

DUKUNGAN BAHASA JAVASCRIPT PADA SHARIF JUDGE

EDWIN PRANAJAYA—2017730027

1 Deskripsi

Pada abad 21 ini, komputer dibutuhkan hampir pada setiap aspek kehidupan kita dan digunakan dengan berbagai cara, tidak terkecuali pendidikan[?]. Memang tidak selalu demikian, dan komputer saat ini digunakan dalam banyak cara. Komputer elektronik digital pertama kali muncul pada tahun 1940-an. Komputer telah berkembang dari mesin yang dirancang untuk melakukan perhitungan sampai menjadi pengolahan informasi dan komunikasi yang dibutuhkan di seluruh dunia. Kursus pada universitas dalam komputasi pada banyak negara dimulai di Departemen Matematika pada akhir 1940-an. Apakah Ilmu Komputer adalah cabang ilmu pengetahuan, cabang teknik, atau apakah itu sesuatu yang sama sekali unik telah lama dibahas, dan baru pada pertengahan 1960-an kursus universitas di Ilmu Komputer sebagai disiplin ilmu yang terpisah menjadi tersedia secara luas. Karena peningkatan komputerisasi departemen pemerintah dan kebutuhan komputasi yang berkembang dari bisnis, pada tahun 1970-an kursus universitas yang berfokus pada bisnis mulai muncul. Kursus tersebut berkembang menjadi yang sering kita sebut sebagai Sistem Informasi. Saat ini, program universitas menggunakan komputer dalam banyak hal. Baik dalam pengajaran maupun dalam penelitian. Pada tahun 1970 sekolah yang berfokus pada komputasi bermunculan di seluruh dunia. Sejumlah kecil sekolah di beberapa negara mulai menggunakan komputer kecil secara bergantian dan juga menggunakan *punch card* seperti yang ditunjukkan pada gambar ?? untuk mengajar pemrograman bersama dengan fasilitas di universitas lokal.

Pada tahun 1970, ketika Universitas A&M Texas menyelenggarakan pertama kali *Association for Computing Machinery (ACM) International Collegiate Programming Contest (ICPC)*, tidak ada yang bisa menebak bahwa, dalam beberapa tahun telah menjadi kontes pemrograman terbesar dan paling bergengsi di dunia [?]. Pada tahun 2015, lebih dari 40.000 mahasiswa dari hampir 3.000 universitas dari 102 negara berpartisipasi dalam fase regional kontes ini (*40th Annual World Finals ACM ICPC 2016*). Kontes tersebut berlangsung selama 5 jam. Peserta diharapkan untuk memecahkan 8 hingga 13 masalah algoritmik. Pemenangnya adalah tim yang pertama kali memecahkan masalah paling banyak. Komponen utama dari lingkungan kontes ini adalah sistem yang secara otomatis memverifikasi kebenaran solusi yang diajukan oleh peserta. Sistem tersebut Menilai kebenaran solusi yang diajukan berdasarkan hasil yang diperoleh dari pelaksanaannya pada tes yang telah ditentukan. Sistem ini juga memastikan bahwa solusi tidak melebihi batas pemanfaatan sumber daya (seperti waktu dan memori). Berdasarkan evaluasi yang dilakukan, peringkat online semua peserta dihitung dan disajikan secara real-time selama kontes berlangsung.

