

SKRIPSI

IMPLEMENTASI EDITOR KODE PADA SHARIF JUDGE



Nicholas Aditya Halim

NPM: 2017730018

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2022**

UNDERGRADUATE THESIS

CODE EDITOR IMPLEMENTATION ON SHARIF JUDGE



Nicholas Aditya Halim

NPM: 2017730018

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2022**

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 CodeIgniter 3	5
2.1.1 Model-View-Controller	5
2.1.2 URL CodeIgniter	7
2.2 Twig	7
2.3 Bash	8
2.4 PDF.js	8
2.5 Ace	9
3 ANALISIS	11
3.1 Analisis Sistem Kini	11
3.1.1 Fitur SharIF Judge	11
3.1.2 Model, View, Controller	17
3.1.3 Antrean Penilaian Kode	26
3.1.4 Penyimpanan <i>File</i> Jawaban	26
3.2 Analisis Sistem Usulan	26
4 PERANCANGAN	27
4.1 Rancangan Antarmuka	27
4.2 Menampilkan soal	28
4.3 Editor Kode	28
4.4 Menyimpan dan Memuat Kode	28
4.5 Menjalankan Kode dengan Tes Kasus	29
4.6 Mengumpulkan Kode dari Editor	29
5 IMPLEMENTASI DAN PENGUJIAN	31
5.1 Lingkungan Implementasi dan Pengujian	31
5.2 Implementasi	32
5.2.1 Tampilan Antarmuka	32
5.2.2 Menampilkan Soal	32
5.2.3 Editor Kode	33

5.2.4	Menyimpan dan Memuat Kode	33
5.2.5	Menjalankan Kode dengan Tes Kasus	33
5.2.6	Mengumpulkan Kode dari Editor	34
5.3	Pengujian	34
5.3.1	Pengujian Fungsional	34
5.3.2	Pengujian Eksperimental	34
6	KESIMPULAN DAN SARAN	39
6.1	Kesimpulan	39
6.2	Saran	39
	DAFTAR REFERENSI	41
	A KODE PROGRAM	43
	B HASIL EKSPERIMEN	59

DAFTAR GAMBAR

2.1	<i>Flow Chart</i> CodeIgniter	5
3.1	Halaman Dashboard	11
3.2	Halaman Settings	12
3.3	Halaman Users	12
3.4	Halaman Notifications	13
3.5	Halaman Assignments	13
3.6	Halaman Problems	14
3.7	Halaman Submit	14
3.8	Halaman Final Submissions	15
3.9	Halaman All Submissions	15
3.10	Halaman Scoreboard	16
3.11	Halaman Hall of Fame	16
3.12	Halaman 24-Hour Log	17
3.13	<code>side_bar.twig</code>	21
3.14	<code>submit.twig</code>	22
3.15	<code>error_404.php</code>	22
4.1	Rancangan antarmuka halaman Submit	27
5.1	Antarmuka halaman Submit	32
5.2	Halaman Submit	35
5.3	Tampilan unggah <i>file</i>	35
5.4	Tampilan IDE	36

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Online judge adalah sebuah sistem *online* yang berfungsi untuk mengevaluasi kode program yang dikumpulkan oleh pengguna. Kode program kemudian dikompilasi dan diuji pada lingkungan yang serupa. *Online judge* sering kali digunakan dalam sistem pemrograman kompetitif dan edukasi pemrograman [1].

Sharif Judge adalah sebuah *online judge* untuk bahasa pemrograman C, C++, Java dan Python. Antarmuka web Sharif Judge dibangun menggunakan PHP dengan *framework* CodeIgniter, disertai *backend* menggunakan Bash [2].

SharIF Judge (dengan IF kapital) adalah modifikasi dari Sharif Judge yang disesuaikan untuk kebutuhan spesifik Teknik Informatika Unpar. SharIF Judge digunakan pada beberapa mata kuliah pemrograman untuk mempermudah proses pengumpulan dan penilaian kode program [3].

Dengan adanya situasi pandemi Covid-19, seluruh kegiatan kuliah wajib dilaksanakan secara *online*. Pada umumnya, kegiatan praktikum dan ujian pada mata kuliah pemrograman Teknik Informatika Unpar dapat diawasi secara langsung oleh dosen dan asisten dosen di lab komputer. Namun, pengawasan menjadi lebih sulit untuk dilakukan saat kuliah dilaksanakan secara *online*. Diperlukan sebuah cara untuk mengawasi mahasiswa selama kuliah *online* berlangsung.

Integrated Development Environment (IDE) adalah sebuah aplikasi editor teks dengan fitur yang membantu penggunaannya untuk menulis kode dengan lebih cepat dan efisien. Sebuah IDE pada umumnya memiliki kemampuan untuk mengedit, mengompilasi, menjalankan, dan menguji kode program [4]. Pada umumnya, mahasiswa menggunakan aplikasi IDE seperti Netbeans untuk membuat kode program yang kemudian diunggah ke SharIF Judge untuk dinilai.

Pada skripsi ini akan diimplementasikan editor kode pada SharIF Judge. SharIF Judge sebelumnya sudah memiliki kemampuan untuk mengompilasi, menjalankan, dan menguji kode. Dengan implementasi editor kode, SharIF Judge dapat menjadi sebuah IDE yang mampu memfasilitasi proses penulisan kode, lalu mengompilasi, menjalankan, dan mengujinya.

Dengan implementasi IDE berbasis web pada SharIF Judge, selanjutnya dapat ditambahkan fitur yang dapat membantu pengawasan terhadap mahasiswa selama kegiatan kuliah seperti merekam ketikan dan mendeteksi ketika mahasiswa membuka *tab* atau aplikasi lain.

Perangkat lunak diuji pada kuliah Dasar-dasar Pemrograman semester ganjil 2021/2022 Teknik Informatika Unpar. Pada kuliah ini terdapat 2 alamat *judge* yang digunakan, yaitu <http://daspro.labftis.net> untuk latihan, dan <http://daspro-quiz.labftis.net> untuk kuis.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada skripsi ini adalah sebagai berikut:

- Bagaimana mengimplementasikan *Integrated Development Environment* sehingga mahasiswa dapat mengetik dan menjalankan kode dalam SharIF Judge?
- Bagaimana tanggapan pengguna terhadap implementasi *Integrated Development Environment* pada SharIF Judge?

1.3 Tujuan

Tujuan yang ingin dicapai skripsi ini adalah sebagai berikut:

- Mengimplementasikan *Integrated Development Environment* sehingga mahasiswa dapat mengetik dan menjalankan kode dalam SharIF Judge.
- Mendapatkan umpan balik dari tanggapan pengguna terhadap implementasi *Integrated Development Environment* pada SharIF Judge.

1.4 Batasan Masalah

Batasan masalah pada skripsi ini adalah sebagai berikut:

- Perangkat lunak skripsi ini hanya akan diuji pada *judge* latihan kuliah Dasar-dasar Pemrograman.

1.5 Metodologi

Metodologi pengerjaan skripsi ini adalah sebagai berikut:

1. Melakukan studi mengenai komponen yang diperlukan untuk membuat IDE berbasis web.
2. Mempelajari struktur SharIF Judge.
3. Merancang IDE berbasis web untuk SharIF Judge.
4. Mengimplementasikan IDE berbasis web pada SharIF Judge.
5. Melakukan pengujian dan eksperimen.
6. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematika pembahasan skripsi ini adalah sebagai berikut:

- Bab 1 Pendahuluan
Membahas latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.
- Bab 2 Landasan Teori
Membahas teori-teori yang digunakan dalam skripsi ini, yaitu CodeIgniter 3, Twig, Shell Script, PDF.js, dan Ace.
- Bab 3 Analisis
Membahas analisis terhadap perangkat lunak SharIF Judge.

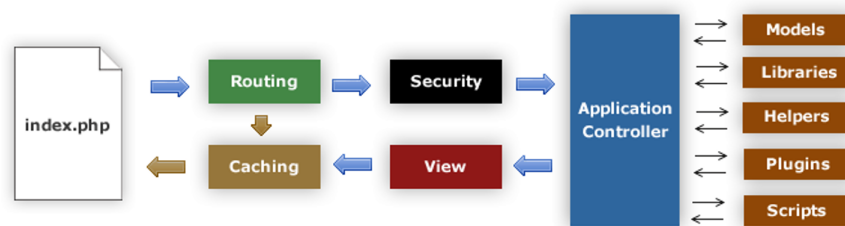
-
- 1 • Bab 4 Perancangan
 - 2 Membahas perancangan fitur yang diimplementasikan pada SharIF Judge.
 - 3 • Bab 5 Implementasi dan Pengujian
 - 4 Membahas implementasi fitur pada SharIF Judge dan pengujian yang dilakukan.

BAB 2

LANDASAN TEORI

2.1 CodeIgniter 3

CodeIgniter adalah sebuah *framework* untuk membangun situs web menggunakan PHP. Tujuan utamanya adalah untuk mempercepat pembuatan proyek dengan menyediakan *library* yang lengkap untuk fungsi-fungsi yang umum digunakan, serta antarmuka yang sederhana dan struktur yang logis untuk mengakses *library* tersebut [5].



Gambar 2.1: *Flow Chart* CodeIgniter

Gambar 2.1 mengilustrasikan bagaimana data mengalir pada sistem CodeIgniter.

1. *File* index.php berfungsi sebagai *front controller*, menginisialisasi *resource* utama untuk menjalankan CodeIgniter.
2. Router meneliti *request* HTTP dan menentukan apa yang harus dilakukan.
3. Jika terdapat *file cache*, maka langsung dikirimkan ke *browser*.
4. Sebelum *controller* dimuat, seluruh *request* HTTP dan data dari user disaring terlebih dahulu untuk keamanan.
5. *Controller* memuat *model*, *library* utama, dan *resource* lainnya yang diperlukan.
6. *View* akhir lalu dikirim ke browser untuk dilihat. *Cache* akan dibuat terlebih dahulu bila diaktifkan.

2.1.1 Model-View-Controller

CodeIgniter menggunakan pola arsitektur MVC (*Model-View-Controller*) sebagai dasarnya. MVC memisahkan proses logika aplikasi dari presentasi. Dengan demikian, halaman web dapat memuat sedikit *script* karena presentasinya terpisah dari *scripting* PHP.

Model

Model merepresentasikan struktur data. Biasanya *model* memiliki fungsi-fungsi yang membantu dalam mengambil, memasukkan, dan memperbarui informasi pada *database*. Pada CodeIgniter, *model* adalah sebuah kelas yang mengekstensi `CI_Model` dan terletak di direktori `application/models/`.

Kode 2.1: Contoh *model*

```

5
61 class Blog_model extends CI_Model {
72
83     public $title;
94     public $content;
105    public $date;
116
127    public function get_last_ten_entries()
138    {
149        $query = $this->db->get('entries', 10);
150        return $query->result();
161    }
172
183    public function insert_entry()
194    {
205        $this->title   = $_POST['title']; // please read the below note
216        $this->content = $_POST['content'];
227        $this->date    = time();
238
249        $this->db->insert('entries', $this);
250    }
261
272    public function update_entry()
283    {
294        $this->title   = $_POST['title'];
305        $this->content = $_POST['content'];
316        $this->date    = time();
327
338        $this->db->update('entries', $this, array('id' => $_POST['id']));
349    }
350
361 }

```

Kode 2.1 merupakan contoh sebuah kelas *model* pada CodeIgniter. Kelas tersebut mengekstensi `CI_Model` dan memiliki fungsi untuk mengambil, memasukkan, dan memperbarui *database*.

View

View adalah informasi yang ditampilkan kepada pengguna. Pada CodeIgniter, *view* merupakan sebuah halaman web atau sebagian dari halaman web yang terletak di direktori `application/view/`.

Kode 2.2: Contoh *view*

```

43
44 1 <html>
45 2 <head>
46 3     <title>My Blog</title>
47 4 </head>
48 5 <body>
49 6     <h1>Welcome to my Blog!</h1>
50 7 </body>
51 8 </html>

```

Kode 2.2 merupakan contoh sebuah *view*. *View* pada CodeIgniter harus dipanggil melalui *Controller* dan tidak pernah dipanggil secara langsung.

Controller

Controller adalah perantara dari *model* dan *view*, serta *resource* lainnya yang diperlukan untuk memproses *request* HTTP dan menghasilkan sebuah halaman web. Pada CodeIgniter, *controller* adalah sebuah kelas yang mengekstensi `CI_Controller` dan terletak di direktori `application/controllers/`.

Kode 2.3: Contoh *controller*

```

1
21 class Blog extends CI_Controller {
32
43     public function index()
54     {
65         echo 'Hello_World!';
76     }
87
98     public function comments()
109    {
110        echo 'Look_at_this!';
121    }
132 }

```

Kode 2.1 merupakan contoh sebuah kelas *controller* pada CodeIgniter. Kelas tersebut meng-
 tensi `CI_Controller` dan memiliki fungsi `index()` dan `comments()`. Fungsi `index()` akan dipanggil
 secara otomatis jika tidak ada fungsi lain yang dipanggil.

Kode 2.4: Contoh memuat *model* dan menampilkan *view*

```

18
191 class Blog_controller extends CI_Controller {
202
213     public function blog()
224     {
235         $this->load->model('blog');
246
257         $data['query'] = $this->blog->get_last_ten_entries();
268
279         $this->load->view('blog', $data);
280     }
291 }
302 }

```

Pada CodeIgniter, *model* dan *view* hanya dapat dimuat melalui *controller*. Pada contoh kode 2.4,
 fungsi `blog()` pada *controller* memuat *model* untuk mengambil data dari *database*, lalu menampilkan
view yang memuat data tersebut.

2.1.2 URL CodeIgniter

URL pada CodeIgniter menggunakan *segment-based approach* yang dirancang untuk lebih mudah
 dibaca oleh *search engine* dan manusia. Berikut ini adalah contoh sebuah URL pada CodeIgniter:

`example.com/class/function/ID`

- Bagian pertama, `class` merepresentasikan kelas *controller* yang akan dipanggil.
- Bagian kedua, `function` merepresentasikan fungsi yang akan dipanggil.
- Bagian ketiga dan seterusnya, `ID` merepresentasikan variabel yang akan digunakan.

2.2 Twig

Twig adalah sebuah *template engine* untuk PHP. Sebuah *template* Twig memuat *variable* atau
expression yang nantinya akan diubah menjadi *value* saat *template* dievaluasi, serta *tag* yang
 mengontrol logika *template* [6].

Kode 2.5: Contoh *template* Twig

```

46
471 <!DOCTYPE html>
482 <html>
493     <head>
504         <title>My Webpage</title>
515     </head>
526     <body>

```

```

17     <ul id="navigation">
18         {% for item in navigation %}
19         <li><a href="{{_item.href_}}">{{ item.caption }}</a></li>
20         {% endfor %}
21     </ul>
22
23     <h1>My Webpage</h1>
24     {{ a_variable }}
25 </body>
26 </html>

```

Kode 2.5 merupakan contoh sebuah template Twig. Terdapat dua jenis *delimiter*, yaitu `{% ... %}` dan `{{ ... }}`. *Delimiter* `{% ... %}` digunakan untuk menjalankan *statement* seperti `for` dan `if`, sementara *delimiter* `{{ ... }}` digunakan untuk menampilkan nilai dari *variable* atau *expression*.

2.3 Bash

Shell adalah sebuah program pada sistem operasi Unix yang menerima perintah tertulis dan mengirimnya ke sistem operasi untuk dijalankan. Pada umumnya, perangkat Linux menggunakan program bernama Bourne Again SHell (Bash) sebagai *shell*. Bash merupakan program yang disempurnakan dari *shell* Unix pertama yang diciptakan oleh Steve Bourne [7].

Shell script adalah sebuah *file* yang menyimpan rangkaian perintah. *Shell* akan membaca *file* tersebut dan menjalankan rangkaian perintah seperti jika perintah tersebut dimasukkan secara langsung pada *command line*. Keunikan dari *shell* adalah kemampuannya sebagai *command line interface* dan sebagai *scripting language interpreter*. Artinya, hal yang dapat dilakukan melalui *command line* dapat dilakukan sebagai *script*, dan hal yang dapat dilakukan sebagai *script* dapat dilakukan melalui *command line*.

2.4 PDF.js

PDF.js adalah sebuah library JavaScript yang berfungsi untuk menampilkan *file* Portable Document Format (PDF) menggunakan HTML5 *Canvas* [8]. PDF.js terdiri dari 3 *layer*:

- **Core** merupakan bagian dimana proses *parse* dan *interpret* dilakukan terhadap *binary* PDF.
- **Display** mengambil *layer core* sebagai API yang lebih mudah digunakan untuk menampilkan PDF dan mengambil informasi lainnya dari sebuah dokumen.
- **Viewer** membangun *layer display* sebagai halaman website dengan *user interface* yang dapat ditampilkan di browser.

Kode 2.6: Contoh kode untuk menggunakan PDF.js

```

34
35 1 <!DOCTYPE html>
36 2 <html>
37 3   <iframe src="/web/viewer.html?file=sample.pdf"></iframe>
38 4 </html>

```

Salah satu cara untuk menampilkan *file* PDF menggunakan PDF.js adalah dengan *embed* *layer viewer* yang sudah tersedia melalui `web/viewer.js` pada sebuah `iframe`. Kode 2.6 merupakan contoh kode *embed* PDF.js untuk menampilkan sebuah file PDF contoh `sample.pdf`.

2.5 Ace

Ace adalah sebuah library JavaScript yang berfungsi sebagai *code editor*. Ace memiliki fitur-fitur yang dapat ditemukan di *code editor* pada umumnya [9]. Berikut ini merupakan beberapa fitur utama dari Ace:

- *Syntax highlighting* untuk lebih dari 110 bahasa pemrograman.
- *Indent* dan *outdent* otomatis.
- Kemampuan *cut*, *copy*, dan *paste*.
- *Drag and drop* teks menggunakan mouse.

Berikut ini adalah beberapa kelas yang terdapat pada Ace:

- **Ace**

Kelas utama yang digunakan mempersiapkan Ace pada browser. Salah satu fungsi yang dimiliki:

- `edit(String | DOMELEMENT el)`
Embed Ace pada elemen yang disediakan.

- **Anchor**

Menangani posisi *pointer* pada dokumen.

- **BackgroundTokenizer**

Bekerja di latar belakang untuk melakukan tokenisasi pada dokumen saat ini dan menyimpan baris yang sudah ditokenisasi sebagai *cache*.

- **Document**

Menyimpan teks dari dokumen.

- **EditSession**

Menyimpan seluruh *state* untuk **Editor** dan menyediakan cara untuk mengubahnya dengan mudah. Beberapa fungsi yang dimiliki:

- `getMode()`
Mengembalikan mode *syntax highlighting* editor yang sedang digunakan.
- `setMode()`
Mengubah mode *syntax highlighting* editor.

- **Editor**

Entry point utama untuk seluruh kegunaan Ace. Beberapa fungsi yang dimiliki:

- `getReadOnly()`
Mengembalikan `true` jika editor sedang menggunakan pengaturan *read-only*.
- `getTheme()`
Mengembalikan alamat tema editor yang sedang digunakan.
- `getValue()`
Mengembalikan isi teks editor.
- `setReadOnly(Boolean readOnly)`
Mengubah pengaturan *read-only*.
- `setTheme(String style)`
Mengubah tema editor.
- `setValue(String val, Number cursorPos)`
Mengubah isi teks editor.

