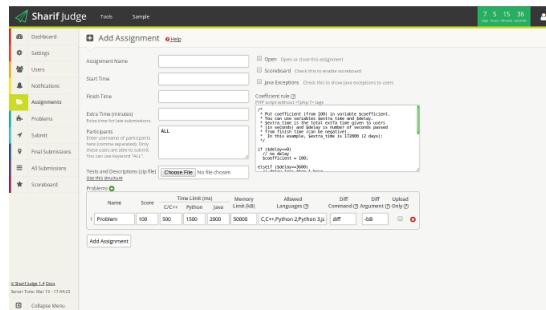
- ¹³ Kode 2.28 menunjukkan contoh untuk menambahkan pengguna.

Kode 2.28: Contoh kode untuk menambahkan pengguna

```
14 # This is a comment!
15 # This is another comment!
16
17 instructor,instructor@sharifjudge.ir,Instructor One,123456,head_instructor
18 instructor2,instructor2@sharifjudge.ir,Instructor Two,random[7],instructor
19 student1,st1@sharifjudge.ir,Student One,random[6],student
20 student2,st2@sharifjudge.ir,Student Two,random[6],student
21 student3,st3@sharifjudge.ir,Student Three,random[6],student
22 student4,st4@sharifjudge.ir,Student Four,random[6],student
23 student5,st5@sharifjudge.ir,Student Five,random[6],student
24 student6,st6@sharifjudge.ir,Student Six,random[6],student
25 student7,st7@sharifjudge.ir,Student Seven,random[6],student
```

27 2.2.5 Menambah Assignment

- ²⁸ Pengguna dapat menambahkan *assignment* baru melalui bagian *Add* pada halaman *Assignments*.
²⁹ yang ditunjukkan pada Gambar 2.2.



Gambar 2.2: Tampilan halaman *SharIF Judge* untuk menambahkan *assignment*

- Berikut merupakan beberapa pengaturan pada halaman *Add Assignments*:

- 31 • Assignment Name

- Assignment* akan ditampilkan sesuai dengan masukan pada daftar *assignment*.

- ### 33 • Start Time

- Pengguna tidak dapat mengumpulkan *assignment* sebelum waktu dimulai("Start Time"). Format pengaturan waktu untuk waktu mulai adalah MM/DD/YYYY HH:MM:SS dengan contoh 08/31/2013 12:00:00.

1 • ***Finish Time, Extra Time***

2 Pengguna tidak dapat mengumpulkan *assignment* setelah *Finish Time* dan *Extra Time* bera-
 3 akhir. Pengumpulan *Assignment* pada *Extra Time* akan dikalikan sesuai dengan koefisien. Peng-
 4 guna harus menulis skrip PHP untuk menghitung koefisien pada *field Coefficient Rule*. Format
 5 pengaturan waktu untuk waktu selesai sama seperti waktu mulai yakni MM/DD/YYYY HH:MM:SS
 6 dan format waktu tambahan menggunakan menit dengan contoh 120 (2 jam) atau 48*60 (2
 7 hari).

8 • ***Participants***

9 Pengguna dapat memasukan *username* setiap partisipan atau menggunakan kata kunci *ALL* un-
 10 tuk membiarkan seluruh pengguna melakukan pengumpulan. Contoh: **admin**, **instructor1**,
 11 **instructor2**, **student1**, **student2**, **student3**, **student4**.

12 • ***Tests***

13 Pengguna dapat mengunggah *test case* dalam bentuk *zip file* sesuai dengan struktur pada
 14 subbab [2.2.7](#).

15 • ***Open***

16 Pengguna dapat membuka dan menutup *assigment* untuk pengguna *student* melalui pilihan
 17 ini. Pengguna selain *student* tetap dapat mengumpulkan *assignment* apabila *assignment* sudah
 18 ditutup.

19 • ***Score Board***

20 Pengguna dapat menghidupkan dan mematikan *score board* melalui pilihan ini.

21 • ***Java Exceptions***

22 Pengguna dapat menghidupkan dan mematikan fungsi untuk menunjukan *java exceptions*
 23 kepada pengguna *students* dan tidak akan memengaruhi kode yang sudah di *judge* sebelumnya.
 24 Kode [2.29](#) menunjukan tampilan apabila fitur *java exceptions* dinyalakan:

Kode 2.29: Contoh tampilan fitur *Java Exceptions*

```

25
26   1 Test 1
27   2 ACCEPT
28   3 Test 2
29   4 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
30   5 Test 3
31   6 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
32   7 Test 4
33   8 ACCEPT
34   9 Test 5
35  10 ACCEPT
36  11 Test 6
37  12 ACCEPT
38  13 Test 7
39  14 ACCEPT
40  15 Test 8
41  16 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
42  17 Test 9
43  18 Runtime Error (java.lang.StackOverflowError)
44  19 Test 10

```

20 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)

3 • ***Archived Assignment***

4 Pengguna dapat menghidupkan fitur ini dan *assignment* akan dibentuk dengan waktu selesai
 5 2038-01-18 00:00:00 (UTC + 7) dengan kata lain pengguna memiliki waktut tidak terhingga
 6 untuk mengumpulkan *assignment*.

7 • ***Coefficient Rule***

8 Pengguna dapat menuliskan skrip PHP pada bagian ini yang akan menghitung koefisien dikalikan
 9 dengan skor. Pengguna harus memasukan koefisien dari 100 dalam variabel `$coefficient`.
 10 Pengguna dapat menggunakan variabel `$extra_time` dan `$delay`. `$extra_time` merupakan
 11 total dari waktu tambahan yang diberikan kepada pengguna dalam detik sedangkan `$delay`
 12 merupakan waktu dalam detik yang melewati waktu selesai yang dapat diisikan berupa negatif.
 13 Skrip PHP pada bagian ini tidak boleh mengandung tag `<?php`, `<?`, dan `?>`. Kode 2.30
 14 menunjukan contoh skrip dimana `$extra_time` adalah 172800 detik atau 2 hari.

Kode 2.30: Contoh skrip PHP

```
15 if ($delay<=0)
16   // no delay
17   $coefficient = 100;
18
19
20 elseif ($delay<=3600)
21   // delay less than 1 hour
22   $coefficient = ceil(100-((30*$delay)/3600));
23
24 elseif ($delay<=86400)
25   // delay more than 1 hour and less than 1 day
26   $coefficient = 70;
27
28 elseif ((($delay-86400)<=3600)
29   // delay less than 1 hour in second day
30   $coefficient = ceil(70-((20*($delay-86400))/3600));
31
32 elseif ((($delay-86400)<=86400)
33   // delay more than 1 hour in second day
34   $coefficient = 50;
35
36 elseif ($delay > $extra_time)
37   // too late
38   $coefficient = 0;
```

40 • ***Name***

41 Merupakan nama dari masalah pada *assignments*.

42 • ***Score***

43 Merupakan skor dari masalah pada *assignments*.

44 • ***Time Limit***

45 Pengguna dapat menentukan batas waktu untuk menjalankan kode dalam satuan milidetik.
 46 Bahasa *Python* dan *Java* biasanya memiliki waktu lebih lambat dari *C/C++* sehingga
 47 membutuhkan waktu lebih lama.

1 • ***Memory Limit***

2 Pengguna dapat menentukan batas memori dalam *kilobytes*, namun penggunaan pembatasan
3 memori tidak terlalu akurat.

4 • ***Allowed Languages***

5 Pengguna dapat menentukan bahasa setiap masalah pada *assignment* yang dipisahkan oleh
6 koma. Terdapat beberapa bahasa yang tersedia yaitu *C*, *C++*, *Java*, *Python 2*, *Python 3*,
7 *Zip*, *PDF*, dan *TXT*. Pengguna dapat memakai *Zip*, *PDF*, dan *TXT* apabila opsi *Upload Only*
8 dinyalakan. Contoh : *C*, *C++*, *Zip* atau *Python 2,Python 3*.

9 • ***Diff Command***

10 *Diff Command* digunakan untuk membandingkan hasil keluaran dengan keluaran yang benar.
11 Secara *default*, *SharIF Judge* menggunakan *diff*, namun pengguna dapat menggantinya pada
12 bagian ini dan bagian ini tidak boleh mengandung spasi.

13 • ***Diff Arguments***

14 Pengguna dapat mengatur argumen untuk *diff arguments* pada bagian ini. Pengguna dapat
15 melihat *man diff* untuk daftar lengkap argumen *diff*. *SharIF Judge* terdapat dua buah opsi
16 baru yakni *ignore* dan *identical*.

- 17 – *ignore* : *SharIF Judge* mengabaikan semua baris baru dan spasi.
18 – *identical* : *SharIF Judge* tidak mengabaikan apapun namun, keluaran dari *file* yang
19 dikumpulkan harus identik dengan *test case* agar dapat diterima.

20 • ***Upload Only***

21 Pengguna dapat menghidupkan *Upload only*, namun *SharIF Judge* tidak akan menilai masalah
22 tersebut. Pengguna dapat memakai *Zip*, *PDF*, dan *TXT* pada *allowed languages* apabila
23 pengguna menghidupkan bagian ini.

24 **2.2.6 *Sample Assignment***

25 Berikut merupakan contoh dari *assignment* untuk melakukan pengujian perangkat lunak *SharIF*
26 *Judge*. Penambahan *Assignment* dapat dilakukan dengan memencet tombol *Add* pada halaman
27 *Assignment*.

28 ***Problems***

29 1. *Problem 1 (Sum)*: Program pengguna dapat membaca *integer n*, membaca *n integers* dan
30 mengeluarkan hasil dari *integer* tersebut. Tabel 2.3 menunjukkan contoh pemasukan dan
31 keluaran untuk *Problem 1*.

Tabel 2.3: Tabel contoh masukan dan keluaran Problem 1

| Sample Input | Sample Output |
|--------------|---------------|
| 5 | |
| 53 78 0 4 9 | 145 |

- 1 2. *Problem 2 (Max)*: Program pengguna dapat membaca *integer n*, membaca *n integer*, dan
 2 mengeluarkan total dari dua buah *integer* terbesar diantara *n integer*. Tabel 2.4 menunjukkan
 3 contoh pmasukan dan keluaran untuk *Problem 2*.

Tabel 2.4: Tabel contoh masukan dan keluaran Problem 2

| Sample Input | Sample Output |
|----------------------------------|---------------|
| 7 162 173 159 164 181 158 175 | 356 |

- 4 3. *Problem 3 (Upload)*: Pengguna dapat mengunggah *file c* dan *zip* dan *problem* ini tidak akan
 5 dinilai karena hanya bersifat *Upload Only*.

6 Tests

- 7 Pengguna dapat menemukan *file zip* pada direktori *Assignments*. Kode 2.31 merupakan susunan
 8 pohon dari ketiga *problems* diatas.

Kode 2.31: Susunan pohon untuk ketiga *problems*

```

9  .
10 1 |-- p1
11 2 |   |-- in
12 3 |   |   |-- input1.txt
13 4 |   |   |-- input2.txt
14 5 |   |   |-- input3.txt
15 6 |   |   |-- input4.txt
16 7 |   |   |-- input5.txt
17 8 |   |   |-- input6.txt
18 9 |   |   |-- input7.txt
19 0 |   |   |-- input8.txt
20 1 |   |   |-- input9.txt
21 2 |   |   |-- input10.txt
22 3 |   |-- out
23 4 |       |-- output1.txt
24 5 |       |-- output2.txt
25 6 |       |-- tester.cpp
26 7 |       |-- desc.md
27 8 |-- p2
28 9 |   |-- in
29 0 |   |   |-- input1.txt
30 1 |   |   |-- input2.txt
31 2 |   |   |-- input3.txt
32 3 |   |   |-- input4.txt
33 4 |   |   |-- input5.txt
34 5 |   |   |-- input6.txt
35 6 |   |   |-- input7.txt
36 7 |   |   |-- input8.txt
37 8 |   |   |-- input9.txt
38 9 |   |   |-- input10.txt
39 0 |   |-- out
40 1 |       |-- output1.txt
41 2 |       |-- output2.txt
42 3 |       |-- output3.txt
43 4 |       |-- output4.txt
44 5 |       |-- output5.txt
45 6 |       |-- output6.txt
46 7 |       |-- output7.txt
47 8 |       |-- output8.txt
48 9 |       |-- output9.txt
49 0 |       |-- output10.txt
50 1 |       |-- desc.md
51 2 |       |-- Problem2.pdf

```

```

213 |--- p3
214 |   --- desc.md
215 --- SampleAssignment.pdf

```

5 *Problem 1* menggunakan metode "*Tester*" untuk mengecek hasil keluaran, sehingga memiliki
 6 *file tester.cpp* (*Tester Script*). *Problem 2* menggunakan metode "*Output Comparison*" untuk
 7 mengecek hasil keluaran, sehingga memiliki dua buah direktori *in* dan *out* yang berisi *test case*.
 8 *Problem 3* merupakan *problem "Upload-Only"* sehingga tidak perlu dilakukan pengecekan.

9 *Sample Solutions*

10 Berikut merupakan *sample solutions* untuk *problem 1*:

11 Kode refkode:samplesolutions1 menunjukkan contoh solusi untuk bahasa *C*.

Kode 2.32: Contoh skrip PHP

```

12 #include<stdio.h>
13 1 int main(){
14 2     int n;
15 3     scanf("%d",&n);
16 4     int i;
17 5     int sum =0 ;
18 6     int k;
19 7     for(i=0 ; i<n ; i++){
20 8         scanf("%d",&k);
21 9         sum+=k;
22 0     }
23 1     printf("%d\n",sum);
24 2     return 0;
25 3 }
26 4

```

28 Kode refkode:samplesolutions2 menunjukkan contoh solusi untuk bahasa *C++*.

Kode 2.33: Contoh solusi *problem 1* bahasa *C++*

```

29 #include<stdio.h>
30 1 int main(){
31 2     int n;
32 3     scanf("%d",&n);
33 4     int i;
34 5     int sum =0 ;
35 6     int k;
36 7     for(i=0 ; i<n ; i++){
37 8         scanf("%d",&k);
38 9         sum+=k;
39 0     }
40 1     printf("%d\n",sum);
41 2     return 0;
42 3 }
43 4

```

45 Kode refkode:samplesolutions3 menunjukkan contoh solusi untuk bahasa *Java*.

Kode 2.34: Contoh solusi *problem 1* bahasa *Java* label

```

46 import java.util.Scanner;
47 1 class sum
48 2 {
49 3     public static void main(String[] args)
50 4     {
51 5         Scanner sc = new Scanner(System.in);
52 6         int n = sc.nextInt();
53 7         int sum =0;
54 8         for (int i=0 ; i<n ; i++)
55 9         {
56 10             int a = sc.nextInt();
57 11             sum += a;
58 12         }
59 13         System.out.println(sum);
60 14     }
61 15 }
62 16

```

- 1 Berikut merupakan contoh solusi untuk *problem 2*:
 2 Kode refkode:samplesolutions21 menunjukan contoh solusi untuk bahasa *C*

Kode 2.35: Contoh solusi *problem 2* bahasa *C*

```

3 1 #include<stdio.h>
4 2 int main(){
5 3     int n , m1=0, m2=0;
6 4     scanf("%d",&n);
7 5     for(;n--;){
8 6         int k;
9 7         scanf("%d",&k);
10 8         if(k>=m1){
11 9             m2=m1;
12 10            m1=k;
13 11        }
14 12        else if(k>m2)
15 13            m2=k;
16 14    }
17 15    printf("%d",m1+m2);
18 16    return 0;
19 17 }
```

- 22 Kode refkode:samplesolutions22 menunjukan contoh solusi untuk bahasa *C++*

Kode 2.36: Contoh solusi *problem 2* bahasa *C++*

```

23 1 #include<iostream>
24 2 using namespace std;
25 3 int main(){
26 4     int n , m1=0, m2=0;
27 5     cin >> n;
28 6     for(;n--;){
29 7         int k;
30 8         cin >> k;
31 9         if(k>=m1){
32 10            m2=m1;
33 11            m1=k;
34 12        }
35 13        else if(k>m2)
36 14            m2=k;
37 15    }
38 16    cout << (m1+m2) << endl ;
39 17    return 0;
40 18 }
```

43 2.2.7 Test Structure

- 44 Penambahan *assignment* harus disertakan dengan *file zip* berisikan *test cases*. *File zip* ini sebaiknya
 45 berisikan *folder* untuk setiap masalah dengan nama *p1*, *p1*, dan *p3* selain masalah *Upload-Only*.

46 Metode Pemeriksaan

- 47 Terdapat dua buah metode untuk melakukan pemeriksaan yakni metode *Input Output* dan metode
 48 *Tester*. Berikut merupakan penjelasan untuk setiap metode pemeriksaan:

49 1. Metode perbandingan *Input Output*

50 Pada metode ini, pengguna harus memberi masukan dan keluaran pada *folder problem*. Per-
 51 angkat lunak akan memberikan *file test input* ke kode pengguna dan melakukan perbandingan
 52 dengan hasil keluaran kode pengguna. *File input* harus berada didalam *folder in* dengan
 53 nama *input1.txt*, *input2.txt*, dst. *File output* harus berada di dalam *folder out* dengan nama
 54 *output1.txt* dan *output2.txt*.

55 2. Metode perbandingan *Tester*

1 Pada metode ini, pengguna harus menyediakan *file input test*, sebuah *file C++*, dan *file output test* yang bersifat opsional. Perangkat lunak akan memberikan *file input test* ke kode pengguna dan mengambil keluaran pengguna. Selanjutnya, *tester.cpp* akan mengambil masukan pengguna, keluaran tes, dan keluaran program pengguna. Jika keluaran pengguna benar maka perangkat lunak akan mengembalikan 1 sedangkan apabila salah maka perangkat lunak akan mengembalikan 0. Kode 2.37 menunjukkan templat yang dapat digunakan pengguna untuk menuliskan *file tester.cpp*.

Kode 2.37: Templat kode *tester.cpp*

```

8  /*
9   * tester.cpp
10  */
11
12
13 #include <iostream>
14 #include <fstream>
15 #include <string>
16 using namespace std;
17 int main(int argc, char const *argv[])
18 {
19
20     ifstream test_in(argv[1]); /* This stream reads from test's input file */
21     ifstream test_out(argv[2]); /* This stream reads from test's output file */
22     ifstream user_out(argv[3]); /* This stream reads from user's output file */
23
24     /* Your code here */
25     /* If user's output is correct, return 0, otherwise return 1 */
26
27     ...
28 }
29 }
```

31 *Sample File*

32 Pengguna dapat menemukan *file sample test* pada direktori *assignments*. Kode 2.38 menunjukkan contoh dari pohon *file sample test*.

Kode 2.38: Contoh pohon *file* dari *sample test*

```

34 .
35 |
36 |--- p1
37 | |--- in
38 | | |--- input1.txt
39 | | |--- input2.txt
40 | | |--- input3.txt
41 | | |--- input4.txt
42 | | |--- input5.txt
43 | | |--- input6.txt
44 | | |--- input7.txt
45 | | |--- input8.txt
46 | | |--- input9.txt
47 | | |--- input10.txt
48 | |--- out
49 | | |--- output1.txt
50 | |--- tester.cpp
51 |
52 |--- p2
53 | |--- in
54 | | |--- input1.txt
55 | | |--- input2.txt
56 | | |--- input3.txt
57 | | |--- input4.txt
58 | | |--- input5.txt
59 | | |--- input6.txt
60 | | |--- input7.txt
61 | | |--- input8.txt
62 | | |--- input9.txt
63 | | |--- input10.txt
64 | |--- out
65 | | |--- output1.txt
66 | | |--- output2.txt
67 | | |--- output3.txt
```

```

1   33 |   |   |-- output4.txt
2   34 |   |   |-- output5.txt
3   35 |   |   |-- output6.txt
4   36 |   |   |-- output7.txt
5   37 |   |   |-- output8.txt
6   38 |   |   |-- output9.txt
7   39 |   |   --- output10.txt

```

9 *Problem 1* menggunakan metode perbandingan *Tester*, sehingga memiliki file **tester.cpp**.
10 Kode 2.39 menunjukan file untuk pengujian *problem 1*.

Kode 2.39: Kode metode perbandingan *tester* dengan bahasa *tester.cpp*

```

11  /*
12  * tester.cpp
13  */
14
15
16  #include <iostream>
17  #include <fstream>
18  #include <string>
19  using namespace std;
20  int main(int argc, char const *argv[])
21 {
22
23     ifstream test_in(argv[1]); /* This stream reads from test's input file */
24     ifstream test_out(argv[2]); /* This stream reads from test's output file */
25     ifstream user_out(argv[3]); /* This stream reads from user's output file */
26
27     /* Your code here */
28     /* If user's output is correct, return 0, otherwise return 1 */
29     /* e.g.: Here the problem is: read n numbers and print their sum: */
30
31     int sum, user_output;
32     user_out >> user_output;
33
34     if ( test_out.good() ) // if test's output file exists
35     {
36         test_out >> sum;
37     }
38     else
39     {
40         int n, a;
41         sum=0;
42         test_in >> n;
43         for (int i=0 ; i<n ; i++){
44             test_in >> a;
45             sum += a;
46         }
47     }
48
49     if (sum == user_output)
50         return 0;
51     else
52         return 1;
53 }
54

```

56 *Problem 2* menggunakan metode perbandingan *Input Output* sehingga memiliki *folder in* dan
57 *folder out* berisikan *test case*. Sedangkan *problem 3* merupakan *Upload Only*, sehingga tidak
58 memiliki *folder* apapun.

59 2.2.8 Deteksi Kecurangan

60 *SharIF Judge* menggunakan Moss untuk mendeteksi kode yang serupa. Moss atau *Measure of*
61 *Software Similarity* merupakan sistem otomatis untuk menentukan kesamaan atau kemiripan dalam
62 program. Penggunaan utama Moss adalah untuk melakukan pemeriksaan plagiarisme pada kelas
63 *programming*. Pengguna dapat mengirim hasil kode terakhir atau *Final Submission* menuju *server*
64 Moss dengan satu klik. Penggunaan Moss memiliki beberapa hal yang harus diatur oleh pengguna
65 sebagai berikut:

1. Pengguna harus mendapatkan *Moss User id* dan melakukan pengaturan pada *SharIF Judge*.
 Untuk mendapatkan *Moss User id*, pengguna dapat mendaftar pada halaman <http://theory.stanford.edu/~aiken/moss/>. Pengguna selanjutkan akan mendapatkan *email* berupa skrip *perl* berisikan *user id* pengguna. Kode 2.40 menunjukan contoh dari potongan skrip *perl*.

Kode 2.40: Contoh potongan skrip *perl*

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
1   ...
2
3
4 $server = 'moss.stanford.edu';
5 $port = '7690';
6 $noreq = "Request not sent.";
7 $usage = "usage: moss [-x] [-l language] [-d] [-b basefile1] ... [-b basefilen] [-m #] [-c \"string\"] file1 file2 file3
8     ...";
9 #
10 # The userid is used to authenticate your queries to the server; don't change it!
11 #
12 $userid=YOUR_MOSS_USER_ID;
13 #
14 #
15 # Process the command line options. This is done in a non-standard
16 # way to allow multiple -b's.
17 #
18 $opt_l = "c";    # default language is c
19 $opt_m = 10;
20 $opt_d = 0;
21 ...
22 ...

```

User id pada skrip *perl* diatas dapat digunakan pada *SharIF Judge* untuk mendeteksi kecurangan. Pengguna dapat menyimpan *userid* pada halaman *SharIF Judge Moss* dan *SharIF Judge* akan menggunakan *userid* tersebut untuk mendeteksi kecurangan.

2. *Server* pengguna harus memiliki instalasi *perl* untuk menggunakan *Moss*.
 3. Pengguna dianjurkan untuk mendeteksi kode yang mirip setelah *assignment* selesai karena *SharIF Judge* akan mengirimkan hasil akhir kepada *Moss* dan pengguna *students* dapat mengganti hasil akhir sebelum *assignment* selesai.

37 2.2.9 Keamanan

- 38 *SharIF Judge* terdapat beberapa langkah untuk melakukan pengaturan keamanan perangkat lunak.
 39 Berikut merupakan langkah untuk melakukan pengaturan keamanan *SharIF Judge*:

40 1. Menggunakan *Sandbox*

Pengguna harus memastikan untuk menggunakan *sandbox* untuk bahasa *C/C++* dan menyalakan *Java Security Manager (Java Policy)* untuk bahasa *java*. Penggunaan *sandbox* dapat dilihat pada subbab 2.2.10.

44 2. Menggunakan *Shield*

Pengguna harus memastikan untuk menggunakan *shield* untuk bahasa *C, C++,* dan *Python*. Penggunaan *shield* dapat dilihat pada subbab 2.2.11.

3. Menjalankan sebagai *Non-Priviledge User*

Pengguna diwajibkan menjalankan kode yang telah dikumpulkan sebagai *non-priviledge user*. *Non-Priviledge User* adalah *user* yang tidak memiliki akses jaringan, tidak dapat menulis *file* apapun, dan tidak dapat membangun banyak proses.

Diasumsikan bahwa PHP dijalankan sebagai pengguna **www-data** pada server. Membangun *user* baru **restricted-user** dan melakukan pengaturan *password*. Selanjutnya, jalankan **sudo visudo** dan tambahkan kode **www-data ALL=(restricted_user) NOPASSWD: ALL** pada baris terakhir *file sudoers*.

- Pada *file tester\runcode.sh* ubah kode:

Kode 2.41: Kode *runcode.sh* sebelum ditambahkan

```
10
11  if $TIMEOUT_EXISTS; then
12    timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
13  else
14    $CMD <$IN >out 2>err
15  fi
```

menjadi Kode 2.42.

Kode 2.42: Kode *runcode.sh* setelah ditambahkan

```
18
19  if $TIMEOUT_EXISTS; then
20    sudo -u restricted_user timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
21  else
22    sudo -u restricted_user $CMD <$IN >out 2>err
23  fi
```

dan *uncomment* baris yang ditunjukan pada Kode 2.43.

Kode 2.43: Kode *runcode.sh* awal

```
26
27  sudo -u restricted_user pkill -9 -u restricted_user
```

- Mematikan akses jaringan untuk *restricted_user*

Pengguna dapat mematikan akses jaringan untuk *restricted_user* di *linux* menggunakan *iptables*. Setelah dimatikan lakukan pengujian menggunakan *ping* sebagai *restricted_user*.

- Menolak izin menulis

Pastikan tidak ada direktori atau *file* yang dapat ditulis oleh *restricted_user*.

- Membatasi jumlah proses

Pengguna dapat membatasi jumlah proses dari *restricted_user* dengan menambahkan melalui *file /etc/security/limits.conf* seperti yang ditunjukan pada Kode 2.44.

Kode 2.44: Contoh kode untuk membatasi jumlah proses

```
38
39  1 restricted_user soft nproc 3
40  2 restricted_user hard nproc 5
```

4. Menggunakan dua *server*

Pengguna dapat menggunakan dua *server* untuk antar muka web, menangani permintaan web, dan mengguna *server* lainnya untuk menjalankan kode yang sudah dikumpulkan. Penggunaan dua *server* mengurangi risiko menjalankan kode yang sudah dikumpulkan. Pengguna harus mengganti *source code SharIF Judge* untuk memakai hal ini.

1 2.2.10 Sandboxing

2 *SharIF Judge* menjalankan banyak *arbitrary codes* yang pengguna kumpulkan. *SharIF Judge*
 3 harus menjalankan kode pada lingkungan terbatas sehingga memerlukan perkakas untuk mela-
 4 kukan *sandbox* kode yang sudah dikumpulkan. Pengguna dapat meningkatkan keamanan dengan
 5 menghidupkan *shield* bersama dengan *Sandbox*.

6 C/C++ Sandboxing

7 *SharIF Judge* menggunakan *EasySandbox* untuk melakukan *sandboxing* kode *C/C++*. *EasySan-
 8 dbox* berguna untuk membatasi kode yang berjalan menggunakan *seccomp*. *Seccomp* merupakan
 9 mekanisme *sandbox* pada *Linux kernel*. Secara *default* *EasySandbox* dimatikan pada *SharIF Judge*,
 10 namun pengguna dapat menghidupkannya melalui halaman *Settings*. Selain itu, pengguna juga
 11 harus melakukan "*build EasySandbox*" sebelum menyalakannya. Berikut merupakan cara melakukan
 12 *build EasySandbox*:

13 File *EasySandbox* terdapat pada direktori **tester/easysandbox**. Untuk membangun *EasySan-
 14 dbox* jalankan:

Kode 2.45: Kode *runcode.sh* awal

```
15 1 $ cd tester/easysandbox
16 2 $ chmod +x runalltests.sh
17 3 $ chmod +x runtest.sh
18 4 $ make runtests
```

21 Jika keluaran berupa *message All test passed!* maka, *EasySandbox* berhasil dibangun dan
 22 dapat dinyalakan pada perangkat lunak.

23 Java Sandboxing

24 Secara *default*, *Java Sandbox* sudah dinyalakan menggunakan *Java Security Manager*. Pengguna
 25 dapat menghidupkan atau mematikannya pada halaman *Settings*.

26 2.2.11 Shield

27 *Shield* merupakan mekanisme sangat simpel untuk melarang jalannya kode yang berpotensi berba-
 28 haya. *Shield* bukan solusi *sandboxing* karena *shield* hanya menyediakan proteksi sebagian terhadap
 29 serangan kecil. Proteksi utama terhadap kode tidak terpercaya adalah dengan menghidupkan
 30 *Sandbox* yang dapat dilihat pada subbab 2.2.10.

31 Shield untuk C/C++

32 Dengan menghidupkan *shield* untuk *c/c++*, *SharIF Judge* hanya perlu menambahkan **#define** pada
 33 awal kode yang dikumpulkan sebelum menjalankannya. Sebagai contoh, pengguna dapat melarang
 34 penggunaan *goto* dengan menambahkan Kode 2.46 pada awal kode yang sudah dikumpulkan.

Kode 2.46: Kode *shield* untuk melarang penggunaan *goto*

```
35 1 #define goto YouCannotUseGoto
```

- 1 Dengan Kode 2.46, semua kode yang menggunakan `goto` akan mendapatkan *compilation error*.
 2 Apabila pengguna menghidupkan *shield*, semua kode yang mengandung `#undef` akan mendapatkan
 3 *compilation error*. *Shield* untuk *C* dan *C++* dapat dikonfigurasikan menggunakan cara berikut:
 4 • Menghidupkan *shield* untuk *C/C++*
 5 Pengguna dapat menghidupkan atau mematikan *shield* pada halaman *settings*.
 6 • Menambahkan aturan untuk *C/C++*
 7 Daftar aturan `#define` untuk bahasa *C* terdapat pada halaman `tester/shield/defc.h` dan
 8 `tester/shield/defcpp.h` untuk bahasa *C++*. Pengguna dapat menambahkan aturan baru
 9 pada *file* tersebut pada halaman *settings*. Kode refkode:define menunjukkan contoh sintaks
 10 pada untuk menambahkan aturan *shield*.

Kode 2.47: Sintaks aturan `#define`

```

11 /*
12 13
14 14 @file defc.h
15 15 There should be a newline at end of this file.
16 16 Put the message displayed to user after // in each line
17 17
18 18 e.g. If you want to disallow goto, add this line:
19 19 #define goto errorNo13 //Goto is not allowed
20 20
21 21 */
22 22
23 23 #define system errorNo1 // "system" is not allowed
24 24 #define freopen errorNo2 //File operation is not allowed
25 25 #define fopen errorNo3 //File operation is not allowed
26 26 #define fprintf errorNo4 //File operation is not allowed
27 27 #define fscanf errorNo5 //File operation is not allowed
28 28 #define feof errorNo6 //File operation is not allowed
29 29 #define fclose errorNo7 //File operation is not allowed
30 30 #define ifstream errorNo8 //File operation is not allowed
31 31 #define ofstream errorNo9 //File operation is not allowed
32 32 #define fork errorNo10 //Fork is not allowed
33 33 #define clone errorNo11 //Clone is not allowed
34 34 #define sleep errorNo12 //Sleep is not allowed

```

36 Pada akhir baris *file defc.h* dan *defcpp.h* harus terdapat baris baru. Terdapat banyak
 37 aturan yang tidak dapat digunakan pada *g++*, seperti aturan `#define fopen errorNo3`
 38 untuk bahasa *C++* karena akan menghasilkan *compile error*.

39 *Shield* untuk *Python*

40 Penggunaan *shield* untuk *python* dapat dihidupkan melalui halaman *settings*. Dengan menghidupkan
 41 *shield* untuk *python*, *SharIF Judge* hanya menambahkan beberapa kode pada baris awal kode
 42 yang sudah dikumpulkan sebelum dijalankan. Penambahan kode digunakan untuk mencegah
 43 pemakaian fungsi berbahaya. Kode-kode tersebut terletak pada *file tester/shield/shield_py2.py*
 44 dan *tester/shield/shield_py3.py*. Kode 2.48 menunjukkan cara untuk keluar dari *shield* untuk
 45 *python* menggunakan fungsi yang telah di daftar hitamkan.

Kode 2.48: Cara keluar dari *shield* untuk *python*

```

46
47 1
48 2 # @file shield_py3.py
49 3
50 4 import sys
51 5 sys.modules['os']=None
52 6
53 7 BLACKLIST = [
54 8     '__import__', # deny importing modules
55 9     'eval', # eval is evil
56 0     'open',

```

```

11   'file',
12   'exec',
13   'execfile',
14   'compile',
15   'reload',
16   #'input'
17 ]
18 for func in BLACKLIST:
19     if func in __builtins__.__dict__:
20         del __builtins__.__dict__[func]

```

2.3 CodeIgniter 4[3]

CodeIgniter 4 merupakan versi terbaru dari framework *CodeIgniter* yang berfungsi untuk membantu pembentukan situs web. *CodeIgniter 4* dapat dipasang menggunakan *composer* ataupun dipasang secara manual dengan mengunduhnya dari situs web resmi. Berikut merupakan sintaks untuk melakukan pemasangan menggunakan *composer*.

```
composer create-project codeigniter4/appstarter project-root
```

Sintaks diatas akan mengunduh dan melakukan instalasi projek *CodeIgniter 4* dengan nama *project-root*. Sintaks *codeigniter4/appstarter* berfungsi untuk mengunduh aplikasi *skeleton* dari projek *CodeIgniter 4* yang berisikan kebutuhan data untuk melakukan pembangunan sebuah aplikasi. Setelah dilakukan pemasangan, *CodeIgniter 4* memiliki lima buah direktori dengan struktur aplikasi sebagai berikut:

- *app/*

Direktori *app* berisikan semua kode yang dibutuhkan untuk menjalankan aplikasi web yang dibentuk. Direktori ini terdiri dari beberapa direktori yang terletak didalamnya sebagai berikut:

- *Config/* berfungsi dalam menyimpan *file* konfigurasi aplikasi web pengguna.
- *Controllers/* berfungsi sebagai penentu alur program yang dibentuk.
- *Database/* berfungsi sebagai penyimpan *file migrations* dan *seeds*.
- *Filters/* berfungsi dalam menyimpan *file* kelas *filter*.
- *Helpers/* berfungsi dalam menyimpan koleksi fungsi mandiri.
- *Language/* berfungsi sebagai tempat penyimpanan *string* dalam berbagai bahasa.
- *Libraries/* berfungsi dalam penyimpanan kelas yang tidak termasuk kategori lain.
- *Models/* berfungsi untuk merepresentasikan data dari *database*.
- *ThirdParty/* *library* *ThirdParty* yang dapat digunakan pada aplikasi.
- *Views/* berisikan *file* HTML yang ditampilkan kepada pengguna.

- *public/*

Direktori *public* merupakan akar dari situs web dan berisikan data-data yang dapat diakses oleh pengguna melalui *browser*. Direktori ini berisikan *file .htaccess*, *index.php*, dan seluruh *assets* dari aplikasi *website* yang dibentuk seperti gambar, CSS, ataupun *JavaScript*.

- *writable/*

Direktori *writable* berisikan data-data yang mungkin perlu ditulis oleh perangkat lunak seperti *file cache*, *logs*, dan *uploads*. Pengguna dapat menambahkan direktori baru sesuai dengan kebutuhan sehingga menambahkan keamanan pada direktori utama.

1 • **tests/**

2 Direktori ini menyimpan *file test* dan tidak perlu dipindahkan ke *server produksi*. Direktori
 3 ini terdapat direktori **_support** berisikan berbagai jenis kelas *mock* dan keperluan yang dapat
 4 dipakai pengguna dalam menulis *tests*.

5 • **vendor/ atau system/**

6 Direktori ini berisikan *file* yang diperlukan dalam menjalani *framework* dan tidak boleh
 7 dimodifikasi oleh pengguna. Pengguna dapat melakukan *extend* atau membangun kelas baru
 8 untuk menambahkan fungsi yang diperlukan.

9 **2.3.1 Models- Views-Controllers**

10 *CodeIgniter 4* menggunakan struktur *MVC* untuk mengatur *file* agar lebih simpel dalam menemukan
 11 *file* yang diperlukan. *MVC* menyimpan data, presentasi, dan alur program dalam *file* yang berbeda.

12 **Models**

13 *Models* berfungsi dalam menyimpan dan mengambil data dari tabel spesifik pada *database*. Data
 14 tersebut dapat berupa pengguna, pemberitahuan blog, transaksi, dll. *Models* pada umumnya
 15 disimpan pada direktori **app/Models** dan memiliki *namespace* sesuai dengan lokasi dari *file* tersebut.
 16 Kode 2.49 menunjukkan contoh dari *models* bernama **UserModel**.

Kode 2.49: Contoh *Models*

```

17
18 1 <?php
19 2
20 3
21 4 namespace App\Models;
22 5
23 6 use CodeIgniter\Model;
24 7
25 8 class UserModel extends Model
26 9 {
27 10     protected $table      = 'users';
28 11     protected $primaryKey = 'id';
29 12
30 13     protected $useAutoIncrement = true;
31 14
32 15     protected $returnType   = 'array';
33 16     protected $useSoftDeletes = true;
34 17
35 18     protected $allowedFields = ['name', 'email'];
36 19
37 20     // Dates
38 21     protected $useTimestamps = false;
39 22     protected $dateFormat    = 'datetime';
40 23     protected $createdField  = 'created_at';
41 24     protected $updatedField  = 'updated_at';
42 25     protected $deletedField  = 'deleted_at';
43 26
44 27     // Validation
45 28     protected $validationRules   = [];
46 29     protected $validationMessages = [];
47 30     protected $skipValidation   = false;
48 31     protected $cleanValidationRules = true;
49 32
50 33 }
```

52 Kode 2.49 merupakan contoh *file Models* yang dapat digunakan pada *controllers*. *Models* tersebut
 53 terkoneksi dengan tabel **users** dengan *primarykey id*. *Model* pada *CodeIgniter 4* juga dapat
 54 digunakan untuk mencari, menyimpan, dan menghapus data untuk setiap tabel spesifik. Kode 2.50
 55 menunjukkan contoh penggunaan *model* dengan menggabungkan *query* dengan metode pencarian
 56 *model* untuk mencari seluruh data **active**.

Kode 2.50: Contoh penggunaan *model* untuk mencari data spesifik

```
1 <?php
2
3 $users = $userModel->where('active', 1)->findAll();
```

6 *Views*

- 7 *Views* merupakan halaman berisikan HTML dan sedikit PHP yang ditampilkan kepada pengguna.
 - 8 *Views* juga dapat berupa pecahan halaman seperti *header*, *footer*, ataupun *sidebar*. *Views* biasanya
 - 9 terdapat pada `app/Views` dan mendapatkan data berupa variabel dari *controller* untuk ditampilkan.

Kode 2.51: Contoh *Views*

```
10 <html>
11 1   <head>
12 2     <title>My Blog</title>
13 3   </head>
14 4   <body>
15 5     <h1>Welcome to my Blog!</h1>
16 6   </body>
17 7 </html>
18 8
```

- Kode 2.51 menunjukan contoh *view* pada direktori `app/Views` yang akan menampilkan tulisan `Welcome to my Blog`. *View* ini dapat ditampilkan melalui controller seperti yang ditunjukan pada Kode 2.52.

Kode 2.52: Contoh menampilkan *Views* pada *controller*

```
23
24 1 <?php
25 2
26 3 namespace App\Controllers;
27 4
28 5 use CodeIgniter\Controller;
29 6
30 7 class Blog extends Controller
31 8 {
32 9     public function index()
33 10    {
34 11        return view('blog_view')
35 12    }
36 13 }
```

- 38 Kode 2.52 merupakan contoh memanggil *views* pada *file controllers*. Kode ini mengembalikan
39 halaman `blog_view` pada *controller* dengan nama `Blog`.

40 *Controllers*

- 41* *Contollers* merupakan kelas untuk mengambil atau memberikan data dari *models* menuju *views* untuk
42 ditampilkan kepada pengguna. Setiap pembentukan *controllers* dibentuk harus memperpanjang
43 kelas *BaseController*. Kode 2.53 menunjukkan contoh *controllers* yang dibentuk dengan nama
44 *Helloworld*.

Kode 2.53: Contoh *Controllers* pada *CodeIgniter 4*

```
45
46 1 <?php
47 2
48 3 namespace App\Controllers;
49 4
50 5 class HelloWorld extends BaseController
51 6 {
52 7     public function index()
53 8     {
54 9         return 'Hello_World!';
55 0     }
56 1 }
```

```

11
12     public function comment()
13     {
14         return 'I_am_not_flat!';
15     }
16 }
```

8 Kode 2.53 akan mengembalikan kata *Hello World* pada url:

```
9           example.com/index.php/helloworld/
```

10 Selain itu, *CodeIgniter 4* menyediakan fungsi bernama *Controller Filters* dan kelas *IncomingRequest*. *Controller Filter* memiliki fungsi untuk membiarkan pengguna membangun sebuah kondisi sebelum ataupun sesudah *controller* dijalankan. Kode 2.54 menunjukkan contoh pembentukan sebuah *filters* dengan nama *MyFilter*.

Kode 2.54: Contoh *Controllers Filters* pada *CodeIgniter 4*

```

14 15 <?php
16 17 namespace App\Filters;
18 19
19 20 use CodeIgniter\Filters\FilterInterface;
20 21 use CodeIgniter\HTTP\RequestInterface;
21 22 use CodeIgniter\HTTP\ResponseInterface;
22 23
23 24 class MyFilter implements FilterInterface
24 25 {
25 26     public function before(RequestInterface $request, $arguments = null)
26 27     {
27 28         $auth = service('auth');
28 29
29 30         if (! $auth->isLoggedIn()) {
30 31             return redirect()->to(site_url('login'));
31 32         }
32 33     }
33 34 }
```

35 Kode 2.54 berfungsi contoh kode yang akan melakukan pengecekan apakah pengguna sudah
36 melakukan *login* sebelum *controller* dijalankan. Kode ini akan disimpan pada direktori *app/Filters*.
37 Selanjutnya pengguna perlu menambahkan konfigurasi *filter* pada *routes*. Sintaks dibawah akan
38 melakukan pengecekan pada *controller Dashboard::index* sebelum dan setelah *controller* tersebut
39 dijalankan menggunakan *filter* dengan nama *admin-auth*.

```
40 $routes->get('/', 'Dashboard::index', ['filter' => 'admin-auth:dual,noreturn']);
```

41 Selanjutnya kelas *IncomingRequest* menyediakan representasi *object-oriented* sebuah *HTTP*
42 *request* dari *client* seperti *browser*. Kode 2.55 menunjukkan contoh *IncomingRequest* untuk mengakses
43 data yang telah dikirimkan oleh *client*.

Kode 2.55: Contoh mengakses data menggunakan *IncomingRequest*

```

44 45 <?php
46 47 namespace App\Controllers;
48 49
49 50 use CodeIgniter\Controller;
50 51
51 52 class UserController extends Controller
52 53 {
53 54     public function index()
54 55     {
55 56         if ($this->request->isAJAX()) {
56 57             $something = $this->request->getVar('foo');
57 58         }
58 59     }
59 60 }
```

- 1 Sintaks `isAJAX` pada Kode 2.55 berfungsi untuk melakukan pengecekan apakan data yang dikirimkan
 2 melalui *AJAX*. Sedangkan sintaks `getVar('foo')` berfungsi untuk mengambil data yang telah
 3 dikirimkan dengan nama *foo*. Selain itu apabila pengguna ingin mengakses kelas ini diluar *controller*,
 4 pengguna dapat menginisiasikan kelas ini menggunakan sintaks berikut:

```
5           $request = \Config\Services::request();
```

- 6 Sintaks diatas akan menginisiasikan kelas *request* dan menyimpannya menuju sebuah variabel yang
 7 dapat diakses oleh seluruh fungsi pada kelas tersebut.

8 2.3.2 Autoloading Files

- 9 *CodeIgniter 4* menyediakan fitur *autoloader* yang dapat digunakan dengan sedikit konfigurasi.
 10 Fitur ini dapat menemukan *namespaced classes* individual yang menggunakan struktur direktori
 autoloading pada PSR-4. Fitur ini juga dapat digunakan bersamaan dengan *autoloader* lain
 seperti *composer*. Konfigurasi fitur ini dapat dilakukan pada direktori `app/Config/Autoload.php`
 yang berisikan dua buah *array* utama yakni `classmap` dan `psr4`. Kode 2.56 menunjukkan contoh
 konfigurasi menggunakan *namespace PSR-4* untuk melakukan *mapping* menuju direktori.

Kode 2.56: Contoh konfigurasi menggunakan *namespace PSR-4*.

```
15 1 <?php
16 2
17 3 namespace Config;
18 4
19 5 use CodeIgniter\Config\AutoloadConfig;
20 6
21 7 class Autoload extends AutoloadConfig
22 8 {
23 9     // ...
24 10    public $psr4 = [
25 11        APP_NAMESPACE => APPPATH, // For custom app namespace
26 12        'Config'      => APPPATH . 'Config',
27 13    ];
28 14
29 15    // ...
30 16 }
```

- 33 *Key* dari setiap baris merupakan *namespace* itu sendiri sedangkan *value* dari *array* merupakan
 34 *path* dari direktori. Pengguna dapat melakukan pengecekan terhadap konfigurasi *namespace*
 35 menggunakan sintaks sebagai berikut:

```
36           php spark namespaces
```

- 37 Sintaks diatas dapat dijalankan melalui *command line* pada aplikasi. Konfigurasi selanjutnya
 38 menggunakan *classmap* untuk menghubungkan terhadap *third-party libraries* yang tidak memiliki
 39 *namespace*. Kode 2.57 menunjukkan contoh konfigurasi menggunakan *classmap*. *Key* pada *array*
 40 *classmap* merupakan nama kelas yang ingin ditemukan sedangkan *value* merupakan *path* dari kelas
 41 itu sendiri.

Kode 2.57: Contoh konfigurasi menggunakan *classmap*.

```
42 1 <?php
43 2
44 3 namespace Config;
45 4
46 5 use CodeIgniter\Config\AutoloadConfig;
47 6
48 7 class Autoload extends AutoloadConfig
49 8 {
```

```
19 // ...
20 public $classmap = [
21     'Markdown' => APPPATH . 'ThirdParty/markdown.php',
22 ];
23
24 // ...
25 }
```

9 2.3.3 Configuration

- ¹⁰ Konfigurasi pada *CodeIgniter 4* terletak pada direktori `app/Config`. Konfigurasi pada *CodeIgniter*
¹¹ *4* tidak ditempatkan pada satu buah *file* melainkan setiap kelas yang membutuhkan konfigurasi
¹² memiliki *file* yang berbeda. Pengguna dapat mengakses *file configuration* dengan beberapa cara
¹³ sebagai berikut:

Kode 2.58: Contoh mengakses file *configuration*.

```
14 <?php
15 // Creating new configuration object by hand
16 // Creating new configuration object by hand
17 $config = new \Config\Pager();
18
19 // Get shared instance with config function
20 $config = config('Pager');
21
22 // Access config class with namespace
23 $config = config('Config\\Pager');
24 $config = config(\Config\Pager::class);
25
26 // Creating a new object with config function
27 $config = config('Pager', false);
```

- Kode 2.58 menunjukkan beberapa contoh untuk mengakses file *configuration*. Pengguna dapat mengakses *configuration* secara manual menggunakan sintaks `new`, pengguna dapat mengakses *configuration* menggunakan fungsi `config` dengan sintaks `config('namafile')`, menggunakan *namspace*, dan membentuk objek baru menggunakan fungsi `config`. Selanjutnya pengguna dapat mengakses properti yang terdapat pada file `config` tersebut menggunakan sintaks berikut:

```
34         $pageSize = $config->perPage;
```

- 35 Sintaks diatas akan mengambil properti `perPage` yang terdapat pada variabel `config` yang sudah
36 diinisiasi. Selain menggunakan `config` yang terdapat pada `CodeIgniter 4`, pengguna dapat membentuk
37 file `config` secara manual. Kode 2.59 menunjukkan contoh isi file `config` yang dibentuk manual.

Kode 2.59: Contoh pembentukan file *configuration*.

```
38 <?php
39
40
41 namespace Config;
42
43 use CodeIgniter\Config\BaseConfig;
44
45 class CustomClass extends BaseConfig
46 {
47     public $siteName = 'My_Great_Site';
48     public $siteEmail = 'webmaster@example.com';
49     // ...
50 }
51
```

- 52 *File config* ini akan disimpan pada direktori `app/Config` dengan nama kelas yang akan `extends`
53 dengan `BaseConfig`. Selain menggunakan file *config*, pengguna juga dapat melakukan konfigurasi
54 menggunakan variabel *environment*. Penggunaan file *environment* merupakan cara melakukan
55 konfigurasi terbaik untuk saat ini dikarenakan kemudahan untuk mengubah konfigurasi pada saat

- 1 melakukan *deploy* aplikasi. Konfigurasi dapat diubah tanpa harus mengubah kode. *CodeIgniter 4*
 2 menyediakan sebuah file bernama *.env* atau yang akan disebut *.env* selanjutnya. *.env* merupakan
 3 sebuah file yang terletak pada akar aplikasi. File ini dapat berisikan seluruh konfigurasi yang
 4 diperlukan oleh aplikasi. Berikut merupakan contoh variabel yang disimpan pada file *.env*.

Kode 2.60: Contoh variabel yang disimpan pada file *.env*.

```
5 S3_BUCKET = dotenv
6 SECRET_KEY = super_secret_key
7 CI_ENVIRONMENT = development
```

- 10 Kode 2.60 menunjukkan contoh variabel konfigurasi yang disimpan pada file *.env*. Variabel dapat
 11 berisikan konfigurasi yang bersifat pribadi seperti *password* dan *API keys*. Namun, konfigurasi hanya
 12 bersifat sebagai pengganti data sehingga pengguna tidak dapat membentuk data baru melainkan
 13 hanya mengubah data yang sudah ada sebelumnya.

14 2.3.4 *CodeIgniter URLs*

- 15 *CodeIgniter 4* menggunakan pendekatan *segment-based* dibandingkan menggunakan *query-string*
 16 untuk menghasilkan URL sehingga ramah manusia dan mesin pencari. Berikut merupakan contoh
 17 *URL* yang dihasilkan oleh *CodeIgniter 4*:

```
18 https://www.example.com/ci-blog/blog/news/2022/10?page=2
```

- 19 *CodeIgniter 4* menghasilkan *URL* seperti diatas dengan membaginya menjadi beberapa bagian
 20 sebagai berikut:

- 21 • URL dasar merupakan URL dasar dari aplikasi web yang ditunjukan sebagai *baseURL*. URL
 22 dasar ini berbentuk <https://www.example.com/ci-blog/>.
 - 23 • *URI Path* merupakan alamat yang dituju yaitu /ci-blog/blog/news/2022/10
 - 24 • *Route* juga merupakan alamat yang dituju tanpa URL dasar yaitu /blog/news/2022/10
 - 25 • *Query* merupakan hasil dari *query* yang ingin ditampilkan yaitu page=2
- 26 Secara *default*, *CodeIgniter 4* membangun *URL* dengan *index.php* namun, pengguna dapat
 27 menghapus *file index.php* pada URL yang dibentuk. Pengguna dapat menghapus *index.php* sesuai
 28 dengan *server* yang digunakan. Berikut merupakan contoh dua buah *server* yang umum dipakai:

29 *Apache Web Server*

- 30 Pengguna dapat URL melalui *file .htaccess* dengan menyalakan ekstensi *mod_rewrite*. Kode 2.61
 31 menunjukan contoh *file .htaccess* untuk menghapus *index.php* pada *URL* yang dibentuk.

Kode 2.61: Contoh *file .htaccess* pada *Apache Web Server*.

```
32 RewriteEngine On
33 RewriteCond %{REQUEST_FILENAME} !-f
34 RewriteCond %{REQUEST_FILENAME} !-d
35 RewriteRule ^(.*)$ index.php/$1 [L]
```

- 38 Kode 2.61 memperlakukan semua *HTTP Request* selain dari direktori dan *file* yang ada sebagai
 39 permintaan *file index.php*.

1 NGINX

2 Pengguna dapat mengubah URL menggunakan `try_files` yang akan mencari URI dan mengirimkan permintaan pada URL yang ingin dihilangkan. Kode 2.62 menunjukkan contoh penggunaan
3 `try_files` untuk menghapus `index.php` pada URL.

Kode 2.62: Contoh penggunaan `try-files`.

```
5 location / {
6   try_files $uri $uri/ /index.php$is_args$args;
7 }
8 }
```

10 Kode 2.62 akan mencari file atau direktori sesuai dengan URI yang diterima. Selanjutnya *request*
11 akan dikirimkan menuju file `index.php` bersamaan dengan argumen yang dikirimkan.

12 2.3.5 Error Handling

13 *CodeIgniter 4* melaporkan *error* melalui *exceptions*. Laporan ini akan dikirimkan sesuai dengan
14 konfigurasi *environment* pengguna kecuali *environment production*. Kode 2.63 menunjukkan cara
15 mengirimkan pesan *error* kepada pengguna.

Kode 2.63: Contoh penggunaan `error handling`.

```
16 throw new \Exception('Some message goes here');
```

19 Kode 2.63 akan mengirimkan pesan berupa `Some message goes here` kepada pengguna pada
20 *environment development* dan *testing*. Selain itu pengguna dapat mengambil *error* apabila terdapat
21 sebuah method yang memungkinkan untuk dikirimkan sebuah *exception*. Kode 2.64 menunjukkan
22 cara untuk mengambil *exception* pada sebuah method.

Kode 2.64: Contoh penggunaan `error handling` untuk mengambil *exception*.

```
23 try {
24   $user = $userModel->find($id);
25 } catch (\Exception $e) {
26   exit($e->getMessage());
27 }
28 }
```

30 Kode 2.64 akan mengirimkan pesan sesuai dengan yang telah ditentukan oleh pengguna apabila
31 method yang dijalankan terdapat sebuah *error*. Selain menentukan pesan *error*, pengguna juga
32 dapat memberikan *error* berupa 404 seperti yang ditunjukkan pada Kode 2.65.

Kode 2.65: Contoh penggunaan `error handling` untuk memberikan *error* berupa 404.

```
33 if (! $page = $pageModel->find($id)) {
34   throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
35 }
36 }
```

38 Kode 2.65 merupakan contoh sintaks untuk memberikan sebuah *error* berupa 404 atau halaman
39 tidak ditemukan. Kode 2.65 akan melakukan pengecekan terhadap *model* dan akan memberikan
40 *error* apabila kondisi tidak ditemukan. Pengguna juga dapat memberikan pesan pada *exception*
41 tersebut yang dapat ditampilkan pada *environment development* dan *testing*.

42 2.3.6 URI Routing

43 *CodeIgniter 4* menyediakan fitur URI *routing* yang berfungsi untuk menentukan *route* dari setiap
44 URL dan *controller* yang diinginkan. *CodeIgniter 4* menyediakan dua buah cara *routing* yang dapat
45 digunakan sebagai berikut:

1 **Defined Route Routing**

- 2 Pengguna dapat mendefinisikan *route* secara manual untuk *URL* yang lebih fleksibel. Kode 2.66
 3 menunjukan contoh *route* yang didefinisikan secara manual untuk membentuk URL menuju kelas
 4 **Catalog** dengan metode `productLookup`.

Kode 2.66: Contoh *route* yang didefinisikan secara manual

```
5 6 1 <?php
7 2
8 3 $routes->get('product/(:num)', 'Catalog::productLookup');
```

- 10 Pengguna juga dapat memakai beberapa HTTP *verb* seperti *GET*, *POST*, dan *PUT*. Selain menulis
 11 *route* secara individu, pengguna dapat melakukan *grouping* pada route seperti yang ditunjukan
 12 Kode 2.67.

Kode 2.67: Contoh *route* yang menggunakan *grouping* manual

```
13 14 1 <?php
15 2
16 3 $routes->group('admin', static function ($routes) {
17 4   $routes->get('users', 'Admin\Users::index');
18 5   $routes->get('blog', 'Admin\Blog::index');
19 6});
```

- 21 Kode 2.67 menunjukan contoh penggunaan *grouping* untuk URI `admin/users` dan `admin/blog`.
 22 Fitur ini memudahkan pengguna untuk menentukan *route* pada setiap fungsi yang ingin dibentuk
 23 pada *controller* `admin`.

24 **Auto Routing**

- 25 Pengguna dapat mendefinisikan *route* secara otomatis melalui fitur *Auto Routing* apabila tidak
 26 terdapat *route* yang didefinisikan. *Auto Routing* membentuk *URI* berdasarkan nama dan fungsi
 27 pada *controller* yang dituju. Pengguna dapat menyalakan fitur ini pada `app\Config\Routes.php`
 28 dengan cara sebagai berikut:

```
29           public bool $autoRoute = true;
```

- 30 Pengguna juga perlu mengubah `$autoRoutesImproved` menjadi `true` pada file `app\Config\Feature.php`.
 31 Selain menggunakan *auto routing* baru, pengguna dapat menggunakan *Auto Routing (Legacy)* seperti
 32 yang terdapat pada *CodeIgniter 3* dengan cara seperti berikut:

```
33           $routes->setAutoRoute(true);
```

- 34 Penggunaan *Auto Routing (Legacy)* tidak disarankan karena fitur ini melewatkkan pengecekan
 35 *controller filters* dan proteksi CSRF.

36 **2.3.7 Database**

- 37 *CodeIgniter 4* menyediakan kelas *database* yang dapat menyimpan, memasukan, memperbarui, dan
 38 menghapus data pada *database* sesuai dengan konfigurasi. Pengguna dapat melakukan konfigurasi
 39 untuk *database* yang ingin dikoneksikan melalui direktori `app\Config\Database.php` atau *file*
 40 `.env`. Kode 2.68 menunjukan contoh konfigurasi untuk database bernama `database_name` dengan
 41 *username* root.

Kode 2.68: Contoh konfigurasi *database* pada *CodeIgniter 4*.

