

PERANCANGAN APLIKASI *LEARNING BY DOING* INTERAKTIF UNTUK MENDUKUNG PEMBELAJARAN BAHASA PEMROGRAMAN

Mochammad Kautsar Sophan¹, Arik Kurniawati²

^{1,2}Teknik Informatika, Universitas Trunojoyo Madura
Email: ¹kautsar@if.trunojoyo.ac.id, ²arik@if.trunojoyo.ac.id

(Naskah masuk: 15 Januari 2018, diterima untuk diterbitkan: 29 April 2018)

Abstrak

Pembelajaran bahasa pemrograman adalah salah satu mata kuliah dasar untuk mengembangkan kompetensi keahlian pemrograman di bidang teknik Informatika, salah satunya adalah mata kuliah Algoritma dan Pemrograman. Sesuai dengan analisis instruksional, mata kuliah ini memberikan kompetensi tentang konsep dasar Algoritma Pemrograman yang akan menjadi dasar bagi pengembangan dan penerapan mata kuliah-mata kuliah selanjutnya. Banyak mahasiswa kurang memahami dasar-dasar pemrograman, sehingga menemui kesulitan ketika mengerjakan tugas-tugas mata kuliah yang membutuhkan keahlian pemrograman.

Melalui inovasi pembelajaran ini, dikembangkan sebuah media pembelajaran baru dan interaktif bagi mahasiswa dengan mengintegrasikan berbagai faktor yang mempengaruhi permasalahan yang telah disebutkan sebelumnya. Aplikasi *learning by doing* interaktif untuk mendukung pembelajaran bahasa pemrograman ini dibangun menggunakan pendekatan *Web Framework* menggunakan *Codeigniter*. Aplikasi ini juga memanfaatkan fitur *share* kode pemrograman yang dikembangkan oleh *Trinket*.

Hasil penelitian menunjukkan bahwa kebermanfaatan aplikasi ini mampu membuat ketertarikan terhadap belajar pemrograman sebesar 79%. Ketertarikan yang tinggi ini membuat mahasiswa sering mencoba kode-kode program secara mandiri sehingga jumlah mahasiswa yang lulus dengan kemampuan baik selisihnya 14% dibandingkan dengan pembelajaran biasa.

Kata kunci: bahasa pemrograman, pembelajaran, *learning by doing*, *trinket*

DESIGN OF INTERACTIVE APPLICATIONS "*LEARNING BY DOING*" TO SUPPORT LEARNING PROGRAMMING

Abstract

Learning programming language is one of the basic courses to develop the competence of programming skills in Informatics engineering, one of which is the course of Algorithm and Programming. Based on instructional analysis, this lectures provides competence on the basic concepts of Programming Algorithm which will be the basis for the development and application of further lectures. Many students do not understand the basics of programming, so they have difficulty when get assignment that require programming skills.

This learning innovation, developed an interactive new learning media that can minimize several factors that affect student learning problems. Application "learning by doing Interactive" to support Learning Programming Language was built using a Web Framework approach using Codeigniter. This application also utilizes the programming code share feature developed by Trinket.

The results showed that the usefulness of this application is able to make an interest in learning programming by 79%. This high attractiveness makes students often try the program codes independently and result the ability of the number of students who graduated with a good is 14% compared with ordinary learning.

Keywords: *programming language, learning, trinket*

1. PENDAHULUAN

Pembelajaran pemrograman merupakan hal yang utama dalam program studi Teknik Informatika atau program studi yang sejenis. Namun demikian, masih banyak ditemukan

beberapa mahasiswa yang kurang memahami dasar-dasar pemrograman, sehingga menemui kesulitan ketika mengerjakan tugas-tugas mata kuliah yang membutuhkan keahlian pemrograman. Hal ini juga menjadi penghambat pada pengerjaan tugas akhir, walau terkadang mahasiswa-

mahasiswa tersebut sudah lulus mata kuliah yang berkaitan dengan pemrograman.