- 1 • **Range**
2 Mengindikasi sebuah daerah pada editor.
- 3 • **Scrollbar**
4 Menangani *scrollbar* editor.
- 5 • **Search**
6 Menangani seluruh operasi pencarian teks pada dokumen.
- 7 • **Selection**
8 Menyimpan posisi kursor dan seleksi teks pada editor.
- 9 • **TokenIterator**
10 Menyediakan fungsi untuk membaca dokumen sebagai aliran token.
- 11 • **Tokenizer**
12 Menerima sejumlah aturan dan membuat **Tokenizer**.
- 13 • **UndoManager**
14 Menangani fungsi *undo* pada editor.
- 15 • **VirtualRenderer**
16 Menggambar tampilan yang terlihat di layar.

Kode 2.7: Contoh kode untuk menggunakan Ace

```

17 <!DOCTYPE html>
18 <html>
19 <head>
20 <title>ACE in Action</title>
21 </head>
22 <body>
23 <div id="editor">
24   function foo(items) {
25     var x = "All this is syntax highlighted";
26     return x;
27   }
28 </div>
29 <script src="/ace-builds/src-noconflict/ace.js" type="text/javascript" charset="utf-8"></script>
30 <script>
31   var editor = ace.edit("editor");
32   editor.setTheme("ace/theme/monokai");
33   editor.session.setMode("ace/mode/javascript");
34 </script>
35 </body>
36 </html>

```

41 Kode 2.7 merupakan contoh kode untuk menempatkan editor Ace pada sebuah elemen `div`
 42 dengan id `editor`. Terdapat berbagai konfigurasi pada Ace, pada contoh ini digunakan tema
 43 *monokai* dan mode *syntax highlighting* untuk JavaScript.

BAB 3

ANALISIS

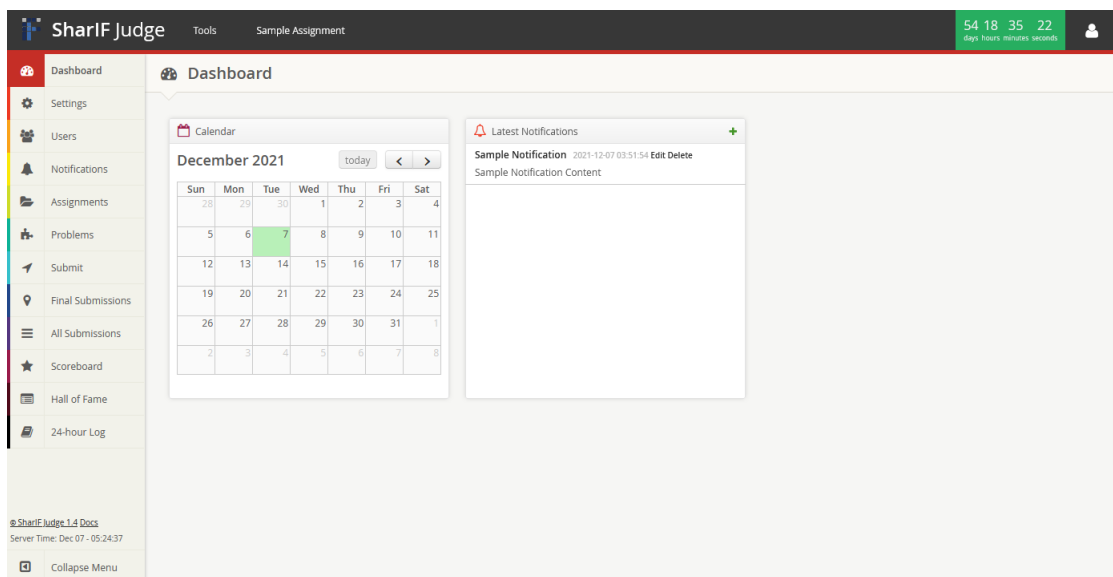
3.1 Analisis Sistem Kini

SharIF Judge adalah sebuah *online judge* dengan fungsi utama untuk mengevaluasi kode program yang dikumpulkan oleh pengguna secara otomatis. SharIF Judge digunakan pada beberapa mata kuliah pemrograman Teknik Informatika Unpar untuk mempermudah proses pengumpulan dan penilaian kode program. Antarmuka web Sharif Judge dibangun menggunakan PHP dengan *framework* CodeIgniter, disertai *backend* menggunakan Bash [3].

3.1.1 Fitur SharIF Judge

Halaman dan tampilan yang tersedia pada pengguna SharIF Judge bergantung pada *role* akun yang digunakan pengguna tersebut. Pada bagian ini, *role* akun yang digunakan adalah *admin*. Berikut ini adalah halaman yang terdapat pada SharIF Judge dengan fitur dan kegunaannya:

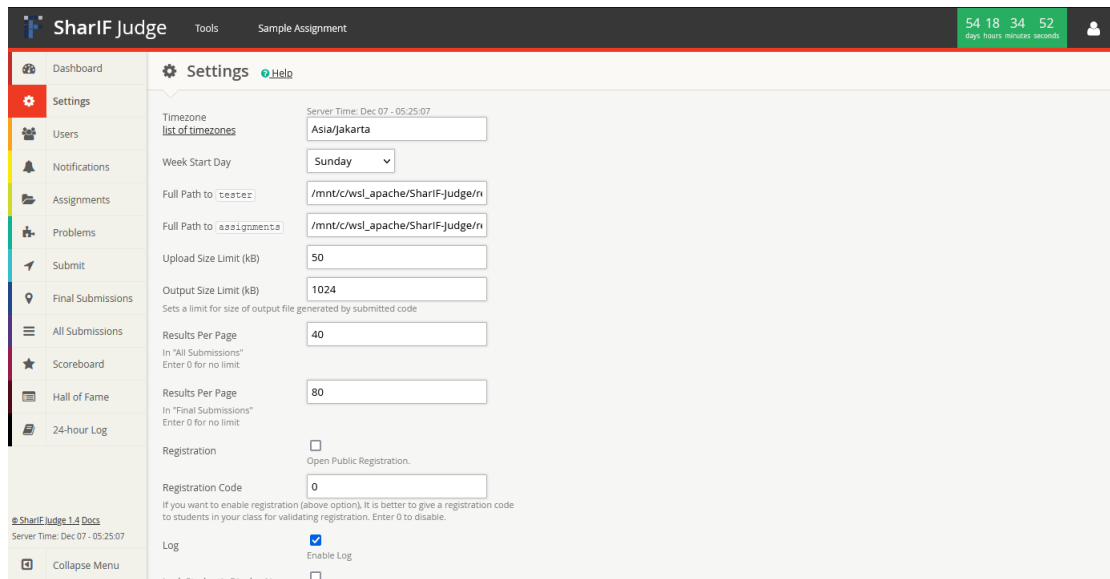
Dashboard



Gambar 3.1: Halaman Dashboard

Gambar 3.1 menunjukkan halaman Dashboard. Pada halaman ini terdapat kalender yang menunjukkan durasi setiap *assignment* dan daftar notifikasi.

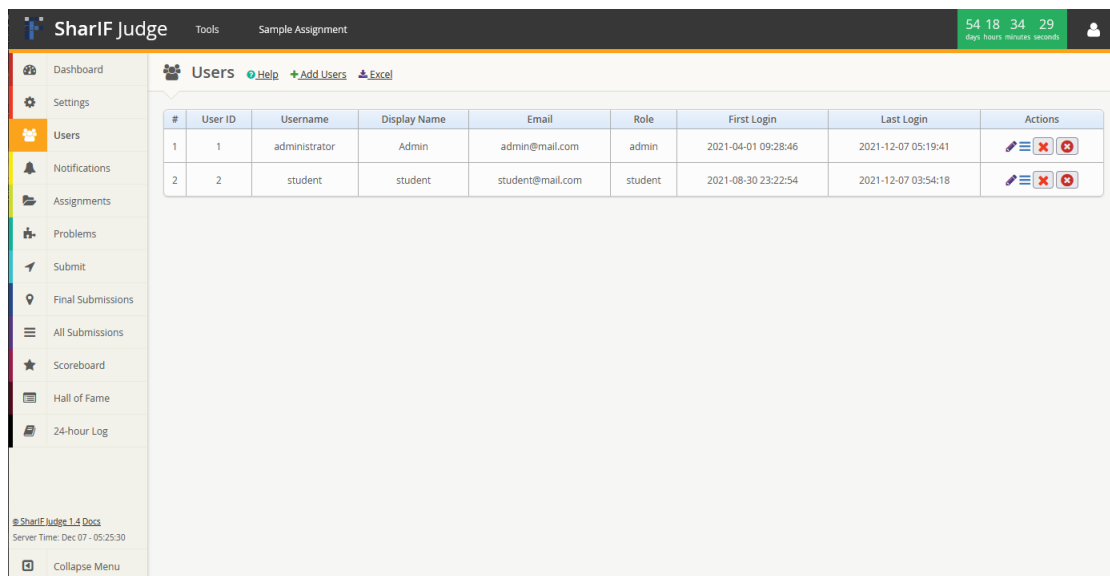
1 Settings



Gambar 3.2: Halaman Settings

2 Gambar 3.2 menunjukkan halaman Settings. Pada halaman ini terdapat berbagai pengaturan yang
3 ada pada SharIF Judge.

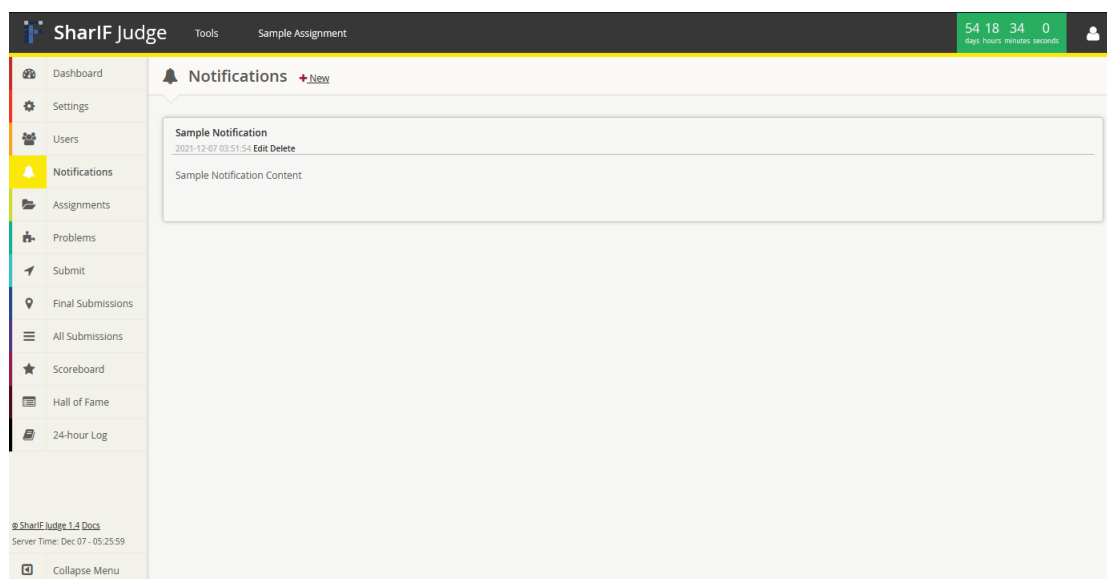
4 Users



Gambar 3.3: Halaman Users

5 Gambar 3.3 menunjukkan halaman Users. Pada halaman ini terdapat *list* seluruh pengguna yang
6 terdaftar pada SharIF Judge. Pengguna juga dapat membuat, mengubah, dan menghapus pengguna.

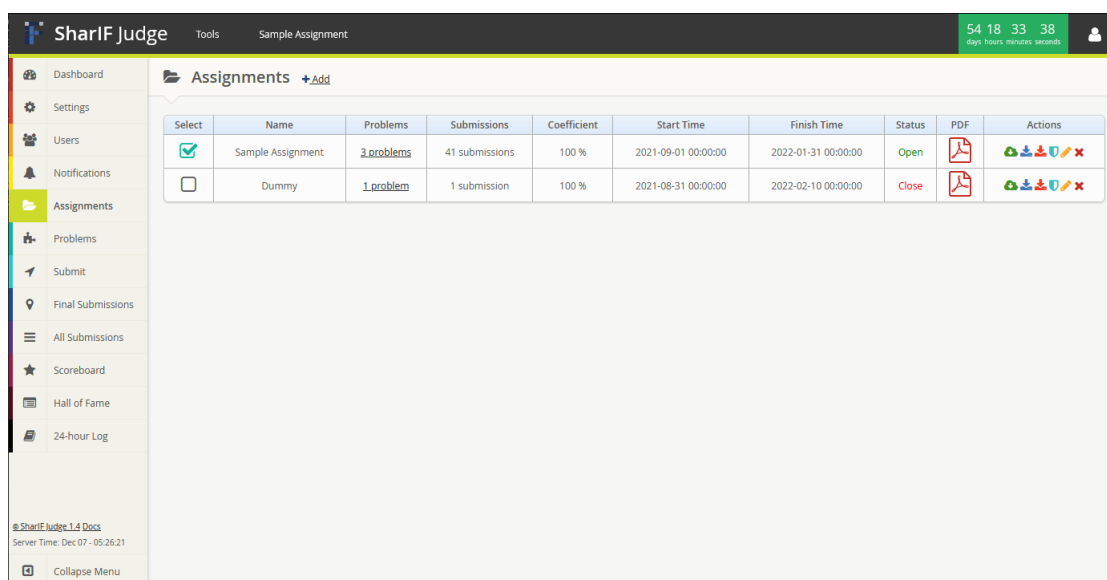
1 Notifications



Gambar 3.4: Halaman Notifications

- 2 Gambar 3.4 menunjukkan halaman Notifications. Pada halaman ini terdapat *list* seluruh notifikasi.
- 3 Pengguna juga dapat membuat, mengubah, dan menghapus notifikasi.

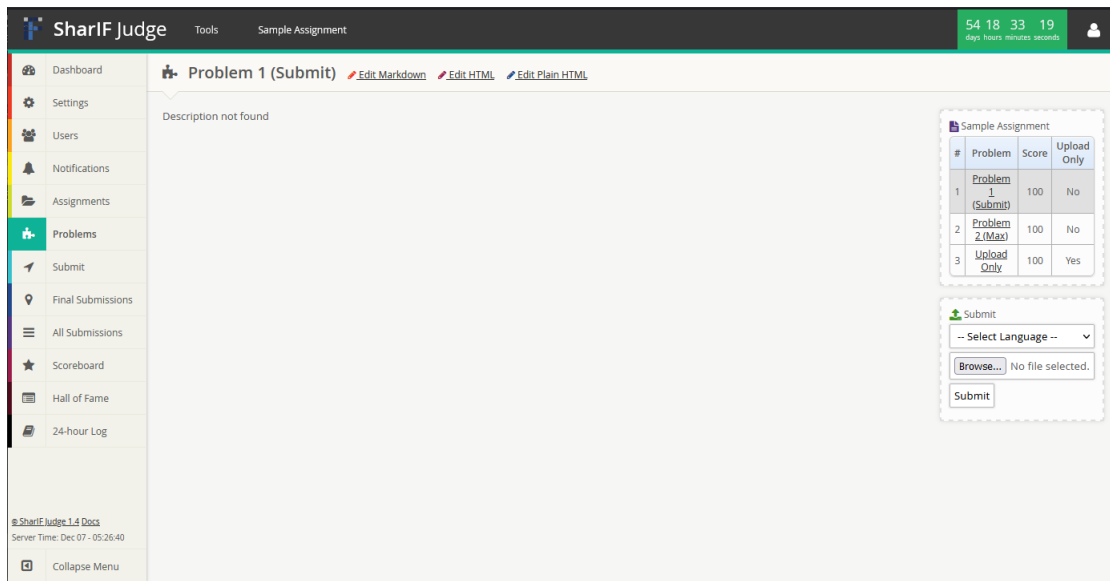
4 Assignments



Gambar 3.5: Halaman Assignments

- 5 Gambar 3.5 menunjukkan halaman Assignments. Pada halaman ini terdapat *list* seluruh *assignment*.
- 6 Pengguna juga dapat membuat, mengubah, dan menghapus *assignment*. Salah satu *assignment*
- 7 pada halaman ini harus dipilih untuk dapat menggunakan beberapa fitur lainnya pada SharIF
- 8 Judge. Soal dalam bentuk PDF juga dapat diunduh melalui halaman ini.

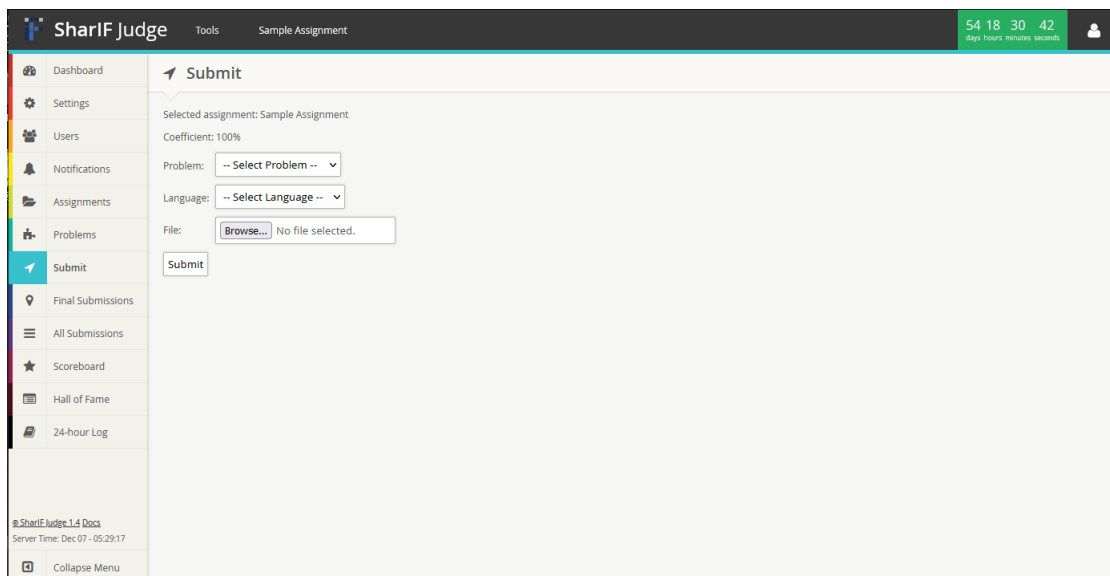
1 Problems



Gambar 3.6: Halaman Problems

2 Gambar 3.6 menunjukkan halaman Problems. Pada halaman ini terdapat detil dari setiap *problem*
3 dari *assignment* yang dipilih. Pengguna juga dapat mengunggah file untuk dikumpulkan sebagai
4 *submission* untuk *problem* yang dipilih.

