



Identifikasi *Extranous Cognitive Load* Siswa Dalam Mengembangkan *Computational Thinking Skill* Melalui Pembelajaran Jaring-Jaring Makanan Berbasis Snap!

Eni Nuraeni^{1*}, Tika Triwahyuni¹, Amprasto¹, Irvan Permana²

¹Universitas Pendidikan Indonesia, Jalan Dr. Setiabudi no 229 Bandung, Indonesia

²Universitas Pakuan, Bogor, Indonesia

*Email: eninuraeni@upi.edu.

DOI: 10.24815/jpsi.v10i1.22924

Article History:

Received: October 2, 2021

Accepted: December 8, 2021

Revised: November 28, 2021

Published: December 20, 2021

Abstract. Food webs learning using the Snap! is one of the learning strategies that are expected to help improve students' computational thinking. For students, this learning strategy were something new and can cause Extranous Cognitive Load (ECL). The purpose of this study was to identify students' ECL in food web learning using the Snap! to develop computational thinking skills. The research method used in this study was a pre-experimental design with a modified research design from an iterative action design. The sampling technique was purposive sampling. The sample in this study consisted of 30 seventh grade students at SMPN 2 Bandung. The research instrument used in this study was a student mental effort questionnaire to measure ECL, field notes, and a computational thinking test. Based on the results of the study, students' ECL was relatively low and increased at each meeting, except for the second meeting. Students experience an increase in their computational thinking skills after participating in food web learning using the Snap! computational model. The results of the N-Gain analysis also show that the improvement of students' computational thinking is in the moderate category and is quite effective.

Keyword: Extranous Cognitive Load, computational thinking, web food, computational model, snap!

Pendahuluan

Berkembangnya ilmu pengetahuan dan teknologi menuntut masyarakat untuk memiliki kecakapan hidup di tengah ketatnya persaingan global. Dalam hal tersebut masyarakat memerlukan berbagai keterampilan untuk memenuhi tantangan abad ke-21. Perkembangan pesat dalam dunia teknologi telah membawa perubahan besar pada dunia pendidikan. Keterampilan berpikir untuk menghadapi tantangan abad 21 pun berkembang pesat. *Computational thinking* (CT) adalah keterampilan abad ke-21 yang diharapkan mampu menjawab tantangan perkembangan tersebut. Moon dkk (2020) mendefinisikan CT sebagai kemampuan peserta didik untuk mendekati tugas-tugas yang tidak terstruktur secara sistematis berdasarkan pada pemikiran algoritmik dalam komputasi. Moon (2020) membahas adanya enam aspek CT yaitu dekomposisi, abstraksi, algoritma, *debugging*, iterasi, dan generalisasi. Baik dekomposisi dan abstraksi termasuk pemolaan masalah sebagai potongan yang dapat dikelola dan kemudian mengekstraksi fitur. Algoritma mewakili penalaran logis peserta didik untuk mendekati masalah terbatas. *Debugging* dan iterasi

adalah langkah-langkah untuk mengevaluasi pekerjaan mereka untuk menemukan, mengoreksi, dan menyaring kesalahan saat menerapkan solusi. Terakhir, generalisasi adalah tindakan mencari peluang untuk mentransfer solusi ke konteks yang lebih luas. Dengan memanfaatkan konsep komputasi, seseorang dapat berpikir secara berurutan dan logis untuk memecahkan sains yang kompleks dan masalah rekayasa. Computational thinking merupakan logika yang mendasari ilmu komputer. Peel, dkk. (2021) menyatakan bahwa CT dapat diterapkan pada disiplin ilmu lain dan memiliki manfaat untuk kehidupan sehari-hari.

Dalam beberapa tahun terakhir ini computational thinking banyak diintegrasikan ke dalam kurikulum pendidikan wajib di berbagai negara. Yasin (2020) menyatakan bahwa dengan memiliki *computational thinking*, siswa dapat belajar dan berpikir abstrak, algoritmik dan logis, serta siap untuk memecahkan masalah kompleks dan terbuka. Hal yang sama diungkapkan oleh Peel, dkk. (2021) yang mengungkapkan bahwa kemampuan berpikir komputasi dapat membantu untuk mendekonstruksi, mengabstraksi dan mengeneralisasi informasi serta berpikir secara berurutan dan algoritmik untuk memecahkan masalah dan menjelaskan suatu fenomena.

