

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/347646698>

# Computational thinking pemecahan masalah di abad ke-21

Book · December 2020

CITATIONS

0

READS

1,295

4 authors:



**Swasti Maharani**

Universitas PGRI Madiun

43 PUBLICATIONS 53 CITATIONS

SEE PROFILE



**Toto Nusantara**

State University of Malang

131 PUBLICATIONS 381 CITATIONS

SEE PROFILE



**Abdur Rahman Asari**

Universitas Negeri Malang, Malang, Jawa Timur, Indonesia

127 PUBLICATIONS 297 CITATIONS

SEE PROFILE



**Abd. Qohar**

State University of Malang

62 PUBLICATIONS 121 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Critical thinking [View project](#)



Teaching for Critical Thinking [View project](#)

# Computational **THINKING**

*(Pemecahan Masalah di Abad ke-21)*



Swasti Maharani  
Toto Nusantara  
Abdur Rahman As'ari  
Abd. Qohar

### Sanksi Pelanggaran Pasal 113

Undang-undang Nomor 28 Tahun 2014

#### Tentang Hak Cipta

1. Setiap orang yang dengan atau tanpa hak melakukan pelanggaran terhadap hak ekonomi yang sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan ancaman pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp. 100.000.000 (seratus juta rupiah)
2. Setiap orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000 (lima ratus juta rupiah).
3. Setiap orang dengan tanpa hak dan/atau tanpa izin Pencipta atau Pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk penggunaan Secara Komersial dipidana dengan pidana penjara paling lama (empat) tahun dan/atau pidana denda paling banyak Rp. 1.000.000.000 (satu miliar rupiah).
4. Setiap orang memenuhi unsur sebagaimana dimaksud ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/ atau pidana denda paling banyak Rp. 4.000.000.000 (empat miliar rupiah).

# **COMPUTATIONAL THINKING**

**Pemecahan Masalah  
di Abad Ke-21**

# COMPUTITONAL THINKING

## Pemecahan Masalah di Abad Ke-21

© Swasti Maharani, Toto Nusantara, Abdur Rahman As'ari, Abd. Qohar

Editor : Addy Septyawan, Iin Susilowati, Ebitya Fajar Subeqi, Elzra  
Melasevix

Layout : Team WADE Publish

Design Cover : Team WADE Publish

Sumber Gambar: <https://www.freepik.com/>

Diterbitkan oleh:



Anggota IKAPI 182/JTI/2017

Cetakan Pertama, September 2020

ISBN: 978-623-7548-63-8

Hak Cipta dilindungi undang-undang.

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa seizin tertulis dari Penerbit.

Perpustakaan Nasional: Katalog Dalam Terbitan (KDT)

xii+174 hlm; 15,5x23 cm

# KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kehadirat Allah SWT karena buku ini telah selesai disusun. Buku ini disusun sebagai bentuk salah satu luaran hibah Penelitian Disertasi Doktor yang didanai oleh Kementerian Riset dan Teknologi/Badan Riset dan Inovasi Nasional (RISTEK-BRIN). Buku ini disusun agar dapat membantu para civitas akademika, penulis, dan semua pembaca dalam mempelajari konsep-konsep *Computational Thinking*.

Penulis menyadari jika isi di dalam buku ini masih mempunyai kekurangan, namun penulis menyakini sepenuhnya bahwa sekecil apapun buku ini tetap akan memberikan banyak manfaat bagi pembaca.

Akhir kata untuk penyempurnaan buku ini, maka kritik dan saran daripembaca sangatlah berguna untuk penulis kedepannya.

Madiun, Juli 2020

Penulis



# DAFTAR ISI

<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xi
<b>BAB 1 SEJARAH COMPUTATIONAL THINKING</b> .....	1
A.    Pendahuluan .....	1
B.    Sejarah CT.....	2
<b>BAB 2 PENGERTIAN <i>COMPUTATIONAL THINKING</i></b> .....	7
A.    Pengertian <i>Computational Thinking</i> (CT) .....	7
B.    Konsep <i>Computational Thinking</i> (CT).....	12
C.    Komponen <i>Computational Thinking</i> .....	13
<b>BAB 3 <i>COMPUTATIONAL THINKING</i> (CT) PADA BIDANG     TEKNOLOGI</b> .....	17
A.    Hubungan CT dan Penggunaan Teknologi.....	17
B.    Pengajaran Pemrograman dengan CT.....	19
1.    Orientasi Pemrograman .....	22
2.    Diagram Alur Pemrograman.....	22
3.    Pemrograman Terstruktur .....	23
4.    Pemrograman Deklaratif.....	23
5.    Pemrograman berorientasi pada Objek .....	23
6.    Pendekatan pemrograman Paralel.....	24
7.    Pemrograman berbasis masalah .....	24
8.    Struktur Informasi .....	25



C.	Belajar Pemrograman dengan CT Melalui Permainan Digital .....	26
D.	Mengembangkan Keterampilan CT dan Belajar Pengantar pemrograman.....	35
<b>BAB 4</b>	<b>COMPUTATIONAL THINKING PADA SAINS .....</b>	<b>39</b>
A.	Revolusi Komputer Memengaruhi Cara Berpikir Seseorang Tentang Sains, Eksperimen, dan Penelitian. ....	39
1.	Asal usul Istilah Ilmu Komputasi .....	41
2.	Pemikiran komputasi muncul dari dalam bidang ilmiah – itu tidak diimpor dari ilmu komputer. ....	42
3.	Apa itu Pemikiran Komputasi? .....	43
4.	Apa itu Model Komputasi? .....	45
5.	Apakah Semua Masalah Dapat Diselesaikan dengan Komputasi?.....	47
B.	Berpikir Komputasi di Kelas Sains.....	49
<b>BAB 5</b>	<b>COMPUTATIONAL THINKING (CT) PADA MATEMATIKA .....</b>	<b>51</b>
A.	Konsep Matematika dalam CT .....	51
B.	CT dalam Meningkatkan Keterampilan Matematika .....	54
C.	Pemikiran Komputasi dan Matematika .....	55
D.	Mendefinisikan Berpikir Komputasi untuk Matematika dan Ruang Kelas Sains.....	57
1.	Mengapa Membawa Pemikiran Komputasi ke Ruang Kelas Matematika dan Sains? .....	58
2.	Peran Tumbuh Komputasi dalam Matematika dan Sains.....	59
E.	Berpikir Komputasi dan Penalaran Matematis.....	61
<b>BAB 6</b>	<b>COMPUTATIONAL THINKING DI BIDANG PENDIDIKAN.....</b>	<b>65</b>

A.	Pemikiran Komputasi Menjadi Keterampilan Dasar bagi Siswa.....	65
B.	Konstruksionisme dan Kerangka Berpikir Komputasi..	70
C.	Spekulasi CT dalam Pendidikan di Masa Yang Akan Datang .....	73
<b>BAB 7 COMPUTATIONAL THINKING PADA PENDIDIKAN MATEMATIKA .....</b>		<b>77</b>
A.	Pendekatan <i>Computational Thinking</i> .....	77
B.	<i>Computational Thinking</i> dalam Pendidikan Matematika	78
C.	Bagaimana Meluangkan Waktu untuk Berpikir Komputasi.....	82
D.	Pembelajaran Matematika melalui Berpikir Komputasi	84
E.	Mengapa CT dalam Pendidikan Matematika?.....	85
<b>BAB 8 HASIL-HASIL PENELITIAN COMPUTATIONAL THINKING (CT).....</b>		<b>87</b>
<b>DAFTAR PUSTAKA.....</b>		<b>149</b>
<b>TENTANG PENULIS .....</b>		<b>169</b>

