

## TUGAS AKHIR

KONVERSI SHARIF JUDGE DARI BERBASIS CODEIGNITER  
3 MENJADI BERBASIS CODEIGNITER 4



Filipus

NPM: 6181901074

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2024



## FINAL PROJECT

SHARIF JUDGE CONVERSION FROM CODEIGNITER 3  
BASED TO CODEIGNITER 4 BASED



Filipus

NPM: 6181901074

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2024



## **LEMBAR PENGESAHAN**

### **KONVERSI SHARIF JUDGE DARI BERBASIS CODEIGNITER 3 MENJADI BERBASIS CODEIGNITER 4**

**Filipus**

**NPM: 6181901074**

**Bandung, 11 Januari 2024**

**Menyetujui,**

**Pembimbing**

**Pascal Alfadian, Nugroho, M.Comp.**

**Ketua Tim Penguji**

**Anggota Tim Penguji**

**Husnul Hakim, M.T.**

**Elisati Hulu, M.T.**

**Mengetahui,**

**Ketua Program Studi**

**Mariskha Tri Adithia, P.D.Eng**



## **PERNYATAAN**

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa tugas akhir dengan judul:

### **KONVERSI SHARIF JUDGE DARI BERBASIS CODEIGNITER 3 MENJADI BERBASIS CODEIGNITER 4**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal 11 Januari 2024



Filipus  
NPM: 6181901074



## ABSTRAK

*SharIF Judge* merupakan sebuah *online judge* yang dikustomisasi untuk keperluan tugas pada prodi Informatika Universitas Katolik Parahyangan. *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* cepat karena hanya membutuhkan sedikit sumber daya untuk menjalankannya. *framework* ini juga menyediakan banyak *library* yang dapat digunakan untuk melakukan pembangunan. 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 tugas akhir 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*. Pengujian dilakukan dengan membandingkan fitur yang terdapat pada *SharIF Judge* berbasis *CodeIgniter 4* dan *SharIF Judge* berbasis *CodeIgniter 3*. Pengujian menghasilkan tidak terdapat perubahan pada fitur yang terdapat pada *SharIF Judge* berbasis *CodeIgniter 4* dan *SharIF Judge* berbasis *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 the needs of tasks in the Computer Science program at Parahyangan Catholic University. Initially named Sharif Judge, it was an open-source software developed by Mohammad Javad with the purpose of evaluating code in various languages such as C, C++, Java, and Python. The SharIF Judge software was built using CodeIgniter 3, a PHP-based framework. Later on, it was forked and modified to become SharIF Judge, incorporating additional functionalities tailored to the needs of the Computer Science program at UNPAR (Parahyangan Catholic University) for collecting students' assignments and exams.

Currently, SharIF Judge utilizes CodeIgniter 3, known for its efficiency in resource usage and providing a variety of libraries for development. However, since CodeIgniter 3 has entered a maintenance phase, it no longer receives updates from its developers and has lost documentation on its official website. Therefore, in this final project, SharIF Judge will be converted to the CodeIgniter 4 framework.

The conversion involves moving files to a new directory and rewriting functions present in SharIF Judge. Additionally, some libraries are removed and replaced since they are no longer available in CodeIgniter 4. The absent libraries are substituted with new functions present 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 that it operates similarly to the CodeIgniter 3-based version. The testing compares the features of SharIF Judge based on CodeIgniter 4 with those of SharIF Judge based on CodeIgniter 3, resulting in no changes observed in the features between the two versions.

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



*Tugas Akhir ini dipersembahkan kepada kedua orang tua. . . ?*



## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Masa Esa atas seluruh berkat dan kekuatan yang diberikan kepada penulis sehingga dapat menyelesaikan tugas akhir dengan judul "Konversi *SharIF Judge* dari berbasis *CodeIgniter 3* menjadi berbasis *CodeIgniter 4*". Penulis juga ingin mengucapkan terima kasih sebesar-besarnya kepada beberapa pihak atas dukungan dan bantuan yang telah diberikan. Penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua dan keluarga penulis yang selalu memberikan dukungan kepada penulis.
2. Bapak Pascal Alfadian sebagai dosen pembimbing yang telah membimbing serta membantu memberikan arahan pada tugas akhir ini.
3. Seluruh rekan kerja DNAArtworks yang telah membantu dalam penggerjaan tugas akhir ini.
4. Kepada seluruh teman-teman yang telah meneman, membantu, dan memberi dukungan selama penggerjaan tugas akhir ini.
5. Pihak-pihak lain yang telah membantu penulisan skripsi ini, yang terus memberikan doa dan semangat kepada penulis.

Akhir kata, penulis ingin mengucapkan permohonan maaf atas kekurangan-kekurangan pada tugas akhir ini. Penulis berharap agar tugas akhir ini dapat bermanfaat bagi pembaca yang hendak melakukan penelitian serupa.

Bandung, Januari 2024

Penulis



# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xxi</b>
<b>DAFTAR TABEL</b>	<b>xxiii</b>
<b>DAFTAR KODE PROGRAM</b>	<b>xxviii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	3
1.4 Batasan Masalah . . . . .	3
1.5 Metodologi . . . . .	3
1.6 Sistematika Pembahasan . . . . .	3
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 <i>CodeIgniter 3</i> [1] . . . . .	5
2.1.1 <i>Model-View-Controller</i> . . . . .	5
2.1.2 <i>CodeIgniter URLs</i> . . . . .	7
2.1.3 <i>Helpers</i> . . . . .	8
2.1.4 <i>Libraries</i> . . . . .	8
2.1.5 <i>Database</i> . . . . .	13
2.1.6 <i>URI Routing</i> . . . . .	14
2.1.7 <i>Error Handling</i> . . . . .	15
2.1.8 <i>Auto-loading</i> . . . . .	15
2.2 <i>SharIF Judge</i> . . . . .	15
2.2.1 Struktur Aplikasi . . . . .	15
2.2.2 Instalasi . . . . .	15
2.2.3 <i>Clean URLs</i> . . . . .	16
2.2.4 <i>Users</i> . . . . .	17
2.2.5 Menambah <i>Assignment</i> . . . . .	18
2.2.6 <i>Sample Assignment</i> . . . . .	21
2.2.7 <i>Test Structure</i> . . . . .	23
2.2.8 Deteksi Kecurangan . . . . .	25
2.2.9 Keamanan . . . . .	26
2.2.10 <i>Sandboxing</i> . . . . .	27
2.2.11 <i>Shield</i> . . . . .	28
2.3 <i>CodeIgniter 4</i> [2] . . . . .	29
2.3.1 <i>Models-Views-Controllers</i> . . . . .	30
2.3.2 <i>Autoloading Files</i> . . . . .	32

2.3.3	<i>Configuration</i>	33
2.3.4	<i>CodeIgniter URLs</i>	34
2.3.5	<i>Error Handling</i>	35
2.3.6	<i>URI Routing</i>	35
2.3.7	<i>Database</i>	36
2.3.8	<i>Library</i>	38
2.3.9	<i>Helpers</i>	42
2.4	Konversi <i>CodeIgniter 3 ke CodeIgniter 4</i> [2]	42
2.4.1	Struktur Aplikasi	42
2.4.2	<i>Routing</i>	42
2.4.3	<i>Model, View, dan Controller</i>	43
2.4.4	<i>Class Loading</i>	44
2.4.5	<i>Configuration</i>	44
2.4.6	<i>Database</i>	44
2.4.7	<i>Migrations</i>	44
2.4.8	<i>Routing</i>	44
2.4.9	<i>Libraries</i>	45
2.4.10	<i>Helpers</i>	46
2.4.11	<i>Framework</i>	46
<b>3</b>	<b>ANALISIS</b>	<b>47</b>
3.1	Analisis Sistem Kini	47
3.1.1	<i>Model</i>	47
3.1.2	<i>View</i>	51
3.1.3	<i>Controller</i>	60
3.1.4	<i>Assets</i>	64
3.1.5	<i>Config</i>	65
3.1.6	<i>Libraries</i>	66
3.2	Analisis Sistem Usulan	70
3.2.1	Persiapan <i>CodeIgniter 4</i>	70
3.2.2	Struktur Aplikasi	70
3.2.3	<i>Routing</i>	71
3.2.4	<i>Model, View, and Controller</i>	72
3.2.5	<i>Libraries</i>	74
3.2.6	<i>Configuration</i>	77
3.2.7	<i>Database</i>	78
3.2.8	<i>Error Handling</i>	78
3.2.9	<i>Helpers</i>	78
<b>4</b>	<b>PERANCANGAN</b>	<b>81</b>
4.1	Perancangan Perubahan Struktur Aplikasi	81
4.1.1	<i>app/Config</i>	81
4.1.2	<i>Controllers</i>	82
4.1.3	<i>Model</i>	85
4.1.4	<i>View</i>	87
4.1.5	<i>Filters</i>	87
4.1.6	<i>Routing</i>	88
4.1.7	<i>Libraries</i>	88
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>89</b>
5.1	Lingkungan Implementasi dan Pengujian	89
5.2	Implementasi	90

5.2.1	Instalasi <i>CodeIgniter 4</i>	90
5.2.2	<i>app/Config</i>	90
5.2.3	<i>Controllers</i>	96
5.2.4	<i>Filters</i>	98
5.2.5	<i>Helpers</i>	100
5.2.6	<i>Libraries</i>	100
5.2.7	<i>Models</i>	107
5.2.8	<i>View</i>	108
5.2.9	<i>public</i>	110
5.2.10	<i>restriced</i>	110
5.2.11	Kode Program	110
5.2.12	Basis Data	110
5.3	Pengujian Fungsional	111
5.4	Pengujian Eksperimental	118
5.4.1	<i>Ubuntu 22.04</i>	118
5.4.2	<i>Ubuntu 23.04</i>	119
5.4.3	<i>Ubuntu 23.10</i>	120
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>123</b>
6.1	Kesimpulan	123
6.2	Saran	123
<b>DAFTAR REFERENSI</b>		<b>125</b>
<b>A</b>	<b>KODE PROGRAM</b>	<b>127</b>
A.1	<i>Controller</i>	127
A.2	<i>Model</i>	146
A.3	<i>View</i>	159
A.4	Kode lainnya	169



## 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> . . . . .	18
3.1 Tampilan Halaman <i>Dashboard</i> . . . . .	52
3.2 Tampilan Halaman <i>Profile</i> . . . . .	52
3.3 Tampilan Halaman <i>Settings</i> . . . . .	53
3.4 Tampilan Halaman <i>Users</i> . . . . .	53
3.5 Tampilan Halaman <i>Notifications</i> . . . . .	54
3.6 Tampilan Halaman <i>Assignments</i> . . . . .	54
3.7 Tampilan Halaman <i>Problems</i> . . . . .	55
3.8 Tampilan Halaman <i>Submit</i> . . . . .	55
3.9 Tampilan Halaman <i>Final Submission</i> . . . . .	56
3.10 Tampilan Halaman <i>All Submission</i> . . . . .	56
3.11 Tampilan Halaman <i>Scoreboard</i> . . . . .	57
3.12 Tampilan Halaman <i>Hall of Fame</i> . . . . .	57
3.13 Tampilan Halaman <i>24-hour Log</i> . . . . .	58
3.14 Tampilan Halaman <i>ReJudge</i> . . . . .	58
3.15 Tampilan Halaman <i>Submission Queue</i> . . . . .	59
3.16 Tampilan Halaman <i>Cheat Detection</i> . . . . .	59
3.17 Pemindahan struktur aplikasi menuju <i>CodeIgniter 4</i> . . . . .	70
5.1 Tampilan halaman <i>login SharIF Judge</i> . . . . .	116
5.2 Tampilan halaman <i>dashboard SharIF Judge</i> . . . . .	116
5.3 Tampilan halaman <i>assignments SharIF Judge</i> . . . . .	117
5.4 Tampilan halaman <i>add assignments SharIF Judge</i> . . . . .	117
5.5 Tampilan halaman <i>all submissions SharIF Judge</i> . . . . .	118



## DAFTAR TABEL

2.1	Tabel tingkat pengguna . . . . .	17
2.2	Tabel izin aksi setiap pengguna . . . . .	17
2.3	Tabel contoh masukan dan keluaran <i>Problem 1</i> . . . . .	21
2.4	Tabel contoh masukan dan keluaran <i>Problem 2</i> . . . . .	21
5.1	Perangkat Keras Lingkungan Pembangunan . . . . .	89
5.2	Perangkat Lunak Lingkungan Pembangunan . . . . .	89
5.3	Perangkat Keras Lingkungan Pengujian . . . . .	90
5.4	Perangkat Lunak Lingkungan Pengujian . . . . .	90
5.5	Struktur baru tabel <i>shj_sessions</i> . . . . .	111
5.6	Tabel Pengujian Fungsional . . . . .	111
5.6	Sambungan dari halaman sebelumnya . . . . .	112
5.6	Sambungan dari halaman sebelumnya . . . . .	113
5.6	Sambungan dari halaman sebelumnya . . . . .	114
5.6	Sambungan dari halaman sebelumnya . . . . .	115
5.6	Sambungan dari halaman sebelumnya . . . . .	116



## DAFTAR KODE PROGRAM

2.1	Contoh <i>model</i> pada <i>CodeIgniter 3</i> . . . . .	6
2.2	Contoh <i>view</i> pada <i>CodeIgniter 3</i> . . . . .	6
2.3	Contoh <i>controller</i> pada <i>CodeIgniter 3</i> . . . . .	7
2.4	Contoh <i>file .htaccess</i> pada halaman index.php . . . . .	8
2.5	Contoh URL sebelum dan sesudah ditambahkan akhiran . . . . .	8
2.6	<i>File application/config.php</i> . . . . .	8
2.7	Contoh kelas <i>library</i> pada <i>CodeIgniter 3</i> . . . . .	8
2.8	Contoh pemanggilan <i>library</i> pada <i>file controller</i> . . . . .	9
2.9	Contoh penambahan <i>tag</i> pada <i>file view</i> . . . . .	9
2.10	Contoh pengiriman email melalui <i>controller</i> . . . . .	10
2.11	Contoh konfigurasi preferensi <i>library email</i> secara manual . . . . .	10
2.12	Contoh <i>controller</i> untuk melakukan validasi dan penyimpanan . . . . .	10
2.13	Contoh konfigurasi untuk menyimpan data pada <i>database</i> . . . . .	11
2.14	Contoh penggunaan <i>library Zip Encoding</i> . . . . .	12
2.15	Contoh <i>library</i> yang dibentuk . . . . .	12
2.16	Contoh konfigurasi <i>database</i> . . . . .	13
2.17	Contoh penggunaan <i>query</i> . . . . .	13
2.18	Contoh membangun <i>database</i> menggunakan <i>CodeIgniter 3</i> . . . . .	13
2.19	Contoh menambahkan kolom dengan konfigurasinya menggunakan <i>CodeIgniter3</i> . . . . .	13
2.20	Contoh URL yang sudah dimetakan . . . . .	14
2.21	Struktur aplikasi <i>SharIF Judge</i> . . . . .	15
2.22	Kode untuk melakukan pengetesan fungsi . . . . .	16
2.23	Contoh <i>path</i> pada halaman index.php . . . . .	16
2.24	Contoh pengaturan koneksi untuk <i>database</i> . . . . .	16
2.25	Contoh URL <i>SharIF Judge</i> . . . . .	17
2.26	Contoh hasil URL <i>SharIF Judge</i> yang telah ditulis ulang . . . . .	17
2.27	Sintaks untuk menambahkan pengguna . . . . .	18
2.28	Contoh kode untuk menambahkan pengguna . . . . .	18
2.29	Contoh tampilan fitur <i>Java Exceptions</i> . . . . .	19
2.30	Contoh skrip PHP . . . . .	20
2.31	Susunan pohon untuk ketiga <i>problems</i> . . . . .	21
2.32	Contoh solusi <i>problem 1</i> bahasa <i>C</i> . . . . .	22
2.33	Contoh solusi <i>problem 1</i> bahasa <i>C++</i> . . . . .	22
2.34	Contoh solusi <i>problem 1</i> bahasa <i>Java</i> . . . . .	23
2.35	Contoh solusi <i>problem 2</i> bahasa <i>C</i> . . . . .	23
2.36	Contoh solusi <i>problem 2</i> bahasa <i>C++</i> . . . . .	23
2.37	Templat kode <i>tester.cpp</i> . . . . .	24
2.38	Contoh pohon <i>file</i> dari <i>sample test</i> . . . . .	24
2.39	Kode metode perbandingan <i>tester</i> dengan bahasa <i>tester.cpp</i> . . . . .	25
2.40	Contoh potongan skrip <i>perl</i> . . . . .	25
2.41	Kode <i>runcode.sh</i> sebelum ditambahkan . . . . .	26
2.42	Kode <i>runcode.sh</i> setelah ditambahkan . . . . .	26

2.43 Kode <i>runcode.sh</i> awal . . . . .	26
2.44 Contoh kode untuk membatasi jumlah proses . . . . .	27
2.45 Kode <i>runcode.sh</i> awal . . . . .	27
2.46 Kode <i>shield</i> untuk melarang penggunaan <i>goto</i> . . . . .	28
2.47 Sintaks aturan <i>#define</i> . . . . .	28
2.48 Cara keluar dari <i>shield</i> untuk <i>python</i> . . . . .	29
2.49 Contoh <i>Models</i> pada <i>CodeIgniter 4</i> . . . . .	30
2.50 Contoh penggunaan <i>model</i> untuk mencari data spesifik . . . . .	30
2.51 Contoh <i>Views</i> pada <i>CodeIgniter 4</i> . . . . .	31
2.52 Contoh menampilkan <i>Views</i> pada <i>controller</i> . . . . .	31
2.53 Contoh <i>Controllers</i> pada <i>CodeIgniter 4</i> . . . . .	31
2.54 Contoh <i>Controllers Filters</i> pada <i>CodeIgniter 4</i> . . . . .	31
2.55 Contoh mengakses data menggunakan <i>IncomingRequest</i> . . . . .	32
2.56 Contoh konfigurasi menggunakan <i>namespace PSR-4</i> . . . . .	32
2.57 Contoh konfigurasi menggunakan <i>classmap</i> . . . . .	33
2.58 Contoh mengakses <i>file configuration</i> . . . . .	33
2.59 Contoh pembentukan <i>file configuration</i> . . . . .	34
2.60 Contoh variabel yang disimpan pada <i>file .env</i> . . . . .	34
2.61 Contoh <i>file .htaccess</i> pada <i>Apache Web Server</i> . . . . .	34
2.62 Contoh penggunaan <i>try-files</i> . . . . .	35
2.63 Contoh penggunaan <i>error handling</i> . . . . .	35
2.64 Contoh penggunaan <i>error handling</i> untuk mengambil <i>exception</i> . . . . .	35
2.65 Contoh penggunaan <i>error handling</i> untuk memberikan <i>error</i> berupa 404. . . . .	35
2.66 Contoh <i>route</i> yang didefinisikan secara manual . . . . .	36
2.67 Contoh <i>route</i> yang menggunakan <i>grouping</i> manual . . . . .	36
2.68 Contoh konfigurasi <i>database</i> pada <i>CodeIgniter 4</i> . . . . .	36
2.69 Contoh konfigurasi <i>database</i> pada <i>file .env</i> . . . . .	37
2.70 Contoh penggunaan <i>query</i> menggunakan konfigurasi pada <i>CodeIgniter 4</i> . . . . .	37
2.71 Contoh pembentukan tabel melalui <i>database forge</i> . . . . .	37
2.72 Contoh kode untuk melakukan konfigurasi <i>email</i> . . . . .	38
2.73 Contoh kode untuk melakukan pengiriman <i>email</i> . . . . .	38
2.74 Contoh kode untuk konfigurasi penyimpanan <i>session</i> . . . . .	39
2.75 Contoh <i>form</i> untuk melakukan pengunggahan <i>file</i> . . . . .	39
2.76 Contoh kode <i>controller</i> untuk melakukan validasi dan penyimpanan. . . . .	40
2.77 Contoh kode untuk melakukan pengumpulan data. . . . .	41
2.78 Contoh kode untuk melakukan validasi data yang sudah dikumpulkan. . . . .	41
2.79 Contoh kode untuk menetapkan aturan untuk validasi data yang sudah dikumpulkan. . . . .	42
2.80 Contoh kode pembentukan aturan secara manual pada <i>file Validation.php</i> . . . . .	42
2.81 Penambahan kode pada <i>file migration</i> . . . . .	44
2.82 Perubahan kode untuk melakukan validasi. . . . .	46
3.1 Contoh kode penggunaan <i>Library Unzip</i> . . . . .	66
3.2 Contoh <i>view</i> menggunakan <i>library Twig</i> . . . . .	66
3.3 Contoh kode penggunaan <i>Library Password_hash</i> . . . . .	67
3.4 Contoh kode penggunaan <i>Library Parsedown</i> . . . . .	68
3.5 Contoh hasil teks sesudah dilakukan <i>parsedown</i> . . . . .	68
3.6 Contoh penggunaan <i>library upload</i> . . . . .	68
3.7 Contoh penggunaan <i>library input</i> . . . . .	69
3.8 Contoh penggunaan <i>library URI</i> . . . . .	69
3.9 Contoh penggunaan <i>library session</i> . . . . .	69
3.10 Contoh fungsi untuk mengambil data seluruh user pada <i>CodeIgniter 3</i> . . . . .	72
3.11 Hasil konversi fungsi untuk mengambil data seluruh user . . . . .	72

3.12 Contoh <i>view</i> menggunakan <i>twig</i> . . . . .	73
3.13 Contoh <i>view</i> menggunakan <i>php</i> . . . . .	73
3.14 Penggunaan <i>initController</i> untuk inisiasi fungsi menuju variabel . . . . .	73
3.15 Contoh perubahan <i>library request</i> . . . . .	74
3.16 Contoh perubahan <i>library emails</i> . . . . .	74
3.17 Contoh perubahan <i>library upload</i> . . . . .	74
3.18 Contoh sintaks untuk mengambil <i>extension</i> dan data <i>file</i> . . . . .	74
3.19 Contoh perubahan penambahan sintaks pada aturan yang dibentuk manual . . . . .	75
3.20 Contoh perubahan penggunaan <i>library Zip Archive</i> untuk melakukan <i>zip</i> . . . . .	76
3.21 Contoh perubahan penggunaan <i>library Zip Archive</i> untuk melakukan <i>unzip</i> . . . . .	76
3.22 Contoh perubahan <i>library MY_Form_validation</i> pada <i>CodeIgniter 4</i> . . . . .	77
3.23 Contoh perubahan <i>library Parsedown</i> pada <i>CodeIgniter 4</i> . . . . .	77
3.24 Contoh pemanggilan <i>Error Handling</i> pada <i>CodeIgniter 4</i> . . . . .	78
3.25 Contoh pemanggilan <i>Error Handling 404</i> pada <i>CodeIgniter 4</i> . . . . .	78
 5.1 Kode <i>application/config/config.php</i> yang dipindahkan menuju <i>.env</i> . . . . .	90
5.2 Pemindahan kode pada <i>Constant</i> . . . . .	91
5.3 Pemindahan <i>app/config/database.php</i> menuju <i>.env</i> . . . . .	91
5.4 Pemindahan <i>app/config/secrets.php</i> menuju <i>Email.php</i> . . . . .	92
5.5 Pemindahan <i>app/config/config.php</i> menuju <i>Encryption.php</i> . . . . .	92
5.6 Penambahan nama <i>filters</i> untuk didefinisikan menuju <i>routes</i> . . . . .	92
5.7 Penambahan <i>routes</i> yang digunakan pada aplikasi <i>SharIF Judge</i> . . . . .	92
5.8 Penambahan <i>file Secrets.example.php</i> . . . . .	94
5.9 Pemindahan <i>file config</i> menuju <i>Security.php</i> . . . . .	94
5.10 Pemindahan <i>file config</i> menuju <i>Session.php</i> . . . . .	95
5.11 Perancangan aturan yang dibentuk secara manual pada <i>file Validation.php</i> . . . . .	95
5.12 Perubahan kode <i>controller Logs.php</i> pada <i>CodeIgniter 4</i> . . . . .	97
5.13 Penambahan sintaks pada <i>BaseController</i> . . . . .	97
5.14 Perancangan inisiasi <i>request</i> pada <i>__construct</i> . . . . .	98
5.15 Perancangan penggunaan <i>request</i> . . . . .	98
5.16 Perancangan kode pada <i>Filters CheckCLI.php</i> . . . . .	98
5.17 Perancangan kode pada <i>Filters CheckInstallAndLogin.php</i> . . . . .	98
5.18 Perancangan kode pada <i>Filters CheckLogin.php</i> . . . . .	99
5.19 Perancangan kode pada <i>Filters CheckLoginandCLI.php</i> . . . . .	99
5.20 Pemindahan kode pada <i>Filters CheckLoginandAjax.php</i> . . . . .	99
5.21 Pemindahan kode pada <i>Filters CheckLoginandLevelAdmin.php</i> . . . . .	99
5.22 Pemindahan kode pada <i>Filters CheckLoginandLevelHead.php</i> . . . . .	100
5.23 Penambahan <i>filter</i> pada <i>routes</i> . . . . .	100
5.24 Perubahan sintaks <i>error handling</i> . . . . .	100
5.25 Perubahan sintaks <i>error handling 404</i> . . . . .	101
5.26 Perubahan penggunaan sintaks pada <i>library emails</i> . . . . .	101
5.27 Perancangan perubahan <i>library upload</i> pada <i>CodeIgniter 4</i> . . . . .	101
5.28 Perancangan inisiasi dan penambahan sintaks <i>library URIs</i> pada <i>__construct</i> . .	102
5.29 Perancangan inisiasi <i>library session</i> . . . . .	102
5.30 Perancangan penggunaan <i>library session</i> . . . . .	102
5.31 Perancangan inisiasi <i>validation</i> pada <i>__construct</i> . . . . .	102
5.32 Perancangan perubahan konfigurasi aturan pada <i>library validation</i> . . . . .	103
5.33 Perancangan perubahan penggunaan <i>validation</i> pada <i>CodeIgniter 4</i> . . . . .	103
5.34 Perancangan perubahan <i>zip encoding</i> menjadi <i>zip archive</i> . . . . .	103
5.35 Perancangan perubahan <i>unzip</i> menggunakan <i>zip archive</i> pada <i>controller</i> . . . . .	104
5.36 Perancangan perubahan <i>library MY_Form_validation</i> pada <i>CodeIgniter 4</i> . . . . .	105
5.37 Contoh perubahan <i>library MY_Form_validation</i> pada <i>CodeIgniter 4</i> . . . . .	106

5.38 Perancangan perubahan <i>library Parsedown</i> pada <i>CodeIgniter 4</i> . . . . .	106
5.39 Perancangan perubahan inisiasi <i>library Parsedown</i> pada <i>controller CodeIgniter 4</i> . . . . .	106
5.40 Perancangan perubahan <i>library Shj_pagination</i> pada <i>CodeIgniter 4</i> . . . . .	107
5.41 Perancangan perubahan <i>model</i> pada <i>CodeIgniter 4</i> . . . . .	107
5.42 Perubahan sintaks pada <i>model</i> . . . . .	107
5.43 Perubahan <i>view</i> pada <i>Login.php</i> . . . . .	109
5.44 Penambahan kode pada <i>controller Login.php</i> . . . . .	110
5.45 Pembangunan <i>sandbox</i> pada <i>Ubuntu 22.04</i> . . . . .	118
5.46 <i>Error message</i> pembangunan <i>sandbox</i> pada <i>Ubuntu 22.04</i> . . . . .	119
5.47 Pembangunan <i>sandbox</i> pada <i>Ubuntu 23.04</i> . . . . .	119
5.48 <i>Error message</i> pembangunan <i>sandbox</i> pada <i>Ubuntu 23.04</i> . . . . .	119
5.49 Pembangunan <i>sandbox</i> pada <i>Ubuntu 23.10</i> . . . . .	120
5.50 <i>Error message</i> pembangunan <i>sandbox</i> pada <i>Ubuntu 23.10</i> . . . . .	120
A.1 Assignments.php . . . . .	127
A.2 BaseController.php . . . . .	134
A.3 Submissions.php . . . . .	134
A.4 Submit.php . . . . .	140
A.5 AssignmentModel.php . . . . .	146
A.6 ScoreboardModel.php . . . . .	151
A.7 UserModel.php . . . . .	153
A.8 add_assignment.php . . . . .	159
A.9 settings.php . . . . .	162
A.10 assignments.php . . . . .	165
A.11 dashboard.php . . . . .	166
A.12 problems.php . . . . .	167
A.13 base.php . . . . .	168
A.14 CheckInstallAndLogin.php . . . . .	169
A.15 CheckLoginandisAjax.php . . . . .	170
A.16 CheckLoginandLevelAdmin.php . . . . .	170
A.17 Validation.php . . . . .	171

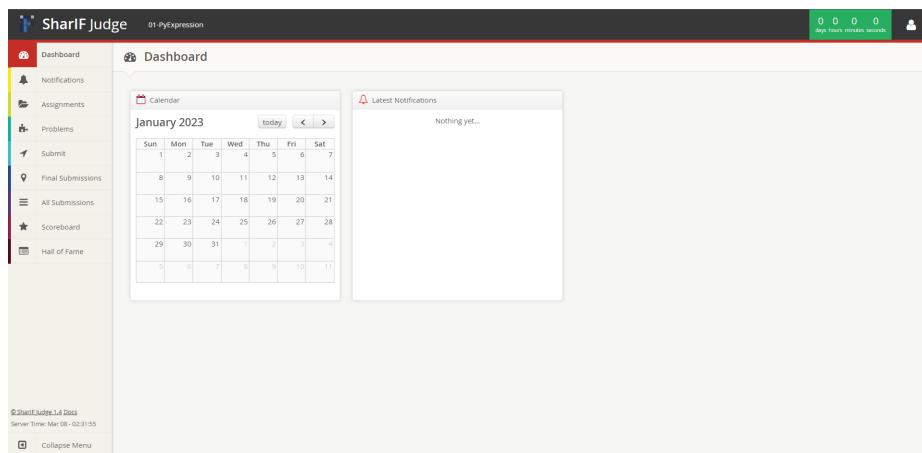
# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Tugas merupakan suatu bentuk pembelajaran dan penilaian yang diberikan oleh pengajar kepada pelajar untuk membantu pelajar mendalami materi yang sudah diberikan[3]. Pembagian tugas yang diberikan dapat dibagi menjadi 2 jenis yakni tugas individu dan tugas kelompok. Tugas individu merupakan tugas yang hanya ditanggung oleh satu individu. Sedangkan tugas kelompok merupakan tugas yang ditanggung oleh beberapa individu. Tugas yang telah diterima selanjutnya akan dikumpulkan kepada pengajar dan diberikan penilaian berdasarkan tingkat ketepatan jawaban dari tugas tersebut. Pengumpulan dan pengecekan tugas terutama *coding* secara manual memiliki kekurangan dimana diperlukan banyak langkah dalam melakukan pengecekan dan pengiriman nilai. Selain itu, pengecekan secara manual juga terdapat kekurangan yakni kesulitan dalam menentukan pengecekan plagiarisme antara tugas pelajar. Oleh karena itu, dibutuhkan perangkat lunak untuk melakukan pengecekan secara otomatis salah satunya adalah *Online Judge*.

*Online Judge* merupakan sebuah perangkat lunak yang dapat melakukan pengecekan program sesuai dengan standar yang sudah diberikan. Perangkat lunak dapat menerima jawaban dari pelajar dan melakukan pengecekan secara otomatis. Perangkat lunak selanjutnya memberikan keluaran berupa nilai dari pelajar tersebut[4]. Salah satu perangkat lunak *Online Judge* terdapat pada prodi Informatika Universitas Katolik Parahyangan bernama *SharIF Judge*. Gambar 1.1 menunjukkan tampilan perangkat lunak *SharIF Judge* yang terletak pada Universitas Katolik Parahyangan.



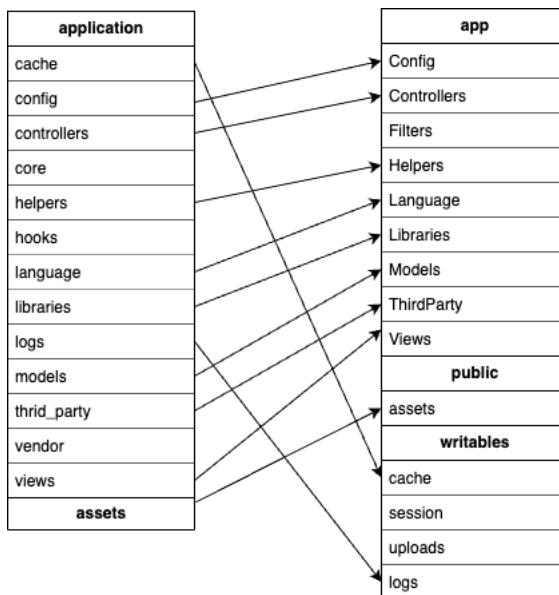
Gambar 1.1: Tampilan halaman *SharIF Judge*

*SharIF Judge* pada awalnya bernama *Sharif Judge* yang merupakan sebuah perangkat lunak *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* [5]. *Sharif Judge* kemudian di *fork* dan dimodifikasi menjadi *SharIF Judge* dengan

penambahan fungsi sesuai kebutuhan Informatika UNPAR untuk mengumpulkan tugas dan ujian mahasiswa[5]. *SharIF Judge* menggunakan autentikasi *Remote Authentication Dial In Service* (RADIUS) dan *Light Weight Directory Protocol* (LDAP) yang memungkinkan autentikasi menuju *database* pusat pada RADIUS dan direktori pada LDAP. Autentikasi ini membutuhkan *server* yang terintegrasi untuk menyimpan data-data yang diperlukan.

*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* cepat karena hanya membutuhkan sedikit sumber daya untuk menjalankannya. *Framework* ini juga menyediakan banyak *library* untuk melakukan pembangunan[1]. Namun, *CodeIgniter 3* sudah memasuki fase *maintenance*<sup>1</sup> 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*.

*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[2]. Perubahan direktori ditunjukkan oleh Gambar 1.2.



Gambar 1.2: Pemindahan struktur aplikasi menuju *CodeIgniter 4*

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

## 1.2 Rumusan Masalah

- Bagaimana cara melakukan konversi *SharIF Judge* pada *CodeIgniter 3* menjadi *CodeIgniter 4*?
- Bagaimana cara mengimplementasikan hasil analisis konversi pada *framework* *CodeIgniter 3* ke *CodeIgniter 4* pada aplikasi *SharIF Judge*.

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

- Bagaimana cara menguji *SharIF Judge* berbasis *CodeIgniter 4*?

### 1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

- Melakukan konversi dengan megubah kode sesuai dengan standar *CodeIgniter 4*.
- Mengimplementasikan hasil analisis konversi pada *framework CodeIgniter 3* ke *CodeIgniter 4* pada aplikasi *SharIF Judge*.
- Melakukan pengujian dan perbandingan setiap fitur pada *SharIF Judge* berbasis *CodeIgniter 3* dengan *SharIF Judge* berbasis *CodeIgniter 4*.

### 1.4 Batasan Masalah

Batasan masalah pada pembentukan tugas akhir ini adalah Autentikasi RADIUS dan LDAP tidak diuji karena tidak memiliki *server* yang terintegrasi.

### 1.5 Metodologi

Metodologi yang dilakukan dalam melakukan penelitian ini adalah sebagai berikut:

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

### 1.6 Sistematika Pembahasan

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

1. **Bab 1:** Pendahuluan

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

2. **Bab 2:** Landasan Teori

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

3. **Bab 3:** Analisis

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

4. **Bab 4:** Perancangan

Bab ini berisi mengenai rancangan perangkat lunak yang akan dikonversi.

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

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

6. **Bab 6:** Kesimpulan dan Saran

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

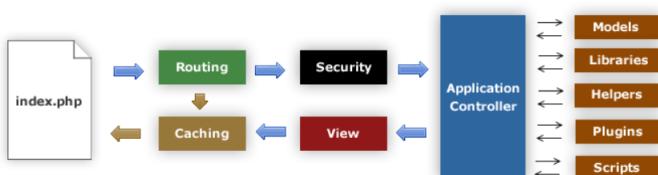


## BAB 2

### LANDASAN TEORI

#### 2.1 *CodeIgniter 3* [1]

*CodeIgniter 3* merupakan sebuah *framework opensource* yang berfungsi untuk mempermudah pengguna dalam membangun aplikasi *website* menggunakan bahasa PHP. *CodeIgniter 3* juga merupakan *framework* ringan yang menggunakan struktur *Model-View-Controller*, dan menghasilkan format URL tanpa *extension view* tersebut. *CodeIgniter 3* memiliki tujuan untuk membantu pengguna dalam membangun aplikasi web lebih cepat dengan menyediakan beragam *library* dan tampilan dan *logic* yang simpel. *Code Igniter 3* memiliki *flow chart* aplikasi yang ditunjukkan pada Gambar 2.1.



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

Berikut merupakan pembagian *flow chart* aplikasi *CodeIgniter 3* seperti yang ditunjukkan pada Gambar 2.1:

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

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

Sintaks diatas akan memuat *model* dengan nama *model\_name*. Selanjutnya, pengguna dapat membangun 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*

```
1 class Blog_model extends CI_Model {
2
3     public $title;
4     public $content;
5     public $date;
6
7     public function get_last_ten_entries()
8     {
9         $query = $this->db->get('entries', 10);
10        return $query->result();
11    }
12
13    public function insert_entry()
14    {
15        $this->title    = $_POST['title']; // please read the below note
16        $this->content  = $_POST['content'];
17        $this->date     = time();
18
19        $this->db->insert('entries', $this);
20    }
21
22    public function update_entry()
23    {
24        $this->title    = $_POST['title'];
25        $this->content  = $_POST['content'];
26        $this->date     = time();
27
28        $this->db->update('entries', $this, array('id' => $_POST['id']));
29    }
30}
31}
```

Berikut merupakan penjelasan beberapa fungsi pada Kode 2.1:

- *get\_last\_ten\_entries()* yang berfungsi untuk mengambil dan mengembalikan 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*.

## Views

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

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

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

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

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

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

### **Controller**

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

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

```
1 <?php
2 class Blog extends CI_Controller {
3
4     public function index()
5     {
6         echo 'Hello_World!';
7     }
8
9     public function comments()
10    {
11        echo 'Look_at_this!';
12    }
13 }
```

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

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

Metode *index* akan secara otomatis dipanggil menjadi URL dan pengguna juga dapat memberi 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

```

1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteCond %{REQUEST_FILENAME} !-d
4 RewriteRule ^(.*)$ index.php/$1 [L]

```

Metode *negative* memungkinkan htaccess untuk melakukan pengecekan menggunakan kondisi negatif `!-f` dan `!-d`. Aturan pada Kode 2.4 menyebabkan *HTTP Request* selain yang terdapat pada 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 menunjukkan contoh URL sebelum dan sesudah ditambahkan akhiran berupa `.html`:

Kode 2.5: Contoh URL sebelum dan sesudah ditambahkan akhiran

```

1 example.com/index.php/products/view/shoes
2
3 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

```

1 $config['enable_query_strings'] = FALSE;
2 $config['controller_trigger'] = 'c';
3 $config['function_trigger'] = 'm';

```

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

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

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

### 2.1.3 Helpers

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

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

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

### 2.1.4 Libraries

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

Kode 2.7: Contoh kelas library pada *CodeIgniter 3*

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3

```

```

4 class Someclass {
5     public function some_method()
6     {
7     }
8 }
9

```

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

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

```

1 $params = array('type' => 'large', 'color' => 'red');
2
3 $this->load->library('someclass', $params);

```

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

### Kelas *Input*

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

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

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

### *JavaScript*

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

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

Pengguna selanjutnya harus melakukan inisiasi *library* pada halaman *view tag <head>*. Kode 2.9 menunjukkan penambahan *tag* pada halaman *view*.

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

```

1 <?php echo $library_src;?>
2
3 <?php echo $script_head;?>

```

Sintaks *\$library\_src* merupakan lokasi *library* yang akan dimuat sedangkan *\$script\_head* merupakan lokasi untuk fungsi yang akan dijalankan. Selanjutnya *javascript* dapat diinisiasi pada *controller* menggunakan sintaks berikut:

```
$this->javascript
```

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

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

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

```
$this->jquery
```

## Kelas *Email*

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

- Beberapa Protokol: *Mail*, *Sendmail*, dan *Send Mail Transfer Protocol(SMTP)*
- Enkripsi *Transport Layer Security(TLS)* dan *Secure Sockets Layer(SSL)* untuk SMTP
- Beberapa Penerima
- *Carbon Copy(CC)* dan *Blind Carbon Copy(BCC)*
- HTML atau *email* teks biasa
- Lampiran
- Pembungkus kata
- Prioritas
- Mode BCC *Batch*, memisahkan daftar *email* besar menjadi beberapa BCC kecil.
- Alat *Debugging email*

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

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

```

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

```

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

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

```

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

```

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

## Kelas *File Uploading*

Pengunggahan *file* terdapat empat urutan proses sebagai berikut:

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

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

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

```
<?php
```

```

2| class Upload extends CI_Controller {
3|
4|     public function __construct()
5|     {
6|         parent::__construct();
7|         $this->load->helper('array', 'url');
8|     }
9|
10|    public function index()
11|    {
12|        $this->load->view('upload_form', array('error' => ' '));
13|    }
14|
15|    public function do_upload()
16|    {
17|        $config['upload_path']          = './uploads/';
18|        $config['allowed_types']        = 'gif|jpg|png';
19|        $config['max_size']             = 100;
20|        $config['max_width']            = 1024;
21|        $config['max_height']           = 768;
22|
23|        $this->load->library('upload', $config);
24|
25|        if ( ! $this->upload->do_upload('userfile'))
26|        {
27|            $error = array('error' => $this->upload->display_errors());
28|
29|            $this->load->view('upload_form', $error);
30|        }
31|        else
32|        {
33|            $data = array('upload_data' => $this->upload->data());
34|
35|            $this->load->view('upload_success', $data);
36|        }
37|
38|    }
39}
40?>
```

Kode 2.12 menunjukan 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.

### **Session**

*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:

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

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

```
$this->session
```

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

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

Sintaks diatas akan mengambil seluruh data *session* dengan nama *item*. Secara *default*, aplikasi akan menyimpan data *session* pada *file* namun pengguna dapat mengubah konfigurasi sesuai dengan kebutuhan. Kode 2.13 menunjukan konfigurasi untuk menyimpan data pada *database* dengan nama tabel *ci\_sessions*.

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

```

1 $config['sess_driver'] = 'database';
2 $config['sess_save_path'] = 'ci_sessions';
```

## Kelas *URI*

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

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

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*.