Pada forum International Olympiad in Informatics (IOI) terdapat diskusi mengenai pengalaman yang diperoleh selama olimpiade tersebut, termasuk persiapan masalah menarik dan perangkat lunak pendukungnya. Entri terpenting di antara perangkat lunak tersebut adalah sistem yang secara otomatis mengevaluasi solusi yang diajukan oleh peserta. Sistem seperti itu disebut sebagai *online judge*. Istilah *online judge* pertama kali diperkenalkan oleh Kurnia Lim, dan Cheang pada tahun 2001 sebagai platform online yang mendukung evaluasi kode sumber, biner, atau bahkan tekstual yang sepenuhnya otomatis dan real-time yang diajukan oleh peserta yang bersaing dalam tantangan tertentu. Akhirnya tidak hanya perangkat lunak itu saja yang penting, tetapi juga deskripsi masalah dan kualitas kasus uji yang disiapkan dan disimpan dalam sistem. Forišek mencatat bahwa penulis masalah, yang dievaluasi secara otomatis selama kompetisi seperti ACM ICPC dan IOI, harus memberikan perhatian khusus pada jenis masalah dan persiapan kasus uji. Dia menunjukkan bahwa beberapa jenis masalah, seperti pencarian substring, dapat dengan mudah diselesaikan dengan menggunakan algoritma heuristik. Selain itu, ia mempresentasikan beberapa masalah dari kompetisi ICPC dan IOI yang dapat diselesaikan dengan menggunakan algoritma yang umumnya salah dan masih menerima skor evaluasi yang sangat baik.

Selain digunakan untuk kompetisi, telah dikatakan pula dapat digunakan untuk membantu dalam akademis. Pemrograman adalah sebuah keterampilan yang diperoleh melalui latihan. Untuk membantu siswa, tugas pemrograman diberikan kepada siswa. Jumlah tugas pemrograman mengungkapkan seberapa besar penguasaan pada modul. Sebagian besar modul tingkat yang lebih tinggi akan mengasumsikan bahwa siswa sudah nyaman dengan berbagai bahasa pemrograman.

Ilmu Komputer tersebut pada abad 21 ini, sudah terdapat pada banyak universitas termasuk Universitas Katolik Parahyangan (UNPAR). Ilmu Komputer banyak mengajarkan bagaimana cara berpikir sistematis agar cara berpikir bisa lebih terstruktur. Pemrograman merupakan salah satu keterampilan utama yang diajarkan pada Fakultas Teknologi Informasi dan Sains di UNPAR. Hampir setiap modul ilmu komputer melibatkan beberapa bentuk pemrograman. Tak perlu dikatakan, beberapa modul ilmu komputer membutuhkan lebih banyak pemrograman daripada yang lain. Pada umumnya, penilaian dilakukan secara manual. dalam penilaian ini, dibutuhkan mahasiswa untuk mengumpulkan pekerjaan mereka dalam bentuk elektronik, cetakan, atau bahkan tulisan tangan. Dan penilaiannya pun dilakukan setelah selesai pengumpulan atau setelah batas waktu yang ditentukan. Penilaian dilakukan dengan melihat seluruh pekerjaan mahasiswa yang dikumpulkan dan dinilai 1 demi 1. Melakukan penilaian dengan hal tersebut mengambil tenaga yang cukup besar dan terkadang pemberian nilai tidak cukup hanya satu kali saja, kadang diperlukan untuk dilakukan pengecekan kembali apakah nilai yang diberikan sudah sesuai atau belum. Jika hanya 1 sampai 5 pelajar bukan menjadi permasalahan, namun jika ada pelajar lebih dari 50 atau bahkan 100 akan sulit untuk diperiksa bersamaan dengan kondisi dimana penilaian dilakukan terkadang dilakukan secara subyektif. Cara ini dilakukan karena penilaian dilakukan berdasarkan cara, metode, atau gaya dari penilai tersebut. Pada

umumnya, pelajar menyelesaikan persoalan tersebut memiliki caranya masing-masing, dan bahkan menggunakan metode yang berbeda dengan yang sudah diajarkan oleh pengajar. Namun terkadang penilai dapat memberikan nilai yang kecil karena tidak sesuai meskipun memiliki hasil yang benar. Selain itu, penilaian dapat dipengaruhi oleh keadaan emosional dari penilai tersebut yang dapat berubah secara signifikan dari waktu ke waktu. Sebagai tambahan terkadang penilai memiliki tujuannya masing-masing. Sebagai contoh mereka ingin menyelesaikan penilaian tersebut secepat mungkin, sehingga dapat terjadi kesalahan.

