

## COMPUTATIONAL THINKING DAN ALICE 3

<sup>1</sup>Mohamad Nurkamal Fauzan

Prodi/Jurusan D4 Teknik Informatika Politeknik Pos Indonesia

Jln. Sari Asih No. 54 Kode Pos 40151 Bandung, Jawa Barat

[m.nurkamal.f@poltekpos.ac.id](mailto:m.nurkamal.f@poltekpos.ac.id)

### ABSTRAK

Teknologi digital berkembang demikian pesatnya hal ini dapat terlihat pada era revolusi industri 4.0, kebutuhan untuk membuat solusi dalam bentuk software menjadi hal yang umum terjadi. Kemampuan untuk berkomunikasi dengan perangkat keras melalui software menjadi suatu keharusan. Bahasa pemrograman mulai diajarkan dari tingkat dasar dengan kedalaman tertentu. Industri membutuhkan tenaga yang handal untuk dapat berkarya melalui solusi berbentuk software. Metode computational thinking dengan Alice 3 digunakan untuk meningkatkan kemampuan decomposition dan abstraction pemrograman pada siswa. Hasil pengujian menunjukkan bahwa penerapan computational thinking dengan mengembangkan source code yang sudah diberikan berdampak lebih baik dibanding jika siswa membuat code program dari nol.

**Kata Kunci :** *Computational thinking, Alice 3, decomposition, abstraction.*

### ABSTRACT

*Digital technology is developing so rapidly this can be seen in the era of the industrial revolution 4.0, the need to make solutions in the form of software is becoming a common thing. The ability to communicate with hardware via software is a must. Programming languages begin to be taught from the basic level with a certain depth. Industry needs reliable personnel to be able to work through software solutions. The computational thinking method with Alice 3 is used to improve students' decomposition and abstraction programming abilities. The test results show that the application of computational thinking by developing the source code that has been given has a better impact than if students code programs from scratch.*

**Keywords:** *Computational thinking, Alice 3, decomposition, abstraction.*

---

## 1. PENDAHULUAN

### 1.1 Latar Belakang Masalah

Teknologi digital berkembang demikian pesatnya hal ini dapat terlihat pada era revolusi industri 4.0, kebutuhan untuk membuat solusi dalam bentuk software menjadi hal yang umum terjadi. Kemampuan untuk berkomunikasi dengan perangkat keras melalui software menjadi suatu keharusan. Bahasa pemrograman mulai diajarkan dari tingkat dasar dengan kedalaman tertentu. Industri membutuhkan tenaga yang handal untuk dapat berkarya melalui solusi berbentuk software, proses bisnis perusahaan harus dapat diterjemahkan oleh programmer. Perguruan tinggi harus mencetak luaran siswa yang sesuai kebutuhan industri. Pada perguruan tinggi, kurikulum dibentuk untuk membekali siswa sehingga memiliki kompetensi sesuai standar. Pada awal tahun perkuliahan, mahasiswa teknik informatika secara umum akan mendapat mata kuliah algoritma dan struktur data,

hal ini perlu untuk menjadi fondasi pada kegiatan perkuliahan selanjutnya. Pada penerimaan mahasiswa baru angkatan 2020 program studi D4 teknik informatika diperoleh jumlah mahasiswa seperti tabel 1.

**Tabel 1 Jumlah mahasiswa**

Kelas	Siswa
1A	19
1B	23
1C	27

Untuk mengukur kemampuan siswa baru dalam hal analisa maka diberikan tugas untuk membuat flowchart dan penjelasan dari game yang harus direview dari play store. Game merupakan hasil porting dari Greenfoot ke android karena memiliki basis bahasa pemrograman yang sama yaitu Java. Jumlah game yang harus direview sebanyak 10, dan merupakan hasil karya dari siswa angkatan sebelumnya (2019). Dari tugas yang diberikan, rata-rata kemampuan siswa dalam memecah masalah yang kompleks menjadi bagian-bagian yang lebih

kecil (decomposition) masih dibawah harapan begitu juga dalam hal literasi menyampaikan informasi atau langkah yang penting (abstraction), hal ini dapat dilihat pada tabel 2.

**Tabel 2 Decomposition dan abstraction (data diolah)**

	A	B	C
<b>decomposition</b>	65	69	63
<b>abstraction</b>	70	70	65

**1.2 Identifikasi Masalah**

Berdasarkan latar belakang yang telah dikemukakan di atas, masalah dapat diidentifikasi sebagai berikut :

- Bagaimana meningkatkan kemampuan decomposition dan abstraction suatu masalah pada siswa.

**1.3 Tujuan**

Adapun tujuan dari penelitian ini, yaitu :

- Meningkatkan kemampuan decomposition dan abstraction pemrograman pada siswa.