Hal ini mungkin disebabkan ada beberapa faktor antara lain : pemrograman bukan pembelajaran yang mudah untuk dipelajari karena berkenaan dengan konsep yang abstrak, sumber pustaka yang menjadi rujukan mahasiswa dalam belajar tidak banyak, kurang pemahamannya mahasiswa dalam memahami instruksi dalam pemrograman, pembelajaran di kelas yang melibatkan banyak orang dengan kemampuan yang berbeda-beda sehingga pengajar kesulitan dalam merancang metode yang sesuai untuk kebutuhan masing-masing mahasiswa (Lahtinen dkk. 2005). Metode pengajaran yang konvensional dalam pembelajaran pemrograman yang diterapkan di sebagian besar universitas di Lituania menyulitkan mahasiswa dalam memahami pembelajaran pemrograman sehingga mempengaruhi motivasi tingkat belajar dan akhirnya menurunkan popularitas studi tentang Teknologi Informasi (Adžgauskienė, 2012)

Bagaimanapun juga, belajar bahasa pemrograman merupakan tantangan yang kompleks sehingga membutuhkan perangkat yang tepat sebagai media pembelajarannya (Bednarick dkk, 2005). Bahkan terkadang mahasiswa yang mengalami kesulitan dalam belajar pemrograman pada awal semester, saat pada semester-semester selanjutnya jika ada pilihan tentang bahasa pemrograman lanjut, mereka akan cenderung untuk menghindarinya (Dorn dkk, 2003).

Oleh karena itu, diperlukan pendekatan pembelajaran yang mendukung visualisasi program untuk memudahkan penjelasan secara teori dan praktek bahasa pemrograman. Ditambah juga fasilitas yang memungkinkan mahasiswa dapat mengerjakan secara langsung dan bisa mendapatkan respon balik dari sistem jika mengalami kesalahan sintaks. Serta dimungkinkan juga mahasiswa bisa mencoba di lain waktu untuk mengulang lagi materi tersebut dengan praktek mandiri. Hal ini akan sangat membantu permasalahan tersebut (Pich dkk, 2008).

Progam studi Teknik Informatika Universitas Trunojoyo Madura, menempatkan mata kuliah untuk belajar bahasa pemrograman pada awal semester di mata kuliah Algoritma Pemrograman. Saat ini metode pembelajaran yang digunakan tidak hanya berbasis teks saja, namun juga secara visual dengan memanfaatkan media pembelajaran *online* yang menyediakan materi-materi permasalahan pemrograman yang harus diselesaikan. Sehingga keaktifan mahasiswa semakin mandiri. Dosen bisa membuat kelas *virtual* dengan berisi seluruh mahasiswa yang mengambil mata kuliah pemrograman. Sehingga dosen bisa memantau hasil pengerjaan mahasiswa secara *online*. Media ini juga menggunakan

Bahasa Pemrograman Python. Soal-soal materi ini bisa dijadikan sebagai kegiatan praktikum mahasiswa yang terpantau salah satu media pembelajaran yang sudah ada yaitu *Pythonroom* (Hulaloop, 2017). Namun kekurangan dari media ini adalah materi sudah tersedia, tidak ada penambahan materi yang bisa dilakukan dosen.

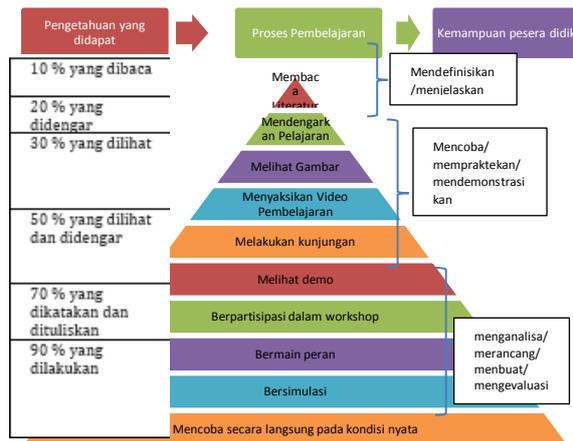
Hal inilah yang melatar belakangi untuk membuat media baru serupa namun dosen bisa membuat materi sendiri, sesuai dengan GBPP dan SAP pembelajaran prodi Teknik Informatika Universitas Trunojoyo. Dalam media ini, tidak hanya materi Algoritma Pemrograman saja yang bisa terfasilitasi, tapi juga mata kuliah lain yang berisi bahasa pemrograman yang menggunakan python. Pemilihan bahasa pemrograman Python dikarenakan Python merupakan bahasa pemrograman yang mudah dipelajari, literatur yang tersedia juga banyak dan mengikuti tren saat ini. Berdasarkan survei yang dilakukan oleh Pierre Carbonelle pada tahun 2016 menunjukkan bahwa bahasa pemrograman *Python* berada pada peringkat ke-2 setelah *Java*, dan trend ini terus meningkat sebesar 4% dalam 10 tahun terakhir (Carbonelle, 2016).