# DAFTAR TABEL

Tabel 1. Aktivitas Praktik Computational Thinking .....	12
Tabel 2. Komponen Computational Thinking.....	13
Tabel 3. Unsur-unsur Computational Thinking.....	16
Tabel 4. Karakter CT pada Game Program Your Robot .....	36

# DAFTAR GAMBAR

Gambar 1. Tampilan Game Program Your Robot .....	30
--	----



# BAB 1

## SEJARAH COMPUTATIONAL THINKING

### A. Pendahuluan

Gagasan tentang *Computational Thinking* (CT) merupakan gagasan yang kontroversial karena tidak ada definisi universal dari istilah ini dalam sastra (Lockwood & Mooney, 2017; Moreno-León, Román-González, & Robles, 2018). Konsep CT dan praktik pemrograman sulit untuk digambarkan dalam literatur karena banyak studi CT atau diskusi teori yang menggunakan pemrograman sebagai konteksnya (Fletcher dan Lu, 2009; Hambruch et al., 2009; Lee et al., 2011). Hal ini dapat membingungkan pembaca dan sering menimbulkan anggapan bahwa CT sama dengan pemrograman komputer atau paling tidak mereka beranggapan bahwa CT membutuhkan penggunaan bahasa pemrograman. CT yang berfokus pada pengembangan ketrampilan berpikir ini sementara terdapat dalam mata pelajaran di luar ilmu komputer atau *computer science* (CS). CT tidak perlu mensyaratkan penggunaan pemrograman dan juga tidak ada sarjana CT yang menyatakan bahwa pemrograman harus menjadi konteks di mana keterampilan ini dikembangkan. Untuk lebih memahami mengapa CT mengambil rute alternatif ini untuk mengembangkan keterampilan berpikir, alangkah lebih baik jika kita harus mengetahui sejarah penelitian mengenai pemrograman dan keterampilan berpikir. Sebagaimana George Santayana yang mengatakan bahwa, “*Those who cannot remember the past are condemned to repeat it*” (Voogt et al., 2015).

## B. Sejarah CT

Sejarah adanya CT berawal pada tahun 1950-an, namun sebenarnya ide CT telah ada jauh sebelumnya. CT melibatkan ide-ide seperti abstraksi, representasi data, dan pengorganisasian data secara logis, yang juga lazim dalam jenis pemikiran lain, seperti pemikiran ilmiah, pemikiran teknik, pemikiran sistem, pemikiran desain, pemikiran berbasis model, dan sejenisnya. Baik secara gagasan maupun istilah, CT bukanlah hal baru, sebelumnya ada istilah-istilah seperti algoritme, pemikiran yang prosedural, pemikiran algoritmik, dan literasi komputasi yang dipelopori oleh Alan Perlis dan Donald Knuth. Istilah CT digunakan untuk pertama kalinya oleh Seymour Papert pada tahun 1980 dan digunakan kembali pada tahun 1996 dalam konteks pendidikan matematika. Hal tersebut merujuk pada implementasi pemikiran prosedural anak-anak melalui pemrograman komputer. CT dapat digunakan untuk memecahkan masalah pada skala yang rumit secara algoritmik, dan sering digunakan untuk mewujudkan peningkatan efisiensi yang besar.

Istilah CT diungkap kembali oleh Jeannette Wing pada tahun 2006. Wing membawa istilah CT ke asosiasi pendidik komputer sains atau *Computer Science Teachers Association* (CSTA) dalam tulisannya yang termuat di *ACM Communications*. Karyanya tersebut mengungkapkan bahwa berpikir komputasi adalah keterampilan mendasar bagi semua orang, bukan hanya ilmuwan komputer, dan berpendapat pentingnya mengintegrasikan ide-ide komputasi ke dalam mata pelajaran lain di sekolah. Serangkaian pertanyaan pun muncul tentang CT, diantaranya yaitu bagaimana memasukkan CT ke dalam pendidikan siswa dan bagaimana mengkomunikasikan prinsip, istilah, dan cara berpikir CT yang mendalam kepada para ahli (guru). Selain itu Wing juga berpendapat bahwa CT

harus dianggap sebagai keterampilan inti, tidak hanya dalam proses pemrograman komputer, tetapi dalam setiap tindakan yang melibatkan kemampuan analitis manusia. Makalah Wing menyarankan bahwa CT melibatkan penggunaan abstraksi dan dekomposisi pada waktu-waktu berikut: selama resolusi tugas yang kompleks, ketika memilih representasi yang tepat dari suatu masalah, dan ketika memodelkan aspek-aspek yang relevan dari suatu masalah untuk membuatnya dapat ditelusuri. Dia menggambarannya sebagai bentuk penalaran heuristik yang digunakan untuk menemukan solusi, dan abstraksi bertingkat yang digunakan manusia.

Menggunakan definisi Wing sebagai titik awal, akademisi lain telah mengusulkan berbagai pandangan dan proposal yang berkaitan dengan CT, seperti Barr & Stephenson di tahun 2011 yang melakukan penelitian dengan memasukkan CT ke dalam pendidikan komputer sains; Brennan & Resnick di tahun 2012 mengembangkan kerangka berpikir dari CT yang terdiri dari tiga dimensi yaitu *computational concepts* (konsep yang melibatkan pemrograman, seperti iterasi, paralelisme, dll.), *computational practices* (praktik yang dikembangkan perancang saat mereka terlibat dengan konsep, seperti men-debug proyek atau mencampur ulang karya orang lain), dan *computational perspectives* (perspektif yang dibentuk desainer tentang dunia di sekitar mereka dan tentang diri mereka sendiri); kemudian Wing pada 2008 yang mengemukakan bahwa CT akan mempengaruhi semua orang di setiap bidang. Pandangan ini menimbulkan tantangan pendidikan baru bagi masyarakat kita, terutama bagi anak-anak di Indonesia. Dalam berpikir tentang komputasi, perlu diselaraskan dengan tiga bidang yaitu sains, teknologi, dan masyarakat. Di sisi lain banyak peneliti yang mengembangkan strategi dan alat untuk mengevaluasi CT. Diantaranya yaitu Voogt, Fisser, Good, Mishra, dan Yadav



tahun 2015 melakukan penelitian tentang bagaimana strategi memasukkan CT ke dalam sekolah formal dan nonformal. Selanjutnya, Román-González, Pérez-González, & Jiménez-Fernández pada tahun 2017 mengembangkan asesmen dan validasi dari CT; dan Rojas-López & García-Peñalvo di tahun 2018 meneliti tentang bagaimana cara meningkatkan rasa percaya diri siswa dalam memecahkan masalah melalui keterampilan yang dilatihkan Computational Thinking. Banyaknya karya terbaru tentang topik ini menunjukkan bahwa bidang penelitian CT masih dalam tahap awal, fase embrionik. Analisis literatur dalam bidang CT mengungkapkan bahwa elemen-elemen kunci, umumnya terkait dengan matematika dan logika, adalah fitur-fitur penting yang terhubung dari CT.