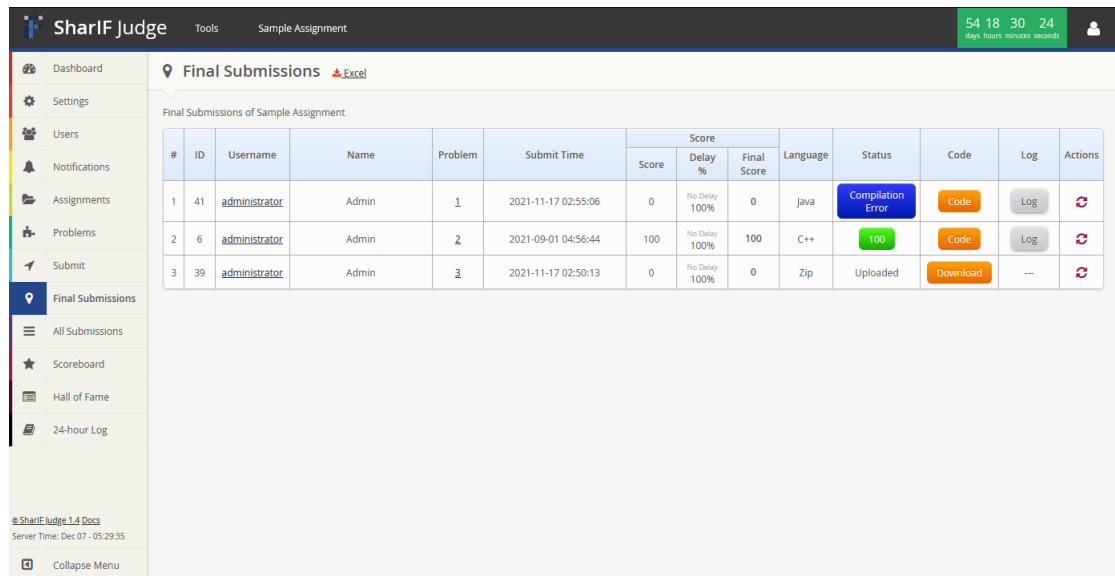
5 Submit



Gambar 3.7: Halaman Submit

6 Gambar 3.7 menunjukkan halaman Submit. Pada halaman ini, pengguna dapat mengunggah file
7 untuk dikumpulkan sebagai *submission* dari *problem* yang dipilih.

1 Final Submissions

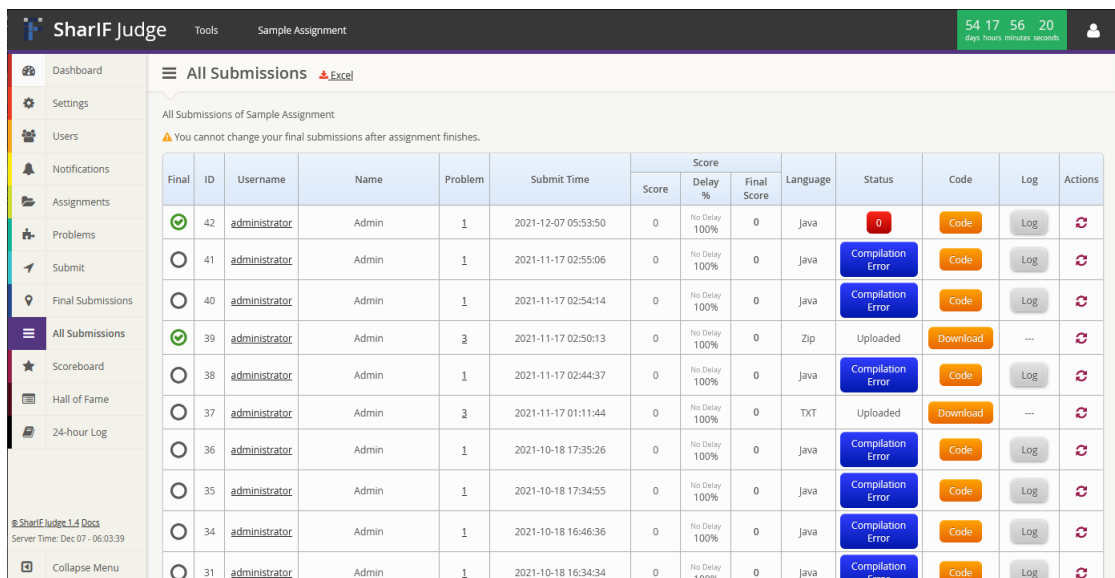


#	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
1	41	administrator	Admin	1	2021-11-17 02:55:06	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
2	6	administrator	Admin	2	2021-09-01 04:56:44	100	No Delay 100%	100	C++	100	Code	Log	
3	39	administrator	Admin	3	2021-11-17 02:50:13	0	No Delay 100%	0	Zip	Uploaded	Download	---	

Gambar 3.8: Halaman Final Submissions

- 2 Gambar 3.8 menunjukkan halaman Final Submissions. Pada halaman ini, terdapat *list* seluruh
 3 *final submission* untuk *assignment* yang dipilih. Pengguna juga dapat melihat *file* atau kode yang
 4 diunggah, dan nilai yang didapatkannya.

5 All Submissions



Final	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
✓	42	administrator	Admin	1	2021-12-07 05:53:50	0	No Delay 100%	0	Java	0	Code	Log	
○	41	administrator	Admin	1	2021-11-17 02:55:06	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
○	40	administrator	Admin	1	2021-11-17 02:54:14	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
✓	39	administrator	Admin	3	2021-11-17 02:50:13	0	No Delay 100%	0	Zip	Uploaded	Download	---	
○	38	administrator	Admin	1	2021-11-17 02:44:37	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
○	37	administrator	Admin	3	2021-11-17 01:11:44	0	No Delay 100%	0	TXT	Uploaded	Download	---	
○	36	administrator	Admin	1	2021-10-18 17:35:26	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
○	35	administrator	Admin	1	2021-10-18 17:34:55	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
○	34	administrator	Admin	1	2021-10-18 16:46:36	0	No Delay 100%	0	Java	Compilation Error	Code	Log	
○	31	administrator	Admin	1	2021-10-18 16:34:34	0	No Delay 100%	0	Java	Compilation Error	Code	Log	

Gambar 3.9: Halaman All Submissions

- 6 Gambar 3.9 menunjukkan halaman All Submissions. Pada halaman ini, terdapat *list* seluruh
 7 *submission* untuk *assignment* yang dipilih. Pengguna juga dapat melihat *file* atau kode yang
 8 diunggah, dan nilai yang didapatkannya. Untuk setiap *problem*, sebuah *submission* dapat dipilih

1 sebagai *final submission* melalui halaman ini.

2 Scoreboard

#	Username	Name	Problem 1 (Submit)	Problem 2 (Max)	Upload Only	Total
1	administrator	Admin	0 18:00:55	100 04:56	0 18:00:50	100 37:59:52

Gambar 3.10: Halaman Scoreboard

3 Gambar 3.10 menunjukkan halaman Scoreboard. Pada halaman ini, terdapat *list* nilai pengguna
4 untuk setiap *problem* pada *assignment* yang dipilih.

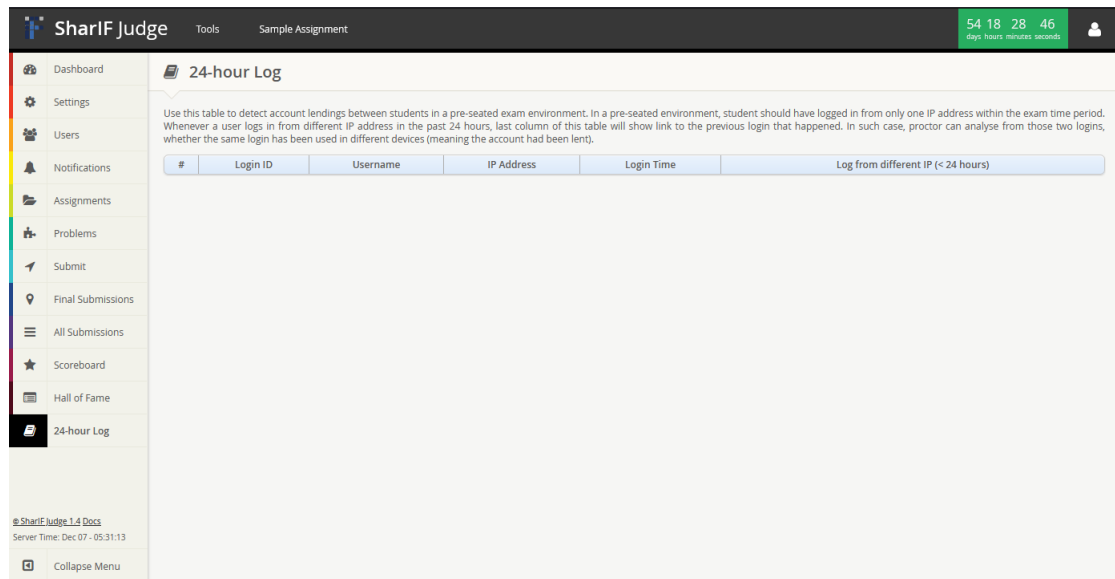
5 Hall of Fame

Rank	Username	Display Name	Total Score
1	administrator	Admin	100

Gambar 3.11: Halaman Hall of Fame

6 Gambar 3.11 menunjukkan halaman Hall of Fame. Pada halaman ini, terdapat *list* pengguna secara
7 berurutan berdasarkan total nilai yang didapatkannya dari seluruh *assignment*.

1 24-Hour Log



Gambar 3.12: Halaman 24-Hour Log

- 2 Gambar 3.12 menunjukkan halaman 24-Hour Log. Pada halaman ini, terdapat *list* yang mencatat
 3 bila akun yang sama melakukan *login* dengan *IP address* yang berbeda dalam jangka waktu 24 jam.
 4 Fitur ini digunakan untuk mendeteksi adanya peminjaman akun.

5 3.1.2 Model, View, Controller

- 6 SharIF Judge menggunakan *framework* CodeIgniter 3. Seperti yang dibahas pada bagian 2.1.1,
 7 *framework* CodeIgniter menerapkan pola arsitektur MVC, dengan komponen-komponen *model*, *view*,
 8 dan *controller*.

9 Model

10 Berikut ini adalah *model* pada SharIF Judge:

11 • Assignment_model

12 Model untuk menangani tabel `shj_assignments`. Fungsi yang dimiliki:

- 13 – `add_assignment($id, $edit)`
 14 Menambah atau memperbarui sebuah *assignment*.
- 15 – `delete_assignment($assignment_id)`
 16 Menghapus sebuah *assignment*.
- 17 – `all_assignments()`
 18 Mengambil seluruh *assignment*.
- 19 – `new_assignment_id()`
 20 Menentukan *integer* terkecil yang dapat digunakan sebagai id *assignment* baru.
- 21 – `all_problems($assignment_id)`
 22 Mengambil seluruh *problem* dari *assignment*.

```

1      - problem_info($assignment_id, $problem_id)
2      Mengambil sebuah problem.
3      - assignment_info($assignment_id)
4      Mengambil sebuah assignment.
5      - is_participant($participants, $username)
6      Mengembalikan TRUE jika $username terdapat dalam $participants.
7      - increase_total_submits($assignment_id)
8      Meningkatkan jumlah total submit sebuah assignment sebanyak satu.
9      - set_moss_time($assignment_id)
10     Memperbarui "Moss Update Time" untuk sebuah assignment.
11     - get_moss_time($assignment_id)
12     Mengambil "Moss Update Time" untuk sebuah assignment.
13     - save_problem_description($assignment_id, $problem_id, $text, $type)
14     Menambah atau memperbarui deskripsi sebuah problem.
15     - _update_coefficients($a_id, $extra_time, $finish_time, $new_late_rule)
16     Memperbarui koefisien seluruh submission pada sebuah assignment.
17 • Hof_model
18 Model untuk menangani informasi hall of fame. Fungsi yang dimiliki:
19     - get_all_final_submission()
20     Mengambil seluruh final submission.
21     - get_all_user_assignments($username)
22     Mengambil seluruh assignment dan problem untuk user tertentu.
23 • Logs_model
24 Model untuk menangani tabel shj_logins. Fungsi yang dimiliki:
25     - insert_to_logs($username, $ip_address)
26     Menambah sebuah catatan login dan menghapus catatan yang sudah melebihi 24 jam.
27     - get_all_logs()
28     Mengambil seluruh catatan login.
29 • Notifications_model
30 Model untuk menangani tabel shj_notifications. Fungsi yang dimiliki:
31     - get_all_notifications()
32     Mengambil seluruh notifikasi.
33     - get_latest_notifications()
34     Mengambil 10 notifikasi terbaru.
35     - add_notification($title, $text)
36     Menambah notifikasi baru.
37     - update_notification($id, $title, $text)
38     Memperbarui sebuah notifikasi.
39     - delete_notification($id)
40     Menghapus sebuah notifikasi.
41     - get_notification($notif_id)
42     Mengambil sebuah notifikasi.

```

```

1      - have_new_notification($time)
2      Mengembalikan TRUE jika terdapat notifikasi setelah $time.
3  • Queue_model
4      Model untuk menangani tabel shj_queue. Fungsi yang dimiliki:
5      - in_queue($username, $assignment, $problem)
6      Mengembalikan TRUE jika sebuah submission sudah berada dalam antrean.
7      - get_queue()
8      Mengambil seluruh antrean.
9      - empty_queue()
10     Mengosongkan antrean.
11     - add_to_queue($submit_info)
12     Menambahkan sebuah submission ke dalam antrean.
13     - rejudge($assignment_id, $problem_id)
14     Menambahkan seluruh submission dari sebuah problem ke dalam antrean untuk dinilai
15     ulang.
16     - rejudge_single($submission)
17     Menambahkan sebuah submission ke dalam antrean untuk dinilai ulang.
18     - get_first_item()
19     Mengambil entry pertama dari antrean.
20     - remove_item($username, $assignment, $problem, $submit_id)
21     Menghapus sebuah entry dari antrean.
22     - save_judge_result_in_db ($submission, $type)
23     Menyimpan hasil penilaian ke dalam database.
24  • Scoreboard_model
25     Model untuk menangani tabel shj_scoreboard. Fungsi yang dimiliki:
26     - _generate_scoreboard($assignment_id)
27     Membuat scoreboard untuk sebuah assignment.
28     - update_scoreboards()
29     Memperbarui scoreboard untuk seluruh assignment.
30     - update_scoreboard($assignment_id)
31     Memperbarui scoreboard untuk sebuah assignment.
32     - get_scoreboard($assignment_id)
33     Mengambil scoreboard untuk sebuah assignment.
34  • Settings_model
35     Model untuk menangani tabel shj_settings. Fungsi yang dimiliki:
36     - get_setting($key)
37     Mengambil sebuah pengaturan.
38     - set_setting($key, $value)
39     Memperbarui sebuah pengaturan.
40     - get_all_settings()
41     Mengambil seluruh pengaturan.
42     - set_settings($settings)

```

Memperbarui beberapa pengaturan.

- **Submit_model**

Model untuk menangani tabel `shj_submissions`. Fungsi yang dimiliki:

- `get_submission($uname, $assignment, $problem, $submit_id)`
Mengambil sebuah *submission*.
- `get_final_submissions($a_id, $u_lv, $uname, $p_num, $f_user, $f_prblm)`
Mengambil seluruh *final submission* untuk sebuah *assignment*.
- `get_all_submissions($a_id, $u_lv, $uname, $p_num, $f_user, $f_prblm)`
Mengambil seluruh *submission* untuk sebuah *assignment*.
- `count_final_submissions($a_id, $u_lv, $uname, $f_user, $f_prblm)`
Menghitung jumlah *final submission* dari *user* tertentu.
- `count_all_submissions($a_id, $u_lv, $uname, $f_user, $f_prblm)`
Menghitung jumlah *submission* dari *user* tertentu.
- `set_final_submission($uname, $assignment, $problem, $submit_id)`
Memperbarui sebuah *submission* menjadi *final*.
- `add_upload_only($submit_info)`
Menambahkan hasil dari *submission upload only* ke dalam *database*.

- **User**

Model untuk menangani informasi preferensi setiap *user*. Fungsi yang dimiliki:

- `select_assignment($assignment_id)`
Menetapkan *assignment* yang dipilih.
- `save_widget_positions($positions)`
Memperbarui posisi *widget*.
- `get_widget_positions()`
Mengambil posisi *widget*.

- **User_model**

Model untuk menangani tabel `shj_users`. Fungsi yang dimiliki:

- `have_user($username)`
Mengembalikan TRUE jika terdapat *user* dengan nama `$username`.
- `user_id_to_username($user_id)`
Mengembalikan *username* dari *user* dengan *id* tertentu.
- `username_to_user_id($username)`
Mengembalikan *id* dari *user* dengan *username* tertentu.
- `have_email($email, $username)`
Mengembalikan TRUE jika terdapat *user* selain `$username` dengan email `$email`.
- `add_user($username, $email, $display_name, $password, $role)`
Menambahkan sebuah *user* baru.
- `add_users($text, $send_mail, $delay)`
Menambahkan beberapa *user* baru.
- `delete_user($user_id)`
Menghapus sebuah *user*.
- `delete_submissions($user_id)`

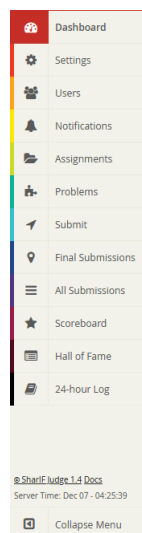

```

1      Menghapus seluruh submission dari sebuah user.
2      - validate_user($username, $password)
3      Mengembalikan TRUE jika $username dan $password valid untuk login.
4      - selected_assignment($username)
5      Mengembalikan assignment yang dipilih sebuah user.
6      - get_names()
7      Mengembalikan nama dari user.
8      - update_profile($user_id)
9      Memperbarui sebuah user.
10     - send_password_reset_mail($email)
11     Mengirim email untuk reset password.
12     - passchange_is_valid($passchange_key)
13     Mengembalikan TRUE jika kunci untuk reset password valid.
14     - reset_password($passchange_key, $newpassword)
15     Memperbarui password menjadi kunci reset password.
16     - get_all_users()
17     Mengambil seluruh user.
18     - get_user($user_id)
19     Mengambil sebuah user.
20     - update_login_time($username)
21     Memperbarui catatan login sebuah user.

```

22 View

23 *View* pada SharIF Judge terbagi menjadi beberapa folder:



Gambar 3.13: side_bar.twig

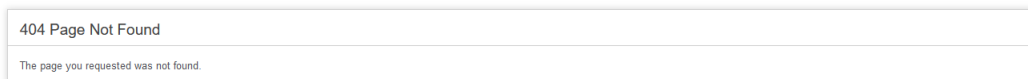
24 • templates

25 Menyimpan komponen dasar halaman. Gambar 3.13 adalah salah satu contoh komponen
 26 dasar halaman.

Gambar 3.14: `submit.twig`

- **pages**

Menyimpan komponen utama halaman. Gambar 3.14 adalah salah satu contoh komponen utama halaman.

Gambar 3.15: `error_404.php`

- **errors**

Menyimpan tampilan halaman error. Gambar 3.15 adalah salah satu contoh halaman error.

