



Studi Komparatif Program Visual Dinamis untuk Pembelajaran Algoritma dan Pemrograman Berorientasi Objek

Kursehi Falgenti

Prodi Sistem Informasi, STMIK Nusamandiri Jakarta
falgenti.kfe@nusamandiri.ac.id

Abstract

As beginners, many first-year students have difficulty understanding object-oriented programming material. To help students learn algorithmic and object-oriented programming material researchers have developed visual programming (PV). Visual programming is a tool to facilitate learning programming. The concept of learning to use PV visualizes the work processes of algorithms and programming. This research aims to compare three dynamic PV tools for object-oriented learning programming that are the most studied. To determine the PV to be compared, a survey was conducted in an online journal database, such as IEEE explore, ACM, and several well-known online publishers. From the survey results, three dynamic PVs were chosen, most widely discussed, namely Jeliot 3, Ville and Jive. All three tools are installed and studied. Comparison results show that each dynamic PV has advantages on certain characteristics. The instructor can choose visual programming by considering the advantages of each PV.

Keywords: algorithms and programming, object oriented programming, visual programming

Abstrak

Sebagai pemula, banyak mahasiswa tingkat pertama kesulitan memahami materi pemrograman berorientasi objek. Untuk membantu mahasiswa mempelajari materi algoritma dan pemrograman berorientasi objek peneliti telah mengembangkan pemrograman visual (PV). Pemrograman visual merupakan sebuah tools untuk memudahkan pembelajaran pemrograman. Konsep belajar menggunakan PV memvisualisasikan proses kerja algoritma dan pemrograman. Penilitin ini bertujuan melakukan komparasi tiga tools PV dinamis untuk pembelajaran pemrograman berorientasi objek yang paling banyak diteliti. Untuk menentukan PV yang akan dikomparasi dilakukan survey di di database jurnal online, seperti IEEE explore, ACM, dan beberapa publisher online terkenal. Dari hasil survey dipilih tiga PV dinamis yang palng banyak dibahas yaitu Jeliot 3, Ville dan Jive. Ketiga tools di instal dan dipelajari. Hasil komparasi menunjukkan masing-masing PV dinamis memiliki keunggulan pada karakteristik tertentu. Instruktur dapat memilih pemrograman visual dengan dengan mempertimbangkan keunggulan masing-masing PV.

Kata kunci: algoritma dan pemrograman, pemrograman berorientasi objek, pemrograman visual

1. Pendahuluan

Pemrograman adalah topik utama yang diajarkan kepada mahasiswa di fakultas ilmu komputer. Tahun pertama kuliah mereka mendapatkan matakuliah algoritma dan pemrograman. Matakuliah ini dapat meningkatkan pemahaman konsep dasar pemrograman dan meningkatkan pengetahuan komputasi dan keterampilan pemecahan masalah [1]. Mereka mempelajari paradigma algoritma dan pemrograman berorientasi prosedur dilanjutkan dengan paradigma pemrograman beroreientasi objek. Istilah dalam pemrograman berorientasi objek lebih sulit dipelajari bila mahasiswa tidak memahami pemrograman berorientasi prosedur dengan baik [2]. Pada pemrograman berorientasi objek aliran kontrol dan fungsinya terdistribusi. Mahasiswa pemula sulit membentuk representasi mental aliran kontrol dan fungsi pada pemrograman berorientasi objek

[3]. Selain itu mereka butuh waktu lama memahami paradigma pemrograman berorientasi objek karena kurva pembelajaran pemrograman berorientasi objek lebih panjang dan karakter program berorientasi objek lebih luas pembahasannya.

sebagian besar siswa adalah pembelajar visual, disisi lain 75% dan 83% proses pembelajaran dan bahwdalam proses pembelajaran materi dan informasi disampaikan instruktur insecara verbal[4]