Kalelioğlu et al. (2016) telah mengusulkan kerangka kerja terbuka (belum difinalisasi), di mana CT didefinisikan sebagai proses penyelesaian masalah dengan tahapan berikut: identifikasi masalah; pengumpulan data/representasi/analisis; pembuatan solusi/seleksi/perencanaan; implementasi solusi; dan penilaian/peningkatan solusi. Tahapan-tahapan ini adalah kategori tindakan yang konsisten dengan perspektif Wing; setiap kategori melibatkan beberapa tindakan kognitif. Identifikasi masalah menyiratkan abstraksi dan dekomposisi. Abstraksi merupakan elemen penting dari CT (Wing, 2008) dan memungkinkan kognitif untuk menghilangkan aspek yang tidak relevan dari masalah sehingga dapat fokus pada komponen penting. Kemampuan ini terkait erat dengan dekomposisi, yang memungkinkan kognitif untuk membagi masalah yang kompleks menjadi yang lebih kecil untuk mengusulkan solusi yang sesuai (Council, 2011). Pengumpulan data, representasi, dan analisis memungkinkan kognitif untuk memahami masalah secara mendalam melalui pemetaan pola, pengenalan pola, dan konseptualisasi. Pembuatan solusi,

seleksi, dan perencanaan membutuhkan pemikiran algoritmik. Lebih khusus lagi, pemikiran algoritmik memungkinkan kognitif untuk fokus pada struktur masalah dan solusi masalah, dan untuk memesannya dalam serangkaian langkah logis. Ini benar-benar terhubung ke konsep "algoritma," yang biasanya digambarkan sebagai "output alami" dari CT (Aho, 2012). Berpikir secara algoritmik menghasilkan algoritma output yang harus jelas, tidak ambigu, dan dapat ditiru (Kalelioğlu et al., 2016). Selama implementasi solusi, operasi otomatisasi, pemodelan, dan simulasi dapat dilakukan; selama tahap penilaian dan peningkatan, solusi dapat dinilai, diuji, didebug, dan digeneralisasi (diterapkan) untuk berbagai masalah.

Pada tahun 2014, Inggris yang pertama memasukkan CT ke dalam kurikulum sekolah dengan cara menambah materi pemrograman pada pembelajaran siswa di sekolah dasar dan menengah. Tujuan dimasukkannya materi tersebut bukan untuk mencetak programmer namun untuk mengenalkan CT sejak dini kepada siswa. Inggris berkeyakinan bahwa CT akan membuat siswa-siswa lebih cerdas dan lebih cepat memahami teknologi yang ada di sekitar mereka. Lain halnya dengan Amerika, di tahun yang sama lembaga dari Amerika (Code.org) menyelenggarakan beberapa acara untuk mempromosikan kegunaan dari belajar pemrograman, mulai dari "*Computer Science Education Week*" untuk anak-anak usia sekolah dan "*Hour of Code*" yang didukung oleh Bill Gates, Mark Zuckerberg, Jack Dorsey, William salah satu personil Black Eyed Peas.



# BAB 2

## PENGERTIAN

### *COMPUTATIONAL THINKING*

#### A. Pengertian *Computational Thinking* (CT)

Apa itu *Computational Thinking*? Untuk menjawab pertanyaan ini sebenarnya cukup menantang. Secara internasional, para peneliti *Computational Thinking* (CT) hingga saat ini telah menghabiskan banyak waktu untuk berdebat tentang bagaimana mendefinisikan CT. Tantangan utama dalam mencoba mendefinisikan CT adalah bagaimana kita memikirkan gagasan definisi. Ada ketegangan tersendiri yang melekat dalam upaya mendefinisikan CT dan hal tersebut berkaitan dengan pemikiran terkait kualitas “inti” CT *versus* kualitas “periferal” tertentu.

Terlepas dari tantangan tersebut, pada tahun 2011, sebuah workshop yaitu “*Workshop on Computational Thinking*” diselenggarakan yang mana banyak orang datang bersama untuk mengeksplorasi apa inti dari konsep CT yang seharusnya. Sejumlah sarjana, terutama pendidik ilmu komputer, telah berusaha untuk mendefinisikan CT. Beberapa orang di dalam workshop ini berpendapat dan mendukung adanya upaya dalam menentukan definisi CT yang ketat dan konsisten. Sebaliknya, Voogt dkk pada tahun 2015 berpendapat bahwa mencoba untuk mendefinisikan CT tidak perlu dilakukan. Menurutnya, memahami CT tidak boleh dilakukan dengan membuat definisi dalam arti biasa (dengan kata lain, membuat daftar kondisi yang harus dipenuhi sesuatu sebelum dianggap cocok). Voogt et al. berpendapat bahwa mendefinisikan CT

memiliki kesulitan yang sama seperti dengan mencoba mendefinisikan apa definisi 'permainan'. (Apakah setiap tujuan permainan, yang setidaknya satu pemain harus melawan pemain yang lain? Apakah setiap permainan harus memiliki konsep menang? Haruskah setiap permainan memasukkan unsur keberuntungan atau keacakan?) Seperti halnya pemahaman kita tentang permainan, mereka juga berpendapat bahwa pendekatan untuk mendefinisikan CT haruslah lebih kabur dan dianggap sebagai serangkaian persamaan dan hubungan yang saling bersilangan dan tumpang tindih.

Selain itu, ada alasan lain untuk membuat pendefinisian CT sebagai upaya yang menantang. Pertama, CT sangat terkait dengan *Computer Science* (CS), yang dengan sendirinya mengalami masalah untuk didefinisikan secara jelas. Seperti CS, CT mencakup serangkaian ide abstrak dan konkret. Keduanya dapat digunakan dalam penerapan *universal*, dan cakupan yang luas. Sehingga untuk membuat definisi CT yang kuat, juga sebanding dengan membuat CT sulit untuk didefinisikan secara ringkas.

CT merupakan ide yang baru dan juga ide yang lama. Merupakan ide yang baru dalam arti bahwa subjek tiba-tiba menjadi topik hangat yang diperdebatkan pada tahun 2006 setelah perbincangan yang dilakukan oleh Wing. Bagaimanapun juga, banyak ide inti dari CT telah dibahas selama beberapa dekade, dan di sepanjang jalan orang mengemasnya dengan cara yang berbeda. Misalnya pada tahun 1980, Seymour Papert dari Massachusetts *Institute of Technology* memelopori teknik yang disebutnya sebagai "*Procedural Thinking*". *Procedural Thinking* merupakan sebuah pemikiran yang sekarang kita anggap sebagai CT. Dengan menggunakan pemikiran prosedural, Papert bertujuan untuk memberikan metode kepada siswa untuk memecahkan masalah

menggunakan komputer sebagai alat. Idenya tersebut dimaksudkan agar siswa akan belajar bagaimana membuat solusi algoritmik yang kemudian dapat dilakukan oleh komputer, untuk ini dia menggunakan bahasa pemrograman Logo. Tulisan-tulisan Papert telah banyak menginspirasi munculnya CT, meskipun pada kenyataannya CT telah menyimpang dari ide asli ini dalam beberapa hal.

Namun demikian, selama 10 tahun setelah pembicaraan Wing, sejumlah definisi ringkas telah dicoba untuk dikemukakan. Sementara mereka menyebutkan gagasan yang serupa, tampaknya terdapat beberapa perbedaan terkait apa yang mereka katakan. Mungkin Voogt et al. benar, bahwa harapan terbaik untuk memahami CT adalah membangun konsep-konsep yang tumpang tindih dan saling bersilangan. Akan lebih baik jika kita mengetahui hasil diskusi beberapa ilmuwan atau peneliti CT terkait definisi CT. Beberapa hasil diskusi mereka terkait definisi CT akan dipaparkan sebagai berikut.