Pembahasan dalam penelitian ini, diawali dengan latar belakang mengapa Aplikasi *Learning By Doing* Interaktif dibangun, kemudian dilanjutkan dengan metode yang digunakan dan tahapan rancangan sistem yang dilakukan serta hasil dan pembahasan dari ujioba yang telah dilakukan.

2. METODE

Penelitian ini akan mengembangkan media pembelajaran untuk mendukung mata kuliah pemrograman yang menggunakan Python, salah satunya Algoritma Pemrograman. Media ini mengembangkan konsep visualisasi dan *learning by doing*. Konsep *learning by doing* ini telah diterapkan di berbagai universitas seperti, Universitas Jinan di Cina untuk mata kuliah *Software Engineering* (Ma dkk, 2014), pada Universitas Aalto di Finland untuk mata kuliah *introductory programming* (Xia, 2017), di *Masinde Muliro University of Science and Technology* untuk mata kuliah *Programming Introductory* dan *Procedural Programming* (Stephen dkk, 2011). Gambar 1 berikut adalah ilustrasi teori *Dale* yang menjelaskan pengaruh penggunaan berbagai variasi media pembelajaran dalam dunia pendidikan, dimana mahasiswa akan mendapatkan lebih banyak informasi jika mereka melakukan apa yang mereka dengar, baca, dan tuliskan. Inilah merupakan konsep dari *learning by doing*. Teori tersebut menyatakan bahwa jumlah prosentase pengetahuan yang didapat akan sangat tergantung dari keterlibatan dalam proses pembelajaran (Stephen, 2011). Konsep *learning*

by doing ini akan sangat membantu dalam proses pembelajaran pemrograman.



Gambar 1. Teori Dale

Sebenarnya ada beberapa metode lain untuk mengenalkan bahasa pemrograman, yaitu:

1. *Extreme Apprenticeship Method*, yakni *learning by doing* yang dilanjutkan dengan proses *feedback*. Metode ini telah diaplikasikan pada *CS I programming* di jurusan *Computer Science* pada Universitas Helsinki. Hasil yang ditunjukkan ternyata dapat menurunkan jumlah mahasiswa yang *drop out* terhadap mata kuliah tersebut dibandingkan dengan metode pembelajaran konvensional sebelumnya (Vihavainen, 2011).
2. *Pair Programming*, yakni memprogram dengan berpasangan. Metode ini adalah kolaborasi tim dalam pembelajaran. Dengan bekerjasama maka kemungkinan terjadi kesalahan semakin sedikit, kemudahan memahami dengan saling berdiskusi lebih mempermudah pembelajaran. Sehingga hasil evaluasi pembelajaran dengan *pair programming* lebih menjanjikan daripada *solo programming* dalam mengerjakan proyek dan peningkatan hasil pembelajaran (Nan dkk, 2008).
3. *Visual teaching approach*, dengan memvisualisasikan bagaimana kode program dijalankan (eksekusi per baris kode) dan bagaimana output yang dihasilkan setelah perintah kode tersebut dijalankan maka akan mempermudah pemahaman bagi mahasiswa yang mengambil mata kuliah pemrograman ini. Demonstrasi ini juga ditunjang dengan video animasi sehingga pemahaman mahasiswa semakin meningkat dalam memahami pemrograman (Shehane, 2014).

Dalam penelitian Kelleher, dkk (2006) tersebut, menyebutkan bahwa sebanyak 55% mahasiswa tidak lulus ujian dalam kursus pemrograman yang telah diikuti. Namun setelah

pindah ke *tools* pemrograman yang menggunakan visualisasi dalam pembelajaran bahasa pemrograman *Python* maka dapat meningkatkan jumlah mahasiswa yang tidak lulus menjadi 38%. *Tools* yang digunakan adalah salah satunya adalah *Turtle*. Hal ini menunjukkan bahwa visualisasi program dapat meningkatkan perhatian mahasiswa terhadap materi yang diajarkan.