Peneliti terus berusaha meningkatkan efektivitas pengenalan pemrograman berorientasi objek. Beberapa peneliti mengembangkan tool untuk pembelajaran pengenalan pemrograman, salah satunya menggunakan pemrograman visual (PV). Pemrograman Visual atau Bahasa Pemrograman Visual adalah bahasa yang terdiri dari objek berwujud, mewakili logika diskrit, dan konsep komputasi. Bahasa ini disusun oleh bidang dua dimensi

atau blok grafik, yang saling terkait dengan beberapa deskripsi tekstual dan satu set port input dan output. Beberapa PV telah dibuat untuk mendukung pembelajaran pemrograman berorientasi objek diantaranya; BlueJ [5], [6], [7], Jeliot [8], Jive [9], Ville [10] merupakan PV untuk pembelajaran Java. SFC (Structured Flow Chart) Editor [11] dan BACCII++ [12] merupakan PV untuk pembelajaran pemrograman C++. Beberapa PV memiliki tujuan spesifik, seperti; App Inventor 2 [13] untuk pemrograman mobil pada sistem operasi android. Scratch [14] lebih spesifik untuk anak-anak usia 8-16 tahun. Beberapa PV memngkominasikan multimedia dalam pembelajaran seperti Alice [28] dan Visa[29]. Kodu [30] dirancang khusus untuk pengembangan game dan menawarkan prinsip-prinsip khusus yang berasal dari skenario game. Masing-masing PV memiliki karakteristik yang berbeda. Karakteristik-karakteristik yang dimiliki masing-masing PV perlu dipelajari untuk memilih pemrograman visual terbaik bagi mahasiswa pemula.

PV telah digunakan di negara maju untuk membantu mahasiswa pemula dalam meningkatkan keterampilan pemrograman. PV memudahkan mahasiswa membangun model mental sehingga cepat memahami eksekusi program maupun algoritma [15]. PV membantu mahasiswa yang memiliki rasa percaya diri rendah dalam proses pembelajaran. Mahasiswa yang memiliki rasa percaya diri rendah memiliki peningkatan kemampuan yang lebih tinggi dalam analisis konsep pemrograman dasar setelah belajar dengan dukungan PV [1]. Mahasiswa yang memiliki rasa rendah diri lebih banyak ditemukan di negara berkembang. Menurut Apiola and Tedre [16] dalam pelajaran pemrograman mahasiswa-mahasiswa dinegara berkembang memiliki kemampuan belajar yang rendah sehingga mereka memiliki pengetahuan pemrograman yang dangkal. Karena itu perlu usaha kreatif untuk meningkatkan keterampilan dan pengetahuan pemrograman mahasiswa dinegara berkembang. Salah satu usaha kreatif tersebut adalah menggunakan PV sebagai tool pendukung dalam mata kuliah algoritma dan pemrograman khususnya dalam pembelajaran pemrograman berorientasi objek.

Penelitian ini bertujuan membantu instruktur menemukan PV terbaik untuk mendukung pembelajaran algoritma dan pemrograman bagi mahasiswa pemula. Karakteristik masing-masing PV dievaluasi dan dikomparasi. Hasil komparasi merupakan karakteristik-karakteristik unggulan pada masing-masing PV. Hasil komparasi dapat menjadi pertimbangan instruktur dalam memilih PV yang cocok dengan lingkungan belajarnya

2. Metode Penelitian

Penelitian ini adalah studi komparatif tiga PV dinamis, Jeliot 3, Ville dan Jive. Tahapan penelitian terdiri dari; pertama, melakukan studi tentang topik PV dari literatur-literatur jurnal di database jurnal online, seperti IEEE explore, ACM, dan beberapa publisher online terkenal. Kedua, Memilih tiga PV yang banyak diteliti

dan dipublikasikan. Dari literatur ditemukan PV online berbasis web, PV desktop diinstall pa PC dan V untuk perangkat mobil. PV yang dipilih adalah PV dalam bentuk aplikasi desktop. Aplikasi yang bisa dipakai dalam kelas mendukung kegiatan belajar mengajar. Ketiga, PV dinamis yang diilih di-install dan dipelajari dan melakukan komparasi untuk menemukan keunggulan masing-masing PV.