1. Wing (2008) berpendapat bahwa CT melengkapi pemikiran dalam matematika dan teknik dengan fokus pada perancangan sistem yang membantu memecahkan masalah kompleks yang dihadapi manusia. Konsep CT ini meliputi, abstraksi (alat mental komputasi, yang diperlukan untuk menyelesaikan masalah), *layers* (masalah perlu diselesaikan pada tingkat yang berbeda) dan hubungan antara *layers* dan abstraksi. Gagasan abstraksi dan kemampuan siswa untuk menghadapi berbagai tingkat abstraksi, serta untuk berpikir secara algoritmik dan untuk memahami konsekuensi dari skala (data besar), dan sangat mendasar bagi CT .
2. Aho (2012) lebih lanjut berpendapat bahwa CT merupakan proses berpikir yang terlibat dalam merumuskan masalah sehingga solusinya dapat direpresentasikan sebagai

langkah komputasi dan algoritmik. Bagian penting dari proses ini adalah menemukan model komputasi yang sesuai untuk merumuskan masalah dan mendapatkan solusinya

3. Denning (2009) mengemukakan CT memiliki sejarah panjang dalam ilmu komputer sejak 1950-an ketika dikenal sebagai pemikiran algoritmik yang berarti “orientasi mental untuk merumuskan masalah sebagai konversi dari beberapa input ke output dan mencari algoritma untuk melakukan konversi”. Beberapa ilmu komputer (CS) juga berpendapat bahwa pemrograman tidak penting dalam pengajaran pemikiran komputasi.
4. Lu dan Fletscher (2009) bahkan mengatakan bahwa penekanan pada pemrograman mungkin menghalangi siswa menjadi tertarik pada ilmu komputer. Singkatnya, pemikiran komputasi adalah cara konseptual untuk “secara sistematis, benar, dan efisien memproses informasi dan tugas” untuk menyelesaikan masalah yang kompleks.

Banyak orang di bidang pendidikan, khususnya teknologi pendidikan, setuju dengan komunitas pendidikan sains komputer bahwa CT adalah keterampilan di abad ke-21 yang penting. Berdasarkan definisi dan konsep inti CT seperti yang diberikan oleh para ilmuwan komputer, beberapa definisi telah muncul untuk apa CT dimasukkan ke dalam pelajaran wajib di sekolah. Kunci dalam semua definisi ini adalah fokus pada keterampilan, kebiasaan dan disposisi yang diperlukan untuk menyelesaikan masalah yang kompleks dengan bantuan komputasi dan komputer. CT mencakup kemampuan untuk membedakan beberapa tingkat abstraksi dan menerapkan penalaran matematis dan pemikiran berbasis desain. Mishra dan Yadav (2013) berpendapat bahwa CT melampaui interaksi komputer manusia yang khas; sebaliknya, mereka juga

berpendapat bahwa kreativitas manusia dapat ditambah dengan pemikiran komputasi, khususnya dengan penggunaan otomatisasi dan pemikiran algoritmik. Secara khusus, Mishra dan Yadav menyarankan bahwa pemikiran komputasi dapat memindahkan siswa dari konsumen teknologi menjadi produsen teknologi untuk menciptakan bentuk-bentuk baru alat bantu ekspresi dan menumbuhkan kreativitas.

Di samping itu, ada beberapa pihak lain yang juga mengembangkan definisi untuk CT diantaranya, lokakarya National Academy of Sciences pada tahun 2010 memberikan laporan yang membedakan pemikiran komputasi dari literasi komputer, pemrograman komputer, dan aplikasi komputer (yaitu permainan), dan memperluas istilah untuk memasukkan konsep inti dari disiplin ilmu komputer, seperti abstraksi, dekomposisi, generalisasi pola, visualisasi, pemecahan masalah, dan pemikiran algoritmik. Selanjutnya, inisiatif yang dilakukan oleh Furber pada tahun 2012, yang menawarkan definisi ringkas dari pemikiran komputasi sebagai “proses mengenali aspek komputasi di dunia yang mengelilingi kita, dan menerapkan alat dan teknik dari ilmu komputer untuk memahami dan alasan tentang sistem dan proses alami dan buatan”. Didukung lagi dengan adanya lokakarya yang diselenggarakan oleh *Computer Science Teachers Association (CSTA)* dan *the International Society for Technology in Education (ISTE)*. CSTA dan ISTE bekerja sama dengan para pemimpin dari pendidikan tinggi, industri, dan pendidikan K-12, mengembangkan definisi operasional pemikiran komputasi sebagai proses penyelesaian masalah yang mencakup, tetapi tidak terbatas pada elemen-elemen berikut: (a) Merumuskan masalah dengan cara yang memungkinkan kita untuk menggunakan komputer dan alat lain untuk membantu menyelesaikannya; (b) Mengelola dan menganalisis data secara logis; (c) Mewakili data melalui



abstraksi, seperti model dan simulasi; (d) Mengotomatiskan solusi melalui pemikiran algoritmik (misalnya serangkaian langkah yang dipesan); (e) Mengidentifikasi, menganalisis, dan mengimplementasikan solusi yang mungkin dengan tujuan mencapai kombinasi langkah dan sumber daya yang paling efisien dan efektif; dan (f) Generalisasi dan transfer proses penyelesaian masalah ini ke berbagai masalah.

## B. Konsep *Computational Thinking* (CT)

Konsep umum pemikiran komputasi dapat dibangun berdasarkan definisi-definisi yang telah dikemukakan oleh para peneliti CT seperti yang disebutkan di atas. Mereka menawarkan sedikit wawasan tentang bagaimana CT dapat diterapkan dalam praktek di bidang pendidikan. Definisi secara praktis dari pemikiran komputasi dan konstituennya diperlukan sebelum target pencapaian dan program pendidikan dapat dibuat di dalam kelas. Seorang pendidik dituntut untuk membantu siswanya dalam mempersiapkan masa depan di era digitalisasi yang semakin meningkat. Mengintegrasikan CT ke dalam aktivitas, pelajaran, dan kurikulum tidak hanya mendukung pengembangan keterampilan baru, tetapi juga meningkatkan pembelajaran dan keterlibatan dalam setiap disiplin ilmu.

CSTA dan ISTE telah menyediakan rubrik aktivitas untuk pemikiran komputasi pada tahun 2011, 2015, dan 2016. Tabel 1. berikut adalah daftar kegiatan atau serangkaian proses pada CT, yang diurutkan berdasarkan kata kunci.

*Tabel 1. Aktivitas Praktik Computational Thinking*

<b>Kata Kunci</b>	<b>Sumber</b>
Merumuskan, mengorganisir, menganalisis, memodelkan, abstraksi, berpikir algoritmik,	ISTE (2011)

<b>Kata Kunci</b>	<b>Sumber</b>
mengotomatisasi, efisiensi, generalisasi, mentransfer.	
Kreativitas, berpikir algoritmik, berpikir kritis, pemecahan masalah, bekerja sama	ISTE (2015); Oden et. al. (2015)
Analisis data, berpikir abstrak, berpikir algoritmik, pemodelan, representasi, pemecahan masalah menjadi komponen yang lebih sederhana, otomatisasi.	ISTE (2015) ( <i>Computational Thinker Definition</i> )

Seperti yang ditunjukkan oleh definisi tersebut, daftar kegiatan memberikan kerangka kerja bagi para pendidik, menggambarkan pencapaian pendidikan yang harus dituju dan menguraikan metode untuk pelaksanaan penilaian dan evaluasi pencapaian ini. Misalnya pendidik menggunakan rubrik-rubrik ini akan tahu bahwa mengajar dengan menggunakan media pembelajaran di kelas misalnya game, tidak hanya dimaksudkan untuk membantu siswa bersenang-senang saat merancang game. Namun mereka juga akan menggunakan pengalaman sebagai media untuk menanamkan beberapa konsep dan kemampuan yang diuraikan dalam tabel 1.

### C. Komponen Computational Thinking

Komponen dasar pemikiran komputasi juga merupakan sumber divergensi di antara para peneliti. Untuk menetapkan dasar analisis lebih lanjut, komponen yang digunakan oleh berbagai peneliti telah disediakan pada tabel 2. sebagai berikut.