**1.4 Ruang lingkup**

Penelitian ini hanya mencakup siswa baru kelas 1A-1C D4TI Politeknik Pos Indonesia tahun 2020 sebagai penelitian awal.

**2. METODE PENELITIAN**

Strategi untuk memecahkan masalah dan tujuan dibuat menjadi beberapa tahapan ditunjukkan pada gambar 2-1.

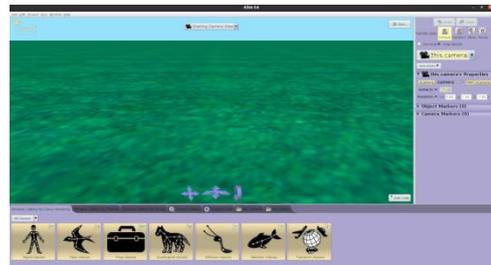


**Gambar 2-1 Metode penelitian**

**2. 1. Studi literatur**

Berdasarkan identifikasi masalah maka diperlukan suatu teknik pengajaran yang baik dan kurikulum yang terstruktur. Menurut Sudarwan, Pengajaran (teaching) yaitu teknik dan metode kerja guru dalam mentranformasikan konten pengetahuan, merangsang mengawasi dan

menfasilitasi pengembangan siswa untuk mencapai tujuan pembelajaran [1]. Konten dan kurikulum untuk keperluan penelitian ini diambil dari Oracle Academy. Oracle Academy menawarkan kepada pengajar kurikulum yang menarik menggunakan teknologi yang relevan dengan industri yang membantu siswa bersiap untuk langkah selanjutnya dalam pendidikan atau karir [2]. Software yang digunakan adalah Alice (gambar 2-2).



**Gambar 2-2 Tampilan Alice 3**

Alice adalah lingkungan pemrograman yang dirancang khusus sebagai alat pengajaran / pembelajaran untuk memungkinkan pemrogram pemula membuat animasi dan game menggunakan dunia 3D [3, 4].

Pada penelitian Dann dkk menunjukan bahwa Alice 3 memperkenalkan konsep dasar yang dikombinasikan dengan materi yang dirancang untuk mengajar transformasi kode dari Alice 3 ke Java memiliki dampak positif yang signifikan secara statistik pada pembelajaran siswa [5].

**2. 2. Pemilihan Metode**

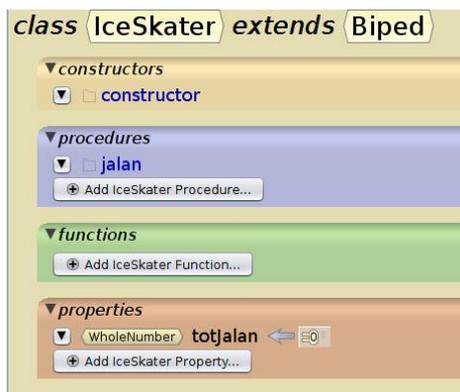
Untuk mencapai tujuan penelitian maka metode computational thinking diterapkan pada siswa[6], hal ini diharapkan dapat mengembangkan kualitas seperti abstract thinking, problem solving, pattern recognition, dan logical reasoning [7]. Menurut Kules, computational thinking melengkapi pemikiran kritis sebagai cara berpikir untuk memecahkan masalah, membuat keputusan, dan berinteraksi dengan dunia kita. Hal ini terkait konsep dan teknik seperti abstraksi, dekomposisi, desain algoritmik, generalisasi, evaluasi dan iterasi dari ilmu komputer dan informasi [8]. Menurut artikel pada BBC, terdapat empat teknik utama (landasan) pada computational thinking[9]:

1. Decomposition, memecahkan masalah atau sistem yang kompleks menjadi bagian-bagian yang lebih kecil dan lebih mudah diatur
2. Pattern recognition, mencari persamaan di antara dan di dalam masalah
3. Abstraction, berfokus pada informasi penting saja, mengabaikan detail yang tidak relevan
4. Algorithms, mengembangkan solusi langkah demi langkah untuk masalah, atau aturan yang harus diikuti untuk menyelesaikan masalah

Langkah yang digunakan berikutnya adalah menentukan bentuk project atau tugas untuk mahasiswa yang mewakili computational thinking dan hasil akhirnya dapat dibandingkan. Untuk keperluan pengujian dan pengukuran maka ditetapkan bahwa tugas berupa memodifikasi game 3d yang sudah disiapkan di github dan membuat dari awal game. Computational thinking mencakup pembuatan story board, flowchart setiap object berikut interaksinya berikut logika atau algoritma game.