```

1 21 <?php
2 32
3 43 namespace Config;
4 54
5 65 use CodeIgniter\Database\Config;
6 76
7 87 class Database extends Config
8 98 {
9 109     public $default = [
10 110         'DSN' => '',
11 121         'hostname' => 'localhost',
12 132         'username' => 'root',
13 143         'password' => '',
14 154         'database' => 'database_name',
15 165         'DBDriver' => 'MySQLi',
16 176         'DBPrefix' => '',
17 187         'pConnect' => true,
18 198         'DBDebug' => true,
19 209         'charset' => 'utf8',
20 220         'DBCollat' => 'utf8_general_ci',
21 231         'swapPre' => '',
22 242         'encrypt' => false,
23 253         'compress' => false,
24 264         'strictOn' => false,
25 275         'failover' => [],
26 286         'port' => 3306,
27 297     ];
28 308
29 319 // ...
30 330 }
31 340
32 350 }
```

33 Selain itu, konfigurasi juga dapat dilakukan pada file `.env` untuk mempermudah dalam pe-
 34 ngubahan data pada saat melakukan *deploy*. Kode 2.69 menunjukkan contoh konfigurasi pada file
 35 `.env`:

Kode 2.69: Contoh konfigurasi *database* pada file `.env`.

```

36 371 database.default.username = 'root';
37 382 database.default.password = '';
38 393 database.default.database = 'ci4';
```

41 Kode 2.69 akan menyimpan konfigurasi pada grup *default* dengan *username* berupa `root`, tanpa
 42 menggunakan *password*, dan juga dengan nama *database* `ci4`. Selain untuk melakukan koneksi
 43 *database*, kelas ini dapat digunakan untuk menambahkan, menghapus, dan memperbarui data
 44 pada *database*. Kode 2.70 menunjukkan contoh penggunaan *query* untuk mengambil data *title*, *content*,
 45 dan *date* pada tabel *users*.

Kode 2.70: Contoh penggunaan *query* menggunakan konfigurasi pada *CodeIgniter 4*.

```

46 471 <?php
47 482
48 493 $builder = $db->table('users');
49 504 $builder->select('title,_content,_date');
50 515 $query = $builder->get();
```

53 *CodeIgniter 4* juga menyediakan fitur untuk membangun *database* melalui fitur bernama *Database*
 54 *Forge*. Pengguna dapat membangun, mengubah, menghapus tabel dan juga menambahkan *field*
 55 pada tabel tersebut. Kode 2.71 menunjukkan contoh pembentukan *database*.

Kode 2.71: Contoh pembentukan tabel melalui *database forge*.

```

56 571 <?php
57 582
58 593 $fields = [
59 604     'id' => [
60 615         'type' => 'INT',
61 626         'constraint' => 5,
62 637         'unsigned' => true,
```

```

18     'auto_increment' => true,
29 ],
30 'title' => [
41     'type'      => 'VARCHAR',
42     'constraint' => '100',
43     'unique'    => true,
44 ],
45 'author' => [
46     'type'      => 'VARCHAR',
47     'constraint' => 100,
48     'default'   => 'King_of_Town',
49 ],
50 'description' => [
51     'type'    => 'TEXT',
52     'null'   => true,
53 ],
54 'status' => [
55     'type'      => 'ENUM',
56     'constraint' => ['publish', 'pending', 'draft'],
57     'default'   => 'pending',
58 ],
59 ];
60 $forge->addField($fields);
61 $forge->createTable('table_name');

```

26 Kode 2.71 akan membentuk *database* dengan tabel bernama *table_name* yang berisikan beberapa *field* seperti *id* dan *title*.

28 2.3.8 Library

29 *CodeIgniter 4* menyediakan berbagai *library* untuk membantu pengguna dalam pembentukan aplikasi
30 *website*. Berikut merupakan beberapa *library* yang disediakan oleh *CodeIgniter 4*:

31 Kelas Email

32 *CodeIgniter* menyediakan kelas *email* dengan fitur sebagai berikut:

- 33 • Beberapa Protokol: *Mail*, *Sendmail*, dan *SMTP*
- 34 • Enkripsi TLS dan SSL untuk *SMTP*
- 35 • Beberapa Penerima
- 36 • CC dan BCC
- 37 • HTML atau *email* teks biasa
- 38 • Lampiran
- 39 • Pembungkus kata
- 40 • Prioritas
- 41 • Mode BCC *Batch*, memisahkan daftar *email* besar menjadi beberapa BCC kecil.
- 42 • Alat *Debugging email*

43 Pengguna dapat melakukan konfigurasi pada file *app/Config/Email.php* untuk melakukan
44 pengiriman *email*. Kode 2.72 menunjukkan contoh konfigurasi preferensi pengiriman *email* secara
45 manual.

Kode 2.72: Contoh kode untuk melakukan konfigurasi *email*.

```

46 47 <?php
48 49 $config['protocol'] = 'sendmail';
50 51 $config['mailPath'] = '/usr/sbin/sendmail';
52 53 $config['charset'] = 'iso-8859-1';
54 55 $config['wordWrap'] = true;
56 57
57 58 $email->initialize($config);

```

1 Kode 2.72 menentukan konfigurasi protokol agar menggunakan `sendmail`, *server path* pada
 2 *Sendmail*, penggunaan karakter *email*, dan menyalakan fitur *wordwrap*. Selain itu, pengguna dapat
 3 melakukan pengiriman *email* sesuai dengan kebutuhan. Kode 2.73 menunjukkan contoh penggunaan
 4 kelas *email* untuk mengirim *email*.

Kode 2.73: Contoh kode untuk melakukan pengiriman *email*.

```
5 <?php
6
7
8 $email = \Config\Services::email();
9
10 $email->setFrom('your@example.com', 'Your_Name');
11 $email->setTo('someone@example.com');
12 $email->setCC('another@another-example.com');
13 $email->setBCC('them@their-example.com');
14
15 $email->setSubject('Email_Test');
16 $email->setMessage('Testing_the_email_class.');
17
18 $email->send();
```

20 Kode 2.73 mengirimkan *email* dari `your@example.com` kepada `someone@example.com` dengan subjek
 21 `Email Test` dan pesan `Testing the email class`. *Email* juga akan dikirimkan melalui CC kepada
 22 '`another@another-example.com`' dan mengirimkan BCC menuju '`them@their-example.com`'.

23 Session

24 *Library session* memperbolehkan aplikasi untuk melihat jejak dari aktivitas dari pengguna aplikasi
 25 tersebut. Data *session* dapat disimpan pada beberapa tempat yakni *file*, *database*, *memcached*,
 26 *redis*, dan *array*. *Session* dapat dilakukan inisiasi dengan sintaks sebagai berikut:

```
27 $session = \Config\Services::session($config);
```

28 Sintaks diatas akan melakukan inisiasi *library session* dan pengguna dapat memanggil *library*
 29 melalui variabel *session*. *Session* bekerja saat halaman aplikasi dari pengguna termuat. Aplikasi
 30 akan melakukan pengecekan terhadap *session cookie* dan melakukan pembaharuan sesuai dengan
 31 *session cookie* yang didapat. Pengguna juga dapat mengambil data yang tersimpan pada *session*
 32 menggunakan sintaks berikut:

```
33 $item = $session->get('item');
```

34 Sintaks diatas akan mengambil data *session* dengan nama *item* dan menyimpannya menuju sebuah
 35 variabel. Selain itu, *session* juga dapat dihapus menggunakan sintaks sebagai berikut:

```
36 $session->destroy();
```

37 Sintaks diatas akan membersihkan *session* dari pengguna seperti saat melakukan *logout*. Secara
 38 *default session* akan menyimpan data menuju file namun, pengguna dapat melakukan konfigurasi
 39 sesuai dengan keperluan. Kode 2.74 menunjukkan konfigurasi untuk menyimpan data *session* menuju
 40 *database*.

Kode 2.74: Contoh kode untuk konfigurasi penyimpanan *session*.

```
41
42 class Session extends BaseConfig
43 {
44     // ...
45     public string $driver = 'CodeIgniter\Session\Handlers\DatabaseHandler';
46 }
```

```

16     // ...
27     public string $savePath = 'ci_sessions';
38
49     // ...
50 }

```

- 7 Kode 2.74 akan mengubah penyimpanan menggunakan `databasehandler` dengan tabel penyimpanan bernama `ci_sessions`. Konfigurasi dapat dilakukan pada direktori `app\Config` dengan nama `file Session.php`.

10 *Working with Uploaded Files*

- 11 *CodeIgniter 4* menyediakan *library* untuk melakukan pengunggahan *file*. Pengunggahan *file* terdapat 12 empat buah proses sebagai berikut:

- 13 1. Dibentuk sebuah form untuk pengguna memilih dan mengunggah *file*.
 - 14 2. Setelah *file* diunggah, *file* akan dipindahkan menuju direktori yang ditentukan.
 - 15 3. Pada pengiriman dan pemindahan *file* dilakukan validasi sesuai dengan ketentuan yang ada.
 - 16 4. Setelah *file* diterima akan dikeluarkan pesan berhasil.
- 17 Perangkat lunak akan menerima *file* dari *form* yang nantinya akan dilakukan validasi pada *controller*.
- 18 Kode 2.75 menunjukkan contoh *form* untuk melakukan pengunggahan *file*.

Kode 2.75: Contoh *form* untuk melakukan pengunggahan *file*.

```

19 <!DOCTYPE html>
20 <html lang="en">
21 <head>
22   <title>Upload Form</title>
23 </head>
24 <body>
25
26
27 <?php foreach ($errors as $error): ?>
28   <li><?= esc($error) ?></li>
29 <?php endforeach ?>
30
31 <?= form_open_multipart('upload/upload') ?>
32   <input type="file" name="userfile" size="20">
33   <br><br>
34   <input type="submit" value="upload">
35 </form>
36
37 </body>
38 </html>
39

```

- 40 Kode 2.75 dibentuk menggunakan *form helper* dan dapat mengembalikan pesan apabila terdapat 41 *error*. Setelah dilakukan penerimaan *file*, perangkat lunak akan mengirimkan *file* menuju *controller* 42 untuk dilakukan validasi dan penyimpanan. Kode 2.76 menunjukkan contoh *controller* untuk 43 melakukan validasi dan penyimpanan.

Kode 2.76: Contoh kode *controller* untuk melakukan validasi dan penyimpanan.

```

44 <?php
45
46
47 namespace App\Controllers;
48
49 use CodeIgniter\Files\File;
50
51 class Upload extends BaseController
52 {
53   protected $helpers = ['form'];
54
55   public function index()
56   {
57     return view('upload_form', ['errors' => []]);
58   }
59 }

```

```

16     public function upload()
17     {
18         $validationRule = [
19             'userfile' => [
20                 'label' => 'Image_File',
21                 'rules' => [
22                     'uploaded[userfile]',
23                     'is_image[userfile]',
24                     'mime_in[userfile,image/jpg,image/jpeg,image/gif,image/png,image/webp]',
25                     'max_size[userfile,100]',
26                     'max_dims[userfile,1024,768]',
27                 ],
28             ],
29         ];
30         if (! $this->validate($validationRule)) {
31             $data = ['errors' => $this->validator->getErrors()];
32
33             return view('upload_form', $data);
34         }
35
36         $img = $this->request->getFile('userfile');
37
38         if (! $img->hasMoved()) {
39             $filepath = WRITEPATH . 'uploads/' . $img->store();
40
41             $data = ['uploaded_fileinfo' => new File($filepath)];
42
43             return view('upload_success', $data);
44         }
45
46         $data = ['errors' => 'The_file_has_already_been_moved.'];
47
48         return view('upload_form', $data);
49     }
50 }
```

37 Kode 2.76 terdapat dua buah fungsi sebagai berikut:

- 38 • `index()` berfungsi untuk mengembalikan *view* bernama `upload_form`
- 39 • `upload()` berfungsi dalam memberikan aturan ukuran, dimensi, dan jenis *file* untuk melakukan
- 40 validasi. Setelah validasi berhasil fungsi ini melakukan penyimpanan pada direktori `uploads`
- 41 dan mengembalikan halaman *view*.

42 ***Working with URIs***

43 *CodeIgniter 4* menyediakan solusi berbasis *object oriented* untuk bekerja dengan URI. *Library* ini

44 simpel dan memastikan struktur selalu benar tanpa melihat URI yang kompleks. Berikut merupakan

45 sintaks untuk membentuk sebuah *URI*:

```
46     $uri = new \CodeIgniter\HTTP\URI('http://www.example.com/some/path');
```

47 Sintaks diatas akan membentuk URI dengan nama `example.com` dengan beberapa *segment* bernama

48 `some` dan `path`. URI yang sudah dibentuk selanjutnya dapat diambil menggunakan sintaks berikut:

```
49             $uri = $this->request->getUri();
```

50 Sintaks diatas akan mengambil URI yang terdapat pada halaman tersebut. Selain dapat mengambil

51 URI, *library* ini juga dapat mengambil *segment* yang terdapat pada *URI*. Berikut merupakan sintaks

52 untuk mengambil *segement* yang terdapat pada *URI*.

```
53             $segments = $uri->getSegments();
```

54 Sintaks diatas akan mengambil *segment* yang tersedia pada *URI* dan menyimpannya dalam sebuah

55 *array* berisikan *value* dari setiap *segment* yang tersedia.

1 Validation

2 *CodeIgniter 4* menyediakan *library* untuk melakukan validasi terhadap data yang dikirimkan oleh
 3 pengguna. Data yang divalidasi dapat diberikan aturan-aturan sesuai dengan konfigurasi pengguna.
 4 Validasi akan melalui data yang dikirimkan oleh *form*. Kode 2.77 menunjukkan contoh *form* yang
 5 digunakan untuk mengirimkan data menuju *controller*.

Kode 2.77: Contoh kode untuk melakukan pengumpulan data.

```

6 <html>
7  <head>
8    <title>My Form</title>
9  </head>
10 <body>
11
12
13 <?= validation_list_errors() ?>
14
15 <?= form_open('form') ?>
16
17   <h5>Username</h5>
18   <input type="text" name="username" value="<?= set_value('username') ?>" size="50">
19
20   <h5>Password</h5>
21   <input type="text" name="password" value="<?= set_value('password') ?>" size="50">
22
23   <h5>Password Confirm</h5>
24   <input type="text" name="passconf" value="<?= set_value('passconf') ?>" size="50">
25
26   <h5>Email Address</h5>
27   <input type="text" name="email" value="<?= set_value('email') ?>" size="50">
28
29   <div><input type="submit" value="Submit"></div>
30
31 <?= form_close() ?>
32
33 </body>
34 </html>
```

36 *Form* yang dibentuk dapat mengembalikan *error* menggunakan sintaks `validation_list_errors()`.
 37 Selanjutnya akan digunakan fungsi `form_open` untuk membuka *tag form* sesuai dengan URL fungsi
 38 *controller* yang sudah dibentuk. Setiap input akan diberikan *name* untuk mengambil data pada
 39 *controller*. Setelah *tag form* selesai maka akan ditutup dengan sintaks `form_close`. Setelah itu
 40 data-data yang sudah dimasukan akan dikirimkan menuju *controller* seperti yang ditunjukan pada
 41 kode 2.78.

Kode 2.78: Contoh kode untuk melakukan validasi data yang sudah dikumpulkan.

```

42 <?php
43
44
45 namespace App\Controllers;
46
47 class Form extends BaseController
48 {
49   protected $helpers = ['form'];
50
51   public function index()
52   {
53     if (! $this->request->is('post')) {
54       return view('signup');
55     }
56
57     $rules = [];
58
59     if (! $this->validate($rules)) {
60       return view('signup');
61     }
62
63     // If you want to get the validated data.
64     $validData = $this->validator->getValidated();
65
66     return view('success');
67   }
68 }
```

26 }

- 3 Data-data yang sudah diberikan oleh pengguna akan diambil dan divalidasi menggunakan *controller*
 4 pada Kode 2.78. Sintaks `if (! $this->request->is('post'))` akan melakukan pengecekan
 5 apakah *request* yang diberikan berupa *post* atau tidak. Selanjutnya dapat ditentukan aturan pada
 6 variabel **rules** yang nantinya dilakukan validasi menggunakan fungsi **validate**. Fungsi **validate**
 7 akan mengecek data yang diberikan dan menentukan apakah sudah sesuai dengan aturan yang ada
 8 atau belum. Kode 2.79 menunjukkan contoh pembentukan aturan sesuai dengan *name* setiap *input*
 9 *form*.

Kode 2.79: Contoh kode untuk menetapkan aturan untuk validasi data yang sudah dikumpulkan.

```
10 $rules = [
11   'username' => 'required|max_length[30]',
12   'password' => 'required|max_length[255]|min_length[10]',
13   'passconf' => 'required|max_length[255]|matches[password]',
14   'email'    => 'required|max_length[254]|valid_email',
15 ];
16 
```

- 18 Kode 2.79 melakukan pengecekan terhadap setiap *input form* sesuai dengan aturan yang telah
 19 ditetapkan. Aturan-aturan tersebut dapat diganti sesuai dengan kebutuhan dari pengguna. Selain
 20 menggunakan aturan yang disediakan *CodeIgniter 4*, pengguna dapat membentuk aturannya sendiri
 21 pada file *Validation.php*. Kode 2.80 menunjukkan contoh aturan yang dibentuk secara manual.
 22 Aturan yang dibentuk secara manual dapat digunakan sama seperti penggunaan aturan lainnya.

Kode 2.80: Contoh kode pembentukan aturan secara manual pada file *Validation.php*.

```
23 <?php
24 class MyRules
25 {
26   public function even($value): bool
27   {
28     return (int) $value % 2 === 0;
29   }
30 }
31 
```

34 2.3.9 *Helpers*

- 35 *Helpers* merupakan fitur pada *CodeIgniter 4* yang menyediakan beberapa fungsi untuk pengguna
 36 dalam membangun aplikasi *website*. *Helpers* dapat dimuat oleh pengguna menggunakan sintaks
 37 sebagai berikut:

```
38           <?php
39           helper('helpers_name');
```

- 40 Setelah dilakukan pemanggilan, pengguna dapat memakai fungsi-fungsi yang disediakan sesuai
 41 dengan *helpers* yang digunakan. Fungsi-fungsi itu antara lain adalah *form*, *array*, dan *text*.

42 2.4 Konversi CodeIgniter 3 ke CodeIgniter 4[3]

- 43 Konversi CodeIgniter 3 ke CodeIgniter 4 diperlukan penulisan ulang karena terdapat banyak
 44 implementasi yang berbeda. Konversi ke CodeIgniter 4 diawali dengan melakukan instalasi projek
 45 baru CodeIgniter 4. Instalasi dapat dilakukan dengan mengunduh file melalui situs resmi ataupun
 46 dapat dilakukan melalui *composer*.

1 2.4.1 Struktur Aplikasi

2 Struktur direktori pada CodeIgniter 4 memiliki perubahan yang terdiri *app*, *public*, dan *writable*.
3 Direktori app merupakan perubahan dari direktori application dengan isi yang hampir sama dengan
4 beberapa perubahan nama dan perpindahan direktori. Pada CodeIgniter 4 terdapat direktori *public*
5 yang bertujuan sebagai direktori akar pada aplikasi *website*. Selanjutnya terdapat direktori *writable*
6 yang berisikan *cache data*, *logs*, dan *session data*.

7 2.4.2 Routing

8 *CodeIgniter 4* meletakan *route* pada file `app\Config\Routes.php`. *CodeIgniter 4* memiliki fitur
9 *auto routing* seperti pada *CodeIgniter 3* namun, secara *default* dimatikan. Fitur *auto routing*
10 memungkinkan untuk dinyalakan serupa dengan pada *CodeIgniter 3* namun tidak direkomendasikan
11 karena alasan *security*.

12 2.4.3 Model, View, dan Controller

13 Struktur MVC pada *CodeIgniter 4* berbeda dengan *CodeIgniter 3* dimana terdapat perbedaan
14 penyimpanan direktori untuk ketiga file tersebut. Berikut merupakan penjelasan mengenai struktur
15 MVC:

16 Model

17 *Model* pada *CodeIgniter 4* terletak pada direktori `app\Models`. Pembaharuan *Model* dapat dilakukan
18 melalui cara sebagai berikut:

- 19 1. Pengguna harus memindahkan seluruh file *model* menuju direktori `app/Models`
- 20 2. Pengguna harus menambahkan `namespace App\Models`; setelah pembukaan tag PHP.
- 21 3. Pengguna juga harus menambahkan `use CodeIgniter\Model`; setelah kode diatas `namespace`.
- 22 4. Pengguna harus mengganti `extends CI_Model` menjadi `extends Model`.
- 23 5. Terakhir pemanggilan *model* berubah dari sintaks `$this->load->model('x');` menjadi
24 `$this->x = new X();`.

25 View

26 *View* pada *CodeIgniter 4* terdapat di `app\Views` dengan sintaks yang harus diubah. Pembaharuan
27 *view* dapat dilakukan dengan cara sebagai berikut:

- 28 1. Pengguna perlu memindahkan seluruh file *views* menuju `app/Views`
- 29 2. Pengguna perlu mengubah sintaks:

30 `$this->load->view('directory_name/file_name')`

31 menjadi sintaks berikut:

32 `return view('directory_name/file_name');`

- 33 3. Pengguna juga perlu mengubah sintaks:

34 `$content = $this->load->view('file', $data, TRUE);`

menjadi sintaks berikut:

```
1           $content = view('file', $data);.
```

- 3 4. Pada *file views* pengguna dapat mengubah sintaks <?php echo \$title; ?> menjadi sintaks
4 <?= \$title ?>.
- 5 5. Pengguna juga perlu menghapus apabila terdapat sintaks defined('BASEPATH') OR exit('No
6 direct script access allowed');

7 *Controller*

8 *Controller* pada CodeIgniter 4 terdapat di `app\Controllers` dan diperlukan beberapa perubahan.

9 Pengguna dapat melakukan pembaharuan *controller* menggunakan cara sebagai berikut:

- 10 1. Pengguna harus memindahkan seluruh *file controller* menuju `app\Controllers`.
- 11 2. Pengguna juga harus menambahkan sintaks `namespace App\Controllers;` setelah pembukaan *tag PHP*.
- 13 3. Selanjutnya pengguna harus mengubah `extends CI_Controller` menjadi `extends BaseController`.
- 14 4. Pengguna juga harus menghapus apabila terdapat baris `defined('BASEPATH') OR exit('No
15 direct script access allowed');`.

16 **2.4.4 Class Loading**

17 Pada *CodeIgniter 4* sudah tidak terdapat *superobject* dengan komponen *framework* yang terinjeksi
18 sebagai properti pada *controller*. Kelas yang sudah dibentuk akan diinisiasi di tempat yang
19 membutuhkan dan komponen *framework* akan diatur oleh *service*. *Autoloader* pada *CodeIgniter 4*
20 secara otomatis menangani lokasi kelas dengan standar *PSR-4* di dalam direktori `App`.

21 **2.4.5 Configuration**

22 *File configuration* CodeIgniter 4 terdapat pada `app\Config` dengan penulisan sedikit berbeda
23 dengan versi sebelumnya. Penulisan berubah dari yang sebelumnya menggunakan *array* akan
24 berubah menjadi menggunakan variabel. Pengguna hanya perlu melakukan pemindahan data
25 menuju CodeIgniter 4 dan apabila menggunakan *file config custom* maka, diperlukan penulisan
26 ulang pada direktori Config dengan melakukan *extend* pada `CodeIgniter\Config\BaseConfig`.
27 Beberapa konfigurasi juga akan dipindahkan menuju file `.env`.

28 **2.4.6 Database**

29 Penggunaan *database* pada CodeIgniter 4 hanya berubah sedikit dibandingkan dengan versi sebelumnya. Pengguna dapat melakukan pembaharuan menggunakan cara sebagai berikut:

- 31 1. Memindahkan data-data kredensial menuju `file app\Config\Database.php`
- 32 2. Mengubah sintaks untuk memuat *database* diubah menjadi `$db = db_connect();`
- 33 3. Penggunaan beberapa *database* diubah menggunakan sintaks `$db = db_connect('group_name');`
- 34 4. Mengubah seluruh *query* dari `$this->db` menjadi `$db`
- 35 5. Mengubah sintaks *query* seperti `$query->result();` menjadi `$query->getResult();`

- 1 6. Selain itu, terdapat pembaharuan *Query Builder Class* yang harus diinisiasi menggunakan
 2 sintaks `$builder = $db->table('mytable');`. Kelas ini dapat dipakai untuk menjalankan
 3 *query* menggunakan sintaks `$builder->get();`

4 2.4.7 *Migrations*

- 5 Perubahan perlu dilakukan pada nama *file* menjadi nama dengan cap waktu. Selanjutnya dilakukan penghapusan kode `defined('BASEPATH') OR exit('No direct script access allowed');`
 6 dan menambahkan dua buah kode setelah membuka tag PHP seperti yang ditunjukan pada Kode
 7 8 [2.81](#).

Kode 2.81: Penambahan kode pada *file migration*.

```
9   \item \verb|namespace App\Database\Migrations;|
10 1   \item \texttt{use CodeIgniter\Database\Migration;}
```

- 13 Setelah itu, `extends CI_Migration` diubah menjadi `extends Migration`. Terakhir, terdapat
 14 perubahan pada nama metode *Forge* dari yang sebelumnya bernama `$this->dbforge->add_field`
 15 menjadi menggunakan *camelCase* `$this->forge->addField`.

16 2.4.8 *Routing*

- 17 Pengguna dapat melakukan pembaharuan *routing* dengan cara sebagai berikut:
- 18 1. Pengguna dapat memakai *Auto Routing* seperti pada *CodeIgniter 3* dengan menyalakan *Auto Routing(Legacy)*.
 - 20 2. Terdapat perubahan dari `(:any)` menjadi `(:segment)`.
 - 21 3. Pengguna juga harus mengubah sintaks pada `app/Config/Routes.php` sebagai berikut:
 - 22 • `$route['journals'] = 'blogs';` menjadi `$routes->add('journals', 'Blogs::index');`
 Sintaks diatas berguna untuk memanggil fungsi *index* pada *controller Blogs*.
 - 24 • `$route['product/(:any)'] = 'catalog/product_lookup'` menjadi sintaks `$routes->add('product', 'Catalog::productLookup');`.

26 2.4.9 *Libraries*

- 27 *CodeIgniter 4* menyediakan *library* untuk digunakan dan dapat diinstall apabila diperlukan. Pemanggilan *library* berubah dari `$this->load->library('x');` menjadi `$this->x = new X();`.
 28 Terdapat beberapa *library* yang harus di perbarui dengan sedikit perubahan. Berikut merupakan
 30 beberapa *libraries* yang terdapat pembaharuan:

31 *Emails*

- 32 Perubahan *email* hanya terdapat pada nama dari *method* dan pemanggilan *library email*. Pemanggilan *library* berubah dari `$this->load->library('email');` menjadi `$email = service('email');`
 33 dan selanjutnya perlu dilakukan perubahan pada semua `$this->email` menjadi `$email`. Selanjutnya beberapa pemanggilan *method* berubah dengan tambahan *set* didepannya seperti *from* menjadi *setFrom*.

1 ***Working with Uploaded Files***

2 Terdapat banyak perubahan dimana pada *CodeIgniter 4* pengguna dapat mengecek apakah *file* telah
 3 terunggah tanpa *error*. *Library* ini juga mempermudah pengguna untuk melakukan penyimpanan
 4 *file*. Pada *CodeIgniter 4* melakukan akses pada *uploaded file* dengan nama *userinfo* dilakukan
 5 dengan sintaks sebagai berikut:

```
6     $file = $this->request->getFile('userinfo')
```

7 selanjutnya dapat dilakukan validasi dengan cara sebagai berikut:

```
8         $file->isValid()
```

9 *File* yang sudah berhasil diunggah dapat disimpan dengan sintaks berikut:

```
10    $path = $this->request->getFile('userinfo')->store('head_img/' , 'user_name.jpg');.
```

11 Sintaks diatas akan mengambil file dengan atribut nama *userinfo* dan menyimpannya pada
 12 direktori *head_img* dengan nama file *user_name.jpg*.

13 ***HTML Tables***

14 Tidak terdapat banyak perubahan pada *framework* versi terbaru hanya perubahan pada nama *method*
 15 dan pemanggilan *library*. Perubahan pemanggilan *library* dari *\$this->load->library('table')*;
 16 menjadi *\$table = new \CodeIgniter\View\Table();* dan perlu dilakukan perubahan setiap
 17 *\$this->table* menjadi *\$table*. Selain itu, terdapat beberapa perubahan pada penamaan *method*
 18 dari *underscored* menjadi *camelCase* seperti *set_heading* menjadi *setHeading*.

19 ***Localization***

20 *CodeIgniter 4* mengembalikan *file* bahasa menjadi *array* sehingga perlu dilakukan beberapa perubah-
 21 an. Pertama, perlu dilakukan konfigurasi *default language* pada perangkat lunak. Selanjutnya mela-
 22 kukan pemindahan *file* bahasa pada *CodeIgniter 3* menuju *app\Language\<locale>*. *File-file* baha-
 23 sa *CodeIgniter 3* perlu dilakukan penghapusan semua *loader* *\$this->lang->load(\$file, \$lang);*.
 24 Terakhir mengubah *method* pemanggilan bahasa dari *\$this->lang->line('error_email_missing')*
 25 menjadi *echo lang('Errors.errorEmailMissing');*

26 ***Validations***

27 Pengguna dapat melakukan pembaharuan pada *validations* melalui cara sebagai berikut:

- 28 1. Pengguna harus mengubah kode pada *view* dari *<?php echo validation_errors(); ?>*
 menjadi *<?= validation_list_errors() ?>*
- 30 2. Pengguna perlu mengubah beberapa kode pada *controller* sebagai berikut:

- 31 • *\$this->load->helper(array('form' , 'url'))*; menjadi *helper(['form' , 'url'])*;
- 32 • Pengguna perlu menghapus kode *\$this->load->library('form_validation')*;
- 33 • *if (\$this->form_validation->run() == FALSE)* menjadi *if (!\$this->validate([]))*
- 34 • *\$this->load->view('myform');*
- 35 menjadi sintaks *return view('myform' , ['validation' => \$this->validator,]);*

- 1 3. Pengguna juga perlu mengubah kode seperti yang ditunjukan pada Kode 2.82 untuk melakukan
2 validasi *form*.

Kode 2.82: Perubahan kode untuk melakukan validasi.

