

Evaluasi Implementasi Virtual Programming Lab Pada E-Learning

Meylanie Olivya¹, Zawiyah Saharuna², Iin Karmila Yusri³

^{1,2,3} Teknik Elektro, Politeknik Negeri Ujung Pandang

meylanie@poliupg.ac.id¹, zawiyah@poliupg.ac.id², iin.yusri@poliupg.ac.id³

Abstract



Penyebaran penyakit yang disebabkan oleh Virus Corona (COVID-19) pada tahun 2020, menyebabkan perubahan terhadap pelaksanaan kegiatan belajar mengajar yang secara menyeluruh dituntut untuk dilaksanakan secara on-line (daring). Namun, pelaksanaan perkuliahan di perguruan tinggi vokasi cukup berbeda termasuk pada Politeknik Negeri Ujung Pandang, karena perkuliahan lebih banyak dilaksanakan secara praktek. Oleh karena itu, cukup sulit melaksanakan seluruh kegiatan perkuliahan secara daring. Meskipun demikian, hal ini dapat diatasi dengan menggunakan virtual reality, simulator, dan lain sebagainya. Salah satu simulator pemrograman yang dapat diimplementasikan pada e-Learning adalah Virtual Programming Lab (VPL). Simulator ini menyediakan suatu lingkungan pemrograman yang terdiri atas editor, compiler, asesmen, dan pengecek plagiat. Asesmen otomatis pada VPL membutuhkan test cases yang perlu dibuat sesuai dengan soal tugas pemrograman yang diberikan. Dengan demikian, mahasiswa dapat mengetahui nilainya berdasarkan hasil dari test cases, sehingga mereka dapat melakukan improvisasi terhadap program sehingga program dapat berhasil dijalankan dan memenuhi semua test cases-nya. Hasil evaluasi terhadap implementasi VPL pada e-learning pada Program Studi Teknik Komputer dan Jaringan, menunjukkan bahwa fitur-fitur VPL dapat berfungsi dengan baik dalam mengevaluasi tugas-tugas mahasiswa. Kemudian, pada hasil evaluasi berdasarkan user acceptance, menunjukkan bahwa sebagian besar mahasiswa dapat menerima penggunaan VPL pada e-learning.

Keywords: Virtual Programming Lab, e-learning, implementasi, evaluasi

I. PENDAHULUAN

Pembelajaran jarak jauh merupakan salah satu model pembelajaran yang memungkinkan terjadinya kegiatan belajar mengajar antar dosen dan mahasiswa yang tidak berada dalam satu tempat (jarak jauh). Model pembelajaran ini telah banyak digunakan oleh perguruan tinggi yang menyelenggarakan kelas jarak jauh. Namun, dengan merebaknya penyakit yang disebabkan oleh Virus Corona (COVID-19) menyebabkan perubahan terhadap pelaksanaan kegiatan belajar mengajar. Kegiatan perkuliahan secara menyeluruh dituntut untuk dilaksanakan secara on-line (daring) baik kelas jarak jauh maupun kelas regular. Namun, pelaksanaan perkuliahan di perguruan tinggi vokasi cukup berbeda, karena perkuliahan lebih banyak dilaksanakan secara praktek. Oleh karena itu, cukup sulit melaksanakan seluruh kegiatan perkuliahan secara daring pada perguruan tinggi vokasi. Meskipun demikian, hal ini dapat diatasi dengan menggunakan virtual reality, simulator, dan lain sebagainya.

Politeknik Negeri Ujung Pandang (PNUP) adalah salah satu perguruan tinggi vokasi di Indonesia. Pelaksanaan perkuliahan di PNUP telah didukung dengan adanya Learning Management System (LMS) yang biasa disebut sebagai e-Learning. Dalam hal ini, sebelum pandemi COVID-19, e-Learning biasanya digunakan sebagai wadah penyimpanan materi maupun pengumpulan tugas mahasiswa secara daring. Penyampaian materi dan praktikum dilakukan secara langsung. Namun, pada masa pandemi, penyampaian materi dilakukan melalui video conference, sedangkan, praktikum dapat menggunakan virtual reality dan simulator.

Salah satu program studi (PS) di PNUP adalah PS Teknik Komputer dan Jaringan (PS TKJ). Pada PS TKJ, terdapat banyak mata kuliah yang menerapkan praktek, khususnya praktek pemrograman. Salah satu simulator pemrograman yang bisa diimplementasikan pada e-Learning adalah Virtual Programming Lab (VPL).