*Tabel 2. Komponen Computational Thinking*

<b>Komponen</b>	<b>Sumber</b>
Abstraksi, Algoritma, Otomasi,	Barr & Stephenson

Komponen	Sumber
Dekomposisi Masalah, Paralelisasi, Simulasi	(2011)
Abstraksi, Otomasi, Analisis	Lee et al. (2011)
Abstraksi, Berpikir Algoritmik, Dekomposisi, Evaluasi, Generalisasi	Selby & Woollard (2013)
Abstraksi, Algoritmik, Dekomposisi, Debugging, Generalisasi	Angeli et al. (2016)
Abstraksi, Algoritma, Otomasi, Dekomposisi Masalah, Generalisasi	Wing (2006, 2008, 2011)

Bagian komponen CT yang tepat mungkin akan berbeda-beda, namun konsep-konsep penting yang mereka wakili sebagian besar memiliki inti yang sama yang dapat digunakan terhadap seluruh bidang. Kemampuan CT pada dasarnya adalah seperangkat keterampilan yang diperlukan untuk mengubah masalah di dunia nyata yang kompleks dan berantakan menjadi permasalahan yang mudah dipahami, sebagian didefinisikan menjadi bentuk yang dapat ditangani oleh komputer yang tidak memikirkan bantuan lebih lanjut dari manusia (BCS, 2014). Dengan demikian, buku ini akan menggunakan definisi abstraksi, generalisasi, dekomposisi, algoritmik, dan *debugging*.

Terlepas dari kenyataan bahwa saat ini tidak ada satupun definisi bulat dari pemikiran komputasi, tampaknya adil untuk menyimpulkan bahwa, berdasarkan literatur yang ditinjau dalam beberapa penelitian, para peneliti telah menerima bahwa pemikiran komputasi adalah proses berpikir yang memanfaatkan unsur-unsur abstraksi, generalisasi,

dekomposisi, pemikiran algoritmik, dan *debugging* (mendeteksi dan mengoreksi kesalahan). “Abstraksi” (*abstraction*) merupakan keterampilan yang menghilangkan karakteristik atau atribut dari suatu objek atau entitas untuk menguranginya menjadi seperangkat karakteristik yang mendasar (Wing, 2011). Sementara abstraksi mengurangi kompleksitas dengan menyembunyikan detail yang tidak relevan, “generalisasi” (*generalization*) mengurangi kompleksitas dengan mengganti beberapa entitas yang melakukan fungsi serupa dengan konstruksi tunggal (Thalheim, 2000). “Abstraksi” dan “generalisasi” sering digunakan bersama karena abstrak digeneralisasikan melalui parameterisasi untuk memberikan utilitas yang lebih besar. “Dekomposisi” (*decomposition*) merupakan keterampilan memecahkan masalah kompleks menjadi masalah yang lebih sederhana (National Research Council, 2010). Pemikiran “algoritmik” (*algorithmic*) merupakan keterampilan memecahkan masalah yang terkait dengan menyusun solusi langkah demi langkah untuk setiap permasalahan dan berbeda dengan pengkodean (yaitu, keterampilan teknis yang diperlukan untuk menggunakan bahasa pemrograman) (Selby, 2014). Selain itu, gagasan algoritmik tentang pengurutan (yaitu, perencanaan algoritma, yang melibatkan tindakan dalam urutan yang benar), dan gagasan algoritmik tentang aliran kontrol (yaitu, urutan di mana instruksi individu atau langkah-langkah dalam suatu algoritma dievaluasi) juga dipertimbangkan elemen penting dari pemikiran komputasi (Selby, 2014). “*Debugging*” merupakan keterampilan untuk mengenali ketika tindakan tidak sesuai dengan instruksi, dan keterampilan untuk memperbaiki kesalahan (Selby, 2014). Adapun unsur-unsur CT dapat dilihat pada Tabel 3 dibawah ini.

Tabel 3. Unsur-unsur Computational Thinking

No.	Unsur	Definisi
1.	<i>Abstraction</i>	Kemampuan untuk memutuskan informasi apa tentang suatu entitas/objek yang diketahui untuk disimpan dan informasi apa yang harus diabaikan (Wing, 2011)
2.	<i>Generalization</i>	Kemampuan untuk merumuskan solusi dalam istilah umum sehingga dapat diterapkan pada masalah yang berbeda (Selby, 2014)
3.	<i>Decomposition</i>	Kemampuan untuk memecahkan masalah yang kompleks menjadi bagian-bagian kecil atau sederhana yang lebih mudah untuk dipajami dan diselesaikan (National Research Council, 2010; Wing, 2011).
4.	<i>Algorithms</i>	Kemampuan untuk merancang serangkaian operasi/ tindakan secara bertahap (selangkah demi selangkah) tentang cara menyelesaikan suatu masalah (Selby, 2014).
	<i>a. Sequencing</i>	Kemampuan untuk menempatkan tindakan atau proses dalam urutan yang benar (Selby, 2014).
	<i>b. Flow of control</i>	Kemampuan mengurutkan urutan di mana instruksi/tindakan dieksekusi (Selby, 2014).
5.	<i>Debugging</i>	Kemampuan untuk mengidentifikasi, menghapus, dan memperbaiki kesalahan (Selby, 2014).

# **BAB 3**

## **COMPUTATIONAL THINKING (CT) PADA BIDANG TEKNOLOGI**

### **A. Hubungan CT dan Penggunaan Teknologi**

Penggunaan teknologi yang efektif menuntut bagaimana cara untuk menalar secara logis dan mampu untuk memprediksi apa yang akan terjadi serta mampu mengetahui mengapa sesuatu hal tersebut tidak bekerja. Penalaran logis juga penting ketika hendak mengevaluasi konten yang sudah dibuat sebelumnya. Dalam menyelesaikan suatu tugas yang menuntut keterampilan pada bidang teknologi perlu adanya suatu rancangan mulai dari awal hingga akhir. Seperti urutan langkah-langkah dalam membuat suatu aplikasi, sehingga dalam hal ini keterampilan untuk manajemen proyek perlu digunakan untuk memecah masalah yang kompleks menjadi bagian-bagian yang lebih kecil. Teknologi juga memiliki peran untuk yaitu untuk mempermudah dalam hal kolaborasi sehingga proyek dapat diselesaikan dengan tepat waktu. Adanya data, media, dan alat serta teknik yang digunakan untuk mengelola proyek tersebut, seseorang menjadi lebih tahu berbagai macam proses yang telah dilalui, yang pada akhirnya dapat digunakan untuk mengambil perspektif kritis pada pengerjaan proyek yang telah dilakukan sebelumnya. Tindakan ini perlu dilakukan untuk mempertimbangkan apakah seseorang tersebut telah mencapai hasil yang telah ditetapkan. Pemikiran seperti ini disebut dengan pemikiran komputasi. Pemikiran komputasi erat kaitannya dengan literasi digital.

Hampir setiap bidang dalam pekerjaan dan kehidupan telah didominasi oleh teknologi digital. Menjadi bagian dari digital saat ini bukan hanya tentang menjelajahi *web*, menggunakan teknologi untuk berkomunikasi, atau berpartisipasi dalam media sosial serta pembuatan suatu aplikasi (*programming*). Namun, melibatkan pengetahuan bagaimana hal-hal tersebut dibuat, memecahkan masalah, merencanakan sistem, berkontribusi melalui pembuatan, dan memahami konsekuensi sosial dan etika. Perkembangan teknologi, seperti komputer dalam berbagai bentuk dan *smartphone* yang telah menjadi bagian tak terpisahkan dari kehidupan sosial.