```
3 1 <?php
4 2
5 3 $isValid = $this->validate([
6 4   'name' => 'required|min_length[3]',
7 5   'email' => 'required|valid_email',
8 6   'phone' => 'required|numeric|max_length[10]',
9 7 ]);
10 7 ]);
```

12 2.4.10 *Helpers*

- 13 *Helpers* tidak terdapat banyak perubahan namun, beberapa *helpers* pada *CodeIgniter 3* tidak
14 terdapat pada *CodeIgniter 4* sehingga perlu perubahan pada implementasi fungsinya. *Helpers* dapat
15 di dimuat secara otomatis menggunakan `app\Config\Autoload.php`

16 2.4.11 *Framework*

- 17 Pengguna tidak membutuhkan direktori *core* dan tidak membutuhkan kelas `MY_X` pada direktori
18 *libraries* untuk memperpanjang atau mengganti potongan CI4. Pengguna dapat membangun kelas
19 dimanapun dan menambahkan metode pada `app\Config\Services.php`.

1

BAB 3

2

ANALISIS

3 3.1 Analisis Sistem Kini

4 Seperti dibahas pada bab 2.2, *SharIF Judge* merupakan sebuah *online judge* yang dikustomisasi
5 sesuai dengan kebutuhan Informatika UNPAR. *SharIF Judge* dibentuk menggunakan *framework*
6 *CodeIgniter 3* yang menerapkan arsitektur *Model-View-Controller* atau MVC. Arsitektur ini me-
7 misahkan pemrosesan data pada *Model*, memisahkan logika pada *Controller*, dan memisahkan
8 tampilan pada *View*. Selain itu, terdapat direktori **assets** yang berisi seluruh kebutuhan
9 pengguna untuk ditampilkan seperti *javascript* dan gambar. Terakhir terdapat direktori **config** yang
10 berisi konfigurasi aplikasi dan *library* yang dibentuk dan digunakan oleh *SharIF Judge*.

11 **3.1.1 Model**

12 *Model* terdapat pada direktori **application/models**. Direktori ini berisi kelas *model* dengan
13 fungsi-fungsi untuk mengolah data pada aplikasi. Berikut merupakan *model* pada *SharIF Judge*
14 beserta setiap fungsinya:

15 **Assignment_model.php**

16 **Assignment_model.php** terdapat beberapa fungsi untuk memproses data pada tabel *assignment*.
17 Berikut merupakan fungsi-fungsi dari **Assignment_model.php**:

- 18 • **add_assignment**
Fungsi ini berguna untuk menambahkan atau memperbarui *assignment* pada *database*.
- 19 • **delete_assignment**
Fungsi ini berguna untuk menghapus *assignment* pada *database*.
- 20 • **all_assignments**
Fungsi ini berguna untuk mengembalikan seluruh *assignment* beserta informasi setiap *assig-
nment* tersebut.
- 21 • **new_assignment_id**
Fungsi ini berguna untuk mencari *id* terkecil yang dapat digunakan untuk menambahkan
22 *assignment* baru.
- 23 • **all_problems**
Fungsi ini berguna untuk mengembalikan seluruh *problems* dari *assignment* yang ada.
- 24 • **problem_info**
Fungsi ini berguna untuk mengembalikan baris tabel untuk *problem* yang diberikan.

- 1 • **assignment_info**
- 2 Fungsi ini berguna untuk mengembalikan baris tabel untuk **assignment** yang diberikan.
- 3 • **is_participant**
- 4 Fungsi ini berguna untuk mengecek apakah *username* merupakan peserta *assignments* atau
- 5 tidak.
- 6 • **increase_total_submits**
- 7 Fungsi ini berguna untuk menambahkan satu buah total *submit*.
- 8 • **set_moss_time**
- 9 Fungsi ini berguna untuk memperbarui "*Moss Update Time*" untuk *assignment* tertentu.
- 10 • **get_moss_time**
- 11 Fungsi ini berguna untuk mengembalikan "*Moss Update Time*" untuk *assignment* tertentu.
- 12 • **save_problem_description**
- 13 Fungsi ini berguna untuk menambahkan atau memperbarui deskripsi *problem*.
- 14 • **_update_coefficients**
- 15 Fungsi ini dipanggil pada fungsi **add_assignment** yang berguna untuk memperbarui koefisien
- 16 dari *assignment* tertentu.

17 **Hof_model.php**

18 Berikut merupakan fungsi-fungsi dari **Hof_model.php** yang berguna untuk mengambil data untuk
19 ditampilkan pada halaman *Hall of Fame*.

- 20 • **get_all_final_submission**
- 21 Fungsi ini berguna untuk mengambil seluruh data *final submission*.
- 22 • **get_all_user_assignments**
- 23 Fungsi ini berguna untuk mengambil detail *assignment* dan *problem* berdasarkan pengguna.

24 **Logs_model.php**

25 Berikut merupakan fungsi-fungsi dari **Logs_model.php** yang berguna untuk mencatat *log* pada
26 beberapa tabel:

- 27 • **insert_to_logs**
- 28 Fungsi ini berguna untuk mencatat *log* pada tabel *login*.
- 29 • **get_all_logs**
- 30 Fungsi ini berguna untuk mengembalikan seluruh *log* dari tabel *login* dalam bentuk *array*.

31 **Notifications_model.php**

32 **Notifications_model.php** terdapat beberapa fungsi untuk memproses data pada tabel *notifications*.
33 Berikut merupakan fungsi-fungsi dari *model* tersebut:

- 34 • **get_all_notifications**
- 35 Fungsi ini berguna untuk mengembalikan seluruh *notifications* dalam bentuk *array*.
- 36 • **get_latest_notifications**
- 37 Fungsi ini berguna untuk mengembalikan sepuluh *notification* terbaru.
- 38 • **add_notification**
- 39 Fungsi ini berguna untuk menambahkan data *notification* baru.

- 1 • **update_notification**
2 Fungsi ini berguna untuk memperbarui *notification* tertentu.
- 3 • **delete_notification**
4 Fungsi ini berguna untuk menghapus *notification* tertentu.
- 5 • **get_notification**
6 Fungsi ini berguna untuk mengembalikan *notification* tertentu dalam bentuk *array*.
- 7 • **have_new_notification**
8 Fungsi ini berguna untuk mengecek apakah terdapat *notification* setelah waktu tertentu.

9 **Queue_model.php**

10 Berikut merupakan fungsi-fungsi dari **Queue_model.php** yang berguna untuk memproses data pada
11 halaman *queue*.

- 12 • **in_queue**
13 Fungsi ini berguna untuk mengecek apakah *submission* pengguna tertentu sudah dalam
14 antrean *submission*.
- 15 • **get_queue**
16 Fungsi ini berguna untuk mengembalikan data seluruh antrian *submission*.
- 17 • **empty_queue**
18 Fungsi ini berguna untuk menghapus seluruh antrean *submission*.
- 19 • **add_to_queue**
20 Fungsi ini berguna untuk memasukan *submission* kedalam tabel *queue*.
- 21 • **rejudge**
22 Fungsi ini berguna untuk menambahkan *submission* kedalam antrean untuk dilakukan *rejudge*.
- 23 • **rejudge_single**
24 Fungsi ini berguna untuk menambahkan satu buah *submission* kedalam antrean untuk dilaku-
25 kan *rejudge*.
- 26 • **get_first_item**
27 Fungsi ini berguna untuk mengambil data *submission* pertama dalam antrean.
- 28 • **remove_item**
29 Fungsi ini berguna untuk menghapus data tertentu dalam antrean.
- 30 • **save_judge_result_in_db**
31 Fungsi ini berguna untuk menyimpan hasil dari *judge* ke dalam *database*.
- 32 • **add_to_queue_exec**
33 Fungsi ini berguna untuk menambahkan data *dummy* pada antrean.

34 **Scoreboard_model.php**

35 Berikut merupakan fungsi-fungsi dari **Scoreboard_model.php** yang berguna untuk memproses data
36 untuk ditampilkan pada halaman *Score Board*.

- 37 • **_generate_scoreboard**
38 Fungsi ini dipanggil pada fungsi **update_scoreboard** dan berfungsi untuk menghasilkan
39 *scoreboard* dari *final submission*.

1 • **update_scoreboards**

2 Fungsi ini berguna untuk memperbarui *cache scoreboard* dari seluruh *assignment*. Fungsi
3 ini dipanggil setiap terdapat penghapusan pengguna atau penghapusan seluruh *assignment*
4 pengguna.

5 • **update_scoreboard**

6 Fungsi ini berguna untuk memperbarui *cache scoreboard* dari sebuah *assignment*. Fungsi ini
7 menyimpan kode HTML menuju *database* pada tabel *scoreboard*. Fungsi ini dipanggil setelah
8 *judge* atau *rejudge*.

9 • **get_scoreboard**

10 Fungsi ini berguna untuk mengambil seluruh *cache scoreboard* dari *assignment* tertentu dalam
11 bentuk HTML *text*.

12 **Settings_model.php**

13 Berikut merupakan fungsi-fungsi dari **Settings_model.php** yang berguna untuk memproses data
14 untuk ditampilkan pada tabel *settings*.

15 • **get_setting**

16 Fungsi ini berguna untuk mengembalikan data *setting* tertentu.

17 • **set_setting**

18 Fungsi ini berguna untuk memperbarui sebuah *setting* pada *database*.

19 • **get_all_settings**

20 Fungsi ini berguna untuk mengembalikan seluruh data *setting* dalam bentuk *array*.

21 • **set_settings**

22 Fungsi memperbarui beberapa *setting*.

23 **Submit_model.php**

24 Berikut merupakan fungsi-fungsi dari **Submit_model.php** yang berguna untuk memproses data
25 yang berkaitan dengan *submission*.

26 • **get_submission**

27 Fungsi ini berguna untuk mengembalikan baris data *submission* tertentu.

28 • **get_final_submissions**

29 Fungsi ini berguna untuk mengembalikan seluruh *final submission*.

30 • **get_all_submissions**

31 Fungsi ini berguna untuk mengembalikan seluruh *submission*.

32 • **count_final_submissions**

33 Fungsi ini berguna untuk menghitung seluruh *final submission*.

34 • **count_all_submissions**

35 Fungsi ini berguna untuk menghitung seluruh *submission*.

36 • **set_final_submission**

37 Fungsi ini berguna untuk memilih *submission* tertentu menjadi *final submission*.

38 • **add_upload_only**

39 Fungsi ini berguna untuk menambahkan hasil dari *upload only problem* ke dalam *database*.

1 **User_model.php**

2 Berikut merupakan fungsi-fungsi dari `User_model.php` yang berguna untuk memproses data pada
3 tabel `users`.

4 • **have_user**

5 Fungsi ini berguna untuk mengecek apakah terdapat pengguna dengan `username` tertentu.

6 • **user_id_to_username**

7 Fungsi ini berguna untuk mengubah `user id` menjadi `username`.

8 • **username_to_user_id**

9 Fungsi ini berguna untuk mengubah `username` menjadi `user id`.

10 • **have_email**

11 Fungsi ini berguna untuk mengecek apakah terdapat `username` dengan `email` tertentu.

12 • **add_user**

13 Fungsi ini berguna untuk menambahkan sebuah pengguna.

14 • **add_users**

15 Fungsi ini berguna untuk menambahkan banyak pengguna.

16 • **delete_user**

17 Fungsi ini berguna untuk menghapus pengguna tertentu.

18 • **delete_submissions**

19 Fungsi ini berguna untuk menghapus seluruh `submission` pada pengguna tertentu.

20 • **validate_user**

21 Fungsi ini berguna untuk mengecek `username` dan `password` apakah sesuai dengan data yang
22 dimasukan.

23 • **selected_assignment**

24 Fungsi ini berguna untuk mengembalikan `selected assignment` untuk pengguna tertentu.

25 • **get_names**

26 Fungsi ini berguna untuk mengembalikan nama dari `username` pengguna tertentu.

27 • **update_profile**

28 Fungsi ini berguna untuk memperbarui profil dari pengguna seperti nama, `email`, `password`,
29 dan `role`.

30 • **send_password_reset_mail**

31 Fungsi ini berguna untuk menghasilkan `password reset key` dan mengirim `email` untuk mela-
32 kukan `reset password`.

33 • **passchange_is_valid**

34 Fungsi ini berguna untuk mengecek apakah `password reset key` yang diberikan sesuai atau
35 tidak.

36 • **reset_password**

37 Fungsi ini berguna untuk mengatur ulang `password` sesuai dengan `password key` tertentu.

38 • **get_all_users**

39 Fungsi ini berguna untuk mengembalikan seluruh pengguna untuk halaman `users`.

40 • **get_user**

41 Fungsi ini berguna untuk mengembalikan baris data untuk pengguna tertentu.

42 • **update_login_time**

1 Fungsi ini berguna untuk memperbarui *login time* dan *last login time* untuk pengguna
 2 tertentu.

3 **User.php**

4 Berikut merupakan fungsi-fungsi dari **User.php** yang berguna untuk memproses data pada tabel
 5 *users*.

- 6 • **select_assignment**

7 Fungsi ini berguna untuk mengatur *assignment* yang dipilih oleh pengguna.

- 8 • **save_widget_positions**

9 Fungsi ini berguna untuk memperbarui posisi dari *dashboard widget* pada *database*.

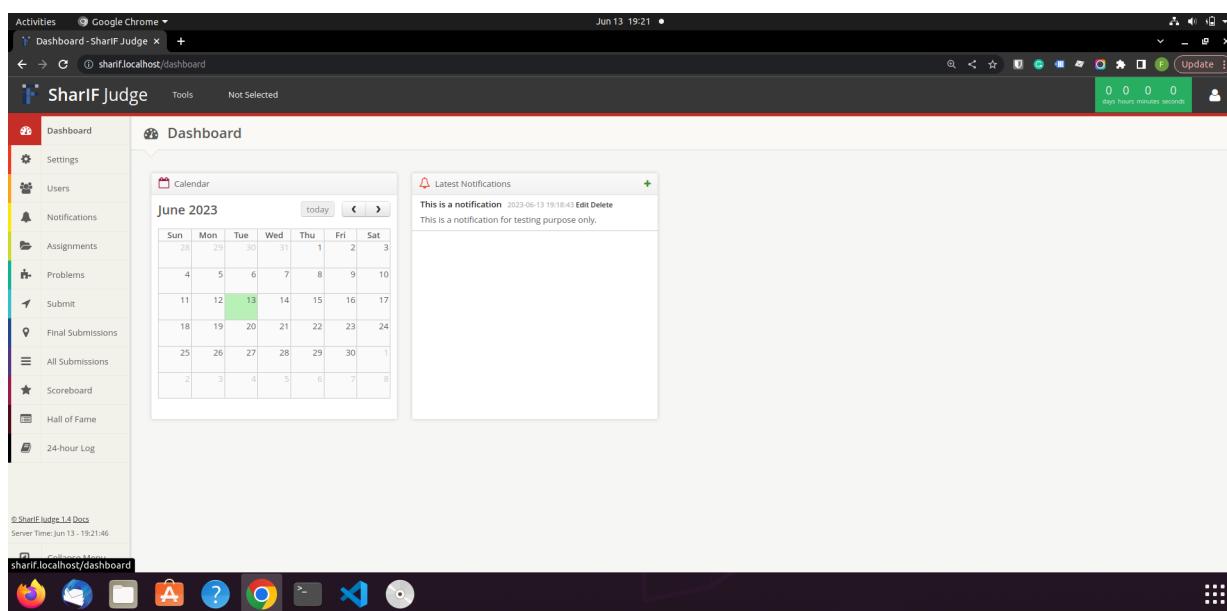
- 10 • **get_widget_positions**

11 Fungsi ini berguna untuk mengembalikan data *dashboard widget*.

12 **3.1.2 View**

13 *View* terdapat pada direktori **application/views**. Direktori ini berisikan seluruh *file* untuk tam-
 14 pilan halaman *SharIF Judge* yang dipisahkan oleh direktori sesuai dengan fungsi dan aksesnya.
 15 Direktori tersebut dibagi menjadi tiga buah direktori utama yakni *error*, *pages*, dan *templates*.
 16 Direktori *error* berisikan tampilan halaman *error* yang akan dilihat oleh pengguna. Direktori *pages*
 17 merupakan tampilan utama *SharIF Judge* yang terbagi lagi menjadi dua buah direktori yakni *admin*
 18 dan *authentication*. Direktori *admin* berisikan tampilan halaman untuk *role admin*. Direktori
 19 *authentication* berisikan tampilan halaman untuk akses pengguna seperti *Login*, *Register*, dan *Reset
 20 Password*. Direktori *templates* terdiri dari tampilan yang digunakan oleh seluruh tampilan halaman
 21 seperti *header* dan *side bar*. Berikut merupakan tampilan halaman pada *SharIF Judge*:

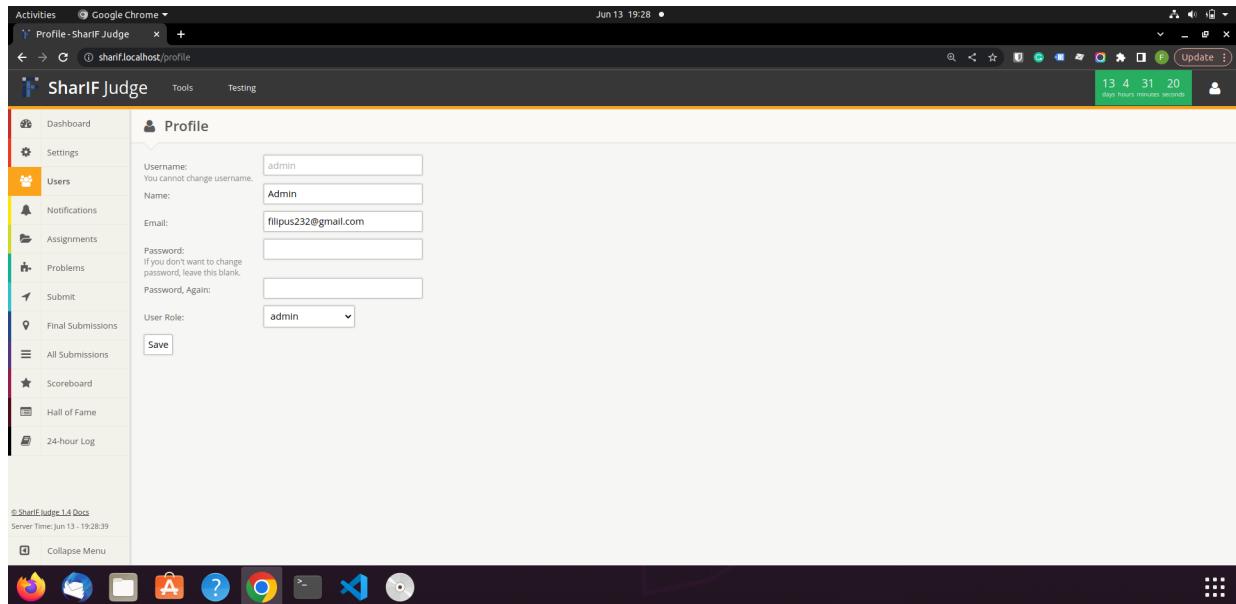
22 **Dashboard**



Gambar 3.1: Tampilan Halaman Dashboard

- 1 Gambar 3.1 menunjukkan tampilan halaman *dashboard* yang terdapat pada semua *role* pengguna.

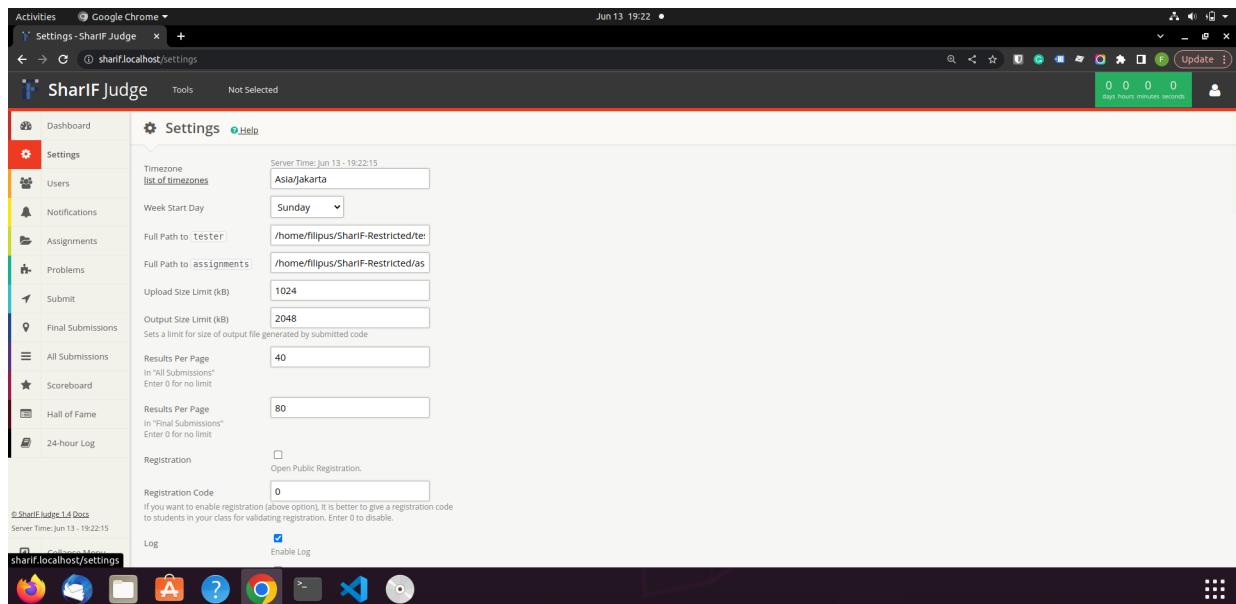
2 *Profile*



Gambar 3.2: Tampilan Halaman Profile

- 3 Gambar 3.2 menunjukkan tampilan halaman *profile* yang terdapat pada semua *role* pengguna. Namun, 4 terdapat fitur yang tidak dapat digunakan oleh *siswa* dan *instructor* yakni mengganti role.

5 *Settings*



Gambar 3.3: Tampilan Halaman Settings

- 1 Gambar 3.3 menunjukkan tampilan halaman *settings* yang terdapat hanya pada *role* admin dan *head instructor*.

3 *Users*

| # | User ID | Username | Display Name | Email | Role | First Login | Last Login | Actions |
|---|---------|------------|----------------|---------------------------|-----------------|---------------------|---------------------|---------|
| 1 | 1 | admin | Admin | filipus232@gmail.com | admin | 2022-12-15 17:17:42 | 2023-06-11 19:17:53 | |
| 2 | 2 | instructor | Instructor One | instructor@sharifjudge.ir | head_instructor | 2022-12-16 12:19:44 | 2022-12-16 12:19:44 | |
| 3 | 4 | I19074 | Student One | st1@sharifjudge.ir | student | 2022-12-16 12:13:55 | 2022-12-20 01:00:04 | |
| 4 | 5 | student2 | Student Two | st2@sharifjudge.ir | student | Never | Never | |
| 5 | 6 | student3 | Student Three | st3@sharifjudge.ir | student | Never | Never | |
| 6 | 7 | student4 | Student Four | st4@sharifjudge.ir | student | Never | Never | |
| 7 | 8 | student5 | Student Five | st5@sharifjudge.ir | student | Never | Never | |
| 8 | 9 | student6 | Student Six | st6@sharifjudge.ir | student | Never | Never | |

Gambar 3.4: Tampilan Halaman Users

- 4 Gambar 3.4 menunjukkan tampilan halaman *users* yang terdapat hanya pada *role* admin dan *head instructor*.

6 *Notifications*

This is a notification

2023-06-13 19:18:43 [Edit](#) [Delete](#)

This is a notification for testing purpose only.

Gambar 3.5: Tampilan Halaman Notifications

- 1 Gambar 3.5 menunjukkan tampilan halaman *notifications* yang terdapat pada semua *role* pengguna.

2 *Assignments*

| Select | Name | Problems | Submissions | Coefficient | Start Time | Finish Time | Status | PDF | Actions |
|--------------------------|---------|-----------|---------------|-------------|---------------------|---------------------|--------|-----|---------|
| <input type="checkbox"/> | Testing | 1 problem | 0 submissions | 100 % | 2023-06-13 00:00:00 | 2023-06-27 00:00:00 | Close | | |

Gambar 3.6: Tampilan Halaman Assignments

- 3 Gambar 3.6 menunjukkan tampilan halaman *assignments* yang terdapat pada semua *role* pengguna.
 4 Namun, terdapat bagian yang tidak dapat diakses oleh *role* siswa dan *instructor* yakni bagian *actions*.

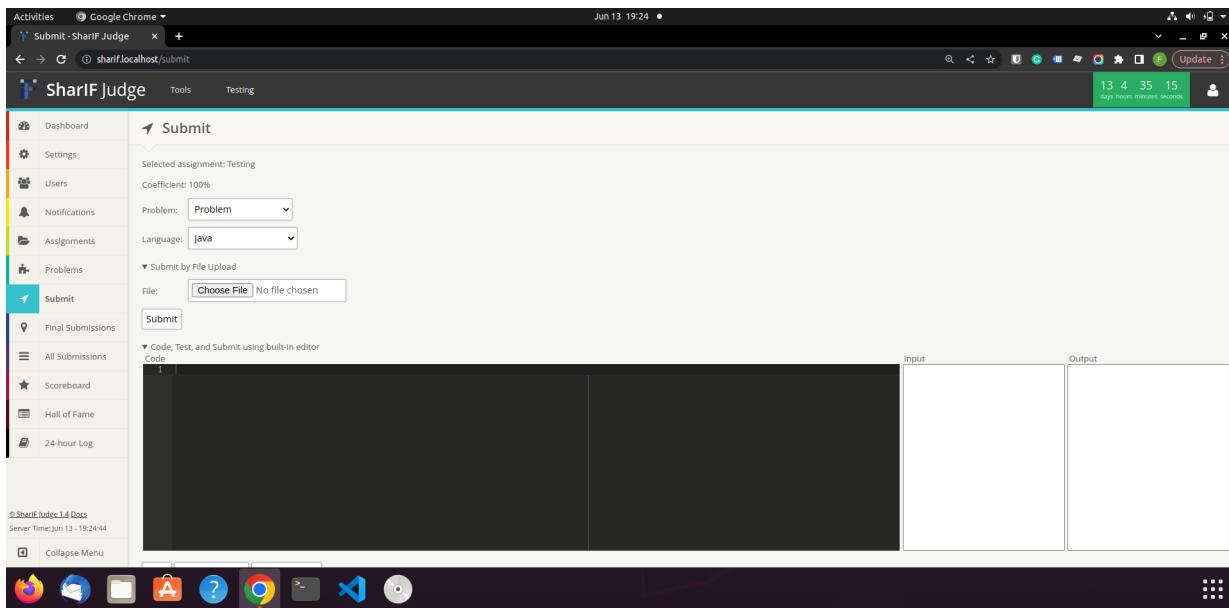
6 *Problems*

| # | Problem | Score | Upload Only |
|---|---------|-------|-------------|
| 1 | Problem | 100 | No |

Gambar 3.7: Tampilan Halaman Problems

- 1 Gambar 3.7 menunjukkan tampilan halaman *problems* yang terdapat pada semua *role* pengguna.

2 *Submit*



Gambar 3.8: Tampilan Halaman Submit

- 3 Gambar 3.8 menunjukkan tampilan halaman *submit* yang terdapat pada semua *role* pengguna.

4 *Final Submissions*

| # | ID | Username | Name | Problem | Submit Time | Score | Delay % | Final Score | Language | Status | Code | Log | Actions |
|---|----|----------|-------|---------|---------------------|-------|------------------|-------------|----------|-------------------|------|-----|---------|
| 1 | 1 | admin | Admin | 1 | 2023-06-13 19:26:24 | 0 | No delay 100% | 0 | java | Compilation Error | Code | Log | |

Gambar 3.9: Tampilan Halaman Final Submission

- 5 Gambar 3.9 menunjukkan tampilan halaman *submit* yang terdapat pada semua *role* pengguna.

1 All Submissions

| Final | ID | Username | Name | Problem | Submit Time | Score | | | Language | Status | Code | Log | Actions |
|--------------------------------------|----|----------|-------|---------|---------------------|-------|------------------|-------------|----------|-------------------|---|--|---|
| | | | | | | Score | Delay % | Final Score | | | | | |
| ● | 2 | admin | Admin | 1 | 2023-06-13 19:28:03 | 0 | No Delay 100% | 0 | Py 3 | Syntax Error | Code | Log | Actions |
| ○ | 1 | admin | Admin | 1 | 2023-06-13 19:26:24 | 0 | No Delay 100% | 0 | Java | Compilation Error | Code | Log | Actions |

Gambar 3.10: Tampilan Halaman All Submission

- 2 Gambar 3.10 menunjukkan tampilan halaman *All Submission* yang terdapat pada semua *role* pengguna.
- 3

4 Scoreboard

| # | Username | Name | Problem | Total |
|---|----------|-------|---------|------------|
| 1 | admin | Admin | 100 | 0 (0:0) |

Gambar 3.11: Tampilan Halaman Scoreboard

- 5 Gambar 3.10 menunjukkan tampilan halaman *All Submission* yang terdapat pada semua *role* pengguna.
- 6

1 Hall of Fame

| Rank | Username | Display Name | Total Score |
|------|----------|--------------|-------------|
| 1 | admin | Admin | 0 |

Gambar 3.12: Tampilan Halaman Hall of Fame

2 Gambar 3.12 menunjukkan tampilan halaman *Hall of Fame* yang terdapat pada semua *role* pengguna.

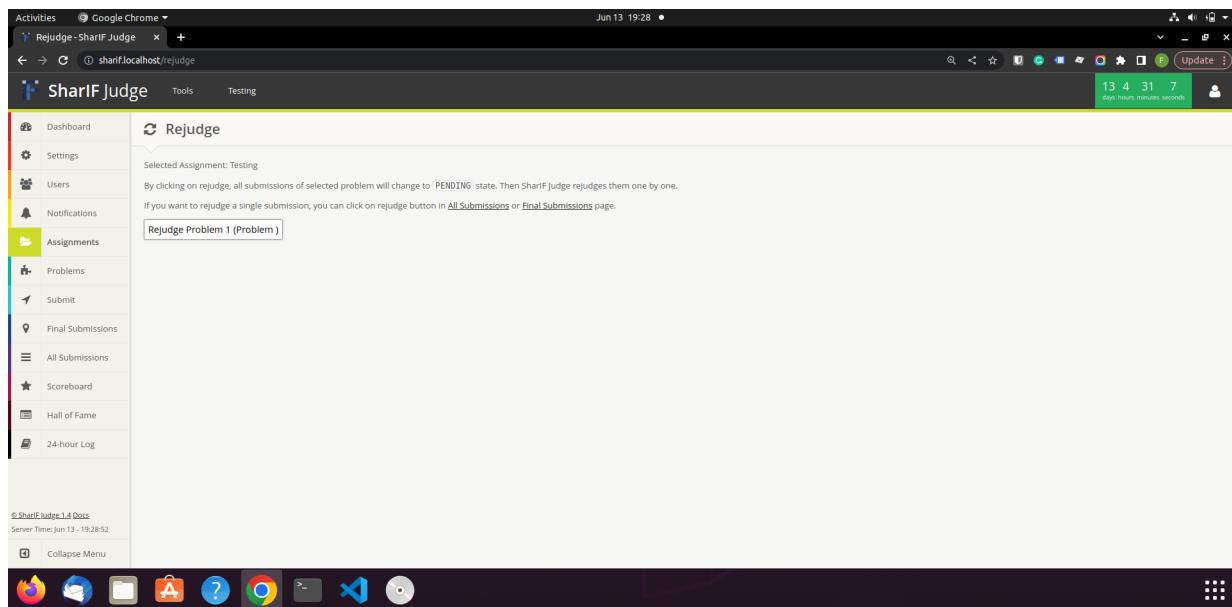
3 24-hour Log

| # | Login ID | Username | IP Address | Login Time | Log from different IP (< 24 hours) |
|---|----------|----------|------------|------------|------------------------------------|
| | | | | | |

Gambar 3.13: Tampilan Halaman 24-hour Log

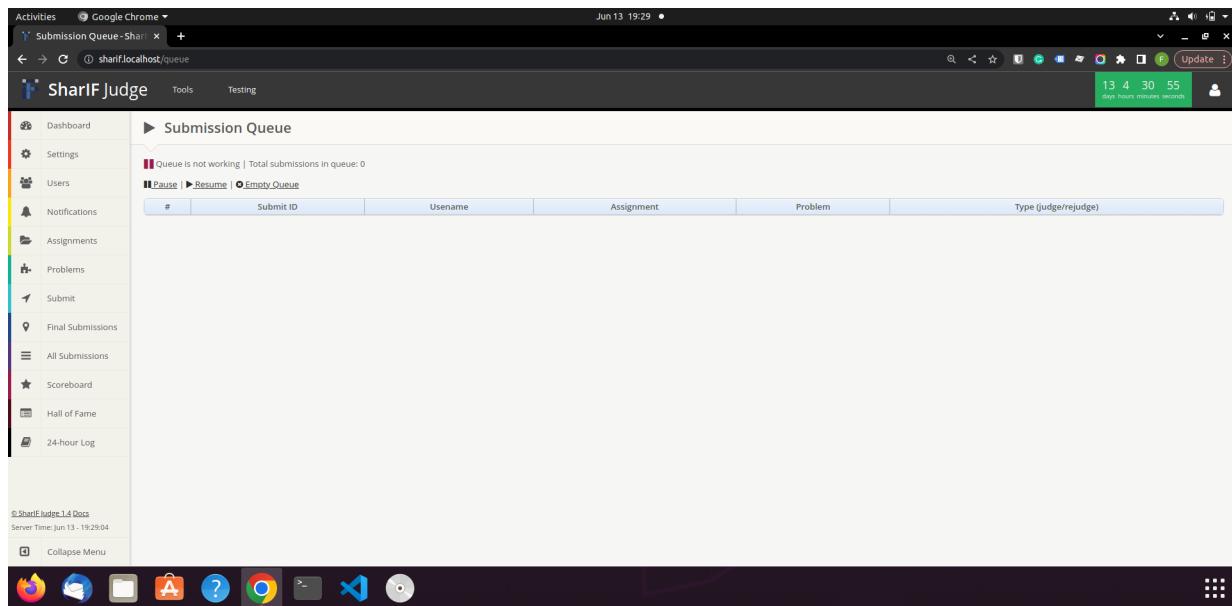
4 Gambar 3.13 menunjukkan tampilan halaman *24-hour Log* yang terdapat hanya pada *role admin* dan *head instructor*.

1 ***Rejudge***



Gambar 3.14: Tampilan Halaman ReJudge

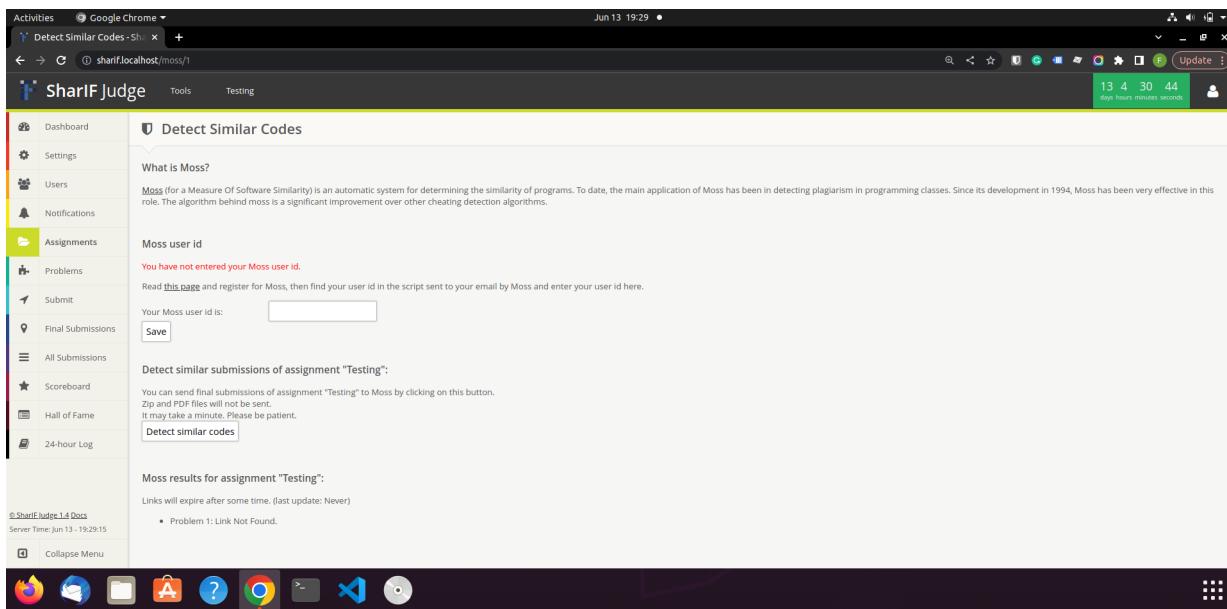
- 2 Gambar 3.14 menunjukkan tampilan halaman *ReJudge* yang terdapat hanya pada *role admin* dan *head instructor*.
- 4 ***Submission Queue***



Gambar 3.15: Tampilan Halaman Submission Queue

- 5 Gambar 3.15 menunjukkan tampilan halaman *Submission Queue* yang terdapat hanya pada *role admin* dan *head instructor*.

1 Cheat Detection



Gambar 3.16: Tampilan Halaman Cheat Detection

- 2 Gambar 3.16 menunjukkan tampilan halaman *Cheat Detection* yang terdapat hanya pada *role admin*
3 dan *head instructor*.

4 3.1.3 Controller

- 5 *Controller* berada pada direktori `application/controller`. Direktori ini berisikan kelas *controller*
6 dengan fungsi-fungsi untuk mengambil atau memberikan data menuju *models*. Data-data ini
7 selanjutnya dialihkan menuju *views* untuk ditampilkan kepada pengguna. Berikut merupakan
8 *controller* pada *SharIF Jugde* beserta fungsi-fungsinya:

9 Assignments.php

- 10 Berikut merupakan fungsi-fungsi pada *controller* `Assignments.php`:

- 11 • **index**

12 Fungsi ini berguna untuk mengambil dan memberikan data menuju halaman `assignments.twig`
13 menggunakan `Assignment_model.php`.

- 14 • **select**

15 Fungsi ini berguna untuk memilih *assignment* melalui *top bar* menggunakan AJAX.

- 16 • **pdf**

17 Fungsi ini berguna untuk mengunduh *assignment* atau *problem* dalam bentuk pdf.

- 18 • **downloadtestsdesc**

19 Fungsi ini berguna untuk mengompres dan mengunduh data *test* dan deskripsi sebuah
20 *assignment*.

- 21 • **download_submissions**

22 Fungsi ini berguna untuk mengompres dan mengunduh kode terakhir sebuah *assignment*

1 pengguna.
2 • **delete**
3 Fungsi ini berguna untuk menghapus *assignment*.
4 • **add**
5 Fungsi ini berguna untuk mengambil data masukan pengguna untuk menambahkan atau
6 mengubah *assignment*.
7 • **_add**
8 Fungsi ini berguna untuk menambah atau mengubah *assignment*.
9 • **edit**
10 Fungsi ini berguna untuk mengecek *role* pengguna apakah dapat mengubah *assignment*.
11 Selanjutnya pengguna dikembalikan menuju fungsi **add**.
12 • **pdfCheck** Fungsi ini berguna untuk mengecek *file PDF* dari sebuah *assignment*.

13 **Dashboard.php**

14 Berikut merupakan fungsi-fungsi pada *controller Dashboard.php*:

15 • **index**
16 Fungsi ini berguna untuk mengambil dan memberikan data menuju halaman *dashboard* meng-
17 gunakan tiga buah *model*. *Model* tersebut terdiri dari **Assignment_model**, **Settings_model**,
18 dan **Notifications_model**.
19 • **widget_positions**
20 Fungsi ini berguna untuk menyimpan data *widget* pengguna menggunakan AJAX.

21 **Install.php**

22 *Controller Install.php* hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna untuk
23 membentuk tabel yang dibutuhkan oleh *SharIF Judge* pada *database*. Selain itu, fungsi ini juga
24 berguna untuk menerima data pengguna *admin* yang pertama kali melakukan instalasi *SharIF*
25 *Judge* pada perangkat.

26 **Login.php**

27 Berikut merupakan fungsi-fungsi pada *controller Login.php*:

28 • **_registration_code**
29 Fungsi ini berguna untuk memeriksa kode registrasi.
30 • **index**
31 Fungsi ini berguna untuk melakukan validasi *username* dan *password* pengguna pada halaman
32 *login*. Selain itu, fungsi ini juga memperbarui *log* pada tabel *login*.
33 • **register**
34 Fungsi ini berguna untuk melakukkan validasi dalam pembentukan akun pada halaman
35 *register*.
36 • **logout**
37 Fungsi ini berguna untuk menghancurkan *session* dari pengguna dan memindahkan pengguna
38 menuju halaman *login*.

- 1 • **lost**
- 2 Fungsi ini berguna untuk mengirim *email* lupa password.
- 3 • **reset**
- 4 Fungsi ini berguna untuk melakukan *reset password* pengguna.

5 **Logs.php**

6 Controller **Logs.php** hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna untuk
7 mengambil dan memberikan data pada halaman *logs* menggunakan **Logs_model**.

8 **Moss.php**

9 Berikut merupakan fungsi-fungsi pada *controller Moss.php*:

- 10 • **index**
- 11 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *moss*. Fungsi ini
12 juga dapat melakukan deteksi kode menuju Moss berdasarkan permintaan.
- 13 • **update**
- 14 Fungsi ini berguna untuk memperbarui **moss_userid** yang dimasukan oleh pengguna.
- 15 • **_detec**
- 16 Fungsi ini berguna untuk mengirimkan data untuk dilakukan pengecekan terhadap *final*
17 *submission* melalui **shell**.

18 **Notification.php**

19 Berikut merupakan fungsi-fungsi pada *controller Notification.php*:

- 20 • **index**
- 21 Fungsi ini berguna untuk mengambil dan mengembalikan data pada halaman *notifications*
22 menggunakan **assignment_model** dan **notifications_model**.
- 23 • **add**
- 24 Fungsi ini berguna untuk menambahkan data *notifications* baru.
- 25 • **edit**
- 26 Fungsi ini berguna untuk memperbarui data *notifications*.
- 27 • **delete**
- 28 Fungsi ini berguna untuk menghapus data *notifications* tertentu.
- 29 • **check**
- 30 Fungsi ini berguna memeriksa *notifications* baru.

31 **Problems.php**

32 Berikut merupakan fungsi-fungsi pada *controller Problems.php*:

- 33 • **index**
- 34 Fungsi ini berguna untuk mengambil dan memberikan data *problems* sesuai dengan *assignment*
35 tertentu pada halaman *problems*.
- 36 • **edit**
- 37 Fungsi ini berguna untuk memperbarui deskripsi *problems* pada *assignment* tertentu.

1 **Profile.php**

2 Berikut merupakan fungsi-fungsi pada *controller Profile.php*:

3 • **index**

4 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *profile*. Selain itu,
5 fungsi ini berguna untuk melakukan pembaharuan data *profile*.

6 • **_password_check**

7 Fungsi ini berguna untuk melakukan validasi terhadap *password* yang akan dimasukkan
8 pengguna sesuai dengan aturan.

9 • **_password_again_check**

10 Fungsi ini berguna untuk melakukan validasi terhadap pengulangan *password* yang dimasukkan
11 pengguna.

12 • **_email_check**

13 Fungsi ini berguna untuk melakukan validasi terhadap *email* yang dimasukkan pengguna

14 • **_role_check**

15 Fungsi ini berguna untuk melakukan validasi *role* pengguna.

16 **Queue.php**

17 • **index**

18 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *queue* menggunakan
19 tiga buah *model*. *Model* tersebut adalah *Assignment_model*, *queue_model*, dan
20 *settings_model*.

21 • **pause**

22 Fungsi ini berguna untuk memberhentikan proses *queue*.

23 • **resume**

24 Fungsi ini berguna untuk melanjutkan proses *queue*.

25 • **empty_queue**

26 Fungsi ini berguna untuk menghapus data tabel *queue*.

27 **Queueprocess.php**

28 Controller *Queueprocess.php* hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna
29 untuk menjalankan proses *judge* berdasarkan *queue* satu demi satu sesuai antrean. Fungsi
30 ini menggunakan beberapa *model* yaitu *Queue_model*, *Submit_model*, *Assignments_model*, dan
31 *Settings_model*.

32 **Rejudge.php**

33 Berikut merupakan fungsi-fungsi pada *controller Rejudge.php*:

34 • **index**

35 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *rejudge* menggunakan
36 *Assignment_model*. Fungsi ini juga berguna untuk melakukan *rejudge* kepada sebuah
37 *problem* tertentu.

- 1 • **rejudge_single**

2 Fungsi ini berguna untuk melakukan *rejudge* pada satu buah masalah tertentu.

3 **Scoreboard.php**

4 Berikut merupakan fungsi-fungsi pada *controller Scoreboard.php*:

- 5 • **index**

6 Fungsi ini berguna untuk memberikan data pada halaman *scoreboard* menggunakan dua buah
7 *model*. *Model* tersebut adalah **Assignment_model** dan **Scoreboard_model**.

8 **Server_time.php**

9 Controller **Server_time.php** hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna
10 untuk mengeluarkan *server_time*.

11 **Settings.php**

12 Berikut merupakan fungsi-fungsi pada *controller Settings.php*:

- 13 • **index**

14 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *settings* menggu-
15 nakan **Settings_model** dan **Assignment_model**.

- 16 • **update**

17 Fungsi ini berguna untuk mengambil masukan dan memperbarui data pada halaman
18 *settings* berdasarkan masukan tersebut. Data tersebut nantinya akan disimpan pada *database*
19 menggunakan fungsi **Settings_model**.

20 **Submission.php**

21 Berikut merupakan fungsi-fungsi pada *controller Submission.php*:

- 22 • **_download_excel**

23 Fungsi ini berguna untuk mengubah dan mengunduh data-data dari *submission* yang dipilih
24 menjadi format *excel*.

- 25 • **final_excel**

26 Fungsi ini berguna untuk mengunduh data *final submissions*.

- 27 • **all_excel**

28 Fungsi ini berguna untuk mengunduh data seluruh *submissions*.

- 29 • **the_final**

30 Fungsi ini berguna untuk memberikan data pada halaman *Final Submissions* menggunakan be-
31 berapa *model*. *Model* tersebut terdiri dari **Submit_model**, **Settings_model**, dan **User_model**.

- 32 • **all**

33 Fungsi ini berguna untuk mengembalikan data pada halaman *All Submissions* menggu-
34 nakan beberapa *model*. *Model* tersebut terdiri dari **Submit_model**, **Settings_model**, dan
35 **User_model**.

- 1 • **select**
- 2 Fungsi ini berguna untuk memilih *submission* yang akan dijadikan *final submission* oleh
- 3 pengguna.
- 4 • **_check_type**
- 5 Fungsi ini berguna untuk melakukan pengecekan tipe *submission* yang telah dikumpulkan
- 6 oleh pengguna.
- 7 • **view_code**
- 8 Fungsi ini berguna untuk memperlihatkan *submission* yang telah dikumpulkan oleh pengguna
- 9 sesuai dengan tipenya.
- 10 • **download_file**
- 11 Fungsi ini berguna untuk mengunduh hasil dari *submission* yang telah dikumpulkan oleh
- 12 pengguna.

13 **Submit.php**

- 14 Berikut merupakan fungsi-fungsi pada *controller Submit.php*:
- 15 • **_language_to_type**
- 16 Fungsi ini berguna untuk mengubah bahasa pemrograman menjadi tipe sesuai dengan pilihan
- 17 pengguna.
- 18 • **_language_to_ext**
- 19 Fungsi ini berguna untuk mengubah bahasa pemrograman menjadi ekstensi sesuai dengan
- 20 pilihan pengguna.
- 21 • **_match**
- 22 Fungsi ini berguna untuk mencocokan tipe dengan ekstensi dari bahasa pemrogramannya.
- 23 • **_check_language**
- 24 Fungsi ini berguna untuk melakukan pengecekan terhadap bahasa pemrograman yang digu-
- 25 nakan.
- 26 • **index**
- 27 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *submit* menggunakan
- 28 **Assignment_model**.
- 29 • **_upload**
- 30 Fungsi ini berguna untuk menyimpan jawaban dan memasukannya ke *queue* untuk dinilai.
- 31 • **load**
- 32 Fungsi ini berguna untuk memuat kode dari *editor file*.
- 33 • **save**
- 34 Fungsi ini berguna untuk meyimpan kode menuju *editor file* dan mengirim ataupun menjalan-
- 35 kannya.
- 36 • **_submit**
- 37 Fungsi ini berguna untuk menambahkan kode pada *queue* untuk dilakukan *judge*.
- 38 • **_execute**
- 39 Fungsi ini berguna untuk menambahkan kode untuk dijalankan atau di *queue*.
- 40 • **get_output** Fungsi ini berguna untuk memuat *file* menjadi hasil eksekusi.

1 **User.php**

2 Berikut merupakan fungsi-fungsi pada *controller User.php*:

- 3 • **index**

4 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *users*.

- 5 • **add**

6 Fungsi ini berguna untuk menambahkan pengguna baru sesuai dengan masukan.

- 7 • **delete**

8 Fungsi ini berguna untuk menghapus pengguna yang dipilih.

- 9 • **delete_submissions**

10 Fungsi ini berguna untuk menghapus *submission* dari sebuah pengguna.

- 11 • **list_excel**

12 Fungsi ini berguna untuk menghasilkan dan mengunduh data pengguna pada format *excel*.

13 **3.1.4 Assets**

14 Direktori ini berisikan seluruh kebutuhan pengguna seperti *library javascript* dan juga gambar yang
15 akan ditampilkan pada aplikasi. Berikut merupakan isi dari direktori ini beserta kegunaannya.

- 16 • Direktori *ace*

17 Direktori ini berisikan *javascript* yang berfungsi untuk menambahkan *code editor* pada aplikasi.

- 18 • Direktori *font*

19 Direktori ini berisikan seluruh *font* yang digunakan oleh aplikasi *SharIF Judge*.

- 20 • Direktori *fullcalendar*

21 Direktori ini berisikan *javascript* dan *css* yang berfungsi untuk menambahkan kalender pada
22 aplikasi.

- 23 • Direktori *gridster*

24 Direktori ini berisikan *javascript*, *css*, dan gambar yang berfungsi untuk membentuk kolom
25 *grid* dengan sifat *drag and drop*.

- 26 • Direktori *images*

27 Direktori ini berisikan seluruh gambar yang digunakan pada aplikasi *SharIF Judge* seperti
28 banner dan juga logo.

- 29 • Direktori *js*

30 Direktori ini berisikan *javascript* dan *jquery*. *Javascript* dan *jquery* yang terdapat pada
31 direktori ini dibentuk secara manual dan diambil dari situs resmi.

- 32 • Direktori *nano_scroller*

33 Direktori ini berisikan *jquery* dan *css* yang berfungsi untuk membentuk *scrollbar* pada aplikasi.

- 34 • Direktori *noty*

35 Direktori ini berisikan *javascript* yang berfungsi untuk mempermudah membentuk pesan *alert*,
36 *success*, dan sebagainya.

- 37 • Direktori *pdfjs*

38 Direktori ini berisikan *javascript* yang berfungsi untuk melakukan *parsing* dan *rendering file*
39 *pdf*.

- 40 • Direktori *reveal*

41 Direktori ini berisikan *jquery* dan *css* yang berfungsi untuk menampilkan *popup* berupa

- 1 halaman yang dikonfigurasikan.
- 2 • Direktori *snippet*
3 Direktori ini berisikan *javascript* dan *css* yang berfungsi sebagai *template* dalam membentuk
4 kode *javascript*.
- 5 • Direktori *styles*
6 Direktori ini berisikan *css* yang berfungsi untuk memperindah aplikasi yang dibentuk.
- 7 • Direktori *tinymce*
8 Direktori ini berisikan *javascript* dan *css* yang berfungsi untuk membentuk *WYSIWYG editor*
9 pada sebuah aplikasi.

10 **3.1.5 Config**

11 Direktori ini berisikan seluruh konfigurasi aplikasi *SharIF Judge* seperti *database*, URL aplikasi,
12 dan lainnya. Berikut merupakan isi dari direktori ini beserta kegunaannya:

- 13 • **autoload.php**

14 File ini berisikan seluruh data yang ingin diinisiasi saat aplikasi dijalankan seperti *libraries*,
15 *helpers*, dan *model*.

- 16 • **config.php**

17 File ini berisikan konfigurasi aplikasi *SharIF Judge* seperti *base url* aplikasi, *encryption key*,
18 konfigurasi *session*, konfigurasi *cookie*, dan *csfr*.

- 19 • **constants.php**

20 File ini berisikan data yang bersifat *global* pada aplikasi.

- 21 • **database.php**

22 File ini berisikan konfigurasi *database* yang digunakan oleh pengguna.

- 23 • **doctypes.php**

24 File ini berisikan deklarasi tipe dokumen untuk *HTML*.

- 25 • **foreign_chars.php**

26 File ini terjemahan data *foreign character* yang terdapat pada *text helper*.

- 27 • **memcached.php**

28 File ini berisikan konfigurasi *server* untuk menyimpan data *cache* pada server.

- 29 • **migration.php**

30 File ini berisikan konfigurasi untuk melakukan *migration* seperti tipe *migration* dan *path file*
31 *migration* disimpan.

- 32 • **mime.php**

33 File ini berisikan tipe-tipe *mime* yang digunakan oleh *upload helper*.

- 34 • **profiler.php**

35 File ini berisikan konfigurasi untuk *profiler*.

- 36 • **routes.php**

37 File ini berisikan konfigurasi *route* dan *route* yang didefinisikan secara manual.

- 38 • **secrets.php**

39 File ini dibentuk oleh pengguna yang berisikan data-data untuk melakukan autentikasi dan
40 konfigurasi *email*.

- 41 • **smiley.php**

1 File ini berisikan terjemahan dari sintaks *emoticon* menjadi gambar yang digunakan pada
 2 *emoticon helper*.

3 • ***twig.php***

4 File ini berisikan konfigurasi *template engine twig*.

5 • ***user_agents.php***

6 File ini berisikan tipe dari *platform*, *browser*, dan lainnya.

7 **3.1.6 Libraries**

8 *SharIF Judge* menggunakan beberapa *library* yang dibentuk secara manual maupun yang sudah
 9 tersedia pada *Codeigniter 3*. Berikut merupakan *library* yang dipakai oleh *SharIF Judge*:

10 ***Unzip***

11 *Unzip* merupakan sebuah *library* yang dibentuk oleh Phil Sturgeon. *Library* ini mewajibkan
 12 pengguna untuk menyalakan *extension Zlib* sebelum dapat digunakan. *Library Unzip* berfungsi
 13 untuk mengextraksi *file* dengan *extension .zip* menuju direktori yang ditentukan dan dapat
 14 mengeluarkan *error* yang sesuai. *Library* ini juga dapat memberi batasan *extension* apa yang
 15 diinginkan dari *file* tersebut. Kode 3.1 menunjukkan contoh penggunaan *library Unzip* pada *SharIF*
 16 *Judge*.

Kode 3.1: Contoh kode penggunaan *Library Unzip*

```
17 $this->load->library('unzip');
18 $this->unzip->allow(array('txt', 'cpp', 'html', 'md', 'pdf'));
19 $extract_result = $this->unzip->extract($u_data['full_path'], $tmp_dir);
```

22 Sintaks `$this->load->library('unzip');` berfungsi untuk melakukan *load library Unzip* agar
 23 dapat digunakan pada fungsi tersebut. Selanjutnya sintaks `$this->unzip->allow();` berfung-
 24 si untuk memberikan batasan *extension* apa saja yang diinginkan dari *file* tersebut. Terakhir
 25 sintaks `$extract_result = $this->unzip->extract($u_data['full_path'], $tmp_dir);` ber-
 26 fungsi untuk melakukan ekstraksi terhadap *file zip* pada `$u_data['full_path']` tersebut menuju
 27 direktori pada variabel `tmp_dir`.

28 ***Twig***

29 *Twig* merupakan sebuah *template engine library* yang digunakan untuk mempermudah dalam
 30 membentuk *view* pada aplikasi. *Twig* terintegrasi dengan fungsi-fungsi pada *CodeIgniter 3* sehingga
 31 dapat menggunakan seluruh fungsi yang terdapat pada *CodeIgniter 3*. Kode 3.2 merupakan contoh
 32 penggunaan *Twig* pada *SharIF Judge*.

Kode 3.2: Contoh *view* menggunakan *library Twig*

```
33 34 1  {% set title = 'Login' %} 
35 2  {% include 'templates/simple_header.twig' %} 
36 3 
37 4  {{ form_open() }} 
38 5  <div class="box login"> 
39 6 
40 7  <div class="judge_logo"> 
41 8  <a href="{{ site_url() }}></a> 
42 9  </div> 
43 10 <div class="login_form">
```

```

12   <div class="login1">
13     <p>
14       <label for="form_username">Username</label><br/>
15       <input id="form_username" type="text" name="username" required="required" pattern="[0-9a-zA-Z]{3,20}" title="The
16         Username field must be between 3 and 20 characters in length, and contain only digits and lowercase
17         letters" class="sharif_input" value="{{ set_value('username') }}" autofocus="autofocus"/>
18       {{ form_error('username', '<div class="shj_error">', '</div>') }}
19     </p>
20     <p>
21       <label for="form_password">Password</label><br/>
22       <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The
23         Password field must be at least 6 characters in length" class="sharif_input"/>
24       {{ form_error('password', '<div class="shj_error">', '</div>') }}
25     </p>
26     {% if error %}
27       <div class="shj_error">Incorrect username or password.</div>
28     {% endif %}
29   </div>
30   <div class="login2">
31     <p style="margin:0;">
32       {% if registration_enabled %}
33         <a href="{{ site_url('register') }}>Register</a> |
34       {% endif %}
35         <a href="{{ site_url('login/lost') }}>Reset Password</a>
36         <input type="submit" value="Login" id="sharif_submit"/>
37     </p>
38   </div>
39 </div>
40 </form>
41 </body>
42 </html>

```

35 Twig pada *view SharIF Judge* menggunakan dua buah *delimiters* yakni `{{ }}` dan `{% %}`.
 36 *Delimiters* `{{ }}` memiliki fungsi untuk mengembalikan *expression* seperti variabel ataupun fungsi
 37 *CodeIgniter 3*. Contoh fungsi yang dikembalikan oleh *delimiters* pada kode diatas adalah `form_open`
 38 yang merupakan sebuah fungsi pada *CodeIgniter 3* untuk membuka *tag form*. Sedangkan *delimiters*
 39 `{% %}` memiliki fungsi untuk mengeksekusi fungsi PHP seperti *for-loops* atau *if else*. Contoh fungsi
 40 yang dieksekusi pada kode diatas adalah `if` yang berfungsi untuk mengecek kondisi tertentu.

41 *Password_hash*

42 *Password_hash* merupakan sebuah *library* yang dibentuk oleh *phpass*. *Library* ini berfungsi untuk
 43 melakukan enkripsi *password* dan melakukan verifikasi *password*. *Library* ini mendukung beberapa
 44 metode enkripsi antara lain *CRYPT_BLOWFISH* dan *CRYPT_EXT_DES*. Kode 3.3 menunjukan
 45 contoh penggunaan *library* ini pada *SharIF Judge*.

Kode 3.3: Contoh kode penggunaan *Library Password_hash*

```

46 $this->load->library('password_hash', array(8, FALSE));
47 $user['password'] = $this->password_hash->HashPassword($this->input->post('password'));
48

```

50 Kode 3.3 melakukan *load* pada *library Password_hash*. Selanjutnya sintaks akan melakukan
 51 *hashing* pada *input* menggunakan algoritma yang ditentukan dan menyimpannya pada sebuah
 52 variabel.

53 *MY_Form_validation*

54 *MY_Form_validation* merupakan *library* yang dibentuk secara manual untuk menambahkan fungsi
 55 validasi yang sudah tersedia pada *CodeIgniter 3*. *Library* ini memiliki dua buah fungsi yakni:

1 • **required**

2 Fungsi ini berguna untuk melakukan pengecekan terhadap sebuah *input* apakah berisikan
3 sebuah *array* kosong ataupun *string* kosong.

4 • **lowercase**

5 Fungsi ini berguna untuk melakukan pengecekan apakah *input* berisikan kata-kata dengan
6 huruf kecil atau tidak.

7 ***MY_Profiler***

8 *MY_Profiler* merupakan *library* yang dibentuk secara manual dan merupakan perpanjangan dari
9 *CI_Profiler*. *Library* ini berfungsi untuk mengembalikan data yang telah dijalankan oleh *profiler*
10 dan menyimpannya pada halaman pengguna. *Library* ini tidak akan digunakan lagi karena tidak
11 terdapat pada *CodeIgniter 4* dan tidak dipakai pada *SharIF Judge*.

12 ***Parsedown***

13 *Parsedown* merupakan sebuah *library* yang dibentuk oleh Emanuil Rusev. *Library* ini berfungsi
14 untuk mengubah teks dengan sintaks *markdown* menjadi teks dalam bentuk *file* lain seperti *HTML*.
15 Kode 3.4 merupakan contoh penggunaan *library parsedown*.

Kode 3.4: Contoh kode penggunaan *Library Parsedown*

```
16 1     $this->load->library('parsedown');
17 2     $html = $this->parsedown->parse(file_get_contents("$assignment_dir/p$i/desc.md"));
```

20 Kode 3.4 akan menginisiasi *library parsedown* pada sintaks baris pertama. Selanjutnya isi dari
21 *file desc.md* akan diambil dan dilakukan *parsedown* menjadi *file* *HTML*. Berikut merupakan contoh
22 dari teks sebelum dan sesudah di *parsedown*:

23 Ini adalah list :

- 24 1. Satu
- 25 2. Dua
- 26 3. Tiga

Kode 3.5: Contoh hasil teks sesudah dilakukan *parsedown*

```
27 1     <p>Ini adalah list</p>
28 2     <ol>
29 3       <li>Satu</li>
30 4       <li>Dua</li>
31 5       <li>Tiga</li>
```

34 Kode 3.5 menunjukan contoh kode *HTML* sesudah dilakukan *parsedown*. Angka 1, 2, dan 3
35 akan diubah menjadi *list* yang dapat ditampilkan pada *HTML*.

36 ***Phpxcel***

37 *Phpxcel* merupakan sebuah *library* yang dibentuk oleh *PHPExcel*. *Library* ini memiliki fungsi untuk
38 mengubah data yang ada pada PHP menjadi *file excel*.

39 ***Shj_pagination***

40 *Shj_pagination* merupakan *library* yang dibentuk secara manual dengan fungsi untuk membatasi
41 maksimal data pada setiap halaman sesuai dengan konfigurasinya.

1 ***Upload***

2 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk menerima
 3 masukan dan mengunggah *file* yang dimasukan oleh pengguna. Kode 3.6 menunjukan contoh
 4 penggunaan *library upload* pada *SharIF Judge* untuk melakukan *upload* sebuah *file*.

Kode 3.6: Contoh penggunaan *library upload*

```

5  $this->load->library('upload');
6  $config = array(
7      'upload_path' => $assignments_root,
8      'allowed_types' => 'zip',
9  );
10  );
11  $this->upload->initialize($config);
12  $zip_uploaded = $this->upload->do_upload('tests_desc');
13  $u_data = $this->upload->data();
14
15
16
17
18

```

15 Kode 3.6 akan melakukan inisiasi terhadap *library upload*. Sedangkan sintaks selanjutnya akan
 16 menentukan dan menginisiasikan konfigurasi terhadap *path* dan tipe apa saja yang dapat dikirimkan
 17 oleh pengguna. Setelah itu akan dilakukan *upload* menggunakan sintaks *do_upload*. Terakhir
 18 seluruh data *file* yang telah dilakukan *upload* disimpan menuju variabel.

19 ***Input***

20 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk menerima
 21 *input* dari pengguna. Kode 3.7 menunjukan penggunaan *library input* pada *SharIF Judge* untuk
 22 menerima input *post* dengan *input form* bernama **name**.

Kode 3.7: Contoh penggunaan *library input*

```

23
24  $this->load->library('upload');
25  $names = $this->input->post('name');
26
27
28
29
30
31
32
33
34
35
36
37
38

```

27 ***Email***

28 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk mengi-
 29 rimkan *email* kepada orang yang dituju sesuai dengan konfigurasinya.

30 ***Form validation***

31 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk melakukan
 32 validasi terhadap data yang dimasukan oleh pengguna sesuai dengan aturan yang telah ditetapkan.

33 ***Kelas URI***

34 *Library* ini berfungsi untuk mengambil informasi yang terdapat pada *segment* dalam sebuah *URI*.
 35 Kode 3.8 merupakan penggunaan *library* ini untuk mengambil *segment* dan mengubahnya menjadi
 36 sebuah *array*.

Kode 3.8: Contoh penggunaan *library URI*

```

37  $input = $this->uri->uri_to_assoc();
38
39
40
41

```

40 Kode 3.8 merupakan fungsi untuk mengambil informasi yang tersedia pada *segment*. Hasil data
 41 dari pengambilan tersebut berupa *associative array*.

1 *Session*

2 *Library* ini berfungsi untuk menyimpan informasi setiap pengguna memakai perangkat lunak.
3 Data-data ini selanjutnya disimpan menuju *database* dengan nama tabel `shj_session`. Data-data
4 yang telah disimpan selanjutnya dapat diambil kembali untuk melakukan pengecekan ataupun
5 disimpan menuju tabel lainnya. Berikut merupakan penggunaan *library session* untuk mengambil
6 data pengguna:

Kode 3.9: Contoh penggunaan *library session*

7 1 `$this->session->userdata('logged_in')`

10 Kode 3.9 merupakan penggunaan *library* ini untuk mengambil data dengan nama `logged_in`.
11 *SharIF Judge* menggunakan tabel untuk menyimpan *session* dengan isi kolom berupa `session_id`,
12 `ip_address`, `user_agent`, `last_activity`, dan `user_data`.

13 **3.2 Analisis Sistem Usulan**

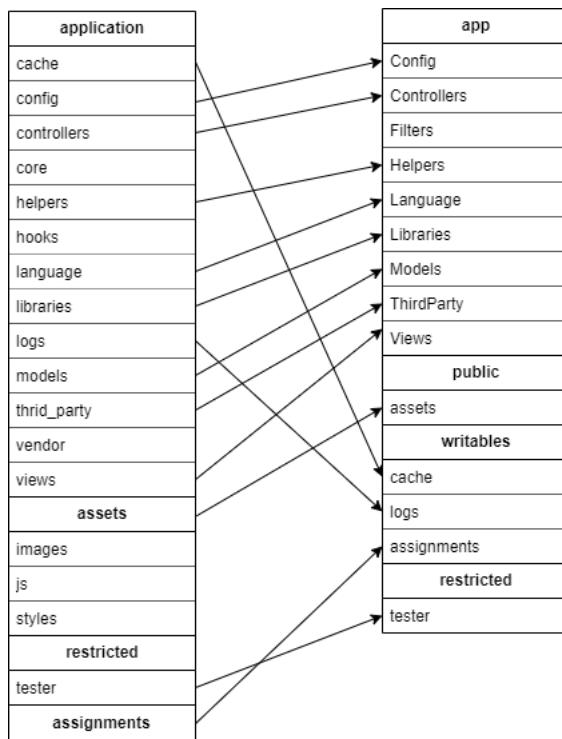
14 Konversi *CodeIgniter 3* menuju *CodeIgniter 4* diperlukan penulisan ulang karena terdapat perubahan
15 struktur aplikasi dan beberapa fungsi yang memiliki pemanggilan berbeda dan harus dilakukan
16 pembaharuan.

17 **3.2.1 Persiapan *CodeIgniter 4***

18 Konversi dimulai dengan mempersiapkan aplikasi *CodeIgniter 4* dengan mengunduh ataupun
19 memasangnya melalui *Composer*. Pengguna juga perlu memasang komponen pendukung seperti
20 `phppoffice`, `radius`, dan `adldap2`.

21 **3.2.2 Struktur Aplikasi**

22 Struktur aplikasi pada *CodeIgniter 3* dan *CodeIgniter 4* memiliki perubahan sehingga perlu dilakukan
23 pemindahan *file* menuju direktori *CodeIgniter 4*. Gambar 3.17 menunjukkan pemindahan struktur
24 aplikasi *SharIF Judge* pada *CodeIgniter 3* menuju *CodeIgniter 4*.



Gambar 3.17: Pemindahan struktur aplikasi menuju *CodeIgniter 4*

1 Berikut merupakan rincian direktori yang dipindahkan menuju *CodeIgniter 4*:

2 Application

3 Direktori-direktori **application** pada *CodeIgniter 3* dipindahkan dengan penyesuaian menuju
4 direktori **app** terkecuali direktori **vendor**, **cache** dan **core**. Berikut merupakan direktori yang
5 dipindahkan dari direktori *application* menuju direktori *app*:

- 6 • **application/config** akan dipindahkan menuju **app/Config**.
- 7 • **application/controllers** akan dipindahkan menuju **app/Controllers**.
- 8 • **application/helpers** akan dipindahkan menuju **app/Helpers**.
- 9 • **application/language** akan dipindahkan menuju **app/Language**.
- 10 • **application/libraries** akan dipindahkan menuju **app/Libraries**.
- 11 • **application/models** akan dipindahkan menuju **app/Models**.
- 12 • **application/views** akan dipindahkan menuju **app/views**.

13 Config

14 Data dari *file* pada direktori *config* *CodeIgniter 3* dipindahkan dengan penyesuaian menuju direktori
15 **Config**. Berikut merupakan *file-file* yang dipindahkan dengan penyesuaian:

- 16 • **autoload.php** akan dipindahkan menuju **BaseController** seperti **helpers** dan terdapat peng-
17 hapusan *autoload library* dan *model* karena terdapat perbedaan cara kerja.
- 18 • **config.php** akan dipecah dan dipindahkan menjadi tiga buah *file* yakni **Config/app.php**,
19 **Config/Session.php**, dan **Config/Security.php**.
- 20 • **constants.php** akan dipindahkan menuju **Config/Constants.php**.

- 1 • `database.php` akan dipindahkan menuju `Config/Database.php`.
- 2 • `routes.php` akan dipindahkan menuju `Config/Routes.php`.
- 3 • `secrets.php` akan dibentuk baru pada direktori `Config`.

4 ***Public***

5 *CodeIgniter 3* tidak menyediakan direktori akar berupa *public* sehingga terdapat perubahan struktur.
6 Struktur berubah dimana direktori yang sebelumnya terdapat pada `application` akan dipindah
7 menuju direktori `public`. Berikut merupakan direktori yang dipindahkan menuju direktori `public`:

- 8 • `assets` akan dipindahkan menuju `public/assets`.

9 Selain struktur aplikasi diatas, *SharIF Judge* memiliki dua buah direktori terpisah diluar direktori
10 utama bernama `assignments` dan `tester`. Direktori `assignments` ini berfungsi untuk menyimpan
11 seluruh *file* yang telah dikumpulkan sedangkan direktori `tester` digunakan untuk melakukan perco-
12 baan untuk keamanan *sandbox*. Kedua direktori ini harus dapat ditulis oleh *PHP* sehingga akan
13 dipindahkan menuju direktori `writables`.

14 ***Writables***

15 Direktori ini merupakan direktori berisikan seluruh direktori yang dapat ditulis oleh PHP. Berikut
16 merupakan direktori yang dipindahkan menuju `writables`:

- 17 • `application/cache` akan dipindahkan menuju `writables/cache`.
- 18 • `application/log` akan dipindahkan menuju `writables/logs`.
- 19 • `assignments` akan dipindahkan menuju `writables/assignments`.

20 ***3.2.3 Routing***

21 *Routing* pada aplikasi *SharIF Judge* menggunakan *auto routing* yang disediakan oleh *CodeIgniter*
22 3. *Auto routing* akan membentuk *url* sesuai dengan `controller` dan `method` yang telah dibentuk
23 tanpa harus didefinisikan secara manual. Penggunaan *auto routing* seperti pada *CodeIgniter 3*
24 memiliki kekurangan pada bagian keamanan dimana `filter` pada `controller` dan proteksi CSRF
25 dilewati. Sehingga, konversi pada aplikasi *SharIF Judge* akan menggunakan *URI Routing* yang
26 didefinisikan secara manual untuk alasan keamanan dan *url* yang fleksibel. Berikut merupakan
27 contoh `route` yang didefinisikan secara manual:

28 \$routes->post("login/register", 'Login::register');

29 *Route* akan didefinisikan secara eksplisit sesuai dengan fungsi dan metodenya karena alasan keamanan
30 dan juga konfigurasi `filters` yang berbeda pada setiap `routes`.

31 ***3.2.4 Model, View, and Controller***

32 *CodeIgniter 4* memiliki perubahan baik dari kegunaan dan cara pemanggilan *Model, View, and*
33 *Controller*. Berikut merupakan perubahan yang terjadi:

1 Model

2 Model pada *CodeIgniter 4* memiliki perubahan dimana *model* dapat digunakan untuk mengambil
 3 data pada satu buah tabel spesifik. Konversi *model* dari *CodeIgniter 3* menuju *CodeIgniter 4* dapat
 4 dilakukan menggunakan dua buah cara yakni:

- 1 Menggunakan *Model* dari *CodeIgniter 3*

6 *Model* akan dipindahkan menuju direktori `app/Models` dan dilakukan beberapa perubahan.
 7 Kode 3.10 merupakan contoh fungsi yang terdapat pada *CodeIgniter 3*.

Kode 3.10: Contoh fungsi untuk mengambil data seluruh user pada *CodeIgniter 3*

```
8
9 public function get_all_users()
10 {
11     return $this->db->order_by('role', 'asc')->order_by('id')->get('users')->result_array();
12 }
```

14 Kode 3.10 mengambil data dari tabel `users` dengan hasil berupa *array* dan diurutkan sesuai
 15 dengan `role` dan `id`nya. Kode akan dilakukan konversi dengan pengubahan sintaks. Berikut
 16 merupakan hasil konversi fungsi untuk mengambil data seluruh *user*.

Kode 3.11: Hasil konversi fungsi untuk mengambil data seluruh user

```
17
18 public function get_all_users()
19 {
20     return $this->db->table('users')->orderBy('role', 'asc')->orderBy('id')->get()->getResultArray();
21 }
```

23 Kode 3.11 merupakan hasil konversi *model SharIF Judge* dari *CodeIgniter 3* menjadi *CodeIgniter 4*. Pada *CodeIgniter 4* diperlukan untuk memberikan nama tabel yang akan diakses dan
 24 juga terdapat perubahan sintaks menjadi *camelCase*.

- 2 Menggunakan *Model* pada *CodeIgniter 4*

27 *CodeIgnier 4* menyediakan fungsi *model* yang dapat dibentuk melalui *command line* untuk
 28 sebuah tabel spesifik. Pengguna dapat membentuk *model* melalui *command line* menggunakan
 29 sintaks sebagai berikut.

30 `make:model <name>`

31 *Model* pada *CodeIgniter 4* menyediakan fungsi untuk mengambil, memasukan, dan memper-
 32 baharui data dari sebuah tabel spesifik tanpa harus membentuk secara manual fungsi-fungsi
 33 tersebut. Pengguna dapat melakukan inisiasi dan memakai fungsi untuk mengambil data
 34 menggunakan sintaks berikut.

35 `$userModel = new \App\Models\UserModel();`

36 `$user = $userModel->findAll();`

37 Sintaks diatas akan mengambil seluruh data dari `$userModel` sesuai dengan konfigurasi yang
 38 telah dilakukan pengguna. Pengguna juga dapat melakukan *create*, *update*, dan *delete* melalui
 39 sintaks berikut.

40 `$this->Notifications_model->delete('notifications', array('id' => $id));`

- 41 Sintaks diatas akan menghapus data dari tabel `notifications` sesuai dengan `id` yang diberikan.
 42 Fungsi ini tidak perlu membentuk sintaks secara manual pada *file models* melainkan hanya perlu

1 memanggil *model* yang ingin digunakan. Konversi aplikasi *SharIF Judge* akan menggunakan cara
 2 pertama dengan penghapusan dan pembaharuan beberapa sintaks. Selain itu, terdapat perubahan
 3 nama *model* dari yang sebelumnya menggunakan *snake case* menjadi *PascalCase* agar sesuai dengan
 4 standar PHP.

5 *View*

6 *View* pada aplikasi *SharIF Judge* menggunakan *template engine* bernama *Twig*. *Twig* merupakan
 7 sebuah *template engine* untuk bahasa pemrograman PHP yang berguna untuk mempermudah
 8 dalam membentuk tampilan sebuah aplikasi *webiste*. *Twig* tidak terintegrasi pada *CodeIgniter*
 9 sehingga terdapat perubahan pada sintaks yang telah dipasang. Selain itu, terdapat beberapa
 10 perubahan fungsi pada *CodeIgniter 4* sehingga perlu dilakukan penyesuaian seperti pengubahan *file*
 11 *extension* dari *.twig* menjadi *.php*. Konversi *SharIF Judge* menuju *CodeIgniter 4* dilakukan dengan
 12 mengubah *view* dari sebelumnya menggunakan *twig* menjadi menggunakan *PHP*. Seluruh *delimiters*
 13 akan diubah menggunakan fitur yang terdapat pada *CodeIgniter 4*. Kode 3.12 menunjukkan contoh
 14 konversi yang dilakukan.

Kode 3.12: Contoh *view* menggunakan *twig*

```
15 15 {% if registration_enabled %}  
16 16   <a href="{{ site_url('register') }}>Register</a> |  
17 17   {% endif %}
```

20 menjadi kode berikut:

Kode 3.13: Contoh *view* menggunakan *php*

```
21 21 <?php if($registration_enabled): ?>  
22 22   <a href=<?=_site_url('register')_?>>Register</a> |  
23 23 <?php endif ?>
```

26 Seluruh sintaks *twig* akan diubah menjadi sintaks *PHP* dari *CodeIgniter 4* seperti yang ditunjukkan
 27 pada kode 3.13.

28 *Controller*

29 *Controller* pada *CodeIgniter 4* memiliki fungsi sama dengan pada *CodeIgniter 3* sehingga hanya
 30 akan dipindahkan dan dilakukan perubahan pada sintaks yang ada. Sintaks yang dihapus berupa
 31 *defined* dan akan digantikan dengan *namespace*. Beberapa sintaks untuk memanggil *models*
 32 dan pengembalian *view* juga akan disesuaikan. Namun, terdapat perubahan pada *constructor*
 33 dimana pada *CodeIgniter 4* terdapat *initController* yang akan dipanggil setelah *constructor*
 34 dijalankan. *Constructor* pada *PHP* tidak diperbolehkan untuk mengembalikan apapun sehingga
 35 terdapat beberapa pemindahan fungsi seperti *redirect()* menuju *filters*. Selain itu, konversi akan
 36 tetap menggunakan *__construct* dengan pemindahan beberapa sintaks menuju *initController*
 37 seperti pemanggilan *helpers* dan variabel yang dapat diakses pada seluruh *controller*. Kode 3.14
 38 menunjukkan contoh penggunaan *initController* untuk melakukan inisiasi fungsi menuju variabel.

Kode 3.14: Penggunaan *initController* untuk inisiasi fungsi menuju variabel

```
39 39 protected $config;  
40 40  
41 41 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)  
42 42 {  
43 43   // Do Not Edit This Line  
44 44 }
```

```

16 parent::initController($request, $response, $logger);
27 // Preload any models, libraries, etc, here.
49
50 // E.g.: $this->session = \Config\Services::session();
51 $this->config = Config('Secrets');
52 }

```

9 Kode 3.14 memanggil variabel `$config` berisikan data dari file `config Secrets.php` yang dapat
 10 diakses oleh seluruh *controller* tanpa perlu melakukan inisiasi lagi. Fungsi lainnya akan dipindahkan
 11 sesuai dengan yang ada pada *CodeIgniter 3* dengan pembaharuan. Selain pembaharuan, akan
 12 terdapat pemindahan variabel *global* yang sebelumnya telah diinisiasi menuju *controller*.

13 IncomingRequest

14 *Input* digantikan menggunakan fungsi *request* dengan perubahan beberapa fungsi. Kode 3.15
 15 menunjukkan perubahan yang terdapat dalam menerima *input* dari pengguna.

Kode 3.15: Contoh perubahan *library request*

```

16 $this->request = \Config\Services::request();
17 $names => $this->request->getPost('name'),
18

```

20 Kode 3.15 melakukan inisiasi terhadap *library* tersebut namun inisiasi tersebut hanya diperlukan
 21 pada file *model* karena fungsi ini sudah terinisiasi pada *controller*. Sintaks selanjutnya mengambil da-
 22 ta bernama `name` yang telah dikirimkan oleh pengguna melalui *post*. Konversi aplikasi *SharIF Judge*
 23 akan menggunakan fungsi ini dengan beberapa perubahan sintaks sesuai dengan dokumentasinya.

24 3.2.5 Libraries

25 *Libraries* pada *CodeIgniter 3* memiliki perubahan dan penghapusan pada *CodeIgnier 4* sehingga
 26 perlu dilakukan pembaharuan. Berikut merupakan *libraries* yang dipakai pada *SharIF Judge*:

27 Emails

28 *Emails* pada *CodeIgniter 4* terdapat perubahan sintaks dan cara pemanggilan sehingga akan
 29 dipindahkan sesuai dengan sintaks yang baru. Sintaks berubah dari yang sebelumnya menggunakan
 30 *snake case* menjadi menggunakan *camelcase*. Kode 3.16 menunjukkan contoh penggunaan *library*
 31 *email*.

Kode 3.16: Contoh perubahan *library emails*