3. Hasil dan Pembahasan

Sebagian besar mahasiswa pemula kesulitan memahami bagaimana sebuah program bekerja ketika kode program yang ditulis di eksekusi, penyebabnya menurut [8] mahasiswa kesulitan membangun mind map. Ketika mempelajari dasar program, mahasiswa pemula belajar banyak istilah baru. Mempelajari istilah baru membingungkan pemula [17]. Mereka memiliki kemampuan terbatas mengerjakan tugas-tugas dasar seperti memahami eksekusi program. Selain itu mereka juga bermasalah dengan perencanaan dan penerapan algoritma.

Paradigma berorientasi prosedur dan paradigm berorientasi objek merupakan dua pendekatan yang diajarkan kepada pemula dalam pelajaran algoritma dan pemrograman. Kedua paradigm memiliki karakter yang berbeda. Pemrograman berorientasi objek lebih sulit dipelajari karena banyak istilah baru yang jauh berbeda dengan istilah pada pemrograman berorientasi prosedur. Teori yang harus dipelajari untuk memahami pemrograman berorientasi objek lebih banyak. Sehingga membutuhkan waktu yang lama untuk menguasainya.

Java adalah salah satu Bahasa pemrograman berorientasi objek yang paling banyak digunakan di kampus [4], [18]. Java banyak digunakan karena free. Agar pembelajaran pemrograman berorientasi objek menggunakan Java lebih efektif peneliti mengembangkan metode pengajaran dengan dukungan pemrograman visual dalam penyampaian materi. Banyak tool pemrograman visual yang telah dibuat dengan tujuan yang berbeda. Ada yang hanya fokus untuk memvisualisasikan algoritma ada juga yang juga berfungsi sebagai tool untuk memvisualisasikan pemrograman.

Bahasa pemrograman klasik identik dengan pemrograman dengan teks, luaranya berupa makna yang berasal dari urutan linier elemen leksikal. Sedangkan pemrograman visual terdiri dari elemen teks dan grafis. Banyak bahasa PV hanya memungkinkan pengguna secara visual membangun pohon sintaks abstrak dari program teks.

Sistem visual memiliki sifat ekspresif dan efektif [19]. Sistem ekspresif menunjukkan jenis visualisasi apa yang dapat dihasilkan dengan sistem, dan juga metode yang ditentukan oleh sistem untuk menghasilkan dan berinteraksi dengan visualisasi tersebut. Sedangkan sistem yang efektif didefinisikan sebagai sejauh mana sistem menyediakan fungsionalitas yang benar-benar ingin digunakan oleh orang (kegunaan) dan juga sejauh

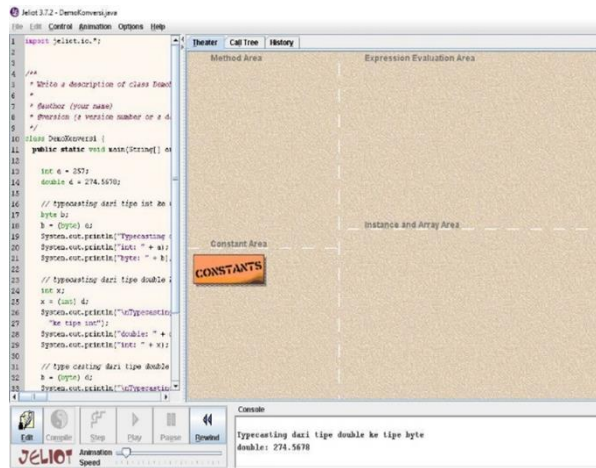
mana orang dapat Manfaatkan fungsionalitas itu. Dengan dukungan animasi dalam PV, pelajar mendapatkan proses yang lebih elaboratif (integrasi dengan pengetahuan sebelumnya) dan berdampak baik untuk pembelajaran yang lebih baik [20].