VPL menyediakan suatu lingkungan pemrograman yang terdiri atas editor,

compiler, asesmen, dan pengecek plagiarisme (Culligan & Casey, 2018). Pembuatan program sederhana menggunakan bahasa pemrograman seperti C, Java, Python, dan lain sebagainya dapat dilakukan dengan VPL secara daring (Cardoso dkk, 2018). Selain itu, asesmen otomatis dan pengecek plagiarisme yang tersedia dapat memudahkan dalam melakukan evaluasi terhadap tugas-tugas pemrograman mahasiswa. Kedua komponen VPL tersebut, dapat menggantikan tugas dosen dalam mengoreksi tugas-tugas pemrograman mahasiswa.

Namun, meskipun demikian, VPL tidak dapat langsung digunakan terlebih jika asesmen otomatis dibutuhkan. Asesmen otomatis pada VPL membutuhkan test cases yang perlu dibuat sesuai dengan soal tugas pemrograman yang diberikan. Test cases ini akan menguji kebenaran program yang dibuat, sekaligus menilai program tersebut berdasarkan test cases yang berhasil dipenuhi. Pembuatan test cases cukup rumit karena harus dibuat berdasarkan soal dan kemungkinan-kemungkinan output programnya. Mahasiswa dapat mengetahui nilainya berdasarkan hasil dari test cases, sehingga mereka dapat melakukan improvisasi terhadap program sehingga program dapat berhasil dijalankan dan memenuhi semua test cases-nya. Dengan demikian, mahasiswa dapat memperoleh pengalaman seperti praktek secara langsung di laboratorium, karena terdapat test cases yang dapat mengoreksi program seperti layaknya seorang dosen.

Berdasarkan uraian di atas, perlu adanya evaluasi terhadap implementasi VPL pada e-Learning. Implementasi VPL yang dimaksud dalam hal ini adalah pembuatan soal beserta test cases-nya. Evaluasi penting dilakukan untuk mengetahui kinerja VPL dalam memberikan pengalaman perkuliahan praktek pemrograman meskipun dilaksanakan secara virtual. Dengan demikian, dapat ditentukan kelayakan dari implementasi VPL dalam pembelajaran daring, tidak hanya selama masa pandemi, namun juga dapat digunakan untuk pembelajaran jarak jauh khususnya untuk mata kuliah pemrograman.

II. KAJIAN LITERATUR

A. *Virtual Programming Lab* (VPL)

VPL atau *Virtual Programming Lab* adalah salah satu plugin penilai otomatis

dalam *Moodle Learning Management System* (LMS). VPL menyediakan suatu lingkungan pengembangan secara daring untuk tugas-tugas pemrograman. Bahasa pemrograman yang dapat digunakan pada VPL sangat beragam yaitu C, C++, C#, Java, Pascal, PHP, Python, dan lain sebagainya (Cardoso dkk, 2018). VPL juga memiliki penilai otomatis dan semi otomatis, deteksi plagiat, dan menawarkan fitur-fitur yang dapat dikonfigurasi sesuai dengan tugas pemrograman yang diberikan (Culligan & Casey, 2018). Dalam hal ini, seorang koordinator mata kuliah dapat membuat sendiri rubrik penilai tugas yang akan menilai tugas pemrograman secara otomatis. Dengan demikian, mahasiswa akan memperoleh umpan balik secara otomatis ketika melakukan pemrograman dengan menggunakan VPL, sehingga mahasiswa dapat mengetahui kesalahan program yang dibuatnya.

Arsitektur VPL terdiri atas tiga elemen (Cardoso dkk, 2018) sebagai berikut:

1. Modul Moodle; Modul ini berfungsi sebagai plugin Moodle agar dapat digunakan dalam LMS Moodle. Fitur-fitur modul ini yaitu manajemen pengajuan tugas, penilai, dan pengecek plagiat
2. Editor kode program yang dapat dijalankan pada browser (Mozilla Firefox, Google Chrome, dsb). Elemen ini memungkinkan user melakukan edit, run, dan tes program yang telah dibuat tanpa harus menginstal compiler
3. Server Linux (jail server) yang menyediakan lingkungan atau wadah tempat tugas-tugas pemrograman dieksekusi dan di-evaluasi secara daring dan aman.

