

# Rancang Bangun Modul Penjurian *Daring* *Moodle* Menggunakan *Pysandbox*

Kharisma Muchammad, Dwi Sunaryono dan Wahyu Suadi

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

*E-mail:* dwi@if.its.ac.id

**Abstrak**—Dalam pengajaran ilmu komputer, pemrograman merupakan mata kuliah yang harus diterima oleh setiap mahasiswa. Dalam prakteknya, penilaian tugas pemrograman acap kali tidak dilakukan secara menyeluruh karena pemeriksaan secara menyeluruh memakan waktu yang lama jika dilakukan secara manual. Salah satu solusi untuk masalah tersebut adalah dengan menggunakan penjurian daring. Dalam pembuatan penjurian daring sendiri terdapat beberapa masalah, salah satunya adalah memastikan kode program tidak merusak komputer. Makalah ini mengajukan suatu modul penjurian daring untuk Moodle menggunakan Pysandbox. Modul yang dihasilkan dapat memeriksa soal dengan kasus uji tunggal ataupun jamak, kasus uji dengan toleransi ataupun tidak dan pengamanan proses uji program hasil kompilasi dalam lingkungan kotak pasir (*sandbox*).

**Kata Kunci**—Moodle, penjurian daring, *sandbox*.

## I. PENDAHULUAN

SALAH satu tujuan dari pengajaran ilmu komputer, informatika, sistem informasi dan teknik komputer adalah untuk memberikan kemampuan pemrograman pada peserta didik [1]. Untuk mencapai tujuan tersebut, pengajar biasanya memberikan tugas pemrograman dan memeriksanya. Hanya saja dengan bentuk penilaian secara manual, terdapat beberapa masalah seperti:

- 1) Menilai tingkat kebenaran dan efektivitas  
Penilaian manual tidak dapat melihat tingkat kebenaran dan efektivitas program yang dihasilkan dengan cepat.
- 2) Model pendekatan yang banyak untuk permasalahan yang sama  
Fokus dari penilaian manual terfokus pada algoritma program dan bukan pada keluaran program. Hal ini menyebabkan penilai akan kesulitan menilai model solusi di luar solusi yang dia pahami.
- 3) Penekanan pada estetika  
Untuk memudahkan proses koreksi program, penilai biasanya menekankan pada faktor estetika seperti komentar, indentasi, modularitas dan nama variabel.
- 4) Faktor manusia  
Penilai manusia cenderung inkonsisten dalam menilai. Selalu terdapat faktor suasana hati (*mood*) dalam menilai.

Salah satu solusi dari masalah di atas adalah mengotomasi proses penilaian dengan menggunakan penjurian daring.

Penjurian daring merupakan sistem yang bertujuan menguji program dalam kontes pemrograman. Sistem ini secara umum mampu melakukan kompilasi, eksekusi kode, dan uji coba program. Program yang diuji akan menghadapi beberapa batasan seperti batasan waktu maupun memori [2].

Moodle adalah salah satu aplikasi *Course Management System* (CMS) berbasis PHP dan bersifat sumber terbuka. Moodle menggunakan arsitektur yang memungkinkan pengguna menambahkan fungsionalitas melalui pengaya tanpa mempengaruhi fungsionalitas Moodle yang lain.

Python merupakan bahasa pemrograman yang bertujuan umum, terinterpretasi dan tingkat tinggi. Python didesain oleh Guido von Rossum pada 1989 dan sampai makalah ini ditulis telah sampai versi 3. Namun karena masalah *backward compatibility* Python 2 masih dikembangkan.

Pysandbox merupakan salah satu pustaka untuk melakukan pengotakpasiran (*sandboxing*) yaitu pengisolasian program dalam suatu lingkungan dimana program tidak dapat mengakses sumber daya lebih dari yang diijinkan [3]. Pengotakpasiran dipakai untuk menguji program yang tidak dipercaya agar tidak merusak sistem [5].

