

**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

SharIF Judge adalah sebuah *online judge* (sebuah sistem *online* yang berfungsi untuk mengevaluasi kode program) untuk bahasa pemrograman C, C++, Java dan Python yang dibangun menggunakan CodeIgniter dan Bash. SharIF Judge digunakan pada beberapa mata kuliah pemrograman Teknik Informatika Unpar untuk mempermudah proses pengumpulan dan penilaian kode program. Situasi pandemi Covid-19 menyebabkan seluruh kegiatan kuliah dilaksanakan secara *online*. Pada umumnya, kegiatan praktikum dan ujian 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*. *Integrated Development Environment* (IDE) akan diimplementasikan pada SharIF Judge, dengan kemampuan untuk memfasilitasi proses penulisan kode, lalu mengompilasi, menjalankan, dan mengujinya. Sebagai sebuah IDE, 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. Fitur melihat soal, mengetik, menyimpan, menjalankan, dan mengumpulkan kode melalui IDE diimplementasikan dan diuji pada mata kuliah Dasar-dasar Pemrograman semester ganjil 2021/2022 Teknik Informatika Unpar. Berdasarkan hasil pengujian, seluruh masalah yang ditemukan berhasil diperbaiki, dan seluruh fitur yang diimplementasikan sudah berfungsi dengan baik.

**Kata-kata kunci:** *Online judge, Integrated Development Environment*



## **ABSTRACT**

SharIF Judge is an online judge (an online system that serves to evaluate program code) for C, C++, Java and Python built using CodeIgniter and Bash. SharIF Judge is used on several programming courses in Unpar Informatics Engineering Study Program to help with code submission and scoring. The Covid-19 Pandemic caused every learning activities to be done online. Usually, practical lectures and exams in Unpar Informatics Engineering can be supervised directly by lecturers and assistants in the computer lab. However, supervision becomes more difficult to do when lectures are carried out online. Integrated Development Environment (IDE) will be implemented on SharIF Judge with the ability to facilitate the process of code writing, then compile, run, and test it. As an IDE, further features can be added to help supervise students during learning activities, such as recording typing activities and detecting application inactivity. Features to show problems, type, save, run, and submit code from IDE is implemented and tested on a programming course in Unpar Informatics Engineering. Based on the results, every problem encountered has been fixed successfully and all implemented features have performed adequately.

**Keywords:** Online judge, Integrated Development Environment





# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>ix</b>
<b>DAFTAR GAMBAR</b>	<b>xi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
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 . . . . .	3
<b>2 LANDASAN TEORI</b>	<b>5</b>
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
<b>3 ANALISIS</b>	<b>11</b>
3.1 Analisis Sistem Kini . . . . .	11
3.1.1 Istilah-istilah Umum . . . . .	11
3.1.2 Fitur-fitur SharIF Judge . . . . .	12
3.1.3 Model, View, Controller . . . . .	21
3.1.4 Penyimpanan Kode . . . . .	30
3.1.5 Antrean Penilaian Kode . . . . .	31
3.2 Analisis Sistem Usulan . . . . .	31
<b>4 PERANCANGAN</b>	<b>35</b>
4.1 Rancangan Antarmuka . . . . .	35
4.2 Rancangan Perubahan Kode . . . . .	36
4.2.1 Menampilkan soal . . . . .	36
4.2.2 Mengedit Kode . . . . .	36
4.2.3 Menyimpan Kode . . . . .	37
4.2.4 Memuat Kode . . . . .	37
4.2.5 Menjalankan Kode dengan Tes Kasus . . . . .	37
4.2.6 Mengumpulkan Kode Melalui IDE . . . . .	38
<b>5 IMPLEMENTASI DAN PENGUJIAN</b>	<b>39</b>
5.1 Lingkungan Implementasi dan Pengujian . . . . .	39
5.2 Implementasi . . . . .	40

5.2.1	Tampilan Antarmuka . . . . .	40
5.2.2	Menampilkan Soal . . . . .	40
5.2.3	Editor Kode . . . . .	41
5.2.4	Menyimpan dan Memuat Kode . . . . .	41
5.2.5	Menjalankan Kode dengan Tes Kasus . . . . .	41
5.2.6	Mengumpulkan Kode Melalui IDE . . . . .	44
5.3	Pengujian Fungsional . . . . .	44
5.4	Pengujian Eksperimental . . . . .	45
5.4.1	Perubahan melalui GitHub . . . . .	45
5.4.2	Survei . . . . .	50
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>59</b>
6.1	Kesimpulan . . . . .	59
6.2	Saran . . . . .	59
	<b>DAFTAR REFERENSI</b>	<b>61</b>
	<b>A KODE PROGRAM</b>	<b>63</b>

## DAFTAR GAMBAR

1.1 Tahapan penelitian SharIF Judge . . . . .	2
2.1 <i>Flow Chart</i> CodeIgniter . . . . .	5
3.1 Bagan hubungan <i>assignment</i> , <i>problem</i> , dan <i>submission</i> pada SharIF Judge . . . . .	12
3.2 Use Case Diagram SharIF Judge . . . . .	12
3.3 Halaman Dashboard . . . . .	13
3.4 Halaman Settings . . . . .	14
3.5 Halaman Users . . . . .	14
3.6 Halaman Notifications . . . . .	15
3.7 Halaman Assignments . . . . .	16
3.8 Halaman Problems . . . . .	16
3.9 Halaman Submit . . . . .	17
3.10 Halaman Final Submissions . . . . .	18
3.11 Halaman All Submissions . . . . .	18
3.12 Halaman Scoreboard . . . . .	19
3.13 Halaman Hall of Fame . . . . .	20
3.14 Halaman 24-Hour Log . . . . .	20
3.15 Diagram kelas SharIF Judge . . . . .	21
3.16 Tampilan <i>top bar</i> pada <i>top_bar.twig</i> dan <i>side bar</i> pada <i>side_bar.twig</i> . . . . .	26
3.17 Tampilan Dashboard pada <i>dashboard.twig</i> . . . . .	26
3.18 Tampilan halaman <i>error</i> pada <i>error_404.php</i> . . . . .	27
3.19 Use Case Diagram Fitur Usulan . . . . .	32
4.1 Rancangan antarmuka halaman Submit . . . . .	35
4.2 Diagram kelas perubahan pada SharIF Judge . . . . .	36
5.1 Tampilan antarmuka halaman Submit . . . . .	40
5.2 Tampilan antarmuka setelah perubahan . . . . .	45
5.3 Tampilan antarmuka unggah <i>file</i> . . . . .	46
5.4 Tampilan antarmuka IDE . . . . .	46
5.5 Hasil survei bagian 1 pertanyaan 1 . . . . .	50
5.6 Hasil survei bagian 1 pertanyaan 2 . . . . .	51
5.7 Hasil survei bagian 1 pertanyaan 4 . . . . .	51
5.8 Hasil survei bagian 2 pertanyaan 1 . . . . .	52
5.9 Hasil survei bagian 2 pertanyaan 2 . . . . .	52
5.10 Hasil survei bagian 2 pertanyaan 4 . . . . .	53
5.11 Hasil survei bagian 3 pertanyaan 1 . . . . .	53
5.12 Hasil survei bagian 3 pertanyaan 2 . . . . .	54
5.13 Hasil survei bagian 3 pertanyaan 4 . . . . .	54
5.14 Hasil survei bagian 4 pertanyaan 1 . . . . .	55
5.15 Hasil survei bagian 4 pertanyaan 2 . . . . .	55
5.16 Hasil survei bagian 4 pertanyaan 4 . . . . .	56

5.17 Hasil survei bagian 5 pertanyaan 1 . . . . .	56
5.18 Hasil survei bagian 5 pertanyaan 2 . . . . .	57
5.19 Hasil survei bagian 5 pertanyaan 4 . . . . .	57

# 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].

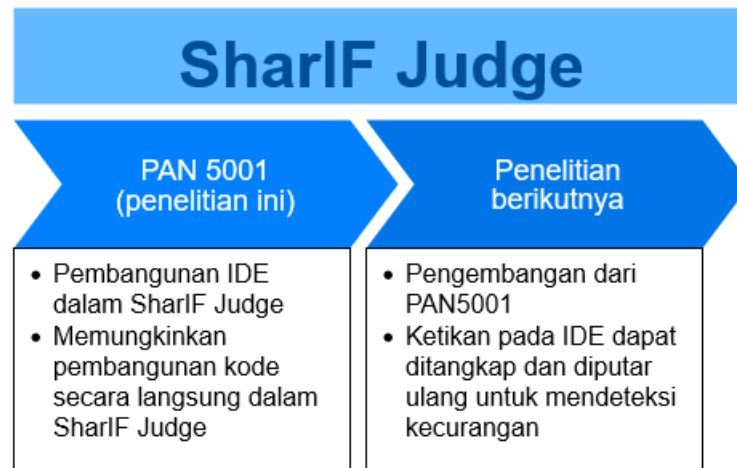
Sharif Judge (dengan if kecil) adalah sebuah *online judge* untuk bahasa pemrograman C, C++, Java dan Python. Perangkat lunak ini diciptakan oleh Mohammad Javad Naderi dan bersifat *open source*. Sharif Judge dibangun dengan menggunakan *framework* CodeIgniter dan Bash [2].

SharIF Judge (dengan IF kapital) adalah modifikasi cabang dari Sharif Judge oleh Stillmen Vallian 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 mulai tahun 2020, 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 dengan fitur yang lengkap untuk membantu penggunaanya dalam proses pengembangan kode. Sebuah IDE pada umumnya memiliki kemampuan untuk mengedit, mengompilasi, menjalankan, dan menguji kode program [4]. Mahasiswa pada umumnya menggunakan aplikasi IDE seperti Netbeans untuk membuat kode program yang kemudian diunggah ke SharIF Judge untuk dinilai.

Gambar 1.1 menggambarkan tahapan penelitian SharIF Judge. Pada skripsi ini, akan dibangun IDE pada SharIF Judge, sehingga seluruh proses pembangunan kode dapat dilakukan langsung di dalam SharIF Judge. Pada penelitian selanjutnya, dapat ditambahkan fitur pada IDE untuk membantu pengawasan terhadap mahasiswa selama kegiatan kuliah, misalnya dengan menangkap dan memutar ulang ketikan pada IDE.



Gambar 1.1: Tahapan penelitian SharIF Judge

## 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

Perangkat lunak diuji pada kuliah Dasar-dasar Pemrograman semester ganjil 2021/2022 Teknik Informatika Unpar. Pada kuliah ini digunakan 2 *judge* terpisah untuk latihan dan kuis. Berdasarkan keputusan dosen koordinator, perangkat lunak skripsi ini hanya akan diuji pada *judge* latihan.

## 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 berhubungan dengan penelitian ini, yaitu CodeIgniter 3, Twig, Bash, PDF.js, dan Ace.

- Bab 3 Analisis

Membahas analisis terhadap perangkat lunak SharIF Judge.

- Bab 4 Perancangan

Membahas perancangan fitur yang diimplementasikan pada SharIF Judge.

- Bab 5 Implementasi dan Pengujian

Membahas implementasi fitur pada SharIF Judge dan pengujian yang dilakukan.

- Bab 6 Kesimpulan dan Saran

Membahas kesimpulan dari penelitian ini dan saran untuk penelitian berikutnya.



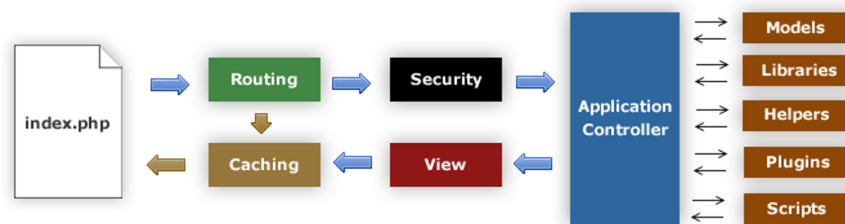


## BAB 2

### 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

Bourne Again SHell (Bash) program yang disempurnakan dari *shell* Unix pertama yang diciptakan oleh Steve Bourne [7]. *Shell* adalah sebuah program pada sistem operasi Unix yang menerima perintah tertulis dan mengirimnya ke sistem operasi untuk dijalankan.

*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*.

Berikut ini merupakan beberapa *command* yang tersedia pada Bash:

- **grep regexp**  
Mencari pola yang sesuai dengan **regexp**, kemudian mencetak seluruh baris yang sesuai dengan pola tersebut.
- **sed s/regexp/replacement/**  
Mencari pola yang sesuai dengan **regexp** dan menggantinya dengan **replacement**.

## 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 lapisan:

- **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

```

39 <!DOCTYPE html>
40 <html>
41 <body>
42 <iframe src="/web/viewer.html?file=sample.pdf"></iframe>
43 </body>
44 </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 bahasa pemrograman.
- *Indent* dan *outdent* otomatis.
- Kemampuan *cut*, *copy*, dan *paste*.

- Kemampuan *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*.

- 1       – `setTheme(String style)`
- 2       Mengubah tema editor.
- 3       – `setValue(String val, Number cursorPos)`
- 4       Mengubah isi teks editor.
- 5       • **Range**
- 6       Mengindikasi sebuah daerah pada editor.
- 7       • **Scrollbar**
- 8       Menangani *scrollbar* editor.
- 9       • **Search**
- 10      Menangani seluruh operasi pencarian teks pada dokumen.
- 11      • **Selection**
- 12      Menyimpan posisi kursor dan seleksi teks pada editor.
- 13      • **TokenIterator**
- 14      Menyediakan fungsi untuk membaca dokumen sebagai aliran token.
- 15      • **Tokenizer**
- 16      Menerima sejumlah aturan dan membuat **Tokenizer**.
- 17      • **UndoManager**
- 18      Menangani fungsi *undo* pada editor.
- 19      • **VirtualRenderer**
- 20      Menggambar tampilan yang terlihat di layar.

Kode 2.7: Contoh kode untuk menggunakan Ace

