

DUKUNGAN BAHASA JAVASCRIPT PADA SHARIF JUDGE

EDWIN PRANAJAYA—2017730027

1 Data Skripsi

Pembimbing utama/tunggal: **Pascal Alfadian Nugroho**

Pembimbing pendamping: -

Kode Topik : **PAN5291**

Topik ini sudah dikerjakan selama : **1 semester**

Pengambilan pertama kali topik ini pada : Semester **52 - Genap 21/22**

Pengambilan pertama kali topik ini di kuliah : **Skripsi 1**

Tipe Laporan : **B** - Dokumen untuk reviewer pada presentasi dan **review Skripsi 1**

2 Latar Belakang

SharIF-Judge adalah *online judge* gratis dan open source untuk bahasa pemrograman C, C++, Java dan Python. SharIF-Judge merupakan hasil *fork* dari Sharif Judge yang dibuat oleh Mohammad Javad Naderi. Versi bercabang ini mengandung banyak perbaikan, sebagian besar karena kebutuhan fakultas Informatika UNPAR.

SharIF-Judge biasa digunakan untuk mempermudah evaluasi kode program. SharIF-Judge digunakan pada beberapa kuliah di Informatika(IF), seperti Dasar Pemrograman dan Algoritma Struktur Data. Tujuan SharIF-Judge digunakan pada perkuliahan, yaitu agar dapat mempermudah kegiatan belajar mengajar berlangsung. Mahasiswa dapat dengan mudah melakukan pengiriman kode program atau *file* ke SharIF-Judge dan dapat langsung melihat nilainya. Pengajar yang bertanggung jawab atas kegiatan belajar mengajar tersebut tidak akan kesusahan dalam mengumpulkan hasil pekerjaan para pelajar.

Alur dari penggunaan SharIF-Judge ini diawali dengan pengajar yang membuat soal terlebih dahulu. Setelah soal disiapkan, pengajar dapat membuat *assignment* dengan click tombol *add assignment*. Dida-lamnya, pengajar diwajibkan untuk memasukan nama *assignment*, waktu dimulainya pengerjaan, waktu selesai pengerjaan, waktu tambahan pengerjaan, daftar peserta, deskripsi soal, dan kunci jawaban dari soal yang sudah dibuat oleh pengajar. Meskipun SharIF-Judge dapat membaca berbagai bahasa pemrograman, JavaScript bukan salah satu yang dapat dibaca oleh SharIF-Judge.

JavaScript adalah bahasa skrip sisi klien yang berjalan sepenuhnya di dalam *browser web*. Sebagai contoh *menu drop-down* yang mencolok, memindahkan teks, dan mengubah konten yang sekarang tersebar luas di situs web. Semua interaksi tersebut diaktifkan melalui JavaScript. Berdasarkan survey yang diberikan pada stackoverflow pada tahun 2021, selama 9 tahun berturut-turut, JavaScript merupakan bahasa pemrograman yang paling sering digunakan. 64,96% developer dari 83,052 responden di dunia menggunakannya.

Berdasarkan data diatas, ditunjukan bahwa JavaScript termasuk dalam bahasa pemrograman yang sangat populer dan banyak digunakan. Sangat disayangkan jika SharIF-Judge tidak dapat membaca bahasa tersebut. Akan kesulitan bagi para pengajar karena harus mengikuti perkembangan zaman, namun pemeriksaannya tidak mudah. Kostumisasi dari SharIF-Judge akan dilakukan agar *online judge* tersebut dapat membaca JavaScript. Untuk mempermudah kostumasi, pengerjaan akan dilakukan dengan menggunakan OS Linux. Namun dikarenakan OS yang digunakan adalah Windows, akan digunakan *virtual machine* agar dapat mengerjakan dengan OS Linux pada Windows.

3 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah penelitian sebagai berikut:

Bagaimanakah cara agar SharIF-Judge memahami bahasa pemrograman JavaScript untuk dievaluasi.

4 Tujuan

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah sebagai berikut:

Melakukan modifikasi pada SharIF-Judge agar dapat menerima soal JavaScript dan dapat melakukan evaluasi pada JavaScript.

5 Detail Perkembangan Pengerjaan Skripsi

Detail bagian pekerjaan skripsi sesuai dengan rencan kerja/laporan perkembangan terkahir :

1. Melakukan instalasi pada SharIF-Judge

Status : Ada sejak rencana kerja skripsi.