Controller

Berikut ini adalah *controller* pada SharIF Judge:

- **Assignments**

Controller untuk menangani *assignments*. Fungsi yang dimiliki:

- `select()`

Memilih *assignment* yang sedang ditampilkan.

```

1      - pdf($assignment_id, $problem_id)
2      Mengunduh file PDF dari sebuah assignment.
3      - downloadtestsdesc($assignment_id)
4      Mengunduh file test case dari sebuah assignment.
5      - download_submissions($type, $assignment_id)
6      Mengunduh seluruh file final submission dari sebuah assignment.
7      - delete($assignment_id)
8      Menghapus sebuah assignment.
9      - add()
10     Menambah atau memperbarui assignment.
11     - edit($assignment_id)
12     Memperbarui assignment.
13 • Dashboard
14     Controller untuk menangani halaman Dashboard. Fungsi yang dimiliki:
15     - widget_positions()
16     Menyimpan posisi widget dari user.
17 • Halloffame
18     Controller untuk menangani halaman Hall of Fame . Fungsi yang dimiliki:
19     - hof_details()
20     Mengambil data yang diperlukan untuk hall of fame.
21 • Install
22     Controller untuk menangani instalasi SharIF Judge.
23 • Login
24     Controller untuk menangani halaman-halaman login. Fungsi yang dimiliki:
25     - register()
26     Registrasi user baru dan menampilkan halaman register.
27     - logout()
28     Log out user saat ini dan mengalihkan ke halaman login.
29     - lost()
30     Menangani email dan menampilkan halaman untuk meminta reset password.
31     - reset($passchange_key)
32     Memproses dan menampilkan halaman untuk ubah reset password.
33 • Logs
34     Controller untuk menangani halaman 24-hour Log.
35     - index() Mengambil data yang diperlukan dan menampilkan halaman 24-hour Log.
36 • Moss
37     Controller untuk menangani halaman Detect Similar Codes . Fungsi yang dimiliki:
38     - update($assignment_id)
39     Memperbarui informasi pada halaman Detect Similar Codes.
40     - _detect($assignment_id)
41     Menjalankan Moss untuk mendeteksi kesamaan kode.
42 • Notifications

```

Controller untuk menangani halaman *Notifications*. Fungsi yang dimiliki:

- `add()`
Menambahkan notifikasi baru dan menampilkan halaman *New Notification*.
- `edit($notif_id)`
Memperbarui sebuah notifikasi.
- `delete()`
Menghapus sebuah notifikasi.
- `check()`
Memeriksa adanya notifikasi baru.

- **Problems**

Controller untuk menangani halaman *Problems*. Fungsi yang dimiliki:

- `index($assignment_id, $problem_id = 1)`
Mengambil data yang diperlukan dan menampilkan halaman *Problems*.
- `edit($type = 'md', $assignment_id, $problem_id = 1)`
Memperbarui deskripsi *problem* dan menampilkan halaman *Edit Problem Description*.

- **Profile**

Controller untuk menangani halaman *Profile*. Fungsi yang dimiliki:

- `index($user_id)`
Mengambil data yang diperlukan dan menampilkan halaman *Profile*.
- `_password_check($str)`
Memeriksa apakah *password* sesuai dengan syarat.
- `_password_again_check($str)`
Memeriksa apakah *password again* sama dengan *password* yang dimasukkan.
- `_email_check($email)`
Memeriksa apakah terdapat user dengan alamat email tertentu.
- `_role_check($role)`
Memeriksa *role* yang dimiliki *user*.

- **Queue**

Controller untuk menangani halaman *Queue*. Fungsi yang dimiliki:

- `index()`
Mengambil data yang diperlukan dan menampilkan halaman *Queue*.
- `pause()`
Memberhentikan antrean.
- `resume()`
Melanjutkan antrean.
- `empty_queue()`
Mengosongkan antrean.

- **Queueprocess**

Controller untuk menangani proses penilaian kode. Fungsi yang dimiliki:

- `run()`
Menilai kode satu per satu dari antrean.

- **Rejudge**

Controller untuk menangani halaman *Rejudge*. Fungsi yang dimiliki:

- `index()`

Mengambil data yang diperlukan dan menampilkan halaman *Rejudge*.

- `rejudge_single()`

Melakukan penilaian ulang untuk sebuah *submission*.

- **Server_time**

Controller untuk menangani sinkronisasi waktu server. Fungsi yang dimiliki:

- `index()`

Mengembalikan waktu server.

- **Submissions**

Controller untuk menangani unduh *submissions* menjadi file Excel. Fungsi yang dimiliki:

- `_download_excel($view)`

Menggunakan *library* PHPEXcel untuk membuat file excel.

- `final_excel()`

Mengunduh data *final submissions* sebagai file excel.

- `all_excel()`

Mengunduh data *final submissions* sebagai file excel.

- `the_final()`

Mengambil dan menampilkan data *final submissions* yang akan diunduh.

- `all()`

Mengambil dan menampilkan data *submissions* yang akan diunduh.

- `select()`

Memilih *final submission*.

- `view_code()`

Menampilkan kode, *result*, atau *log* dari *submission*.

- `download_file()`

Mengunduh file excel.

- **Submit**

Controller untuk menangani *submissions*. Fungsi yang dimiliki:

- `_language_to_type($language)`

Mengembalikan kode singkatan dari bahasa pemrograman.

- `_match($type, $extension)`

Memeriksa apakah bahasa pemrograman dan tipe file sesuai.

- `_check_language($str)`

Memeriksa apakah bahasa pemrograman yang dipilih valid.

- `index()`

Mengambil data yang diperlukan dan menampilkan halaman *Submit*.

- `_upload()`

Menyimpan file yang diunggah dan menambahkannya ke dalam antrian.

- **Users**

Controller untuk menangani halaman *Users*. Fungsi yang dimiliki:

- `index()`

- 1 Mengambil data yang diperlukan dan menampilkan halaman *Users*.
- 2 – `add()`
- 3 Menambah *user* baru dan menampilkan halaman *Add Users*.
- 4 – `delete()`
- 5 Menghapus *user*.
- 6 – `delete_submissions()`
- 7 Menghapus seluruh *submission* dari sebuah *user*.
- 8 – `list_excel()`
- 9 Menggunakan *library* PHPExcel untuk membuat file excel dari *list user*.

3.1.3 Antrean Penilaian Kode

Pada SharIF Judge, seluruh kode yang dikumpulkan pengguna akan dijalankan satu per satu dalam antrean untuk dinilai. Tahap-tahap yang dilalui sebuah kode hingga penilaian selesai adalah sebagai berikut:

- *Controller Submit* menyimpan *file* kode pada folder sesuai dengan *assignment* dan *problem* yang dipilih.
- *Model Queue_model* menyimpan alamat kode sebagai sebuah *entry submission* di tabel *submission*, lalu kode dimasukkan dalam antrean sebagai sebuah *entry* di tabel *queue*.
- *Controller Queueprocess* membaca *entry* tabel *queue* satu per satu untuk dinilai dengan menjalankan *tester.sh*.
- *tester.sh* mengompilasi kode, menjalankan kode dengan tes kasus, menilai hasilnya dengan kunci jawaban, lalu mengembalikan hasil penilaian.
- *Controller Queueprocess* menyimpan nilai kembalian pada database *submissions* dan menghapus *entry* dari tabel *queue*.

3.1.4 Penyimpanan *File* Jawaban

3.2 Analisis Sistem Usulan

Fitur yang akan diimplementasikan pada SharIF Judge adalah sebagai berikut:

- Melihat soal
Saat ini SharIF Judge memiliki kemampuan untuk menyimpan soal dalam bentuk PDF. Soal tersebut akan ditampilkan secara langsung pada *browser*.
- Mengetik kode
Editor teks yang memiliki kemampuan untuk membantu pembuatan kode, seperti *syntax highlighting*.
- Menyimpan kode
Kemampuan untuk menyimpan dan memuat kembali kode yang sudah dibuat pada server.
- Menjalankan kode dengan tes kasus
Kode yang sudah dibuat dapat dijalankan dengan tes kasus yang disediakan oleh pengguna.
- Mengumpulkan Kode dari Editor
Melakukan *submit* kode yang sudah dibuat pada editor.

BAB 4

PERANCANGAN

Bab ini membahas perancangan untuk seluruh fitur yang diimplementasi pada perangkat lunak SharIF Judge.

4.1 Rancangan Antarmuka

Gambar 4.1: Rancangan antarmuka halaman Submit

Seluruh fitur akan diimplementasikan pada halaman Submit. Gambar 4.1 menunjukkan rancangan antarmuka halaman Submit. Pada halaman Submit sudah terdapat *dropdown* untuk memilih *problem* yang akan dikerjakan, dan bahasa pemrograman yang akan digunakan. Kedua *dropdown* tersebut juga akan digunakan pada fitur yang akan diimplementasikan. *Dropdown problem* digunakan untuk menentukan kode yang akan disimpan dan dimuat. Sementara *dropdown language* digunakan untuk memilih *mode syntax highlighting* pada editor kode.

4.2 Menampilkan soal

SharIF Judge sudah memiliki fitur untuk menyimpan soal dalam bentuk PDF, namun untuk melihat soal tersebut, soal harus diunduh terlebih dahulu. Agar pengguna dapat melihat soal secara langsung di halaman *web*, digunakan *library* PDF.js untuk menampilkan *file* PDF soal di halaman Submit.

Untuk menampilkan soal PDF, dilakukan perubahan sebagai berikut:

- *Controller Assignments*:
 - Fungsi *pdf*:
Penambahan kondisi untuk mencegah dialog unduh *file*.
- *View Submit*:
 - Penambahan *iframe* untuk menampilkan PDF.js.

4.3 Editor Kode

Digunakan *library* Ace untuk menambahkan editor kode pada halaman Submit.

Untuk mengimplementasikan editor kode, perlu dilakukan perubahan sebagai berikut:

- *View Submit*:
 - Penambahan *div* sebagai tempat untuk menampilkan editor Ace.
 - Penambahan *script* untuk konfigurasi Ace.
 - Penambahan *script* untuk menyesuaikan mode *syntax highlighting* Ace dengan pilihan bahasa pemrograman pada *dropdown*.

4.4 Menyimpan dan Memuat Kode

Seluruh *submission* yang diunggah oleh pengguna pada SharIF Judge akan disimpan pada folder **Assignments** sesuai dengan *assignment* dan *problem* yang dipilih. Kode pada editor kode juga akan disimpan pada folder yang sama sebagai sebuah *file* txt saat pengguna menekan tombol Save. Kode yang sudah tersimpan akan otomatis dimuat pada editor kode saat pengguna memilih *problem* pada *dropdown*.

Untuk menyimpan dan mengambil kode, perlu dilakukan perubahan sebagai berikut:

- *Controller Submit*:
 - Fungsi *save*:
Fungsi baru untuk menyimpan kode pada *file* txt.
 - Fungsi *load*:
Fungsi baru untuk memuat kode dari *file* txt.
- *View Submit*:
 - Penambahan *button* untuk menyimpan kode.
 - Penambahan *script* untuk memanggil fungsi *save* pada *controller* Submit.
 - Penambahan *script* untuk memanggil fungsi *load* pada *controller* Submit.

4.5 Menjalankan Kode dengan Tes Kasus

Fitur ini memanfaatkan sistem antrean eksekusi kode yang sudah tersedia pada SharIF Judge. Diperlukan beberapa perubahan agar kode yang sudah tersimpan pada editor dapat dimasukkan ke dalam antrean, dan *output* dari kode dapat ditampilkan.

Untuk menjalankan kode dengan tes kasus, dilakukan perubahan sebagai berikut:

- **tester.sh:**
 - Penambahan kondisi untuk menjalankan kode tanpa penilaian dan mencatat hasilnya pada *file* txt.
- **Controller Submit:**
 - Fungsi `_execute`:
Fungsi baru untuk memasukkan kode dari editor ke antrean. Fungsi ini dipanggil oleh fungsi `save("execute")`.
 - Fungsi `get_output`:
Fungsi baru untuk memuat hasil eksekusi kode dari *file* txt.
- **Controller Queueprocess:**
 - Fungsi `run`:
Penambahan kondisi untuk menjalankan `tester.sh` tanpa penilaian.
- **View Submit:**
 - Penambahan `textarea` untuk *input*.
 - Penambahan `textarea` untuk *output*.
 - Penambahan `button` untuk menjalankan kode.
 - Penambahan *script* untuk memanggil fungsi `save("execute")` pada *controller* Submit.
 - Penambahan *script* untuk memanggil fungsi `get_output` pada *controller* Submit.

4.6 Mengumpulkan Kode dari Editor

Fitur ini memanfaatkan fitur *submit* yang sudah tersedia pada SharIF Judge, namun kode yang digunakan adalah kode yang sudah tersimpan pada editor, sebagai ganti dari unggah *file*.

Untuk mengumpulkan kode dari editor, perlu dilakukan perubahan sebagai berikut:

- **Controller Submit:**
 - Fungsi `_submit`:
Fungsi baru untuk mengumpulkan kode. Fungsi ini dipanggil oleh fungsi `save("submit")`.
- **View Submit:**
 - Penambahan `button` untuk mengumpulkan kode.
 - Penambahan *script* untuk memanggil fungsi `save("submit")` pada *controller* Submit.

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai implementasi dan pengujian perangkat lunak SharIF Judge.

5.1 Lingkungan Implementasi dan Pengujian

Implementasi perangkat lunak dilakukan pada beberapa lingkungan yang berbeda:

- Lingkungan *development*:

Perangkat lokal milik penulis yang digunakan untuk pembangunan perangkat lunak. Spesifikasi lingkungan ini adalah sebagai berikut:

- Perangkat Keras:

- * *Processor*: Intel Core i5-7600 3.5GHz

- * *Random Access Memory*: 16GB DDR4

- * *Storage*: 500GB

- Perangkat Lunak:

- * *Operating System*: Windows 10 Home 64-bit

- * *Windows Subsystem for Linux*: Ubuntu 20.04.2 LTS

- Lingkungan *staging*:

Lingkungan *server* yang digunakan untuk menguji perangkat lunak selama pembangunan. Spesifikasi lingkungan ini adalah sebagai berikut:

- Perangkat Keras:

- * *Processor*: Intel DO-Regular 2.4GHz

- * *Random Access Memory*: 1GB

- * *Storage*: 25GB

- Perangkat Lunak:

- * *Operating System*: Ubuntu 20.04.3 LTS

- Lingkungan *production*:

Lingkungan *server* yang digunakan pada kuliah Dasar-dasar Pemrograman dengan alamat <http://daspro.labftis.net>. Spesifikasi lingkungan ini adalah sebagai berikut:

- Perangkat Keras:

- * *Processor*: Intel Xeon E5-2603 1.70GHz

- * *Random Access Memory*: 8GB

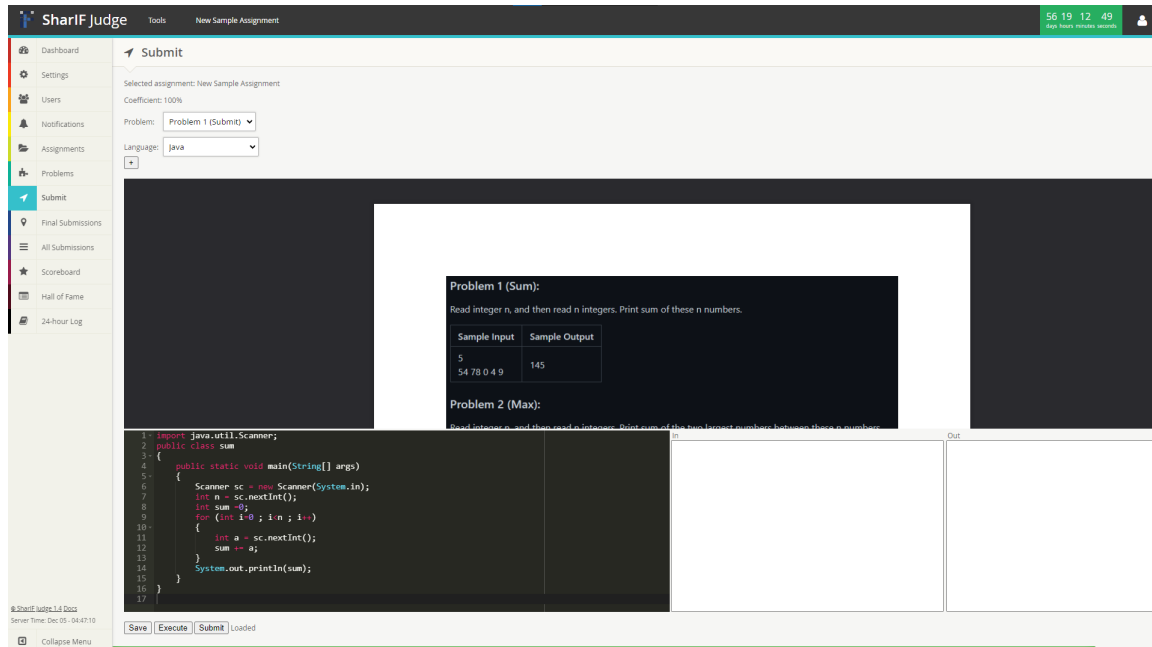
- * *Storage*: 1TB

- Perangkat Lunak:

- * *Operating System*: Ubuntu 16.04.6 LTS

5.2 Implementasi

5.2.1 Tampilan Antarmuka



Gambar 5.1: Antarmuka halaman Submit

Gambar 5.1 merupakan tampilan antarmuka pada halaman Submit yang sudah diimplementasikan. Seluruh perubahan pada `submit.twig` dapat dilihat pada lampiran A.7. Seluruh *style* dan *script* untuk halaman ini terletak di *file* terpisah. *Stylesheet* yang terdapat di `assets\styles\submit.css` dapat dilihat pada lampiran A.9. *Script* yang terdapat di `assets\js\shj_submit.js` dapat dilihat pada lampiran A.8.

5.2.2 Menampilkan Soal

Soal PDF ditampilkan pada `iframe` yang berisi `viewer.html` milik PDF.js. URL dari *file* PDF yang akan ditampilkan adalah URL yang mengarah ke fungsi `pdf` pada controller `Assignments`. URL ini dikirim ke `viewer.html` PDF.js sebagai parameter GET bernama `site_url`.

Fungsi `pdf` pada controller `Assignments` menggunakan fungsi `force_download` yang menyebabkan munculnya dialog unduh pada *browser*. Agar *file* PDF dapat dibaca oleh *browser* dan tidak memunculkan dialog unduh, *file* PDF akan dikembalikan dengan fungsi `die` dengan header `Content-Type: application/pdf`. Ditambahkan parameter `$no_download` pada fungsi `pdf` untuk menentukan apakah *file* PDF soal akan diunduh atau ditampilkan oleh PDF.js. Jika `$no_download` bernilai `FALSE`, maka PDF akan diunduh melalui fungsi `force_download` seperti semula. Jika `$no_download` bernilai `TRUE`, maka PDF akan dikembalikan melalui fungsi `die`. Kode untuk perubahan ini dapat dilihat pada lampiran A.2

5.2.3 Editor Kode

Ace menggunakan sebuah elemen `div` sebagai tempat untuk menampilkan editor kodenya. Editor ace dimuat dan dikonfigurasi melalui JavaScript yang terdapat di `shj_submit.js`. Ditambahkan juga beberapa fungsi untuk mengubah konfigurasi *syntax highlighting* sesuai dengan bahasa yang dipilih, dan untuk mengaktifkan atau menonaktifkan editor kode jika *problem* dan bahasa belum dipilih. Kode JavaScript untuk konfigurasi editor Ace dapat dilihat pada lampiran A.8.