Di Indonesia *computational thinking* dikenal oleh masyarakat semenjak Kemendikbud mengeluarkan Permendikbud Nomor 35, 36, dan 37 pada tahun 2018 mengenai informatika sebagai mata pelajaran pilihan di tingkat SMP dan SMA. Dalam lampiran Permendikbud Nomor 37, secara resmi dimuat istilah *computational thinking* sebagai salah satu Kompetensi Dasar yang dipelajari dalam mata pelajaran Informatika. Terdapat dua cara yang dapat dilakukan dalam mengajarkan *computational thinking* terhadap siswa (Cotton, 1991). Cara yang pertama yaitu dengan melakukan aktivitas belajar khusus untuk mengembangkan *computational thinking* dan cara yang kedua yaitu dengan mengintegrasikan kemampuan *computational thinking* dalam mata pelajaran-mata pelajaran yang sudah ada seperti mata pelajaran biologi. Berdasarkan informasi ini, kurikulum di Indonesia menerapkan cara pertama untuk mengajarkan *computational thinking*. Mengingat pentingnya kemampuan ini dalam membangun abstraksi dan penyelesaian berbagai masalah, maka kemampuan ini perlu dikembangkan dengan cara kedua yaitu diintegrasikan ke dalam mata pelajaran.

Sejak pandemi Covid-19 melanda dunia termasuk Indonesia tahun 2020, terjadi perubahan besar pada sektor pendidikan. Kegiatan belajar mengajar yang umumnya dilakukan secara langsung atau tatap muka, kini harus dilakukan secara jarak jauh (daring). Hal ini terjadi juga di Indonesia. Semua sektor pendidikan dari pendidikan dasar hingga perguruan tinggi dipaksa melaksanakan pembelajaran secara daring. Perubahan mode belajar dari luring ke daring membawa dampak yang kurang baik terhadap pengembangan berbagai kemampuan yang diperlukan oleh siswa termasuk kemampuan penguasaan konsep dan *computational thinking*. Rubenstein & Chor (2014) menegaskan bahwa pendekatan komputasi dalam konteks biologi dan praktikumnya bukan hanya digunakan sebagai alat, tetapi integrasi pemikiran komputasi dan algoritma untuk mendesain eksperimen, menghasilkan, mengintegrasikan dan menganalisis data serta pemodelan.

Berbagai penelitian telah mengungkapkan pentingnya *computational thinking* bagi siswa. Peel dan Friedrichsen (2018) berpendapat bahwa ruang kelas biologi adalah konteks ideal untuk pembelajaran CT karena proses biologis berfungsi sebagai suatu sistem, dan memahami bagaimana sistem itu berfungsi membutuhkan pemikiran algoritmik dan keterampilan memecahkan masalah. Berdasarkan penelitian yang telah dilakukan oleh Preston (2018), siswa sekolah menengah masih sering mengalami kesulitan dalam memahami jaring-jaring makanan dan hubungannya dalam ekosistem. Padahal, kemampuan untuk menafsirkan rantai makanan dan jaring makanan berperan sangat penting dalam pemahaman konsep hubungan antar organisme dalam domain biologi. Oleh karena itu materi ini memiliki potensi yang besar untuk mengembangkan CT.

Pembelajaran menggunakan model komputasi dengan platform pemrograman *Snap!* Adalah salah satu strategi pembelajaran yang diharapkan dapat membantu meningkatkan *computational thinking* siswa. *Snap!* adalah pemrograman visual dan *drag-and-drop* dimana siswa dapat membangun blok kode sendiri dan menambahkan kode yang lebih kompleks (José, 2017). Saat belajar menggunakan *Snap!* siswa belajar membangun blok kode secara sederhana untuk menciptakan simulasi. Jaringan makanan merupakan materi biologi di SMA yang dapat disimulasikan menggunakan blok-blok kode. Jaringan-jaringan makanan menggambarkan hubungan makan antar spesies dalam suatu komunitas, interaksi antar spesies dan struktur komunitas, serta dinamika transfer energi dalam suatu ekosistem (Hui, 2012). Berdasarkan uraian tentang pemrograman *Snap!* Dan karakteristik materi, strategi pembelajaran jaringan-jaringan menggunakan model komputasi *Snap!* secara daring mungkin akan mengalami berbagai hambatan. Selain kedua hal tersebut, hambatan dapat muncul karena strategi ini merupakan hal yang baru bagi siswa. Pembelajaran jaringan-jaringan makanan umumnya dilakukan dengan strategi pembelajaran ekspositori oleh guru, sehingga perubahan strategi pembelajaran yang dialami oleh siswa akan memberikan beban pada sistem kognitif mereka.