```

32 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
33 $this->email->setTo($user[1]);
34 $this->email->setSubject('SharIF_Judge_Username_and_Password');
35 $text = $this->settings_model->get_setting('add_user_mail');
36 $text = str_replace('{SITE_URL}', base_url(), $text);
37 $text = str_replace('{ROLE}', $user[4], $text);
38 $text = str_replace('{USERNAME}', $user[0], $text);
39 $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
40 $text = str_replace('{LOGIN_URL}', base_url(), $text);
41 $this->email->setMessage($text);
42 $this->email->send()
43

```

45 Kode 3.16 memiliki sintaks dengan nama sama namun terdapat perubahan menjadi *camelCase*.

1 ***Working with Uploaded Files***

2 *Upload* akan diganti menggunakan fungsi *Working with uploaded files* dengan beberapa penggantian
 3 dan penghapusan fungsi. *Working with uploaded files* terdapat perubahan pada beberapa sintaks
 4 dan validasi terhadap *file* yang telah diunggah. Kode 3.17 merupakan perubahan yang terdapat
 5 dalam melakukan *upload* sebuah *file*.

Kode 3.17: Contoh perubahan *library upload*

```
6 $zip_uploaded = $this->request->getFile('tests_desc');
7 $pdf_uploaded->move("$assignment_dir");
8 }
```

10 Kode 3.17 merupakan perubahan dalam melakukan *upload* sebuah *file*. Sintaks *getFile* akan
 11 mengambil *file* bernama *tests_desc* yang dikirimkan oleh pengguna. Sintaks selanjutnya akan
 12 memindahkan *file* tersebut menuju direktori yang telah ditentukan. Pada *CodeIgniter 4* sudah
 13 tidak terdapat konfigurasi aturan dan pengambilan data terhadap *file* yang diterima. Sehingga
 14 fungsi tersebut akan digantikan oleh *validation* dan beberapa fitur yang terdapat pada *library ini*.
 15 Kode 3.18 menunjukkan contoh sintaks untuk melakukan pengecekan *extension* dan pengambilan
 16 data berupa nama dari *file*.

Kode 3.18: Contoh sintaks untuk mengambil *extension* dan data *file*

```
17 if($pdf_uploaded->getExtension() != "pdf"){
18   $zip_uploaded->getName()
19 }
20 }
```

22 ***Working with URIs***

23 Kelas *URI* digantikan dengan fungsi *working with URIs* dengan perubahan fungsi. Berikut meru-
 24 pakan contoh pengambilan sebuah *segment* yang terdapat pada *URI*:

```
25 $temp_inp = $this->uri->getSegments();
```

26 Sintaks diatas akan mengambil semua *segment* yang tersedia pada *URI*. *Segment* yang sudah diambil
 27 akan disimpan menjadi sebuah *array*.

28 ***Session***

29 Pemakaian *library session* terdapat perubahan dalam sintaks dan melakukan inisiasi *library*. *Library*
 30 *session* perlu dilakukan inisiasi pada *CodeIgniter 4*. Berikut merupakan contoh inisiasi *library* ini:

```
31 $this->session = session();
```

32 Sintaks diatas akan melakukan inisiasi terhadap *library* yang nantinya dapat dipakai pada seluruh
 33 *file*. Selanjutnya terdapat perubahan cara pengambilan data berikut merupakan contoh pengambilan
 34 data pada *CodeIgniter 4*:

```
35 $this->session->get('logged_in')
```

36 Sintaks diatas akan mengambil data *session* dengan nama *logged_in*. Penggunaan berubah dari
 37 yang sebelumnya menggunakan sintaks *userdata*. Selanjutnya data tersebut akan disimpan menuju
 38 *database* dengan perubahan dan penghapusan kolom. Kolom tabel akan berubah menjadi empat
 39 buah yakni *id*, *ip_address*, *timestamp*, dan data.

1 Validation

2 *Form_validation* diganti menggunakan fungsi *validation* dengan perubahan dan pengapusan beberapa fungsi. Berikut merupakan contoh penetapan aturan untuk melakukan validasi saat mengumpulkan sebuah data pada *form*:

```
5         $validate->setRule('username', 'username',
6             'required|min_length[3]|max_length[20]');
```

7 Sintaks diatas akan melakukan validasi terhadap *input* yang dimasukan oleh pengguna. Sintaks
8 penetapan aturan berubah menggunakan *camelCase* dari yang sebelumnya berupa *set_rules*
9 menjadi *setRule*. Aturan untuk validasi tetap menggunakan fungsi yang tersedia pada *CodeIgniter*
10 tanpa perubahan namun, terdapat pemindahan aturan yang dibentuk manual dan penambahan
11 beberapa sintaks. Kode 3.19 menunjukkan pemindahan aturan dengan penambahan beberapa sintaks
12 menuju *app/Config/Validation.php*.

Kode 3.19: Contoh perubahan penambahan sintaks pada aturan yang dibentuk manual

```
13 public function email_check($email, $edit_username):bool
14 {
15     $edit_username = explode('/', $edit_username);
16     $user_model = new UserModel();
17     if ($user_model->have_email($email, $edit_username[1]))
18         return FALSE;
19     return TRUE;
20 }
21
22 }
```

23 Aturan pada Kode 3.19 ditambahkan sintaks *explode* untuk memecah *array* dan juga terdapat
24 inisiasi *model*. Aturan ini dapat digunakan seperti aturan lainnya pada sintaks *setRule*. Namun,
25 *CodeIgniter 4* tidak menyediakan fungsi *form_error* sehingga pengembalian *error* diubah dengan
26 menggunakan fungsi baru bernama *validation_errors()*. Fungsi tersebut dapat digunakan untuk
27 mengembalikan *error* setiap *input* apabila terdapat data yang tidak sesuai dengan aturan. *Error*
28 tersebut dapat ditampilkan pada halaman *view* menggunakan sintaks berikut:

```
29 <?= $validation->getError('username'); ?>
```

30 Sintaks diatas akan mengembalikan *error* terhadap *form* dengan nama *username* apabila tidak
31 sesuai dengan aturan yang sudah ditentukan. Variabel *validation* dikirimkan dari *controller*
32 berisikan *library* dari *validation* tersebut.

33 Zip Archive

34 *Zip Encoding* dan *Unzip* digantikan dengan fungsi PHP *zip archive* karena sudah tidak tersedia pada
35 *CodeIgniter 4*. Fungsi *zip archive* terdapat beberapa perbedaan sehingga akan disesuaikan dengan
36 fungsi-fungsi *Zip Encoding*. Kode 3.20 menunjukkan penggunaan *zip archive* untuk melakukan *zip*.

Kode 3.20: Contoh perubahan penggunaan *library Zip Archive* untuk melakukan *zip*

```
37
38 $this->zip = new \ZipArchive();
39 $this->zip->open($zipname, ZipArchive::CREATE);
40 $this->zip->addFile($file_path,"{$item['username']}/p{$item['problem']}.".filetype_to_extension($item['file_type']));
```

42 Kode 3.20 berfungsi untuk melakukan inisiasi terhadap *ZipArchive* dan membuka sebuah *file zip*.
43 Sintaks *addFile* selanjutnya memasukan *file* bernama *file_path* menuju zip yang terdapat pada

1 *array item*. Selain untuk membentuk *file zip*, *zip archive* juga dapat melakukan *unzip* terhadap *file zip*. Kode 3.21 menunjukan penggunaan *ZipArchive* dalam melakukan *unzip*.

Kode 3.21: Contoh perubahan penggunaan *library ZipArchive* untuk melakukan *unzip*

```
3 1 $this->unzip = new ZipArchive();
4 2     // Create a temp directory
5 3     $tmp_dir = "$assignments_root/$tmp_dir_name";
6 4     // Extract new test cases and descriptions in temp directory
7 5     $this->unzip->open("$assignments_root/".$zip_uploaded->getName());
8 6     $extract_result = $this->unzip->extractTo($tmp_dir);
```

11 Kode 3.21 melakukan inisiasi terhadap *ZipArchive*. Kode selanjutnya akan membuka sebuah *zip*
 12 pada *assignments_root* dengan nama yang dibentuk. Terakhir fungsi *extractTo* akan melakukan
 13 extraksi *file* pada direktori yang sudah dibentuk.

14 **Twig**

15 *Library* ini tidak digunakan untuk membentuk *view* pada *CodeIgniter 4* namun, akan ada penggunaan
 16 sebuah fungsi *Twig* yang dibentuk pada direktori *app/Libraries*. Fungsi tersebut bernama
 17 *extra_time_formatter* yang memiliki fungsi untuk mengubah input yang diberikan menjadi
 18 format jam dikali enam puluh menit.

19 **Unzip**

20 *Library* ini tidak akan digunakan dan digantikan oleh *ZipArchive* yang disediakan oleh PHP. Terdapat
 21 perubahan sintaks pada *ZipArchive* sehingga akan disesuaikan. Perubahan dapat dilihat pada
 22 subbab 3.2.5 kode 3.21.

23 **Password_hash**

24 *Library* ini tidak akan digunakan dan akan digantikan oleh *password hash* yang disediakan oleh
 25 PHP .*Library Password_hash* merekomendasikan pegguna untuk menggunakan fungsi *native* yang
 26 disediakan oleh PHP apabila aplikasi mendukung PHP versi 5.5 ke atas. Sehingga, akan dilakukan
 27 konversi menggunakan fungsi yang disediakan oleh PHP bernama *password_hash()*. Seluruh
 28 penggunaan *library* ini akan diubah menggunakan fungsi yang disediakan oleh PHP dengan metode
 29 *hashing* sama yaitu *CRYPT_BLOWFISH*. Perubahan fungsi *hashing* ini bersifat *backward compatible*
 30 sehingga dapat menggunakan *database* aplikasi terdahulu tanpa perlu membentuk data baru. Berikut
 31 merupakan contoh pengubahan kode dari *phpass* menjadi *password_hash*.

32 'password' => \$this->password_hash->HashPassword(\$password)

33 menjadi

34 'password' => password_hash(\$password,PASSWORD_BCRYPT)

35 Sintaks *password_hash()* diatas menerima dua buah parameter yakni data yang ingin di enkripsi
 36 dan tipe enkripsi. Enkripsi akan menggunakan sintaks *PASSWORD_BCRYPT* yang menggunakan tipe
 37 *hash* berupa *CRYPT_BLOWFISH*.

1 MY_Form_validation

2 *Library MY_Form_validation* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini
 3 akan digunakan kembali dengan perubahan `extends` menjadi menuju `Validation`, penghapusan
 4 sintaks `defined`, dan akan ada penambahan `namespace` pada baris awal `file`. Kode 3.22 merupakan
 5 contoh penambahan `namespace` dan penggantian `extends` pada *library* ini.

Kode 3.22: Contoh perubahan *library MY_Form_validation* pada *CodeIgniter 4*

```
6 namespace App\Libraries;
7 1
8 2
9 3 use CodeIgniter\Validation\Validation;
10 4
11 5 class MY_Form_validation extends Validation
```

13 Kode 3.22 menghapus sintaks `defined` dan menggantikannya dengan penambahan `namespace`. Selain
 14 itu, kelas *library* akan `extends Validation`.

15 Parsedown

16 *Library Parsedown* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini akan digunakan
 17 kembali dengan penambahan `namespace` pada baris awal `file` dan penghapusan sintaks `defined`.
 18 Kode 3.23 merupakan contoh penambahan `namespace` dan juga penambahan sintaks `defined`.

Kode 3.23: Contoh perubahan *library Parsedown* pada *CodeIgniter 4*

```
19 namespace App\Libraries;
20 1
21 2
22 3 class Parsedown
```

24 Kode 3.23 menghapus sintaks `defined` dan menggantikannya dengan penambahan `namespace`.

25 Phpexcel

26 *Library* ini akan digunakan kembali namun tidak akan dipindahkan menuju `app/Libraries`. *Library*
 27 akan dilakukan instalasi melalui `composer` dengan sintaks berikut:

```
28 composer require phoffice/phpexcel
```

29 Sintaks diatas akan dijalankan pada akar dari aplikasi dan tidak terdapat perubahan terhadap
 30 penggunaan sintaks ini.

31 Shj_pagination

32 *Library* ini akan digunakan kembali dan dipindahkan menuju direktori `app/Libraries`. Selain itu,
 33 terdapat penambahan `namespace` pada baris awal `file` dan penghapusan sintaks `defined`.

34 3.2.6 Configuration

35 *Configuration* terdapat perubahan nama dari yang sebelumnya `application/config/config.php`
 36 menjadi `app/Config/App.php` dan penambahan `file` dengan nama `app/Config/Secrets.php`. Ber-
 37 hubung dengan perubahan nama tersebut, terdapat beberapa perpindahan sintaks menuju direktori
 38 baru tersebut. Berikut merupakan sintaks yang dipindahkan menuju `app/Config/Security.php`:
 39

- `$config['csrf_protection'] = TRUE;`

```

1   • $config['csrf_token_name'] = 'shj_csrf_token';
2   • $config['csrf_cookie_name'] = 'shjcsrftoken';
3   • $config['csrf_expire'] = 7200;
4   • $config['csrf_regeneration'] = FALSE;

5   Configurations yang telah dipindahkan akan diubah dari yang sebelumnya menggunakan array
6   menjadi menggunakan variable. Seluruh configurations pada CodeIgniter 3 akan dipindahkan
7   menuju CodeIgniter 4 sesuai dengan direktori dan fungsinya. Sedangkan app/Config/App.php
8   dan dipindahkan menuju file .env karena alasan kemudahan untuk penggantian environment dalam
9   melakukan deploy.

```

10 3.2.7 Database

11 *Database* pada *CodeIgniter 4* memiliki fungsi yang sama pada *CodeIgniter 3* sehingga akan dilakukan
 12 pemindahan konfigurasi sesuai dengan yang ada pada *CodeIgniter 3*. Namun, terjadi beberapa
 13 perubahan pada sintaks untuk melakukan koneksi ke *database* dan beberapa sintaks untuk melakukan
 14 *query*. Sintaks koneksi *database* akan berubah dari `$this->load->database();` menjadi `db =`
 15 `db_connect()`. Selain itu, pemanggilan fungsi *query builder* berubah menggunakan *camelcase* dari
 16 yang sebelumnya menggunakan *snakecase*. Seperti dari yang sebelumnya menggunakan sintaks
 17 `get_where` menjadi `getWhere`.

18 *Database* pada aplikasi *SharIF Judge* menggunakan *autoload* yang dapat memuat beberapa
 19 fungsi secara otomatis. Namun, pada *CodeIgniter 4* tidak akan menggunakan *autoload* karena
 20 terdapat perbedaan fungsi *autoload* pada *CodeIgniter 4* sehingga akan dimuat menggunakan `$db =`
 21 `db_connect()` pada *controller* yang membutuhkan.

22 3.2.8 Error Handling

23 *Error Handling* terdapat perubahan sintaks dan perubahan cara pemanggilan. Berikut merupakan
 24 cara pemanggilan *error handling*:

Kode 3.24: Contoh pemanggilan *Error Handling* pada *CodeIgniter 4*

```

25 1 <?php
26 2
27 3 throw new \Exception('Some_message_goes_here');

```

30 Kode 3.24 merupakan cara pemanggilan *error hadnling* pada *CodeIgniter 4*. Sintaks ini akan
 31 memberikan *exception* dengan pesan yang telah ditentukan. Selain memberikan pesan, pengguna
 32 juga dapat mengembalikan *error* berupa 404 dengan sintaks berikut:

Kode 3.25: Contoh pemanggilan *Error Handling 404* pada *CodeIgniter 4*

```

33 1 throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();

```

36 Kode 3.25 akan mengembalikan *error* berupa 404 halaman tidak ditemukan.

37 3.2.9 Helpers

38 *Helpers* tidak terdapat banyak perubahan namun, terdapat perubahan pemanggilan *helpers* dan
 39 beberapa penghapusan *helpers*. Berikut merupakan *helpers* yang dihapus dan diubah cara pemang-
 40 gilannya:

- 1 • *Download Helper* ini sudah tidak tersedia pada *CodeIgniter 4* sehingga perlu dilakukan
2 pengubahan dengan menggunakan fungsi *HTTP Response*. *HTTP Response* menyediakan
3 fungsi bernama *Force File Download* yang berguna untuk mengunduh sebuah *file* menuju
4 perangkat pengguna. Fungsi ini dapat dipanggil menggunakan sintaks berikut.

```
5                  return $this->response->download($name, $data);
```

6 Sintaks diatas mengembalikan *file* yang ingin diunduh pengguna dengan dua buah parame-
7 ter. Parameter pertama berupa nama *file* yang ingin diunduh sedangkan parameter kedua
8 merupakan data dalam *file* tersebut.

- 9 • **redirect()**

10 Fungsi ini memiliki perubahan pada *CodeIgniter 4* dimana **redirect()** tidak langsung meng-
11 arahkan kepada *url* yang dibentuk. Pengguna harus mengembalikan fungsi *redirect()* menggu-
12 nakan **return** dengan sintaks sebagai berikut.

```
13                  return redirect()->to('login/form')
```

14 Sintaks diatas akan mengembalikan pengguna menuju *url login/form* yang sudah dibentuk
15 pada *routes*.

- 16 Konversi *helpers* akan dipindahkan dari yang sebelumnya menggunakan *autoload* menuju
17 **initController** dengan menambahkan pada variabel *helpers*.

1

BAB 4

2

PERANCANGAN

3 Bab ini membahas perancangan untuk seluruh implementasi *SharIF Judge* pada *CodeIgniter 4*.

4 **4.1 Instalasi *CodeIgniter 4***

5 *CodeIgniter 4* diinstalasi menggunakan *composer*. *Composer* merupakan sebuah *dependency manager* untuk PHP yang memungkinkan pengguna untuk melakukan instalasi seluruh kebutuhan untuk menjalankan program berbasis PHP. Instalasi akan dilakukan menggunakan sintaks sebagai berikut:

6 `composer create-project codeigniter4/appstarter SharIF-JudgeCI4`

7 Sintaks diatas akan mengunduh dan melakukan instalasi *template* projek *CodeIgniter 4*. Template berisikan *skeleton* dari projek *CodeIgniter 4* yang berisikan data untuk melakukan *development* sebuah aplikasi.

8 **4.2 Perubahan Struktur Aplikasi**

9 Struktur aplikasi *SharIF Judge* dipindahkan seperti pemetaan pada bab 3 gambar 3.17. Pemindahan dilakukan dengan penyesuaian dan perubahan beberapa sintaks dan fungsi yang digunakan. Struktur aplikasi pada *CodeIgniter 4* berisikan sebagai berikut :

10 **4.2.1 app/Config**

11 *File config* pada *CodeIgniter 3* dipindahkan sesuai dengan pemetaan pada gambar 3.17. Direktori berisikan data-data pada *application/Config*. Beberapa data pada direktori ini dipindahkan menuju *file* yang terdapat pada *CodeIgniter 4*. Terdapat juga penambahan *file Secrets.php* yang dibentuk secara manual. Berikut merupakan rincian isi direktori yang dipindahkan:

12 **app/Config/App.php**

13 *File* ini tidak digunakan sehingga dibiarkan berisi sintaks *default* dan seluruh data akan dipindahkan menuju *file .env*. Kode 4.1 menunjukan isi dari *file .env* yang dipindahkan dari direktori *application/config.php*:

14 Kode 4.1: Kode *application/config/config.php* yang dipindahkan menuju *.env*

15 `261 app.baseURL = 'http://sharif.localhost/'`

1 Kode 4.1 akan menentukan *url* dasar dari aplikasi menjadi `http://sharif.localhost/`. *URL*
 2 dibentuk spesifik berupa `http://sharif.localhost/` karena dijalankan pada server lokal sehingga
 3 tidak bertabrakan dengan server lokal lainnya. *URL* juga diganti karena tidak dapat diimple-
 4 mentasikan pada *CodeIgniter 4* sehingga URL ditentukan manual sesuai dengan *domain* yang
 5 dibentuk.

6 **app/Config/AutoLoad.php**

7 *File* ini tidak digunakan dan dibiarkan menggunakan konfigurasi *default* karena terdapat perubahan
 8 cara kerja kelas ini. Fitur ini pada *CodeIgniter 4* tidak dapat melakukan inisiasi otomatis terhadap
 9 *model*, *controller*, dan *library* saat aplikasi dijalankan. Namun kelas ini hanya akan mencari
 10 lokasi *file* sesuai dengan konfigurasinya saat kelas tersebut pertama kali diinisiasi. Kelas *helpers*
 11 yang diinisiasi dipindahkan menuju *BaseController* sedangkan kelas *model* dan *libraries* akan
 12 dipanggil menggunakan *PSR-4* pada *file-file* yang membutuhkan kelas tersebut.

13 **app/Config/Cache.php**

14 *File* ini tidak berubah dan akan tetap menggunakan konfigurasi *default* karena tidak terdapat perubahan
 15 pada *SharIF Judge* versi *CodeIgniter 3*.

16 **app/Config/Constants.php**

17 *File* ini berisikan seluruh data yang dipindahkan dari *application/config/constants.php*. Kode
 18 4.2 menunjukkan data yang dipindahkan menuju *Constant.php*.

Kode 4.2: Pemindahan kode pada *Constant*

```
19 define('SHJ_VERSION', '1.4');
20
21
22 /*Code editor related constants*/
23 define('EDITOR_FILE_NAME', "editor");
24 define('EDITOR_FILE_EXT', "txt");
25 define('EDITOR_IN_NAME', "exec_in");
26 define('EDITOR_OUT_NAME', "exec_out");
27 define('EDITOR_SUBMIT_ID', 0);
```

29 Kode 4.2 menunjukkan pemindahan dari *application/config/constants.php* menuju *Constant.php*.
 30 Sintaks SHJ_VERSION akan digunakan pada tampilan sidebar sedangkan sintaks EDITOR digunakan
 31 pada fungsi submit. Sintaks yang telah dipindahkan akan bersifat *global* sehingga dapat diakses
 32 oleh seluruh *file* pada *CodeIgniter 4*.

33 **app/Config/Cookie.php**

34 *File* ini tidak terdapat perubahan dan tetap menggunakan konfigurasi *default* karena tidak terdapat
 35 perubahan pada *SharIF Judge* versi *CodeIgniter 3*.

36 **app/Config/Database.php**

37 *File* ini dibiarkan menggunakan konfigurasi *default* karena akan dipindahkan menuju *file .env*. Kode
 38 4.3 menunjukkan pemindahan menuju *file .env* yang dipindahkan dari direktori *application/config.php*.

Kode 4.3: Pemindahan app/config/database.php menuju .env

```
1 database.default.hostname = db
2 database.default.database = sharif
3 database.default.username = root
4 database.default.password =
5 database.default.DBDriver = MySQLi
6 database.default.DBPrefix = shj_
```

Kode 4.3 menunjukkan pemindahan kode dari `database.php` menuju `.env`. Kode akan berisikan `hostname` yang digunakan, nama `database` yang digunakan, `username` dari `database`, `password`, `DBDriver`, dan `DBPrefix` yang digunakan.

12 app/Config>Email.php

¹³ File ini tidak digunakan dan dibiarkan menggunakan konfigurasi *default* karena konfigurasi *email* pada aplikasi *SharIF Judge* diletakan pada file *Secrets.php*.

File ini berisikan seluruh data yang dipindahkan dari `application/config/secrets.php`. Kode 4.4 menunjukan pemindahan konfigurasi `email`.

Kode 4.4: Pemindahan app/config/secrets.php menuju Email.php

```
17  
18 1    public string $protocol = 'smtp';  
19 2    public string $SMTPHost = 'ssl://smtp.mailgun.org';  
20 3    public string $mailType = 'html';
```

Kode 4.4 menunjukkan pemindahan konfigurasi *email* yang terdapat pada `secrets.php`. Kode akan berisikan *protocol* yang digunakan untuk mengirim *email*, *host* yang akan digunakan, dan tipe *email* yang dikirimkan. Seluruh data akan disimpan menuju variabel pada *CodeIgniter 4* dan terdapat perubahan nama variabel menjadi *camelCase*.

26 app/Config/Encryption.php

²⁷ File ini berisikan seluruh data yang dipindahkan dari `application/config/config.php`. Kode [4.5](#)
²⁸ menunjukkan pemindahan konfigurasi enkripsi yang akan digunakan.

Kode 4.5: Pemindahan app/config/config.php menuju Encryption.php

```
29     public string $key = 'bj42CAigTwLHUhz3p0vfRM0EJFeQD9uG';  
30 1
```

³² Kode 4.5 menunjukkan pemindahan *key* dari enkripsi yang akan digunakan. Data akan disimpan
³³ menuju variabel pada *CodeIgniter 4*.

34 app/Config/Filters.php

³⁵ File ini berisikan seluruh nama *filters* yang telah dibentuk. Kode 4.6 menunjukan pembangunan
³⁶ nama *filters* sesuai dengan kelas yang sudah dibentuk.

Kode 4.6: Penambahan nama *filters* untuk didefinisikan menuju *routes*

```
37     public array $aliases = [
38 1      'csrf'           => CSRF::class,
39 2      'toolbar'        => DebugToolbar::class,
40 3      'honeypot'       => Honeypot::class,
41 4      'invalidchars'  => InvalidChars::class,
42 5      'secureheaders'  => SecureHeaders::class,
43 6      'check-installandlogin' => CheckInstallAndLogin::class,
44 7      'check-login'    => CheckLogin::class,
45 8      'check-loginandlevelAdmin' => CheckLoginandLevelAdmin::class,
46 9      'check-loginandlevelHead'  => CheckLoginandLevelHead::class,
```

```

11     'check-loginandcli' => CheckLoginandCLI::class,
12     'check-loginandisajax' => CheckLoginandAjax::class,
13     'check-iscli' => CheckCLI::class,
14 ]

```

6 Kode 4.6 menunjukan penambahan nama *filters* untuk dipanggil menuju *routes*. Penamaan ini
7 akan ditambahkan pada *array \$aliases* dengan nama lain *filters* dan nama kelasnya.

8 app/Config/Routes.php

9 File ini berisikan seluruh pemindahan dan penambahan manual dari *routes* pada SharIF Judge.
10 Kode 4.7 menunjukan pembangunan *routes* aplikasi secara manual.

Kode 4.7: Penambahan *routes* yang digunakan pada aplikasi SharIF Judge

```

11     $routes->setDefaultNamespace('App\Controllers');
12     $routes->setDefaultController('Dashboard');
13     $routes->setDefaultMethod('index');
14     $routes->setTranslateURIDashes(false);
15     $routes->set404Override();
16     $routes->setAutoRoute(false);
17 // The Auto Routing (Legacy) is very dangerous. It is easy to create vulnerable apps
18 // where controller filters or CSRF protection are bypassed.
19 // If you don't want to define all routes, please use the Auto Routing (Improved).
20 // Set '$autoRoutesImproved' to true in 'app/Config/Feature.php' and set the following to true.
21 // $routes->setAutoRoute(false);
22
23 /*
24 *
25 * -----
26 * Route Definitions
27 * -----
28 */
29
30 // We get a performance increase by specifying the default
31 // route since we don't have to scan directories.
32 $routes->get('/', 'Dashboard::index', ['filter' => 'check-installandlogin:dual,noreturn']);
33 $routes->get('/dashboard', 'Dashboard::index', ['filter' => 'check-installandlogin:dual,noreturn']);
34 $routes->post('/dashboard/widget_positions', 'Dashboard::widget_positions');
35 // Authentication
36 $routes->add('/install', 'Install::index');
37 $routes->add('/login', 'Login::index');
38 $routes->post('login/register', 'Login::register');
39 $routes->get('login/lost', 'Login::lost');
40 $routes->post('login/lost', 'Login::lost');
41 $routes->get('/reset/:any', 'Login::reset/$1');
42 $routes->post('login/reset/:any', 'Login::reset/$1');
43 $routes->get('register', 'Login::register');
44 $routes->post('register', 'Login::register');
45 $routes->get('/settings', 'Settings::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
46 $routes->post('/settings', 'Settings::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
47 $routes->add('/profile/:num', 'Profile::index/$1', ['filter' => 'check-login:dual,noreturn']);
48 $routes->get('logout', 'Login::logout');
49 $routes->add('/settings/update', 'Settings::update');
50 // Users
51 $routes->get('/profile', 'Profile::index');
52 $routes->get('/users', 'Users::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
53 $routes->get('/users/add', 'Users::add', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
54 $routes->post('/users/add', 'Users::add', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
55 $routes->post('/users/delete_submissions', 'Users::delete_submissions', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
56 $routes->get('users/list_excel', 'Users::list_excel', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
57 // Notifications and scoreboard
58 $routes->get('/notifications', 'Notifications::index', ['filter' => 'check-login:dual,noreturn']);
59 $routes->get('/notifications/add', 'Notifications::add', ['filter' => 'check-login:dual,noreturn']);
60 $routes->post('/notifications/add', 'Notifications::add', ['filter' => 'check-login:dual,noreturn']);
61 $routes->add('/notifications/edit/:num', 'Notifications::edit/$1', ['filter' => 'check-login:dual,noreturn']);
62 $routes->post('/notifications/delete', 'Notifications::delete', ['filter' => 'check-login:dual,noreturn']);
63 $routes->post('/notifications/check', 'Notifications::check', ['filter' => 'check-login:dual,noreturn']);
64 $routes->get('scoreboard', 'Scoreboard::index', ['filter' => 'check-loginandcli:dual,noreturn']);
65 $routes->add('/server_time', 'Server_time::index', ['filter' => 'check-loginandisajax:dual,noreturn']);
66 // Logs, hof, moss, rejudge, queue
67 $routes->get('logs', 'Logs::index', ['filter' => 'check-login:dual,noreturn']);
68 $routes->get('halloffame', 'Halloffame::index', ['filter' => 'check-login:dual,noreturn']);
69 $routes->get('moss/:num', 'Moss::index/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);
70 $routes->post('moss/:num', 'Moss::index/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);
71 $routes->get('moss/update/:num', 'Moss::update/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);
72 $routes->post('moss/update/:num', 'Moss::update/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);

```

```

52 $routes->get('queue', 'Queue::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
53 $routes->post('queue', 'Queue::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
54 $routes->post('queue/resume', 'Queue::resume', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
55 $routes->post('queue/pause', 'Queue::pause', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
56 $routes->post('queue/empty_queue', 'Queue::empty_queue', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
57 $routes->cli('queueprocess/run', 'Queueprocess::run');
58 $routes->get('rejudge', 'Rejudge::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
59 $routes->post('rejudge', 'Rejudge::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
60 $routes->get('rejudge/:num', 'Rejudge::index/$1', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
61 $routes->post('rejudge/rejudge_single', 'Rejudge::rejudge_single', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
62 //Assignments
63 $routes->get('/assignments', 'Assignments::index', ['filter' => 'check-login:dual,noreturn']);
64 $routes->get('/assignments/add', 'Assignments::add', ['filter' => 'check-login:dual,noreturn']);
65 $routes->post('/assignments/add', 'Assignments::add', ['filter' => 'check-login:dual,noreturn']);
66 $routes->post('/assignments/select', 'Assignments::select', ['filter' => 'check-login:dual,noreturn']);
67 $routes->get('/assignments/edit/:num', 'Assignments::edit/$1', ['filter' => 'check-login:dual,noreturn']);
68 $routes->post('/assignments/edit/:num', 'Assignments::edit/$1', ['filter' => 'check-login:dual,noreturn']);
69 $routes->get('/assignments/delete/:num', 'Assignments::delete/$1', ['filter' => 'check-login:dual,noreturn']);
70 $routes->post('/assignments/delete/:num', 'Assignments::delete/$1', ['filter' => 'check-login:dual,noreturn']);
71 $routes->get('/assignments/pdf/:any', 'Assignments::pdf/$1/$2/$3', ['filter' => 'check-login:dual,noreturn']);
72 $routes->get('/assignments/pdf/:num', 'Assignments::pdf/$1', ['filter' => 'check-login:dual,noreturn']);
73 $routes->post('/assignments/pdfCheck/:num', 'Assignments::pdfCheck/$1');
74 $routes->get('/assignments/pdfCheck/:num', 'Assignments::pdfCheck/$1');
75 $routes->get('assignments/downloadtestsdesc/:num', 'Assignments::downloadtestsdesc/$1');
76 $routes->get('assignments/download_submissions/:any/:num', 'Assignments::download_submissions/$1/$2');
77 // Submit and submissions
78 $routes->get('submit', 'Submit::index', ['filter' => 'check-login:dual,noreturn']);
79 $routes->post('submit', 'Submit::index', ['filter' => 'check-login:dual,noreturn']);
80 $routes->get('submit/load/:num', 'Submit::load/$1', ['filter' => 'check-login:dual,noreturn']);
81 $routes->post('submit/save/:any', 'Submit::save/$1');
82 $routes->post('submit/save', 'Submit::save');
83 $routes->get('submit/get_output/:num', 'Submit::get_output/$1');
84 $routes->get('submissions/all/user/:any', 'Submissions::all');
85 $routes->get('submissions/final', 'Submissions::the_final', ['filter' => 'check-login:dual,noreturn']);
86 $routes->get('submissions/final/:any', 'Submissions::the_final/$1', ['filter' => 'check-login:dual,noreturn']);
87 $routes->post('submissions/view_code', 'Submissions::view_code');
88 $routes->post('submissions/select', 'Submissions::select');
89 $routes->get('/submissions/all_excel', 'Submissions::all_excel');
90 $routes->get('/submissions/final_excel', 'Submissions::final_excel');
91 $routes->get('submissions/download_file/:any/:num/:num', 'Submissions::download_file/$1/$2/$3/$4');
92 $routes->get('submissions/all', 'Submissions::all');
93 // Problems
94 $routes->get('/problems', 'Problems::index', ['filter' => 'check-login:dual,noreturn']);
95 $routes->get('/problems/:num', 'Problems::index/$1');
96 $routes->get('/problems/:num/:num', 'Problems::index/$1/$2');
97 $routes->get('/problems/edit/:any/:num/:num', 'Problems::edit/$1/$2/$3');
98 $routes->post('/problems/edit/:any/:num/:num', 'Problems::edit/$1/$2/$3');

```

- 49 Kode 4.7 menunjukkan seluruh *routes* yang didefinisikan secara explisit sesuai dengan kegunaannya.
50 Terdapat juga penambahan *filters* untuk melakukan pengecekan sebelum *controller* diinisiasi.
51 Pembangunan *routes* secara manual dilakukan karena alasan keamanan dan juga konfigurasi *filters*
52 yang berbeda untuk setiap *controller*.

53 app/Config/Secrets.php

- 54 File ini merupakan file yang ditambahkan diluar dari struktur standart dari aplikasi *CodeIgniter 4*.
55 File ini berisikan seluruh kebutuhan konfigurasi yang tidak terdapat pada *config* lainnya. Kode 4.8
56 menunjukan pemindahan file *Secrets.php* menuju *app/Config*.

Kode 4.8: Penambahan file *secrets*

```

57 namespace Config;
58 use CodeIgniter\Config\BaseConfig;
59 class Secrets extends BaseConfig
60 {
61     public $shj_authenticate = 'radius';
62     public $shj_radius = [
63         "server" => "localhost",
64         "secret" => "i-have-no-secret"
65     ];
66 }

```

```

13 // LDAP Settings refers to
14 // @link https://adldap2.github.io/Adldap2/#/setup?id=options
15 public $shj_ldap = [
16     "hosts" => ["dc3.ftis.unpar"],
17     "base_dn" => "DC=FTIS,DC=UNPAR",
18     "username" => "",
19     "password" => "",
20     "account_suffix" => "@ftis.unpar"
21 ];
22
23 public $shj_mail = [
24     'protocol' => 'smtp',
25     'SMTPHost' => 'ssl://smtp.mailgun.org',
26     'SMTPPort' => 465,
27     'SMTPUser' => '',
28     'SMTPPass' => '',
29     'mailType' => 'html',
30     'charset' => 'utf-8'
31 ];
32

```

Kode 4.8 menunjukan pemindahan kode menuju app/Config. Terdapat penghapusan sintaks `defined('BASEPATH') OR exit('No direct script access allowed');` pada baris pertama kalimat. Sintaks yang telah dihapus akan digantikan oleh sintaks *class* seperti pada baris pertama dan nama kelas akan *extend BaseConfig*. Terdapat juga penambahan sintaks *namespace* dan *use*. File ini akan digunakan dan dipanggil pada *controller* untuk melakukan autentikasi pengguna.

app/Config/Security.php

File ini berisikan seluruh konfigurasi yang dipindahkan dari application/config/config.php. Kode 4.9 menunjukan pemindahan data token *csrf* dan *cookie* menuju Security.php.

Kode 4.9: Pemindahan file config menuju Security.php

```

31 public string $tokenName = 'shj_csrf_token';
32 public string $cookieName = 'shjcsrftoken';

```

Kode 4.9 menunjukan pemindahan sintaks menuju Security.php. Sintaks pada baris pertama merupakan nama *token* yang akan digunakan sedangkan sintaks kedua merupakan nama *cookie* yang akan digunakan. Sintaks pada CodeIgniter 4 tidak akan menggunakan *array* namun akan disimpan menuju variabel. Sintaks lainnya tidak akan diubah karena akan menggunakan konfigurasi *default* dari CodeIgniter 4.

app/Config/Session.php

File ini berisikan seluruh konfigurasi yang dipindahkan dari application/config/config.php. Kode 4.10 menunjukan pemindahan data konfigurasi *session* menuju Session.php.

Kode 4.10: Pemindahan file config menuju Session.php

```

43 public string $driver = 'CodeIgniter\Session\Handlers\DatabaseHandler';
44 public string $cookieName = 'shjsession';

```

Data yang dipindahkan berupa *cookieName* yang selanjutnya disimpan menuju sebuah variabel. Selanjutnya terdapat perubahan *driver* dimana akan menggunakan *database handler* yang menyimpan data menuju *database*.

1 app/Config/Validation.php

2 File ini berisikan aturan yang dibentuk secara manual oleh pengguna untuk melakukan validasi
3 sebelum *request* diproses oleh aplikasi. Kode 4.11 menunjukkan aturan validasi data yang dibentuk
4 secara manual oleh pengguna.

Kode 4.11: Perancangan aturan yang dibentuk secara manual pada file Validation.php

```
6 1 class MyRules
7 2 {
8 3     public function password_check($str): bool
9 4     {
10 5         if ($strlen($str) == 0 OR ($strlen($str) >= 6 && $strlen($str) <= 200))
11 6             return TRUE;
12 7         return FALSE;
13 8     }
14 9
15 0     public function password_again_check($str) :bool
16 1     {
17 2         $request = \Config\Services::request();
18 3         if ($request->getPost('password') !== $request->getPost('password_again'))
19 4             return FALSE;
20 5         return TRUE;
21 6     }
22 7
23 8     public function role_check($role)
24 9     {
25 0         $user = new User();
26 1         if ($user->level <= 2)
27 2             return ($role == '');
28 3
29 4         // Admins can change everybody's user role:
30 5         $roles = array('admin', 'head_instructor', 'instructor', 'student');
31 6         return in_array($role, $roles);
32 7     }
33 8
34 9     /**
35 0      * Checks whether a user with this email exists
36 1      */
37 2     public function email_check($email,$edit_username):bool
38 3     {
39 4         $edit_username = explode(',', $edit_username);
40 5         $user_model = new UserModel();
41 6         if ($user_model->have_email($email, $edit_username))
42 7             return FALSE;
43 8         return TRUE;
44 9     }
45 0
46 1     /**
47 2      * checks whether the entered registration code is correct or not
48 3      */
49 4      */
50 5     public function _registration_code($code){
51 6         $settings_model = new SettingsModel();
52 7         $rc = $settings_model->get_setting('registration_code');
53 8         if ($rc == '0')
54 9             return TRUE;
55 0         if ($rc == $code)
56 1             return TRUE;
57 2         return FALSE;
58 3     }
59 4
60 5     /**
61 6      * Required
62 7      */
63 8      *
64 9      * @param string
65 0      * @return bool
66 1      */
67 2     public function required($str)
68 3     {
69 4         return is_array($str) ? (bool) count($str) : ($str !== '');
70 5     }
71 6
72 7     // -----
73 8
74 9
75 0     /**
76 1      * Is Lowercase
77 2
```

```

72  *
73  * @param $str
74  * @return bool
75  */
76  public function lowercase($str)
77  {
78      return (strtolower($str) === $str);
79  }
80
81  // -----
82
83
84  public function _check_language($str)
85  {
86      if ($str=='0')
87          return FALSE;
88      if (in_array( strtolower($str),array('c', 'c++', 'python_2', 'python_3', 'java', 'zip', 'pdf', 'txt')))
89          return TRUE;
90      return FALSE;
91  }
92
93
94  // -----
95  // Used in Submissions.php
96
97  public function _check_type($type)
98  {
99      return ($type === 'code' || $type === 'result' || $type === 'log');
100 }
101 }
```

32 Kode 4.11 menunjukan pembangunan aturan secara manual yang dipindahkan dari *controller*
33 *Login.php*, *Profile.php*, *Submissions.php*, dan *Submit.php*. Aturan yang dibentuk secara
34 manual dapat digunakan seperti aturan lainnya yang tersedia pada *CodeIgniter 4*. Aturan yang
35 dibentuk terdapat beberapa penambahan seperti pemanggilan *library*, *model*, dan penambahan
36 sintaks untuk memecah *array*.

37 4.2.2 Controllers

38 *Controller* terdapat perubahan pada bagian fungsi *__construct()* dimana sekarang tidak dapat
39 mengembalikan sesuatu. Oleh karena itu, akan dibentuk beberapa *filters* 4.2.1 untuk melakukan
40 pengecekan terhadap fungsi yang sebelumnya terdapat pada *__construct()*.

41 app/Controllers

42 Direktori ini berisikan seluruh *controller* yang dipindahkan dari *application/controllers*. Beri-
43 kut merupakan isi pada direktori ini:

- 44 • *Assignments.php*
- 45 • *BaseController.php*
- 46 • *Dashboard.php*
- 47 • *Halloffame.php*
- 48 • *Install.php*
- 49 • *Login.php*
- 50 • *Logs.php*
- 51 • *Moss.php*
- 52 • *Notifications.php*
- 53 • *Problems.php*
- 54 • *Profile.php*

- 1 • Queue.php
- 2 • Queueprocess.php
- 3 • Rejudge.php
- 4 • Scoreboard.php
- 5 • Server_time.php
- 6 • Settings.php
- 7 • Submissions.php
- 8 • Submit.php
- 9 • Users.php

10 Controllers terdapat perubahan dan penambahan baik dalam kelas yang dilakukan *extends*
 11 maupun dalam pemanggilan kelas lain seperti *model* dan *helpers*. Beberapa *helpers* yang dipanggil
 12 melalui *autoload* dipindahkan menuju *BaseController*. Kode 4.12 menunjukan perubahan yang
 13 terdapat pada *controller Logs.php*.

Kode 4.12: Perubahan kode *controller Logs.php* pada *CodeIgniter 4*

```

14 namespace App\Controllers;
15 1
16 2
17 3 use App\Controllers\BaseController;
18 4 use App\Models\AssignmentModel;
19 5 use App\Models\LogsModel;
20 6 use App\Models\User;
21 7
22 8 class Logs extends BaseController
23 9 {
24 10     protected $session;
25 11     protected $user;
26 12     protected $logs_model;
27 13     protected $assignment_model;
28 14
29 15     public function __construct()
30 16     {
31 17         $this->session = session();
32 18         $this->logs_model = new LogsModel();
33 19         $this->assignment_model = new AssignmentModel();
34 20         $this->user = new User();
35 21         if ( $this->user->level <= 2 ) // permission denied
36 22             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
37 23     }
38 24
39 25     public function index()
40 26     {
41 27
42 28         $data = array(
43 29             'logs' => $this->logs_model->get_all_logs(),
44 30             'selected' => 'logs',
45 31             'user' => $this->user,
46 32             'all_assignments' => $this->assignment_model->all_assignments(),
47 33             'finish_time' => $this->user->selected_assignment['finish_time'],
48 34             'extra_time' => $this->user->selected_assignment['extra_time'],
49 35         );
50 36
51 37         return view('pages/admin/logs', $data);
52 38     }
53 39 }
```

55 Kode 4.12 terdapat perubahan dimana sekarang akan *extends BaseController*. Terdapat juga
 56 penghapusan sintaks `defined('BASEPATH') OR exit('No direct script access allowed');`.
 57 Terdapat penambahan sintaks *namespace* dan juga beberapa sintaks untuk memanggil *models*.
 58 *Controller* juga memiliki perubahan dalam mengembalikan *view* dimana sekarang menggunakan
 59 sintaks `return view`. Selain itu terdapat penambahan pada *BaseController.php* untuk melakukan
 60 inisiasi terhadap *helpers* dan juga beberapa *library* yang akan digunakan. Kode 4.13 menunjukan
 61 penambahan *helpers* yang dapat diakses oleh seluruh *controller*

Kode 4.13: Penambahan sintaks pada BaseController

```

1 protected $helpers = ['text','url','shj_helper','form','cookie','string','filesystem'];
2 /**
3  * Be sure to declare properties for any property fetch you initialized.
4  * The creation of dynamic property is deprecated in PHP 8.2.
5  */
6 protected $parsedown;
7 protected $twig;
8
9 /**
10 * Constructor.
11 */
12 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
13 {
14     // Do Not Edit This Line
15     parent::initController($request, $response, $logger);
16
17     // Preload any models, libraries, etc, here.
18     $this->parsedown = new Parsedown();
19     $this->twig = new Twig();
20
21 }

```

Kode 4.13 menunjukkan pemindahan dan penambahan *helpers* dari *autoload* menuju *BaseController*. *Helpers* akan ditambahkan menuju *array \$helpers* sedangkan beberapa *thirdparty library* akan dilakukan inisiasi di dalam fungsi *initController*. Seluruh *helpers* yang telah dipindahkan dapat diakses pada seluruh *controller*. Penambahan ini dilakukan untuk mempermudah dalam penggunaan *helpers* pada seluruh *controller* agar tidak perlu diinisiasi lagi pada setiap *controller* yang membutuhkan.

30 IncomingRequest

31 *Input* digantikan oleh fungsi *request* dengan perubahan cara pemanggilan sintaks. *Validation* hanya
32 akan diinisiasikan pada fungsi *__construct* setiap *model* yang membutuhkan. Fungsi tidak akan
33 diinisiasikan pada *controller* karena *CodeIgniter 4* sudah menyediakan fitur ini pada *controller*
34 sehingga hanya perlu dilakukan pemanggilan. Kode 4.14 menunjukkan inisiasi *request* pada *model*.

Kode 4.14: Perancangan inisiasi *request* pada *__construct*

```

35
36 protected $request;
37
38
39 public function __construct()
40 {
41     $this->request = \Config\Services::request();
42 }

```

44 Kode 4.14 menunjukkan inisiasi yang dilakukan pada *model*. *Controller* tidak perlu melakukan
45 inisiasi terhadap fungsi ini karena sudah terinisiasi pada *BaseController*. Pengambilan *input* yang
46 dilakukan oleh pengguna dilakukan menggunakan sintaks pada kode 4.15.

Kode 4.15: Perancangan penggunaan *request*

```

47 if($this->request->isAJAX())
48 $this->request->getPost('assignment_select')

```

51 Kode 4.15 menunjukkan cara penggunaan *request* untuk melakukan pengecekan dan pengambilan
52 data. Sintaks berubah dari yang sebelumnya menggunakan *snakecase* menjadi *camelCase*.

53 4.2.3 Filters

54 Pada *CodeIgniter 4* *__construct()* tidak dapat mengembalikan sesuatu oleh karena itu akan
55 dibentuk beberapa *filters* untuk melakukan pengecekan. Beberapa *filters* yang dibentuk antara lain

berfungsi untuk mengecek apakah dijalankan dari *command line interface*, apakah sudah *install* dan *login*, apakah sudah *login*, apakah sudah *login* dan dijalankan dari *command line interface*, apakah sudah *login* dan apakah *request* berupa *AJAX*, dan apakah sudah *login* dan pengecekan terhadap *role* pengguna. Terdapat total tujuh buah *filters* yang akan dibentuk. Kode 4.16 menunjukan sintaks untuk melakukan pengecekan apakah dijalankan dari *command line interface*.

Kode 4.16: Perancangan kode pada *Filters CheckCLI.php*

```

6  public function before(RequestInterface $request, $arguments = null)
7  {
8      $request = \Config\Services::request();
9
10     if ($request->isCLI())
11         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
12
13 }

```

Kode 4.16 menunjukan pemindahan kode dari *__construct* menuju *filters CheckCLI*. *__construct* terdapat pada *controller* dimana fungsi diatas digunakan pada *controller Queueprocessphp*. Kode ini akan mengecek apakah *request* yang diberikan oleh pengguna melalui *command line interface*. Apabila *request* yang diberikan bukan melalui *command line interface* maka dikembalikan error berupa halaman tidak ditemukan. Kode 4.17 menunjukan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan *install* dan *login*.

Kode 4.17: Perancangan kode pada *Filters CheckInstallAndLogin.php*

```

21  public function before(RequestInterface $request, $arguments = null)
22  {
23      $db = Database::connect();
24
25      if ( !$db->tableExists('sessions'))
26          return redirect()->to('/install');
27
28      $session = \Config\Services::session();
29      if ( !$session->get('logged_in')) // if not logged in
30          return redirect()->to('/login');
31
32 }

```

Kode 4.17 menunjukan pemindahan dari *__construct* pada *controller Dashboard*. Fungsi pada *filters* ini akan melakukan pengecekan apakah sudah terdapat tabel *session* dan apakah pengguna sudah melakukan *login*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.18 menunjukan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan *login*.

Kode 4.18: Perancangan kode pada *Filters CheckLogin.php*

```

39  public function before(RequestInterface $request, $arguments = null)
40  {
41      $session = \Config\Services::session();
42
43      if ( !$session->get('logged_in')) // if not logged in
44          return redirect()->to('Login');
45
46 }

```

Kode 4.18 menunjukan pemindahan dari *__construct* pada *controller Profile, Notifications, Logs, Assignments, Submit, Submission, dan Problems*. Fungsi pada *filters* ini akan melakukan pengecekan apakah pengguna sudah melakukan *login*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang sesuai. Kode 4.19 menunjukan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna menjalankan *request* dari *command line interface* dan apakah pengguna sudah melakukan *login*.

Kode 4.19: Perancangan kode pada *Filters CheckLoginandCLI.php*

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $session = \Config\Services::session();
4     $request = \Config\Services::request();
5
6     if ($request->isCLI())
7         return;
8     if ( !$session->get('logged_in') ) // if not logged in
9         return redirect()->to('login');
10 }
11

```

Kode 4.19 menunjukan pemindahan dari `__construct` pada *controller Scoreboard*. Fungsi pada *filters* ini akan melakukan pengecekan apakah *controller* dijalankan melalui *command line* dan apakah pengguna sudah *login*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.20 menunjukan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah *request* yang diberikan pengguna berupa *ajax*.

Kode 4.20: Pemindahan kode pada *Filters CheckLoginandAjax.php*

```

19 public function before(RequestInterface $request, $arguments = null)
20 {
21     $session = \Config\Services::session();
22     $request = \Config\Services::request();
23
24     if ( !$session->get('logged_in') ) // if not logged in
25         return redirect()->to('/login');
26     if ( !$request->isAJAX() )
27         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
28 }
29
30

```

Kode 4.20 menunjukan pemindahan dari `__construct` pada *controller Server_time*. Fungsi pada *filters* ini akan melakukan pengecekan apakah pengguna sudah *login* dan apakah *request* berupa *ajax*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.21 menunjukan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah *role* dari pengguna dapat mengakses situs tersebut.

Kode 4.21: Pemindahan kode pada *Filters CheckLoginandLevelAdmin.php*

```

37 public function before(RequestInterface $request, $arguments = null)
38 {
39     $session = \Config\Services::session();
40     $user = new User();
41
42     if ( !$session->get('logged_in') ) // if not logged in
43         return redirect()->to('/login');
44     if ( $user->level <= 2 ) // permission denied
45         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
46 }
47
48

```

Kode 4.21 menunjukan pemindahan dari `__construct` pada *controller Server_time*. Fungsi pada *filters* ini akan melakukan pengecekan terhadap *level* pengguna apakah *admin* dan apakah pengguna sudah melakukan *login*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.22 menunjukan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah *role* dari pengguna dapat mengakses situs tersebut.

Kode 4.22: Pemindahan kode pada *Filters CheckLoginandLevelHead.php*

```

55 public function before(RequestInterface $request, $arguments = null)
56 {
57

```

```

13     $session = \Config\Services::session();
14     $user = new User();
15
16     if ( !$session->get('logged_in')) // if not logged in
17         return redirect()->to('/Login');
18     if ( $user->level <= 1) // permission denied
19         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
20 }
```

10 Kode 4.21 menunjukan pemindahan dari `__construct` pada `controller Server_time`. Fungsi
 11 pada `filters` ini akan melakukan pengecekan `level` pengguna dan apakah pengguna sudah melakukan
 12 `login`. Selanjutnya `filter` ini akan mengembalikan pengguna kepada `routes` yang sesuai apabila
 13 tidak sesuai dengan kondisinya. Setelah dibentuk `filters`, selanjutnya akan ditambahkan menuju
 14 `file Filters.php` untuk mendefiniskan nama yang selanjutnya akan dimasukkan menuju `routes`.
 15 Penambahan menuju `file Filters.php` dapat dilihat pada subbab 4.2.1. Setelah ditambahkan, `filters`
 16 akan dipanggil pada `routes` yang membutuhkan pengecekan. Kode 4.23 menunjukan penambahan
 17 `filters` pada `routes` sesuai dengan kebutuhannya.

Kode 4.23: Penambahan `filter` pada `routes`

```

18 $routes->get('/settings','Settings::index',['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
```

21 Kode 4.23 menambahkan `filter` yang sudah dibentuk dan dinamakan setelah penulisan nama
 22 `controller` dan fungsinya.

23 4.2.4 *Helpers*

24 Direktori ini berisikan `helpers` yang dibentuk secara manual oleh pengguna bernama `shj_helper`.
 25 `Helpers` terdapat perubahan dan penghapusan sintaks. Sintaks `defined('BASEPATH')` OR `exit('No`
 26 `direct script access allowed');` akan dihapus dan akan ditambahkan beberapa `model` yang
 27 digunakan pada `helper` ini. `Model` yang ditambahkan berupa `settings_model`.

28 4.2.5 *Libraries*

29 `Libraries` terdapat beberapa perubahan dan penghapusan fungsi sehingga akan digantikan. Berikut
 30 merupakan perancangan perubahan fungsi pada `CodeIgniter 4`.

31 *Error Handling*

32 `Error Handling` pada `CodeIgniter 4` terdapat perubahan sintaks dan cara pemanggilan sehingga
 33 akan digantikan dengan sintaks yang baru. Berikut merupakan pemanggilan `error handling`:

Kode 4.24: Perubahan sintaks `error handling`

```

34 throw new \Exception('SharIF_Judge_is_already_installed.');
```

37 Kode 4.24 menunjukan perubahan sintaks untuk memberikan `error` dengan pesan yang ditentukan.
 38 Selain itu, terdapat perubahan dalam memberikan `error` berupa 404. Kode 4.25 menunjukan sintaks
 39 untuk memberikan `error` berupa 404 halaman tidak ditemukan.

Kode 4.25: Perubahan sintaks `error handling 404`

```

40 throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_finished.');
```

- 1 Kode 4.25 akan mengembalikan *error* berupa halaman tidak ditemukan dengan pesan **Selected**
 2 **assignment has finished.** dalam *environment development* dan *testing*. Pada *environment*
 3 *production* tidak ditampilkan pesan *error* namun hanya akan ditampilkan halaman tidak ditemukan.

4 *Emails*

- 5 *Emails* pada *CodeIgniter 4* terdapat perubahan sintaks dan cara pemanggilan sehingga akan dipin-
 6 dahkan sesuai dengan sintaks yang baru. Sintaks berubah dari yang sebelumnya menggunakan
 7 *snakecase* menjadi menggunakan *camelcase*. Kode 4.26 menunjukkan penggunaan *library email*.

Kode 4.26: Perubahan penggunaan sintaks pada *library emails*

```

8 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
10.2 $this->email->setTo($user[1]);
11.3 $this->email->setSubject('SharIF_Judge_Username_and_Password');
12.4 $text = $this->settings_model->get_setting('add_user_mail');
13.5 $text = str_replace('{SITE_URL}', base_url(), $text);
14.6 $text = str_replace('{ROLE}', $user[4], $text);
15.7 $text = str_replace('{USERNAME}', $user[0], $text);
16.8 $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
17.9 $text = str_replace('{LOGIN_URL}', base_url(), $text);
18.0 $this->email->setMessage($text);
38.1 $this->email->send()

```

- 21 Kode 4.26 memiliki sintaks dengan nama sama namun terdapat perubahan menjadi *camelcase*.

22 *Working with Uploaded Files*

- 23 *Working with uploaded files* terdapat perubahan pada beberapa sintaks dan validasi terhadap *file*
 24 yang telah diunggah. Konversi aplikasi *SharIF Judge* akan menggunakan fungsi ini dengan beberapa
 25 perubahan sintaks sesuai dengan dokumentasinya. Kode 4.27 menunjukkan perubahan yang terdapat
 26 pada *library* ini.

