

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

Digitally signed
by Pascal
Alfadian Nugroho

Pascal Alfadian, Nugroho, M.Comp.

Ketua Tim Penguji

Digitally
signed by
Husnul
Hakim

Husnul Hakim, M.T.

Anggota Tim Penguji

Digitally signed
by Elisati Hulu

Elisati Hulu, M.T.

Mengetahui,

Ketua Program Studi

Digitally signed
by Lionov

Lionov, Ph.D.

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 dijatuhkan 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 program studi 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. 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 assignments in the Computer Science program at Parahyangan Catholic University. Initially named Sharif Judge, it was an open-source software developed by Mohammad Javad. The purpose of SharIF Judge is to evaluate code written in various languages such as C, C++, Java, and Python. The software is built using the CodeIgniter 3 framework, which is a PHP-based framework. Subsequently, SharIF Judge was forked and modified to meet the specific requirements of the Computer Science department at UNPAR (Parahyangan Catholic University) for collecting student assignments and exams.

As of now, SharIF Judge continues to use CodeIgniter 3. However, this framework has entered the maintenance phase, receiving no further updates from its creators. Eventually, CodeIgniter 3 may become unusable and lose documentation on its official website. Therefore, in this final project, SharIF Judge will undergo conversion to the CodeIgniter 4 framework.

The conversion involves transferring files to a new directory and rewriting functions found in SharIF Judge. Additionally, some libraries are removed and replaced because they are no longer available in CodeIgniter 4. Libraries that are no longer present are substituted with new functions found in CodeIgniter 4 and PHP. This conversion does not alter the features of SharIF Judge, and thorough testing is conducted for each feature to ensure it functions the same way as the CodeIgniter 3-based SharIF Judge.

Testing is carried out by comparing the features of SharIF Judge based on CodeIgniter 4 with those of SharIF Judge based on CodeIgniter 3. The results of the testing confirm that there are no changes in the features between SharIF Judge based on CodeIgniter 4 and SharIF Judge based on CodeIgniter 3.

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

*Tugas Akhir ini dipersembahkan kepada Tuhan Yang Maha Esa,
Keluarga, dan rekan-rekan sesama mahasiswa.*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha 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 DNArtworks yang telah membantu dalam pengerjaan tugas akhir ini.
4. Kepada seluruh teman-teman yang telah menemani, membantu, dan memberi dukungan selama pengerjaan 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 yang terdapat 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

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxiii
DAFTAR KODE PROGRAM	xxviii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 <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	16
2.2.3 <i>Clean URLs</i>	17
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>	24
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>	33

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>	36
2.3.7	<i>Database</i>	37
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]	43
2.4.1	Struktur Aplikasi	43
2.4.2	<i>Routing</i>	43
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>	45
2.4.9	<i>Libraries</i>	45
2.4.10	<i>Helpers</i>	46
2.4.11	<i>Framework</i>	46
3	ANALISIS	47
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>	78
3.2.7	<i>Database</i>	78
3.2.8	<i>Error Handling</i>	78
3.2.9	<i>Helpers</i>	78
4	PERANCANGAN	81
4.1	app/Config	81
4.1.1	app/Config/App.php	81
4.1.2	app/Config/Autoload.php	81
4.1.3	app/Config/Cache.php	81
4.1.4	app/Config/Constants.php	81
4.1.5	app/Config/Cookie.php	81
4.1.6	app/Config/Database.php	81
4.1.7	app/Config/Email.php	81
4.1.8	app/Config/Encryption.php	81
4.1.9	app/Config/Filters.php	82
4.1.10	app/Config/Routes.php	82
4.1.11	app/Config/Secrets.example.php	82