Seiring dengan perkembangan tersebut, penting bagi seseorang untuk berkontribusi dalam dunia digital seperti bagaimana berinteraksi dengan penggunaan teknologi, membuat keputusan tentang bagaimana teknologi membentuk dunia, dan menemukan solusi untuk masalah yang dihadapi. Oleh karena itu, pendidik membutuhkan dukungan untuk mengembangkan pemahaman tentang apa yang penting bagi para siswa untuk dipelajari sehingga siswa dapat memainkan peran aktif dalam dunia digital yang selalu berubah. Dalam banyak hal, pemikiran komputasi masuk dalam ranah Literasi Digital. Literasi digital adalah kemampuan untuk menemukan, mengatur, memahami, mengevaluasi, dan membuat informasi menggunakan teknologi digital untuk masyarakat berbasis pengetahuan. Literasi Digital juga dianggap sebagai kemampuan memahami konten dan alat digital yang dapat mencakup kemampuan yang lebih maju untuk menciptakan alat produk, dan layanan teknologi baru. Perlu dicatat bahwa ada beberapa aspek berpikir komputasi yang dapat dikembangkan tanpa teknologi digital.

Salah satu hal yang dapat diterapkan dalam aspek ini adalah studi komputer. Di dalam studi komputer terdapat pemrograman komputer yang lebih umum dikenal dengan istilah "Pengkodean". Pengkodean adalah tentang memberi tahu komputer apa yang harus dilakukan. Orang menggunakan bahasa pemrograman atau pengkodean tertentu yang mendefinisikan bagaimana kode harus ditulis sehingga komputer dapat memahaminya. Pengkodean dapat membantu siswa mengembangkan banyak aspek berpikir komputasi.

## **B. Pengajaran Pemrograman dengan CT**

Komputer merupakan suatu alat perkembangan teknologi yang dominan di awal abad ke-21 yang dapat mencakup semua aspek kehidupan ekonomi, sosial dan pribadi. Komputer merupakan perangkat keras yang dikendalikan oleh perangkat lunak, yaitu kode yang ditulis dalam bahasa pemrograman. Pemrograman adalah pembangun dari perangkat lunak. Dengan demikian pemrograman merupakan kegiatan mendasar. Namun, selama 70 tahun setelah komputer dibangun, tidak ada pedagogi yang digunakan untuk pengajaran pemrograman. Banyak kesulitan antara pemecahan masalah dan pemrograman. Setelah masalah dikarakterisasi dengan baik, maka relatif lebih mudah untuk mewujudkan solusi dari perangkat lunak. Namun, mengajarkan pemecahan masalah dalam pemrograman sulit. Pemecahan masalah tampaknya menjadi kreatif, holistik, dialektis, multi dimensi, dan proses yang berulang. Meskipun ada teknik yang lain, masalah tidak dapat diselesaikan dengan menghafal melainkan dengan teknik secara mekanik. Lebih jauh, secara historis, pendekatan dalam pengajaran pemrograman telah gagal menjelaskan kompleksitas dalam pemecahan masalah, hanya berfokus pada pemrograman itu sendiri, hanya mengeksplorasi



pemecahan masalah secara parsial dan dangkal. Untuk mengatasi hal tersebut pada tahun 2006 muncullah pendekatan terintegrasi untuk pemecahan masalah dan pemrograman. Pendekatan tersebut adalah *Computational Thinking* (CT). CT memiliki keuntungan luar biasa yang lebih dari pendekatan sebelumnya yang sangat menekankan pemecahan masalah. CT melibatkan pembuatan model pada dunia nyata, untuk memahaminya dan mengubahnya dapat diprediksi dengan suatu cara atau langkah. Seperti halnya pada pemrograman komputer yang menjadi model sebagai program. Pemrograman merupakan jembatan dari model dan komputer. Dalam pemrograman terdapat berbagai bahasa dan notasi yang digunakan untuk membuat model dan menulis program. Jadi berikut adalah kaitan antara Ilmu Komputer, TIK dan *Computational Thinking*,

- *Computational Thinking* memberi cara untuk memahami masalah
- TIK menawarkan masalah untuk berpikir secara *Computational Thinking* dalam mencari solusi
- Ilmu Komputer memberikan konsep untuk berpikir secara *Computational Thinking* dalam mencari sinergitas antara teori dan praktek, merespon kebutuhan sosial dari TIK dengan perangkat lunak, yaitu program.

Untuk mengetahui suatu masalah diselesaikan dan informasi yang relevan untuk menyelesaikan masalah serta informasi tersebut dapat berubah agar masalah dapat dipecahkan dengan menerapkan wawasan Niklaus Wirth bahwa Algoritma + Struktur Data = Program, maka untuk menyelesaikan masalah tersebut dapat dikatakan bahwa Informasi + Perhitungan = Solusi. Namun, bagian tersulit dari pemecahan masalah adalah mengkarakterisasi masalah. CT telah dipandang sebagai

pendekatan kunci untuk pemecahan masalah. CT didasarkan pada empat tahap, yaitu:

- Dekomposisi: kemampuan untuk memecahkan masalah menjadi submasalah
- Pengenalan Pola: kemampuan untuk melihat kesamaan, perbedaan, properti atau tren di suatu data
- Generalisasi Pola: kemampuan untuk mengekstrak detail yang tidak perlu dan menggeneralisasi yang diperlukan untuk mendefinisikan konsep atau ide secara umum
- Desain Algoritma: kemampuan untuk membangun proses langkah demi langkah yang berulang untuk menyelesaikan masalah khusus.

Secara lebih rinci, CT dapat dijelaskan seperti berikut ini,

- Dekomposisi yang didasarkan pada penggalian dasar masalah dan melibatkan
  - a) Memecah masalah menjadi sub-masalah yang lebih kecil
  - b) Mengidentifikasi sub-informasi yang diperlukan untuk menyelesaikan sub-masalah
- Identifikasi Pola yang didasarkan pada menemukan perbedaan dan melibatkan,
  - a) Mencari pola di antara masalah, untuk memikirkan apakah masalah sudah terlihat
  - b) Mencari pola dalam informasi, untuk mengetahui dan mempertimbangkan hubungan yang berguna dalam informasi
- Generalisasi Pola dan abstraksi yang merupakan dari identifikasi pola, dengan melibatkan berbagai perbedaan yang diidentifikasi dalam mengidentifikasi pola, untuk
  - a) Menemukan kasus umum untuk suatu masalah.
  - b) Memperjelas suatu informasi
- Desain Algoritma, dimana menyangkut mengenai,

- a) Urutan langkah-langkah dari informasi awal ke masalah yang sedang dipecahkan
- b) Bagaimana sub masalah tersebut terhubung
- c) Bagaimana informasi berubah diantara langkah-langkah

Mudah untuk menuliskan beberapa tahapan tetapi sulit untuk bagaimana berlaku secara praktis dalam pemecahan masalah untuk pemrograman. Pemrograman tidak akan sulit apabila mengetahui bagaimana menyelesaikan masalah.

### **1. Orientasi Pemrograman**

Pendekatan tertua untuk pemrograman adalah hanya menuliskan kode. Bahasa pemrograman tidak menawarkan pendekatan apapun untuk pemecahan masalah di luar cara merumuskan algoritma. Beberapa bahasa pemrograman yang digunakan dalam industri antara lain, *FORTRAN*, *COBOL*, *C/C++* dan *Java*. Keterampilan dalam menuliskan kode atau bahasa pemrograman sangatlah dibutuhkan bagi pemula. Seperti penggunaan bahasa pemrograman *Pascal* dan *BASIC*. Bahasa seperti *HTML*, banyak digunakan dalam pengajaran awal ketrampilan TIK untuk mendesain web, digunakan untuk tata letak dan memasang antar muka.

### **2. Diagram Alur Pemrograman**

Diagram telah lama digunakan dalam merancang perangkat lunak dan pengajaran pemrograman. Penggunaan diagram alur adalah untuk mendesai perangkat lunak. Paling sederhana, diagram alur disajikan pada kota untuk keputusan dan perintah, dihubungkan oleh tanda panah atau busur untuk menunjukkan aliran kontrol. Dengan demikian, terjadi penggabungan antara dekomposisi dan Algoritma pada tahapan CT. Diagram alur telah lama menjadi tambahan untuk belajar bahasa pemrograman. Namun, apabila dalam diagram

alur terdapat kotak perintah yang tidak sesuai maka program tersebut tidak bisa dijalankan.