Penelitian-penelitian terhadap penggunaan VPL telah dilakukan sejak tahun 2012. Diawali dengan penelitian terhadap pengembangan VPL yang merupakan suatu lingkungan pengembangan program sederhana yang mampu menilai program secara otomatis (Rodriguez-del-Pino dkk, 2012). Kemudian diikuti dengan penelitian-penelitian terhadap penggunaan VPL. Penelitian ini didasarkan pada harapan akan kemudahan untuk menilai tugas-tugas pemrograman, diantaranya penelitian yang dilakukan oleh Dominique Thiebaut pada tahun 2015. Penelitian ini mengevaluasi fitur penilai otomatis dari VPL. Hasil penelitian ini adalah bahwa VPL sangat

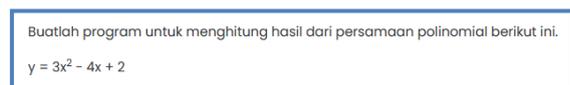
memudahkan pengajar dalam pembelajaran pemrograman (Thiebaut, 2015). Kemudian pada tahun 2017, dilakukan penelitian terhadap penggunaan fitur-fitur VPL dalam bahasa pemrograman python (Pisani & Carvalho, 2017). Pada tahun 2017 pula, dilakukan penelitian terhadap penggunaan VPL dalam kompetisi pemrograman (Wirawan dkk, 2017). Kemudian pada tahun 2019, terdapat penelitian terhadap penggunaan VPL dalam keterampilan pemrograman mahasiswa tingkat pemula (Cardoso dkk, 2020).

B. Penggunaan VPL

Untuk menggunakan VPL, perlu dilakukan konfigurasi yang disesuaikan dengan tipe tugas. VPL memperbolehkan beragam aktivitas seperti contoh atau percobaan praktikum, tes cloze, latihan-latihan soal, dan juga latihan pengembangan kode program (Cardoso dkk, 2018). Semua aktivitas harus didefinisikan, dikonfigurasi, dan diatur oleh pengajar. Konfigurasi-konfigurasi tersebut adalah sebagai berikut:

1. Periode pengumpulan tugas
2. Pembatasan terhadap pengumpulan tugas, seperti jumlah file yang dapat dimasukkan, user yang boleh memasukkan tugas, proses eksekusi file, dsb
3. Cara tugas program dievaluasi dan dinilai
4. Test cases
5. Visibilitas penilaian
6. Evaluasi bergantung pada ukuran byte program
7. Fitur copy/paste dapat dinonaktifkan Penilai otomatis dalam VPL dapat

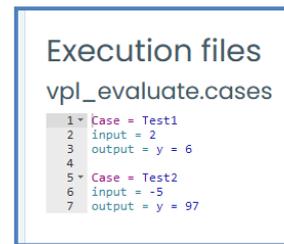
berfungsi dengan baik dengan adanya test cases. Adapun fungsi dari test cases adalah memeriksa kebenaran output dari program yang dihasilkan oleh mahasiswa. Suatu program meskipun dapat dijalankan, program tersebut juga harus dapat memberi output yang benar. Pada Gambar 1 berikut adalah contoh soal latihan pemrograman yang diberikan kepada mahasiswa.



Gambar 1. Contoh soal latihan pemrograman

Berdasarkan contoh soal yang diberikan, pengajar dapat menyusun test cases untuk memeriksa output dari program jawaban soal tersebut. Semakin rumit soal, maka jumlah test cases pun akan bertambah. Pada Gambar 2 menunjukkan contoh test cases yang

dibuat berdasarkan soal yang diberikan pada Gambar 1.



Gambar 2. Test cases soal

III. METODE PENELITIAN

Dalam penggunaannya, VPL dapat memeriksa kode-kode program dalam berbagai bahasa seperti bahasa C dan Python. Kedua bahasa pemrograman ini digunakan pada Mata Kuliah (MK) Algoritma dan Pemrograman (ALPRO) dalam semester ganjil pada PS TKJ PNU. Oleh karena itu, VPL dapat digunakan pada mata kuliah tersebut.

Soal-soal latihan pemrograman dibuat berdasarkan capaian pembelajaran beberapa pokok bahasan dalam MK ALPRO. Adapun pokok bahasan tersebut berkaitan dengan teknik pemrograman dasar secara umum, yaitu:

1. Dasar Pemrograman
2. Struktur Percabangan
3. Struktur Loop

Adapun jumlah soal setiap pokok bahasan bergantung pada capaian pembelajarannya.