Untuk pengembangan aplikasi ini dibangun dengan pendekatan *Web Framework*, *Codeigniter*. *Codeigniter* banyak di manfaatkan oleh para *webdeveloper* untuk mengembangkan *website* atau aplikasi (Kasurinen dkk, 2008). *Codeigniter* menggunakan konsep *design pattern Model View Controller* atau *MVC*. Pendekatan konsep ini digunakan di *Codeigniter* dalam proses pengembangan aplikasi web. Dalam konsep ini, kode program dibagi dalam 3 kategori, *Model*, *View* dan *Controller*. *Model* digunakan untuk melakukan koneksi ke database. *View* digunakan untuk melakukan presentasi atau penyajian aplikasi ke user (*User Interface*). Sedangkan *Controller* digunakan untuk membangun logika aplikasi yang menghubungkan antara *View* dan *Model*.

Trinket adalah sebuah media pembelajaran yang bersifat *open source* yang dibangun oleh tim *Trinket*. Program ini bisa *download* pada <https://trinket.io/>. *Trinket* ini seperti *YouTube* videos, dimana cukup dengan *embed* atau menggunakan link dari aplikasi tersebut agar bisa ditampilkan pada website yang akan dibangun.

Program *Trinket* ini memang digunakan secara khusus untuk mempermudah belajar bahasa pemrograman *Python* dengan tampilan menarik yang didukung dengan blok kode pemrograman dan tampilan warna. Fitur yang disediakan oleh *Trinket* sangat mendukung pembelajaran *learning by doing*, seperti respon balik dari sistem jika kode pemrograman yang dituliskan salah, hasil program bisa langsung terlihat setelah dijalankan, fasilitas *share* kode program dan hasilnya (Yehuda, 2013).

Trinket ini tidak hanya bisa dimanfaatkan untuk mata kuliah *Algoritma Pemrograman* tapi juga bisa mata kuliah lain yang mendukung pembelajaran menggunakan bahasa pemrograman *Python*, seperti mata kuliah *Struktur Data*. Namun *Trinket* ini hanya bisa digunakan untuk bahasa pemrograman *Python* tidak bisa bahasa pemrograman yang lain.

3. PERANCANGAN SISTEM

Tahapan pengembangan aplikasi ini mengikuti standar *System Development Life Cycle* dengan pendekatan model spiral. Sistem ini dikembangkan bertahap mulai dari modul dasar sampai dengan pengembangan fitur secara lengkap.

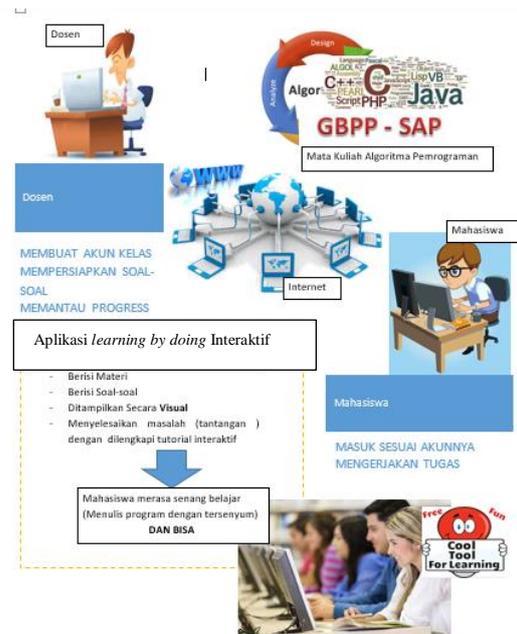
3.1. Analisa Kebutuhan

Berdasarkan hasil analisa awal, maka sistem aplikasi nantinya adalah sebagai berikut ini :

- Berisi materi mata kuliah. Sistem dikelola oleh dosen, ada pokok bahasan dan sub pokok bahasan. Setiap bahasan / sub pokok bahasan ada tautan materi atau referensi.
- Mahasiswa dikelompokkan berdasarkan grup kelas dan mempunyai akses terhadap sistem ini. Dan bisa melakukan uji coba kode *Python* di sistem, baik dalam bentuk *script* ataupun blok, kode pemrograman bisa di-*share* kepada dosen. Selain itu mahasiswa juga dapat mengerjakan soal berbasis Komputer (*Computer Based Test*) yang diberikan oleh dosen sesuai dengan grup kelas yang diikuti.
- Dosen bisa mendapatkan informasi siapa saja mahasiswa yang telah mengakses materi, dan siapa saja mahasiswa yang sedang login di sistem. Dosen mempunyai fitur dalam menyiapkan materi, soal ujian serta dapat *share* kode program untuk mahasiswa.
- Tampilan pembelajaran ini mendukung *visual learning*, yang bersifat interaktif antara dosen dan mahasiswa.