Untuk menanggulangi hal tersebut, digunakan *online judge*[?]. Jika *online judge* dapat digunakan untuk mempercepat evaluasi pada kompetisi program, maka seharusnya *online judge* juga dapat membantu pada proses penilaian nilai pelajar. Sebagai contohnya adalah penilaian yang dilakukan secara otomatis. Sebuah program akan mengambil program lain dan melaporkan penilaiannya dimana penilaiannya tersebut diterjemahkan ke dalam nilai numerik. *Online judge* mengevaluasi kode dari sumber algoritme yang dikirimkan oleh pelajar atau pengguna, yang selanjutnya dikompilasi dan diuji dengan tes kasus yang sudah disiapkan. Dengan menggunakan penilaian secara otomatis, semua permasalahan dari penilaian secara manual terselesaikan.

Aspek penting yang perlu diperhatikan saat pengembangan *online judge* adalah ketepatan pengukuran waktu eksekusi[?]. Batas waktu untuk eksekusi kasus uji tunggal sering diukur dalam *millisecond* dan dengan demikian metode analisis kinerja yang digunakan selama evaluasi harus cukup sensitif dan deterministik untuk secara tepat membedakan fraksi waktu yang kecil dan memastikan pengukuran yang dapat direproduksi dari eksekusi berurutan dari kode yang sama untuk kasus uji tertentu. Tetapi secara umum, tujuan sistem penilaian online adalah evaluasi algoritme berbasis *cloud* yang aman, andal, dan berkelanjutan yang dikirimkan oleh pengguna yang didistribusikan di seluruh dunia.

Salah satu contoh *Online judge* adalah Sharif Judge. Sharif Judge adalah online judge gratis untuk bahasa pemrograman C, C++, Java dan Python ¹. 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. Salah satu universitas yang menggunakan Sharif Judge adalah Universitas Katolik Parahyangan. Sharif Judge dapat membaca berbagai bahasa pemrograman, namun JavaScript bukan salah satu yang dapat dibacanya.

JavaScript adalah bahasa yang lebih banyak kelebihan daripada kekurangannya[?]. Reputasi dari JavaScript berkembang dari yang belum ada sampai sangat dibutuhkan di seluruh dunia hanya dalam waktu yang sempit. JavaScript dibutuhkan karena merupakan bahasa *browser web*. Hubungannya dengan browser web menjadikannya salah satu bahasa pemrograman yang paling populer di dunia.

Karena popularitas JavaScript, sangat disayangkan jika Sharif Judge tidak dapat membaca bahasa tersebut. Karena JavaScript merupakan bahasa yang akan terus terpakai. 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 Sharif Judge. Untuk mempermudah kostumisasi, pengerjaan akan dilakukan dengan menggunakan OS Linux. Namun dikarenakan OS yang digunakan adalah Windows, akan digunakan docker agar dapat mengerjakan dengan OS Linux pada Windows. Sehingga, ketika pengajar membuat soal tentang JavaScript, pengajar dapat membuat soal dan tes kasus terhadap JavaScript. Para pelajar pun dapat dengan mudah mengetahui apakah kode yang mereka buat tentang JavaScript sudah sesuai atau belum. Dengan JavaScript dapat dibaca oleh Sharif Judge, akan dengan mudah dan cepat evaluasi terhadap pekerjaan para pelajar.

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

2 Rumusan Masalah

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

1. Bagaimana cara membuat soal dan tes kasus yang benar pada Sharif Judge.
2. Bagaimana cara agar Sharif Judge dapat membaca JavaScript.
3. Bagaimana cara agar Sharif Judge dapat memahami bahasa pemrograman JavaScript untuk dievaluasi.