```

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

```

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

## BAB 3

## ANALISIS

### 3.1 Analisis Sistem Kini

Seperti dibahas pada bab 1, 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. SharIF Judge dibangun menggunakan PHP dengan *framework* CodeIgniter dan Bash.

#### 3.1.1 Istilah-istilah Umum

Pada SharIF Judge, digunakan istilah-istilah berikut ini:

- *Assignment*

Merepresentasikan sebuah tugas yang diberikan. Pada sebuah *assignment*, ditentukan waktu mulai, tenggat waktu, partisipan, dan soal-soal yang diberikan. Untuk setiap *assignment*, dapat diunggah sebuah *file* PDF yang berisi seluruh soal, serta tes kasus dan jawaban yang digunakan untuk menilai kode.

- *Problem*

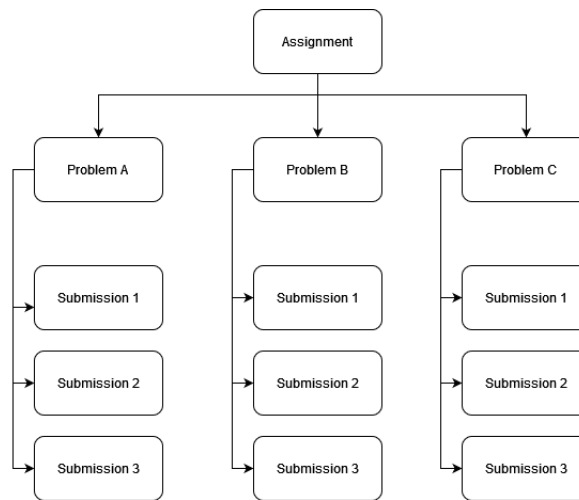
Merepresentasikan sebuah soal yang terdapat pada *assignment*. Untuk setiap *problem*, dapat ditentukan ekstensi *file* yang diizinkan, serta batas waktu dan memori kode. Sebuah *problem* dapat bersifat *upload only*, yang berarti *file* jawaban hanya diunggah, namun tidak dinilai dengan tes kasus.

- *Submission*

Merepresentasikan sebuah jawaban dari seorang pengguna untuk sebuah *problem*. Untuk setiap *submission*, pengguna mengunggah sebuah *file* yang kemudian akan dinilai.

- *Final Submission*

Merepresentasikan sebuah jawaban akhir dari seorang pengguna. Seorang pengguna dapat mengunggah *submission* berkali-kali untuk menguji dan memperbarui jawaban, namun harus memilih satu buah *submission* sebagai jawaban final.



Gambar 3.1: Bagan hubungan *assignment*, *problem*, dan *submission* pada SharIF Judge

- 1 Gambar 3.1 menunjukkan hubungan *assignment*, *problem*, dan *submission* pada SharIF Ju-
- 2 dge. Sebuah *assignment* dapat terdiri beberapa *problem*. Untuk setiap *problem*, pengguna dapat
- 3 mengumpulkan *submission* berkali-kali, lalu memilih salah satunya sebagai *final submission*.

#### 3.1.2 Fitur-fitur SharIF Judge

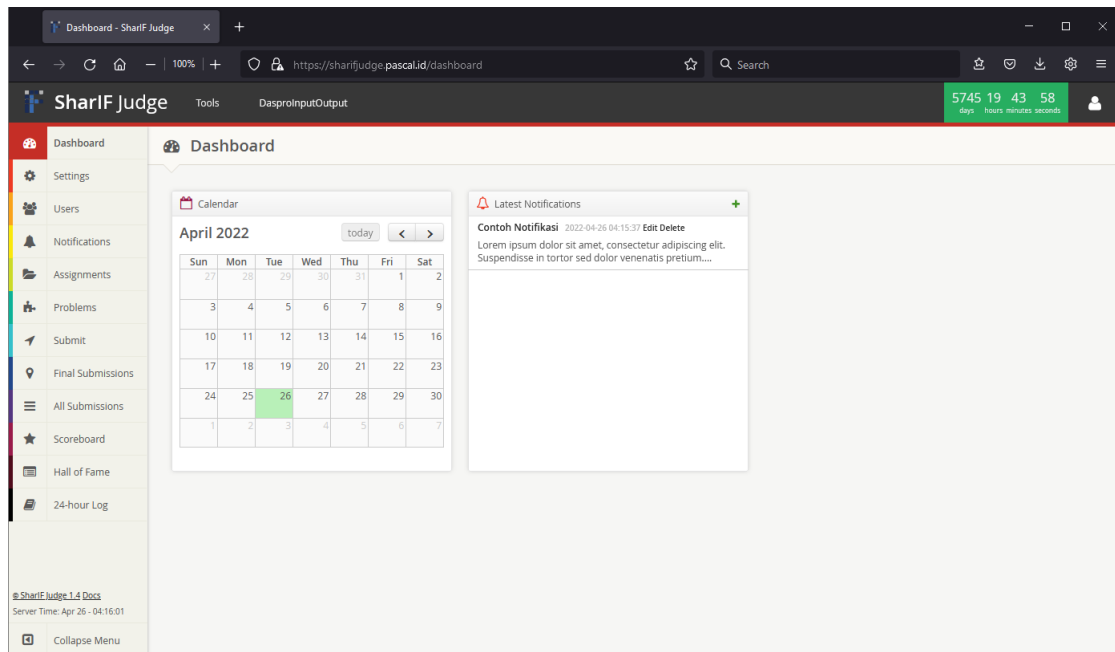


Gambar 3.2: Use Case Diagram SharIF Judge



- 1 Halaman dan tampilan yang tersedia pada pengguna SharIF Judge bergantung pada *role* akun yang  
2 digunakan pengguna tersebut. Pada gambar 3.2, terdapat Use Case Diagram untuk setiap *role* yang  
3 tersedia. Berikut ini adalah setiap halaman dan fitur yang terdapat pada SharIF Judge untuk *role*  
4 *admin*:

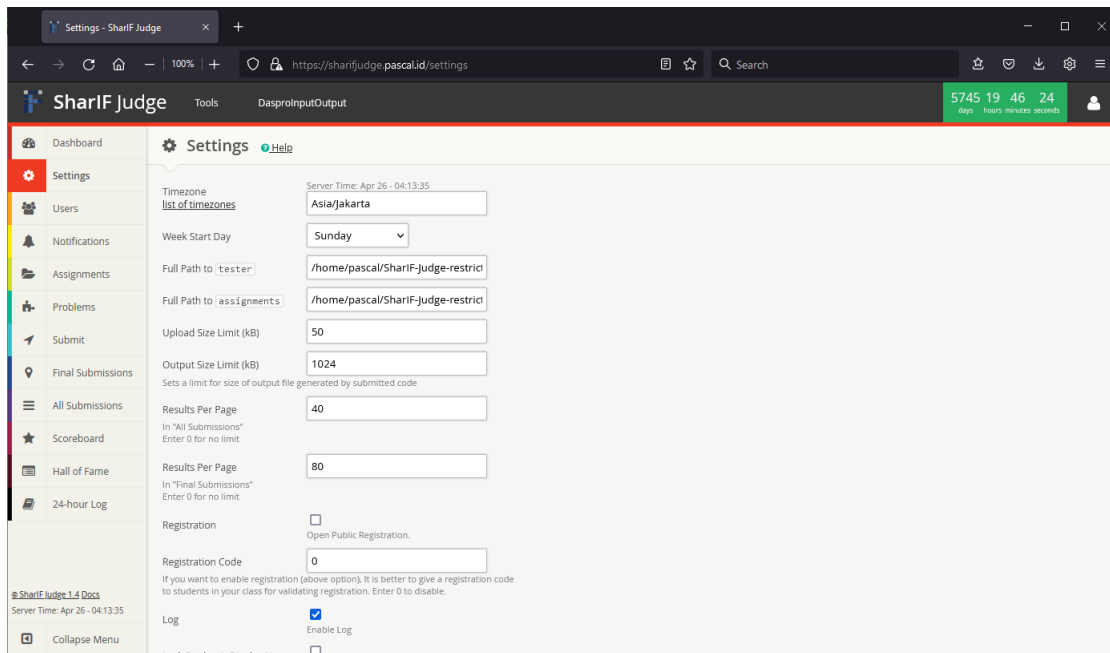
## 5 Dashboard



Gambar 3.3: Halaman Dashboard

- 6 Gambar 3.3 menunjukkan halaman Dashboard. Pada halaman ini terdapat kalender yang menun-  
7 jukkan durasi setiap *assignment* dan daftar notifikasi.

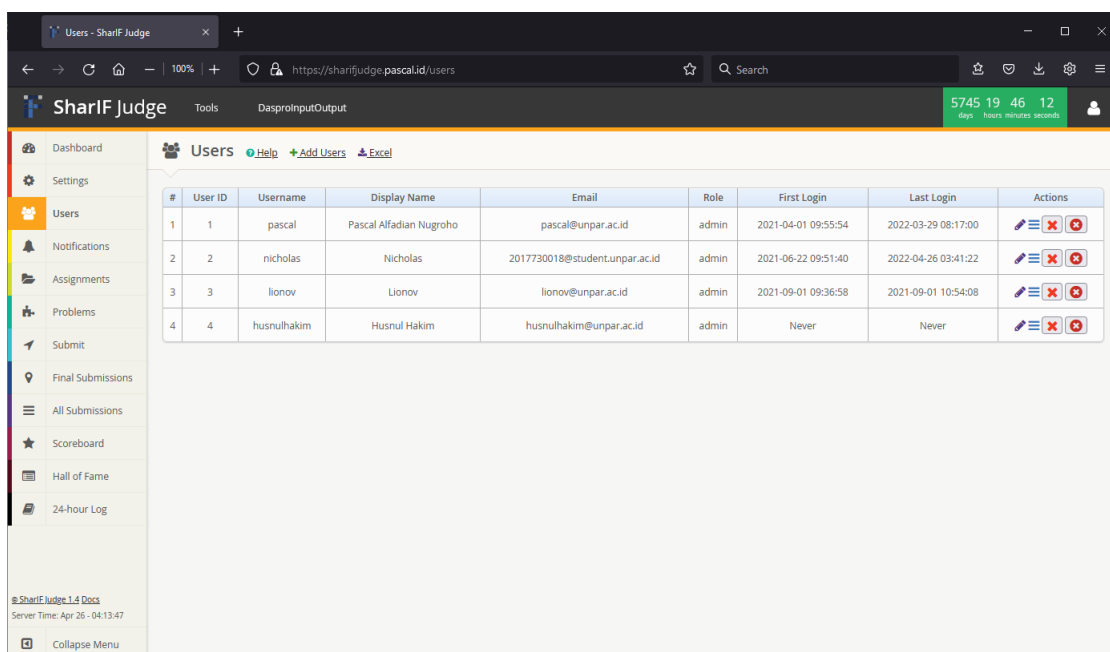
## 1 Settings



Gambar 3.4: Halaman Settings

- 2 Gambar 3.4 menunjukkan halaman Settings. Pada halaman ini terdapat berbagai pengaturan yang  
 3 ada pada SharIF Judge.

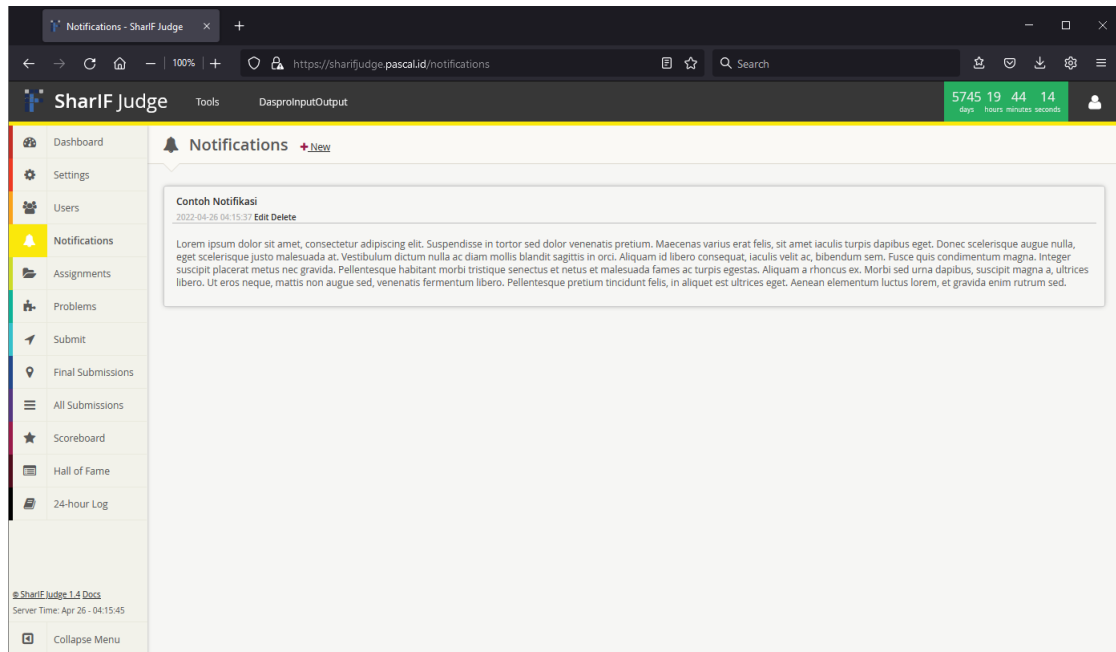
## 4 Users



Gambar 3.5: Halaman Users

- 1 Gambar 3.5 menunjukkan halaman Users. Pada halaman ini terdapat daftar seluruh pengguna
- 2 yang terdaftar pada SharIF Judge. Pengguna juga dapat membuat, mengubah, dan menghapus
- 3 pengguna.

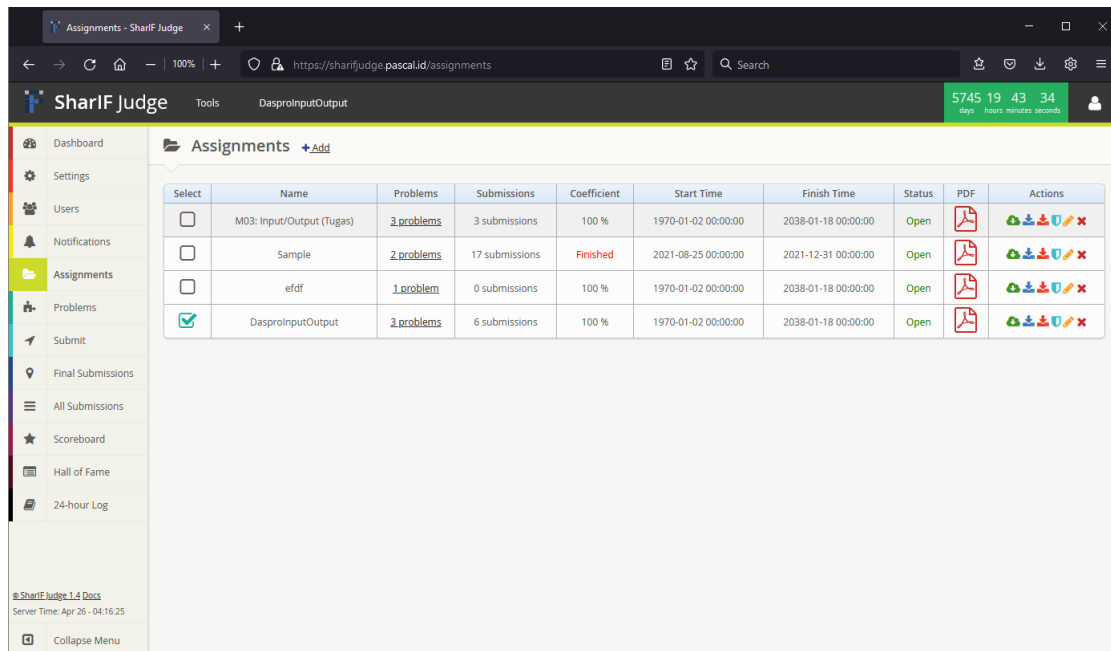
#### 4 Notifications



Gambar 3.6: Halaman Notifications

- 5 Gambar 3.6 menunjukkan halaman Notifications. Pada halaman ini terdapat daftar seluruh notifikasi.
- 6 Pengguna juga dapat membuat, mengubah, dan menghapus notifikasi.

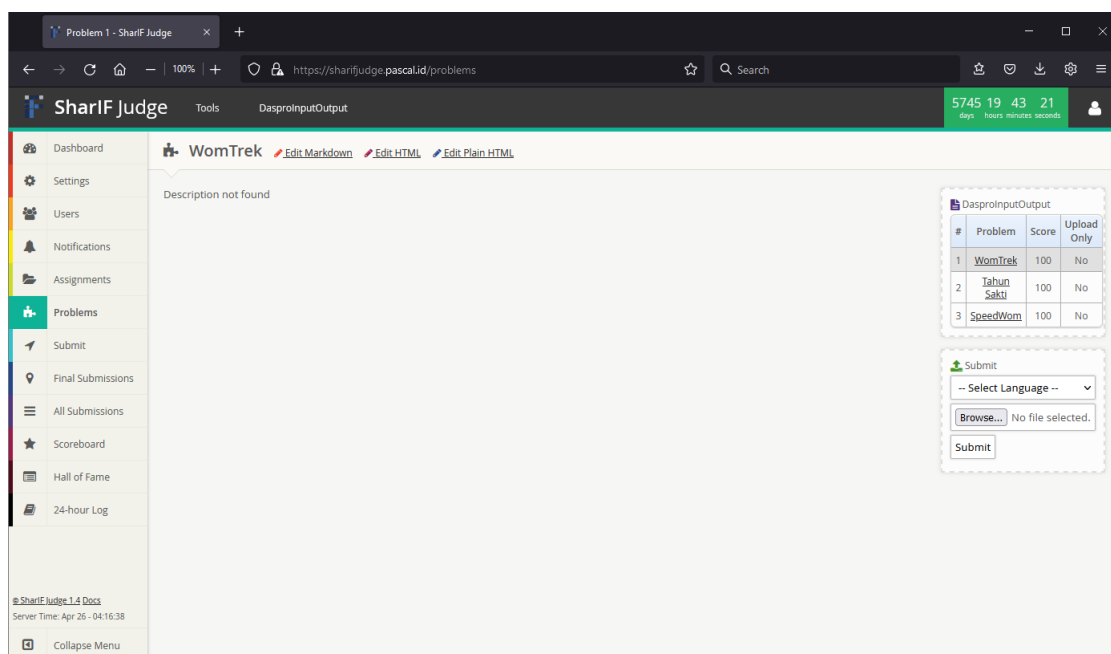
## 1 Assignments



Gambar 3.7: Halaman Assignments

Gambar 3.7 menunjukkan halaman Assignments. Pada halaman ini terdapat daftar seluruh *assignment*. Pengguna juga dapat membuat, mengubah, dan menghapus *assignment*. Salah satu *assignment* pada halaman ini harus dipilih untuk dapat menggunakan beberapa fitur lainnya pada SharIF Judge. Soal dalam bentuk PDF juga dapat diunduh melalui halaman ini.

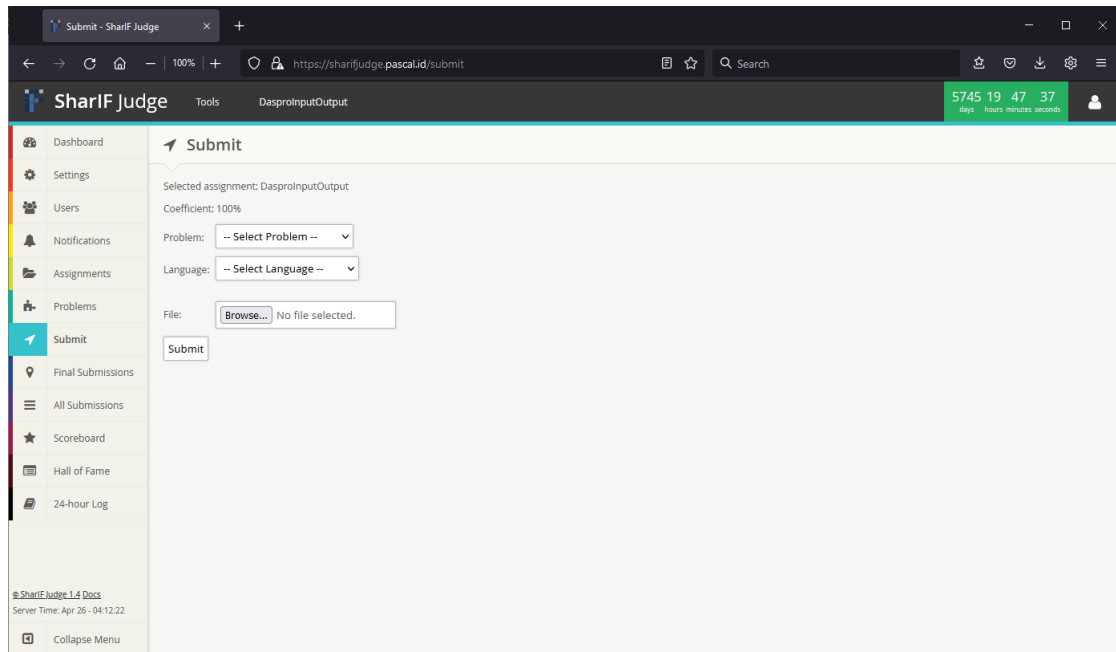
## 6 Problems



Gambar 3.8: Halaman Problems

- 1 Gambar 3.8 menunjukkan halaman Problems. Pada halaman ini terdapat detil dari setiap *problem*
- 2 dari *assignment* yang dipilih. Pengguna juga dapat mengunggah file untuk dikumpulkan sebagai
- 3 *submission* untuk *problem* yang dipilih.

#### 4 Submit



Gambar 3.9: Halaman Submit