3.2. Rancangan Aplikasi

Aplikasi Referensi *Python* Interaktif Untuk Pemrograman Algoritma ini berbasis web, dimana materi bersumber pada SAP dan GBPP Mata Kuliah Algoritma Pemrograman dan dilengkapi soal-soal sebagai penunjang kegiatan praktikum. Aplikasi ini memungkinkan dosen membuat kelas virtual dengan berisikan seluruh account dari semua mahasiswa yang mengambil mata kuliah tersebut. Dengan adanya kelas *virtual* ini, progress dari setiap mahasiswa bisa terpantau. Sedangkan mahasiswa menggunakan aplikasi ini dapat mengaksesnya melalui akun yang diberikan oleh sistem. Mahasiswa bisa mengerjakan soal-soal materi sesuai dengan yang dijelaskan saat kuliah ataupun saat ujian. Jawaban dari mahasiswa dapat berupa pilihan ganda dan esai yang berupa kode program. Rancangan konten dari aplikasi ini diilustrasikan pada Gambar 2 berikut.



Gambar 2. Desain Ilustrasi Aplikasi *learning by doing* Interaktif

3.3. Rancangan Sistem

1) Rancangan User dan fitur

Berikut ini adalah fitur yang ada dalam masing-masing user. User meliputi: Dosen dan Mahasiswa

Dosen, memiliki wewenang dalam :

1. *Create/modify* pokok bahasan
2. *Create / modify* sub pokok bahasan
3. *Create Group*
4. *Close / Archive Group*
5. *Monitoring Group*
6. *Question Bank, CBT*
7. *Open CBT for specific Group*

Mahasiswa, memiliki wewenang dalam :

1. *Register as Student*
2. *Join to Group*
3. *Access Material*
4. *Access Live Code*
5. *Access CBT*

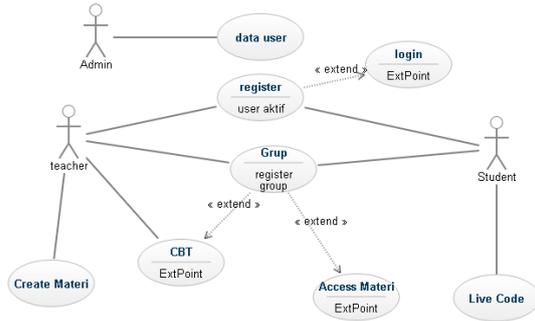
2) Struktur Database

Struktur *database* yang akan ada dalam aplikasi ini meliputi:

1. Pokok Bahasan, meliputi: Kode, Judul, Uraian, *URL*, *dropboxURL*
2. Sub Pokok Bahasan, meliputi:
 - a. Kode, Judul
 - b. IDPokokbahasan(parent)
 - c. Uraian, *URL*, *dropboxURL*
3. *Group*, meliputi : Kode, Nama *Group*, status *Group*(*open,close,archive*)
4. *Student*: Kode, Nama, email, nohp, alamat, keterangan
5. *Student activity*: tgl, log
6. *Question Bank*: id, question, ans1, ans2, ans3, and ans4

3) Use Case Diagram

Rancangan *Use Case Diagram* dari aplikasi ini adalah seperti Gambar 3 berikut.



Gambar 3. Rancangan *Use Case Diagram*

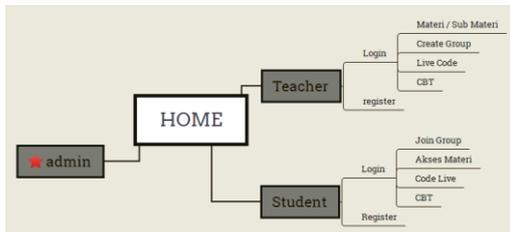
Pada rancangan aplikasi yang dibangun, ada *user* Admin yang berhak melakukan pengelolaan data *user*. Selanjutnya *user* Dosen, berhak melakukan pembuatan kelas baru, aktivasi *user* mahasiswa, membuat materi, menyiapkan soal untuk CBT. *User* mahasiswa bisa akses materi, dan bisa melakukan *live code*.

