

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

Situasi pandemi Covid-19 menyebabkan seluruh kegiatan kuliah 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*.

Pada mata kuliah pemrograman Teknik Informatika Unpar, dimanfaatkan SharIF Judge, sebuah *online judge* (sebuah sistem *online* yang berfungsi untuk mengevaluasi kode program) untuk mempermudah proses pengumpulan dan penilaian kode program. Pada SharIF judge, pengumpulan kode dilakukan dengan mengunggah kode program, yang biasanya dibuat oleh mahasiswa dengan menggunakan aplikasi eksternal. Karena proses pembuatan kode program tidak terintegrasi, SharIF Judge tidak memiliki kemampuan untuk mengawasi proses pembuatan kode program.

Pada skripsi ini, diimplementasikan *Integrated Development Environment* (IDE), sebuah aplikasi dengan kemampuan untuk mengedit, mengompilasi, dan menjalankan kode program [1] pada SharIF Judge, sehingga seluruh proses pembuatan kode program dapat dilakukan langsung di dalam SharIF Judge. Dengan implementasi IDE, seluruh proses pembuatan kode program terintegrasi pada SharIF Judge, dan selanjutnya dapat dimanfaatkan untuk pengawasan.

SharIF Judge dengan implementasi IDE diuji selama 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 berfungsi dengan baik.

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

ABSTRACT

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.

SharIF Judge is an online judge (an online system that serves to evaluate program code) utilized on several programming courses in Unpar Informatics Engineering Study Program to help with code submission and scoring. Code submission in SharIF Judge was done by file upload, which are made by students using external applications. SharIF Judge was unable to supervise code development process they are not integrated within SharIF Judge.

By implementing Integrated Development Environment (IDE), an application with the ability to write, compile, and run code on SharIF Judge, all code development process is now integrated, and further features can be added to help supervise students during learning activities.

SharIF Judge with IDE implemented is tested on odd semester 2021/2022 *Dasar-dasar Pemrograman* (Basic 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

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 SharIF Judge	5
2.1.1 Instalasi	5
2.1.2 Fitur-fitur	6
2.2 CodeIgniter 3	6
2.2.1 Model-View-Controller	7
2.2.2 URL CodeIgniter	9
2.3 Twig	9
2.4 Bash	9
2.5 PDF.js	10
2.6 Ace	10
3 ANALISIS	13
3.1 Analisis Sistem Kini	13
3.1.1 Istilah-istilah Umum	13
3.1.2 Halaman-halaman pada SharIF Judge	14
3.1.3 Model, View, Controller	20
3.1.4 Penyimpanan Kode	30
3.1.5 Antrean Penilaian Kode	31
3.2 Analisis Sistem Usulan	31
3.2.1 Fitur-fitur	31
3.2.2 Use Case	33
4 PERANCANGAN	37
4.1 Rancangan Antarmuka	37
4.2 Rancangan Perubahan Kode	38
4.2.1 Menampilkan soal	38
4.2.2 Mengedit Kode	38
4.2.3 Menyimpan Kode	39
4.2.4 Memuat Kode	39
4.2.5 Menjalankan Kode dengan Tes Kasus	39

4.2.6	Mengumpulkan Kode Melalui IDE	40
5	IMPLEMENTASI DAN PENGUJIAN	41
5.1	Lingkungan Implementasi dan Pengujian	41
5.2	Implementasi	42
5.2.1	Tampilan Antarmuka	42
5.2.2	Menampilkan Soal	42
5.2.3	Editor Kode	43
5.2.4	Menyimpan dan Memuat Kode	43
5.2.5	Menjalankan Kode dengan Tes Kasus	43
5.2.6	Mengumpulkan Kode Melalui IDE	46
5.3	Pengujian Fungsional	46
5.4	Pengujian Eksperimental	47
5.4.1	Perubahan melalui GitHub	47
5.4.2	Survei	52
6	KESIMPULAN DAN SARAN	61
6.1	Kesimpulan	61
6.2	Saran	61
	DAFTAR REFERENSI	63
	A KODE PROGRAM CONTROLLER	65
	B KODE PROGRAM MODEL	69
	C KODE PROGRAM VIEW	71
	D KODE PROGRAM LAINNYA	75

DAFTAR GAMBAR

1.1 Tahapan penelitian SharIF Judge	2
2.1 Use Case Diagram SharIF Judge	6
2.2 <i>Flow Chart</i> CodeIgniter	7
3.1 Bagan hubungan <i>assignment</i> , <i>problem</i> , dan <i>submission</i> pada SharIF Judge	14
3.2 Halaman Dashboard	14
3.3 Halaman Settings	15
3.4 Halaman Users	15
3.5 Halaman Notifications	16
3.6 Halaman Assignments	16
3.7 Halaman Problems	17
3.8 Halaman Submit	17
3.9 Halaman Final Submissions	18
3.10 Halaman All Submissions	18
3.11 Halaman Scoreboard	19
3.12 Halaman Hall of Fame	19
3.13 Halaman 24-Hour Log	20
3.14 Diagram kelas SharIF Judge	21
3.15 Tampilan Dashboard pada <code>dashboard.twig</code>	25
3.16 Tampilan <i>top bar</i> pada <code>top_bar.twig</code> dan <i>side bar</i> pada <code>side_bar.twig</code>	26
3.17 Tampilan halaman <i>error</i> 404 pada <code>error_404.php</code>	26
3.18 Use Case Diagram Fitur Usulan	34
4.1 Rancangan antarmuka halaman Submit	37
4.2 Diagram kelas perubahan pada SharIF Judge	38
5.1 Tampilan antarmuka halaman Submit	42
5.2 Tampilan antarmuka setelah perubahan	48
5.3 Tampilan antarmuka unggah <i>file</i>	48
5.4 Tampilan antarmuka IDE	49
5.5 Hasil survei bagian 1 pertanyaan 1	53
5.6 Hasil survei bagian 1 pertanyaan 2	53
5.7 Hasil survei bagian 1 pertanyaan 4	54
5.8 Hasil survei bagian 2 pertanyaan 1	54
5.9 Hasil survei bagian 2 pertanyaan 2	55
5.10 Hasil survei bagian 2 pertanyaan 4	55
5.11 Hasil survei bagian 3 pertanyaan 1	56
5.12 Hasil survei bagian 3 pertanyaan 2	56
5.13 Hasil survei bagian 3 pertanyaan 4	57
5.14 Hasil survei bagian 4 pertanyaan 1	57
5.15 Hasil survei bagian 4 pertanyaan 2	58
5.16 Hasil survei bagian 4 pertanyaan 4	58

5.17 Hasil survei bagian 5 pertanyaan 1	59
5.18 Hasil survei bagian 5 pertanyaan 2	59
5.19 Hasil survei bagian 5 pertanyaan 4	60

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dengan adanya situasi pandemi Covid-19 mulai tahun 2020, seluruh kegiatan kuliah wajib dilaksanakan secara *online*. Sebelum pandemi, kegiatan praktikum dan ujian kuliah pemrograman Teknik Informatika Unpar dilaksanakan di lab komputer, sehingga dapat di diawasi secara langsung oleh dosen dan asisten dosen . Namun, pengawasan menjadi lebih sulit untuk dilakukan saat kuliah dilaksanakan secara *online*.

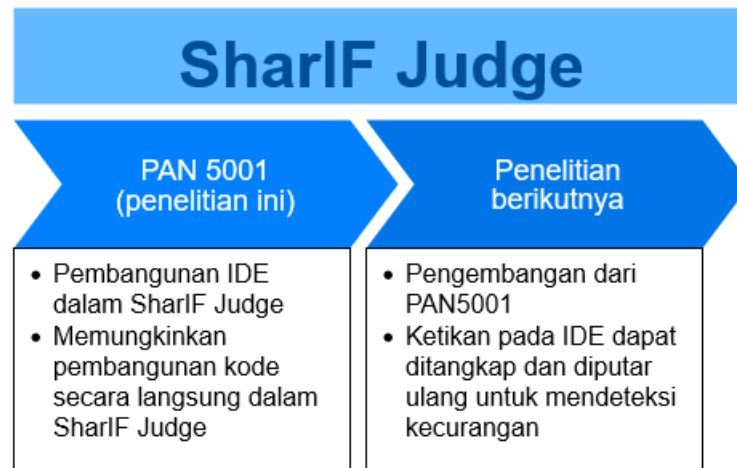
Pada mata kuliah pemrograman Teknik Informatika Unpar, dimanfaatkan SharIF Judge, sebuah *online judge* untuk bahasa pemrograman C, C++, Java dan Python untuk mempermudah proses pengumpulan dan penilaian kode program. *Online judge* adalah sebuah sistem *online* yang berfungsi untuk mengevaluasi kode program yang dikumpulkan oleh pengguna. Kode program yang dikumpulkan akan dikompilasi dan diuji pada lingkungan yang serupa, kemudian diberi nilai berdasarkan hasil yang didapatkan [2].

SharIF Judge (dengan IF kapital) adalah modifikasi cabang oleh Stillmen Vallian terhadap Sharif Judge (dengan if kecil) yang diciptakan oleh Mohammad Javad Naderi dengan *framework* CodeIgniter dan Bash [3]. Modifikasi ini dilakukan untuk menyesuaikan Sharif Judge terhadap kebutuhan spesifik Teknik Informatika Unpar [4].

Pada SharIF judge, pengumpulan kode dilakukan dengan mengunggah kode program, yang biasanya dibuat oleh mahasiswa dengan menggunakan aplikasi eksternal. Karena proses pembuatan kode program tidak terintegrasi, SharIF Judge tidak memiliki kemampuan untuk mengawasi proses pembuatan kode program. Sebelum pandemi, pengawasan dapat dilakukan secara langsung oleh dosen dan asisten dosen di lab komputer. Namun, ketika kuliah dilakukan secara *online*, diperlukan cara lain untuk mengawasi mahasiswa.

Pada skripsi ini, diimplementasikan *Integrated Development Environment* (IDE), sebuah aplikasi dengan kemampuan untuk mengedit, mengompilasi, dan menjalankan kode program [1] pada SharIF Judge, sehingga seluruh proses pembuatan kode program dapat dilakukan langsung di dalam SharIF Judge. Dengan demikian, proses pembuatan kode terintegrasi pada SharIF Judge, dan dapat dimanfaatkan untuk pengawasan.

Gambar 1.1 menggambarkan tahapan penelitian SharIF Judge. Pada skripsi ini, diimplementasikan IDE pada SharIF Judge, sehingga seluruh proses pembuatan kode program 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:

1. Bagaimana mengimplementasikan *Integrated Development Environment* sehingga mahasiswa dapat mengedit, mengompilasi, dan menjalankan kode dalam SharIF Judge?
2. Bagaimana tanggapan pengguna terhadap implementasi *Integrated Development Environment* pada SharIF Judge?

1.3 Tujuan

Tujuan yang ingin dicapai skripsi ini adalah sebagai berikut:

1. Mengimplementasikan *Integrated Development Environment* sehingga mahasiswa dapat mengedit, mengompilasi, dan menjalankan kode dalam SharIF Judge.
2. 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 SharIF Judge

SharIF Judge (dengan IF kapital) adalah modifikasi cabang dari Sharif Judge (dengan if kecil), sebuah *online judge* gratis untuk bahasa pemrograman C, C++, Java dan Python yang diciptakan oleh Mohammad Javad Naderi dengan *framework* CodeIgniter dan Bash[3]. Modifikasi ini dilakukan oleh Stillmen Vallian untuk menyesuaikan Sharif Judge terhadap kebutuhan-kebutuhan spesifik Teknik Informatika Unpar. SharIF Judge digunakan pada beberapa mata kuliah pemrograman Teknik Informatika Unpar untuk mempermudah proses pengumpulan dan penilaian kode program. [4]

2.1.1 Instalasi

Untuk menjalankan SharIF Judge, diperlukan sebuah *server* Linux yang memenuhi prasyarat berikut ini:

- *Webserver* dengan PHP versi 5.3 atau lebih
- PHP Command Line Interface (CLI)
- *Database* MySql atau PostgreSQL
- PHP memiliki akses untuk menjalankan *shell commands* dengan fungsi `shell_exec`
- Kemampuan untuk mengompilasi dan menjalankan kode yang dikumpulkan (`gcc`, `g++`, `javac`, `java`, `python2` dan `python3`)
- Perl

Pada perangkat yang sudah memenuhi prasyarat, instalasi SharIF Judge dapat dilakukan dengan cara berikut ini:

1. Unduh versi terakhir dari Sharif Judge dan menempatkannya pada direktori publik.
2. Pindahkan folder `system` dan `application` ke luar direktori publik. Kemudian simpan alamatnya pada `index.php`.
3. Buat sebuah *Database* MySql atau PostgreSQL.
4. Atur pengaturan koneksi *database* pada `application/config/database.php`.
5. Atur agar direktori `application/cache/Twig` dapat ditulis oleh php.
6. Buka halaman utama SharIF Judge pada *browser* dan ikuti proses instalasi.
7. Log in dengan akun admin
8. Pindahkan folder `tester` dan `assignments` ke luar direktori publik. Kemudian simpan alamatnya pada halaman pengaturan.

2.1.2 Fitur-fitur

Pada SharIF Judge, pengguna terbagi menjadi beberapa *role*. *Role* yang tersedia pada SharIF Judge adalah sebagai berikut:

1. *admin*
2. *head instructor*
3. *instructor*
4. *student*

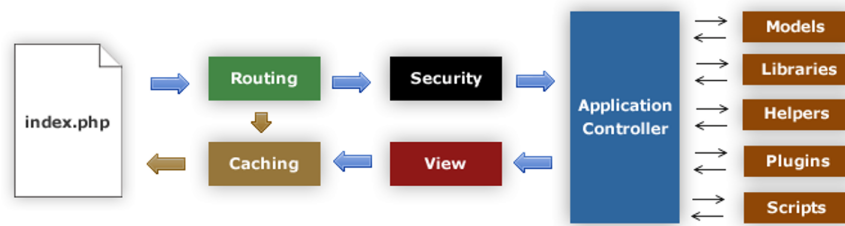
Halaman dan tampilan yang tersedia pada pengguna SharIF Judge bergantung pada *role* akun yang digunakan pengguna tersebut. Pada gambar 2.1, terdapat Use Case Diagram untuk setiap *role* yang tersedia.



Gambar 2.1: Use Case Diagram SharIF Judge

2.2 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 2.2: *Flow Chart* CodeIgniter

Gambar 2.2 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.

2.2.1 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*

```

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

```

```

24         $this->title    = $_POST['title'];
25         $this->content  = $_POST['content'];
26         $this->date     = time();
27
28         $this->db->update('entries', $this, array('id' => $_POST['id']));
29     }
30
31 }

```

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*

```

15 <html>
16 <head>
17     <title>My Blog</title>
18 </head>
19 <body>
20     <h1>Welcome to my Blog!</h1>
21 </body>
22 </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*

```

31 class Blog extends CI_Controller {
32
33     public function index()
34     {
35         echo 'Hello_World!';
36     }
37
38     public function comments()
39     {
40         echo 'Look_at_this!';
41     }
42 }

```

Kode 2.3 merupakan contoh sebuah kelas *controller* pada CodeIgniter. Kelas tersebut mengekstensi `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*

```

48 class Blog_controller extends CI_Controller {
49
50     public function blog()
51     {
52         $this->load->model('blog');
53
54         $data['query'] = $this->blog->get_last_ten_entries();
55     }
56 }

```

```

19         $this->load->view('blog', $data);
20     }
31 }
42 }

```

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.

2.2.2 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.

2.3 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

```

20 <!DOCTYPE html>
21 <html>
22 <head>
23 <title>My Webpage</title>
24 </head>
25 <body>
26 <ul id="navigation">
27 {% for item in navigation %}
28 <li><a href="{{_item.href_}}">{{ item.caption }}</a></li>
29 {% endfor %}
30 </ul>
31 <h1>My Webpage</h1>
32 {{ a_variable }}
33 </body>
34 </html>
35
36

```

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*.

2.4 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**.

2.5 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

```

19 <!DOCTYPE html>
20 <html>
21   <iframe src="/web/viewer.html?file=sample.pdf"></iframe>
22 </html>
23
24

```

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**.

2.6 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*.

– `setTheme(String style)`

Mengubah tema editor.

– `setValue(String val, Number cursorPos)`

Mengubah isi teks editor.

- **Range**

Mengindikasi sebuah daerah pada editor.

- **Scrollbar**

Menangani *scrollbar* editor.

- **Search**

Menangani seluruh operasi pencarian teks pada dokumen.

- **Selection**

Menyimpan posisi kursor dan seleksi teks pada editor.

- **TokenIterator**

Menyediakan fungsi untuk membaca dokumen sebagai aliran token.

- **Tokenizer**

Menerima sejumlah aturan dan membuat **Tokenizer**.

- **UndoManager**
Menangani fungsi *undo* pada editor.
- **VirtualRenderer**
Menggambar tampilan yang terlihat di layar.

Kode 2.7: Contoh kode untuk menggunakan Ace

