2. LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada bab ini akan dijelaskan mengenai teori-teori yang akan digunakan dalam penulisan skripsi dan pembuatan aplikasi.

2.1.1 Online Judge

Online Judge adalah sebuah sistem untuk melakukan evaluasi kebenaran pada suatu kode program pada tugas atau ujian pemrograman (Kurnia et al., 2001). Sebuah sistem *online judge* umumnya digunakan dalam kontes pemrograman, edukasi, dan rekrutmen karyawan. Sedangkan, prosedur evaluasi pada sebuah sistem online judge umumnya terdiri dari 3 tahapan, yaitu: *submission, assessment,* dan *scoring* (Yibo, et al., 2019).

Menurut penelitian dari Kurnia et al. (2001), sistem *online judge* dibuat untuk mengatasi permasalahan pada sistem penilaian secara manual. Beberapa masalah tersebut antara lain kesulitan sistem manual dalam menghadapi jumlah peserta yang semakin bertambah banyak dibandingkan jumlah *grader* atau pengoreksi yang ada. Selain itu, *grader* umumnya akan mengoreksi setelah *deadline*, dibandingkan dengan *online judge* yang akan mengoreksi secara langsung sehingga peserta dapat melihat hasilnya dan menentukan apakah ingin mencoba lagi untuk mendapatkan nilai yang lebih baik (Kurnia et al., 2001).

Penelitian Watanobe et al. (2022) membahas mengenai *requirement,* arsitektur, dan pengalaman dari pembuatan sebuah *online judge system*, disebutkan bahwa mempunyai *functional requirements* dan *non-functional requirements.* Berikut ini adalah rangkuman dari beberapa *functional requirement* dan *non-functional requirements* menurut penelitian tersebut.

- *Functional Requirements*

  Sebuah *online judge system* dapat menerima sebuah *submission request* yang terdiri dari bahasa pemrograman *lang*, dari sebuah *user u,* untuk menyelesaikan soal *p*. Terdapat 2 tipe submisi, satu adalah *custom*, di mana *user* dapat memberikan *input* dan *output* melalui

submisi, dan yang lain adalah *actual*, yang mengevaluasi program menggunakan *input, output*, dan validator yang sudah didefinisikan oleh soal. Terdapat pula 4 cara mengevaluasi kode program, menggunakan perbedaan dalam segi teks *output* yang dihasilkan oleh eksekusi kode program dengan *input* yang ditentukan, kedua mengevaluasi sebuah *output* yang berupa angka *real* dengan toleransi *error,* selain itu dapat menggunakan program yang memvalidasi hasil output secara eksternal, dan terakhir menggunakan sebuah *reactive program* yang berinteraksi dengan program *user*. Hasil dari proses evaluasi di atas adalah pesan apabila *source code* tidak dapat dikompilasi, yaitu *Compile Error* dan apabila program mengalami error ketika dijalankan dapat mengembalikan pesan *Runtime Error*. Selain itu, ada error yang dikeluarkan oleh *judge* atas persyaratan tambahan yang sudah diberikan dari pembuat soal, seperti *Time Limit Exceeded* apabila eksekusi program melebihi waktu tertentu dan *Memory Limit Exceeded* apabila eksekusi program melebihi penggunaan memori tertentu. Hasil pengecekan jawaban dapat mengeluarkan *Wrong Answer* apabila kesalahan pada hasil *output* ataupun *Accepted* apabila hasil *output* dapat diterima.