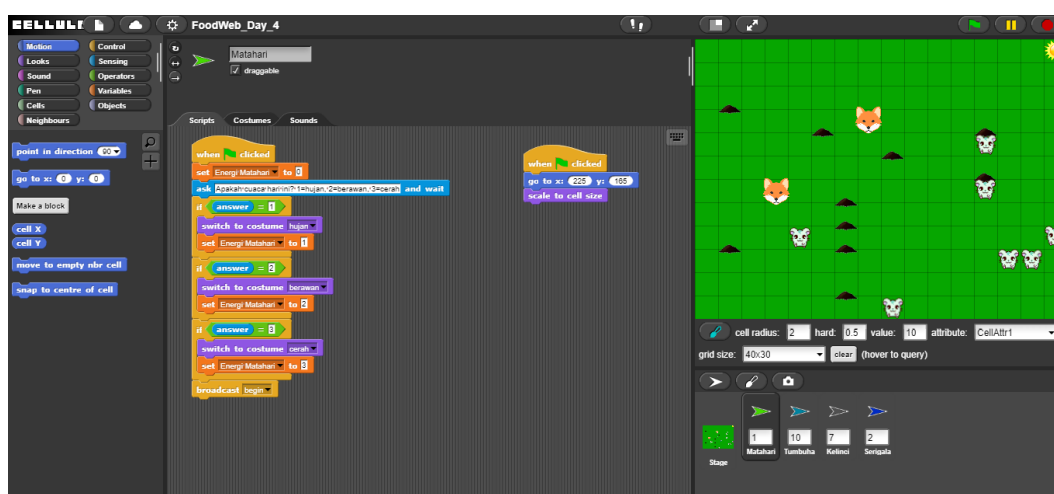
Beban kognitif adalah aktivitas memori kerja dalam melakukan tugas belajar tertentu yang dapat membebani sistem pengolahan kognitif (Scharfenberg & Bogner, 2010). Terdapat tiga komponen beban kognitif yaitu *Intrinsic Cognitive load* (ICL), *Extraneous Cognitive load* (ECL) dan *Germane Cognitive load* (GCL). Dalam teori beban kognitif, ECL digambarkan oleh besarnya usaha mental siswa ketika memahami suatu materi dan mengembangkan suatu kemampuan tertentu. *Extraneous Cognitive load* (ECL) merupakan beban dari luar yang disebabkan oleh strategi pembelajaran yang dapat menghambat proses belajar siswa (Scharfenberg & Bogner, 2010). Proses berpikir berhubungan dengan aktivitas memori kerja ketika siswa mengalami proses belajar (Rahmat, dkk., 2017). Seberapa keras seseorang mencoba memproses secara aktif informasi yang disajikan disebut sebagai usaha mental (Kirschner, 2012). Oleh karena itu, maka identifikasi ECL dapat dilakukan dengan pengukuran usaha mental yang diterima oleh siswa. Hal ini berarti bahwa strategi yang diterapkan oleh guru dapat menentukan usaha-usaha mental siswa selama proses pembelajaran.

Berbagai penelitian difokuskan pada strategi meningkatkan *computational thinking* selama proses pembelajaran, misalnya tentang pengembangan CT yang signifikan pada pembelajaran translasi sistensis protein dengan *lightboot game* oleh Peel dan Friedrichsen (2018). Akan tetapi penelitian tentang usaha mental yang dialami oleh siswa selama proses integrasi CT ini belum banyak diungkap. Dalam artikel ini dikaji usaha mental yang ditempuh siswa untuk mempelajari konsep jaringan-jaringan makanan menggunakan model komputasi pemrograman *Snap!* untuk meningkatkan kemampuan *computational thinking*. Terdapat dua pertanyaan yang akan dijawab di dalam penelitian yaitu bagaimana kemampuan *computational thinking* siswa sebelum dan setelah pembelajaran dan bagaimana ECL siswa selama pembelajaran jaringan-jaringan makanan menggunakan model komputasi *Snap!*?

Metode

Peneliti memodifikasi desain penelitian satu kelompok perlakuan berulang (*repeated measures design*) untuk mengambil data usaha mental dan *computational thinking* siswa. Data tentang usaha mental diambil pada setiap akhir pembelajaran. Partisipan dalam penelitian merupakan 30 siswa kelas VII dari SMP Negeri di Bandung dengan teknik pemilihan *purposive sampling*. Penggunaan aplikasi *Snap!* memerlukan komputer/laptop pribadi, dan memiliki keterampilan dasar dalam menggunakan aplikasi seperti *Whatsapp*, *Zoom* dan *web browser*.