Setelah dilakukan penyusunan soal latihan, kemudian dirancang test cases untuk setiap soal tersebut. Setiap program jawaban soal latihan pemrograman dapat memiliki satu atau lebih jenis output, sehingga jumlah test cases juga akan bergantung pada jumlah output.

A. Penyusunan Soal-Soal Latihan

Tabel 1 berisi soal-soal latihan untuk pokok bahasan Dasar Pemrograman berdasarkan capaian pembelajarannya. Capaian pembelajaran untuk pokok bahasan tersebut adalah mahasiswa mampu menerapkan teknik pemrograman dasar meliputi tipe data, variabel, input dan output, pernyataan, dan operator untuk membuat program sederhana.

Tabel 1 Soal Latihan Pokok Bahasan Dasar Pemrograman

N	Soal (tampilan pada e-learning)
0	

No	Soal (tampilan pada e-learning)
1	<pre>1 /* Program mencetak bilangan bulat yang diinput oleh user */ 2 3 #include <stdio.h> 4 5 main() { 6 int x; 7 8 printf("Masukkan sebuah bilangan bulat: "); 9 scanf("%d", &x); 10 11 printf("\nKamu memasukkan bilangan bulat %d\n", x); 12 }</pre>
2	<p>Buatlah program untuk menghitung hasil dari persamaan polinomial berikut ini.</p> $y = 3x^2 - 4x + 2$ <p>Samakan tampilan program seperti berikut ini</p> <pre>x = 2 y = 6</pre>
3	<p>Buatlah program yang menghitung keliling dan luas lingkaran dengan jari-jari yang diinputkan oleh user</p> <p>Contoh output:</p> <pre>Masukkan jari-jari: 3 keliling = 18.84 luas = 28.26</pre>

B. Perancangan Penilai Otomatis

Penilai otomatis dalam hal ini menggunakan test cases. Nilai yang diperoleh mahasiswa untuk setiap soal bergantung pada jumlah test cases yang dipenuhi. Misalnya soal no.3 pada Tabel 1, mahasiswa diminta untuk membuat program menghitung hasil dari persamaan kuadrat yang telah ditentukan. Akar dari persamaan kuadrat memiliki dua kemungkinan yaitu dapat berupa bilangan positif maupun negatif. Oleh karena itu, dibuatkan dua test cases yang menguji kebenaran program terhadap dua bilangan input positif maupun negatif. Test cases untuk soal no.3 tersebut ditunjukkan pada Gambar 3.

Buatlah program untuk menghitung hasil dari persamaan polinomial berikut ini.

$$y = 3x^2 - 4x + 2$$

Samakan tampilan program seperti berikut ini

```
x = 2
y = 6
```

Execution files

vpl_evaluate.cases

```
1 * Case = Test1
2 input = 2
3 output = y = 6
4
5 * Case = Test2
6 input = -5
7 output = y = 97
```

Test cases

Gambar 3. Contoh test cases soal no.2 pada Tabel 1

Adapun contoh test cases lainnya terdapat pada Gambar 4 dan 5 berikut ini.

Execution files

vpl_evaluate.cases

```
1 * Case = Test1
2 input = 4
3 output = Masukkan sebuah bilangan bulat:
4 Kamu memasukkan bilangan bulat 4
```

Gambar 4. Contoh test cases soal no.1 pada Tabel 1

Execution files

vpl_evaluate.cases

```
1 * Case = Test1
2 input = 3
3 output = keliling = 18.84
4 luas = 28.26
```

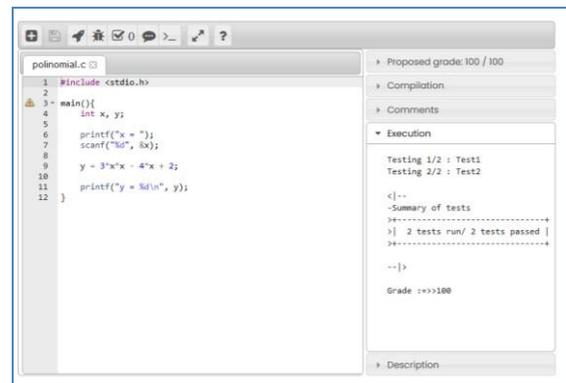
Gambar 5. Contoh test cases soal no.3 pada Tabel 1

IV. HASIL DAN PEMBAHASAN

A. Pengujian Penilai Otomatis

Setelah membuat test cases, kemudian dilakukan pengujian terhadap penilai otomatis. Hal ini bertujuan untuk mengetahui realibilitas penilai otomatis tersebut.

