

# Pengembangan Fitur Penilaian dan Perangkingan pada Automatic Programming Assessment Tools

RINA HARIMURTI<sup>1</sup>, ANITA QOIRIAH<sup>2</sup>, ANDI IWAN NH<sup>3</sup>, ASMUNIN<sup>4</sup>

<sup>1,2,3,4</sup> Jurusan Teknik Informatika FT Unesa Surabaya

<sup>1</sup>[rinaharimurti@unesa.ac.id](mailto:rinaharimurti@unesa.ac.id)

<sup>2</sup>[anitaqoiriah@unesa.ac.id](mailto:anitaqoiriah@unesa.ac.id)

<sup>3</sup>[andy134ks@unesa.ac.id](mailto:andy134ks@unesa.ac.id)

<sup>4</sup>[asmunin@unesa.ac.id](mailto:asmunin@unesa.ac.id)

**Abstrak**— Praktikum Pemrograman Dasar merupakan wadah yang efektif dan telah terbukti untuk meningkatkan kemampuan mahasiswa dalam pemrograman. Dalam kegiatan ini, mahasiswa diberikan permasalahan, kemudian mahasiswa menyusun algoritma dan mencari solusi dan menuliskan kode program. Setelah program selesai ditulis, setiap mahasiswa harus melakukan asistensi dan demo ke dosen pengampu/asisten praktikum. Dalam melakukan demo ke dosen pengampu/asisten membutuhkan waktu yang tidak sedikit, dan tidak hanya sekali. Aplikasi Automatic Programming Assessment digunakan untuk melakukan evaluasi secara mendalam terhadap source code yang diunggah oleh praktikan. Aplikasi ini akan melakukan pengecekan jenis bahasa pemrograman yang digunakan, ukuran file, error syntax, lama eksekusi, runtime error, time limit, wrong answer, dan compilation error. Jika hasil evaluasi tidak ada masalah, maka program dianggap benar (Accepted). Praktik/mahasiswa dapat secara live menulis kode program kemudian dikirim, atau langsung mengirim file kode program. Terhadap file yang terkirim, aplikasi akan melakukan evaluasi dan akan menampilkan hasil evaluasi. Hasil pengujian yang dilakukan dapat disimpulkan bahwa fitur penilaian otomatis terhadap kode program yang dikirim oleh mahasiswa dapat berjalan dengan baik. Hal ini dibuktikan dengan kesamaan hasil penilaian antara penilaian manual dan penilaian melalui aplikasi. Fitur rangking juga dapat menampilkan urutan mahasiswa yang telah memperoleh nilai dengan baik berdasarkan nilai (point) dan persentase AC (Accepted Answer)

**Kata Kunci**— Automatic Assessment, Grading System, Autograding, Online Judge.

## I. PENDAHULUAN

Pemrograman komputer merupakan skill yang harus dikuasai oleh mahasiswa ilmu komputer. Fakta yang ada, banyak mahasiswa yang tidak memiliki kemampuan pemrograman yang baik. Metode terbaik dalam meningkatkan kemampuan pemrograman adalah melalui praktik langsung menulis program untuk memecahkan berbagai permasalahan mulai yang sederhana sampai kompleks. Dengan banyak melakukan praktik, pengalaman menjadi lebih banyak dan skill pemrograman juga menjadi meningkat.

Praktikum pemrograman dasar adalah bentuk implementasi dari peningkatan skill pemrograman mahasiswa dengan banyak melakukan praktek menulis kode program secara langsung. Pada kegiatan ini, mahasiswa diberi permasalahan dalam bentuk modul praktikum yang terbagi menjadi 8 modul, dan 1 final project. Setiap modul praktikum berisi soal

praktikum, baik yang bersifat individu maupun kelompok. Untuk soal yang bersifat individu, dikerjakan secara live pada saat praktikum, dan hasilnya langsung didemokan ke dosen/asisten, sedangkan untuk soal kelompok dikerjakan bersama-sama. Setiap akhir selesai praktikum, mahasiswa membuat laporan praktikum dan dikumpulkan. Final project dikerjakan dipertemuan akhir kegiatan praktikum dengan kasus permasalahan yang lebih besar dan kompleks.