Model komputasi yang digunakan oleh siswa adalah platform pemrograman *Snap!* yaitu sebuah pemrograman visual dan *drag-and-drop*. Dengan *Snap!* Siswa dapat membangun blok kode sendiri dan menambahkan blok kode yang lebih kompleks (José, 2017). *Snap!* dikembangkan oleh Jens Mönig dan Brian Harvey dibawah pengawasan *University of California* di Berkeley dengan tujuan membawa kekuatan bahasa pemrograman *Scheme*, dan konsep ilmu komputernya ke dalam lingkungan seperti platform pemrograman *Scratch*. Kemampuan untuk membuat blok sendiri dari blok lain adalah salah satu prinsip dasar dari *Snap!*. Dengan kata lain, *Snap!* memungkinkan pengguna membuat ekspresi khusus yang terlihat dan berperilaku seperti ekspresi yang telah ada sebelumnya (Romagosa, 2019).



Gambar 1. Tampilan dari Platform Pemrograman *Snap!*

Selama kegiatan pembelajaran yang dilakukan dalam empat kali pertemuan, siswa diberikan panduan untuk membangun suatu model simulasi aliran energi dalam rantai makanan. Di awal pertemuan, kegiatan pembelajaran dilakukan secara *unplugged*, dimana pembelajaran yang dilakukan tidak mewajibkan siswa untuk menggunakan komputer. Namun, pada pertemuan selanjutnya, yaitu pertemuan kedua, ketiga, dan keempat, siswa mulai diminta untuk memperbaiki urutan blok kode, memodifikasi blok kode yang telah ada, dan menambahkan blok kode yang baru agar simulasi rantai makanan dapat berjalan dengan lancar.

Pembelajaran dilaksanakan selama empat pertemuan. Pada pertemuan pertama siswa mempelajari konsep dasar jaring-jaring makanan, dan istilah-istilah pemrograman. Pada pertemuan kedua, guru mendemonstrasikan dasar-dasar pemrograman di *Snap!* diikuti oleh siswa secara mandiri. Pada pertemuan ketiga, siswa memodifikasi model rantai makanan menggunakan *Snap!* dengan informasi berupa panduan tertulis dari guru. Pada pertemuan keempat siswa membuat pemrograman untuk *sprite* baru dalam model rantai makanan di *Snap!* dengan informasi berupa panduan tertulis dari guru. Detail strategi pembelajaran disajikan dalam Tabel 1.

Tabel 1. Rincian Kegiatan pembelajaran

Pertemuan	Strategi
1	Pembelajaran tanpa komputer tentang konsep dasar ekosistem dan jaring-jaring makanan
2	Pengenalan food web yang digunakan dan istilah dalam program Pengenalan fitur-fitur pada web yang digunakan. memodelkan sistem rantai makanan. merancang dan membuat pemodelan antara matahari dan tumbuhan dalam perannya pada sistem rantai makanan.
3	merancang dan membuat pemodelan antara matahari, tumbuhan bunga matahari dan hamster dalam perannya pada sistem rantai makanan
4	merancang dan membuat pemodelan antara matahari, tumbuhan, hamster dan serigala dalam perannya pada rantai makanan

Data *computational thinking* siswa diambil menggunakan *instrument test computational thinking* (Wiebe dkk, 2019) yang diberikan di awal dan akhir pembelajaran. Instrumen tes CT terdiri dari 25 soal pilihan ganda. Analisis data diolah dengan menghitung Normalitas Gain (N-gain). Kriteria N-gain menggunakan Hake (1999).

Soal-soal *computational thinking* yang dikembangkan oleh Wiebe dkk. (2019) sudah mencakup seluruh komponen dalam *computational thinking*, yaitu abstraksi, generalisasi, dekomposisi, algoritmik dan *debugging*. Kisi-kisi dari soal instrumen tes *computational thinking* dapat dilihat pada Tabel 2.