## Kelas *Zip Encoding*

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

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

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

```
$this->zip
```

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

Kode 2.14: Contoh penggunaan *library Zip Encoding*

```
1 $name = 'mydata1.txt';
2 $data = 'A_Data_String!';
3
4 $this->zip->add_data($name, $data);
5
6 // Write the zip file to a folder on your server. Name it "my_backup.zip"
7 $this->zip->archive('/path/to/directory/my_backup.zip');
8
9 // Download the file to your desktop. Name it "my_backup.zip"
10 $this->zip->download('my_backup.zip');
```

Selain menggunakan *library* yang sudah disediakan, pengguna dapat membangun dan memperluas *libraries* sendiri sesuai dengan kebutuhan. Kode 2.15 merupakan contoh *library* yang dibentuk oleh 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 }
```

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* untuk *database* bernama *database\_name* dengan *username* *root* tanpa sebuah *password*.

Kode 2.16: Contoh konfigurasi *database*

```

1 $db['default'] = array(
2     'dsn' => '',
3     'hostname' => 'localhost',
4     'username' => 'root',
5     'password' => '',
6     'database' => 'database_name',
7     'dbdriver' => 'mysqli',
8     'dbprefix' => '',
9     'pconnect' => TRUE,
10    'db_debug' => TRUE,
11    'cache_on' => FALSE,
12    'cachedir' => '',
13    'char_set' => 'utf8',
14    'dbcollat' => 'utf8_general_ci',
15    'swap_pre' => '',
16    'encrypt' => FALSE,
17    'compress' => FALSE,
18    'stricton' => FALSE,
19    'failover' => array()
20 );

```

*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*

```

1 $this->db->select('*');
2 $this->db->from('blogs');
3 $this->db->join('comments', 'comments.id=_blogs.id');
4 $query = $this->db->get();

```

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

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

Setelah dilakukan inisiasi pengguna dapat membangun *database* menggunakan kelas **Forge**. Kode 2.18 menunjukkan contoh untuk membangun *database* dengan nama *db\_name*.

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

```

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

```

Selain itu, pengguna juga dapat menambahkan kolom dengan konfigurasinya. Kode 2.19 menunjukkan contoh penambahan kolom sesuai dengan kebutuhan pengguna.

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

```

1 $fields = array(
2     'blog_id' => array(
3         'type' => 'INT',
4         'constraint' => 5,
5         'unsigned' => TRUE,
6         'auto_increment' => TRUE
7     ),
8     'blog_title' => array(
9         'type' => 'VARCHAR',
10        'constraint' => '100',
11        'unique' => TRUE,
12    ),
13    'blog_author' => array(
14        'type' => 'VARCHAR',
15        'constraint' => '100',
16        'default' => 'King_of_Town',
17    ),
18    'blog_description' => array(
19        'type' => 'TEXT',
20        'null' => TRUE,

```

```

21     ),
22 );
23 $this->dbforge->add_field($fields)
24 $this->dbforge->create_table('table_name');

```

Kode 2.19 menunjukkan contoh penggunaan *database* untuk menambahkan kolom sesuai dengan tipe, batas dari data yang disimpan, penambahan otomatis, dan isi *default* dari kolom pada tabel *table\_name*.

### 2.1.6 URI Routing

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

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

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

Kode 2.20: Contoh URL yang sudah dimetakan

```

1 example.com/product/1/
2 example.com/product/2/
3 example.com/product/3/
4 example.com/product/4/

```

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

#### WildCards

*Route wildcard* biasanya berisikan kode seperti berikut:

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

*Route* diatas dibagi menjadi dua buah yakni:

1. Bagian segmen URL

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

2. Bagian kelas dan metode

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

#### Ekspresi Reguler

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

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

Ekspresi diatas menghasilkan URI *products/shirts/123* yang memanggil kelas *controller* dan metode *id\_123*. Pengguna juga dapat mengambil segmen banyak seperti berikut:

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

### 2.1.7 Error Handling

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

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

Sintaks diatas menerima parameter berupa pesan, kode *HTTP*, dan *heading* yang ditampilkan pada halaman *error*. Sintaks ini akan mengembalikan halaman *error* dengan nama **error\_general.php**.

### 2.1.8 Auto-loading

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

## 2.2 SharIF Judge

*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.

### 2.2.1 Struktur Aplikasi

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

Kode 2.21: Struktur aplikasi *SharIF Judge*

```

1 -- application
2   |-- cache
3   |-- config
4   |-- controllers
5   |-- core
6   |-- helpers
7   |-- hooks
8   |-- language
9   |-- libraries
10  |-- logs
11  |-- models
12  |-- third_party
13  |-- vendor
14  |-- views
15  |-- assets
16    |-- images
17    |-- js
18    |-- styles
19  |-- restricted
20    |-- tester
21  |-- system
22  |-- assignments

```

### 2.2.2 Instalasi

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

## Persyaratan

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

- Diperlukan *webserver* dengan versi PHP 5.3 atau lebih baru.
- Pengguna dapat menjalankan PHP pada *command line*. Pada *Ubuntu* diperlukan instalasi paket *php5-cli*.
- *MySQL database* dengan ekstensi *Mysqli* untuk PHP atau *PostgreSQL database*.
- PHP harus memiliki akses untuk menjalankan perintah melalui fungsi *shell\_exec*. Kode 2.22 menunjukkan sintaks untuk melakukan pengetesan fungsi PHP pada *shell*.

Kode 2.22: Kode untuk melakukan pengetesan fungsi

```
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.

## 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 menunjukkan contoh path pada file *index.php*.

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

```
1 $system_path = '/home/mohammad/secret/system';
2 application_folder = '/home/mohammad/secret/application';
```

3. Membangun *database MySQL* atau *PostgreSQL* untuk *SharIF Judge*. Jangan melakukan instalasi paket koneksi *database* apapun untuk *C*, *C++*, *Java*, atau *Python*.
4. 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 menunjukkan contoh pengaturan koneksi untuk nama, *host*, dan prefix pada *database*.

Kode 2.24: Contoh pengaturan koneksi untuk *database*

```
1 /* Enter database connection settings here: */
2 'dbdriver' => 'postgre', // database driver (mysqli, postgre)
3 'hostname' => 'localhost', // database host
4 'username' => '', // database username
5 'password' => '', // database password
6 'database' => '', // database name
7 'dbprefix' => 'shj_', // table prefix
```

5. Mengatur konfigurasi RADIUS *server* dan *mail server* pada file *application/config/secrets.example.php* dan menyimpannya dengan nama *secrets.php*.
6. Mengatur *application/cache/Twig* agar dapat ditulis oleh PHP.
7. Membuka halaman utama *SharIF Judge* pada *web browser* dan mengikuti proses instalasi.
8. Melakukan *login* dengan akun admin.
9. 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.

### 2.2.3 Clean URLs

Secara *default*, *index.php* merupakan bagian dari seluruh URL pada *SharIF Judge*. Kode 2.25 menunjukkan contoh dari URL *SharIF Judge*.

Kode 2.25: Contoh URL SharIF Judge

```

1 http://example.mjnaderi.ir/index.php/dashboard
2
3 http://example.mjnaderi.ir/index.php/users/add

```

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

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

```

1 http://example.mjnaderi.ir/dashboard
2
3 http://example.mjnaderi.ir/users/add

```

### Cara Mengaktifkan *Clean URLs*

Pengguna dapat mengaktifkan *clean URL* dengan cara sebagai berikut:

- Mengganti nama file `.htaccess2` pada direktori utama menjadi `.htaccess`.
- Mengganti `$config['index_page'] = 'index.php';` menjadi `$config['index_page'] = '';` pada file `application\config\config.php`.

#### 2.2.4 Users

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

Tabel 2.1: Tabel tingkat pengguna

User Role	User Level
Admin	3
Head Instructor	2
Instructor	1
Student	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>	✓	✓	✓	✓

## Menambahkan Pengguna

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

Kode 2.27: Sintaks untuk menambahkan pengguna

```

1 # USERNAME,EMAIL,DISPLAY-NAME,PASSWORD,ROLE
2
3 * Usernames may contain lowercase letters or numbers and must be between 3 and 20 characters in length.
4 * Passwords must be between 6 and 30 characters in length.
5 * You can use RANDOM[n] for password to generate random n-digit password.
6 * ROLE must be one of these: 'admin', 'head_instructor', 'instructor', 'student'

```

Kode 2.28 menunjukkan contoh untuk menambahkan pengguna.

Kode 2.28: Contoh kode untuk menambahkan pengguna

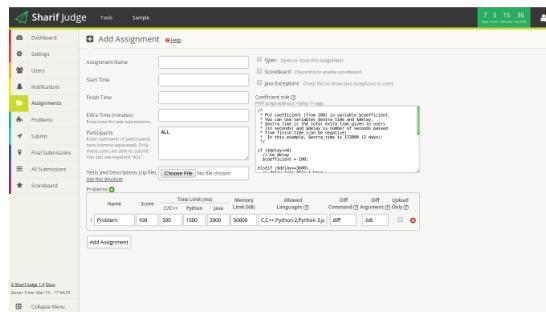
```

1 # This is a comment!
2 # This is another comment!
3 instructor,instructor@sharifjudge.ir,Instructor One,123456,head_instructor
4 instructor2,instructor2@sharifjudge.ir,Instructor Two,random[7],instructor
5 student1,st1@sharifjudge.ir,Student One,random[6],student
6 student2,st2@sharifjudge.ir,Student Two,random[6],student
7 student3,st3@sharifjudge.ir,Student Three,random[6],student
8 student4,st4@sharifjudge.ir,Student Four,random[6],student
9 student5,st5@sharifjudge.ir,Student Five,random[6],student
10 student6,st6@sharifjudge.ir,Student Six,random[6],student
11 student7,st7@sharifjudge.ir,Student Seven,random[6],student

```

### 2.2.5 Menambah Assignment

Pengguna dapat menambahkan *assignment* baru melalui bagian *Add* pada halaman *Assignments* yang ditunjukan pada Gambar 2.2.



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

Berikut merupakan beberapa pengaturan pada halaman *Add Assignments*:

- ***Assignment Name***

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

- ***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.

- ***Finish Time, Extra Time***

Pengguna tidak dapat mengumpulkan *assignment* setelah *Finish Time* dan *Extra Time* berakhir. Pengumpulan *Assignment* pada *Extra Time* akan dikalikan sesuai dengan koefisien. Pengguna harus menulis skrip PHP untuk menghitung koefisien pada field *Coefficient Rule*. Format

pengaturan waktu untuk waktu selesai sama seperti waktu mulai yakni MM/DD/YYYY HH:MM:SS dan format waktu tambahan menggunakan menit dengan contoh 120 (2 jam) atau 48\*60 (2 hari).

- ***Participants***

Pengguna dapat memasukan *username* setiap partisipan atau menggunakan kata kunci *ALL* untuk membiarkan seluruh pengguna melakukan pengumpulan. Contoh: `admin, instructor1, instructor2, student1, student2, student3, student4.`

- ***Tests***

Pengguna dapat mengunggah *test case* dalam bentuk *zip file* sesuai dengan struktur pada subbab 2.2.7.

- ***Open***

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

- ***Score Board***

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

- ***Java Exceptions***

Pengguna dapat menghidupkan dan mematikan fungsi untuk menunjukan *java exceptions* kepada pengguna *students* dan tidak akan memengaruhi kode yang sudah di *judge* sebelumnya. Kode 2.29 menunjukan tampilan apabila fitur *java exceptions* dinyalakan:

Kode 2.29: Contoh tampilan fitur *Java Exceptions*

```

1 Test 1
2 ACCEPT
3 Test 2
4 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
5 Test 3
6 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
7 Test 4
8 ACCEPT
9 Test 5
10 ACCEPT
11 Test 6
12 ACCEPT
13 Test 7
14 ACCEPT
15 Test 8
16 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
17 Test 9
18 Runtime Error (java.lang.StackOverflowError)
19 Test 10
20 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)

```

- ***Archived Assignment***

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

- ***Coefficient Rule***

Pengguna dapat menuliskan skrip PHP pada bagian ini yang akan menghitung koefisien dikalikan dengan skor. Pengguna harus memasukan koefisien dari 100 dalam variabel `$coefficient`.

Pengguna dapat menggunakan variabel `$extra_time` dan `$delay`. `$extra_time` merupakan total dari waktu tambahan yang diberikan kepada pengguna dalam detik sedangkan `$delay` merupakan waktu dalam detik yang melewati waktu selesai yang dapat diisikan berupa negatif. Skrip PHP pada bagian ini tidak boleh mengandung tag `<?php`, `<?`, dan `?>`. Kode 2.30 menunjukkan contoh skrip dimana `$extra_time` adalah 172800 detik atau 2 hari.

Kode 2.30: Contoh skrip PHP

```

1 if ($delay<=0)
2 // no delay
3 $coefficient = 100;
4
5 elseif ($delay<=3600)
6 // delay less than 1 hour
7 $coefficient = ceil(100-((30*$delay)/3600));
8
9 elseif ($delay<=86400)
10 // delay more than 1 hour and less than 1 day
11 $coefficient = 70;
12
13 elseif ((($delay-86400)<=3600)
14 // delay less than 1 hour in second day
15 $coefficient = ceil(70-((20*($delay-86400))/3600));
16
17 elseif ((($delay-86400)<=86400)
18 // delay more than 1 hour in second day
19 $coefficient = 50;
20
21 elseif ($delay > $extra_time)
22 // too late
23 $coefficient = 0;

```

- **Name**

Merupakan nama dari masalah pada *assignments*.

- **Score**

Merupakan skor dari masalah pada *assignments*.

- **Time Limit**

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

- **Memory Limit**

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

- **Allowed Languages**

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

- **Diff Command**

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

- ***Diff Arguments***

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

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

- ***Upload Only***

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

### 2.2.6 Sample Assignment

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

#### Problems

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

2. *Problem 2 (Max)*: Program pengguna dapat membaca *integer n*, membaca *n integer*, dan mengeluarkan total dari dua buah *integer* terbesar diantara *n integer*. Tabel 2.4 menunjukkan contoh pemasukan 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

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

#### Tests

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

Kode 2.31: Susunan pohon untuk ketiga *problems*

```

1 .
2 |-- p1
3 |   |-- in
4 |   |   |-- input1.txt
5 |   |   |-- input2.txt

```

```

6 |     |-- input3.txt
7 |     |-- input4.txt
8 |     |-- input5.txt
9 |     |-- input6.txt
10 |    |-- input7.txt
11 |    |-- input8.txt
12 |    |-- input9.txt
13 |    --- input10.txt
14 |   -- out
15 |   --- output1.txt
16 |   -- tester.cpp
17 |   --- desc.md
18 -- p2
19 |   -- in
20 |     |-- input1.txt
21 |     |-- input2.txt
22 |     |-- input3.txt
23 |     |-- input4.txt
24 |     |-- input5.txt
25 |     |-- input6.txt
26 |     |-- input7.txt
27 |     |-- input8.txt
28 |     |-- input9.txt
29 |     --- input10.txt
30 |   -- out
31 |     |-- output1.txt
32 |     |-- output2.txt
33 |     |-- output3.txt
34 |     |-- output4.txt
35 |     |-- output5.txt
36 |     |-- output6.txt
37 |     |-- output7.txt
38 |     |-- output8.txt
39 |     |-- output9.txt
40 |     --- output10.txt
41 |   -- desc.md
42 |   --- Problem2.pdf
43 -- p3
44 |   --- desc.md
45 --- SampleAssignment.pdf

```

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

### Sample Solutions

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

Kode 2.32 menunjukkan contoh solusi *problem 1* untuk bahasa *C*.

Kode 2.32: Contoh solusi *problem 1* bahasa *C*

```

1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     int i;
6     int sum =0 ;
7     int k;
8     for(i=0 ; i<n ; i++){
9         scanf("%d",&k);
10        sum+=k;
11    }
12    printf("%d\n",sum);
13    return 0;
14 }

```

Kode 2.33 menunjukkan contoh solusi *problem 1* untuk bahasa *C++*.

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

```

1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     int i;
6     int sum =0 ;
7     int k;
8     for(i=0 ; i<n ; i++){
9         scanf("%d",&k);
10        sum+=k;
11    }
12    printf("%d\n",sum);
13    return 0;
14 }

```

Kode 2.34 menunjukkan contoh solusi *problem 1* untuk bahasa *Java*.

Kode 2.34: Contoh solusi *problem 1* bahasa Java

```

1 import java.util.Scanner;
2 class sum
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8         int sum =0;
9         for (int i=0 ; i<n ; i++)
10        {
11            int a = sc.nextInt();
12            sum += a;
13        }
14        System.out.println(sum);
15    }
16 }
```

Berikut merupakan contoh solusi untuk *problem 2*:

Kode 2.35 menunjukan contoh solusi *problem 2* untuk bahasa C

Kode 2.35: Contoh solusi *problem 2* bahasa C

```

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

Kode 2.36 menunjukan contoh solusi *problem 2* untuk bahasa C++

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

```

1 #include<iostream>
2 using namespace std;
3 int main(){
4     int n , m1=0, m2=0;
5     cin >> n;
6     for(;n--){ 
7         int k;
8         cin >> k;
9         if(k>=m1){
10            m2=m1;
11            m1=k;
12        }
13        else if(k>m2)
14            m2=k;
15    }
16    cout << (m1+m2) << endl ;
17    return 0;
18 }
```

## 2.2.7 Test Structure

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

### Metode Pemeriksaan

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

#### 1. Metode perbandingan *Input Output*

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

## 2. Metode perbandingan *Tester*

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*

```

1  /*
2   * tester.cpp
3   */
4
5 #include <iostream>
6 #include <fstream>
7 #include <string>
8 using namespace std;
9 int main(int argc, char const *argv[])
10 {
11
12     ifstream test_in(argv[1]); /* This stream reads from test's input file */
13     ifstream test_out(argv[2]); /* This stream reads from test's output file */
14     ifstream user_out(argv[3]); /* This stream reads from user's output file */
15
16     /* Your code here */
17     /* If user's output is correct, return 0, otherwise return 1 */
18
19     ...
20 }
21 
```

### *Sample File*

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*

```

1  -- p1
2  |-- in
3  |  |-- input1.txt
4  |  |-- input2.txt
5  |  |-- input3.txt
6  |  |-- input4.txt
7  |  |-- input5.txt
8  |  |-- input6.txt
9  |  |-- input7.txt
10 |  |-- input8.txt
11 |  |-- input9.txt
12 |  --- input10.txt
13 |  -- out
14 |    --- output1.txt
15 |  -- tester.cpp
16 -- p2
17 |-- in
18 |  |-- input1.txt
19 |  |-- input2.txt
20 |  |-- input3.txt
21 |  |-- input4.txt
22 |  |-- input5.txt
23 |  |-- input6.txt
24 |  |-- input7.txt
25 |  |-- input8.txt
26 |  |-- input9.txt
27 |  --- input10.txt
28 |  -- out
29 |    --- output1.txt
30 |    --- output2.txt
31 |    --- output3.txt
32 |    --- output4.txt
33 |    --- output5.txt
34 |    --- output6.txt
35 |    --- output7.txt
36 |    --- output8.txt
37 |    --- output9.txt
38 |    --- output10.txt
39 
```

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

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

```

1  /*
2   * tester.cpp
3   */
4
5 #include <iostream>
6 #include <fstream>
7 #include <string>
8 using namespace std;
9 int main(int argc, char const *argv[])
10 {
11
12     ifstream test_in(argv[1]);    /* This stream reads from test's input file */
13     ifstream test_out(argv[2]);   /* This stream reads from test's output file */
14     ifstream user_out(argv[3]);  /* This stream reads from user's output file */
15
16     /* Your code here */
17     /* If user's output is correct, return 0, otherwise return 1 */
18     /* e.g.: Here the problem is: read n numbers and print their sum: */
19
20     int sum, user_output;
21     user_out >> user_output;
22
23     if ( test_out.good() ) // if test's output file exists
24     {
25         test_out >> sum;
26     }
27     else
28     {
29         int n, a;
30         sum=0;
31         test_in >> n;
32         for (int i=0 ; i<n ; i++){
33             test_in >> a;
34             sum += a;
35         }
36     }
37
38     if (sum == user_output)
39         return 0;
40     else
41         return 1;
42 }
43

```

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

## 2.2.8 Deteksi Kecurangan

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

- 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 menunjukkan contoh dari potongan skrip *perl*.

Kode 2.40: Contoh potongan skrip *perl*

```

1 ...
2 ...
3
4 $server = 'moss.stanford.edu';
5 $port = '7690';
6 $nreq = "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.

### 2.2.9 Keamanan

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

#### 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](#).

#### 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

```

1 if $TIMEOUT_EXISTS; then
2     timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
3 else
4     $CMD <$IN >out 2>err
5 fi

```

menjadi Kode [2.42](#).

Kode 2.42: Kode *runcode.sh* setelah ditambahkan

```

1 if $TIMEOUT_EXISTS; then
2     sudo -u restricted_user timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
3 else
4     sudo -u restricted_user $CMD <$IN >out 2>err
5 fi

```

dan *uncomment* baris yang ditunjukan pada Kode [2.43](#).

Kode 2.43: Kode *runcode.sh* awal

```

1 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 ditunjukkan pada Kode 2.44.

Kode 2.44: Contoh kode untuk membatasi jumlah proses

```

1 restricted_user    soft    nproc   3
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.

##### 2.2.10 Sandboxing

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

##### C/C++ Sandboxing

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

File *EasySandbox* terdapat pada direktori *tester/easyandbox*. Untuk membangun *EasySandbox* jalankan:

Kode 2.45: Kode *runcode.sh* awal

```

1 $ cd tester/easyandbox
2 $ chmod +x runalltests.sh
3 $ chmod +x runtest.sh
4 $ make runtests

```

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

##### Java Sandboxing

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

### 2.2.11 *Shield*

*Shield* merupakan mekanisme sangat simpel untuk mlarang jalannya kode yang berpotensi berbahaya. *Shield* bukan solusi *sandboxing* karena *shield* hanya menyediakan proteksi sebagian terhadap serangan kecil. Proteksi utama terhadap kode tidak terpercaya adalah dengan menghidupkan *Sandbox* yang dapat dilihat pada subbab 2.2.10.

#### *Shield* untuk C/C++

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

Kode 2.46: Kode *shield* untuk mlarang penggunaan `goto`

```
1 #define goto YouCannotUseGoto
```

Dengan Kode 2.46, semua kode yang menggunakan `goto` akan mendapatkan *compilation error*. Apabila pengguna menghidupkan *shield*, semua kode yang mengandung `#undef` akan mendapatkan *compilation error*. *Shield* untuk *C* dan *C++* dapat dikonfigurasikan menggunakan cara berikut:

- Menghidupkan *shield* untuk *C/C++*  
Pengguna dapat menghidupkan atau mematikan *shield* pada halaman *settings*.
- Menambahkan aturan untuk *C/C++*  
Daftar aturan `#define` untuk bahasa *C* terdapat pada halaman *tester/shield/defc.h* dan *tester/shield/defcpp.h* untuk bahasa *C++*. Pengguna dapat menambahkan aturan baru pada *file* tersebut pada halaman *settings*. Kode refkode:define menunjukkan contoh sintaks pada untuk menambahkan aturan *shield*.

Kode 2.47: Sintaks aturan `#define`

```
1 /*
2
3 @file defc.h
4 There should be a newline at end of this file.
5 Put the message displayed to user after // in each line
6
7 e.g. If you want to disallow goto, add this line:
8 #define goto errorNo13 //Goto is not allowed
9
10 */
11
12 #define system errorNo1 // "system" is not allowed
13 #define freopen errorNo2 //File operation is not allowed
14 #define fopen errorNo3 //File operation is not allowed
15 #define fprintf errorNo4 //File operation is not allowed
16 #define fscanf errorNo5 //File operation is not allowed
17 #define feof errorNo6 //File operation is not allowed
18 #define fclose errorNo7 //File operation is not allowed
19 #define ifstream errorNo8 //File operation is not allowed
20 #define ofstream errorNo9 //File operation is not allowed
21 #define fork errorNo10 //Fork is not allowed
22 #define clone errorNo11 //Clone is not allowed
23 #define sleep errorNo12 //Sleep is not allowed
```

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

#### *Shield* untuk *Python*

Penggunaan *shield* untuk *python* dapat dihidupkan melalui halaman *settings*. Dengan menghidupkan *shield* untuk *python*, *SharIF Judge* hanya menambahkan beberapa kode pada baris awal kode yang sudah dikumpulkan sebelum dijalankan. Penambahan kode digunakan untuk mencegah pemakaian fungsi berbahaya. Kode-kode tersebut terletak pada *file tester/shield/shield\_py2.py* dan *tester/shield/shield\_py3.py*. Kode 2.48 menunjukkan cara untuk keluar dari *shield* untuk *python* menggunakan fungsi yang telah di daftar hitamkan.

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

```

1 # @file shield_py3.py
2
3 import sys
4 sys.modules['os']=None
5
6 BLACKLIST = [
7     '__import__', # deny importing modules
8     'eval', # eval is evil
9     'open',
10    'file',
11    'exec',
12    'execfile',
13    'compile',
14    'reload',
15    #'input'
16 ]
17 for func in BLACKLIST:
18     if func in __builtins__._dict_:
19         del __builtins__._dict_[func]
20

```

### 2.3 CodeIgniter 4[2]

*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 menyimpan 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.

- *tests/*

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

- `vendor/` atau `system/`

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

### 2.3.1 Models- Views- Controllers

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

#### Models

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

Kode 2.49: Contoh *Models* pada *CodeIgniter 4*

```

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

```

Kode 2.49 merupakan contoh *file Models* yang dapat digunakan pada *controllers*. *Models* tersebut terkoneksikan dengan tabel `users` dengan *primarykey* `id`. *Model* pada *CodeIgniter 4* juga dapat digunakan untuk mencari, menyimpan, dan menghapus data untuk setiap tabel spesifik. Kode 2.50 menunjukkan contoh penggunaan *model* dengan menggabungkan *query* dengan metode pencarian *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();

```

#### Views

*Views* merupakan halaman berisikan HTML dan sedikit PHP yang ditampilkan kepada pengguna. *Views* juga dapat berupa pecahan halaman seperti *header*, *footer*, ataupun *sidebar*. *Views* biasanya

terdapat pada `app/Views` dan mendapatkan data berupa variabel dari `controller` untuk ditampilkan.

Kode 2.51: Contoh *Views* pada *CodeIgniter 4*

```

1 <html>
2   <head>
3     <title>My Blog</title>
4   </head>
5   <body>
6     <h1>Welcome to my Blog!</h1>
7   </body>
8 </html>
```

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

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

```

1 <?php
2 namespace App\Controllers;
3
4 use CodeIgniter\Controller;
5
6 class Blog extends Controller
7 {
8   public function index()
9   {
10    return view('blog_view');
11  }
12 }
```

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

### *Controllers*

*Controllers* merupakan kelas untuk mengambil atau memberikan data dari *models* menuju *views* untuk ditampilkan kepada pengguna. Setiap pembentukan *controllers* dibentuk harus memperpanjang kelas `BaseController`. Kode 2.53 menunjukkan contoh *controllers* yang dibentuk dengan nama `Helloworld`.

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

```

1 <?php
2 namespace App\Controllers;
3
4 class Helloworld extends BaseController
5 {
6   public function index()
7   {
8     return 'Hello_World!';
9   }
10
11  public function comment()
12  {
13    return 'I_am_not_flat!';
14  }
15 }
```

Kode 2.53 akan mengembalikan kata `Hello World` pada URL:

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

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*

```

1 <?php
2 namespace App\Filters;
3
4 use CodeIgniter\Filters\FilterInterface;
```

```

6| use CodeIgniter\HTTP\RequestInterface;
7| use CodeIgniter\HTTP\ResponseInterface;
8|
9| class MyFilter implements FilterInterface
10|
11| {
12|     public function before(RequestInterface $request, $arguments = null)
13|     {
14|         $auth = service('auth');
15|
16|         if (! $auth->isLoggedIn()) {
17|             return redirect()->to(site_url('login'));
18|         }
19|     }
}

```

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

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

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

Kode 2.55: Contoh mengakses data menggunakan *IncomingRequest*

```

1<?php
2
3namespace App\Controllers;
4
5use CodeIgniter\Controller;
6
7class UserController extends Controller
8{
9    public function index()
10    {
11        if ($this->request->isAJAX()) {
12            $something = $this->request->getVar('foo');
13        }
14    }
}

```

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

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

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

### 2.3.2 Autoloading Files

*CodeIgniter 4* menyediakan fitur *autoloader* yang dapat digunakan dengan sedikit konfigurasi. Fitur ini dapat menemukan *namespaced classes* individual yang menggunakan struktur direktori *autoloading* pada *PHP Standart Recommendation(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 menunjukan contoh konfigurasi menggunakan *namespace PSR-4* untuk melakukan *mapping* menuju direktori.

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

```

1<?php
2
3namespace Config;
4
5use CodeIgniter\Config\AutoloadConfig;
6

```

```

7| class Autoload extends AutoloadConfig
8|
9|     // ...
10|    public $psr4 = [
11|        APP_NAMESPACE => APPPATH, // For custom app namespace
12|        'Config'      => APPPATH . 'Config',
13|    ];
14|
15|    // ...
16}

```

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

```
php spark namespaces
```

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

Kode 2.57: Contoh konfigurasi menggunakan *classmap*.

```

1<?php
2
3namespace Config;
4
5use CodeIgniter\Config\AutoloadConfig;
6
7class Autoload extends AutoloadConfig
8{
9    // ...
10   public $classmap = [
11       'Markdown' => APPPATH . 'ThirdParty/markdown.php',
12   ];
13
14   // ...
15}

```

### 2.3.3 Configuration

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

Kode 2.58: Contoh mengakses *file configuration*.

```

1<?php
2// Creating new configuration object by hand
3$config = new \Config\Pager();
4
5// Get shared instance with config function
6$config = config('Pager');
7
8// Access config class with namespace
9$config = config('Config\\Pager');
10$config = config(\Config\Pager::class);
11
12// Creating a new object with config function
13$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 *namespace*, dan membangun objek baru menggunakan fungsi *config*. Selanjutnya pengguna dapat mengakses properti yang terdapat pada *file config* tersebut menggunakan sintaks berikut:

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

Sintaks diatas akan mengambil properti *perPage* yang terdapat pada variabel *config* yang sudah diinisiasi. Selain menggunakan *config* yang terdapat pada *CodeIgniter 4*, pengguna dapat membangun *file config* secara manual. Kode 2.59 menunjukkan contoh isi *file config* yang dibentuk manual.

Kode 2.59: Contoh pembentukan *file configuration*.

```

1 <?php
2
3 namespace Config;
4
5 use CodeIgniter\Config\BaseConfig;
6
7 class CustomClass extends BaseConfig
8 {
9     public $siteName = 'My_Great_Site';
10    public $siteEmail = 'webmaster@example.com';
11    // ...
12 }

```

*File config* ini akan disimpan pada direktori `app/Config` dengan nama kelas yang akan `extends` dengan `BaseConfig`. Selain menggunakan *file config*, pengguna juga dapat melakukan konfigurasi menggunakan variabel *environment*. Penggunaan *file environment* merupakan cara melakukan konfigurasi terbaik untuk saat ini dikarenakan kemudahan untuk mengubah konfigurasi pada saat melakukan *deploy* aplikasi. Konfigurasi dapat diubah tanpa harus mengubah kode. *CodeIgniter 4* menyediakan sebuah *file* bernama `dotenv` atau yang akan disebut `.env` selanjutnya. `.env` merupakan sebuah *file* yang terletak pada akar aplikasi. *File* ini dapat berisikan seluruh konfigurasi yang diperlukan oleh aplikasi. Berikut merupakan contoh variabel yang disimpan pada *file .env*.

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

```

1 S3_BUCKET = dotenv
2 SECRET_KEY = super_secret_key
3 CI_ENVIRONMENT = development

```

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

### 2.3.4 *CodeIgniter URLs*

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

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

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

- URL dasar merupakan URL dasar dari aplikasi web yang ditunjukan sebagai `baseURL`. URL dasar ini berbentuk `https://www.example.com/ci-blog/`.
- *URI Path* merupakan alamat yang dituju yaitu `/ci-blog/blog/news/2022/10`
- *Route* juga merupakan alamat yang dituju tanpa URL dasar yaitu `/blog/news/2022/10`
- *Query* merupakan hasil dari *query* yang ingin ditampilkan yaitu `page=2`

Secara *default*, *CodeIgniter 4* membangun URL dengan `index.php` namun, pengguna dapat menghapus *file index.php* pada URL yang dibentuk. Pengguna dapat menghapus `index.php` sesuai dengan *server* yang digunakan. Berikut merupakan contoh dua buah *server* yang umum dipakai:

#### *Apache Web Server*

Pengguna dapat URL melalui *file .htaccess* dengan menyalakan ekstensi `mod_rewrite`. Kode 2.61 menunjukkan contoh *file .htaccess* untuk menghapus `index.php` pada URL yang dibentuk.

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

```

1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteCond %{REQUEST_FILENAME} !-d
4 RewriteRule ^(.*)$ index.php/$1 [L]

```

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

### NGINX

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 `try_files` untuk menghapus *index.php* pada URL.

Kode 2.62: Contoh penggunaan `try-files`.

```
1 location / {
2     try_files $uri $uri/ /index.php$is_args$args;
3 }
```

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

### 2.3.5 Error Handling

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

Kode 2.63: Contoh penggunaan `error handling`.

```
1 throw new \Exception('Some_message_goes_here');
```

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

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

```
1 try {
2     $user = $userModel->find($id);
3 } catch (\Exception $e) {
4     exit($e->getMessage());
5 }
```

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

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

```
1 if (! $page = $pageModel->find($id)) {
2     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
3 }
```

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

### 2.3.6 URI Routing

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

### **Defined Route Routing**

Pengguna dapat mendefinisikan *route* secara manual untuk URL yang lebih fleksibel. Kode 2.66 menunjukkan contoh *route* yang didefinisikan secara manual untuk membangun URL menuju kelas **Catalog** dengan metode **productLookup**.

Kode 2.66: Contoh *route* yang didefinisikan secara manual

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

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

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

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

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

### **Auto Routing**

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

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

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

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

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

### **2.3.7 Database**

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

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

```
1 <?php
2
3 namespace Config;
4
5 use CodeIgniter\Database\Config;
6
7 class Database extends Config
8 {
9     public $default = [
10         'DSN' => '',
11         'hostname' => 'localhost',
```

```

12     'username' => 'root',
13     'password' => '',
14     'database' => 'database_name',
15     'DBDriver' => 'MySQLi',
16     'DBPrefix' => '',
17     'pConnect' => true,
18     'DBDebug' => true,
19     'charset' => 'utf8',
20     'DBCollat' => 'utf8_general_ci',
21     'swapPre' => '',
22     'encrypt' => false,
23     'compress' => false,
24     'strictOn' => false,
25     'failover' => [],
26     'port' => 3306,
27   ];
28
29 // ...
30 }

```

Selain itu, konfigurasi juga dapat dilakukan pada file .env untuk mempermudah dalam pengubahan data pada saat melakukan *deploy*. Kode 2.69 menunjukkan contoh konfigurasi pada file .env:

Kode 2.69: Contoh konfigurasi database pada file .env.

```

1 database.default.username = 'root';
2 database.default.password = '';
3 database.default.database = 'ci4';

```

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

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

```

1 <?php
2
3 $builder = $db->table('users');
4 $builder->select('title,_content,_date');
5 $query = $builder->get();

```

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

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

```

1 <?php
2
3 $fields = [
4   'id' => [
5     'type'          => 'INT',
6     'constraint'   => 5,
7     'unsigned'      => true,
8     'auto_increment' => true,
9   ],
10  'title' => [
11    'type'          => 'VARCHAR',
12    'constraint'   => '100',
13    'unique'        => true,
14  ],
15  'author' => [
16    'type'          => 'VARCHAR',
17    'constraint'   => 100,
18    'default'       => 'King_of_Town',
19  ],
20  'description' => [
21    'type'          => 'TEXT',
22    'null'          => true,
23  ],
24  'status' => [
25    'type'          => 'ENUM',
26    'constraint'   => ['publish', 'pending', 'draft'],
27    'default'       => 'pending',
28  ],
29 ];
30 $forge->addField($fields);
31 $forge->createTable('table_name');

```

Kode 2.71 akan membangun *database* dengan tabel bernama *table\_name* yang berisikan beberapa *field* seperti *id* dan *title*.

### 2.3.8 Library

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

#### Kelas Email

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

- Beberapa Protokol: *Mail*, *Sendmail*, dan *SMTP*
- Enkripsi *Transport Layer Security*(TLS) dan *Secure Sockets Layer*(SSL) untuk *SMTP*
- Beberapa Penerima
- *Carbon Copy*(CC) dan *Blind Carbon Copy*(BCC)
- HTML atau *email* teks biasa
- Lampiran
- Pembungkus kata
- Prioritas
- Mode BCC *Batch*, memisahkan daftar *email* besar menjadi beberapa BCC kecil.
- Alat *Debugging email*

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

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

```

1 <?php
2
3 $config['protocol'] = 'sendmail';
4 $config['mailPath'] = '/usr/sbin/sendmail';
5 $config['charset'] = 'iso-8859-1';
6 $config['wordWrap'] = true;
7
8 $email->initialize($config);

```

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

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

```

1 <?php
2
3 $email = \Config\Services::email();
4
5 $email->setFrom('your@example.com', 'Your_Name');
6 $email->setTo('someone@example.com');
7 $email->setCC('another@another-example.com');
8 $email->setBCC('them@their-example.com');
9
10 $email->setSubject('Email_Test');
11 $email->setMessage('Testing_the_email_class.');
12
13 $email->send();

```

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

#### Session

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

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

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

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

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

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

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

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

```
1 class Session extends BaseConfig
2 {
3     // ...
4     public string $driver = 'CodeIgniter\Session\Handlers\DatabaseHandler';
5
6     // ...
7     public string $savePath = 'ci_sessions';
8
9     // ...
10 }
```

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**.

### **Working with Uploaded Files**

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

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

Perangkat lunak akan menerima *file* dari *form* yang nantinya akan dilakukan validasi pada *controller*. Kode 2.75 menunjukkan contoh *form* untuk melakukan pengunggahan *file*.

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

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <title>Upload Form</title>
5 </head>
6 <body>
7
8 <?php foreach ($errors as $error): ?>
9     <li><?php esc($error) ?></li>
10 <?php endforeach ?>
11
12 <?= form_open_multipart('upload/upload') ?>
13     <input type="file" name="userfile" size="20">
14     <br><br>
15     <input type="submit" value="upload">
16 </form>
17
18 </body>
19 </html>
```

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

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

```

1 <?php
2
3 namespace App\Controllers;
4
5 use CodeIgniter\Files\File;
6
7 class Upload extends BaseController
8 {
9     protected $helpers = ['form'];
10
11    public function index()
12    {
13        return view('upload_form', ['errors' => []]);
14    }
15
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 }

```

Kode 2.76 terdapat dua buah fungsi sebagai berikut:

- *index()* berfungsi untuk mengembalikan *view* bernama *upload\_form*
- *upload()* berfungsi dalam memberikan aturan ukuran, dimensi, dan jenis *file* untuk melakukan validasi. Setelah validasi berhasil fungsi ini melakukan penyimpanan pada direktori *uploads* dan mengembalikan halaman *view*.

### *Working with URIs*

*CodeIgniter 4* menyediakan solusi berbasis *object oriented* untuk bekerja dengan *URI*. *Library* ini simpel dan memastikan struktur selalu benar tanpa melihat *URI* yang kompleks. Berikut merupakan sintaks untuk membangun sebuah *URI*:

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

Sintaks diatas akan membangun *URI* dengan nama *example.com* dengan beberapa *segment* bernama *some* dan *path*. *URI* yang sudah dibentuk selanjutnya dapat diambil menggunakan sintaks berikut:

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

Sintaks diatas akan mengambil *URI* yang terdapat pada halaman tersebut. Selain dapat mengambil *URI*, *library* ini juga dapat mengambil *segment* yang terdapat pada *URI*. Berikut merupakan sintaks untuk mengambil *segement* yang terdapat pada *URI*.

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

Sintaks diatas akan mengambil *segment* yang tersedia pada *URI* dan menyimpannya dalam sebuah *array* berisikan *value* dari setiap *segment* yang tersedia.

### Validation

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

Kode 2.77: Contoh kode untuk melakukan pengumpulan data.

```

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

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

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

```

1 <?php
2
3 namespace App\Controllers;
4
5 class Form extends BaseController
6 {
7   protected $helpers = ['form'];
8
9   public function index()
10  {
11    if (! $this->request->is('post')) {
12      return view('signup');
13    }
14
15    $rules = [];
16
17    if (! $this->validate($rules)) {
18      return view('signup');
19    }
20
21    // If you want to get the validated data.
22    $validData = $this->validator->getValidated();
23
24    return view('success');
25  }
26}
```

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

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

```

1 $rules = [
2     'username' => 'required|max_length[30]',
3     'password' => 'required|max_length[255]|min_length[10]',
4     'passconf' => 'required|max_length[255]|matches[password]',
5     'email'    => 'required|max_length[254]|valid_email',
6 ];

```

Kode 2.79 melakukan pengecekan terhadap setiap *input form* sesuai dengan aturan yang telah ditetapkan. Aturan-aturan tersebut dapat diganti sesuai dengan kebutuhan dari pengguna. Selain menggunakan aturan yang disediakan *CodeIgniter 4*, pengguna dapat membangun aturannya sendiri pada file *Validation.php*. Kode 2.80 menunjukkan contoh aturan yang dibentuk secara manual. 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*.

```

1 <?php
2
3 class MyRules
4 {
5     public function even($value): bool
6     {
7         return (int) $value % 2 === 0;
8     }
9 }

```

### 2.3.9 *Helpers*

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

```

<?php
helper('helpers_name');

```

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

## 2.4 Konversi *CodeIgniter 3* ke *CodeIgniter 4* [2]

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

### 2.4.1 Struktur Aplikasi

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

### 2.4.2 *Routing*

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

### 2.4.3 Model, View, dan Controller

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

#### *Model*

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

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

#### *View*

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

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

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

menjadi sintaks berikut:

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

3. Pengguna juga perlu mengubah sintaks:

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

menjadi sintaks berikut:

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

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

#### *Controller*

*Controller* pada *CodeIgniter 4* terdapat di `app\Controllers` dan diperlukan beberapa perubahan. Pengguna dapat melakukan pembaharuan *controller* menggunakan cara sebagai berikut:

1. Pengguna perl memindahkan seluruh file *controller* menuju `app/Controllers`.
2. Pengguna perlu menambahkan sintaks namespace `App\Controllers`; setelah pembukaan tag PHP.
3. Pengguna perlu mengubah `extends CI_Controller` menjadi `extends BaseController`.
4. Pengguna juga perlu menghapus apabila terdapat baris `defined('BASEPATH') OR exit('No direct script access allowed');`.

#### 2.4.4 Class Loading

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

#### 2.4.5 Configuration

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

#### 2.4.6 Database

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

1. Memindahkan data-data kredensial menuju *file app\Config\Database.php*
2. Mengubah sintaks untuk memuat *database* diubah menjadi `$db = db_connect();`
3. Penggunaan beberapa *database* diubah menjadi sintaks `$db = db_connect('group_name');`
4. Mengubah seluruh *query* dari `$this->db` menjadi `$db`
5. Mengubah sintaks *query* seperti `$query->result();` menjadi `$query->getResult();`
6. Selain itu, terdapat pembaharuan *Query Builder Class* yang harus diinisiasi menggunakan sintaks `$builder = $db->table('mytable');`. Kelas ini dapat dipakai untuk menjalankan *query* menggunakan sintaks `$builder->get();`

#### 2.4.7 Migrations

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

Kode 2.81: Penambahan kode pada *file migration*.

```
1 namespace App\Database\Migrations;
2 use CodeIgniter\Database\Migration;
```

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

#### 2.4.8 Routing

Pengguna dapat melakukan pembaharuan *routing* dengan cara sebagai berikut:

1. Pengguna dapat memakai *Auto Routing* seperti pada *CodeIgniter 3* dengan menyalakan *Auto Routing(Legacy)*.
2. Terdapat perubahan dari `(:any)` menjadi `(:segment)`.
3. Pengguna juga harus mengubah sintaks pada *app\Config\Routes.php* sebagai berikut:
  - `$route['journals'] = 'blogs';` menjadi `$routes->add('journals', 'Blogs::index');`
  - `$route['product/(:any)'] = 'catalog/product_lookup'` menjadi sintaks `$routes->add('product/(:segment)', 'Catalog::productLookup');`

#### 2.4.9 Libraries

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

##### Emails

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

##### Working with Uploaded Files

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

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

selanjutnya dapat dilakukan validasi dengan cara sebagai berikut:

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

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

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

Sintaks diatas akan mengambil *file* dengan atribut nama `userfile` dan menyimpannya pada direktori `head_img` dengan nama *file* `user_name.jpg`.

##### HTML Tables

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

##### Localization

*CodeIgniter 4* mengembalikan *file* bahasa menjadi *array* sehingga perlu dilakukan beberapa perubahan. Berikut merupakan langkah-langkah untuk melakukan pembaharuan *localization*:

1. Melakukan konfigurasi *default language* pada perangkat lunak.
2. Melakukan pemindahan *file* bahasa pada *CodeIgniter 3* menuju `app\Language\<locale>`.
3. Penghapusan semua *loader* `$this->lang->load($file, $lang);`
4. Mengubah *method* pemanggilan bahasa dari `$this->lang->line('error_email_missing')` menjadi `echo lang('Errors.errorEmailMissing');`

### **Validations**

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

1. Pengguna harus mengubah kode pada *view* dari <?php echo validation\_errors(); ?> menjadi <?= validation\_list\_errors() ?>
2. Pengguna perlu mengubah beberapa sintaks pada *controller* sebagai berikut:
  - \$this->load->helper(array('form', 'url')); menjadi helper(['form', 'url']);
  - Pengguna perlu menghapus sintaks \$this->load->library('form\_validation');
  - if (\$this->form\_validation->run() == FALSE) menjadi if (!\$this->validate([]))
  - \$this->load->view('myform');  
menjadi sintaks return view('myform', ['validation' => \$this->validator,]);
3. Pengguna juga perlu mengubah kode seperti yang ditunjukan pada Kode 2.82 untuk melakukan validasi *form*.