Kode 4.27: Perancangan perubahan *library upload* pada *CodeIgniter 4*

```

27 $zip_uploaded = $this->request->getFile('tests_desc');
28.1 if ( $_FILES['tests_desc']['error'] === UPLOAD_ERR_NO_FILE ){
29.2     $this->messages[] = array(
30.3         'type' => 'notice',
31.4         'text' => "Notice:_You_did_not_upload_any_zip_file_for_tests._If_needed,_upload_by_editing_assignment."
32.5     );
33.6 }
34.7 }
35.8 elseif ( $zip_uploaded->getExtension() != "zip" ){
36.9     $this->messages[] = array(
37.0         'type' => 'error',
38.1         'text' => "Error:_Error_uploading_tests_zip_file:_The_filetype_you_are_attempting_to_upload_is_not_allowed."
39.2     );
40.3 }
41.4 else{
42.5     $zip_uploaded->move($assignments_root);
43.6     $this->messages[] = array(
44.7         'type' => 'success',
45.8         'text' => "Tests_(zip_file)_uploaded_successfully."
46.9     );
47.0 }

```

- 49 Kode 4.27 menunjukkan perubahan yang terdapat pada *library upload*. Pengambilan *file* digantikan
 50 dengan sintaks `$this->request->getFile('tests_desc')` dengan parameter berupa nama dari
 51 *tag form* yang sudah dibentuk. Selanjutnya dilakukan pengecekan terhadap *file* yang sudah di-
 52 unggah dan memberikan beberapa *error message* sesuai dengan kondisinya. *File* yang sudah di-
 53 unggah dipindahkan menuju direktori yang disimpan pada variabel `assignments_root`.

- 54 Konfigurasi aturan terhadap *upload* sudah tidak terdapat pada *CodeIgniter 4* sehingga akan
 55 diubah dan digantikan pada kondisi di baris kedua. Kondisi tersebut dibentuk untuk melakukan

- 1 pengecekan tipe *file* yang sudah diunggah dan memberikan *error*. *Error* yang dihasilkan juga
 2 sudah tidak dinamik karena tidak menggunakan *validation* sehingga disesuaikan dengan *error* yang
 3 terdapat pada *CodeIgniter 4*.

4 ***Working with URIs***

- 5 Kelas *URI* digantikan oleh fungsi *working with URIs* dengan perubahan dan penambahan fungsi.
 6 *Library* ini akan diinisiasi pada fungsi `__construct controller Submissions.php`. Kode 4.28
 7 menunjukkan inisiasi *library* ini dan terdapat penambahan sintaks karena terdapat fungsi yang
 8 dihapus.

Kode 4.28: Perancangan inisiasi dan penambahan sintaks *library URIs* pada `__construct`

```

9 $temp_inp = $this->uri->getSegments();
10 1 $input=[];
11 2 foreach($temp_inp as $key => $value){
12 3     if($key != ($count($temp_inp)-1) && ($key%2 != 0)){
13 4         $input[$value] = $temp_inp[$key+1];
14 5     }else{
15 6         break;
16 7     }
17 8 }
18 9 }
```

- 20 Kode 4.28 menunjukkan sintaks untuk mengambil *segment* yang terdapat pada *URI*. *Segment* ini akan
 21 disimpan dalam sebuah *array* dan tidak membentuk *associative array*. Oleh karena itu, terdapat
 22 penambahan sintaks `foreach` untuk mengubah *array* tersebut menjadi sebuah *associative array*.

23 ***Session***

- 24 Sintaks penggunaan *session* akan berubah dan *session* akan dilakukan inisiasi sebelum pemakaian.
 25 Kode 4.29 menunjukkan perubahan dalam melakukan inisiasi pada *session*.

Kode 4.29: Perancangan inisiasi *library session*

```

26 1
27 2 protected $session;
28 3
29 4 public function selected_assignment($username){
30 5     $this->session = session();
31 6 }
32 7 }
```

- 35 *Session* akan diinisiasi pada setiap *file* yang menggunakan *library* ini. Penggunaan *session*
 36 terdapat perubahan dari sintaks untuk pengambilan data, membersihkan data, dan penetapan data.
 37 Kode 4.30 menunjukkan perubahan dari sintaks untuk menggunakan *library session*.

Kode 4.30: Perancangan penggunaan *library session*

```

38 1 $this->session->get('logged_in')
39 2 $this->session->set($login_data);
40 3 $this->session->destroy();
```

- 43 Sintaks pada baris pertama berfungsi untuk mengambil data dengan nama *logged_in*. Sintaks pada
 44 baris kedua berfungsi untuk menetapkan sebuah data *session* pada variabel *login_data*. Terakhir
 45 pada baris ketiga merupakan sintaks untuk membersihkan data pada *session* pengguna tersebut.

1 Validation

2 *Form_validation* digantikan oleh fungsi *validation* dengan perubahan dan pengapusan beberapa
 3 fungsi. *Validation* akan diinisiasi pada fungsi *__construct* setiap *controller* dan *model* yang
 4 membutuhkan. Kode 4.31 menunjukkan inisiasi *validation* pada *controller*.

Kode 4.31: Perancangan inisiasi *validation* pada *__construct*

```
5
6 1
7 2 protected $validation;
8 3
9 4     public function __construct()
10 5     {
11 6         $this->validation = \Config\Services::validation();
12 7     }
```

14 Setiap *controller* yang menggunakan *validation* akan dideklarasikan sebuah variabel bernama
 15 *validation* agar dapat dipanggil pada seluruh fungsi yang membutuhkan pada kelas tersebut.
 16 *Validation* dilanjutkan dengan penetapan aturan untuk *tag form* yang diinginkan. Kode 4.32
 17 menunjukkan contoh penetapan aturan terhadap *input* yang akan masukan oleh pengguna dengan
 18 nama *username*.

Kode 4.32: Perancangan perubahan konfigurasi aturan pada *library validation*

```
19
20 1 $this->validation->setRule('username', 'username', 'required|min\_length[3]|max\_length[20]');
```

22 Sintaks untuk melakukan penetapan aturan berubah menjadi *setRule*. Beberapa aturan yang
 23 dibentuk secara manual akan dipindahkan menuju *file Validation.php* dan terdapat penghapusan
 24 aturan yang sudah tidak terdapat pada *CodeIgniter 4*. Pembangunan aturan dapat dilihat pada sub
 25 bab 4.2.1 kode 4.11. Aturan yang sudah dibentuk dapat digunakan seperti aturan lainnya dengan
 26 cara menulis nama kelasnya. Setelah aturan ditetapkan, *validation* akan dieksekusi berdasarkan
 27 *request* dari pengguna dan dilakukan validasi. Kode 4.33 menunjukkan perubahan penggunaan
 28 *validation* terhadap data yang sudah diberikan oleh pengguna.

Kode 4.33: Perancangan perubahan penggunaan *validation* pada *CodeIgniter 4*

```
29
30 1 if ($this->validation->withRequest($this->request)->run())
31 2     {
32 3         if ( !$this->request->isAJAX() ){
33 4             exit;
34 5         }else{
35 6             list($ok, $error) = $this->user_model->add_users(
36 7                 $this->request->getPost('new_users'),
37 8                 $this->request->getPost('send_mail'),
38 9                 $this->request->getPost('delay')
39 0             );
40 1             return view('pages/admin/add_user_result', array('ok' => $ok, 'error' => $error));
41 2         }
42 3     }
43 4 }
```

44 Kode 4.33 menjalankan *validation* berdasarkan *request* dari pengguna. *Validation* akan tetap
 45 menggunakan sintaks *run()* namun akan ada penambahan sintaks *withRequest* dimana validasi
 46 akan dijalankan setiap ada HTTP *Request* dari pengguna. Namun, *CodeIgniter 4* tidak me-
 47 nyediakan fungsi *form_error* sehingga akan diubah dengan menggunakan fungsi baru bernama
 48 *validation_errors()*. Fungsi tersebut dapat digunakan untuk mengembalikan *error* apabila
 49 terdapat data yang tidak sesuai dengan aturan. *Error* tersebut dapat ditampilkan pada halaman
 50 *view* menggunakan sintaks berikut:

```
51 <?= $validation->hasError('username') ? $validation->getError('username') : '' ?>
```

1 Sintaks diatas akan melakukan pengecekan apakah terdapat *error* dari *form* bernama *username*
 2 dan apabila terdapat maka akan dikembalikan dara *error* yang berasal dari *validation*. Variabel
 3 *validation* akan dikirimkan dari *controller* berisikan *library validation*.

4 Zip Archive

5 *Zip Encoding* akan digantikan dengan fungsi PHP *zip archive* karena sudah tidak tersedia pada
 6 *CodeIgniter 4*. Fungsi *zip archive* terdapat beberapa perbedaan sehingga akan disesuaikan dengan
 7 fungsi-fungsi yang ada. Kode 4.34 menunjukan perubahan yang terdapat pada *zip encoding* menjadi
 8 *zip archive*.

Kode 4.34: Perancangan perubahan *zip encoding* menjadi *zip archive*

```

9 $this->zip = new \ZipArchive();
10 1 $this->zip->open($zipname, ZipArchive::CREATE);
11 2     for ($i=1 ; $i<=$number_of_problems ; $i++)
12 3     {
13 4
14 5
15 6         $path = "$root_path/p{$i}/in";
16 7         $options = ['add_path' => "p{$i}/in/", 'remove_all_path' => TRUE];
17 8         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
18 9
19 0         $path = "$root_path/p{$i}/out";
20 1         $options = ['add_path' => "p{$i}/out/", 'remove_all_path' => TRUE];
21 2         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
22 3
23 4         $path = "$root_path/p{$i}/tester.cpp";
24 5         if (file_exists($path))
25 6             $this->zip->addFile($path,"p{$i}/tester.cpp");
26 7
27 8         $pdf_files = glob("$root_path/p{$i}/*.pdf");
28 9         if ($pdf_files)
29 0         {
30 1             $path = $pdf_files[0];
31 2             $this->zip->addFile($path,"p{$i}/".shj_basename($path));
32 3
33 4
34 5         $path = "$root_path/p{$i}/desc.html";
35 6         if (file_exists($path))
36 7             $this->zip->addFile($path,"p{$i}/desc.html");
37 8
38 9         $path = "$root_path/p{$i}/desc.md";
39 0         if (file_exists($path))
40 1             $this->zip->addFile($path,"p{$i}/desc.md");
41 2
42 3
43 4         $pdf_files = glob("$root_path/*.pdf");
44 5         if ($pdf_files)
45 6         {
46 7             $path = $pdf_files[0];
47 8             $this->zip->addFile($path,shj_basename($path));
48 9         }
49 0         $this->zip->close();
50 1
51 2         header('Content-Type:application/zip');
52 3         header('Content-disposition:attachment;filename=' . $zipname);
53 4         header('Content-Length:' . filesize($zipname));
54 5         readfile($zipname);
55 6

```

56 *Zip archive* akan dilakukan inisiasi pada variabel *zip*. Selanjutnya akan dibuka *file zip* meng-
 57 gunakan sintaks *open* yang menerima dua buah parameter. Parameter pertama berisikan nama
 58 *zip file* yang ingin dibentuk sedangkan parameter kedua berisikan mode *zip* yang diinginkan. Sin-
 59 taks *addGlob* akan memasukan seluruh *file* sesuai dengan pattern yang telah ditentukan. Sintaks
 60 *addFile* akan memasukan *file* sesuai dengan *path* yang telah ditentukan. Selanjutnya *zip* akan
 61 ditutup menggunakan sintaks *close* dan dapat diunduh oleh pengguna menggunakan fungsi *header*
 62 yang disediakan oleh PHP.

Selain untuk membentuk *file zip*, *zip archive* juga mendukung fungsi untuk melakukan *unzip* sehingga akan menggantikan *library unzip* yang sebelumnya dibentuk oleh Phil Sturgeon. Kode [4.35](#) menunjukkan penggunaan *zip archive* untuk *unzip* sebuah *file* pada *controller*.

Kode 4.35: Perancangan perubahan *unzip* menggunakan *zip archive* pada *controller*

```
4 $this->unzip = new ZipArchive();
5 // Create a temp directory
6 $tmp_dir_name = "shj_tmp_directory";
7 $tmp_dir = "$assignments_root/$tmp_dir_name";
8
9 // Extract new test cases and descriptions in temp directory
10 $this->unzip->open("$assignments_root/".$zip_uploaded->getName());
11 $extract_result = $this->unzip->extractTo($tmp_dir);
12
13 $this->messages[] = array(
14     'type' => 'error',
15     'text' => "_Zip_Extraction_Error:_".$this->unzip->getStatusString(),
16 );
17
```

Kode 4.35 akan melakukan inisiasi terhadap *zip archive*. Selanjutnya kode membentuk dua buah variabel berisikan nama direktori yang dituju. Sintaks `open` akan membuka sebuah *file zip* sesuai dengan nama *zip* yang ditentukan. Kode 4.35 selanjutnya akan melakukan *extract file zip* tersebut menuju direktori yang telah dibentuk dan menyimpannya pada sebuah variabel. Terakhir pengguna dapat mengambil status dari hasil *extract* yang telah dilakukan dan menyimpannya menuju sebuah variabel.

24 Password_hash

25 *Library* ini tidak akan digunakan dan akan digantikan oleh *password hash* yang disediakan oleh
26 PHP .*Library Password_hash* merekomendasikan pegguna untuk menggunakan fungsi *native* yang
27 disediakan oleh PHP apabila aplikasi mendukung PHP versi 5.5 ke atas. Sehingga, akan dilakukan
28 konversi menggunakan fungsi yang disediakan oleh PHP bernama `password_hash()`. Seluruh
29 penggunaan *library* ini akan diubah menggunakan fungsi yang disediakan oleh PHP dengan metode
30 *hashing* sama yaitu *CRYPT_BLOWFISH*. Perubahan fungsi *hashing* ini bersifat *backward compatible*
31 sehingga dapat menggunakan *database* aplikasi terdahulu tanpa perlu membentuk data baru. Berikut
32 merupakan contoh pengubahan kode dari *phpass* menjadi *password hash*.

33 'password' => \$this->password_hash->HashPassword(\$password)

34 menjadi

```
35     'password' => password_hash($password,PASSWORD_BCRYPT)
```

Sintaks `password_hash()` diatas menerima dua buah parameter yakni data yang ingin di enkripsi dan tipe enkripsi. Enkripsi akan menggunakan sintaks `PASSWORD_BCRYPT` yang menggunakan tipe `hash` berupa `CRYPT_BLOWFISH`. Selain itu, terdapat fungsi untuk melakukan pengecekan `password` yang sudah di enkripsi. Berikut merupakan contoh pengubahan kode untuk melakukan pengecekan `password` yang sudah di enkripsi.

41 password_verify(\$password, \$query->getRow()->password)

42 Sintaks diatas menerima dua buah parameter dengan parameter pertama berupa masukan dari
43 pengguna dan parameter berikutnya merupakan *hash* dari *password* yang sudah disimpan. Fungsi ini
44 akan mengembalikan data berupa *true* apabila *password* sama dan *false* apabila *password* berbeda.

1 *Library* yang terdapat pada *CodeIgniter 4* juga dapat diextend dan dibentuk sesuai dengan
 2 kebutuhan. Berikut merupakan *library* yang dibentuk oleh pengguna dan akan dipindahkan menuju
 3 app/Libraries.

4 ***Twig***

5 *Library* ini tidak akan digunakan untuk membentuk *view* pada *CodeIgniter 4* namun, akan ada
 6 penggunaan sebuah fungsi *Twig* yang akan dibentuk pada direktori app/Libraries. Fungsi tersebut
 7 bernama **extra_time_formatter** yang memiliki fungsi untuk mengubah input yang diberikan
 8 menjadi format jam dikali enam puluh menit. Kode 4.36 menunjukkan fungsi yang dibentuk pada
 9 direktori app/Libraries dengan nama *Twig.php*. Fungsi ini nantinya digunakan pada halaman
 10 *add_assignment.php*.

Kode 4.36: Perancangan perubahan *library MY_Form_validation* pada *CodeIgniter 4*

```
11 1 <?php
12 2 namespace App\libraries;
13 3
14 4 class Twig
15 5 {
16 6
17 7     /**
18 8      * Required
19 9      *
20 9
21 0      * @param  string
22 1      * @return bool
23 2      */
24 3 public function extra_time_formatter($extra_time)
25 4 {
26 5     // convert to minutes
27 6     $extra_time = floor($extra_time/60);
28 7     // convert to H*60
29 8     if ($extra_time % 60 == 0 )
30 9         $extra_time = ($extra_time/60) . '*60';
32 0     return $extra_time;
32 1 }
32 2 }
```

35 ***MY_Form_validation***

36 *Library MY_Form_validation* akan dipindahkan menuju direktori app/Libraries. *Library* ini
 37 akan digunakan kembali dengan perubahan *extends* menjadi menuju Validation, penghapusan
 38 sintaks **defined**, dan akan ada penambahan *namespace* pada baris awal file. Kode 3.22 menunjukkan
 39 contoh penambahan *namespace* dan penggantian *extends* pada *library* ini.

Kode 4.37: Contoh perubahan *library MY_Form_validation* pada *CodeIgniter 4*

```
40 1 namespace App\libraries;
42 2
43 3 use CodeIgniter\Validation\Validation;
44 4
45 5 class MY_Form_validation extends Validation
```

47 Kode 4.37 menghapus sintaks **defined** dan menggantikannya dengan penambahan *namespace*. Selain
 48 itu, kelas *library* akan *extends Validation*.

49 ***Parsedown***

50 *Library Parsedown* akan dipindahkan menuju direktori app/Libraries. *Library* ini akan digunakan
 51 kembali dengan penambahan *namespace* pada baris awal file dan penghapusan sintaks **defined**.
 52 Kode 4.38 menunjukkan contoh penambahan *namespace* dan juga penambahan sintaks *defined*.

Kode 4.38: Perancangan perubahan *library Parsedown* pada *CodeIgniter 4*

```

21 namespace App\Libraries;
22
23 class Parsedown

```

- 6 Kode 4.38 menghapus sintaks `defined` dan menggantikannya dengan penambahan *namespace*.
 7 Penggunaan *library* ini tidak akan berubah sehingga tidak terdapat perbedaan sintaks. Namun,
 8 terdapat perubahan cara inisiasi *library* ini dimana sekarang akan menggunakan sintaks `new` dan
 9 dilakukan inisiasi pada *BaseController*. Kode 4.39 menunjukkan perubahan cara melakukan inisiasi
 10 *library parsedown*.

Kode 4.39: Perancangan perubahan inisiasi *library Parsedown* pada *controller CodeIgniter 4*

```

11 protected $parsedown;
12
13 /**
14 * Constructor.
15 */
16
17 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
18 {
19     // Do Not Edit This Line
20     parent::initController($request, $response, $logger);
21
22     // Preload any models, libraries, etc, here.
23
24     // E.g.: $this->session = \Config\Services::session();
25     $this->parsedown = new Parsedown();
26 }

```

- 28 *Library parsedown* akan dilakukan inisiasi menuju variabel `parsedown` yang sudah dibentuk
 29 pada luar fungsi. Inisiasi dilakukan pada *BaseController* untuk mempermudah pemanggilan pada
 30 beberapa *controller*.

31 *Phpxcel*

- 32 *Library* ini akan digunakan kembali namun tidak akan dipindahkan menuju `app/Libraries`. *Library*
 33 akan dilakukan instalasi melalui *composer* dengan sintaks berikut:

34 composer require phpoffice/phpexcel

- 35 Sintaks diatas akan dijalankan pada akar dari aplikasi dan tidak terdapat perubahan terhadap
 36 penggunaan sintaks ini.

37 *Shj_pagination*

- 38 *Library* ini akan digunakan kembali dan dipindahkan menuju direktori `app/Libraries`. Selain itu,
 39 terdapat penambahan *namespace* pada baris awal *file* dan penghapusan sintaks `defined`. Kode 4.40
 40 menunjukkan penambahan perubahan dan penambahan sintaks pada *library Shj_pagination*.

Kode 4.40: Perancangan perubahan *library Shj_pagination* pada *CodeIgniter 4*

```

41 namespace App\Libraries;
42
43 class Shj_pagination
44 {

```

- 47 Selanjutnya *library* dapat dipanggil menggunakan sintaks berikut:

48 \$shj_pagination = new Shj_pagination(\$config);

- 49 Sintaks diatas akan memanggil *library* dan memasukan konfigurasi yang telah dibentuk pada
 50 variabel `config`.

1 4.2.6 *Models*

2 Direktori ini akan berisikan seluruh *model* yang dipindahkan dari *CodeIgniter 3*. Berikut merupakan
 3 isi pada direktori ini:

- 4 • AssignmentModel.php
- 5 • HofModel.php
- 6 • LogsModel.php
- 7 • NotificationsModel.php
- 8 • QueueModel.php
- 9 • ScoreboardModel.php
- 10 • SettingsModel.php
- 11 • SubmitModel.php
- 12 • User.php
- 13 • UserModel.php

14 Seluruh *Model* akan diganti penamaannya dari yang sebelumnya menggunakan *snakecase* menjadi
 15 *PascalCase*. Kode 4.41 menunjukkan perubahan sintaks yang terdapat pada *model*.

Kode 4.41: Perancangan perubahan *model* pada *CodeIgniter 4*

```

16 namespace App\Models;
17 1
18 2
19 3 use App\Libraries\Parsedown;
20 4 use CodeIgniter\Model;
21 5 use App\Models\SettingsModel;
22 6 use App\Models\ScoreboardModel;
23 7
24 8 class AssignmentModel extends Model
25 9 {
26 0   protected $settings_model;
27 1   protected $scoreboard_model;
28 2   protected $request;
29 3   protected $parsedown;
30 4
31 5   public function __construct()
32 6   {
33 7     parent::__construct();
34 8     $this->settings_model = new SettingsModel();
35 9     $this->request = \Config\Services::request();
36 0   }
37 1
38 2 /**
39 3 * Add New Assignment to DB / Edit Existing Assignment
40 4 *
41 5 * @param $id
42 6 * @param bool $edit
43 7 * @return bool
44 8 */
45 9 public function add_assignment($id, $edit = FALSE)
46 0 {
47 1   $this->scoreboard_model = new ScoreboardModel();

```

49 Terdapat penghapusan sintaks **defined** dan juga terdapat perubahan dalam *extends* dimana
 50 sekarang akan *extends Model*. Selain itu terdapat pemanggilan *model* lainnya menggunakan
 51 sintaks **new** yang disimpan melalui variabel yang sudah dibentuk. Terdapat juga perubahan cara
 52 pengambilan data menggunakan fungsi *database* yang ditunjukan pada Kode 4.42.

Kode 4.42: Perubahan sintaks pada *model*

```

53
54 1 public function all_assignments()
55 2 {
56 3   $result = $this->db->table('assignments')->orderBy('id')->get()->getResultArray();
57 4   $assignments = [];
58 5   foreach ($result as $item)
59 6   {

```

```

17     $assignments[$item['id']] = $item;
18 }
19 return $assignments;
20 }
```

6 Pengambilan data pada *CodeIgniter 4* perlu mendefinisikan nama tabel yang akan diambil
 7 datanya. Selanjutnya sintaks akan bersifat sama namun terdapat perubahan dari yang sebelumnya
 8 *snakecase* menjadi *camelCase*. *Model* yang telah dibentuk dapat dipanggil menuju *controller*
 9 menggunakan sintaks berikut:

```
10     $this->settings_model = new SettingsModel();
```

11 Sintaks diatas akan memanggil **SettingsModel** dan menyimpannya menuju variabel. Fungsi
 12 *database* juga dapat digunakan *file* lain dengan melakukan inisiasi dengan sintaks berikut:

```
13     $this->db = db_connect();
```

14 4.2.7 View

15 Direktori ini berisikan seluruh *view* yang dipindahkan dari *CodeIgniter 3*. Berikut merupakan isi
 16 dari direktori *view*:

17 errors

18 Direktori ini berisikan *file default* dari *CodeIgniter 4* karena tidak terdapat perubahan pada *SharIF*
 19 *Judge CodeIgniter 3*.

20 pages

- 21 • assignments.php
- 22 • dashboard.php
- 23 • halloffame.php
- 24 • notifications.php
- 25 • problems.php
- 26 • profile.php
- 27 • scoreboard_table.php
- 28 • scoreboard.php
- 29 • submissions.php
- 30 • submit.php
- 31 • Direktori admin:
 - 32 – add_assignment.php
 - 33 – add_notification.php
 - 34 – add_user_result.php
 - 35 – add_user.php
 - 36 – delete_assignment.php
 - 37 – edit_problem_html.php
 - 38 – edit_problem_md.php
 - 39 – edit_problem_plain.php

```

1      - install.php
2      - logs.php
3      - moss.php
4      - queue.php
5      - rejudge.php
6      - settings.php
7      - users.php
8 • Direktori authentication:
9      - login.php
10     - lost.php
11     - register_success.php
12     - register.php
13     - reset_password.php
14 • Direktori templates:
15     - base.php
16     - side_bar.php
17     - simple_header.php
18     - top_bar.php

```

19 *View* akan diubah menggunakan *extension .php* karena *twig* sudah tidak terintegrasi *CodeIgniter*
20 4. Seluruh *file view* akan diubah menjadi *extension .php* dari yang sebelumnya menggunakan *.twig*.
21 Seluruh *delimiters* juga akan diubah menggunakan fungsi pada *CodeIgniter 4*. Perubahan *view*
22 ditunjukan pada kode 4.43.

Kode 4.43: Perubahan *view* pada *Login.php*

```

23 <!-- {#
24 # SharIF Judge
25 # file: login.twig
26 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
27 #} -->
28 <!DOCTYPE html>
29 <html lang="en">
30 <?= $this->include('templates/simple_header')?>
31 <?= form_open() ?>
32 <div class="box_login">
33     <div class="judge_logo">
34         <a href="<?= site_url() ?>"></a>
35     </div>
36     <div class="login_form">
37         <div class="login1">
38             <p>
39                 <label for="form_username">Username</label><br/>
40                 <input id="form_username" type="text" name="username" required="required" pattern="[0-9a-zA-Z]{3,20}" title="The_
41                     Username_field_must_be_between_3_and_20_characters_in_length,_and_contain_only_digits_and_lowercase_
42                     letters" class="sharif_input" value="<?= set_value('username') ?>" autofocus="autofocus"/>
43                 <?= isset($this->errors['username']) ?>
44             </p>
45             <p>
46                 <label for="form_password">Password</label><br/>
47                 <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The_
48                     Password_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
49                 <?= isset($this->errors['password']) ?>
50             </p>
51             <?php if ($error): ?>
52                 <div class="shj_error">Incorrect username or password.</div>
53             <?php endif ?>
54         </div>
55         <div class="login2">
56             <p style="margin:0;">
57

```

```

B5      <?php if ($registration_enabled): ?>
26      <a href="=site_url('register')?&gt;"Register</a> |
37      <?php endif ?>
48      <a href="=site_url('login/lost')?&gt;"Reset Password</a>
59      <input type="submit" value="Login" id="sharif_submit"/>
60      </p>
71      </div>
82      </div>
93
104     </div>q
115     <?= form_close() ?>
126   </body>
127 </html>

```

15 Delimiters {{ }} digantikan menjadi <?= ?> sedangkan {%, %} digantikan menjadi <?php ?>. 16 Delimiters yang memanggil fungsi pada *CodeIgniter 4* akan digantikan menjadi <?= ?>. Perubahan 17 juga terdapat pada sintaks dari yang sebelumnya menggunakan **include** akan digantikan meng- 18 gunakan fungsi *CodeIgniter 4* berupa **\$this->include**. Selain terdapat perubahan *extension* dan 19 *delimiters*, terdapat juga penambahan kode pada *Controller* karena tidak mendukung pembangunan 20 variabel *global* pada *View*. Kode 4.44 menunjukkan contoh penambahan kode pada *Controller*.

Kode 4.44: Penambahan kode pada *controller Login.php*

```

21
22 $data = [
23   'error' => FALSE,
24   'registration_enabled' => $this->settings_model->get_setting('enable_registration'),
25   'title' => 'Login',
26   'validationError' => $this->validation
27 ];

```

29 Penambahan data terjadi karena halaman *view PHP* tidak dapat mendeklarasikan variabel 30 separa *global* sehingga data-data seperti *title* tidak dapat diakses oleh *view* lainnya.

31 4.2.8 public

32 assets

33 Direktori ini berisikan seluruh data yang dapat dilihat oleh pengguna seperti *javascript*, gambar, 34 dan lainnya. Berikut merupakan isi pada direktori ini:

- 35 • ace
- 36 • font
- 37 • fullcalendar
- 38 • gridster
- 39 • images
- 40 • js
- 41 • nano_scroller
- 42 • noty
- 43 • pdfjs
- 44 • reveal
- 45 • snippet
- 46 • styles
- 47 • tinymce

48 Direktori *assets* akan dipindah tanpa ada perubahan apapun.

1 4.2.9 restriced

- 2 Direktori ini berisikan direktori **tester** yang berfungsi untuk melakukan *sandbox* dan *shield*.
- 3 Direktori ini dipindahkan tanpa ada perubahan apapun.

1

BAB 5

2

IMPLEMENTASI DAN PENGUJIAN

3 Bab ini membahas mengenai implementasi dan pengujian konversi *SharIF Judge*.

4 **5.1 Lingkungan Implementasi dan Pengujian**

5 Implementasi perangkat lunak *SharIF Judge* dilakukan pada dua buah lingkungan. Lingkungan
6 pertama digunakan untuk membangun perangkat lunak sedangkan lingkungan kedua digunakan
7 untuk melakukan pengujian. Berikut merupakan spesifikasi lingkungan implementasi dan pengujian
8 yang digunakan:

9 1. Lingkungan *Development*

10 Tabel 5.1 menunjukkan spesifikasi perangkat keras lingkungan *development*.

Tabel 5.1: Perangkat Keras Lingkungan *Development*

| Parameter | Nilai |
|-----------------------------------|-------------------|
| Perangkat Keras | Macbook Pro M1 |
| <i>Processor</i> | <i>M1 Pro</i> |
| <i>Random Access Memory (RAM)</i> | 16 GB |
| <i>Storage</i> | 512 GB <i>SSD</i> |

11 Tabel 5.2 menunjukkan spesifikasi perangkat lunak lingkungan *development*.

Tabel 5.2: Perangkat Lunak Lingkungan *Development*

| Parameter | Nilai |
|---------------------------|--|
| Sistem Operasi | <i>Debian 11-slim</i> |
| Bahasa Pemrograman | <i>PHP, JavaScript, CSS, dan HTML</i> |
| <i>Framework</i> | <i>CodeIgniter 4.2.3</i> |
| <i>Code Editor</i> | <i>Visual Studio Code 1.84.2 (Universal)</i> |
| Perangkat Lunak Pendukung | <i>Docker Version 4.21.1 (114176)</i> <i>Composer 2.6.5</i> <i>Google Chrome Version 119.0.6045.159 (Official Build) (arm64)</i> <i>MariaDB 10.5.8</i> <i>phpMyAdmin 4.8</i> |

12 2. Lingkungan *Staging*

13 Tabel 5.3 menunjukkan spesifikasi perangkat keras lingkungan *staging*.

Tabel 5.3: Perangkat Keras Lingkungan *Staging*

| Parameter | Nilai |
|-----------------------------------|----------------------------|
| <i>Processor</i> | <i>Regular Intel 1vCPU</i> |
| <i>Random Access Memory (RAM)</i> | 512 MB |
| <i>Storage</i> | 10 GB <i>SSD</i> |

¹ Tabel 5.4 menunjukkan spesifikasi perangkat lunak lingkungan *staging*.

Tabel 5.4: Perangkat Lunak Lingkungan *Staging*

| Parameter | Nilai |
|---------------------------|--|
| Sistem Operasi | Ubuntu 20.04 |
| Perangkat Lunak Pendukung | <i>Apache Server 2.4.41</i> <i>Composer 2.6.3</i> <i>MariaDB 10.3.38</i> <i>PHP 8.1</i> |

² 5.2 Implementasi

³ Berikut merupakan hasil implementasi dari analisis dan perancangan perangkat lunak yang telah
⁴ dibentuk:

⁵ 5.2.1 Kode Program

⁶ Perubahan dan penambahan kode program untuk melakukan konversi perangkat lunak telah
⁷ dijabarkan pada bab 4. Seluruh perubahan dan penambahan dibentuk menggunakan bahasa
⁸ pemrograman PHP dengan bantuan beberapa perangkat lunak tambahan. Kode program untuk
⁹ *controller* dapat dilihat pada Lampiran A. Kode program untuk *model* dapat dilihat pada Lampiran
¹⁰ B. Kode program untuk *view* dapat dilihat pada Lampiran C. Kode program lainnya dapat dilihat
¹¹ pada Lampiran D.

¹² 5.2.2 Basis Data

¹³ Basis data terdapat perubahan pada tabel *sessions* dimana perubahan terjadi pada nama kolom dan
¹⁴ penghapusan kolom. Terdapat perubahan nama kolom *session_id* menjadi *id*, *user_data* menjadi
¹⁵ *data*, dan *last_activity* menjadi *timestamp*. Terdapat penghapusan kolom *user_agent*, perubahan
¹⁶ tipe *data* kolom *timestamp* dan *data*. Tabel 5.5 menunjukkan struktur tabel baru pada *shj_sessions*.

Tabel 5.5: Struktur baru tabel *shj_sessions*

| Atribut | Tipe Data | Ukuran | Default |
|----------------------------------|-----------|--------|---------------------|
| <i>id</i> (<i>primary key</i>) | varchar | 128 | None |
| <i>ip_address</i> | varchar | 45 | None |
| <i>timestamp</i> | timestamp | - | current_timestamp() |
| <i>data</i> | blob | - | None |

1 5.3 Pengujian Fungsional

- 2 Pengujian fungsional dilakukan pada lingkungan *staging* yang bertujuan untuk melakukan pengujian
3 terhadap seluruh fitur pada perangkat lunak *SharIF Judge*. Pengujian dilakukan untuk memastikan
4 setiap fitur yang terdapat pada *SharIF Judge* versi *CodeIgniter 4* dapat berjalan dengan baik setelah
5 dilakukan konversi. Berikut merupakan pengujian terhadap fitur-fitur pada *SharIF Judge*:
- 6 1. Melakukan instalasi *SharIF Judge*
7 • Reaksi yang diharapkan: *Database* terbentuk dan akun *admin* tersimpan pada *database*.
8 • Reaksi yang ditemukan: *Database* terbentuk dan akun *admin* tersimpan pada *database*.
9 Halaman *error* dengan pesan **SharIF Judge is already installed.** ditampilkan
10 apabila tabel sudah terbentuk.
- 11 2. Melakukan pengamanan terhadap *sandbox*
12 • Reaksi yang diharapkan: Dapat membangun *sandbox* tanpa mengembalikan *error*.
13 • Reaksi yang ditemukan: Pembangunan *sandbox* mengembalikan *success message* berupa
14 **All tests passed!**.
- 15 3. Melakukan *login*
16 • Reaksi yang diharapkan: Pengguna masuk menuju *dashboard* menggunakan akun yang
17 telah dibentuk.
18 • Reaksi yang ditemukan: Pengguna masuk menuju *dashboard* dengan akun yang telah
19 dibentuk. Pengguna tidak dapat *login* dan aplikasi mengembalikan *error message* berupa
20 *Incorrect username or password* apabila *username* atau *password* yang dimasukan salah.
- 21 4. Melakukan *register*
22 • Reaksi yang diharapkan: Pengguna berhasil melakukan *register* dan aplikasi mengembalikan
23 *Registered successfully*.
24 • Reaksi yang ditemukan: Pengguna berhasil melakukan *register* dan aplikasi mengembalikan
25 *Registered successfully*. Aplikasi mengembalikan *error message* berupa **Registration is Closed** apabila fitur *register* tidak dinyalakan.
- 26 5. Melakukan *reset password* dengan *email* yang telah terdaftar
27 • Reaksi yang diharapkan: Dikirimkan *email* berupa *link* untuk melakukan *reset password*,
28 pengguna diarahkan menuju situs *SharIF Judge*, dan pengguna memasukan *password*
29 baru untuk *login*.
30 • Reaksi yang ditemukan: Dikirimkan *email* berupa *link* untuk melakukan *reset password*,
31 pengguna diarahkan menuju situs *SharIF Judge*, dan pengguna dapat memasukan
32 *password* baru untuk *login*.
- 33 6. *Top bar* pengguna *admin*
34 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown tools*, *dropdown select assignments*,
35 *timer assignment*, dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*,
36 dan *cheat detection*.
37 • Reaksi yang ditemukan: *Top bar* memiliki *dropdown tools*, *dropdown select assignments*,
38 *timer assignment*, dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*,
39 dan *cheat detection*.
- 40 7. *Top bar* pengguna *head instructor*
41 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown tools*, *dropdown select assignments*,

1 *timer assignment*, dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*,
2 dan *cheat detection*.

- 3 • Reaksi yang ditemukan: *Top bar* terdapat *dropdown tools*, *dropdown select assignments*,
4 *timer assignment*, dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*,
5 dan *cheat detection*.

6 8. *Top bar* pengguna *instructor*

- 7 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown select assignments*, *timer assignment*,
8 dan *icon profile*.
9 • Reaksi yang ditemukan: *Top bar* terdapat *dropdown select assignments*, *timer assignment*,
10 dan *icon profile*.

11 9. *Top bar* pengguna *student*

- 12 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown select assignments*, *timer assignment*,
13 dan *icon profile*.
14 • Reaksi yang ditemukan: *Top bar* terdapat *dropdown select assignments*, *timer assignment*,
15 dan *icon profile*.

16 10. *Side bar* pengguna *admin*

- 17 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard*, *settings*, *users*, *notifications*,
18 *assignments*, *problems*, *submit*, *final submissions*, *all submissions*, *scoreboard*, *hall of*
19 *fame*, dan *24-hour log*.
20 • Reaksi yang ditemukan: *Sidebar* berisikan *dashboard*, *settings*, *users*, *notifications*,
21 *assignments*, *problems*, *submit*, *final submissions*, *all submissions*, *scoreboard*, *hall of*
22 *fame*, dan *24-hour log*.

23 11. *Side bar* pengguna *head instructor*

- 24 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard*, *notifications*, *assignments*, *pro-*
25 *blems*, *submit*, *final submissions*, *all submissions*, *scoreboard*, dan *hall of fame*.
26 • Reaksi yang ditemukan: *Sidebar* berisikan *dashboard*, *notifications*, *assignments*, *problems*,
27 *submit*, *final submissions*, *all submissions*, *scoreboard*, dan *hall of fame*.

28 12. *Side bar* pengguna *instructor*

- 29 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard*, *notifications*, *assignments*, *pro-*
30 *blems*, *submit*, *final submissions*, *all submissions*, *scoreboard*, dan *hall of fame*.
31 • Reaksi yang ditemukan: *Sidebar* berisikan *dashboard*, *notifications*, *assignments*, *problems*,
32 *submit*, *final submissions*, *all submissions*, *scoreboard*, dan *hall of fame*.

33 13. *Side bar* pengguna *student*

- 34 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard*, *notifications*, *assignments*, *pro-*
35 *blems*, *submit*, *final submissions*, *all submissions*, *scoreboard*, dan *hall of fame*.
36 • Reaksi yang ditemukan: *Sidebar* berisikan *dashboard*, *notifications*, *assignments*, *problems*,
37 *submit*, *final submissions*, *all submissions*, *scoreboard*, dan *hall of fame*.

38 14. *Dashboard* pengguna *admin* dan *head instructor*

- 39 • Reaksi yang diharapkan: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments*
40 dan *latest notification* dengan *icon* untuk menambahkan *notification*.
41 • Reaksi yang ditemukan: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments*
42 dan *latest notification* dengan *icon* untuk menambahkan *notification*.

1 15. *Dashboard* pengguna *instructor* dan *student*

- 2 • Reaksi yang diharapkan: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments*
3 dan *latest notification*.
- 4 • Reaksi yang ditemukan: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments*
5 dan *latest notification*.

6 16. *Profile* pengguna *admin*

- 7 • Reaksi yang diharapkan: Berisikan *username*, *name*, *email*, *password*, *password again*,
8 dan *user role*.
- 9 • Reaksi yang ditemukan: Berisikan *username*, *name*, *email*, *password*, *password again*,
10 dan *user role*.

11 17. *Profile* pengguna *head instructor*, *instructor*, dan *student*

- 12 • Reaksi yang diharapkan: Berisikan *username*, *name*, *email*, *password*, dan password
13 again.
- 14 • Reaksi yang ditemukan: Berisikan *username*, *name*, *email*, *password*, dan password
15 again.

16 18. Mengubah *password* akun pada *profile*

- 17 • Reaksi yang diharapkan: *Password* pada akun berubah dan dapat dilakukan oleh seluruh
18 pengguna.
- 19 • Reaksi yang ditemukan: *Password* pada akun berubah dan dapat dilakukan oleh seluruh
20 pengguna.

21 19. Mengubah *role*

- 22 • Reaksi yang diharapkan: Penggantian *role* hanya dapat dilakukan oleh *admin*.
- 23 • Reaksi yang ditemukan: Penggantian *role* hanya dapat dilakukan oleh *admin* dan
24 mengembalikan *error* apabila pengguna bukan *admin*.

25 20. Halaman *settings*

- 26 • Reaksi yang diharapkan: Halaman hanya dapat diakses oleh *admin* dan mengeluarkan
27 *error message* apabila menggunakan *role* lain.
- 28 • Reaksi yang ditemukan: Halaman hanya dapat diakses oleh *admin* dan mengeluarkan
29 *error message* berupa 404 apabila diakses oleh *role* lain.

30 21. Melakukan *sign out*

- 31 • Reaksi yang diharapkan: Pengguna kembali ke halaman *login* dan *session* pengguna
32 dihancurkan.
- 33 • Reaksi yang ditemukan: Pengguna kembali ke halaman *login* dan *session* pengguna
34 dihancurkan.

35 22. Mengubah konfigurasi pada halaman *settings*

- 36 • Reaksi yang diharapkan: Konfigurasi yang diubah disimpan menuju *database*.
- 37 • Reaksi yang ditemukan: Konfigurasi yang diubah disimpan menuju *database*

38 23. Menyalakan fitur *registration*

- 39 • Reaksi yang diharapkan: Halaman *login* terdapat tombol *register* dan pengguna dapat
40 melakukan *register*.
- 41 • Reaksi yang ditemukan: Halaman *login* terdapat tombol *register* dan pengguna dapat
42 melakukan *register*.

- 1 24. Menyalakan fitur *lock student display name*
 - 2 • Reaksi yang diharapkan: Fitur untuk mengubah *display name* telah *disable*.
 - 3 • Reaksi yang ditemukan: Fitur untuk mengubah *display name* telah *disable*.
- 4 25. Halaman *users*
 - 5 • Reaksi yang diharapkan: Halaman hanya dapat diakses oleh *admin*.
 - 6 • Reaksi yang ditemukan: Halaman hanya dapat diakses oleh *admin* dan mengembalikan *error* berupa 404 apabila pengguna bukan *admin*.
- 8 26. Menambah *users* dan mengirimkan email kepada setiap pengguna
 - 9 • Reaksi yang diharapkan: Pengguna dapat ditambahkan lebih dari satu dan email dikirimkan berisikan data pengguna.
 - 11 • Reaksi yang ditemukan: Pengguna bertambah dan email dikirimkan berisikan data dari pengguna berupa *username* dan *password*.
- 13 27. Mengunduh pengguna dalam format *excel*
 - 14 • Reaksi yang diharapkan: Seluruh data pada halaman *users* terunduh pada perangkat pengguna dalam format *excel*.
 - 16 • Reaksi yang ditemukan: Seluruh data pada halaman *users* terunduh pada perangkat pengguna dalam format *excel*.
- 18 28. Mengganti data pengguna lain
 - 19 • Reaksi yang diharapkan: Data pengguna lain hanya dapat diubah oleh *admin*. Data berubah dan pengguna dapat mengakses menggunakan data yang baru.
 - 21 • Reaksi yang ditemukan: Data pengguna lain hanya dapat diubah oleh *admin* dan mengembalikan *error* berupa 404 apabila pengguna bukan *admin*. Data berubah dan pengguna dapat mengakses menggunakan data yang baru.
- 24 29. Melihat seluruh pengumpulan setiap pengguna
 - 25 • Reaksi yang diharapkan: Pengguna *student* hanya dapat melihat seluruh pengumpulannya. Pengguna *admin*, *head instructor*, dan *instructor* dapat melihat pengumpulan pengguna lain.
 - 28 • Reaksi yang ditemukan: Pengguna *student* hanya dapat melihat seluruh pengumpulannya. Pengguna *admin*, *head instructor*, dan *instructor* dapat melihat pengumpulan pengguna lain.
- 31 30. Menghapus pengguna
 - 32 • Reaksi yang diharapkan: Hanya dapat dilakukan oleh *admin* dan pengguna terhapus dari *database*.
 - 34 • Reaksi yang ditemukan: Hanya dapat dilakukan oleh *admin* dan apabila dilakukan oleh pengguna lain mengembalikan *error* 404. Data pengguna terhapus dari *database*.
- 36 31. Menghapus *submission* pengguna
 - 37 • Reaksi yang diharapkan: Seluruh *submission* dari pengguna tersebut terhapus.
 - 38 • Reaksi yang ditemukan: Seluruh *submission* dari pengguna tersebut terhapus.
- 39 32. Halaman *notifications*
 - 40 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna namun tombol *add*, *edit* dan *delete* hanya terdapat pada pengguna *admin* dan *head instructor*.
 - 42 • Reaksi yang ditemukan: Dapat diakses oleh seluruh pengguna namun tombol *add*, *edit*

1 dan *delete* hanya terdapat pada pengguna *admin* dan *head instructor*.

2 33. Halaman *assignments*

- 3 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna dan dapat memilih
4 *assignment*.
5 • Reaksi yang ditemukan: Dapat diakses oleh seluruh pengguna dan dapat memilih
6 *assignment*.

7 34. Memilih *assignment*

- 8 • Reaksi yang diharapkan: Waktu pada *top bar* berganti menjadi penghitung mundur
9 waktu *assignment* selesai. Apabila terdapat *extra time* maka muncul tampilan dan
10 penghitung mundur *extra time*.
11 • Reaksi yang ditemukan: Waktu pada *top bar* berganti menjadi penghitung mundur waktu
12 *assignment* selesai. Apabila terdapat *extra time* maka muncul tampilan dan penghitung
13 mundur *extra time*.

14 35. Menambah *assignment*

- 15 • Reaksi yang diharapkan: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Assignment*
16 bertambah pada direktori *assignments* dan data terextract. Dapat menambahkan
17 *problem* dan melakukan konfigurasi sesuai dengan kebutuhan.
18 • Reaksi yang ditemukan: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Assignment*
19 bertambah pada direktori *assignments* dan data terextract. Dapat menambahkan
20 *problem* dan melakukan konfigurasi sesuai dengan kebutuhan.

21 36. Mengunduh *pdf* soal

- 22 • Reaksi yang diharapkan: *PDF* soal terunduh menuju perangkat pengguna.
23 • Reaksi yang ditemukan: *PDF* soal terunduh menuju perangkat pengguna.

24 37. Mengunduh *test case* dan soal *PDF*

- 25 • Reaksi yang diharapkan: *Test case* dan soal *pdf* terunduh menuju perangkat pengguna
26 dengan *extension zip*.
27 • Reaksi yang ditemukan: *Test case* dan soal *pdf* terunduh menuju perangkat pengguna
28 dengan *extension zip*.

29 38. Mengunduh *final submission* berdasarkan pengguna

- 30 • Reaksi yang diharapkan: Seluruh *final submission* dari setiap pengguna yang meng-
31 umpulkan terunduh pada perangkat dengan *extension zip*. Seluruh *file* yang terunduh
32 dikelompokan pada direktori berdasarkan pengguna.
33 • Reaksi yang ditemukan: Seluruh *final submission* dari setiap pengguna yang mengum-
34 pulkan terunduh pada perangkat dengan *extension zip*. Seluruh *file* yang terunduh
35 dikelompokan pada direktori berdasarkan pengguna.

36 39. Mengunduh *final submission* berdasarkan *problem*

- 37 • Reaksi yang diharapkan: Seluruh *final submission* dari pengguna yang mengumpulkan
38 terunduh pada perangkat dengan *extension zip*. Seluruh *file* yang terunduh dikelompokan
39 pada direktori berdasarkan *problem*.
40 • Reaksi yang ditemukan: Seluruh *final submission* dari pengguna yang mengumpulkan
41 terunduh pada perangkat dengan *extension zip*. Seluruh *file* yang terunduh dikelompokan
42 pada direktori berdasarkan *problem*.

1 40. Mendeteksi *similar code* pada *assignment*

- 2 • Reaksi yang diharapkan: Pengguna diarahkan menuju halaman *cheat detection* dengan
3 *id* dari *assignment* yang dipilih. Dihasilkan tautan yang berisikan hasil dari pengecekan
4 *similar code*.
5 • Reaksi yang ditemukan: Pengguna diarahkan menuju halaman *cheat detection* dengan *id*
6 dari *assignment* yang dipilih. Dihasilkan tautan yang berisikan hasil dari pengecekan
7 *similar code*.

8 41. Melakukan *edit assignment*

- 9 • Reaksi yang diharapkan: *Assignment* berubah sesuai dengan konfigurasi pengguna.
10 • Reaksi yang ditemukan: *Assignment* berubah sesuai dengan konfigurasi pengguna.

11 42. Menghapus *assignment*

- 12 • Reaksi yang diharapkan: *Assignment terhapus* dari halaman *assignments*.
13 • Reaksi yang ditemukan: *Assignment terhapus* dari halaman *assignments*.

14 43. Halaman *problems*

- 15 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna dan pengguna dapat
16 mengumpulkan jawaban melalui halaman tersebut. Namun, dekripsi *problem* hanya dapat
17 dilakukan oleh *admin* dan *head instructor*.
18 • Reaksi yang ditemukan: Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna
19 dan pengguna dapat mengumpulkan jawaban pada halaman tersebut. Namun, dekripsi
20 *problem* hanya dapat dilakukan oleh *admin* dan *head instructor*.

21 44. Menambahkan deskripsi *problem* menggunakan *markdown*

- 22 • Reaksi yang diharapkan: Deskripsi *problem* ditambahkan dengan tipe bahasa *markdown*.
23 Deskripsi ditampilkan pada halaman *problem*.
24 • Reaksi yang ditemukan: Deskripsi *problem* ditambahkan dengan tipe bahasa *markdown*.
25 Deskripsi ditampilkan pada halaman *problem*.

26 45. Menambahkan deskripsi *problem* menggunakan *HTML*

- 27 • Reaksi yang diharapkan: Deskripsi *problem* ditambahkan dengan tipe bahasa *HTML*.
28 Deskripsi ditampilkan pada halaman *problem*.
29 • Reaksi yang ditemukan: Deskripsi *problem* ditambahkan dengan tipe bahasa *HTML*.
30 Deskripsi ditampilkan pada halaman *problem*.

31 46. Menambahkan deskripsi *problem* menggunakan *plain HTML*

- 32 • Reaksi yang diharapkan: Deskripsi *problem* ditambahkan dengan tipe bahasa *plain*
33 *HTML*. Deskripsi ditampilkan pada halaman *problem*.
34 • Reaksi yang ditemukan: Deskripsi *problem* ditambahkan dengan tipe bahasa *plain HTML*.
35 Deskripsi ditampilkan pada halaman *problem*.

36 47. Halaman *submit*

- 37 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna. Pengguna dapat memilih
38 *problem* dan bahasa pemrogramannya. *PDF* soal ditampilkan. Pengguna dapat langsung
39 mencoba dan mengumpulkan kode melalui *editor* kode.
40 • Reaksi yang ditemukan: Dapat diakses oleh seluruh pengguna. Pengguna dapat memilih
41 *problem* dan bahasa pemrogramannya. *PDF* soal ditampilkan. Pengguna dapat langsung
42 mencoba dan mengumpulkan kode melalui *editor* kode.

1 48. Halaman *final submissions*

- 2 • Reaksi yang diharapkan: Melihat seluruh *final submission* dari seluruh pengguna apabila
3 pengguna berupa *admin*, *head instructor*, dan *instructor*. Melihat seluruh *final submission*
4 dari pengguna apabila pengguna *student*.
5 • Reaksi yang ditemukan: Melihat seluruh *final submission* dari seluruh pengguna apabila
6 pengguna berupa *admin*, *head instructor*, dan *instructor*. Melihat seluruh *final submission*
7 dari pengguna apabila pengguna *student*.

8 49. Mengunduh data *final submission* dalam format *excel*

- 9 • Reaksi yang diharapkan: Seluruh data dari halaman *final submission* terunduh kepada
10 perangkat pengguna dalam format *excel*.
11 • Reaksi yang ditemukan: Seluruh data dari halaman *final submission* terunduh kepada
12 perangkat pengguna dalam format *excel*.

13 50. Melihat status dari *submission* dan *test case problem*

- 14 • Reaksi yang diharapkan: Memperlihatkan status dan nilai dari kode. Setiap *test case*
15 memperlihatkan apakah benar, salah, atau *runtime error*
16 • Reaksi yang ditemukan: Memperlihatkan status dan nilai dari kode. Setiap *test case*
17 memperlihatkan apakah benar, salah, atau *runtime error*

18 51. Melakukan *filter* berdasarkan nama pengguna

- 19 • Reaksi yang diharapkan: Hanya terlihat *final submissions* dari pengguna tersebut.
20 • Reaksi yang ditemukan: Hanya terlihat *final submissions* dari pengguna tersebut.

21 52. Melakukan *filter* berdasarkan *problem*

- 22 • Reaksi yang diharapkan: Hanya terlihat *final submissions* dari *problem* tersebut.
23 • Reaksi yang ditemukan: Hanya terlihat *final submissions* dari *problem* tersebut.

24 53. Melakukan *rejudge*

- 25 • Reaksi yang diharapkan: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Su-*
26 *bmission* dilakukan penilaian ulang dan mengubah status sementara menjadi *pending*.
27 • Reaksi yang ditemukan: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Su-*
28 *bmission* dilakukan penilaian ulang dan mengubah status sementara menjadi *pending*.

29 54. Halaman *all submissions*

- 30 • Reaksi yang diharapkan: Melihat seluruh *submission* yang telah dikumpulkan dan dapat
31 memilih *submission* yang ingin dijadikan hasil akhir. Melihat status dari *submission*
32 pengguna. Melihat kode yang sudah dikumpulkan. Melihat *log* dari setiap *submission*
33 apabila pengguna berupa *admin*, *head instructor*, dan *instructor*.
34 • Reaksi yang ditemukan: Melihat seluruh *submission* yang telah dikumpulkan dan dapat
35 memilih *submission* yang ingin dijadikan hasil akhir. Melihat status dari *submission*
36 pengguna. Melihat kode yang sudah dikumpulkan. Melihat *log* dari setiap *submission*
37 apabila pengguna berupa *admin*, *head instructor*, dan *instructor*.

38 55. Melakukan *filter* berdasarkan nama pengguna

- 39 • Reaksi yang diharapkan: Hanya terlihat *all submissions* dari pengguna tersebut.
40 • Reaksi yang ditemukan: Hanya terlihat *all submissions* dari pengguna tersebut.

41 56. Melakukan *filter* berdasarkan *problem*

- 42 • Reaksi yang diharapkan: Hanya terlihat *all submissions* dari *problem* tersebut.

- Reaksi yang ditemukan: Hanya terlihat *all submissions* dari *problem* tersebut.
57. Mengunduh data *all submission* dalam format *excel*
- Reaksi yang diharapkan: Seluruh data dari halaman *all submission* terunduh kepada perangkat pengguna dalam format *excel*.
 - Reaksi yang ditemukan: Seluruh data dari halaman *all submission* terunduh kepada perangkat pengguna dalam format *excel*.
58. Halaman *hall of fame*
- Reaksi yang diharapkan: Melihat skor total dari seluruh *assignment* setiap pengguna dan mengurutkannya dari yang terbesar.
 - Reaksi yang ditemukan: Melihat skor total dari seluruh *assignment* setiap pengguna dan mengurutkannya dari yang terbesar.
59. Halaman *24-hour log*
- Reaksi yang diharapkan: Hanya dapat diakses oleh *admin* dan berisikan seluruh data perangkat dari pengguna. Pengguna yang melakukan *login* dari *ip address* yang berbeda terdeteksi dan ditampilkan.
 - Reaksi yang ditemukan: Hanya dapat diakses oleh *admin* dan berisikan seluruh data perangkat dari pengguna. Pengguna yang melakukan *login* dari *ip address* yang berbeda terdeteksi dan ditampilkan
60. Halaman *submission queue*
- Reaksi yang diharapkan: Dapat memberhentikan, melanjutkan, dan membersihkan antrian dari *submission* yang ada.
 - Reaksi yang ditemukan: Dapat memberhentikan, melanjutkan, dan membersihkan antrian dari *submission* yang ada.