5.2.4 Menyimpan dan Memuat Kode

Kode akan disimpan pada *file* txt bernama `editor.txt`. Nama dan tipe file ini disimpan sebagai konstanta pada `constants.php`. Perubahan kode `constants.php` dapat dilihat pada lampiran A.1.

Untuk menyimpan kode, fungsi `save` ditambahkan pada *controller* `Submit`. Fungsi ini mengambil isi dari editor kode melalui POST lalu menyimpannya pada `editor.txt`. Kode untuk fungsi `save` dapat dilihat pada lampiran A.3 baris 72–135.

Untuk memuat kode, fungsi `load` ditambahkan pada *controller* `Submit`. Bila tersedia, fungsi ini mengambil isi dari `editor.txt` lalu mengembalikan isinya. Kode untuk fungsi `load` dapat dilihat pada lampiran A.3 baris 39–64.

Fungsi `save` dan `load` dipanggil melalui *AJAX request* pada `shj_submit.js`. Fungsi `save` dipanggil ketika tombol Save ditekan, sementara fungsi `load` dipanggil ketika pengguna memilih *problem* pada *dropdown*. Kode JavaScript untuk mengirimkan *AJAX request* tersebut dapat dilihat pada lampiran A.8 baris 77–99 dan baris 23–47.

5.2.5 Menjalankan Kode dengan Tes Kasus

Pada sistem antrian kode yang dibahas pada bagian 3.1.3, tabel `shj_queue` tidak menyimpan alamat dan ekstensi *file*, namun tabel ini menyimpan `submit_id` sebagai referensi untuk tabel `shj_submissions`, dimana alamat dan ekstensi *file* tersimpan. Selain itu, `submit_id` juga digunakan untuk menyimpan nilai yang dihasilkan oleh `tester.sh`. Karena itu, agar kode dapat dimasukkan pada antrian, kode perlu dikumpulkan sebagai *submission* terlebih dahulu.

Agar kode dari editor dapat dijalankan melalui antrian yang sama, perlu dilakukan langkah-langkah berikut ini:

1. Kode yang sudah disimpan sebagai *file* txt disimpan kembali dengan ekstensi yang tepat.
2. *Input* tes kasus disimpan sebagai *file* txt.
3. Membuat *entry* pada tabel `shj_submission` yang bersifat sementara untuk menyimpan alamat dan ekstensi *file* kode. Dikarenakan `submit_id` untuk setiap *submission* selalu dimulai dari 1, *submission* dapat digunakan `submit_id = 0`.
4. Membuat *entry* pada tabel `Queue` dengan `submit_id = 0` dan `type = "exec"` untuk menandakan bahwa kode ini bukan *submission* yang akan dinilai.
5. Ditambahkan parameter dan fungsi pada `tester.sh` yang menjalankan kode dengan tes kasus yang sudah disimpan tanpa melakukan penilaian, lalu menyimpan hasil *output* kode sebagai *file* txt.
6. *File* txt *output* dimuat dan ditampilkan pada halaman submit.

Input dan *output* kode akan disimpan pada *file* txt bernama `exec_in.txt` dan `exec_out.txt`. Nama dan tipe file ini disimpan sebagai konstanta pada `constants.php`. Ditambahkan juga konstanta yang akan digunakan sebagai `submit_id` antrean dari seluruh kode yang akan dijalankan. Perubahan kode `constants.php` dapat dilihat pada lampiran A.1.

Untuk menyimpan kode, fungsi `_execute` ditambahkan pada *controller* `Submit`. Fungsi ini dijalankan oleh fungsi `save("execute")` setelah *file* berhasil disimpan. Kode disimpan kembali dengan ekstensi yang sesuai, lalu informasinya disimpan melalui *model* `Queue_model`. Perubahan kode ini dapat dilihat pada lampiran A.3

Penamaan File Java

5.2.6 Mengumpulkan Kode dari Editor

5.3 Pengujian

5.3.1 Pengujian Fungsional

Pengujian fungsional dilakukan secara lokal pada perangkat penulis. Berikut ini pengujian yang dilakukan terhadap fitur-fitur yang sudah diimplementasi:

Tabel 5.1: Tabel Pengujian Fungsional

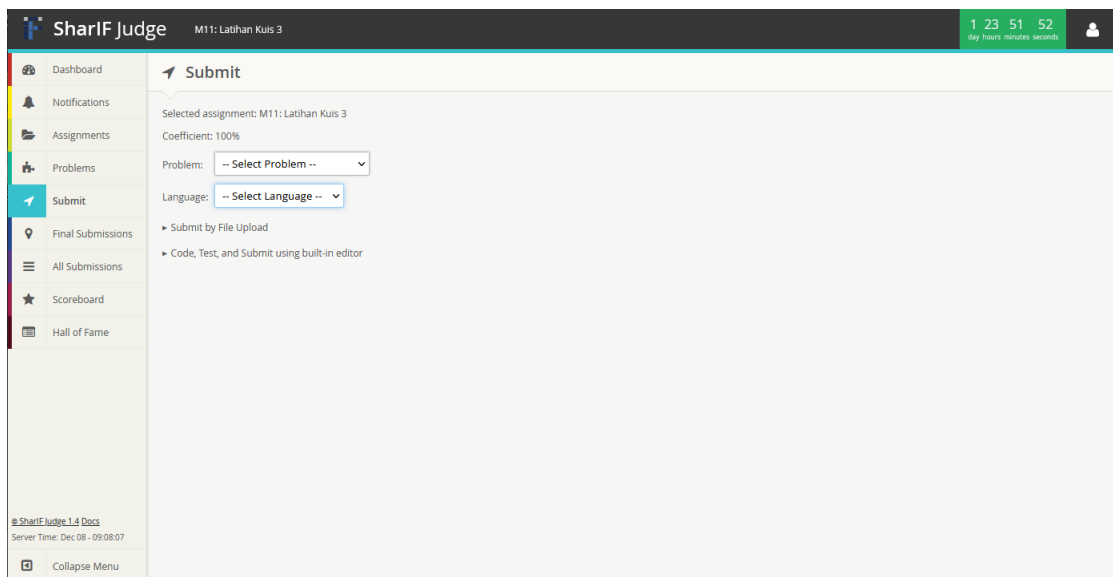
No.	Aksi Pengguna	Reaksi yang diharapkan	Reaksi Perangkat Lunak
1	Membuka halaman Submit, dengan <i>assignment</i> yang dipilih memiliki PDF soal	PDF Soal ditampilkan	sesuai
2	Memilih <i>problem</i> dan <i>language</i> pada dropdown	Editor kode dan tombol Save, Submit, Execute diaktifkan	sesuai
3	Mengetik kode pada editor kode	Kode yang diketik memiliki <i>syntax highlighting</i> sesuai dengan bahasa yang dipilih	sesuai
4	Menekan tombol save	Kode disimpan ditandai dengan <i>status</i> "Saved"	sesuai
5	Memilih <i>problem</i> pada dropdown setelah menyimpan kode	Kode dimuat pada editor kode	sesuai
6	Menekan tombol Execute	<i>Output</i> kode sesuai dengan tes kasus ditampilkan	sesuai
7	Menekan tombol Submit	Pengguna diarahkan ke halaman All Submissions dengan kode pada editor berhasil dikumpulkan dan dinilai	sesuai

5.3.2 Pengujian Eksperimental

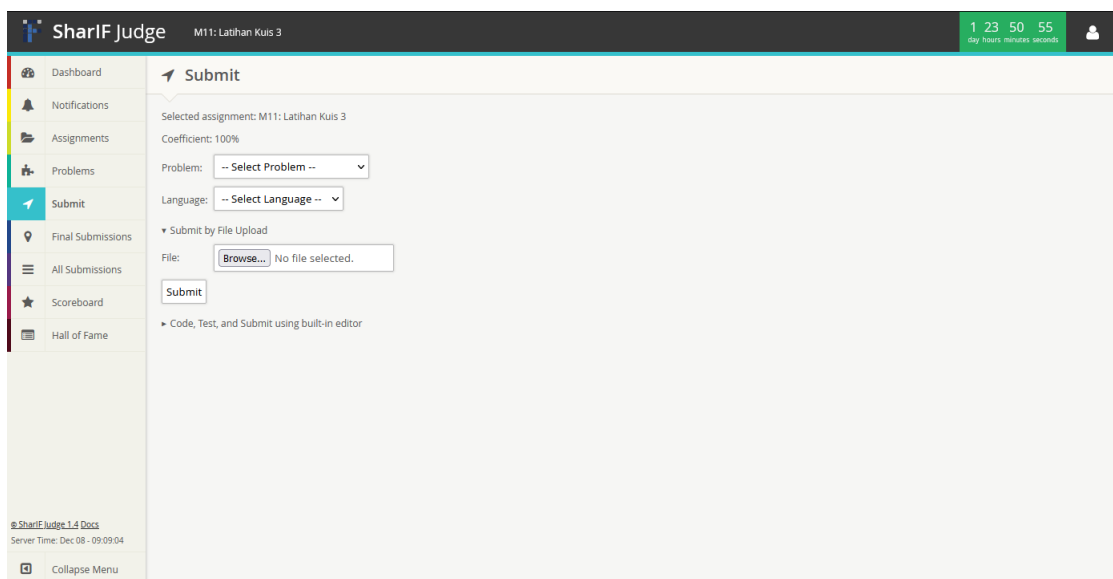
Pengujian eksperimental dilakukan pada mata kuliah Dasar-dasar Pemrograman semester 51 Teknik Informatika Unpar. Perangkat lunak diuji pada *judge* dengan alamat <http://daspro.labftis.net>. Seluruh persoalan dan masukan yang diterima selama mata kuliah Dasar-dasar Pemrograman dicatat pada <https://github.com/athlonneo/SharIF-Judge/issues>.

1 Perubahan Tampilan Antarmuka

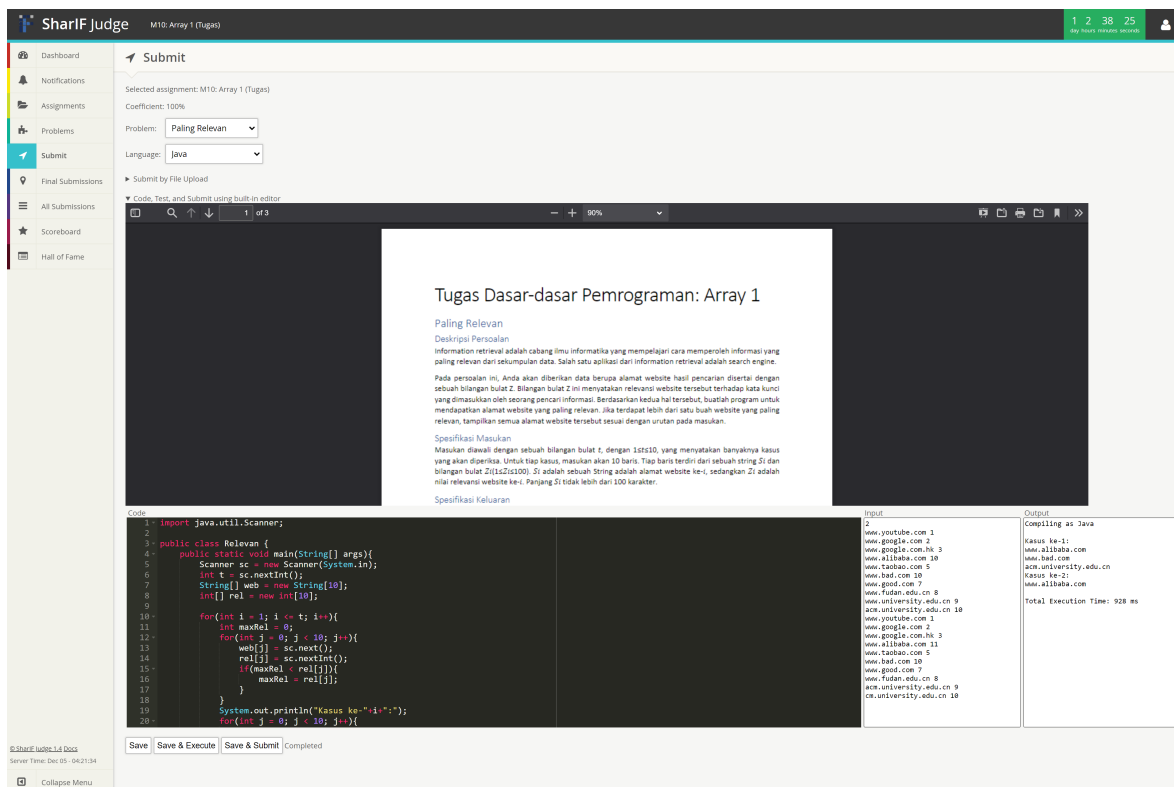
- 2 Tercatat pada *issue #2*, masukan dari salah satu dosen Dasar-dasar pemrograman adalah peru-
- 3 bahan tampilan antarmuka untuk memperjelas fungsi dan meningkatkan kenyamanan pengguna.
- 4 Perubahan yang disarankan adalah sebagai berikut:
- 5 • Memberi jarak antara PDF Viewer dengan *scrollbar*.
 - 6 • Meningkatkan ukuran editor kode.
 - 7 • Mengubah teks tombol "Execute" menjadi "Save & Execute".
 - 8 • Mengubah teks tombol "Submit" menjadi "Save & Submit".
 - 9 • Memisahkan antarmuka unggah *file* dengan IDE.



Gambar 5.2: Halaman Submit



Gambar 5.3: Tampilan unggah *file*



Gambar 5.4: Tampilan IDE

- 1 Gambar 5.2 menunjukkan antarmuka halaman Submit setelah perubahan dari masukan ini.
- 2 Terdapat elemen *summary* pada halaman ini yang dapat ditekan untuk menampilkan antarmuka
- 3 unggah file seperti pada gambar 5.3 dan antarmuka IDE seperti pada gambar 5.4.

4 **Submit Jawaban Soal Upload Only Melalui IDE**

- 5 Tercatat pada *issue* #3, ditemukan masalah jika mengumpulkan jawaban untuk soal *upload only*,
- 6 *status* pada halaman Submissions menjadi "File Not Found".

- 7 Masalah ini disebabkan karena variabel `$this->problem['is_upload_only']` pada fungsi
- 8 `_submit` milik *controller* Submit tidak tersedia, sehingga kode dilanjutkan ke dalam antrian untuk
- 9 dinilai, namun *file* kunci jawaban tidak tersedia, sehingga dikembalikan *status* "File Not Found".

- 10 Untuk menyelesaikan masalah ini, dimanfaatkan variabel `$this->problems` yang sudah tersedia
- 11 pada *controller* Submit untuk mengisi variabel `$this->problem['is_upload_only']`. Perubahan
- 12 ini dapat dilihat pada kode 5.1.

Kode 5.1: Perubahan pada Submit.php

```

13
14 1 @@ -378,8 +378,15 @@ class Submit extends CI_Controller
15 2     $file_name = $file_name.'-'.($this->user->selected_assignment['total_submits']+1);
16 3     $file_path = $user_dir.'/'.$file_name.'.'.$file_ext;
17 4
18 5 +
19 6 +     foreach($this->problems as $item)
20 7 +     {
21 8 +         $this->problem = $item;
22 9 +         break;
23 0 +     }
24 1 +
25 2 +     if (!write_file($file_path, $data)){
26 3         $response = json_encode(array(status=>FALSE, message=>'Unable to submit'));
27 4     }
28 5

```


-
- 1 **Tidak Bisa *Submit* Melalui Unggah File**
 - 2 ***Submit* Jawaban txt Melalui IDE**
 - 3 **Tampilan PDF Viewer jika File PDF Tidak Tersedia**

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil analisis, implementasi, dan pengujian fitur pada perangkat lunak SharIF Judge yang dikembangkan, diperoleh kesimpulan sebagai berikut:

- Dengan memanfaatkan fitur yang sudah ada dan menambahkan fitur baru, halaman Submit pada SharIF Judge dapat berfungsi sebagai *Integrated Development Environment* (IDE), dengan kemampuan untuk mengedit, mengompilasi, menjalankan, dan menguji kode program. Berikut ini adalah fitur-fitur yang diimplementasikan:
 - Menampilkan soal
 - Mengedit Kode
 - Menyimpan dan Memuat Kode
 - Menjalankan Kode dengan Tes Kasus
 - Mengumpulkan kode dari editor
- Seluruh masukan dan penyelesaian masalah yang ditemukan saat pengujian di mata kuliah Dasar-dasar Pemrograman sudah diimplementasikan.
- Hasil Survei

6.2 Saran

Berdasarkan hasil pengembangan yang dilakukan, berikut adalah saran-saran untuk pengembangan selanjutnya:

- Mengimplementasikan fitur merekam ketikan dengan memanfaatkan Ace, untuk membantu pengawasan terhadap mahasiswa selama kegiatan kuliah.
- Menguji perangkat lunak pada mata kuliah pemrograman lainnya, untuk mendapatkan umpan balik dari mahasiswa yang sudah berpengalaman menggunakan SharIF Judge versi sebelumnya.

DAFTAR REFERENSI

- [1] Wasik, S., Antczak, M., Badura, J., Laskowski, A., dan Sternal, T. (2017) A Survey on Online Judge Systems and Their Applications. *ACM Computing Surveys*, **51**, 3:1–3:34.
- [2] Version 1.4 (2014) *Sharif Judge Documentation*. Mohammad Javad Naderi. Tehran, Iran.
- [3] Vallian, S. (2018) Kustomisasi Sharif Judge Untuk Kebutuhan Program Studi Teknik Informatika. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [4] Satav, S., Satao, K., dan Satpathy, S. K. (2011) A Comparative Study and Critical Analysis of Various Integrated Development Environments of C, C++, and Java Languages for Optimum Development. *Universal Journal of Applied computer Science and Technology*, **1**, 9–15.
- [5] Version 3.1.11 (2019) *CodeIgniter User Guide*. British Columbia Institute of Technology. Burnaby, Canada.
- [6] Version 1.44.5 (2021) *Twig Documentation*. Symfony SAS. Clichy, France.
- [7] Shotts, W. (2019) *The Linux Command Line*, 5th edition. No Starch Press, San Francisco, USA.
- [8] Version 2.7.570 (2021) *PDF.js*. Mozilla Corporation. Mountain View, United States.
- [9] Version 1.4.13 (2021) *Ace API Reference*. Ajax.org B.V. Amsterdam, The Netherlands.