Salah satu PV yang banyak digunakan adalah BlueJ. BlueJ dirancang dan dikembangkan untuk tujuan pendidikan. BlueJ dikembangkan tahun 1999 oleh Michael Kölling and John Rosenberg dari Universitas Monash. Bulan Maret 2009 proyek BlueJ menjadi free dan menjadi software open source di bawah lisensi GNU GPL dengan pengecualian pada classpath exception. BlueJ adalah tool pemrograman visual yang bersifat statis. fokus PV statis pada pemformatan struktur program dan memvisualkan hubungan antara komponen program. BlueJ memiliki tampilan kelas yang menunjukkan hubungan antara kelas dan objek yang berisi semua objek yang diinisialisasi. Selain BlueJ. Pemrograman visual berorientasi objek untuk bahasa python adalah Python Tutor[27]. Program visual ini juga memiliki fungsi maju dan mundur satu eksekusi untuk memvisualisasikan kondisi run-time dari struktur data program. Dengan menggunakan fitur-fitur ini, mahasiswa dapat men-debug program dapat melakukan perbaikan bug dan mendapatkan solusi yang benar.

Pemrograman visual dinamis memvisualisasikan pelaksanaan program secara bertahap, pengguna dapat melihat tahapan-tahapan proses ketika baris program dieksekusi. Jeliot 3, JIVE dan VILLE adalah PV yang bersifat dinamis. Ketiganya, memiliki tujuan dasar dan beberapa fitur yang sama. Namun, perbedaan ada pada tingkat abstraksi dari visualisasi.

3.1 Jeliot 3

Jeliot telah melewati banyak tahap perkembangan, Mulai dari sistem pendahulunya Eliot dan terus dikembangkan menjadi Jeliot I, Jeliot 2000 dan versi terbaru Jeliot 3. Versi Jeliot 3 diimplementasikan di Universitas Joensuu Oleh Niko Myller dan Andre's Moreno di bawah supervisi Dari Mordechai Ben-Ari dan Erkki Sutinen. Eliot and Jeliot I lebih mirip seperti algorithma visualization, sedangkan Jeliot 2000 dan Jeliot 3 merupakan pemrograman visual yang lebih interaktif dibandingkan pendahulunya karena mempresentasikan proses eksekusi program dengan animasi.



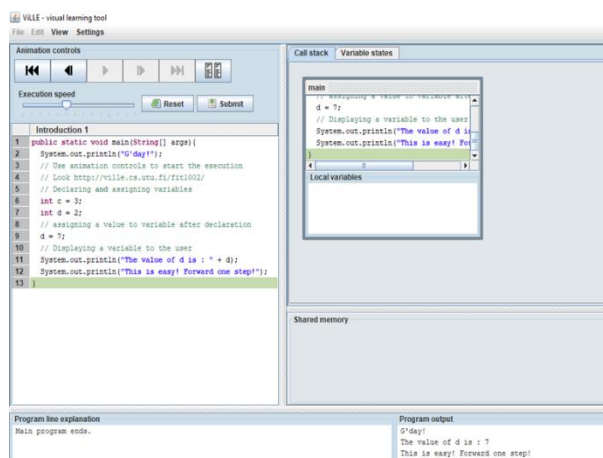
Gambar. 1 Tampilan antar muka Jeliot 3 versi desktop

Bentuk antar muka Jeliot 3 dapat dilihat pada gambar 1. Bagian kiri merupakan list program, sementara ditengah terdiri dari 4 bagian yaitu method area, expression evaluation area, constanta area dan instance and array area. Masing-masing area akan aktif memvisualisasikan apa saja yang terjadi saat program dieksekusi. Sedangkan bagian kanan bawah merupakan console yang menampilkan output.

Jeliot saat ini mengembangkan versi module Moodle dan Jeliot Mobil dengan fungsi yang terbatas. Pada versi mobil pengguna hanya bisa mempelajari dasar-dasar pemrograman java menggunakan perangkat mobile.

3.2 Ville

Ville dikembangkan oleh Teemu Rajala, Mikko-Jussi Laakso, Erkki Kaila, dan Tapio Salakoski dari University of Turku, Finlandia. Berbeda dengan tool pemrograman visual lainnya yang hanya bisa menggunakan Java, Ville dibangun dari syntax editor, dengan pendekatan ini pengguna bisa menambahkan Bahasa baru ke dalam tool VILLE. Sampai saat ini Ville bisa digunakan untuk Bahasa berorientasi objek seperti Java, C++ Python, PHP dan bahasa Pseudo.