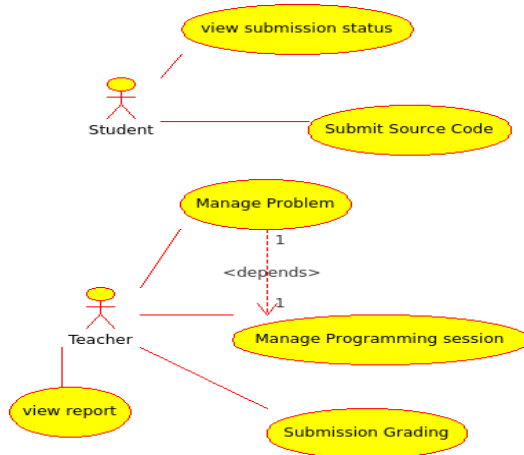
*Daemon* adalah program komputer yang berjalan di belakang layar dan berfungsi memberikan fungsionalitas tertentu pada pengguna tanpa ada interaksi langsung dengan pengguna [4]. Istilah *daemon* umum dipakai dalam sistem operasi mirip UNIX sedangkan dalam sistem operasi Windows lebih dikenal istilah *services*.

## II. METODOLOGI

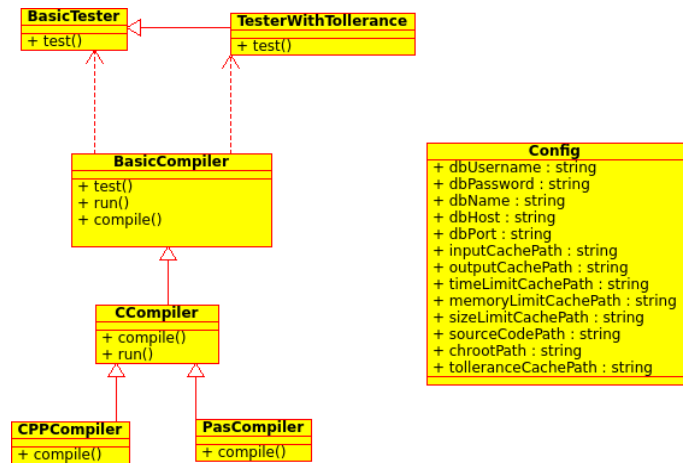
### A. Analisis Kebutuhan

Modul yang dihasilkan mampu mengolah soal bertipe *batch* dengan kasus uji tunggal ataupun jamak, dengan toleransi numerik atau tidak. Modul juga mampu menjalankan kode hasil kompilasi dalam suatu lingkungan kotak pasir.