```
5
61 <!DOCTYPE html>
72 <html>
83 <head>
94 <title>ACE in Action</title>
105 </head>
116 <body>
127
138 <div id="editor">
149 function foo(items) {
150     var x = "All_this_is_syntax_highlighted";
161     return x;
172 }
183 </div>
194
205 <script src="/ace-builds/src-noconflict/ace.js" type="text/javascript" charset="utf-8"></script>
216 <script>
227     var editor = ace.edit("editor");
238     editor.setTheme("ace/theme/monokai");
249     editor.session.setMode("ace/mode/javascript");
250 </script>
261 </body>
272 </html>
```

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

BAB 3

ANALISIS

3.1 Analisis Sistem Kini

Seperti dibahas pada bab 2.1, 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.

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*

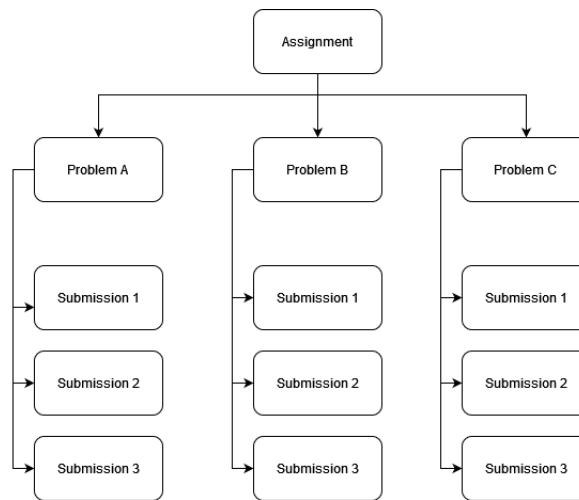
Merepresentasikan sebuah soal yang terdapat pada *assignment*. Untuk setiap *problem*, dapat ditentukan ekstensi *file* yang diizinkan, serta batas waktu dan memori kode. Sebuah *problem* dapat bersifat *upload only*, yang berarti *file* jawaban hanya diunggah, namun tidak dinilai dengan tes kasus.

- *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.



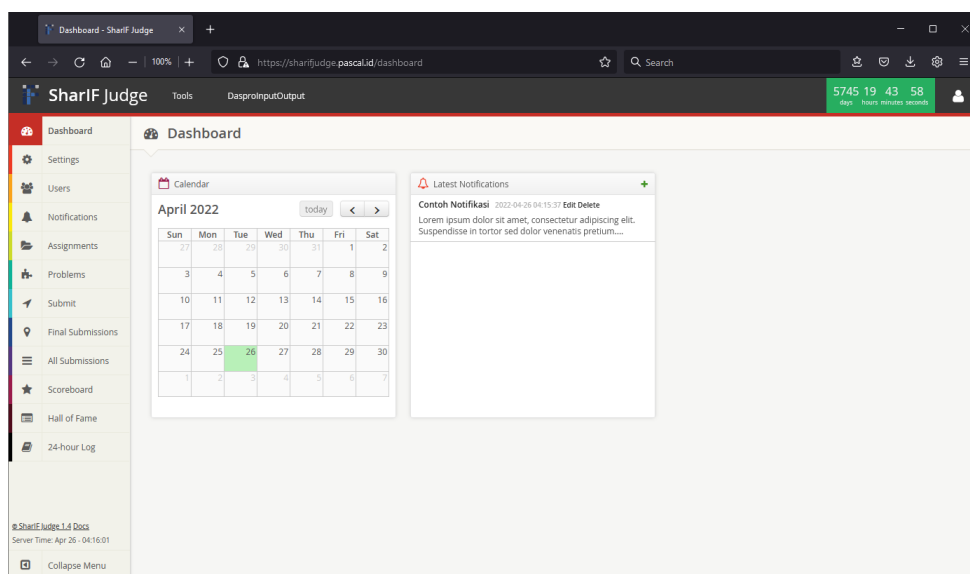
Gambar 3.1: Bagan hubungan *assignment*, *problem*, dan *submission* pada SharIF Judge

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

4 3.1.2 Halaman-halaman pada SharIF Judge

- 5 Halaman yang tersedia pada pengguna SharIF Judge bergantung pada *role* akun yang digunakan
- 6 pengguna tersebut. Berikut ini adalah setiap halaman dan fitur yang terdapat pada SharIF Judge
- 7 untuk *role admin*:

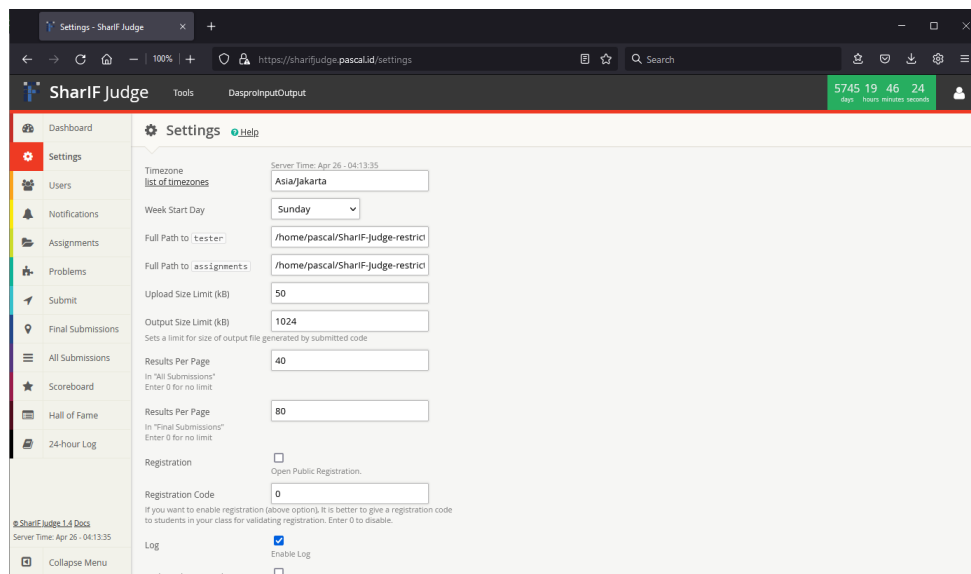
8 Dashboard



Gambar 3.2: Halaman Dashboard

- 9 Gambar 3.2 menunjukkan halaman Dashboard. Pada halaman ini terdapat kalender yang menun-
- 10 jukkan durasi setiap *assignment* dan daftar notifikasi.

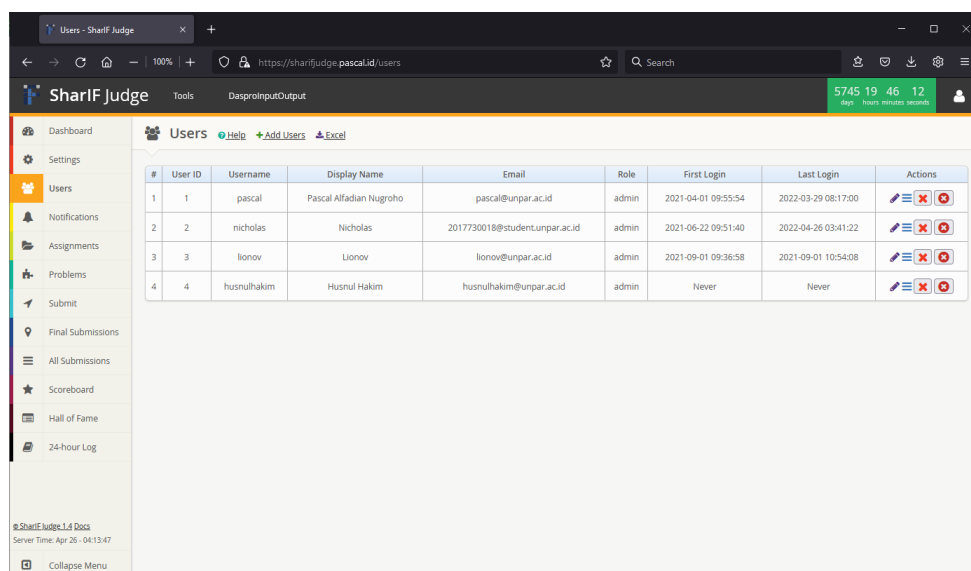
1 Settings



Gambar 3.3: Halaman Settings

Gambar 3.3 menunjukkan halaman Settings. Pada halaman ini terdapat berbagai pengaturan yang ada pada SharIF Judge.

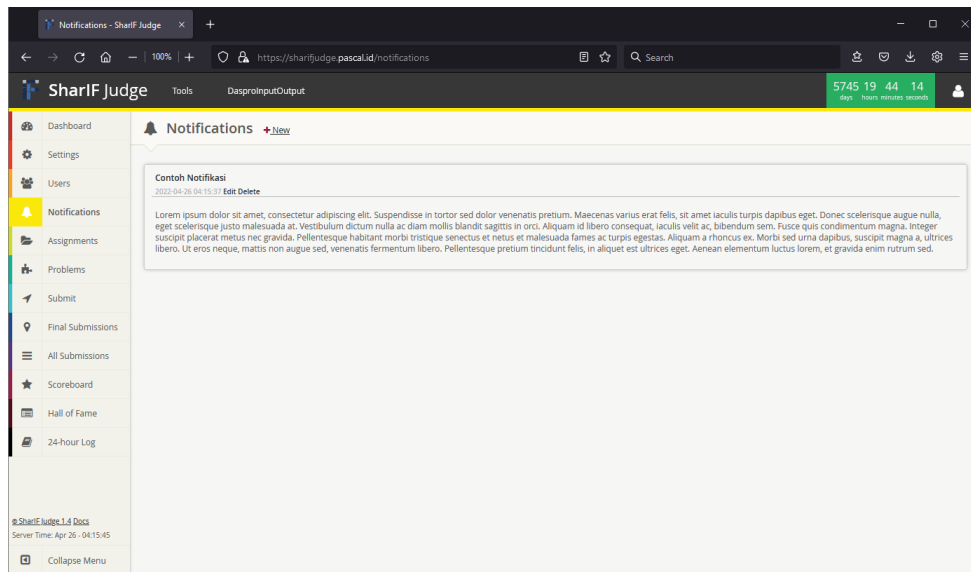
4 Users



Gambar 3.4: Halaman Users

Gambar 3.4 menunjukkan halaman Users. Pada halaman ini terdapat daftar seluruh pengguna yang terdaftar pada SharIF Judge. Pengguna juga dapat membuat, mengubah, dan menghapus pengguna.

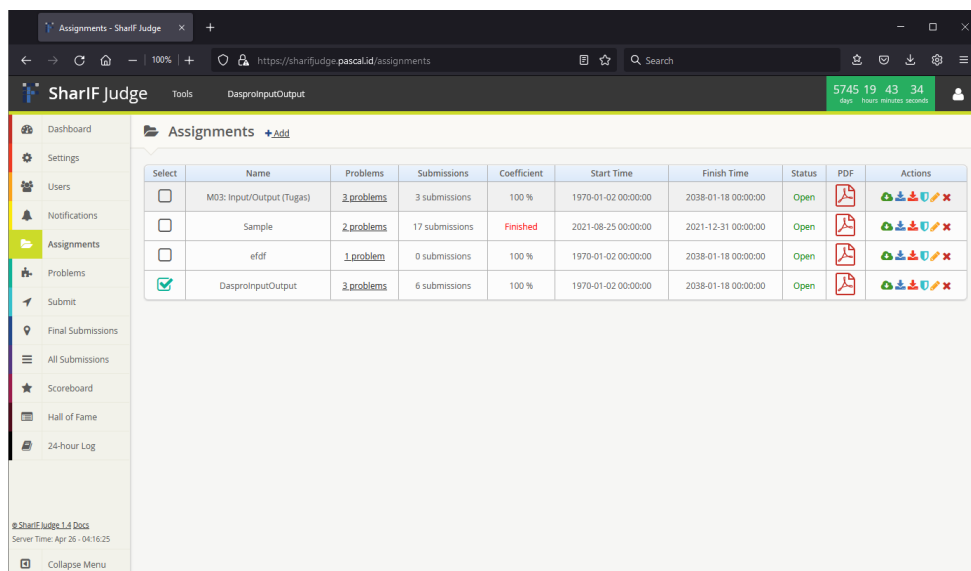
1 Notifications



Gambar 3.5: Halaman Notifications

- 2 Gambar 3.5 menunjukkan halaman Notifications. Pada halaman ini terdapat daftar seluruh notifikasi.
- 3 Pengguna juga dapat membuat, mengubah, dan menghapus notifikasi.

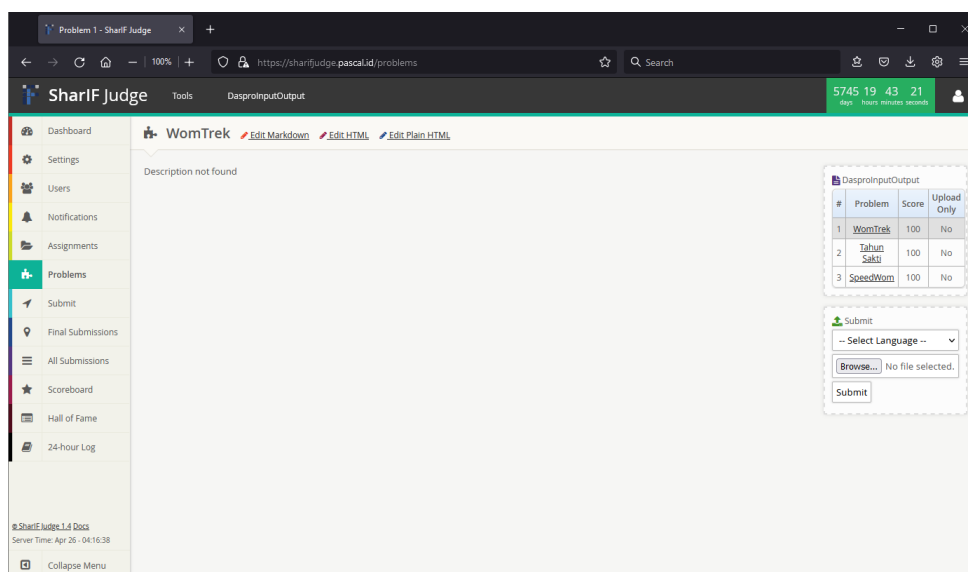
4 Assignments



Gambar 3.6: Halaman Assignments

- 5 Gambar 3.6 menunjukkan halaman Assignments. Pada halaman ini terdapat daftar seluruh
- 6 *assignment*. Pengguna juga dapat membuat, mengubah, dan menghapus *assignment*. Salah satu
- 7 *assignment* pada halaman ini harus dipilih untuk dapat menggunakan beberapa fitur lainnya pada
- 8 SharIF Judge. Soal dalam bentuk PDF juga dapat diunduh melalui halaman ini.

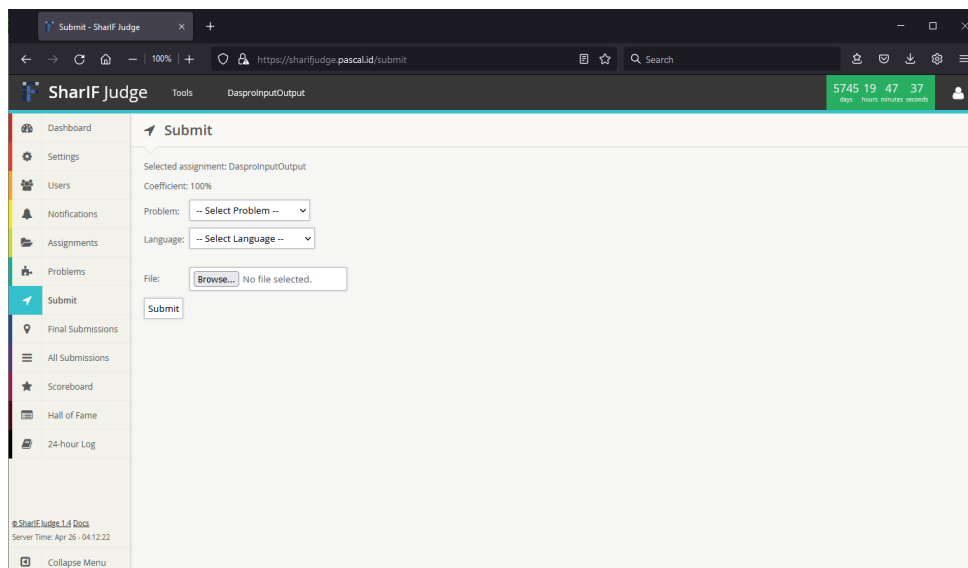
1 Problems



Gambar 3.7: Halaman Problems

- 2 Gambar 3.7 menunjukkan halaman Problems. Pada halaman ini terdapat detil dari setiap *problem*
3 dari *assignment* yang dipilih. Pengguna juga dapat mengunggah file untuk dikumpulkan sebagai
4 *submission* untuk *problem* yang dipilih.

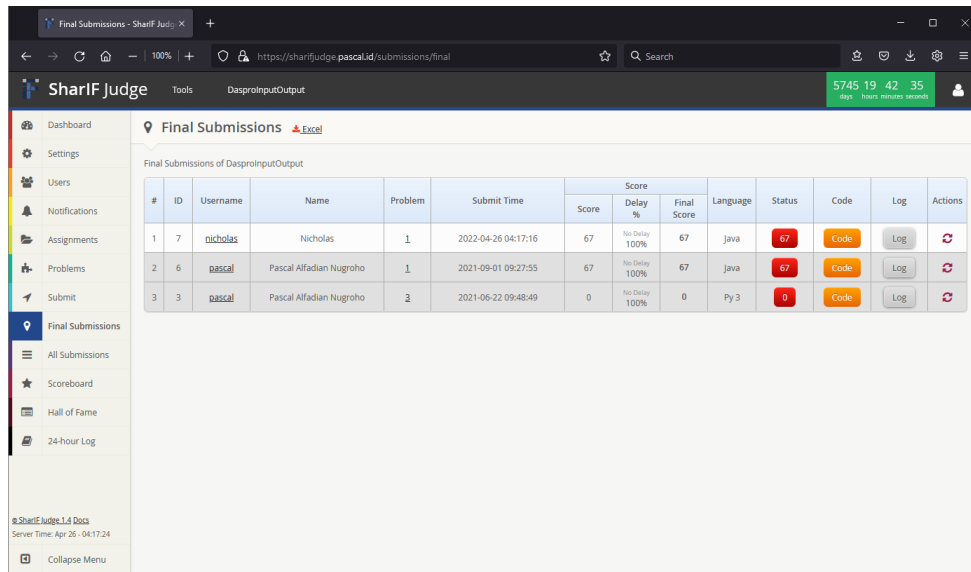
5 Submit



Gambar 3.8: Halaman Submit

- 6 Gambar 3.8 menunjukkan halaman Submit. Pada halaman ini, pengguna dapat mengunggah file
7 untuk dikumpulkan sebagai *submission* dari *problem* yang dipilih.

1 Final Submissions

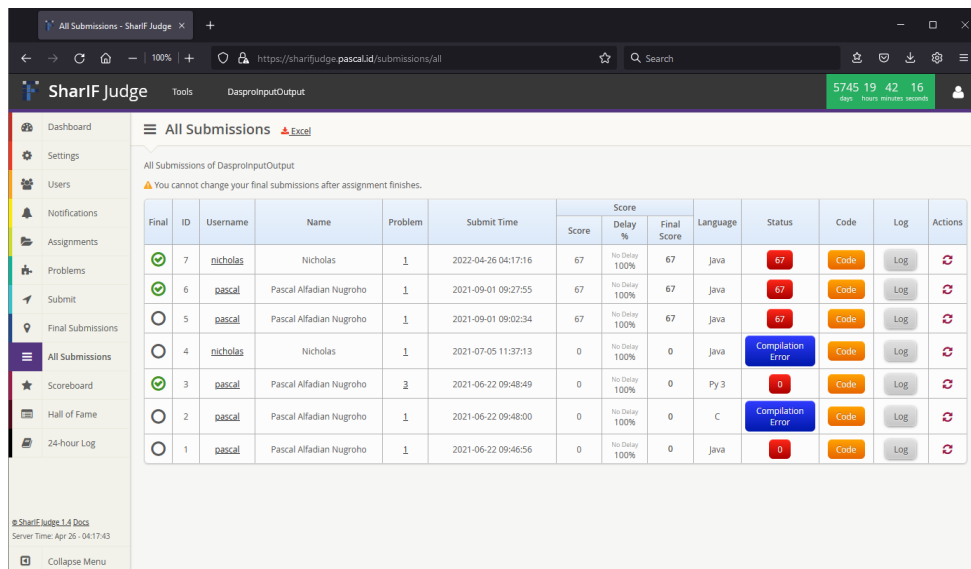


#	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
1	7	nicholas	Nicholas	1	2022-04-26 04:17:16	67	No Delay 100%	67	Java	67	Code	Log	
2	6	pascal	Pascal Alfadian Nugroho	1	2021-09-01 09:27:55	67	No Delay 100%	67	Java	67	Code	Log	
3	3	pascal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

Gambar 3.9: Halaman Final Submissions

- 2 Gambar 3.9 menunjukkan halaman Final Submissions. Pada halaman ini, terdapat daftar seluruh
 3 *final submission* untuk *assignment* yang dipilih. Pengguna juga dapat melihat *file* atau kode yang
 4 diunggah, dan nilai yang didapatkannya.

5 All Submissions

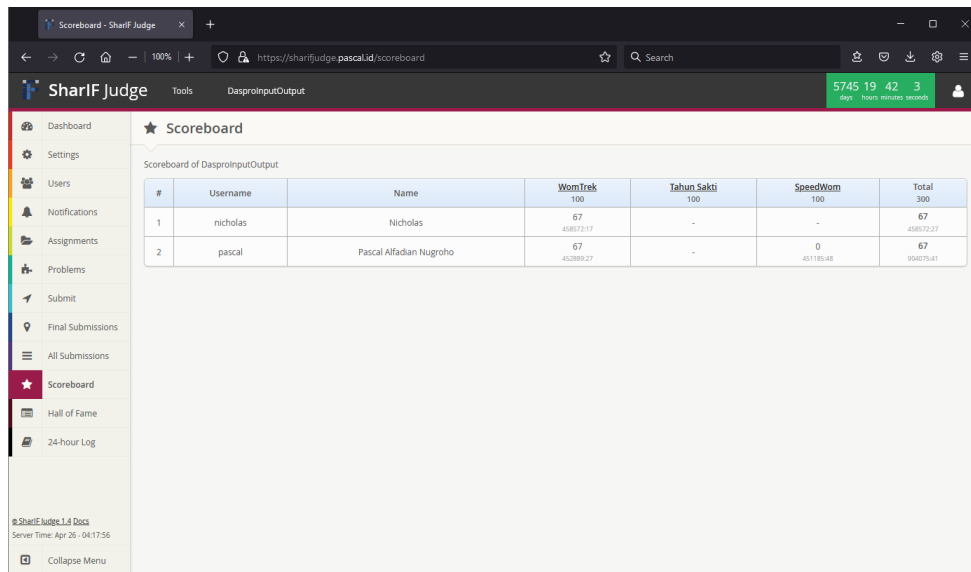


Final	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
✓	7	nicholas	Nicholas	1	2022-04-26 04:17:16	67	No Delay 100%	67	Java	67	Code	Log	
✓	6	pascal	Pascal Alfadian Nugroho	1	2021-09-01 09:27:55	67	No Delay 100%	67	Java	67	Code	Log	
○	5	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	

Gambar 3.10: Halaman All Submissions

- 6 Gambar 3.10 menunjukkan halaman All Submissions. Pada halaman ini, terdapat daftar seluruh
 7 *submission* untuk *assignment* yang dipilih. Pengguna juga dapat melihat *file* atau kode yang
 8 diunggah, dan nilai yang didapatkannya. Untuk setiap *problem*, sebuah *submission* dapat dipilih
 9 sebagai *final submission* melalui halaman ini.

1 Scoreboard



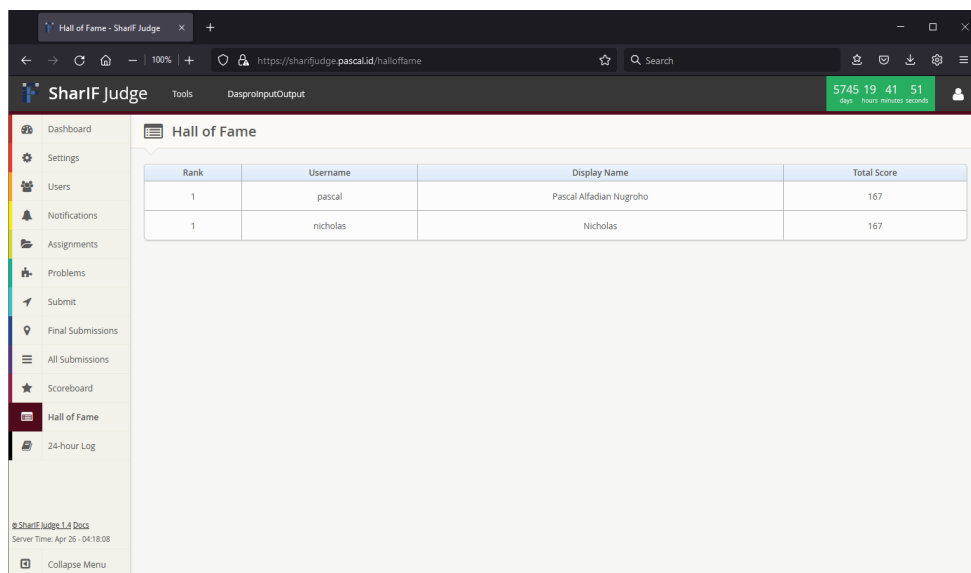
Scoreboard of DasproInputOutput

#	Username	Name	WomTrek 100	Tahun Sakti 100	SpeedWom 100	Total 300
1	nicholas	Nicholas	67 458672.17	-	-	67 458672.17
2	pascal	Pascal Alfadian Nugroho	67 452889.27	-	0 451185.48	67 904075.41

Gambar 3.11: Halaman Scoreboard

- 2 Gambar 3.11 menunjukkan halaman Scoreboard. Pada halaman ini, terdapat daftar nilai pengguna
3 untuk setiap *problem* pada *assignment* yang dipilih.

4 Hall of Fame



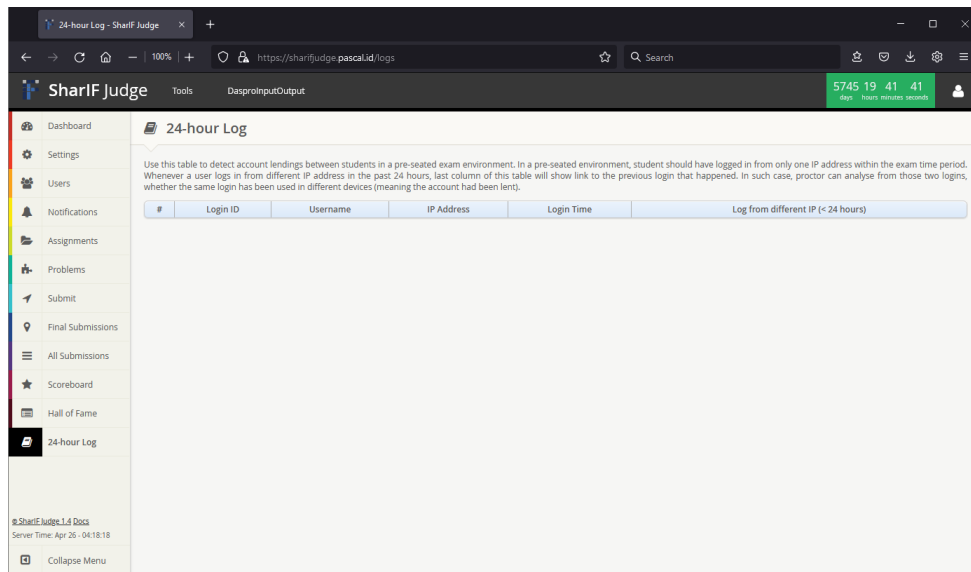
Hall of Fame

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

Gambar 3.12: Halaman Hall of Fame

- 5 Gambar 3.12 menunjukkan halaman Hall of Fame. Pada halaman ini, terdapat daftar pengguna
6 secara berurutan berdasarkan total nilai yang didapatkannya dari seluruh *assignment*.

1 24-Hour Log

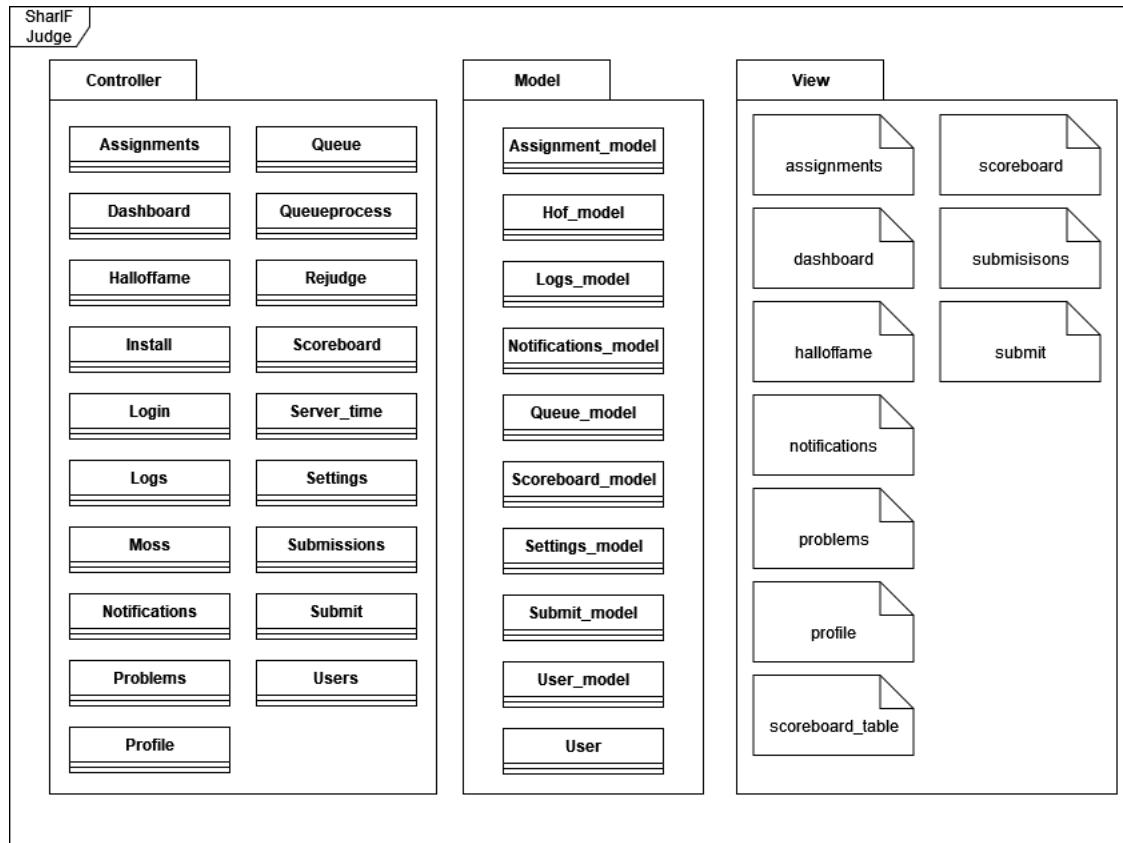


Gambar 3.13: Halaman 24-Hour Log

- 2 Gambar 3.13 menunjukkan halaman 24-Hour Log. Pada halaman ini, terdapat daftar yang mencatat
 3 bila akun yang sama melakukan *login* dengan *IP address* yang berbeda dalam jangka waktu 24 jam.
 4 Fitur ini digunakan untuk mendeteksi adanya peminjaman akun.

5 3.1.3 Model, View, Controller

- 6 SharIF Judge menggunakan *framework* CodeIgniter 3. Seperti yang dibahas pada bagian 2.2.1,
 7 *framework* CodeIgniter menerapkan pola arsitektur MVC, dengan komponen-komponen *model*, *view*,
 8 dan *controller*. Diagram kelas SharIF Judge dapat dilihat pada gambar 3.14. Digunakan simbol
 9 komentar untuk komponen *view* pada diagram ini, karena *view* tidak berbentuk kelas.



Gambar 3.14: Diagram kelas SharIF Judge

1 Model

Berikut ini adalah *model* pada SharIF Judge:

- **Assignment_model**

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

- `add_assignment($id, $edit)`
Menambah atau memperbarui sebuah *assignment*.
- `delete_assignment($assignment_id)`
Menghapus sebuah *assignment*.
- `all_assignments()`
Mengambil seluruh *assignment*.
- `new_assignment_id()`
Menentukan *integer* terkecil yang dapat digunakan sebagai id *assignment* baru.
- `all_problems($assignment_id)`
Mengambil seluruh *problem* dari *assignment*.
- `problem_info($assignment_id, $problem_id)`
Mengambil sebuah *problem*.
- `assignment_info($assignment_id)`
Mengambil sebuah *assignment*.
- `is_participant($participants, $username)`
Mengembalikan TRUE jika `$username` terdapat dalam `$participants`.