Gambar. 2. Tampilan antar muka Ville

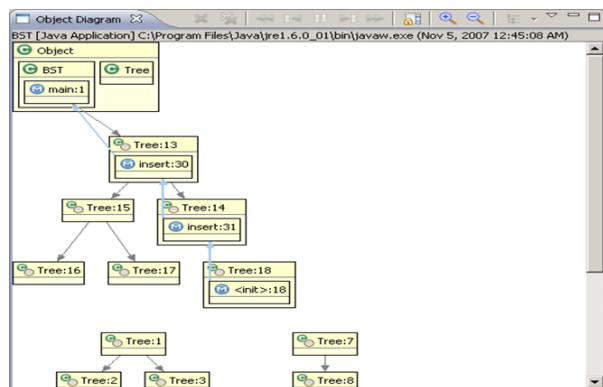
Penjelasan bagian-bagian dari antarmuka Ville pada gambar 2. Pada bagian kanan atas merupakan pengatur animasi, dibawahnya adalah listing program. Bagian paling kiri bawah merupakan penjelasan line program yang sedang dieksekusi. Pada bagian kanan atas merupakan call stack yang menandai bagaian line program yg dieksekusi. Kotak dibawahnya menunjukkan varaiabel local yang sedang digunakan. Dipojok kanan bawah merupakan output program.

Ville sebagai VP kini dikembangkan menjadi Colaborative Learning evirontment [21]. Ville versi colaboratif learning menjadi aplikasi klien server. Pada versi terberu ini vllle menyediakan metode pembejaran pair proraming dan per review. Versi *Colaborative lerning* tidak hanya untuk pembelajaran pemograman berorientasi objek tapi juga dikembangkan untuk pembelajaran matematika untuk pendidikan dasar.

3.3 Jive

JIVEe dikembangkan oleh Paul Gestwicki, Bharat Jayaraman dari Department of Computer Science and Engineering University at Buffalo Amerika. JIVE merupakan pendekatan baru untuk memvisualisasikan keadaan eksekusi program berorientasi objek khususnya Java. Jive dirancang sebagai alat eksekusi visual interaktif untuk Java. Hal ini memungkinkan eksekusi interaktif dari program pengguna, memberikan tampilan grafis dari keadaan eksekusi dengan cara yang jelas. tampilan jive ada dua jenis; objek diagram view dan sequence diagram. Ville merupakan plug in dari IDE eclipse. Jive dipasang sebagai tool pendukung pada IDE Eclipse.

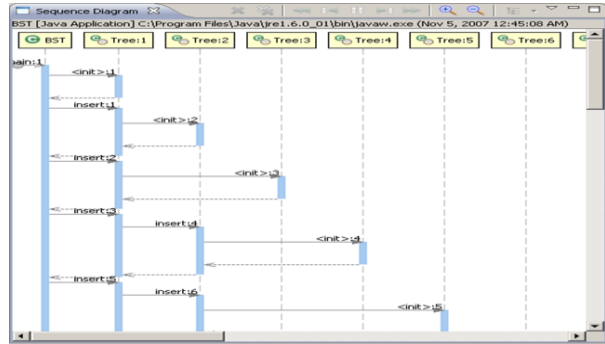
Antar muka Jive dapat dilihat pada gambar 3 dan 4. Jive Menampilkan objek-objek yang aktif saat program dieksekusi, selain itu juga menampilkan diagram yang merepresentasikan referensi objek. Aktivasi method juga muncul dalam diagram pada objek yang sesuai atau konteks statis. Koneksi berwarna antara aktivasi method mewakili tautan kembali. Setiap thread diberi warna sendiri seperti pada gambar 3.



Gambar. 3 Tampilan Tampilan objek diagram pada Jive antar muka Jive

Pada gambar 4 dapat dilihat sequence diagram yang menggambarkan eksekusi suatu program. Objek dan

konteks statis tercantum di bagian atas diagram. Aktivasi method muncul sebagai persegi panjang yang muncul secara vertikal yang ditempatkan di bawah objek atau konteks pelaksanaan statis. Seperti halnya diagram objek, setiap thread diberi warna sendiri. Koneksi solid dalam diagram mewakili pemanggilan method, dan koneksi putus-putus merepresentasikan pengembalian method.