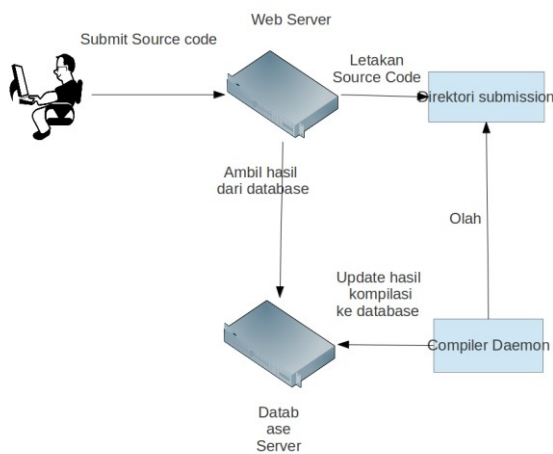
### B. Diagram Kasus Guna



Gambar 1. Diagram Kasus Guna



Gambar 3. Diagram Kelas *Daemon* Kompilasi

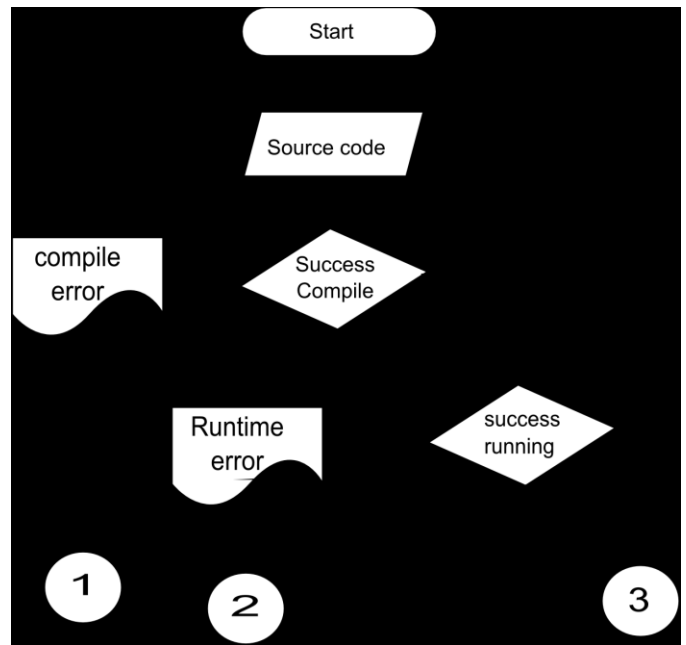


Gambar 2. Arsitektur Perangkat Lunak

Gambar 1 menunjukkan diagram kasus guna sistem. Secara umum sistem menerima masukan berupa deskripsi sesi dan jawaban. Deskripsi sesi berisi sejumlah deskripsi soal beserta batasannya. Jawaban berupa status jawaban dan nilai dari jawaban tersebut.

C. Arsitektur Perangkat Lunak

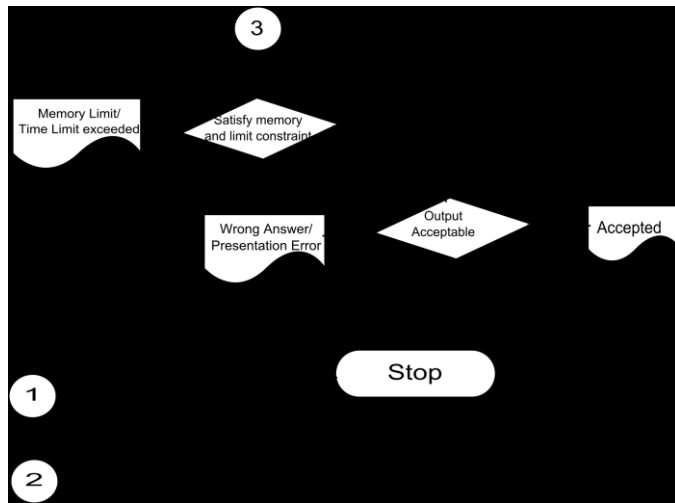
Gambar 2 menggambarkan bagaimana modul bekerja. Pertama-tama dosen membuat sesi pemrograman beserta soal, kasus uji, dan batasan uji. Setelah sesi dibuka, mahasiswa mengirim jawaban ke sistem melalui antarmuka *web*. Kemudian sistem meletakkan berkas jawaban pada direktori submisi. *Daemon* kompilasi akan secara berkala memeriksa direktori submisi dan mengolah (mengkompilasi dan menguji jawaban) berkas yang berada pada direktori tersebut. Hasil uji dari *daemon* kompilasi akan dimasukkan pada *server* basis data. Dosen maupun mahasiwa dapat melihat laporan melalui antarmuka *web* dari modul.



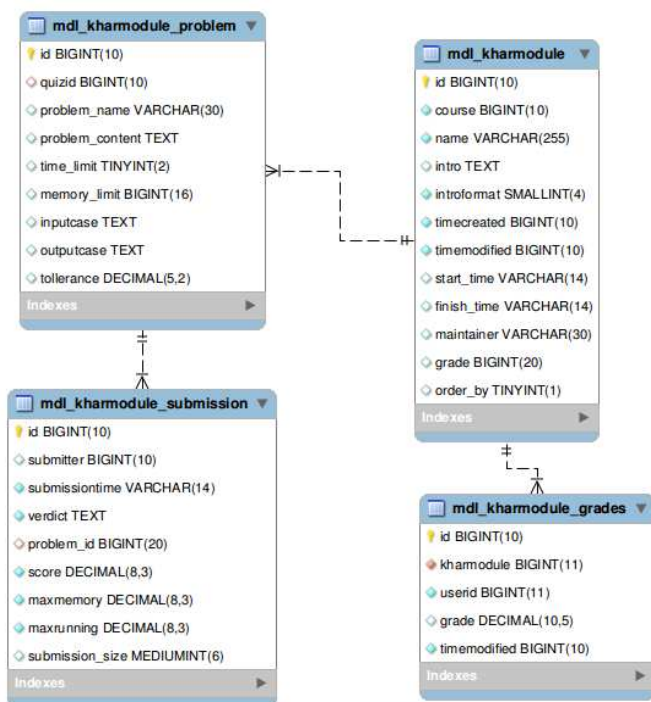
Gambar 4. Diagram Alur Kerja *Daemon* (bagian 1)

D. Diagram Kelas

Diagram kelas *daemon* kompilasi dapat dilihat pada Gambar 3. Pada gambar tersebut dapat dilihat bahwa terdapat kelas kompilator abstrak bernama *BasicCompiler*. Kelas ini akan di-*extends* ke kelas kompilator spesifik untuk bahasa pemrograman yang didukung yaitu CPP, Pascal dan C. Dapat dilihat pula bahwa kelas *BasicCompiler* memiliki atribut kelas penguji yaitu *BasicTester* dan *TesterWithTolerance*.