```

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

```

-
- 1 – `get_queue()`
 - 2 Mengambil seluruh antrean.
 - 3 – `empty_queue()`
 - 4 Mengosongkan antrean.
 - 5 – `add_to_queue($submit_info)`
 - 6 Menambahkan sebuah *submission* ke dalam antrean.
 - 7 – `rejudge($assignment_id, $problem_id)`
 - 8 Menambahkan seluruh *submission* dari sebuah *problem* ke dalam antrean untuk dinilai ulang.
 - 9 – `rejudge_single($submission)`
 - 10 Menambahkan sebuah *submission* ke dalam antrean untuk dinilai ulang.
 - 11 – `get_first_item()`
 - 12 Mengambil baris pertama dari antrean.
 - 13 – `remove_item($username, $assignment, $problem, $submit_id)`
 - 14 Menghapus sebuah baris dari antrean.
 - 15 – `save_judge_result_in_db ($submission, $type)`
 - 16 Menyimpan hasil penilaian ke dalam *database*.
 - 17 – `save_judge_result_in_db ($submission, $type)`
 - 18 Menyimpan hasil penilaian ke dalam *database*.
 - 19 • **Scoreboard_model**
 - 20 Model untuk menangani tabel `shj_scoreboard`. Fungsi yang dimiliki:
 - 21 – `_generate_scoreboard($assignment_id)`
 - 22 Membuat *scoreboard* untuk sebuah *assignment*.
 - 23 – `update_scoreboards()`
 - 24 Memperbarui *scoreboard* untuk seluruh *assignment*.
 - 25 – `update_scoreboard($assignment_id)`
 - 26 Memperbarui *scoreboard* untuk sebuah *assignment*.
 - 27 – `get_scoreboard($assignment_id)`
 - 28 Mengambil *scoreboard* untuk sebuah *assignment*.
 - 29 • **Settings_model**
 - 30 Model untuk menangani tabel `shj_settings`. Fungsi yang dimiliki:
 - 31 – `get_setting($key)`
 - 32 Mengambil sebuah pengaturan.
 - 33 – `set_setting($key, $value)`
 - 34 Memperbarui sebuah pengaturan.
 - 35 – `get_all_settings()`
 - 36 Mengambil seluruh pengaturan.
 - 37 – `set_settings($settings)`
 - 38 Memperbarui beberapa pengaturan.
 - 39 • **Submit_model**
 - 40 Model untuk menangani tabel `shj_submissions`. Fungsi yang dimiliki:
 - 41 – `get_submission($uname, $assignment, $problem, $submit_id)`
 - 42 Mengambil sebuah *submission*.
 - 43 – `get_final_submissions($a_id, $u_lv, $uname, $p_num, $f_user, $f_prblm)`

```

1      Mengambil seluruh final submission untuk sebuah assignment.
2      – get_all_submissions($a_id, $u_lv, $uname, $p_num, $f_user, $f_prblm)
3      Mengambil seluruh submission untuk sebuah assignment.
4      – count_final_submissions($a_id, $u_lv, $uname, $f_user, $f_prblm)
5      Menghitung jumlah final submission dari user tertentu.
6      – count_all_submissions($a_id, $u_lv, $uname, $f_user, $f_prblm)
7      Menghitung jumlah submission dari user tertentu.
8      – set_final_submission($uname, $assignment, $problem, $submit_id)
9      Memperbarui sebuah submission menjadi final.
10     – add_upload_only($submit_info)
11     Menambahkan hasil dari submission upload only ke dalam database.

```

- User

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

```

14     – select_assignment($assignment_id)
15     Menetapkan assignment yang dipilih.
16     – save_widget_positions($positions)
17     Memperbarui posisi widget.
18     – get_widget_positions()
19     Mengambil posisi widget.

```

- User_model

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