Tabel 2. Kisi-kisi Instrumen Tes Kemampuan *Computational Thinking* Siswa

No.	Komponen <i>Computational Thinking</i>	Indikator <i>Computational Thinking</i>	Jumlah Soal
1.	Abstraksi	Siswa mampu memutuskan informasi yang penting untuk disimpan dan informasi yang harus diabaikan.	5
2.	Generalisasi	Siswa mampu merumuskan solusi dalam istilah umum sehingga dapat diterapkan pada masalah yang berbeda.	5
3.	Dekomposisi	Siswa mampu memecahkan masalah yang kompleks menjadi bagian-bagian kecil atau sederhana yang lebih mudah untuk dipahami dan diselesaikan.	5
4.	Algoritmik	Siswa mampu menyusun langkah-langkah solusi untuk memecahkan permasalahan yang diberikan.	5
5.	<i>Debugging</i>	Siswa mampu mengenali ketika terdapat tindakan yang tidak sesuai dengan instruksi, dan dapat memperbaiki kesalahan tersebut.	5
Total			25

Data ECL diambil menggunakan Kuisisioner berisi 20 soal pernyataan dengan pilihan jawaban Skala Likert 1-7 (Tabel 3) yang menggambarkan usaha mental siswa. Data usaha mental dirata-ratakan, dan dikategorisasi menurut Paas (1992) kemudian dipersentasekan

untuk memudahkan interpretasi ke dalam ECL. Terdapat 4 buah instrumen kuesioner untuk usaha mental 4 pertemuan.

Tabel 3. Kuisisioner *Subjective rating scale* Usaha Mental

Skor	Keterangan Tanggapan
1	Sangat Setuju
2	Setuju
3	Cukup Setuju
4	Netral
5	Cukup Tidak Setuju
6	Tidak Setuju
7	Sangat Tidak Setuju

(Sumber: Paas, 1992)

Hasil dan Pembahasan

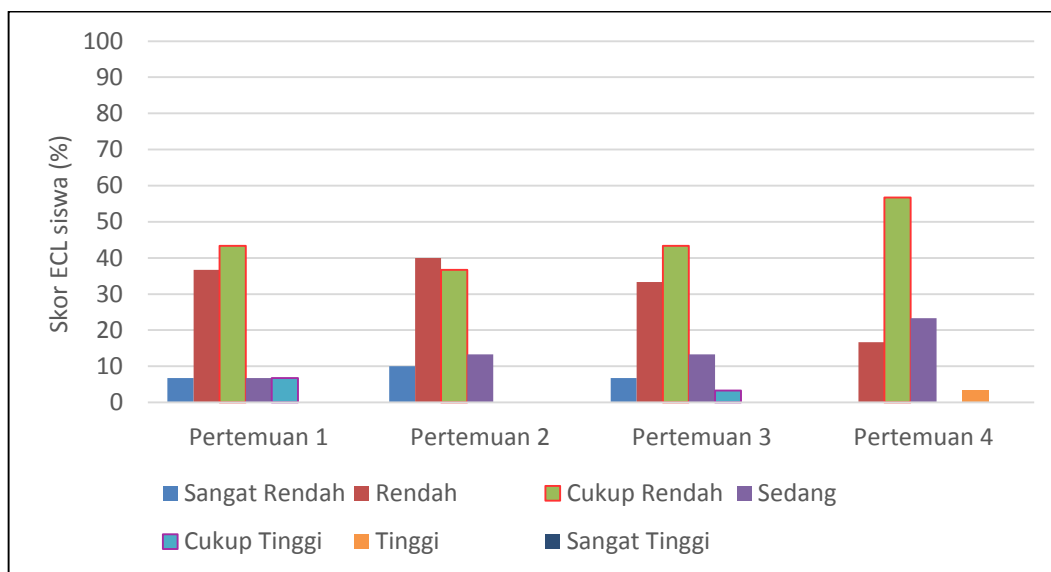
Computational thinking dalam penelitian ini dikembangkan melalui penerapan pembelajaran berbasis web dengan strategi pada Tabel 1. Kemampuan siswa dalam CT disajikan dalam Tabel 4.

Tabel 4. Rekapitulasi N-gain *Computational Thinking* Siswa

Komponen	Rekapitulasi Hasil Analisis N-gain
Rata-rata N-gain	0,56
Minimum	0,21
Maksimum	0,76
Kategori	Sedang atau Cukup Efektif

Pada Tabel 4 terlihat bahwa terjadi peningkatan nilai *computational thinking* siswa yang cukup baik dengan kategori sedang. Hasil perhitungan menunjukkan bahwa nilai rata-rata N-gain *computational thinking* siswa adalah 0,56 %, nilai maksimum N-gain adalah sebesar 0,76 dan nilai minimum sebesar 0,21. Moon, dkk. (2020) mengungkapkan bahwa *computational thinking* tidak hanya terbatas pada tindakan logis untuk pemrograman komputer tetapi secara luas menggambarkan pendekatan kinerja sistematis dalam penyelesaian masalah untuk menjawab berbagai tantangan.