Gambar 4. Tampilan antar muka Sequence diagram pada Jive Tampilan antar muka Jive

Dengan penerapan pemograman visual diharapkan dapat membantu instruktur menyampaikan materi pelajaran pemograman lebih efektif. Interaksi antara instruktur dan mahasiswa diharapkan dapat lebih ekspresif. Mahasiswa dapat menyerap materi pemograman berorientasi objek dengan lebih baik sehingga dapat meningkatkan proses pembelajaran.

3.4 Komparasi ville, Jeliot 3 dan Jive

Ketiga tool PV dinamis ini dirancang untuk pengajaran bahasa pemograman berorientasi objek untuk pemula. Beberapa studi telah melakukan komparasi PV seperti Studi [10] lebih spesifik membandngkan karakteristik beberapa studi Jeliot 3, Vlille dan Jive dan Studi [25] lebih banakPV. Dari hasil studi diatas dan analisa masing-masing PV dengan menginstal dan mempelajari masing-masing PV I komputer pribadi dapat dikelompokkan keunggulan masing-masing PV seperti yang ditampilkan di tabel 1, tabel 2 dan dan tabel 3.

Tabel 1. Keunggulan Ville

PV	Keunggulan
Ville	Dapat menjalankan beragam bahasa pemograman berorientasi objek seperti Java, C++ Syntax bisa diedit Tersedia contoh-contoh soal dengan deskripsinya Contoh soal dapat dipublikasikan Kontrol visualisasi bisa si stop dan dimundurkan Tersedia penjelasan baris-baris program Tersedia informasi peran dari variabel Evaluasi ekspresi secara verbal

Karakteristik Ville yang dapat menjalankan berbagai jenis bahasa pemograman berorientasi objek dan memiliki kompiler yang dibangun sendiri adalah karakteristik pembeda Ville dengan VP lainnya. Sedangkan karakteristi yang tidak dimiliki oleh Ville

tidak menyediakan interface bagi user memberikan input saat eksekusi program. Ville memiliki karakteristik unggul lebih banyak diantara PV yang di pelajari. Tersedianya contoh soal memudahkan instruktur mengevaluais kemampuan mahasiswa. Contoh-contoh soal yang disediakan dapat diedit, memudahkan instruktur membuat variasi soal.

Tabel 2 Keunggulan Jeliot 3

PV	Keunggulan
Jeliot 3	Evaluasi ekspresinya dipresentasikan menggunakan grafik Interaksi dengan user lebih interaktif dengan fasilitas input dari user saat eksekusi program Contoh program bisa dibuat user dan bisa disimpan

Karakteristik utama Jeliot 3 yang menjadi pembeda dengan VP lain adalah evaluasi ekspresinya ditampilkan dalam bentuk grafik dan menyediakan pilihan input oleh user saat eksekusi.

Tabel 3 Keunggulan Jive

PV	Keunggulan
Jive	Tersedia kontrol visualisasi eksekusi program dapat menghentikan proses eksekusi untuk mempelajari langkah-langkah yang sudah dieksekusi Algoritma program dipresentasikan dalam bentuk grafik Kontrol visualisasi dinamis, bisa digeser maju dan mundur

Karakteristik utama Jive yang menjadi pembeda dengan PV lain adalah presentasi algoritma dalam bentuk grafik, fungsi ini akan lebih memudahkan pengguna memahami algoritma dalam baris kode. Interaksi Jive dengan IDE Java Eclipse memudahkan penggunaan PV ini tanpa harus melakukan instalasi dari awal.

Studi tentang keuntungan penerapan pemrograman visual di dibandingkan dengan dengan metode konvensional dalam proses belajar mengajar di kampus-kampus negara maju seperti studi Boyle [22] menyatakan pendekatan grafik dalam pengenalan pemrograman berperan terhadap mahasiswa dalam transisi kurikulum di London Metropolitan University. Studi [23] mengevaluais penggunaan Jeliot 3 di Universitas Joensuu Finlandia, hasil studi menunjukkan Jeliot 3 meningkatkan kemampuan para pemula untuk memahami pernyataan if else dan loop, memahami objek, dan menelusuri kesalahan dari kode program. Studi terhadap versi mobile jeliot [26] Studi untuk melihat efektivitas penggunaan pemrograman visual dinamis Ville di Universitas Turku Finlandia dilakukan oleh [24]. Dari hasil studinya dapat disimpulkan pemrograman visual dapat meningkatkan pembelajaran siswa. Studi oleh [25] menunjukkan pemrograman visual memberi dampak positif terhadap pembelajaran pengenalan program.