Kode 2.82: Perubahan kode untuk melakukan validasi.

```

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

### **2.4.10 Helpers**

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

### **2.4.11 Framework**

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

## BAB 3

# ANALISIS

### 3.1 Analisis Sistem Kini

Seperti dibahas pada bab 2.2, *SharIF Judge* merupakan sebuah *online judge* yang dikustomisasi sesuai dengan kebutuhan Informatika UNPAR. *SharIF Judge* dibentuk menggunakan *framework CodeIgniter 3* yang menerapkan arsitektur *Model-View-Controller* atau MVC. Arsitektur ini memisahkan pemrosesan data pada *Model*, memisahkan logika pada *Controller*, dan memisahkan tampilan pada *View*. Selain itu, terdapat direktori **assets** yang berisi seluruh kebutuhan pengguna untuk ditampilkan seperti *javascript* dan gambar. Terakhir terdapat direktori **config** yang berisi konfigurasi aplikasi dan *library* yang dibentuk dan digunakan oleh *SharIF Judge*. Analisis dimulai dengan melakukan eksplorasi setiap fungsi yang terdapat pada MVC aplikasi *SharIF Judge*. Berikut merupakan hasil temuan dari eksplorasi yang telah dilakukan:

#### 3.1.1 *Model*

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

##### **Assignment\_model.php**

**Assignment\_model.php** terdapat beberapa fungsi untuk memproses data pada tabel *assignment*. Berikut merupakan fungsi-fungsi dari **Assignment\_model.php**:

- **add\_assignment**  
Fungsi ini berguna untuk menambahkan atau memperbarui *assignment* pada *database*.
- **delete\_assignment**  
Fungsi ini berguna untuk menghapus *assignment* pada *database*.
- **all\_assignments**  
Fungsi ini berguna untuk mengembalikan seluruh *assignment* beserta informasi setiap *assignment* tersebut.
- **new\_assignment\_id**  
Fungsi ini berguna untuk mencari *id* terkecil yang dapat digunakan untuk menambahkan *assignment* baru.
- **all\_problems**  
Fungsi ini berguna untuk mengembalikan seluruh *problems* dari *assignment* yang ada.
- **problem\_info**  
Fungsi ini berguna untuk mengembalikan baris tabel untuk *problem* yang diberikan.
- **assignment\_info**  
Fungsi ini berguna untuk mengembalikan baris tabel untuk *assignment* yang diberikan.
- **is\_participant**  
Fungsi ini berguna untuk mengecek apakah *username* merupakan peserta *assignments* atau tidak.

- **increase\_total\_submits**  
Fungsi ini berguna untuk menambahkan satu buah total *submit*.
- **set\_moss\_time**  
Fungsi ini berguna untuk memperbarui "Moss Update Time" untuk *assignment* tertentu.
- **get\_moss\_time**  
Fungsi ini berguna untuk mengembalikan "Moss Update Time" untuk *assignment* tertentu.
- **save\_problem\_description**  
Fungsi ini berguna untuk menambahkan atau memperbarui deskripsi *problem*.
- **\_update\_coefficients**  
Fungsi ini dipanggil pada fungsi **add\_assignment** yang berguna untuk memperbarui koefisien dari *assignment* tertentu.

#### Hof\_model.php

Berikut merupakan fungsi-fungsi dari **Hof\_model.php** yang berguna untuk mengambil data untuk ditampilkan pada halaman *Hall of Fame*.

- **get\_all\_final\_submission**  
Fungsi ini berguna untuk mengambil seluruh data *final submission*.
- **get\_all\_user\_assignments**  
Fungsi ini berguna untuk mengambil detail *assignment* dan *problem* berdasarkan pengguna.

#### Logs\_model.php

Berikut merupakan fungsi-fungsi dari **Logs\_model.php** yang berguna untuk mencatat *log* pada beberapa tabel:

- **insert\_to\_logs**  
Fungsi ini berguna untuk mencatat *log* pada tabel *login*.
- **get\_all\_logs**  
Fungsi ini berguna untuk mengembalikan seluruh *log* dari tabel *login* dalam bentuk *array*.

#### Notifications\_model.php

**Notifications\_model.php** terdapat beberapa fungsi untuk memproses data pada tabel *notifications*. Berikut merupakan fungsi-fungsi dari *model* tersebut:

- **get\_all\_notifications**  
Fungsi ini berguna untuk mengembalikan seluruh *notifications* dalam bentuk *array*.
- **get\_latest\_notifications**  
Fungsi ini berguna untuk mengembalikan sepuluh *notification* terbaru.
- **add\_notification**  
Fungsi ini berguna untuk menambahkan data *notification* baru.
- **update\_notification**  
Fungsi ini berguna untuk memperbarui *notification* tertentu.
- **delete\_notification**  
Fungsi ini berguna untuk menghapus *notification* tertentu.
- **get\_notification**  
Fungsi ini berguna untuk mengembalikan *notification* tertentu dalam bentuk *array*.
- **have\_new\_notification**  
Fungsi ini berguna untuk mengecek apakah terdapat *notification* setelah waktu tertentu.

#### Queue\_model.php

Berikut merupakan fungsi-fungsi dari **Queue\_model.php** yang berguna untuk memproses data pada halaman *queue*.

- **in\_queue**

Fungsi ini berguna untuk mengecek apakah *submission* pengguna tertentu sudah dalam antrean *submission*.

- **get\_queue**

Fungsi ini berguna untuk mengembalikan data seluruh antrian *submission*.

- **empty\_queue**

Fungsi ini berguna untuk menghapus seluruh antrean *submission*.

- **add\_to\_queue**

Fungsi ini berguna untuk memasukan *submission* kedalam tabel *queue*.

- **rejudge**

Fungsi ini berguna untuk menambahkan *submission* kedalam antrean untuk dilakukan *rejudge*.

- **rejudge\_single**

Fungsi ini berguna untuk menambahkan satu buah *submission* kedalam antrean untuk dilakukan *rejudge*.

- **get\_first\_item**

Fungsi ini berguna untuk mengambil data *submission* pertama dalam antrean.

- **remove\_item**

Fungsi ini berguna untuk menghapus data tertentu dalam antrean.

- **save\_judge\_result\_in\_db**

Fungsi ini berguna untuk menyimpan hasil dari *judge* ke dalam *database*.

- **add\_to\_queue\_exec**

Fungsi ini berguna untuk menambahkan data *dummy* pada antrean.

### **Scoreboard\_model.php**

Berikut merupakan fungsi-fungsi dari **Scoreboard\_model.php** yang berguna untuk memproses data untuk ditampilkan pada halaman *Score Board*.

- **\_generate\_scoreboard**

Fungsi ini dipanggil pada fungsi **update\_scoreboard** dan berfungsi untuk menghasilkan *scoreboard* dari *final submission*.

- **update\_scoreboards**

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

- **update\_scoreboard**

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

- **get\_scoreboard**

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

### **Settings\_model.php**

Berikut merupakan fungsi-fungsi dari **Settings\_model.php** yang berguna untuk memproses data untuk ditampilkan pada tabel *settings*.

- **get\_setting**

Fungsi ini berguna untuk mengembalikan data *setting* tertentu.

- **set\_setting**

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

- **get\_all\_settings**

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

- **set\_settings**  
Fungsi memperbarui beberapa *setting*.

#### **Submit\_model.php**

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

- **get\_submission**  
Fungsi ini berguna untuk mengembalikan baris data *submission* tertentu.
- **get\_final\_submissions**  
Fungsi ini berguna untuk mengembalikan seluruh *final submission*.
- **get\_all\_submissions**  
Fungsi ini berguna untuk mengembalikan seluruh *submission*.
- **count\_final\_submissions**  
Fungsi ini berguna untuk menghitung seluruh *final submission*.
- **count\_all\_submissions**  
Fungsi ini berguna untuk menghitung seluruh *submission*.
- **set\_final\_submission**  
Fungsi ini berguna untuk memilih *submission* tertentu menjadi *final submission*.
- **add\_upload\_only**  
Fungsi ini berguna untuk menambahkan hasil dari *upload only problem* ke dalam *database*.

#### **User\_model.php**

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

- **have\_user**  
Fungsi ini berguna untuk mengecek apakah terdapat pengguna dengan *username* tertentu.
- **user\_id\_to\_username**  
Fungsi ini berguna untuk mengubah *user id* menjadi *username*.
- **username\_to\_user\_id**  
Fungsi ini berguna untuk mengubah *username* menjadi *user id*.
- **have\_email**  
Fungsi ini berguna untuk mengecek apakah terdapat *username* dengan *email* tertentu.
- **add\_user**  
Fungsi ini berguna untuk menambahkan sebuah pengguna.
- **add\_users**  
Fungsi ini berguna untuk menambahkan banyak pengguna.
- **delete\_user**  
Fungsi ini berguna untuk menghapus pengguna tertentu.
- **delete\_submissions**  
Fungsi ini berguna untuk menghapus seluruh *submission* pada pengguna tertentu.
- **validate\_user**  
Fungsi ini berguna untuk mengecek *username* dan *password* apakah sesuai dengan data yang dimasukan.
- **selected\_assignment**  
Fungsi ini berguna untuk mengembalikan *selected assignment* untuk pengguna tertentu.
- **get\_names**  
Fungsi ini berguna untuk mengembalikan nama dari *username* pengguna tertentu.
- **update\_profile**  
Fungsi ini berguna untuk memperbarui profil dari pengguna seperti *nama*, *email*, *password*, dan *role*.

- **send\_password\_reset\_mail**  
Fungsi ini berguna untuk menghasilkan *password reset key* dan mengirim *email* untuk melakukan *reset password*.
- **passchange\_is\_valid**  
Fungsi ini berguna untuk mengecek apakah *password reset key* yang diberikan sesuai atau tidak.
- **reset\_password**  
Fungsi ini berguna untuk mengatur ulang *password* sesuai dengan *password key* tertentu.
- **get\_all\_users**  
Fungsi ini berguna untuk mengembalikan seluruh pengguna untuk halaman *users*.
- **get\_user**  
Fungsi ini berguna untuk mengembalikan baris data untuk pengguna tertentu.
- **update\_login\_time**  
Fungsi ini berguna untuk memperbaharui *login time* dan *last login time* untuk pengguna tertentu.

### User.php

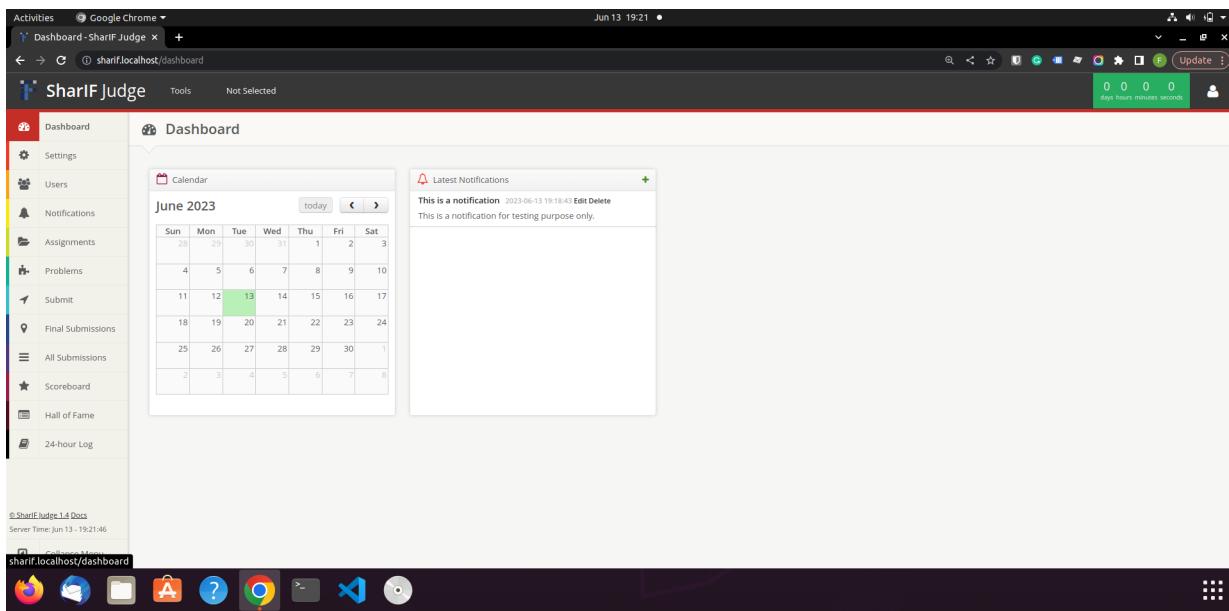
Berikut merupakan fungsi-fungsi dari *User.php* yang berguna untuk memproses data pada tabel *users*.

- **select\_assignment**  
Fungsi ini berguna untuk mengatur *assignment* yang dipilih oleh pengguna.
- **save\_widget\_positions**  
Fungsi ini berguna untuk memperbaharui posisi dari *dashboard widget* pada *database*.
- **get\_widget\_positions**  
Fungsi ini berguna untuk mengembalikan data *dashboard widget*.

### 3.1.2 View

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

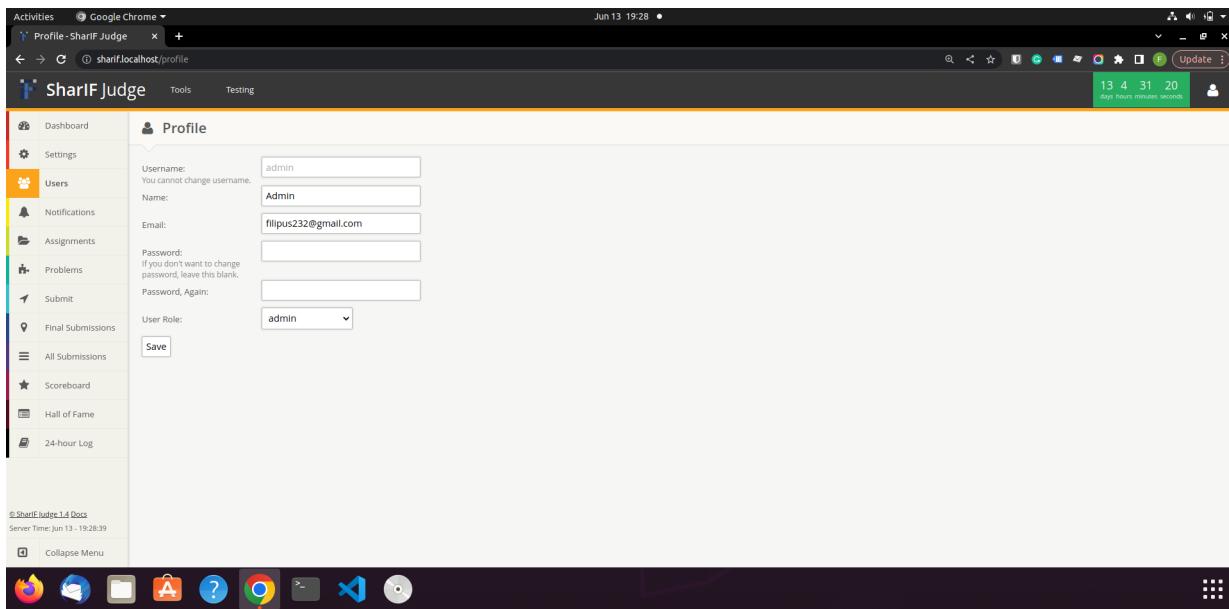
## Dashboard



Gambar 3.1: Tampilan Halaman Dashboard

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

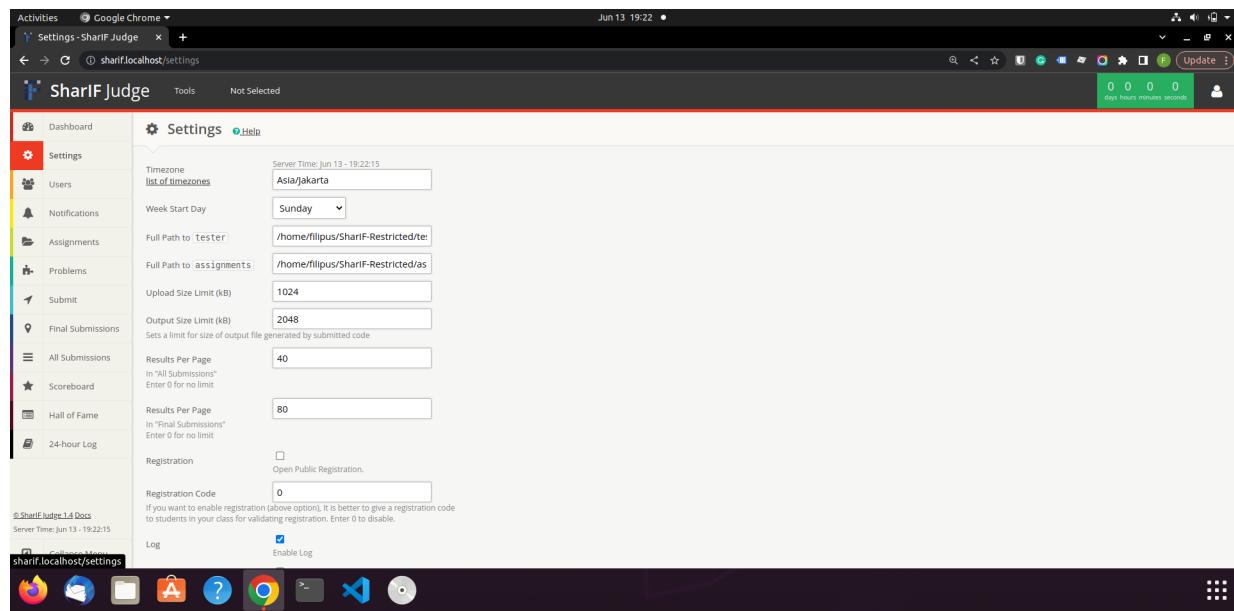
## Profile



Gambar 3.2: Tampilan Halaman Profile

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

### Settings



Gambar 3.3: Tampilan Halaman Settings

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

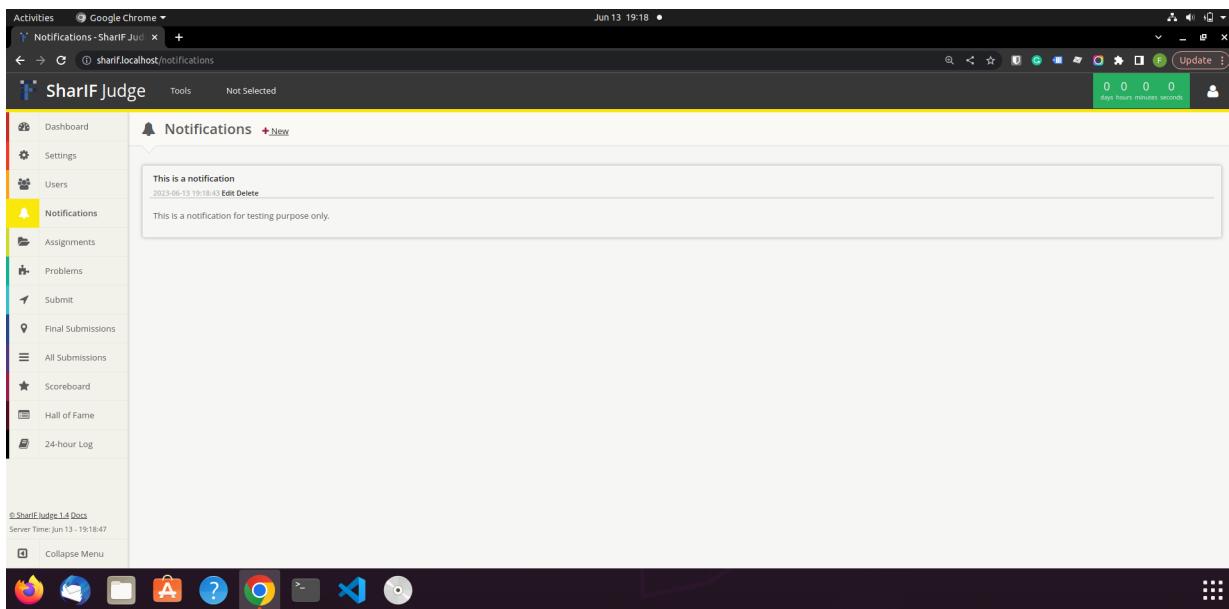
### Users

#	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	2023-06-13 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	119074	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

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

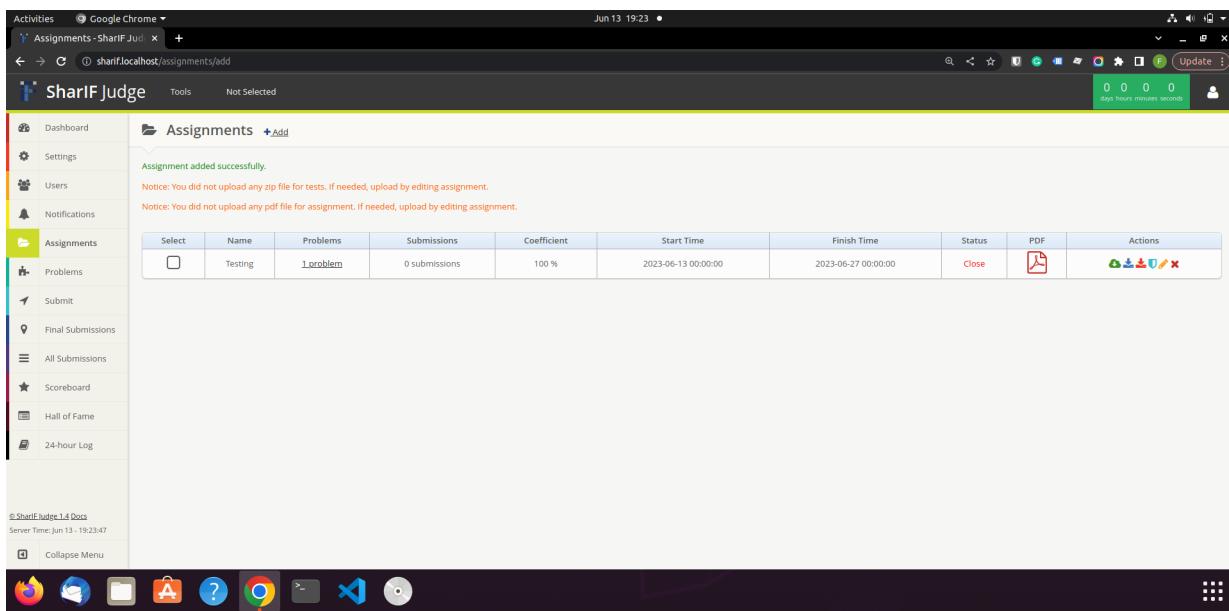
## Notifications



Gambar 3.5: Tampilan Halaman Notifications

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

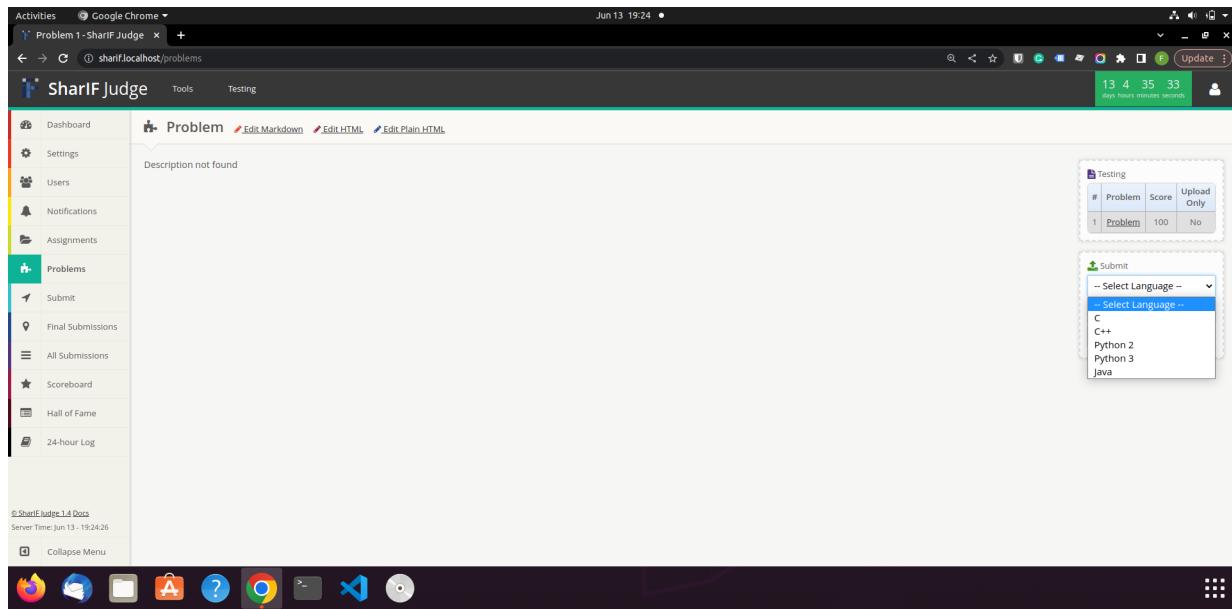
## Assignments



Gambar 3.6: Tampilan Halaman Assignments

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

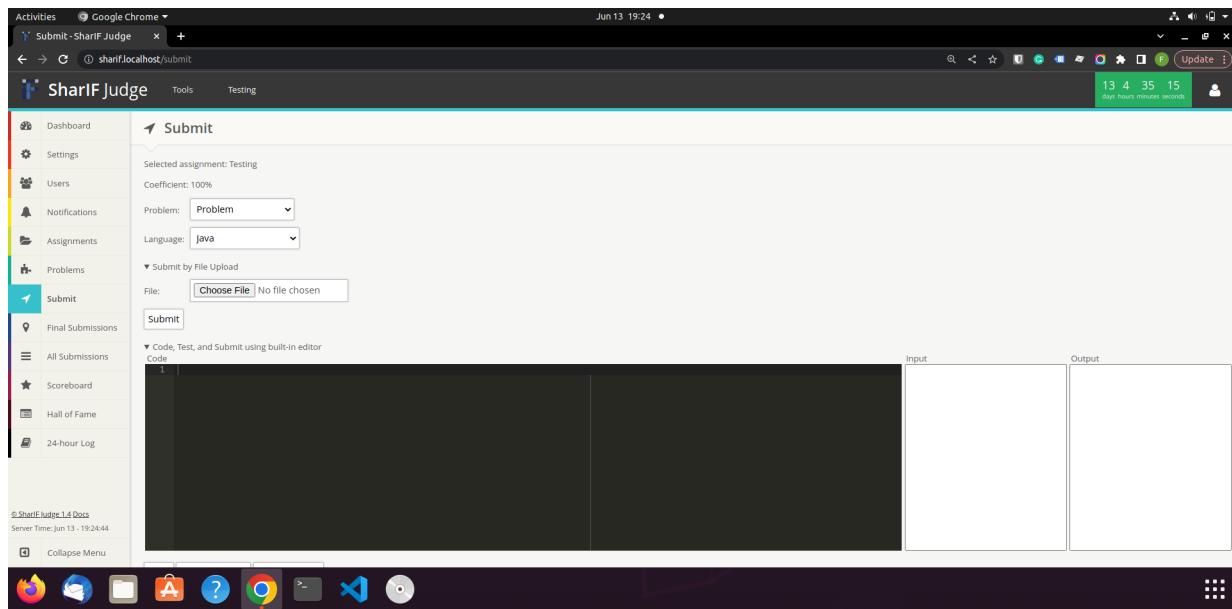
### Problems



Gambar 3.7: Tampilan Halaman Problems

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

### Submit



Gambar 3.8: Tampilan Halaman Submit

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

### Final Submissions

#	ID	Username	Name	Problem	Submit Time	Score				Language	Status	Code	Log	Actions
						Score	Delay %	Final Score	Language					
1	1	admin	Admin	1	2023-06-13 19:26:24	0	No Delay 100%	0	java	Compilation Error	<span style="background-color: blue; color: white; padding: 2px;">Code</span>	<span style="background-color: grey; color: white; padding: 2px;">Log</span>	<span style="color: red; font-size: small;">↻</span>	

Gambar 3.9: Tampilan Halaman Final Submission

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

### All Submissions

Final	ID	Username	Name	Problem	Submit Time	Score				Language	Status	Code	Log	Actions
						Score	Delay %	Final Score	Language					
✓	2	admin	Admin	1	2023-06-13 19:28:03	0	No Delay 100%	0	Py 3	Syntax Error	<span style="background-color: orange; color: white; padding: 2px;">Code</span>	<span style="background-color: grey; color: white; padding: 2px;">Log</span>	<span style="color: red; font-size: small;">↻</span>	
✗	1	admin	Admin	1	2023-06-13 19:26:24	0	No Delay 100%	0	java	Compilation Error	<span style="background-color: orange; color: white; padding: 2px;">Code</span>	<span style="background-color: grey; color: white; padding: 2px;">Log</span>	<span style="color: red; font-size: small;">↻</span>	

Gambar 3.10: Tampilan Halaman All Submission

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

### Scoreboard

The screenshot shows the Sharf Judge application interface. The left sidebar has a 'Scoreboard' section selected. The main content area is titled 'Scoreboard of Testing' and displays a table with one row. The table columns are '#', 'Username', 'Name', 'Problem', and 'Total'. The data row is: #1, Username: admin, Name: Admin, Problem: 100, Total: 100. At the top right, there is a timer showing '13 4 32 45 days hours minutes seconds' with an 'Update' button. The bottom status bar shows the server time as 'Jun 13 19:27:14'.

#	Username	Name	Problem	Total
1	admin	Admin	100	100

Gambar 3.11: Tampilan Halaman Scoreboard

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

### Hall of Fame

The screenshot shows the Sharf Judge application interface. The left sidebar has a 'Hall of Fame' section selected. The main content area is titled 'Hall of Fame' and displays a table with one row. The table columns are 'Rank', 'Username', 'Display Name', and 'Total Score'. The data row is: Rank 1, Username: admin, Display Name: Admin, Total Score: 0. At the top right, there is a timer showing '13 4 31 43 days hours minutes seconds' with an 'Update' button. The bottom status bar shows the server time as 'Jun 13 19:28:16'.

Rank	Username	Display Name	Total Score
1	admin	Admin	0

Gambar 3.12: Tampilan Halaman Hall of Fame

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

## 24-hour Log

Gambar 3.13: Tampilan Halaman 24-hour Log

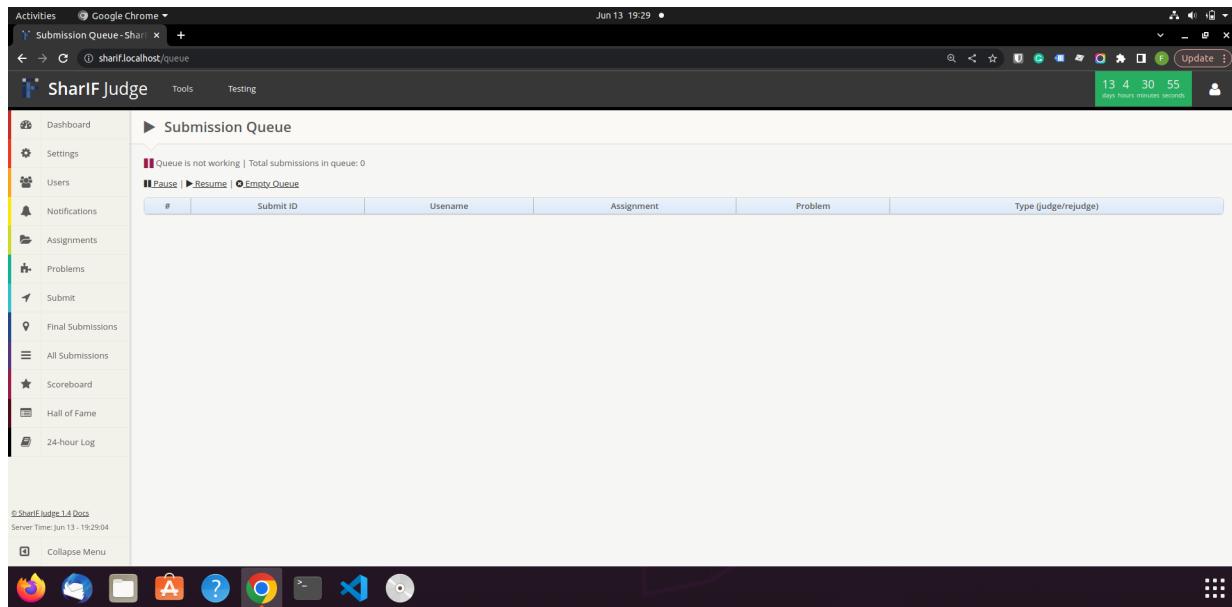
Gambar 3.13 menunjukkan tampilan halaman *24-hour Log* yang terdapat hanya pada *role admin* dan *head instructor*.

## Rejudge

Gambar 3.14: Tampilan Halaman ReJudge

Gambar 3.14 menunjukkan tampilan halaman *ReJudge* yang terdapat hanya pada *role admin* dan *head instructor*.

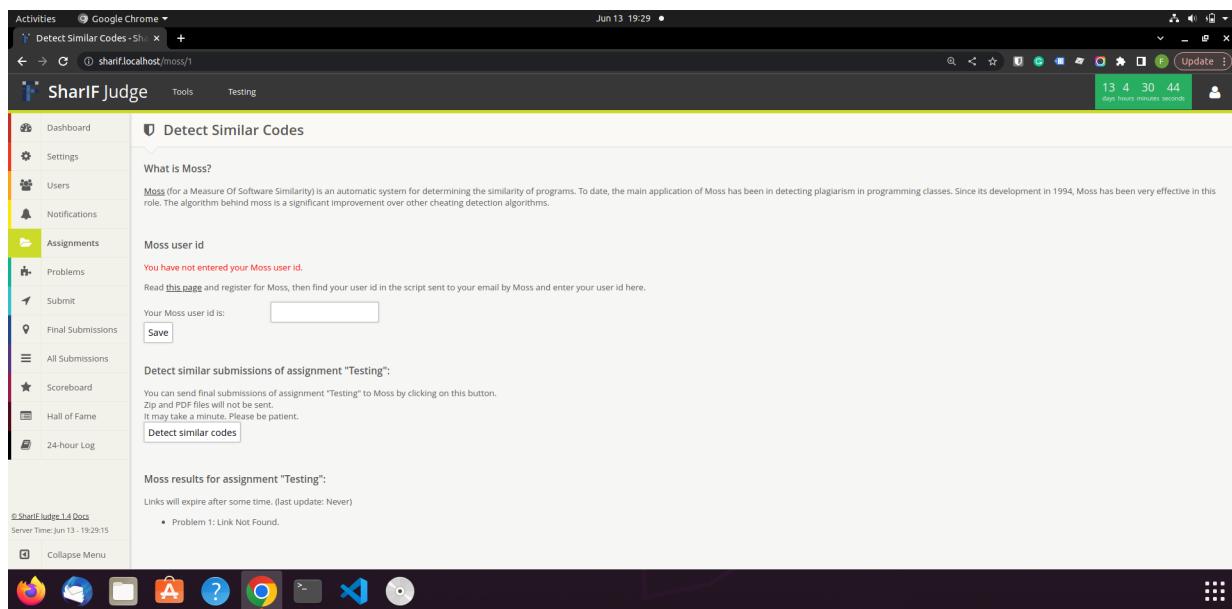
### Submission Queue



Gambar 3.15: Tampilan Halaman Submission Queue

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

### Cheat Detection



Gambar 3.16: Tampilan Halaman Cheat Detection

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

### 3.1.3 Controller

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

#### `Assignments.php`

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

- `index`  
Fungsi ini berguna untuk mengambil dan memberikan data menuju halaman `assignments.twig` menggunakan `Assignment_model.php`.
- `select`  
Fungsi ini berguna untuk memilih `assignment` melalui `top bar` menggunakan AJAX.
- `pdf`  
Fungsi ini berguna untuk mengunduh `assignment` atau `problem` dalam bentuk pdf.
- `downloadtestsdesc`  
Fungsi ini berguna untuk mengompres dan mengunduh data `test` dan deskripsi sebuah `assignment`.
- `download_submissions`  
Fungsi ini berguna untuk mengompres dan mengunduh kode terakhir sebuah `assignment` pengguna.
- `delete`  
Fungsi ini berguna untuk menghapus `assignment`.
- `add`  
Fungsi ini berguna untuk mengambil data masukan pengguna untuk menambahkan atau mengubah `assignment`.
- `_add`  
Fungsi ini berguna untuk menambah atau mengubah `assignment`.
- `edit`  
Fungsi ini berguna untuk mengecek `role` pengguna apakah dapat mengubah `assignment`. Selanjutnya pengguna dikembalikan menuju fungsi `add`.
- `pdfCheck` Fungsi ini berguna untuk mengecek `file PDF` dari sebuah `assignment`.

#### `Dashboard.php`

Berikut merupakan fungsi-fungsi pada `controller Dashboard.php`:

- `index`  
Fungsi ini berguna untuk mengambil dan memberikan data menuju halaman `dashboard` menggunakan tiga buah `model`. `Model` tersebut terdiri dari `Assignment_model`, `Settings_model`, dan `Notifications_model`.
- `widget_positions`  
Fungsi ini berguna untuk menyimpan data `widget` pengguna menggunakan AJAX.

#### `Install.php`

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

### >Login.php

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

- **\_registration\_code**

Fungsi ini beguna untuk memeriksa kode registrasi.

- **index**

Fungsi ini berguna untuk melakukan validasi *username* dan *password* pengguna pada halaman *login*. Selain itu, fungsi ini juga memperbarui *log* pada tabel *login*.

- **register**

Fungsi ini berguna untuk melakukan validasi dalam pembentukan akun pada halaman *register*.

- **logout**

Fungsi ini berguna untuk menghancurkan *session* dari pengguna dan memindahkan pengguna menuju halaman *login*.

- **lost**

Fungsi ini berguna untuk mengirim *email* lupa password.

- **reset**

Fungsi ini berguna untuk melakukan *reset password* pengguna.

### Logs.php

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

### Moss.php

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

- **index**

Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *moss*. Fungsi ini juga dapat melakukan deteksi kode menuju Moss berdasarkan permintaan.

- **update**

Fungsi ini berguna untuk memperbarui *moss\_userid* yang dimasukan oleh pengguna.

- **\_detec**

Fungsi ini berguna untuk mengirimkan data untuk dilakukan pengecekan terhadap *final submission* melalui *shell*.

### Notification.php

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

- **index**

Fungsi ini berguna untuk mengambil dan mengembalikan data pada halaman *notifications* menggunakan *assignment\_model* dan *notifications\_model*.

- **add**

Fungsi ini berguna untuk menambahkan data *notifications* baru.

- **edit**

Fungsi ini berguna untuk memperbarui data *notifications*.

- **delete**

Fungsi ini berguna untuk menghapus data *notifications* tertentu.

- **check**

Fungsi ini berguna memeriksa *notifications* baru.

### Problems.php

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

- **index**  
Fungsi ini berguna untuk mengambil dan memberikan data *problems* sesuai dengan *assignment* tertentu pada halaman *problems*.
- **edit**  
Fungsi ini berguna untuk memperbaharui deskripsi *problems* pada *assignment* tertentu.

**Profile.php**

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

- **index**  
Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *profile*. Selain itu, fungsi ini berguna untuk melakukan pembaharuan data *profile*.
- **\_password\_check**  
Fungsi ini berguna untuk melakukan validasi terhadap *password* yang akan dimasukkan pengguna sesuai dengan aturan.
- **\_password\_again\_check**  
Fungsi ini berguna untuk melakukan validasi terhadap pengulangan *password* yang dimasukkan pengguna.
- **\_email\_check**  
Fungsi ini berguna untuk melakukan validasi terhadap *email* yang dimasukkan pengguna
- **\_role\_check**  
Fungsi ini berguna untuk melakukan validasi *role* pengguna.

**Queue.php**

- **index**  
Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *queue* menggunakan tiga buah *model*. *Model* tersebut adalah *Assignment\_model*, *queue\_model*, dan *settings\_model*.
- **pause**  
Fungsi ini berguna untuk memberhentikan proses *queue*.
- **resume**  
Fungsi ini berguna untuk melanjutkan proses *queue*.
- **empty\_queue**  
Fungsi ini berguna untuk menghapus data tabel *queue*.

**Queueprocess.php**

*Controller Queueprocess.php* hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna untuk menjalankan proses *judge* berdasarkan *queue* satu demi satu sesuai antrean. Fungsi ini menggunakan beberapa *model* yaitu *Queue\_model*, *Submit\_model*, *Assignments\_model*, dan *Settings\_model*.

**Rejudge.php**

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

- **index**  
Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *rejudge* menggunakan *Assignment\_model*. Fungsi ini juga berguna untuk melakukan *rejudge* kepada sebuah *problem* tertentu.
- **rejudge\_single**  
Fungsi ini berguna untuk melakukan *rejudge* pada satu buah masalah tertentu.

### Scoreboard.php

Berikut merupakan fungsi-fungsi pada controller `Scoreboard.php`:

- `index`

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

### Server\_time.php

*Controller Server\_time.php* hanya memiliki satu buah fungsi bernama `index`. Fungsi ini berguna untuk mengeluarkan *server\_time*.

### Settings.php

Berikut merupakan fungsi-fungsi pada controller `Settings.php`:

- `index`

Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *settings* menggunakan `Settings_model` dan `Assignment_model`.

- `update`

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

### Submission.php

Berikut merupakan fungsi-fungsi pada controller `Submission.php`:

- `_download_excel`

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

- `final_excel`

Fungsi ini berguna untuk mengunduh data *final submissions*.

- `all_excel`

Fungsi ini berguna untuk mengunduh data seluruh *submissions*.

- `the_final`

Fungsi ini berguna untuk memberikan data pada halaman *Final Submissions* menggunakan beberapa *model*. *Model* tersebut terdiri dari `Submit_model`, `Settings_model`, dan `User_model`.

- `all`

Fungsi ini berguna untuk mengembalikan data pada halaman *All Submissions* menggunakan beberapa *model*. *Model* tersebut terdiri dari `Submit_model`, `Settings_model`, dan `User_model`.

- `select`

Fungsi ini berguna untuk memilih *submission* yang akan dijadikan *final submission* oleh pengguna.

- `_check_type`

Fungsi ini berguna untuk melakukan pengecekan tipe *submission* yang telah dikumpulkan oleh pengguna.

- `view_code`

Fungsi ini berguna untuk memperlihatkan *submission* yang telah dikumpulkan oleh pengguna sesuai dengan tipenya.

- `download_file`

Fungsi ini berguna untuk mengunduh hasil dari *submission* yang telah dikumpulkan oleh pengguna.

**Submit.php**

Berikut merupakan fungsi-fungsi pada *controller Submit.php*:

- **\_language\_to\_type**  
Fungsi ini berguna untuk mengubah bahasa pemrograman menjadi tipe sesuai dengan pilihan pengguna.
- **\_language\_to\_ext**  
Fungsi ini berguna untuk mengubah bahasa pemrograman menjadi ekstensi sesuai dengan pilihan pengguna.
- **\_match**  
Fungsi ini berguna untuk mencocokkan tipe dengan ekstensi dari bahasa pemrogramannya.
- **\_check\_language**  
Fungsi ini berguna untuk melakukan pengecekan terhadap bahasa pemrograman yang digunakan.
- **index**  
Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *submit* menggunakan *Assignment\_model*.
- **\_upload**  
Fungsi ini berguna untuk menyimpan jawaban dan memasukannya ke *queue* untuk dinilai.
- **load**  
Fungsi ini berguna untuk memuat kode dari *editor file*.
- **save**  
Fungsi ini berguna untuk meyimpan kode menuju *editor file* dan mengirim ataupun menjalankannya.
- **\_submit**  
Fungsi ini berguna untuk menambahkan kode pada *queue* untuk dilakukan *judge*.
- **\_execute**  
Fungsi ini berguna untuk menambahkan kode untuk dijalankan atau di *queue*.
- **get\_output** Fungsi ini berguna untuk memuat *file* menjadi hasil eksekusi.

**User.php**

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

- **index**  
Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *users*.
- **add**  
Fungsi ini berguna untuk menambahkan pengguna baru sesuai dengan masukan.
- **delete**  
Fungsi ini berguna untuk menghapus pengguna yang dipilih.
- **delete\_submissions**  
Fungsi ini berguna untuk menghapus *submission* dari sebuah pengguna.
- **list\_excel**  
Fungsi ini berguna untuk menghasilkan dan mengunduh data pengguna pada format *excel*.

**3.1.4 Assets**

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

- **Direktori ace**  
Direktori ini berisikan *javascript* yang berfungsi untuk menambahkan *code editor* pada aplikasi.
- **Direktori font**  
Direktori ini berisikan seluruh *font* yang digunakan oleh aplikasi *SharIF Judge*.

- Direktori *fullcalendar*  
Direktori ini berisikan *javascript* dan *css* yang berfungsi untuk menambahkan kalender pada aplikasi.
- Direktori *gridster*  
Direktori ini berisikan *javascript*, *css*, dan gambar yang berfungsi untuk membangun kolom *grid* dengan sifat *drag and drop*.
- Direktori *images*  
Direktori ini berisikan seluruh gambar yang digunakan pada aplikasi *SharIF Judge* seperti banner dan juga logo.
- Direktori *js*  
Direktori ini berisikan *javascript* dan *jquery*. *Javascript* dan *jquery* yang terdapat pada direktori ini dibentuk secara manual dan diambil dari situs resmi.
- Direktori *nano\_scroller*  
Direktori ini berisikan *jquery* dan *css* yang berfungsi untuk membangun *scrollbar* pada aplikasi.
- Direktori *noty*  
Direktori ini berisikan *javascript* yang berfungsi untuk mempermudah membangun pesan *alert*, *success*, dan sebagainya.
- Direktori *pdfjs*  
Direktori ini berisikan *javascript* yang berfungsi untuk melakukan *parsing* dan *rendering file pdf*.
- Direktori *reveal*  
Direktori ini berisikan *jquery* dan *css* yang berfungsi untuk menampilkan *popup* berupa halaman yang dikonfigurasikan.
- Direktori *snippet*  
Direktori ini berisikan *javascript* dan *css* yang berfungsi sebagai *template* dalam membangun kode *javascript*.
- Direktori *styles*  
Direktori ini berisikan *css* yang berfungsi untuk memperindah aplikasi yang dibentuk.
- Direktori *tinymce*  
Direktori ini berisikan *javascript* dan *css* yang berfungsi untuk membangun *WYSIWYG editor* pada sebuah aplikasi.

### 3.1.5 *Config*

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

- **autoload.php**  
*File* ini berisikan seluruh data yang ingin diinisiasikan saat aplikasi dijalankan seperti *libraries*, *helpers*, dan *model*.
- **config.php**  
*File* ini berisikan konfigurasi aplikasi *SharIF Judge* seperti *base url* aplikasi, *encryption key*, konfigurasi *session*, konfigurasi *cookie*, dan *csrf*.
- **constants.php**  
*File* ini berisikan data yang bersifat *global* pada aplikasi.
- **database.php**  
*File* ini berisikan konfigurasi *database* yang digunakan oleh pengguna.
- **doctypes.php**  
*File* ini berisikan deklarasi tipe dokumen untuk *HTML*.
- **foreign\_chars.php**  
*File* ini terjemahan data *foreign character* yang terdapat pada *text helper*.
- **memcached.php**  
*File* ini berisikan konfigurasi *server* untuk menyimpan data *cache* pada server.

- **migration.php**  
File ini berisikan konfigurasi untuk melakukan *migration* seperti tipe *migration* dan *path* file *migration* disimpan.
- **mime.php**  
File ini berisikan tipe-tipe *mime* yang digunakan oleh *upload helper*.
- **profiler.php**  
File ini berisikan konfigurasi untuk *profiler*.
- **routes.php**  
File ini berisikan konfigurasi *route* dan *route* yang didefinisikan secara manual.
- **secrets.php**  
File ini dibentuk oleh pengguna yang berisikan data-data untuk melakukan autentikasi dan konfigurasi *email*.
- **smiley.php**  
File ini berisikan terjemahan dari sintaks *emoticon* menjadi gambar yang digunakan pada *emoticon helper*.
- **twig.php**  
File ini berisikan konfigurasi *template engine twig*.
- **user\_agents.php**  
File ini berisikan tipe dari *platform*, *browser*, dan lainnya.

### 3.1.6 Libraries

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

#### Unzip

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

Kode 3.1: Contoh kode penggunaan *Library Unzip*