Penilai otomatis akan bekerja pada saat user menjalankan program yang telah dibuat pada code editor VPL. Pada saat program dijalankan, VPL akan melakukan pengujian terhadap input dan output sesuai test cases, sehingga akan menghasilkan nilai secara otomatis. Gambar 6 menunjukkan ruang kerja pada VPL.

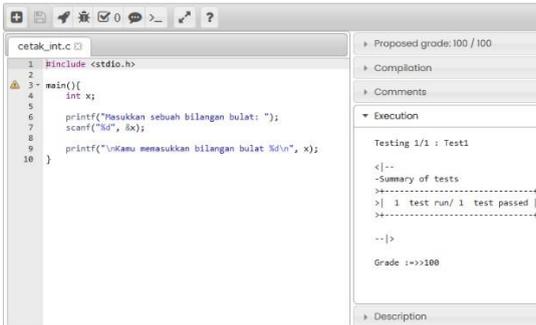


Gambar 6. Ruang kerja VPL

Pada ruang kerja VPL berdasarkan gambar yang ditunjukkan pada Gambar 6, terdapat tab *Proposed Grade*, *Compilation*, *Comments*, *Execution*, dan *Description*. Tab *Proposed Grade* menunjukkan nilai hasil pengujian program, tab *Compilation* menunjukkan hasil kompilasi program, tab *Comments* menunjukkan rangkuman hasil tes, tab *Execution* menampilkan hasil pengujian dari setiap test cases, dan tab *Description* menampilkan soal latihan.

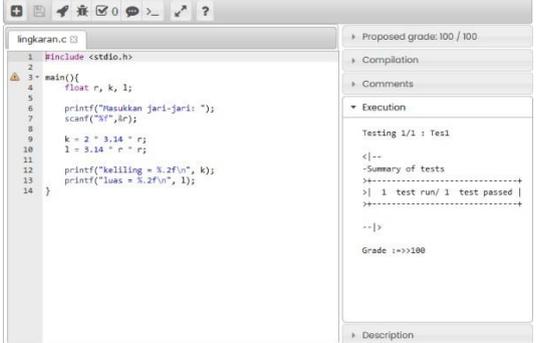
Pada tab *execution* terdapat hasil pengujian terhadap program berdasarkan test cases. Oleh karena terdapat dua test cases maka dilakukan dua kali pengujian. Pada tab tersebut juga terdapat rangkuman hasil test cases beserta nilai yang diperoleh. Adapun hasil pengujian yang dilakukan terhadap test cases yang dibuat berdasarkan Tabel 1, ditunjukkan pada Tabel 2.

Tabel 5.6. Hasil Pengujian Penilai Otomatis

Hasil Pengujian	
Soal no.1 Tabel 1	
	<p>Proposed grade: 100 / 100</p> <p>Execution</p> <pre>Testing 1/1 : Test1</pre> <p>Grade : >>>100</p>

Hasil pengujian menunjukkan bahwa program yang dibuat dapat memenuhi test cases yang diberikan sehingga memperoleh nilai 100.

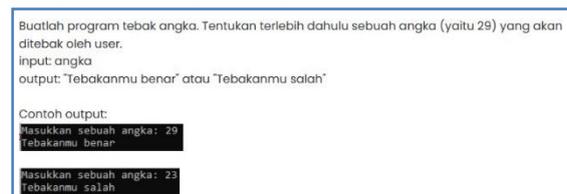
Soal no.3 Tabel 1

	<p>Proposed grade: 100 / 100</p> <p>Execution</p> <pre>Testing 1/1 : Test1</pre> <p>Grade : >>>100</p>
--	---

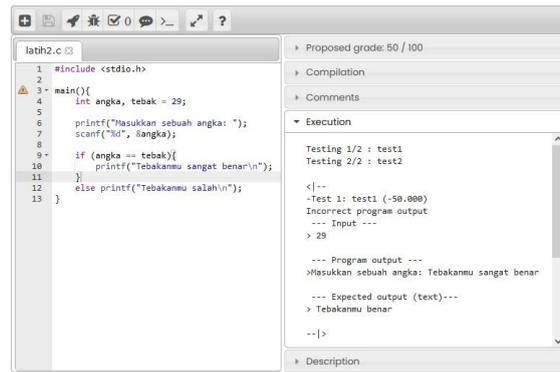
Hasil pengujian menunjukkan bahwa program yang dibuat dapat memenuhi test cases yang diberikan sehingga memperoleh nilai 100.