LAMPIRAN A

KODE PROGRAM

Kode A.1: Perubahan pada constants.php

```
1 diff --git a/application/config/constants.php b/application/config/constants.php
2 index e71097b..6938566 100644
3 --- a/application/config/constants.php
4 +++ b/application/config/constants.php
5 @@ -109,5 +109,12 @@ define('EXIT_DATABASE', 8); // database error
6 define('EXIT__AUTO_MIN', 9); // lowest automatically-assigned error code
7 define('EXIT__AUTO_MAX', 125); // highest automatically-assigned error code
8
9 /*Code editor related constants*/
10 +define('EDITOR_FILE_NAME', "editor");
11 +define('EDITOR_FILE_EXT', "txt");
12 +define('EDITOR_IN_NAME', "exec_in");
13 +define('EDITOR_OUT_NAME', "exec_out");
14 +define('EDITOR_SUBMIT_ID', 0);
15 +
16 /* End of file constants.php */
17 /* Location: ./application/config/constants.php */
18 \ No newline at end of file
```

Kode A.2: Perubahan pada Assignments.php

```
1 diff --git a/application/controllers/Assignments.php b/application/controllers/Assignments.php
2 index 28154c0..42652a6 100644
3 --- a/application/controllers/Assignments.php
4 +++ b/application/controllers/Assignments.php
5 @@ -97,14 +97,14 @@ class Assignments extends CI_Controller
6 /**
7  * Download pdf file of an assignment (or problem) to browser
8  */
9 - public function pdf($assignment_id, $problem_id = NULL)
10 + public function pdf($assignment_id, $problem_id = NULL, $no_download = FALSE)
11 {
12     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
13     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
14     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
15
16     // Find pdf file
17     if ($problem_id === NULL)
18 +     if ($problem_id === NULL || $problem_id === "null")
19         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{assignment_id}/*.pdf";
20     else
21         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{assignment_id}/p{
22         $problem_id}/*.pdf";
23 @@ -120,10 +120,24 @@ class Assignments extends CI_Controller
24     elseif ( $shj_now() < $starttime)
25         show_error('Selected assignment has not started.');
```

```
25
26 // Download the file to browser
27 $this->load->helper('download')->helper('file');
28 +
29 +
30 $filename = shj_basename($pdf_files[0]);
31 - force_download($filename, file_get_contents($pdf_files[0]), TRUE);
32 + // Download the file to browser
33 + if($no_download === FALSE){
34 +     $this->load->helper('download')->helper('file');
35 +     force_download($filename, file_get_contents($pdf_files[0]), TRUE);
36 + }
37 + else{
38 +     $content = file_get_contents($pdf_files[0]);
39 +     header('Content-Type: application/pdf');
40 +     die($content);
41 + }
42 }
```

Kode A.3: Perubahan pada Submit.php

```
1 diff --git a/application/controllers/Submit.php b/application/controllers/Submit.php
2 index 1cc43f4..2e7a16c 100644
3 --- a/application/controllers/Submit.php
4 +++ b/application/controllers/Submit.php
```

```

5 @@ -64,6 +66,26 @@ class Submit extends CI_Controller
6     }
7
8
9 + // -----
10 +
11 +
12 +     public function _language_to_ext($language)
13 +     {
14 +         $language = strtolower($language);
15 +         switch ($language) {
16 +             case 'c': return 'c';
17 +             case 'c++': return 'cpp';
18 +             case 'python 2': return 'py';
19 +             case 'python 3': return 'py';
20 +             case 'java': return 'java';
21 +             case 'zip': return 'zip';
22 +             case 'pdf': return 'pdf';
23 +             case 'txt': return 'txt';
24 +             default: return FALSE;
25 +         }
26 +     }
27 +
28 +
29 + // -----
30 +
31 +
32 @@ -238,5 +259,236 @@ class Submit extends CI_Controller
33     }
34
35
36 + // -----
37 +
38 + /**
39 +  * Load code from editor file
40 +  */
41 + public function load($problem_id){
42 +     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'.$this->user->selected_assignment['id'].'/p/'.$problem_id.'/'.
43 +     $this->user->username;
44 +     $file_path = $user_dir.'/'.EDITOR_FILE_NAME.'.'.EDITOR_FILE_EXT;
45 +     $input_path = $user_dir.'/'.EDITOR_IN_NAME.'.'.EDITOR_FILE_EXT;
46 +     $output_path = $user_dir.'/'.EDITOR_OUT_NAME.'.'.EDITOR_FILE_EXT;
47 +
48 +     $this->load->helper('file');
49 +     if(!write_file($input_path, ' ')){
50 +         if(!write_file($output_path, ' ')){
51 +             if (!file_exists($file_path)){
52 +                 $response = json_encode(array(content=>'', message=>'No saved file'));
53 +             }
54 +             else{
55 +                 $file_content = file_get_contents($file_path);
56 +                 if ($file_content === FALSE){
57 +                     $response = json_encode(array(content=>'', message=>'Unable to load'));
58 +                 }
59 +                 else{
60 +                     addslashes($file_content);
61 +                     $response = json_encode(array(content=>$file_content, message=>'Loaded'));
62 +                 }
63 +             }
64 +             echo $response;
65 +         }
66 +     }
67 +
68 + // -----
69 +
70 + /**
71 +  * Save code to editor file and submit/execute if needed
72 +  */
73 + public function save($type = FALSE){
74 +     $data = $_POST['code_editor'];
75 +     $problem_id = $_POST['problem_id'];
76 +     $language = $_POST['language'];
77 +
78 +     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'.$this->user->selected_assignment['id'].'/p/'.$problem_id.'/'.
79 +     $this->user->username;
80 +     if (!file_exists($user_dir)){
81 +         mkdir($user_dir, 0700);
82 +     }
83 +     $file_path = $user_dir.'/'.EDITOR_FILE_NAME.'.'.EDITOR_FILE_EXT;
84 +     $input_path = $user_dir.'/'.EDITOR_IN_NAME.'.'.EDITOR_FILE_EXT;
85 +
86 +     $this->load->helper('file');
87 +     if (!write_file($file_path, $data)){
88 +         $response = json_encode(array(status=>FALSE, message=>'Unable to save'));
89 +         echo $response;
90 +     }
91 +     else{
92 +         $response = json_encode(array(status=>TRUE, message=>'Saved'));
93 +         if($type === FALSE){
94 +             echo $response;
95 +         }
96 +         else{
97 +             $now = shj_now();
98 +             if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $this->problem['id']
99 +             ')){
100 +                 $response = json_encode(array(status=>FALSE, message=>'You have already submitted for this problem. Your last
101 +                 submission is still in queue.'));
102 +                 echo $response;
103 +             }
104 +         }
105 +     }
106 + }

```



```

100 +         else if ($this->user->level==0 && !$this->user->selected_assignment['open']){
101 +             $response = json_encode(array(status=>FALSE, message=>'Selected assignment has been closed.'));
102 +             echo $response;
103 +         }
104 +         else if ($now < strtotime($this->user->selected_assignment['start_time'])){
105 +             $response = json_encode(array(status=>FALSE, message=>'Selected assignment has not started.'));
106 +             echo $response;
107 +         }
108 +         else if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['
109 + extra_time']){
110 +             $response = json_encode(array(status=>FALSE, message=>'Selected assignment has finished.'));
111 +             echo $response;
112 +         }
113 +         else if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->
114 + username)){
115 +             $response = json_encode(array(status=>FALSE, message=>'You are not registered for submitting.'));
116 +             echo $response;
117 +         }
118 +         else{
119 +             if($type == 'submit'){
120 +                 $this->_submit($data, $problem_id, $language, $user_dir);
121 +             }
122 +             else if($type == 'execute'){
123 +                 $editor_input = $_POST['editor_input'];
124 +                 if (!write_file($input_path, $editor_input)){
125 +                     $response = json_encode(array(status=>FALSE, message=>'Unable to write input file'));
126 +                     echo $response;
127 +                 }
128 +                 else{
129 +                     $this->_execute($data, $problem_id, $language, $user_dir);
130 +                 }
131 +             }
132 +         }
133 +     }
134 + }
135 +
136 +
137 +
138 + // -----
139 +
140 + /**
141 +  * Add code to queue for judging
142 +  */
143 + private function _submit($data, $problem_id, $language, $user_dir){
144 +     $file_type = $this->_language_to_type(strtolower(trim($language)));
145 +     $file_ext = $this->_language_to_ext(strtolower(trim($language)));
146 +     $file_name = EDITOR_FILE_NAME;
147 +     $file_fname = $file_name.'-'.($this->user->selected_assignment['total_submits']+1);
148 +     $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
149 +
150 +     foreach($this->problems as $item)
151 +         if ($item['id'] == $problem_id)
152 +             {
153 +                 $this->problem = $item;
154 +                 break;
155 +             }
156 +
157 +     if (!write_file($file_path, $data)){
158 +         $response = json_encode(array(status=>FALSE, message=>'Unable to submit'));
159 +     }
160 +     else{
161 +         $this->load->model('submit_model');
162 +
163 +         $submit_info = array(
164 +             'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
165 +             'username' => $this->user->username,
166 +             'assignment' => $this->user->selected_assignment['id'],
167 +             'problem' => $problem_id,
168 +             'file_name' => $file_fname,
169 +             'main_file_name' => $file_name,
170 +             'file_type' => $file_type,
171 +             'coefficient' => $this->coefficient,
172 +             'pre_score' => 0,
173 +             'time' => shj_now_str(),
174 +         );
175 +         if ($this->problem['is_upload_only'] == 0)
176 +             {
177 +                 $this->queue_model->add_to_queue($submit_info);
178 +                 process_the_queue();
179 +             }
180 +         else
181 +             {
182 +                 $this->submit_model->add_upload_only($submit_info);
183 +             }
184 +
185 +         $response = json_encode(array(status=>TRUE, message=>'Submitted'));
186 +         echo $response;
187 +     }
188 + }
189 +
190 +
191 + // -----
192 +
193 + /**
194 +  * Add code to queue for execution only
195 +  */
196 + private function _execute($data, $problem_id, $language, $user_dir){

```

```

197 +     $file_type = $this->_language_to_type(strtolower(trim($language)));
198 +     $file_ext = $this->_language_to_ext(strtolower(trim($language)));
199 +     $file_name = EDITOR_FILE_NAME;
200 +     $file_fname = $file_name.'-'.EDITOR_SUBMIT_ID;
201 +     $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
202 +     $output_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
203 +
204 +     if (!write_file($file_path, $data)){
205 +         $response = json_encode(array(status=>FALSE, message=>'Unable to execute', debug=>$file_path));
206 +     }
207 +     else{
208 +         $submit_info = array(
209 +             'submit_id' => EDITOR_SUBMIT_ID,
210 +             'username' => $this->user->username,
211 +             'assignment' => $this->user->selected_assignment['id'],
212 +             'problem' => $problem_id,
213 +             'file_name' => $file_fname,
214 +             'main_file_name' => $file_name,
215 +             'file_type' => $file_type,
216 +             'coefficient' => $this->coefficient,
217 +             'pre_score' => 0,
218 +             'time' => shj_now_str(),
219 +         );
220 +
221 +         if($this->queue_model->add_to_queue_exec($submit_info)){
222 +             if (!write_file($output_path, 'Queueing...')){
223 +                 $response = json_encode(array(status=>FALSE, message=>'Unable to write output file'));
224 +             }
225 +             else{
226 +                 process_the_queue();
227 +                 $response = json_encode(array(status=>TRUE, message=>'Executing'));
228 +             }
229 +         }
230 +         else{
231 +             $response = json_encode(array(status=>FALSE, message=>'Still in queue'));
232 +         }
233 +     }
234 +     echo $response;
235 + }
236 +
237 + // -----
238 +
239 + /**
240 +  * Load output file as execution result
241 +  */
242 + public function get_output($problem_id){
243 +     $user_dir = rtrim($this->assignment_root, '/') . '/assignment-'. $this->user->selected_assignment['id'] . '/p'. $problem_id . '/'.
244 +     $this->user->username;
245 +     $file_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
246 +
247 +     if (!file_exists($file_path)){
248 +         $response = json_encode(array(status=>FALSE, content=>''));
249 +     }
250 +     else{
251 +         $this->load->helper('file');
252 +         $file_content = file_get_contents($file_path);
253 +         if ($file_content === FALSE){
254 +             $response = json_encode(array(status=>FALSE, content=>''));
255 +         }
256 +         else{
257 +             $complete_status = strpos($file_content, 'Total Execution Time');
258 +             if($complete_status === FALSE){
259 +                 $response = json_encode(array(status=>FALSE, content=>$file_content));
260 +             }
261 +             else{
262 +                 $response = json_encode(array(status=>TRUE, content=>$file_content));
263 +             }
264 +         }
265 +     }
266 +     echo $response;
267 + }
268 + }

```

Kode A.4: Perubahan pada Queueprocess.php

```

1 diff --git a/application/controllers/Queueprocess.php b/application/controllers/Queueprocess.php
2 index ed632cc..7b81d7b 100644
3 --- a/application/controllers/Queueprocess.php
4 +++ b/application/controllers/Queueprocess.php
5 @@ -57,7 +57,7 @@ class Queueprocess extends CI_Controller
6     $assignment = $queue_item['assignment'];
7     $assignment_info = $this->assignment_model->assignment_info($assignment);
8     $problem = $this->assignment_model->problem_info($assignment, $queue_item['problem']);
9 -     $type = $queue_item['type']; // $type can be 'judge' or 'rejudge'
10 +     $type = $queue_item['type']; // $type can be 'judge', 'rejudge', or 'exec'
11
12     $submission = $this->submit_model->get_submission($username, $assignment, $problem['id'], $submit_id);
13
14 @@ -86,7 +86,14 @@ class Queueprocess extends CI_Controller
15     $op4 = $this->settings_model->get_setting('enable_py3_shield');
16     $op5 = $this->settings_model->get_setting('enable_java_policy');
17     $op6 = $assignment_info['javaexceptions'];
18 -
19 +     if($type === 'exec') {
20 +         $exec_only = TRUE;
21 +         $op7 = 1;
22 +     }

```

```

23 +         else {
24 +             $op7 = 0;
25 +         }
26 +
27 +         if ($file_type === 'c' OR $file_type === 'cpp')
28 +             $time_limit = $problem['c_time_limit']/1000;
29 +         elseif ($file_type === 'java')
30 @@ -101,7 +108,7 @@ class Queueprocess extends CI_Controller
31 +             $diff_arg = $problem['diff_arg'];
32 +             $output_size_limit = $this->settings_model->get_setting('output_size_limit') * 1024;
33 +
34 -             $cmd = "cd $tester_path;\n./tester.sh $problem_dir ".escapeshellarg($username).' '.escapeshellarg($main_filename).' '.
escapeshellarg($raw_filename)." $file_type $time_limit $time_limit_int $memory_limit $output_size_limit $diff_cmd $diff_arg
$op1 $op2 $op3 $op4 $op5 $op6";
35 +             $cmd = "cd $tester_path;\n./tester.sh $problem_dir ".escapeshellarg($username).' '.escapeshellarg($main_filename).' '.
escapeshellarg($raw_filename)." $file_type $time_limit $time_limit_int $memory_limit $output_size_limit $diff_cmd $diff_arg
$op1 $op2 $op3 $op4 $op5 $op6 $op7";
36 +
37 +             file_put_contents($userdir.'/log', $cmd);

```

Kode A.5: Perubahan pada Queue_model.php

```

1 diff --git a/application/models/Queue_model.php b/application/models/Queue_model.php
2 index e384b43..f42cda4 100644
3 --- a/application/models/Queue_model.php
4 +++ b/application/models/Queue_model.php
5 @@ -50,6 +50,11 @@ class Queue_model extends CI_Model
6 + public function empty_queue ()
7 + {
8 +     return $this->db->empty_table('queue');
9 + }
10 +
11 + //Delete all dummy submission
12 + $this->db->delete('submissions', array(
13 +     'submit_id' => 0,
14 + ));
15 + }
16 +
17 @@ -180,6 +185,16 @@ class Queue_model extends CI_Model
18 +     'assignment' => $assignment,
19 +     'problem' => $problem
20 + ));
21 +
22 + //Delete dummy submission if exec only
23 + if($submit_id == 0){
24 +     $this->db->delete('submissions', array(
25 +         'submit_id' => $submit_id,
26 +         'username' => $username,
27 +         'assignment' => $assignment,
28 +         'problem' => $problem
29 +     ));
30 + }
31 + }
32 + }
33 +
34 @@ -221,4 +236,36 @@ class Queue_model extends CI_Model
35 +     $this->scoreboard_model->update_scoreboard($submission['assignment']);
36 + }
37 +
38 + // -----
39 + /**
40 + * Adds a dummy submission to queue for execution only
41 + */
42 + public function add_to_queue_exec($submit_info)
43 + {
44 +     $query = $this->db->get_where('submissions', array(
45 +         'submit_id' => $submit_info['submit_id'],
46 +         'username' => $submit_info['username'],
47 +         'assignment' => $submit_info['assignment'],
48 +         'problem' => $submit_info['problem']
49 +     ));
50 +     if ($query->num_rows() == 0){
51 +         $submit_info['is_final'] = 0;
52 +         $submit_info['status'] = 'PENDING';
53 +         $this->db->insert('submissions', $submit_info);
54 +
55 +         $this->db->insert('queue', array(
56 +             'submit_id' => $submit_info['submit_id'],
57 +             'username' => $submit_info['username'],
58 +             'assignment' => $submit_info['assignment'],
59 +             'problem' => $submit_info['problem'],
60 +             'type' => 'exec'
61 +         ));
62 +         return TRUE;
63 +     }
64 +     else{
65 +         return FALSE;
66 +     }
67 + }
68 + }
69 + }
70 + }
71 + }
72 \ No newline at end of file

```

Kode A.6: Perubahan pada Submit_model.php

```

1 diff --git a/application/models/Submit_model.php b/application/models/Submit_model.php
2 index 3347d7e..66913e5 100644
3 --- a/application/models/Submit_model.php
4 +++ b/application/models/Submit_model.php
5 @@ -42,6 +42,7 @@ class Submit_model extends CI_Model {
6     public function get_final_submissions($assignment_id, $user_level, $username, $page_number = NULL, $filter_user = NULL,
7         $filter_problem = NULL)
8     {
9         $arr['assignment'] = $assignment_id;
10        + $arr['submit_id !='] = EDITOR_SUBMIT_ID;
11        $arr['is_final'] = 1;
12        if ($user_level === 0) // students can only get final submissions of themselves
13            $arr['username'] = $username;
14 @@ -69,6 +70,7 @@ class Submit_model extends CI_Model {
15     public function get_all_submissions($assignment_id, $user_level, $username, $page_number = NULL, $filter_user = NULL,
16         $filter_problem = NULL)
17     {
18         $arr['assignment'] = $assignment_id;
19        + $arr['submit_id !='] = EDITOR_SUBMIT_ID;
20        if ($user_level === 0)
21            $arr['username'] = $username;
22        elseif ($filter_user !== NULL)
23            $arr['username'] = $username;
24 @@ -94,6 +96,7 @@ class Submit_model extends CI_Model {
25     public function count_final_submissions($assignment_id, $user_level, $username, $filter_user = NULL, $filter_problem = NULL)
26     {
27         $arr['assignment'] = $assignment_id;
28        + $arr['submit_id !='] = EDITOR_SUBMIT_ID;
29        $arr['is_final'] = 1;
30        if ($user_level === 0)
31            $arr['username'] = $username;
32 @@ -111,6 +114,7 @@ class Submit_model extends CI_Model {
33     public function count_all_submissions($assignment_id, $user_level, $username, $filter_user = NULL, $filter_problem = NULL)
34     {
35         $arr['assignment'] = $assignment_id;
36        + $arr['submit_id !='] = EDITOR_SUBMIT_ID;
37        if ($user_level === 0)
38            $arr['username'] = $username;
39        elseif ($filter_user !== NULL)