```

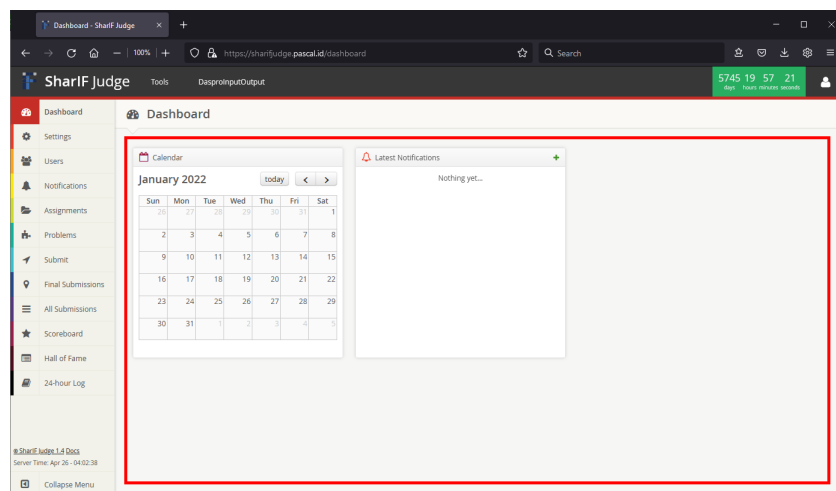
22     – have_user($username)
23     Mengembalikan TRUE jika terdapat user dengan nama $username.
24     – user_id_to_username($user_id)
25     Mengembalikan username dari user dengan id tertentu.
26     – username_to_user_id($username)
27     Mengembalikan id dari user dengan username tertentu.
28     – have_email($email, $username)
29     Mengembalikan TRUE jika terdapat user selain $username dengan email $email.
30     – add_user($username, $email, $display_name, $password, $role)
31     Menambahkan sebuah user baru.
32     – add_users($text, $send_mail, $delay)
33     Menambahkan beberapa user baru.
34     – delete_user($user_id)
35     Menghapus sebuah user.
36     – delete_submissions($user_id)
37     Menghapus seluruh submission dari sebuah user.
38     – validate_user($username, $password)
39     Mengembalikan TRUE jika $username dan $password valid untuk login.
40     – selected_assignment($username)
41     Mengembalikan assignment yang dipilih sebuah user.
42     – get_names()

```

- 1 Mengembalikan nama dari *user*.
- 2 – `update_profile($user_id)`
- 3 Memperbarui sebuah *user*.
- 4 – `send_password_reset_mail($email)`
- 5 Mengirim *email* untuk *reset password*.
- 6 – `passchange_is_valid($passchange_key)`
- 7 Mengembalikan TRUE jika kunci untuk *reset password* valid.
- 8 – `reset_password($passchange_key, $newpassword)`
- 9 Memperbarui *password* menjadi kunci *reset password*.
- 10 – `get_all_users()`
- 11 Mengambil seluruh *user*.
- 12 – `get_user($user_id)`
- 13 Mengambil sebuah *user*.
- 14 – `update_login_time($username)`
- 15 Memperbarui catatan *login* sebuah *user*.

16 View

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



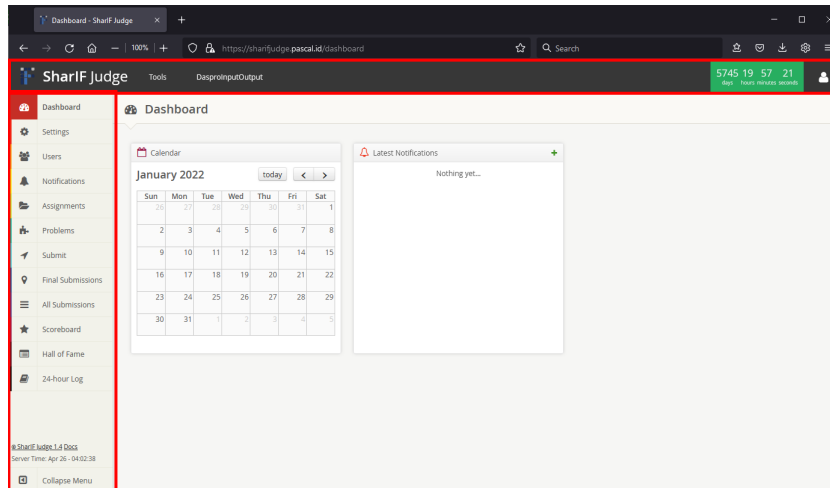
Gambar 3.15: Tampilan Dashboard pada `dashboard.twig`

- 18 • **pages**

19 Menyimpan komponen utama halaman. Pada gambar 3.15, area yang ditandai dengan kotak
 20 merah adalah salah satu contoh komponen utama halaman, yaitu Dashboard. Berikut ini
 21 adalah *view* yang terdapat pada folder ini:

- 22 – `assignments.twig`
- 23 – `dashboard.twig`
- 24 – `halloffame.twig`
- 25 – `notifications.twig`
- 26 – `problems.twig`
- 27 – `profile.twig`

- 1 – scoreboard_table.twig
- 2 – scoreboard.twig
- 3 – submissions.twig
- 4 – submit.twig

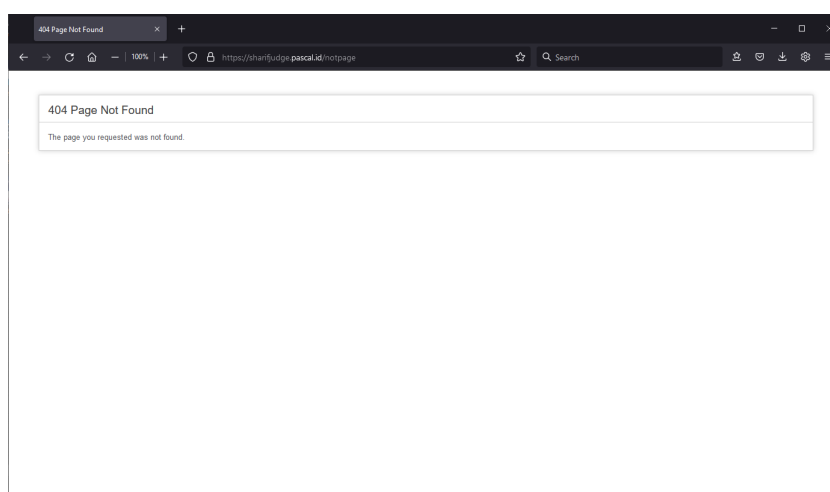


Gambar 3.16: Tampilan *top bar* pada `top_bar.twig` dan *side bar* pada `side_bar.twig`

5 • templates

6 Menyimpan komponen-komponen dasar halaman. Pada gambar 3.16, area yang ditandai
 7 dengan kotak merah adalah contoh komponen dasar halaman, yaitu *top bar* dan *side bar*.
 8 Berikut ini adalah *view* yang terdapat pada folder ini:

- 9 – base.twig
- 10 – side_bar.twig
- 11 – simple_header.twig
- 12 – top_bar.twig



Gambar 3.17: Tampilan halaman *error* 404 pada `error_404.php`

13 • errors

14 Menyimpan tampilan halaman *error*. Gambar 3.17 adalah salah satu contoh halaman *error*,

yaitu *error 404* yang ditampilkan ketika halaman yang diminta tidak ditemukan. Berikut ini adalah *view* yang terdapat pada folder ini:

- `error_404.php`
- `error_db.php`
- `error_general.php`
- `error_php.php`

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 memperbarui *assignment*.
- `edit($assignment_id)`
Memperbarui *assignment*.

- **Dashboard**

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

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

- **Halloffame**

Controller untuk menangani halaman *Hall of Fame* . Fungsi yang dimiliki:

- `hof_details()`
Mengambil data yang diperlukan untuk *hall of fame*.

- **Install**

Controller untuk menangani instalasi SharIF Judge.

- **Login**

Controller untuk menangani halaman-halaman *login*. Fungsi yang dimiliki:

- `register()`
Registrasi *user* baru dan menampilkan halaman *register*.
- `logout()`
Log out user saat ini dan mengalihkan ke halaman *login*.
- `lost()`

Menangani email dan menampilkan halaman untuk meminta *reset password*.

- `reset($passchange_key)`

Memproses dan menampilkan halaman untuk ubah *reset password*.

- **Logs**

Controller untuk menangani halaman *24-hour Log*.

- `index()` Mengambil data yang diperlukan dan menampilkan halaman *24-hour Log*.

- **Moss**

Controller untuk menangani halaman *Detect Similar Codes* . Fungsi yang dimiliki:

- `update($assignment_id)`

Memperbarui informasi pada halaman *Detect Similar Codes*.

- `_detect($assignment_id)`

Menjalankan Moss untuk mendeteksi kesamaan kode.

- **Notifications**

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

- `add()`

Menambahkan notifikasi baru dan menampilkan halaman *New Notification*.

- `edit($notif_id)`

Memperbarui sebuah notifikasi.

- `delete()`

Menghapus sebuah notifikasi.

- `check()`

Memeriksa adanya notifikasi baru.

- **Problems**

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

- `index($assignment_id, $problem_id = 1)`

Mengambil data yang diperlukan dan menampilkan halaman *Problems*.

- `edit($type = 'md', $assignment_id, $problem_id = 1)`

Memperbarui deskripsi *problem* dan menampilkan halaman *Edit Problem Description*.

- **Profile**

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

- `index($user_id)`

Mengambil data yang diperlukan dan menampilkan halaman *Profile*.

- `_password_check($str)`

Memeriksa apakah *password* sesuai dengan syarat.

- `_password_again_check($str)`

Memeriksa apakah *password again* sama dengan *password* yang dimasukkan.

- `_email_check($email)`

Memeriksa apakah terdapat user dengan alamat email tertentu.

- `_role_check($role)`

Memeriksa *role* yang dimiliki *user*.

- **Queue**

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

1 – `index()`

2 Mengambil data yang diperlukan dan menampilkan halaman *Queue*.

3 – `pause()`

4 Memberhentikan antrean.

5 – `resume()`

6 Melanjutkan antrean.

7 – `empty_queue()`

8 Mengosongkan antrean.

9 • **Queueprocess**

10 Controller untuk menangani proses penilaian kode. Fungsi yang dimiliki:

11 – `run()`

12 Menilai kode satu per satu dari antrean.

13 • **Rejudge**

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

15 – `index()`

16 Mengambil data yang diperlukan dan menampilkan halaman *Rejudge*.

17 – `rejudge_single()`

18 Melakukan penilaian ulang untuk sebuah *submission*.

19 • **Server_time**

20 Controller untuk menangani sinkronisasi waktu server. Fungsi yang dimiliki:

21 – `index()`

22 Mengembalikan waktu server.

23 • **Submissions**

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

25 – `_download_excel($view)`

26 Menggunakan *library* PHPExcel untuk membuat file excel.

27 – `final_excel()`

28 Mengunduh data *final submissions* sebagai file excel.

29 – `all_excel()`

30 Mengunduh data *final submissions* sebagai file excel.

31 – `the_final()`

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

33 – `all()`

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

35 – `select()`

36 Memilih *final submission*.

37 – `view_code()`

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

39 – `download_file()`

40 Mengunduh file excel.

41 • **Submit**

42 Controller untuk menangani *submissions*. Fungsi yang dimiliki:

```

1      - _language_to_type($language)
2      Mengembalikan kode singkatan dari bahasa pemrograman.
3      - _match($type, $extension)
4      Memeriksa apakah bahasa pemrograman dan tipe file sesuai.
5      - _check_language($str)
6      Memeriksa apakah bahasa pemrograman yang dipilih valid.
7      - index()
8      Mengambil data yang diperlukan dan menampilkan halaman Submit.
9      - _upload()
10     Menyimpan file yang diunggah dan menambahkannya ke dalam antrian.
11     • Users
12     Controller untuk menangani halaman Users. Fungsi yang dimiliki:
13     - index()
14     Mengambil data yang diperlukan dan menampilkan halaman Users.
15     - add()
16     Menambah user baru dan menampilkan halaman Add Users.
17     - delete()
18     Menghapus user.
19     - delete_submissions()
20     Menghapus seluruh submission dari sebuah user.
21     - list_excel()
22     Menggunakan library PHPExcel untuk membuat file excel dari list user.

```

23 3.1.4 Penyimpanan Kode

24 Pada SharIF Judge, seluruh kode yang diunggah pengguna disimpan pada *folder assignments*.
 25 Lokasi *folder assignments* dapat diatur pada halaman Settings. Kode yang diunggah disimpan
 26 pada *folder assignments* dengan format alamat:

```

27     assignment_<a_id>/p_<p_id>/<nama_pengguna>/<nama_file>-<s_id>.<ekstensi_file>

```

28 Penjelasan untuk format alamat adalah sebagai berikut:

- 29 • <a_id>
30 id dari *assignment* yang dipilih.
- 31 • <p_id>
32 id dari *problem* yang dipilih.
- 33 • <nama_pengguna>
34 Nama pengguna yang mengunggah *file*.
- 35 • <nama_file>
36 Nama *file* yang diunggah.
- 37 • <s_id>
38 id dari *submission* yang diunggah.
- 39 • <ekstensi_file>
40 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` yang terletak pada *folder tester*. Lokasi *folder tester* dapat diatur pada halaman Settings.
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

Pada SharIF Judge, sudah terdapat beberapa fitur yang diperlukan untuk berfungsi sebagai sebuah IDE. Dengan menggabungkan fitur yang sudah ada dengan beberapa fitur tambahan, SharIF Judge dapat berfungsi sebagai IDE.

3.2.1 Fitur-fitur

Agar SharIF Judge dapat berfungsi sebagai IDE, maka seluruh proses pengerjaan soal dari awal hingga akhir harus dapat dilakukan dalam SharIF Judge. Tahap pertama yang dilakukan di luar SharIF Judge adalah mengunduh dan melihat soal PDF, karena itu ditambahkan fitur untuk melihat PDF pada SharIF Judge. Kemudian, seluruh proses pembuatan kode juga harus dapat dilakukan dalam SharIF Judge, maka diperlukan fitur untuk mengedit kode, mengompilasi kode, dan menjalankan kode dengan tes kasus. Agar seluruh proses pengerjaan soal dapat terintegrasi, kode yang sudah dibuat juga harus dapat dikumpulkan sebagai *submission* untuk dinilai. Bagian-bagian berikut ini adalah fitur-fitur yang akan ditambahkan pada SharIF Judge:

Melihat soal

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.

SharIF Judge sudah memiliki fitur untuk menyimpan soal dalam bentuk PDF. Untuk mengambil *file* PDF tersebut, terdapat fungsi `pdf()` pada *controller Assignments*. Karena fitur ini menampilkan *file* PDF yang sama, fungsi tersebut akan digunakan kembali. Namun, karena fungsi `pdf()` menggunakan fungsi `force_download()`, perlu dilakukan perubahan agar tidak muncul dialog unduh pada *browser*. *File* PDF yang didapat kemudian ditampilkan melalui *viewer* milik `PDF.js` yang di *embed* pada halaman Submit.

Mengedit kode

Ditambahkan editor kode Ace yang memiliki kemampuan untuk membantu pembuatan, seperti *syntax highlighting* dan indentasi, sehingga pengguna dapat mengedit kode secara langsung pada SharIF Judge.

Pada halaman submit, terdapat menu *dropdown* yang digunakan untuk memilih bahasa program. Karena seluruh pilihan bahasa sudah terdapat pada *dropdown* tersebut, pada fitur ini dimanfaatkan kembali untuk memilih bahasa *syntax highlighting* yang digunakan pada editor kode Ace.

Menyimpan dan Memuat kode

Ditambahkan fitur untuk menyimpan dan memuat kembali kode yang sudah dibuat pada editor kode, sehingga pengguna dapat menyimpan kode yang sudah dibuat secara daring dan memuatnya kembali ketika *problem* yang sama dipilih.

Seperti dijelaskan pada bagian 3.1.4, SharIF Judge memiliki struktur penyimpanan file yang dipisahkan berdasarkan *assignment*, *problem*, *submission*, dan pengguna. Alamat folder tersebut sudah sesuai dengan kebutuhan fitur ini, sehingga juga digunakan untuk menyimpan kode. Untuk mempermudah penyimpanan dan pemuatan, kode dari editor disimpan dalam bentuk *file* `txt` pada alamat yang sesuai.

Controller yang berfungsi untuk menangani penyimpanan *file* yang diunggah adalah *controller Submit*. Fungsi baru `save()` dan `load()` yang berfungsi untuk menyimpan dan memuat kode ditambahkan pada *controller Submit*.

Menjalankan kode dengan tes kasus

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.

Controller yang berfungsi untuk menangani penyimpanan *file* yang diunggah adalah *controller Submit*. Fungsi baru `_execute()` ditambahkan pada *controller Submit* yang dijalankan melalui fungsi `save("execute")` untuk menyimpan kode terlebih dahulu, kemudian menambahkan kode ke antrean.

Pada sistem antrean 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 antrean, kode perlu dikumpulkan sebagai *submission* terlebih dahulu.

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

1. Kode yang sudah disimpan dengan ekstensi yang tepat dengan alamat folder yang sesuai, seperti dibahas pada bagian 3.1.4.
2. *Input* tes kasus disimpan sebagai *file* txt dengan alamat folder yang sama.
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 dengan alamat folder yang sama.
6. *File* txt *output* dimuat dan ditampilkan pada halaman submit.

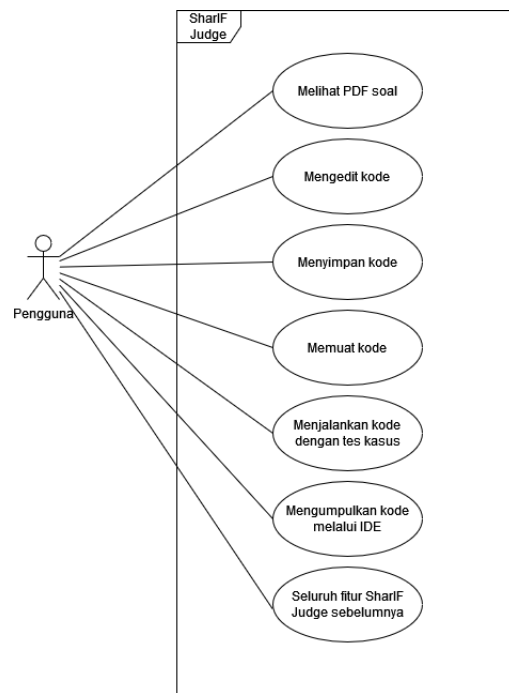
Mengumpulkan kode melalui IDE

Ditambahkan fitur untuk mengumpulkan kode yang sudah dibuat pada editor, sehingga pengguna dapat mengumpulkan kode yang sudah dibuat sebagai *submission* tanpa perlu mengunggah *file*.

Pada fitur ini, kode dari editor akan melalui proses yang sama seperti yang dilalui oleh *file* kode yang diunggah, seperti dijelaskan pada bagian 3.1.5. *Controller* yang berfungsi untuk menangani penyimpanan *file* yang diunggah adalah *controller* Submit. Fungsi baru `_submit()` ditambahkan pada *controller* Submit yang dijalankan melalui fungsi `save("submit")` untuk menyimpan kode terlebih dahulu, kemudian menambahkan kode ke antrean. Kode kemudian akan melalui proses hingga penilaian selesai.

3.2.2 Use Case

Seluruh fitur yang ditambahkan dapat digunakan oleh setiap *role* pengguna, yaitu *Admin*, *Head Instructor*, *Instructor*, *Student*. Pada gambar 3.18, terdapat Use Case Diagram sebagai tambahan dari diagram pada gambar 2.1 untuk fitur yang akan ditambahkan.



Gambar 3.18: Use Case Diagram Fitur Usulan

Berikut ini merupakan skenario untuk setiap fitur yang ditambahkan pada SharIF Judge:

- Melihat soal

- Aktor: Pengguna
- Kondisi Awal: PDF soal 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

- 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

- 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
 - 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
 - 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
 - Pengecualian: Pengguna belum mengedit kode pada editor kode
- Mengumpulkan kode melalui IDE
 - Aktor: Pengguna
 - Kondisi Awal: Pengguna sudah mengedit kode pada editor kode
 - Kondisi Akhir: Sistem menampilkan halaman All Submissions
 - Skenario Normal:
 1. Pengguna mengedit kode pada editor kode
 2. Pengguna menekan menu Submit
 3. Sistem menjalankan dan menilai kode
 4. Sistem menampilkan hasil penilaian pada halaman All Submissions
 - 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.

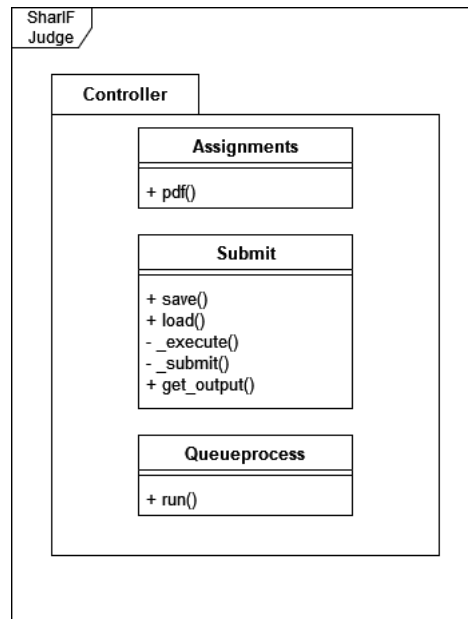
4.1 Rancangan Antarmuka

The image shows a web application interface for SharIF Judge. It features a sidebar on the left with a list of problem categories. The main content area is divided into three sections: 'Submit', 'PDF Viewer', and 'Code Editor'. The 'Submit' section contains dropdown menus for 'Problem' and 'Language', a 'File' input with a 'Browse' button, and a 'Submit' button. The 'PDF Viewer' section displays a sample PDF document. The 'Code Editor' section contains a large text area for code, with 'Save', 'Execute', and 'Submit' buttons at the bottom. To the right of the code editor are two smaller sections: 'Input' and 'Output', each containing a text area for input and output data respectively.

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* melalui AJAX.

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* melalui AJAX.

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 variabel `EXEC_ONLY` sebagai kondisi untuk menjalankan kode tanpa penilaian.
 - Penambahan fungsi `shj_log_exec` untuk mencatat hasil *output* dan pesan *error*.
- *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.

- 1 • *Controller Queueprocess*:
 - 2 – Fungsi `run()`:
 - 3 Penambahan kondisi untuk menjalankan `tester.sh` tanpa penilaian.
- 4 • *View submit*:
 - 5 – Penambahan elemen `textarea` untuk *input*.
 - 6 – Penambahan elemen `textarea` untuk *output*.
 - 7 – Penambahan elemen `button` untuk menjalankan kode.
 - 8 – Penambahan *script* untuk memanggil fungsi `save("execute")` pada *controller Submit*.
 - 9 – Penambahan *script* untuk memanggil fungsi `get_output` pada *controller Submit*.

10 4.2.6 Mengumpulkan Kode Melalui IDE

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

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

- 14 • *Controller Submit*:
 - 15 – Fungsi `_submit()`:
 - 16 Fungsi baru untuk mengumpulkan kode. Fungsi ini dipanggil oleh fungsi `save("submit")`.
- 17 • *View submit*:
 - 18 – Penambahan elemen `button` untuk mengumpulkan kode.
 - 19 – 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

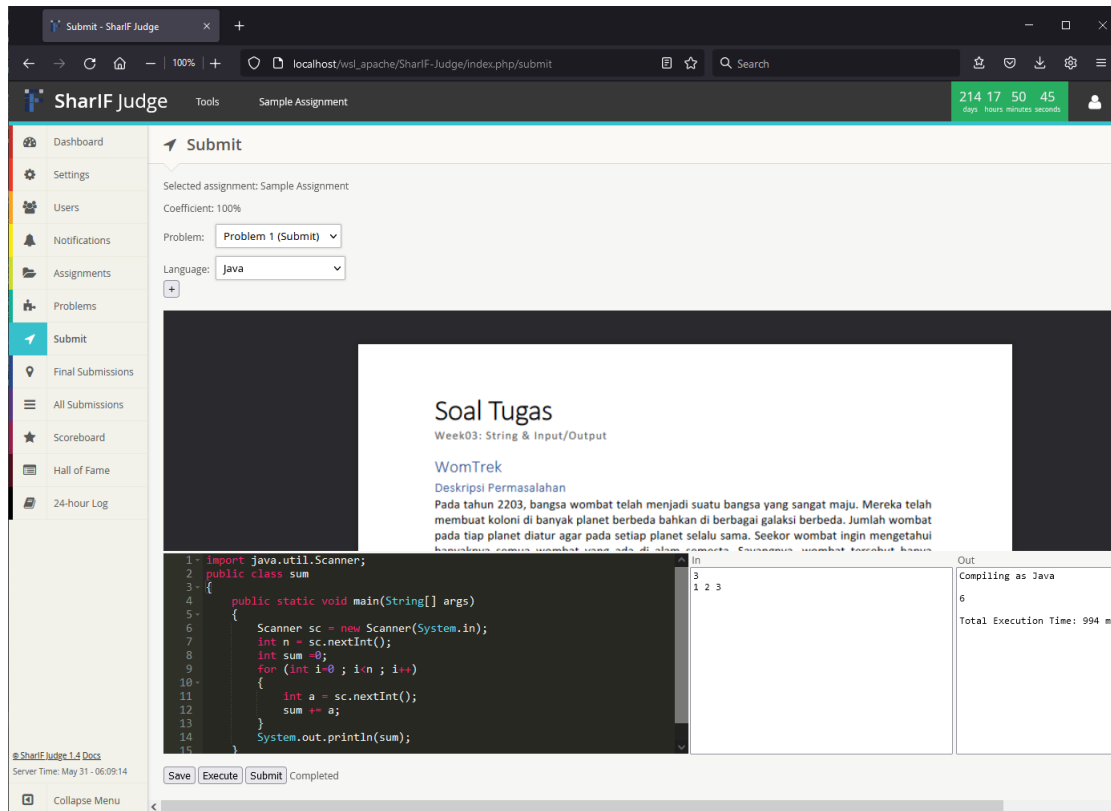
- * *Storage*: 1TB

- 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 C.1. Seluruh *style* dan *script* untuk halaman ini terletak di *file* terpisah. *Stylesheet* yang terdapat di `assets\styles\submit.css` terdapat pada lampiran C.3. *Script* yang terdapat di `assets\js\shj_submit.js` terdapat pada lampiran C.2.

5.2.2 Menampilkan Soal

Soal PDF ditampilkan pada `iframe` dengan id `pdf_viewer` yang menampilkan `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 *query string* `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 `echo` 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

1 `$no_download` bernilai `TRUE`, maka isi PDF akan dikembalikan melalui fungsi `echo` dengan *header*
2 `Content-Type: application/pdf`. Kode untuk perubahan ini terdapat pada lampiran [A.1](#)

3 5.2.3 Editor Kode

4 Ace menggunakan sebuah elemen `div` sebagai tempat untuk menampilkan editor kodenya. Editor
5 Ace dimuat dan dikonfigurasi melalui JavaScript yang terdapat di `shj_submit.js`. Ditambahkan
6 juga beberapa fungsi untuk mengubah konfigurasi *syntax highlighting* sesuai dengan bahasa yang
7 dipilih, dan untuk mengaktifkan atau menonaktifkan editor kode jika *problem* dan bahasa belum
8 dipilih. Kode JavaScript untuk konfigurasi editor Ace terdapat pada lampiran [C.2](#).

9 5.2.4 Menyimpan dan Memuat Kode

10 Kode akan disimpan pada *file* txt bernama `editor.txt` dengan alamat folder yang dibahas pada
11 bagian [3.1.4](#). Nama dan tipe file ini disimpan sebagai konstanta pada `constants.php`. Perubahan
12 kode `constants.php` terdapat pada lampiran [D.1](#).

13 Untuk menyimpan kode, fungsi `save` ditambahkan pada *controller* `Submit`. Fungsi ini mengambil
14 isi dari editor kode melalui POST lalu menyimpannya pada `editor.txt`. Kode untuk fungsi `save`
15 terdapat pada lampiran [A.2](#) baris 72–135.

16 Untuk memuat kode, fungsi `load` ditambahkan pada *controller* `Submit`. Bila tersedia, fungsi ini
17 mengambil isi dari `editor.txt` lalu mengembalikan isinya. Kode untuk fungsi `load` terdapat
18 pada lampiran [A.2](#) baris 39–64.

19 Fungsi `save` dan `load` dipanggil melalui *AJAX request* pada *view* `Submit`. Fungsi `save` dipanggil
20 ketika tombol `Save` ditekan, sementara fungsi `load` dipanggil ketika pengguna memilih *problem*
21 pada *dropdown*. Kode untuk mengirimkan *AJAX request* tersebut terdapat pada lampiran [C.2](#) baris
22 77–99 dan baris 23–47.

23 5.2.5 Menjalankan Kode dengan Tes Kasus

24 Pada sistem antrean kode yang dibahas pada bagian [3.1.5](#), tabel `shj_queue` tidak menyimpan
25 alamat dan ekstensi *file*, namun tabel ini menyimpan `submit_id` sebagai referensi untuk tabel
26 `shj_submissions`, dimana alamat dan ekstensi *file* tersimpan. Selain itu, `submit_id` juga digunakan
27 untuk menyimpan nilai yang dihasilkan oleh `tester.sh`. Karena itu, agar kode dapat dimasukkan
28 pada antrean, kode perlu dikumpulkan sebagai *submission* terlebih dahulu.

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

- 31 1. Kode yang sudah disimpan sebagai *file* txt disimpan kembali dengan ekstensi yang tepat
32 dengan alamat folder yang dibahas pada bagian [3.1.4](#).
- 33 2. *Input* tes kasus disimpan sebagai *file* txt dengan alamat folder yang sama.
- 34 3. Membuat baris baru pada tabel `shj_submission` yang bersifat sementara untuk menyimpan
35 alamat dan ekstensi *file* kode. Dikarenakan `submit_id` untuk setiap *submission* selalu dimulai
36 dari 1, dapat digunakan `submit_id = 0`.
- 37 4. Membuat baris baru pada tabel `shj_queue` dengan `submit_id = 0` dan `type = "exec"` untuk
38 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 dengan alamat folder yang sama.

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 D.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.2 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 B.1.

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 B.2.

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 tersimpan pada `exec_in.txt`, kemudian hasilnya akan disimpan pada `exec_out.txt`. Perubahan ini terdapat pada lampiran D.2 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.2 baris 240–267. Kode untuk mengirimkan *AJAX request* terdapat pada lampiran C.2 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 D.2 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.
 - `+`
 - `[\w]+`

Mengambil hanya 1 token yang sesuai.
 - `(?=\ is public)`

Mencari kalimat "is public" tanpa memasukkannya 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*.