Hasil pengujian test cases pada Tabel 2 menunjukkan bahwa test cases dapat menguji kebenaran program yang dibuat. Hal ini disebabkan setiap test cases dibuat berdasarkan soal.

Pengujian kedua dilakukan atas kemungkinan bahwa program dibuat berdasarkan instruksi soal dan kreativitas pemrogram. Pengujian ini dilakukan terhadap soal pada Gambar 7 dengan mengubah kode program sehingga dapat menghasilkan output yang tidak sesuai dengan test cases. Hasil pengujian ditunjukkan pada Gambar 8.



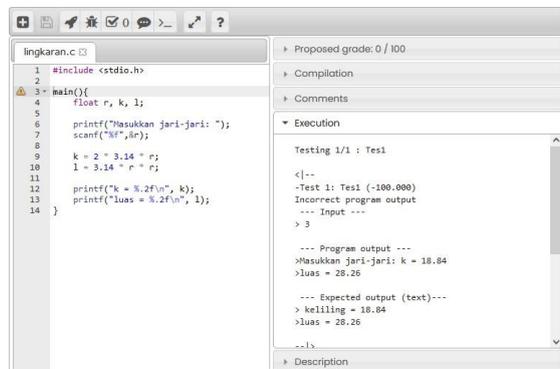
Gambar 7. Contoh soal Pokok Bahasan Struktur Percabangan



Gambar 8. Pengujian dengan output program yang berbeda dengan test cases pada soal Gambar 7

Pada Gambar 8, dalam tab Execution, dapat diketahui bahwa program yang dibuat tidak memenuhi test cases pertama. Perubahan yang dilakukan pada kode program adalah seperti yang ditunjukkan dalam kotak biru pada gambar. Sedangkan output yang diharapkan pada test cases pertama adalah kalimat "Tebakanmu benar". Oleh karena itu, output program tidak sesuai dengan output yang diharapkan pada test cases pertama, sehingga dihasilkan nilai 50.

Demikian pula halnya yang terjadi pada soal no.3 pada Tabel 1, jika dilakukan pengujian seperti di atas. Hasil Pengujian ditunjukkan pada Gambar 9.



Gambar 9. Pengujian dengan output program yang berbeda dengan test cases pada soal no.3 Tabel 1.

Berdasarkan kedua pengujian yang ditunjukkan pada Gambar 8 dan 9, dapat disimpulkan bahwa meskipun program dapat berjalan dengan benar, namun jika tidak sesuai dengan output yang diharapkan pada test cases, maka penilai otomatis akan menghasilkan nilai 0. Nilai ini hanya untuk test cases yang tidak dipenuhi.

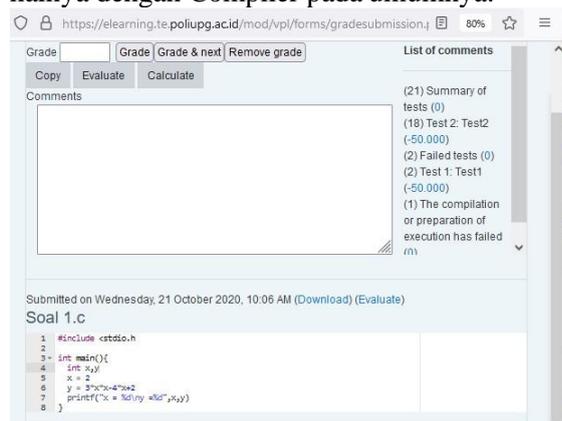
B. Evaluasi terhadap Fitur-Fitur VPL

Evaluasi ini dilakukan dengan mengamati hasil yang diperoleh mahasiswa dari ketiga pokok bahasan yang diujikan. Ketiga pokok

bahasan tersebut diselenggarakan dalam tiga minggu secara berurutan. Dalam kurun waktu tersebut, diharapkan dapat diamati kemampuan VPL dalam mengevaluasi tugas mahasiswa. Dari setiap pokok bahasan akan diambil sampel satu soal latihan untuk diamati hasil penggunaan VPL.