- *Non-Functional Requirements*

  Menurut Watanobe et al. (2022), terdapat beberapa *non-functional requirements* yang perlu dimiliki oleh sebuah *online judge system*, yaitu:

  - *Immediacy,* kemampuan untuk memberikan *feedback* secara *real-time.*
  - *Security*, kemampuan untuk mengisolasi lingkungan di mana kode dijalankan dengan sistem *host*.
  - *Consistency*, kemampuan untuk menjalankan kode program dalam suatu lingkungan yang konsisten,
  - *Newness*, kemampuan untuk mendukung versi terbaru dari suatu bahasa pemrograman.
  - *Robustness*, kapabilitas untuk bisa tersedia dan berjalan selama 24 jam sehari, tanpa perlu kehadiran *administrator* yang mengelola secara manual.
  - *Correctness*, kemampuan sistem untuk bisa memberikan evaluasi yang adil, tepat, dan akurat. Jawaban yang benar tidak boleh ditetapkan sebagai jawaban yang salah dan sebaliknya.
    - *Sustainability*, kemampuan untuk menyimpan data mengenai submisi, *log,* dan hasil evaluasi secara terus-menerus.

- *Portability,* kemampuan untuk bisa diinstall dan dijalankan dengan mudah untuk menyediakan layanan jika diperlukan pada sistem atau peranti keras yang berbeda.
- *Transparency,* adanya transparansi dalam pemrosesan status dan laporan hasil evaluasi.
- *Scalability,* dapat mudah *scale-up* bergantung pada jumlah *user* yang menggunakan.
- *Availability,* dapat terus berjalan dan siap menerima submisi pada waktu apapun.
- *Economic,* efisien dalam menggunakan *resource* yang ada.

Beberapa contoh sistem online judge dalam bidang edukasi adalah *Codechef, Codecademy, URI Online Judge*, dan lain-lain (Wasik, et al 2019).

2.1.2 Judge0

Judge0 adalah sebuah *code execution system* yang *open-source* dan *scalable* (Judge0 CE - API Docs, n.d.) dan akses disediakan dalam bentuk web API.

Beberapa fitur yang disediakan oleh Judge0 antara lain:

- Kemudahan dalam menginstal
- Dokumentasi API yang lengkap.
- Arsitektur yang *scalable.*
- Proses kompilasi dan eksekusi program secara *sandboxed*.
- Dukungan untuk lebih dari 60 bahasa pemrograman.
- Kompilasi program dengan lebih dari satu file.
- Dukungan untuk memberikan *compiler options*, *arguments*, dan batasan memori dan waktu.
- Hasil eksekusi program yang detil.

Secara proses *deployment*, Judge0 dapat digunakan melalui *RapidAPI* yang sudah dkelola secara *scaling* dan *maintenance*, namun berbayar. Selain melalui hal tersebut, juga dapat di *deploy* secara *on-premise* dengan Docker Image yang sudah disediakan.

2.2. Tinjauan Studi

Berikut merupakan penelitian yang telah dilakukan sebelumnya:

2.2.1 Online Judge System and Its Applications in C Language Teaching (2016)

- Penelitian dilakukan terhadap dampak dari penggunaan *Online Judge* untuk kegiatan pembelajaran bahasa pemrograman C pada *Hangzhou Dianzi University*, Tiongkok.
- Beberapa hasil yang didapat antara lain:
    - Perbedaan antara mahasiswa yang berperforma baik dengan mahasiswa yang berperforma buruk semakin terlihat dengan penggunaan *online judge.* Hal ini terjadi karena pemberian soal dilakukan secara acak sehingga mengurangi adanya kecurangan dalam pengerjaan soal.
    - Tingkat ketertarikan mahasiswa, persentase mahasiswa yang mempunyai kemampuan lebih dari rata-rata dan finalis pada kompetisi pemrograman mengalami peningkatan sejak menggunakan sistem *online judge*. Ketertarikan terhadap pemrograman meningkat dari 40% menjadi 73%, persentase mahasiswa dengan kemampuan luar biasa meningkat dari 10% menjadi 34%, dan finalis di kompetisi nasional meningkat dari 5% menjadi 20%.