4.1.12	app/Config/Security.php	82
4.1.13	app/Config/Session.php	82
4.1.14	app/Config/Validation.php	82
4.2	Controllers	82
4.2.1	Assignments.php	82
4.2.2	BaseController	83
4.2.3	Dashboard.php	83
4.2.4	Halloffame.php	83
4.2.5	Install.php	83
4.2.6	Login.php	83
4.2.7	Logs.php	83
4.2.8	Moss.php	84
4.2.9	Notification.php	84
4.2.10	Problems.php	84
4.2.11	Profile.php	84
4.2.12	Queue.php	84
4.2.13	Queueprocess.php	84
4.2.14	Rejudge.php	85
4.2.15	Scoreboard.php	85
4.2.16	Server_time.php	85
4.2.17	Settings.php	85
4.2.18	Submission.php	85
4.2.19	Submit.php	85
4.2.20	Users.php	85
4.3	Model	86
4.3.1	Assignment_model.php	86
4.3.2	Hof_model.php	86
4.3.3	Logs_model.php	86
4.3.4	Notifications_model.php	86
4.3.5	Queue_model.php	86
4.3.6	Scoreboard_model.php	87
4.3.7	Settings_model.php	87
4.3.8	Submit_model.php	87
4.3.9	User_model.php	87
4.3.10	User.php	87
4.4	View	87
4.5	<i>Filters</i>	88
4.6	<i>Routing</i>	88
4.7	<i>Libraries</i>	88
5	IMPLEMENTASI DAN PENGUJIAN	89
5.1	Lingkungan Implementasi dan Pengujian	89
5.2	Implementasi	90
5.2.1	Instalasi <i>CodeIgniter 4</i>	90
5.2.2	app/Config	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	public	110

5.2.10	restriced	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
6	KESIMPULAN DAN SARAN	123
6.1	Kesimpulan	123
6.2	Saran	123
	DAFTAR REFERENSI	125
A	KODE PROGRAM	127
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 <i>CodeIgniter 3</i> 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 <i>SharIF Judge</i> pada <i>CodeIgniter 3</i> ke <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	18
2.3	Tabel contoh masukan dan keluaran <i>Problem 1</i>	21
2.4	Tabel contoh masukan dan keluaran <i>Problem 2</i>	22
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 <i>index.php</i>	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	11
2.13	Contoh konfigurasi untuk menyimpan data pada <i>database</i>	12
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>	14
2.20	Contoh URL yang sudah dimetakan	14
2.21	Struktur aplikasi <i>SharIF Judge</i>	15
2.22	Kode untuk melakukah pengetesan fungsi	16
2.23	Contoh <i>path</i> pada halaman <i>index.php</i>	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>	22
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>	23
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>	26
2.41	Kode <i>runcode.sh</i> sebelum ditambahkan	27
2.42	Kode <i>runcode.sh</i> setelah ditambahkan	27

2.43	Kode <i>runcode.sh</i> awal	27
2.44	Contoh kode untuk membatasi jumlah proses	27
2.45	Kode <i>runcode.sh</i> awal	28
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	31
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>	32
2.55	Contoh mengakses data menggunakan <i>IncomingRequest</i>	32
2.56	Contoh konfigurasi menggunakan <i>namespace PSR-4</i> .	33
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> .	35
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.	36
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> .	37
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> .	40
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.	42
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> .	45
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>	75
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 CheckLoginandisAjax.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	<i>Assignments.php</i>	127
A.2	<i>BaseController.php</i>	134
A.3	<i>Submissions.php</i>	134
A.4	<i>Submit.php</i>	140
A.5	<i>AssignmentModel.php</i>	146
A.6	<i>ScoreboardModel.php</i>	151
A.7	<i>UserModel.php</i>	153
A.8	<i>add_assignment.php</i>	159
A.9	<i>settings.php</i>	162
A.10	<i>assignments.php</i>	165
A.11	<i>dashboard.php</i>	166
A.12	<i>problems.php</i>	167
A.13	<i>base.php</i>	168
A.14	<i>CheckInstallAndLogin.php</i>	169
A.15	<i>CheckLoginandisAjax.php</i>	170
A.16	<i>CheckLoginandLevelAdmin.php</i>	170
A.17	<i>Validation.php</i>	171