```

1 $this->load->library('unzip');
2 $this->unzip->allow(array('txt', 'cpp', 'html', 'md', 'pdf'));
3 $extract_result = $this->unzip->extract($u_data['full_path'], $tmp_dir);

```

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

#### Twig

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

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

```

1 {%- set title = 'Login' %}
```

```

2  {% include 'templates/simple_header.twig' %}
3
4  {{ form_open() }}
5      <div class="box login">
6
7          <div class="judge_logo">
8              <a href="{{ site_url() }}>
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', '

', '

') }}
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', '

', '

') }}
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                         <a href="{{ site_url('login/lost') }}>Reset Password</a>
35                         <input type="submit" value="Login" id="sharif_submit"/>
36                     </p>
37                 </div>
38             </div>
39         </form>
40     </body>
41 </html>

```

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

### ***Password\_hash***

*Password\_hash* merupakan sebuah *library* yang dibentuk oleh *phpass*. *Library* ini berfungsi untuk melakukan enkripsi *password* dan melakukan verifikasi *password*. *Library* ini mendukung beberapa metode enkripsi antara lain *CRYPT\_BLOWFISH* dan *CRYPT\_EXT\_DES*. Kode 3.3 menunjukan contoh penggunaan *library* ini pada *SharIF Judge*.

Kode 3.3: Contoh kode penggunaan *Library Password\_hash*

```

1 $this->load->library('password_hash', array(8, FALSE));
2 $user['password'] = $this->password_hash->HashPassword($this->input->post('password'));

```

Kode 3.3 melakukan *load* pada *library Password\_hash*. Selanjutnya sintaks akan melakukan *hashing* pada *input* menggunakan algoritma yang ditentukan dan menyimpannya pada sebuah variabel.

### ***MY\_Form\_validation***

*MY\_Form\_validation* merupakan *library* yang dibentuk secara manual untuk menambahkan fungsi validasi yang sudah tersedia pada *CodeIgniter 3*. *Library* ini memiliki dua buah fungsi yakni:

- **required**

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

- **lowercase**

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

### ***MY\_Profiler***

*MY\_Profiler* merupakan *library* yang dibentuk secara manual dan merupakan perpanjangan dari *CI\_Profiler*. *Library* ini berfungsi untuk mengembalikan data yang telah dijalankan oleh *profiler* dan menyimpannya pada halaman pengguna. *Library* ini tidak akan digunakan lagi karena tidak terdapat pada *CodeIgniter 4* dan tidak dipakai pada *SharIF Judge*.

### ***Parsedown***

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

Kode 3.4: Contoh kode penggunaan *Library Parsedown*

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

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

Ini adalah list :

1. Satu
2. Dua
3. Tiga

Kode 3.5: Contoh hasil teks sesudah dilakukan *parsedown*

```
1 <p>Ini adalah list</p>
2 <ol>
3 <li>Satu</li>
4 <li>Dua</li>
5 <li>Tiga</li>
```

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

### ***Phpxcel***

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

### ***Shj\_pagination***

*Shj\_pagination* merupakan *library* yang dibentuk secara manual dengan fungsi untuk membatasi maksimal data pada setiap halaman sesuai dengan konfigurasinya.

### ***Upload***

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

Kode 3.6: Contoh penggunaan *library upload*

```
1 $this->load->library('upload');
2 $config = array(
3     'upload_path' => $assignments_root,
```

```

4     'allowed_types' => 'zip',
5 );
6 $this->upload->initialize($config);
7 $zip_uploaded = $this->upload->do_upload('tests_desc');
8 $u_data = $this->upload->data();

```

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

### ***Input***

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

Kode 3.7: Contoh penggunaan *library input*

```

1 $this->load->library('upload');
2 $names = $this->input->post('name');

```

### ***Email***

*Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk mengirimkan *email* kepada orang yang dituju sesuai dengan konfigurasinya.

### ***Form validation***

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

### ***Kelas URI***

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

Kode 3.8: Contoh penggunaan *library URI*

```

1 $input = $this->uri->uri_to_assoc();

```

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

### ***Session***

*Library* ini berfungsi untuk menyimpan informasi setiap pengguna memakai perangkat lunak. Data-data ini selanjutnya disimpan menuju *database* dengan nama tabel *shj\_session*. Data-data yang telah disimpan selanjutnya dapat diambil kembali untuk melakukan pengecekan ataupun disimpan menuju tabel lainnya. Berikut merupakan penggunaan *library session* untuk mengambil data pengguna:

Kode 3.9: Contoh penggunaan *library session*

```

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

```

Kode 3.9 merupakan penggunaan *library* ini untuk mengambil data dengan nama *logged\_in*. *SharIF Judge* menggunakan tabel untuk menyimpan *session* dengan isi kolom berupa *session\_id*, *ip\_address*, *user\_agent*, *last\_activity*, dan *user\_data*.

## 3.2 Analisis Sistem Usulan

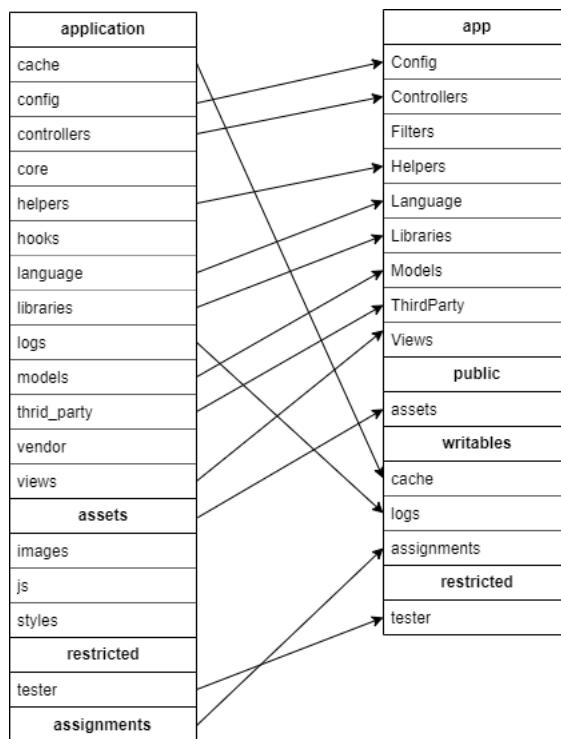
Seperti yang sudah dijelaskan pada sub bab 3.1, sistem saat ini masih menggunakan struktur pada *CodeIgniter 3*. *CodeIgniter 3* akan konversi menuju *CodeIgniter 4* tanpa perubahan fitur dan tampilan. Konversi *CodeIgniter 3* menuju *CodeIgniter 4* diperlukan penulisan ulang karena terdapat perubahan struktur aplikasi dan beberapa fungsi yang memiliki pemanggilan berbeda dan harus dilakukan pembaharuan. Oleh karena itu, berikut merupakan sistem usulan untuk melakukan konversi *SharIF Judge*:

### 3.2.1 Persiapan *CodeIgniter 4*

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

### 3.2.2 Struktur Aplikasi

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



Gambar 3.17: Pemindahan struktur aplikasi menuju *CodeIgniter 4*

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

#### Application

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

- *application/config* akan dipindahkan menuju *app/Config*.

- `application/controllers` akan dipindahkan menuju `app/Controllers`.
- `application/helpers` akan dipindahkan menuju `app/Helpers`.
- `application/languange` akan dipindahkan menuju `app/Languange`.
- `application/libraries` akan dipindahkan menuju `app/Libraries`.
- `application/models` akan dipindahkan menuju `app/Models`.
- `application/views` akan dipindahkan menuju `app/views`.

### **Config**

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

- `autoload.php` akan dipindahkan menuju `BaseController` seperti `helpers` dan terdapat penghapusan `autoload library` dan `model` karena terdapat perbedaan cara kerja.
- `config.php` akan dipecah dan dipindahkan menjadi tiga buah *file* yakni `Config/app.php`, `Config/Session.php`, dan `Config/Security.php`.
- `constants.php` akan dipindahkan menuju `Config/Constants.php`.
- `database.php` akan dipindahkan menuju `Config/Database.php`.
- `routes.php` akan dipindahkan menuju `Config/Routes.php`.
- `secrets.php` akan dibentuk baru pada direktori `Config`.

### **Public**

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

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

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

### **Writables**

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

- `application/cache` akan dipindahkan menuju `writables/cache`.
- `application/log` akan dipindahkan menuju `writables/logs`.
- `assignments` akan dipindahkan menuju `writables/assignments`.

### **3.2.3 Routing**

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

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

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

### 3.2.4 Model, View, and Controller

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

#### Model

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

1. Menggunakan *Model* dari *CodeIgniter 3*

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

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

```
1 public function get_all_users()
2 {
3     return $this->db->order_by('role', 'asc')->order_by('id')->get('users')->result_array();
4 }
```

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

Kode 3.11: Hasil konversi fungsi untuk mengambil data seluruh user

```
1 public function get_all_users()
2 {
3     return $this->db->table('users')->orderBy('role', 'asc')->orderBy('id')->get()->getResultSet();
4 }
```

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 juga terdapat perubahan sintaks menjadi *camelCase*.

2. Menggunakan *Model* pada *CodeIgniter 4*

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

```
make:model <name>
```

*Model* pada *CodeIgniter 4* menyediakan fungsi untuk mengambil, memasukan, dan memperbarui data dari sebuah tabel spesifik tanpa harus membangun secara manual fungsi-fungsi tersebut. Pengguna dapat melakukan inisiasi dan memakai fungsi untuk mengambil data menggunakan sintaks berikut.

```
$userModel = new \App\Models\UserModel();
$user = $userModel->findAll();
```

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

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

Sintaks diatas akan menghapus data dari tabel `notifications` sesuai dengan `id` yang diberikan. Fungsi ini tidak perlu membangun sintaks secara manual pada *file models* melainkan hanya perlu memanggil *model* yang ingin digunakan. Konversi aplikasi *SharIF Judge* akan menggunakan cara pertama dengan penghapusan dan pembaharuan beberapa sintaks. Selain itu, terdapat perubahan nama *model* dari yang sebelumnya menggunakan *snake case* menjadi *PascalCase* agar sesuai dengan standar PHP.

### **View**

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

Kode 3.12: Contoh *view* menggunakan *twig*

```
1  {% if registration_enabled %}  
2      <a href="{{ site_url('register') }}>Register</a> |  
3  {% endif %}
```

menjadi kode berikut:

Kode 3.13: Contoh *view* menggunakan *php*

```
1  <?php if($registration_enabled): ?>  
2      <a href=<?=site_url('register')_?>>Register</a> |  
3  <?php endif ?>
```

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

### **Controller**

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

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

```
1 protected $config;  
2  
3 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)  
4 {  
5     // Do Not Edit This Line  
6     parent::__construct($request, $response, $logger);  
7  
8     // Preload any models, libraries, etc, here.  
9  
10    // E.g.: $this->session = \Config\Services::session();  
11    $this->config = Config('Secrets');  
12 }
```

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

### **IncomingRequest**

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

Kode 3.15: Contoh perubahan *library request*

```

1 $this->request = \Config\Services::request();
2 $names => $this->request->getPost('name'),

```

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

### 3.2.5 Libraries

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

#### Emails

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

Kode 3.16: Contoh perubahan *library emails*

```

1 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
2     $this->email->setTo($user[1]);
3     $this->email->setSubject('SharIF_Judge_Username_and_Password');
4     $text = $this->settings_model->get_setting('add_user_mail');
5     $text = str_replace('{SITE_URL}', base_url(), $text);
6     $text = str_replace('{ROLE}', $user[4], $text);
7     $text = str_replace('{USERNAME}', $user[0], $text);
8     $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
9     $text = str_replace('{LOGIN_URL}', base_url(), $text);
10    $this->email->setMessage($text);
11    $this->email->send()

```

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

#### Working with Uploaded Files

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

Kode 3.17: Contoh perubahan *library upload*

```

1 $zip_uploaded = $this->request->getFile('tests_desc');
2 $pdf_uploaded->move("$assignment_dir");

```

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

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

```

1 if($pdf_uploaded->getExtension() != "pdf"){
2     $zip_uploaded->getName()
3 }

```

### **Working with URIs**

Kelas *URI* digantikan dengan fungsi *working with URIs* dengan perubahan fungsi. Berikut merupakan contoh pengambilan sebuah *segment* yang terdapat pada *URI*:

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

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

### **Session**

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

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

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

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

Sintaks diatas akan mengambil data *session* dengan nama *logged\_in*. Penggunaan berubah dari yang sebelumnya menggunakan sintaks *userdata*. Selanjutnya data tersebut akan disimpan menuju *database* dengan perubahan dan penghapusan kolom. Kolom tabel akan berubah menjadi empat buah yakni *id*, *ip\_address*, *timestamp*, dan data.

### **Validation**

*Form\_validation* diganti menggunakan fungsi *validation* dengan perubahan dan pengapusn beberapa fungsi. Berikut merupakan contoh penetapan aturan untuk melakukan validasi saat mengumpulkan sebuah data pada *form*:

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

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

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

```
1  public function email_check($email, $edit_username):bool
2  {
3      $edit_username = explode(',', $edit_username);
4      $user_model = new UserModel();
5      if ($user_model->have_email($email, $edit_username[1]))
6          return FALSE;
7      return TRUE;
8  }
```

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

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

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

### **Zip Archive**

*Zip Encoding* dan *Unzip* digantikan dengan fungsi PHP *zip archive* karena sudah tidak tersedia pada *CodeIgniter 4*. Fungsi *zip archive* terdapat beberapa perbedaan sehingga akan disesuaikan dengan 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*

```
1 $this->zip = new \ZipArchive();
2 $this->zip->open($zipname, ZipArchive::CREATE);
3 $this->zip->addFile($file_path, "{$item['username']}/p{$item['problem']}.".filetype_to_extension($item['file_type']));
```

Kode 3.20 berfungsi untuk melakukan inisiasi terhadap *ZipArchive* dan membuka sebuah *file zip*. Sintaks *addFile* selanjutnya memasukan *file* bernama *file\_path* menuju *zip* yang terdapat pada *array item*. Selain untuk membangun *file zip*, *zip archive* juga dapat melakukan *unzip* terhadap *file zip*. Kode 3.21 menunjukkan penggunaan *ZipArchive* dalam melakukan *unzip*.

Kode 3.21: Contoh perubahan penggunaan *library Zip Archive* untuk melakukan *unzip*

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

Kode 3.21 melakukan inisiasi terhadap *ZipArchive*. Kode selanjutnya akan membuka sebuah *zip* pada *assignments\_root* dengan nama yang dibentuk. Terakhir fungsi *extractTo* akan melakukan extraksi *file* pada direktori yang sudah dibentuk.

### **Twig**

*Library* ini tidak digunakan untuk membangun *view* pada *CodeIgniter 4* namun, akan ada penggunaan sebuah fungsi *Twig* yang dibentuk pada direktori *app/Libraries*. Fungsi tersebut bernama *extra\_time\_formatter* yang memiliki fungsi untuk mengubah input yang diberikan menjadi format jam dikali enam puluh menit.

### **Unzip**

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

### **Password\_hash**

*Library* ini tidak akan digunakan dan akan digantikan oleh *password hash* yang disediakan oleh PHP. *Library Password\_hash* merekomendasikan pegguna untuk menggunakan fungsi *native* yang disediakan oleh PHP apabila aplikasi mendukung PHP versi 5.5 ke atas. Sehingga, akan dilakukan konversi menggunakan fungsi yang disediakan oleh PHP bernama *password\_hash()*. Seluruh penggunaan *library* ini akan diubah menggunakan fungsi yang disediakan oleh PHP dengan metode *hashing* sama yaitu *CRYPT\_BLOWFISH*. Perubahan fungsi *hashing* ini bersifat *backward compatible* sehingga dapat menggunakan *database* aplikasi terdahulu tanpa perlu membangun data baru. Berikut merupakan contoh pengubahan kode dari *phpass* menjadi *password\_hash*.

```
'password' => $this->password_hash->HashPassword($password)
```

menjadi

```
'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 `CRIPT_BLOWFISH`.

### ***MY\_Form\_validation***

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

Kode 3.22: Contoh perubahan *library MY\_Form\_validation* pada *CodeIgniter 4*

```
1 namespace App\Libraries;
2
3 use CodeIgniter\Validation\Validation;
4
5 class MY_Form_validation extends Validation
```

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

### ***Parsedown***

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

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

```
1 namespace App\Libraries;
2
3 class Parsedown
```

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

### ***Phpexcel***

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

```
composer require phoffice/phpexcel
```

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

### ***Shj\_pagination***

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

## **3.2.6 Configuration**

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

- \$config['csrf\_protection'] = TRUE;
- \$config['csrf\_token\_name'] = 'shj\_csrf\_token';
- \$config['csrf\_cookie\_name'] = 'shjcsrftoken';
- \$config['csrf\_expire'] = 7200;
- \$config['csrf\_regenerate'] = FALSE;

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

### 3.2.7 Database

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

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

### 3.2.8 Error Handling

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

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

```
1 <?php
2
3 throw new \Exception('Some_message_goes_here');
```

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

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

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

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

### 3.2.9 Helpers

*Helpers* tidak terdapat banyak perubahan namun, terdapat perubahan pemanggilan *helpers* dan beberapa penghapusan *helpers*. Berikut merupakan *helpers* yang dihapus dan diubah cara pemanggilannya:

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

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

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

- **redirect()**

Fungsi ini memiliki perubahan pada *CodeIgniter 4* dimana **redirect()** tidak langsung mengarahkan kepada URL yang dibentuk. Pengguna harus mengembalikan fungsi **redirect()** menggunakan **return** dengan sintaks sebagai berikut.

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

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

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



## BAB 4

# PERANCANGAN

Bab ini membahas perancangan untuk seluruh implementasi *SharIF Judge* pada *CodeIgniter 4*. Seluruh perancangan dilakukan dengan cara membandingkan kedua kode pada *SharIF Judge* berbasis *CodeIgniter 3* dan *SharIF Judge* berbasis *CodeIgniter 4*.

### 4.1 Perancangan Perubahan Struktur Aplikasi

Struktur aplikasi *SharIF Judge* dipindahkan seperti pemetaan pada bab 3 gambar 3.17. Pemindahan dilakukan dengan penyesuaian dan perubahan beberapa fungsi yang digunakan. Berikut merupakan perancangan struktur baru pada *CodeIgniter 4*:

#### 4.1.1 app/Config

##### app/Config/App.php

File ini tidak digunakan sehingga dibiarkan berisi sintaks *default*. Seluruh data akan dipindahkan menuju file .env. Data-data yang dipindahkan antara lain adalah `baseURL`.

##### app/Config/Autoload.php

File ini tidak terdapat perubahan karena terdapat perbedaan cara penggunaan antara pada *CodeIgniter 3* dan *CodeIgniter 4*.

##### app/Config/Cache.php

File ini tidak terdapat perubahan karena pada *CodeIgniter 3* menggunakan konfigurasi *default*.

##### app/Config/Constants.php

File ini tidak terdapat perubahan namun terdapat pemindahan beberapa sintaks dari `constants.php`.

##### app/Config/Cookie.php

File ini tidak terdapat perubahan karena pada *CodeIgniter 3* menggunakan konfigurasi *default*.

##### app/Config/Database.php

File ini tidak digunakan sehingga dibiarkan berisi sintaks *default*. Seluruh data akan dipindahkan menuju file .env. Data-data yang dipindahkan antara lain adalah konfigurasi `database` aplikasi.

##### app/Config/Email.php

File ini tidak terdapat perubahan karena pada *CodeIgniter 3* menggunakan konfigurasi *default*.

**app/Config/Encryption.php**

File ini terdapat perubahan pada *encryption key* yang dipindahkan dari `config/config.php`.

**app/Config/Filters.php**

File ini terdapat penambahan penamaan seluruh *filters* yang telah dibentuk.

**app/Config/Routes.php**

File ini terdapat penambahan manual dari *routes* pada *SharIF Judge*.

**app/Config/Secrets.example.php**

File ini dipindahkan menuju direktori `Config` dengan perubahan `extends` dan penghapusan sintaks `defined`.

**app/Config/Security.php**

File ini terdapat perubahan pada nama `token` dan nama `cookie`.

**app/Config/Session.php**

File ini terdapat perubahan tempat penyimpanan *session* dan nama `cookie`.

**app/Config/Validation.php**

File ini berisikan seluruh aturan yang dibentuk secara manual oleh pengguna. Terdapat pemindahan seluruh aturan untuk melakukan validasi dan terdapat penambahan sintaks untuk memproses data.

#### 4.1.2 Controllers

Controller dipindahkan seluruhnya dari direktori `application/controllers` menuju direktori `app/Controllers`. Terdapat penghapusan sintaks `defined` dan perubahan `extends`. Berikut merupakan perubahan yang terdapat pada setiap *controller*:

**Assignments.php**

Berikut merupakan perubahan yang terdapat pada *controller* `Assignments.php`:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan *library* `Form_validation` menjadi `Validation`.
- Perubahan *library* `ZipEncoding` dan `UnZip` menjadi `Zip Archive`.
- Perubahan *library* `Input` menjadi `Request`.
- Perubahan `Upload` menjadi `Working with uploaded files`.
- Pemindahan pengecekan aplikasi menuju *Filters*.

**BaseController**

*Controller* `BaseController.php` merupakan penambahan *controller* baru pada *CodeIgniter 4*. Berikut merupakan penambahan sintaks pada *controller* ini:

- Pemanggilan *helpers*
- Pemanggilan *library third party* seperti `Parsedown` dan `Twig`.

### Dashboard.php

Berikut merupakan perubahan yang terdapat pada *controller Dashboard.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Halloffame.php

Berikut merupakan perubahan yang terdapat pada *controller Halloffame.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan *library Input* menjadi *Request*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Install.php

Berikut merupakan perubahan yang terdapat pada *controller Install.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan cara penggunaan *query*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Login.php

Berikut merupakan perubahan yang terdapat pada *controller Login.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan cara penggunaan *library session*.
- Perubahan *library Form\_validation* menjadi *Validation*.

### Logs.php

Berikut merupakan perubahan yang terdapat pada *controller Logs\_model*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Moss.php

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

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### **Notification.php**

Berikut merupakan perubahan yang terdapat pada controller *Notification.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### **Problems.php**

Berikut merupakan perubahan yang terdapat pada controller *Problems.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### **Profile.php**

Berikut merupakan perubahan yang terdapat pada controller *Profile.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### **Queue.php**

Berikut merupakan perubahan yang terdapat pada controller *Queue.php*

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan sintaks pengembalian *error*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### **Queueprocess.php**

Berikut merupakan perubahan yang terdapat pada controller *Queueprocess.php*:

- Perubahan cara pemanggilan *model* yang digunakan.
- Perubahan cara pengambilan data pada file *config*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### **Rejudge.php**

Berikut merupakan perubahan yang terdapat pada controller *Rejudge.php*:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Perubahan sintaks pengembalian *error*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Scoreboard.php

Berikut merupakan perubahan yang terdapat pada controller Scoreboard.php:

- Perubahan cara pemanggilan *model* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Server\_time.php

Controller ini hanya terdapat pemindahan pengecekan aplikasi menuju *Filters*.

### Settings.php

Berikut merupakan perubahan yang terdapat pada controller Settings.php:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Submission.php

Berikut merupakan perubahan yang terdapat pada controller Submission.php:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Perubahan cara penggunaan *library session*.
- Perubahan cara penggunaan *library URI*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Submit.php

Berikut merupakan perubahan yang terdapat pada controller Submit.php:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

### Users.php

Berikut merupakan perubahan yang terdapat pada controller User.php:

- Perubahan cara pemanggilan *model* dan *library* yang digunakan.
- Perubahan cara mengembalikan halaman *view*.
- Perubahan *library Input* menjadi *Request*.
- Perubahan *library Form\_validation* menjadi *Validation*.
- Perubahan cara penggunaan *library session*.
- Pemindahan pengecekan aplikasi menuju *Filters*.

#### 4.1.3 Model

Model dipindahkan seluruhnya dari direktori application/models menuju direktori app/Models. Berikut merupakan perubahan yang terdapat pada setiap model:

### **Assignment\_model.php**

Berikut merupakan perubahan yang terdapat pada `Assignment_model.php`:

- Perubahan nama *file* menjadi `AssignmentModel.php`.
- Perubahan cara pemanggilan *model* dan *library* lain yang digunakan.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.
- Perubahan penggunaan `input` menjadi `request`.
- Penambahan sintaks `table` pada seluruh *query*.

### **Hof\_model.php**

Berikut merupakan perubahan yang terdapat pada `Hof_model.php`:

- Perubahan nama *file* menjadi `HofModel.php`.
- Perubahan cara pemanggilan *model* lain yang digunakan.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

### **Logs\_model.php**

Berikut merupakan perubahan yang terdapat pada `Logs_model.php`:

- Perubahan nama *file* menjadi `LogsModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

### **Notifications\_model.php**

Berikut merupakan perubahan yang terdapat pada `Notifications_model.php`:

- Perubahan nama *file* menjadi `NotificationsModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

### **Queue\_model.php**

Berikut merupakan perubahan yang terdapat pada `Queue_model.php`:

- Perubahan nama *file* menjadi `QueueModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan cara pemanggilan *model* lain yang digunakan.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

### **Scoreboard\_model.php**

Berikut merupakan perubahan yang terdapat pada `Scoreboard_model.php`:

- Perubahan nama *file* menjadi `ScoreboardModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan cara pemanggilan *model* lain yang digunakan.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.
- Perubahan sintaks pengembalian halaman.

### **Settings\_model.php**

Berikut merupakan perubahan yang terdapat pada `Settings_model.php`:

- Perubahan nama *file* menjadi `SettingsModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

### **Submit\_model.php**

Berikut merupakan perubahan yang terdapat pada `Submit_model.php`:

- Perubahan nama *file* menjadi `SubmitModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan cara pemanggilan *model* lain yang digunakan.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

### **User\_model.php**

Berikut merupakan perubahan yang terdapat pada `User_model.php`:

- Perubahan nama *file* menjadi `UserModel.php`.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan cara pemanggilan *model* dan *library* lain yang digunakan.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.
- Perubahan *library* `HashPassword` menjadi `password_hash`.

### **User.php**

Berikut merupakan perubahan yang terdapat pada `User.php`:

- Perubahan cara pemanggilan *library* yang digunakan.
- Penambahan sintaks `table` pada seluruh *query*.
- Perubahan sintaks dari *snake case* menjadi *camelCase*.

#### **4.1.4 View**

*View* dipindahkan seluruhnya dari direktori `application/views` menuju direktori `app/Views`. *View* berubah dari yang sebelumnya menggunakan *template engine* bernama Twig menjadi menggunakan PHP. Seluruh *extension* dari *file View* berubah dari `.twig` menjadi `.php`. Selain itu terdapat perubahan beberapa *delimiters*. *Delimiters* yang menggunakan sintaks PHP berubah menjadi `<?php ?>`. *Delimiters* yang menggunakan fitur pada Twig berubah menjadi `<?= ?>`. Terdapat juga perubahan fungsi pada *template engine* Twig menjadi PHP dan perubahan cara pengambilan data dari *controller*.

#### **4.1.5 Filters**

*Filters* merupakan fitur baru pada *CodeIgniter 4*. Fitur ini memungkinkan untuk melakukan pengecekan sebelum dan setelah *controller* dijalankan. Fitur ini digunakan karena *controller* pada *CodeIgniter 4* tidak dapat mengembalikan apapun sehingga seluruh pengecekan pada *controller* dipindahkan menuju fitur ini. Berikut merupakan perancangan *filter* yang dibentuk:

- `CheckCLI.php`  
*Filters* ini merupakan pemindahan dari *controller Queueprocess.php*.
- `CheckInstallAndLogin.php`  
*Filters* ini merupakan pemindahan dari *controller Dashboard.php*.
- `CheckLogin.php`  
*Filters* ini merupakan pemindahan dari *controller Profile.php*, `Logs.php`, `Notification.php`, `Assignments.php`, `Submit.php`, `Submissions.php`, dan `Problems.php`.
- `CheckLoginandCLI.php`  
*Filters* ini merupakan pemindahan dari *controller Dashboard.php* dan `Scoreboard.php`.
- `CheckLoginandAjax.php`  
*Filters* ini merupakan pemindahan dari *controller Server\_time.php*.
- `CheckLoginandLevelAdmin.php`  
*Filters* ini merupakan pemindahan dari *controller Settings.php* dan `Users.php`.

- `CheckLoginandLevelHead.php`  
*Filters* ini merupakan pemindahan dari *controller* `Moss.php`, `Queue.php` dan `Rejudge.php`.

#### 4.1.6 *Routing*

*Routing* pada *SharIF Judge* berbasis *CodeIgniter 4* didefinisikan secara manual pada seluruh *controller* dan fungsinya.

#### 4.1.7 *Libraries*

Terdapat penghapusan *library* `Zip Encoding`, `UnZip`, `Twig`, dan `Password_hash`. *Library* lainnya dipindahkan menuju direktori **Libraries** dengan pengapusn sintaks `defined` dan penambahan sintaks `namespace`. *Library* `Twig` digunakan kembali untuk melakukan pengubahan format waktu dan tidak digunakan untuk menampilkan halaman.

## BAB 5

# IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai implementasi dan pengujian konversi *SharIF Judge*.

### 5.1 Lingkungan Implementasi dan Pengujian

Implementasi perangkat lunak *SharIF Judge* dilakukan pada dua buah lingkungan. Lingkungan pertama digunakan untuk membangun perangkat lunak sedangkan lingkungan kedua digunakan untuk melakukan pengujian. Berikut merupakan spesifikasi lingkungan implementasi dan pengujian yang digunakan:

#### 1. Lingkungan Pembangunan

Tabel 5.1 menunjukkan spesifikasi perangkat keras lingkungan pembangunan.

Tabel 5.1: Perangkat Keras Lingkungan Pembangunan

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 SSD

Tabel 5.2 menunjukkan spesifikasi perangkat lunak lingkungan pembangunan.

Tabel 5.2: Perangkat Lunak Lingkungan Pembangunan

Parameter	Nilai
Sistem Operasi	<i>MACOS Sonoma Version 14.2</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>Debian 11-slim</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> <i>PHP 8.1</i>

#### 2. Lingkungan Pengujian

Tabel 5.3 menunjukkan spesifikasi perangkat keras lingkungan pengujian.

Tabel 5.3: Perangkat Keras Lingkungan Pengujian

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>

Tabel 5.4 menunjukkan spesifikasi perangkat lunak lingkungan pengujian.

Tabel 5.4: Perangkat Lunak Lingkungan Pengujian

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 8.1.23</i> <i>phpMyAdmin 4.9.5</i>

## 5.2 Implementasi

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. Berikut merupakan hasil implementasi dari analisis dan perancangan perangkat lunak yang telah dibentuk:

### 5.2.1 Instalasi *CodeIgniter 4*

*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:

```
composer create-project codeigniter4/appstarter SharIF-JudgeCI4
```

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.

### 5.2.2 app/Config

*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:

#### app/Config/App.php

*File* ini tidak digunakan sehingga dibiarkan berisi sintaks *default* dan seluruh data akan dipindahkan menuju *file .env*. Kode 5.1 menunjukan isi dari *file .env* yang dipindahkan dari direktori *application/config.php*:

Kode 5.1: Kode *application/config/config.php* yang dipindahkan menuju *.env*

```
1 app.baseURL = 'http://sharif.localhost/'
```

Kode 5.1 akan menentukan URL dasar dari aplikasi menjadi `http://sharif.localhost/`. URL dibentuk spesifik berupa `http://sharif.localhost/` karena dijalankan pada server lokal sehingga tidak bertabrakan dengan server lokal lainnya. URL juga diganti karena tidak dapat diimplementasikan pada *CodeIgniter 4* sehingga URL ditentukan manual sesuai dengan *domain* yang dibentuk.

#### **app/Config/Autoload.php**

File ini tidak digunakan dan dibiarkan menggunakan konfigurasi *default* karena terdapat perubahan cara kerja kelas ini. Fitur ini pada *CodeIgniter 4* tidak dapat melakukan inisiasi otomatis terhadap *model*, *controller*, dan *library* saat aplikasi dijalankan. Namun kelas ini hanya akan mencari lokasi *file* sesuai dengan konfigurasinya saat kelas tersebut pertama kali diinisiasi. Kelas *helpers* yang diinisiasi dipindahkan menuju *BaseController* sedangkan kelas *model* dan *libraries* akan dipanggil menggunakan *PSR-4* pada *file-file* yang membutuhkan kelas tersebut.

#### **app/Config/Cache.php**

File tidak berubah dan akan tetap menggunakan konfigurasi *default* karena tidak terdapat perubahan pada *SharIF Judge* versi *CodeIgniter 3*.

#### **app/Config/Constants.php**

File ini berisikan seluruh data yang dipindahkan dari `application/config/constants.php`. Kode 5.2 menunjukkan data yang dipindahkan menuju `Constant.php`.

Kode 5.2: Pemindahan kode pada *Constant*

```

1 define('SHJ_VERSION', '1.4');
2 /*Code editor related constants*/
3 define('EDITOR_FILE_NAME', "editor");
4 define('EDITOR_FILE_EXT', "txt");
5 define('EDITOR_IN_NAME', "exec_in");
6 define('EDITOR_OUT_NAME', "exec_out");
7 define('EDITOR_SUBMIT_ID', 0);
8

```

Kode 5.2 menunjukkan pemindahan data dari file `application/config/constants.php` menuju `Constant.php`. Sintaks `SHJ_VERSION` akan digunakan pada tampilan `sidebar` sedangkan sintaks `EDITOR` digunakan pada fungsi `submit`. Sintaks yang telah dipindahkan akan bersifat *global* sehingga dapat diakses oleh seluruh *file* pada *CodeIgniter 4*.

#### **app/Config/Cookie.php**

File ini tidak terdapat perubahan dan tetap menggunakan konfigurasi *default* karena tidak terdapat perubahan pada *SharIF Judge* versi *CodeIgniter 3*.

#### **app/Config/Database.php**

File ini dibiarkan menggunakan konfigurasi *default* karena akan dipindahkan menuju `.env`. Kode 5.3 menunjukkan pemindahan menuju `.env` yang dipindahkan dari `application/config.php`.

Kode 5.3: 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 5.3 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.

### app/Config>Email.php

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 5.4 menunjukan pemindahan konfigurasi *email*.

Kode 5.4: Pemindahan *app/config/secrets.php* menuju *Email.php*

```
1 public string $protocol = 'smtp';
2 public string $SMTPHost = 'ssl://smtp.mailgun.org';
3 public string $mailType = 'html';
```

Kode 5.4 menunjukan 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*.

### app/Config/Encryption.php

File ini berisikan seluruh data yang dipindahkan dari *application/config/config.php*. Kode 5.5 menunjukan pemindahan konfigurasi enkripsi yang akan digunakan.

Kode 5.5: Pemindahan *app/config/config.php* menuju *Encryption.php*

```
1 public string $key = 'bj42CAiqTwLHUhz3p0vFRM0EJFeQD9uG';
```

Kode 5.5 menunjukan pemindahan *key* dari enkripsi yang akan digunakan. Data akan disimpan menuju variabel pada *CodeIgniter 4*.

### app/Config/Filters.php

File ini berisikan seluruh nama *filters* yang telah dibentuk. Kode 5.6 menunjukan pembangunan nama *filters* sesuai dengan kelas yang sudah dibentuk.

Kode 5.6: Penambahan nama *filters* untuk didefinisikan menuju *routes*

```
1 public array $aliases = [
2     'csrf'          => CSRF::class,
3     'toolbar'        => DebugToolbar::class,
4     'honeypot'       => Honeypot::class,
5     'invalidchars'   => InvalidChars::class,
6     'secureheaders'  => SecureHeaders::class,
7     'check-installandlogin' => CheckInstallAndLogin::class,
8     'check-login'    => CheckLogin::class,
9     'check-loginandLevelAdmin' => CheckLoginandLevelAdmin::class,
10    'check-loginandLevelHead' => CheckLoginandLevelHead::class,
11    'check-loginandCLI' => CheckLoginandCLI::class,
12    'check-loginandisajax' => CheckLoginandAjax::class,
13    'check-iscli'     => CheckCLI::class,
14 ]
```

Kode 5.6 menunjukan penambahan nama *filters* untuk dipanggil menuju *routes*. Penamaan ini akan ditambahkan pada array *\$aliases* dengan nama lain *filters* dan nama kelasnya.

### app/Config/Routes.php

File ini berisikan seluruh pemindahan dan penambahan manual dari *routes* pada *SharIF Judge*. Kode 5.7 menunjukan pembangunan *routes* aplikasi secara manual.