```

1      - .*
2      Mencari semua karakter sebanyak 0 atau lebih kali.
3      - class
4      Mencari kata "class".
5      - +
6      Mencari karakter spasi sebanyak 1 atau lebih kali.
7  5. sed 's/ .*$/ /'
8      Mencari karakter spasi dan seluruh karakter setelahnya, lalu menggantikannya dengan string
9      kosong.
10     • ' .*$'
11     Mencari karakter spasi dan seluruh karakter setelahnya hingga akhir dari string.
12     - ' '
13     Mencari karakter spasi.
14     - .*
15     Mencari semua karakter sebanyak 0 atau lebih kali.
16     - $
17     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 antrean melalui *model* `Queue_model` untuk dinilai. Perubahan kode ini terdapat pada lampiran [A.2](#) 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 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. Perangkat lunak diuji pada *judge* latihan dengan alamat <http://daspro.labftis.net>. Perangkat lunak digunakan oleh seluruh mahasiswa Dasar-dasar Pemrograman selama 1 semester di luar kegiatan kuis. Selama durasi kelas, mahasiswa dan dosen melaporkan persoalan yang ditemukan dan memberi saran pada perangkat lunak. Seluruh persoalan dan masukan yang diterima dicatat pada GitHub penulis, dengan alamat <https://github.com/athlonneo/SharIF-Judge>. Menjelang akhir semester, dibagikan survei pada mahasiswa Dasar-dasar Pemrograman untuk menilai perangkat lunak sekaligus mendapatkan masukan yang lebih banyak.

5.4.1 Perubahan melalui GitHub

Bagian-bagian berikut ini merupakan persoalan dan masukan yang diterima dan dicatat pada GitHub penulis sebagai *issues*.

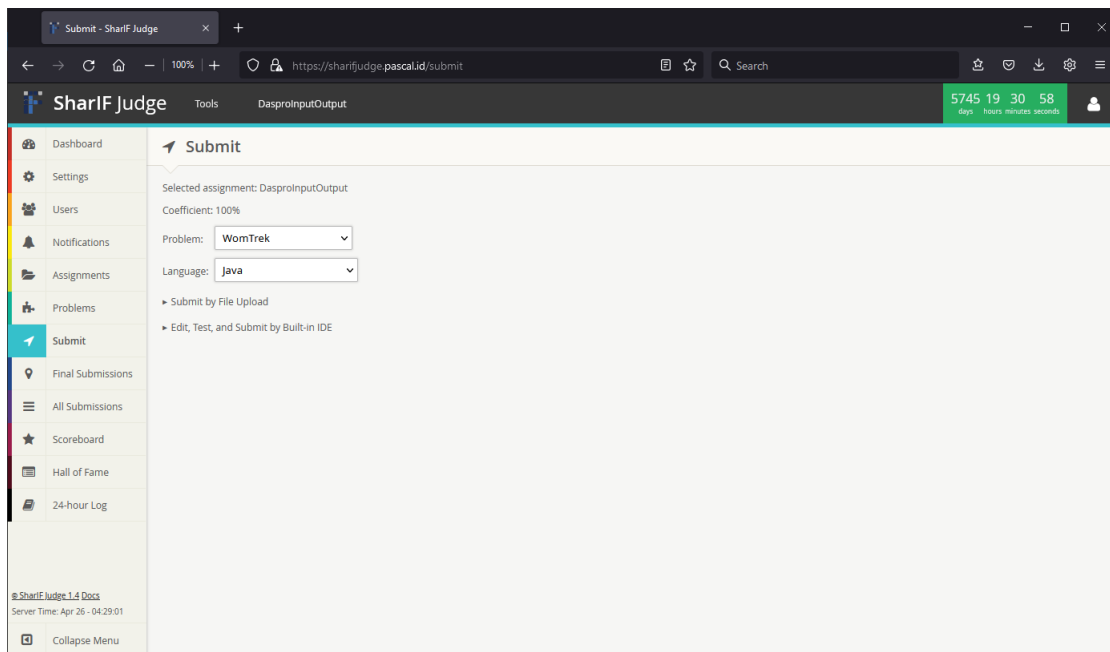
Perubahan Tampilan Antarmuka

Tercatat pada *issue* #2¹, 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:

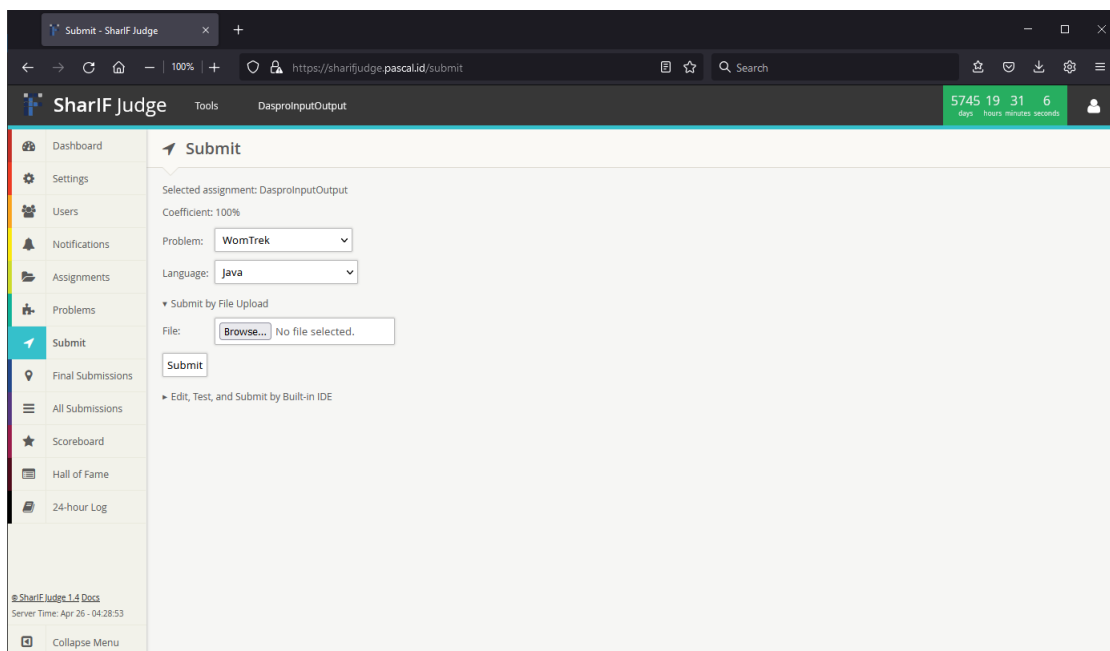
- Memberi jarak antara PDF Viewer dengan *scrollbar*.
- Meningkatkan ukuran editor kode.
- Mengubah teks tombol "Execute" menjadi "Save & Execute".
- Mengubah teks tombol "Submit" menjadi "Save & Submit".

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

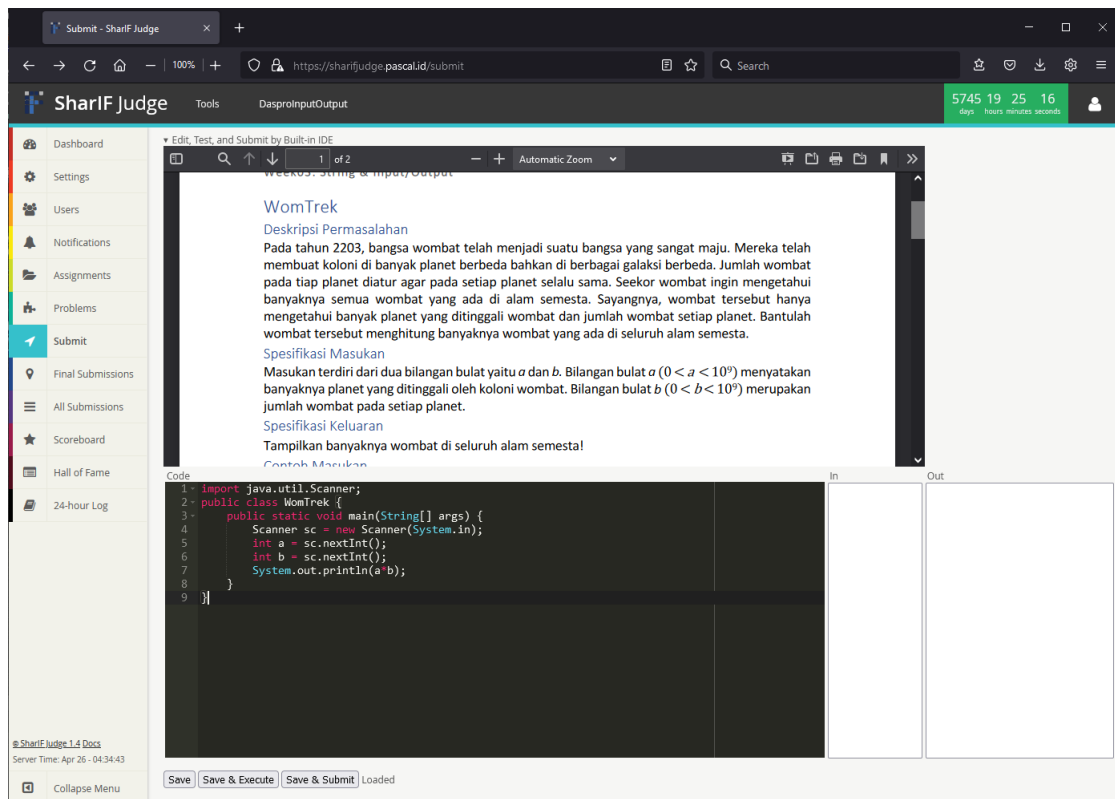
- 1 • Memisahkan antarmuka unggah *file* dengan IDE.



Gambar 5.2: Tampilan antarmuka setelah perubahan



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.

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

5 Tercatat pada *issue* #3², ditemukan masalah jika mengumpulkan jawaban untuk soal *upload only*,
 6 *status* pada halaman Submissions menjadi "File Not Found".

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

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

Kode 5.1: Perubahan pada Submit.php

```

13 @@ -378,8 +378,15 @@ class Submit extends CI_Controller
14 1
15 2     $file_name = $file_name.'-'.($this->user->selected_assignment['total_submits']+1);
16 3     $file_path = $user_dir.'/'.$file_name.'.'.$file_ext;
17 4
18 5 +     foreach($this->problems as $item)
19 6 +         if ($item['id'] == $problem_id)
20 7 +         {
21 8 +             $this->problem = $item;
22 9 +             break;
23 0 +         }
24 1 +
  
```

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

```

112         if (!write_file($file_path, $data)){
123             $response = json_encode(array(status=>FALSE, message=>'Unable to submit'));
124         }

```

5 Tidak Dapat *Submit* Melalui Unggah File

Tercatat pada *issue* #7³, salah satu mahasiswa mencoba melakukan *submit* melalui unggah *file*, namun tidak tercatat pada halaman *Submissions*. Ketika mahasiswa tersebut mencoba mengunggah kode yang sama melalui IDE, *submission* berhasil tercatat.