Hasil identifikasi usaha mental siswa selama pembelajaran jaring-jaring makanan berbasis web disajikan dalam Gambar 2. Data pada Gambar 2 menunjukkan bahwa usaha mental yang dilakukan oleh siswa selama empat pertemuan mengalami fluktuatif.



Gambar 2. Data persentase siswa berdasarkan ECL

Kegiatan pembelajaran pertemuan pertama berfokus untuk mengembangkan pemahaman konsep dasar jaring-jaring makanan dan istilah-istilah yang berhubungan dengan pemrograman, seperti “model ilmiah”, dan belajar membuat *pseudocode*. Strategi yang diterapkan sama seperti pembelajaran yang biasa siswa alami. Berdasarkan Gambar 1, pada pertemuan pertama ini ditemukan bahwa usaha mental siswa didominasi oleh kategori rendah. Pemrosesan informasi yang dilakukan selama pembelajaran *unplugged* di pertemuan pertama tidak mengalami hambatan yang berarti. Temuan ini sejalan dengan Lim & Chen (2015). Tahapan pertama dalam membangun suatu model komputasi adalah perumusan model konseptual, yaitu seperangkat ide tentang operasi dasar sistem yang diinginkan atau katalog komponen yang terlibat dan bagaimana setiap komponen dapat berinteraksi satu sama lain. Meskipun demikian istilah-istilah yang dikemukakan dalam pertemuan ini merupakan sebuah informasi yang baru bagi siswa, sehingga dapat menambah usaha mental yang mereka butuhkan untuk memproses informasi tersebut. Adapun dalam kegiatan membuat *pseudocode* kelinci diperlukan pemikiran langkah-langkah siklus makhluk hidup yang logis, sehingga dibutuhkan usaha mental yang cukup tinggi (Gambar 3).

Pada pertemuan kedua, terlihat persentase siswa dengan usaha mental rendah dan sangat rendah mengalami penurunan. Sementara siswa dengan usaha mental tinggi dan cukup tinggi mengalami kenaikan. Kondisi ini menunjukkan bahwa pembelajaran dirasakan meningkat tingkat kesulitannya oleh siswa. Pengenalan fitur-fitur pada web yang digunakan menggunakan perangkat laptop dan berinteraksi langsung dengan food web, memodelkan sistem rantai makanan, merancang dan membuat pemodelan antara matahari dan tumbuhan dalam perannya pada sistem rantai makanan diproses lebih berat dibandingkan materi pertemuan pertama.

Menggunakan bahasa untuk menggambarkan langkah (atau perilaku) yang kita amati dapat membantu kita menerjemahkan sains ke dalam kode. Ini membantu kita dalam membuat simulasi komputer. Kami menyebutnya pseudocode.		
<p>Bayangkan kamu sedang mengamati kondisi cuaca dan mendeskripsikan apa yang terjadi berdasarkan kondisi tersebut. Contoh pseudocode mungkin akan terlihat seperti ini:</p> <p>JIKA kondisi cerah:</p> <ul style="list-style-type: none"> ● MATAHARI tampak cerah ● Matahari menghasilkan lebih banyak energi untuk tumbuhan <p>JIKA kondisi hujan:</p> <ul style="list-style-type: none"> ● MATAHARI muncul di balik awan ● Matahari menghasilkan energi untuk tumbuhan <p>JIKA kondisi berawan:</p> <ul style="list-style-type: none"> ● MATAHARI akan tertutup sepenuhnya ● Matahari menghasilkan energi yang sangat sedikit untuk tumbuhan 	<p>Sekarang kamu mencobanya. Lengkapi pseudocode yang mendeskripsikan pertumbuhan dari suatu tumbuhan yang dimulai di bawah ini.</p> <p>JIKA tumbuhan adalah biji:</p> <ul style="list-style-type: none"> ○ Biji memiliki cukup energi: ○ Biji tumbuh menjadi kecambah ○ Kecambah kehilangan energi <p>JIKA tumbuhan adalah kecambah:</p> <p>JIKA kecambah memperoleh cukup _____:</p> <ul style="list-style-type: none"> ○ Kecambah _____ menjadi tumbuhan muda ○ Tumbuhan muda _____ <p>JIKA tumbuhan ini tumbuhan muda:</p> <p>JIKA tumbuhan muda memperoleh cukup _____:</p> <ul style="list-style-type: none"> ○ Tubuh muda _____ menjadi _____ ○ _____ 	<p>Seperti apa pseudocode mendeskripsikan kelinci? Tuliskan pseudocode kamu di bawah ini. Pastikan untuk memasukkan bagaimana kelinci memperoleh energi.</p> <p>JIKA kelinci kekurangan energi:</p> <p>JIKA kelinci memiliki cukup energi:</p> <p>JIKA seekor kelinci kehilangan semua energinya:</p>