Kode 5.7: Penambahan *routes* yang digunakan pada aplikasi *SharIF Judge*

```
1 $routes->setDefaultNamespace('App\Controllers');
2 $routes->setDefaultController('Dashboard');
3 $routes->setDefaultMethod('index');
4 $routes->setTranslateURIDashes(false);
5 $routes->set404Override();
6 $routes->setAutoRoute(false);
7 // The Auto Routing (Legacy) is very dangerous. It is easy to create vulnerable apps
8 // where controller filters or CSRF protection are bypassed.
9 // If you don't want to define all routes, please use the Auto Routing (Improved).
10 // Set '$autoRoutesImproved' to true in 'app/Config/Feature.php' and set the following to true.
```

```

11 // $routes->setAutoRoute(false);
12 /*
13 * -----
14 * Route Definitions
15 * -----
16 */
17 /*
18 // We get a performance increase by specifying the default
19 // route since we don't have to scan directories.
20 $routes->get('/', 'Dashboard::index',[ 'filter' => 'check-installandlogin:dual,noreturn']);
21 $routes->get('/dashboard', 'Dashboard::index',[ 'filter' => 'check-installandlogin:dual,noreturn']);
22 $routes->post('/dashboard/widget_positions', 'Dashboard::widget_positions');
23 // Authentication
24 $routes->add('/install','Install::index');
25 $routes->add('/login','Login::index');
26 $routes->post('login/register','Login::register');
27 $routes->get('login/lost','Login::lost');
28 $routes->post('login/lost','Login::lost');
29 $routes->get('/reset/(:any)', 'Login::reset/$1');
30 $routes->post('login/reset/(:any)', 'Login::reset/$1');
31 $routes->get('/register','Login::register');
32 $routes->post('/register','Login::register');
33 $routes->get('/settings','Settings::index',[ 'filter' => 'check-loginandlevelAdmin:dual,noreturn']);
34 $routes->post('/settings','Settings::index',[ 'filter' => 'check_loginandlevelAdmin:dual,noreturn']);
35 $routes->add('/profile/(:num)','Profile::index/$1',[ 'filter' => 'check-login:dual,noreturn']);
36 $routes->get('/logout','Login::logout');
37 $routes->add('/settings/update', 'Settings::update');
38 // Users
39 $routes->get('/profile', 'Profile::index');
40 $routes->get('/users', 'Users::index',[ 'filter' => 'check-loginandlevelAdmin:dual,noreturn']);
41 $routes->get('/users/add', 'Users::add',[ 'filter' => 'check-loginandlevelAdmin:dual,noreturn']);
42 $routes->post('/users/add', 'Users::add',[ 'filter' => 'check-loginandlevelAdmin:dual,noreturn']);
43 $routes->post('/users/delete_submissions', 'Users::delete_submissions',[ 'filter' => 'check-loginandlevelAdmin:dual,noreturn']);
44 $routes->get('users/list_excel', 'Users::list_excel',[ 'filter' => 'check-loginandlevelAdmin:dual,noreturn']);
45 $routes->add('/notifications_and_scoreboard'
46 // Notifications and scoreboard
47 $routes->get('/notifications', 'Notifications::index',[ 'filter' => 'check-login:dual,noreturn']);
48 $routes->get('/notifications/add', 'Notifications::add',[ 'filter' => 'check-login:dual,noreturn']);
49 $routes->post('/notifications/add', 'Notifications::add',[ 'filter' => 'check-login:dual,noreturn']);
50 $routes->add('/notifications/edit/(:num)', 'Notifications::edit/$1',[ 'filter' => 'check-login:dual,noreturn']);
51 $routes->post('/notifications/delete', 'Notifications::delete',[ 'filter' => 'check-login:dual,noreturn']);
52 $routes->post('/notifications/check', 'Notifications::check',[ 'filter' => 'check-login:dual,noreturn']);
53 $routes->get('/scoreboard', 'Scoreboard::index',[ 'filter' => 'check-loginandlevelcli:dual,noreturn']);
54 $routes->add('/server_time','Server_time::index',[ 'filter' => 'check-loginandajax:dual,noreturn']);
55 // Logs, hof, moss, rejudge, queue
56 $routes->get('logs', 'Logs::index',[ 'filter' => 'check-login:dual,noreturn']);
57 $routes->get('halloffame', 'Halhoffame::index',[ 'filter' => 'check-login:dual,noreturn']);
58 $routes->get('moss/(:num)', 'Moss::index/$1',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
59 $routes->post('moss/(:num)', 'Moss::index/$1',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
60 $routes->get('moss/update/(:num)', 'Moss::update/$1',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
61 $routes->post('moss/update/(:num)', 'Moss::update/$1',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
62 $routes->get('queue', 'Queue::index',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
63 $routes->post('queue', 'Queue::index',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
64 $routes->post('queue/resume', 'Queue::resume',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
65 $routes->post('queue/pause', 'Queue::pause',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
66 $routes->post('queue/empty_queue', 'Queue::empty_queue',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
67 $routes->cli('queueprocess/run', 'Queueprocess::run');
68 $routes->get('rejudge', 'Rejudge::index',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
69 $routes->post('rejudge', 'Rejudge::index',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
70 $routes->get('rejudge/(:num)', 'Rejudge::index/$1',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
71 $routes->post('rejudge/rejudge_single', 'Rejudge::rejudge_single',[ 'filter' => 'check-loginandlevelHead:dual,noreturn']);
72 //Assignments
73 $routes->get('/assignments', 'Assignments::index',[ 'filter' => 'check-login:dual,noreturn']);
74 $routes->get('/assignments/add', 'Assignments::add',[ 'filter' => 'check-login:dual,noreturn']);
75 $routes->post('/assignments/add', 'Assignments::add',[ 'filter' => 'check-login:dual,noreturn']);
76 $routes->post('/assignments/select', 'Assignments::select',[ 'filter' => 'check-login:dual,noreturn']);
77 $routes->get('/assignments/edit/(:num)', 'Assignments::edit/$1',[ 'filter' => 'check-login:dual,noreturn']);
78 $routes->post('/assignments/edit/(:num)', 'Assignments::edit/$1',[ 'filter' => 'check-login:dual,noreturn']);
79 $routes->get('/assignments/delete/(:num)', 'Assignments::delete/$1',[ 'filter' => 'check-login:dual,noreturn']);
80 $routes->post('/assignments/delete/(:num)', 'Assignments::delete/$1',[ 'filter' => 'check-login:dual,noreturn']);
81 $routes->get('/assignments/pdf/(:num)/(:any)/(:any)', 'Assignments::pdf/$1/$2/$3',[ 'filter' => 'check-login:dual,noreturn']);
82 $routes->get('/assignments/pdf/(:num)', 'Assignments::pdf/$1',[ 'filter' => 'check-login:dual,noreturn']);
83 $routes->post('/assignments/pdfCheck/(:num)', 'Assignments::pdfCheck/$1');
84 $routes->get('/assignments/pdfCheck/(:num)', 'Assignments::pdfCheck/$1');
85 $routes->get('assignments/downloadtestsdesc/(:num)', 'Assignments::downloadtestsdesc/$1');
86 $routes->get('assignments/download_submissions/(:any)/(:num)', 'Assignments::download_submissions/$1/$2');
87 // Submit and submissions
88 $routes->get('submit', 'Submit::index',[ 'filter' => 'check-login:dual,noreturn']);
89 $routes->post('submit', 'Submit::index',[ 'filter' => 'check-login:dual,noreturn']);
90 $routes->get('submit/load/(:num)', 'Submit::load/$1',[ 'filter' => 'check-login:dual,noreturn']);
91 $routes->post('submit/save/(:any)', 'Submit::save/$1');
92 $routes->post('submit/save', 'Submit::save');
93 $routes->get('submit/get_output/(:num)', 'Submit::get_output/$1');
94 $routes->get('submissions/all/user/(:any)', 'Submissions::all');
95 $routes->get('submissions/final', 'Submissions::the_final',[ 'filter' => 'check-login:dual,noreturn']);
96 $routes->get('submissions/final/(:any)', 'Submissions::the_final/$1',[ 'filter' => 'check-login:dual,noreturn']);
97 $routes->post('submissions/view_code', 'Submissions::view_code');
98 $routes->post('/submissions/select', 'Submissions::select');
99 $routes->get('/submissions/all_excel', 'Submissions::all_excel');
100 $routes->get('/submissions/final_excel', 'Submissions::final_excel');
101 $routes->get('submissions/download_file/(:any)/(:num)/(:num)', 'Submissions::download_file/$1/$2/$3/$4');
102 $routes->get('submissions/all', 'Submissions::all');
103 // Problems
104 $routes->get('/problems', 'Problems::index',[ 'filter' => 'check-login:dual,noreturn']);
105 $routes->get('/problems/(:num)', 'Problems::index/$1');
106 $routes->get('/problems/(:num)/(:num)', 'Problems::index/$1/$2');
107 $routes->get('/problems/edit/(:any)/(:num)/(:num)', 'Problems::edit/$1/$2/$3');
108 $routes->post('/problems/edit/(:any)/(:num)', 'Problems::edit/$1/$2/$3');

```

Kode 5.7 menunjukkan seluruh *routes* yang didefinisikan secara explisit sesuai dengan kegunaannya. Terdapat juga penambahan *filters* untuk melakukan pengecekan sebelum *controller* diinisiasi. Pembangunan *routes* secara manual dilakukan karena alasan keamanan dan juga konfigurasi *filters* yang berbeda untuk setiap *controller*.

#### app/Config/Secrets.example.php

File ini merupakan file yang ditambahkan diluar dari struktur standart dari aplikasi *CodeIgniter 4*. File ini berisikan seluruh kebutuhan konfigurasi yang tidak terdapat pada config lainnya. Kode 5.8 menunjukkan pemindahan file *secrets.example.php* menuju app/Config/Secrets.example.php.

Kode 5.8: Penambahan file Secrets.example.php

```

1 namespace Config;
2
3 use CodeIgniter\Config\BaseConfig;
4 class Secrets extends BaseConfig
5 {
6     public $shj_authenticate = 'radius';
7
8     public $shj_radius = [
9         "server" => "localhost",
10        "secret" => "i-have-no-secret"
11    ];
12
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 5.8 menunjukkan 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 dan konfigurasi *email* pengguna.

#### app/Config/Security.php

File ini berisikan seluruh konfigurasi yang dipindahkan dari application/config/config.php. Kode 5.9 menunjukkan pemindahan data token *csrf* dan *cookie* menuju Security.php.

Kode 5.9: Pemindahan file config menuju Security.php

```

1 public string $tokenName = 'shj_csrf_token';
2 public string $cookieName = 'shjcsrftoken';
```

Kode 5.9 menunjukkan 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 5.10 menunjukkan pemindahan data konfigurasi *session* menuju Session.php.

Kode 5.10: Pemindahan file config menuju Session.php

```

1 public string $driver = 'CodeIgniter\Session\Handlers\DatabaseHandler';
2 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`.

### app/Config/Validation.php

*File* ini berisikan aturan yang dibentuk secara manual oleh pengguna untuk melakukan validasi sebelum *request* diproses oleh aplikasi. Kode 5.11 menunjukkan aturan validasi data yang dibentuk secara manual oleh pengguna.

Kode 5.11: Perancangan aturan yang dibentuk secara manual pada file Validation.php

```

1 class MyRules
2 {
3     public function password_check($str): bool
4     {
5         if (strlen($str) == 0 OR (strlen($str) >= 6 && strlen($str) <= 200))
6             return TRUE;
7         return FALSE;
8     }
9
10    public function password_again_check($str) :bool
11    {
12        $request = \Config\Services::request();
13        if ($request->getPost('password') !== $request->getPost('password_again'))
14            return FALSE;
15        return TRUE;
16    }
17
18    public function role_check($role)
19    {
20        $user = new User();
21        if ($user->level <= 2)
22            return ($role == ' ');
23
24        // Admins can change everybody's user role:
25        $roles = array('admin', 'head_instructor', 'instructor', 'student');
26        return in_array($role, $roles);
27    }
28
29    /**
30     * Checks whether a user with this email exists
31     */
32    public function email_check($email,$edit_username):bool
33    {
34        $edit_username = explode('/', $edit_username);
35        $user_model = new UserModel();
36        if ($user_model->have_email($email, $edit_username))
37            return FALSE;
38        return TRUE;
39    }
40
41    /**
42     * checks whether the entered registration code is correct or not
43     */
44
45    public function _registration_code($code){
46        $settings_model = new SettingsModel();
47        $rc = $settings_model->get_setting('registration_code');
48        if ($rc == '0')
49            return TRUE;
50        if ($rc == $code)
51            return TRUE;
52        return FALSE;
53    }
54
55    /**
56     * Required
57     *
58     * @param string
59     * @return bool
60     */
61    public function required($str)
62    {
63        return is_array($str) ? (bool) count($str) : ($str !== ' ');
64    }
65
66
67    /**
68     * Is Lowercase
69     *
70     * @param $str
71     * @return bool
72     */
73    public function is_lowercase($str)
74    {
75        return ctype_lower($str);
76    }

```

```

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}

```

Kode 5.11 menunjukan pembangunan aturan secara manual yang dipindahkan dari *controller* *Login.php*, *Profile.php*, *Submissions.php*, dan *Submit.php*. Aturan yang dibentuk secara manual dapat digunakan seperti aturan lainnya yang tersedia pada *CodeIgniter 4*. Aturan yang dibentuk terdapat beberapa penambahan seperti pemanggilan *library*, *model*, dan penambahan sintaks untuk memecah *array*.

### 5.2.3 Controllers

*Controller* terdapat perubahan pada bagian fungsi *\_\_construct()* dimana sekarang tidak dapat mengembalikan sesuatu. Oleh karena itu, akan dibentuk beberapa *filters* 5.2.2 untuk melakukan pengecekan terhadap fungsi yang sebelumnya terdapat pada *\_\_construct()*.

#### app/Controllers

Direktori ini berisikan seluruh *controller* yang dipindahkan dari *application/controllers*. Berikut merupakan isi pada direktori ini:

- *Assignments.php*
- *BaseController.php*
- *Dashboard.php*
- *Halloffame.php*
- *Install.php*
- *Login.php*
- *Logs.php*
- *Moss.php*
- *Notifications.php*
- *Problems.php*
- *Profile.php*
- *Queue.php*
- *Queueprocess.php*
- *Rejudge.php*
- *Scoreboard.php*
- *Server\_time.php*
- *Settings.php*
- *Submissions.php*
- *Submit.php*
- *Users.php*

*Controllers* terdapat perubahan dan penambahan baik dalam kelas yang dilakukan *extends* maupun dalam pemanggilan kelas lain seperti *model* dan *helpers*. Beberapa *helpers* yang dipanggil melalui *autoload* dipindahkan menuju *BaseController*. Kode 5.12 menunjukkan perubahan yang terdapat pada *controller Logs.php*.

Kode 5.12: Perubahan kode *controller Logs.php* pada *CodeIgniter 4*

```

1 namespace App\Controllers;
2
3 use App\Controllers\BaseController;
4 use App\Models\AssignmentModel;
5 use App\Models\LogsModel;
6 use App\Models\User;
7
8 class Logs extends BaseController
9 {
10     protected $session;
11     protected $user;
12     protected $logs_model;
13     protected $assignment_model;
14
15     public function __construct()
16     {
17         $this->session = session();
18         $this->logs_model = new LogsModel();
19         $this->assignment_model = new AssignmentModel();
20         $this->user = new User();
21         if ($this->user->level <= 2) // permission denied
22             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
23     }
24
25     public function index()
26     {
27
28         $data = array(
29             'logs' => $this->logs_model->get_all_logs(),
30             'selected' => 'logs',
31             'user' => $this->user,
32             'all_assignments' => $this->assignment_model->all_assignments(),
33             'finish_time' => $this->user->selected_assignment['finish_time'],
34             'extra_time' => $this->user->selected_assignment['extra_time'],
35         );
36
37         return view('pages/admin/logs', $data);
38     }
39 }
```

Kode 5.12 terdapat perubahan dimana sekarang akan *extends BaseController*. Terdapat juga penghapusan sintaks `defined('BASEPATH')` OR `exit('No direct script access allowed');`. Terdapat penambahan sintaks *namespace* dan juga beberapa sintaks untuk memanggil *models*. *Controller* juga memiliki perubahan dalam mengembalikan *view* dimana sekarang menggunakan sintaks *return view*. Selain itu terdapat penambahan pada *BaseController.php* untuk melakukan inisiasi terhadap *helpers* dan juga beberapa *library* yang akan digunakan. Kode 5.13 menunjukkan penambahan *helpers* yang dapat diakses oleh seluruh *controller*.

Kode 5.13: 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 5.13 menunjukkan pemindahan dan penambahan *helpers* dari file *autoload.php* menuju *BaseController*. *Helpers* 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.

### IncomingRequest

*Input* digantikan oleh fungsi *request* dengan perubahan cara pemanggilan sintaks. *Validation* hanya akan diinisiasi pada fungsi *\_\_construct* setiap *model* yang membutuhkan. Fungsi tidak akan diinisiasi pada *controller* karena *CodeIgniter 4* sudah menyediakan fitur ini pada *controller* sehingga hanya perlu dilakukan pemanggilan. Kode 5.14 menunjukkan inisiasi *request* pada *model*.

Kode 5.14: Perancangan inisiasi *request* pada *\_\_construct*

```

1 protected $request;
2
3     public function __construct()
4     {
5         $this->request = \Config\Services::request();
6     }
7 
```

Kode 5.14 menunjukkan inisiasi yang dilakukan pada *model*. *Controller* tidak perlu melakukan inisiasi terhadap fungsi ini karena sudah terinisiasi pada *BaseController*. Pengambilan *input* yang dilakukan oleh pengguna dilakukan menggunakan sintaks pada kode 5.15.

Kode 5.15: Perancangan penggunaan *request*

```

1 if($this->request->isAJAX())
2 $this->request->getPost('assignment_select')
3 
```

Kode 5.15 menunjukkan cara penggunaan *request* untuk melakukan pengecekan dan pengambilan data. Sintaks berubah dari yang sebelumnya menggunakan *snakecase* menjadi *camelCase*.

### 5.2.4 Filters

Pada *CodeIgniter 4* *\_\_construct()* tidak dapat mengembalikan sesuatu oleh karena itu akan 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 5.16 menunjukkan sintaks untuk melakukan pengecekan apakah dijalankan dari *command line interface*.

Kode 5.16: Perancangan kode pada *Filters CheckCLI.php*

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $request = \Config\Services::request();
4
5     if ($request->isCLI())
6         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
7 } 
```

Kode 5.16 menunjukkan 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 5.17 menunjukkan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan *install* dan *login*.

Kode 5.17: Perancangan kode pada *Filters CheckInstallAndLogin.php*

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $db = Database::connect();
4
5     if ( !$db->tableExists('sessions'))
6         return redirect()->to('/install');
7
8     $session = \Config\Services::session(); 
```

```

9     if ( !$session->get('logged_in') ) // if not logged in
10    return redirect()->to('/login');
11 }

```

Kode 5.17 menunjukkan 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 5.18 menunjukkan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan *login*.

Kode 5.18: Perancangan kode pada *Filters CheckLogin.php*

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $session = \Config\Services::session();
4
5     if ( !$session->get('logged_in') ) // if not logged in
6         return redirect()->to('login');
7 }

```

Kode 5.18 menunjukkan 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 5.19 menunjukkan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna menjalankan *request* dari *command line interface* dan apakah pengguna sudah melakukan *login*.

Kode 5.19: 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 }

```

Kode 5.19 menunjukkan 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 5.20 menunjukkan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan *login* dan apakah *request* yang diberikan pengguna berupa *ajax*.

Kode 5.20: Pemindahan kode pada *Filters CheckLoginandAjax.php*

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $session = \Config\Services::session();
4     $request = \Config\Services::request();
5
6     if ( !$session->get('logged_in') ) // if not logged in
7         return redirect()->to('/login');
8     if ( !$request->isAJAX() )
9         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
10 }

```

Kode 5.20 menunjukkan 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 5.21 menunjukkan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan *login* dan apakah *role* dari pengguna dapat mengakses situs tersebut.

Kode 5.21: Pemindahan kode pada *Filters CheckLoginandLevelAdmin.php*

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $session = \Config\Services::session();
4     $user = new User();

```

```

5     if ( !$session->get('logged_in')) // if not logged in
6         return redirect()->to('/Login');
7     if ( $user->level <= 2) // permission denied
8         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
9 }
10

```

Kode 5.21 menunjukkan 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 5.22 menunjukkan `filters` yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah `role` dari pengguna dapat mengakses situs tersebut.

Kode 5.22: Pemindahan kode pada `Filters CheckLoginandLevelHead.php`

```

1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $session = \Config\Services::session();
4     $user = new User();
5
6     if ( !$session->get('logged_in')) // if not logged in
7         return redirect()->to('/Login');
8     if ( $user->level <= 1) // permission denied
9         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
10

```

Kode 5.21 menunjukkan pemindahan dari `__construct` pada `controller Server_time`. Fungsi pada `filters` ini akan melakukan pengecekan `level` pengguna dan apakah pengguna sudah melakukan `login`. Selanjutnya `filter` ini akan mengembalikan pengguna kepada `routes` yang sesuai apabila tidak sesuai dengan kondisinya. Setelah dibentuk `filters`, selanjutnya akan ditambahkan menuju `file Filters.php` untuk mendefinisikan nama yang selanjutnya akan dimasukkan menuju `routes`. Penambahan menuju `file Filters.php` dapat dilihat pada subbab 5.2.2. Setelah ditambahkan, `filters` akan dipanggil pada `routes` yang membutuhkan pengecekan. Kode 5.23 menunjukkan penambahan `filters` pada `routes` sesuai dengan kebutuhannya.

Kode 5.23: Penambahan `filter` pada `routes`

```
$routes->get('/settings', 'Settings::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
```

Kode 5.23 menambahkan `filter` yang sudah dibentuk dan dinamakan setelah penulisan nama `controller` dan fungsinya.

### 5.2.5 *Helpers*

Direktori ini berisikan `helpers` yang dibentuk secara manual oleh pengguna bernama `shj_helper`. `Helpers` terdapat perubahan dan penghapusan sintaks. Sintaks `defined('BASEPATH') OR exit('No direct script access allowed');` akan dihapus dan akan ditambahkan beberapa `model` yang digunakan pada `helper` ini. `Model` yang ditambahkan berupa `settings_model`.

### 5.2.6 *Libraries*

`Libraries` terdapat beberapa perubahan dan penghapusan fungsi sehingga akan digantikan. Berikut merupakan perancangan perubahan fungsi pada `CodeIgniter 4`.

#### *Error Handling*

`Error Handling` pada `CodeIgniter 4` terdapat perubahan sintaks dan cara pemanggilan sehingga akan digantikan dengan sintaks yang baru. Berikut merupakan pemanggilan `error handling`:

Kode 5.24: Perubahan sintaks `error handling`

```
1     throw new \Exception('SharIF_Judge_is_already_installed');
```

Kode 5.24 menunjukkan perubahan sintaks untuk memberikan *error* dengan pesan yang ditentukan. Selain itu, terdapat perubahan dalam memberikan *error* berupa 404. Kode 5.25 menunjukkan sintaks untuk memberikan *error* berupa 404 halaman tidak ditemukan.

Kode 5.25: Perubahan sintaks *error handling* 404

```
1 throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_finished.');
```

Kode 5.25 akan mengembalikan *error* berupa halaman tidak ditemukan dengan pesan **Selected assignment has finished.** dalam *environment development* dan *testing*. Pada *environment production* tidak ditampilkan pesan *error* namun hanya akan ditampilkan halaman tidak ditemukan.

### Emails

*Emails* pada *CodeIgniter 4* terdapat perubahan sintaks dan cara pemanggilan sehingga akan dipindahkan sesuai dengan sintaks yang baru. Sintaks berubah dari yang sebelumnya menggunakan *snakecase* menjadi menggunakan *camelcase*. Kode 5.26 menunjukkan penggunaan *library email*.

Kode 5.26: Perubahan penggunaan sintaks pada *library emails*

```
1 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
2     $this->email->setTo($user[1]);
3     $this->email->setSubject('SharIF_Judge_Username_and_Password');
4     $text = $this->settings_model->get_setting('add_user_mail');
5     $text = str_replace('{SITE_URL}', base_url(), $text);
6     $text = str_replace('{ROLE}', $user[4], $text);
7     $text = str_replace('{USERNAME}', $user[0], $text);
8     $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
9     $text = str_replace('{LOGIN_URL}', base_url(), $text);
10    $this->email->setMessage($text);
11    $this->email->send()
```

Kode 5.26 memiliki sintaks dengan nama sama namun terdapat perubahan menjadi *camelcase*.

### Working with Uploaded Files

*Working with uploaded files* terdapat perubahan pada beberapa sintaks dan validasi terhadap *file* yang telah diunggah. Konversi aplikasi *SharIF Judge* akan menggunakan fungsi ini dengan beberapa perubahan sintaks sesuai dengan dokumentasinya. Kode 5.27 menunjukkan perubahan yang terdapat pada *library* ini.

Kode 5.27: Perancangan perubahan *library upload* pada *CodeIgniter 4*

```
1 $zip_uploaded = $this->request->getFile('tests_desc');
2     if ( $_FILES['tests_desc']['error'] === UPLOAD_ERR_NO_FILE ){
3         $this->messages[] = array(
4             'type' => 'notice',
5             'text' => "Notice: You did not upload any zip file for tests. If needed, upload by editing assignment."
6         );
7     }
8     elseif ( $zip_uploaded->getExtension() != "zip" ){
9         $this->messages[] = array(
10            'type' => 'error',
11            'text' => "Error: Error uploading tests zip file: The filetype you are attempting to upload is not allowed."
12        );
13    }
14    else{
15        $zip_uploaded->move($assignments_root);
16        $this->messages[] = array(
17            'type' => 'success',
18            'text' => "Tests (zip_file) uploaded successfully."
19        );
20    }
```

Kode 5.27 menunjukkan perubahan yang terdapat pada *library upload*. Pengambilan *file* digantikan dengan sintaks `$this->request->getFile('tests_desc')` dengan parameter berupa nama dari *tag form* yang sudah dibentuk. Selanjutnya dilakukan pengecekan terhadap *file* yang sudah diunggah dan memberikan beberapa *error message* sesuai dengan kondisinya. *File* yang sudah diunggah dipindahkan menuju direktori yang disimpan pada variabel `assignments_root`.

Konfigurasi aturan terhadap *upload* sudah tidak terdapat pada *CodeIgniter 4* sehingga akan diubah dan digantikan pada kondisi di baris kedua. Kondisi tersebut dibentuk untuk melakukan pengecekan tipe *file* yang sudah diunggah dan memberikan *error*. *Error* yang dihasilkan juga

sudah tidak dinamik karena tidak menggunakan *validation* sehingga disesuaikan dengan *error* yang terdapat pada *CodeIgniter 4*.

### **Working with URIs**

Kelas *URI* digantikan oleh fungsi *working with URIs* dengan perubahan dan penambahan fungsi. *Library* ini akan diinisiasi pada fungsi `__construct controller Submissions.php`. Kode 5.28 menunjukkan inisiasi *library* ini dan terdapat penambahan sintaks karena terdapat fungsi yang dihapus.

Kode 5.28: Perancangan inisiasi dan penambahan sintaks *library URIs* pada `__construct`

```

1 $temp_inp = $this->uri->getSegments();
2     $input=[];
3     foreach($temp_inp as $key => $value){
4         if($key != (count($temp_inp)-1) && ($key%2 != 0)){
5             $input[$value] = $temp_inp[$key+1];
6         }else{
7             break;
8         }
9     }

```

Kode 5.28 menunjukkan sintaks untuk mengambil *segment* yang terdapat pada *URI*. *Segment* ini akan disimpan dalam sebuah *array* dan tidak membentuk *associative array*. Oleh karena itu, terdapat penambahan sintaks `foreach` untuk mengubah *array* tersebut menjadi sebuah *associative array*.

### **Session**

Sintaks penggunaan *session* akan berubah dan *session* akan dilakukan inisiasi sebelum pemakaian. Kode 5.29 menunjukkan perubahan dalam melakukan inisiasi pada *session*.

Kode 5.29: Perancangan inisiasi *library session*

```

1 protected $session;
2
3 public function selected_assignment($username){
4     $this->session = session();
5 }
6
7 }

```

*Session* akan diinisiasi pada setiap *file* yang menggunakan *library* ini. Penggunaan *session* terdapat perubahan dari sintaks untuk pengambilan data, membersihkan data, dan penetapan data. Kode 5.30 menunjukkan perubahan dari sintaks untuk menggunakan *library session*.

Kode 5.30: Perancangan penggunaan *library session*

```

1 $this->session->get('logged_in')
2 $this->session->set($login_data);
3 $this->session->destroy();

```

Sintaks pada baris pertama berfungsi untuk mengambil data dengan nama *logged\_in*. Sintaks pada baris kedua berfungsi untuk menetapkan sebuah data *session* pada variabel *login\_data*. Terakhir pada baris ketiga merupakan sintaks untuk membersihkan data pada *session* pengguna tersebut.

### **Validation**

*Form\_validation* digantikan oleh fungsi *validation* dengan perubahan dan pengapusan beberapa fungsi. *Validation* akan diinisiasi pada fungsi `__construct` setiap *controller* dan *model* yang membutuhkan. Kode 5.31 menunjukkan inisiasi *validation* pada *controller*.

Kode 5.31: Perancangan inisiasi *validation* pada `__construct`

```

1 protected $validation;
2
3 public function __construct()
4 {
5     $this->validation = \Config\Services::validation();
6 }
7

```

Setiap *controller* yang menggunakan *validation* akan dideklarasikan sebuah variabel bernama *validation* agar dapat dipanggil pada seluruh fungsi yang membutuhkan pada kelas tersebut. *Validation* dilanjutkan dengan penetapan aturan untuk *tag form* yang diinginkan. Kode 5.32 menunjukkan contoh penetapan aturan terhadap *input* yang akan masukan oleh pengguna dengan nama *username*.

Kode 5.32: Perancangan perubahan konfigurasi aturan pada *library validation*

```
1 $this->validation->setRule('username', 'username', 'required|min\_length[3]|max\_length[20]);
```

Sintaks untuk melakukan penetapan aturan berubah menjadi *setRule*. Beberapa aturan yang dibentuk secara manual akan dipindahkan menuju *file Validation.php* dan terdapat penghapusan aturan yang sudah tidak terdapat pada *CodeIgniter 4*. Pembangunan aturan dapat dilihat pada sub bab 5.2.2 kode 5.11. Aturan yang sudah dibentuk dapat digunakan seperti aturan lainnya dengan cara menulis nama kelasnya. Setelah aturan ditetapkan, *validation* akan dieksekusi berdasarkan *request* dari pengguna dan dilakukan validasi. Kode 5.33 menunjukkan perubahan penggunaan *validation* terhadap data yang sudah diberikan oleh pengguna.

Kode 5.33: Perancangan perubahan penggunaan *validation* pada *CodeIgniter 4*

```
1 if ($this->validation->withRequest($this->request)->run())
2 {
3     if ( !$this->request->isAJAX() ){
4         exit;
5     }else{
6         list($ok, $error) = $this->user_model->add_users(
7             $this->request->getPost('new_users'),
8             $this->request->getPost('send_mail'),
9             $this->request->getPost('delay')
10            );
11        return view('pages/admin/add_user_result', array('ok' => $ok, 'error' => $error));
12    }
13 }
```

Kode 5.33 menjalankan *validation* berdasarkan *request* dari pengguna. *Validation* akan tetap menggunakan sintaks *run()* namun akan ada penambahan sintaks *withRequest* dimana validasi akan dijalankan setiap ada HTTP *Request* dari pengguna. Namun, *CodeIgniter 4* tidak menyediakan fungsi *form\_error* sehingga akan diubah dengan menggunakan fungsi baru bernama *validation\_errors()*. Fungsi tersebut dapat digunakan untuk mengembalikan *error* apabila terdapat data yang tidak sesuai dengan aturan. *Error* tersebut dapat ditampilkan pada halaman *view* menggunakan sintaks berikut:

```
<?= $validation->hasError('username') ? $validation->getError('username') : '' ?>
```

Sintaks diatas akan melakukan pengecekan apakah terdapat *error* dari *form* bernama *username* dan apabila terdapat maka akan dikembalikan data *error* yang berasal dari *validation*. Variabel *validation* akan dikirimkan dari *controller* berisikan *library validation*.

### Zip Archive

*Zip Encoding* akan digantikan dengan fungsi PHP *zip archive* karena sudah tidak tersedia pada *CodeIgniter 4*. Fungsi *zip archive* terdapat beberapa perbedaan sehingga akan disesuaikan dengan fungsi-fungsi yang ada. Kode 5.34 menunjukkan perubahan yang terdapat pada *zip encoding* menjadi *zip archive*.

Kode 5.34: Perancangan perubahan *zip encoding* menjadi *zip archive*

```
1 $this->zip = new \ZipArchive();
2     $this->zip->open($zipname, ZipArchive::CREATE);
3     for ($i=1 ; $i<=$number_of_problems ; $i++)
4     {
5
6         $path = "$root_path/p{$i}/in";
7         $options = ['add_path' => "{$i}/in/", 'remove_all_path' => TRUE];
8         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
9
10        $path = "$root_path/p{$i}/out";
11        $options = ['add_path' => "{$i}/out/", 'remove_all_path' => TRUE];
```

```

12 $this->zip->addGlob($path.'*.{txt}', GLOB_BRACE, $options);
13 $path = "$root_path/p{$i}/tester.cpp";
14 if ($file_exists($path))
15     $this->zip->addFile($path, "p{$i}/tester.cpp");
16
17 $pdf_files = glob("$root_path/p{$i}/*.pdf");
18 if ($pdf_files)
19 {
20     $path = $pdf_files[0];
21     $this->zip->addFile($path, "p{$i}/".shj_basename($path));
22 }
23
24 $path = "$root_path/p{$i}/desc.html";
25 if ($file_exists($path))
26     $this->zip->addFile($path, "p{$i}/desc.html");
27
28 $path = "$root_path/p{$i}/desc.md";
29 if ($file_exists($path))
30     $this->zip->addFile($path, "p{$i}/desc.md");
31 }
32
33 $pdf_files = glob("$root_path/*.pdf");
34 if ($pdf_files)
35 {
36     $path = $pdf_files[0];
37     $this->zip->addFile($path, shj_basename($path));
38 }
39
40 $this->zip->close();
41
42 header('Content-Type: application/zip');
43 header('Content-disposition: attachment; filename=' . $zipname);
44 header('Content-Length: ' . filesize($zipname));
45 readfile($zipname);

```

*Zip archive* akan dilakukan inisiasi pada variabel `zip`. Selanjutnya akan dibuka *file zip* menggunakan sintaks `open` yang menerima dua buah parameter. Parameter pertama berisikan nama *zip file* yang ingin dibentuk sedangkan parameter kedua berisikan mode *zip* yang diinginkan. Sintaks `addGlob` akan memasukan seluruh *file* sesuai dengan pattern yang telah ditentukan. Sintaks `addFile` akan memasukan *file* sesuai dengan *path* yang telah ditentukan. Selanjutnya *zip* akan ditutup menggunakan sintaks `close` dan dapat diunduh oleh pengguna menggunakan fungsi `header` 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 5.35 menunjukkan penggunaan *zip archive* untuk *unzip* sebuah *file* pada *controller*.

Kode 5.35: Perancangan perubahan *unzip* menggunakan *zip archive* pada *controller*

```

1 $this->unzip = new ZipArchive();
2 // Create a temp directory
3 $tmp_dir_name = "shj_tmp_directory";
4 $tmp_dir = "$assignments_root/$tmp_dir_name";
5
6 // Extract new test cases and descriptions in temp directory
7 $this->unzip->open("$assignments_root/", $zip_uploaded->getName());
8 $extract_result = $this->unzip->extractTo($tmp_dir);
9 $this->messages[] = array(
10     'type' => 'error',
11     'text' => "_Zip_Extraction_Error:_".$this->unzip->getStatusString(),
12 );

```

Kode 5.35 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 5.35 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.

### ***Password\_hash***

*Library* ini tidak akan digunakan dan akan digantikan oleh *password hash* yang disediakan oleh PHP. *Library Password\_hash* merekomendasikan pegguna untuk menggunakan fungsi *native* yang disediakan oleh PHP apabila aplikasi mendukung PHP versi 5.5 ke atas. Sehingga, akan dilakukan konversi menggunakan fungsi yang disediakan oleh PHP bernama `password_hash()`. Seluruh penggunaan *library* ini akan diubah menggunakan fungsi yang disediakan oleh PHP dengan metode *hashing* sama yaitu *CRYPT\_BLOWFISH*. Perubahan fungsi *hashing* ini bersifat *backward compatible*

sehingga dapat menggunakan *database* aplikasi terdahulu tanpa perlu membentuk data baru. Berikut merupakan contoh pengubahan kode dari *phpass* menjadi *password\_hash*.

```
'password' => $this->password_hash->HashPassword($password)
```

menjadi

```
'password' => password_hash($password, PASSWORD_BCRYPT)
```

Sintaks *password\_hash()* diatas menerima dua buah parameter yakni data yang ingin dienkripsi dan tipe enkripsi. Enkripsi akan menggunakan sintaks *PASSWORD\_BCRYPT* yang menggunakan tipe *hash* berupa *CRIPT\_BLOWFISH*. Selain itu, terdapat fungsi untuk melakukan pengecekan *password* yang sudah dienkripsi. Berikut merupakan contoh pengubahan kode untuk melakukan pengecekan *password* yang sudah dienkripsi.

```
password_verify($password, $query->getRow()->password)
```

Sintaks diatas menerima dua buah parameter dengan parameter pertama berupa masukan dari pengguna dan parameter berikutnya merupakan *hash* dari *password* yang sudah disimpan. Fungsi ini akan mengembalikan data berupa *true* apabila *password* sama dan *false* apabila *password* berbeda.

*Library* yang terdapat pada *CodeIgniter 4* juga dapat diextend dan dibentuk sesuai dengan kebutuhan. Berikut merupakan *library* yang dibentuk oleh pengguna dan akan dipindahkan menuju *app/Libraries*.

### *Twig*

*Library* ini tidak akan digunakan untuk membentuk *view* pada *CodeIgniter 4* namun, akan ada penggunaan sebuah fungsi *Twig* yang akan dibentuk pada direktori *app/Libraries*. Fungsi tersebut bernama *extra\_time\_formatter* yang memiliki fungsi untuk mengubah input yang diberikan menjadi format jam dikali enam puluh menit. Kode 5.36 menunjukkan fungsi yang dibentuk pada direktori *app/Libraries* dengan nama *Twig.php*. Fungsi ini nantinya digunakan pada halaman *add\_assignment.php*.

Kode 5.36: Perancangan perubahan *library MY\_Form\_validation* pada *CodeIgniter 4*

```

1 <?php
2 namespace App\Libraries;
3
4 class Twig
5 {
6
7     /**
8      * Required
9      *
10     * @param  string
11     * @return bool
12     */
13    public function extra_time_formatter($extra_time)
14    {
15        // convert to minutes
16        $extra_time = floor($extra_time/60);
17        // convert to H:60
18        if ($extra_time % 60 == 0 )
19            $extra_time = ($extra_time/60) . ':00';
20        return $extra_time;
21    }
22 }
```

### *MY\_Form\_validation*

*Library MY\_Form\_validation* akan dipindahkan menuju direktori *app/Libraries*. *Library* ini akan digunakan kembali dengan perubahan *extends* menjadi menuju *Validation*, penghapusan sintaks *defined*, dan akan ada penambahan *namespace* pada baris awal file. Kode 3.22 menunjukkan contoh penambahan *namespace* dan penggantian *extends* pada *library* ini.

Kode 5.37: Contoh perubahan *library MY\_Form\_validation* pada *CodeIgniter 4*

```
1 namespace App\Libraries;
2
3 use CodeIgniter\Validation\Validation;
4
5 class MY_Form_validation extends Validation
```

Kode 5.37 mengapus sintaks `defined` dan menggantikannya dengan penambahan *namespace*. Selain itu, kelas *library* akan *extends Validation*.

### *Parsedown*

*Library Parsedown* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini akan digunakan kembali dengan penambahan *namespace* pada baris awal *file* dan penghapusan sintaks `defined`. Kode 5.38 menunjukan contoh penambahan *namespace* dan juga penambahan sintaks `defined`.

Kode 5.38: Perancangan perubahan *library Parsedown* pada *CodeIgniter 4*

```
1 namespace App\Libraries;
2
3 class Parsedown
```

Kode 5.38 menghapus sintaks `defined` dan menggantikannya dengan penambahan *namespace*. Penggunaan *library* ini tidak akan berubah sehingga tidak terdapat perbedaan sintaks. Namun, terdapat perubahan cara inisiasi *library* ini dimana sekarang akan menggunakan sintaks `new` dan dilakukan inisiasi pada *BaseController*. Kode 5.39 menunjukan perubahan cara melakukan inisiasi *library parsedown*.

Kode 5.39: Perancangan perubahan inisiasi *library Parsedown* pada *controller CodeIgniter 4*

```
1 protected $parsedown;
2
3 /**
4 * Constructor.
5 */
6 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
7 {
8     // Do Not Edit This Line
9     parent::initController($request, $response, $logger);
10
11    // Preload any models, libraries, etc, here.
12
13    // E.g.: $this->session = \Config\Services::session();
14    $this->parsedown = new Parsedown();
15 }
```

*Library parsedown* akan dilakukan inisiasi menuju variabel `parsedown` yang sudah dibentuk pada luar fungsi. Inisiasi dilakukan pada *BaseController* untuk mempermudah pemanggilan pada beberapa *controller*.

### *Phpxcel*

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

```
composer require phpoffice/phpexcel
```

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

### *Shj\_pagination*

*Library* ini akan digunakan kembali dan dipindahkan menuju direktori `app/Libraries`. Selain itu, terdapat penambahan *namespace* pada baris awal *file* dan penghapusan sintaks `defined`. Kode 5.40 menunjukan penambahan perubahan dan penambahan sintaks pada *library Shj\_pagination*.

Kode 5.40: Perancangan perubahan *library Shj\_pagination* pada *CodeIgniter 4*

```

1 namespace App\Libraries;
2
3 class Shj_pagination
4 {

```

Selanjutnya *library* dapat dipanggil menggunakan sintaks berikut:

```
$shj_pagination = new Shj_pagination($config);
```

Sintaks diatas akan memanggil *library* dan memasukan konfigurasi yang telah dibentuk pada variabel *config*.

### 5.2.7 Models

Direktori ini akan berisikan seluruh *model* yang dipindahkan dari *CodeIgniter 3*. Berikut merupakan isi pada direktori ini:

- AssignmentModel.php
- HofModel.php
- LogsModel.php
- NotificationsModel.php
- QueueModel.php
- ScoreboardModel.php
- SettingsModel.php
- SubmitModel.php
- User.php
- UserModel.php

Seluruh *Model* akan diganti penamaannya dari yang sebelumnya menggunakan *snakecase* menjadi *PascalCase*. Kode 5.41 menunjukan perubahan sintaks yang terdapat pada *model*.

Kode 5.41: Perancangan perubahan *model* pada *CodeIgniter 4*

```

1 namespace App\Models;
2
3 use App\Libraries\Parsedown;
4 use CodeIgniter\Model;
5 use App\Models\SettingsModel;
6 use App\Models\ScoreboardModel;
7
8 class AssignmentModel extends Model
9 {
10     protected $settings_model;
11     protected $scoreboard_model;
12     protected $request;
13     protected $parsedown;
14
15     public function __construct()
16     {
17         parent::__construct();
18         $this->settings_model = new SettingsModel();
19         $this->request = \Config\Services::request();
20     }
21
22     /**
23      * Add New Assignment to DB / Edit Existing Assignment
24      *
25      * @param $id
26      * @param bool $edit
27      * @return bool
28      */
29     public function add_assignment($id, $edit = FALSE)
30     {
31         $this->scoreboard_model = new ScoreboardModel();

```

Terdapat penghapusan sintaks *defined* dan juga terdapat perubahan dalam *extends* dimana sekarang akan *extends Model*. Selain itu terdapat pemanggilan *model* lainnya menggunakan sintaks *new* yang disimpan melalui variabel yang sudah dibentuk. Terdapat juga perubahan cara pengambilan data menggunakan fungsi *database* yang ditunjukan pada Kode 5.42.

Kode 5.42: Perubahan sintaks pada *model*

```

1 public function all_assignments()

```

```

2| {
3|     $result = $this->db->table('assignments')->orderBy('id')->get()->getResultArray();
4|     $assignments = [];
5|     foreach ($result as $item)
6|     {
7|         $assignments[$item['id']] = $item;
8|     }
9|     return $assignments;
10}

```

Pengambilan data pada *CodeIgniter 4* perlu mendefinisikan nama tabel yang akan diambil datanya. Selanjutnya sintaks akan bersifat sama namun terdapat perubahan dari yang sebelumnya *snakecase* menjadi *camelCase*. *Model* yang telah dibentuk dapat dipanggil menuju *controller* menggunakan sintaks berikut:

```
$this->settings_model = new SettingsModel();
```

Sintaks diatas akan memanggil *SettingsModel* dan menyimpannya menuju variabel. Fungsi *database* juga dapat digunakan *file* lain dengan melakukan inisiasi dengan sintaks berikut:

```
$this->db = db_connect();
```

### 5.2.8 View

Direktori ini berisikan seluruh *view* yang dipindahkan dari *CodeIgniter 3*. Berikut merupakan isi dari direktori *view*:

#### errors

Direktori ini berisikan *file default* dari *CodeIgniter 4* karena tidak terdapat perubahan pada *SharIF Judge CodeIgniter 3*.

#### pages

- assignments.php
- dashboard.php
- halloffame.php
- notifications.php
- problems.php
- profile.php
- scoreboard\_table.php
- scoreboard.php
- submissions.php
- submit.php
- Direktori admin:
  - add\_assignment.php
  - add\_notification.php
  - add\_user\_result.php
  - add\_user.php
  - delete\_assignment.php
  - edit\_problem\_html.php
  - edit\_problem\_md.php
  - edit\_problem\_plain.php
  - install.php
  - logs.php
  - moss.php
  - queue.php
  - rejudge.php

- settings.php
- users.php
- Direktori authentication:
  - login.php
  - lost.php
  - register\_success.php
  - register.php
  - reset\_password.php
- Direktori templates:
  - base.php
  - side\_bar.php
  - simple\_header.php
  - top\_bar.php

*View* akan diubah menggunakan *extension .php* karena *twig* sudah tidak terintegrasi *CodeIgniter 4*. Seluruh *file view* akan diubah menjadi *extension .php* dari yang sebelumnya menggunakan *.twig*. Seluruh *delimiters* juga akan diubah menggunakan fungsi pada *CodeIgniter 4*. Perubahan *view* ditunjukan pada kode 5.43.