4) Site Map Aplikasi

Site Map dari aplikasi ini memiliki struktur diagram seperti yang ditunjukkan pada gambar 4. Dari halaman depan, ada menu untuk masuk ke halaman admin, *Teacher* (Dosen), dan Student (Mahasiswa).

Dari halaman *Teacher*, ada halaman untuk Login dan register. Setelah login, *teacher* bisa melakukan akses materi dan sub materi, membuat grup pembelajaran, akses *live code*, dan CBT.

Dari halaman *Student*, setelah login, *student* bisa melakukan *Join Group*, akses materi, mencoba *live code*, dan melakukan CBT.



Gambar 4. Struktur Site Map Aplikasi

Sedangkan Fitur Utama Aplikasi ini, meliputi :

- a. **Materi**, bahasan dan sub pokok bahasan dari materi kuliah.
- b. **Live Coding**, Fitur untuk menulis kode program serta dapat menjalankan dalam *browser*. Kode program ini dapat disimpan, di-download atau di-share kepada *user* yang lain (melalui : *twitter*, *facebook*, *google+*, *email* dan *link*). Termasuk jika mahasiswa ingin mengirimkan kode program kepada dosen, misal untuk demo program atau ujian.
- c. **CBT**, fitur *Computer Based Test*. merupakan tes berbasis komputer teknik penyampaian

soal yang tidak lagi menggunakan kertas (*paperless*), baik untuk naskah soal maupun lembar jawaban. Sistem skoring atau koreksi langsung dilakukan oleh komputer. Mahasiswa dapat mengakses fitur CBT yang telah dibuat oleh dosen.

5) Struktur MVC

Struktur MVC dari aplikasi ini adalah seperti yang diuraikan dalam Tabel 1.

MVC	Jenis	Keterangan
Model	M_admin	Untuk akses tabel
	M_grup	Untuk akses tabel
	M_group_member	Untuk akses tabel
	M_materi	Untuk akses tabel
	M_student	Untuk akses tabel
	M_student_log	Untuk akses tabel
	M_subMateri	Untuk akses tabel
	M_teacher	Untuk akses tabel
Controller	C_admin	Berisi fungsi untuk user admin
	C_group	Berisi fungsi untuk mengelola grup
	C_group_member	Berisi fungsi untuk grup member
	C_landing_page	Berisi fungsi sebagai halaman awal aplikasi
	C_materi	Berisi fungsi untuk mengelola data materi
	C_student	Berisi fungsi untuk user student
	C_subMateri	Berisi fungsi untuk mengelola sub materi
C_teacher	Berisi fungsi untuk user teacher	
View		View secara umum digunakan untuk menyajikan data ke user.

6) Rancangan Fitur *Live Coding*

Sedangkan implementasi menu *Live Coding*, fitur ini dibangun menggunakan *Trinket*. *Link* untuk *embed Trinket* tersebut diletakan pada web aplikasi ini. Berikut ini adalah implementasi link dari aplikasi *Trinket*.

- Untuk tipe script :

```
<iframe src="https://trinket.io/embed/python/bfff64c8c5" width="100%" height="356" frameborder="0" marginwidth="0" marginheight="0" allowfullscreen></iframe>
```
- Untuk tipe blocks :

```
<iframe src="https://trinket.io/embed/blocks/4645538d17" width="100%" height="356" frameborder="0" marginwidth="0" marginheight="0" allowfullscreen></iframe>
```

```
marginheight="0"
allowfullscreen></iframe>
```

4. HASIL DAN PEMBAHASAN

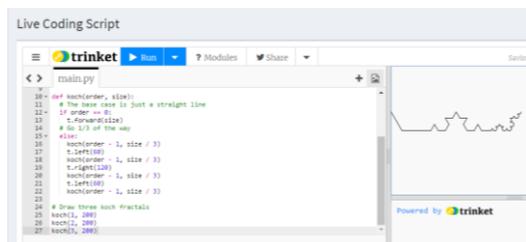
Bagian ini adalah membahas tentang bagaimana hasil uji coba yang telah dilakukan setelah aplikasi di hosting di internet. Ujicoba dilakukan dengan 2 tahap, yakni pengujian secara fungsionalitas sistem dan ujicoba hasil pembelajaran menggunakan media aplikasi ini dengan mata kuliah Algoritma Pemrograman.