- 5 Gambar 3.9 menunjukkan halaman Submit. Pada halaman ini, pengguna dapat mengunggah file
- 6 untuk dikumpulkan sebagai *submission* dari *problem* yang dipilih.

## 1 Final Submissions

Final Submissions of DasproInputOutput

#	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
1	7	nicholas	Nicholas	1	2022-04-26 04:17:16	67	No Delay 100%	67	Java	67	Code	Log	↻
2	6	pascal	Pascal Alfadian Nugroho	1	2021-09-01 09:27:55	67	No Delay 100%	67	Java	67	Code	Log	↻
3	3	pascal	Pascal Alfadian Nugroho	3	2021-06-22 09:48:49	0	No Delay 100%	0	Py 3	0	Code	Log	↻

5745 19 42 35 days hours minutes seconds

SharIF Judge 1.4 Docs  
Server Time: Apr 26 - 04:17:24

Gambar 3.10: Halaman Final Submissions

Gambar 3.10 menunjukkan halaman Final Submissions. Pada halaman ini, terdapat daftar seluruh *final submission* untuk *assignment* yang dipilih. Pengguna juga dapat melihat *file* atau kode yang diunggah, dan nilai yang didapatkannya.

## 5 All Submissions

All Submissions of DasproInputOutput

⚠ You cannot change your final submissions after assignment finishes.

Final	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
✓	7	nicholas	Nicholas	1	2022-04-26 04:17:16	67	No Delay 100%	67	Java	67	Code	Log	↻
✓	6	pascal	Pascal Alfadian Nugroho	1	2021-09-01 09:27:55	67	No Delay 100%	67	Java	67	Code	Log	↻
○	5	pascal	Pascal Alfadian Nugroho	1	2021-09-01 09:02:34	67	No Delay 100%	67	Java	67	Code	Log	↻
○	4	nicholas	Nicholas	1	2021-07-05 11:37:13	0	No Delay 100%	0	Java	Compilation Error	Code	Log	↻
✓	3	pascal	Pascal Alfadian Nugroho	3	2021-06-22 09:48:49	0	No Delay 100%	0	Py 3	0	Code	Log	↻
○	2	pascal	Pascal Alfadian Nugroho	1	2021-06-22 09:48:00	0	No Delay 100%	0	C	Compilation Error	Code	Log	↻
○	1	pascal	Pascal Alfadian Nugroho	1	2021-06-22 09:46:56	0	No Delay 100%	0	Java	0	Code	Log	↻

5745 19 42 16 days hours minutes seconds

SharIF Judge 1.4 Docs  
Server Time: Apr 26 - 04:17:43

Gambar 3.11: Halaman All Submissions

- 1 Gambar 3.11 menunjukkan halaman All Submissions. Pada halaman ini, terdapat daftar seluruh  
2 *submission* untuk *assignment* yang dipilih. Pengguna juga dapat melihat *file* atau kode yang  
3 diunggah, dan nilai yang didapatkannya. Untuk setiap *problem*, sebuah *submission* dapat dipilih  
4 sebagai *final submission* melalui halaman ini.

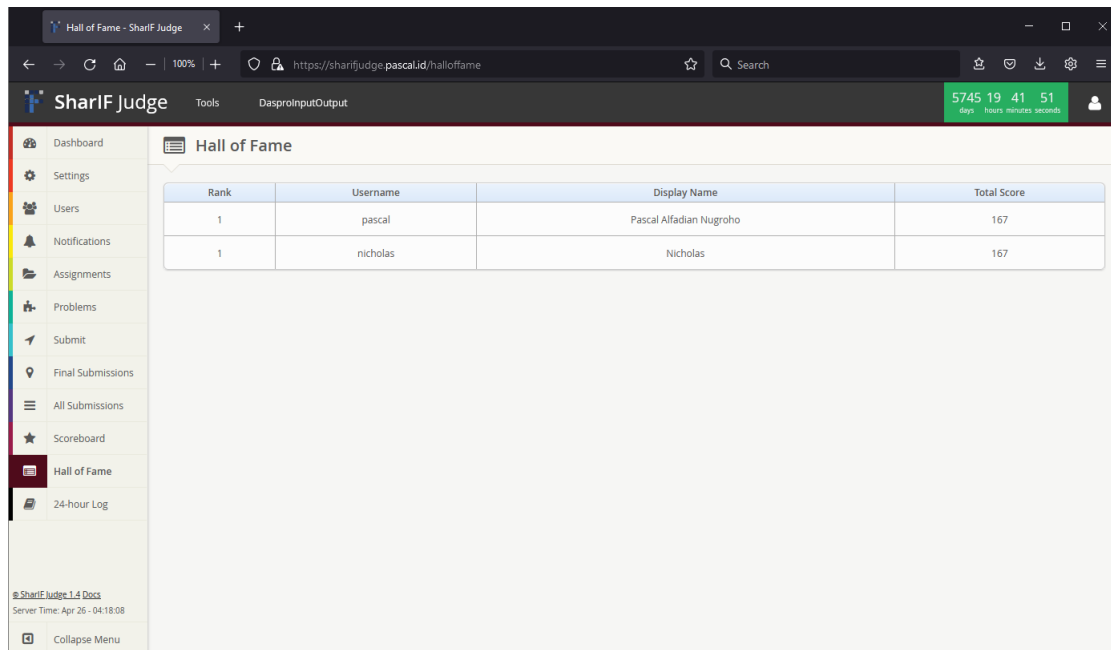
## 5 Scoreboard

#	Username	Name	WomTrek 100	Tahun Sakti 100	SpeedWom 100	Total 300
1	nicholas	Nicholas	67 438572.17	-	-	67 438572.17
2	pascal	Pascal Alfadian Nugroho	67 432889.27	-	0 451185.48	67 934875.41

Gambar 3.12: Halaman Scoreboard

- 6 Gambar 3.12 menunjukkan halaman Scoreboard. Pada halaman ini, terdapat daftar nilai pengguna  
7 untuk setiap *problem* pada *assignment* yang dipilih.

## 1 Hall of Fame

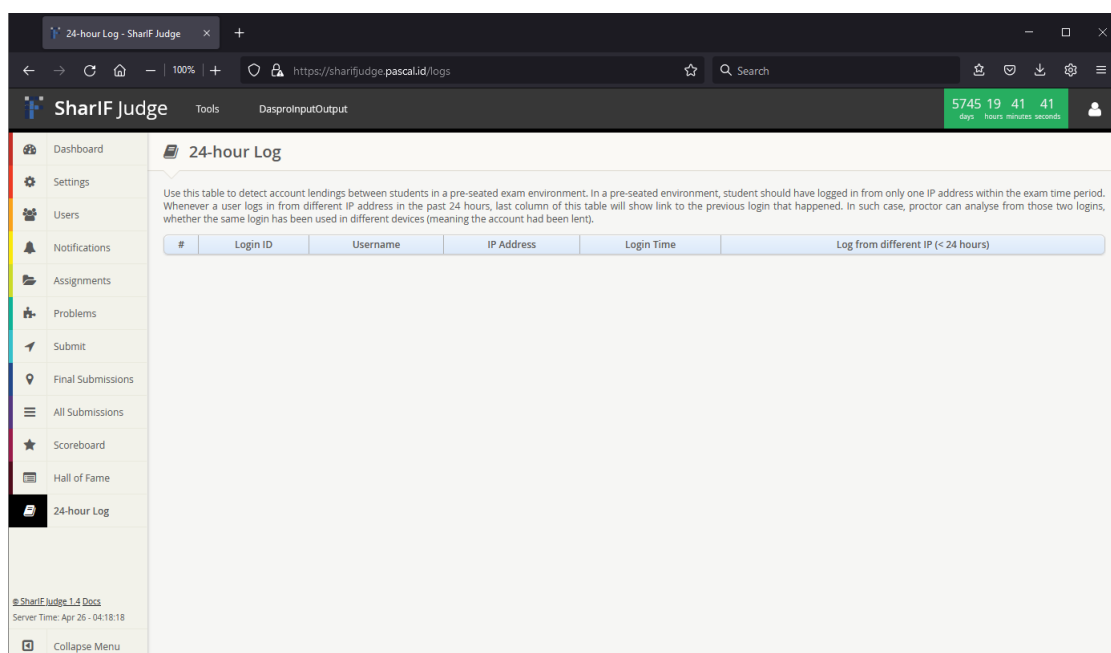


Rank	Username	Display Name	Total Score
1	pascal	Pascal Alfadian Nugroho	167
1	nicholas	Nicholas	167

Gambar 3.13: Halaman Hall of Fame

- 2 Gambar 3.13 menunjukkan halaman Hall of Fame. Pada halaman ini, terdapat daftar pengguna  
 3 secara berurutan berdasarkan total nilai yang didapatkannya dari seluruh *assignment*.

## 4 24-Hour Log



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

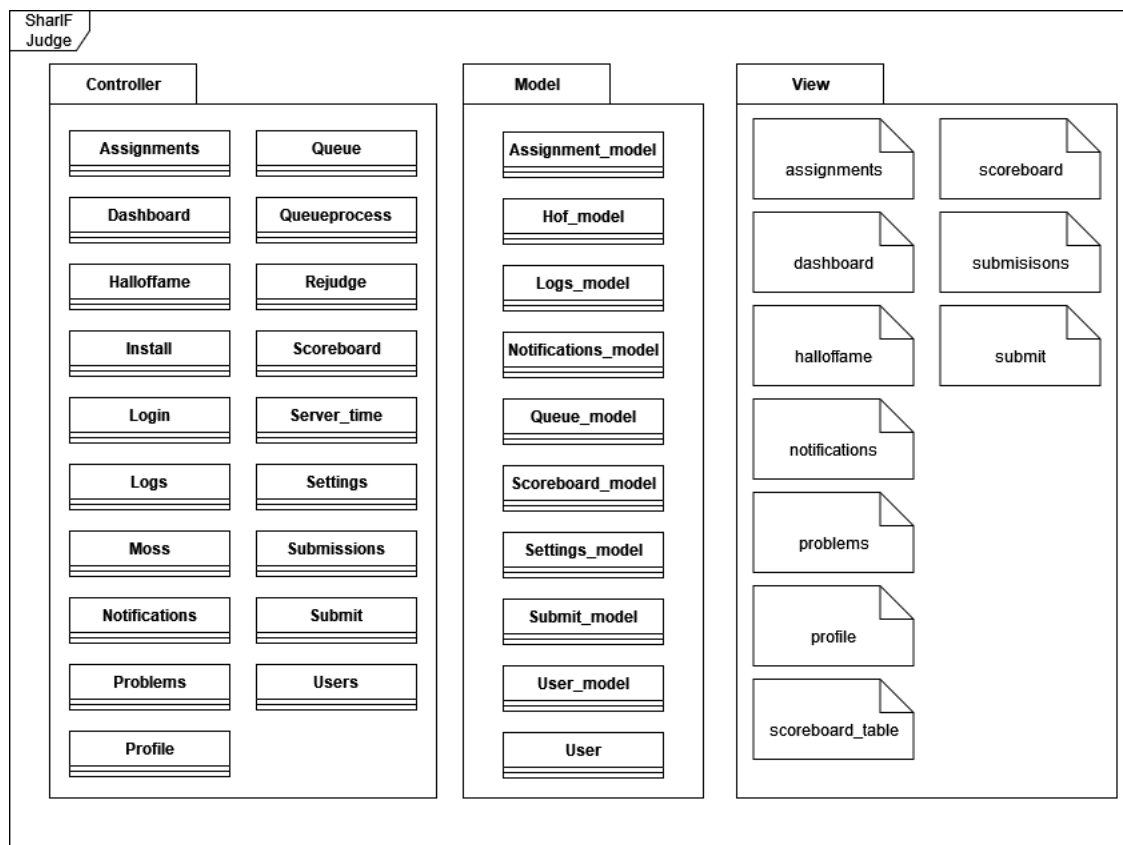
Gambar 3.14: Halaman 24-Hour Log



- 1 Gambar 3.14 menunjukkan halaman 24-Hour Log. Pada halaman ini, terdapat daftar yang mencatat
- 2 bila akun yang sama melakukan *login* dengan *IP address* yang berbeda dalam jangka waktu 24 jam.
- 3 Fitur ini digunakan untuk mendeteksi adanya peminjaman akun.

### 4 3.1.3 Model, View, Controller

- 5 SharIF Judge menggunakan *framework* CodeIgniter 3. Seperti yang dibahas pada bagian 2.1.1,
- 6 *framework* CodeIgniter menerapkan pola arsitektur MVC, dengan komponen-komponen *model*, *view*,
- 7 dan *controller*. Diagram kelas SharIF Judge dapat dilihat pada gambar 3.15. Digunakan simbol
- 8 komentar untuk komponen *view* pada diagram ini, karena *view* tidak berbentuk kelas.



Gambar 3.15: Diagram kelas SharIF Judge

### 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, $edit)`  
Menambah atau memperbarui sebuah *assignment*.
- 15 – `delete_assignment($assignment_id)`  
Menghapus sebuah *assignment*.
- 16 – `all_assignments()`  
Mengambil seluruh *assignment*.

---