5.4 Pengujian Experimental

Pengujian experimental dilakukan pada tiga buah versi sistem operasi *Ubuntu* yakni versi 22.04, 23.04, dan 22.10. Pengujian ini bertujuan untuk menguji perangkat lunak apakah dapat berjalan pada sistem operasi yang berbeda.

5.4.1 *Ubuntu* 22.04

Pengujian experimental pertama dilakukan pada sistem operasi *Ubuntu* dengan versi 22.04. Pengujian dilakukan dengan melakukan instalasi dan melakukan pengamanan terhadap *sandbox*. Setelah dilakukan instalasi terdapat persoalan dalam melakukan pengamanan terhadap *sandbox*. Berikut merupakan persoalan yang didapatkan:

Sandbox tidak terbangun

Pengujian dilakukan dengan membangun *sandbox* untuk kode C/C++. Pembangunan *sandbox* dilakukan pada direktori *tester/easybox* dengan menjalankan kode berikut:

Kode 5.1: Pembangunan *sandbox* pada *Ubuntu* 22.04

```
36
37 1 $ cd tester/easybox
38 2 $ chmod +x runalltests.sh
```

```
13 $ chmod +x runtest.sh
34 $ make runtests
```

4 Kode 5.1 memindahkan pengguna menuju direktori `tester/easysandbox`. Selanjutnya akses
 5 `file runalltests` dan `runtest` diubah agar dapat dieksekusi. Terakhir `file runtests` dijalankan
 6 menggunakan sintaks `make`. Pembangunan *sandbox* menghasilkan pesan berupa `All tests passed!`
 7 Namun pada sistem operasi ini, *sandbox* tidak dapat dibentuk dan menghasilkan *error message*
 8 seperti yang ditunjukkan pada Kode 5.2.

Kode 5.2: *Error message* pembangunan *sandbox* pada *Ubuntu 22.04*

```
9
10 1 ./runalltests.sh t/test01 t/test02 t/test03 t/test04 t/test05 t/test06 t/test07 t/test08 t/test09 t/test10 t/test11 t/test12 t/
11   test13 t/test14
12 2 Executing t/test01...passed!
13 3 Executing t/test02...passed!
14 4 Executing t/test03...passed!
15 5 Executing t/test04...passed!
16 6 Executing t/test05...passed!
17 7 Executing t/test06...passed!
18 8 Executing t/test07...passed!
19 9 Executing t/test08...passed!
20 0 Executing t/test09...1a2
21 1 > Hello, C++ world
22 2 failed (output mismatch, expected [<<entering SECCOMP mode>>
23 3 Hello, C++ world], got [<<entering SECCOMP mode>>])
24 4 Executing t/test10...1a2
25 5 > Hello from the constructor!
26 6 failed (output mismatch, expected [<<entering SECCOMP mode>>
27 7 Hello from the constructor!], got [<<entering SECCOMP mode>>])
28 8 Executing t/test11...passed!
29 9 Executing t/test12...1a2
30 0 > Here we are in main()
31 1 > Hello from the destructor!
32 2 failed (output mismatch, expected [<<entering SECCOMP mode>>
33 3 Here we are in main()
34 4 Hello from the destructor!], got [<<entering SECCOMP mode>>])
35 5 Executing t/test13...passed!
36 6 Executing t/test14...1a2
37 7 > 500500
38 8 failed (output mismatch, expected [<<entering SECCOMP mode>>
39 9 500500], got [<<entering SECCOMP mode>>])
40 0 4 test(s) failed
41 1 make: *** [Makefile:31: runtests] Error 1
```

43 Kode 5.2 merupakan keluaran dari pembangunan *sandbox* pada *Ubuntu 22.04*. Terdapat *error*
 44 *message* berupa keluaran tidak sesuai dengan yang seharusnya. Pembangunan dapat dikatakan tidak
 45 berhasil karena hanya sebagian *test case* yang berhasil dilewati dan tidak terdapat pesan berupa
 46 `All tests passed!`. *Error message* ini dihasilkan karena terdapat perbedaan versi pada *seccomp*
 47 atau *secure computing mode* sehingga tidak dapat dijalankan pada versi teratas pada *Ubuntu*.

48 5.4.2 *Ubuntu 23.04*

49 Pengujian experimental kedua dilakukan pada sistem operasi *Ubuntu* dengan versi 23.04. Pengujian
 50 dilakukan dengan melakukan instalasi dan melakukan pengamanan terhadap *sandbox*. Setelah
 51 dilakukan instalasi terdapat persoalan dalam melakukan pengamanan terhadap *sandbox*. Berikut
 52 merupakan persoalan yang didapatkan:

Kode 5.3: Pembangunan *sandbox* pada *Ubuntu 23.04*

```
53
54 1 $ cd tester/easysandbox
55 2 $ chmod +x runalltests.sh
56 3 $ chmod +x runtest.sh
57 4 $ make runtests
```

1 Kode 5.3 memindahkan pengguna menuju direktori `tester/easy sandbox`. Selanjutnya akses
 2 file `runalltests` dan `runttest` diubah agar dapat dieksekusi. Terakhir file `runtests` dijalankan
 3 menggunakan sintaks `make`. Pembangunan `sandbox` menghasilkan pesan berupa `All tests passed!`.
 4 Namun pada sistem operasi ini, `sandbox` tidak dapat dibentuk dan menghasilkan *error message*
 5 seperti yang ditunjukkan pada Kode 5.4.

Kode 5.4: *Error message* pembangunan `sandbox` pada *Ubuntu* 23.04

```

6 71 /runalltests.sh t/test01 t/test02 t/test03 t/test04 t/test05 t/test06 t/test07 t/test08 t/test09 t/test10 t/test11 t/test12 t/
7   test13 t/test14
8 Executing t/test01...failed (exit code mismatch, expected 0, got 139)
9 Executing t/test02...1a2
10 > Hello, world
11 failed (output mismatch, expected [<<entering SECCOMP mode>>
12 Hello, world], got [<<entering SECCOMP mode>>])
13 Executing t/test03...failed (exit code mismatch, expected 137, got 139)
14 Executing t/test04...1a2
15 > 500500
16 failed (output mismatch, expected [<<entering SECCOMP mode>>
17 500500], got [<<entering SECCOMP mode>>])
18 Executing t/test05...1a2
19 > Hello, world
20 failed (output mismatch, expected [<<entering SECCOMP mode>>
21 Hello, world], got [<<entering SECCOMP mode>>])
22 Executing t/test06...failed (exit code mismatch, expected 137, got 139)
23 Executing t/test07...1a2
24 > 59
25 failed (output mismatch, expected [<<entering SECCOMP mode>>
26 59], got [<<entering SECCOMP mode>>])
27 Executing t/test08...failed (exit code mismatch, expected 0, got 139)
28 Executing t/test09...1a2
29 > Hello, C++ world
30 failed (output mismatch, expected [<<entering SECCOMP mode>>
31 Hello, C++ world], got [<<entering SECCOMP mode>>])
32 Executing t/test10...1a2
33 > Hello from the constructor!
34 failed (output mismatch, expected [<<entering SECCOMP mode>>
35 Hello from the constructor!], got [<<entering SECCOMP mode>>])
36 Executing t/test11...failed (exit code mismatch, expected 137, got 139)
37 Executing t/test12...1a2,3
38 > Here we are in main()
39 > Hello from the destructor!
40 failed (output mismatch, expected [<<entering SECCOMP mode>>
41 Here we are in main()
42 Hello from the destructor!], got [<<entering SECCOMP mode>>])
43 Executing t/test13...1a2
44 > Hello from the destructor!
45 failed (output mismatch, expected [<<entering SECCOMP mode>>
46 Hello from the destructor!], got [<<entering SECCOMP mode>>])
47 Executing t/test14...1a2
48 > 500500
49 failed (output mismatch, expected [<<entering SECCOMP mode>>
50 500500], got [<<entering SECCOMP mode>>])
51 14 test(s) failed
52 make: *** [Makefile:31: runtests] Error 1
53

```

55 Kode 5.4 merupakan keluaran dari pembangunan `sandbox` pada *Ubuntu* 23.04. Terdapat *error*
 56 *message* berupa keluaran tidak sesuai dengan yang seharusnya. Pembangunan dapat dikatakan
 57 tidak berhasil karena seluruh *test case* tidak berhasil dilewati dan tidak terdapat pesan berupa `All`
 58 `tests passed!`. *Error message* ini dihasilkan karena terdapat perbedaan versi pada `seccomp` atau
 59 `secure computing mode` sehingga tidak dapat dijalankan pada versi teratas pada *Ubuntu*.

60 5.4.3 *Ubuntu* 23.10

61 Pengujian experimental ketiga dilakukan pada sistem operasi *Ubuntu* dengan versi 23.10. Pengujian
 62 dilakukan dengan melakukan instalasi dan melakukan pengamanan terhadap `sandbox`. Setelah
 63 dilakukan instalasi terdapat persoalan dalam melakukan pengamanan terhadap `sandbox`. Berikut
 64 merupakan persoalan yang didapatkan:

Kode 5.5: Pembangunan *sandbox* pada *Ubuntu 23.10*

```

1  $ cd tester/easysandbox
2  $ chmod +x runalltests.sh
3  $ chmod +x runtest.sh
4  $ make runtests

```

7 Kode 5.5 memindahkan pengguna menuju direktori `tester/easysandbox`. Selanjutnya akses
 8 file `runalltests` dan `runtest` diubah agar dapat dieksekusi. Terakhir file `runtests` dijalankan
 9 menggunakan sintaks `make`. Pembangunan *sandbox* menghasilkan pesan berupa `All tests passed!`.
 10 Namun pada sistem operasi ini, *sandbox* tidak dapat dibentuk dan menghasilkan *error message*
 11 seperti yang dijelaskan pada Kode 5.6.

Kode 5.6: *Error message* pembangunan *sandbox* pada *Ubuntu 23.10*

```

12 g++ -g -Wall -D_BSD_SOURCE -o t/test09 t/test09.cpp
13 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,
14   from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,
15   from /usr/include/c++/13/bits/requirements_hosted.h:31,
16   from /usr/include/c++/13/iostream:38,
17   from t/test09.cpp:3:
18 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
19 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"
20  |   ~~~~~
21 g++ -g -Wall -D_BSD_SOURCE -o t/test10 t/test10.cpp
22 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,
23   from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,
24   from /usr/include/c++/13/bits/requirements_hosted.h:31,
25   from /usr/include/c++/13/iostream:38,
26   from t/test10.cpp:3:
27 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
28 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"
29  |   ~~~~~
30 g++ -g -Wall -D_BSD_SOURCE -o t/test11 t/test11.cpp
31 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,
32   from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,
33   from /usr/include/c++/13/bits/requirements_hosted.h:31,
34   from /usr/include/c++/13/iostream:38,
35   from t/test11.cpp:4:
36 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
37 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"
38  |   ~~~~~
39 g++ -g -Wall -D_BSD_SOURCE -o t/test12 t/test12.cpp
40 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,
41   from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,
42   from /usr/include/c++/13/bits/requirements_hosted.h:31,
43   from /usr/include/c++/13/iostream:38,
44   from t/test12.cpp:3:
45 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
46 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"
47  |   ~~~~~
48 gcc -std=c99 -g -Wall -D_BSD_SOURCE -o t/test13 t/test13.c
49 In file included from /usr/include/x86_64-linux-gnu/bits/libc-header-start.h:33,
50   from /usr/include/stdio.h:27,
51   from t/test13.c:3:
52 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
53 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"
54  |   ~~~~~
55 g++ -g -Wall -D_BSD_SOURCE -o t/test14 t/test14.cpp
56 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,
57   from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,
58   from /usr/include/c++/13/bits/requirements_hosted.h:31,
59   from /usr/include/c++/13/iostream:38,
60   from t/test14.cpp:4:
61 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
62 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"
63  |   ~~~~~
64 ./runalltests.sh t/test01 t/test02 t/test03 t/test04 t/test05 t/test06 t/test07 t/test08 t/test09 t/test10 t/test11 t/test12 t/
65   test13 t/test14
66 Executing t/test01...failed (exit code mismatch, expected 0, got 139)
67 Executing t/test02...1a2
68 > Hello, world
69 failed (output mismatch, expected [<<entering SECCOMP mode>>
70 Hello, world], got [<<entering SECCOMP mode>>])
71 Executing t/test03...failed (exit code mismatch, expected 137, got 139)
72 Executing t/test04...1a2
73 > 500500

```

```

62| failed (output mismatch, expected [<<entering SECCOMP mode>>
63| 500500], got [<<entering SECCOMP mode>>])
64 Executing t/test05...1a2
65 > Hello, world
66 failed (output mismatch, expected [<<entering SECCOMP mode>>
67 Hello, world], got [<<entering SECCOMP mode>>])
68 Executing t/test06...failed (exit code mismatch, expected 137, got 139)
69 Executing t/test07...1a2
70 > 59
71 failed (output mismatch, expected [<<entering SECCOMP mode>>
72 59], got [<<entering SECCOMP mode>>])
73 Executing t/test08...failed (exit code mismatch, expected 0, got 139)
74 Executing t/test09...1a2
75 > Hello, C++ world
76 failed (output mismatch, expected [<<entering SECCOMP mode>>
77 Hello, C++ world], got [<<entering SECCOMP mode>>])
78 Executing t/test10...1a2
79 > Hello from the constructor!
80 failed (output mismatch, expected [<<entering SECCOMP mode>>
81 Hello from the constructor!], got [<<entering SECCOMP mode>>])
82 Executing t/test11...failed (exit code mismatch, expected 137, got 139)
83 Executing t/test12...1a2,3
84 > Here we are in main()
85 > Hello from the destructor!
86 failed (output mismatch, expected [<<entering SECCOMP mode>>
87 Here we are in main()
88 Hello from the destructor!], got [<<entering SECCOMP mode>>])
89 Executing t/test13...1a2
90 > Hello from the destructor!
91 failed (output mismatch, expected [<<entering SECCOMP mode>>
92 Hello from the destructor!], got [<<entering SECCOMP mode>>])
93 Executing t/test14...1a2
94 > 500500
95 failed (output mismatch, expected [<<entering SECCOMP mode>>
96 500500], got [<<entering SECCOMP mode>>])
97 14 test(s) failed
98 make: *** [Makefile:31: runtests] Error 1

```

39 Kode 5.6 merupakan keluaran dari pembangunan *sandbox* pada *Ubuntu 23.10*. Terdapat *error message* berupa *warning* dan *test case* yang tidak berhasil dijalankan. Pembangunan dapat dikatakan
 40 tidak berhasil karena seluruh *test case* tidak berhasil dilewati dan tidak terdapat pesan berupa **All**
 41 **tests passed!**. *Error message* ini dihasilkan karena terdapat perbedaan versi pada *seccomp* atau
 42 *secure computing mode* sehingga tidak dapat dijalankan pada versi teratas pada *Ubuntu*.

1

BAB 6

2

KESIMPULAN DAN SARAN

3 Bab ini akan membahas kesimpulan yang telah diraih selama proses analisis, perancangan, im-
4 plementasi, pengujian yang telah dibentuk, serta saran-saran untuk pengembangan yang dapat
5 dipertimbangkan.

6 6.1 Kesimpulan

7 Berdasarkan proses dan hasil analisis, perancangan, implementasi, dan pengujian perangkat lunak
8 yang telah dibentuk, maka diperoleh kesimpulan sebagai berikut:

9 1. Perangkat lunak *SharIF Judge* telah berhasil dikonversi menuju *CodeIgniter 4* kecuali pengu-
10 jian terhadap autentikasi RADIUS dan LDAP. Pengujian tidak dilakukan karena terdapat
11 keterbatasan *server*. Seluruh fitur yang terdapat pada *SharIF Judge CodeIgniter 4* berjalan
12 sesuai dengan *SharIF Judge* pada *CodeIgniter 3* dengan beberapa perubahan sintaks.

13 6.2 Saran

14 Berdasarkan hasil analisis, perancangan, dan implementasi yang telah dilakukan, berikut merupakan
15 saran-saran untuk pengembangan selanjutnya:

- 16 1. Membentuk *Sandbox* untuk melakukan *compile* bahasa pemrograman *C*, *CPP*, dan *Python*
17 yang terintegrasi pada *Ubuntu* versi 22.04.
- 18 2. Melakukan dokumentasi terhadap fitur-fitur yang akan dibentuk karena dokumentasi terbatas
19 hanya pada pembentukan awal *SharIF Judge*.
- 20 3. Melakukan pengujian autentikasi *RADIUS* dan *LDAP*.

DAFTAR REFERENSI

- [1] Version 3.1.13 (2022) *CodeIgniter User Guide*. CodeIgniter Foundation. Richmond, Canada.
- [2] Version 1.4 (2023) *SharIF Judge Documentation*. Informatika UNPAR. Jl. Ciumbuleuit No. 94, Bandung.
- [3] Version 4.3 (2023) *CodeIgniter User Guide*. CodeIgniter Foundation. Richmond, Canada.
- [4] Prihatini, F. N. dan Indudewi, D. (2016) Kesadaran dan Perilaku Plagiarisme dikalangan Mahasiswa(Studi pada Mahasiswa Fakultas Ekonomi Jurusan Akuntansi Universitas Semarang). *Dinamika Sosial Budaya*, **18**, 68–75.
- [5] Kurnia, A., Lim, A., dan Cheang, B. (2001) Online judge. *Computers & Education*, **18**, 299–315.

LAMPIRAN A

KODE PROGRAM *CONTROLLERS*

Kode A.1: Assignments.php

```
1 <?php
2
3 namespace App\Controllers;
4
5 use App\Controllers\BaseController;
6 use App\Libraries\Unzip;
7 use App\Models\AssignmentModel;
8 use App\Models\SettingsModel;
9 use App\Models\SubmitModel;
10 use App\Models\User;
11 use ZipArchive;
12
13 /**
14 * SharIF Judge online judge
15 * @file Assignments.php
16 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
17 */
18
19 class Assignments extends BaseController
20 {
21
22     private $messages;
23     private $edit_assignment;
24     private $edit;
25     protected $assignment_model;
26     protected $validation;
27     protected $user;
28     protected $assignment;
29     protected $settings_model;
30     protected $zip;
31     protected $submit_model;
32     protected $unzip;
33     protected $parsedown;
34
35
36 // -----
37
38
39     public function __construct()
40     {
41         $this->messages = array();
42         $this->edit_assignment = array();
43         $this->edit = FALSE;
44         $this->assignment_model = new AssignmentModel();
45         $this->settings_model = new SettingsModel();
46         $this->validation = \Config\Services::validation();
47         $this->user = new User();
48     }
49
50
51 // -----
52
53
54     public function index()
55     {
56         $data = array(
57             'all_assignments' => $this->assignment_model->all_assignments(),
58             'messages' => $this->messages,
59             'selected' => 'assignments',
60             'user' => $this->user,
61             'finish_time' => $this->user->selected_assignment['finish_time'],
62             'extra_time' => $this->user->selected_assignment['extra_time'],
63         );
64
65         foreach ($data['all_assignments'] as &$item)
66         {
67             $extra_time = $item['extra_time'];
68             $delay = shj_now() - strtotime($item['finish_time']);
69             ob_start();
70             if ( eval($item['late_rule']) === FALSE )
71                 $coefficient = "error";
72             if (!isset($coefficient))
73                 $coefficient = "error";
74             ob_end_clean();
75             $item['coefficient'] = $coefficient;
76         }
77     }
78 }
```

```

76         $item['finished'] = ($delay > $extra_time);
77     }
78
79     return view('pages/assignments', $data);
80 }
81
82 // -----
83
84 /**
85 * Used by ajax request (for select assignment from top bar)
86 */
87 public function select()
88 {
89     if ( ! $this->request->isAJAX() )
90         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
91
92     $this->validation->setRule('assignment_select', 'Assignment', 'required|integer|greater_than[0]');
93
94     if ($this->validation->withRequest($this->request)->run())
95     {
96         $this->user->select_assignment($this->request->getPost('assignment_select'));
97         $this->assignment = $this->assignment_model->assignment_info($this->request->getPost('assignment_select'));
98         $json_result = array(
99             'done' => 1,
100             'finish_time' => $this->assignment['finish_time'],
101             'extra_time' => $this->assignment['extra_time'],
102         );
103     }
104     else{
105         $json_result = array('done' => 0, 'message' => 'Input_Error');
106     }
107     $this->response->setHeader('Content-Type:', 'application/json; charset=utf-8');
108     echo json_encode($json_result);
109 }
110
111 // -----
112
113
114
115
116 /**
117 * Download pdf file of an assignment (or problem) to browser
118 */
119 public function pdf($assignment_id, $problem_id = NULL, $no_download = FALSE)
120 {
121     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
122     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
123     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
124
125     // Find pdf file
126     if ($problem_id === NULL || $problem_id === "null")
127         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/*.pdf";
128     else
129         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/p{$problem_id}/*.pdf";
130     $pdf_files = glob($pattern);
131     if ( ! $pdf_files )
132         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("File_not_found");
133     elseif ( ! $this->assignment_model->assignment_info($assignment_id)[ 'open' ] )
134         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("Selected_assignment_has_been_closed.");
135     elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)[ 'participants' ], $this->user->username) )
136         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("You_are_not_registered_for_submitting.");
137     elseif ( shj_now() > $finishtime + $extratime )
138         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("Selected_assignment_has_finished.");
139     elseif ( shj_now() < $starttime )
140         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("Selected_assignment_has_not_started.");
141
142
143     $filename = shj_basename($pdf_files[0]);
144     // Download the file to browser
145     if($no_download === FALSE){
146         return $this->response->download($filename, file_get_contents($pdf_files[0]));
147     }
148     else{
149         $content = file_get_contents($pdf_files[0]);
150         header('Content-Type: application/pdf');
151         die($content);
152     }
153 }
154
155 // -----
156
157 /**
158 * Compressing and downloading test data and descriptions of an assignment to the browser
159 */
160 public function downloadtestsdesc($assignment_id = FALSE)
161 {
162     if ($assignment_id === FALSE)
163         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
164
165
166
167
168
169
170
171
172

```

```

173|     if ( $this->user->level <= 1 ) // permission denied
174|         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
175|
176|     $assignment = $this->assignment_model->assignment_info($assignment_id);
177|
178|     $number_of_problems = $assignment['problems'];
179|
180|     $root_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
181|     "/$assignment[$assignment_id]";
182|     $zipname = "assignment{$assignment_id}_tests_desc_.date('Y-m-d_H-i', shj_now()).zip";
183|     $this->zip = new \ZipArchive();
184|     $this->zip->open($zipname, ZipArchive::CREATE);
185|     for ($i=1 ; $i<=$number_of_problems ; $i++)
186|     {
187|
188|         $path = "$root_path/p{$i}/in";
189|         $options = ['add_path' => 'p{$i}/in/', 'remove_all_path' => TRUE];
190|         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
191|
192|         $path = "$root_path/p{$i}/out";
193|         $options = ['add_path' => 'p{$i}/out/', 'remove_all_path' => TRUE];
194|         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
195|
196|         $path = "$root_path/p{$i}/tester.cpp";
197|         if (file_exists($path))
198|             $this->zip->addFile($path,"p{$i}/tester.cpp");
199|
200|         $pdf_files = glob("$root_path/p{$i}/*.pdf");
201|         if ($pdf_files)
202|         {
203|             $path = $pdf_files[0];
204|             $this->zip->addFile($path,"p{$i}/".shj_basename($path));
205|         }
206|
207|         $path = "$root_path/p{$i}/desc.html";
208|         if (file_exists($path))
209|             $this->zip->addFile($path,"p{$i}/desc.html");
210|
211|         $path = "$root_path/p{$i}/desc.md";
212|         if (file_exists($path))
213|             $this->zip->addFile($path,"p{$i}/desc.md");
214|
215|         $pdf_files = glob("$root_path/*.*");
216|         if ($pdf_files)
217|         {
218|             $path = $pdf_files[0];
219|             $this->zip->addFile($path,shj_basename($path));
220|         }
221|
222|     $this->zip->close();
223|
224|     header('Content-Type:application/zip');
225|     header('Content-disposition:attachment;filename=' . $zipname);
226|     header('Content-Length:' . filesize($zipname));
227|     readfile($zipname);
228|     // $this->zip->download("assignment{$assignment_id}_tests_desc_.date('Y-m-d_H-i', shj_now()).zip");
229}
230
231
232// -----
233
234
235 /**
236 * Compressing and downloading final codes of an assignment to the browser
237 */
238 public function download_submissions($type = FALSE, $assignment_id = FALSE)
239{
240    if ($type !== 'by_user' && $type !== 'by_problem')
241        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
242    if ($assignment_id === FALSE || ! is_numeric($assignment_id))
243        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
244    if ( $this->user->level == 0 ) // permission denied
245        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
246
247    $this->submit_model = new SubmitModel();
248    $items = $this->submit_model->get_final_submissions($assignment_id, $this->user->level, $this->user->username);
249    $zipname = "assignment{$assignment_id}_submissions_{$type}_date('Y-m-d_H-i', shj_now()).zip";
250
251    $this->zip = new \ZipArchive();
252    $this->zip->open($zipname, ZipArchive::CREATE);
253    $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
254
255    foreach ($items as $item)
256    {
257        $file_path = $assignments_root.
258            "/$assignment[$item['assignment']]/" . $item['problem'] . "/" . $item['username'] . "/" . $item['file_name'] . "."
259            . filetype_to_extension($item['file_type']);
260        if ( ! file_exists($file_path) )
261            continue;
262        $file = file_get_contents($file_path);
263
264        if ($type === 'by_user')
265            $this->zip->addFile($file_path, "{$item['username']}/{$item['problem']}." . filetype_to_extension($item['file_type']));
266        elseif ($type === 'by_problem')
267            $this->zip->addFile($file_path, "problem_{$item['problem']}/{$item['username']}." . filetype_to_extension($item['file_type']));
268    }
269    $this->zip->close();

```

```

270     header('Content-Type:_application/zip');
271     header('Content-disposition:_attachment;_filename='.$zipname);
272     header('Content-Length:_' . filesize($zipname));
273     readfile($zipname);
274 }
275
276
277 // -----
278
279
280 /**
281 * Delete assignment
282 */
283 public function delete($assignment_id = FALSE)
284 {
285     if ($assignment_id === FALSE)
286         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
287     if ($this->user->level <= 1) // permission denied
288         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
289
290     $assignment = $this->assignment_model->assignment_info($assignment_id);
291
292     if ($assignment['id'] === 0)
293         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
294
295     if ($this->request->getPost('delete') === 'delete')
296     {
297         $this->assignment_model->delete_assignment($assignment_id);
298         return redirect()->to('assignments');
299     }
300
301     $data = array(
302         'all_assignments' => $this->assignment_model->all_assignments(),
303         'id' => $assignment_id,
304         'name' => $assignment['name'],
305         'selected' => 'assignments',
306         'icon' => 'fa-times',
307         'title' => 'Delete Assignments',
308         'user' => $this->user,
309         'finish_time' => $this->user->selected_assignment['finish_time'],
310         'extra_time' => $this->user->selected_assignment['extra_time'],
311
312     );
313
314     return view('pages/admin/delete_assignment', $data);
315 }
316
317
318 // -----
319
320
321 /**
322 * This method gets inputs from user for adding/editing assignment
323 */
324 public function add()
325 {
326
327     if ($this->user->level <= 1) // permission denied
328         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
329
330
331     if ($this->request->is('post')){
332         if ($this->add()) // add/edit assignment
333         {
334             //if ( ! $this->edit) // if adding assignment (not editing)
335             //{
336                 // goto Assignments page
337                 return $this->index();
338             //}
339         }
340     }
341
342     $data = array(
343         'all_assignments' => $this->assignment_model->all_assignments(),
344         'messages' => $this->messages,
345         'edit' => $this->edit,
346         'default_late_rule' => $this->settings_model->get_setting('default_late_rule'),
347         'selected' => 'assignments',
348         'icon' => 'fa-edit',
349         'title' => 'Assignments',
350         'user' => $this->user,
351         'finish_time' => $this->user->selected_assignment['finish_time'],
352         'extra_time' => $this->user->selected_assignment['extra_time'],
353         'twig' => $this->twig,
354         'validationError' => $this->validation,
355
356     );
357
358     if ($this->edit)
359     {
360         $data['edit_assignment'] = $this->assignment_model->assignment_info($this->edit_assignment);
361         if ($data['edit_assignment']['id'] === 0)
362             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
363         $data['problems'] = $this->assignment_model->all_problems($this->edit_assignment);
364     }
365     else
366     {
367
368

```

```

369     $names = $this->request->getPost('name');
370     if ($names === NULL)
371         $data['problems'] = array(
372             array(
373                 'id' => 1,
374                 'name' => 'Problem_',
375                 'score' => 100,
376                 'c_time_limit' => 500,
377                 'python_time_limit' => 1500,
378                 'java_time_limit' => 2000,
379                 'memory_limit' => 50000,
380                 'allowed_languages' => 'C,C++,Python_2,Python_3,Java',
381                 'diff_cmd' => 'diff',
382                 'diff_arg' => '-bB',
383                 'is_upload_only' => 0
384             )
385         );
386     else
387     {
388         $names = $this->request->getPost('name');
389         $scores = $this->request->getPost('score');
390         $c_tl = $this->request->getPost('c_time_limit');
391         $py_tl = $this->request->getPost('python_time_limit');
392         $java_tl = $this->request->getPost('java_time_limit');
393         $ml = $this->request->getPost('memory_limit');
394         $ft = $this->request->getPost('languages');
395         $dc = $this->request->getPost('diff_cmd');
396         $da = $this->request->getPost('diff_arg');
397         $data[ 'problems' ] = array();
398         $uo = $this->request->getPost('is_upload_only');
399         if ($uo === NULL)
400             $uo = array();
401         for ($i=0; $i<count($names); $i++){
402             array_push($data['problems'], array(
403                 'id' => $i+1,
404                 'name' => $names[$i],
405                 'score' => $scores[$i],
406                 'c_time_limit' => $c_tl[$i],
407                 'python_time_limit' => $py_tl[$i],
408                 'java_time_limit' => $java_tl[$i],
409                 'memory_limit' => $ml[$i],
410                 'allowed_languages' => $ft[$i],
411                 'diff_cmd' => $dc[$i],
412                 'diff_arg' => $da[$i],
413                 'is_upload_only' => in_array($i+1,$uo)?1:0,
414             ));
415         }
416     }
417 }
418
419 return view('pages/admin/add_assignment', $data);
420 }
421
422 // -----
423
424
425 /**
426 * Add/Edit assignment
427 */
428
429 private function _add()
430 {
431     // Check permission
432     if ($this->user->level <= 1) // permission denied
433         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
434
435     $this->validation->setRule('assignment_name', 'assignment_name', 'required|max_length[50]');
436     $this->validation->setRule('start_time', 'start_time', 'required');
437     $this->validation->setRule('finish_time', 'finish_time', 'required');
438     $this->validation->setRule('extra_time', 'extra_time', 'required');
439     $this->validation->setRule('participants', 'participants', 'string');
440     $this->validation->setRule('late_rule', 'coefficient_rule', 'required');
441     $this->validation->setRule('name.*', 'problem_name', 'required|max_length[50]');
442     $this->validation->setRule('score.*', 'problem_score', 'required|integer');
443     $this->validation->setRule('c_time_limit.*', 'C/C++_time_limit', 'required|integer');
444     $this->validation->setRule('python_time_limit.*', 'python_time_limit', 'required|integer');
445     $this->validation->setRule('java_time_limit.*', 'java_time_limit', 'required|integer');
446     $this->validation->setRule('memory_limit.*', 'memory_limit', 'required|integer');
447     $this->validation->setRule('languages.*', 'languages', 'required');
448     $this->validation->setRule('diff_cmd.*', 'diff_command', 'required');
449     $this->validation->setRule('diff_arg.*', 'diff_argument', 'required');
450
451     // Validate input data
452     if ( !$this->validation->withRequest($this->request)->run() ){
453         return FALSE;
454     }
455
456     // Preparing variables
457
458     if ($this->edit)
459         $the_id = $this->edit_assignment;
460     else
461         $the_id = $this->assignment_model->new_assignment_id();
462
463     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
464     $assignment_dir = "$assignments_root/assignment_{$the_id}";
465
466     // Adding/Editing assignment in database
467     if ( !$this->assignment_model->add_assignment($the_id, $this->edit))

```

```

468{
469    $this->messages[] = array(
470        'type' => 'error',
471        'text' => 'Error'.($this->edit?'updating':'adding').'_assignment.'
472    );
473    return FALSE;
474}
475
476$this->messages[] = array(
477    'type' => 'success',
478    'text' => 'Assignment'.($this->edit?'updated':'added').'_successfully.'
479);
480
481// Create assignment directory
482if( !file_exists($assignment_dir) ){
483    mkdir($assignment_dir, 0700, true);
484}
485
486// Upload Tests (zip file)
487$files = glob($assignments_root.'/*.zip');
488foreach($files as $file){
489    if(is_file($file)){
490        // Delete the given file
491        unlink($file);
492    }
493}
494$zip_uploaded = $this->request->getFile('tests_desc');
495if( $_FILES['tests_desc']['error'] === UPLOAD_ERR_NO_FILE ){
496    $this->messages[] = array(
497        'type' => 'notice',
498        'text' => "Notice: You did not upload any zip file for tests. If needed, upload by editing assignment."
499    );
500}
501elseif( $zip_uploaded->getExtension() != "zip" ){
502    $this->messages[] = array(
503        'type' => 'error',
504        'text' => "Error: Error uploading tests zip file: The filetype you are attempting to upload is not allowed."
505    );
506}
507else{
508    $zip_uploaded->move($assignments_root);
509    $this->messages[] = array(
510        'type' => 'success',
511        'text' => "Tests(zip file) uploaded successfully."
512    );
513}
514
515// Upload PDF File of Assignment
516$old_pdf_files = glob("$assignment_dir/*.pdf");
517$pdf_uploaded = $this->request->getFile('pdf');
518if( $_FILES['pdf']['error'] === UPLOAD_ERR_NO_FILE){
519    $this->messages[] = array(
520        'type' => 'notice',
521        'text' => "Notice: You did not upload any pdf file for assignment. If needed, upload by editing assignment."
522    );
523}
524elseif( $pdf_uploaded->getExtension() != "pdf" ){
525    $this->messages[] = array(
526        'type' => 'error',
527        'text' => "Error: Error uploading pdf file of assignment: The filetype you are attempting to upload is not allowed."
528    );
529}
530else{
531    $pdf_uploaded->move("$assignment_dir");
532    {
533        foreach($old_pdf_files as $old_name){
534            unlink($old_name);
535        }
536        $this->messages[] = array(
537            'type' => 'success',
538            'text' => 'PDF file uploaded successfully.'
539        );
540    }
541}
542
543
544// Extract Tests (zip file)
545if( $zip_uploaded->getClientOriginalExtension() == "zip" ){ // if zip file is uploaded
546    $this->unzip = new ZipArchive();
547    // Create a temp directory
548    $tmp_dir_name = "shj_tmp_directory";
549    $tmp_dir = "$assignments_root/$tmp_dir_name";
550    shell_exec("rm -rf $tmp_dir; mkdir $tmp_dir");
551
552    // Extract new test cases and descriptions in temp directory
553    $this->unzip->open("$assignments_root/".$zip_uploaded->getName());
554    $extract_result = $this->unzip->extractTo($tmp_dir);
555
556    // Remove the zip file
557    unlink("$assignments_root/".$zip_uploaded->getName());
558
559    if( $extract_result )
560    {
561        // Remove previous test cases and descriptions
562        shell_exec("cd $assignment_dir;" .
563                  ". rm -rf */in; rm -rf */out; rm -f */tester.cpp; rm -f */tester.executable;" .
564                  ". rm -f */desc.html; rm -f */desc.md; rm -f */*.pdf;");
565        if( glob("$tmp_dir/*.pdf") )
566    }

```

```

566         shell_exec("cd $assignment_dir; rm -f *.pdf");
567         // Copy new test cases from temp dir
568         shell_exec("cd $assignments_root; cp -R $tmp_dir_name/* $assignment_{$the_id};");
569         $this->messages[] = array(
570             'type' => 'success',
571             'text' => 'Tests_(zip_file)_extracted_successfully.'
572         );
573     }
574     else
575     {
576         $this->messages[] = array(
577             'type' => 'error',
578             'text' => 'Error:_Error_extracting_zip_archive.'
579         );
580         $this->messages[] = array(
581             'type' => 'error',
582             'text' => "_Zip_Extraction_Error:_" . $this->unzip->getStatusString(),
583         );
584     }
585     // Remove temp directory
586     shell_exec("rm -rf $tmp_dir");
587 }
588
589 // Create problem directories and parsing markdown files
590
591 for ($i=1; $i <= $this->request->getPost('number_of_problems'); $i++)
592 {
593     if ( ! file_exists("$assignment_dir/p{$i}")
594         mkdir("$assignment_dir/p{$i}", 0700);
595     elseif (file_exists("$assignment_dir/p{$i}/desc.md"))
596     {
597         $html = $this->parsedown->parse(file_get_contents("$assignment_dir/p{$i}/desc.md"));
598         file_put_contents("$assignment_dir/p{$i}/desc.html", $html);
599     }
600 }
601
602 return TRUE;
603 }
604
605
606 // -----
607
608
609 public function edit($assignment_id)
610 {
611
612     if ($this->user->level <= 1) // permission denied
613         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
614
615     $this->edit_assignment = $assignment_id;
616     $this->edit = TRUE;
617
618     // redirect to add function
619     return $this->add();
620 }
621
622
623
624 // -----
625
626
627 /**
628 * Check PDF File Availability
629 */
630
631 public function pdfCheck($assignment_id, $problem_id = NULL)
632 {
633     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
634     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
635     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
636
637     // Find pdf file
638     if ($problem_id === NULL || $problem_id === "null")
639         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/*.pdf";
640     else
641         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/p{$problem_id}/*.pdf";
642     $pdf_files = glob($pattern);
643
644     if ( ! $pdf_files )
645         $response = json_encode(array('status'=>FALSE));
646     elseif ( !$this->assignment_model->assignment_info($assignment_id)['open'] )
647         $response = json_encode(array('status'=>FALSE));
648     elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)['participants'],
649                 $this->user->username) )
650         $response = json_encode(array('status'=>FALSE));
651     elseif ( shj_now() > $finishtime + $extratime )
652         $response = json_encode(array('status'=>FALSE));
653     elseif ( shj_now() < $starttime )
654         $response = json_encode(array('status'=>FALSE));
655     else
656         $response = json_encode(array('status'=>TRUE));
657
658     echo $response;
659 }

```

Kode A.2: BaseController.php

```

1 <?php
2
3 namespace App\Controllers;
4
5 use App\Libraries\Parsedown;
6 use App\Libraries\Unzip;
7 use App\Libraries\Twig;
8 use CodeIgniter\Controller;
9 use CodeIgniter\HTTP\CLIRequest;
10 use CodeIgniter\HTTP\IncomingRequest;
11 use CodeIgniter\HTTP\RequestInterface;
12 use CodeIgniter\HTTP\ResponseInterface;
13 use Psr\Log\LoggerInterface;
14
15 /**
16  * Class BaseController
17  *
18  * BaseController provides a convenient place for loading components
19  * and performing functions that are needed by all your controllers.
20  * Extend this class in any new controllers:
21  *     class Home extends BaseController
22  *
23  * For security be sure to declare any new methods as protected or private.
24  */
25 abstract class BaseController extends Controller
26 {
27     /**
28      * Instance of the main Request object.
29      *
30      * @var CLIRequest|IncomingRequest
31      */
32     protected $request;
33
34     /**
35      * An array of helpers to be loaded automatically upon
36      * class instantiation. These helpers will be available
37      * to all other controllers that extend BaseController.
38      *
39      * @var array
40      */
41     protected $helpers = ['text','url','shj_helper','form','cookie','string','filesystem'];
42
43     /**
44      * Be sure to declare properties for any property fetch you initialized.
45      * The creation of dynamic property is deprecated in PHP 8.2.
46      */
47     protected $parsedown;
48     protected $twig;
49
50     /**
51      * Constructor.
52      */
53     public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
54     {
55         // Do Not Edit This Line
56         parent::initController($request, $response, $logger);
57
58         // Preload any models, libraries, etc, here.
59
60         // E.g.: $this->session = \Config\Services::session();
61         $appPaths = new \Config.Paths();
62         $appViewPaths = $appPaths->viewDirectory;
63         $this->parsedown = new Parsedown();
64         $this->twig = new Twig();
65     }
66 }

```

Kode A.3: Submissions.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Submissions.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\SettingsModel;
12 use App\Models\SubmitModel;
13 use App\Models\User;
14 use App\Models\UserModel;
15 use App\Libraries\Shj_pagination;
16 use App\Models\ScoreboardModel;
17 use PhpOffice\PhpSpreadsheet\Spreadsheet;
18 use PhpOffice\PhpSpreadsheet\IOFactory;
19 use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
20 use PhpOffice\PhpSpreadsheet\Style\Fill;
21 use PhpOffice\PhpSpreadsheet\Style\Border;
22 use PhpOffice\PhpSpreadsheet\Style\Alignment;
23
24 class Submissions extends BaseController
25 {
26
27     private $problems;

```

```

28| private $filter_user;
29| private $filter_problem;
30| private $page_number;
31|
32| protected $assignment_model;
33| protected $submit_model;
34| protected $user;
35| protected $validation;
36| protected $settings_model;
37| protected $user_model;
38| protected $scoreboard_model;
39| protected $uri;
40|
41| // -----
42|
43|
44|
45| public function __construct()
46| {
47|     $this->user = new User();
48|     $this->assignment_model = new AssignmentModel();
49|     $this->submit_model = new SubmitModel();
50|     $this->validation = \Config\Services::validation();
51|     $this->settings_model = new SettingsModel();
52|     $this->user_model = new UserModel();
53|     $this->problems = $this->assignment_model->all_problems($this->user->selected_assignment['id']);
54|     $this->uri = service('uri');
55|     $temp_inp = $this->uri->getSegments();
56|     $this->filter_user = $this->filter_problem = NULL;
57|     $this->page_number = 1;
58|     $input = [];
59|     foreach($temp_inp as $key => $value){
60|         if($key != (count($temp_inp)-1)){
61|             $input[$value] = $temp_inp[$key+1];
62|         }else{
63|             break;
64|         }
65|     }
66|     if (array_key_exists('user', $input) && $input['user'])
67|         if ($this->user->level > 0) // students are not able to filter submissions by user
68|             $this->filter_user = ctype_alnum($input['user'])? $input['user']:NULL;
69|     if (array_key_exists('problem', $input) && $input['problem'])
70|         $this->filter_problem = is_numeric($input['problem'])? $input['problem']:NULL;
71|     if (array_key_exists('page', $input) && $input['page'])
72|         $this->page_number = is_numeric($input['page'])? $input['page']:1;
73|
74| }
75|
76|
77|
78| // -----
79|
80|
81|
82| /**
83| * Uses PHPExcel library to generate excel file of submissions
84| */
85| private function _download_excel($view)
86| {
87|     if ( ! in_array($view, array('all', 'final')))
88|         exit;
89|
90|     $now = shj_now_str(); // current time
91|
92|     // Load PHPExcel library
93|     $phpspreadsheet = new Spreadsheet();
94|
95|     // Set document properties
96|     $phpspreadsheet->getProperties()->setCreator('SharIF_Judge')
97|         ->setLastModifiedBy('SharIF_Judge')
98|         ->setTitle('SharIF_Judge_Users')
99|         ->setSubject('SharIF_Judge_Users')
100|        ->setDescription("List_of_ShariF_Judge_Users_('.\$now.')");
101|
102|     // Name of the file sent to browser
103|     $output_filename = 'judge_'.$view.'_submissions';
104|
105|     // Set active sheet
106|     $phpspreadsheet->setActiveSheetIndex(0);
107|     $sheet = $phpspreadsheet->getActiveSheet();
108|
109|     // Add current assignment, time, username filter, and problem filter to document
110|     $sheet->fromArray(array('Assignment:', $this->user->selected_assignment['name']), null, 'A1', true);
111|     $sheet->fromArray(array('Time:', $now), null, 'A2', true);
112|     $sheet->fromArray(array('Username_Filter:', $this->filter_user?$this->filter_user:'No_filter'), null, 'A3', true);
113|     $sheet->fromArray(array('Problem_Filter:', $this->filter_problem?$this->filter_problem:'No_filter'), null, 'A4', true);
114|
115|     // Prepare header
116|     if ($this->user->level === 0)
117|         $header=array('Final','Problem','Submit_Time','Score','Delay_(HH:MM)','Coefficient','Final_Score','Language','Status')
118|             ;
119|     else{
120|         $header=array('Final','Submit_ID','Username','Name','Problem','Submit_Time','Score','Delay_(HH:MM)','Coefficient',
121|             'Final_Score','Language','Status');
122|         if ($view === 'final'){
123|             array_unshift($header, "#2");
124|             array_unshift($header, "#1");
125|         }
126|     }
127| }
```

```

125
126 // Add header to document
127 $sheet->fromArray($header, null, 'A6', true);
128 $highest_column = $sheet->getHighestColumn();
129
130 // Set custom style for header
131 $sheet->getStyle('A6:'.$highest_column.'6')->applyFromArray(
132     array(
133         'fill' => array(
134             'fillType' => Fill::FILL_SOLID,
135             'color' => array('rgb' => '173C45')
136         ),
137         'font' => array(
138             'bold' => true,
139             'color' => array('rgb' => 'FFFFFF'),
140             //'size' => 14
141         )
142     )
143 );
144
145 // Prepare data (in $rows array)
146 if ($view === 'final')
147     $items = $this->submit_model->get_final_submissions($this->user->selected_assignment['id'], $this->user->level, $this->user->username, NULL, $this->filter_user, $this->filter_problem);
148 else
149     $items = $this->submit_model->get_all_submissions($this->user->selected_assignment['id'], $this->user->level, $this->user->username, NULL, $this->filter_user, $this->filter_problem);
150
151 $names = $this->user_model->get_names();
152
153 $finish = strtotime($this->user->selected_assignment['finish_time']);
154 $i=0; $j=0; $un='';
155 $rows = array();
156 foreach ($items as $item){
157     $i++;
158     if ($item['username'] != $un)
159         $j++;
160     $un = $item['username'];
161
162     $pi = $this->problems[$item['problem']];
163
164     $pre_score = ceil($item['pre_score']*$pi['score']/10000);
165
166     $checked='';
167     if ($item['is_final'])
168         $checked='*';
169
170     $delay = strtotime($item['time'])-$finish;
171     if ($item['coefficient'] === 'error')
172         $final_score = 0;
173     else
174         $final_score = ceil($pre_score*$item['coefficient']/100);
175
176     if ($this->user->level === 0)
177         $row = array(
178             $checked,
179             $item['problem'].'.'.$pi['name'],
180             $item['time'],
181             $pre_score,
182             ($delay==0?'No_Delay':time_hhmm($delay)),
183             $item['coefficient'],
184             $final_score,
185             filetype_to_language($item['file_type']),
186             $item['status'],
187         );
188     else {
189         $row = array(
190             $checked,
191             $item['submit_id'],
192             $item['username'],
193             $names[$item['username']],
194             $item['problem'].'.'.$pi['name'],
195             $item['time'],
196             $pre_score,
197             ($delay==0?'No_Delay':time_hhmm($delay)),
198             $item['coefficient'],
199             $final_score,
200             filetype_to_language($item['file_type']),
201             $item['status'],
202         );
203     }
204     if ($view === 'final'){
205         array_unshift($row,$j);
206         array_unshift($row,$i);
207     }
208 }
209 array_push($rows, $row);
210
211 // Add rows to document
212 $sheet->fromArray($rows, null, 'A7', true);
213 // Add alternative colors to rows
214 for ($i=7; $i<count($rows)+7; $i++){
215     $sheet->getStyle('A'.$i.':'.$highest_column.$i)->applyFromArray(
216         array(
217             'fill' => array(
218                 'fillType' => Fill::FILL_SOLID,
219                 'color' => array('rgb' => (($i%2)?'F0F0F0':'FAFAFA'))
220             )
221         )
222     );
}

```

```

222         )
223     );
224 }
225
226 // Set text align to center
227 $sheet->getStyle( $sheet->calculateWorksheetDimension() )
228     ->getAlignment()
229     ->setHorizontal(Alignment::HORIZONTAL_CENTER);
230
231 // Making columns autosize
232 for ($i=2;$i<$count($header);$i++)
233     $sheet->getColumnDimension(chr(65+$i))->setAutoSize(true);
234
235 // Set Border
236 $sheet->getStyle('A7:'.$highest_column.$sheet->getHighestRow())->applyFromArray(
237     array(
238         'borders' => array(
239             'outline' => array(
240                 'borderStyle' => Border::BORDER_THIN,
241                 'color' => array('rgb' => '444444'),
242             ),
243         ),
244     )
245 );
246
247 // Send the file to browser
248
249 $ext = 'xlsx';
250 if ( ! class_exists('ZipArchive') ) // If class ZipArchive does not exist, export to excel5 instead of excel 2007
251     $ext = 'xls';
252
253 header('Content-Type:_application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
254 header('Content-Disposition:_attachment;filename="'.$output_filename.'.'.$ext.'"');
255 header('Cache-Control:_max-age=0');
256 $objWriter = IOFactory::createWriter($phpspreadsheet, ucfirst($ext));
257 $objWriter->save('php://output');
258 }
259
260
261
262
263 // -----
264
265
266
267
268 public function final_excel()
269 {
270     $this->download_excel('final');
271 }
272
273
274
275 public function all_excel()
276 {
277     $this->download_excel('all');
278 }
279
280
281
282 // -----
283
284
285
286
287
288 public function the_final()
289 {
290
291     if ( ! is_numeric($this->page_number) )
292         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
293
294     if ($this->page_number<1)
295         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
296
297     $config = array(
298         'base_url' => site_url('submissions/final'.($this->filter_user?'/user/'.$this->filter_user:'')).($this->filter_problem?
299             '/problem/'.$this->filter_problem:''),
300         'cur_page' => $this->page_number,
301         'total_rows' => $this->submit_model->count_final_submissions($this->user->selected_assignment['id'], $this->user->
302             level, $this->user->username, $this->filter_user, $this->filter_problem),
303         'per_page' => $this->settings_model->get_setting('results_per_page_final'),
304         'num_links' => 5,
305         'full_ul_class' => 'shj_pagination',
306         'cur_li_class' => 'current_page'
307     );
308
309     if ($config['per_page'] == 0)
310         $config['per_page'] = $config['total_rows'];
311     $shj_pagination = new Shj_pagination($config);
312
313     $submissions = $this->submit_model->get_final_submissions($this->user->selected_assignment['id'], $this->user->level,
314         $this->user->username, $this->page_number, $this->filter_user, $this->filter_problem);
315
316     $names = $this->user_model->get_names();
317
318     foreach ($submissions as &$item)
319     {
320         $item['name'] = $names[$item['username']];

```

```

318     $item['fullmark'] = ($item['pre_score'] == 10000);
319     $item['pre_score'] = ceil($item['pre_score']*$this->problems[$item['problem']]['score']/10000);
320     $item['delay'] = strtotime($item['time'])-strtotime($this->user->selected_assignment['finish_time']);
321     $item['language'] = filetype_to_language($item['file_type']);
322     if ($item['coefficient'] === 'error')
323         $item['final_score'] = 0;
324     else
325         $item['final_score'] = ceil($item['pre_score']*$item['coefficient']/100);
326     }
327
328
329     $data = array(
330         'view' => 'final',
331         'all_assignments' => $this->assignment_model->all_assignments(),
332         'problems' => $this->problems,
333         'submissions' => $submissions,
334         'excel_link' => site_url('submissions/final_excel'.($this->filter_user?'/user/'.$this->filter_user:'').($this->
335             filter_problem?'/problem/'.$this->filter_problem:'')),
336         'filter_user' => $this->filter_user,
337         'filter_problem' => $this->filter_problem,
338         'pagination' => $shj_pagination->create_links(),
339         'page_number' => $this->page_number,
340         'per_page' => $config['per_page'],
341         'selected' => "final_submissions",
342         'user' => $this->user,
343         'finish_time' => $this->user->selected_assignment['finish_time'],
344         'extra_time' => $this->user->selected_assignment['extra_time'],
345     );
346
347     return view('pages/submissions', $data);
348 }
349
350
351
352 // -----
353
354
355
356
357 public function all()
358 {
359
360     if ( ! is_numeric($this->page_number))
361         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
362
363     if ($this->page_number < 1)
364         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
365
366     $config = array(
367         'base_url' => site_url('submissions/all'.($this->filter_user?'/user/'.$this->filter_user:'').($this->filter_problem?'/
368             problem/'.$this->filter_problem:'')),
369         'cur_page' => $this->page_number,
370         'total_rows' => $this->submit_model->count_all_submissions($this->user->selected_assignment['id'], $this->user->level,
371             $this->user->username, $this->filter_user, $this->filter_problem),
372         'per_page' => $this->settings_model->get_setting('results_per_page_all'),
373         'num_links' => 5,
374         'full_ul_class' => 'shj_pagination',
375         'cur_li_class' => 'current_page'
376     );
377     if ($config['per_page']==0)
378         $config['per_page'] = $config['total_rows'];
379
380     $shj_pagination = new Shj_pagination($config);
381     $submissions = $this->submit_model->get_all_submissions($this->user->selected_assignment['id'], $this->user->level, $this-
382         >user->username, $this->page_number, $this->filter_user, $this->filter_problem);
383     $names = $this->user_model->get_names();
384
385     foreach ($submissions as &$item)
386     {
387         $item['name'] = $names[$item['username']];
388         $item['fullmark'] = ($item['pre_score'] == 10000);
389         $item['pre_score'] = ceil($item['pre_score']*$this->problems[$item['problem']]['score']/10000);
390         $item['delay'] = strtotime($item['time'])-strtotime($this->user->selected_assignment['finish_time']);
391         $item['language'] = filetype_to_language($item['file_type']);
392         if ($item['coefficient'] === 'error')
393             $item['final_score'] = 0;
394         else
395             $item['final_score'] = ceil($item['pre_score']*$item['coefficient']/100);
396     }
397
398     $data = array(
399         'view' => 'all',
400         'all_assignments' => $this->assignment_model->all_assignments(),
401         'problems' => $this->problems,
402         'submissions' => $submissions,
403         'excel_link' => site_url('submissions/all_excel'.($this->filter_user?'/user/'.$this->filter_user:'').($this->
404             filter_problem?'/problem/'.$this->filter_problem:'')),
405         'filter_user' => $this->filter_user,
406         'filter_problem' => $this->filter_problem,
407         'pagination' => $shj_pagination->create_links(),
408         'selected' => "all_submissions",
409         'user' => $this->user,
410         'finish_time' => $this->user->selected_assignment['finish_time'],
411         'extra_time' => $this->user->selected_assignment['extra_time'],
412     );
413
414     return view('pages/submissions', $data);
415 }

```

```

412
413
414
415
416 // -----
417
418
419
420
421 /**
422 * Used by ajax request (for selecting final submission)
423 */
424 public function select()
425 {
426     if ( ! $this->request->isAJAX() )
427         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
428
429     // Students cannot change their final submission after finish_time + extra_time
430     if ($this->user->level === 0)
431         if ( $hj_now() > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time'])
432         {
433             $json_result = array(
434                 'done' => 0,
435                 'message' => 'This_assignment_is_finished._You_CANNOT_change_your_final_submissions.'
436             );
437             $this->response->setHeader('Content-Type:_application/json;', 'charset=utf-8');
438             echo json_encode($json_result);
439             return;
440         }
441
442     $this->validation->setRule('submit_id', 'Submit_ID', 'integer|greater_than[0]');
443     $this->validation->setRule('problem', 'problem', 'integer|greater_than[0]');
444     $this->validation->setRule('username', 'Username', 'required|min_length[3]|max_length[20]|alpha_numeric');
445
446     if ($this->validation->withRequest($this->request)->run())
447     {
448         $username = $this->request->getPost('username');
449         if ($this->user->level === 0)
450             $username = $this->user->username;
451
452         $res = $this->submit_model->set_final_submission(
453             $username,
454             $this->user->selected_assignment['id'],
455             $this->request->getPost('problem'),
456             $this->request->getPost('submit_id')
457         );
458
459         if ($res) {
460             // each time a user changes final submission, we should update scoreboard of that assignment
461             $this->scoreboard_model = new ScoreboardModel();
462             $this->scoreboard_model->update_scoreboard($this->user->selected_assignment['id']);
463             $json_result = array('done' => 1);
464         }
465         else
466             $json_result = array('done' => 0, 'message' => 'Selecting_Final_Submission_Failed');
467     }
468     else
469         $json_result = array('done' => 0, 'message' => 'Input_Error');
470
471     $this->response->setHeader('Content-Type:_application/json;', 'charset=utf-8');
472     echo json_encode($json_result);
473 }
474
475
476 /**
477 * For "view code" or "view result" or "view log"
478 */
479 public function view_code()
480 {
481     if ( ! $this->request->isAJAX() )
482         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
483     $this->validation->setRule('type','type','_check_type');
484     $this->validation->setRule('username','username','required|min_length[3]|max_length[20]|alpha_numeric');
485     $this->validation->setRule('assignment','assignment','assignment','integer|greater_than[0]');
486     $this->validation->setRule('problem','problem','integer|greater_than[0]');
487     $this->validation->setRule('submit_id','submit_id','integer|greater_than[0]');
488
489     if($this->validation->withRequest($this->request)->run())
490     {
491         $submission = $this->submit_model->get_submission(
492             $this->request->getPost('username'),
493             $this->request->getPost('assignment'),
494             $this->request->getPost('problem'),
495             $this->request->getPost('submit_id')
496         );
497         if ($submission === FALSE)
498             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
499
500         $type = $this->request->getPost('type'); // $type is 'code', 'result', or 'log'
501
502         if ($this->user->level === 0 && $type === 'log')
503             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
504
505         if ($this->user->level === 0 && $this->user->username != $submission['username'])
506             exit('Don\'t_try_to_see_submitted_codes:');
507
508         if ($type === 'result')
509             $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');

```