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

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

Karena masalah ini berasal dari Sharif Judge langsung dan tidak berkaitan dengan skripsi ini, *pull request* ditujukan langsung ke repositori <https://github.com/ifunpar/SharIF-Judge> sebagai *pull request* #15⁴.

Kode 5.2: Perubahan pada *Install.php*

```

21 @@ -78,8 +78,8 @@ public function index()
22     'status'      => array('type' => 'VARCHAR', 'constraint' => 100),
23     'pre_score'   => array('type' => 'INT', 'constraint' => 11),
24     'coefficient' => array('type' => 'VARCHAR', 'constraint' => 6),
25     'file_name'   => array('type' => 'VARCHAR', 'constraint' => 30),
26     'main_file_name' => array('type' => 'VARCHAR', 'constraint' => 30),
27     'file_name'   => array('type' => 'VARCHAR', 'constraint' => 100),
28     'main_file_name' => array('type' => 'VARCHAR', 'constraint' => 100),
29     'file_type'   => array('type' => 'VARCHAR', 'constraint' => 6),
30 );
31 $this->dbforge->add_field($fields);
32

```

34 *Submit* Jawaban txt Melalui IDE

Tercatat pada *issue* #10⁵, 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*

```

42 @@ -68,6 +68,12 @@ $(document).ready(function(){
43     editor.session.setMode("ace/mode/c-cpp");
44

```

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

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

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

```

13         disableEditor(false);
14     }
15 +     else if(this.value.toLowerCase().includes("txt")){
16 +         editor.session.setMode("ace/mode/plain-text");
17 +         disableEditor(false);
18 +         $("#editor_execute").prop("disabled", true);
19 +         $("#editor_input").prop("disabled", true);
20 +     }
21     else{
22         editor.session.setMode("ace/mode/plain-text");
23         disableEditor(true);

```

13 Tampilan PDF Viewer jika PDF Soal Tidak Tersedia

14 Tercatat pada *issue* #11⁶, sebelumnya PDF.js akan menampilkan pesan *error* jika file PDF tidak
15 ditemukan. Karena pesan ini kurang deskriptif bagi pengguna, disarankan untuk menyembunyikan
16 PDF *viewer* jika file PDF soal tidak tersedia.

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

Kode 5.4: Perubahan pada `Assignments.php`

```

22 @ -588,7 +583,43 @@ class Assignments extends CI_Controller
23 // redirect to add function
24 $this->add();
25 }
26 +
27 +
28 +
29 +
30 + // -----
31 +
32 +
33 +
34 + /**
35 +  * Check PDF File Availability
36 +  */
37 + public function pdfCheck($assignment_id, $problem_id = NULL)
38 + {
39 +     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
40 +     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
41 +     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
42 +
43 +     // Find pdf file
44 +     if ($problem_id === NULL || $problem_id === "null")
45 +     {
46 +         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/')."/assignment-{$assignment_id}/*.pdf";
47 +     }
48 +     else
49 +     {
50 +         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/')."/assignment-{$assignment_id}/p{
51 +             $problem_id}/*.pdf";
52 +         $pdf_files = glob($pattern);
53 +
54 +         if ( ! $pdf_files )
55 +             $response = json_encode(array(status=>FALSE));
56 +         elseif (!$this->assignment_model->assignment_info($assignment_id)['open'])
57 +             $response = json_encode(array(status=>FALSE));
58 +         elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)['participants
59 +             '], $this->user->username) )
60 +             $response = json_encode(array(status=>FALSE));
61 +         elseif ( shj_now() > $finishtime + $extratime )
62 +             $response = json_encode(array(status=>FALSE));
63 +         elseif ( shj_now() < $starttime )
64 +             $response = json_encode(array(status=>FALSE));
65 +         else
66 +             $response = json_encode(array(status=>TRUE));
67 +
68 +         echo $response;
69 +     }
70 + }

```

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

```

113 }
114 }

```

Kode 5.5: Perubahan pada `submit.twig`

```

4  @ -15,7 +15,21 @@
5  <link rel="stylesheet" type='text/css' href="{{ base_url('assets/styles/submit.css') }}" />
6  <script>
7    shj.p={};
8    - {{ problems_js|raw }}
9    + {{ problems_js|raw }};
10   + $.ajax({
11     +   url: "{{ site_url('assignments/pdfCheck/' ~ user.selected_assignment.id) }}",
12     +   cache: false,
13     +   success: function(data){
14       +     data = JSON.parse(data);
15       +     if(data.status){
16         +       $("#pdf_viewer").attr('src', "{{ base_url('assets/pdfjs/web/viewer.html?file=' ~ site_url('assignments/pdf/'
17         ~ user.selected_assignment.id ~ '/null/true') }}"");
18         +       $("#pdf_viewer").show();
19       +     }
20     +   },
21     +   error: function (error){
22       +     console.error(error);
23     +   },
24   + });
25   </script>
26   <script src="{{ base_url('assets/ace/ace.js') }}"></script>
27   <script type='text/javascript' src="{{ base_url('assets/js/shj-submit.js') }}"></script>

```

5.4.2 Survei

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

1. Menampilkan soal
2. Editor kode
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:

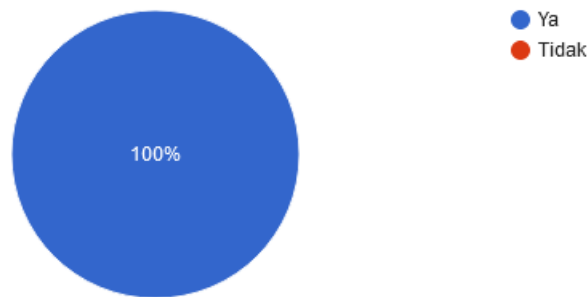
1. Apakah Anda sudah mencoba fitur ini? (Partisipan menjawab dengan ya atau tidak)
2. Apakah terdapat kendala saat menggunakan fitur ini? (Partisipan menjawab dengan ya atau tidak)
3. Jika ya, apa kendala yang dialami? (Partisipan menjawab dengan kalimat sendiri)
4. Apakah fitur ini sudah cukup nyaman 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)

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



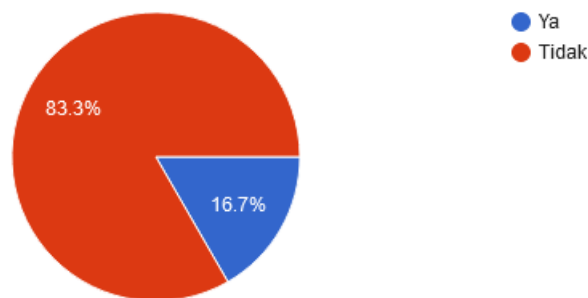
Gambar 5.5: Hasil survei bagian 1 pertanyaan 1

(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.

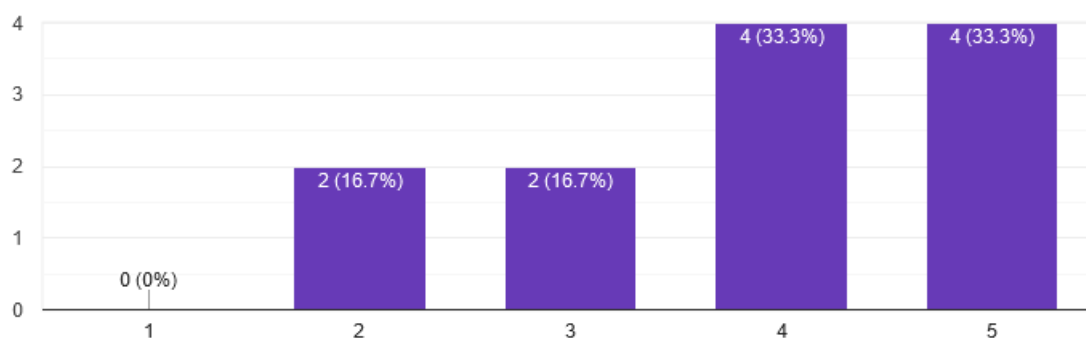
(c) Jika ya, apa kendala yang dialami?

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?

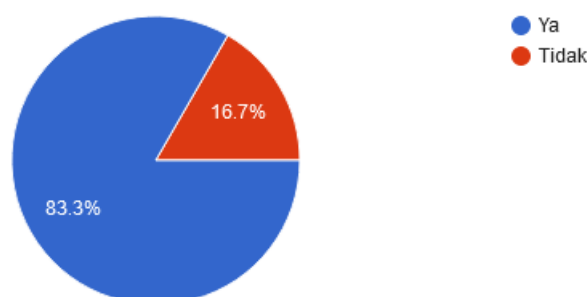
Berikut ini adalah masukan yang didapatkan:

- Pastikan pdf bisa muncul di berbagai *device*
- Mengubah tampilan agar soal lebih mudah dibaca
- Buat dark theme

2. Editor kode

Apakah Anda sudah mencoba fitur ini?

12 responses



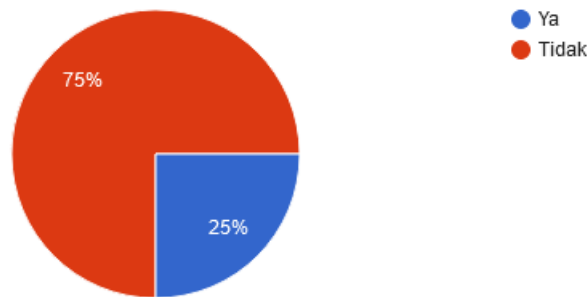
Gambar 5.8: Hasil survei bagian 2 pertanyaan 1

(a) Apakah Anda sudah mencoba fitur ini?

83% dari partisipan menjawab ya, hal ini menunjukkan bahwa mayoritas partisipan 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

(b) Apakah terdapat kendala saat menggunakan fitur ini?

75% dari partisipan menjawab tidak, hal ini menunjukkan bahwa mayoritas partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.9.

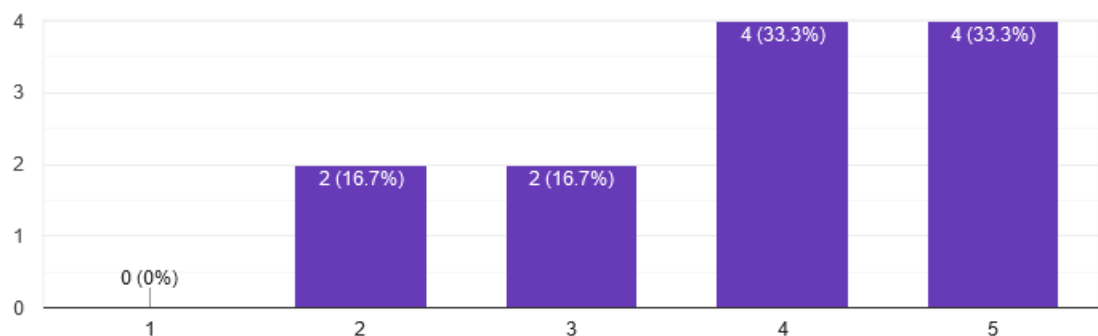
(c) Jika ya, apa kendala yang dialami?

Berikut ini kendala yang dialami:

- Terkadang jenis file tidak muncul
- Ukuran editor kurang besar
- 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.

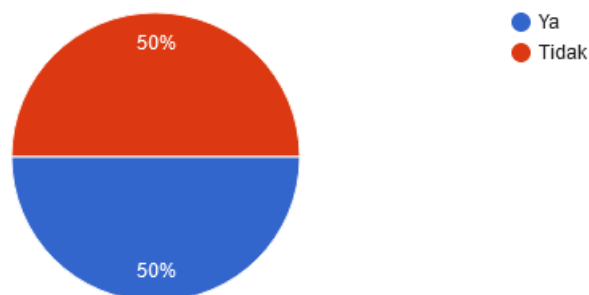
(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

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



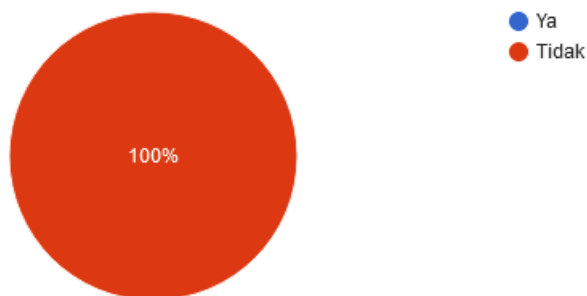
Gambar 5.11: Hasil survei bagian 3 pertanyaan 1

(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?

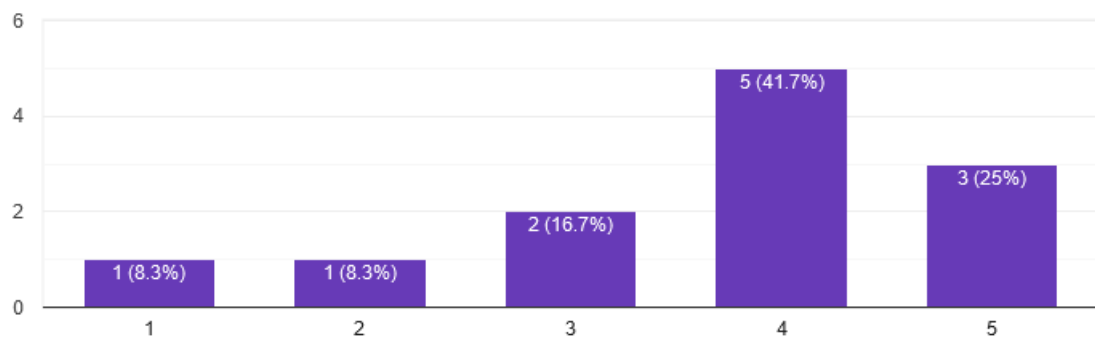
100% dari partisipan menjawab tidak, hal ini menunjukkan bahwa seluruh partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.12.

(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?

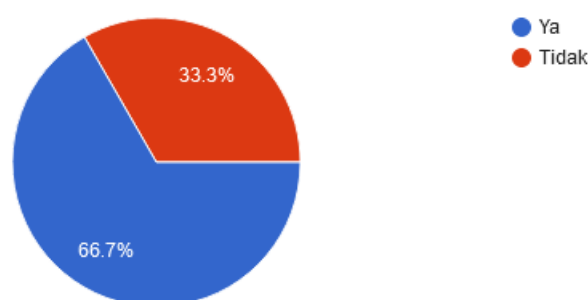
Berikut ini adalah masukan yang didapatkan:

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

4. Menjalankan kode dengan tes kasus

Apakah Anda sudah mencoba fitur ini?

12 responses



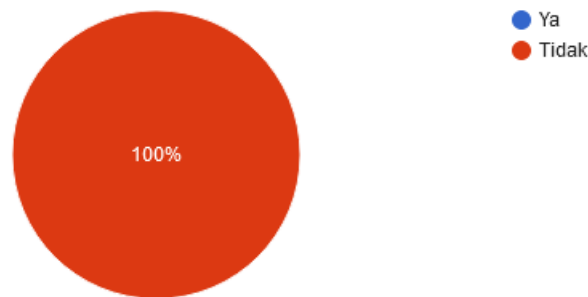
Gambar 5.14: Hasil survei bagian 4 pertanyaan 1

(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?

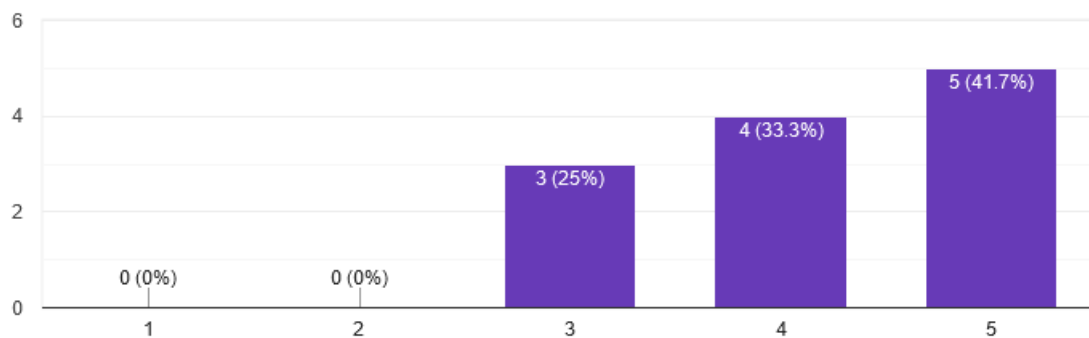
100% dari partisipan menjawab tidak, hal ini menunjukkan bahwa seluruh partisipan tidak menemukan kendala pada fitur ini. Hasil untuk pertanyaan ini dapat dilihat pada gambar 5.15.

(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.

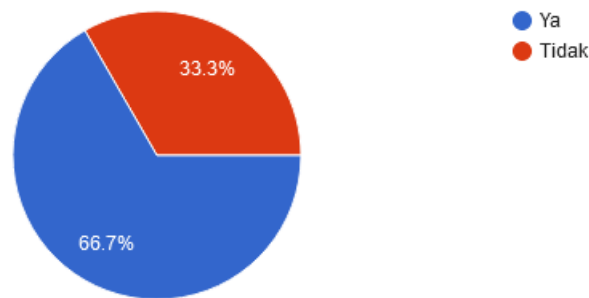
(e) Apakah ada pendapat/saran/masukan untuk fitur ini?

Tidak ada partisipan yang memberi masukan pada fitur ini.

5. Mengumpulkan kode melalui IDE

Apakah Anda sudah mencoba fitur ini?

12 responses



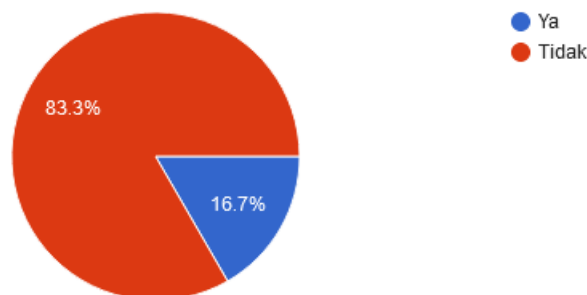
Gambar 5.17: Hasil survei bagian 5 pertanyaan 1

(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.