Hasil : Instalasi SharIF-Judge dilakukan melalui github yang dimiliki oleh Informatika UNPAR. Untuk dapat menjalankan SharIF-Judge, diwajibkan untuk memiliki server Linux dan mengikuti persyaratan sebagai berikut:

- Menjalankan PHP versi 5.3 atau versi yang lebih baru.
- Pengguna dapat menjalankan PHP dari command line dan pengguna perlu menginstall paket PHP CLI.
- Memiliki *Mysql* atau *PostgreSql database*.
- PHP memiliki akses untuk menjalankan perintah *shell* terutama untuk fungsi *shell_exec*.
- Untuk melakukan proses kompilasi dan menjalankan kode yang dikumpulkan adalah (*gcc, g++, javac, java, python2, python3 commands*)
- Disarankan untuk melakukan instalasi *Perl* dengan alasan agar memiliki ketepatan waktu, penggunaan *memory* yang terbatas dan memaksimalkan batas ukuran pada *output* kode yang dikirimkan.

Jika persyaratan diatas telah selesai dilakukan, dapat dilanjutkan untuk melakukan instalasi sebagai berikut:

- Mengunduh versi terakhir dari Sharif Judge dan unpack file yang berhasil diunduh, letakan pada direktori html publik
- Untuk mempermudah pindahkan folder *system* dan *application* keluar dari direktori publik dan masukan *path* lengkap pada file index.php

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

- Membuat sebuah *Mysql* atau *PostgreSql database* untuk Sharif Judge.
- Mengatur koneksi database di file application/config/database.php.

```
/* Enter database connection settings here: */
'dbdriver' => 'postgre', // database driver (mysqli, postgre)
'hostname' => 'localhost', // database host
```

```

'username' => ' , // database username
'password' => ' , // database password
'database' => ' , // database name
'dbprefix' => 'shj_', // table prefix
/*****/

```

- Membuat direktori *application/cache/Twig* agar dapat ditulis oleh PHP
- Membuka halaman utama SharIF-Judge pada web browser
- *Log in* menggunakan akun *admin*
- Memindahkan folder *tester* dan *assignments* di luar direktori publik lalu simpan *path* lengkap pada halaman *Settings* . Dua folder tersebut harus dapat ditulis oleh PHP. File-file yang diunggah akan disimpan di folder *assignments* sehingga tidak dapat diakses publik.

2. Menulis dokumen skripsi

Status : Ada sejak rencana kerja skripsi.

Hasil : Dokumen skripsi telah dituliskan hingga Bab 3 yaitu analisis sistem kini dan analisis sistem usulan.

3. Mempelajari javascript

Status : Ada sejak rencana kerja skripsi.

Hasil : Melakukan studi mengenai javascript. Javascript berfungsi untuk membantu membuat fungsionalitas pada halaman web, sedangkan HTML dan CSS membantu membuat desain halaman web. Javascript memiliki beberapa fungsi yang dapat digunakan, yaitu

- **String.** Merupakan nilai yang terdiri dari teks dan dapat berisi huruf, angka, simbol, tanda baca, dan emoji. berikut merupakan contoh kode

EXAMPLE

```
'This is a string.'
```

- **Numbers.** Merupakan nilai yang dapat digunakan dalam operasi matematika

EXAMPLE

```
12345;
```

- **Booleans.** Merupakan nilai yang dapat berupa TRUE atau FALSE dimana nilai ini berarti hanya antara benar atau salah

EXAMPLE

```
var kitchenLights = false;
```

```
kitchenLights = true;
```

```
kitchenLights;
```

OUTPUT:

```
true
```

- **Operators.** Simbol antara nilai yang memungkinkan operasi yang berbeda seperti penambahan, pengurangan, perkalian, dll.

EXAMPLE:

```
1+2;
```

OUTPUT:

```
3
```

- **Variables.** Merupakan sesuatu yang akan dideklarasikan.

EXAMPLE

```
var x = 100;
```

- **Functions.** Merupakan blok kode yang dapat digunakan kembali yang melakukan tugas tertentu, mengambil beberapa bentuk input dan mengembalikan output.

EXAMPLE

```
function addTwoNumbers(x, y) {
    return x + y;
}
```

- **Conditionals.** Merupakan cara untuk engontrol perilaku dalam JavaScript dan menentukan apakah potongan kode dapat dijalankan atau tidak.

EXAMPLE:

```
if (10 > 5) {
    var outcome = "if block";
}
```

OUTPUT:

"if block"

- **Arrays.** Merupakan nilai seperti wadah yang dapat menampung nilai, Elemen pada array tidak semuanya harus memiliki tipe nilai yang sama. Elemen dapat berupa nilai JavaScript apa pun bahkan array lain.

EXAMPLE:

```
var breakfast = ["coffee", "croissant"];
breakfast;
```

OUTPUT:

["coffee", "croissant"]

- **Objects.** Merupakan variabel yang berisi beberapa nilai data Nilai dalam objek JavaScript dikenal sebagai properti.