Gambar 3. Langkah-langkah Membuat *Pseudocode* pada Pertemuan Pertama Pembelajaran *Food Web* Menggunakan Model Komputasi

Pertemuan ketiga, terjadi kenaikan persentase siswa dengan kategori usaha mental cukup rendah dan cukup tinggi. Hal ini menunjukkan bahwa pertemuan ketiga cukup menyita upaya kognitif siswa dalam memproses informasi tentang merancang dan membuat pemodelan antara matahari, tumbuhan bunga matahari dan hamster dalam perannya pada sistem rantai makanan. Penambahan variabel yang dimodelkan di dalam web menambah sedikit beban terhadap kognitif siswa. Pada pertemuan ini, siswa diminta memodifikasi model rantai makanan dalam *platform* pemrograman *Snap!*. Guru memberikan informasi berupa panduan tertulis kepada siswa, dan membiarkan siswa mengeksplor sendiri langkah-langkah yang tepat untuk memodifikasi model rantai makanan dalam *platform* pemrograman *Snap!*, sehingga dibutuhkan usaha mental yang cukup tinggi. Hal inilah yang menjadi beban siswa. Sedikit bimbingan guru dan mengandalkan pengetahuan awal dari pertemuan kedua.

Pada pertemuan keempat, terjadi kenaikan persentase jumlah siswa dengan kategori usaha mental sedang yang cukup besar. Meskipun ada penurunan pada siswa dengan kategori rendah dan sangat rendah, kontribusi siswa dengan kategori sedang dan cukup rendah perlu diperhatikan. Keadaan ini menunjukkan bahwa penambahan variabel pada pertemuan 2 diproses lebih berat oleh sebagian siswa. Apabila dibandingkan dengan pertemuan ketiga, pertemuan keempat memiliki tahapan pemrograman yang lebih kompleks. Siswa diminta untuk membuat kode skrip secara lengkap untuk *sprite* baru, sedangkan di pertemuan ketiga, siswa hanya diminta untuk memodifikasi kode skrip dari *sprite* yang telah disediakan. Oleh karena itu, diperlukan usaha mental yang lebih besar pada pertemuan keempat.

Data pada Gambar 2 menunjukkan bahwa siswa dengan kategori usaha mental tinggi hingga sangat tinggi berada di persentase sangat rendah. Berdasarkan data rata-rata persentase usaha mental pada Gambar 2 tersebut, dapat disimpulkan bahwa pada semua pertemuan keadaan usaha mental siswa cukup stabil. Berdasarkan hal tersebut ditemukan bahwa siswa mengalami beban yang ringan selama pembelajaran menggunakan web untuk materi jaring-jaring makanan.

Pembelajaran dengan menggunakan pemrograman Snap! membantu siswa untuk memiliki *computational thinking* sehingga penambahan variabel pada pemrograman dapat diselesaikan dengan baik oleh siswa. Temuan ini menunjukkan bahwa strategi yang diterapkan dapat mereduksi besarnya interaktivitas konten yang harus dipelajari oleh siswa. Interaktivitas konten yang tinggi dapat lebih banyak sumber daya kognitif yang tersedia pada siswa. Penambahan variabel pada pemrograman secara bertahap kompleksitasnya memenuhi strategi untuk menurunkan beban kognitif siswa. Kapasitas memori kerja siswa untuk memproses informasi tentang jaring-jaring makanan melalui pembuatan pemrograman Snap! menjadi lebih optimal. Rahmat dkk. (2017) menjelaskan bahwa kapasitas memori kerja siswa bergantung kepada level pengetahuan awal yang dimilikinya. Pemahaman tentang istilah-istilah pemrograman di pertemuan pertama memberikan pengetahuan awal yang cukup untuk melanjutkan ke pemrograman secara bertahap di pertemuan-pertemuan selanjutnya. Dengan pengetahuan awal yang memadai ini kapasitas memori kerja siswa menjadi optimal.