### **3. Pemrograman Terstruktur**

Bahasa pemrograman awal sangat bergantung pada konstruksi dan orientasi pada sebuah platform yang mendasarinya, khususnya kemampuan untuk bercabang ke berbagai konstruksi dalam memori. Penggunaan pemrograman terstruktur secara luas dianjurkan berdasarkan penggunaan konstruksi dasar urutan, pilihan dan pengulangan. Seperti diagram alur, ini menggabungkan tahapan dekomposisi dan algoritma CT. Pemrograman terstruktur dilengkapi dengan pemrograman modular. Sebuah pendekatan di mana program disusun dari komponen yang diidentifikasi. Teknik untuk mengidentifikasi modul adalah menguraikan langkah-langkah utama, mulai dari diagram alur dan menyembunyikan informasi, dimana modul abstrak dari detail yang mendasarinya, biasanya terbentuk dari struktur data dan operasi terkait, penentuan objek orientasi awal.

### **4. Pemrograman Deklaratif**

Menyusul pengumuman proyek sistem komputer generasi ke-5 di Jepang, muncul minat baru pada pedagogik dalam penggunaan bahasa pemrograman untuk pengajaran awal pemrograman. Secara umum, bahasa-bahasa tersebut bersumber pada logika matematika. Pemrograman deklaratif sangat menekankan pada *Computational Thinking* pada teknik abstraksi dan identifikasi pola serta generalisasi, untuk mengetahui hubungan antara algoritma dan struktur data.

### **5. Pemrograman berorientasi pada Objek**

Pendekatan kontemporer untuk pengajaran awal pemrograman di perguruan tinggi adalah berorientasi pada objek, dimana komponen (kelas) merangkul kedua data struktur dan algoritma (metode) untuk memanipulasi

bahasapemrograman. Sekarang, Bahasapemrograman *Java* lebih mendominasi untuk pengajaranpemrograman berorientasi pada objek, diikuti oleh C++ dan bahasapemrograman *Phyton*. Salah satu pendekatan kontemporer yang populer untuk pengajaranpemrograman berorientasi objek adalah berbasis pada komponen terlebih dahulu.pemrograman berorientasi pada objek dilengkapi dengan UML (*Unified Modeling Language*), pendekatan diagram, yang mengacu pada pendekatan terstruktur sebelumnya.

## **6. Pendekatanpemrograman Paralel**

Pendekatanpemrograman Paralel adalah suatu pendekatan dimana penyelesaian masalah sitesis sangat penting untuk penggunaan sumber daya komputasi dan taksonomi masalah banyak digunakan. Ada tida metodologi untuk membangun program paralel yang memiliki resonansi kuat dengan *Computational Thinking* yaitu berkaitan dengan tahap-tahap pemartisian dan komunikasi, mengidentifikasi tugas dan tahap pemetaan. Tahap partisi melibatkan dekomposisi informasi dan fungsional suatu komponen. Kemudian, komponen-komponen tersebut dikelompokkan untuk mengoptimalkan alokasi ke tugas processor yang tersedia untuk meminimalkan komunikasi informasi.

## **7. Pemograman berbasis masalah**

Pemograman berbasis masalah, peserta didik didorong untuk memahami, memodifikasi dan memperluas solusi yang ada untuk masalah maupun sub masalah. Selain itupemrograman berbasis masalah sebenarnya juga berbasis proyek belajar. Disini, penekanan pendekatanpemrograman berbasis adalah pemahaman tentan solusi yang spesifik dalam konteks masalah, bukan pada bagaimana solusi itu berasal atau digeneralisasikan ke solusi yang lain.pemrograman berbasis masalah, juga mendorong peserta didik untuk mengeksplorasi

pada fase awal, peserta didik dilengkapi dengan konsep dasar. Pendekatan berbasis masalah, menggunakan tahap perbaikan dan perulangan pada dekomposisi masalah.

Dari berbagai pendekatan pemrograman di atas, dapat diketahui bahwa ada beberapa tahap dari CT yang digunakan, yaitu dekomposisi dan penggunaan desain algoritma. Jadi, dalam pendekatan berorientasi pemrograman dapat dikatakan bahwa penggunaan CT dapat dirumuskan dalam berbagai pendekatan pemrograman di atas.

## **8. Struktur Informasi**

Kunci dalam pemrograman adalah bagaimana caranya untuk mengarakterisasi informasi. Informasi adalah gabungan dari elemen dasar dan struktur. Konsep pemrograman yang harus dilakukan dengan mewakili dan mengimplementasikan informasi dari suatu masalah dan menerapkan struktur informasi masalah. Setelah itu adalah bagaimana menemukan informasi, lalu informasi tersebut dikarakterisasi dan diorganisasikan secara sederhana, linear, tidak teratur dan diperbaiki. Generalisasi struktur informasi adalah dengan menerapkan pemikiran komputasi sehingga terbentuklah struktur informasi sebagai data tipe abstrak dengan cara untuk membuat struktur baru, menambahkan informasi, memeriksa apakah ada informasi, mencari informasi, mengubah informasi serta terakhir adalah menghapus informasi. Dari informasi ke perhitungan atau setiap langkah dari informasi tentunya kita dapat melihat operasi pada tipe abstrak seperti kata kerja suatu bahasa. Selanjutnya, membuat kalimat dengan menerapkan kata kerja ke kata benda. Yaitu, struktur komputasi pada struktur informasi ke dalam algoritma.

Jadi, pemrograman dapat menerapkan berpikir secara CT dimana Dekomposisi + Abstraksi + Pola + Algoritma sehingga dapat memunculkan suatu solusi yaitu dengan informasi +

perhitungan dari masalah. CT merupakan kerangka untuk menyelesaikan suatu masalah dengan tahapan-tahapan yang sesuai sehingga didapatkannya suatu pengalaman dalam memecahkan masalah.

### **C. Belajar Pemrograman dengan CT Melalui Permainan Digital**

Pada bagian ini akan diuraikan mengenai model permainan inovatif untuk belajar ketrampilan berpikir secara *Computational Thinking* melalui permainan digital. Permainan ini telah dirancang untuk melatih dan mengembangkan keterampilan dalam CT dengan pengetahuan pemrograman dalam ilmu komputer, sangat penting untuk konsep identitas dengan definisi yang tepat. Ilmu komputer sangat penting untuk memahami masalah dan untuk mencari solusi yang efektif untuk dikembangkan. CT dapat diakses dan diterapkan oleh semua orang, seperti peserta didik yang digunakan untuk mengembangkan ketrampilan dalam CT pada pemrograman khususnya. Untuk itulah keterampilan CT dapat diterapkan pada pembelajaran berbasis *game*. Pembelajaran pada jenis ini cenderung fokus pada aspek motivasi permainan dan tujuan utamanya dan keterlibatan peserta didik dengan berbagai hasil belajar. Terlepas dari upaya ini, beberapa pembelajaran menunjukkan cara dari suatu permainan dapat dikaitkan dengan CT dan bagaimana pendidikan pemrograman dapat didukung dengan bermain *game*. Dengan mengeksplorasi kerangka kerja pembelajaran berbasis permainan dapat meningkatkan keterampilan CT untuk mendukung pembelajaran pemrograman.