EXAMPLE:

```
var hodgepodge = [100, "paint", [200, "brush"], false];
hodgepodge;
```

OUTPUT:

[100, "paint", [200, "brush"], false]

- **Input and Output.** Untuk javascript, diperlukan menggunakan Modul *Readline*. pada Node.js Modul *Readline* memungkinkan pembacaan aliran input baris demi baris.

```
var readline = require('readline');
```

4. Melakukan kostumisasi pada SharIF-Judge

Status : ada sejak rencana kerja skripsi

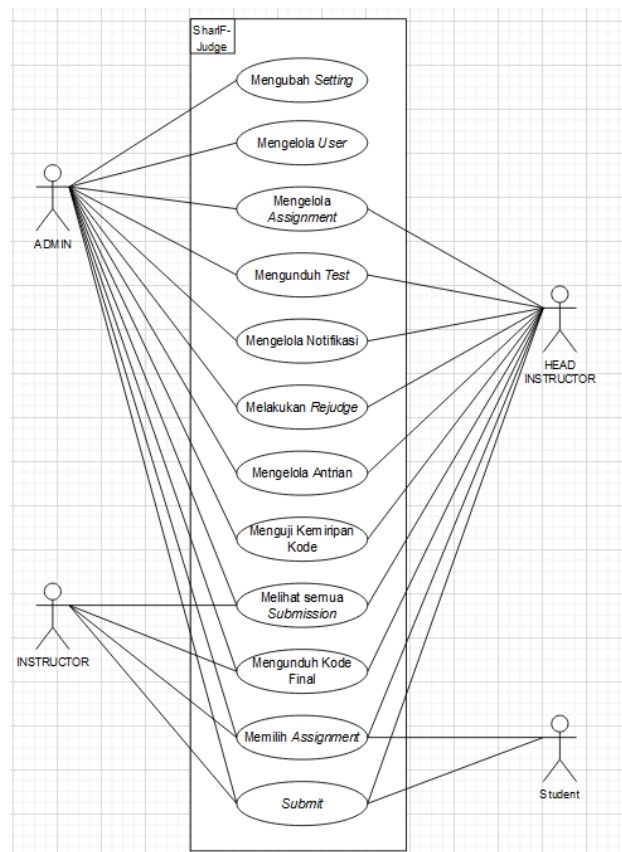
Hasil : Sudah memberikan perubahan pada tampilan sehingga pengguna dapat memilih *javascript* namun belum memiliki fungsi. Hal tersebut dikarenakan ditemukan bug sehingga diprioritaskan dokumen dan dilanjutkan pada skripsi 2. Bug yang ditemukan seperti:

- Ketika ingin memasukan file *test case* muncul error yang disebabkan pada config.php terdapat typo
- Ketika *add assignment* dan memasukan *test case*, *test case* tersebut dikatakan file not found. Hal ini disebabkan oleh *End of Line Sequence* pada *tester.sh* yang berawalan dari tipe CLRF dan dirubah menjadi LF. CR LF adalah singkatan dari "Carriage Return, Line Feed" yang merupakan sisa digital dari mesin tik klasik. Dengan mesin tik, Pengguna harus mendorong "carriage" (benda yang menahan kertas) kembali ke tempatnya, maka "Carriage Return". Sedangkan LF memindahkan kursor ke bawah ke baris berikutnya tanpa kembali ke awal baris.

5. Melakukan eksplorasi pada SharIF-Judge

Status : Ada sejak rencana kerja skripsi.

- **Hasil :** SharIF-Judge adalah *online judge* gratis dan open source untuk bahasa pemrograman C, C++, Java dan Python. SharIF-Judge merupakan hasil *fork* dari Sharif Judge yang dibuat oleh Mohammad Javad Naderi. Versi bercabang ini mengandung banyak perbaikan, sebagian besar karena kebutuhan fakultas Informatika UNPAR. Antarmuka web ditulis dalam PHP (kerangka CodeIgniter) dan backend utama ditulis dalam BASH.-Python di SharIF Judge belum di-sandbox. Hanya tingkat keamanan rendah yang disediakan untuk python. SharIF-Judge dibuat dengan menggunakan sebuah *framework* yang bernama CodeIgniter. CodeIgniter merupakan framework PHP dengan memiliki arsitektur MVC(Model, View, Controller) untuk mempermudah pembangunan web. Berikut ini merupakan fitur-fitur dari SharIF-Judge:



Gambar 1: Use Case Diagram

Pada Gambar 1 ditunjukan bahwa terdapat 4 *role* pada SharIF-Judge dari yang memiliki fungsi paling banyak hingga paling sedikit berturut-turut yaitu Admin, *Head Instructor*, *Instructor*, dan *Student*.