Selama kegiatan praktikum, setiap mahasiswa harus mendemokan hasilnya ke dosen/asisten. Demo ke dosen/asisten tidak hanya dilakukan sekali, akan tetapi dilakukan berkali-kali sampai dosen/asisten menerima hasil praktikum mahasiswa.

Berdasarkan permasalahan tersebut diatas, maka perlu tersedia suatu aplikasi/tools yang dapat membantu meringankan tugas dosen/asisten terkait kegiatan praktikum pemrograman dasar. Tools ini diharapkan dapat menggantikan sebagian besar tugas dosen/asisten dengan tidak menghilangkan karakter kegiatan praktikum itu sendiri. Tools ini harus dapat menampilkan soal praktikum, live coding, unggah kode program, melakukan evaluasi terhadap kode program yang telah diunggah, melakukan kompilasi, menampilkan hasil kompilasi, dan melakukan grading. Selain itu, tools ini juga dapat menampilkan skor dan rangking dari semua peserta praktikum yang telah unggah kode program.

## II. KAJIAN PUSTAKA

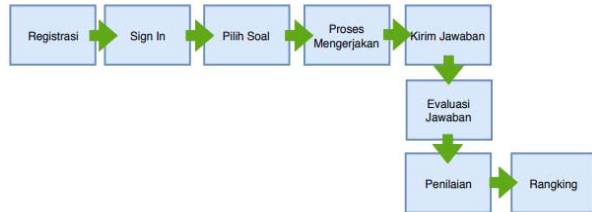
Kajian pustaka yang mendasari penelitian ini adalah sebagai berikut.

### A. Online Judge System

Merupakan sistem yang mampu melakukan evaluasi secara mendalam dan melakukan kompilasi terhadap kode program yang diunggah. Evaluasi mendalam terhadap kode sumber meliputi ukuran file, waktu eksekusi, batasan memori, runtime error, dan compilation error. Online Judge selama ini telah banyak digunakan untuk kegiatan competitive programming, baik yang dilakukan perguruan tinggi maupun perlombaan tingkat nasional seperti GEMASTIK. Beberapa contoh online judge yang banyak digunakan untuk competitive programming antara lain: SPOJ, mooshak, codechef.com, topcoder.com, dan codeforce.com. Sedangkan yang digunakan dilingkungan kampus antara lain: Assyst, The

Autograder, Web-CAT, dan Autolab. Istilah lain dari Online Judge antara lain Autograding System, Automated Assessment System, dan Automated Evaluation System.

Secara umum, alur kerja dari Online Judge adalah: (1) Memilih soal yang akan dikerjakan, (2) Menulis kode program sesuai spesifikasi soal, (3) Unggah kode program ke aplikasi Online Judge, (4) Aplikasi Online Judge melakukan evaluasi terhadap kode program dan melakukan kompilasi, (5) Aplikasi Online Judge melakukan skoring dan perangkaan, (6) Menampilkan hasil. Ilustrasi alur kerja Online Judge tersaji dalam gambar 1.



Gbr. 1 Alur Online Judge System

Berdasarkan gambar 1, proses utama dalam Online Judge adalah evaluasi dan kompilasi kode program. Pada proses ini, dilakukan evaluasi secara mendalam kode program sebelum dilakukan proses kompilasi. Evaluasi yang dilakukan meliputi memory, cpu time, maximum file size, time limit dan restricted function. Proses kompilasi akan dilakukan jika selama proses evaluasi tidak terdapat error. Selama proses kompilasi, jenis error yang mungkin terjadi antara lain runtime error, internal error, dan compilation error.