```

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

```

```

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

```

Memperbarui sebuah pengaturan.

– `get_all_settings()`

Mengambil seluruh pengaturan.

– `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)`

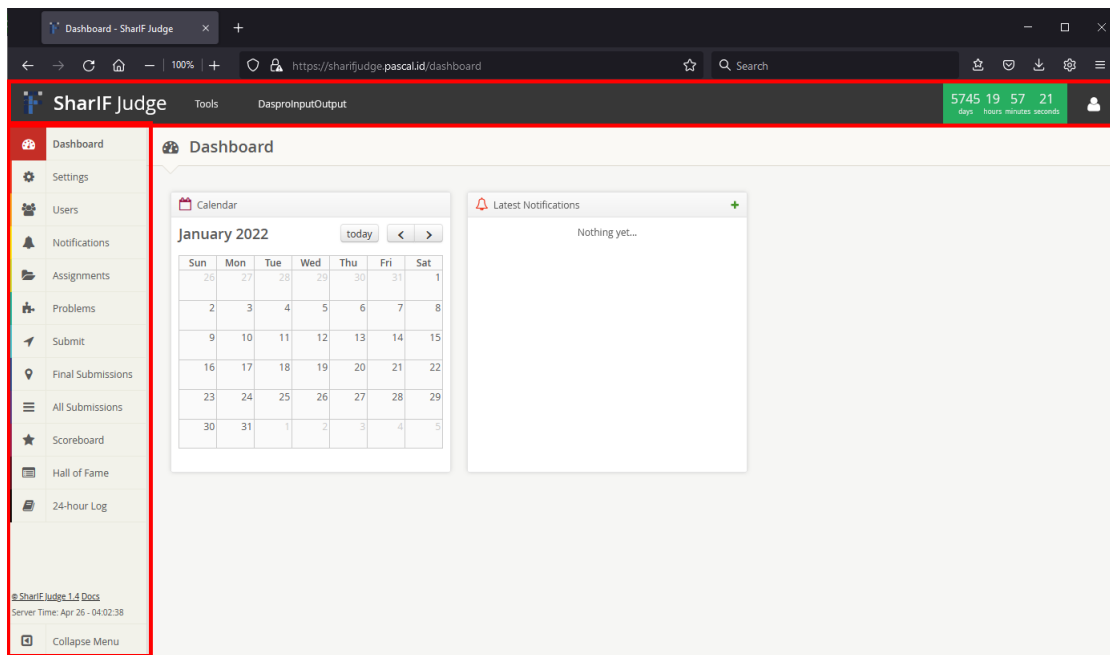
Menambahkan sebuah *user* baru.

– `add_users($text, $send_mail, $delay)`

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

## 26 View

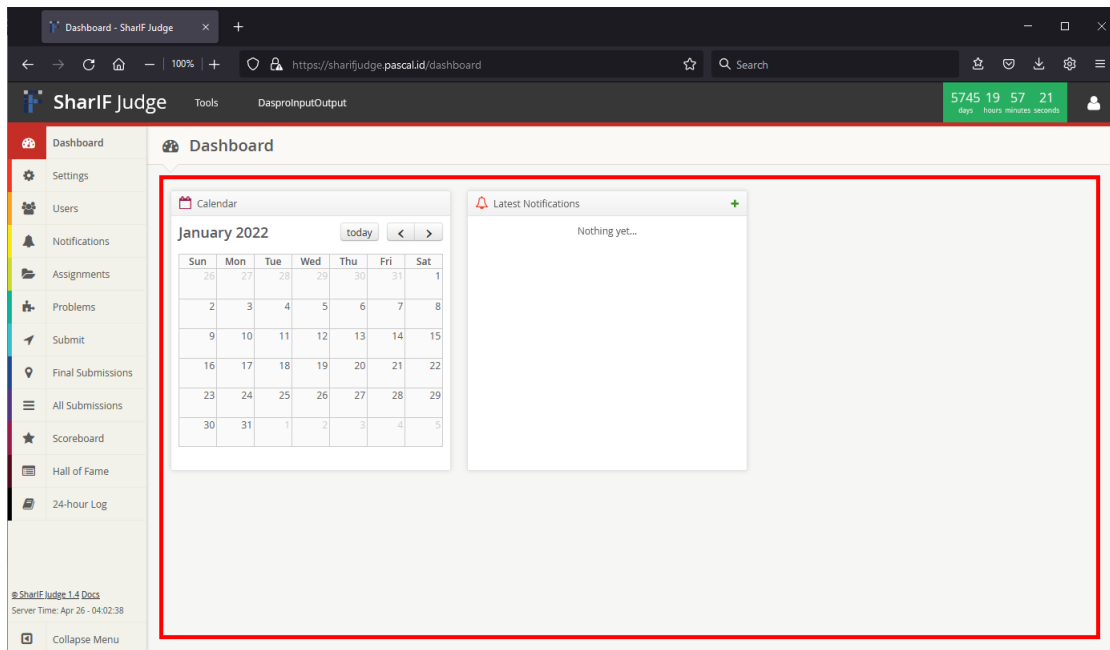
27 *View* pada SharIF Judge terbagi menjadi beberapa folder:



Gambar 3.16: Tampilan *top bar* pada `top_bar.twig` dan *side bar* pada `side_bar.twig`

- **templates**

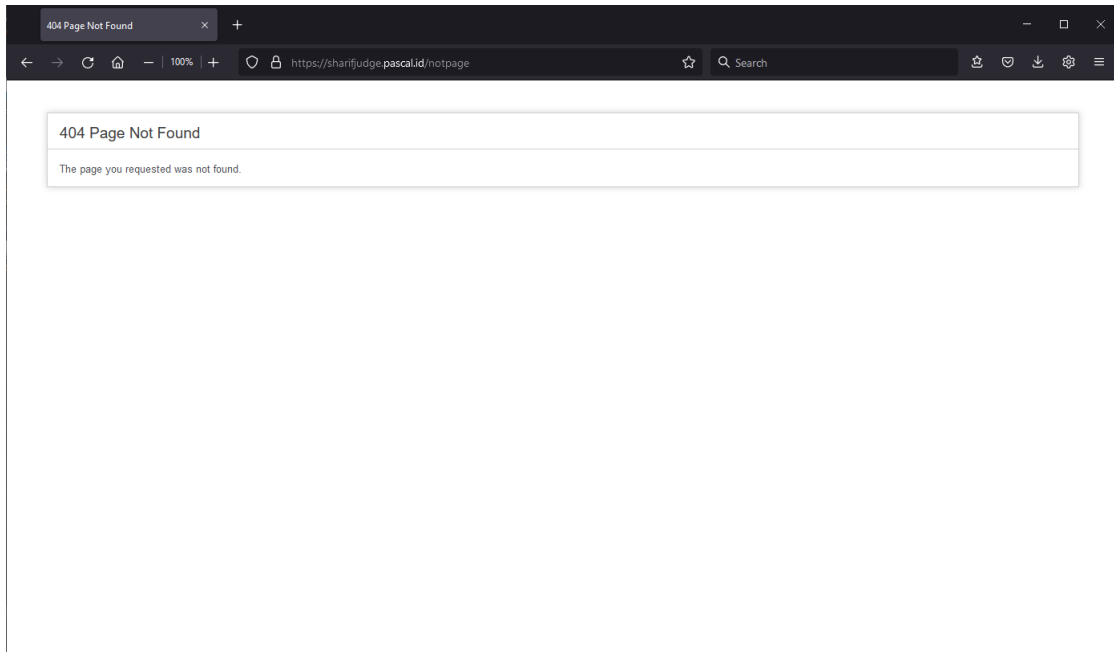
Menyimpan komponen-komponen dasar halaman. Pada gambar 3.16, area yang ditandai dengan kotak merah adalah komponen dasar *top bar* dan *side bar*.



Gambar 3.17: Tampilan Dashboard pada `dashboard.twig`

- **pages**

Menyimpan komponen utama halaman. Pada gambar 3.17, area yang ditandai dengan kotak merah adalah komponen utama halaman Dashboard.



Gambar 3.18: Tampilan halaman *error* pada `error_404.php`

- **errors**

Menyimpan tampilan halaman error. Gambar 3.18 adalah contoh halaman error 404.

### Controller

Berikut ini adalah *controller* pada SharIF Judge:

- **Assignments**

Controller untuk menangani *assignments*. Fungsi yang dimiliki:

- `select()`  
Memilih *assignment* yang sedang ditampilkan.
- `pdf($assignment_id, $problem_id)`  
Mengunduh *file* PDF dari sebuah *assignment*.
- `downloadtestsdesc($assignment_id)`  
Mengunduh *file test case* dari sebuah *assignment*.
- `download_submissions($type, $assignment_id)`  
Mengunduh seluruh *file final submission* dari sebuah *assignment*.
- `delete($assignment_id)`  
Menghapus sebuah *assignment*.
- `add()`  
Menambah atau memperbaiki *assignment*.
- `edit($assignment_id)`  
Memperbarui *assignment*.

- **Dashboard**

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

- `widget_positions()`  
Menyimpan posisi *widget* dari *user*.

- 1     • **Halloffame**  
2         Controller untuk menangani halaman *Hall of Fame* . Fungsi yang dimiliki:  
3             – **hof\_details()**  
4                 Mengambil data yang diperlukan untuk *hall of fame*.
- 5     • **Install**  
6         Controller untuk menangani instalasi SharIF Judge.
- 7     • **Login**  
8         Controller untuk menangani halaman-halaman *login*. Fungsi yang dimiliki:  
9             – **register()**  
10                 Registrasi *user* baru dan menampilkan halaman *register*.  
11             – **logout()**  
12                 Log out user saat ini dan mengalihkan ke halaman *login*.  
13             – **lost()**  
14                 Menangani email dan menampilkan halaman untuk meminta *reset password*.  
15             – **reset(\$passchange\_key)**  
16                 Memproses dan menampilkan halaman untuk ubah *reset password*.
- 17    • **Logs**  
18         Controller untuk menangani halaman *24-hour Log*.  
19             – **index()** Mengambil data yang diperlukan dan menampilkan halaman *24-hour Log*.
- 20    • **Moss**  
21         Controller untuk menangani halaman *Detect Similar Codes* . Fungsi yang dimiliki:  
22             – **update(\$assignment\_id)**  
23                 Memperbarui informasi pada halaman *Detect Similar Codes*.  
24             – **\_detect(\$assignment\_id)**  
25                 Menjalankan Moss untuk mendeteksi kesamaan kode.
- 26    • **Notifications**  
27         Controller untuk menangani halaman *Notifications*. Fungsi yang dimiliki:  
28             – **add()**  
29                 Menambahkan notifikasi baru dan menampilkan halaman *New Notification*.  
30             – **edit(\$notif\_id)**  
31                 Memperbarui sebuah notifikasi.  
32             – **delete()**  
33                 Menghapus sebuah notifikasi.  
34             – **check()**  
35                 Memeriksa adanya notifikasi baru.
- 36    • **Problems**  
37         Controller untuk menangani halaman *Problems*. Fungsi yang dimiliki:  
38             – **index(\$assignment\_id, \$problem\_id = 1)**  
39                 Mengambil data yang diperlukan dan menampilkan halaman *Problems*.  
40             – **edit(\$type = 'md', \$assignment\_id, \$problem\_id = 1)**  
41                 Memperbarui deskripsi *problem* dan menampilkan halaman *Edit Problem Description*.
- 42    • **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()`

- 1        Mengunduh data *final submissions* sebagai file excel.
- 2        – `the_final()`
- 3        Mengambil dan menampilkan data *final submissions* yang akan diunduh.
- 4        – `all()`
- 5        Mengambil dan menampilkan data *submissions* yang akan diunduh.
- 6        – `select()`
- 7        Memilih *final submission*.
- 8        – `view_code()`
- 9        Menampilkan kode, *result*, atau *log* dari *submission*.
- 10       – `download_file()`
- 11       Mengunduh file excel.
- 12    • **Submit**
- 13    Controller untuk menangani *submissions*. Fungsi yang dimiliki:
- 14       – `_language_to_type($language)`
- 15       Mengembalikan kode singkatan dari bahasa pemrograman.
- 16       – `_match($type, $extension)`
- 17       Memeriksa apakah bahasa pemrograman dan tipe file sesuai.
- 18       – `_check_language($str)`
- 19       Memeriksa apakah bahasa pemrograman yang dipilih valid.
- 20       – `index()`
- 21       Mengambil data yang diperlukan dan menampilkan halaman *Submit*.
- 22       – `_upload()`
- 23       Menyimpan file yang diunggah dan menambahkannya ke dalam antrian.
- 24    • **Users**
- 25    Controller untuk menangani halaman *Users*. Fungsi yang dimiliki:
- 26       – `index()`
- 27       Mengambil data yang diperlukan dan menampilkan halaman *Users*.
- 28       – `add()`
- 29       Menambah *user* baru dan menampilkan halaman *Add Users*.
- 30       – `delete()`
- 31       Menghapus *user*.
- 32       – `delete_submissions()`
- 33       Menghapus seluruh *submission* dari sebuah *user*.
- 34       – `list_excel()`
- 35       Menggunakan *library* PHPExcel untuk membuat file excel dari *list user*.

### 3.1.4 Penyimpanan Kode

Pada SharIF Judge, seluruh kode yang diunggah pengguna disimpan dalam *folder assignments*. Lokasi *folder assignments* dapat diubah pada halaman Settings. Kode yang diunggah disimpan pada *folder assignments* dengan format alamat `assignment_<assignment_id>/p_<problem_id>/<nama_pengguna>/<nama_file_yang_diupload>--<submit_id>.<ekstensi_file>`, dengan penjelasan sebagai berikut:

- `<assignment_id>`

id dari *assignment* yang dipilih.

- `<problem_id>`

id dari *problem* yang dipilih.

- `<nama_pengguna>`

Nama pengguna yang mengunggah *file*.

- `<nama_file_yang_diupload>`

Nama *file* yang diunggah.

- `<submit_id>`

id dari *submission* yang diunggah.

- `<ekstensi_file>`

Ekstensi *file* yang diunggah.

Sebagai contoh, seorang pengguna bernama `user` mengunggah sebuah *file* bernama `mycode.java` untuk *problem* ke-3 pada *assignment* dengan id 5. *Submission* pengguna ini adalah *submission* ke-20 untuk *problem* tersebut, sehingga `submit_id` adalah 20. Maka alamat penyimpanan untuk contoh ini adalah `assignment_5/p_3/user/mycode-20.java`.

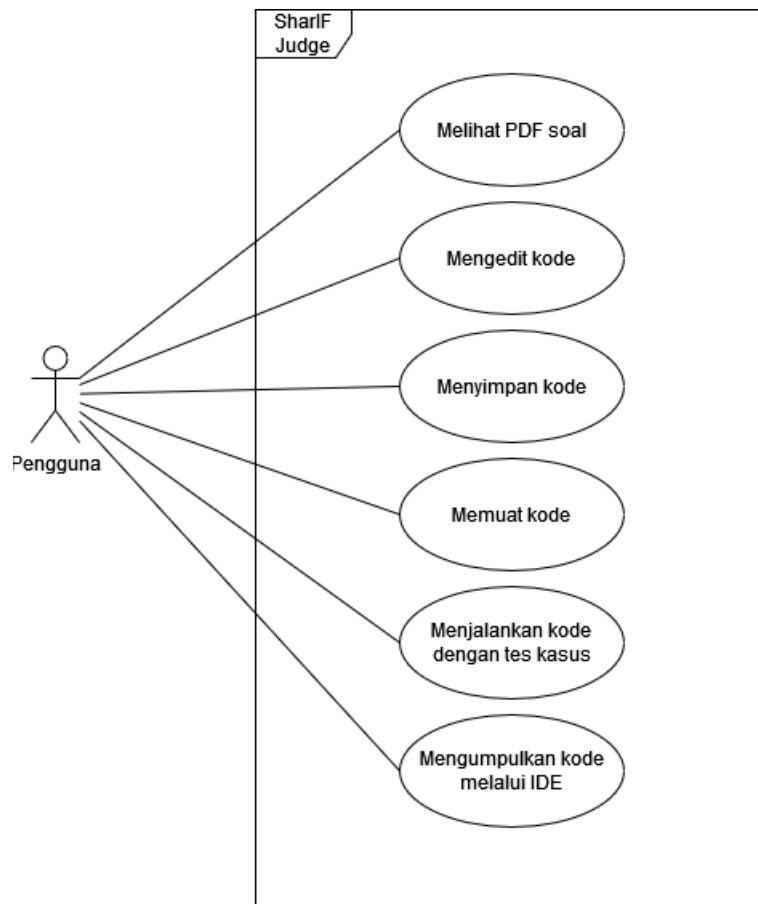
### 3.1.5 Antrean Penilaian Kode

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:

1. *Controller Submit* menyimpan *file* kode pada folder sesuai dengan penjelasan pada bab 3.1.4.
2. *Model Queue\_model* menyimpan *submission* pada basis data `shj_submissions`, lalu dimasukkan dalam antrean pada basis data `shj_queue`.
3. *Controller Submit* memanggil fungsi `process_the_queue()` yang menjalankan fungsi `run()` pada *controller Queueprocess*.
4. *Controller Queueprocess* membaca setiap baris basis data `shj_queue` satu per satu untuk dinilai dengan menjalankan `tester.sh`.
5. `tester.sh` mengompilasi kode, menjalankan kode dengan tes kasus, menilai hasilnya dengan kunci jawaban, lalu mengembalikan hasil penilaian.
6. *Controller Queueprocess* menyimpan nilai kembalian pada basis data `shj_submissions` dan menghapus baris dari basis data `shj_queue`.

## 3.2 Analisis Sistem Usulan

Agar SharIF Judge dapat menjadi sebuah IDE, SharIF Judge harus mampu untuk memfasilitasi seluruh proses pembuatan kode. Pada gambar 3.19, terdapat Use Case Diagram sebagai tambahan dari diagram pada gambar 3.2 untuk fitur yang akan ditambahkan. Seluruh fitur yang akan ditambahkan dapat digunakan oleh setiap *role* pengguna.



Gambar 3.19: Use Case Diagram Fitur Usulan

Berikut ini merupakan skenario untuk masing-masing fitur yang akan ditambahkan pada SharIF Judge:

- Melihat soal

Akan ditambahkan fitur untuk menampilkan soal PDF di halaman submit untuk *assignment* yang dipilih, sehingga pengguna dapat melihat soal secara langsung pada SharIF Judge tanpa perlu mengunduhnya terlebih dahulu.

- Aktor: Pengguna
- Kondisi Awal: PDF soal sudah tersedia
- Kondisi Akhir: PDF soal ditampilkan
- Skenario Normal:
  1. Pengguna memilih menu Submit
  2. Sistem menampilkan halaman Submit
  3. Sistem menampilkan PDF soal
- Pengecualian: PDF soal tidak tersedia

- Mengedit kode

Akan ditambahkan editor teks yang memiliki kemampuan untuk membantu pembuatan kode, seperti *syntax highlighting*, sehingga pengguna dapat mengedit kode secara langsung pada SharIF Judge.

- Aktor: Pengguna

- Kondisi Awal: Pengguna sudah memilih *assignment* dan *problem* yang akan dikerjakan
- Kondisi Akhir: Editor kode ditampilkan dan berfungsi
- Skenario Normal:
  1. Pengguna memilih menu Submit
  2. Sistem menampilkan halaman Submit
  3. Sistem menampilkan editor kode
  4. Pengguna mengedit kode pada editor kode
- Pengecualian: Pengguna belum memilih *assignment* atau *problem*
- Menyimpan kode

Akan ditambahkan fitur untuk menyimpan kode yang sudah dibuat pada editor kode, sehingga pengguna dapat menyimpan kode yang sudah dibuat secara daring.

  - Aktor: Pengguna
  - Kondisi Awal: Pengguna sudah memilih *assignment* dan *problem* yang akan dikerjakan
  - Kondisi Akhir: Kode tersimpan pada sistem
  - Skenario Normal:
    1. Pengguna mengedit kode pada editor kode
    2. Pengguna menekan menu Save
    3. Sistem menyimpan kode
  - Pengecualian: Pengguna belum memilih *assignment* atau *problem*
- Memuat kode

Akan ditambahkan fitur untuk memuat kode yang sudah dibuat pada editor kode, sehingga pengguna dapat melanjutkan kode yang sudah disimpan ketika *problem* yang sama dipilih.

  - Aktor: Pengguna
  - Kondisi Awal: Pengguna sudah menyimpan kode untuk *problem* yang dipilih
  - Kondisi Akhir: Kode dimuat pada editor kode
  - Skenario Normal:
    1. Pengguna memilih *problem* yang akan dikerjakan
    2. Sistem memuat kode pada editor kode
  - Pengecualian: Pengguna belum menyimpan kode untuk *problem* yang dipilih
- Menjalankan kode dengan tes kasus

Akan ditambahkan fitur untuk menjalankan kode pada editor kode dengan tes kasus yang disediakan pengguna, sehingga pengguna dapat menguji kode yang sudah dibuat secara langsung pada SharIF Judge.

  - Aktor: Pengguna
  - Kondisi Awal: Pengguna sudah mengedit kode pada editor kode
  - Kondisi Akhir: Hasil dari kode dengan tes kasus ditampilkan pada area Output
  - Skenario Normal:
    1. Pengguna mengedit kode pada editor kode
    2. Pengguna memasukkan tes kasus pada area Input
    3. Pengguna menekan menu Execute
    4. Sistem menjalankan kode dengan tes kasus
    5. Sistem menampilkan hasil dari kode pada area Output

- 1       – Pengecualian: Pengguna belum mengedit kode pada editor kode
- 2   • Mengumpulkan kode melalui IDE
- 3       Akan ditambahkan fitur untuk mengumpulkan kode yang sudah dibuat pada editor, sehingga
- 4       pengguna dapat mengumpulkan kode yang sudah dibuat sebagai *submission* tanpa perlu
- 5       mengunggah *file*.
- 6       – Aktor: Pengguna
- 7       – Kondisi Awal: Pengguna sudah mengedit kode pada editor kode
- 8       – Kondisi Akhir: Sistem menampilkan halaman All Submissions
- 9       – Skenario Normal:
- 10       1. Pengguna mengedit kode pada editor kode
- 11       2. Pengguna menekan menu Submit
- 12       3. Sistem menjalankan dan menilai kode
- 13       4. Sistem menampilkan hasil penilaian pada halaman All Submissions
- 14       – Pengecualian: Pengguna belum mengedit kode pada editor kode

## BAB 4

## PERANCANGAN

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

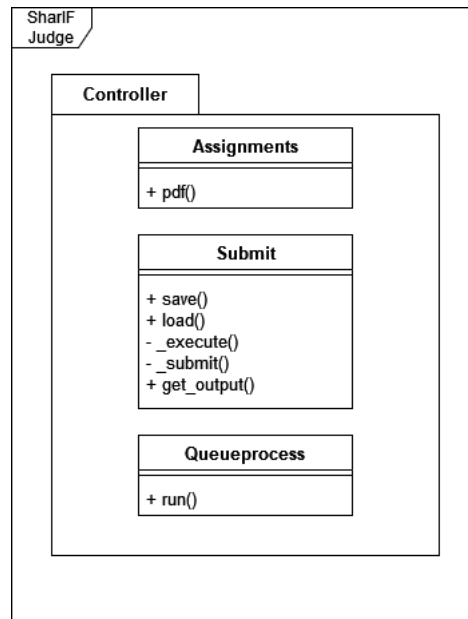
### 4.1 Rancangan Antarmuka

The image shows a web browser window with the title 'SharIF Judge'. The interface is divided into a sidebar on the left and a main content area. The sidebar contains a list of problem categories. The main content area is divided into three sections: 'Submit', 'PDF Viewer', and 'Code Editor'. The 'Submit' section contains dropdowns for 'Problem' and 'Language', a 'File' input with a 'Browse' button, and a 'Submit' button. The 'PDF Viewer' section displays a PDF document. The 'Code Editor' section contains a text area for code, with 'Save', 'Execute', and 'Submit' buttons below it. To the right of the code editor are two columns labeled 'Input' and 'Output' for displaying program results.

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 Rancangan Perubahan Kode



Gambar 4.2: Diagram kelas perubahan pada SharIF Judge

Untuk mengimplementasikan fitur-fitur, diperlukan perubahan kode berikut ini pada SharIF Judge. Diagram kelas pada gambar 4.2 menunjukkan kelas-kelas yang mengalami perubahan.

### 4.2.1 Menampilkan soal

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

Untuk menampilkan soal PDF, dilakukan perubahan sebagai berikut:

- *Controller Assignments*:

- Fungsi `pdf`:

Penambahan kondisi untuk mencegah dialog unduh PDF soal, karena soal akan ditampilkan.

- *View submit*:

- Penambahan elemen `iframe` sebagai tempat untuk menampilkan PDF soal.

### 4.2.2 Mengedit Kode

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

Untuk mengimplementasikan editor kode, perlu dilakukan perubahan sebagai berikut:

- *View submit*:

- Penambahan elemen `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*.



### 4.2.3 Menyimpan 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 saat pengguna menekan tombol Save. Kode disimpan sebagai sebuah *file* teks dengan ekstensi txt untuk memudahkan pemuatan kode dan mencegah tersimpannya banyak file dengan ekstensi berbeda.

Untuk menyimpan kode, perlu dilakukan perubahan sebagai berikut:

- *Controller Submit*:
  - Fungsi *save*:  
Fungsi baru untuk menyimpan kode pada *file* teks.
- *View submit*:
  - Penambahan elemen *button* untuk menyimpan kode.
  - Penambahan *script* untuk memanggil fungsi *save* pada *controller Submit*.

### 4.2.4 Memuat Kode

Kode yang sudah tersimpan sebagai akan otomatis dimuat pada editor kode saat pengguna memilih *problem* pada *dropdown*.

Untuk memuat kode, perlu dilakukan perubahan sebagai berikut:

- *Controller Submit*:
  - Fungsi *load*:  
Fungsi baru untuk memuat kode dari *file* teks.
- *View submit*:
  - Penambahan *script* untuk memanggil fungsi *load* pada *controller Submit*.

### 4.2.5 Menjalankan Kode dengan Tes Kasus

Fitur ini memanfaatkan sistem antrean eksekusi kode yang sudah tersedia pada SharIF Judge. Diperlukan beberapa perubahan agar kode pada editor dapat dimasukkan ke dalam antrean, dijalankan *input* tes kasus, 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* teks dengan ekstensi 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* teks.
- *Controller Queueprocess*:
  - Fungsi *run*:  
Penambahan kondisi untuk menjalankan *tester.sh* tanpa penilaian.

- *View submit*:
  - Penambahan elemen `textarea` untuk *input*.
  - Penambahan elemen `textarea` untuk *output*.
  - Penambahan elemen `button` untuk menjalankan kode.
  - Penambahan *script* untuk memanggil fungsi `save("execute")` pada *controller Submit*.
  - Penambahan *script* untuk memanggil fungsi `get_output` pada *controller Submit*.

#### 4.2.6 Mengumpulkan Kode Melalui IDE

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 elemen `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

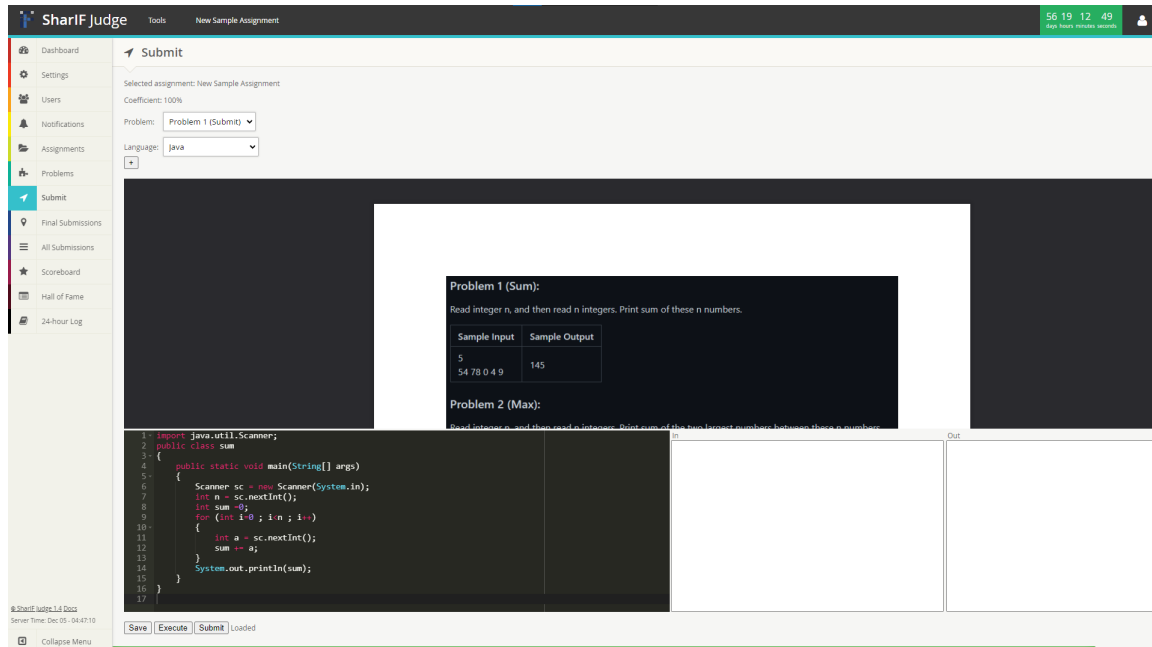
- \* *Storage*: 1TB

- Perangkat Lunak:

- \* *Operating System*: Ubuntu 16.04.6 LTS

## 5.2 Implementasi

### 5.2.1 Tampilan Antarmuka



Gambar 5.1: Tampilan antarmuka halaman Submit

Gambar 5.1 merupakan tampilan antarmuka pada halaman Submit yang sudah diimplementasikan. Seluruh perubahan pada `submit.twig` terdapat pada lampiran A.7. Seluruh *style* dan *script* untuk halaman ini terletak di *file* terpisah. *Stylesheet* yang terdapat di `assets\styles\submit.css` terdapat pada lampiran A.9. *Script* yang terdapat di `assets\js\shj_submit.js` terdapat 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` milik 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 isi PDF akan dikembalikan melalui fungsi `die` dengan header `Content-Type: application/pdf`. Kode untuk perubahan ini terdapat 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 terdapat 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` terdapat 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` terdapat 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` terdapat pada lampiran A.3 baris 39–64.

Fungsi `save` dan `load` dipanggil melalui *AJAX request* pada *view* `Submit`. Fungsi `save` dipanggil ketika tombol `Save` ditekan, sementara fungsi `load` dipanggil ketika pengguna memilih *problem* pada *dropdown*. Kode untuk mengirimkan *AJAX request* tersebut terdapat 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 3.1.5, 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 baris baru pada tabel `shj_submission` yang bersifat sementara untuk menyimpan alamat dan ekstensi *file* kode. Dikarenakan `submit_id` untuk setiap *submission* selalu dimulai dari 1, dapat digunakan `submit_id = 0`.
4. Membuat baris baru pada tabel `shj_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` terdapat pada lampiran A.1.

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 terdapat pada lampiran A.3 baris 193–235.

Fungsi `add_to_queue_exec` dan `delete_exec_submission` juga ditambahkan pada *model* `Queue_model` untuk menambah dan menghapus kode dengan `submit_id = 0` pada tabel `shj_queue`. Perubahan kode ini terdapat pada lampiran A.5.

Beberapa fungsi pada *model* `Submit_model` perlu diubah agar antrean dengan `submit_id = 0` tidak dihitung dan diambil pada fungsi-fungsi tersebut. Perubahan kode ini terdapat pada lampiran A.6.

Perubahan juga dilakukan pada `tester.sh` dengan menambahkan 1 buah parameter untuk membedakan apakah kode akan dinilai atau dijalankan dengan tes kasus dari IDE. Kode akan dijalankan dengan tes kasus yang tersipman pada `exec_in.txt`, kemudian hasilnya akan disimpan pada `exec_out.txt`. Perubahan ini terdapat pada lampiran A.10 baris 515–629.

Untuk mengambil hasil pada `exec_out.txt`, ditambahkan fungsi `get_output` pada *controller* `Submit`. Fungsi `get_output` akan mengembalikan status `TRUE` jika *output* mengandung kalimat "Total Execution Time", yang berarti kode sudah selesai dijalankan. Fungsi ini akan dipanggil melalui *AJAX request* pada *view* `Submit` secara berkala setiap 1 detik, dan akan dihentikan jika status yang dikembalikan `TRUE`. Perubahan kode pada *controller* `Submit` terdapat pada lampiran A.3 baris 240–267. Kode untuk mengirimkan *AJAX request* terdapat pada lampiran A.8 baris 128–176.

## Penamaan File Java

Pada Java, nama *file* harus disesuaikan dengan nama kelas utama pada kode. Hal ini menyebabkan masalah pada fitur ini karena seluruh kode akan disimpan dengan nama `editor`. Karena itu, diperlukan perubahan bagian kompilasi Java pada `tester.sh`. Perubahan ini terdapat pada lampiran A.10 baris 66–121.

Cara pertama yang dilakukan adalah dengan membaca *error* yang dihasilkan Java, yaitu "Class <nama kelas> is public, should be declared in a file named <nama kelas>.java". Untuk mengambil nama kelas dari *error* ini, digunakan *command* berikut:

```
1. grep -e '\<should be declared in a file named\>'
```

Mengambil baris yang mengandung kalimat "should be declared in a file named".

- `-e`

Opsi untuk menggunakan regex sebagai pola yang dicari.

- `'\<should be declared in a file named\>'`

Mencari kalimat "should be declared in a file named".

```
2. grep -Po '[\w]+?(?=\ is public)'
```

Mengambil kata sebelum kalimat " is public".

- -P  
Opsi untuk menggunakan regex Perl agar kelas `[\w]` dapat digunakan.
- -o  
Mengembalikan hanya bagian yang sesuai dengan pola yang dicari.
- '`[\w]+?(?=\ is public)`'  
Mencari 1 kata sebelum kalimat " is public".
  - `[\w]`  
Kelas yang merepresentasikan sebuah kata.
  - `+`  
Mengambil hanya 1 token yang sesuai.
  - `(?=\ is public)`  
Mencari kalimat "is public" tanpa memasukannya ke dalam hasil.

Dengan cara tersebut, nama kelas bisa didapatkan, namun setiap kode harus dikompilasi terlebih dahulu untuk mendapatkan *error* lalu dikompilasi lagi dengan nama yang sesuai. Untuk mencegah hal ini, dicoba untuk mengambil nama kelas utama dari kode dengan *command* berikut:

1. `grep -e 'public class\|public static void main\>'`  
Mengambil semua baris yang mengandung kalimat "public class" atau "public static void main".
  - -e  
Opsi untuk menggunakan regex sebagai pola yang dicari.
  - '`public class\|public static void main\>`'  
Mencari kalimat "public class" atau "public static void main".
2. `grep -B1 'public static void main'`  
Mengambil baris yang mengandung kalimat "public static void main" dan 1 baris sebelumnya.
  - -B1  
Opsi untuk menyertakan 1 baris sebelum.
  - '`public static void main`'  
Mencari kalimat "public static void main".
3. `grep '\<class\>'`  
Mengambil baris yang mengandung kata "class".
  - '`\<class\>`'  
Mencari kata "class".
4. `sed 's/^.*class \+//'`  
Mencari kata dari awal *string* hingga kata "class ", lalu menggantikannya dengan *string* kosong.
  - '`^.*class \+`'  
Mencari kata dari awal *string* hingga kata "class ".
    - `^`  
Mencari awal dari *string*.
    - `.*`  
Mencari semua karakter sebanyak 0 atau lebih kali.
    - `class`  
Mencari kata "class".