Kode 5.43: Perubahan *view* pada *Login.php*

```

1 <!-- {#
2 # SharIF Judge
3 # file: login.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <!DOCTYPE html>
7 <html lang="en">
8 <?= $this->include('templates/simple_header')?>
9
10 <?= form_open() ?>
11   <div class="box_login">
12
13     <div class="judge_logo">
14       <a href="<?= _site_url() ?>"></a>
15     </div>
16
17     <div class="login_form">
18       <div class="login1">
19         <p>
20           <label for="form_username">Username</label><br/>
21           <input id="form_username" type="text" name="username" required="required" pattern="[0-9a-z]{3,20}" title="The_
22             Username_field_must_be_between_3_and_20_characters_in_length,_andContain_only_digits_and_lowercase_
23             letters" class="sharif_input" value="<?= _set_value('username') ?>" autofocus="autofocus"/>
24           <?= iset($this->errors['username']) ?>
25         </p>
26         <p>
27           <label for="form_password">Password</label><br/>
28           <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The_
29             Password_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
30           <?= iset($this->errors['password']) ?>
31         </p>
32         <?php if ($error): ?>
33           <div class="shj_error">Incorrect username or password.</div>
34         <?php endif ?>
35       </div>
36       <div class="login2">
37         <p style="margin:0;">
38           <?php if ($registration_enabled): ?>
39           <a href="<?= _site_url('register') ?>">Register</a> |
40           <?php endif ?>
41           <a href="<?= _site_url('login/lost') ?>">Reset Password</a>
42           <input type="submit" value="Login" id="sharif_submit"/>
43         </p>
44       </div>
45     </div>
46   </body>
47 </html>
```

*Delimiters {{ }}* digantikan menjadi *<?= ?>* sedangkan *{% %}* digantikan menjadi *<?php ?>*. *Delimiters* yang memanggil fungsi pada *CodeIgniter 4* akan digantikan menjadi *<?= ?>*. Perubahan juga terdapat pada sintaks dari yang sebelumnya menggunakan *include* akan digantikan menggunakan fungsi *CodeIgniter 4* berupa *\$this->include*. Selain terdapat perubahan *extension* dan *delimiters*, terdapat juga penambahan kode pada *Controller* karena tidak mendukung pembangunan variabel *global* pada *View*. Kode 5.44 menunjukan contoh penambahan kode pada *Controller*.

Kode 5.44: Penambahan kode pada *controller Login.php*

```

1 $data = [
2     'error' => FALSE,
3     'registration_enabled' => $this->settings_model->get_setting('enable_registration'),
4     'title' => 'Login',
5     'validationError' => $this->validation
6 ];

```

Penambahan data terjadi karena halaman *view* PHP tidak dapat mendeklarasikan variabel secara *global* sehingga data-data seperti *title* tidak dapat diakses oleh *view* lainnya.

### 5.2.9 public

#### assets

Direktori ini berisikan seluruh data yang dapat dilihat oleh pengguna seperti *javascript*, gambar, dan lainnya. Berikut merupakan isi pada direktori ini:

- ace
- font
- fullcalendar
- gridster
- images
- js
- nano\_scroller
- noty
- pdfjs
- reveal
- snippet
- styles
- tinymce

Direktori *assets* akan dipindah tanpa ada perubahan apapun.

### 5.2.10 restriced

Direktori ini berisikan direktori *tester* yang berfungsi untuk melakukan *sandbox* dan *shield*. Direktori ini dipindahkan tanpa ada perubahan apapun.

### 5.2.11 Kode Program

Perubahan dan penambahan kode program untuk melakukan konversi perangkat lunak telah dijabarkan pada bab 4. Seluruh perubahan dan penambahan dibentuk menggunakan bahasa pemrograman PHP dengan bantuan beberapa perangkat lunak tambahan. Kode program untuk *controller* dapat dilihat pada Lampiran A bab A.1. Kode program untuk *model* dapat dilihat pada Lampiran A bab A.2. Kode program untuk *view* dapat dilihat pada Lampiran A bab A.3. Kode program lainnya dapat dilihat pada Lampiran A bab A.4.

### 5.2.12 Basis Data

Basis data terdapat perubahan pada tabel *sessions* dimana perubahan terjadi pada nama kolom dan penghapusan kolom. Terdapat perubahan nama kolom *session\_id* menjadi *id*, *user\_data* menjadi *data*, dan *last\_activity* menjadi *timestamp*. Terdapat penghapusan kolom *user\_agent*, perubahan 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 (primary key)</i>	varchar	128	None
<i>ip_address</i>	varchar	45	None
<i>timestamp</i>	timestamp	-	current_timestamp()
<i>data</i>	blob	-	None

### 5.3 Pengujian Fungsional

Pengujian fungsional dilakukan pada lingkungan pengujian yang bertujuan untuk melakukan pengujian terhadap seluruh fitur pada perangkat lunak *SharIF Judge*. Pengujian dilakukan untuk memastikan setiap fitur yang terdapat pada *SharIF Judge* versi *CodeIgniter 4* dapat berjalan dengan baik setelah dilakukan konversi. Pengujian dilakukan dengan membandingkan fitur yang terdapat pada *SharIF Judge* versi *CodeIgniter 3* dan *SharIF Judge* versi *CodeIgniter 4*. Tabel 5.6 menunjukkan pengujian terhadap fitur-fitur pada *SharIF Judge*.

Tabel 5.6: Tabel Pengujian Fungsional

No	Aksi Pengguna	Reaksi yang diharapkan	Reaksi
1	Melakukan instalasi <i>SharIF Judge</i>	<i>Database</i> terbentuk dan akun admin tersimpan pada <i>database</i>	Sesuai
2	Melakukan pengamanan terhadap <i>sandbox</i>	Dapat membangun <i>sandbox</i> tanpa mengembalikan error	Sesuai
3	Melakukan <i>Login</i>	Pengguna masuk menuju dashboard menggunakan akun yang telah dibentuk	Sesuai
4	Melakukan <i>Register</i>	Pengguna berhasil melakukan <i>register</i> dan aplikasi mengembalikan <i>Registered successfully</i>	Sesuai
5	Melakukan <i>reset password</i> dengan <i>email</i> yang telah terdaftar	Dikirimkan <i>email</i> berupa <i>link</i> untuk melakukan <i>reset password</i> , pengguna diarahkan menuju situs <i>SharIF Judge</i> , dan pengguna memasukan <i>password</i> baru untuk <i>login</i>	Sesuai
6	<i>Top bar</i> pengguna <i>admin</i>	<i>Top bar</i> terdapat <i>dropdown tools</i> , <i>dropdown select assignments</i> , <i>timer assignment</i> , dan <i>icon profile</i> . <i>Drop-down tools</i> berisikan <i>rejudge</i> , <i>submission queue</i> , dan <i>cheat detection</i>	Sesuai
7	<i>Top bar</i> pengguna <i>head instructor</i>	<i>Top bar</i> terdapat <i>dropdown tools</i> , <i>dropdown select assignments</i> , <i>timer assignment</i> , dan <i>icon profile</i> . <i>Drop-down tools</i> berisikan <i>rejudge</i> , <i>submission queue</i> , dan <i>cheat detection</i>	Sesuai
8	<i>Top bar</i> pengguna <i>instructor</i>	<i>Top bar</i> terdapat <i>dropdown select assignments</i> , <i>timer assignment</i> , dan <i>icon profile</i>	Sesuai

Bersambung ke halaman berikutnya

Tabel 5.6: Sambungan dari halaman sebelumnya

No	Aksi Pengguna	Reaksi yang diharapkan	Reaksi
9	<i>Top bar</i> pengguna <i>student</i>	<i>Top bar</i> terdapat <i>dropdown select assignments, timer assignment, dan icon profile</i>	Sesuai
10	<i>Side bar</i> pengguna <i>admin</i>	<i>Sidebar</i> berisikan <i>dashboard, settings, users, notifications, assignments, problems, submit, final submissions, all submissions, scoreboard, hall of fame, dan 24-hour log</i>	Sesuai
11	<i>Side bar</i> pengguna <i>head instructor</i>	<i>Sidebar</i> berisikan <i>dashboard, notifications, assignments, problems, submit, final submissions, all submissions, scoreboard, dan hall of fame</i>	Sesuai
12	<i>Side bar</i> pengguna <i>instructor</i>	<i>Sidebar</i> berisikan <i>dashboard, notifications, assignments, problems, submit, final submissions, all submissions, scoreboard, dan hall of fame</i>	Sesuai
13	<i>Side bar</i> pengguna <i>student</i>	<i>Sidebar</i> berisikan <i>dashboard, notifications, assignments, problems, submit, final submissions, all submissions, scoreboard, dan hall of fame</i>	Sesuai
14	<i>Dashboard</i> pengguna <i>admin</i> dan <i>head instructor</i>	<i>Dashboard</i> berisikan <i>calendar dengan jangka waktu assignments dan latest notification dengan icon untuk menambahkan notification</i>	Sesuai
15	<i>Dashboard</i> pengguna <i>instructor</i> dan <i>student</i>	<i>Dashboard</i> berisikan <i>calendar dengan jangka waktu assignments dan latest notification</i>	Sesuai
16	<i>Profile</i> pengguna <i>admin</i>	Berisikan <i>username, name, email, password, password again, dan user role</i>	Sesuai
17	<i>Profile</i> pengguna <i>head instructor, instructor, dan student</i>	Berisikan <i>username, name, email, password, dan password again</i>	Sesuai
18	Mengubah <i>password</i> akun pada <i>profile</i>	<i>Password</i> pada akun berubah dan dapat dilakukan oleh seluruh pengguna	Sesuai
19	Mengubah <i>role</i>	Penggantian <i>role</i> hanya dapat dilakukan oleh <i>admin</i>	Sesuai
20	Halaman <i>settings</i>	Halaman hanya dapat diakses oleh <i>admin</i> dan mengeluarkan <i>error message</i> apabila menggunakan <i>role</i> lain	Sesuai
21	Melakukan <i>sign out</i>	Pengguna kembali ke halaman <i>login</i> dan <i>session</i> pengguna dihancurkan	Sesuai
22	Mengubah konfigurasi pada halaman <i>settings</i>	Konfigurasi yang diubah disimpan menuju <i>database</i>	Sesuai
23	Menyalakan fitur <i>registration</i>	Halaman <i>login</i> terdapat tombol <i>register</i> dan pengguna dapat melakukan <i>register</i>	Sesuai

Bersambung ke halaman berikutnya

Tabel 5.6: Sambungan dari halaman sebelumnya

No	Aksi Pengguna	Reaksi yang diharapkan	Reaksi
24	Menyalakan fitur <i>lock student display name</i>	<i>Input</i> untuk mengubah <i>display name</i> telah <i>disable</i>	Sesuai
25	Halaman <i>users</i>	Halaman hanya dapat diakses oleh <i>admin</i>	Sesuai
26	Menambah <i>users</i> dan mengirimkan email kepada setiap pengguna	Pengguna dapat ditambahkan lebih dari satu dan email dikirimkan berisikan data pengguna	Sesuai
27	Mengunduh pengguna dalam format <i>excel</i>	Seluruh data pada halaman <i>users</i> terunduh pada perangkat pengguna dalam format <i>excel</i>	Sesuai
28	Mengganti data pengguna lain	Data pengguna lain hanya dapat diubah oleh <i>admin</i> . Data berubah dan pengguna dapat mengakses menggunakan data yang baru	Sesuai
29	Melihat seluruh pengumpulan setiap pengguna	Pengguna <i>student</i> hanya dapat melihat seluruh pengumpulannya. Pengguna <i>admin</i> , <i>head instructor</i> , dan <i>instructor</i> dapat melihat pengumpulan pengguna lain	Sesuai
30	Menghapus pengguna	Hanya dapat dilakukan oleh <i>admin</i> dan pengguna terhapus dari <i>database</i>	Sesuai
31	Menghapus <i>submission</i> pengguna	Seluruh <i>submission</i> dari pengguna tersebut terhapus	Sesuai
32	Halaman <i>notifications</i>	Dapat diakses oleh seluruh pengguna namun tombol <i>add</i> , <i>edit</i> dan <i>delete</i> hanya terdapat pada pengguna <i>admin</i> dan <i>head instructor</i>	Sesuai
33	Halaman <i>assignments</i>	Dapat diakses oleh seluruh pengguna dan dapat memilih <i>assignment</i>	Sesuai
34	Memilih <i>assignment</i>	Waktu pada <i>top bar</i> berganti menjadi penghitung mundur waktu <i>assignment</i> selesai. Apabila terdapat <i>extra time</i> maka muncul tampilan dan penghitung mundur <i>extra time</i>	Sesuai
35	Menambah <i>assignment</i>	Hanya dapat dilakukan oleh <i>admin</i> dan <i>head instructor</i> . <i>Assignment</i> bertambah pada direktori <i>assignments</i> dan data terextract. Dapat menambahkan <i>problem</i> dan melakukan konfigurasi sesuai dengan kebutuhan	Sesuai
36	Mengunduh <i>pdf</i> soal	<i>PDF</i> soal terunduh menuju perangkat pengguna	Sesuai
37	Mengunduh <i>test case</i> dan soal <i>PDF</i>	<i>Test case</i> dan soal <i>pdf</i> terunduh menuju perangkat pengguna dengan <i>extension zip</i>	Sesuai

Bersambung ke halaman berikutnya

Tabel 5.6: Sambungan dari halaman sebelumnya

No	Aksi Pengguna	Reaksi yang diharapkan	Reaksi
38	Mengunduh <i>final submission</i> berdasarkan pengguna	Seluruh <i>final submission</i> dari setiap pengguna yang mengumpulkan terunduh pada perangkat dengan <i>extension zip</i> . Seluruh <i>file</i> yang terunduh dikelompokan pada direktori berdasarkan pengguna	Sesuai
39	Mengunduh <i>final submission</i> berdasarkan <i>problem</i>	Seluruh <i>final submission</i> dari pengguna yang mengumpulkan terunduh pada perangkat dengan <i>extension zip</i> . Seluruh <i>file</i> yang terunduh dikelompokan pada direktori berdasarkan <i>problem</i>	Sesuai
40	Mendeteksi <i>similar code</i> pada <i>assignment</i>	Pengguna diarahkan menuju halaman <i>cheat detection</i> dengan <i>id</i> dari <i>assignment</i> yang dipilih. Dihasilkan tautan yang berisikan hasil dari pengecekan <i>similar code</i>	Sesuai
41	Melakukan <i>edit assignment</i>	<i>Assignment</i> berubah sesuai dengan konfigurasi pengguna	Sesuai
42	Menghapus <i>assignment</i>	<i>Assignment</i> terhapus dari halaman <i>assignments</i>	Sesuai
43	Halaman <i>problems</i>	Dapat diakses oleh seluruh pengguna dan pengguna dapat mengumpulkan jawaban melalui halaman tersebut. Namun, dekripsi <i>problem</i> hanya dapat dilakukan oleh <i>admin</i> dan <i>head instructor</i>	Sesuai
44	Menambahkan deskripsi <i>problem</i> menggunakan <i>markdown</i>	Deskripsi <i>problem</i> ditambahkan dengan tipe bahasa <i>markdown</i> . Deskripsi ditampilkan pada halaman <i>problem</i>	Sesuai
45	Menambahkan deskripsi <i>problem</i> menggunakan <i>HTML</i>	Deskripsi <i>problem</i> ditambahkan dengan tipe bahasa <i>HTML</i> . Deskripsi ditampilkan pada halaman <i>problem</i>	Sesuai
46	Menambahkan deskripsi <i>problem</i> menggunakan <i>plain HTML</i>	Deskripsi <i>problem</i> ditambahkan dengan tipe bahasa <i>plain HTML</i> . Deskripsi ditampilkan pada halaman <i>problem</i>	Sesuai
47	Halaman <i>submit</i>	Dapat diakses oleh seluruh pengguna. Pengguna dapat memilih <i>problem</i> dan bahasa pemrogramannya. <i>PDF</i> soal ditampilkan. Pengguna dapat langsung mencoba dan mengumpulkan kode melalui <i>editor</i> kode	Sesuai

Bersambung ke halaman berikutnya

Tabel 5.6: Sambungan dari halaman sebelumnya

No	Aksi Pengguna	Reaksi yang diharapkan	Reaksi
48	Halaman <i>final submissions</i>	Melihat seluruh <i>final submission</i> dari seluruh pengguna apabila pengguna berupa <i>admin</i> , <i>head instructor</i> , dan <i>instructor</i> . Melihat seluruh <i>final submission</i> dari pengguna apabila pengguna <i>student</i>	Sesuai
49	Mengunduh data <i>final submission</i> dalam format <i>excel</i>	Seluruh data dari halaman <i>final submission</i> terunduh kepada perangkat pengguna dalam format <i>excel</i>	Sesuai
50	Melihat status dari <i>submission</i> dan <i>test case problem</i>	Memperlihatkan status dan nilai dari kode. Setiap <i>test case</i> memperlihatkan apakah benar, salah, atau <i>runtime error</i>	Sesuai
51	Melakukan <i>filter</i> berdasarkan nama pengguna	Hanya terlihat <i>final submissions</i> dari pengguna tersebut	Sesuai
52	Melakukan <i>filter</i> berdasarkan <i>problem</i>	Hanya terlihat <i>final submissions</i> dari <i>problem</i> tersebut	Sesuai
53	Melakukan <i>rejudge</i>	Hanya dapat dilakukan oleh <i>admin</i> dan <i>head instructor</i> . <i>Submission</i> dilakukan penilaian ulang dan mengubah status sementara menjadi <i>pending</i>	Sesuai
54	Halaman <i>all submissions</i>	Melihat seluruh <i>submission</i> yang telah dikumpulkan dan dapat memilih <i>submission</i> yang ingin dijadikan hasil akhir. Melihat status dari <i>submission</i> pengguna. Melihat kode yang sudah dikumpulkan. Melihat <i>log</i> dari setiap <i>submission</i> apabila pengguna berupa <i>admin</i> , <i>head instructor</i> , dan <i>instructor</i>	Sesuai
55	Melakukan <i>filter</i> berdasarkan nama pengguna	Hanya terlihat <i>all submissions</i> dari pengguna tersebut	Sesuai
56	Melakukan <i>filter</i> berdasarkan <i>problem</i>	Hanya terlihat <i>all submissions</i> dari <i>problem</i> tersebut	Sesuai
57	Mengunduh data <i>all submission</i> dalam format <i>excel</i>	Seluruh data dari halaman <i>all submission</i> terunduh kepada perangkat pengguna dalam format <i>excel</i>	Sesuai
58	Halaman <i>hall of fame</i>	Melihat skor total dari seluruh <i>assignment</i> setiap pengguna dan mengurutkannya dari yang terbesar	Sesuai
59	Halaman <i>24-hour log</i>	Hanya dapat diakses oleh <i>admin</i> dan berisikan seluruh data perangkat dari pengguna. Pengguna yang melakukan <i>login</i> dari <i>ip address</i> yang berbeda terdeteksi dan ditampilkan	Sesuai

Bersambung ke halaman berikutnya

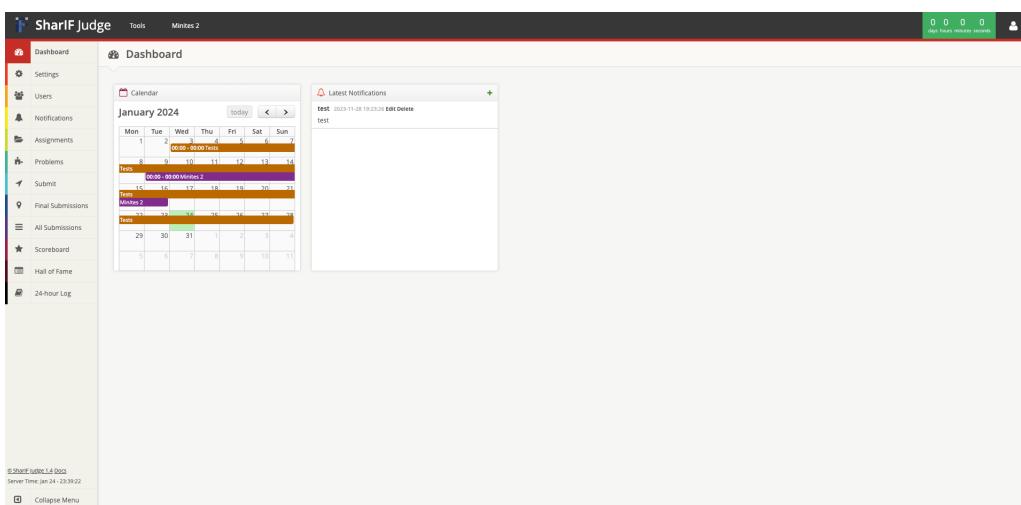
Tabel 5.6: Sambungan dari halaman sebelumnya

No	Aksi Pengguna	Reaksi yang diharapkan	Reaksi
60	Halaman <i>submission queue</i>	Dapat memberhentikan, melanjutkan, dan membersihkan antrian dari <i>submission</i> yang ada	Sesuai

Gambar 5.1 menunjukan hasil pengujian fungsional Halaman *login*

Gambar 5.1: Tampilan login *dashboard SharIF Judge*

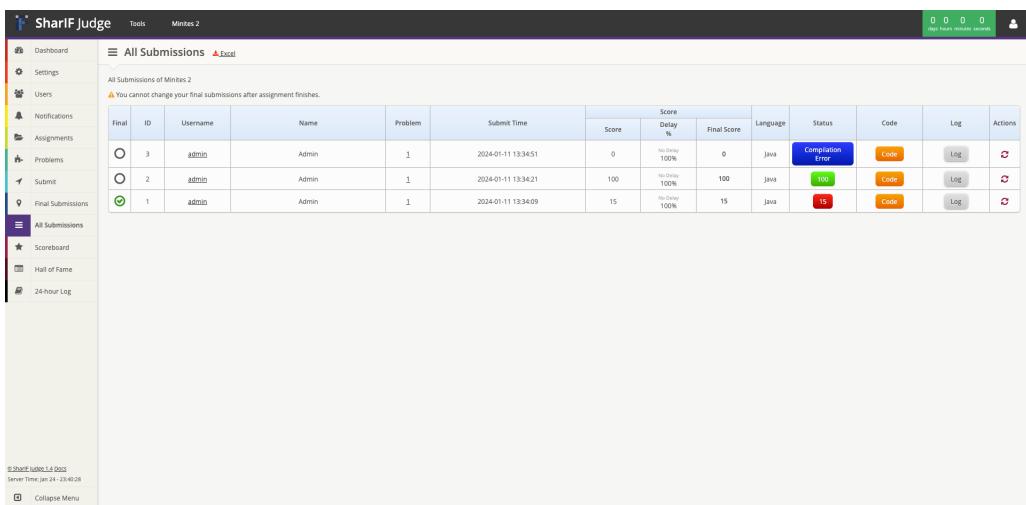
Gambar 5.2 menunjukan hasil pengujian fungsional Halaman *dashboard*

Gambar 5.2: Tampilan *dashboard SharIF Judge*

Gambar 5.3 menunjukan hasil pengujian fungsional Halaman *assignments*

Gambar 5.3: Tampilan *assignments* SharIF JudgeGambar 5.4 menunjukkan hasil pengujian fungsional Halaman *add assignments*

Gambar 5.4: Tampilan *add assignments* SharIF JudgeGambar 5.5 menunjukkan hasil pengujian fungsional Halaman *all submissions*



Gambar 5.5: Tampilan *all submissions SharIF Judge*

## 5.4 Pengujian Eksperimental

Pengujian dimulai dengan melakukan wawancara kepada admin Lab FTIS untuk menanyakan sistem operasi yang digunakan untuk menjalankan *SharIF Judge*. Wawancara tersebut menghasilkan jawaban dimana *SharIF Judge* dijalankan pada sistem operasi *Ubuntu* versi 16.04. Versi *Ubuntu* tersebut sudah tidak mendapat dukungan sehingga tidak digunakan untuk pembangunan perangkat lunak. Pengujian dilakukan untuk mengetahui apakah *SharIF Judge* dapat dijalankan pada beberapa versi *Ubuntu* yang berbeda.

Pengujian pertama kali dilakukan pada sistem operasi *Ubuntu* versi 22.04. Pengujian dilakukan dengan melakukan instalasi perangkat lunak *SharIF Judge* dan melakukan pengamanan terhadap *sandbox*. Setelah dilakukan instalasi terdapat persoalan dalam membangun *sandbox* dimana *sandbox* tidak dapat dibangun dan mengembalikan *error*. Oleh karena itu, dilakukan sebuah wawancara kepada admin Lab FTIS untuk menanyakan sistem operasi yang digunakan untuk menjalankan *SharIF Judge*. Wawancara tersebut menghasilkan jawaban bahwa sistem operasi yang digunakan untuk menjalankan *SharIF Judge* adalah *Ubuntu* versi 16.04.

Pengujian selanjutnya dilakukan dengan melakukan  *downgrade*  sistem operasi menuju  *Ubuntu*  versi 20.04. Perangkat lunak dapat berjalan dengan baik tanpa masalah pada  *Ubuntu*  versi 20.04. Pengujian kemudian dilanjutkan dengan melakukan pengujian eksperimental untuk menguji setiap versi sistem operasi  *Ubuntu*  terbaru. Pengujian eksperimental 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 eksperimental 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/easysandbox` dengan menjalankan kode berikut:

Kode 5.45: Pembangunan *sandbox* pada *Ubuntu 22.04*

```

1 $ cd tester/easysandbox
2 $ chmod +x runalltests.sh
3 $ chmod +x runtest.sh
4 $ make runtests

```

Kode 5.45 memindahkan pengguna menuju direktori `tester/easysandbox`. Selanjutnya akses file `runalltests` dan `runtest` diubah agar dapat dieksekusi. Terakhir file `runtests` dijalankan menggunakan sintaks `make`. Pembangunan *sandbox* menghasilkan pesan berupa `All tests passed!`. Namun pada sistem operasi ini, *sandbox* tidak dapat dibentuk dan menghasilkan *error message* seperti yang ditunjukkan pada Kode 5.46.

Kode 5.46: *Error message* pembangunan *sandbox* pada *Ubuntu 22.04*

```

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/
   test13 t/test14
2 Executing t/test01...passed!
3 Executing t/test02...passed!
4 Executing t/test03...passed!
5 Executing t/test04...passed!
6 Executing t/test05...passed!
7 Executing t/test06...passed!
8 Executing t/test07...passed!
9 Executing t/test08...passed!
10 Executing t/test09...la2
11 > Hello, C++ world
12 failed (output mismatch, expected [<<entering SECCOMP mode>>
13 Hello, C++ world], got [<<entering SECCOMP mode>>])
14 Executing t/test10...la2
15 > Hello from the constructor!
16 failed (output mismatch, expected [<<entering SECCOMP mode>>
17 Hello from the constructor!], got [<<entering SECCOMP mode>>])
18 Executing t/test11...passed!
19 Executing t/test12...la2,3
20 > Here we are in main()
21 > Hello from the destructor!
22 failed (output mismatch, expected [<<entering SECCOMP mode>>
23 Here we are in main()
24 Hello from the destructor!], got [<<entering SECCOMP mode>>])
25 Executing t/test13...passed!
26 Executing t/test14...la2
27 > 500500
28 failed (output mismatch, expected [<<entering SECCOMP mode>>
29 500500], got [<<entering SECCOMP mode>>])
30 4 test(s) failed
31 make: *** [Makefile:31: runtests] Error 1

```

Kode 5.46 merupakan keluaran dari pembangunan *sandbox* pada *Ubuntu 22.04*. Terdapat *error message* berupa keluaran tidak sesuai dengan yang seharusnya. Pembangunan dapat dikatakan tidak berhasil karena hanya sebagian *test case* yang berhasil dilewati dan tidak terdapat pesan berupa `All tests passed!`. *Error message* ini dihasilkan karena terdapat perbedaan versi pada *seccomp* atau *secure computing mode* sehingga tidak dapat dijalankan pada versi teratas pada *Ubuntu*.

#### 5.4.2 *Ubuntu 23.04*

Pengujian eksperimental kedua dilakukan pada sistem operasi *Ubuntu* dengan versi 23.04. 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:

Kode 5.47: Pembangunan *sandbox* pada *Ubuntu 23.04*

```

1 $ cd tester/easysandbox
2 $ chmod +x runalltests.sh
3 $ chmod +x runtest.sh
4 $ make runtests

```

Kode 5.47 memindahkan pengguna menuju direktori `tester/easysandbox`. Selanjutnya akses file `runalltests` dan `runtest` diubah agar dapat dieksekusi. Terakhir file `runtests` dijalankan menggunakan sintaks `make`. Pembangunan *sandbox* menghasilkan pesan berupa `All tests passed!`. Namun pada sistem operasi ini, *sandbox* tidak dapat dibentuk dan menghasilkan *error message* seperti yang ditunjukkan pada Kode 5.48.

Kode 5.48: *Error message* pembangunan *sandbox* pada *Ubuntu 23.04*

```

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/
   test13 t/test14

```

```

2| Executing t/test01...failed (exit code mismatch, expected 0, got 139)
3 Executing t/test02...1a2
4 > Hello, world
5 failed (output mismatch, expected [<<entering SECCOMP mode>>
6 Hello, world], got [<<entering SECCOMP mode>>])
7 Executing t/test03...failed (exit code mismatch, expected 137, got 139)
8 Executing t/test04...1a2
9 > 500500
10 failed (output mismatch, expected [<<entering SECCOMP mode>>
11 500500], got [<<entering SECCOMP mode>>])
12 Executing t/test05...1a2
13 > Hello, world
14 failed (output mismatch, expected [<<entering SECCOMP mode>>
15 Hello, world], got [<<entering SECCOMP mode>>])
16 Executing t/test06...failed (exit code mismatch, expected 137, got 139)
17 Executing t/test07...1a2
18 > 59
19 failed (output mismatch, expected [<<entering SECCOMP mode>>
20 59], got [<<entering SECCOMP mode>>])
21 Executing t/test08...failed (exit code mismatch, expected 0, got 139)
22 Executing t/test09...1a2
23 > Hello, C++ world
24 failed (output mismatch, expected [<<entering SECCOMP mode>>
25 Hello, C++ world], got [<<entering SECCOMP mode>>])
26 Executing t/test10...1a2
27 > Hello from the constructor!
28 failed (output mismatch, expected [<<entering SECCOMP mode>>
29 Hello from the constructor!], got [<<entering SECCOMP mode>>])
30 Executing t/test11...failed (exit code mismatch, expected 137, got 139)
31 Executing t/test12...1a2,3
32 > Here we are in main()
33 > Hello from the destructor!
34 failed (output mismatch, expected [<<entering SECCOMP mode>>
35 Here we are in main()
36 Hello from the destructor!], got [<<entering SECCOMP mode>>])
37 Executing t/test13...1a2
38 > Hello from the destructor!
39 failed (output mismatch, expected [<<entering SECCOMP mode>>
40 Hello from the destructor!], got [<<entering SECCOMP mode>>])
41 Executing t/test14...1a2
42 > 500500
43 failed (output mismatch, expected [<<entering SECCOMP mode>>
44 500500], got [<<entering SECCOMP mode>>])
45 14 test(s) failed
46 make: *** [Makefile:31: runtests] Error 1

```

Kode 5.48 merupakan keluaran dari pembangunan *sandbox* pada *Ubuntu* 23.04. Terdapat *error message* berupa keluaran tidak sesuai dengan yang seharusnya. Pembangunan dapat dikatakan tidak berhasil karena seluruh *test case* tidak berhasil dilewati dan tidak terdapat pesan berupa **All tests passed!**. *Error message* ini dihasilkan karena terdapat perbedaan versi pada *seccomp* atau *secure computing mode* sehingga tidak dapat dijalankan pada versi teratas pada *Ubuntu*.

#### 5.4.3 Ubuntu 23.10

Pengujian eksperimental ketiga dilakukan pada sistem operasi *Ubuntu* dengan versi 23.10. 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:

Kode 5.49: Pembangunan *sandbox* pada *Ubuntu* 23.10

```

1 $ cd tester/easy sandbox
2 $ chmod +x runalltests.sh
3 $ chmod +x runtest.sh
4 $ make runtests

```

Kode 5.49 memindahkan pengguna menuju direktori *tester/easy sandbox*. Selanjutnya akses file *runalltests* dan *runtest* diubah agar dapat dieksekusi. Terakhir file *runtests* dijalankan menggunakan sintaks *make*. Pembangunan *sandbox* menghasilkan pesan berupa **All tests passed!**. Namun pada sistem operasi ini, *sandbox* tidak dapat dibentuk dan menghasilkan *error message* seperti yang dijelaskan pada Kode 5.50.

Kode 5.50: *Error message* pembangunan *sandbox* pada *Ubuntu* 23.10

```

1 g++ -g -Wall -D_BSD_SOURCE -o t/test09 t/test09.cpp
2 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,
3          from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,
4          from /usr/include/c++/13/bits/requirements.h:31,
5          from /usr/include/c++/13/iostream:38,
6          from t/test09.cpp:3:
7 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]
8   196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"

```

```

9 |   ^~~~~~  

10 g++ -g -Wall -D_BSD_SOURCE -o t/test10 t/test10.cpp  

11 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,  

12     from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,  

13     from /usr/include/c++/13/bits/requirements.h:31,  

14     from /usr/include/c++/13/iostream:38,  

15     from t/test10.cpp:3:  

16 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]  

17 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"  

18 |   ^~~~~~  

19 g++ -g -Wall -D_BSD_SOURCE -o t/test11 t/test11.cpp  

20 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,  

21     from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,  

22     from /usr/include/c++/13/bits/requirements.h:31,  

23     from /usr/include/c++/13/iostream:38,  

24     from t/test11.cpp:4:  

25 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]  

26 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"  

27 |   ^~~~~~  

28 g++ -g -Wall -D_BSD_SOURCE -o t/test12 t/test12.cpp  

29 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,  

30     from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,  

31     from /usr/include/c++/13/bits/requirements.h:31,  

32     from /usr/include/c++/13/iostream:38,  

33     from t/test12.cpp:3:  

34 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]  

35 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"  

36 |   ^~~~~~  

37 gcc -std=c99 -g -Wall -D_BSD_SOURCE -o t/test13 t/test13.c  

38 In file included from /usr/include/x86_64-linux-gnu/bits/libc-header-start.h:33,  

39     from /usr/include/stdio.h:27,  

40     from t/test13.c:3:  

41 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]  

42 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"  

43 |   ^~~~~~  

44 g++ -g -Wall -D_BSD_SOURCE -o t/test14 t/test14.cpp  

45 In file included from /usr/include/x86_64-linux-gnu/c++/13/bits/os_defines.h:39,  

46     from /usr/include/x86_64-linux-gnu/c++/13/bits/c++config.h:679,  

47     from /usr/include/c++/13/bits/requirements.h:31,  

48     from /usr/include/c++/13/iostream:38,  

49     from t/test14.cpp:4:  

50 /usr/include/features.h:196:3: warning: #warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE" [-Wcpp]  

51 196 | # warning "_BSD_SOURCE and _SVID_SOURCE are deprecated, use _DEFAULT_SOURCE"  

52 |   ^~~~~~  

53 ./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/  

      test13 t/test14  

54 Executing t/test01...failed (exit code mismatch, expected 0, got 139)  

55 Executing t/test02...1a2  

56 > Hello, world  

57 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Hello, world], got [<<entering SECCOMP mode>>])  

58 Executing t/test03...failed (exit code mismatch, expected 137, got 139)  

59 Executing t/test04...1a2  

60 Executing t/test05...1a2  

61 > 500500  

62 failed (output mismatch, expected [<<entering SECCOMP mode>>  

500500], got [<<entering SECCOMP mode>>])  

63 Executing t/test06...1a2  

64 > Hello, world  

65 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Hello, world], got [<<entering SECCOMP mode>>])  

66 Executing t/test07...1a2  

67 > 59  

68 failed (output mismatch, expected [<<entering SECCOMP mode>>  

59], got [<<entering SECCOMP mode>>])  

69 Executing t/test08...failed (exit code mismatch, expected 0, got 139)  

70 Executing t/test09...1a2  

71 > Hello from the constructor!  

72 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Hello from the constructor!], got [<<entering SECCOMP mode>>])  

73 Executing t/test10...1a2  

74 Executing t/test11...1a2  

75 > Hello, C++ world  

76 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Hello, C++ world], got [<<entering SECCOMP mode>>])  

77 Executing t/test12...1a2  

78 Executing t/test13...1a2  

79 > Hello from the destructor!  

80 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Hello from the destructor!], got [<<entering SECCOMP mode>>])  

81 Executing t/test14...1a2  

82 Executing t/test15...1a2  

83 Executing t/test16...1a2  

84 > Here we are in main()  

85 > Hello from the destructor!  

86 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Here we are in main()  

Hello from the destructor!], got [<<entering SECCOMP mode>>])  

87 Executing t/test17...1a2  

88 Executing t/test18...1a2  

89 Executing t/test19...1a2  

90 > Hello from the destructor!  

91 failed (output mismatch, expected [<<entering SECCOMP mode>>  

Hello from the destructor!], got [<<entering SECCOMP mode>>])  

92 Executing t/test20...1a2  

93 Executing t/test21...1a2  

94 > 500500  

95 failed (output mismatch, expected [<<entering SECCOMP mode>>  

500500], got [<<entering SECCOMP mode>>])  

96 14 test(s) failed  

97 make: *** [Makefile:31: runtests] Error 1

```

Kode 5.50 merupakan keluaran dari pembangunan *sandbox* pada *Ubuntu 23.10*. Terdapat *error message* berupa *warning* dan *test case* yang tidak berhasil dijalankan. Pembangunan dapat dikatakan tidak berhasil karena seluruh *test case* tidak berhasil dilewati dan tidak terdapat pesan berupa **All tests passed!**. *Error message* ini dihasilkan karena terdapat perbedaan versi pada *seccomp* atau

*secure computing mode* sehingga tidak dapat dijalankan pada versi teratas pada *Ubuntu*.

## BAB 6

### KESIMPULAN DAN SARAN

Bab ini akan membahas kesimpulan yang telah diraih selama proses analisis, perancangan, implementasi, pengujian yang telah dibentuk, serta saran-saran untuk pengembangan yang dapat dipertimbangkan.

#### 6.1 Kesimpulan

Berdasarkan proses dan hasil analisis, perancangan, implementasi, dan pengujian perangkat lunak yang telah dibentuk, maka diperoleh kesimpulan sebagai berikut:

1. Perangkat lunak *SharIF Judge* telah berhasil dikonversi menuju *CodeIgniter 4* kecuali pengujian terhadap autentikasi RADIUS dan LDAP. Pengujian tidak dilakukan karena terdapat keterbatasan *server*. Seluruh fitur yang terdapat pada *SharIF Judge CodeIgniter 4* berjalan sesuai dengan *SharIF Judge* pada *CodeIgniter 3* dengan beberapa perubahan sintaks.

#### 6.2 Saran

Berdasarkan hasil analisis, perancangan, dan implementasi yang telah dilakukan, berikut merupakan saran-saran untuk pengembangan selanjutnya:

1. Membentuk *Sandbox* untuk melakukan *compile* bahasa pemrograman *C*, *CPP*, dan *Python* yang terintegrasi pada *Ubuntu* versi 22.04, 23.04, dan 23.10.
2. Melakukan dokumentasi terhadap fitur-fitur yang akan dibentuk karena dokumentasi terbatas hanya pada pembentukan awal *SharIF Judge*.
3. Melakukan pengujian autentikasi *RADIUS* dan *LDAP*.



## **DAFTAR REFERENSI**

- [1] Version 3.1.13 (2022) *CodeIgniter User Guide*. CodeIgniter Foundation. Richmond,Canada.
- [2] Version 4.3 (2023) *CodeIgniter User Guide*. CodeIgniter Foundation. Richmond,Canada.
- [3] 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.
- [4] Kurnia, A., Lim, A., dan Cheang, B. (2001) Online judge. *Computers & Education*, **18**, 299–315.
- [5]



## LAMPIRAN A

### KODE PROGRAM

Kode perangkat lunak hanya akan dilampirkan beberapa *file* dengan perubahan terbanyak. Lampiran kode lainnya tidak ditampilkan karena dapat memperpanjang dan memperberat dokumen tugas akhir ini. Lampiran lainnya dapat dilihat pada <https://github.com/filipussetio/SharIF-JudgeCI4>.