Penelitian terkait Online Judge antara lain: (Patil, 2010) yang mengembangkan online judge untuk bahasa pemrograman Java, dengan nama Javabrat yang dapat melakukan evaluasi secara cepat dan memberikan feedback. (Norazah Yusof, 2012), mengembangkan plugin moodle dengan nama Jassess, yang dapat melakukan grading bahasa pemrograman Java secara semi otomatis, menampilkan error jika selama kompilasi terdapat error, menentukan grading hanya berdasarkan kesuksesan kompilasi. (Karol Danutama, 2013), mengembangkan Autograder yang terintegrasi dengan LMS Moodle, dan (Timotius Nugraha C, 2013) yang mengembangkan tools untuk evaluasi kode program secara realtime.

Modul aplikasi yang terdapat pada online judge diantaranya adalah problem (soal), users (pengguna), submission, contest (kompetisi), statistik, news (informasi), online editor, online compile, online grading dan compiler setting. Setiap fitur dapat dikonfigurasi dengan mudah sesuai dengan kebutuhan. Modul problem digunakan untuk pengelolaan soal yakni tambah soal, edit, tampilkan soal, dan pengaturan lain yang terkait. Modul users digunakan untuk pengaturan pengguna diantaranya registrasi, login, ganti password, dan pengaturan profil. Modul submission digunakan untuk pengelolaan kode program yang telah dikirim oleh pengguna. Modul contest digunakan untuk pembuatan kompetisi, soal kompetisi, join kompetisi, penilaian kompetisi dan pengaturan kompetisi yang lain. Modul statistik berisi statistik hasil online compilation, statistik pengguna, dan statistik yang lain. Modul news

digunakan untuk tambah, edit, tampil dan edit terkait pengumuman/informasi pada Online Judge System.

Pada Setiap Online Judge System selalu tersedia online editor, yang memungkinkan pengguna dapat menuliskan kode program. Selain itu, terdapat juga online compiler yang digunakan untuk melakukan proses kompilasi secara online sesuai bahasa pemrograman yang dipilih. Bahasa pemrograman yang didukung pada online compiler antara lain C++, Java, Python, C#, C, dan lain-lain. Online compiler diperlukan pada saat proses evaluasi kode program sehingga dapat dilakukan penilaian secara otomatis.

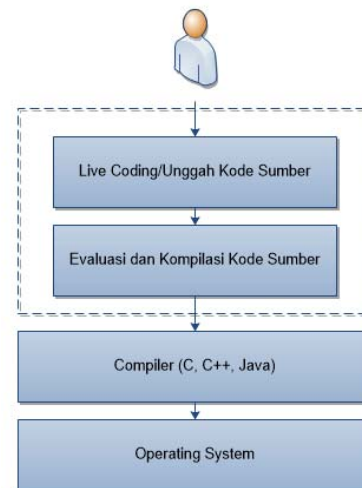
### B. Model Penilaian Otomatis

Proses penilaian dimulai ketika pengguna mengirimkan jawaban dalam bentuk kode program (source code) kemudian dilakukan proses kompilasi oleh online compiler. Jika terdapat kesalahan pada saat kompilasi maka akan ditampilkan pesan error, dan jika tidak terdapat kesalahan maka dilakukan proses penilaian (grading). Pesan error pada saat kompilasi antara lain: AC (Accepted), WA (Wrong Answer), Compile Error, Time Limit Exceeded, Memory Limit Exceeded dan lain-lain.

Fitur penilaian yang dikembangkan pada aplikasi ini mengacu pada bobot soal dan jumlah benar hasil pencocokan dengan kunci jawaban. Semakin tinggi bobot soal berbanding lurus dengan tingkat kesulitan dan kompleksitas. Semakin banyak jumlah kunci jawaban dalam bentuk input dan output maka persentase nilai menjadi lebih kecil.

### C. Arsitektur Aplikasi

Arsitektur aplikasi yang diusulkan terdiri dari tiga lapisan antara lain: live coding/unggah kode sumber, evaluasi kode sumber, dan compiler. Ilustrasi arsitektur tersaji pada gambar 2.