```

1      - +
2      Mencari karakter spasi sebanyak 1 atau lebih kali.
3  5. sed 's/ .*$/ /'
4      Mencari karakter spasi dan seluruh karakter setelahnya, lalu menggantikannya dengan string
5      kosong.
6      • ' .*$'
7      Mencari karakter spasi dan seluruh karakter setelahnya hingga akhir dari string.
8      - ' '
9      Mencari karakter spasi.
10     - .*
11     Mencari semua karakter sebanyak 0 atau lebih kali.
12     - $
13     Mencari akhir dari string.

```

### 5.2.6 Mengumpulkan Kode Melalui IDE

Untuk mengumpulkan kode melalui IDE, fungsi `_submit` ditambahkan pada *controller Submit*. Fungsi ini dijalankan oleh fungsi `save("submit")` setelah *file* berhasil disimpan. Kode disimpan kembali dengan ekstensi yang sesuai, lalu informasi kode dimasukkan ke dalam antrian melalui *model Queue\_model* untuk dinilai. Perubahan kode ini terdapat pada lampiran A.3 baris 140–188.

## 5.3 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
1	Membuka halaman Submit	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



## 5.4 Pengujian Eksperimental

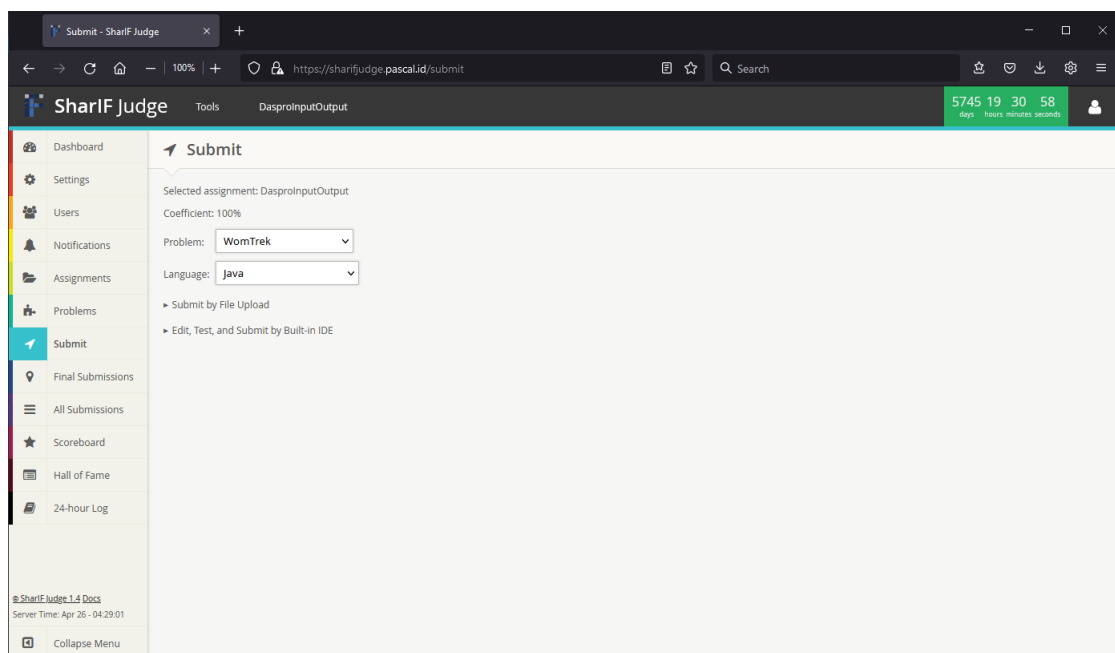
Pengujian eksperimental dilakukan pada mata kuliah Dasar-dasar Pemrograman semester 51 Teknik Informatika Unpar. Perangkat lunak diuji pada *judge* dengan alamat <http://daspro.labftis.net>. Seluruh persoalan dan masukan yang diterima selama mata kuliah Dasar-dasar Pemrograman dicatat pada <https://github.com/athlonneo/SharIF-Judge/issues>.

### 5.4.1 Perubahan melalui GitHub

#### Perubahan Tampilan Antarmuka

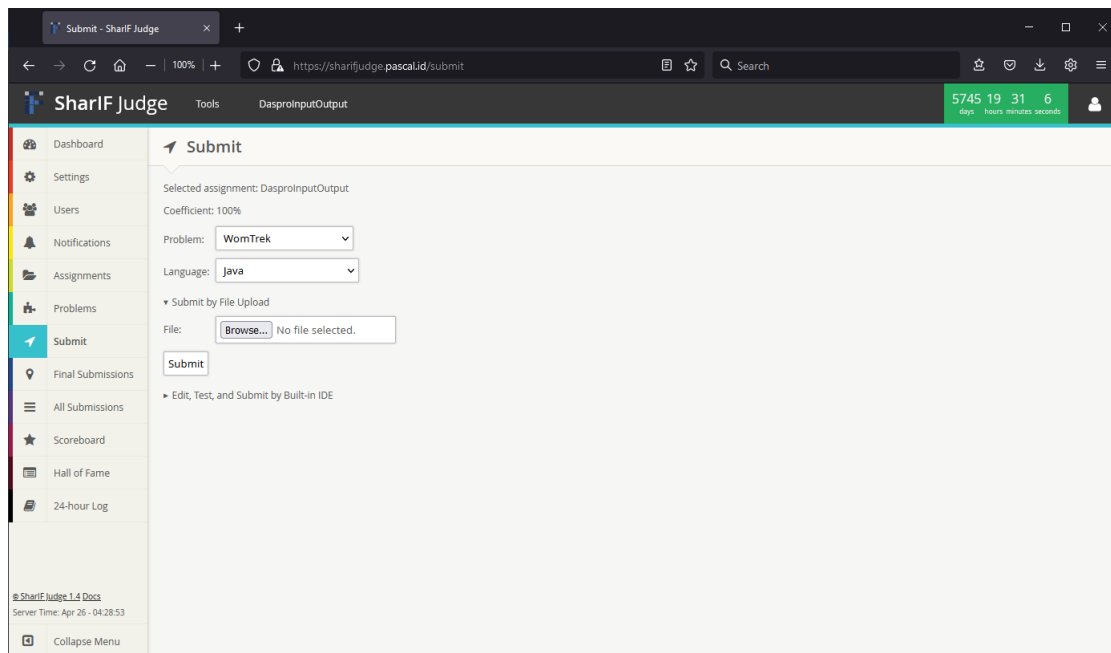
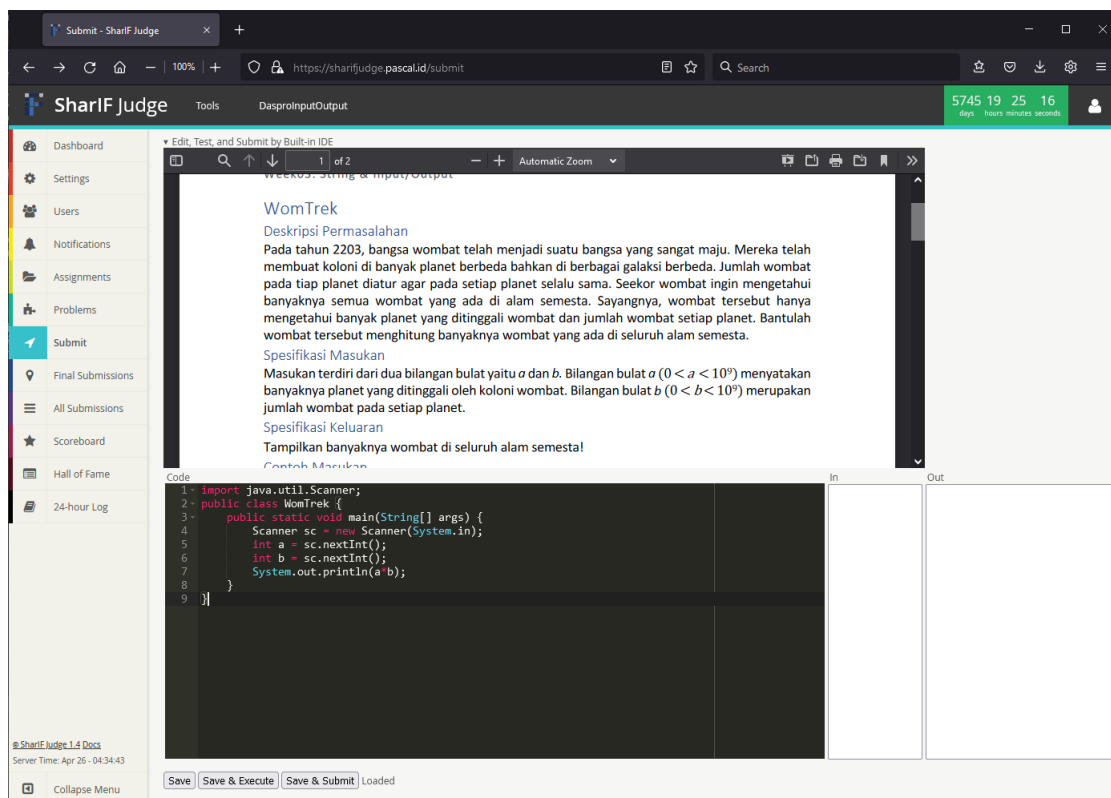
Tercatat pada *issue* #2<sup>1</sup>, masukan dari salah satu dosen Dasar-dasar pemrograman adalah perubahan tampilan antarmuka untuk memperjelas fungsi dan meningkatkan kenyamanan pengguna. Perubahan yang disarankan adalah sebagai berikut:

- Memberi jarak antara PDF Viewer dengan *scrollbar*.
- Meningkatkan ukuran editor kode.
- Mengubah teks tombol "Execute" menjadi "Save & Execute".
- Mengubah teks tombol "Submit" menjadi "Save & Submit".
- Memisahkan antarmuka unggah *file* dengan IDE.



Gambar 5.2: Tampilan antarmuka setelah perubahan

<sup>1</sup><https://github.com/athlonneo/SharIF-Judge/issues/2>

Gambar 5.3: Tampilan antarmuka unggah *file*

Gambar 5.4: Tampilan antarmuka 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.

### 1 *Submit Jawaban Soal Upload Only Melalui IDE*

2 Tercatat pada *issue* #3<sup>2</sup>, ditemukan masalah jika mengumpulkan jawaban untuk soal *upload only*,  
3 *status* pada halaman *Submissions* menjadi "File Not Found".

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

7 Untuk menyelesaikan masalah ini, dimanfaatkan variabel `$this->problems` yang sudah tersedia  
8 pada *controller* `Submit` untuk mengisi variabel `$this->problem['is_upload_only']`. Perubahan  
9 ini terdapat pada kode 5.1 .

Kode 5.1: Perubahan pada `Submit.php`

```
10 @@ -378,8 +378,15 @@ class Submit extends CI_Controller
11 1
12 2     $file_fname = $file_name.'-'.($this->user->selected_assignment['total_submits']+1);
13 3     $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
14 4
15 5 +         foreach($this->problems as $item)
16 6 +             if ($item['id'] == $problem_id)
17 7 +             {
18 8 +                 $this->problem = $item;
19 9 +                 break;
20 0 +             }
21 1 +
22 2     if (!write_file($file_path, $data)){
23 3         $response = json_encode(array(status=>FALSE, message=>'Unable to submit'));
24 4     }
```

### 26 *Tidak Dapat Submit Melalui Unggah File*

27 Tercatat pada *issue* #7<sup>3</sup>, salah satu mahasiswa mencoba melakukan *submit* melalui unggah *file*,  
28 namun tidak tercatat pada halaman *Submissions*. Ketika mahasiswa tersebut mencoba mengunggah  
29 kode yang sama melalui IDE, *submission* berhasil tercatat.

30 Pada *database* tabel `shj_submissions`, kolom `file_name` dan `main_file_name` menyimpan  
31 nama *file* yang diunggah. Nama *file* yang diunggah ditambah dengan `submit_id` untuk membedakan  
32 nama *file* setiap *submission*. Pada kasus ini, `submit_id` sudah mencapai ratusan (3 karakter),  
33 sehingga nama *file* yang diunggah mahasiswa berjumlah 31 karakter ketika ditambah dengan  
34 `submit_id`. Dikarenakan kolom `file_name` dan `main_file_name` disimpan sebagai `varchar(30)`,  
35 nama *file* yang diunggah melebihi batas 30 karakter, sehingga tidak dapat disimpan pada *database*.

36 Solusi untuk menyelesaikan masalah ini adalah dengan meningkatkan batas kolom `file_name` dan  
37 `main_file_name` dari menjadi `varchar(100)`. Perubahan kode pada *controller* `Install` terdapat  
38 pada kode 5.2

39 Karena masalah ini berasal dari Sharif Judge langsung dan tidak berkaitan dengan skripsi  
40 ini, *pull request* ditujukan langsung ke repositori <https://github.com/ifunpar/SharIF-Judge>  
41 sebagai *pull request* #15<sup>4</sup>.

Kode 5.2: Perubahan pada `Install.php`

```
42 @@ -78,8 +78,8 @@ public function index()
43 1
44 2     'status'      => array('type' => 'VARCHAR', 'constraint' => 100),
45 3     'pre_score'   => array('type' => 'INT', 'constraint' => 11),
46 4     'coefficient' => array('type' => 'VARCHAR', 'constraint' => 6),
```

<sup>2</sup><https://github.com/athlonneo/SharIF-Judge/issues/3>

<sup>3</sup><https://github.com/athlonneo/SharIF-Judge/issues/7>

<sup>4</sup><https://github.com/ifunpar/SharIF-Judge/pull/15>

```

15 -         'file_name' => array('type' => 'VARCHAR', 'constraint' => 30),
16 -         'main_file_name' => array('type' => 'VARCHAR', 'constraint' => 30),
17 +         'file_name' => array('type' => 'VARCHAR', 'constraint' => 100),
18 +         'main_file_name' => array('type' => 'VARCHAR', 'constraint' => 100),
19         'file_type' => array('type' => 'VARCHAR', 'constraint' => 6),
20     );
21     $this->dbforge->add_field($fields);

```

## 9 Submit Jawaban txt Melalui IDE

Tercatat pada *issue* #10<sup>5</sup>, sebelumnya IDE akan dinonaktifkan jika tipe file yang dipilih bukan bahasa program, yaitu txt dan zip. Karena editor kode dapat digunakan untuk mengedit file txt juga, disarankan untuk mengaktifkan IDE saat txt dipilih.

Perubahan dilakukan pada `shj_submit.js` untuk menambahkan kondisi jika txt dipilih pada dropdown Language. Jika txt dipilih, maka editor kode akan diaktifkan dengan mode *syntax highlighting* untuk *plain text*. Area Input dan tombol Execute dinonaktifkan karena fitur menjalankan kode tidak dapat digunakan untuk *file* txt.

Kode 5.3: Perubahan pada `shj_submit.js`

```

17 @@ -68,6 +68,12 @@ $(document).ready(function(){
18     editor.session.setMode("ace/mode/c-cpp");
19     disableEditor(false);
20 }
21 + else if(this.value.toLowerCase().includes("txt")){
22 +     editor.session.setMode("ace/mode/plain_text");
23 +     disableEditor(false);
24 +     $("#editor_execute").prop("disabled", true);
25 +     $("#editor_input").prop("disabled", true);
26 + }
27 + else{
28     editor.session.setMode("ace/mode/plain_text");
29     disableEditor(true);
30 }

```

## 32 Tampilan PDF Viewer jika PDF Soal Tidak Tersedia

Tercatat pada *issue* #11<sup>6</sup>, sebelumnya PDF.js akan menampilkan pesan *error* jika file PDF tidak ditemukan. Karena pesan ini kurang deskriptif bagi pengguna, disarankan untuk menyembunyikan PDF *viewer* jika *file* PDF soal tidak tersedia.

Pada *controller* Assignments ditambahkan fungsi `pdfCheck` yang mengembalikan `TRUE` jika *file* PDF tersedia, dan `FALSE` jika *file* PDF tidak tersedia. Fungsi ini akan dipanggil saat *view* Submit dimuat, yang kemudian menentukan apakah PDF *viewer* akan ditampilkan sesuai dengan kembalian yang didapat. Perubahan pada *controller* Assignments terdapat pada kode 5.4, perubahan pada *view* Submit terdapat pada 5.5.

Kode 5.4: Perubahan pada `Assignments.php`

```

41 @@ -588,7 +583,43 @@ class Assignments extends CI_Controller
42     // redirect to add function
43     $this->add();
44 }
45 +
46 +
47 +
48 +
49 + // -----
50 +
51 +
52 +

```

<sup>5</sup><https://github.com/athlonneo/SharIF-Judge/issues/10>

<sup>6</sup><https://github.com/athlonneo/SharIF-Judge/issues/11>

```

12 + /**
13 +  * Check PDF File Availability
14 +  */
15 + public function pdfCheck($assignment_id, $problem_id = NULL)
16 + {
17 +     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
18 +     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
19 +     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
20 +
21 +     // Find pdf file
22 +     if ($problem_id === NULL || $problem_id === "null")
23 +         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/')."/assignment_{ $assignment_id }/*.pdf";
24 +     else
25 +         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/')."/assignment_{ $assignment_id }/p{
26 + $problem_id }/*.pdf";
27 +     $pdf_files = glob($pattern);
28 +
29 +     if ( ! $pdf_files )
30 +         $response = json_encode(array(status=>FALSE));
31 +     elseif ( !$this->assignment_model->assignment_info($assignment_id)['open'] )
32 +         $response = json_encode(array(status=>FALSE));
33 +     elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)['participants
34 + '], $this->user->username) )
35 +         $response = json_encode(array(status=>FALSE));
36 +     elseif ( shj_now() > $finishtime + $extratime )
37 +         $response = json_encode(array(status=>FALSE));
38 +     elseif ( shj_now() < $starttime )
39 +         $response = json_encode(array(status=>FALSE));
40 +     else
41 +         $response = json_encode(array(status=>TRUE));
42 +
43 +     echo $response;
44 + }

```

Kode 5.5: Perubahan pada submit.twig