4. Kesimpulan

Berdasarkan fungsinya, ketiga PV yang telah dibahas memiliki kemampuan visualisasi yang hampir sama. Pembedanya adalah kemampuan masing-masing PV memvisualisasikan jenis bahasa pemrograman berorientasi objek yang berbeda. Ville memiliki keunggulan dalam menampilkan eksekusi program yang ditulis dengan bahasa Java, C++ Python, PHP dan bahasa Pseudo, sedangkan jeliot dan Jive hanya mampu menampilkan eksekusi program yang ditulis dengan Bahasa Java. Keunggulan Ville yang lain adalah adanya materi pemrograman berorientasi objek dalam aplikasi, berupa list program sederhana yang dapat digunakan untuk pembelajaran algoritma dan pemrograman berorientasi objek. Instruktur juga dapat menambahkan materi praktikum yang mereka anggap penting ke dalam Ville

Jeliot 3 memiliki keunggulan dalam menampilkan grafik memvisualisasikan evaluasi ekspresinya dengan dukungan animasi. Mahasiswa lebih mudah memahami proses eksekusi program dengan dukungan visualisasi dalam bentuk animasi. Sedangkan Jive memiliki keunggulan dalam memvisualisasikan algoritma, fungsi ini sangat penting bagi mahasiswa pemula memahami algoritma yang bekerja saat program dieksekusi, membantu membentuk representasi mental dari baris-baris kode yang mereka tulis.

Dari keunggulan karakteristik masing-masing PV instruktur dapat memilih PV yang cocok dengan lingkungan dimana mereka memberikan pengajaran. Memandu mahasiswa pemula memahami lebih mudah proses eksekusi program berorientasi objek

Daftar Rujukan

- [1] C. Tsai., "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy", *Computers in Human Behavior* 95 224-232, 2019.
- [2] K. C. Thramboulidis, "A Sequence of Assignments to Teach Object-Oriented Programming: a Constructivism Design-First Approach" *Informatics in Education - An International Journal* Vol.2 No.1 p 103-112, 2013.
- [3] S. Wiedenbeck., V. Ramalingam., Suseela. S., and Cynthia. C. "A comparison of the comprehension of object-oriented and procedural programs by novice programmers". *Interacting with Computer* Vol 11 No.3 1999.
- [4] M.J. Kirschner, 2015. *Visual programming in three dimensions: visual representations of computational mental models.* Massachusetts Institute of Technology.
- [5] .
- [6] M. Kolling., & J. Rosenberg, "Objects first with Java and BlueJ" (seminar session). *ACM SIGCSE Bulletin*, 32(1) 2000. 429.
- [7] M. Kolling., & J. Rosenberg. "Guidelines for teaching object orientation with Java" *ACM SIGCSE Bulletin*, 33(3), 2001, 33 – 36.
- [8] M. Kolling., B. Quig., A. Patterson., & J. Rosenberg. "The BlueJ system and its pedagogy" *Computer Science Education*, 13(4), 2003 249 – 268.
- [9] B. Levy., M. Ben-Ari dan P.A. Uronen., "The Jeliot 2000 program animation system." *Computers & Education* 40 (1) 2003,15-21