2.2.2 An Experimental Online Judge System Based on Docker Container for Learning and Teaching Assistance (Yibo et al., 2019)

- Penelitian ini mengangkat masalah dari kekurangan beberapa *online judge* untuk edukasi yang ada. Hal itu berupa kurangnya implementasi manajemen kelas, mahasiswa, *course*, pengajar, dan berbagai hak akses, soal-soal yang sistematis sesuai dengan course, fitur anti plagiarisme, dan visualisasi perolehan mahasiswa secara statistik.
- Sistem *online judge* ini digunakan di *Nanyang Institute of Technology*, Tiongkok pada tahun ajaran 2018 hingga 2019 dan digunakan oleh 1460 mahasiswa.
- Sistem *online judge* yang dibuat terdiri dari *web server* dan *container resource pool* yang melakukan manajemen *Docker Container* yang digunakan untuk menjalankan kode.

2.2.3 OJPOT: online judge & practice oriented teaching idea in programming courses. (Wang et al., 2015)

- Penelitian ini berupaya meningkatkan antusiasme mahasiswa dalam mempelajari mata kuliah *Programming Foundation* yang diketahui memiliki materi yang banyak dan membosankan bagi mahasiswa. Penelitian ini pertama dilakukan di Information School Zhejiang University of Finance and Economy (ZUFE) pada semester pertama periode 2010-2011 dan kedua di College of Computer Science, Chongqing University (CQU) pada semester pertama periode 2011–2012.
- Solusi yang diajukan adalah melakukan proses belajar mengajar dengan metode OJPOT (*Online Judge & Practice Oriented Teaching*), yang terdiri dari hal berikut:
    - Adanya *contest* sebagai metode pembelajaran.
    - Penggunaan *online judge* dan latihan soal yang diberikan setelah materi kelas diberikan.
    - Adanya *course project* yang diberikan secara berkelompok.
- Hasil yang didapatkan setelah observasi yaitu kelas yang menggunakan metode ini memiliki antusiasme yang lebih tinggi dibandingkan kelompok lain.

2.2.4 DMOJ (Home - DMOJ: Modern Online Judge, n.d.)

- Website dan judge dibuat menggunakan Python dan framework Django. Eksekutor program diimplementasikan dengan memblokir beberapa system call untuk keamanan.
- Beberapa fitur yang dimiliki oleh aplikasi ini antara lain (DMOJ: Modern Online Judge, 2022):
    - Mendukung lebih dari 60 bahasa pemrograman, kustomisasi hecker dan validator, serta dapat membatasi penggunaan resource pada proses eksekusi.
    - Mendukung konfigurasi dalam sistem kontes pemrograman, deteksi plagiasi menggunakan Stanford MOSS, serta dapat membatasi kontes untuk pengguna tertentu saja.
    - Adanya dukungan untuk menggunakan LaTeX dalam penulisan soal, serta melakukan export menjadi PDF untuk memudahkan proses distribusi.
- Kekurangan yang dimiliki yaitu fitur terfokus untuk kontes pemrograman, sehingga belum ada pendataan yang sesuai dengan kebutuhan self-paced learning Studio Pemrograman Universitas Kristen Petra.

2.2.5 DOMJudge (DOMjudge - Programming Contest Jury System, n.d.)

- Terdiri dari 2 komponen, yaitu web dan judge host. Web bertugas menyajikan website, sedangkan judge host bertugas menjalankan kode.
- Fitur-fitur terdiri dari:
    - Sistem judge otomatis dengan jumlah host yang scalable
    - Mempunyai web interface yang simple dan portabel
    - Sistem yang modular untuk memasukkan validator dan *compiler* bahasa pemrograman
    - Memiliki informasi detail bagi *jury* seperti *submissions*, *judging*, dan *diffs* serta opsi tambahan seperti *rejudge*, *clarification*, serta *resubmit* untuk juri.
- Kekurangan sistem yaitu berfokus untuk pengujian kontes pemrograman saja serta belum ada pendataan yang sesuai dengan kebutuhan Studio Pemrograman di Universitas Kristen Petra.