Mempelajari koding sederhana melibatkan berbagai alat analisis mental yang melekat dalam bidang ilmu komputer, termasuk pemikiran rekursif, menerapkan abstraksi saat menghitung, menyelesaikan tugas-tugas kompleks, dan menggunakan penalaran heuristik untuk menemukan solusi (Bers, 2018). Membuat kode-kode sederhana yang kompleksitasnya ditingkatkan dapat mendorong siswa mengembangkan *computational thinking*. Strategi yang diterapkan dalam 4 pertemuan tidak menghasilkan *extraneous cognitive load* yang berat bagi siswa. Petunjuk yang jelas pada pertemuan 2-4 untuk membantu siswa mengkode telah membantu mereka merumuskan koding yang sesuai untuk memodelkan jaring makanan. Menurut teori belajar kognitif, petunjuk untuk *self-explanation* membantu membimbing pemikiran siswa dan membantu mengidentifikasi apa yang siswa lakukan jika siswa perlu merumuskan penjelasan untuk memproses informasi yang relevan untuk pembelajaran.

Kesimpulan

Kesimpulan hasil penelitian ini yaitu bahwa *computational thinking* siswa dapat ditingkatkan dengan menerapkan pembelajaran berbasis web menggunakan pemrograman Snap!. Beban kognitif yang dialami siswa selama pembelajaran yang sistematis, kompleksitas konten dan proses kodingnya termasuk ke dalam kategori ringan. Penambahan variabel yang dikoding membantu siswa mengoptimalkan pemrosesan informasi tentang konsep jaring-jaring makanan

Daftar Pustaka

- Bers, M.U. 2018. Coding and computational thinking in early childhood: the impact of scratchjr in Europe. *European Journal of STEM Education*, 3(3):1-13.
- Cotton, K. 1991. *Teaching thinking skills*. Northwest Regional Educational Laboratory, School Improvement Program, Portland.
- Lim, B.L. & Chen, C.J. 2021. Computational thinking (algorithms) through unplugged programming activities: exploring upper primary students' learning experiences. *International Journal of Academic Research in Business and Social Sciences*, 11(14):384-403.

- José, F. & Francisco, G.P . 2017. Improving computational thinking using follow and give instructions. *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, 2017 October*, p.1-7.
- Hake, R.R. 1999. *Analyzing change/gain score*. [Online] Tersedia: <https://web.physics.indiana.edu/sdi/AnalyzingChange-Gain.pdf> (2 Juli 2021).
- Hui, D. 2012. Food web: concept and applications. *Nature Education Knowledge*, 3(12):612.
- Kirschner, P.A. & Kirschner F. 2012. *Mental effort*, dalam Seel, N.M. (eds), *Encyclopedia of the sciences of learning*, Springer, Boston.
- Moon, J., Do, J., Lee, D., & Choi, G.W. 2020. A conceptual framework for teaching computational thinking in personalized OERs. *Smart Learning Environments*, 7(6):1-19.
- Paas, F. 1992. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology*, 84(4):429-434.
- Peel, A., Sadler, T. D., & Friedrichsen, P. 2021. Using unplugged computational thinking to scaffold natural selection learning. *The American Biology Teacher*, 83(2):112-117.
- Peel, A. & Friedrichsen, P. 2018. Algorithms, abstractions, and iterations: teaching computational thinking using protein synthesis translation. *The American Biology Teacher*, 80(1):21-28.
- Preston, C. 2018. Food webs: Implications for instruction. *The American Biology Teacher*, 80(5):331-338.
- Rahmat, A., Soesilawaty, S.A., Nuraeni, E., & Hidayat, T. 2017. Controlling cognitive load of high school student in biology class. *Journal of Science Education*, 2(18):105-108.
- Romagosa, B. 2019. The Snap! programming system. *Encyclopedia of Education and Information Technologies*, p.1-10.
- Rubinstein, A. & Chor, B. 2014. Computational thinking in life science education. *PLOS Computational Biology*, 10(11):1-5.
- Scharfenberg, F.J. & Bogner, F.X. 2010. Instructional efficiency of changing cognitive in an out-of-school laboratory. *International Journal of Science Education*, 32(6):829-824.
- Wiebe, E., London, J., Aksit, O., Mott, B., Boyer, K., & Lester, J. 2019. Development of a lean computational thinking abilities assessment for middle grades students. *Proceeding of SIGCSE, 2019 February*, p.456-461.
- Yasin, M. 2020. *Computational thinking untuk pembelajaran dasar-dasar pemrograman komputer*. [Online]. Diakses dari: https://www.researchgate.net/publication/340637723_computational_thinking_untuk_pembelajaran_dasar-dasar_pemrograman_komputer (10 Januari 2021).