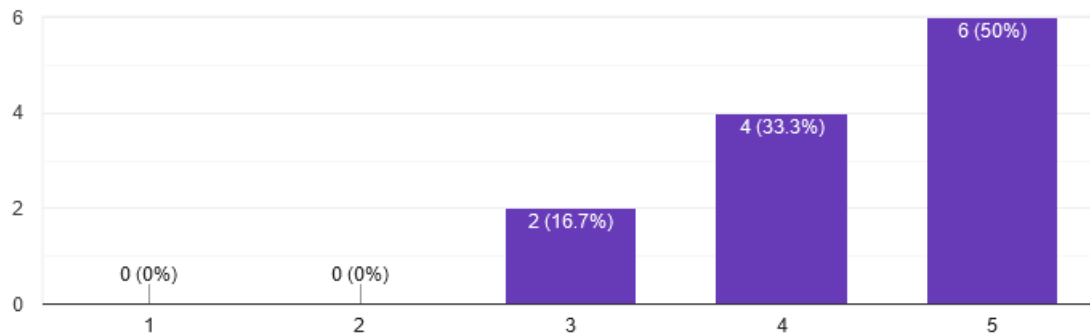
(c) Jika ya, apa kendala yang dialami?

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

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

DAFTAR REFERENSI

- [1] 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.
- [2] 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.
- [3] Version 1.4 (2014) *Sharif Judge Documentation*. Mohammad Javad Naderi. Tehran, Iran.
- [4] Vallian, S. (2018) Kustomisasi Sharif Judge Untuk Kebutuhan Program Studi Teknik Informatika. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [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 CONTROLLER

Kode A.1: 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 ( $hj_now() < $starttime)
25            show_error('Selected assignment has not started.');
```

Kode A.2: 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 . '/'.
    $this->user->username;
43     $file_path = $user_dir . '/' . EDITOR_FILE_NAME . '.' . EDITOR_FILE_EXT;
44     $input_path = $user_dir . '/' . EDITOR_IN_NAME . '.' . EDITOR_FILE_EXT;
45     $output_path = $user_dir . '/' . EDITOR_OUT_NAME . '.' . EDITOR_FILE_EXT;
46
47     $this->load->helper('file');
48     if(!write_file($input_path, ' ')){
49         if(!write_file($output_path, ' ')){
50             if (!file_exists($file_path)){
51                 $response = json_encode(array(content=>'', message=>'No saved file'));
52             }
53             else{
54                 $file_content = file_get_contents($file_path);
55                 if ($file_content === FALSE){
56                     $response = json_encode(array(content=>'', message=>'Unable to load'));
57                 }
58                 else{
59                     addslashes($file_content);
60                     $response = json_encode(array(content=>$file_content, message=>'Loaded'));
61                 }
62             }
63             echo $response;
64         }
65
66 // -----
67 +
68 /**
69 * Save code to editor file and submit/execute if needed
70 */
71 public function save($type = FALSE){
72     $data = $_POST['code_editor'];
73     $problem_id = $_POST['problem_id'];
74     $language = $_POST['language'];
75
76     $user_dir = rtrim($this->assignment_root, '/') . '/assignment-' . $this->user->selected_assignment['id'] . '/p' . $problem_id . '/'.
    $this->user->username;
77     if (!file_exists($user_dir)){
78         mkdir($user_dir, 0700);
79     }
80     $file_path = $user_dir . '/' . EDITOR_FILE_NAME . '.' . EDITOR_FILE_EXT;
81     $input_path = $user_dir . '/' . EDITOR_IN_NAME . '.' . EDITOR_FILE_EXT;
82
83     $this->load->helper('file');
84     if (!write_file($file_path, $data)){
85         $response = json_encode(array(status=>FALSE, message=>'Unable to save'));
86         echo $response;
87     }
88     else{
89         $response = json_encode(array(status=>TRUE, message=>'Saved'));
90         if($type === FALSE){
91             echo $response;
92         }
93         else{
94             $now = shj_now();
95             if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $this->problem['id']
96             ')){
97                 $response = json_encode(array(status=>FALSE, message=>'You have already submitted for this problem. Your last
98                 submission is still in queue.'));
99                 echo $response;
100             }
101             else if ($this->user->level==0 && !$this->user->selected_assignment['open']){
102                 $response = json_encode(array(status=>FALSE, message=>'Selected assignment has been closed.'));
103                 echo $response;
104             }
105             else if ($now < strtotime($this->user->selected_assignment['start_time'])){
106                 $response = json_encode(array(status=>FALSE, message=>'Selected assignment has not started.'));
107                 echo $response;
108             }
109             else if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['
110             extra_time']){
111                 $response = json_encode(array(status=>FALSE, message=>'Selected assignment has finished.'));
112                 echo $response;
113             }
114             else if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->
115             username)){
116                 $response = json_encode(array(status=>FALSE, message=>'You are not registered for submitting.'));
117                 echo $response;
118             }
119             else{
120                 if($type === 'submit'){
121                     $this->_submit($data, $problem_id, $language, $user_dir);
122                 }
123                 else if($type === 'execute'){

```

```

121 +         $editor_input = $_POST['editor_input'];
122 +         if (!write_file($input_path, $editor_input)){
123 +             $response = json_encode(array(status=>FALSE, message=>'Unable to write input file'));
124 +             echo $response;
125 +         }
126 +         else{
127 +             $this->execute($data, $problem_id, $language, $user_dir);
128 +         }
129 +     }
130 + }
131 + }
132 + }
133 +
134 +
135 + }
136 +
137 +
138 + // -----
139 +
140 + /**
141 +  * Add code to queue for judging
142 +  */
143 + private function _submit($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_name = $file_name.'-'.($this->user->selected_assignment['total_submits']+1);
148 +     $file_path = $user_dir.'/'.$file_name.'.'.$file_ext;
149 +
150 +     foreach($this->problems as $item)
151 +     {
152 +         if ($item['id'] == $problem_id)
153 +         {
154 +             $this->problem = $item;
155 +             break;
156 +         }
157 +     }
158 +     if (!write_file($file_path, $data)){
159 +         $response = json_encode(array(status=>FALSE, message=>'Unable to submit'));
160 +     }
161 +     else{
162 +         $this->load->model('submit_model');
163 +
164 +         $submit_info = array(
165 +             'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
166 +             'username' => $this->user->username,
167 +             'assignment' => $this->user->selected_assignment['id'],
168 +             'problem' => $problem_id,
169 +             'file_name' => $file_name,
170 +             'main_file_name' => $file_name,
171 +             'file_type' => $file_type,
172 +             'coefficient' => $this->coefficient,
173 +             'pre_score' => 0,
174 +             'time' => shj_now_str(),
175 +         );
176 +         if ($this->problem['is_upload_only'] == 0)
177 +         {
178 +             $this->queue_model->add_to_queue($submit_info);
179 +             process_the_queue();
180 +         }
181 +         else
182 +         {
183 +             $this->submit_model->add_upload_only($submit_info);
184 +         }
185 +         $response = json_encode(array(status=>TRUE, message=>'Submitted'));
186 +     }
187 +     echo $response;
188 + }
189 +
190 +
191 + // -----
192 +
193 + /**
194 +  * Add code to queue for execution only
195 +  */
196 + private function _execute($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_name = $file_name.'-'.EDITOR_SUBMIT_ID;
201 +     $file_path = $user_dir.'/'.$file_name.'.'.$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_name,
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 $response;
235 + }
236 +
237 +
238 + // -----
239 +
240 + /**
241 +  * Load output file as execution result
242 +  */
243 + public function get_output($problem_id){
244 +     $user_dir = rtrim($this->assignment_root, '/') . '/assignment_' . $this->user->selected_assignment['id'] . '/' . $problem_id . '/';
245 +     $this->user->username;
246 +     $file_path = $user_dir . '/' . EDITOR_OUT_NAME . '/' . EDITOR_FILE_EXT;
247 +
248 +     if (!file_exists($file_path)){
249 +         $response = json_encode(array(status=>FALSE, content=>''));
250 +     }
251 +     else{
252 +         $this->load->helper('file');
253 +         $file_content = file_get_contents($file_path);
254 +         if ($file_content === FALSE){
255 +             $response = json_encode(array(status=>FALSE, content=>''));
256 +         }
257 +         else{
258 +             $complete_status = strpos($file_content, 'Total Execution Time');
259 +             if($complete_status === FALSE){
260 +                 $response = json_encode(array(status=>FALSE, content=>$file_content));
261 +             }
262 +             else{
263 +                 $response = json_encode(array(status=>TRUE, content=>$file_content));
264 +             }
265 +         }
266 +     }
267 +     echo $response;
268 + }

```

Kode A.3: 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
31 @@ -101,7 +108,7 @@ class Queueprocess extends CI_Controller
32     $diff_arg = $problem['diff_arg'];
33     $output_size_limit = $this->settings_model->get_setting('output_size_limit') * 1024;
34
35 -     $cmd = "cd $tester_path;\n./tester.sh $problem_dir ".escapeshellarg($username)." ".escapeshellarg($main_filename)." ".
36 -     escapeshellarg($raw_filename)." $file_type $time_limit $time_limit_int $memory_limit $output_size_limit $diff_cmd $diff_arg
37 -     $op1 $op2 $op3 $op4 $op5 $op6";
38 +     $cmd = "cd $tester_path;\n./tester.sh $problem_dir ".escapeshellarg($username)." ".escapeshellarg($main_filename)." ".
39 +     escapeshellarg($raw_filename)." $file_type $time_limit $time_limit_int $memory_limit $output_size_limit $diff_cmd $diff_arg
40 +     $op1 $op2 $op3 $op4 $op5 $op6 $op7";
41
42     file_put_contents($user_dir.'/log', $cmd);

```


LAMPIRAN B

KODE PROGRAM MODEL

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

```

Kode B.2: 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)

```

LAMPIRAN C

KODE PROGRAM VIEW

Kode C.1: 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
34 @@ -55,18 +47,47 @@
35 </select>
36 {{ form_error('language','<div class="shj_error">','</div>') }}
37 </p>
38 - <p class="input_p">
39 -   <label for="file" class="tiny">File:</label>
40 -   <input type="file" id="file" class="sharif_input medium" name="userfile" />
41 -   {% if upload_state == 'error' %}
42 -   <div class="shj_error">Error uploading file.</div>
43 -   {% elseif upload_state == 'ok' %}
44 -   <div class="shj_ok">File uploaded successfully. See the result in 'All Submissions'.</div>
45 -   {% endif %}
46 - </p>
47 - <p class="input_p">
48 -   <input type="submit" value="Submit" class="sharif_input"/>
49 - </p>
50 + <br>
51 + <details>
52 +   <summary>Submit by File Upload</summary>
53 +   <p class="input_p upload_hidden">
54 +     <label for="file" class="tiny">File:</label>
55 +     <input type="file" id="file" class="sharif_input medium" name="userfile" />
56 +     {% if upload_state == 'error' %}
57 +     <div class="shj_error">Error uploading file.</div>
58 +     {% elseif upload_state == 'ok' %}
59 +     <div class="shj_ok">File uploaded successfully. See the result in 'All Submissions'.</div>
60 +     {% endif %}
61 +   </p>
62 +   <p class="input_p upload_hidden">
63 +     <input type="submit" value="Submit" class="sharif_input"/>
64 +   </p>
65 + </details>
66 </form>
67
68 + <br>
69 + <details>
70 +   <summary>Code, Test, and Submit using built-in editor</summary>
71 +   <iframe id="pdf_viewer" src="{{ base_url('assets/pdfjs/web/viewer.html?file=') ~ site_url('assignments/pdf/' ~ user.
72 +     selected_assignment.id ~ '/null/true')}} "></iframe>
73 +   <div id="ide_wrap">
74 +     <fieldset id="editor_wrap">

```

```

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 C.2: 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++)
55             $('#<option_value="'+shj.p[v][i]+'">'+shj.p[v][i]+'</option>').appendTo('select#languages');
56     });
57
58     $("#select#languages").change(function(){
59         if(this.value.toLowerCase().includes("java")){
60             editor.session.setMode("ace/mode/java");
61             disableEditor(false);
62         }
63         else if(this.value.toLowerCase().includes("python")){
64             editor.session.setMode("ace/mode/python");
65             disableEditor(false);
66         }
67         else if(this.value.toLowerCase().includes("c")){
68             editor.session.setMode("ace/mode/c_cpp");
69             disableEditor(false);
70         }
71         else{
72             editor.session.setMode("ace/mode/plain_text");
73             disableEditor(true);
74         }

```

```

75 | });
76 |
77 | $("#editor_save").click(function(){
78 |     disableEditor(true);
79 |     $.ajax({
80 |         type: "POST",
81 |         url: shj.site_url + 'submit/save',
82 |         data: {
83 |             shj_csrf_token: shj.csrf_token,
84 |             code_editor: editor.getValue(),
85 |             problem_id: $("select#problems").val(),
86 |             language: $("select#languages").val(),
87 |         },
88 |         cache: false,
89 |         success: function(data){
90 |             data = JSON.parse(data);
91 |             $("#ajax_status").html(data.message);
92 |             disableEditor(false);
93 |         },
94 |         error: function (error){
95 |             console.error(error);
96 |             disableEditor(false);
97 |         },
98 |     });
99 | });
100 |
101 | $("#editor_submit").click(function(){
102 |     disableEditor(true);
103 |     $.ajax({
104 |         type: "POST",
105 |         url: shj.site_url + 'submit/save/submit',
106 |         data: {
107 |             shj_csrf_token: shj.csrf_token,
108 |             code_editor: editor.getValue(),
109 |             problem_id: $("select#problems").val(),
110 |             language: $("select#languages").val(),
111 |         },
112 |         cache: false,
113 |         success: function(data){
114 |             data = JSON.parse(data);
115 |             $("#ajax_status").html(data.message);
116 |             disableEditor(false);
117 |             if(data.status){
118 |                 window.location.href = shj.site_url + 'submissions/all';
119 |             }
120 |         },
121 |         error: function (error){
122 |             console.error(error);
123 |             disableEditor(false);
124 |         },
125 |     });
126 | });
127 |
128 | $("#editor_execute").click(function(){
129 |     disableEditor(true);
130 |     $.ajax({
131 |         type: "POST",
132 |         url: shj.site_url + 'submit/save/execute',
133 |         data: {
134 |             shj_csrf_token: shj.csrf_token,
135 |             code_editor: editor.getValue(),
136 |             editor_input: $('textarea#editor_input').val(),
137 |             problem_id: $("select#problems").val(),
138 |             language: $("select#languages").val(),
139 |         },
140 |         cache: false,
141 |         success: function(data){
142 |             data = JSON.parse(data);
143 |             $("#ajax_status").html(data.message);
144 |             if(data.status){
145 |                 (function update() {
146 |                     $.ajax({
147 |                         url: shj.site_url + 'submit/get_output/' + $("select#problems").val(),
148 |                         cache: false,
149 |                         success: function (data){
150 |                             data = JSON.parse(data);
151 |                             $('textarea#editor_output').val(data.content);
152 |                             if(!data.status){
153 |                                 setTimeout(update, 1000);
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 |             }
167 |             else{
168 |                 disableEditor(false);
169 |             }
170 |         },
171 |         error: function (error){
172 |             console.error(error);
173 |             disableEditor(false);

```

```

174         },
175     });
176 });
177
178     loadCode($("#select#problems").val());
179 });

```

Kode C.3: *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 }

```

LAMPIRAN D

KODE PROGRAM LAINNYA

Kode D.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 D.2: 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 +%s%N)/1000000));
51     shj_log "\nTotal Execution Time: $((END-START)) ms"
52 +    shj_log_exec "\nTotal Execution Time: $((END-START)) ms"
```

```

53     echo $@
54     exit 0
55 }
56 @@ -189,7 +209,6 @@ elif [[ $EXT = "java" ]]; then
57 fi
58
59
60
61 COMPILE_BEGIN_TIME=$((date +%sN)/1000000));
62
63 #####
64 @@ -198,24 +217,45 @@ COMPILE_BEGIN_TIME=$((date +%sN)/1000000));
65
66 if [ "$EXT" = "java" ]; then
67     cp ../java.policy java.policy
68 +   MAINCLASS=$(grep -e 'public class\|public static void main\>' ${PROBLEMPATH}/${UN}/${FILENAME}.java | grep -B1 "public
69 +   static void main" | grep '\<class\>' | sed "s/^.*class \+//;s/.*$/")
70 +   if [ $MAINFILENAME != $MAINCLASS ]; then
71 +       MAINFILENAME=$MAINCLASS
72 +   fi
73     cp $PROBLEMPATH/$UN/$FILENAME.java $MAINFILENAME.java
74 -   shj_log "Compiling as Java"
75 +   shj_log_exec "Compiling as Java"
76 +   shj_log_exec ""
77 +   shj_log "Compiling as Java $MAINFILENAME.java"
78     javac $MAINFILENAME.java >/dev/null 2>cerr
79     EXITCODE=$?
80     COMPILE_END_TIME=$((date +%sN)/1000000));
81     shj_log "Compiled. Exit Code=$EXITCODE Execution Time: $((COMPILE_END_TIME-COMPILE_BEGIN_TIME)) ms"
82 -   if [ $EXITCODE -ne 0 ]; then
83 -       shj_log "Compile Error"
84 -       shj_log "$(cat cerr|head -10)"
85 -       echo '<span class="shj_b">Compile Error</span>' >$PROBLEMPATH/$UN/result.html
86 -       echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
87 -       #filepath=$(echo "${JAIL}/${FILENAME}.${EXT}" | sed 's/\//\\//g') #replacing / with \
88 -       (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'/\&quot;/g') >> $PROBLEMPATH/
89 -       $UN/result.html
90 -       #$(cat $JAIL/cerr) >> $PROBLEMPATH/$UN/result.html
91 -       echo "</span>" >> $PROBLEMPATH/$UN/result.html
92 -       cd ..
93 -       rm -r $JAIL >/dev/null 2>/dev/null
94 -       shj_finish "Compilation Error"
95 +   if [ $EXITCODE -ne 0 ]; then
96 +       MSGCHECK=$(cat cerr | grep -e '\<should be declared in a file named\>' | grep -Po '[\w]+?(?=\ is public)')
97 +       echo "$MSGCHECK">> $PROBLEMPATH/$UN/result.html
98 +       if [ -n "${MSGCHECK}" ]; then
99 +           MAINFILENAME=$MSGCHECK
100 +           cp $PROBLEMPATH/$UN/$FILENAME.java $MAINFILENAME.java
101 +           shj_log "Compiling as Java $MAINFILENAME.java"
102 +           javac $MAINFILENAME.java >/dev/null 2>cerr
103 +           EXITCODE=$?
104 +           COMPILE_END_TIME=$((date +%sN)/1000000));
105 +           shj_log "Compiled. Exit Code=$EXITCODE Execution Time: $((COMPILE_END_TIME-COMPILE_BEGIN_TIME)) ms"
106 +       fi
107 +       if [ $EXITCODE -ne 0 ]; then
108 +           shj_log "Compile Error"
109 +           shj_log "$(cat cerr|head -10)"
110 +           echo '<span class="shj_b">Compile Error</span>' >$PROBLEMPATH/$UN/result.html
111 +           echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
112 +           ERRMSG=$(cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'/\&quot;/g')
113 +           echo "$ERRMSG">> $PROBLEMPATH/$UN/result.html
114 +           #filepath=$(echo "${JAIL}/${FILENAME}.${EXT}" | sed 's/\//\\//g') #replacing / with \
115 +           #$(cat $JAIL/cerr) >> $PROBLEMPATH/$UN/result.html
116 +           echo "</span>" >> $PROBLEMPATH/$UN/result.html
117 +           shj_log_exec "$(cat cerr)"
118 +           cd ..
119 +           rm -r $JAIL >/dev/null 2>/dev/null
120 +           shj_finish "Compilation Error"
121 +       fi
122 +   fi
123 fi
124
125 @@ -228,6 +268,8 @@ fi
126 if [ "$EXT" = "py2" ]; then
127     cp $PROBLEMPATH/$UN/$FILENAME.py $FILENAME.py
128     shj_log "Checking Python Syntax"
129 +   shj_log_exec "Checking Python Syntax"
130 +   shj_log_exec ""
131     python2 -0 -m py_compile $FILENAME.py >/dev/null 2>cerr
132     EXITCODE=$?
133     COMPILE_END_TIME=$((date +%sN)/1000000));
134     @@ -239,6 +281,7 @@ if [ "$EXT" = "py2" ]; then
135     echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
136     (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/>/\&gt;/g' | sed 's/'/\&quot;/g') >> $PROBLEMPATH/
137     $UN/result.html
138     echo "</span>" >> $PROBLEMPATH/$UN/result.html
139 +   shj_log_exec "$(cat cerr)"
140 +   cd ..
141     rm -r $JAIL >/dev/null 2>/dev/null
142     shj_finish "Syntax Error"
143 @@ -259,6 +302,8 @@ fi
144 if [ "$EXT" = "py3" ]; then
145     cp $PROBLEMPATH/$UN/$FILENAME.py $FILENAME.py
146     shj_log "Checking Python Syntax"
147 +   shj_log_exec "Checking Python Syntax"
148 +   shj_log_exec ""
149     python3 -0 -m py_compile $FILENAME.py >/dev/null 2>cerr
150     EXITCODE=$?
151     COMPILE_END_TIME=$((date +%sN)/1000000));