3 Tujuan

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

1. Mempelajari cara kerja dan merancang soal pada Sharif Judge.
2. Mempelajari kode program Sharif Judge agar dapat mengintegrasikan JavaScript.
3. Memodifikasi Sharif Judge agar dapat memasukan soal JavaScript dan dapat mengevaluasi JavaScript.

4 Deskripsi Perangkat Lunak

Tuliskan deksripsi dari perangkat lunak yang akan anda hasilkan. Apa saja fitur yang disediakan oleh PL tersebut dan apa saja kemampuan dari PL tersebut. Perhatikan contoh di bawah ini:

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

- Pengguna dapat melihat denah Musem Geologi Bandung dalam bidang dua dimensi. Sedangkan pengunjung direpresentasikan menggunakan lingkaran-lingkaran kecil (tidak menggunakan gambar manusia yang diambil dari atas)
- Pengguna dapat memunculkan atau menghilangkan gambar *flow tiles* pada denah museum.
- Pengguna dapat mengatur jalannya simulasi: memulai(start) simulasi, menunda(pause) simulasi, melanjutkan(continue) simulasi, maupun menghentikan(stop) simulasi
- Pengguna dapat mengatur banyaknya pengunjung di dalam museum, baik melalui perubahan frekuensi kedatangan pengunjung maupun menambahkan dan menghapus pengunjung satu-persatu secara manual.
- Posisi kamera dapat diubah (pergerakan di bidang tiga dimensi) sehingga pengguna dapat melihat simulasi di museum dari berbagai arah.
- Posisi kamera dapat diubah untuk mengikuti perjalanan seorang pengunjung di dalam
- Pengguna dapat memilih apakah akan menggunakan teknik *flow tiles* atau tidak pada saat simulasi berlangsung
- Jenis *flow tiles* yang digunakan dapat diubah-ubah pada saat simulasi sedang berlangsung

5 Detail Pengerjaan Skripsi

Tuliskan bagian-bagian pengerjaan skripsi secara detail. Bagian pekerjaan tersebut mencakup awal hingga akhir skripsi, termasuk di dalamnya pengerjaan dokumentasi skripsi, pengujian, survei, dll.

Bagian-bagian pekerjaan skripsi ini adalah sebagai berikut :

1. Melakukan survei ke Museum Geologi Bandung untuk mendapatkan denah serta mengetahui perilaku pengunjung museum secara umum (arah perjalanan, kecepatan, lama melihat objek, dll)
2. Melakukan analisis pada hasil survei terhadap pergerakan pengunjung di museum dan membuat rancangan denah di komputer yang dilengkapi dengan penghalang dan objek di museum.
3. Melakukan studi literatur mengenai sifat kolektif suatu kerumunan, teknik *social force model* dan teknik *flow tiles*
4. Mempelajari bahasa pemrograman C++ dan cara menggunakan framework OpenSteer
5. Merancang pergerakan kerumunan di dalam museum menggunakan teknik *social force model* dan *flow tiles* serta menggunakan teknik lainnya seperti konsep pathway dan waypoints. Selain itu, dirancang pula adanya waktu tunggu (pada saat pengunjung melihat objek di museum) dan cara pembuatan jalur bagi setiap individu pengunjung
6. Melakukan analisa dan merancang struktur data yang cocok untuk menyimpan penghalang (obstacle)
7. Mengimplementasikan keseluruhan algoritma dan struktur data yang dirancang, dengan menggunakan framework OpenSteer
8. Melakukan pengujian (dan eksperimen) yang melibatkan responde untuk menilai hasil simulasi secara kualitatif
9. Menulis dokumen skripsi

6 Rencana Kerja

Rincian capaian yang direncanakan di Skripsi 1 adalah sebagai berikut:

1. Melakukan studi literatur terhadap kode program yang ada pada SharIF Judge bagian Assignment
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

Sedangkan yang akan diselesaikan di Skripsi 2 adalah sebagai berikut:

- 1.
- 2.
- 3.

Edwin Pranajaya

Menyetujui,

Nama: _____
Pembimbing Tunggal