Berdasarkan pengamatan, terdapat beberapa tugas berstatus no grade yang menunjukkan program tidak dapat dinilai secara otomatis dan juga dua tugas dengan nilai 0. Gambar 10 menunjukkan tugas berstatus no grade yang dimasukkan oleh mahasiswa tersebut.

Pada Gambar 10, diketahui bahwa program yang dibuat memiliki kesalahan syntax, sehingga tidak dapat dikompilasi. Dari gambar tersebut, dapat diketahui pula bahwa proses kompilasi program menggunakan VPL berfungsi dengan baik. Demikian halnya dengan hasil kompilasinya, mahasiswa dapat mengetahui kesalahan kode program sama halnya dengan Compiler pada umumnya.



Gambar 10. Tugas program yang berstatus no grade

B. Evaluasi Berdasarkan User Acceptance

VPL dalam penggunaannya ditanam pada e-learning. Dengan demikian, untuk melakukan evaluasi berdasarkan User Acceptance, maka perlu dibuat dalam bentuk modul. Kemudian, evaluasi ini dilakukan dengan menggunakan kuesioner yang berisi umpan balik pengguna modul yaitu mahasiswa.

Modul ini telah dipraktikkan oleh mahasiswa Angkatan 2020 kelas B pada PS TKJ yang telah mengambil mata kuliah yang sama pada tahun 2020. Hasil evaluasinya ditunjukkan pada Tabel 3.

Tabel 3. Hasil evaluasi umpan balik pengguna

No	Aspek Penilaian	SKOR
1	Aspek Kelayakan Isi	88,42%

2	Aspek Kebahasaan	81,86%
3	Aspek Kemanfaatan	87,71%
4	Aspek Kegrafikan/tata layout	80,43%

Berdasarkan hasil yang diperoleh pada Tabel 3, dapat disimpulkan bahwa sebagian besar pengguna atau mahasiswa dapat menerima modul ini. Dengan demikian, penggunaan VPL pada modul ini, juga dapat diterima.

V. KESIMPULAN

Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa VPL dapat diimplementasikan pada e-Learning untuk mendukung pembelajaran praktik pemrograman. Kinerja VPL juga dapat dikatakan baik, karena mampu menilai secara otomatis program yang dimasukkan oleh mahasiswa.

UCAPAN TERIMA KASIH

Ucapan terima kasih ditujukan kepada Pusat Penelitian dan Pengabdian kepada Masyarakat (P3M) Politeknik Negeri Ujung Pandang karena kami telah disertakan pada Pendanaan Rutin Penelitian PNUP Tahun 2021.

REFERENSI

[1] D. Thiebaut, Automatic Evaluation of Computer Programs Using Moodle’s Virtual Programming Lab (VPL) Plug-In, Journal of Computing Sciences in Colleges, 2015

[2] I.M. Wirawan, H. W. Herwanto, dan I. D. Wahyono, Virtual Programming Laboratory UM (VPLUM) For Programming Competition, 5th International Conference On Electrical, Electronics and Information Engineering (ICEEIE), 2017.

[3] J. C. Rodriguez-del-Pino, E. Rubio-Royo, dan Z. J. Hernandez-Figueroa, A Virtual Programming Lab for Moodle with Automatic Assessment and Anti-Plagiarism Features, Proceedings of The 2012 International Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government, 2012.

[4] M. Cardoso, A.V. de Castro dan A. Rocha, Integration of virtual programming lab in a process of teaching programming EduScrum based, 13th Iberian Conference

- on Information Systems and Technologies (CISTI), 2018.
- [5]M. Cardoso, R. Marques, A.V. de Castro dan A. Rocha, Using Virtual Programming Lab to improve learning programming: The case of Algorithms and Programming. Expert Systems, 2020 - Wiley Online Library, 2020.
- [6]N. Culligan dan K. Casey, Building an Authentic Novice Programming Lab Environment, International Conference on Engaging Pedagogy (ICEP), 2018.
- [7]P. H. Pisani dan A. C. P. L. F. de Carvalho, Applying the Virtual Programming Lab with Python, ICMC Technical Report, 2017.