Gbr. 2 Arsitektur Aplikasi Automatic Programming Assessment

Berdasarkan gambar 2, Praktikan hanya berinteraksi dengan aplikasi selama kegiatan praktikum. Interaksi yang dilakukan antara lain: registrasi, login, memilih soal praktikum, menuliskan kode program secara live, dan mengirim kode program. Setelah kode program terkirim,

Aplikasi kemudian melakukan evaluasi dan kompilasi dengan memanfaatkan compiler C, C++, dan Java yang telah terpasang pada sistem operasi Linux.

### III. METODE PENELITIAN

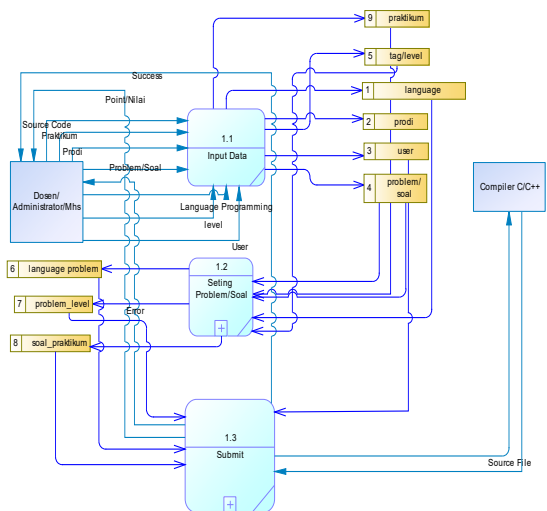
Penelitian ini dilakukan pada jurusan teknik informatika, FT Unesa untuk mata kuliah praktikum pemrograman dasar dengan jumlah mahasiswa sebanyak 120. Bahasa pemrograman yang digunakan adalah C++. Setiap praktikan melakukan registrasi ke aplikasi, login, memilih soal praktikum, kemudian menuliskan kode program baik secara live pada aplikasi maupun menggunakan editor lain. Kode program yang telah selesai dikerjakan kemudian diunggah ke aplikasi. Pengujian dilakukan secara unit testing yang dilakukan selama pengembangan dan acceptance testing, yang dilakukan oleh praktikan. Metode pengembangan aplikasi menggunakan waterfall model.

### IV. IMPLEMENTASI

Semua perangkat lunak yang digunakan selama Implementasi aplikasi menggunakan teknologi open source. Implementasi yang dilakukan meliputi:

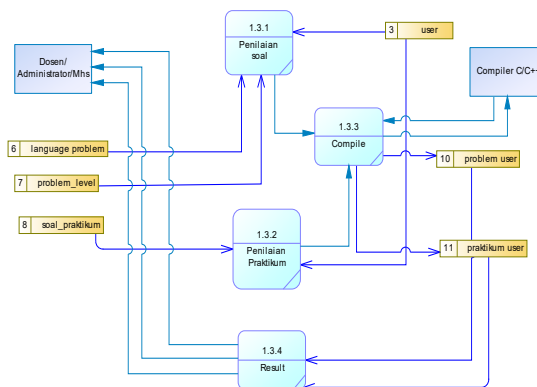
#### A. Proses Bisnis

Implementasi proses bisnis digambarkan dalam bentuk DFD (Data Flow Diagram). Entitas yang terlibat meliputi dosen, asisten, praktikan, dan compiler. DFD level 1 tersaji pada gambar 3 berikut.



Gbr. 3 Data Flow Diagram Level 1

Proses utama pada DFD level 1 antara lain: input data, seting soal, dan unggah kode program. Data store yang digunakan antara lain: praktikm, level, language, prodi, user, dan soal. Proses seting soal dan unggah memiliki subproses yang berisi detail proses yang terjadi. Gambar 4 berikut merupakan sub proses dari submit.



Gbr. 4 Data Flow Diagram subproses Submit

Berdasar gambar 4, sub proses dari submit terdiri dari proses penilaian soal, penilaian praktikum, compile, dan result. Proses penilaian berisi evaluasi terhadap kode program yang dikirim sebelum dilakukan proses kompilasi. Hasil dari proses kompilasi kemudian ditampilkan melalui proses result, baik terdapat error maupun tidak.

