

DUKUNGAN BAHASA JAVASCRIPT PADA SHARIF JUDGE

EDWIN PRANAJAYA—2017730027

1 Deskripsi

Online judge merupakan sebuah sistem yang dirancang untuk melakukan evaluasi dari sumber kode algoritme yang dikirimkan oleh pengguna. Konsep dari *online judge* adalah dengan mengasumsi pengguna mengirimkan solusi berupa kode program, atau bahkan pengguna mengirimkan file yang dapat dijalankan dimana tahap selanjutnya kode ato file tersebut akan dievaluasi yang sering kali menggunakan infrastruktur berbasis *cloud*. Dalam perancangan *online judge* harus dipastikan bahwa *online judge* harus dapat menanggapi berbagai macam serangan seperti memaksakan waktu kompilasi yang tinggi, atau mengakses sumber daya yang dibatasi selama proses evaluasi berlangsung.

Sharif Judge merupakan salah satu *online judge* yang gratis untuk bahasa pemrograman C, C++, Java dan Python¹. Perangkat lunak ini diciptakan oleh Mohammad Javad Naderi pada tahun 2014 dan bersifat open source. Antarmuka Sharif Judge ditulis menggunakan bahasa pemrograman PHP (framework CodeIgniter) dan backend menggunakan BASH.

Sharif Judge biasa digunakan untuk mempermudah evaluasi kode program. dan salah satu universitas yang menggunakannya adalah Universitas Katolik Parahyangan (UNPAR) pada jurusan Informatika (IF). Namun Sharif Judge telah dimodifikasi dan berubah namanya menjadi SharIF-Judge. SharIF-Judge ini digunakan pada beberapa kuliah di IF, seperti Dasar Pemrograman dan Algoritma Struktur Data. Tujuan SharIF-Judge digunakan pada perkuliahan, adalah 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*. Didalamnya, 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 dapat dilihat *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 dari <https://insights.stackoverflow.com/survey/2021>, selama 9 tahun berturut-turut, JavaScript merupakan bahasa pemrograman yang paling sering digunakan. 64,96% orang 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.

¹<https://github.com/mjnaderi/Sharif-Judge>

2 Rumusan Masalah

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

Bagaimana cara agar SharIF Judge dapat memahami bahasa pemrograman JavaScript untuk dievaluasi.

3 Tujuan

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

Memodifikasi SharIF Judge agar dapat menerima soal JavaScript dan dapat mengevaluasi JavaScript.

4 Deskripsi Perangkat Lunak

Perangkat lunak akhir yang akan dibuat memiliki fitur minimal sebagai berikut:

- Perangkat lunak dapat menjalankan fitur yang sebelumnya dimiliki.
- Perangkat lunak dapat melakukan evaluasi terhadap JavaScript.

5 Detail Pengerjaan Skripsi

Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut :

1. Melakukan pembelajaran terhadap kode SharIF Judge
2. Melakukan studi literatur terhadap JavaScript
3. Melakukan studi literatur mengenai ECMAScript
4. Merancang fitur yang akan diimplementasikan
5. Implementasi fitur terhadap perangkat lunak
6. Menguji perangkat lunak ke mata kuliah selama 1 semester
7. Membuat dokumentasi perangkat lunak
8. Menulis dokumen skripsi

6 Rencana Kerja

Rincian capaian yang direncanakan di Skripsi 1 adalah sebagai berikut:

1. Mempelajari kode program SharIF Judge
2. Mempelajari bagaimana program tersebut dapat membaca bahasa pemrograman yang sudah ada.
3. Mencoba mengimplementasikan dari kode program yang sudah ada agar dapat membaca javascript
4. Menuliskan dokumen skripsi

Sedangkan yang akan diselesaikan di Skripsi 2 adalah sebagai berikut:

1. Melakukan testing SharIF-Judge yang di kostumisasi pada mata kuliah di jurusan Infromatika
2. Melakukan *bug fixing* jika ada *bug*
3. Menuliskan dokumen skripsi

Bandung, 22/02/2022

Edwin Pranajaya

Menyetujui,

Pascal Alfadian Nugroho, S.Kom, M.Comp
Pembimbing Tunggal