5.1. Uji Coba Fungsionalitas Sistem

Menu Utama yang akan diujicoba adalah Menu Live Coding. Menu ini terdapat pada pengguna *teacher* (dosen) dan *student* (mahasiswa) yang fungsinya digunakan untuk menulis kode program, menjalankan program dan share kode program kepada pengguna lain, hal ini sangat sesuai untuk pembelajaran di kelas. Seperti dosen untuk menunjukkan kode program tertentu atau mahasiswa ingin demo program terhadap dosennya. Ada 2 jenis tipe yang digunakan dalam fitur ini, berbentuk script dan berbentuk blok yang keduanya berbasis bahasa pemrograman *Python*. Bentuk *script*, seperti *Idle python* pada umumnya, ada tempat untuk menuliskan *script* dan ada tempat untuk melihat output program. Sedangkan berbentuk blok, maka kode program tampil dalam bentuk blok, tapi masih bisa dilihat *script*-nya.

1) Tipe Script

Gambar 5 adalah contoh penggunaan *Trinket* yang berbasis *script*. Menggambar fraktal dengan menggunakan fungsi rekursif.



Gambar 5. Contoh penggunaan fungsi rekursif

2) Tipe Blocks

Gambar 6 adalah contoh penggunaan *trinket* yang berbasis blok. Membuat gambar bunga dengan menggunakan fungsi dan pen.



Gambar 6. Contoh pemrograman untuk grafis

5.2. Uji Coba Hasil Pembelajaran

Aplikasi *learning by doing* interaktif ini telah diujicobakan ke beberapa mahasiswa yang sedang mengambil mata kuliah Algoritma dan Pemrograman. Ada 4 kelas, masing-masing kelas terisi 40 orang mahasiswa dengan 2 orang dosen sebagai pengajar mata kuliah tersebut. Sebagai perbandingan 2 kelas menggunakan aplikasi ini dan 2 kelas yang lain tidak (menggunakan editor idle sebagai *tools* pemrograman *python*). Tabel 2 berikut adalah hasil rekapitulasi ujicoba terhadap penggunaan aplikasi ini.

Tabel 2. Rekapitulasi Hasil Uji Coba

Aspek	Deskripsi	Media Pembelajaran			
		Menggunakan aplikasi		Tanpa aplikasi	
		B	D	A	C
Keaktifan dalam pembelajaran	Mahasiswa aktif mengikuti pembelajaran	78	84	71	80
	Mahasiswa aktif dalam mengerjakan tugas	82	79	74	79
	Mahasiswa belajar mandiri	77	76	69	75
Rata ² (dalam persen)		79	79,67	71,3	78
Rata ² Keaktifan dalam pembelajaran	(dalam persen)	79,335		74,65	
Peningkatan Kemampuan	Hasil evaluasi tugas yang diberikan (Mendapatkan nilai Baik)	54	53	31	51
	Hasil evaluasi ujian (Mendapatkan nilai Baik)	47	67	34	43
	Hasil evaluasi praktikum (Mendapatkan nilai Baik)	60	71	55	53
Rata ² (dalam persen)		53,6	63,67	40	49
Rata ² Peningkatan Kemampuan	(dalam persen)	58,635		44,5	

Hasil analisa dari survei evaluasi pembelajaran menunjukkan bahwa penggunaan aplikasi ini mampu membuat ketertarikan terhadap belajar pemrograman sebesar 79%. Hal ini karena ditunjang visualisasi dari media yang membuat mahasiswa untuk tertarik belajar secara mandiri. Dan hasil evaluasi pembelajaran menunjukkan peningkatan yakni selisih 14% dibandingkan dengan pembelajaran biasa. Hal ini karena mahasiswa dapat melakukan pembelajaran secara berulang, pembelajaran mandiri yang dibantu dengan tutorial. Sehingga sangat sesuai bagi pengenalan awal untuk mahasiswa semester

baru dalam belajar bahasa pemrograman pada mata kuliah Algoritma Pemrograman.