#### B. Database

DBMS (Database Management System) yang digunakan selama implemetnasi adalah postgresQL versi 9.5. tabel-tabel yang digunakan meliputi: soal praktikum, praktikum, prodi, users, problem\_users, language, language\_problem, dan judgments. Implementasi tabel menggunakan database migration pada framework laravel. Struktur tabel users tersaji pada gambar 5.

```
//Creation of the user table
Schema::create('users', function (Blueprint $table) {
    //Columns
    $table->increments('id');
    $table->string('name');
    $table->string('last_name');
    $table->string('nickname')->unique();
    $table->string('small')->unique();
    $table->string('password');
    $table->date('register_date');
    $table->integer('submitted')->unsigned()->default(0);
    $table->integer('accepted')->unsigned()->default(0);
    $table->enum('role',['contestant', 'coach', 'problem_setter', 'admin']->default('contestant'));
    $table->integer('institution_id')->unsigned();
    $table->integer('country_id')->unsigned();
    $table->double('points')->default(0);
    $table->string('avatar')->default('user_default.png');
    $table->boolean('confirmed')->default(0);
    $table->string('confirmation_code')->nullable();
    $table->string('provider')->nullable();
    $table->string('remember_id')->unique()->nullable();
    $table->rememberToken();
    $table->timestamps();
    //Foreign keys
    $table->foreign('institution_id')->references('id')->on('institutions');
    $table->foreign('country_id')->references('id')->on('countries');
});
```

Gbr. 5 Struktur tabel dalam migration

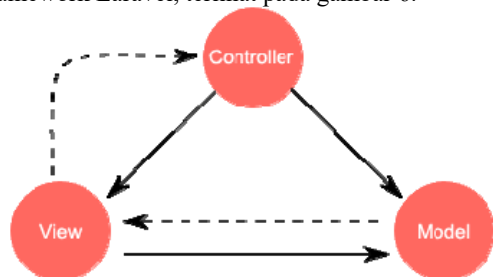
Semua tabel yang digunakan, struktur tabelnya telah didefinisikan pada laravel migration dengan alasan kemudahan ketika terjadi perubahan data.

#### C. Implementasi Program

Aplikasi assessment tool ini memiliki fitur untuk mengatur limit atau batas-batas sebuah program dapat diterima untuk dieksekusi baik bahasa C ataupun C++, sehingga dosen dapat membatasi program yang dikirim oleh praktikan sesuai dengan kemampuan dari hardware yang dimiliki oleh

laboratorium. Fitur limit antara lain: memory limit, source code limit, time limit, dan total time limit.

Apliasi dibangun menggunakan teknologi LAPP (Linux Apache PHP PostgreSQL) dengan menggunakan framework Laravel versi 5.4. Perangkat lunak pendukung yang lain antara lain: Bootstrap, DataTable, dan compiler C++ yang telah terpasang pada sistem operasi linux. Implementasi program menggunakan arsitektur MVC (Model View Controller) pada framework Laravel, terlihat pada gambar 6.



Gbr. 6 Arsitektur MVC

Jumlah file program untuk controller terdiri dari 13 file, view 14 folder, dan model 11 folder. Tiap file berhubungan antara satu dengan yang lain.

Variabel utama yang digunakan antara lain: variabel untuk menyimpan perintah untuk kompilasi (\$gcc, \$gccplus, \$javac), variabel yang menyimpan hasil evaluasi (\$size\_limit, \$time\_limit, \$memory\_limit, \$total\_time\_limit, \$output\_limit), dan variabel array yang berisi pesan hasil kompilasi. Daftar beberapa variabel utama yang digunakan terlihat pada gambar 7.

```
//vars used to compiling and executing
private static $GCC = "/usr/bin/gcc -std=c99 -w ";
private static $GCCPLUSPLUS = "/usr/bin/g++ -w ";
private static $JAVAC = "javac ";
private static $JAVA = "java -Xmx512m -cp ";//-Djava.compiler=NONE -cp ";
private static $PYTHON = "python ";

//vars used to store the limits to evaluate
private static $SIZE_LIMIT = 0;
private static $TIME_LIMIT = 0;
private static $MEMORY_LIMIT = 0;
private static $TOTAL_TIME_LIMIT = 0;
private static $OUTPUT_LIMIT = 50 * 1024; //represented in bytes (50 kb)
```