Gambar 5. Diagram Alur Kerja *Daemon* (bagian 2)



Gambar 6. Diagram Data Fisik

E. Diagram Data Fisik

Diagram data fisik dapat dilihat pada Gambar 6. Terdapat 4 entitas yang disimpan di basis data. *Kharmodule\_problem* menyimpan entitas aktivitas. *Kharmodule\_grade* berfungsi menyimpan nilai bagi buku nilai Moodle. *Kharmodule\_problem* berfungsi menyimpan deskripsi soal dan *Kharmodule\_submission* berfungsi menyimpan jawaban mahasiswa.

III. IMPLEMENTASI

Implementasi modul dilakukan pada *server* Linux Ubuntu 12.04 dengan *kernel* Linux 3.2, Apache 2.2, MySQL 5.5, Python 2.7 dan Moodle 2.3.

Tabel 1. Hasil Pengujian Sistem

Uji Coba	Status
Memasukkan soal	
Kasus uji tunggal	Sukses
Kasus uji jamak	Sukses
Hampiran numerik	Sukses
Menguji Soal	
Kasus uji tunggal	Sukses
Kasus uji jamak	Sukses
Hampiran Numerik	Sukses
Keamanan	Sukses

*Daemon* kompilasi memerlukan beberapa paket perangkat lunak tambahan agar dapat bekerja. Perangkat lunak tersebut antara lain GCC, G++, FPC, fromdos, python-mysqldb, dan libsandbox. Alur kerja *daemon* dalam menentukan putusan suatu submisi dapat dilihat pada Gambar 4 dan Gambar 5. Dalam gambar tersebut dapat dilihat bahwa *daemon* akan melakukan beberapa pemeriksaan sebelum memasukkan hasil ke dalam basis data.

IV. UJI COBA

Uji coba dilakukan melalui komputer klien dengan spesifikasi Ubuntu 12.04 *kernel* Linux 3.2 dengan peramban *web* Mozilla Firefox 17.0.1. Hasil uji coba dapat dilihat pada Tabel 1. Uji coba dilakukan untuk menguji sistem apakah dapat melakukan manajemen soal dan pengolahan submisi dengan baik atau tidak. Pengujian nonfungsionalitas meliputi uji coba keamanan untuk memastikan bahwa program tidak dapat mengakses sumber daya selain yang diperbolehkan.

V. KESIMPULAN

- 1) Kotak pasir dapat dipakai untuk menguji program yang tidak dipercaya sehingga program yang tidak dipercaya tersebut tidak dapat merusak komputer yang menguji.
- 2) *Daemon* dapat menguji beberapa jenis soal *batch* seperti kasus uji tunggal, kasus uji jamak, dan kasus uji hampiran numerik.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Direktorat Jenderal Pendidikan Tinggi yang memaksa penulis harus membuat makalah ini.

DAFTAR PUSTAKA

- [1] B.Cheang et al, "On automated grading of programming assignments in an academic institution", *Computer & Education* 41, 121-131, 2003.
- [2] S Skiena, M Revilla, *Programming Challenges*, New York:Springer Science, 2003.
- [3] L Yu. "Sandbox Libraries (libsandbox & pysandbox)", 2012. [online] Available: <http://openjudge.net/~liuyu/Project/LibSandbox>. [Diakses pada tanggal 26 September 2012].
- [4] E Raymond. "Daemon", 2004. [online] Available: <http://catb.org/~esr/jargon/html/D/daemon.html>. [Diakses pada tanggal 4 Maret 2013].
- [5] I Goldberg et al, "A Secure Environment for Untrusted Helper Application", *Proceeding of the Sixth USENIX UNIX Security Symposium*, 1996.