Pada tipe modifikasi game, para siswa dapat melihat contoh program yang sudah diberikan untuk dipelajari dan akhirnya siswa menambahkan fitur-fitur baru dapat berupa modifikasi terhadap kelas berupa property, procedure dan function. Adapun github diharapkan dapat menjadi forking atau kolaborasi agar terjadi pengembangan source code game [10].

Untuk membuat game diperlukan object-object yang berinteraksi. Object dibuat dari kelas yang dapat dipilih melalui Alice 3, kelas tersebut dapat dimodifikasi (gambar 2-3).



Gambar 2-3 Kelas IceSkater

Berikut potongan procedure pada kelas IceSkater (gambar 2-4):



Gambar 2-4 Procedure jalan dengan input parameter

Sedangkan interaksi antar object dapat dilihat dari susunan kelas yang digunakan (gambar 2-5).



Gambar 2-5 Type hierarchy

Pada tipe pembuatan game dari awal para siswa harus melakukan studi literasi [11] yang mendalam sehingga diperlukan akses ke berbagai sumber.

Hasil kedua tipe ini akan dibandingkan. Setiap siswa akan mempresentasikan hasil project game dan akan dievaluasi setiap langkahnya sehingga mewakili konsep computational thinking.

### 3. HASIL DAN PEMBAHASAN

Dari hasil evaluasi didapat rata-rata nilai untuk tipe 1 (tabel 3-1) dan 2 (tabel 3-2) dengan mengimplementasikan computational thinking:

Tabel 3-1 Tipe 1 rerata nilai, modifikasi game yang diberikan di github

	A	B	C
decomposition	85	85	80
pattern recognition	75	75	70
abstraction	75	80	72
algorithms	70	80	70

Tabel 3-2 Tipe 2 rerata nilai, game dibuat dari awal (nol)

	A	B	C
decomposition	72	75	75
pattern recognition	70	75	70
abstraction	70	70	70
algorithms	70	72	70

Dari perbandingan kedua tipe ini didapatkan rerata tipe 1 sebesar 76.41666667 dan tipe 2 sebesar 71.58333333 sehingga tipe 1 lebih besar 4.83333333 dari tipe 2.

#### 4. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Dari analisis data dan pembahasan yang telah dilakukan, dapat ditarik kesimpulan yang sekaligus menjawab rumusan masalah penelitian sebagai berikut:

1. Penerapan computational thinking dengan mengembangkan source code yang sudah diberikan berdampak lebih baik dibanding jika siswa membuat code program dari nol.

##### 4.2 Saran

Berikut beberapa saran dari hasil laporan.

1. Untuk pengembangan selanjutnya perlu dilakukan pengukuran pada forking dan kolaborasi.
2. Literasi yang baik mungkin dapat dibangun jika siswa aktif berdiskusi atau menjawab pertanyaan pada forum yang membahas mengenai pemrograman, debugging dst.

##### DAFTAR PUSTAKA

- [1] Sudarwan Danim. 2010. *Pedagogi, Andragogi dan Heutagogi*. Bandung; Penerbit Alfabeta
- [2] <https://academy.oracle.com/en/solutions-curriculum.html>
- [3] Pausch, R., & Forlines, C. 2000. Alice: Model, paint & animate —easy-to-use interactive graphics for the web. *SIGGRAPH Comput. Graph.*, 34(2), 42-43.
- [4] Pierce, J., Cobb, T., & Pausch, R. 1998. Alice. *ACM SIGGRAPH 98 Conference abstractsand applications*. (Orlando, Florida, United States).
- [5] Dann, W., Cosgrove, D., Slater, D., Culyba, D., & Cooper, S. (2012). Mediated transfer: Alice 3 to Java. In *SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 141–146). <https://doi.org/10.1145/2157136.2157180>
- [6] Wei, X., Lin, L., Meng, N., Tan, W., Kong, S. C., & Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers and Education*, 160. <https://doi.org/10.1016/j.compedu.2020.104023>
- [7] Angeli, C., & Giannakos, M. (2020, April 1). Computational thinking education: Issues and challenges. *Computers in Human Behavior*. Elsevier Ltd. <https://doi.org/10.1016/j.chb.2019.106185>
- [8] Kules, B. (2016). Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. *Proceedings of the Association for Information Science and Technology*, 53(1), 1–6. <https://doi.org/10.1002/pra2.2016.14505301092>
- [9] <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>
- [10] Zakiah, A., & Fauzan, M. N. (2016). Collaborative Learning Model of Software Engineering using Github for informatics student. In *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/CITSM.2016.7577521>
- [11] Nichols, T. P., & LeBlanc, R. J. (2020). Beyond Apps: Digital Literacies in a Platform Society. *Reading Teacher*, 74(1), 103–109. <https://doi.org/10.1002/trtr.1926>