```

37 @ -15,7 +15,21 @@
38 <link rel="stylesheet" type='text/css' href="{{ base_url('assets/styles/submit.css') }}" />
39 <script>
40     shj.p={};
41     {{ problems_js|raw }}
42     {{ problems_js|raw }};
43     $.ajax({
44         url: "{{ site_url('assignments/pdfCheck/' ~ user.selected_assignment.id) }}",
45         cache: false,
46         success: function(data){
47             data = JSON.parse(data);
48             if(data.status){
49                 $("#pdf_viewer").attr('src', "{{ base_url('assets/pdfjs/web/viewer.html?file=' ~ site_url('assignments/pdf/'
50 ~ user.selected_assignment.id ~ '/null/true') }}" );
51                 $("#pdf_viewer").show();
52             }
53         },
54         error: function (error){
55             console.error(error);
56         },
57     });
58 </script>
59 <script src="{{ base_url('assets/ace/ace.js') }}"></script>
60 <script type='text/javascript' src="{{ base_url('assets/js/shj-submit.js') }}"></script>

```

## 63 Survei

64 Untuk mendapatkan lebih banyak masukan dari pengguna, dibuat sebuah survei untuk diisi oleh  
 65 mahasiswa peserta kuliah Dasar-dasar Pemrograman. Survei ini terbagi menjadi 5 bagian untuk  
 66 masing-masing fitur yang diimplementasikan pada SharIF Judge. Fitur-fitur yang terdapat pada  
 67 survei adalah sebagai berikut:

- 68 1. Menampilkan soal
- 69 2. Editor kode
- 70 3. Menyimpan dan memuat kode

4. Menjalankan kode dengan tes kasus

5. Mengumpulkan kode melalui IDE

Untuk setiap fitur pada survei, terdapat 5 pertanyaan. Pertanyaan yang diberikan adalah sebagai berikut:

1. Apakah Anda sudah mencoba fitur ini? (Partisipan menjawab dengan ya atau tidak)

2. Apakah terdapat kendala saat menggunakan fitur ini? (Partisipan menjawab dengan ya atau tidak)

3. Jika ya, apa kendala yang dialami? (Partisipan menjawab dengan kalimat sendiri)

4. Apakah fitur ini sudah cukup nyaman untuk digunakan? (Partisipan menjawab dengan skala 1 (sangat tidak nyaman) hingga 5 (sangat nyaman))

5. Apakah ada pendapat/saran/masukan untuk fitur ini? (Partisipan menjawab dengan kalimat sendiri)

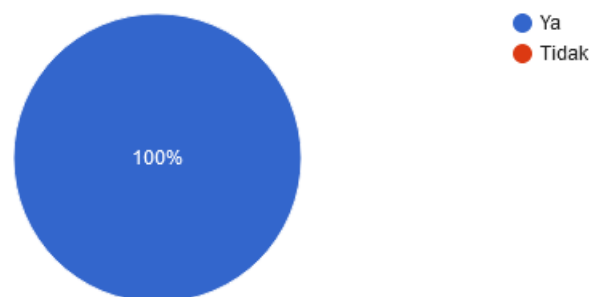
### 5.4.2 Survei

Pada survei ini, didapatkan respon dari 12 orang mahasiswa Dasar-dasar Pemrograman. Berikut ini adalah hasil dari survei untuk setiap fitur:

1. Menampilkan soal

Apakah Anda sudah mencoba fitur ini?

12 responses



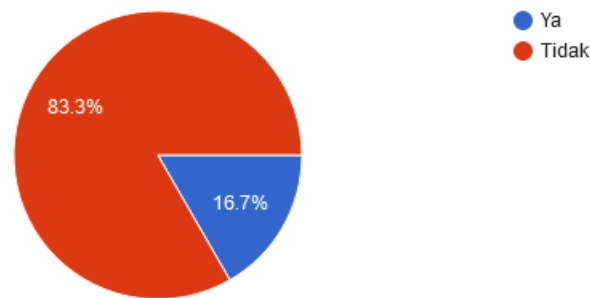
Gambar 5.5: Hasil survei bagian 1 pertanyaan 1

(a) Apakah Anda sudah mencoba fitur ini?

100% dari partisipan menjawab ya, hal ini menunjukkan bahwa seluruh partisipan sudah mencoba fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.5.

Apakah terdapat kendala saat menggunakan fitur ini?

12 responses



Gambar 5.6: Hasil survei bagian 1 pertanyaan 2

(b) Apakah terdapat kendala saat menggunakan fitur ini?

83% dari partisipan menjawab ya, hal ini menunjukkan bahwa mayoritas partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.6.

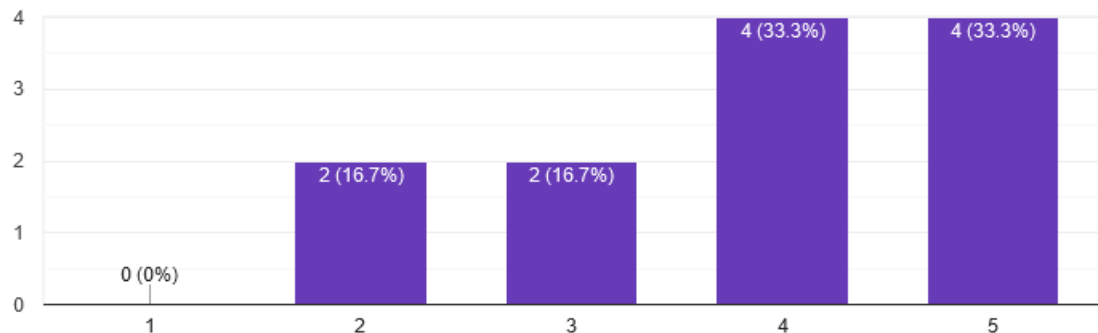
(c) Jika ya, apa kendala yang dialami?

Berikut ini kendala yang dialami:

- Pdf tidak muncul

Apakah fitur ini sudah cukup nyaman untuk digunakan?

12 responses



Gambar 5.7: Hasil survei bagian 1 pertanyaan 4

(d) Apakah fitur ini sudah cukup nyaman untuk digunakan?

Rata-rata skor untuk pertanyaan ini adalah 3.84, dapat disimpulkan bahwa fitur ini sudah cukup nyaman untuk digunakan. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.7.

(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

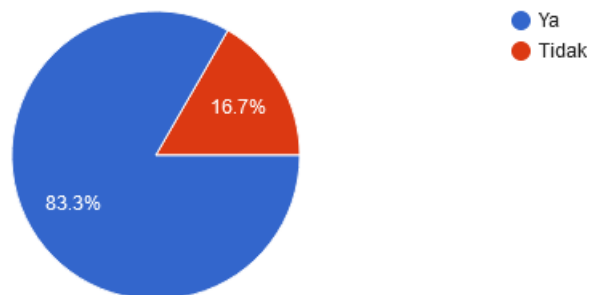
Berikut ini adalah masukan yang didapatkan:

- Pastikan pdf bisa muncul di berbagai *device*

- 1                   • Mengubah tampilan agar soal lebih mudah dibaca
- 2                   • Buat dark theme
- 3    2. Editor kode

Apakah Anda sudah mencoba fitur ini?

12 responses

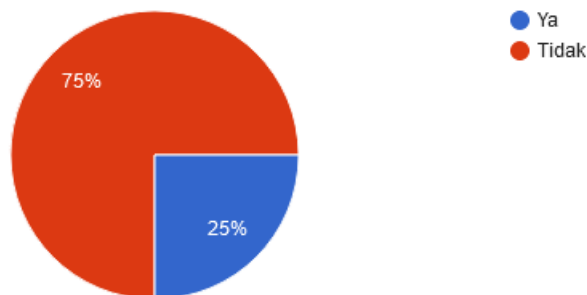


Gambar 5.8: Hasil survei bagian 2 pertanyaan 1

- 4           (a) Apakah Anda sudah mencoba fitur ini?
- 5               83% dari partisipan menjawab ya, hal ini menunjukkan bahwa mayoritas partisipan
- 6               sudah mencoba fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.8.

Apakah terdapat kendala saat menggunakan fitur ini?

12 responses



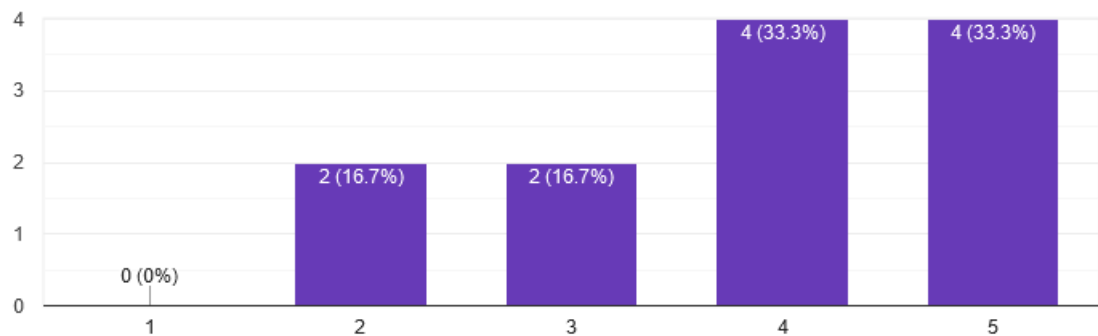
Gambar 5.9: Hasil survei bagian 2 pertanyaan 2

- 7           (b) Apakah terdapat kendala saat menggunakan fitur ini?
- 8               75% dari partisipan menjawab tidak, hal ini menunjukkan bahwa mayoritas partisipan
- 9               tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada
- 10              gambar 5.9.
- 11           (c) Jika ya, apa kendala yang dialami?
- 12               Berikut ini kendala yang dialami:
- 13               • Terkadang jenis file tidak muncul
- 14               • Ukuran editor kurang besar
- 15               • Sulit untuk *scrolling* pada text editor



Apakah fitur ini sudah cukup nyaman untuk digunakan?

12 responses



Gambar 5.10: Hasil survei bagian 2 pertanyaan 4

(d) Apakah fitur ini sudah cukup nyaman untuk digunakan?

Rata-rata skor untuk pertanyaan ini adalah 4.25, dapat disimpulkan bahwa fitur ini sudah cukup nyaman untuk digunakan. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.10.

(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

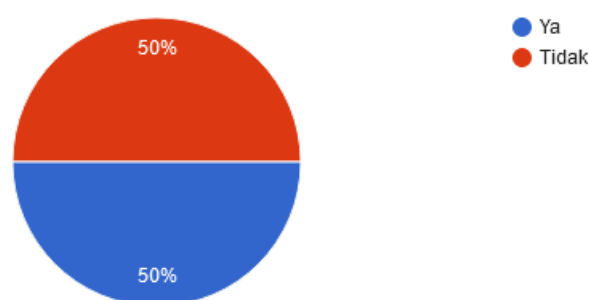
Berikut ini adalah masukan yang didapatkan:

- Mengubah tampilan agar editor lebih mudah digunakan

3. Menyimpan dan memuat kode

Apakah Anda sudah mencoba fitur ini?

12 responses



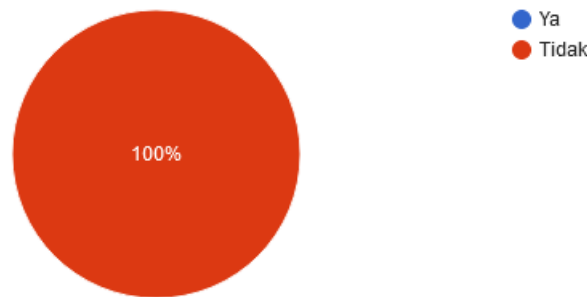
Gambar 5.11: Hasil survei bagian 3 pertanyaan 1

(a) Apakah Anda sudah mencoba fitur ini?

50% dari partisipan menjawab ya, hal ini menunjukkan bahwa hanya sebagian partisipan sudah mencoba fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.11.

Apakah terdapat kendala saat menggunakan fitur ini?

12 responses



Gambar 5.12: Hasil survei bagian 3 pertanyaan 2

(b) Apakah terdapat kendala saat menggunakan fitur ini?

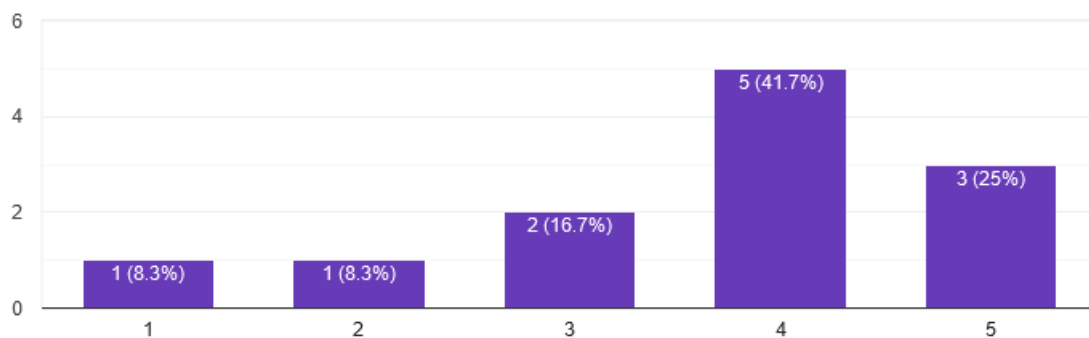
100% dari partisipan menjawab tidak, hal ini menunjukkan bahwa seluruh partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.12.

(c) Jika ya, apa kendala yang dialami?

Tidak ada partisipan yang menemukan kendala pada fitur ini.

Apakah fitur ini sudah cukup nyaman untuk digunakan?

12 responses



Gambar 5.13: Hasil survei bagian 3 pertanyaan 4

(d) Apakah fitur ini sudah cukup nyaman untuk digunakan?

Rata-rata skor untuk pertanyaan ini adalah 3.67, dapat disimpulkan bahwa fitur ini sudah cukup nyaman untuk digunakan. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.13.

(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

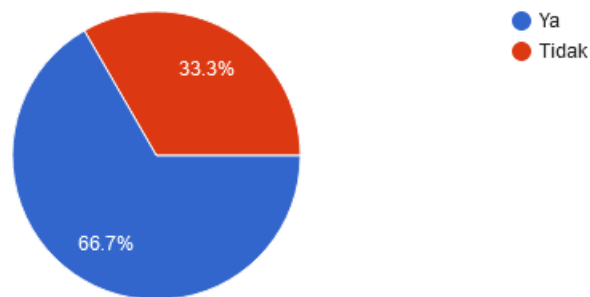
Berikut ini adalah masukan yang didapatkan:

- Tidak perlu ada keterangan Save, cukup Execute dan Submit saja

4. Menjalankan kode dengan tes kasus

Apakah Anda sudah mencoba fitur ini?

12 responses



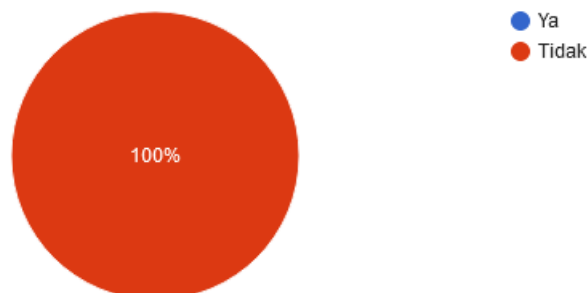
Gambar 5.14: Hasil survei bagian 4 pertanyaan 1

(a) Apakah Anda sudah mencoba fitur ini?

67% dari partisipan menjawab ya, hal ini menunjukkan bahwa hanya sebagian partisipan sudah mencoba fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.14.

Apakah terdapat kendala saat menggunakan fitur ini?

12 responses



Gambar 5.15: Hasil survei bagian 4 pertanyaan 2

(b) Apakah terdapat kendala saat menggunakan fitur ini?

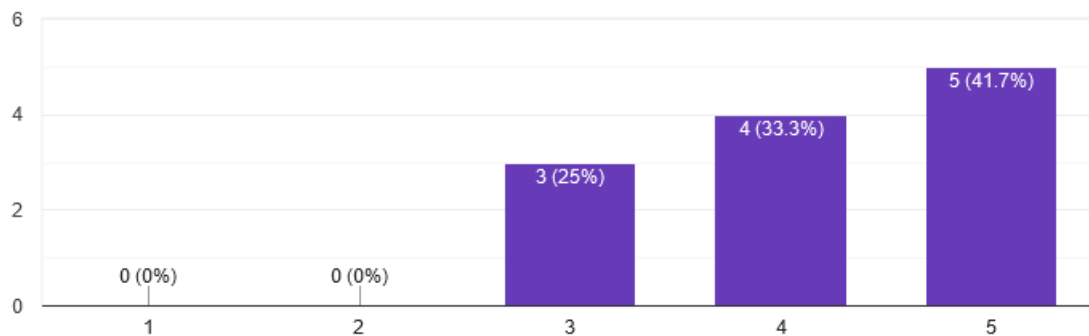
100% dari partisipan menjawab tidak, hal ini menunjukkan bahwa seluruh partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.15.

(c) Jika ya, apa kendala yang dialami?

Tidak ada partisipan yang menemukan kendala pada fitur ini.

Apakah fitur ini sudah cukup nyaman untuk digunakan?

12 responses



Gambar 5.16: Hasil survei bagian 4 pertanyaan 4

(d) Apakah fitur ini sudah cukup nyaman untuk digunakan?

Rata-rata skor untuk pertanyaan ini adalah 4.17, dapat disimpulkan bahwa fitur ini sudah cukup nyaman untuk digunakan. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.16.

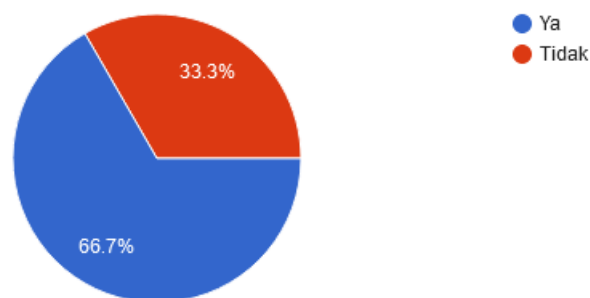
(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

Tidak ada partisipan yang memberi masukan pada fitur ini.

5. Mengumpulkan kode melalui IDE

Apakah Anda sudah mencoba fitur ini?

12 responses



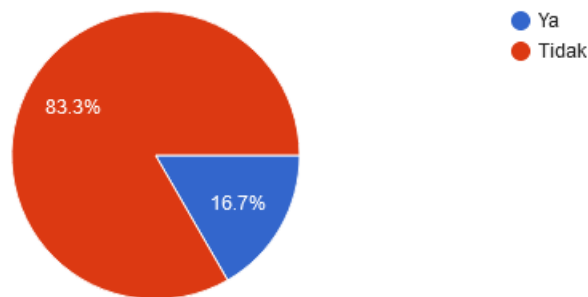
Gambar 5.17: Hasil survei bagian 5 pertanyaan 1

(a) Apakah Anda sudah mencoba fitur ini?

67% dari partisipan menjawab ya, hal ini menunjukkan bahwa hanya sebagian partisipan sudah mencoba fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.17.

Apakah terdapat kendala saat menggunakan fitur ini?

12 responses



Gambar 5.18: Hasil survei bagian 5 pertanyaan 2

(b) Apakah terdapat kendala saat menggunakan fitur ini?

83% dari partisipan menjawab tidak, hal ini menunjukkan bahwa seluruh partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.18.

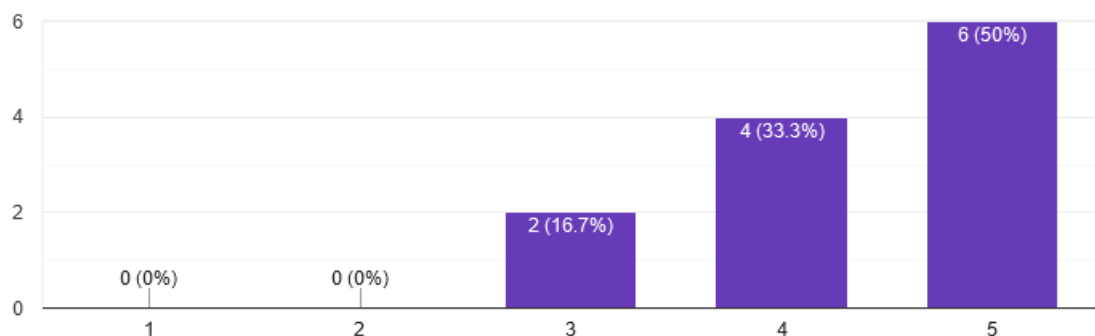
(c) Jika ya, apa kendala yang dialami?

Berikut ini kendala yang dialami:

- Terkadang kode tidak terbaca di judge

Apakah fitur ini sudah cukup nyaman untuk digunakan?

12 responses



Gambar 5.19: Hasil survei bagian 5 pertanyaan 4

(d) Apakah fitur ini sudah cukup nyaman untuk digunakan?

Rata-rata skor untuk pertanyaan ini adalah 4.33, dapat disimpulkan bahwa fitur ini sudah cukup nyaman untuk digunakan. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.19.

(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

Berikut ini adalah masukan yang didapatkan:

- Tidak perlu ada keterangan Save, cukup Submit saja

- 1 Melalui hasil survei, dapat disimpulkan bahwa fitur-fitur yang diimplementasikan pada SharIF
- 2 Judge sudah cukup nyaman untuk digunakan. Sebagian besar masukan dan kendala pada umumnya
- 3 berkaitan dengan antarmuka. Masih banyak mahasiswa yang belum mencoba fitur menyimpan,
- 4 menjalankan, dan mengumpulkan kode. Hal ini kemungkinan disebabkan karena mahasiswa
- 5 didorong untuk menggunakan IDE BlueJ selama kelas Dasar-dasar Pemrograman, yang menyebabkan
- 6 mahasiswa tidak mencoba untuk menggunakan fitur IDE pada SharIF Judge.

## 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 melalui IDE
- Seluruh masukan dan penyelesaian masalah yang ditemukan saat pengujian di mata kuliah Dasar-dasar Pemrograman sudah diimplementasikan, termasuk dengan salah satu penyelesaian masalah yang berasal dari SharIF Judge versi sebelumnya dan tidak berhubungan dengan skripsi ini.
- Melalui hasil survei, dapat disimpulkan bahwa fitur-fitur yang diimplementasikan pada SharIF Judge sudah cukup nyaman untuk digunakan. Namun, terdapat sebagian besar mahasiswa yang belum mencoba beberapa fitur.

### 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 lebih bebas dan berpengalaman dalam menggunakan IDE lain.





## 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.');
```

```
26
27     // Download the file to browser
28     $this->load->helper('download')->helper('file');
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 $response;
65 +         }
66 +     }
67 + // -----
68 +
69 + /**
70 +  * Save code to editor file and submit/execute if needed
71 +  */
72 + public function save($type = FALSE){
73 +     $data = $_POST['code_editor'];
74 +     $problem_id = $_POST['problem_id'];
75 +     $language = $_POST['language'];
76 +
77 +     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'.$this->user->selected_assignment['id'].'/p'.$problem_id.'/'.
78 +     $this->user->username;
79 +     if (!file_exists($user_dir)){
80 +         mkdir($user_dir, 0700);
81 +     }
82 +     $file_path = $user_dir.'/'.EDITOR_FILE_NAME.'.'.EDITOR_FILE_EXT;
83 +     $input_path = $user_dir.'/'.EDITOR_IN_NAME.'.'.EDITOR_FILE_EXT;
84 +
85 +     $this->load->helper('file');
86 +     if (!write_file($file_path, $data)){
87 +         $response = json_encode(array(status=>FALSE, message=>'Unable to save'));
88 +         echo $response;
89 +     }
90 +     else{
91 +         $response = json_encode(array(status=>TRUE, message=>'Saved'));
92 +         if($type === FALSE){
93 +             echo $response;
94 +         }
95 +         else{
96 +             $now = shj_now();
97 +             if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $this->problem['id']
98 +             ')){
99 +                 $response = json_encode(array(status=>FALSE, message=>'You have already submitted for this problem. Your last
100 +                 submission is still in queue.'));
101 +                 echo $response;
102 +             }
103 +         }
104 +     }
105 + }