#### A.1 *Controller*

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: Filipus Setio Nugroho <filipussetio@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 $parsesdown;
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         );
63     }
64 }
```

```

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         $item['finished'] = ($delay > $extra_time);
77     }
78
79     return view('pages/assignments', $data);
80
81 }
82
83 /**
84 * Used by ajax request (for select assignment from top bar)
85 */
86 public function select()
87 {
88     if ( ! $this->request->isAJAX() )
89         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
90
91     $this->validation->setRule('assignment_select', 'Assignment', 'required|integer|greater_than[0]');
92
93     if ($this->validation->withRequest($this->request)->run())
94     {
95         $this->user->select_assignment($this->request->getPost('assignment_select'));
96         $this->assignment = $this->assignment_model->assignment_info($this->request->getPost('assignment_select'));
97         $json_result = array(
98             'done' => 1,
99             'finish_time' => $this->assignment['finish_time'],
100            'extra_time' => $this->assignment['extra_time'],
101        );
102    }
103    else{
104        $json_result = array('done' => 0, 'message' => 'Input_Error');
105    }
106    $this->response->setHeader('Content-Type:', 'application/json; charset=utf-8');
107    echo json_encode($json_result);
108 }
109
110 /**
111 * Download pdf file of an assignment (or problem) to browser
112 */
113 public function pdf($assignment_id, $problem_id = NULL, $no_download = FALSE)
114 {
115     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
116     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
117     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
118
119     // Find pdf file
120     if ($problem_id === NULL || $problem_id === "null")
121         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/*.pdf";
122     else
123         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/p{$problem_id}/*.pdf";
124     $pdf_files = glob($pattern);
125     if ( ! $pdf_files )
126         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("File_not_found");
127     elseif ( !$this->assignment_model->assignment_info($assignment_id)['open'] )
128         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_been_closed.');
129     elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)['participants'], $this->user->username) )
130         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_are_not_registered_for_submitting.');
131     elseif ( shj_now() > $finishtime + $extratime )
132         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_finished.');
133     elseif ( shj_now() < $starttime )
134         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_not_started.');
135
136     $filename = shj_basename($pdf_files[0]);
137     // Download the file to browser
138     if($no_download === FALSE){
139         return $this->response->download($filename, file_get_contents($pdf_files[0]));
140     }
141     else{
142         $content = file_get_contents($pdf_files[0]);
143         header('Content-Type: application/pdf');
144         die($content);
145     }
146 }
147
148
149
150
151
152
153
154
155
156
157
158 }

```

```

159
160
161
162 // -----
163
164
165
166 /**
167 * Compressing and downloading test data and descriptions of an assignment to the browser
168 */
169 public function downloadtestsdesc($assignment_id = FALSE)
170 {
171     if ($assignment_id === FALSE)
172         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
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
216     $pdf_files = glob("$root_path/*.pdf");
217     if ($pdf_files)
218     {
219         $path = $pdf_files[0];
220         $this->zip->addFile($path, shj_basename($path));
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_{$item['assignment']}/p{$item['problem']}/{$item['username']}/{$item['file_name']}.".
259         .filetype_to_extension($item['file_type']);
260     if ( ! file_exists($file_path))
261     {
262     continue;
263     $file = file_get_contents($file_path);
264
265     if ($type === 'by_user')
266     {
267     $this->zip->addFile($file_path,"{$item['username']}/p{$item['problem']}.".filetype_to_extension($item['file_type']
268     ));
269     elseif ($type === 'by_problem')
270     {
271     $this->zip->addFile($file_path,"problem_{$item['problem']}/{$item['username']}.".filetype_to_extension($item['file_type']));
272     }
273     }
274   }
275 }
276
277 // -----
278
279
280 /**
281 * Delete assignment
282 */
283 public function delete($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
315   return view('pages/admin/delete_assignment', $data);
316 }
317
318
319
320 // -----
321
322
323 /**
324 * This method gets inputs from user for adding/editing assignment
325 */
326 public function add()
327 {
328
329   if ($this->user->level <= 1) // permission denied
330     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
331
332
333   if ($this->request->is('post')){
334     if ($this->_add()) // add/edit assignment
335     {
336       //if ( ! $this->edit) // if adding assignment (not editing)
337       //{
338       //  goto Assignments page
339       //  return $this->index();
340       //}
341     }
342   }
343
344   $data = array(
345     'all_assignments' => $this->assignment_model->all_assignments(),
346     'messages' => $this->messages,
347     'edit' => $this->edit,
348     'default_late_rule' => $this->settings_model->get_setting('default_late_rule'),
349     'selected' => 'assignments',
350     'icon' => 'fa-edit',
351     'title' => 'Assignments',
352     'user' => $this->user,
353     'finish_time' => $this->user->selected_assignment['finish_time'],
354   );

```

```

355     'extra_time' => $this->user->selected_assignment['extra_time'],
356     'twig' => $this->twig,
357     'validationError' => $this->validation,
358 );
359
360 if ($this->edit)
361 {
362     $data['edit_assignment'] = $this->assignment_model->assignment_info($this->edit_assignment);
363     if ($data['edit_assignment']['id'] === 0)
364         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
365     $data['problems'] = $this->assignment_model->all_problems($this->edit_assignment);
366 }
367 else
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     }

```

```

454 // Preparing variables
455
456 if ($this->edit)
457     $the_id = $this->edit_assignment;
458 else
459     $the_id = $this->assignment_model->new_assignment_id();
460
461 $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
462 $assignment_dir = "$assignments_root/assignment_{$the_id}";
463
464 // Adding/Editing assignment in database
465
466 if ( !$this->assignment_model->add_assignment($the_id, $this->edit))
467 {
468     $this->messages[] = array(
469         'type' => 'error',
470         'text' => 'Error'.($this->edit?'updating':'adding').'_assignment.'
471     );
472     return FALSE;
473 }
474
475 $this->messages[] = array(
476     'type' => 'success',
477     'text' => 'Assignment'.($this->edit?'updated':'added').'_successfully.'
478 );
479
480 // Create assignment directory
481 if ( !file_exists($assignment_dir) ){
482     mkdir($assignment_dir, 0700, true);
483 }
484
485
486 // Upload Tests (zip file)
487 $files = glob($assignments_root.'/*.zip');
488 foreach($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');
495 if ( $_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 }
501 elseif ( $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 }
507 else{
508     $zip_uploaded->move($assignments_root);
509     $this->messages[] = array(
510         'type' => 'success',
511         'text' => "Tests(zip file) uploaded successfully."
512     );
513 }
514
515
516 // Upload PDF File of Assignment
517 $old_pdf_files = glob("$assignment_dir/*.pdf");
518 $pdf_uploaded = $this->request->getFile("pdf");
519 if ( $_FILES['pdf']['error'] === UPLOAD_ERR_NO_FILE ){
520     $this->messages[] = array(
521         'type' => 'notice',
522         'text' => "Notice: You did not upload any pdf file for assignment. If needed, upload by editing assignment."
523     );
524 }
525 elseif ( $pdf_uploaded->getExtension() != "pdf" ){
526     $this->messages[] = array(
527         'type' => 'error',
528         'text' => "Error: Error uploading pdf file of assignment: The filetype you are attempting to upload is not allowed."
529     );
530 }
531 else{
532     $pdf_uploaded->move("$assignment_dir");
533     {
534         foreach($old_pdf_files as $old_name){
535             unlink($old_name);
536         }
537         $this->messages[] = array(
538             'type' => 'success',
539             'text' => 'PDF file uploaded successfully.'
540         );
541     }
542 }
543
544 // Extract Tests (zip file)
545 if ($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_rf_*/tester.cpp;\"rm_rf_*/tester.executable;" .
564             ."\"rm_rf_*/desc.html;\"rm_rf_*/desc.md;\"rm_rf_*/*.pdf;\"");
565         if (glob("$tmp_dir/*.pdf"))
566             shell_exec("cd_$assignment_dir;\"rm_rf_*.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", $html);
599     }
600 }
601
602 return TRUE;
603
604
605
606 // -----
607
608 public function edit($assignment_id)
609 {
610
611     if ($this->user->level <= 1) // permission denied
612         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
613
614     $this->edit_assignment = $assignment_id;
615     $this->edit = TRUE;
616
617     // redirect to add function
618     return $this->add();
619 }
620
621
622
623
624 // -----
625
626
627
628 /**
629 * Check PDF File Availability
630 */
631 public function pdfCheck($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') )

```

```

649     $response = json_encode(array('status'=>FALSE));
650     elseif ( shj_now() > $finishtime + $extratime)
651         $response = json_encode(array('status'=>FALSE));
652     elseif ( shj_now() < $starttime)
653         $response = json_encode(array('status'=>FALSE));
654     else
655         $response = json_encode(array('status'=>TRUE));
656
657     echo $response;
658 }
659 }
660 }
```

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: Filipus Setio Nugroho <filipussetio@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
29     private $filter_user;
30     private $filter_problem;
31     private $page_number;
32
33     protected $assignment_model;
34     protected $submit_model;
35     protected $user;
36     protected $validation;
37     protected $settings_model;
38     protected $user_model;
39     protected $scoreboard_model;
40     protected $uri;
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     $header=array('Final', 'Submit_ID', 'Username', 'Name', 'Problem', 'Submit_Time', 'Score', 'Delay_(HH:MM)', 'Coefficient', 'Final_Score', 'Language', 'Status');
121     if ($view === 'final'){
122         array_unshift($header, "#2");
123         array_unshift($header, "#1");
124     }
125 }
126
127 // Add header to document
128 $sheet->fromArray($header, null, 'A6', true);
129 $highest_column = $sheet->getHighestColumn();
130
131 // Set custom style for header
132 $sheet->getStyle('A6:'.$highest_column.'6')->applyFromArray(
133     array(
134         'fill' => array(
135             'fillType' => Fill::FILL_SOLID,
136             'color' => array('rgb' => '173C45')
137         ),
138         'font' => array(
139             'bold' => true,
140             'color' => array('rgb' => 'FFFFFF'),
141             //'size' => 14
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*$item['coefficient']/100);
175
176
177     if ($this->user->level === 0)
178         $row = array(
179             $checked,
180             $item['problem'].'.'.$pi['name'],
181             $item['time'],
182             $pre_score,
183             ($delay<=0?'No_Delay':time_hhmm($delay)),
184             $item['coefficient'],
185             $final_score,
186             filetype_to_language($item['file_type']),
187             $item['status'],
188         );
189     else {
190         $row = array(
191             $checked,
192             $item['submit_id'],
193             $item['username'],
194             $names[$item['username']],
195             $item['problem'].'.'.$pi['name'],
196             $item['time'],
197             $pre_score,
198             ($delay<=0?'No_Delay':time_hhmm($delay)),
199             $item['coefficient'],
200             $final_score,
201             filetype_to_language($item['file_type']),
202             $item['status'],
203         );
204     if ($view === 'final'){
205         array_unshift($row,$j);

```

```

206             array_unshift($row,$i);
207         }
208     }
209     array_push($rows, $row);
210 }
211
212 // Add rows to document
213 $sheet->fromArray($rows, null, 'A7', true);
214 // Add alternative colors to rows
215 for ($i=7; $i<count($rows)+7; $i++){
216     $sheet->getStyle('A'.$i.':' . $highest_column.$i)->applyFromArray(
217         array(
218             'fill' => array(
219                 'fillType' => Fill::FILL_SOLID,
220                 'color' => array('rgb' => (($i%2)?'F0F0F0':'FAFAFA'))
221             )
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 public function all_excel()
275 {
276     $this->download_excel('all');
277 }
278
279
280
281
282 // -----
283
284
285
286
287
288 public function the_final()
289 {
290
291     if ( ! is_numeric($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     )

```

```

303     'full_ul_class' => 'shj_pagination',
304     'cur_li_class' => 'current_page'
305   );
306
307   if ($config['per_page'] == 0)
308     $config['per_page'] = $config['total_rows'];
309   $shj_pagination = new Shj_pagination($config);
310
311   $submissions = $this->submit_model->get_final_submissions($this->user->selected_assignment['id'], $this->user->level,
312   $this->user->username, $this->page_number, $this->filter_user, $this->filter_problem);
313
314   $names = $this->user_model->get_names();
315
316   foreach ($submissions as &$item)
317   {
318     $item['name'] = $names[$item['username']];
319     $item['fullmark'] = ($item['pre_score'] == 10000);
320     $item['pre_score'] = ceil($item['pre_score']*$this->problems[$item['problem']]['score']/10000);
321     $item['delay'] = strtotime($item['time'])-strtotime($this->user->selected_assignment['finish_time']);
322     $item['language'] = filetype_to_language($item['file_type']);
323     if ($item['coefficient'] === 'error')
324       $item['final_score'] = 0;
325     else
326       $item['final_score'] = ceil($item['pre_score']*$item['coefficient']/100);
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   );

```

```

397     'all_assignments' => $this->assignment_model->all_assignments(),
398     'problems' => $this->problems,
399     'submissions' => $submissions,
400     'excel_link' => site_url('submissions/all_excel'.($this->filter_user?'/user/'. $this->filter_user:'')).($this->
401         filter_problem?'/problem/'.$this->filter_problem: ''),
402     'filter_user' => $this->filter_user,
403     'filter_problem' => $this->filter_problem,
404     'pagination' => $shj_pagination->create_links(),
405     'selected' => "all_submissions",
406     'user' => $this->user,
407     'finish_time' => $this->user->selected_assignment['finish_time'],
408     'extra_time' => $this->user->selected_assignment['extra_time'],
409   );
410 
411   return view('pages/submissions', $data);
412 }
413 
414 
415 /**
416  * Used by ajax request (for selecting final submission)
417 */
418 public function select()
419 {
420   if ( ! $this->request->isAJAX() )
421     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
422 
423   // Students cannot change their final submission after finish_time + extra_time
424   if ($this->user->level === 0)
425     if ( shj_now() > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['
426       extra_time'])
427     {
428       $json_result = array(
429         'done' => 0,
430         'message' => 'This_assignment_is_finished._You_cannot_change_your_final_submissions.'
431       );
432       $this->response->setHeader('Content-Type:_application/json;', 'charset=utf-8');
433       echo json_encode($json_result);
434       return;
435     }
436 
437   $this->validation->setRule('submit_id', 'Submit_ID', 'integer|greater_than[0]');
438   $this->validation->setRule('problem', 'problem', 'integer|greater_than[0]');
439   $this->validation->setRule('username', 'Username', 'required|min_length[3]|max_length[20]|alpha_numeric');
440 
441   if ($this->validation->withRequest($this->request)->run())
442   {
443     $username = $this->request->getPost('username');
444     if ($this->user->level === 0)
445       $username = $this->user->username;
446 
447     $res = $this->submit_model->set_final_submission(
448       $username,
449       $this->user->selected_assignment['id'],
450       $this->request->getPost('problem'),
451       $this->request->getPost('submit_id')
452     );
453 
454     if ($res) {
455       // each time a user changes final submission, we should update scoreboard of that assignment
456       $this->scoreboard_model = new ScoreboardModel();
457       $this->scoreboard_model->update_scoreboard($this->user->selected_assignment['id']);
458       $json_result = array('done' => 1);
459     }
460     else
461       $json_result = array('done' => 0, 'message' => 'Selecting_Final_Submission_Failed');
462   }
463   else
464     $json_result = array('done' => 0, 'message' => 'Input_Error');
465 
466   $this->response->setHeader('Content-Type:_application/json;', 'charset=utf-8');
467   echo json_encode($json_result);
468 }
469 
470 /**
471  * For "view code" or "view result" or "view log"
472 */
473 public function view_code()
474 {
475   if ( ! $this->request->isAJAX() )
476     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
477   $this->validation->setRule('type', 'type', '_check_type');
478   $this->validation->setRule('username', 'username', 'required|min_length[3]|max_length[20]|alpha_numeric');
479   $this->validation->setRule('assignment', 'assignment', 'integer|greater_than[0]');
480   $this->validation->setRule('problem', 'problem', 'integer|greater_than[0]');
481   $this->validation->setRule('submit_id', 'submit_id', 'integer|greater_than[0]');
482 
483   if($this->validation->withRequest($this->request)->run())
484   {
485     $submission = $this->submit_model->get_submission(
486       $this->request->getPost('username'),
487       $this->request->getPost('assignment'),
488     )
489   }
490 }
491 
```

```

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 }
538
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']}.".filetype_to_extension($submission['file_type']);
568
569     return $this->response->download(
570         "{$submission['file_name']}.".filetype_to_extension($submission['file_type']),
571         file_get_contents($file_path)
572     );
573 }
574
575
576
577 }

```

Kode A.4: Submit.php

```

1 <?php
2 /**
3 * Sharif Judge online judge
4 * @file Submit.php
5 * @author: Filipus Setio Nugroho <filipussetio@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 = $coefficient;
57|
58|     }
59|
60|     // -----
61|
62|     public function _language_to_type($language)
63|     {
64|         $language = strtolower ($language);
65|         switch ($language) {
66|             case 'c': return 'c';
67|             case 'c++': return 'cpp';
68|             case 'python_2': return 'py2';
69|             case 'python_3': return 'py3';
70|             case 'java': return 'java';
71|             case 'zip': return 'zip';
72|             case 'pdf': return 'pdf';
73|             case 'txt': return 'txt';
74|             default: return FALSE;
75|         }
76|     }
77|
78|     // -----
79|
80|
81|     public function _language_to_ext($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);

```

```

106     case 'cpp': return ($extension==='cpp'?TRUE:FALSE);
107     case 'py2': return ($extension==='py'?TRUE:FALSE);
108     case 'py3': return ($extension==='py'?TRUE:FALSE);
109     case 'java': return ($extension==='java'?TRUE:FALSE);
110     case 'zip': return ($extension==='zip'?TRUE:FALSE);
111     case 'pdf': return ($extension==='pdf'?TRUE:FALSE);
112     case 'txt': return ($extension==='txt'?TRUE:FALSE);
113   }
114 }
115
116 /**
117 */
118
119 public function index()
120 {
121   $this->validation->setRule('problem', 'problem', 'required|integer|greater_than[0]', array('greater_than' => 'Select_a_%s.'));
122   $this->validation->setRule('language', 'language', 'required|check_language', array('_check_language' => 'Select_a_valid_%s.'));
123   if($this->request->is('post')){
124     if ($this->validation->withRequest($this->request)->run())
125     {
126       if ($this->upload())
127         return redirect()->to('submissions/all');
128       else
129         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('ErrorUploadingFile:'.$this->validation->listErrors());
130     }
131   }
132
133   $this->data = array(
134     'all_assignments' => $this->assignment_model->all_assignments(),
135     'problems' => $this->problems,
136     'in_queue' => FALSE,
137     'coefficient' => $this->coefficient,
138     'upload_state' => '',
139     'problems_js' => '',
140     'error' => '',
141     'selected' => 'submit',
142     'user' => $this->user,
143     'finish_time' => $this->user->selected_assignment['finish_time'],
144     'extra_time' => $this->user->selected_assignment['extra_time'],
145     'validationError' => $this->validation
146   );
147   foreach ($this->problems as $problem)
148   {
149     $languages = explode(',', $problem['allowed_languages']);
150     $items='';
151     foreach ($languages as $language)
152     {
153       $items = $items.'"'.trim($language).'"';
154     }
155     $items = substr($items,0,strlen($items)-1);
156     $this->data['problems_js'] .= "shj.p[$problem['id']] = [".$items."];";
157   }
158   if ($this->user->selected_assignment['id'] == 0)
159     $this->data['error']= 'Please_select_an_assignment_first.';
160   elseif ($this->user->level == 0 && ! $this->user->selected_assignment['open'])
161     // if assignment is closed, non-student users (admin, instructors) still can submit
162     $this->data['error'] = 'Selected_assignment_is_closed.';
163   elseif (shj_now() < strtotime($this->user->selected_assignment['start_time']))
164     $this->data['error'] = 'Selected_assignment_has_not_started.';
165   elseif (shj_now() > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time']) // deadline = finish_time + extra_time
166     $this->data['error'] = 'Selected_assignment_has_finished.';
167   elseif ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->username))
168     $this->data['error'] = 'You_are_not_registered_for_submitting.';
169   else
170     $this->data['error'] = 'none';
171
172   return view('pages/submit', $this->data);
173 }
174
175 /**
176 */
177 /**
178 */
179 /**
180 * Saves submitted code and adds it to queue for judging
181 */
182
183 private function _upload()
184 {
185   $now = shj_now();
186   foreach($this->problems as $item)
187   {
188     if ($item['id'] == $this->request->getPost('problem'))
189     {
190       $this->problem = $item;
191       break;
192     }
193   }
194   $this->filetype = $this->language_to_type(strtolower(trim($this->request->getPost('language'))));
195   $this->ext = substr(strrchr($_FILES['userfile']['name'],'.'),1); // uploaded file extension
196   $this->file_name = basename($_FILES['userfile']['name'], ".$this->ext"); // uploaded file name without extension
197   if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $this->problem['id']) )
198     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_have_already_submitted_for_this_problem._Your_last_submission_is_still_in_queue.');
199   if ($this->user->level==0 && !$this->user->selected_assignment['open'])
200     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_been_closed.');

```

```

199     if ($now < strtotime($this->user->selected_assignment['start_time']))
200         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_not_started.');
201     if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time'])
202         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_finished.');
203     if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->username) )
204         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_are_not_registered_for_submitting.');
205     $filetypes = explode( ' ', $this->problem['allowed_languages']);
206     foreach ($filetypes as &$filetype)
207     {
208         $filetype = $this->language_to_type(strtolower(trim($filetype)));
209     }
210     if ($_FILES['userfile']['error'] == 4)
211         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('No_file_chosen.');
212     if ( ! in_array($this->filetype, $filetypes))
213         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('This_file_type_is_not_allowed_for_this_problem.');
214     if ( ! $this->match($this->filetype, $this->ext) )
215         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('This_file_type_does_not_match_your_selected_language.');
216     if ( ! preg_match('/^([a-zA-Z0-9\-\_])+$/i', $this->file_name) )
217         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Invalid_characters_in_file_name.');
218
219     $user_dir = rtrim($this->assignment_root, '/').$this->user->selected_assignment['id'].'/p'.$this->problem['id'].'/.'.$this->user->username;
220     if ( !file_exists($user_dir) )
221         mkdir($user_dir, 0700);
222
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_you_are_attempting_to_upload_is_larger_than_the_permitted_size.']);
227     if ($this->validation->withRequest($this->request)->run())
228     {
229         $results = $this->request->getFile('userfile');
230         $results->move($user_dir,$config['file_name']);
231         $result = preg_replace("/\[^.]+\$/i", "", $results->getName());
232         $this->submit_model = new SubmitModel();
233
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 * Load code from editor file
266 */
267
268 public function load($problem_id){
269     $user_dir = rtrim($this->assignment_root, '/').$this->user->selected_assignment['id'].'/p'.$problem_id.'/';
270     $user_dir .= $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 = $user_dir.'/.EDITOR_OUT_NAME.'.EDITOR_FILE_EXT;
274
275     if(!write_file($input_path, '_')){}
276     if(!write_file($output_path, '_')){}
277     if (!file_exists($file_path)){
278         $response = json_encode(array('content'=>'', 'message'=>'No_saved_file'));
279     }
280     else{
281         $file_content = file_get_contents($file_path);
282         if ($file_content === FALSE){
283             $response = json_encode(array('content'=>'', 'message'=>'Unable_to_load'));
284         }
285         else{
286             addslashes($file_content);
287             $response = json_encode(array('content'=>$file_content, 'message'=>'Loaded'));
288         }
289     }
290     echo $response;
291 }
292

```

```

293 // -----
294 /**
295 * Save code to editor file and submit/execute if needed
296 */
297 public function save($type = FALSE){
298     $data = $_POST['code_editor'];
299     $problem_id = $_POST['problem_id'];
300     $language = $_POST['language'];
301
302     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'. $this->user->selected_assignment['id']. '/p' . $problem_id '/';
303     $this->user->username;
304     if (!file_exists($user_dir)){
305         mkdir($user_dir, 0700);
306     }
307     $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     } else{
314         $response = json_encode(array('status'=>TRUE, 'message'=>'Saved'));
315         if ($type === FALSE){
316             echo $response;
317         } else{
318             $now = shj_now();
319             if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $problem_id)){
320                 $response = json_encode(array('status'=>FALSE, 'message'=>'You_have_already_submitted_for_this_problem_.Your_
321                 last_submission_is_still_in_queue.'));
322                 echo $response;
323             } else if ($this->user->level==0 && !$this->user->selected_assignment['open']){
324                 $response = json_encode(array('status'=>FALSE, 'message'=>'Selected_assignment_has_been_closed.'));
325                 echo $response;
326             } else if ($now < strtotime($this->user->selected_assignment['start_time'])){
327                 $response = json_encode(array('status'=>FALSE, 'message'=>'Selected_assignment_has_not_started.'));
328                 echo $response;
329             } else if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['
330                 extra_time']){
331                 $response = json_encode(array('status'=>FALSE, 'message'=>'Selected_assignment_has_finished.'));
332                 echo $response;
333             } else if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'], $this->user->
334                 username)){
335                 $response = json_encode(array('status'=>FALSE, 'message'=>'You_are_not_registered_for_submitting.'));
336                 echo $response;
337             } else{
338                 if($type === 'submit'){
339                     $this->_submit($data, $problem_id, $language, $user_dir);
340                 } else if($type === 'execute'){
341                     $editor_input = $_POST['editor_input'];
342                     if (!write_file($input_path, $editor_input)){
343                         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_write_input_file'));
344                         echo $response;
345                     } else{
346                         $this->_execute($data, $problem_id, $language, $user_dir);
347                     }
348                 }
349             }
350         }
351     }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 // -----
364 /**
365 * Add code to queue for judging
366 */
367 private function _submit($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 = $item;
379             break;
380         }
381
382         if (!write_file($file_path, $data)){
383             $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_submit'));
384         } else{
385             $this->submit_model = new SubmitModel();
386         }
387     }

```

```

388     $submit_info = array(
389         'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
390         'username' => $this->user->username,
391         'assignment' => $this->user->selected_assignment['id'],
392         'problem' => $problem_id,
393         'file_name' => $file_name,
394         'main_file_name' => $file_name,
395         'file_type' => $file_type,
396         'coefficient' => $this->coefficient,
397         'pre_score' => 0,
398         'time' => shj_now_str(),
399     );
400     if ($this->problem['is_upload_only'] == 0)
401     {
402         $this->queue_model->add_to_queue($submit_info);
403         process_the_queue();
404     }
405     else
406     {
407         $this->submit_model->add_upload_only($submit_info);
408     }
409
410     $response = json_encode(array('status'=>TRUE, 'message'=>"Submitted"));
411 }
412 echo $response;
413 }

414

415 // -----
416 /**
417 * Add code to queue for execution only
418 */
419 private function _execute($data, $problem_id, $language, $user_dir){
420     $file_type = $this->language_to_type(strtolower(trim($language)));
421     $file_ext = $this->language_to_ext(strtolower(trim($language)));
422     $file_name = EDITOR_FILE_NAME;
423     $file_fname = $file_name.'.'.$EDITOR_SUBMIT_ID;
424     $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
425     $output_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
426
427     if (!write_file($file_path, $data)){
428         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_execute', 'debug'=>$file_path));
429     }
430     else{
431         $submit_info = array(
432             'submit_id' => EDITOR_SUBMIT_ID,
433             'username' => $this->user->username,
434             'assignment' => $this->user->selected_assignment['id'],
435             'problem' => $problem_id,
436             'file_name' => $file_fname,
437             'main_file_name' => $file_name,
438             'file_type' => $file_type,
439             'coefficient' => $this->coefficient,
440             'pre_score' => 0,
441             'time' => shj_now_str(),
442         );
443
444         if($this->queue_model->add_to_queue_exec($submit_info)){
445             if (!write_file($output_path, 'Queueing...')){
446                 $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_write_output_file'));
447             }
448             else{
449                 process_the_queue();
450                 $response = json_encode(array('status'=>TRUE, 'message'=>'Executing'));
451             }
452         }
453         else{
454             $response = json_encode(array('status'=>FALSE, 'message'=>'Still_in_queue'));
455         }
456     }
457 }
458 echo $response;
459 }

460

461 // -----
462 /**
463 * Load output file as execution result
464 */
465 public function get_output($problem_id){
466     $user_dir = rtrim($this->assignment_root, '/').$this->user->selected_assignment['id'].'/p/'.$problem_id.'/';
467     $this->user->username;
468     $file_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
469
470     if (!file_exists($file_path)){
471         $response = json_encode(array('status'=>FALSE, 'content'=>''));
472     }
473     else{
474
475         $file_content = file_get_contents($file_path);
476         if ($file_content === FALSE){
477             $response = json_encode(array('status'=>FALSE, 'content'=>''));
478         }
479         else{
480             $complete_status = strpos($file_content, 'Total_Execution_Time');
481             if($complete_status === FALSE){
482                 $response = json_encode(array('status'=>FALSE, 'content'=>$file_content));
483             }
484         }
485     }
}

```

```

486         else{
487             $response = json_encode(array('status'=>TRUE, 'content'=>$file_content));
488         }
489     }
490     echo $response;
491 }
492 }
493 }

```

## A.2 Model

Kode A.5: AssignmentModel.php

```

1 <?php
2 /**
3  * ShariF Judge online judge
4  * @file Assignment_model.php
5  * @author: Filipus Setio Nugroho <filipussetio@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     else
77         $this->db->table('assignments')->insert($assignment);
78
79     /* **** Adding problems to "problems" table **** */

```

```

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 {
96     $uo = array();
97 }
98 for ($i=1; $i<=$this->request->getPost('number_of_problems'); $i++)
99 {
100     $items = explode(',', $ft[$i-1]);
101     $ft[$i-1] = '';
102     foreach ($items as $item){
103         $item = trim($item);
104         $item2 = strtolower($item);
105         $item = ucfirst($item2);
106         if ($item2 === 'python2')
107             $item = 'Python_2';
108         elseif ($item2 === 'python3')
109             $item = 'Python_3';
110         elseif ($item2 === 'pdf')
111             $item = 'PDF';
112         $item2 = strtolower($item);
113         if ( ! in_array($item2, array('c','c++','python_2','python_3','java','zip','pdf','txt')))
114             continue;
115         // If the problem is not Upload-Only, its language should be one of {C,C++,Python 2, Python 3,Java}
116         if ( ! in_array($i, $uo) && ! in_array($item2, array('c','c++','python_2','python_3','java')) )
117             continue;
118         $ft[$i-1] .= $item . ",";
119     }
120     $ft[$i-1] = substr($ft[$i-1],0,strlen($ft[$i-1])-1); // remove last ','
121     $problem = array(
122         'assignment' => $id,
123         'id' => $i,
124         'name' => $names[$i-1],
125         'score' => $scores[$i-1],
126         'is_upload_only' => in_array($i,$uo)?1:0,
127         'c_time_limit' => $c_tl[$i-1],
128         'python_time_limit' => $py_tl[$i-1],
129         'java_time_limit' => $java_tl[$i-1],
130         'memory_limit' => $ml[$i-1],
131         'allowed_languages' => $ft[$i-1],
132         'diff_cmd' => $dc[$i-1],
133         'diff_arg' => $da[$i-1],
134     );
135     $this->db->table('problems')->insert($problem);
136 }
137 if ($edit)
138 {
139     // We must update scoreboard of the assignment
140     $this->scoreboard_model->update_scoreboard($id);
141 }
142 // Complete Database Transaction
143 $this->db->transComplete();
144
145 return $this->db->transStatus();
146 }
147
148
149
150 // -----
151
152
153 /**
154 * Delete An Assignment
155 *
156 * @param $assignment_id
157 */
158 public function delete_assignment($assignment_id)
159 {
160     $this->db->transStart();
161
162     // Phase 1: Delete this assignment and its submissions from database
163     $this->db->table('assignments')->delete(['id'=>$assignment_id]);
164     $this->db->table('problems')->delete(['assignment'=>$assignment_id]);
165     $this->db->table('submissions')->delete(['assignment'=>$assignment_id]);
166
167     $this->db->transComplete();
168
169     if ($this->db->transStatus())
170     {
171         // Phase 2: Delete assignment's folder (all test cases and submitted codes)
172         $cmd = 'rm -rf '.rtrim($this->settings_model->get_setting('assignments_root'), '/').'/assignment_'.$assignment_id;
173         shell_exec($cmd);
174     }
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     * Returns database row for given assignment
279     *
280     * @param $assignment_id
281     * @return array
282     */
283     public function assignment_info($assignment_id)
284     {
285         $query = $this->db->table('assignments')->getWhere(['id'=>$assignment_id]);
286         if ($query->getNumRows() != 1){
287             return array(
288                 'id' => 0,
289                 'name' => 'Not_Selected',
290                 'finish_time' => 0,
291                 'extra_time' => 0,
292                 'problems' => 0
293             );
294         }  

295     }  

296     else{
297         return $query->getRowArray();
298     }
299 }
300
301 /**
302 * Is Participant
303 *
304 * Returns TRUE if $username if one of the $participants
305 * Examples for participants: "ALL" or "user1, user2, user3"
306 *
307 * @param $participants
308 * @param $username
309 * @return bool
310 */
311 public function is_participant($participants, $username)
312 {
313     $participants = explode(',', $participants);
314     foreach ($participants as &$participant){
315         $participant = trim($participant);
316     }
317     if(in_array('ALL', $participants))
318         return TRUE;
319     if(in_array($username, $participants))
320         return TRUE;
321     return FALSE;
322 }
323
324 /**
325 * Increase Total Submits
326 *
327 * Increases number of total submits for given assignment by one
328 *
329 * @param $assignment_id
330 * @return mixed
331 */
332 public function increase_total_submits($assignment_id)
333 {
334     // Get total submits
335     $total = $this->db->table('assignments')->select('total_submits')->getWhere(['id'=>$assignment_id])->getRow()->
336         total_submits;
337     // Save total+1 in DB
338     $this->db->table('assignments')->where('id', $assignment_id)->update(['total_submits'=>($total+1)]);
339
340     // Return new total
341     return ($total+1);
342 }
343
344 /**
345 * Set Moss Time
346 *
347 * Updates "Moss Update Time" for given assignment
348 *
349 * @param $assignment_id
350 */
351 public function set_moss_time($assignment_id)
352 {
353     $now = shj_now_str();
354     $this->db->table('assignments')->where('id', $assignment_id)->update(['moss_update'=>$now]);
355 }
356
357 /**
358 * Set Moss Time
359 *
360 * Updates "Moss Update Time" for given assignment
361 *
362 * @param $assignment_id
363 */
364 public function set_moss_time($assignment_id)
365 {
366     $now = shj_now_str();
367     $this->db->table('assignments')->where('id', $assignment_id)->update(['moss_update'=>$now]);
368 }
369
370 /**
371 * Set Moss Time
372 *
373 * Updates "Moss Update Time" for given assignment
374 */
375

```

```

376
377
378
379 /**
380 * Get Moss Time
381 *
382 * Returns "Moss Update Time" for given assignment
383 *
384 * @param $assignment_id
385 * @return string
386 */
387 public function get_moss_time($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 * Save Problem Description
397 *
398 * Saves (Adds/Updates) problem description (html or markdown)
399 *
400 * @param $assignment_id
401 * @param $problem_id
402 * @param $text
403 * @param $type
404 */
405 public function save_problem_description($assignment_id, $problem_id, $text, $type)
406 {
407     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
408
409     if ($type === 'html')
410     {
411         // Remove the markdown code
412         if(file_exists("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md")){
413             unlink("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md");
414         }
415         // Save the html code
416         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.html", $text);
417     }
418     elseif ($type === 'md')
419     {
420         // We parse markdown using Parsedown library
421         $this->parsedown = new Parsedown;
422         // Save the markdown code
423         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md", $text);
424         // Convert markdown to html and save the html
425         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.html", $this->parsedown->parse(
426             $text));
427     }
428 }
429
430
431 /**
432 * Update Coefficients
433 *
434 * Each time we edit an assignment (Update start time, finish time, extra time, or
435 * coefficients rule), we should update coefficients of all submissions of that assignment
436 *
437 * This function is called from add_assignment($id, TRUE)
438 *
439 * @param $assignment_id
440 * @param $extra_time
441 * @param $finish_time
442 * @param $new_late_rule
443 */
444 private function _update_coefficients($assignment_id, $extra_time, $finish_time, $new_late_rule)
445 {
446     $submissions = $this->db->table('submissions')->getWhere(['assignment'=>$assignment_id])->getResultArray();
447
448     $finish_time = strtotime($finish_time);
449
450     foreach ($submissions as $i => $item) {
451         $delay = strtotime($item['time'])-$finish_time;
452         ob_start();
453         if ( eval($new_late_rule) === FALSE )
454             $coefficient = "error";
455         if (!isset($coefficient))
456             $coefficient = "error";
457         ob_end_clean();
458         $submissions[$i]['coefficient'] = $coefficient;
459     }
460
461     // For better performance, we update each 1000 rows in one SQL query
462     $size = count($submissions);
463     for ($i=0; $i<($size-1)/1000; $i++) {
464         if ($this->db->dbdriver === 'postgre')
465             $query = 'UPDATE '.$this->db->prefixTable('submissions').'_AS_t_SET_coefficient=_c.coeff FROM _values_\n';
466         else
467
468
469
470
471
472
473

```

```

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']}'\n"
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 A.6: ScoreboardModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Scoreboard_model.php
5  * @author: Filipus Setio Nugroho <filipussetio@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']]))
56             $penalty[$submission['username']] = 0;
57
58             $total_score[$submission['username']] += $final_score;
59
60             $number_of_submissions = $this->db->table('submissions')->where(array(
61                 'assignment' => $submission['assignment'],
62                 'problem' => $submission['problem'],
63                 'username' => $submission['username'],
64             ))->countAllResults();
65
66
67
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 in 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 ! $scoreboard_enabled )
141     {
142         return;
143     }
144     // Generate the scoreboard
145     list ($scores, $scoreboard) = $this->_generate_scoreboard($assignment_id);
146
147     // Generate the scoreboard's html code
148     // todo: Save Scoreboard as json (generate html at client side)
149     $all_problems = $this->assignment_model->all_problems($assignment_id);
150     $total_score = 0;
151     foreach($all_problems as $i)
152     {
153         $total_score += $i['score'];
154     }
155     $data = array(
156         'assignment_id' => $assignment_id,
157         'problems' => $all_problems,
158         'total_score' => $total_score,
159         'scores' => $scores,
160         'scoreboard' => $scoreboard,
161         'names' => $this->user_model->get_names(),
162     );
163
164     $scoreboard_table = view('pages/scoreboard_table', $data);
165
166     // Minify the scoreboard's html code
167     // $scoreboard_table = $this->output->minify($scoreboard_table, 'text/html');
168
169     // Save the scoreboard's html code in Database

```

```

167|     $query = $this->db->table('scoreboard')->select('assignment')->getWhere(array('assignment'=>$assignment_id));
168|     if ($query->getNumRows() == 0)
169|         $this->db->table('scoreboard')->insert(array('assignment'=>$assignment_id, 'scoreboard'=>$scoreboard_table));
170|     else
171|         $this->db->table('scoreboard')->where('assignment', $assignment_id)->update(array('scoreboard'=>$scoreboard_table));
172|
173|
174| // -----
175|
176|
177|
178| /**
179| * Get Cached Scoreboard
180| *
181| * Returns the cached scoreboard of given assignment as a html text
182| *
183| * @param int $assignment_id
184| * @return string
185| */
186| public function get_scoreboard($assignment_id)
187| {
188|     $query = $this->db->table('scoreboard')->select('scoreboard')->getWhere(array('assignment'=>$assignment_id));
189|
190|     if ($query->getNumRows() != 1)
191|         return 'Scoreboard_not_found';
192|     else
193|         return $query->getRow()->scoreboard;
194|
195|
196}

```

Kode A.7: UserModel.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file User_model.php
5 * @author: Filipus Setio Nugroho <filipussetio@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 /**
35 * Have User
36 *
37 * Returns TRUE if there is a user with username $username in database
38 *
39 * @param $username
40 * @return bool
41 */
42 public function have_user($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
55
56 /**
57 * User ID to Username
58 *
59 * Converts user id to username (returns FALSE if user does not exist)
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/***
79 * Username to User ID
80 *
81 * Converts username to user id (returns FALSE if user does not exist)
82 *
83 * @param $username
84 * @return bool
85 */
86 public function username_to_user_id($username)
87{
88    $query = $this->db->table('users')->select('id')->getWhere(['username'=>$username]);
89    if ($query->getNumRows() == 0)
90        return FALSE;
91    return $query->getRow()->id;
92}
93
94
95// -----
96
97
98/***
99 * Have Email
100 *
101 * Returns TRUE if a user (except $username) with given email exists
102 *
103 * @param $email
104 * @param bool $username
105 * @return bool
106 */
107 public function have_email($email, $username = FALSE)
108{
109    $query = $this->db->table('users')->getWhere(['email'=>$email]);
110    if ($query->getNumRows() >= 1){
111        if($username !== FALSE && $query->getRow()->username == $username)
112            return FALSE;
113        else
114            return TRUE;
115    }
116    return FALSE;
117}
118
119
120// -----
121
122
123/***
124 * Add User
125 *
126 * Adds a single user
127 *
128 * @param $username
129 * @param $email
130 * @param $password
131 * @param $role
132 * @return bool|string
133 */
134 public function add_user($username, $email, $display_name, $password, $role)
135{
136    if( ! ctype_alnum($username) )
137        return 'Username_may_only_contain_alpha-numeric_characters.';
138    if (strlen($username) < 3 OR strlen($username) > 20 OR strlen($password) < 6 OR strlen($password) > 200)
139        return 'Username_or_password_length_error.';
140    if ($this->have_user($username))
141        return 'User_with_this_username_exists.';
142    if ($this->have_email($email))
143        return 'User_with_this_email_exists.';
144    if (strtolower($username) !== $username)
145        return 'Username_must_be_lowercase.';
146    $roles = array('admin', 'head_instructor', 'instructor', 'student');
147    if ( ! in_array($role, $roles) )
148        return 'Users_role_is_not_valid.';
149
150    $user=[
151        'username' => $username,
152        'email' => $email,
153        'display_name' => $display_name,
154        'password' => password_hash($password,PASSWORD_BCRYPT),
155        'role' => $role
156    ];
157    $this->db->table('users')->insert($user);
158    return TRUE; //success
159}
160
161
162// -----
163
164

```

```

165 /**
166 * Add Users
167 *
168 * Adds multiple users
169 *
170 * @param $text
171 * @param $send_mail
172 * @param $delay
173 * @return array
174 */
175 public function add_users($text, $send_mail, $delay)
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_email);
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 * Delete User
251 *
252 * Deletes a user with given user id
253 * Returns TRUE (success) or FALSE (failure)
254 *
255 * @param $user_id
256 * @return bool
257 */
258 public function delete_user($user_id)

```

```

263 {
264     $this->db->transStart();
265
266     $username = $this->user_id_to_username($user_id);
267     if ($username === FALSE)
268         return FALSE;
269     $this->db->table('users')->delete(['id'=>$user_id]);
270     $this->db->table('submissions')->delete(['username' => $username]);
271     // each time we delete a user, we should update all scoreboards
272     $this->scoreboard_model->update_scoreboards();
273
274     $this->db->transComplete();
275
276     if ($this->db->transStatus()) {
277         // Delete submitted files
278         shell_exec("cd_{$this->settings_model->get_setting('assignments_root')};_rm_-r_/*/{$username};");
279         return TRUE; // success
280     }
281     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 $user_id
294 * @return bool
295 */
296
297 public function delete_submissions($user_id)
298 {
299     $this->db->transStart();
300
301     $username = $this->user_id_to_username($user_id);
302     if ($username === FALSE)
303         return FALSE;
304     // delete all submissions from database
305     $this->db->table('submissions')->delete(['username'=>$username]);
306     // each time we delete a user's submissions, we should update all scoreboards
307     $this->scoreboard_model->update_scoreboards();
308
309     $this->db->transComplete();
310
311     if ($this->db->transStatus()) {
312         // delete all submitted files
313         shell_exec("cd_{$this->settings_model->get_setting('assignments_root')};_rm_-r_/*/{$username};");
314         return TRUE; // success
315     }
316
317     return FALSE; // failure
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 $password
330 * @return bool
331 */
332
333 public function validate_user($username, $password)
334 {
335     $query = $this->db->table('users')->getWhere(['username' => $username]);
336     if ($query->getNumRows() != 1)
337         return FALSE;
338     if ($query->getRow()->username !== $username) // needed (because of utf8_general_ci [ci=case insensitive])
339         return FALSE;
340     if (password_verify($password, $query->getRow()->password))
341         return TRUE;
342
343     if($this->config->shj_authenticate == 'radius') {
344         $client = new Radius();
345         $client->setServer($this->config->shj_radius['server']) // RADIUS server address
346         ->setSecret($this->config->shj_radius['secret']);
347         if($client->accessRequest($username, $password))
348             return TRUE;
349     } else if($this->config->shj_authenticate == 'ldap') {
350         try {
351             $ldap = new \Adldap\Adldap();
352             $ldap->addProvider($this->config->shj_ldap);
353
354             $guard = $ldap->getDefaultProvider()->getGuard();
355             return $guard->attempt($username, $password);
356         } catch (\Exception $e) {
357             return FALSE;
358         }
359     }
360     return FALSE;
361 }

```

```

362
363
364 // -----
365
366
367 /**
368 * Selected Assignment
369 *
370 * Returns selected assignment by given username
371 * @param $username
372 * @return mixed
373 */
374 public function selected_assignment($username)
375 {
376     $this->session = session();
377     $query = $this->db->table('users')->select('selected_assignment')->getWhere(['username'=>$username]);
378     if ($query->getNumRows() != 1){//logout
379         $this->session->destroy();
380         return redirect()->to('login');
381     }
382     return $query->getRow()->selected_assignment;
383 }
384
385
386 // -----
387
388
389 /**
390 * Get Display Name
391 *
392 * Returns name of the user with given username
393 *
394 * @return array
395 */
396 public function get_names()
397 {
398     $query = $this->db->table('users')->select('username,_display_name')->get();
399     $tmp = $query->getResultArray();
400     $result = array();
401     foreach ($tmp as $row)
402         $result[$row['username']] = $row['display_name'];
403     return $result;
404 }
405
406
407 // -----
408
409
410 /**
411 * Update Profile
412 *
413 * Updates User Profile (Name, Email, Password, Role)
414 *
415 * @param $user_id
416 * @return bool
417 */
418 public function update_profile($user_id,$email,$password,$role,$display_name)
419 {
420     $query = $this->db->table('users')->getWhere(['id'=>$user_id]);
421     if ($query->getNumRows() != 1)
422         return FALSE;
423     $the_user = $query->getRow();
424     $username = $the_user->username;
425
426     $display_name = $display_name;
427     $locked = $this->settings_model->get_setting('lock_student_display_name');
428
429     if ($locked == 1) {
430         $display_name = $the_user->display_name;
431     }
432
433     $user=array(
434         'display_name' => $display_name,
435         'email' => $email
436     );
437
438     // if a role is provided, change the role
439     // (only admins are able to provide a role)
440     if ($role !== NULL)
441         $user['role'] = $role;
442
443     // if a password is provided, change the password:
444     if ($password != ''){
445         $user['password'] = password_hash($password, PASSWORD_BCRYPT);
446     }
447
448     $this->db->table('users')->where('username', $username)->update($user);
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('Y-m-d H:i:s', $now)));
476
477         // send the email:
478         $this->response = response();
479
480         $this->email->initialize($this->config->shj_mail);
481         $this->email->setNewline("\r\n");
482         $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
483
484         $this->email->setTo($email);
485         $this->email->setSubject('Password Reset');
486         $text = $this->settings_model->get_setting('reset_password_mail');
487         $text = str_replace('{SITE_URL}', base_url(), $text);
488         $text = str_replace('{RESET_LINK}', site_url('login/reset/'.$passchange_key), $text);
489         $text = str_replace('{VALID_TIME}', '1_hour', $text); // links are valid for 1 hour
490         $this->email->setMessage($text);
491         $this->email->send();
492     }
493
494     // -----
495
496     /**
497      * Password Reset Key Is Valid
498      *
499      * Returns TRUE if the given password reset key is valid
500      * And returns an error message if key is invalid
501      *
502      * @param $passchange_key
503      * @return bool|string
504      */
505     public function passchange_is_valid($passchange_key)
506     {
507         $query = $this->db->table('users')->select('passchange_time')->getWhere(['passchange_key'=>$passchange_key]);
508         if ($query->getNumRows() != 1)
509             return 'Invalid_password_reset_link.';
510         $time = strtotime($query->getRow()->passchange_time);
511         $now = shj_now();
512         if ($now-$time > 3600 OR $now-$time < 0) // reset link is valid for 1 hour
513             return 'The_link_is_expired.';
514         return TRUE;
515     }
516
517
518     // -----
519
520
521     /**
522      * Reset Password
523      *
524      * Resets password for given password reset key (in case of lost password)
525      *
526      * @param $passchange_key
527      * @param $newpassword
528      * @return bool
529      */
530     public function reset_password($passchange_key, $newpassword)
531     {
532         $query = $this->db->table('users')->getWhere(array('passchange_key'=>$passchange_key));
533         if ($query->getNumRows() != 1)
534             return FALSE; //failure
535         $this->db->table('users')->where('username', $query->getRow()->username)->update(array('passchange_key'=>'', 'password' => password_hash($newpassword,PASSWORD_BCRYPT)));
536         return TRUE; //success
537     }
538
539
540     // -----
541
542
543     /**
544      * Get All Users
545      *
546      * Returns an array of all users (for Users page)
547      *
548      * @return mixed
549      */
550     public function get_all_users()
551     {
552         return $this->db->table('users')->orderBy('role', 'asc')->orderBy('id')->get()->getResultArray();
553     }
554
555
556

```