```

Kode A.7: Perubahan pada Submit.twig

```

1 diff --git a/application/views/pages/submit.twig b/application/views/pages/submit.twig
2 index 22b54cf..70a9526 100644
3 --- a/application/views/pages/submit.twig
4 +++ b/application/views/pages/submit.twig
5 @@ -12,21 +12,13 @@
6
7
8 {% block other_assets %}
9 -<script>
10 -    shj.p={};
11 -    {{ problems_js|raw }}
12 -    $(document).ready(function(){
13 -        $("select#problems").change(function(){
14 -            var v = $(this).val();
15 -            $('select#languages').empty();
16 -            $('<option value="0" selected="selected">-- Select Language --</option>').appendTo('select#languages');
17 -            if (v==0)
18 -                return;
19 -            for (var i=0;i<shj.p[v].length;i++)
20 -                $('<option value="'+shj.p[v][i]+'>'+shj.p[v][i]+'</option>').appendTo('select#languages');
21 -        });
22 -    });
23 -</script>
24 + <link rel="stylesheet" type="text/css" href="{{ base_url('assets/styles/submit.css') }}" />
25 + <script>
26 +     shj.p={};
27 +     {{ problems_js|raw }}
28 + </script>
29 + <script src="{{ base_url('assets/ace/ace.js') }}"></script>
30 + <script type="text/javascript" src="{{ base_url('assets/js/shj-submit.js') }}"></script>
31 {% endblock %}
32
33 @@ -55,18 +47,47 @@
34 </select>
35 {{ form_error('language', '<div class="shj_error">', '</div>') }}
36
37 </p>
38 <p class="input_p">
39     <label for="file" class="tiny">File:</label>
40     <input type="file" id="file" class="sharif_input medium" name="userfile" />
41     {% if upload_state == 'error' %}
42     <div class="shj_error">Error uploading file.</div>
43     {% elseif upload_state == 'ok' %}
44     <div class="shj-ok">File uploaded successfully. See the result in 'All Submissions'.</div>
45     {% endif %}
46 </p>
47 <p class="input_p">
48     <input type="submit" value="Submit" class="sharif_input"/>
49 </p>
50 <br>
51 <details>
52     <summary>Submit by File Upload</summary>
53     <p class="input_p upload_hidden">
54         <label for="file" class="tiny">File:</label>
55         <input type="file" id="file" class="sharif_input medium" name="userfile" />

```

```

56 +         {% if upload_state == 'error' %}
57 +         <div class="shj_error">Error uploading file.</div>
58 +         {% elseif upload_state == 'ok' %}
59 +         <div class="shj_ok">File uploaded successfully. See the result in 'All Submissions'.</div>
60 +         {% endif %}
61 +     </p>
62 +     <p class="input_p upload_hidden">
63 +         <input type="submit" value="Submit" class="sharif_input"/>
64 +     </p>
65 + </details>
66 </form>
67 +
68 + <br>
69 + <details>
70 +     <summary>Code, Test, and Submit using built-in editor</summary>
71 +     <iframe id="pdf_viewer" src="{% base_url('assets/pdfjs/web/viewer.html?file=') ~ site_url('assignments/pdf/' ~ user.
selected_assignment.id ~ '/null/true') %}" ></iframe>
72 +     <div id="ide_wrap">
73 +         <fieldset id="editor_wrap">
74 +             <legend>Code</legend>
75 +             <div id="code_editor" ></div>
76 +         </fieldset>
77 +         <fieldset id="in_wrap">
78 +             <legend>Input</legend>
79 +             <textarea id="editor_input" class="in_out"></textarea>
80 +         </fieldset>
81 +         <fieldset id="out_wrap">
82 +             <legend>Output</legend>
83 +             <textarea id="editor_output" class="in_out" readonly></textarea>
84 +         </fieldset>
85 +     </div>
86 +     <br>
87 +     <button type="button" id="editor_save" disabled>Save</button>
88 +     <button type="button" id="editor_execute" disabled>Save & Execute</button>
89 +     <button type="button" id="editor_submit" disabled>Save & Submit</button>
90 +     <span id="ajax_status"></span>
91 + </details>
92 {% endif %}
93 -{% endblock %} {# main_content #}
94 \ No newline at end of file
95 +{% endblock %} {# main_content #}

```

Kode A.8: shj_submit.js

```

1 /**
2  * SharIF Judge
3  * @file shj_submit.js
4  *
5  * Javascript codes for "Submit" page
6  */
7
8 $(document).ready(function(){
9     var editor = ace.edit("code_editor");
10    editor.setOptions({
11        theme: "ace/theme/monokai",
12        fontSize: "11pt"
13    });
14
15    function disableEditor(bool) {
16        $("#editor_save").prop("disabled", bool);
17        $("#editor_execute").prop("disabled", bool);
18        $("#editor_submit").prop("disabled", bool);
19        $("#editor_input").prop("disabled", bool);
20        editor.setReadOnly(bool);
21    }
22
23    function loadCode(problem_id){
24        $("#editor_input").val("");
25        $("#editor_output").val("");
26
27        if(problem_id == 0){
28            disableEditor(true);
29            editor.setValue("");
30            $("#ajax_status").html("Select_problem_and_language");
31        }
32        else{
33            disableEditor(true);
34            $.ajax({
35                url: shj.site_url + 'submit/load/' + problem_id,
36                cache: false,
37                success: function (data){
38                    data = JSON.parse(data);
39                    editor.setValue(data.content);
40                    $("#ajax_status").html(data.message);
41                },
42                error: function (error){
43                    console.error(error);
44                },
45            });
46        }
47    }
48
49    $("select#problems").change(function(){
50        var v = $(this).val();
51        loadCode(v);
52        $('select#languages').empty();
53        $('<option value="0" selected="selected">--_Select_Language_--</option>').appendTo('select#languages');
54        for (var i=0;i<shj.p[v].length;i++)

```



```

154         }
155         else{
156             $("#ajax_status").html("Completed");
157             disableEditor(false);
158         }
159     },
160     error: function (error){
161         console.error(error);
162         disableEditor(false);
163     },
164     })();
165 }
166 else{
167     disableEditor(false);
168 }
169 },
170 error: function (error){
171     console.error(error);
172     disableEditor(false);
173 },
174 });
175 });
176 });
177
178 loadCode($("#select#problems").val());
179 });

```

Kode A.9: submit.css

```

1 iframe#pdf_viewer{
2     width:_calc(90vw_-_160px);
3     height:_55vh;
4     padding:_0;
5     border:_0;
6 }
7
8 div#ide_wrap{
9     display:flex;
10    width:_calc(100vw_-_200px);
11    height:_40vh;
12    padding:_0;
13    border:_0;
14 }
15
16 fieldset#editor_wrap{
17     width:_70%;
18     height:_100%;
19     padding:_0;
20     border:_0;
21 }
22
23 div#code_editor{
24     width:_100%;
25     height:_100%;
26     padding:_0;
27     border:_0;
28 }
29
30 fieldset#in_wrap{
31     width:_15%;
32     height:_100%;
33     padding:_0;
34     border:_0;
35 }
36
37 fieldset#out_wrap{
38     width:_15%;
39     height:_100%;
40     padding:_0;
41     border:_0;
42 }
43
44 textarea.in_out{
45     resize:none;
46     width:_100%;
47     height:_100%;
48 }

```

Kode A.10: Perubahan pada tester.sh

```

1 diff --git a/restricted/tester/tester.sh b/restricted/tester/tester.sh
2 index f1ld330..c634872 100755
3 --- a/restricted/tester/tester.sh
4 +++ b/restricted/tester/tester.sh
5 @@ -20,7 +20,7 @@
6
7
8 ##### Example Usage #####
9 -# tester.sh /home/mohammad/judge/homeworks/hw6/p1 mjn problem problem c 1 1 50000 1000000 diff -bB 1 1 1 0 1 1
10 +# tester.sh /home/mohammad/judge/homeworks/hw6/p1 mjn problem problem c 1 1 50000 1000000 diff -bB 1 1 1 0 1 1 1
11 # In this example judge assumes that the file is located at:
12 # /home/mohammad/judge/homeworks/hw6/p1/mjn/problem.c
13 # And test cases are located at:
14
15 @@ -118,6 +118,14 @@ if [ ${17} = "1" ]; then

```

```

16 | else
17 |     DISPLAY_JAVA_EXCEPTION_ON=false
18 | fi
19 | +if [ ${18} = "1" ]; then
20 | + EXEC_ONLY=true
21 | +else
22 | + EXEC_ONLY=false
23 | +fi
24 | +
25 | +EXEC_IN="exec_in.txt"
26 | +
27 |
28 | # DIFFOPTION can also be "ignore" or "exact".
29 | # ignore: In this case, before diff command, all newlines and whitespaces will be removed from both files
30 |
31 | @@ -136,12 +144,24 @@ function shj_log
32 |     fi
33 | }
34 |
35 | +if $EXEC_ONLY; then
36 | + >${PROBLEMPATH}/${UN}/exec_out.txt
37 | +fi
38 | +EXEC_LOG="${PROBLEMPATH}/${UN}/exec_out.txt";
39 | +function shj_log_exec
40 | +{
41 | +     if $EXEC_ONLY; then
42 | +         echo -e "$@" >>EXEC_LOG
43 | +     fi
44 | +}
45 | +
46 |
47 | function shj_finish
48 | {
49 |     # Get Current Time (in milliseconds)
50 |     END=$((date +%sN)/1000000));
51 |     shj_log "\nTotal Execution Time: ${((END-START))} ms"
52 | + shj_log_exec "\nTotal Execution Time: ${((END-START))} ms"
53 |     echo $@
54 |     exit 0
55 | }
56 | @@ -189,7 +209,6 @@ elif [ ${EXT} = "java" ]; then
57 | fi
58 |
59 |
60 |
61 | COMPILE_BEGIN_TIME=$((date +%sN)/1000000));
62 |
63 | #####
64 | @@ -198,24 +217,45 @@ COMPILE_BEGIN_TIME=$((date +%sN)/1000000));
65 |
66 | if [ ${EXT} = "java" ]; then
67 |     cp ../java.policy java.policy
68 | + MAINCLASS=$(grep -e 'public class\|public static void main\>' ${PROBLEMPATH}/${UN}/${FILENAME}.java | grep -B1 "public
69 | + static void main" | grep '\<class\>' | sed 's/^.*class \+//;s/.*$//')
70 | + if [ ${MAINFILENAME} != ${MAINCLASS} ]; then
71 | +     MAINFILENAME=${MAINCLASS}
72 | + fi
73 | + cp ${PROBLEMPATH}/${UN}/${FILENAME}.java ${MAINFILENAME}.java
74 | - shj_log "Compiling as Java"
75 | + shj_log_exec "Compiling as Java"
76 | + shj_log_exec ""
77 | + shj_log "Compiling as Java ${MAINFILENAME}.java"
78 | + javac ${MAINFILENAME}.java >/dev/null 2>cerr
79 | EXITCODE=$?
80 | COMPILE_END_TIME=$((date +%sN)/1000000));
81 | shj_log "Compiled. Exit Code=${EXITCODE} Execution Time: ${((COMPILE_END_TIME-COMPILE_BEGIN_TIME))} ms"
82 | - if [ ${EXITCODE} -ne 0 ]; then
83 | -     shj_log "Compile Error"
84 | -     shj_log "${(cat cerr|head -10)}"
85 | -     echo '<span class="shj_b">Compile Error</span>' >${PROBLEMPATH}/${UN}/result.html
86 | -     echo '<span class="shj_r">' >> ${PROBLEMPATH}/${UN}/result.html
87 | -     #filepath=$(echo "${JAIL}/${FILENAME}.${EXT}" | sed 's/\//\\\/g') #replacing / with \
88 | -     (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'/\&quot;/g') >> ${PROBLEMPATH}/
89 | -     $UN/result.html
90 | -     # (cat ${JAIL}/cerr) >> ${PROBLEMPATH}/${UN}/result.html
91 | -     echo "</span>" >> ${PROBLEMPATH}/${UN}/result.html
92 | -     cd ..
93 | -     rm -r ${JAIL} >/dev/null 2>/dev/null
94 | -     shj_finish "Compilation Error"
95 | + if [ ${EXITCODE} -ne 0 ]; then
96 | +     MSGCHECK=$(cat cerr | grep -e '\<should be declared in a file named\>' | grep -Po '[\w]+?(?=\ is public)')
97 | +     echo "${MSGCHECK}" >> ${PROBLEMPATH}/${UN}/result.html
98 | +     if [ -n "${MSGCHECK}" ]; then
99 | +         MAINFILENAME=${MSGCHECK}
100 | +         cp ${PROBLEMPATH}/${UN}/${FILENAME}.java ${MAINFILENAME}.java
101 | +         shj_log "Compiling as Java ${MAINFILENAME}.java"
102 | +         javac ${MAINFILENAME}.java >/dev/null 2>cerr
103 | +         EXITCODE=$?
104 | +         COMPILE_END_TIME=$((date +%sN)/1000000));
105 | +         shj_log "Compiled. Exit Code=${EXITCODE} Execution Time: ${((COMPILE_END_TIME-COMPILE_BEGIN_TIME))} ms"
106 | +     fi
107 | +     if [ ${EXITCODE} -ne 0 ]; then
108 | +         shj_log "Compile Error"
109 | +         shj_log "${(cat cerr|head -10)}"
110 | +         echo '<span class="shj_b">Compile Error</span>' >${PROBLEMPATH}/${UN}/result.html
111 | +         echo '<span class="shj_r">' >> ${PROBLEMPATH}/${UN}/result.html
112 | +         ERRMSG=$(cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'/\&quot;/g')
113 | +         echo "${ERRMSG}" >> ${PROBLEMPATH}/${UN}/result.html
114 | +         #filepath=$(echo "${JAIL}/${FILENAME}.${EXT}" | sed 's/\//\\\/g') #replacing / with \

```



```

113 +         #(cat $JAIL/cerr) >> $PROBLEMPATH/$UN/result.html
114 +         echo "</span>" >> $PROBLEMPATH/$UN/result.html
115 +         shj_log_exec "$(cat cerr)"
116 +         cd ..
117 +         rm -r $JAIL >/dev/null 2>/dev/null
118 +         shj_finish "Compilation Error"
119 +     fi
120 fi
121 fi
122
123 @@ -228,6 +268,8 @@ fi
124 if [ "$EXT" = "py2" ]; then
125     cp $PROBLEMPATH/$UN/$FILENAME.py $FILENAME.py
126     shj_log "Checking Python Syntax"
127 +     shj_log_exec "Checking Python Syntax"
128 +     shj_log_exec ""
129     python2 -0 -m py_compile $FILENAME.py >/dev/null 2>cerr
130     EXITCODE=$?
131     COMPILE_END_TIME=$((date +%s%N)/1000000);
132 @@ -239,6 +281,7 @@ if [ "$EXT" = "py2" ]; then
133     echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
134     (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/"/\&quot;/g') >> $PROBLEMPATH/
135     $UN/result.html
136     echo "</span>" >> $PROBLEMPATH/$UN/result.html
137 +     shj_log_exec "$(cat cerr)"
138 +     cd ..
139     rm -r $JAIL >/dev/null 2>/dev/null
140     shj_finish "Syntax Error"
141 @@ -259,6 +302,8 @@ fi
142 if [ "$EXT" = "py3" ]; then
143     cp $PROBLEMPATH/$UN/$FILENAME.py $FILENAME.py
144     shj_log "Checking Python Syntax"
145 +     shj_log_exec "Checking Python Syntax"
146 +     shj_log_exec ""
147     python3 -0 -m py_compile $FILENAME.py >/dev/null 2>cerr
148     EXITCODE=$?
149     COMPILE_END_TIME=$((date +%s%N)/1000000);
150 @@ -270,6 +315,7 @@ if [ "$EXT" = "py3" ]; then
151     echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
152     (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/"/\&quot;/g') >> $PROBLEMPATH/
153     $UN/result.html
154     echo "</span>" >> $PROBLEMPATH/$UN/result.html
155 +     shj_log_exec "$(cat cerr)"
156 +     cd ..
157     rm -r $JAIL >/dev/null 2>/dev/null
158     shj_finish "Syntax Error"
159 @@ -295,6 +341,8 @@ if [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
160 EXEFILE="s_$(echo $FILENAME | sed 's/[^a-zA-Z0-9]/g') # Name of executable file
161 cp $PROBLEMPATH/$UN/$FILENAME.$EXT code.c
162 shj_log "Compiling as $EXT"
163 + shj_log_exec "Compiling as $EXT"
164 + shj_log_exec ""
165 if $SANDBOX_ON; then
166     shj_log "Enabling EasySandbox"
167     if cp ../easysandbox/EasySandbox.so EasySandbox.so; then
168 @@ -360,6 +408,7 @@ if [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
169     (cat cerr | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/"/\&quot;/g') >> $PROBLEMPATH/$UN/
170     result.html
171 fi
172 echo "</span>" >> $PROBLEMPATH/$UN/result.html
173 + shj_log_exec "$(cat cerr)"
174 + cd ..
175     rm -r $JAIL >/dev/null 2>/dev/null
176     shj_finish "Compilation Error"
177 @@ -372,194 +421,303 @@ fi
178 ##### TESTING #####
179 -shj_log "\nTesting..."
180 -shj_log "$TST test cases found"
181 +if ! $EXEC_ONLY; then
182 +     shj_log "\nTesting..."
183 +     shj_log "$TST test cases found"
184 -echo "" >$PROBLEMPATH/$UN/result.html
185 +echo "" >$PROBLEMPATH/$UN/result.html
186
187 -if [ -f "$PROBLEMPATH/tester.cpp" ] && [ ! -f "$PROBLEMPATH/tester.executable" ]; then
188 -     shj_log "Tester file found. Compiling tester..."
189 -     TST_COMPILE_BEGIN_TIME=$((date +%s%N)/1000000);
190 -     g++ $PROBLEMPATH/tester.cpp -lm -O2 -o $PROBLEMPATH/tester.executable
191 -     EC=$?
192 -     TST_COMPILE_END_TIME=$((date +%s%N)/1000000);
193 -     if [ $EC -ne 0 ]; then
194 -         shj_log "Compiling tester failed."
195 -         cd ..
196 -         rm -r $JAIL >/dev/null 2>/dev/null
197 -         shj_finish "Invalid Tester Code"
198 -     else
199 -         shj_log "Tester compiled. Execution Time: $((TST_COMPILE_END_TIME-TST_COMPILE_BEGIN_TIME)) ms"
200 +     if [ -f "$PROBLEMPATH/tester.cpp" ] && [ ! -f "$PROBLEMPATH/tester.executable" ]; then
201 +         shj_log "Tester file found. Compiling tester..."
202 +         TST_COMPILE_BEGIN_TIME=$((date +%s%N)/1000000);
203 +         g++ $PROBLEMPATH/tester.cpp -lm -O2 -o $PROBLEMPATH/tester.executable
204 +         EC=$?
205 +         TST_COMPILE_END_TIME=$((date +%s%N)/1000000);
206 +         if [ $EC -ne 0 ]; then
207 +             shj_log "Compiling tester failed."