5. KESIMPULAN

Aplikasi *learning by doing* Interaktif yang diusulkan dalam penelitian ini terbukti dapat mendukung pembelajaran bahasa pemrograman salah satunya untuk mata kuliah Algoritma Pemrograman. Hasil pengujian didapatkan kesimpulan bahwa aplikasi ini mampu meningkatkan ketertarikan mahasiswa dalam belajar bahasa pemrograman dan meningkatkan kemampuan mahasiswa dalam pemrograman.

Aplikasi ini dapat memfasilitasi untuk menampung materi sesuai dengan GBPP dan SAP dan bisa dimanfaatkan untuk mata kuliah lain yang berisi konten pemrograman yang menggunakan bahasa pemrograman Python. Pada aplikasi ini juga memungkinkan mahasiswa bisa melakukan belajar secara mandiri karena ada respon balik dari sistem secara langsung. Serta dimungkinkan untuk melakukan pembelajaran secara berulang-ulang dilain waktu. Selain itu aplikasi ini bisa digunakan untuk demonstrasi kode program dan hasil *run out* dari kode program yang dituliskan.

6. REFERENSI

- ADŽGAUSKIENE, D. V & ANTANAS V. U. 2012. Problems in Choosing Tools and Methods for Teaching Programming, *Informatics in Education*, 11(2), 271–282
- BEDNARICK, R., MORENO, A., MYLLER, N., & SUTINEN, E. 2005. Smart Program Visualization Technologies: Planning A Next Step. *Proceedings Of The Fifth IEEE International Conference On Advanced Learning Technologies (Icalt'05)*.
- CARBONNELLE, P. 2016. PYPL Popularity of Programming Language, <http://Pypl.Github.Io/Pypl.Html>
- DORN, B. & SANDERS, D. 2003. Jeroo: A Tool For Introducing Object-Oriented Programming, *ACM SIGCSE'03*, 19-23 February.
- HULALOOP. 2017. <https://Pythonroom.Com/>
- KASURINEN, J., MIKA, P. & UOLEVI, N. 2008., A Study Of Visualization In Introductory Programming, PPIG, Lancaster
- KELLEHER, C. & PAUSCHY, R. 2006. Lessons Learned From Designing A Programming System To Support Middle School Girls Creating Animated Stories, *Proceedings Of The Visual Languages And Human-Centric Computing IEEE Computer Society Washington, DC, USA*.
- LAHTINEN, E., MUTKA K. A., & JÄRVINEN H.M. 2005. A study of the difficulties of novice programmers, *Proceeding Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05)*, 14-18
- MA, K., TENG, H., DU, L., ZHANG, K. 2014, Project-Driven Learning-by-Doing Method for Teaching Software Engineering Using Virtualization Technology, *International Journal of Emerging Technologies in Learning*, 9(9).
- NAN, I., KAU, B., & RUGELJ, J. 2008. Pair Programming as a Modern Method of Teaching Computer Science *International Journal of Emerging Technologies in Learning*, Vol 3
- PICH, C., N, LEV & GEORGE, G. 2008. Visualization Of Exception Handling Constructs To Support Program Understanding, *Softvis Herrsching Am Ammersee*, 16–17 September. Germany.
- SHEHANE, R. & SHERMAN, S. 2014. Visual teaching model for introducing programming languages, *Journal of Instructional Pedagogy*, Vol 14
- STEPHEN, M., FRANKLIN, W., ELIZABETH, A., JUMA, K., PATRICK O. 2011. Teaching Computer Programming in the 21st Century *International Journal of Science and Technology*, 1(6), 247-252
- VIHAVAINEN, A. & PAKSULA, M. , LUUKKAINEN, M. 2011. Extreme apprenticeship method in teaching programming for beginners, *Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11)* , 93-98
- XIA, B.S. 2017, An In-depth Analysis of Learning Goals in Higher Education: Evidence from the Programming Education, *Journal of Learning Design*
- YEHUDA, Z. E. O. 2013. *Programming With Codeigniter MVC*. Birmingham: Packt Publishing LTD

Halaman ini sengaja dikosongkan