```

557 // -----
558
559 /**
560 * Get User
561 *
562 * Returns database row for given user id
563 *
564 * @param $user_id
565 * @return bool
566 */
567 public function get_user($user_id)
568 {
569     $query = $this->db->table('users')->getWhere(array('id'=>$user_id));
570     if ($query->getNumRows() != 1)
571         return FALSE;
572     return $query->getRow();
573 }
574
575
576 // -----
577
578 /**
579 * Update Login Time
580 *
581 * Updates First Login Time and Last Login Time for given username
582 *
583 */
584 public function update_login_time($username)
585 {
586     helper('shj_helper');
587     $now = shj_now_str();
588
589     $first_login = $this->db->table('users')->select('first_login_time')->getWhere(array('username'=>$username))->getRow()->
590         first_login_time;
591     if ($first_login === NULL)
592         $this->db->table('users')->where('username', $username)->update(array('first_login_time'=>$now));
593
594     $this->db->table('users')->where('username', $username)->update(array('last_login_time'=>$now));
595 }
596
597
598 // -----
599 }
600

```

## A.3 View

Kode A.8: 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.getElementsByTagName('textarea'));
17         $switch = false;
18         $("textarea").keyup(function (e) {
19             if (e.which==27){
20                 tabOverride.set(document.getElementsByTagName('textarea'),$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;?td&gt;&lt;td&gt;PID&lt;/td&gt;
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;\n
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
</pre

```

```

40    <td><button class="delete_problem" type="button" aria-label="Delete_Problem"><i class="fa fa-times-circle fa-lg color1 pointer"
41    "></i></button></td>
42  </tr>;
43  $(document).ready(function(){
44    $('#add').click(function(){
45      $('#problems_table>tbody').append(shj.row.replace(/PID/g, (shj.num_of_problems+1)));
46      shj.num_of_problems++;
47      $('#nop').attr('value', shj.num_of_problems);
48    });
49    $("#form_a_archived_assignment").click(function(){
50      if ($("#form_a_archived_assignment").is(':checked')) {
51        $('#start_time').val('1970-01-02_00:00:00');
52        $('#finish_time').val('2038-01-18_00:00:00');
53        $('#form_extra_time').val('0');
54      }
55      else{
56        $('#start_time').val('');
57        $('#finish_time').val('');
58        $('#form_extra_time').val('');
59      }
60    });
61    $(document).on('click', '.delete_problem', function(){
62      if (shj.num_of_problems==1) return;
63      var row = $(this).parents('tr');
64      row.remove();
65      var i = 0;
66      $('#problems_table>tbody').children('tr').each(function(){
67        i++;
68        $(this).children(':first').html(i);
69        $(this).find(['type="checkbox"]').attr('value',i);
70      });
71      shj.num_of_problems--;
72      $('#nop').attr('value',shj.num_of_problems);
73    });
74    $('#start_time').datetimepicker({
75      timeFormat: 'HH:mm:ss'
76    });
77    $('#finish_time').datetimepicker({
78      timeFormat: 'HH:mm:ss'
79    });
80  </script>
81  <?= $this->endSection() ?>
82
83
84
85  <?= $this->section('title_menu') ?>
86  <span class="title_menu_item">
87    <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>
88  </span>
89  <?= $this->endSection() ?>
90
91
92
93  <?= $this->section('main_content') ?>
94  <?php $msgclasses = ['success' => 'shj_g', 'notice' => 'shj_o', 'error' => 'shj_r'] ?>
95  <?php foreach($messages as $message): ?>
96    <p class="<?= $msgclasses[$message->type] ?>"><? $message->text ?></p>
97  <?php endforeach ?>
98
99  <?php if($edit): ?>
100 <p>
101   <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.
102 </p>
103 <?php endif ?>
104
105 <?= _form_open_multipart($edit ? assignments/edit/.$edit_assignment['id'] : assignments/add) ?>
106 <div class="panel_left">
107   <input type="hidden" name="number_of_problems" id="nop" value="<?=$edit ? $edit_assignment['problems'] : count($problems) ?"/>
108   <p class="input_p">
109     <label for="form_a_name">Assignment_Name</label>
110     <input id="form_a_name" type="text" name="assignment_name" class="sharif_input_medium" value="<?=$edit ? $edit_assignment['name'] : set_value('assignment_name') ?"/>
111     <div class="shj_error"><?=$validationError->hasError('assignment_name') ? $validationError->getError('assignment_name') : '' ?></div>
112   </p>
113   <p class="input_p">
114     <label for="start_time">Start_Time</label>
115     <input id="start_time" type="text" name="start_time" class="sharif_input_medium" value="<?=$edit ? date('m/d/Y H:i:s', strtotime($edit_assignment['start_time'])) : set_value('start_time') ?"/>
116     <div class="shj_error"><?=$validationError->hasError('start_time') ? $validationError->getError('start_time') : '' ?></div>
117   </p>
118   <p class="input_p">
119     <label for="finish_time">Finish_Time</label>
120     <input id="finish_time" type="text" name="finish_time" class="sharif_input_medium" value="<?=$edit ? date('m/d/Y H:i:s', strtotime($edit_assignment['finish_time'])) : set_value('finish_time') ?"/>
121     <div class="shj_error"><?=$validationError->hasError('finish_time') ? $validationError->getError('finish_time') : '' ?></div>
122   </p>
123   <p class="input_p_clear">
124     <label for="form_extra_time">
125       Extra_Time_(minutes)<br>
126       <span class="form_comment">Extra_time_for_late_submissions.</span>
127     </label>
128     <input id="form_extra_time" type="text" name="extra_time" class="sharif_input_medium" value="<?=$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 <span>Only_these_users_areable_to_submit_You_cany_use_keyword_ALL_.</span>
135 <span class="form_comment_clear">Press_esc_to_enable/disable_tabindent</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/Shariif-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 <span>PDF_File_of_Assignment<br>
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 <input checked="" type="checkbox" value="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 <input checked="" type="checkbox" value="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['javaexceptions'] ?> checked' : '' >?
173 <input checked="" type="checkbox" value="1" />
174 <label for="form_a_javaexceptions" class="default">Java_Exceptions</label>
175 <span class="form_comment_space-left">Check_this_to_show_Java_exceptions_to_users</span>
176 <div class="shj_error"><?= $_validationError->hasError('javaexceptions') ?>$_validationError->getError('javaexceptions') : '' ?></div>
177 </p>
178 <p class="input_p">
179 <input id="form_a_archived_assignment" type="checkbox" name="archived_assignment" value="1" <?= $_edit ?>$_edit_assignment['archived_assignment'] ?> checked' : '' >?
180 <input checked="" type="checkbox" value="1" />
181 <label for="form_a_archived_assignment" class="default">Archived_Assignment</label>
182 <span class="form_comment_space-left">Check_this_to_make_an_archived_assignment</span>
183 <div class="shj_error"><?= $_validationError->hasError('archived_assignment') ?>$_validationError->getError('archived_assignment') : '' ?></div>
184 </p>
185 <p class="input_p">
186 <label for="form_late_rule">Coefficient_rule_(<a target="_blank" href="https://github.com/ifunpar/Shariif-Judge/blob/docs/v1.4/add_assignment.md#coefficient-rule">?</a>)</label><br>
187 <span class="form_comment_medium_clear" style="display: block;">>PHP_script_without_&lt;?php_&gt;_tags</span>
188 <span class="form_comment_clear">Press_esc_to_enable/disable_tabindent<br>
189 <input id="form_late_rule" name="late_rule" rows="20" class="sharif_input_add_text"><?= $_edit ?>$_edit_assignment['late_rule'] : $_set_value('late_rule', '$default_late_rule') ?></input>
190 <div class="shj_error"><?= $_validationError->hasError('late_rule') ?>$_validationError->getError('late_rule') : '' ?></div>
191 </p>
192 <table id="problems_table">
193 <thead>
194 <tr>
195 <th rowspan="2" colspan="2">Name</th>
196 <th colspan="3" style="border-bottom: 1px solid #BDBDBD">Time_Limit_(ms)</th>
197 <th rowspan="2" colspan="2" style="border-bottom: 1px solid #BDBDBD">Memory<br>Limit_(KB)</th>
198 <th rowspan="2" colspan="2" style="border-bottom: 1px solid #BDBDBD">Allowed<br>Languages_(<a aria-label="Link_Help_For_Languages" target="_blank" href="https://github.com/ifunpar/Shariif-Judge/blob/docs/v1.4/add_assignment.md#allowed-languages">?</a>)</th>
199 <th rowspan="2" colspan="2" style="border-bottom: 1px solid #BDBDBD">Diff<br>Command_(<a aria-label="Link_Help_For_Diff_Command" target="_blank" href="https://github.com/ifunpar/Shariif-Judge/blob/docs/v1.4/add_assignment.md#diff-command">?</a>)</th>
200 <th rowspan="2" colspan="2" style="border-bottom: 1px solid #BDBDBD">Diff<br>Argument_(<a aria-label="Link_Help_For_Diff_Argument" target="_blank" href="https://github.com/ifunpar/Shariif-Judge/blob/docs/v1.4/add_assignment.md#diff-arguments">?</a>)</th>
201 <th rowspan="2" colspan="2" style="border-bottom: 1px solid #BDBDBD">Upload<br>Only_(<a aria-label="Link_Help_For_Upload_Only" target="_blank" href="https://github.com/ifunpar/Shariif-Judge/blob/docs/v1.4/add_assignment.md#upload-only">?</a>)</th>
202 <th rowspan="2" colspan="2" style="border-bottom: 1px solid #BDBDBD">C/C++<br>Python<br>Java</th>
203 </tr>
204 </thead>
205 <tbody>
206 <tr>
207 <td colspan="2" style="text-align: center;"><?php foreach_($problems_as_$problem) : ?>
208 </tr>
209 <tr>
210 </tr>

```

```

211 <td><?= $problem['id'] ?></td>
212 <td><input_aria-label="Problem_Name" type="text" name="name[]" class="sharif_input_short" value="<?=$problem['name'] ?>" /></td>
213 <td><input_aria-label="Score" type="text" name="score[]" class="sharif_input_tiny2" value="<?=$problem['score'] ?>" /></td>
214 <td><input_aria-label="Time_Limit_for_C" type="text" name="c_time_limit[]" class="sharif_input_tiny2" value="<?=$problem['c_time_limit'] ?>" /></td>
215 <td><input_aria-label="Time_Limit_for_Python" type="text" name="python_time_limit[]" class="sharif_input_tiny2" value="<?=$problem['python_time_limit'] ?>" /></td>
216 <td><input_aria-label="Time_Limit_for_Java" type="text" name="java_time_limit[]" class="sharif_input_tiny2" value="<?=$problem['java_time_limit'] ?>" /></td>
217 <td><input_aria-label="Memory_Limit" type="text" name="memory_limit[]" class="sharif_input_tiny" value="<?=$problem['memory_limit'] ?>" /></td>
218 <td><input_aria-label="Allowed_Languages" type="text" name="languages[]" class="sharif_input_short2" value="<?=$problem['allowed_languages'] ?>" /></td>
219 <td><input_aria-label="Diff_Command" type="text" name="diff_cmd[]" class="sharif_input_tiny" value="<?=$problem['diff_cmd'] ?>" /></td>
220 <td><input_aria-label="Diff_Argument" type="text" name="diff_arg[]" class="sharif_input_tiny" value="<?=$problem['diff_arg'] ?>" /></td>
221 <td><input_aria-label="Upload_Only" type="checkbox" name="is_upload_only[]" class="check" value="<?=$problem['id'] ?>" ?> <?=$problem['is_upload_only'] ? checked : '' ?></td>
222 <td><button_aria-label="Delete_Problem" type="button" class="fa fa-times-circle fa-lg color1 pointer"></i></td>
223 </tr>
224 <?php endforeach ?>
225 </tbody>
226 </table>
227 </p>
228 <div class="shj_error"><?=$validationError->hasError('name.*') ?> $validationError->getError('name.*') : '' ?></div>
229 <div class="shj_error"><?=$validationError->hasError('score.*') ?> $validationError->getError('score.*') : '' ?></div>
230 <div class="shj_error"><?=$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 A.9: settings.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: settings.php
4 # author: Filips Setio Nugroho <filipussetio@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/Sharif-Judge/blob/docs/v1.4/settings.md" target="_blank"><i class="fa fa-question-circle 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_?>/><br>
81    <span class="form_comment_clear">Sets a limit for size of output file generated by submitted code</span>
82    <div class="shj_error"><?= $validationError->hasError('output_size_limit') ? $validationError->getError('output_size_limit') : '' ?></div>
83  </p>
84  <p class="input_p">
85    <label for="form_rpp_all">Results Per Page</label>
86    <input id="form_rpp_all" type="text" name="rpp_all" class="sharif_input_medium" value=<?=$_results_per_page_all_?>/>
87    <p class="form_comment_clear">In 'All_Submissions'<br>Enter 0 for no limit</p>
88    <div class="shj_error"><?= $validationError->hasError('rpp_all') ? $validationError->getError('rpp_all') : '' ?></div>
89  </p>
90  <p class="input_p">
91    <label for="form_rpp_final">Results Per Page</label>
92    <input id="form_rpp_final" type="text" name="rpp_final" class="sharif_input_medium" value=<?=$_results_per_page_final_?>/>
93    <p class="form_comment_clear">In 'Final_Submissions'<br>Enter 0 for no limit</p>
94    <div class="shj_error"><?= $validationError->hasError('rpp_final') ? $validationError->getError('rpp_final') : '' ?></div>
95  </p>
96  <p class="input_p">
97    <input id="form_en_reg" type="checkbox" name="enable_registration" value="1" <?= $enable_registration ? 'checked' : '' ?>/>
98    <label for="form_en_reg">Registration</label><br>
99    <span class="form_comment">Open Public Registration.</span>
100   </p>
101   <p class="input_p">
102     <label for="form_reg_code">Registration Code</label>
103     <input id="form_reg_code" type="text" name="registration_code" class="sharif_input_medium" value=<?=$_registration_code_?>/><br>
104     <p class="form_comment_clear">If you want to enable registration (above option), It is better to give a registration code<br>
105       to students in your class for validating registration. Enter 0 to disable.</p>
106     <div class="shj_error"><?= $validationError->hasError('registration_code') ? $validationError->getError('registration_code') : '' ?></div>
107   </p>
108   <p class="input_p">
109     <input id="form_log" type="checkbox" name="enable_log" value="1" <?= $enable_log ? 'checked' : '' ?>/>
110     <label for="form_log">Log</label><br>
111     <span class="form_comment">Enable Log</span>
112   </p>
113   <p class="input_p">
114     <input id="form_lock_student_display_name" type="checkbox" name="lock_student_display_name" value="1" <?= $lock_student_display_name ? 'checked' : '' ?>/>
115     <label for="form_lock_student_display_name">Lock Student's Display Name</label><br>
116     <span class="form_comment">Student's can't change their display name</span>
117   </p>
118   <p class="input_p">
119     <label for="form_late_rule">Default_Coefficient_Rule</label>
120     <span class="form_comment_clear">PHP_script_without_&lt;?php_?&gt;;_tags</span><br>
121     <span class="form_comment_clear">Press esc to enable/disable tab indent</span><br>
122     <textarea id="form_late_rule" name="default_late_rule" rows="15" class="sharif_input_add_text_clear"><?=$_default_late_rule_?></textarea>
123   </p>
124
125   <h2>Email</h2>
126
127   <p class="input_p">
128     <label for="form_mail_from">Send Emails From</label>
129     <input id="form_mail_from" type="text" name="mail_from" class="sharif_input_medium" value=<?=$_mail_from_?>/>
130     <div class="shj_error"><?= $validationError->hasError('mail_from') ? $validationError->getError('mail_from') : '' ?></div>
131   </p>
132   <p class="input_p">
133     <label for="form_mail_name">Send Emails From Name</label>
134     <input id="form_mail_name" type="text" name="mail_from_name" class="sharif_input_medium" value=<?=$_mail_from_name_?>/>
135     <div class="shj_error"><?= $validationError->hasError('mail_from_name') ? $validationError->getError('mail_from_name') : '' ?></div>
136   </p>

```

```

137 137< p class="input_p">
138 138    <label_for="form_mail_reset">Password_Reset_Email</label>
139 139    <span class="form_comment">You_can_use_{SITE_URL},{RESET_LINK}_and_{VALID_TIME}</span><br>
140 140    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
141 141    <textarea_id="form_mail_reset" name="reset_password_mail" rows="15" class="sharif_input_add_text_clear"><?=$_
142 142      $reset_password_mail_?></textarea>
143 143  </p>
144 144  <label_for="form_mail_add">Add_User_Email</label>
145 145    <span class="form_comment_clear">You_can_use_{SITE_URL},{LOGIN_URL},{ROLE},{USERNAME},and,{PASSWORD}</span><br>
146 146    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
147 147    <textarea_id="form_mail_add" name="add_user_mail" rows="15" class="sharif_input_add_text_clear"><?=$_
148 148      $add_user_mail_?></textarea>
149 149  </p>
150 150
151 151  <h2 class="shj_form">
152 152    <Sandboxing><span class="title_menu_item">
153 153      <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md" target="_blank"><i class="fa fa-
154 154        question-circle_color11"></i> Help</a>
155 155    </span>
156 156  </h2>
157 157  <p class="input_p">
158 158    <input_id="form_easysandbox" type="checkbox" name="enable_easysandbox" value="1" <?=$_enable_easysandbox_?>' checked' <: >
159 159    </p>
160 160  <label_for="form_easysandbox">EasySandbox</label>
161 161  <p class="form_comment">Enable_EasySandbox_for_C/C++.<br>
162 162    You_must<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md#build-easysandbox" target="_blank">
163 163      build_EasySandbox</a>_before_enabling_it.<br>
164 164      <?php_if_!$sandbox_built:>?>
165 165      <span style="color:red;">You_have_not_built_EasySandbox_yet.</span>
166 166    <?php_endif_?>
167 167  </p>
168 168  <p class="input_p">
169 169    <input_id="form_java_policy" type="checkbox" name="enable_java_policy" value="1" <?=$_enable_java_policy_?>' checked' <: >
170 170    </p>
171 171  <label_for="form_java_policy">Java_Policy</label>
172 172  <p class="form_comment">
173 173    <Enable><a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md#java-sandboxing" target="_blank">Java
174 174      Sandboxing</a>
175 175  </p>
176 176
177 177  <h2 class="shj_form">
178 178    <Shield><span class="title_menu_item">
179 179      <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank"><i class="fa fa-question-
180 180        circle_color11"></i> Help</a>
181 181    </span>
182 182  </h2>
183 183
184 184  <p class="input_p">
185 185    <input_id="form_c_sh" type="checkbox" name="enable_c_shield" value="1" <?=$_enable_c_shield_?>' checked' <: > <label_for="form_c_sh">C_Shield</label><br>
186 186    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank">
187 187      Shield</a>_for_C</span>
188 188  </p>
189 189  <p class="input_p">
190 190    <input_id="form_cpp_sh" type="checkbox" name="enable_cpp_shield" value="1" <?=$_enable_cpp_shield_?>' checked' <: > <label_for="form_cpp_sh">C++_Shield</label><br>
191 191    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank">
192 192      Shield</a>_for_C++</span>
193 193  </p>
194 194  <p class="input_p">
195 195    <input_id="form_py2_sh" type="checkbox" name="enable_py2_shield" value="1" <?=$_enable_py2_shield_?>' checked' <: > <label_for="form_py2_sh">Python_2_Shield</label><br>
196 196    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank">
197 197      Shield</a>_for_Python_2</span>
198 198  </p>
199 199  <p class="input_p">
200 200    <input_id="form_py3_sh" type="checkbox" name="enable_py3_shield" value="1" <?=$_enable_py3_shield_?>' checked' <: > <label_for="form_py3_sh">Python_3_Shield</label><br>
201 201    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank">
202 202      Shield</a>_for_Python_3</span>
203 203  </p>
204 204  <p class="input_p">
205 205    <label_for="form_def_c">Shield_Rules_(for_C)</label>
206 206    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
207 207    <textarea_id="form_def_c" name="def_c" rows="15" class="sharif_input_add_text_clear"><?=$_defc_?></textarea>
208 208  </p>
209 209  <p class="input_p">
210 210    <label_for="form_def_cpp">Shield_Rules_(for_C++)</label>
211 211    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
212 212    <textarea_id="form_def_cpp" name="def_cpp" rows="15" class="sharif_input_add_text_clear"><?=$_defcpp_?></textarea>
213 213  </p>
214 214  <p class="input_p">
215 215    <label_for="form_shield_py2">Shield_(for_Python_2)</label>
216 216    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
217 217    <textarea_id="form_shield_py2" name="shield_py2" rows="15" class="sharif_input_add_text_clear"><?=$_shield_py2_?></textarea>
218 218  </p>
219 219  <p class="input_p">
220 220    <label_for="form_shield_py3">Shield_(for_Python_3)</label>
221 221    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
222 222    <textarea_id="form_shield_py3" name="shield_py3" rows="15" class="sharif_input_add_text_clear"><?=$_shield_py3_?></textarea>

```

```

222     textarear>
223     </p>
224     <p class="input_p">
225     <br>
226     <input type="submit" value="Save_Changes" class="sharif_input"/>
227   </form>
228 <?=$this->endSection()?>

```

Kode A.10: 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 <?= $this->section('title_menu') ?>
11 <?php if ($user->level >= 2): ?>
12 <span class="title_menu_item"><a href="=site_url('assignments/add')?&gt;"&gt;&lt;i class="fa_fa-plus_color8"&gt;&lt;/i&gt; Add&lt;/a&gt;&lt;/span&gt;
13 &lt;?php endif ?&gt;
14 &lt;?= $this-&gt;endSection() ?&gt;
15 &lt;?php $this-&gt;section('main_content') ?&gt;
16 &lt;?php $msgclasses = ['success'=&gt; 'shj_g', 'notice'=&gt; 'shj_o', 'error'=&gt; 'shj_r'] ?&gt;
17 &lt;?php foreach ($messages as $message): ?&gt;
18 &lt;p class="<?=$msgclasses[$message['type']]?&gt;"&gt;&lt;?= $message['text'] ?&gt;&lt;/p&gt;
19 &lt;?php endforeach ?&gt;
20 &lt;?php if (count($all_assignments)== 0): ?&gt;
21 &lt;p style="text-align: center;"&gt;Nothing to show...&lt;/p&gt;
22 &lt;br/&gt;
23 &lt;table class="sharif_table"&gt;
24 &lt;thead&gt;
25 &lt;tr&gt;
26   &lt;th&gt;Select&lt;/th&gt;
27   &lt;th&gt;Name&lt;/th&gt;
28   &lt;th&gt;Problems&lt;/th&gt;
29   &lt;th&gt;Submissions&lt;/th&gt;
30   &lt;th&gt;Coefficient&lt;/th&gt;
31   &lt;th&gt;Start Time&lt;/th&gt;
32   &lt;th&gt;Finish Time&lt;/th&gt;
33   &lt;th&gt;Status&lt;/th&gt;
34   &lt;th&gt;PDF&lt;/th&gt;
35   &lt;th&gt;Actions&lt;/th&gt;
36 &lt;?php endif ?&gt;
37 &lt;/tr&gt;
38 &lt;/thead&gt;
39 &lt;?php foreach (array_reverse($all_assignments) as $item): ?&gt;
40 &lt;tr&gt;
41   &lt;td&gt;&lt;i tabindex="0" class="pointer_select_assignment_fa_&lt;?=$item['id']?&gt;_user-&gt;selected_assignment['id']?&gt;_fa-check-square_color6_o?&gt;' fa-square-o'&gt;_?&gt;fa-2x" data-id="<?=$item['id']?&gt;"&gt;&lt;/i&gt;&lt;/td&gt;
42   &lt;td dir="auto"&gt;&lt;?= $item['name'] ?&gt;&lt;/td&gt;
43   &lt;td&gt;&lt;a href="<?=site_url('problems/'.$item['id'])?&gt;"&gt;&lt;?= $item['problems'] ?&gt; problem&lt;?= $item['problems'] != 1 ? 's' : '' ?&gt;&lt;/a&gt;&lt;/td&gt;
44   &lt;td&gt;&lt;?= $item['total_submits'] ?&gt; submission&lt;?= $item['total_submits'] != 1 ? 's' : '' ?&gt;&lt;/td&gt;
45   &lt;td&gt;
46     &lt;?php if ($item['finished']): ?&gt;
47       &lt;span style="color: _red;"&gt;Finished&lt;/span&gt;
48     &lt;?php else: ?&gt;
49       &lt;?= $item['coefficient'] ?&gt; %
50     &lt;?php endif ?&gt;
51   &lt;/td&gt;
52   &lt;td&gt;&lt;?= $item['start_time'] ?&gt;&lt;/td&gt;
53   &lt;td&gt;&lt;?= $item['finish_time'] ?&gt;&lt;/td&gt;
54   &lt;td&gt;
55     &lt;?php if ($item['open']): ?&gt;
56       &lt;span style="color: _green;"&gt;Open&lt;/span&gt;
57     &lt;?php else: ?&gt;
58       &lt;span style="color: _red;"&gt;Close&lt;/span&gt;
59     &lt;?php endif ?&gt;
60   &lt;/td&gt;
61   &lt;td&gt;
62     &lt;?php if ($user-&gt;level &gt; 0): ?&gt;
63       &lt;a href="<?=site_url('assignments/pdf/'.$item['id'])?&gt;"&gt;&lt;img src="<?=base_url('assets/images/pdf.svg')?&gt;" aria-label="Download_PDF_For_Assignment_&lt;?=$item['name']?&gt;"&gt;&lt;/a&gt;
64     &lt;/td&gt;
65   &lt;?php endif ?&gt;
66   &lt;td&gt;
67     &lt;?php if ($user-&gt;level &gt; 0): ?&gt;
68       &lt;a href="<?=site_url('assignments/downloadtestsdesc/'.$item['id'])?&gt;"&gt;&lt;i title="Download_Tests_and_Descriptions" class="fa_fa-cloud-download_fa-lg_color11"&gt;&lt;/i&gt;&lt;/a&gt;
69     &lt;?php endif ?&gt;
70   &lt;?php if ($user-&gt;level &gt;= 1): ?&gt;
71     &lt;a href="<?=site_url('assignments/download_submissions/by_user/'.$item['id'])?&gt;"&gt;&lt;i title="Download_Final_Submissions_(by_user)" class="fa_fa-download_fa-lg_color12"&gt;&lt;/i&gt;&lt;/a&gt;
72     &lt;a href="<?=site_url('assignments/download_submissions/by_problem/'.$item['id'])?&gt;"&gt;&lt;i title="Download_Final_Submissions_(by_problem)" class="fa_fa-download_fa-lg_color2"&gt;&lt;/i&gt;&lt;/a&gt;
73   &lt;?php endif ?&gt;
74   &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
75     &lt;a href="<?=site_url('moss/'.$item['id'])?&gt;"&gt;&lt;i title="Detect_Similar_Codes" class="fa_fa-shield_fa-lg_color7"&gt;&lt;/i&gt;&lt;/a&gt;
76   &lt;?php endif ?&gt;
77 &lt;?php endif ?&gt;
78 &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
79     &lt;a href="<?=site_url('moss/'.$item['id'])?&gt;"&gt;&lt;i title="Detect_Similar_Codes" class="fa_fa-shield_fa-lg_color7"&gt;&lt;/i&gt;&lt;/a&gt;
80   &lt;?php endif ?&gt;
</pre

```

```

81     <?php if ($user->level >= 2): ?>
82         <a href=<?=site_url('assignments/edit/'.$item['id'])?>><i title="Edit" class="fa_fa-pencil_fa-lg_color3"></i></a>
83     </?php endif ?>
84     <?php if ($user->level >= 2): ?>
85         <a href=<?=site_url('assignments/delete/'.$item['id'])?>><i title="Delete" class="fa_fa-times_fa-lg_color1"></i></a>
86     </?php endif ?>
87     </td>
88 </?php endforeach ?>
89 </table>
90 <?php endif ?>
91 <?php $this->endSection() ?>
92

```

Kode A.11: 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
11 <?= $this->section('other_assets') ?>
12 <link rel='stylesheet' type='text/css' href='<?=base_url('assets/fullcalendar/fullcalendar.css')?>' />
13 <script type='text/javascript' src='<?=base_url('assets/fullcalendar/fullcalendar.min.js')?>'></script>
14 <link rel='stylesheet' type='text/css' href='<?=base_url('assets/gridster/jquery.gridster.css')?>' />
15 <script type='text/javascript' src='<?=base_url('assets/gridster/jquery.gridster.min.js')?>'></script>
16 <script type='text/javascript' src='<?=base_url('assets/js/jquery.autoellipsis-1.0.10.min.js')?>'></script>
17 <script>
18 $(document).ready(function () {
19     $('#calendar').fullCalendar({
20         timeFormat: 'HH:mm ~ HH:mm',
21         editable: false,
22         height: 280,
23         firstDay: <?= $week_start ?>,
24         events: [
25             <?php $colors = ['#812C8C', '#FF7500', '#2C578C', '#013440', '#A6222C', '#42758C', '#02A300', '#BA6900']; ?>
26             <?php foreach ($all_assignments as $idx => $assignment): ?>
27                 <?php if ($assignment['archived_assignment'] == '0'): ?>
28                     {id: <?= $assignment['id'] ?>, title: '<?=$assignment['name']?>', start: '<?=$assignment['start_time']?>', end: '<?=$assignment['finish_time']?>', allDay: false, color: '<?=$colors[($idx)%count($colors)]?>'},
29                 <?php endif; ?>
30             <?php endforeach; ?>
31         ]
32     });
33     var gridster = $(".gridster_ul").gridster({
34         widget_margins: [10, 10],
35         widget_base_dimensions: [350, 350],
36         serialize_params: function ($w, wgd) {
37             return {
38                 r: wgd.row,
39                 c: wgd.col,
40                 x: wgd.size_x,
41                 y: wgd.size_y
42             }
43         },
44         draggable: {
45             handle: '.widget_title',
46             stop: function (event, ui) { // send widget positions to server for saving in database
47                 var positions = JSON.stringify(gridster.serialize());
48                 $.post(
49                     "<?=site_url('/dashboard/widget_positions')?>",
50                     {
51                         positions: positions,
52                         shj_csrf_token: shj.csrf_token
53                     }
54                 );
55             }
56         }
57     }).data('gridster');
58     $('.notif_text').ellipsis();
59 });
60 </script>
61 <?= $this->endSection() ?>
62
63 <?= $this->section('main_content') ?>
64 <?php foreach($errors as $error){ ?>
65     <p class="shj_error"><?= $error ?></p>
66 <?php } ?>
67 <div style="height:15px;"></div>
68 <div class="gridster">
69     <?php $i = 0; ?>
70     <ul>
71         <li data-row=<?=_isset($wp[0])?>_<?=_isset($wp[0]['r'])?>_<?=_isset($wp[0]['l'])?>_<?=_isset($wp[0]['c'])?>_<?=_isset($wp[0]['y'])?>_>
72             <?=_isset($wp[0])?>_<?=_isset($wp[0]['x'])?>_<?=_isset($wp[0]['t'])?>_>
73             <div class="shj_widget">
74                 <div class="widget_title"><i class="fa_fa-calendar-o_fa-lg_color10"></i> Calendar</div>
75                 <div class="widget_scrollable_scroll-wrapper">
76                     <div class="scroll-content">
77                         <div class="widget_contents_container" id='calendar'></div>
78                     </div>
79                 </div>
80             </div>
81         </li>
82     <ul>
83

```

```

79        </div>
80    </li>
81
82    <li data-row=<?=isset($wp[1])?> data-col=<?=isset($wp[1])?> data-sizex=<?=isset($wp[1])?>
83        data-sizey=<?=isset($wp[1])?> data-sizex=<?=isset($wp[1])?> data-sizey=<?=isset($wp[1])?>>
84        <div class="shj_widget">
85            <div class="widget_title"><i class="fa fa-bell-o fa-lg_color2"></i>
86                Latest Notifications
87                <?php if ($user->level >= 2): ?>
88                    <a title="New Notification" href=<?=_site_url('notifications/add')?> class="pull-right">
89                        <i class="fa fa-plus_color11"></i>
90                </a>
91            </?php endif ?>
92        </div>
93        <div class="widget_scrollable_scroll-wrapper">
94            <div class="scroll-content">
95                <div class="widget_contents_container">
96                    <?php if (count($notifications) == 0): ?>
97                        <p style="text-align:center;">Nothing yet...</p>
98                    <?php endif ?>
99                    <?php foreach ($notifications as $notification): ?>
100                        <div class="notif" id="number<?=$notification['id']?>" data-id=<?=$notification['id']?>>
101                            <div class="notif_title" dir="auto">
102                                <span class="anchor_ttl_n"><?=$notification['title']?></span>
103                                <span class="notif_meta" dir="ltr">
104                                    <?=$notification['time']?>
105                                    <?php if ($user->level >= 2): ?>
106                                        <span class="anchor_edt_n">Edit</span>
107                                        <span class="anchor_del_n">Delete</span>
108                                    <?php endif ?>
109                                </span>
110                            </div>
111                            <div class="notif_text_latest" dir="auto"><?=$notification['text']?></div>
112                        </div>
113                    </?php endforeach ?>
114                </div>
115            </div>
116        </div>
117    </li>
118
119    <li data-row=<?=isset($wp[2])?> data-col=<?=isset($wp[2])?> data-sizex=<?=isset($wp[2])?>
120        data-sizey=<?=isset($wp[2])?> data-sizex=<?=isset($wp[2])?> data-sizey=<?=isset($wp[2])?>>
121    </li>
122 </ul>
123 </div>
124 <?= $this->endSection() ?>

```

Kode A.12: 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
11 <?= $this->section('other_assets') ?>
12 <link rel='stylesheet' type='text/css' href='<?=base_url("assets/snippet/jquery.snippet.css")?>'>
13 <link rel='stylesheet' type='text/css' href='<?=base_url("assets/snippet/themes/github.css")?>'>
14 <script type='text/javascript' src='<?=base_url("assets/snippet/jquery.snippet.js")?>'></script>
15 <script>
16 $(document).ready(function(){
17     // Syntax highlighting increases the page's height, and we need to update the scroll-bar
18     $('.problem_description').resize(function(){
19         $('.scroll-wrapper').nanoScroller();
20     });
21     // Fix text directions for rtl text
22     $.each($('.problem_description[dir="auto"]'), function(i, element){
23         if (getComputedStyle(element).direction == 'rtl')
24         {
25             $(element).css('direction', 'rtl');
26             $(element).parent('ul,.ol').css('direction', 'rtl');
27         }
28     });
29     // Syntax highlighting
30     $('#pre_code.language-c').parent().snippet('c', {style: shj.color_scheme});
31     $('#pre_code.language-cpp').parent().snippet('cpp', {style: shj.color_scheme});
32     $('#pre_code.language-python').parent().snippet('python', {style: shj.color_scheme});
33     $('#pre_code.language-java').parent().snippet('java', {style: shj.color_scheme});
34 });
35 </script>
36 <?= $this->endSection() ?>
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'])._?&gt;"&gt;&lt;i
47|   class="fa_fa-download_color1"&gt;&lt;/i&gt; PDF&lt;/a&gt;&lt;/span&gt;
48| &lt;/php endif ?&gt;
49| &lt;/php endif ?&gt;
50| &lt;?= $this-&gt;endSection() ?&gt;
51|
52|
53| &lt;?= $this-&gt;section('main_content') ?&gt;
54| &lt;div class="problem_description"&gt;
55|   &lt;?= $problem['description'] ?&gt;
56| &lt;/div&gt;
57|
58| &lt;div id="right_sidebar"&gt;
59|
60|   &lt;div class="problems_widget"&gt;
61|     &lt;p dir="auto"&gt;&lt;i class="fa_fa-file-text_fa-lg_color9"&gt;&lt;/i&gt; &lt;?= $description_assignment['name'] ?&gt;&lt;/p&gt;
62|
63|     &lt;?php if (count($all_problems) == 0): ?&gt;
64|       &lt;p style="text-align:center;"&gt;Nothing to show...&lt;/p&gt;
65|     &lt;?php endif ?&gt;
66|
67|     &lt;table class="sharif_table"&gt;
68|       &lt;thead&gt;
69|         &lt;tr&gt;
70|           &lt;th rowspan="2"#</th>
71|           <th rowspan="2">Problem</th>
72|           <th rowspan="2">Score</th>
73|           <th rowspan="2">Upload<br>Only</th>
74|         </tr>
75|       </thead>
76|       <?php foreach($all_problems as $one_problem): ?>
77|         <tr>
78|           <td><?= $problem['id'] == $one_problem['id'] ? '_class="hl"' : '' ?>>
79|             <td>
80|               <a dir="auto" href="=site_url("problems/".$description_assignment['id'].'/.'.$one_problem['id'])._?&gt;"&gt;&lt;?=
81|                 $one_problem['name'] ?&gt;&lt;/a&gt;
82|             &lt;/td&gt;
83|             &lt;td&gt;&lt;?= $one_problem['score'] ?&gt;&lt;/td&gt;
84|             &lt;td&gt;&lt;?= $one_problem['is_upload_only'] ? 'Yes' : 'No' ?&gt;&lt;/td&gt;
85|           &lt;/tr&gt;
86|         &lt;?php endforeach ?&gt;
87|       &lt;/table&gt;
88|
89|     &lt;?php if ($can_submit): ?&gt;
90|       &lt;div class="problems_widget"&gt;
91|         &lt;p&gt;&lt;i class="fa_fa-upload_fa-lg_color11"&gt;&lt;/i&gt; Submit&lt;/p&gt;
92|         &lt;?= form_open_multipart('submit') ?&gt;
93|         &lt;input type="hidden" name="problem" value="<?= $problem['id'] _?&gt;"/&gt;
94|
95|         &lt;p class="input_p"&gt;
96|           &lt;select id="languages" name="language" class="sharif_input_full-width" aria-label="Select_Language"&gt;
97|             &lt;option value="0" selected="selected"&gt;-- Select Language --&lt;/option&gt;
98|             &lt;?php foreach ($problem['allowed_languages'] as $l): ?&gt;
99|               &lt;option value="<?= $l _?&gt;"<?= $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|
113| </div>
114| <?= $this->endSection() ?> {# main_content #}

```

Kode A.13: 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')_?&gt;"/&gt;
13  &lt;link rel="stylesheet" type="text/css" href="<?=base_url('assets/styles/main.css')_?&gt;"/&gt;
14  &lt;script type="text/javascript" src="<?=base_url('assets/js/jquery-1.10.2.min.js')_?&gt;"/&gt;&lt;/script&gt;
15  &lt;!-- You can use google's_cdn_for_loading_jquery:--&gt;
16  &lt;!--&lt;script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"&gt;&lt;/script&gt;--&gt;
17  &lt;script type="text/javascript" src="<?=base_url('assets/js/jquery-ui-1.10.3.custom.min.js')_?&gt;"/&gt;&lt;/script&gt;
18  &lt;link rel="stylesheet" href="<?=base_url('assets/styles/flick/jquery-ui-1.10.3.custom.min.css')_?&gt;"/&gt;
19  &lt;script type="text/javascript" src="<?=base_url('assets/js/moment.min.js')_?&gt;"/&gt;&lt;/script&gt;
20  &lt;script type="text/javascript" src="<?=base_url('assets/js/jquery.hoverIntent.minified.js')_?&gt;"/&gt;&lt;/script&gt;
21  &lt;script type="text/javascript" src="<?=base_url('assets/js/jquery.cookie.js')_?&gt;"/&gt;&lt;/script&gt;
22  &lt;script type="text/javascript" src="<?=base_url('assets/js/jquery.ba-resize.min.js')_?&gt;"/&gt;&lt;/script&gt;
23  &lt;script type="text/javascript" src="<?=base_url('assets/noty/jquery.noty.js')_?&gt;"/&gt;&lt;/script&gt;
</pre

```

```

25  <script type="text/javascript" src="= base_url('assets/noty/layouts/center.js') ?&gt;"&gt;&lt;/script&gt;
26  &lt;script type="text/javascript" src="<?= base_url('assets/noty/layouts/bottomRight.js') ?&gt;"&gt;&lt;/script&gt;
27  &lt;script type="text/javascript" src="<?= base_url('assets/noty/themes/default.js') ?&gt;"&gt;&lt;/script&gt;
28
29  &lt;link rel='stylesheet' type='text/css' href='<?= base_url('assets/nano_scroller/nanoscroller.css') ?&gt;'&gt;
30  &lt;script type='text/javascript' src="<?= base_url('assets/nano_scroller/jquery.nanoscroller.min.js') ?&gt;"&gt;&lt;/script&gt;
31
32  &lt;link rel='stylesheet' type='text/css' href='<?= base_url('assets/font/font-awesome/css/font-awesome.min.css') ?&gt;'&gt;
33
34  &lt;/head&gt;
35  &lt;script&gt;
36  shj={};
37  shj.site_url_=<?= rtrim(site_url(), '/') ?&gt; '/';
38  shj.base_url_=<?= rtrim(base_url(), '/') ?&gt; '/';
39  shj.csrf_token_=<?= cookie('shjcsrfToken');
40  shj.offset_=moment('<?= shj.nowStr() ?&gt;').diff(moment());
41  shj.time_=moment();
42  shj.finish_time_=moment("<?= $finish_time ?&gt;");
43  shj.extra_time_=moment.duration("<?= $extra_time ?&gt;, 'seconds');
44 //notifications
45  shj.check_for_notifications_=false;
46  shj.notif_check_delay_=30;
47  shj.color_scheme_=github;
48 &lt;/script&gt;
49
50 &lt;script type="text/javascript" src="<?= base_url('assets/js/shj_functions.js') ?&gt;"&gt;&lt;/script&gt;
51
52 &lt;?= $this-&gt;renderSection('other_assets')?&gt;
53
54 &lt;body id="body"&gt;
55 &lt;a href="#page_title" class="skip_to_content"&gt;Skip to content&lt;/a&gt;
56
57 &lt;?= $this-&gt;include('templates/top_bar') ?&gt;
58 &lt;?= $this-&gt;include('templates/side_bar') ?&gt;
59 &lt;div id="main_container" class="scroll-wrapper"&gt;
60   &lt;div class="scroll-content"&gt;
61
62     &lt;div id="page_title"&gt;
63       &lt;i class="fa_&lt;?= $this-&gt;renderSection('icon') ?&gt;"&gt;&lt;/i&gt;
64       &lt;h1 dir="auto"&gt;&lt;?= $this-&gt;renderSection('title') ?&gt;&lt;/h1&gt;
65       &lt;?= $this-&gt;renderSection('title_menu') ?&gt;
66     &lt;/div&gt;
67
68     &lt;div id="main_content"&gt;
69       &lt;?= $this-&gt;renderSection('main_content') ?&gt;
70     &lt;/div&gt;
71   &lt;/div&gt;
72 &lt;/div&gt;
73 &lt;?= $this-&gt;renderSection('body_end') ?&gt;
74 &lt;/body&gt;
75 &lt;/html&gt;
</pre

```

## A.4 Kode lainnya

Berikut merupakan lampiran kode lainnya seperti **Filters**, aturan **Validation**, dan lainnya.

Kode A.14: 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 * 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 }
55

```

Kode A.15: CheckLoginandAjax.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 CheckLoginandAjax 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 A.16: 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 A.17: 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 ( strlen($str) == 0 OR (strlen($str) >= 6 && strlen($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 $str
121 * @return bool
122 */
123
124 public function lowercase($str)
125 {
126     return (strtolower($str) === $str);
127 }
128
129 // -----
130
131
132 public function _check_language($str)
133 {
134     if ($str=='0')
135         return FALSE;
136     if (in_array( strtolower($str),array('c', 'c++', 'python_2', 'python_3', 'java', 'zip', 'pdf', 'txt')))
137         return TRUE;
138     return FALSE;
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 }
```