```

510         "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/result-{
511             $submission['submit_id']].html";
512     elseif ($type === 'code')
513         $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
514         "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/{$submission['
515             file_name']}." . filetype_to_extension($submission['file_type']));
516     elseif ($type === 'log')
517         $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
518         "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/log-{$submission
519             ['submit_id']}";
520     else
521         $file_path = '/nowhere'; // This line is never reached!
522
523     $result = array(
524         'file_name' => $submission['main_file_name'] . '.' . filetype_to_extension($submission['file_type']),
525         'text' => file_exists($file_path) ? file_get_contents($file_path) : 'File Not Found'
526     );
527
528     if ($type === 'code') {
529         $result['lang'] = $submission['file_type'];
530         if ($result['lang'] == 'py2' || $result['lang'] == 'py3')
531             $result['lang'] = 'python';
532     }
533
534     $this->response->setContentType('application/json')->setBody(json_encode($result));
535     $this->response->send();
536
537 } else
538     exit('Are you trying to see other users\' codes?');
539
540
541
542 // -----
543
544
545
546
547 public function download_file()
548 {
549     $username = $this->uri->getSegment(3);
550     $assignment = $this->uri->getSegment(4);
551     $problem = $this->uri->getSegment(5);
552     $submit_id = $this->uri->getSegment(6);
553
554     $submission = $this->submit_model->get_submission(
555         $username,
556         $assignment,
557         $problem,
558         $submit_id
559     );
560     if ($submission === FALSE)
561         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
562
563     if ($this->user->level === 0 && $this->user->username != $submission['username'])
564         exit('Don\'t try to see submitted codes!');
565
566     $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
567     "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/{$submission['file_name']}." .
568         filetype_to_extension($submission['file_type']);
569
570     return $this->response->download(
571         "{$submission['file_name']}." . filetype_to_extension($submission['file_type']),
572         file_get_contents($file_path)
573     );
574 }
575
576 }
577
}

```

Kode A.4: Submit.php

```

1 <?php
2 /**
3 * Sharif Judge online judge
4 * @file Submit.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\QueueModel;
12 use App\Models\SettingsModel;
13 use App\Models\SubmitModel;
14 use App\Models\User;
15
16 class Submit extends BaseController
17 {
18
19     private $data; //data sent to view
20     private $assignment_root;
21     private $problems;
22     private $problem; //submitted problem id
23     private $filetype; //type of submitted file

```

```

24 private $ext; //uploaded file extension
25 private $file_name; //uploaded file name without extension
26 private $coefficient;
27
28 // -----
29 protected $queue_model;
30 protected $settings_model;
31 protected $assignment_model;
32 protected $submit_model;
33 protected $user;
34 protected $validation;
35
36 public function __construct()
37 {
38     $this->queue_model = new QueueModel();
39     $this->settings_model = new SettingsModel();
40     $this->assignment_model = new AssignmentModel();
41     $this->user = new User();
42     $this->validation = \Config\Services::validation();
43     $this->assignment_root = $this->settings_model->get_setting('assignments_root');
44     $this->problems = $this->assignment_model->all_problems($this->user->selected_assignment['id']);
45
46     helper('shj_helper');
47     $extra_time = $this->user->selected_assignment['extra_time'];
48     $delay = shj_now() - strtotime($this->user->selected_assignment['finish_time']);
49     ob_start();
50     if ( isset($this->user->selected_assignment['late_rule']) ? eval($this->user->selected_assignment['late_rule']) === FALSE
51         : '' )
52         $coefficient = "error";
53     if (!isset($coefficient))
54         $coefficient = "error";
55     ob_end_clean();
56     $this->coefficient = $coefficient;
57 }
58
59 // -----
60
61 public function _language_to_type($language)
62 {
63     $language = strtolower ($language);
64     switch ($language) {
65         case 'c': return 'c';
66         case 'c++': return 'cpp';
67         case 'python_2': return 'py2';
68         case 'python_3': return 'py3';
69         case 'java': return 'java';
70         case 'zip': return 'zip';
71         case 'pdf': return 'pdf';
72         case 'txt': return 'txt';
73         default: return FALSE;
74     }
75 }
76
77
78 // -----
79
80
81 public function _language_to_ext($language)
82 {
83     $language = strtolower ($language);
84     switch ($language) {
85         case 'c': return 'c';
86         case 'c++': return 'cpp';
87         case 'python_2': return 'py';
88         case 'python_3': return 'py';
89         case 'java': return 'java';
90         case 'zip': return 'zip';
91         case 'pdf': return 'pdf';
92         case 'txt': return 'txt';
93         default: return FALSE;
94     }
95 }
96
97
98 // -----
99
100
101 public function _match($type, $extension)
102 {
103     switch ($type) {
104         case 'c': return ($extension==='c'?TRUE:FALSE);
105         case 'cpp': return ($extension==='cpp'?TRUE:FALSE);
106         case 'py2': return ($extension==='py'?TRUE:FALSE);
107         case 'py3': return ($extension==='py'?TRUE:FALSE);
108         case 'java': return ($extension==='java'?TRUE:FALSE);
109         case 'zip': return ($extension==='zip'?TRUE:FALSE);
110         case 'pdf': return ($extension==='pdf'?TRUE:FALSE);
111         case 'txt': return ($extension==='txt'?TRUE:FALSE);
112     }
113 }
114
115
116
117 // -----
118
119
120 public function index()
121 {

```

```

122     $this->validation->setRule('problem', 'problem', 'required|integer|greater_than[0]', array('greater_than' => 'Select_a_%s.'));
123     $this->validation->setRule('language', 'language', 'required|check_language', array('_check_language' => 'Select_a_valid_%s.'));
124     if($this->request->is('post')){
125         if ($this->validation->withRequest($this->request)->run())
126         {
127             if ($this->upload())
128                 return redirect()->to('submissions/all');
129             else
130                 throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Error_Uploader_File:' . $this->validation->listErrors());
131         }
132     }
133
134     $this->data = array(
135         'all_assignments' => $this->assignment_model->all_assignments(),
136         'problems' => $this->problems,
137         'in_queue' => FALSE,
138         'coefficient' => $this->coefficient,
139         'upload_state' => '',
140         'problems_js' => '',
141         'error' => '',
142         'selected' => 'submit',
143         'user' => $this->user,
144         'finish_time' => $this->user->selected_assignment['finish_time'],
145         'extra_time' => $this->user->selected_assignment['extra_time'],
146         'validationError' => $this->validation
147     );
148     foreach ($this->problems as $problem)
149     {
150         $languages = explode(',', $problem['allowed_languages']);
151         $items="";
152         foreach ($languages as $language)
153         {
154             $items = $items . trim($language) . ",";
155         }
156         $items = substr($items,0,strlen($items)-1);
157         $this->data['problems_js'] .= 'shj.p[{$problem['id']}]=[$items];';
158     }
159     if ($this->user->selected_assignment['id'] == 0)
160         $this->data['error']= 'Please_select_an_assignment_first.';
161     elseif ($this->user->level == 0 && ! $this->user->selected_assignment['open'])
162         // if assignment is closed, non-student users (admin, instructors) still can submit
163         $this->data['error']= 'Selected_assignment_is_closed.';
164     elseif ($shj_now() < strtotime($this->user->selected_assignment['start_time']))
165         $this->data['error']= 'Selected_assignment_has_not_started.';
166     elseif ($shj_now() > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time']) // deadline = finish_time + extra_time
167         $this->data['error']= 'Selected_assignment_has_finished.';
168     elseif ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->username) )
169         $this->data['error']= 'You_are_not_registered_for_submitting.';
170     else
171         $this->data['error']= 'none';
172
173     return view('pages/submit', $this->data);
174 }
175
176
177 // -----
178
179 /**
180 * Saves submitted code and adds it to queue for judging
181 */
182 private function _upload()
183 {
184     $now = shj_now();
185     foreach($this->problems as $item)
186     {
187         if ($item['id'] == $this->request->getPost('problem'))
188         {
189             $this->problem = $item;
190             break;
191         }
192     }
193     $filetype = $this->language_to_type(strtolower(trim($this->request->getPost('language'))));
194     $this->ext = substr(strrchr($_FILES['userfile']['name'],'.'),1); // uploaded file extension
195     $this->file_name = basename($_FILES['userfile']['name'], ".{$this->ext}"); // uploaded file name without extension
196     if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $this->problem['id']) )
197         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_have_already_submitted_for_this_problem._Your_last_submission_is_still_in_queue.');
198     if ($this->user->level==0 && !$this->user->selected_assignment['open'])
199         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_been_closed.');
200     if ($now < strtotime($this->user->selected_assignment['start_time']))
201         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_not_started.');
202     if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time'])
203         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_finished.');
204     if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->username) )
205         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_are_not_registered_for_submitting.');
206     $filetypes = explode(",",$this->problem['allowed_languages']);
207     foreach ($filetypes as $filetype)
208     {
209         $filetype = $this->language_to_type(strtolower(trim($filetype)));
210     }
211     if ($_FILES['userfile']['error'] == 4)
212         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('No_file_chosen.');
213     if ( ! in_array($this->filetype, $filetypes))
214         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('This_file_type_is_not_allowed_for_this_problem.');
}

```

```

214     if ( ! $this->match($this->filetype, $this->ext) )
215         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('This_file_type_does_not_match_your_selected_
216                                         language.');
216     if ( ! preg_match('/^a-zA-Z0-9_\-(\.)+$/ ', $this->file_name) )
217         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Invalid_characters_in_file_name.');
218
219     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'. $this->user->selected_assignment['id'].'/p' . $this->problem['
220                                         id']. '/'.$this->user->username;
221     if ( !file_exists($user_dir))
222         mkdir($user_dir, 0700);
223
223     $config['max_size'] = $this->settings_model->get_setting('file_size_limit');
224     $config['file_name'] = $this->file_name.".".($this->user->selected_assignment['total_submits']+1).".". $this->ext;
225     $config['max_file_name'] = 20;
226     $this->validation->setRule('userfile', 'userfile', 'max_size[userfile,'.$config['max_size'].']', ['max_size' => 'The_file_
227                                         you_are_attempting_to_upload_is_larger_than_the_permitted_size.']);
228     if ($this->validation->withRequest($this->request)->run())
229     {
230         $results = $this->request->getFile('userfile');
231         $results->move($user_dir,$config['file_name']);
232         $result = preg_replace('/\.\[^\.]+\./', "", $results->getName());
233         $this->submit_model = new SubmitModel();
234
234         $submit_info = array(
235             'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
236             'username' => $this->user->username,
237             'assignment' => $this->user->selected_assignment['id'],
238             'problem' => $this->problem['id'],
239             'file_name' => $result,
240             'main_file_name' => $this->file_name,
241             'file_type' => $this->filetype,
242             'coefficient' => $this->coefficient,
243             'pre_score' => 0,
244             'time' => shj_now_str(),
245         );
246         if ($this->problem['is_upload_only'] == 0)
247         {
248             $this->queue_model->add_to_queue($submit_info);
249             process_the_queue();
250         }
251         else
252         {
253             $this->submit_model->add_upload_only($submit_info);
254         }
255
256         return TRUE;
257     }
258
259     return FALSE;
260 }
261
262
263 // -----
264
265 /**
266 * Load code from editor file
267 */
268 public function load($problem_id){
269     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'. $this->user->selected_assignment['id'].'/p' . $problem_id.'/'.
270                                         $this->user->username;
271     $file_path = $user_dir.'/.EDITOR_FILE_NAME.'.EDITOR_FILE_EXT;
272     $input_path = $user_dir.'/.EDITOR_IN_NAME.'.EDITOR_FILE_EXT;
273     $output_path = $user_dir.'/.EDITOR_OUT_NAME.'.EDITOR_FILE_EXT;
274
274     if(!write_file($input_path, '_')){}
275     if(!write_file($output_path, '_')){}
276     if (!file_exists($file_path)){
277         $response = json_encode(array('content'=>'', 'message'=>'No_saved_file'));
278     }
279     else{
280         $file_content = file_get_contents($file_path);
281         if ($file_content === FALSE){
282             $response = json_encode(array('content'=>'', 'message'=>'Unable_to_load'));
283         }
284         else{
285             addslashes($file_content);
286             $response = json_encode(array('content'=>$file_content, 'message'=>'Loaded'));
287         }
288     }
289     echo $response;
290 }
291
292
293 // -----
294
295 /**
296 * Save code to editor file and submit/execute if needed
297 */
298 public function save($type = FALSE){
299     $data = $_POST['code_editor'];
300     $problem_id = $_POST['problem_id'];
301     $language = $_POST['language'];
302
303     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'. $this->user->selected_assignment['id'].'/p' . $problem_id.'/'.
304                                         $this->user->username;
305     if ( !file_exists($user_dir)){
306         mkdir($user_dir, 0700);
307     }
308     $file_path = $user_dir.'/.EDITOR_FILE_NAME.'.EDITOR_FILE_EXT;

```

```

308     $input_path = $user_dir.'/.EDITOR_IN_NAME.' . EDITOR_FILE_EXT;
309
310     if (!write_file($file_path, $data)){
311         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_save'));
312         echo $response;
313     }
314     else{
315         $response = json_encode(array('status'=>TRUE, 'message'=>'Saved'));
316         if($type === FALSE){
317             echo $response;
318         }
319         else{
320             $now = shj_now();
321             if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $problem_id)){
322                 $response = json_encode(array('status'=>FALSE, 'message'=>'You_have_already_submitted_for_this_problem._Your_
323                 last_submission_is_still_in_queue.'));
324                 echo $response;
325             }
326             else if ($this->user->level==0 && !$this->user->selected_assignment['open']){
327                 $response = json_encode(array('status'=>FALSE, 'message'=>'Selected_assignment_has_been_closed.'));
328                 echo $response;
329             }
330             else if ($now < strtotime($this->user->selected_assignment['start_time'])){
331                 $response = json_encode(array('status'=>FALSE, 'message'=>'Selected_assignment_has_not_started.'));
332                 echo $response;
333             }
334             else if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['
335                 extra_time']){
336                 $response = json_encode(array('status'=>FALSE, 'message'=>'Selected_assignment_has_finished.'));
337                 echo $response;
338             }
339             else if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'], $this->user->
340                 username)){
341                 $response = json_encode(array('status'=>FALSE, 'message'=>'You_are_not_registered_for_submitting.'));
342                 echo $response;
343             }
344             else{
345                 if($type === 'submit'){
346                     $this->submit($data, $problem_id, $language, $user_dir);
347                 }
348                 else if($type === 'execute'){
349                     $editor_input = $_POST['editor_input'];
350                     if (!write_file($input_path, $editor_input)){
351                         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_write_input_file'));
352                         echo $response;
353                     }
354                 }
355             }
356         }
357     }
358 }
359
360 }
361
362 // -----
363
364 /**
365 * Add code to queue for judging
366 */
367 private function _submit($data, $problem_id, $language, $user_dir){
368     $file_type = $this->language_to_type(strtolower(trim($language)));
369     $file_ext = $this->language_to_ext(strtolower(trim($language)));
370     $file_name = EDITOR_FILE_NAME;
371     $file_fname = $file_name.'.'.$this->user->selected_assignment['total_submits']+1;
372     $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
373
374     foreach($this->problems as $item)
375     {
376         if ($item['id'] == $problem_id)
377         {
378             $this->problem = $item;
379             break;
380         }
381     }
382
383     if (!write_file($file_path, $data)){
384         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_submit'));
385     }
386     else{
387         $this->submit_model = new SubmitModel();
388
389         $submit_info = array(
390             'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
391             'username' => $this->user->username,
392             'assignment' => $this->user->selected_assignment['id'],
393             'problem' => $problem_id,
394             'file_name' => $file_fname,
395             'main_file_name' => $file_name,
396             'file_type' => $file_type,
397             'coefficient' => $this->coefficient,
398             'pre_score' => 0,
399             'time' => shj_now_str(),
400         );
401         if ($this->problem['is_upload_only'] == 0)
402         {
403             $this->queue_model->add_to_queue($submit_info);
404             process_the_queue();
405         }
406     }
407 }

```

```

404        }
405    }
406    {
407        $this->submit_model->add_upload_only($submit_info);
408    }
409
410    $response = json_encode(array('status'=>TRUE, 'message'=>"Submitted"));
411}
412echo $response;
413
414
415
416// -----
417
418/** 
419 * Add code to queue for execution only
420 */
421private function _execute($data, $problem_id, $language, $user_dir){
422    $file_type = $this->language_to_type(strtolower(trim($language)));
423    $file_ext = $this->language_to_ext(strtolower(trim($language)));
424    $file_name = EDITOR_FILE_NAME;
425    $file_fname = $file_name.'.'.$EDITOR_SUBMIT_ID;
426    $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
427    $output_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
428
429    if (!write_file($file_path, $data)){
430        $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_execute', 'debug'=>$file_path));
431    }
432    else{
433        $submit_info = array(
434            'submit_id' => EDITOR_SUBMIT_ID,
435            'username' => $this->user->username,
436            'assignment' => $this->user->selected_assignment['id'],
437            'problem' => $problem_id,
438            'file_name' => $file_name,
439            'main_file_name' => $file_name,
440            'file_type' => $file_type,
441            'coefficient' => $this->coefficient,
442            'pre_score' => 0,
443            'time' => shj_now_str(),
444        );
445
446        if($this->queue_model->add_to_queue_exec($submit_info)){
447            if (!write_file($output_path, 'Queueing...')){
448                $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_write_output_file'));
449            }
450            else{
451                process_the_queue();
452                $response = json_encode(array('status'=>TRUE, 'message'=>'Executing'));
453            }
454        }
455        else{
456            $response = json_encode(array('status'=>FALSE, 'message'=>'Still_in_queue'));
457        }
458    }
459    echo $response;
460}
461
462
463// -----
464
465/** 
466 * Load output file as execution result
467 */
468public function get_output($problem_id){
469    $user_dir = rtrim($this->assignment_root, '/').$this->user->selected_assignment['id'].'/p'.$problem_id.'/';
470    $this->user->username;
471    $file_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
472
473    if (!file_exists($file_path)){
474        $response = json_encode(array('status'=>FALSE, 'content'=>''));
475    }
476    else{
477        $file_content = file_get_contents($file_path);
478        if ($file_content === FALSE){
479            $response = json_encode(array('status'=>FALSE, 'content'=>''));
480        }
481        else{
482            $complete_status = strpos($file_content, 'Total_Execution_Time');
483            if($complete_status === FALSE){
484                $response = json_encode(array('status'=>FALSE, 'content'=>$file_content));
485            }
486            else{
487                $response = json_encode(array('status'=>TRUE, 'content'=>$file_content));
488            }
489        }
490    }
491    echo $response;
492}
493}

```

Lampiran kode *controller* lainnya dapat dilihat pada <https://github.com/filipussetioo/SharIF-JudgeCI4/tree/main/app/Controllers>.

LAMPIRAN B

KODE PROGRAM *MODELS*

Kode B.1: AssignmentModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Assignment_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7 namespace App\Models;
8
9 use App\Libraries\Parsedown;
10 use CodeIgniter\Model;
11 use App\Models\SettingsModel;
12 use App\Models\ScoreboardModel;
13
14 class AssignmentModel extends Model
15 {
16     protected $settings_model;
17     protected $scoreboard_model;
18     protected $request;
19     protected $parsedown;
20
21     public function __construct()
22     {
23         parent::__construct();
24         $this->settings_model = new SettingsModel();
25         $this->request = \Config\Services::request();
26     }
27
28     /**
29      * Add New Assignment to DB / Edit Existing Assignment
30      *
31      * @param $id
32      * @param bool $edit
33      * @return bool
34      */
35     public function add_assignment($id, $edit = FALSE)
36     {
37         $this->scoreboard_model = new ScoreboardModel();
38         // Start Database Transaction
39         $this->db->transStart();
40
41         $extra_items = explode('*', $this->request->getPost('extra_time'));
42         $extra_time = 1;
43         foreach($extra_items as $extra_item)
44         {
45             $extra_time *= $extra_item;
46         }
47
48         $archived_assignment = $archived_assignment = $this->request->getPost('archived_assignment')!=NULL ? 1 : 0;
49
50         $assignment = array(
51             'id' => $id,
52             'name' => $this->request->getPost('assignment_name'),
53             'problems' => $this->request->getPost('number_of_problems'),
54             'total_submits' => 0,
55             'open' => ($this->request->getPost('open')==NULL?0:1),
56             'scoreboard' => ($this->request->getPost('scoreboard')==NULL?0:1),
57             'javaexceptions' => ($this->request->getPost('javaexceptions')==NULL?0:1),
58             'description' => '', /* todo */
59             'start_time' => date('Y-m-d H:i:s', strtotime($this->request->getPost('start_time'))),
60             'finish_time' => date('Y-m-d H:i:s', strtotime($this->request->getPost('finish_time'))),
61             'extra_time' => $extra_time*60,
62             'late_rule' => $this->request->getPost('late_rule'),
63             'participants' => $this->request->getPost('participants'),
64             'archived_assignment' => $archived_assignment
65         );
66         if($edit)
67         {
68             $before = $this->db->table('assignments')->getWhere(['id'=>$id])->getRowArray();
69             unset($assignment['total_submits']);
70             $this->db->table('assignments')->where('id', $id)->update($assignment);
71             // each time we edit an assignment, we should update coefficient of all submissions of that assignment
72             if ($assignment['extra_time']==$before['extra_time'] OR $assignment['start_time']==$before['start_time'] OR
73                 $assignment['finish_time']==$before['finish_time'] OR $assignment['late_rule']==$before['late_rule'])
74                 $this->_update_coefficients($id, $assignment['extra_time'], $assignment['finish_time'], $assignment['late_rule']);
75         }
76     }
77 }
```

```

75|     else
76|         $this->db->table('assignments')->insert($assignment);
77|
78|     /* **** Adding problems to "problems" table **** */
79|
80|     //First remove all previous problems
81|     $this->db->table('problems')->delete(['assignment'=>$id]);
82|
83|     //Now add new problems:
84|     $names = $this->request->getPost('name');
85|     $scores = $this->request->getPost('score');
86|     $c_tl = $this->request->getPost('c_time_limit');
87|     $py_tl = $this->request->getPost('python_time_limit');
88|     $java_tl = $this->request->getPost('java_time_limit');
89|     $ml = $this->request->getPost('memory_limit');
90|     $ft = $this->request->getPost('languages');
91|     $dc = $this->request->getPost('diff_cmd');
92|     $da = $this->request->getPost('diff_arg');
93|     $uo = $this->request->getPost('is_upload_only');
94|     if ($uo === NULL)
95|         $uo = array();
96|     for ($i=1; $i<=$this->request->getPost('number_of_problems'); $i++)
97|     {
98|         $items = explode(',', $ft[$i-1]);
99|         $ft[$i-1] = '';
100|        foreach ($items as $item){
101|            $item = trim($item);
102|            $item2 = strtolower($item);
103|            $item = ucfirst($item2);
104|            if ($item2 === 'python2')
105|                $item = 'Python_2';
106|            elseif ($item2 === 'python3')
107|                $item = 'Python_3';
108|            elseif ($item2 === 'pdf')
109|                $item = 'PDF';
110|            $item2 = strtolower($item);
111|            if (! in_array($item2, array('c', 'c++', 'python_2', 'python_3', 'java', 'zip', 'pdf', 'txt')))
112|                continue;
113|            // If the problem is not Upload-Only, its language should be one of {C,C++,Python 2, Python 3,Java}
114|            if (! in_array($i, $uo) && ! in_array($item2, array('c', 'c++', 'python_2', 'python_3', 'java')) )
115|                continue;
116|            $ft[$i-1] .= $item . ",";
117|        }
118|        $ft[$i-1] = substr($ft[$i-1],0,strlen($ft[$i-1])-1); // remove last ','
119|        $problem = array(
120|            'assignment' => $id,
121|            'id' => $i,
122|            'name' => $names[$i-1],
123|            'score' => $scores[$i-1],
124|            'is_upload_only' => in_array($i,$uo)?1:0,
125|            'c_time_limit' => $c_tl[$i-1],
126|            'python_time_limit' => $py_tl[$i-1],
127|            'java_time_limit' => $java_tl[$i-1],
128|            'memory_limit' => $ml[$i-1],
129|            'allowed_languages' => $ft[$i-1],
130|            'diff_cmd' => $dc[$i-1],
131|            'diff_arg' => $da[$i-1],
132|        );
133|        $this->db->table('problems')->insert($problem);
134|    }
135|
136|    if ($edit)
137|    {
138|        // We must update scoreboard of the assignment
139|        $this->scoreboard_model->update_scoreboard($id);
140|    }
141|
142|    // Complete Database Transaction
143|    $this->db->transComplete();
144|
145|    return $this->db->transStatus();
146|
147|
148|
149|
150| // -----
151|
152|
153|
154| /**
155|  * Delete An Assignment
156|  *
157|  * @param $assignment_id
158|  */
159| public function delete_assignment($assignment_id)
160| {
161|     $this->db->transStart();
162|
163|     // Phase 1: Delete this assignment and its submissions from database
164|     $this->db->table('assignments')->delete(['id'=>$assignment_id]);
165|     $this->db->table('problems')->delete(['assignment'=>$assignment_id]);
166|     $this->db->table('submissions')->delete(['assignment'=>$assignment_id]);
167|
168|     $this->db->transComplete();
169|
170|     if ($this->db->transStatus())
171|     {
172|         // Phase 2: Delete assignment's folder (all test cases and submitted codes)
173|         $cmd = 'rm -rf '.rtrim($this->settings_model->get_setting('assignments_root'), '/').'/assignment_'.assignment_id;

```

```

174     shell_exec($cmd);
175 }
176
177
178
179
180 // -----
181
182
183
184 /**
185 * All Assignments
186 *
187 * Returns a list of all assignments and their information
188 *
189 * @return mixed
190 */
191 public function all_assignments()
192 {
193     $result = $this->db->table('assignments')->orderBy('id')->get()->getResultArray();
194     $assignments = [];
195     foreach ($result as $item)
196     {
197         $assignments[$item['id']] = $item;
198     }
199     return $assignments;
200 }
201
202
203
204 // -----
205
206
207
208 /**
209 * New Assignment ID
210 *
211 * Finds the smallest integer that can be used as id for a new assignment
212 *
213 * @return int
214 */
215 public function new_assignment_id()
216 {
217     $max = ($this->db->table('assignments')->selectMax('id', 'max_id')->get()->getRow()->max_id) + 1;
218
219     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
220     while (file_exists($assignments_root.'/'.$assignment_.$max)){
221         $max++;
222     }
223
224     return $max;
225 }
226
227
228
229 // -----
230
231
232
233 /**
234 * All Problems of an Assignment
235 *
236 * Returns an array containing all problems of given assignment
237 *
238 * @param $assignment_id
239 * @return mixed
240 */
241 public function all_problems($assignment_id)
242 {
243     $result = $this->db->table('problems')->orderBy('id')->getWhere(['assignment'=>$assignment_id])->getResultArray();
244     $problems = array();
245     foreach ($result as $row)
246     {
247         $problems[$row['id']] = $row;
248     }
249
250
251
252 // -----
253
254
255
256 /**
257 * Problem Info
258 *
259 * Returns database row for given problem (from given assignment)
260 *
261 * @param $assignment_id
262 * @param $problem_id
263 * @return mixed
264 */
265 public function problem_info($assignment_id, $problem_id)
266 {
267     return $this->db->table('problems')->getWhere(['assignment'=>$assignment_id, 'id'=>$problem_id])->getRowArray();
268 }
269
270
271
272 // -----

```

```

273
274
275
276     /**
277      * Assignment Info
278      *
279      * Returns database row for given assignment
280      *
281      * @param $assignment_id
282      * @return array
283      */
284      public function assignment_info($assignment_id)
285      {
286          $query = $this->db->table('assignments')->getWhere(['id'=>$assignment_id]);
287          if ($query->getNumRows() != 1){
288              return array(
289                  'id' => 0,
290                  'name' => 'Not_Selected',
291                  'finish_time' => 0,
292                  'extra_time' => 0,
293                  'problems' => 0
294              );
295          }else{
296              return $query->getRowArray();
297          }
298      }
299
300
301
302 // -----
303
304
305 /**
306  * Is Participant
307  *
308  * Returns TRUE if $username if one of the $participants
309  * Examples for participants: "ALL" or "user1, user2,user3"
310  *
311  * @param $participants
312  * @param $username
313  * @return bool
314  */
315      public function is_participant($participants, $username)
316      {
317          $participants = explode(',', $participants);
318          foreach ($participants as &$participant){
319              $participant = trim($participant);
320          }
321          if(in_array('ALL', $participants))
322              return TRUE;
323          if(in_array($username, $participants))
324              return TRUE;
325          return FALSE;
326      }
327
328
329
330 // -----
331
332
333
334 /**
335  * Increase Total Submits
336  *
337  * Increases number of total submits for given assignment by one
338  *
339  * @param $assignment_id
340  * @return mixed
341  */
342      public function increase_total_submits($assignment_id)
343      {
344          // Get total submits
345          $total = $this->db->table('assignments')->select('total_submits')->getWhere(['id'=>$assignment_id])->getRow()->
346          total_submits;
347          // Save total+1 in DB
348          $this->db->table('assignments')->where('id', $assignment_id)->update(['total_submits'=>($total+1)]);
349
350          // Return new total
351          return ($total+1);
352      }
353
354
355
356 // -----
357
358
359
360 /**
361  * Set Moss Time
362  *
363  * Updates "Moss Update Time" for given assignment
364  *
365  * @param $assignment_id
366  */
367      public function set_moss_time($assignment_id)
368      {
369          $now = shj_now_str();
370          $this->db->table('assignments')->where('id', $assignment_id)->update(['moss_update'=>$now]);

```

```

371     }
372
373
374
375 // -----
376
377
378 /**
379 * Get Moss Time
380 *
381 * Returns "Moss Update Time" for given assignment
382 *
383 * @param $assignment_id
384 * @return string
385 */
386
387 public function get_moss_time($assignment_id)
388 {
389     $query = $this->db->table('assignments')->select('moss_update')->getWhere(['id'=>$assignment_id]);
390     if($query->getNumRows() != 1) return 'Never';
391     return $query->getRow()->moss_update;
392 }
393
394
395 // -----
396
397
398 /**
399 * Save Problem Description
400 *
401 * Saves (Adds/Updates) problem description (html or markdown)
402 *
403 * @param $assignment_id
404 * @param $problem_id
405 * @param $text
406 * @param $type
407 */
408
409 public function save_problem_description($assignment_id, $problem_id, $text, $type)
410 {
411     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
412
413     if ($type === 'html')
414     {
415         // Remove the markdown code
416         if(file_exists("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md")){
417             unlink("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md");
418         }
419         // Save the html code
420         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.html", $text);
421     }
422     elseif ($type === 'md')
423     {
424         // We parse markdown using Parsedown library
425         $this->parsedown = new Parsedown;
426         // Save the markdown code
427         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md", $text);
428         // Convert markdown to html and save the html
429         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.html", $this->parsedown->parse(
430             $text));
431     }
432 }
433
434
435 // -----
436
437
438 /**
439 * Update Coefficients
440 *
441 * Each time we edit an assignment (Update start time, finish time, extra time, or
442 * coefficients rule), we should update coefficients of all submissions of that assignment
443 *
444 * This function is called from add_assignment($id, TRUE)
445 *
446 * @param $assignment_id
447 * @param $extra_time
448 * @param $finish_time
449 * @param $new_late_rule
450 */
451
452 private function _update_coefficients($assignment_id, $extra_time, $finish_time, $new_late_rule)
453 {
454     $submissions = $this->db->table('submissions')->getWhere(['assignment'=>$assignment_id])->getResultArray();
455
456     $finish_time = strtotime($finish_time);
457
458     foreach ($submissions as $i => $item) {
459         $delay = strtotime($item['time'])-$finish_time;
460         ob_start();
461         if ( eval($new_late_rule) === FALSE )
462             $coefficient = "error";
463         if (!isset($coefficient))
464             $coefficient = "error";
465         ob_end_clean();
466         $submissions[$i]['coefficient'] = $coefficient;
467     }
468 // For better performance, we update each 1000 rows in one SQL query

```

```

469 $size = count($submissions);
470 for ($i=0; $i<=$size-1)/1000; $i++) {
471     if ($this->db->dbdriver === 'postgre')
472         $query = 'UPDATE `'.$this->db->prefixTable('submissions').`_AS_t SET coefficient=_c.coeff FROM _values\n';
473     else
474         $query = 'UPDATE `'.$this->db->prefixTable('submissions').`_SET coefficient=_CASE\n';
475
476     for ($j=1000*$i; $j<1000*($i+1) && $j<$size; $j++){
477         $item = $submissions[$j];
478         if ($this->db->dbdriver === 'postgre'){
479             $query.=("{$assignment_id},{$item['problem']},{$item['username']},{$item['submit_id']},{$item['coefficient']}");
480             if ($j+1<1000*($i+1) && $j+1<$size )
481                 $query.="\n";
482         }
483         else
484             $query.= "WHEN assignment='{$assignment_id}' AND problem='{$item['problem']}' AND username='{$item['username']}' "
485             "AND submit_id='{$item['submit_id']}' THEN {$item['coefficient']}\n";
486     }
487
488     if ($this->db->dbdriver === 'postgre')
489         $query.= `_AS_c(assignment,problem,username,submit_id,coeff)\n"
490             ." WHERE t.assignment=c.assignment AND t.problem=c.problem AND t.username=c.username AND t.submit_id=c.submit_id;";
491     else
492         $query.= "ELSE coefficient\nEND\nWHERE assignment='{$assignment_id}';";
493     $this->db->query($query);
494 }
495
496 }
497 }

```

Kode B.2: ScoreboardModel.php

```

1 <?php
2 /**
3  * Sharif Judge online judge
4  * @file Scoreboard_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class ScoreboardModel extends Model
11 {
12     protected $assignment_model;
13     protected $settings_model;
14     protected $user_model;
15
16     public function __construct()
17     {
18         parent::__construct();
19     }
20
21
22 /**
23  * Generate Scoreboard
24  *
25  * Generates scoreboard for given assignment, from Final Submissions.
26  * This function is called by update_scoreboard
27  *
28  * @param int $assignment_id
29  * @return array
30  */
31 private function _generate_scoreboard($assignment_id)
32 {
33     $this->assignment_model = new AssignmentModel();
34     $this->settings_model = new SettingsModel();
35     $assignment = $this->assignment_model->assignment_info($assignment_id);
36     $submissions = $this->db->table('submissions')->getWhere(array('is_final' => 1 , 'assignment' => $assignment_id))->
37         getResultArray();
38     $total_score = array();
39     $penalty = array();
40     $users = array();
41     $start = strtotime($assignment['start_time']);
42     $submit_penalty = $this->settings_model->get_setting('submit_penalty');
43     $scores = array();
44     foreach ($submissions as $submission){
45
46         $pi = $this->assignment_model->problem_info($assignment_id, $submission['problem']);
47
48         $pre_score = ceil($submission['pre_score']*$pi['score']/10000);
49         if ($submission['coefficient'] === 'error')
50             $final_score = 0;
51         else
52             $final_score = ceil($pre_score+$submission['coefficient']/100);
53         $delay = strtotime($submission['time'])-$start;
54         $scores[$submission['username']][$submission['problem']]['score'] = $final_score;
55         $scores[$submission['username']][$submission['problem']]['time'] = $delay;
56
57         if ( ! isset($total_score[$submission['username']]))
58             $total_score[$submission['username']] = 0;
59         if ( ! isset($penalty[$submission['username']]))
60             $penalty[$submission['username']] = 0;
61
62         $total_score[$submission['username']] += $final_score;
63
64         $number_of_submissions = $this->db->table('submissions')->where(array(

```

```

64         'assignment' => $submission['assignment'],
65         'problem' => $submission['problem'],
66         'username' => $submission['username'],
67     ))->countAllResults();
68
69     $penalty[$submission['username']] += $delay + $number_of_submissions*$submit_penalty;
70     $users[] = $submission['username'];
71 }
72 $scoreboard = array(
73     'username' => array(),
74     'score' => array(),
75     'submit_penalty' => array()
76 );
77 $users = array_unique($users);
78 foreach($users as $username){
79     array_push($scoreboard['username'], $username);
80     array_push($scoreboard['score'], $total_score[$username]);
81     array_push($scoreboard['submit_penalty'], $penalty[$username]);
82 }
83 array_multisort(
84     $scoreboard['score'], SORT_NUMERIC, SORT_DESC,
85     $scoreboard['submit_penalty'], SORT_NUMERIC, SORT_ASC,
86     $scoreboard['username']
87 );
88 return array($scores, $scoreboard);
89 }
90
91 // -----
92
93
94 /**
95 * Update All Scoreboards
96 *
97 * Updates the cached scoreboard of all assignments.
98 * This function is called each time a user is deleted, or all submissions
99 * of a user is deleted.
100 */
101 public function update_scoreboards()
102 {
103     $assignments = $this->db->table('assignments')->select('id')->get()->getResultArray();
104     foreach ($assignments as $assignment){
105         $this->update_scoreboard($assignment['id']);
106     }
107 }
108
109 // -----
110
111
112
113 /**
114 * Update Scoreboard
115 *
116 * Updates the cached scoreboard of given assignment. Saves the html code of
117 * scoreboard table in database.
118 *
119 * This function is called after judging/rejudging a submission, and when one
120 * of these settings is changed for an assignment:
121 *
122 * TODO: Better Implementation (faster and with less db queries)
123 *
124 * - Extra Time
125 * - Start Time
126 * - Finish Time
127 * - Coefficient's Rule
128 * - Enable/Disable Scoreboard
129 *
130 * @param int $assignment_id
131 */
132 public function update_scoreboard($assignment_id)
133 {
134     $this->assignment_model = new AssignmentModel();
135     $this->user_model = new UserModel();
136     // If scoreboard is not enabled, do nothing
137     $scoreboard_enabled = $this->db->table('assignments')->select('scoreboard')->getWhere(array('id'=>$assignment_id))->getRow();
138     ()->scoreboard;
139
140     if ($assignment_id == 0 OR ! $scoreboard_enabled )
141         return;
142
143     // Generate the scoreboard
144     list ($scores, $scoreboard) = $this->generate_scoreboard($assignment_id);
145
146     // Generate the scoreboard's html code
147     // todo: Save Scoreboard as json (generate html at client side)
148     $all_problems = $this->assignment_model->all_problems($assignment_id);
149     $total_score = 0;
150     foreach($all_problems as $i)
151         $total_score += $i['score'];
152     $data = array(
153         'assignment_id' => $assignment_id,
154         'problems' => $all_problems,
155         'total_score' => $total_score,
156         'scores' => $scores,
157         'scoreboard' => $scoreboard,
158         'names' => $this->user_model->get_names(),
159     );
160
161     $scoreboard_table = view('pages/scoreboard_table', $data);

```

```

162    // Minify the scoreboard's html code
163    // $scoreboard_table = $this->output->minify($scoreboard_table, 'text/html');
164
165    // Save the scoreboard's html code in Database
166    $query = $this->db->table('scoreboard')->select('assignment')->getWhere(array('assignment'=>$assignment_id));
167    if ($query->getNumRows() == 0)
168        $this->db->table('scoreboard')->insert(array('assignment'=>$assignment_id, 'scoreboard'=>$scoreboard_table));
169    else
170        $this->db->table('scoreboard')->where('assignment', $assignment_id)->update(array('scoreboard'=>$scoreboard_table));
171
172    }
173
174    // -----
175
176
177    /**
178     * Get Cached Scoreboard
179     *
180     * Returns the cached scoreboard of given assignment as a html text
181     *
182     * @param int $assignment_id
183     * @return string
184     */
185    public function get_scoreboard($assignment_id)
186    {
187        $query = $this->db->table('scoreboard')->select('scoreboard')->getWhere(array('assignment'=>$assignment_id));
188
189        if ($query->getNumRows() != 1)
190            return 'Scoreboard_not_found';
191        else
192            return $query->getRow()->scoreboard;
193    }
194
195}

```

Kode B.3: UserModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file User_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9 use Dapphp\Radius\Radius;
10 use App\Models\ScoreboardModel;
11 use App\Models\SettingsModel;
12
13 class UserModel extends Model
14 {
15     protected $validation;
16     protected $config;
17     protected $email;
18     protected $settings_model;
19     protected $scoreboard_model;
20     protected $session;
21     protected $response;
22
23     public function __construct()
24     {
25         parent::__construct();
26         $this->validation = \Config\Services::validation();
27         $this->config = config('Secrets');
28         $this->settings_model = new SettingsModel();
29         $this->scoreboard_model = new ScoreboardModel();
30         $this->email = \Config\Services::email();
31     }
32
33     /**
34      * Have User
35      *
36      * Returns TRUE if there is a user with username $username in database
37      *
38      * @param $username
39      * @return bool
40      */
41
42     public function have_user($username)
43     {
44         $query = $this->db->table('users')->getWhere(['username'=>$username]);
45         if ($query->getNumRows() == 0)
46             return FALSE;
47         if ($username === $query->getRow()->username) // needed (because of utf8_general_ci [ci=case insensitive])
48             return TRUE;
49         return FALSE;
50     }
51
52
53     /**
54      * User ID to Username
55      *
56      * Converts user id to username (returns FALSE if user does not exist)
57      *
58
59
60

```

```

61     * @param $user_id
62     * @return bool
63     */
64     public function user_id_to_username($user_id)
65     {
66         if( ! is_numeric($user_id))
67             return FALSE;
68         $query = $this->db->table('users')->select('username')->getWhere(['id'=>$user_id]);
69         if ($query->getNumRows() == 0)
70             return FALSE;
71         return $query->getRow()->username;
72     }
73
74     // -----
75
76
77     /**
78     * Username to User ID
79     *
80     * Converts username to user id (returns FALSE if user does not exist)
81     *
82     * @param $username
83     * @return bool
84     */
85     public function username_to_user_id($username)
86     {
87         $query = $this->db->table('users')->select('id')->getWhere(['username'=>$username]);
88         if ($query->getNumRows() == 0)
89             return FALSE;
90         return $query->getRow()->id;
91     }
92
93
94     // -----
95
96
97     /**
98     * Have Email
99     *
100    * Returns TRUE if a user (except $username) with given email exists
101   *
102   * @param $email
103   * @param bool $username
104   * @return bool
105   */
106   public function have_email($email, $username = FALSE)
107   {
108       $query = $this->db->table('users')->getWhere(['email'=>$email]);
109       if ($query->getNumRows() >= 1){
110           if($username !== FALSE && $query->getRow()->username == $username)
111               return FALSE;
112           else
113               return TRUE;
114       }
115       return FALSE;
116   }
117
118
119     // -----
120
121
122     /**
123     * Add User
124     *
125     * Adds a single user
126     *
127     * @param $username
128     * @param $email
129     * @param $password
130     * @param $role
131     * @return bool|string
132     */
133     public function add_user($username, $email, $display_name, $password, $role)
134     {
135         if ( ! ctype_alnum($username) )
136             return 'Username_may_only_contain_alpha-numeric_characters.';
137         if (strlen($username) < 3 OR strlen($username) > 20 OR strlen($password) < 6 OR strlen($password) > 200)
138             return 'Username_or_password_length_error.';
139         if ($this->have_user($username))
140             return 'User_with_this_username_exists.';
141         if ($this->have_email($email))
142             return 'User_with_this_email_exists.';
143         if (strtolower($username) !== $username)
144             return 'Username_must_be_lowercase.';
145         $roles = array('admin', 'head_instructor', 'instructor', 'student');
146         if ( ! in_array($role, $roles))
147             return 'Users_role_is_not_valid.';
148
149         $user=[
150             'username' => $username,
151             'email' => $email,
152             'display_name' => $display_name,
153             'password' => password_hash($password,PASSWORD_BCRYPT),
154             'role' => $role
155         ];
156         $this->db->table('users')->insert($user);
157         return TRUE; //success
158     }
159

```

```

160
161 // -----
162
163 /**
164 * Add Users
165 *
166 * Adds multiple users
167 *
168 * @param $text
169 * @param $send_mail
170 * @param $delay
171 * @return array
172 */
173
174 public function add_users($text, $send_mail, $delay)
175 {
176
177     $lines = preg_split('/\r?\n|\n?\r/', $text);
178     $users_ok = array();
179     $users_error = array();
180
181     // loop over lines of $text :
182     foreach ($lines as $line)
183     {
184         $line = trim($line);
185
186         if (strlen($line) == 0 OR $line[0] == '#')
187             continue; //ignore comments and empty lines
188
189         $parts = preg_split('/,+/', $line);
190         if (count($parts) == 5){
191             if (strtolower(substr($parts[3], 0, 6)) == 'random')
192             {
193                 // generate random password
194                 $len = trim(substr($parts[3], 6), '[]');
195                 if (is_numeric($len)){
196                     $this->load->helper('string');
197                     $parts[3] = shj_random_password($len);
198                 }
199             }
200
201             $result = $this->add_user($parts[0], $parts[1], $parts[2], $parts[3], $parts[4]);
202
203             if ($result === TRUE)
204                 array_push($users_ok, array($parts[0], $parts[1], $parts[2], $parts[3], $parts[4]));
205             else
206                 array_push($users_error, array($parts[0], $parts[1], $parts[2], $parts[3], $parts[4], $result));
207         }
208         else{
209             array_push($users_error, array($parts[0], $parts[1], $parts[2], $parts[3], $parts[4], 'Wrong_Format'));
210             continue; //ignore lines that not contain 5 parts
211         }
212     } // end of loop
213
214     if ($send_mail)
215     {
216         // sending usernames and passwords by email
217         $config = [
218             'mailtype' => 'html',
219             'charset'  => 'iso-8859-1'
220         ];
221         $this->email->initialize($this->config->shj_mail);
222         $this->email->setNewline("\r\n");
223         $count_users = count($users_ok);
224         $counter = 0;
225         foreach ($users_ok as $user)
226         {
227             $counter++;
228             $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('
229                                         mail_from_name'));
230             $this->email->setTo($user[1]);
231             $this->email->setSubject('SharIF_Judge_Username_and_Password');
232             $text = $this->settings_model->get_setting('add_user_mail');
233             $text = str_replace('{SITE_URL}', base_url(), $text);
234             $text = str_replace('{ROLE}', $user[4], $text);
235             $text = str_replace('{USERNAME}', $user[0], $text);
236             $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
237             $text = str_replace('{LOGIN_URL}', base_url(), $text);
238             $this->email->setMessage($text);
239             $this->email->send();
240             if ($counter < $count_users)
241                 sleep($delay);
242         }
243     }
244
245     return array($users_ok, $users_error);
246
247 }
248
249 // -----
250
251 /**
252 * Delete User
253 *
254 * Deletes a user with given user id
255 * Returns TRUE (success) or FALSE (failure)
256
257

```

```

258  *
259  * @param $user_id
260  * @return bool
261  */
262 public function delete_user($user_id)
263 {
    $this->db->transStart();
264
    $username = $this->user_id_to_username($user_id);
    if ($username === FALSE)
        return FALSE;
    $this->db->table('users')->delete(['id'=>$user_id]);
    $this->db->table('submissions')->delete(['username' => $username]);
    // each time we delete a user, we should update all scoreboards
    $this->scoreboard_model->update_scoreboards();
273
    $this->db->transComplete();
274
    if ($this->db->transStatus()) {
        // Delete submitted files
        shell_exec("cd {$this->settings_model->get_setting('assignments_root')}; rm -r /*/{$username};");
        return TRUE; // success
    }
    return FALSE; // failure
}
282
283
284 // -----
285
286
287 /**
288 * Delete Submissions
289 *
290 * Deletes all submissions of user with given user id
291 * Returns TRUE (success) or FALSE (failure)
292 *
293 * @param $user_id
294 * @return bool
295 */
296 public function delete_submissions($user_id)
297 {
    $this->db->transStart();
298
    $username = $this->user_id_to_username($user_id);
    if ($username === FALSE)
        return FALSE;
    // delete all submissions from database
    $this->db->table('submissions')->delete(['username'=>$username]);
    // each time we delete a user's submissions, we should update all scoreboards
    $this->scoreboard_model->update_scoreboards();
307
    $this->db->transComplete();
308
    if ($this->db->transStatus()) {
        // delete all submitted files
        shell_exec("cd {$this->settings_model->get_setting('assignments_root')}; rm -r /*/{$username};");
        return TRUE; // success
    }
315
    return FALSE; // failure
}
317
318
319 // -----
320
321
322
323 /**
324 * Validate User
325 *
326 * Returns TRUE if given username and password is valid for login
327 *
328 * @param $username
329 * @param $password
330 * @return bool
331 */
332 public function validate_user($username, $password)
333 {
    $query = $this->db->table('users')->getWhere(['username' => $username]);
    if ($query->getNumRows() != 1)
        return FALSE;
    if ($query->getRow()->username !== $username) // needed (because of utf8_general_ci [ci=case insensitive])
        return FALSE;
    if (password_verify($password, $query->getRow()->password))
        return TRUE;
341
    if($this->config->shj_authenticate == 'radius') {
        $client = new Radius();
        $client->setServer($this->config->shj_radius['server']) // RADIUS server address
            ->setSecret($this->config->shj_radius['secret']);
        if($client->accessRequest($username, $password))
            return TRUE;
    } else if($this->config->shj_authenticate == 'ldap') {
        try {
            $ldap = new \Adldap\Adldap();
            $ldap->addProvider($this->config->shj_ldap);
352
            $guard = $ldap->getDefaultProvider()->getGuard();
            return $guard->attempt($username, $password);
        } catch (\Exception $e) {
356
    }
}

```

```

357         return FALSE;
358     }
359 }
360 return FALSE;
361 }
362 }
363 }
364 // -----
365
366 /**
367 * Selected Assignment
368 *
369 * Returns selected assignment by given username
370 * @param $username
371 * @return mixed
372 */
373 public function selected_assignment($username)
374 {
375     $this->session = session();
376     $query = $this->db->table('users')->select('selected_assignment')->getWhere(['username'=>$username]);
377     if ($query->getNumRows() != 1){//logout
378         $this->session->destroy();
379         return redirect()->to('login');
380     }
381     return $query->getRow()->selected_assignment;
382 }
383
384
385
386 // -----
387
388 /**
389 * Get Display Name
390 *
391 * Returns name of the user with given username
392 *
393 * @return array
394 */
395 public function get_names()
396 {
397     $query = $this->db->table('users')->select('username,_display_name')->get();
398     $tmp = $query->getResultArray();
399     $result = array();
400     foreach ($tmp as $row)
401         $result[$row['username']] = $row['display_name'];
402     return $result;
403 }
404
405
406 // -----
407
408 /**
409 * Update Profile
410 *
411 * Updates User Profile (Name, Email, Password, Role)
412 *
413 * @param $user_id
414 * @return bool
415 */
416 public function update_profile($user_id,$email,$password,$role,$display_name)
417 {
418     $query = $this->db->table('users')->getWhere(['id'=>$user_id]);
419     if ($query->getNumRows() != 1)
420         return FALSE;
421     $the_user = $query->getRow();
422     $username = $the_user->username;
423
424     $display_name = $display_name;
425     $locked = $this->settings_model->get_setting('lock_student_display_name');
426
427     if ($locked == 1) {
428         $display_name = $the_user->display_name;
429     }
430
431     $user=array(
432         'display_name' => $display_name,
433         'email' => $email
434     );
435
436     // if a role is provided, change the role
437     // (only admins are able to provide a role)
438     if ($role !== NULL)
439         $user['role'] = $role;
440
441     // if a password is provided, change the password:
442     if ($password != ''){
443         $user['password'] = password_hash($password, PASSWORD_BCRYPT);
444     }
445
446     $this->db->table('users')->where('username', $username)->update($user);
447 }
448
449 // -----
450
451
452
453
454 /**
455 */

```

```

456     * Send Password Reset Mail
457     *
458     * Generates a password reset key and sends an email containing the link
459     * for resetting password (in case of password lost)
460     *
461     * @param $email
462     */
463     public function send_password_reset_mail($email)
464     {
465         // exit if $email is invalid:
466         if ( !$this->have_email($email) )
467             return;
468
469         // generate a random password reset key:
470         helper('string');
471         $passchange_key = random_string('alnum', 50);
472
473         // save the key in users table:
474         $now = shj_now();
475         $this->db->table('users')->where('email', $email)->update(array('passchange_key'=>$passchange_key, 'passchange_time'=>date
476             ('Y-m-d H:i:s', $now)));
477
478         // send the email:
479         $this->response = response();
480
481         $this->email->initialize($this->config->shj_mail);
482         $this->email->setNewline("\r\n");
483         $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'
484             ));
485         $this->email->setTo($email);
486         $this->email->setSubject('Password_Reset');
487         $text = $this->settings_model->get_setting('reset_password_mail');
488         $text = str_replace('{SITE_URL}', base_url(), $text);
489         $text = str_replace('{RESET_LINK}', site_url('login/reset/'.$passchange_key), $text);
490         $text = str_replace('{VALID_TIME}', '1_hour', $text); // links are valid for 1 hour
491         $this->email->setMessage($text);
492         if(!$this->email->send())
493             dd($this->email->printDebugger());
494     }
495
496     // -----
497
498     /**
499      * Password Reset Key Is Valid
500      *
501      * Returns TRUE if the given password reset key is valid
502      * And returns an error message if key is invalid
503      *
504      * @param $passchange_key
505      * @return bool|string
506      */
507     public function passchange_is_valid($passchange_key)
508     {
509         $query = $this->db->table('users')->select('passchange_time')->getWhere(['passchange_key'=>$passchange_key]);
510         if ($query->getNumRows() != 1)
511             return 'Invalid_password_reset_link.';
512         $time = strtotime($query->getRow()->passchange_time);
513         $now = shj_now();
514         if ($now-$time > 3600 OR $now-$time < 0) // reset link is valid for 1 hour
515             return 'The_link_is_expired.';
516         return TRUE;
517     }
518
519
520     // -----
521
522     /**
523      * Reset Password
524      *
525      * Resets password for given password reset key (in case of lost password)
526      *
527      * @param $passchange_key
528      * @param $newpassword
529      * @return bool
530      */
531     public function reset_password($passchange_key, $newpassword)
532     {
533         $query = $this->db->table('users')->getWhere(array('passchange_key'=>$passchange_key));
534         if ($query->getNumRows() != 1)
535             return FALSE; //failure
536         $this->db->table('users')->where('username', $query->getRow()->username)->update(array('passchange_key'=>'', 'password' =>
537             password_hash($newpassword,PASSWORD_BCRYPT)));
538         return TRUE; //success
539     }
540
541
542     // -----
543
544     /**
545      * Get All Users
546      *
547      * Returns an array of all users (for Users page)
548      *
549      * @return mixed
550      */
551 
```