```

```

209 +         cd ..
210 +         rm -r $JAIL >/dev/null 2>/dev/null
211 +         shj_finish "Invalid Tester Code"
212 +     else
213 +         shj_log "Tester compiled. Execution Time: ${TST_COMPILE_END_TIME-TST_COMPILE_BEGIN_TIME}) ms"
214 +     fi
215 + fi
216 -fi
217
218 -if [ -f "$PROBLEMPATH/tester.executable" ]; then
219 -    shj_log "Copying tester executable to current directory"
220 -    cp $PROBLEMPATH/tester.executable shj_tester
221 -    chmod +x shj_tester
222 -fi
223 + if [ -f "$PROBLEMPATH/tester.executable" ]; then
224 +     shj_log "Copying tester executable to current directory"
225 +     cp $PROBLEMPATH/tester.executable shj_tester
226 +     chmod +x shj_tester
227 + fi
228
229
230 -PASSEDTES=0
231 + PASSEDTES=0
232
233 -for((i=1;i<=TST;i++)); do
234 -    shj_log "\n=== TEST $i ==="
235 -    echo "<span class=\`shj_b\`>Test $i</span>" >>$PROBLEMPATH/$UN/result.html
236 -
237 -    touch err
238 -
239 -    if [ "$EXT" = "java" ]; then
240 -        if $PERL_EXISTS; then
241 -            ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ".\timeout --just-kill -nosandbox -l
242 -            $OUTLIMIT -t $TIMELIMIT java -mx${MEMLIMIT}k $JAVA_POLICY $MAINFILENAME"
243 -        else
244 -            ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "java -mx${MEMLIMIT}k $JAVA_POLICY
245 -            $MAINFILENAME"
246 -        fi
247 -        EXITCODE=?
248 -        if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
249 -            shj_log "Memory Limit Exceeded"
250 -            echo "<span class=\`shj_o\`>Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
251 -            continue
252 -        fi
253 -        if grep -q -m 1 "Exception in" err; then # show Exception
254 -            javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\[a-zA-Z.\]*' | head -1 | head -c 80'
255 -            javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
256 -            shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
257 -            # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
258 -            if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname\$" ../java_exceptions_list; then
259 -                echo "<span class=\`shj_o\`>Runtime Error ($javaexceptionname)</span>" >>$PROBLEMPATH/$UN/result.html
260 -            fi
261 -        for((i=1;i<=TST;i++)); do
262 -            shj_log "\n=== TEST $i ==="
263 -            echo "<span class=\`shj_b\`>Test $i</span>" >>$PROBLEMPATH/$UN/result.html
264 -
265 -            touch err
266 -
267 -            if [ "$EXT" = "java" ]; then
268 -                if $PERL_EXISTS; then
269 -                    ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ".\timeout --just-kill -nosandbox
270 -                    -l $OUTLIMIT -t $TIMELIMIT java -mx${MEMLIMIT}k $JAVA_POLICY $MAINFILENAME"
271 -                else
272 -                    echo "<span class=\`shj_o\`>Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
273 -                    ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "java -mx${MEMLIMIT}k
274 -                    $JAVA_POLICY $MAINFILENAME"
275 -                fi
276 -                continue
277 -            fi
278 -        elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
279 -            # $TIMEOUT ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
280 -            if $SANDBOX_ON; then
281 -                #LD_PRELOAD=./EasySandbox.so ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
282 -                EXITCODE=?
283 -                if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
284 -                    shj_log "Memory Limit Exceeded"
285 -                    echo "<span class=\`shj_o\`>Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
286 -                    continue
287 -                fi
288 -                if grep -q -m 1 "Exception in" err; then # show Exception
289 -                    javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\[a-zA-Z.\]*' | head -1 | head -c 80'
290 -                    javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
291 -                    shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
292 -                    # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
293 -                    if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname\$" ../java_exceptions_list; then
294 -                        echo "<span class=\`shj_o\`>Runtime Error ($javaexceptionname)</span>" >>$PROBLEMPATH/$UN/result.html
295 -                    fi
296 -                else
297 -                    echo "<span class=\`shj_o\`>Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
298 -                fi
299 -                continue
300 -            fi
301 -            elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
302 -                # $TIMEOUT ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
303 -                if $SANDBOX_ON; then
304 -                    #LD_PRELOAD=./EasySandbox.so ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
305 -                    if $PERL_EXISTS; then
306 -                        ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ".\timeout --just-kill --
307 -                        sandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"
308 -                    else
309 -                        ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "LD_PRELOAD=./EasySandbox.so

```

```

./$EXEFILE"
303 +         fi
304 +         EXITCODE=$?
305 +         # remove <<entering SECCOMP mode>> from beginning of output:
306 +         tail -n +2 out >thetemp && mv thetemp out
307 +     else
308 +         #./$FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
309 +         if $PERL_EXISTS; then
310 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill -
nosandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
311 +         else
312 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./$EXEFILE"
313 +         fi
314 +         EXITCODE=$?
315 +     fi
316 +
317 +     elif [ "$EXT" = "py2" ]; then
318 +         if $PERL_EXISTS; then
319 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --sandbox
-l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
320 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill -nosandbox
-l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -0 $FILENAME.py"
321 +         else
322 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "LD_PRELOAD=./EasySandbox.so ./
$EXEFILE"
323 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -0 $FILENAME.py"
324 +         fi
325 +         EXITCODE=$?
326 +         # remove <<entering SECCOMP mode>> from beginning of output:
327 +         tail -n +2 out >thetemp && mv thetemp out
328 +     else
329 +         #./$FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
330 +
331 +     elif [ "$EXT" = "py3" ]; then
332 +         if $PERL_EXISTS; then
333 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill -nosandbox
-l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
334 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill -nosandbox
-l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -0 $FILENAME.py"
335 +         else
336 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./$EXEFILE"
337 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -0 $FILENAME.py"
338 +         fi
339 +         EXITCODE=$?
340 +     fi
341 +
342 +     elif [ "$EXT" = "py2" ]; then
343 +         if $PERL_EXISTS; then
344 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill -nosandbox -l
$OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -0 $FILENAME.py"
345 +         else
346 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -0 $FILENAME.py"
347 +             shj_log "File Format Not Supported"
348 +             cd ..
349 +             rm -r $JAIL >/dev/null 2>/dev/null
350 +             shj_finish "File Format Not Supported"
351 +         fi
352 +         EXITCODE=$?
353 +
354 +     elif [ "$EXT" = "py3" ]; then
355 +         if $PERL_EXISTS; then
356 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill -nosandbox -l
$OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -0 $FILENAME.py"
357 +             shj_log "Exit Code = $EXITCODE"
358 +
359 +             if ! grep -q "FINISHED" err; then
360 +                 if grep -q "SHJ_TIME" err; then
361 +                     t='grep "SHJ_TIME" err|cut -d" " -f3'
362 +                     shj_log "Time Limit Exceeded ($t s)"
363 +                     echo "<span class=\"shj_o\">Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
364 +                     continue
365 +                 elif grep -q "SHJ_MEM" err; then
366 +                     shj_log "Memory Limit Exceeded"
367 +                     echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
368 +                     continue
369 +                 elif grep -q "SHJ_HANGUP" err; then
370 +                     shj_log "Hang Up"
371 +                     echo "<span class=\"shj_o\">Process hanged up</span>" >>$PROBLEMPATH/$UN/result.html
372 +                     continue
373 +                 elif grep -q "SHJ_SIGNAL" err; then
374 +                     shj_log "Killed by a signal"
375 +                     echo "<span class=\"shj_o\">Killed by a signal</span>" >>$PROBLEMPATH/$UN/result.html
376 +                     continue
377 +                 elif grep -q "SHJ_OUTSIZE" err; then
378 +                     shj_log "Output Size Limit Exceeded"
379 +                     echo "<span class=\"shj_o\">Output Size Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
380 +                     continue
381 +                 fi
382 +             else
383 +                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -0 $FILENAME.py"
384 +                 t='grep "FINISHED" err|cut -d" " -f3'
385 +                 shj_log "Time: $t s"
386 +             fi
387 +             EXITCODE=$?
388 +
389 +     else
390 +         shj_log "File Format Not Supported"
391 +         cd ..
392 +         rm -r $JAIL >/dev/null 2>/dev/null

```

```

393 -     shj_finish "File Format Not Supported"
394 - fi
395 -
396 - shj_log "Exit Code = $EXITCODE"
397 -
398 - if ! grep -q "FINISHED" err; then
399 -     if grep -q "SHJ_TIME" err; then
400 -         t='grep "SHJ_TIME" err|cut -d " " -f3'
401 -         shj_log "Time Limit Exceeded ($t s)"
402 -         echo "<span class=\"shj_o\">Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
403 -         continue
404 -     elif grep -q "SHJ_MEM" err; then
405 -         shj_log "Memory Limit Exceeded"
406 -         echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
407 -         continue
408 -     elif grep -q "SHJ_HANGUP" err; then
409 -         shj_log "Hang Up"
410 -         echo "<span class=\"shj_o\">Process hanged up</span>" >>$PROBLEMPATH/$UN/result.html
411 -         continue
412 -     elif grep -q "SHJ_SIGNAL" err; then
413 -         shj_log "Killed by a signal"
414 -         echo "<span class=\"shj_o\">Killed by a signal</span>" >>$PROBLEMPATH/$UN/result.html
415 -         continue
416 -     elif grep -q "SHJ_OUTSIZE" err; then
417 -         shj_log "Output Size Limit Exceeded"
418 -         echo "<span class=\"shj_o\">Output Size Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
419 -
420 -         if [ $EXITCODE -eq 137 ]; then
421 -             #shj_log "Time Limit Exceeded (Exit code=$EXITCODE)"
422 -             #echo "<span style='color: orange;'>Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
423 -             shj_log "Killed"
424 -             echo "<span class=\"shj_o\">Killed</span>" >>$PROBLEMPATH/$UN/result.html
425 -             continue
426 -         fi
427 -     else
428 -         t='grep "FINISHED" err|cut -d " " -f3'
429 -         shj_log "Time: $t s"
430 -     fi
431 -
432 -     if [ $EXITCODE -eq 137 ]; then
433 -         #shj_log "Time Limit Exceeded (Exit code=$EXITCODE)"
434 -         #echo "<span style='color: orange;'>Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
435 -         shj_log "Killed"
436 -         echo "<span class=\"shj_o\">Killed</span>" >>$PROBLEMPATH/$UN/result.html
437 -         continue
438 -     fi
439 -
440 -     if [ $EXITCODE -ne 0 ]; then
441 -         shj_log "Runtime Error"
442 -         echo "<span class=\"shj_o\">Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
443 -         continue
444 -     fi
445 -
446 -     # checking correctness of output
447 -     ACCEPTED=false
448 -     if [ -f shj_tester ]; then
449 -         ./shj_tester $PROBLEMPATH/in/input$i.txt $PROBLEMPATH/out/output$i.txt out
450 -         EC=$?
451 -         if [ $EC -eq 0 ]; then
452 -             ACCEPTED=true
453 -             if [ $EXITCODE -ne 0 ]; then
454 -                 shj_log "Runtime Error"
455 -                 echo "<span class=\"shj_o\">Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
456 -                 continue
457 -             fi
458 -         else
459 -             cp $PROBLEMPATH/out/output$i.txt correctout
460 -             if [ "$DIFFOPTION" = "ignore" ]; then
461 -                 # Removing all newlines and whitespaces before diff
462 -                 tr -d '\t\n\r\f' <out >tmp1 && mv tmp1 out;
463 -                 tr -d '\t\n\r\f' <correctout >tmp1 && mv tmp1 correctout;
464 -
465 -                 # checking correctness of output
466 -                 ACCEPTED=false
467 -                 if [ -f shj_tester ]; then
468 -                     ./shj_tester $PROBLEMPATH/in/input$i.txt $PROBLEMPATH/out/output$i.txt out
469 -                     EC=$?
470 -                     if [ $EC -eq 0 ]; then
471 -                         ACCEPTED=true
472 -                     fi
473 -                 else
474 -                     cp $PROBLEMPATH/out/output$i.txt correctout
475 -                     if [ "$DIFFOPTION" = "ignore" ]; then
476 -                         # Removing all newlines and whitespaces before diff
477 -                         tr -d '\t\n\r\f' <out >tmp1 && mv tmp1 out;
478 -                         tr -d '\t\n\r\f' <correctout >tmp1 && mv tmp1 correctout;
479 -
480 -                         fi
481 -                         # Add a newline at the end of both files
482 -                         echo "" >> out
483 -                         echo "" >> correctout
484 -                         if [ "$DIFFTOOL" = "diff" ]; then
485 -                             # Add -q to diff options (for faster diff)
486 -                             DIFFFARGUMENT="-q $DIFFFARGUMENT"
487 -                         fi
488 -                         # Compare output files
489 -                         if $DIFFTOOL $DIFFFARGUMENT out correctout >/dev/null 2>/dev/null; then
490 -                             ACCEPTED=true
491 -                         fi

```

```

492 |         fi
493 |         # Add a newline at the end of both files
494 |         echo "" >> out
495 |         echo "" >> correctout
496 |         if [ "$DIFFTOOL" = "diff" ]; then
497 |             # Add -q to diff options (for faster diff)
498 |             DIFFFARGUMENT="-q $DIFFFARGUMENT"
499 |         +
500 |         if $ACCEPTED; then
501 |             shj_log "ACCEPTED"
502 |             echo "<span class=\"shj_g\">ACCEPT</span>" >>$PROBLEMPATH/$UN/result.html
503 |             ((PASSEDTESTS=PASSEDTESTS+1))
504 |         +
505 |         else
506 |             shj_log "WRONG"
507 |             echo "<span class=\"shj_r\">WRONG</span>" >>$PROBLEMPATH/$UN/result.html
508 |         fi
509 |         # Compare output files
510 |         if $DIFFTOOL $DIFFFARGUMENT out correctout >/dev/null 2>/dev/null; then
511 |             ACCEPTED=true
512 |         +
513 |         +
514 |         +
515 | +#####
516 | +##### EXECUTION #####
517 | +#####
518 |         +
519 |         +else
520 |             touch err
521 |             if [ "$EXT" = "java" ]; then
522 |                 if $PERL_EXISTS; then
523 |                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout --just-kill -nosandbox -
524 |                     l $OUTLIMIT -t $TIMELIMIT java -mx${MEMLIMIT}k $JAVA_POLICY $MAINFILENAME"
525 |                 +
526 |                 else
527 |                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "java -mx${MEMLIMIT}k $JAVA_POLICY
528 |                     $MAINFILENAME"
529 |                     fi
530 |                     EXITCODE=?
531 |                     if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
532 |                         shj_log "Memory Limit Exceeded"
533 |                         echo "Memory Limit Exceeded" >>out
534 |                         continue
535 |                     fi
536 |                     if grep -q -m 1 "Exception in" err; then # show Exception
537 |                         javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\.[a-zA-Z\.]*' | head -1 | head -c 80'
538 |                         javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
539 |                         shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
540 |                         # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
541 |                         if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname$" ../java_exceptions_list; then
542 |                             echo "Runtime Error ($javaexceptionname)" >>out
543 |                         +
544 |                         else
545 |                             echo "Runtime Error" >>out
546 |                         +
547 |                         fi
548 |                         continue
549 |                     fi
550 |                     elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
551 |                         if $SANDBOX_ON; then
552 |                             if $PERL_EXISTS; then
553 |                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout --just-kill --
554 |                                 sandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
555 |                             +
556 |                             else
557 |                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "LD_PRELOAD=./EasySandbox.so ./
558 |                                 $EXEFILE"
559 |                                 fi
560 |                                 EXITCODE=?
561 |                                 # remove <<entering SECCOMP mode>> from beginning of output:
562 |                                 tail -n +2 out >thetemp && mv thetemp out
563 |                             +
564 |                             else
565 |                                 if $PERL_EXISTS; then
566 |                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout --just-kill -
567 |                                     nosandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
568 |                                 +
569 |                                 else
570 |                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /$EXEFILE"
571 |                                 +
572 |                                 fi
573 |                                 EXITCODE=?
574 |                             +
575 |                             fi
576 |                             elif [ "$EXT" = "py2" ]; then
577 |                                 if $PERL_EXISTS; then
578 |                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout --just-kill -nosandbox -
579 |                                     l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -0 $FILENAME.py"
580 |                                 +
581 |                                 else
582 |                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "python2 -0 $FILENAME.py"
583 |                                 +
584 |                                 fi
585 |                                 EXITCODE=?
586 |                             +
587 |                             elif [ "$EXT" = "py3" ]; then
588 |                                 if $PERL_EXISTS; then
589 |                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout --just-kill -nosandbox -
590 |                                     l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -0 $FILENAME.py"
591 |                                 +
592 |                                 else
593 |                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "python3 -0 $FILENAME.py"
594 |                                 +
595 |                                 fi
596 |                                 EXITCODE=?
597 |                             +
598 |                             else
599 |                                 shj_log "File Format Not Supported"
600 |                                 cd ..
601 |                                 rm -r $JAIL >/dev/null 2>/dev/null

```

```

584 +         shj_finish "File Format Not Supported"
585 +     fi
586 - fi
587
588 - if $ACCEPTED; then
589 -     shj_log "ACCEPTED"
590 -     echo "<span class=\"shj_g\">ACCEPT</span>" >>$PROBLEMPATH/$UN/result.html
591 -     ((PASSEDTESTS=PASSEDTESTS+1))
592 - else
593 -     shj_log "WRONG"
594 -     echo "<span class=\"shj_r\">WRONG</span>" >>$PROBLEMPATH/$UN/result.html
595 - fi
596 -done
597 + shj_log_exec "$(cat out)"
598 +
599 + if ! grep -q "FINISHED" err; then
600 +     if grep -q "SHJ_TIME" err; then
601 +         t='grep "SHJ_TIME" err|cut -d" " -f3'
602 +         shj_log "Time Limit Exceeded ($t s)"
603 +         shj_log_exec "Time Limit Exceeded ($t s)"
604 +         continue
605 +     elif grep -q "SHJ_MEM" err; then
606 +         shj_log "Memory Limit Exceeded"
607 +         shj_log_exec "Memory Limit Exceeded"
608 +         continue
609 +     elif grep -q "SHJ_HANGUP" err; then
610 +         shj_log "Hang Up"
611 +         shj_log_exec "Hang Up"
612 +         continue
613 +     elif grep -q "SHJ_SIGNAL" err; then
614 +         shj_log "Killed by a signal"
615 +         shj_log_exec "Killed by a signal"
616 +         continue
617 +     elif grep -q "SHJ_OUTSIZE" err; then
618 +         shj_log "Output Size Limit Exceeded"
619 +         shj_log_exec "Output Size Limit Exceeded"
620 +         continue
621 +     fi
622 + else
623 +     t='grep "FINISHED" err|cut -d" " -f3'
624 +     shj_log "Time: $t s"
625 + fi
626 +
627 + cp err $PROBLEMPATH/$UN/exec_err.txt
628 + shj_log "Exit Code = $EXITCODE"
629 +fi
630
631 # After I added the feature for showing java exception name and exception place,
632

```

LAMPIRAN B
HASIL EKSPERIMEN