```

```

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).' '.
35 +             $cmd = "cd $tester_path;\n./tester.sh $problem_dir ".escapeshellarg($username).' '.escapeshellarg($main_filename).' '.
36 +             $op1 $op2 $op3 $op4 $op5 $op6";
37 +             $cmd = "cd $tester_path;\n./tester.sh $problem_dir ".escapeshellarg($username).' '.escapeshellarg($main_filename).' '.
38 +             $op1 $op2 $op3 $op4 $op5 $op6 $op7";
39
40             file_put_contents($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 @@ -221,4 +236,36 @@ class Queue_model extends CI_Model
34     $this->scoreboard_model->update_scoreboard($submission['assignment']);
35 }
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 \ 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         $sarr['assignment'] = $assignment_id;
10        + $sarr['submit_id !='] = EDITOR_SUBMIT_ID;
11        $sarr['is_final'] = 1;
12        if ($user_level === 0) // students can only get final submissions of themselves
13            $sarr['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         $sarr['assignment'] = $assignment_id;
19        + $sarr['submit_id !='] = EDITOR_SUBMIT_ID;
20        if ($user_level === 0)
21            $sarr['username'] = $username;
22        elseif ($filter_user !== NULL)
23            $sarr['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         $sarr['assignment'] = $assignment_id;
28        + $sarr['submit_id !='] = EDITOR_SUBMIT_ID;
29        $sarr['is_final'] = 1;
30        if ($user_level === 0)
31            $sarr['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         $sarr['assignment'] = $assignment_id;
36        + $sarr['submit_id !='] = EDITOR_SUBMIT_ID;
37        if ($user_level === 0)
38            $sarr['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 </p>
37 <p class="input_p">
38 <label for="file" class="tiny">File:</label>
39 <input type="file" id="file" class="sharif_input medium" name="userfile" />
40 {% if upload_state == 'error' %}
41 <div class="shj_error">Error uploading file.</div>
42 {% elseif upload_state == 'ok' %}
43 <div class="shj-ok">File uploaded successfully. See the result in 'All Submissions'.</div>
44 {% endif %}
45 </p>
46 <p class="input_p">
47 <input type="submit" value="Submit" class="sharif_input"/>
48 </p>
49 <br>
50 <details>
51 <summary>Submit by File Upload</summary>
52 <p class="input_p upload_hidden">
53 <label for="file" class="tiny">File:</label>
54 <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: File baru 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             $("#ajax_status").html("Completed");
157             disableEditor(false);
158         }
159     },
160     error: function (error){
161         console.error(error);
162         disableEditor(false);
163     },
164     })());
165 }
166 else{
167     disableEditor(false);
168 }
169 },
170 error: function (error){
171     console.error(error);
172     disableEditor(false);
173 },
174 });
175 });
176 });
177 loadCode($("#select#problems").val());
178 });
179 });

```

Kode A.9: File baru 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 f11d330..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     shj_log "Compiled. Exit Code=$EXITCODE Execution Time: $((COMPILE_END_TIME-COMPILE_BEGIN_TIME)) ms"
81 - if [ $EXITCODE -ne 0 ]; then
82 -     shj_log "Compile Error"
83 -     shj_log "$(cat cerr|head -10)"
84 -     echo '<span class="shj_b">Compile Error</span>' >$PROBLEMPATH/$UN/result.html
85 -     echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
86 -     #filepath=$(echo "${JAIL}/${FILENAME}.${EXT}" | sed 's/\//\\\/g') #replacing / with \
87 -     (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'"/\&quot;/g') >> $PROBLEMPATH/
88 - $UN/result.html
89 -     #(cat $JAIL/cerr) >> $PROBLEMPATH/$UN/result.html
90 -     echo "</span>" >> $PROBLEMPATH/$UN/result.html
91 -     cd ..
92 -     rm -r $JAIL >/dev/null 2>/dev/null
93 -     shj_finish "Compilation Error"
94 + if [ $EXITCODE -ne 0 ]; then
95 +     MSGCHECK=$(cat cerr | grep -e '\<should be declared in a file named\>' | grep -Po '[\w]+?(?=\ is public)')
96 +     echo "$MSGCHECK">> $PROBLEMPATH/$UN/result.html
97 +     if [ -n "${MSGCHECK}" ]; then
98 +         MAINFILENAME=$MSGCHECK
99 +         cp $PROBLEMPATH/$UN/$FILENAME.java $MAINFILENAME.java
100 +         shj_log "Compiling as Java ${MAINFILENAME}.java"
101 +         javac $MAINFILENAME.java >/dev/null 2>cerr
102 +         EXITCODE=$?
103 +         COMPILE_END_TIME=$((date +%sN)/1000000));
104 +         shj_log "Compiled. Exit Code=$EXITCODE Execution Time: $((COMPILE_END_TIME-COMPILE_BEGIN_TIME)) ms"
105 +     fi
106 +     if [ $EXITCODE -ne 0 ]; then
107 +         shj_log "Compile Error"
108 +         shj_log "$(cat cerr|head -10)"
109 +         echo '<span class="shj_b">Compile Error</span>' >$PROBLEMPATH/$UN/result.html
110 +         echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
111 +         ERRMSG=$(cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'"/\&quot;/g')
112 +         echo "$ERRMSG">> $PROBLEMPATH/$UN/result.html
113 +         #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=$((date +%sN)/1000000);
132 @@ -239,6 +281,7 @@ if [ "$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=$((date +%sN)/1000000);
150 @@ -270,6 +315,7 @@ if [ "$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 +%sN)/1000000);
190 -     g++ $PROBLEMPATH/tester.cpp -lm -O2 -o $PROBLEMPATH/tester.executable
191 -     EC=$?
192 -     TST_COMPILE_END_TIME=$((date +%sN)/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 +%sN)/1000000);
203 -             g++ $PROBLEMPATH/tester.cpp -lm -O2 -o $PROBLEMPATH/tester.executable
204 -             EC=$?
205 -             TST_COMPILE_END_TIME=$((date +%sN)/1000000);
206 -             if [ $EC -ne 0 ]; then
207 -                 shj_log "Compiling tester failed."
208 +

```

```

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 ./EXEFIL"
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 -
311 + nosandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
312 +         else
313 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./$EXEFILE"
314 +         fi
315 +         EXITCODE=$?
316 +     fi
317 +
318 +     elif [ "$EXT" = "py2" ]; then
319 -         if $PERL_EXISTS; then
320 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --sandbox
321 - -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
322 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox
323 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -O $FILENAME.py"
324 +         else
325 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "LD_PRELOAD=./EasySandbox.so ./
326 - $EXEFILE"
327 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -O $FILENAME.py"
328 +         fi
329 +         EXITCODE=$?
330 +     fi
331 +
332 +     elif [ "$EXT" = "py3" ]; then
333 -         if $PERL_EXISTS; then
334 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox
335 - -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
336 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox
337 - -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -O $FILENAME.py"
338 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -O $FILENAME.py"
339 +         else
340 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./$EXEFILE"
341 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -O $FILENAME.py"
342 +         fi
343 +         EXITCODE=$?
344 -     fi
345 +
346 +     elif [ "$EXT" = "py2" ]; then
347 -         if $PERL_EXISTS; then
348 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox -l
349 - $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -O $FILENAME.py"
350 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -O $FILENAME.py"
351 +         else
352 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -O $FILENAME.py"
353 +             shj_log "File Format Not Supported"
354 +             cd ..
355 +             rm -r $JAIL >/dev/null 2>/dev/null
356 +             shj_finish "File Format Not Supported"
357 +         fi
358 +         EXITCODE=$?
359 +     fi
360 +
361 +     elif [ "$EXT" = "py3" ]; then
362 -         if $PERL_EXISTS; then
363 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox -l
364 - $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -O $FILENAME.py"
365 +             shj_log "Exit Code = $EXITCODE"
366 +             if ! grep -q "FINISHED" err; then
367 +                 if grep -q "SHJ_TIME" err; then
368 +                     t='grep "SHJ_TIME" err|cut -d" " -f3'
369 +                     shj_log "Time Limit Exceeded ($t s)"
370 +                     echo "<span class=\"shj_o\">Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
371 +                     continue
372 +                 elif grep -q "SHJ_MEM" err; then
373 +                     shj_log "Memory Limit Exceeded"
374 +                     echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
375 +                     continue
376 +                 elif grep -q "SHJ_HANGUP" err; then
377 +                     shj_log "Hang Up"
378 +                     echo "<span class=\"shj_o\">Process hanged up</span>" >>$PROBLEMPATH/$UN/result.html
379 +                     continue
380 +                 elif grep -q "SHJ_SIGNAL" err; then
381 +                     shj_log "Killed by a signal"
382 +                     echo "<span class=\"shj_o\">Killed by a signal</span>" >>$PROBLEMPATH/$UN/result.html
383 +                     continue
384 +                 elif grep -q "SHJ_OUTSIZE" err; then
385 +                     shj_log "Output Size Limit Exceeded"
386 +                     echo "<span class=\"shj_o\">Output Size Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
387 +                     continue
388 +                 fi
389 +             fi
390 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -O $FILENAME.py"
391 +             t='grep "FINISHED" err|cut -d" " -f3'
392 +             shj_log "Time: $t s"
393 +         fi
394 +         EXITCODE=$?
395 +     fi
396 +
397 +     else
398 -         shj_log "File Format Not Supported"
399 +         cd ..
400 +         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 -
441 - if [ $EXITCODE -ne 0 ]; then
442 -     shj_log "Runtime Error"
443 -     echo "<span class=\"shj_o\">Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
444 -     continue
445 - fi
446 -
447 - # checking correctness of output
448 - ACCEPTED=false
449 - if [ -f shj_tester ]; then
450 -     ./shj_tester $PROBLEMPATH/in/input$i.txt $PROBLEMPATH/out/output$i.txt out
451 -     EC=$?
452 -     if [ $EC -eq 0 ]; then
453 -         ACCEPTED=true
454 - +     if [ $EXITCODE -ne 0 ]; then
455 - +         shj_log "Runtime Error"
456 - +         echo "<span class=\"shj_o\">Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
457 - +         continue
458 -     fi
459 - else
460 -     cp $PROBLEMPATH/out/output$i.txt correctout
461 -     if [ "$DIFFOPTION" = "ignore" ]; then
462 -         # Removing all newlines and whitespaces before diff
463 -         tr -d '\t\n\r\f' <out >tmp1 && mv tmp1 out;
464 -         tr -d '\t\n\r\f' <correctout >tmp1 && mv tmp1 correctout;
465 - +
466 - +     # checking correctness of output
467 - +     ACCEPTED=false
468 - +     if [ -f shj_tester ]; then
469 - +         ./shj_tester $PROBLEMPATH/in/input$i.txt $PROBLEMPATH/out/output$i.txt out
470 - +         EC=$?
471 - +         if [ $EC -eq 0 ]; then
472 - +             ACCEPTED=true
473 - +         fi
474 - +     else
475 - +         cp $PROBLEMPATH/out/output$i.txt correctout
476 - +         if [ "$DIFFOPTION" = "ignore" ]; then
477 - +             # Removing all newlines and whitespaces before diff
478 - +             tr -d '\t\n\r\f' <out >tmp1 && mv tmp1 out;
479 - +             tr -d '\t\n\r\f' <correctout >tmp1 && mv tmp1 correctout;
480 - +         fi
481 - +         # Add a newline at the end of both files
482 - +         echo "" >> out
483 - +         echo "" >> correctout
484 - +         if [ "$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 |             DIFFARGUMENT="-q $DIFFARGUMENT"
499 |         +
500 |         if $ACCEPTED; then
501 |             shj_log "ACCEPTED"
502 |             echo "<span class=\"shj_g\">ACCEPT</span>" >>$PROBLEMPATH/$UN/result.html
503 |             ((PASSEDTESTS=PASSEDTESTS+1))
504 |         else
505 |             shj_log "WRONG"
506 |             echo "<span class=\"shj_r\">WRONG</span>" >>$PROBLEMPATH/$UN/result.html
507 |         fi
508 |         # Compare output files
509 |         if $DIFFTOOL $DIFFARGUMENT out correctout >/dev/null 2>/dev/null; then
510 |             ACCEPTED=true
511 |         + done
512 |         +
513 |         +
514 |         +
515 | +#####
516 | +##### EXECUTION #####
517 | +#####
518 |         +
519 |         +else
520 |             touch err
521 |             if [ "$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 |                 else
526 |                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "java -mx${MEMLIMIT}k $JAVA_POLICY
527 | $MAINFILENAME"
528 |                     fi
529 |                     EXITCODE=$?
530 |                     if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
531 |                         shj_log "Memory Limit Exceeded"
532 |                         echo "Memory Limit Exceeded" >>out
533 |                         continue
534 |                     fi
535 |                     if grep -q -m 1 "Exception in" err; then # show Exception
536 |                         javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\.[a-zA-Z\.]*' | head -1 | head -c 80'
537 |                         javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
538 |                         shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
539 |                         # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
540 |                         if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname\$" ../java_exceptions_list; then
541 |                             echo "Runtime Error ($javaexceptionname)" >>out
542 |                         else
543 |                             echo "Runtime Error" >>out
544 |                         fi
545 |                     fi
546 |                     continue
547 |                 fi
548 |                 elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
549 |                     if $PERL_EXISTS; then
550 |                         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout —just-kill —
551 | sandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
552 |                     else
553 |                         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "LD_PRELOAD=./EasySandbox.so ./
554 | $EXEFILE"
555 |                         fi
556 |                         EXITCODE=$?
557 |                         # remove <<entering SECCOMP mode>> from beginning of output:
558 |                         tail -n +2 out >thetemp && mv thetemp out
559 |                     else
560 |                         if $PERL_EXISTS; then
561 |                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout —just-kill —
562 | nosandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./$EXEFILE"
563 |                         else
564 |                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /$EXEFILE"
565 |                         fi
566 |                         EXITCODE=$?
567 |                     fi
568 |                     elif [ "$EXT" = "py2" ]; then
569 |                         if $PERL_EXISTS; then
570 |                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN ". /timeout —just-kill —nosandbox —
571 | l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -0 $FILENAME.py"
572 |                         else
573 |                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "python2 -0 $FILENAME.py"
574 |                         fi
575 |                         EXITCODE=$?
576 |                     elif [ "$EXT" = "py3" ]; 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 python3 -0 $FILENAME.py"
580 |                         else
581 |                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "python3 -0 $FILENAME.py"
582 |                         fi
583 |                         EXITCODE=$?
584 |                     else
585 |                         shj_log "File Format Not Supported"
586 |                         cd ..
587 |                         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
632 # After I added the feature for showing java exception name and exception place,

```