```

552     */
553     public function get_all_users()
554     {
555         return $this->db->table('users')->orderBy('role', 'asc')->orderBy('id')->get()->getResultArray();
556     }
557
558     // -----
559
560
561
562     /**
563      * Get User
564      *
565      * Returns database row for given user id
566      *
567      * @param $user_id
568      * @return bool
569      */
570     public function get_user($user_id)
571     {
572         $query = $this->db->table('users')->getWhere(array('id'=>$user_id));
573         if ($query->getNumRows() != 1)
574             return FALSE;
575         return $query->getRow();
576     }
577
578     // -----
579
580
581
582     /**
583      * Update Login Time
584      *
585      * Updates First Login Time and Last Login Time for given username
586      *
587      */
588     public function update_login_time($username)
589     {
590         helper('shj_helper');
591         $now = shj_now_str();
592
593         $first_login = $this->db->table('users')->select('first_login_time')->getWhere(array('username'=>$username))->getRow() ->
594             first_login_time;
595         if ($first_login === NULL)
596             $this->db->table('users')->where('username', $username)->update(array('first_login_time'=>$now));
597
598         $this->db->table('users')->where('username', $username)->update(array('last_login_time'=>$now));
599     }
600
601     // -----
602 }

```

Lampiran kode *model* lainnya dapat dilihat pada <https://github.com/filipussetioo/SharIF-JudgeCI4/tree/main/app/Models>.

LAMPIRAN C

KODE PROGRAM VIEWS

Kode C.1: add_assignment.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: add_assignment.php
4 # author: Filipus Setio Nugroho <filipussetio@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?><?= $edit ? 'fa-edit' : 'fa-plus-square' ?><?= $this->endSection() ?>
8 <?= $this->section('title') ?><?= $edit ? 'Edit' : 'Add' ?><?= $this->endSection() ?>
9 <?= $this->section('head_title') ?><?= $edit ? 'Edit' : 'Add' ?><?= $this->endSection() ?>
10
11
12 <?= $this->section('other_assets') ?>
13 <script type='text/javascript' src="= _base_url('assets/js/taboverride.min.js') ?&gt;"&gt;&lt;/script&gt;
14 &lt;script&gt;
15     $(document).ready(function(){
16         tabOverride.set(document.getElementsByName('textarea'));
17         $switch = false;
18         $("textarea").keyup(function (e) {
19             if (e.which==27){
20                 tabOverride.set(document.getElementsByName('textarea'),$switch);
21                 $switch = !$switch;
22             }
23         });
24     });
25 &lt;/script&gt;
26 &lt;script type="text/javascript" src="<?= _base_url('assets/js/jquery-ui-timepicker-addon.js') ?&gt;"&gt;&lt;/script&gt;
27 &lt;script&gt;
28     shj.num_of_problems=&lt;?= count($problems) ?&gt;;
29     shj.row='&lt;tr&gt;&lt;td&gt;PID&lt;/td&gt;\n'
30     &lt;td&gt;&lt;input_aria-label="Problem_Name" type="text" name="name[]" class="sharif_input_short" value="Problem_"/&gt;&lt;/td&gt;\n
31     &lt;td&gt;&lt;input_aria-label="Score" type="text" name="score[]" class="sharif_input_tiny2" value="100"/&gt;&lt;/td&gt;\n
32     &lt;td&gt;&lt;input_aria-label="Time_Limit_for_C" type="text" name="c_time_limit[]" class="sharif_input_tiny2" value="500"/&gt;&lt;/td&gt;\n
33     &lt;td&gt;&lt;input_aria-label="Time_Limit_for_Python" type="text" name="python_time_limit[]" class="sharif_input_tiny2" value
34     ="1500"/&gt;&lt;/td&gt;\n
35     &lt;td&gt;&lt;input_aria-label="Time_Limit_for_Java" type="text" name="java_time_limit[]" class="sharif_input_tiny2" value="2000"/&gt;&lt;/td&gt;
36     &gt;\n
37     &lt;td&gt;&lt;input_aria-label="Memory_Limit" type="text" name="memory_limit[]" class="sharif_input_tiny" value="50000"/&gt;&lt;/td&gt;\n
38     &lt;td&gt;&lt;input_aria-label="Allowed_Languages" type="text" name="languages[]" class="sharif_input_short2" value="C,C++,Python_2,
39     Python_3,Java"/&gt;&lt;/td&gt;\n
40     &lt;td&gt;&lt;input_aria-label="Diff_Command" type="text" name="diff_cmd[]" class="sharif_input_tiny" value="diff"/&gt;&lt;/td&gt;\n
41     &lt;td&gt;&lt;input_aria-label="Diff_Argument" type="text" name="diff_arg[]" class="sharif_input_tiny" value=".bB"/&gt;&lt;/td&gt;\n
42     &lt;td&gt;&lt;input_aria-label="Upload_Only" type="checkbox" name="is_upload_only[]" class="check" value="PID"/&gt;&lt;/td&gt;\n
43     &lt;td&gt;&lt;button class="delete_problem" type="button" aria-label="Delete_Problem"&gt;&lt;i class="fa fa-times-circle fa-lg_color1_pointer
44     "&gt;&lt;/i&gt;&lt;/button&gt;&lt;/td&gt;\n
45 &lt;/tr&gt;';
46     $(document).ready(function(){
47         $('#add').click(function(){
48             $('#problems_table&gt;tbody').append(shj.row.replace(/PID/g, (shj.num_of_problems+1)));
49             shj.num_of_problems++;
50             $('#nop').attr('value', shj.num_of_problems);
51         });
52         $("#form_a_archived_assignment").click(function(){
53             if ($('#form_a_archived_assignment').is(':checked')) {
54                 $('#start_time').val('1970-01-02_00:00:00');
55                 $('#finish_time').val('2038-01-18_00:00:00');
56                 $('#form_extra_time').val('0');
57             } else{
58                 $('#start_time').val('');
59                 $('#finish_time').val('');
60                 $('#form_extra_time').val('');
61             });
62             $(document).on('click', '.delete_problem', function(){
63                 if (shj.num_of_problems==1) return;
64                 var row = $(this).parents('tr');
65                 row.remove();
66                 var i = 0;
67                 $('#problems_table&gt;tbody').children('tr').each(function(){
68                     i++;
69                     $(this).children(':first').html(i);
70                     $(this).find('[type="checkbox"]').attr('value',i);
71                 });
72                 shj.num_of_problems--;
73                 $('#nop').attr('value',shj.num_of_problems);
74             });
75         });
76     });
77 </pre

```

```

72    });
73    $('#start_time').datetimepicker({
74        timeFormat: 'HH:mm:ss'
75    });
76    $('#finish_time').datetimepicker({
77        timeFormat: 'HH:mm:ss'
78    });
79 });
80 </script>
81 <?= $this->endSection() ?>
82
83
84 <?= $this->section('title_menu') ?>
85 <span class="title_menu_item">
86     <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md" target="_blank"><i class="fa fa-question-circle color1"></i> Help</a>
87 </span>
88 <?= $this->endSection() ?>
89
90
91
92 <?= $this->section('main_content') ?>
93 <?php $msgclasses = ['success' => 'shj_g', 'notice' => 'shj_o', 'error' => 'shj_r'] ?>
94 <?php foreach($messages as $message): ?>
95     <p class="<?= $msgclasses[$message->type] ?>"><? $message->text ?></p>
96 <?php endforeach ?>
97
98 <?php if($edit): ?>
99 <p>
100     <i class="fa fa-info-circle fa-lg color8"></i> If you don't want to change tests or pdf file, just do not upload its file.
101 </p>
102 <?php endif ?>
103
104 <?= form_open_multipart($edit ? '_assignments/edit/' . $edit_assignment['id'] : '_assignments/add') ?>
105 <div class="panel_left">
106     <input type="hidden" name="number_of_problems" id="nop" value="<?= $edit ? $edit_assignment['problems'] : count($problems) ?>" />
107     <p class="input_p">
108         <label for="form_a_name">Assignment Name</label>
109         <input id="form_a_name" type="text" name="assignment_name" class="sharif_input_medium" value="<?= $edit ? $edit_assignment['name'] : set_value('assignment_name') ?>" />
110         <div class="shj_error"><?=$validationError->hasError('assignment_name') ?> $validationError->getError('assignment_name') : '' ?></div>
111     </p>
112     <p class="input_p">
113         <label for="start_time">Start Time</label>
114         <input id="start_time" type="text" name="start_time" class="sharif_input_medium" value="<?= $edit ? date('m/d/Y H:i:s', strtotime($edit_assignment['start_time'])) : set_value('start_time') ?>" />
115         <div class="shj_error"><?=$validationError->hasError('start_time') ?> $validationError->getError('start_time') : '' ?></div>
116     </p>
117     <p class="input_p">
118         <label for="finish_time">Finish Time</label>
119         <input id="finish_time" type="text" name="finish_time" class="sharif_input_medium" value="<?= $edit ? date('m/d/Y H:i:s', strtotime($edit_assignment['finish_time'])) : set_value('finish_time') ?>" />
120         <div class="shj_error"><?=$validationError->hasError('finish_time') ?> $validationError->getError('finish_time') : '' ?></div>
121     </p>
122     <p class="input_p_clear">
123         <label for="form_extra_time">
124             Extra Time (minutes)<br>
125             <span class="form_comment">Extra time for late submissions.</span>
126         </label>
127     </p>
128     <input id="form_extra_time" type="text" name="extra_time" class="sharif_input_medium" value="<?= $edit ? $twig->extra_time_formatter($edit_assignment['extra_time']) : set_value('extra_time') ?>" />
129     <div class="shj_error"><?=$validationError->hasError('extra_time') ?> $validationError->getError('extra_time') : '' ?></div>
130     </p>
131     <p class="input_p_clear">
132         <label for="form_participants">Participants<br>
133             <span class="form_comment">Enter username of participants here (comma separated).<br>
134             Only these users are able to submit. You can use keyword "ALL".</span>
135             <span class="form_comment_clear">Press "esc" to enable/disable tab indent</span>
136         </label>
137         <textarea id="form_participants" name="participants" rows="5" class="sharif_input_medium"><?= $edit ? $edit_assignment['participants'] : set_value('participants', 'ALL') ?></textarea>
138     </p>
139     <p class="input_p_clear">
140         <label for="form_tests_desc">Tests and Descriptions (zip file)<br>
141             <span class="form_comment">
142                 <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/tests_structure.md" target="_blank">Use this structure</a>
143             </span>
144         </label>
145         <input id="form_tests_desc" type="file" name="tests_desc" class="sharif_input_medium"/>
146     </p>
147     <p class="input_p_clear">
148         <label for="form_pdf">PDF File<br>
149             <span class="form_comment">
150                 PDF file of Assignment
151             </span>
152         </label>
153         <input id="form_pdf" type="file" name="pdf" class="sharif_input_medium"/>
154     </p>
155 </div>
156 <div class="panel_right">
157     <p class="input_p">

```

```

158 <input_id="form_a_open"_type="checkbox"_name="open"_value="1"_{<=?_$_edit_?_($edit_assignment['open'])_?_checked'_&:_?}>:_
159   set_checkbox('open',_1')_?>/>
160 <label_for="form_a_open"_class="default">Open</label>
161 <span_class="form_comment_space-left">Open_or_close_this_assignment</span>
162 <div_class="shj_error"><=?_$_validationError->hasError('open')_?_$_validationError->getError('open')_&:_?></div>
163 </p>
164 <p_class="input_p">
165   <input_id="form_a_scoreboard"_type="checkbox"_name="scoreboard"_value="1"_{<=?_$_edit_?_($edit_assignment['scoreboard'])_?_checked'_&:_?}>:_
166     set_checkbox('scoreboard',_1')_?>/>
167 <label_for="form_a_scoreboard"_class="default">Scoreboard</label>
168 <span_class="form_comment_space-left">Check_this_to_enable_scoreboard</span>
169 <div_class="shj_error"><=?_$_validationError->hasError('scoreboard')_?_$_validationError->getError('scoreboard')_&:_?></div>
170 </p>
171 <p_class="input_p">
172   <input_id="form_a_javaexceptions"_type="checkbox"_name="javaexceptions"_value="1"_{<=?_$_edit_?_($edit_assignment['
173     javaexceptions'])_?_checked'_&:_?}>:_
174     set_checkbox('javaexceptions',_1')_?>/>
175 <label_for="form_a_javaexceptions"_class="default">Java_Exceptions</label>
176 <span_class="form_comment_space-left">Check_this_to_show_Java_exceptions_to_users</span>
177 <div_class="shj_error"><=?_$_validationError->hasError('javaexceptions')_?_$_validationError->getError('javaexceptions')_&:_?></div>
178 </p>
179 <p_class="input_p">
180   <input_id="form_a_archived_assignment"_type="checkbox"_name="archived_assignment"_value="1"_{<=?_$_edit_?_($edit_assignment['
181     archived_assignment'])_?_checked'_&:_?}>:_
182     set_checkbox('archived_assignment',_1')_?>/>
183 <label_for="form_a_archived_assignment"_class="default">Archived_Assignment</label>
184 <span_class="form_comment_space-left">Check_this_to_make_an_archived_assignment</span>
185 <div_class="shj_error"><=?_$_validationError->hasError('archived_assignment')_?_$_validationError->getError('
186     archived_assignment')_&:_?></div>
187 </p>
188 <p_class="input_p">
189 <label_for="form_late_rule">Coefficient_rule_{<a_target="_blank"_href="https://github.com/ifunpar/Sharif-Judge/blob/docs/
190   v1.4/add_assignment.md#coefficient-rule"></a></label><br>
191 <span_class="form_comment_medium_clear" style="display: block;">PHP_script_without_&lt;?php_&gt;_tags</span>
192 <span_class="form_comment_clear">Press_&lt;esc_&gt;_to_enable/disable_tabindent</span><br>
193 <textarea_id="form_late_rule"_name="late_rule"_rows="20"_class="sharif_input_add_text"><=?_$_edit_?_$_edit_assignment['
194   late_rule']_>:_
195     set_value('late_rule',$_default_late_rule),</textarea>
196 <div_class="shj_error"><=?_$_validationError->hasError('late_rule')_?_$_validationError->getError('late_rule')_&:_?></div>
197 </p>
198 <div>
199 <p_class="input_p" id="add_problems">Problems<button_type="button" id="add" aria-label="Add_Problem"><i class="fa_fa-plus-circle_
200   fa-lg_color1_pointer"></i></button>
201 <table_id="problems_table">
202 <thead>
203 <tr>
204 <th>
205 <th colspan="3" style="border-bottom: 1px solid #BDBDBD">Time_Limit_(ms)</th>
206 <th colspan="2" rowspan="2">Name</th>
207 <th colspan="2" rowspan="2">Score</th>
208 <th colspan="3" rowspan="2">Memory<br>Limit_(KB)</th>
209 <th colspan="2" rowspan="2">Allowed<br>Languages<_aria-label="Link_Help_For_Languages" target="_blank" href="https://github.com/
210   ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#allowed-languages"></a></th>
211 <th colspan="2" rowspan="2">Diff<br>Command<_aria-label="Link_Help_For_Diff_Command" target="_blank" href="https://github.com/
212   ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#diff-command"></a></th>
213 <th colspan="2" rowspan="2">Diff<br>Argument<_aria-label="Link_Help_For_Diff_Argument" target="_blank" href="https://github.com/
214   ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#diff-arguments"></a></th>
215 <th colspan="2" rowspan="2">Upload<br>Only<_aria-label="Link_Help_For_Upload_Only" target="_blank" href="https://github.com/ifunpar/
216   /Sharif-Judge/blob/docs/v1.4/add_assignment.md#upload-only"></a></th>
217 <th colspan="2" rowspan="2">C/C++</th><th>Python</th><th>Java</th>
218 </thead>
219 <tbody>
220 <?php foreach($problems_as_$problem) :_?>
221 <tr>
222   <td><=?_$_problem['id']_?></td>
223   <td><input_aria-label="Problem_Name"_type="text"_name="name[]" class="sharif_input_short" value="<=?_$_problem['name']_&
224     ?>"></td>
225   <td><input_aria-label="Score"_type="text"_name="score[]" class="sharif_input_tiny2" value="<=?_$_problem['score']_&
226     ?>"></td>
227   <td><input_aria-label="Time_Limit_for_C"_type="text"_name="c_time_limit[]" class="sharif_input_tiny2" value="<=?_$_problem['
228     $problem['c_time_limit']]_?>"></td>
229   <td><input_aria-label="Time_Limit_for_Python"_type="text"_name="python_time_limit[]" class="sharif_input_tiny2" value=
230     "<=?_$_problem['python_time_limit']_?>"></td>
231   <td><input_aria-label="Time_Limit_for_Java"_type="text"_name="java_time_limit[]" class="sharif_input_tiny2" value="<=?_$_problem['
232     $problem['java_time_limit']]_?>"></td>
233   <td><input_aria-label="Memory_Limit"_type="text"_name="memory_limit[]" class="sharif_input_tiny" value="<=?_$_problem['
234     memory_limit']_?>"></td>
235   <td><input_aria-label="Allowed_Languages"_type="text"_name="languages[]" class="sharif_input_short2" value="<=?_$_problem['
236     $problem['allowed_languages']]_?>"></td>
237   <td><input_aria-label="Diff_Command"_type="text"_name="diff_cmd[]" class="sharif_input_tiny" value="<=?_$_problem['
238     diff_cmd']_?>"></td>
239   <td><input_aria-label="Diff_Argument"_type="text"_name="diff_arg[]" class="sharif_input_tiny" value="<=?_$_problem['
240     diff_arg']_?>"></td>
241   <td><input_aria-label="Upload_Only"_type="checkbox"_name="is_upload_only[]" class="check" value="<=?_$_problem['id']_&
242     ?>"<=?_$_problem['is_upload_only']_?> checked'_&:_?></td>
243   <td><button_aria-label="Delete_Problem" class="delete_problem" type="button"><i class="fa_fa-times-circle_fa-lg_color1_
244     _pointer"></i></button></td>
245 </tr>
246 <?php endforeach?
247 </tbody>
248 </table>
249 <div class="shj_error"><=?_$_validationError->hasError('name.*')_?_$_validationError->getError('name.*')_&:_?></div>
250 <div class="shj_error"><=?_$_validationError->hasError('score.*')_?_$_validationError->getError('score.*')_&:_?></div>
251 <div class="shj_error"><=?_$_validationError->hasError('c_time_limit/*') ? $_validationError->getError('c_time_limit.*') : ''_?></div>

```

```

231 <div class="shj_error"><?= $validationError->hasError('python_time_limit.*') ? $validationError->getError('python_time_limit.*') : '' ?></div>
232 <div class="shj_error"><?= $validationError->hasError('java_time_limit.*') ? $validationError->getError('java_time_limit.*') : '' ?></div>
233 <div class="shj_error"><?= $validationError->hasError('memory_limit.*') ? $validationError->getError('memory_limit.*') : '' ?></div>
234 <div class="shj_error"><?= $validationError->hasError('languages.*') ? $validationError->getError('languages.*') : '' ?></div>
235 <div class="shj_error"><?= $validationError->hasError('diff_cmd.*') ? $validationError->getError('diff_cmd.*') : '' ?></div>
236 <div class="shj_error"><?= $validationError->hasError('diff_arg.*') ? $validationError->getError('diff_arg.*') : '' ?></div>
237 <p class="input_p">
238   <input type="submit" value="<?= $edit ? 'Edit' : 'Add' ?> Assignment" class="sharif_input"/>
239 </p>
240 </form>
241 <?php $this->endSection() ?>
```

Kode C.2: settings.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: settings.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-gear<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Settings<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Settings<?= $this->endSection() ?>
10
11 <script type='text/javascript' src="<?= $base_url('assets/js/taboverride.min.js') ?>"></script>
12 <script>
13 $(document).ready(function(){
14   tabOverride.set(document.getElementsByTagName('textarea'));
15   $switch = false;
16   $("textarea").keyup(function (e) {
17     if (e.which==27){
18       tabOverride.set(document.getElementsByTagName('textarea'),$switch);
19       $switch = !$switch;
20     }
21   });
22 });
23 </script>
24
25 <span class="title_menu_item">
26   <a href="https://github.com/ifunpar/Shafif-Judge/blob/docs/v1.4/settings.md" target="_blank">i class="fa fa-question-circle-o" color6</i> Help</a>
27 </span>
28
29 <?= $this->section('main_content') ?>
30 <p class="input_p">
31 <?php if ($form_status == 'ok') : ?>
32   <div class="shj_ok">Settings updated successfully.</div>
33 <?php elseif ($form_status == 'error') : ?>
34   <div class="shj_error">Error updating settings.</div>
35 <?php endif ?>
36 <?php foreach ($errors as $error) :?>
37   <div class="shj_error"><?= $error ?></div>
38 <?php endforeach ?>
39 <?php if (!($defc)) : ?>
40   <div class="shj_error">"Tester_path" is not correct.</div>
41 <?php endif ?>
42 </p>
43 <?= form_open('settings/update') ?>
44 <p class="input_p">
45   <label for="form_timezone">
46     Timezone<br>
47     <a target="_blank" href="http://www.php.net/manual/en/timezones.php">list of timezones</a>
48   </label>
49   <span class="form_comment_timer"></span><br>
50   <input id="form_timezone" type="text" name="timezone" class="sharif_input_medium" value="<?= $timezone ?>"/>
51   <div class="shj_error"><?= $validationError->hasError('timezone') ? $validationError->getError('timezone') : '' ?></div>
52 </p>
53 <p class="input_p">
54   <label for="form_week">Week Start Day:</label>
55   <select id="form_week" name="week_start" class="sharif_input">
56     <option value="0" ><?= ($week_start == 0) ? 'selected="selected"' : '' ?>>Sunday</option>
57     <option value="1" ><?= ($week_start == 1) ? 'selected="selected"' : '' ?>>Monday</option>
58     <option value="2" ><?= ($week_start == 2) ? 'selected="selected"' : '' ?>>Tuesday</option>
59     <option value="3" ><?= ($week_start == 3) ? 'selected="selected"' : '' ?>>Wednesday</option>
60     <option value="4" ><?= ($week_start == 4) ? 'selected="selected"' : '' ?>>Thursday</option>
61     <option value="5" ><?= ($week_start == 5) ? 'selected="selected"' : '' ?>>Friday</option>
62     <option value="6" ><?= ($week_start == 6) ? 'selected="selected"' : '' ?>>Saturday</option>
63   </select>
64 </p>
65 <p class="input_p">
66   <label for="form_t_path">Full Path to <code>tester</code></label>
67   <input id="form_t_path" type="text" name="tester_path" class="sharif_input_medium" value="<?= $tester_path ?>"/>
68 </p>
69 <p class="input_p">
70   <label for="form_a_path">Full Path to <code>assignments</code></label>
71   <input id="form_a_path" type="text" name="assignments_root" class="sharif_input_medium" value="<?= $assignments_root ?>"/>
72 </p>
73 <p class="input_p">
74   <label for="form_up_limit">Upload Size Limit (kB)</label>
75   <input id="form_up_limit" type="text" name="file_size_limit" class="sharif_input_medium" value="<?= $file_size_limit ?>"/>
76   <div class="shj_error"><?= $validationError->hasError('file_size_limit') ? $validationError->getError('file_size_limit') : '' ?></div>
77 </p>
78 <p class="input_p">
79   <label for="form_out_limit">Output Size Limit (kB)</label>
```

```

80      <input id="form_out_limit" type="text" name="output_size_limit" class="sharif_input_medium" value="=$_output_size_limit_&gt;" /&gt;&lt;br&gt;
81      &lt;span class="form_comment_clear"&gt;Sets a limit for size of output file generated by submitted code&lt;/span&gt;
82      &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('output_size_limit') ? $validationError-&gt;getError('output_size_limit') : '' ?&gt;&lt;/div&gt;
83  &lt;/p&gt;
84  &lt;p class="input_p"&gt;
85    &lt;label for="form_rpp_all"&gt;Results Per Page&lt;/label&gt;
86    &lt;input id="form_rpp_all" type="text" name="rpp_all" class="sharif_input_medium" value="<?=$_results_per_page_all_?"/&gt;
87    &lt;p class="form_comment_clear"&gt;In All Submissions&lt;br&gt;Enter 0 for no limit&lt;/p&gt;
88    &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('rpp_all') ? $validationError-&gt;getError('rpp_all') : '' ?&gt;&lt;/div&gt;
89  &lt;/p&gt;
90  &lt;p class="input_p"&gt;
91    &lt;label for="form_rpp_final"&gt;Results Per Page&lt;/label&gt;
92    &lt;input id="form_rpp_final" type="text" name="rpp_final" class="sharif_input_medium" value="<?=$_results_per_page_final_?"/&gt;
93    &lt;p class="form_comment_clear"&gt;In Final Submissions&lt;br&gt;Enter 0 for no limit&lt;/p&gt;
94    &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('rpp_final') ? $validationError-&gt;getError('rpp_final') : '' ?&gt;&lt;/div&gt;
95  &lt;/p&gt;
96  &lt;p class="input_p"&gt;
97    &lt;input id="form_en_reg" type="checkbox" name="enable_registration" value="1" &lt;?= $enable_registration ? 'checked' : '' ?&gt;
98    &lt;label for="form_en_reg"&gt;Registration&lt;/label&gt;&lt;br&gt;
99    &lt;span class="form_comment"&gt;Open Public Registration.&lt;/span&gt;
100 &lt;/p&gt;
101 &lt;p class="input_p"&gt;
102   &lt;label for="form_reg_code"&gt;Registration Code&lt;/label&gt;
103   &lt;input id="form_reg_code" type="text" name="registration_code" class="sharif_input_medium" value="<?=$_registration_code_?"/&gt;&lt;br&gt;
104   &lt;p class="form_comment_clear"&gt;If you want to enable registration (above option), It is better to give a registration code&lt;br&gt;
105   to students in your class for validating registration. Enter 0 to disable.&lt;/p&gt;
106   &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('registration_code') ? $validationError-&gt;getError('registration_code') : '' ?&gt;&lt;/div&gt;
107 &lt;/p&gt;
108 &lt;p class="input_p"&gt;
109   &lt;input id="form_log" type="checkbox" name="enable_log" value="1" &lt;?= $enable_log ? 'checked' : '' ?&gt;
110   &lt;label for="form_log"&gt;Log&lt;/label&gt;&lt;br&gt;
111   &lt;span class="form_comment"&gt;Enable Log&lt;/span&gt;
112 &lt;/p&gt;
113 &lt;p class="input_p"&gt;
114   &lt;input id="form_lock_student_display_name" type="checkbox" name="lock_student_display_name" value="1" &lt;?= $lock_student_display_name ? 'checked' : '' ?&gt;
115   &lt;label for="form_lock_student_display_name"&gt;Lock Student's Display Name&lt;/label&gt;&lt;br&gt;
116   &lt;span class="form_comment"&gt;Student's can't change their display name&lt;/span&gt;
117 &lt;/p&gt;
118 &lt;p class="input_p"&gt;
119   &lt;label for="form_late_rule"&gt;Default Coefficient Rule&lt;/label&gt;
120   &lt;span class="form_comment_clear"&gt;PHP Script without &lt;?php ?&gt; tags&lt;/span&gt;&lt;br&gt;
121   &lt;span class="form_comment_clear"&gt;Press esc to enable/disable tab indent&lt;/span&gt;&lt;br&gt;
122   &lt;textarea id="form_late_rule" name="default_late_rule" rows="15" class="sharif_input_add_text_clear"&gt;&lt;?=$_default_late_rule_?&gt;&lt;/textarea&gt;
123 &lt;/p&gt;
124
125 &lt;h2 class="shj_form"&gt;Email&lt;/h2&gt;
126
127 &lt;p class="input_p"&gt;
128   &lt;label for="form_mail_from"&gt;Send Emails From&lt;/label&gt;
129   &lt;input id="form_mail_from" type="text" name="mail_from" class="sharif_input_medium" value="<?=$_mail_from_?"/&gt;
130   &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('mail_from') ? $validationError-&gt;getError('mail_from') : '' ?&gt;&lt;/div&gt;
131 &lt;/p&gt;
132 &lt;p class="input_p"&gt;
133   &lt;label for="form_mail_name"&gt;Send Emails From Name&lt;/label&gt;
134   &lt;input id="form_mail_name" type="text" name="mail_from_name" class="sharif_input_medium" value="<?=$_mail_from_name_?"/&gt;
135   &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('mail_from_name') ? $validationError-&gt;getError('mail_from_name') : '' ?&gt;&lt;/div&gt;
136 &lt;/p&gt;
137 &lt;p class="input_p"&gt;
138   &lt;label for="form_mail_reset"&gt;Password Reset Email&lt;/label&gt;
139   &lt;span class="form_comment"&gt;You can use {SITE_URL}, {RESET_LINK} and {VALID_TIME}&lt;/span&gt;&lt;br&gt;
140   &lt;span class="form_comment_clear"&gt;Press esc to enable/disable tab indent&lt;/span&gt;&lt;br&gt;
141   &lt;textarea id="form_mail_reset" name="reset_password_mail" rows="15" class="sharif_input_add_text_clear"&gt;&lt;?=$_reset_password_mail_?&gt;&lt;/textarea&gt;
142 &lt;/p&gt;
143 &lt;p class="input_p"&gt;
144   &lt;label for="form_mail_add"&gt;Add User Email&lt;/label&gt;
145   &lt;span class="form_comment_clear"&gt;You can use {SITE_URL}, {LOGIN_URL}, {ROLE}, {USERNAME} and {PASSWORD}&lt;/span&gt;&lt;br&gt;
146   &lt;span class="form_comment_clear"&gt;Press esc to enable/disable tab indent&lt;/span&gt;&lt;br&gt;
147   &lt;textarea id="form_mail_add" name="add_user_email" rows="15" class="sharif_input_add_text_clear"&gt;&lt;?=$_add_user_email_?&gt;&lt;/textarea&gt;
148 &lt;/p&gt;
149
150
151 &lt;h2 class="shj_form"&gt;
152   Sandboxing &lt;span class="title_menu_item"&gt;
153     &lt;a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md" target="_blank"&gt;&lt;i class="fa fa-question-circle_color11"&gt;&lt;/i&gt; Help&lt;/a&gt;
154   &lt;/span&gt;
155 &lt;/h2&gt;
156
157 &lt;p class="input_p"&gt;
158   &lt;input id="form_easysandbox" type="checkbox" name="enable_easysandbox" value="1" &lt;?=$_enable_easysandbox_?&gt; 'checked' : '' ?&gt;
159   &lt;label for="form_easysandbox"&gt;EasySandbox&lt;/label&gt;
160 &lt;/p&gt;
161 &lt;p class="form_comment"&gt;Enable EasySandbox for C/C++.&lt;br&gt;
162 You must &lt;a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md#build-easysandbox" target="_blank"&gt; build EasySandbox&lt;/a&gt; before enabling it.&lt;br&gt;
163 &lt;?php if (!$_sandbox_built): ?&gt;
</pre

```

```

164 | <span style="color:red;">>You have not built EasySandbox yet.</span>
165 | <?php endif;?>
166 | </p>
167 |
168 | <p class="input_p">
169 | <input id="form_java_policy" type="checkbox" name="enable_java_policy" value="1" <?= $enable_java_policy ?> checked' <?>
170 | </?>
171 | <label for="form_java_policy">Java Policy</label>
172 | </p>
173 | <p class="form_comment">
174 | Enable <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md#java-sandboxing" target="_blank">Java
175 | Sandboxing</a>
176 | </p>
177 |
178 | <h2 class="shj_form">
179 | Shield <span class="title_menu_item">
180 | <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank"><i class="fa fa-question-
181 | circle_color11">Help</a>
182 | </span>
183 | </h2>
184 | <p class="input_p">
185 | <input id="form_c_sh" type="checkbox" name="enable_c_shield" value="1" <?= $enable_c_shield ?> checked' <?>
186 | <label for="form_c_sh">C_Shield</label><br>
187 | <span class="form_comment">Enable <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target=
188 | _blank">Shield</a> for C</span>
189 | <p class="input_p">
190 | <input id="form_cpp_sh" type="checkbox" name="enable_cpp_shield" value="1" <?= $enable_cpp_shield ?> checked' <?>
191 | <label for="form_cpp_sh">C++_Shield</label><br>
192 | <span class="form_comment">Enable <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target=
193 | _blank">Shield</a> for C++</span>
194 | <p class="input_p">
195 | <input id="form_py2_sh" type="checkbox" name="enable_py2_shield" value="1" <?= $enable_py2_shield ?> checked' <?>
196 | <label for="form_py2_sh">Python_2_Shield</label><br>
197 | <span class="form_comment">Enable <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target=
198 | _blank">Shield</a> for Python_2</span>
199 | <p class="input_p">
200 | <input id="form_py3_sh" type="checkbox" name="enable_py3_shield" value="1" <?= $enable_py3_shield ?> checked' <?>
201 | <label for="form_py3_sh">Python_3_Shield</label><br>
202 | <span class="form_comment">Enable <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target=
203 | _blank">Shield</a> for Python_3</span>
204 | <p class="input_p">
205 | <label for="form_def_c">Shield_Rules_(for_C)</label>
206 | <span class="form_comment_clear">Press "esc" to enable/disable tabindent</span><br>
207 | <textarea id="form_def_c" name="def_c" rows="15" class="sharif_input_add_text_clear"><?= $defc ?></textarea>
208 | </p>
209 | <p class="input_p">
210 | <label for="form_def_cpp">Shield_Rules_(for_C++)</label>
211 | <span class="form_comment_clear">Press "esc" to enable/disable tabindent</span><br>
212 | <textarea id="form_def_cpp" name="def_cpp" rows="15" class="sharif_input_add_text_clear"><?= $defcpp ?></textarea>
213 | </p>
214 | <p class="input_p">
215 | <label for="form_shield_py2">Shield_(for_Python_2)</label>
216 | <span class="form_comment_clear">Press "esc" to enable/disable tabindent</span><br>
217 | <textarea id="form_shield_py2" name="shield_py2" rows="15" class="sharif_input_add_text_clear"><?= $shield_py2 ?></
218 | textarea>
219 | </p>
220 | <p class="input_p">
221 | <label for="form_shield_py3">Shield_(for_Python_3)</label>
222 | <span class="form_comment_clear">Press "esc" to enable/disable tabindent</span><br>
223 | <textarea id="form_shield_py3" name="shield_py3" rows="15" class="sharif_input_add_text_clear"><?= $shield_py3 ?></
224 | textarea>
225 | <p class="input_p">
226 | <br>
227 | <input type="submit" value="Save Changes" class="sharif_input"/>
228 | </p>
229 | </form>
230 | <?= $this->endSection() ?>

```

Kode C.3: assignments.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: assignments.php
4 # author: Filipus Setio Nugroho <filipussetio@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-folder-open<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Assignments<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Assignments<?= $this->endSection() ?>
10
11 <?= $this->section('title_menu') ?>
12 <?php if ($user->level >= 2): ?>
13 <span class="title_menu_item"><a href="<?= $site_url('assignments/add') ?>"><i class="fa fa-plus_color8"></i> Add</a></span>
14 <?php endif ?>
15 <?= $this->endSection() ?>
16
17 <?php $this->section('main_content') ?>
18 <?php $msgclasses = ['success'=> 'shj_g', 'notice'=> 'shj_o', 'error'=> 'shj_r'] ?>
19 <?php foreach ($messages as $message): ?>
20   <p class="<?= $msgclasses[$message['type']] ?>"><?= $message['text'] ?></p>

```

```

21| <?php endforeach ?>
22| <?php if (count($all_assignments)== 0): ?>
23| <p style="text-align:center;">Nothing to show...</p>
24| <br/>
25| <br/>
26| <table class="sharif_table">
27| <thead>
28| <tr>
29|   <th>Select</th>
30|   <th>Name</th>
31|   <th>Problems</th>
32|   <th>Submissions</th>
33|   <th>Coefficient</th>
34|   <th>Start Time</th>
35|   <th>Finish Time</th>
36|   <th>Status</th>
37|   <th>PDF</th>
38| <?php if ($user->level > 0): ?>
39|   <th>Actions</th>
40| <?php endif ?>
41| </tr>
42| </thead>
43| <?php foreach (array_reverse($all_assignments) as $item): ?>
44| <tr>
45|   <td><i tabindex="0" class="pointer_select_assignment_fa_<?=$item['id']_<?=$user->selected_assignment['id']_<?'fa-check-square-o_color6'_<'fa-square-o'_?>_fa-2x" data-id="<?=$item['id']_?>"></i></td>
46|   <td dir="auto"><?=$item['name'] ?></td>
47|   <td><a href=<?=_site_url('problems/'.$item['id'])_?>><? $item['problems'] ?> problem<?=$item['problems'] != 1 ? 's' : '' ?></a></td>
48|   <td><?=$item['total_submits'] ?> submission<?=$item['total_submits'] != 1 ? 's' : '' ?></td>
49|   <td>
50|     <?php if ($item['finished']): ?>
51|       <span style="color:_red;">Finished</span>
52|     <?php else: ?>
53|       <?=$item['coefficient'] ?> %
54|     <?php endif ?>
55|   </td>
56|   <td><?=$item['start_time'] ?></td>
57|   <td><?=$item['finish_time'] ?></td>
58|   <td>
59|     <?php if ($item['open']): ?>
60|       <span style="color:_green;">Open</span>
61|     <?php else: ?>
62|       <span style="color:_red;">Close</span>
63|     <?php endif ?>
64|   </td>
65|   <td>
66|     <a href=<?=_site_url('assignments/pdf/'.$item['id'])_?>>"></a>
67|   </td>
68| <?php if ($user->level > 0): ?>
69| <td>
70|   <?php if ($user->level >= 2): ?>
71|     <a href=<?=_site_url('assignments/downloadtestsdesc/'.$item['id'])_?>><i title="Download_Tests_and_Descriptions" class="fa_fa-cloud-download_fa-lg_color11"></i></a>
72|   <?php endif ?>
73|   <?php if ($user->level >= 1): ?>
74|     <a href=<?=_site_url("assignments/download_submissions/by_user/".$item['id'])_?>><i title="Download_Final_Submissions_(by_user)" class="fa_fa-download_fa-lg_color12"></i></a>
75|     <a href=<?=_site_url("assignments/download_submissions/by_problem/".$item['id'])_?>><i title="Download_Final_Submissions_(by_problem)" class="fa_fa-download_fa-lg_color22"></i></a>
76|   <?php endif ?>
77|   <?php if ($user->level >= 2): ?>
78|     <a href=<?=_site_url('moss/'.$item['id'])_?>><i title="Detect_Similar_Codes" class="fa_fa-shield_fa-lg_color7"></i></a>
79|   <?php endif ?>
80|   <?php if ($user->level >= 2): ?>
81|     <a href=<?=_site_url('assignments/edit/'.$item['id'])_?>><i title="Edit" class="fa_fa-pencil_fa-lg_color3"></i></a>
82|   <?php endif ?>
83|   <?php if ($user->level >= 2): ?>
84|     <a href=<?=_site_url('assignments/delete/'.$item['id'])_?>><i title="Delete" class="fa_fa-times_fa-lg_color11"></i></a>
85|   <?php endif ?>
86|   </td>
87| <?php endforeach ?>
88| </table>
89| <?php endif ?>
90| <?php $this->endSection() ?>

```

Kode C.4: dashboard.php

```

1  <!-- {#
2  # SharIF Judge
3  # file: dashboard.php
4  # author: Filipus Setio Nugroho <filipussetio@gmail.com>
5  #} -->
6  <?= $this->extend('templates/base') ?>
7  <?= $this->section('icon') ?>fa-dashboard<?=$this->endSection() ?>
8  <?= $this->section('title') ?>Dashboard<?=$this->endSection() ?>
9  <?= $this->section('head_title') ?>Dashboard<?=$this->endSection() ?>
10 <?= $this->section('other_assets') ?>
11 <link rel='stylesheet' type='text/css' href='<?=_base_url('assets/fullcalendar/fullcalendar.css')_?>' />
12 <script type='text/javascript' src='<?=_base_url('assets/fullcalendar/fullcalendar.min.js')_?>'></script>
13 <link rel='stylesheet' type='text/css' href='<?=_base_url('assets/gridster/jquery.gridster.css')_?>' />

```



```

111             <div class="notif_text_latest" dir="auto"><?= $notification['text'] ?></div>
112         </div>
113     </div>
114     </div>
115     </div>
116     </div>
117     </li>
118   <li data-row=<?=isset($wp[2])?&$wp[2]['r']:2?> data-col=<?=isset($wp[2])?&$wp[2]['c']:1?> data-sizey=<?=isset($wp[2])?&$wp[2]['x']:1?> data-sizex=<?=isset($wp[2])?&$wp[2]['y']:1?>>
119     <li>
120   </ul>
121 </div>
122 <?= $this->endSection() ?>
123

```

Kode C.5: problems.php

```

1 <! -- {#
2 # SharIF Judge
3 # file: problems.php
4 # author: Filipus Setio Nugroho <filipussetio@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-puzzle-piece<?= $this->endSection() ?>
8 <?= $this->section('title') ?><?= $all_problems[$problem['id']]['name'] ?><?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Problem <?= $problem['id'] ?><?= $this->endSection() ?>
10 <?= $this->section('other_assets') ?>
11 <link rel='stylesheet' type='text/css' href='<?=base_url("assets/snippet/jquery.snippet.css")?>' />
12 <link rel='stylesheet' type='text/css' href='<?=base_url("assets/snippet/themes/github.css")?>' />
13 <script type='text/javascript' src='<?=base_url("assets/snippet/jquery.snippet.js")?>'></script>
14 <script>
15 $(document).ready(function(){
16     // Syntax highlighting increases the page's height, and we need to update the scroll-bar
17     $('.problem_description').resize(function(){
18         $('.scroll-wrapper').nanoScroller();
19     });
20     // Fix text directions for rtl text
21     $.each($('.problem_description[dir="auto"]'), function(i, element){
22         if (getComputedStyle(element).direction == 'rtl')
23         {
24             $(element).css('direction', 'rtl');
25             $(element).parent('ul,ol').css('direction', 'rtl');
26         }
27     });
28     // Syntax highlighting
29     $('pre_code.language-c').parent().snippet('c', {style: shj.color_scheme});
30     $('pre_code.language-cpp').parent().snippet('cpp', {style: shj.color_scheme});
31     $('pre_code.language-python').parent().snippet('python', {style: shj.color_scheme});
32     $('pre_code.language-java').parent().snippet('java', {style: shj.color_scheme});
33 });
34 </script>
35 <?= $this->endSection() ?>
36
37
38
39
40 <?= $this->section('title_menu') ?>
41 <?php if ($user->level >= 2): ?>
42 <span class="title_menu_item"><a href=<?=site_url("problems/edit/md/".$description_assignment['id'].'/.'.$problem['id'])?>><i class="fa fa-pencil_color2"></i> Edit Markdown</a></span>
43 <span class="title_menu_item"><a href=<?=site_url("problems/edit/html/".$description_assignment['id'].'/.'.$problem['id'])?>><i class="fa fa-pencil_color10"></i> Edit HTML</a></span>
44 <span class="title_menu_item"><a href=<?=site_url("problems/edit/plain/".$description_assignment['id'].'/.'.$problem['id'])?>><i class="fa fa-pencil_color8"></i> Edit Plain HTML</a></span>
45 <?php if ($problem['has_pdf']): ?>
46 <span class="title_menu_item"><a href=<?=site_url("assignments/pdf/".$description_assignment['id'].'/.'.$problem['id'])?>><i class="fa fa-download_color1"></i> PDF</a></span>
47 <?php endif ?>
48 <?php endif ?>
49 <?= $this->endSection() ?>
50
51
52
53 <?= $this->section('main_content') ?>
54 <div class="problem_description">
55     <?= $problem['description'] ?>
56 </div>
57
58 <div id="right_sidebar">
59     <div class="problems_widget">
60         <p dir="auto"><i class="fa fa-file-text_fa-lg_color9"></i> <?= $description_assignment['name'] ?></p>
61
62         <?php if (count($all_problems) == 0): ?>
63             <p style="text-align:center;">Nothing to show...</p>
64         <?php endif ?>
65
66         <table class="sharif_table">
67             <thead>
68                 <tr>
69                     <th rowspan="2">#</th>
70                     <th rowspan="2">Problem</th>
71                     <th rowspan="2">Score</th>
72                     <th rowspan="2">Upload<br>Only</th>
73                 </tr>
74             </thead>
75             <tbody>
76                 <?php foreach($all_problems as $one_problem): ?>

```

```

77     <tr <?= $problem['id'] == $one_problem['id'] ? '_class="hl"' : '' ?>>
78     <td><?= $one_problem['id'] ?></td>
79     <td>
80         <a dir="auto" href="<?= site_url("problems/" . $description_assignment['id'] . '/' . $one_problem['id']) . ?>"><?=
81             $one_problem['name'] ?></a>
82         </td>
83         <td><?= $one_problem['score'] ?></td>
84         <td><?= $one_problem['is_upload_only'] ? 'Yes' : 'No' ?></td>
85     </tr>
86     <?php endforeach ?>
87   </table>
88 </div>
89 <?php if ($can_submit): ?>
90 <div class="problems_widget">
91   <p><i class="fa_fa-upload_fa-lg_color11"></i> Submit</p>
92   <?= form_open_multipart('submit') ?>
93   <input type="hidden" name="problem" value="<?= $problem['id'] . ?>"/>
94
95   <p class="input_p">
96     <select id="languages" name="language" class="sharif_input_full-width" aria-label="Select_Language">
97       <option value="0" selected="selected">-- Select Language --</option>
98       <?php foreach ($problem['allowed_languages'] as $l): ?>
99         <option value="<?= $l . ?>"><?= $l ?></option>
100    <?php endforeach ?>
101   </select>
102 </p>
103   <p class="input_p">
104     <input type="file" id="file" class="sharif_input_full-width" name="userfile" aria-label="Upload_File"/>
105   </p>
106   <p class="input_p">
107     <input type="submit" value="Submit" class="sharif_input"/>
108   </p>
109 </form>
110 </div>
111 <?php endif ?>
112 <?php endSection() ?> {# main_content #}
113 <?= $this->endSection() ?> {# main_content #}
114

```

Kode C.6: base.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: base.php
4 # author: Filipus Setio Nugroho <filipussetio@gmail.com>
5 #} -->
6 <?= $this->renderSection('base')?>
7 <!DOCTYPE html>
8 <html lang="en">
9 <head>
10   <title><?= $this->renderSection('head_title') ?> - SharIF Judge</title>
11   <meta content="text/html" charset="UTF-8">
12   <link rel="icon" href="<?= base_url('assets/images/favicon.ico') . ?>"/>
13   <link rel="stylesheet" type="text/css" href="<?= base_url('assets/styles/main.css') . ?>"/>
14   <script type="text/javascript" src="<?= base_url('assets/js/jquery-1.10.2.min.js') . ?>"/></script>
15   <!-- You can use google's_cdn_for_loading_jquery:-->
16   <!--<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>-->
17   <script type="text/javascript" src="<?= base_url('assets/js/jquery-ui-1.10.3.custom.min.js') . ?>"/></script>
18   <link rel="stylesheet" href="<?= base_url('assets/styles/flick/jquery-ui-1.10.3.custom.min.css') . ?>"/>
19   <script type="text/javascript" src="<?= base_url('assets/js/moment.min.js') . ?>"/></script>
20   <script type="text/javascript" src="<?= base_url('assets/js/jquery.hoverIntent.minified.js') . ?>"/></script>
21   <script type="text/javascript" src="<?= base_url('assets/js/jquery.cookie.js') . ?>"/></script>
22   <script type="text/javascript" src="<?= base_url('assets/js/jquery.ba-resize.min.js') . ?>"/></script>
23
24   <script type="text/javascript" src="<?= base_url('assets/noty/jquery.noty.js') . ?>"/></script>
25   <script type="text/javascript" src="<?= base_url('assets/noty/layouts/center.js') . ?>"/></script>
26   <script type="text/javascript" src="<?= base_url('assets/noty/layouts/bottomRight.js') . ?>"/></script>
27   <script type="text/javascript" src="<?= base_url('assets/noty/themes/default.js') . ?>"/></script>
28
29   <link rel='stylesheet' type='text/css' href='<?= base_url('assets/nano_scroller/nanoscroller.css') ?>'/>
30   <script type='text/javascript' src='<?= base_url('assets/nano_scroller/jquery.nanoscroller.min.js') . ?>'/></script>
31
32   <link rel='stylesheet' type='text/css' href='<?= base_url('assets/font/fontawesome/css/fontawesome.min.css') ?>'/>
33
34 </head>
35 <script>
36 shj={};
37 shj.site_url='<?= rtrim(site_url(), '/') ?>/';
38 shj.base_url='<?= rtrim(base_url(), '/') ?>';
39 shj.csrf_token='$_COOKIE['shj csrf token'];
40 shj.offset='<?= moment('<?= shj_now_str() ?>').diff(moment())';
41 shj.time='<?= moment()';
42 shj.finish_time='<?= $finish_time ?>';
43 shj.extra_time='<?= moment.duration(<?= $extra_time ?>, 'seconds')';
44 //notifications
45 shj.check_for_notifications=false;
46 shj.notification_check_delay=30;
47 shj.color_scheme='github';
48 </script>
49 <script type="text/javascript" src="<?= base_url('assets/js/shj_functions.js') . ?>"/></script>
50 <?=$this->renderSection('other_assets')?>
51
52 <body id="body">
53   <a href="#page_title" class="skip" href="#">Skip to content</a>
54
55

```

```
57| <?=_$this->include('templates/top_bar')_?>
58| <?=_$this->include('templates/side_bar')_?>
59| <div_id="main_container"_class="scroll-wrapper">
60|   <div_class="scroll-content">
61|
62|     <div_id="page_title">
63|       <i_class="fa_<?=_$this->renderSection('icon')_?>"></i>
64|       <h1_dir="auto"><?=_$this->renderSection('title')_?></h1>
65|       <?= _$this->renderSection('title_menu')?>
66|     </div>
67|
68|     <div_id="main_content">
69|       <?= _$this->renderSection('main_content')?>
70|     </div>
71|   </div>
72| </div>
73| <?= _$this->renderSection('body_end')_?>
74| </body>
75| </html>
```

Lampiran kode *view* lainnya dapat dilihat pada <https://github.com/filipussetioo/SharIF-JudgeCI4/tree/main/app/Views>.

LAMPIRAN D

KODE PROGRAM LAINNYA

Kode D.1: CheckInstallAndLogin.php

```
1 <?php
2
3 namespace App\Filters;
4
5 use CodeIgniter\Filters\FilterInterface;
6 use CodeIgniter\HTTP\RequestInterface;
7 use CodeIgniter\HTTP\ResponseInterface;
8 use Config\Database;
9
10 class CheckInstallAndLogin implements FilterInterface
11 {
12     /**
13      * Do whatever processing this filter needs to do.
14      * By default it should not return anything during
15      * normal execution. However, when an abnormal state
16      * is found, it should return an instance of
17      * CodeIgniter\HTTP\Response. If it does, script
18      * execution will end and that Response will be
19      * sent back to the client, allowing for error pages,
20      * redirects, etc.
21      *
22      * @param RequestInterface $request
23      * @param array|null      $arguments
24      *
25      * @return mixed
26      */
27     public function before(RequestInterface $request, $arguments = null)
28     {
29         $db = Database::connect();
30
31         if ( !$db->tableExists('sessions'))
32             return redirect()->to('/install');
33
34         $session = \Config\Services::session();
35         if ( !$session->get('logged_in')) // if not logged in
36             return redirect()->to('/login');
37     }
38
39     /**
40      * Allows After filters to inspect and modify the response
41      * object as needed. This method does not allow any way
42      * to stop execution of other after filters, short of
43      * throwing an Exception or Error.
44      *
45      * @param RequestInterface $request
46      * @param ResponseInterface $response
47      * @param array|null      $arguments
48      *
49      * @return mixed
50      */
51     public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)
52     {
53         //
54     }
55 }
```

Kode D.2: CheckLoginandisAjax.php

```
1 <?php
2
3 namespace App\Filters;
4
5 use CodeIgniter\Filters\FilterInterface;
6 use CodeIgniter\HTTP\RequestInterface;
7 use CodeIgniter\HTTP\ResponseInterface;
8
9 class CheckLoginandisAjax implements FilterInterface
10 {
11     /**
12      * Do whatever processing this filter needs to do.
13      * By default it should not return anything during
14      * normal execution. However, when an abnormal state
15      * is found, it should return an instance of
```

```

16 * CodeIgniter\HTTP\Response. If it does, script
17 * execution will end and that Response will be
18 * sent back to the client, allowing for error pages,
19 * redirects, etc.
20 *
21 * @param RequestInterface $request
22 * @param array|null      $arguments
23 *
24 * @return mixed
25 */
26 public function before(RequestInterface $request, $arguments = null)
27 {
28     $session = \Config\Services::session();
29     $request = \Config\Services::request();
30
31     if ( !$session->get('logged_in') ) // if not logged in
32         return redirect()->to('/login');
33     if ( !$request->isAJAX() )
34         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
35 }
36
37 /**
38 * Allows After filters to inspect and modify the response
39 * object as needed. This method does not allow any way
40 * to stop execution of other after filters, short of
41 * throwing an Exception or Error.
42 *
43 * @param RequestInterface $request
44 * @param ResponseInterface $response
45 * @param array|null      $arguments
46 *
47 * @return mixed
48 */
49 public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)
50 {
51     //
52 }
53 }
```

Kode D.3: CheckLoginandLevelAdmin.php

```

1 <?php
2
3 namespace App\Filters;
4
5 use App\Models\User;
6 use CodeIgniter\Filters\FilterInterface;
7 use CodeIgniter\HTTP\RequestInterface;
8 use CodeIgniter\HTTP\ResponseInterface;
9
10 class CheckLoginandLevelAdmin implements FilterInterface
11 {
12     /**
13      * Do whatever processing this filter needs to do.
14      * By default it should not return anything during
15      * normal execution. However, when an abnormal state
16      * is found, it should return an instance of
17      * CodeIgniter\HTTP\Response. If it does, script
18      * execution will end and that Response will be
19      * sent back to the client, allowing for error pages,
20      * redirects, etc.
21      *
22      * @param RequestInterface $request
23      * @param array|null      $arguments
24      *
25      * @return mixed
26      */
27     public function before(RequestInterface $request, $arguments = null)
28     {
29         $session = \Config\Services::session();
30         $user = new User();
31
32         if ( !$session->get('logged_in') ) // if not logged in
33             return redirect()->to('/login');
34         if ( $user->level <= 2 ) // permission denied
35             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
36     }
37
38     /**
39      * Allows After filters to inspect and modify the response
40      * object as needed. This method does not allow any way
41      * to stop execution of other after filters, short of
42      * throwing an Exception or Error.
43      *
44      * @param RequestInterface $request
45      * @param ResponseInterface $response
46      * @param array|null      $arguments
47      *
48      * @return mixed
49      */
50     public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)
51     {
52         //
53     }
54 }
```

Kode D.4: Validation.php

```

1 <?php
2
3 namespace Config;
4
5 use App\Models\SettingsModel;
6 use App\Models\User;
7 use App\Models\UserModel;
8 use CodeIgniter\Config\BaseConfig;
9 use CodeIgniter\Validation\StrictRules\CreditCardRules;
10 use CodeIgniter\Validation\StrictRules\FileRules;
11 use CodeIgniter\Validation\StrictRules\FormatRules;
12 use CodeIgniter\Validation\StrictRules\Rules;
13
14 class Validation extends BaseConfig
15 {
16     // -----
17     // Setup
18     // -----
19
20     /**
21      * Stores the classes that contain the
22      * rules that are available.
23      *
24      * @var string[]
25      */
26     public array $ruleSets = [
27         Rules::class,
28         FormatRules::class,
29         FileRules::class,
30         CreditCardRules::class,
31         MyRules::class,
32     ];
33
34     /**
35      * Specifies the views that are used to display the
36      * errors.
37      *
38      * @var array<string, string>
39      */
40     public array $templates = [
41         'list' => 'CodeIgniter\Validation\Views\list',
42         'single' => 'CodeIgniter\Validation\Views\single',
43     ];
44     // -----
45     // Rules
46     // -----
47 }
48
49 class MyRules
50 {
51     public function password_check($str): bool
52     {
53         if ($str == 0 OR ($str == 6 && $str == 200))
54             return TRUE;
55         return FALSE;
56     }
57
58     public function password_again_check($str) :bool
59     {
60         $request = \Config\Services::request();
61         if ($request->getPost('password') != $request->getPost('password_again'))
62             return FALSE;
63         return TRUE;
64     }
65
66     public function role_check($role)
67     {
68         $user = new User();
69         if ($user->level <= 2)
70             return ($role == ' ');
71
72         // Admins can change everybody's user role:
73         $roles = array('admin', 'head_instructor', 'instructor', 'student');
74         return in_array($role, $roles);
75     }
76
77     /**
78      * Checks whether a user with this email exists
79      */
80     public function email_check($email, $edit_username):bool
81     {
82         $edit_username = explode('/', $edit_username);
83         $user_model = new UserModel();
84         if ($user_model->have_email($email, $edit_username[1]))
85             return FALSE;
86         return TRUE;
87     }
88
89     /**
90      * checks whether the entered registration code is correct or not
91      */
92
93     public function _registration_code($code){
94         $settings_model = new SettingsModel();
95         $rc = $settings_model->get_setting('registration_code');
96         if ($rc == '0')
97             return TRUE;
98         if ($rc == $code)

```

```
99         return TRUE;
100    return FALSE;
101 }
102
103 /**
104 * Required
105 *
106 * @param string
107 * @return bool
108 */
109 public function required($str)
110 {
111     return is_array($str) ? (bool) count($str) : ($str !== '');
112 }
113
114 // -----
115
116
117 /**
118 * Is Lowercase
119 *
120 * @param $str
121 * @return bool
122 */
123 public function lowercase($str)
124 {
125     return (strtolower($str) === $str);
126 }
127
128 // -----
129
130
131 public function _check_language($str)
132 {
133     if ($str=='0')
134         return FALSE;
135     if (in_array( strtolower($str),array('c', 'c++', 'python_2', 'python_3', 'java', 'zip', 'pdf', 'txt')))
136         return TRUE;
137     return FALSE;
138 }
139
140
141
142 // -----
143 // Used in Submissions.php
144
145 public function _check_type($type)
146 {
147     return ($type === 'code' || $type === 'result' || $type === 'log');
148 }
149 }
```