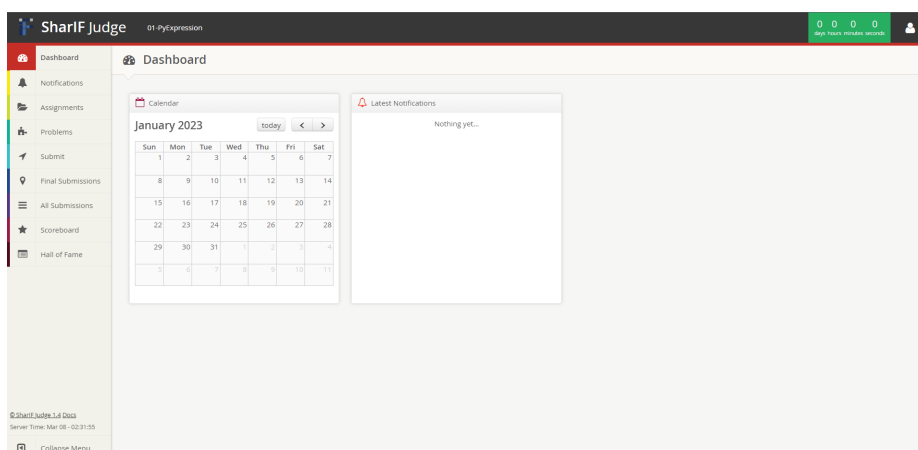
BAB 1

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, terdapat kekurangan pada pengecekan secara manual 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 program studi Informatika Universitas Katolik Parahyangan bernama *SharIF Judge*. Gambar 1.1 menunjukkan tampilan perangkat lunak *SharIF Judge* yang terletak pada Universitas Katolik Parahyangan.



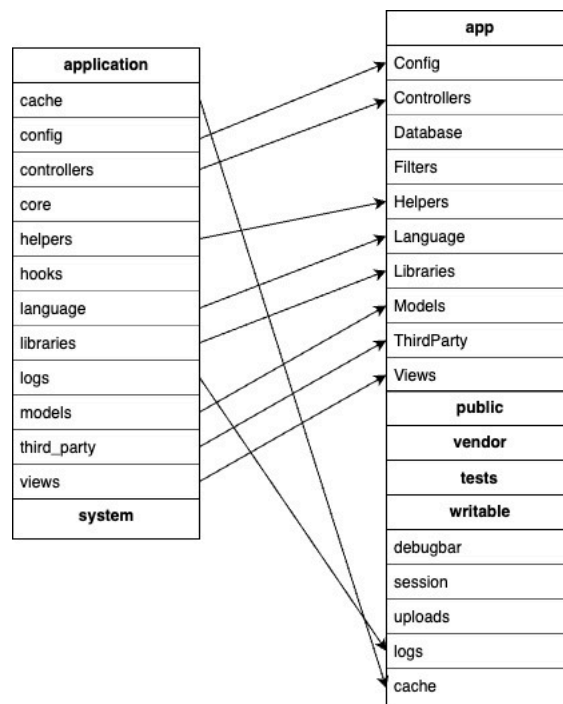
Gambar 1.1: Tampilan halaman *SharIF Judge*

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*.

Sharif Judge kemudian di *fork* dan dimodifikasi menjadi *SharIF Judge* dengan penambahan fungsi sesuai kebutuhan Informatika UNPAR untuk mengumpulkan tugas dan ujian mahasiswa. *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 yang membagi *file* menjadi 3 buah yaitu *Model*, *View*, dan *Controller* atau disebut MVC. 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*¹ sehingga tidak mendapatkan pembaharuan lebih lanjut dari pembentuknya. *CodeIgniter 3* pada akhirnya tidak dapat digunakan kembali dan 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 *CodeIgniter 3* 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*. Konversi dilakukan agar *SharIF Judge* dapat digunakan kembali apabila *CodeIgniter 3* sudah tidak mendapat dukungan.

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

1.2 Rumusan Masalah

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

1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

- Melakukan konversi dengan mengubah kode sesuai dengan standar *CodeIgniter 4*.
- 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.