Gbr. 7 Potongan kode program variabel yang digunakan

Isi dari variabel array yang digunakan untuk menampilkan pesan dari hasil evaluasi dan kompilasi terlihat pada gambar 8.

```
private static $VERDICTS = array(
    "JDG" => "Judging",
    "UOQ" => "Unqualified",
    "IE" => "Internal Error",
    "SLE" => "Size Limit Exceeded",
    "RF" => "Restricted Function",
    "CE" => "Compilation Error",
    "RTE" => "Runtime Error",
    "TLE" => "Time Limit Exceeded",
    "MLE" => "Memory Limit Exceeded",
    "OLE" => "Output Limit Exceeded", //Output limit currently is 64mb
    "PE" => "Presentation Error",
    "WA" => "Wrong Answer",
    "AC" => "Accepted"
);
```

Gbr. 8 Variabel Array untuk output hasil evaluasi program

Kode program utama adalah pada proses evaluasi dan kompilasi. Proses evaluasi yang dilakukan antara lain:

eval\_size\_limit, eval\_restricted, eval\_wa, eval\_compilation\_error, dan eval\_runtime\_error. Sedangkan proses kompilasi dilakukan sesuai pengaturan bahasa yang digunakan meliputi C, C++, Java dan python.

### V. UJICOBA

Pengujian dilakukan melalui dua tahap yakni pada saat pengembangan aplikasi (development testing) dan user testing. pengujian pada saat pengembangan bertujuan untuk menemukan bug pada program dan melakukan perbaikan. Cara yang dilakukan untuk pengujian ini adalah unit testing, componen testing, dan system testing. Pengujian unit testing terlihat pada gambar 9.

```
nafls@kingpin:~/www/lv/riset/laravel-tdd (master)
$ vendor/bin/phpunit
PHPUnit 6.4.0 by Sebastian Bergmann and contributors.

.
Time: 102 ms, Memory: 10.00MB

There was 1 error:

1) Tests\Feature\ExampleTest::testBasicTest
Error: Call to undefined method Tests\Feature\ExampleTest::assertStatus()

/home/nafls/www/lv/riset/laravel-tdd/tests/Feature/ExampleTest.php:19

ERRORS!
Tests: 2, Assertions: 1, Errors: 1
nafls@kingpin:~/www/lv/riset/laravel-tdd (master)
$
```

Gbr. 9 Unit Testing program

Pada gambar 9 menunjukkan terdapat error yang harus diperbaiki yang terletak pada file ExampleTest pada baris 20. Setelah dilakukan perbaikan, dan dilakukan unit testing kembali maka pesan error sudah tidak ada, lihat gambar 10.

```
nafls@kingpin:~/www/lv/riset/laravel-tdd (master)
$ vendor/bin/phpunit
PHPUnit 6.4.0 by Sebastian Bergmann and contributors.

..
Time: 102 ms, Memory: 10.00MB

OK (2 tests, 2 assertions)
nafls@kingpin:~/www/lv/riset/laravel-tdd (master)
$
```

Gbr. 10 Hasil Unit Testing program

Ujicoba dilaksanakan pada mahasiswa jurusan Teknik Informatika yang memprogram mata kuliah pemrograman dasar. Rujukan soal praktikum yang dilakukan saat ujicoba adalah modul praktikum. Berdasarkan modul praktikum, mahasiswa mengerjakan tugas praktikum yang terkait dengan pemrograman, baik memperbaiki program yang salah maupun membuat solusi permasalahan dalam bentuk program.