```



```

149 @@ -270,6 +315,7 @@ if [ "$EXT" = "py3" ]; then
150     echo '<span class="shj_r">' >> $PROBLEMPATH/$UN/result.html
151     (cat cerr | head -10 | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/"/\&quot;/g') >> $PROBLEMPATH/
152     $UN/result.html
153     echo "</span>" >> $PROBLEMPATH/$UN/result.html
154 +     shj_log_exec "(cat cerr)"
155     cd ..
156     rm -r $JAIL >/dev/null 2>/dev/null
157     shj_finish "Syntax Error"
158 @@ -295,6 +341,8 @@ if [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
159     EXEFILE="s_$(echo $FILENAME | sed 's/[^a-zA-Z0-9]//g')".# Name of executable file
160     cp $PROBLEMPATH/$UN/$FILENAME.$EXT code.c
161     shj_log "Compiling as $EXT"
162 +     shj_log_exec "Compiling as $EXT"
163 +     shj_log_exec ""
164     if $SANDBOX_ON; then
165         shj_log "Enabling EasySandbox"
166         if cp ../easysandbox/EasySandbox.so EasySandbox.so; then
167 @@ -360,6 +408,7 @@ if [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
168     (cat cerr | sed 's/&/\&amp;/g' | sed 's/</\&lt;/g' | sed 's/"/\&quot;/g') >> $PROBLEMPATH/$UN/
169     result.html
170     fi
171     echo "</span>" >> $PROBLEMPATH/$UN/result.html
172 +     shj_log_exec "(cat cerr)"
173     cd ..
174     rm -r $JAIL >/dev/null 2>/dev/null
175     shj_finish "Compilation Error"
176 @@ -372,194 +421,303 @@ fi
177 ##### TESTING #####
178 -shj_log "\nTesting..."
179 -shj_log "$TST test cases found"
180 +if ! $EXEC_ONLY; then
181 +     shj_log "\nTesting..."
182 +     shj_log "$TST test cases found"
183
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 +             cd ..
209 +             rm -r $JAIL >/dev/null 2>/dev/null
210 +             shj_finish "Invalid Tester Code"
211 +         else
212 +             shj_log "Tester compiled. Execution Time: $((TST_COMPILE_END_TIME-TST_COMPILE_BEGIN_TIME)) ms"
213 +             fi
214 +         fi
215     fi
216 -fi
217
218 -if [ -f "$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 -PASSEDTST=0
230 +     PASSEDTST=0
231
232 -for((i=1;i<=TST;i++)); do
233 -     shj_log "\n=== TEST $i ==="
234 -     echo "<span class='shj_b'>Test $i</span>" >>$PROBLEMPATH/$UN/result.html
235 -     touch err
236 -     if [ "$EXT" = "java" ]; then
237 -         if $PERL_EXISTS; then
238 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout --just-kill --nosandbox -l
239 -             $OUTLIMIT -t $TIMELIMIT java -mx${MEMLIMIT}k $JAVA_POLICY $MAINFILENAME"
240 -         else
241 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "java -mx${MEMLIMIT}k $JAVA_POLICY
242 -             $MAINFILENAME"
243 -         fi

```

```

244 - fi
245 - EXITCODE=?
246 - if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
247 -     shj_log "Memory Limit Exceeded"
248 -     echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
249 -     continue
250 - fi
251 - if grep -q -m 1 "Exception in" err; then # show Exception
252 -     javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\.[a-zA-Z\.]*' | head -1 | head -c 80'
253 -     javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
254 -     shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
255 -     # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
256 -     if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname\$" ../java_exceptions_list; then
257 -         echo "<span class=\"shj_o\">Runtime Error ($javaexceptionname)</span>" >>$PROBLEMPATH/$UN/result.html
258 + for((i=1;i<=TST;i++)); do
259 +     shj_log "\n=== TEST $i ==="
260 +     echo "<span class=\"shj_b\">Test $i</span>" >>$PROBLEMPATH/$UN/result.html
261 +
262 +     touch err
263 +
264 +     if [ "$EXT" = "java" ]; then
265 +         if $PERL_EXISTS; then
266 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout ---just-kill ---nosandbox
267 -             else
268 -                 echo "<span class=\"shj_o\">Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
269 +             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "java -mx${MEMLIMIT}k
270 +             $JAVA_POLICY $MAINFILENAME"
271 +             fi
272 +             continue
273 -         elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
274 -             # $TIMEOUT ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
275 -             if $SANDBOX_ON; then
276 -                 #LD_PRELOAD=./EasySandbox.so ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
277 +                 EXITCODE=?
278 +                 if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
279 +                     shj_log "Memory Limit Exceeded"
280 +                     echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
281 +                     continue
282 +                 fi
283 +                 if grep -q -m 1 "Exception in" err; then # show Exception
284 +                     javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\.[a-zA-Z\.]*' | head -1 | head -c 80'
285 +                     javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
286 +                     shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
287 +                     # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
288 +                     if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname\$" ../java_exceptions_list; then
289 +                         echo "<span class=\"shj_o\">Runtime Error ($javaexceptionname)</span>" >>$PROBLEMPATH/$UN/result.html
290 +                     else
291 +                         echo "<span class=\"shj_o\">Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
292 +                     fi
293 +                     continue
294 +                 fi
295 +                 elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
296 +                     # $TIMEOUT ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
297 +                     if $SANDBOX_ON; then
298 +                         #LD_PRELOAD=./EasySandbox.so ./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
299 +                         if $PERL_EXISTS; then
300 +                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout ---just-kill ---
301 +                             sandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"
302 +                         else
303 +                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "LD_PRELOAD=./EasySandbox.so
304 +                             ./EXEFILE"
305 +                             fi
306 +                             # remove <<entering SECCOMP mode>> from beginning of output:
307 +                             tail -n +2 out >thetemp && mv thetemp out
308 +                         else
309 +                             #./FILENAME <$PROBLEMPATH/in/input$i.txt >out 2>/dev/null
310 +                             if $PERL_EXISTS; then
311 +                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout ---just-kill ---
312 +                                 nosandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"
313 +                             else
314 +                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./EXEFILE"
315 +                             fi
316 +                             EXITCODE=?
317 +                             fi
318 +                             elif [ "$EXT" = "py2" ]; then
319 -                             if ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout ---just-kill ---sandbox
320 +                             -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"
321 +                             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout ---just-kill ---nosandbox
322 -                             -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -O $FILENAME.py"
323 +                             else
324 +                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "LD_PRELOAD=./EasySandbox.so ./
325 +                                 $EXEFILE"
326 +                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -O $FILENAME.py"
327 +                                 fi
328 +                                 EXITCODE=?
329 +                                 # remove <<entering SECCOMP mode>> from beginning of output:
330 +                                 tail -n +2 out >thetemp && mv thetemp out
331 +                             else
332 +                                 ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt >out 2>/dev/null
333 +                             elif [ "$EXT" = "py3" ]; then
334 +                                 if $PERL_EXISTS; then
335 +                                     ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt ./timeout ---just-kill ---nosandbox
336 +                                     -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"

```

```

334 +         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox
-l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -0 $FILENAME.py"
335 -     else
336 -         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./EXEFILE"
337 +         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -0 $FILENAME.py"
338 -     fi
339 -     EXITCODE=?
340 - fi
341 -
342 - elif [ "$EXT" = "py2" ]; then
343 -     if $PERL_EXISTS; then
344 -         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox -l
345 - $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -0 $FILENAME.py"
346 -     else
347 -         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python2 -0 $FILENAME.py"
348 +         shj_log "File Format Not Supported"
349 +         cd ..
350 +         rm -r $JAIL >/dev/null 2>/dev/null
351 +         shj_finish "File Format Not Supported"
352 -     fi
353 -     EXITCODE=?
354 -
355 - elif [ "$EXT" = "py3" ]; then
356 -     if $PERL_EXISTS; then
357 -         ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "./timeout --just-kill --nosandbox -l
358 - $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -0 $FILENAME.py"
359 -         shj_log "Exit Code = $EXITCODE"
360 +
361 +         if ! grep -q "FINISHED" err; then
362 +             if grep -q "SHJ_TIME" err; then
363 +                 t='grep "SHJ_TIME" err|cut -d" " -f3'
364 +                 shj_log "Time Limit Exceeded ($t s)"
365 +                 echo "<span class=\"shj_o\">Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
366 +                 continue
367 +             elif grep -q "SHJ_MEM" err; then
368 +                 shj_log "Memory Limit Exceeded"
369 +                 echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
370 +                 continue
371 +             elif grep -q "SHJ_HANGUP" err; then
372 +                 shj_log "Hang Up"
373 +                 echo "<span class=\"shj_o\">Process hanged up</span>" >>$PROBLEMPATH/$UN/result.html
374 +                 continue
375 +             elif grep -q "SHJ_SIGNAL" err; then
376 +                 shj_log "Killed by a signal"
377 +                 echo "<span class=\"shj_o\">Killed by a signal</span>" >>$PROBLEMPATH/$UN/result.html
378 +                 continue
379 +             elif grep -q "SHJ_OUTSIZE" err; then
380 +                 shj_log "Output Size Limit Exceeded"
381 +                 echo "<span class=\"shj_o\">Output Size Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
382 +                 continue
383 +             fi
384 -         else
385 -             ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/in/input$i.txt "python3 -0 $FILENAME.py"
386 -             t='grep "FINISHED" err|cut -d" " -f3'
387 -             shj_log "Time: $t s"
388 -         fi
389 -         EXITCODE=?
390 -     else
391 -         shj_log "File Format Not Supported"
392 -         cd ..
393 -         rm -r $JAIL >/dev/null 2>/dev/null
394 -         shj_finish "File Format Not Supported"
395 -     fi
396 -
397 -     shj_log "Exit Code = $EXITCODE"
398 -
399 -     if ! grep -q "FINISHED" err; then
400 -         if grep -q "SHJ_TIME" err; then
401 -             t='grep "SHJ_TIME" err|cut -d" " -f3'
402 -             shj_log "Time Limit Exceeded ($t s)"
403 -             echo "<span class=\"shj_o\">Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
404 -             continue
405 -         elif grep -q "SHJ_MEM" err; then
406 -             shj_log "Memory Limit Exceeded"
407 -             echo "<span class=\"shj_o\">Memory Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
408 -             continue
409 -         elif grep -q "SHJ_HANGUP" err; then
410 -             shj_log "Hang Up"
411 -             echo "<span class=\"shj_o\">Process hanged up</span>" >>$PROBLEMPATH/$UN/result.html
412 -             continue
413 -         elif grep -q "SHJ_SIGNAL" err; then
414 -             shj_log "Killed by a signal"
415 -             echo "<span class=\"shj_o\">Killed by a signal</span>" >>$PROBLEMPATH/$UN/result.html
416 -             continue
417 -         elif grep -q "SHJ_OUTSIZE" err; then
418 -             shj_log "Output Size Limit Exceeded"
419 -             echo "<span class=\"shj_o\">Output Size Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
420 +
421 +         if [ $EXITCODE -eq 137 ]; then
422 +             #shj_log "Time Limit Exceeded (Exit code=$EXITCODE)"
423 +             #echo "<span style='color: orange;'>Time Limit Exceeded</span>" >>$PROBLEMPATH/$UN/result.html
424 +             shj_log "Killed"
425 +             echo "<span class=\"shj_o\">Killed</span>" >>$PROBLEMPATH/$UN/result.html
426 +             continue
427 -         fi
428 -     else
429 -         t='grep "FINISHED" err|cut -d" " -f3'
430 -         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 -     fi
455 -     if [ $EXITCODE -ne 0 ]; then
456 -         shj_log "Runtime Error"
457 -         echo "<span class='shj_o'>Runtime Error</span>" >>$PROBLEMPATH/$UN/result.html
458 -         continue
459 -     fi
460 - else
461 -     cp $PROBLEMPATH/out/output$i.txt correctout
462 -     if [ "$DIFFOPTION" = "ignore" ]; then
463 -         # Removing all newlines and whitespaces before diff
464 -         tr -d '\t\n\r\f' <out >tmp1 && mv tmp1 out;
465 -         tr -d '\t\n\r\f' <correctout >tmp1 && mv tmp1 correctout;
466 -     fi
467 -     # checking correctness of output
468 -     ACCEPTED=false
469 -     if [ -f shj_tester ]; then
470 -         ./shj_tester $PROBLEMPATH/in/input$i.txt $PROBLEMPATH/out/output$i.txt out
471 -         EC=$?
472 -         if [ $EC -eq 0 ]; then
473 -             ACCEPTED=true
474 -         fi
475 -     else
476 -         cp $PROBLEMPATH/out/output$i.txt correctout
477 -         if [ "$DIFFOPTION" = "ignore" ]; then
478 -             # Removing all newlines and whitespaces before diff
479 -             tr -d '\t\n\r\f' <out >tmp1 && mv tmp1 out;
480 -             tr -d '\t\n\r\f' <correctout >tmp1 && mv tmp1 correctout;
481 -         fi
482 -         # Add a newline at the end of both files
483 -         echo "" >> out
484 -         echo "" >> correctout
485 -         if [ "$DIFFTOOL" = "diff" ]; then
486 -             # Add -q to diff options (for faster diff)
487 -             DIFFFARGUMENT="-q $DIFFFARGUMENT"
488 -         fi
489 -         # Compare output files
490 -         if $DIFFTOOL $DIFFFARGUMENT out correctout >/dev/null 2>/dev/null; then
491 -             ACCEPTED=true
492 -         fi
493 -     fi
494 -     # Add a newline at the end of both files
495 -     echo "" >> out
496 -     echo "" >> correctout
497 -     if [ "$DIFFTOOL" = "diff" ]; then
498 -         # Add -q to diff options (for faster diff)
499 -         DIFFFARGUMENT="-q $DIFFFARGUMENT"
500 -     fi
501 -     if $ACCEPTED; then
502 -         shj_log "ACCEPTED"
503 -         echo "<span class='shj_g'>ACCEPT</span>" >>$PROBLEMPATH/$UN/result.html
504 -         ((PASSEDTESTS=PASSEDTESTS+1))
505 -     else
506 -         shj_log "WRONG"
507 -         echo "<span class='shj_r'>WRONG</span>" >>$PROBLEMPATH/$UN/result.html
508 -     fi
509 -     # Compare output files
510 -     if $DIFFTOOL $DIFFFARGUMENT out correctout >/dev/null 2>/dev/null; then
511 -         ACCEPTED=true
512 -     fi
513 - done
514 -
515 - #####
516 - ##### EXECUTION #####
517 - #####
518 -
519 - else
520 -     touch err
521 -     if [ "$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

```

```

527 + EXITCODE=?
528 + if grep -iq -m 1 "Too small initial heap" out || grep -q -m 1 "java.lang.OutOfMemoryError" err; then
529 + shj_log "Memory Limit Exceeded"
530 + echo "Memory Limit Exceeded" >>out
531 + continue
532 + fi
533 + if grep -q -m 1 "Exception in" err; then # show Exception
534 + javaexceptionname='grep -m 1 "Exception in" err | grep -m 1 -oE 'java\[a-zA-Z\].*' | head -1 | head -c 80'
535 + javaexceptionplace='grep -m 1 "$MAINFILENAME.java" err | head -1 | head -c 80'
536 + shj_log "Exception: $javaexceptionname\nMaybe at:$javaexceptionplace"
537 + # if DISPLAY_JAVA_EXCEPTION_ON is true and the exception is in the trusted list, we show the exception name
538 + if $DISPLAY_JAVA_EXCEPTION_ON && grep -q -m 1 "^$javaexceptionname\$" ../java_exceptions_list; then
539 + echo "Runtime Error ($javaexceptionname)" >>out
540 + else
541 + echo "Runtime Error" >>out
542 + fi
543 + continue
544 + fi
545 + elif [ "$EXT" = "c" ] || [ "$EXT" = "cpp" ]; then
546 + if $SANDBOX_ON; then
547 + if $PERL_EXISTS; then
548 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "timeout --just-kill --
sandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"
549 + else
550 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "LD_PRELOAD=./EasySandbox.so ./
EXEFILE"
551 + fi
552 + EXITCODE=?
553 + # remove <<entering SECCOMP mode>> from beginning of output:
554 + tail -n +2 out >thetemp && mv thetemp out
555 + else
556 + if $PERL_EXISTS; then
557 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "timeout --just-kill --
nosandbox -l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT ./EXEFILE"
558 + else
559 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "EXEFILE"
560 + fi
561 + EXITCODE=?
562 + fi
563 +
564 + elif [ "$EXT" = "py2" ]; then
565 + if $PERL_EXISTS; then
566 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "timeout --just-kill --nosandbox -
l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python2 -0 $FILENAME.py"
567 + else
568 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "python2 -0 $FILENAME.py"
569 + fi
570 + EXITCODE=?
571 +
572 + elif [ "$EXT" = "py3" ]; then
573 + if $PERL_EXISTS; then
574 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "timeout --just-kill --nosandbox -
l $OUTLIMIT -t $TIMELIMIT -m $MEMLIMIT python3 -0 $FILENAME.py"
575 + else
576 + ./runcode.sh $EXT $MEMLIMIT $TIMELIMIT $TIMELIMITINT $PROBLEMPATH/$UN/$EXEC_IN "python3 -0 $FILENAME.py"
577 + fi
578 + EXITCODE=?
579 +
580 + else
581 + shj_log "File Format Not Supported"
582 + cd ..
583 + 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,
```