- [10] P. Gestwicki. and B. Jayaraman. "Interactive visualization of Java programs" In Proceedings of Symposia on Human Centric Computing Languages and Environments. 2002, 226-235
- [11] T. Rajala, M.-J. Laakso, E. Kaila, & T. Salakoski. VILLE – A language-independent program visualization tool. Proceedings of the Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007), Koli National Park, Finland, November 15-18, 2007. Conferences in Research and Practice in Information Technology, Vol. 88, Australian Computer Society. Raymond Lister and Simon, Eds
- [12] T. Waats. "The SFC editor: a graphical tool for algorithm development." *J. Comput. Small Coll.* 20, 2 (Dec. 2004), 73-85.
- [13] B. A. Calloni, D. J. Bagert, and H. P. Haiduk, Iconic programming proves effective for teaching the first-year programming sequence. In Proceedings of the Twenty-Eighth SIGCSE Technical Symposium on Computer Science Education (San Jose, California, United States, February 27 - March 01, 1997). J. E. Miller, Ed. SIGCSE '97. ACM, New York, NY, 262-266
- [14] D. Wolber, H. Abelson, and M. Friedman. Democratizing Computing with App Inventor. *GetMobile: Mobile Comp. and Comm.* 18, 4 (October 2014), 53–58.
- [15] J. Maloney, M. Resnick, N. Rusk, B. Silverman, & E. Eastmond. "The scratch programming language and environment." *ACM Trans. Comput. Educ.* 10, 4, 2010 Article 16.
- [16] A. Al-syakaf, M. Omar, M. Ahmad., "Social Worked-Examples Technique to Enhance Student Engagement in Program Visualization" *Baghdad Science Journal* Vol.16 (Special Issue) 2019
- [17] M Apiola, dan M. Tedre, "New perspectives on the pedagogy of programming in a developing country context." *Computer Science Education*, Vol. 22(13) 2012
- [18] M. Ben-Ari, N. Ragonis. & L. R. Ben-Bassat. A vision of visualization in teaching object-oriented programming. Second Program Visualization Workshop. Hornstrup Centret, Denmark. 2002 84-90
- [19] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, & J Paterson. A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin* Vol.39 No.4 2007
- [20] C. D. Hundhausen, "A Meta-Study of Software Visualization Effectiveness." *Journal of Visual Languages & Computing* Volume 13 Issue 3, June 2002, Pages 259-290
- [21] K. Scheiter, P. Gerjets, & R. Catrambone. "Making the Abstract Concrete: Visualizing Mathematical Solution Procedures." *Computers in Human Behavior.* 22, .12002 9–25.
- [22] M Laakso, E. Kaila, T. Rajala. "ViLLE – collaborative education tool: Designing and utilizing an exercise-based learning environment." *Education and Information Technologies*, 01/2018, 1573-7608
- [23] T. Boyle, C. Bradley, P. Chalk, R. Jones, & Pickard. D. P. "Using blended learning to improve student success rates in learning to program." *Journal of Educational Media*, Vol. 28, Nos. 2–3 2003
- [24] O. Kannusmäki, A. Moreno, N. Myller, and E Sutinen. What a novice wants: Students using program visualization in distance programming course. Proceedings of the Third Program Visualization Workshop (PVW'04), 2004,126-133
- [25] T. Rajala, M.J. Laakso, E. Kaila, & T. Salakoski, "Effectiveness of Program Visualization: A Case Study with the ViLLE Tool." *Journal of Information Technology Education: Innovations in Practice*, Vol. 7 2008
- [26] J. Sorva, V. Karavirta, dan L. Malmi, "A Review of Generic Program Visualization Systems for Introductory Programming Education." *ACM Transactions on Computing Education (TOCE)*. Vol. 13 No. 4, November 2013
- [27] P. J. Guo, "Online python tutor: Embeddable web-based program visualization for cs education," in Proceeding of the 44th ACM Technical Symposium on Computer Science Education, ser. SIGCSE '13, 2013.
- [28] M. Conway, S. Audia, T. Burnette, D. Cosgrove, and K. Christiansen, "Alice: lessons learned from building a 3D system for novices", In Proceedings of the SIGCHI conference on Human factors in computing systems, The Hague: ACM, 2000, pp. 486 – 493.
- [29] I. Reif, and T. Orehovački, "ViSA: Visualization of Sorting Algorithms", In Proceedings of the 35th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija: IEEE, 2012. pp. 1146-1151.
- [30] A.Fowler, T. Fristce, & M. Maclauren, (2012). *Kodu Game Lab: a programming environment*. The Computer Games Journal, 1(1), 17–28. Retrieved from www.computergamesjournal.com