Dalam mengerjakan pemrograman, mahasiswa bisa langsung menggunakan online editor yang tersedia atau dikerjakan secara offline dilaptop masing-masing. Kode program yang telah selesai dikerjakan, kemudian dikirim (submit) melalui aplikasi Automatic Programming Assessment dengan terlebih dahulu melakukan proses login. Aplikasi kemudian melakukan evaluasi secara mendalam terhadap kode program yang telah dikirim. Jika status hasil evaluasi adalah AC (Accepted) maka dilakukan proses penilaian. Proses penilaian dilakukan dengan mencocokkan input yang diberikan dengan output yang dihasilkan untuk beberapa kasus. Apapun hasil pencocokan, akan ditampilkan

hasil penilaiannya. Contoh hasil ujicoba tersaji pada gambar 11 berikut.

### Submission of A Minus B by *asmunin*

[View source](#)  
[Resubmit](#)

#### Execution Results

Test case #1: **WA** [0.013s, 1.63 MB] (0/1)  
Test case #2: **WA** [0.013s, 1.63 MB] (0/1)  
Test case #3: **WA** [0.012s, 1.63 MB] (0/1)  
Test case #4: **WA** [0.009s, 1.63 MB] (0/1)  
Test case #5: **WA** [0.008s, 1.63 MB] (0/1)  
Test case #6: **WA** [0.007s, 1.63 MB] (0/1)  
Test case #7: **WA** [0.007s, 1.63 MB] (0/1)  
Test case #8: **WA** [0.007s, 1.63 MB] (0/1)  
Test case #9: **WA** [0.007s, 1.63 MB] (0/1)  
Test case #10: **WA** [0.007s, 1.63 MB] (0/1)

Final score: 0/10 (0.0/3 points)

Gbr. 7 Hasil Penilaian Aplikasi

Berdasarkan gambar 5, total point yang diperoleh mahasiswa adalah 3 point dengan jumlah kasus pencocokan input dan output sebanyak 10. Nilai akhir yang diperoleh mahasiswa adalah hasil test case dari jumlah test case yang diberikan, dan hasilnya kemudian dikalikan dengan total point dari soal yang diberikan. Karena dari 10 kali pencocokan tidak ada yang benar, yang ditandai dengan WA (Wrong Answer), maka nilai yang diperoleh mahasiswa adalah  $0/10 \times 3 = 0$ . Berdasarkan nilai yang telah diperoleh, aplikasi kemudian melakukan perankingan secara descending (dari nilai tinggi ke nilai rendah).

## VI. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa fitur penilaian otomatis terhadap kode program yang dikirim oleh mahasiswa dapat berjalan dengan baik. Hal ini dibuktikan dengan kesamaan hasil penilaian antara penilaian manual dan penilaian melalui aplikasi. Fitur ranking juga dapat menampilkan urutan mahasiswa yang telah memperoleh nilai dengan baik berdasarkan nilai (point) dan persentase AC (Accepted Answer). Perlu pengembangan lebih lanjut agar aplikasi dapat memberikan penilaian tidak hanya berdasarkan input dan output, melainkan penilaian terhadap indentansi program, waktu eksekusi, dan kode program yang terpecah menjadi beberapa file.

## REFERENSI

- [1] Kurnia, A., Lim, A., Cheang, B (2001), Online Judge, Computer and Education 36, 299-315.
- [2] Kosowski, A., Malafiesjski, M., Noinski T (2008). Application of an Online Judge & Contester System in Academic Tuition. ICWL 2007, 343-354.
- [3] Kevin A, N., Jean H, G., Dieters, V (2010), Marking student program using graph similarity, Computer & Education 54, 545-561.
- [4] Liang, H., Chen, C., Zhong, Z (2017), Design and implementation of Online Automatic Judging System, IOP Conf. Series: Earth and Environment Science 69.
- [5] Sharma, H., Jain, R., Shekhar, S (2017), Secure Online Judge in Cloude Environment, IOSR Journal of Computer Engineering (IOSR-JCE) Volume 19, 102-106.
- [6] Gui Ping Wang, Shu Yu Chen, Xin Yang & Rui Feng (2015), OJPOT: online judge & practice oriented teaching idea in programming courses, European Journal of Engineering Education.