Select	Name	Problems	Submissions	Coefficient	Start Time	Finish Time	Status	PDF	Actions
<input type="checkbox"/>	ada	1 problem	0 submissions	Finished	2022-06-06 00:00:00	2022-06-08 00:00:00	Open		
<input type="checkbox"/>	aaaa	1 problem	0 submissions	Finished	2022-06-05 00:00:00	2022-06-08 00:00:00	Open		
<input checked="" type="checkbox"/>	baba	1 problem	4 submissions	Finished	2022-05-17 00:00:00	2022-05-26 00:00:00	Open		

Gambar 2: Halaman *Assignment*

Pada Gambar 2 merupakan halaman *Assignment*, bagi role *student* halaman *assignment* ini berfungsi sebagai tempat untuk melihat seluruh *assignment* yang ada dan dapat memilih salah satunya. Namun, untuk role Admin, dan *Head Instructor* halaman *assignment* dapat berguna untuk menambahkan *assignment* pada *add assignment*, edit, atau bahkan menghapus *assignment*.

Assignment Name:

Start Time:

Finish Time:

Extra Time (minutes):

Participants: ALL

Tests and Descriptions (zip file): No file chosen

PDF File: No file chosen

Coefficient rule (2):

```

PHP script without <?php ?> tags
Press "test" to enable/disable tabindent

/*
 * Put coefficient (from 100) in variable $coefficient.
 * You can use variables: $extra_time and $delay.
 * $extra_time is the total extra time given to users
 * (in seconds) and $delay is number of seconds passed
 * from finish time (can be negative).
 * In this example, $extra_time is 172800 (2 days):
 */

if ($delay==0)
// no delay
$coefficient = 100;
elseif ($delay>3600)
// delay less than 1 hour
$coefficient = ceil(100/(30*$delay/3600));
elseif ($delay>86400)
// delay more than 1 hour and less than 1 day
$coefficient = 70;

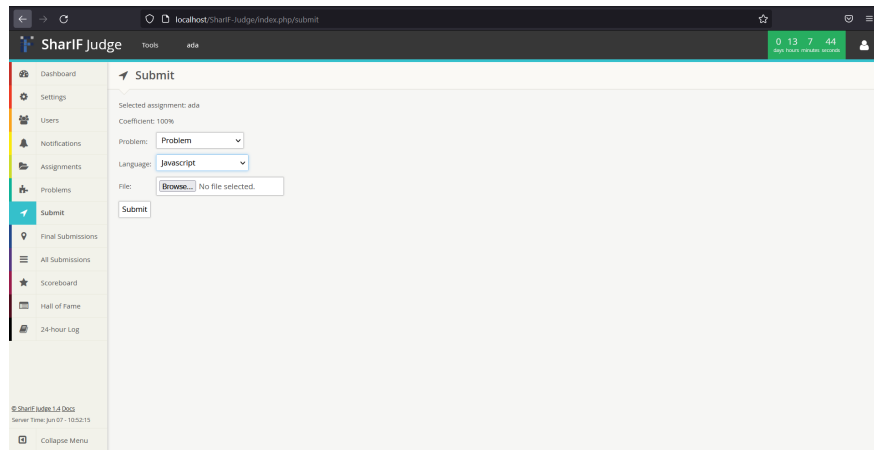
```

Name	Score	Time Limit (ms)	Memory Limit (KB)	Allowed Languages (2)	Diff	Command (2)	Argument (2)	Upload
1 Problem	100	500 1500 2000	50000	C,C++,Python 2,Python 3,j	diff	-bB		<input type="checkbox"/>

Add Assignment

Gambar 3: Halaman *Add Assignment*

Gambar 3 merupakan halaman pada *Add Assignment*. Halaman tersebut memiliki fungsi untuk Admin dan *Head Instructor* menambahkan *Assignment*. Pada bagian *Allowed Language*, akan dilakukan kostumisasi agar javascript menjadi salah satu bahasa yang diterima oleh SharIF-Judge.

Gambar 4: Halaman *Submit*

Gambar 4 merupakan halaman pada *Submit*. Halaman tersebut memiliki fungsi ketika pengguna ingin melakukan *upload* file untuk dievaluasi. Pada halaman *submit* pengguna terlebih dahulu memilih permasalahan yang mana yang akan dikumpulkan dan dengan bahasa pemrograman apa permasalahan tersebut dievaluasi. Setelah itu pengguna dapat *upload* file dan tekan tombol *submit*.

6. Melakukan pengujian dan eksperimen

Status : Ada sejak rencana kerja skripsi.

Hasil : Akan dilakukan pada Skripsi 2.

6 Pencapaian Rencana Kerja

Langkah-langkah kerja yang berhasil diselesaikan dalam Skripsi 1 ini adalah sebagai berikut:

1. Mempelajari struktur pada SharIF-Judge
2. Melakukan studi terhadap JavaScript
3. Mencoba membuat program sederhana javascript yang menerima input dari console
4. Menyelesaikan *bug* yang ada pada SharIF-Judge

Bandung, 21/06/2021

Edwin Pranajaya

Menyetujui,

Nama: Pascal Alfadian Nugroho
Pembimbing Tunggal