Tugas dalam pemrograman sering diakui sebagai kegiatan yang membuat frustrasi dan menuntut siswa untuk dapat membuat suatu program. Banyak penelitian menyatakan

bahawa metode pengajaran yang buruk, rendahnya interaksi dengan siswa dan kurangnya minat adalah suatu masalah dalam belajar pemrograman. Siswa perlu menunjukkan bagaimana memahami pola yang jelas dalam pemrograman daripada hanya berfokus pada sintaksis dan semantik pemrograman komputer. Untuk itu, pemikiran CT telah menjadi titik focus dari pembelajaran terbaru terutama dalam disiplin ilmu komputer untuk mengintegrasikan CT ke dalam kurikulum dasar.

Penggunaan alat pemrograman visual cerita dan pembuatan permainan digital yang menarik dan memotivasi diusulkan sebagai kerangka kerja untuk mengajar CT dan pemrograman komputer. Alat pemrograman visual sering dianggap ideal karena memungkinkan siswa untuk membuat berbagai abstraksi secara cepat tanpa perlu kode program yang berlebihan untuk membuat skenario kerja, siswa dapat memilih karakter dan perilaku yang berbeda dari tempat pilihan dan kemudian membangun adegan dimana setiap adegan berisi karakter dan perilakunya. Abstraksi kompleks dapat dibuat melalui penggabungan atribut karakter dan perilaku yang pasti membutuhkan pemahaman tentang urutan pemrograman, persyaratan, lietrasi dan metode. Lebih lanjut, kegiatan ini menghapus aturan sintaksis bahasapemrograman asli dan mengizinkan siswa untuk memikirkan representasi secara pragmatik. Terlepas dari keunggulan ini, penelitian terbaru berpendapat bahwa CT tidak sama dengan pemrograman meski alat pemrograman visual dapat berguna untuk menciptakan antusiasme dalam pemrograman. Dapat dikatakan apakah siswa mampu melakukan kegiatan pemrograman dan mengembangkan CT melalui mekanisme seperti ini. Misalnya, siswa mungkin mengembangkan pemrograman yang baik



dengan berlatih atau secara spontan mendapatkan strategi CT, tetapi hanya sedikit umpan balik yang terjadi.

Hal ini mungkin terjadi ketika siswa membuat skenario linear yang berfungsi tanpa memperhatikan pola yang dapat digunakan kembali dan praktik pemrograman lain. Dalam hal ini, siswa dapat membuat *output* (luaran) yang berfungsi dengan merancang strategi pemrograman, seperti pernyataan yang diulang berkali-kali, tanpa menggunakan loop. Karena, siswa belum memiliki tingkat pengetahuan yang diperlukan untuk mengembangkan solusi yang lebih baik. Dengan kata lain, alat pemrograman visual memerlukan mekanisme *debug* untuk mendukung siswa dalam mengembangkan abstraksi serta mempraktekkan pemrograman dengan baik.

CT yang efisien harus mendukung siswa dalam mengembangkan praktik pemrograman dengan panduan yang jelas melalui umpan balik yang relevan yang akan masuk akal bagi mereka. Pendekatan populer lain yang diikuti untuk mengajar pengantar pemrograman adalah penggunaan permainan kreasi digital melalui tahap pengembangan permainan. Penugasan dalam konteks ini dapat dikategorikan berbasis tugas modul, permainan papan strategis dan pengembangan aplikasi mobile. Tujuan dari pendekatan ini adalah untuk memahami konsep pemrograman abstrak dan tidak perlu belajar tentang membangun *game*. Namun, dalam pengembangan sebuah *game*, membutuhkan upaya yang cukup besar dikarenakan siswa tidak hanya membuat *game* sederhana, sementara pembuatan *game* tingkat lanjut membutuhkan penggunaan konsep spesifik *game* yang kompleks. Meskipun konsep-konsep tersebut penting untuk dipelajari, perlu untuk mendukungnya dengan CT sebagai pola dalam konteks dan aplikasi mandiri, yang berarti bahwa sekali siswa mengerti caranya secara konseptual dalam menyajikan pola, maka

mereka dapat mentransfernya dan menggunakan dalam konteks yang lain. Belajar pemrograman melalui *game* dapat menggunakan pemandu dalam model pembelajaran berbasis modul. Salah satu aspek kunci dari pembelajaran berbasis ini adalah struktur konstruksivitas yang melekat.

Pemecahan masalah dalam pembuatan *game* digital adalah dengan pemikiran secara logis melalui penggunaan berbagai komputasi. Termasuk dekomposisi masalah untuk mengidentifikasi masalah atau menghasilkan solusi alternatif. Pada level ini siswa membedakan masalah dan memutuskan apakah masalah ini dapat diselesaikan secara komputasi. Selanjutnya siswa mengevaluasi masalah dan menentukan sesuai kriteria untuk mengembangkan abstraksi yang berlaku. Membangun algoritma melibatkan pembangunan prosedur langkah demi langkah untuk menyelesaikan masalah tertentu. Dengan membangun algoritma, siswa menulis logaritma untuk melihat kemampuan mereka dalam CT. Akhirnya, kelak apabila siswa dapat memainkan sebuah permainan maka mereka mampu memvisualisasikan cara kerja struktur data dari permainan tersebut.

Pemilihan teknik pada algoritma merupakan bagian penting dari pemikiran secara komputasi yang saat ini berkembang. Selain itu yang tak kalah pentingnya adalah *debugging*. *Debugging* adalah menganalisis masalah dan mengevaluasinya. *Debugging* adalah pusat dari CT karena melibatkan pemikiran kritis dan prosedural. Tahap berikutnya adalah simulasi. Simulasi juga dapat disebut dengan membangun model sesuai dengan algoritma yang telah dituliskan. Dalam simulasi, siswa merancang model atau menjalankan model sebagai tes untuk membuat suatu keputusan tentang keadaan yang harus dipertimbangkan saat

menyelesaikan abstraksi mereka. Setelah siswa melakukan simulasi, langkah selanjutnya adalah sosialisasi.

Sosialisasi mengacu pada aspek sosial dari CT, yang melibatkan koordinasi, kerjasama dan kompetisi selama tahap penyelesaian masalah, pembangunan algoritma, *debugging* dan simulasi. Pada tahap sosialisasi ini memungkinkan pertukaran pendapat dan mendorong pengembangan strategi. Sosialisasi juga merupakan salah satu fitur khas pada CT. Untuk itulah, pengembangan sebuah *game* yang dapat mengembangkan keterampilan sesuai dengan CT dengan menggunakan pemodelan pemrograman. Salah satu *game* yang dikembangkan adalah *Program Your Robot*. *Program Your Robot* adalah sebuah prototipe *game* yang dapat mengembangkan keterampilan untuk mendukung dalam berlatih berbagai ketrampilan pada CT serta belajar dan menggunakan konstruksi pemrograman.



Gambar 1. Tampilan Game Program Your Robot

Pada Game ini, pemain dapat mensimulasikan permainan teka-teki, dimana pemain dapat mengontrol robot dengan memberikan berbagai perintah. Hasil dari game ini adalah untuk membantu robot mencapai pintu teleportasi dimana setiap level hanya berisi satu pintu teleportasi. Selama bermain *game*, pemain perlu merancang algoritma sebagai solusi

melalui penggunaan pemrograman dan simbol untuk menemukan jalan sehingga robot dapat mencapai pintu teleportasi. Perintah yang dapat diberikan pemain kepada robot dibagi menjadi dua bagian, perintah untuk tindakan dan perintah pemrograman. Perintah tindakan digunakan untuk menggerakkan robot yang terdiri dari kisi-kisi perintah, sedangkan perintah pemrograman secara tidak langsung mempengaruhi tindakan ini dan memfasilitasi perancangan algoritma. Kedua jenis perintah tersebut dapat digunakan dengan diseret dari *bar* yang telah disediakan dan menjatuhkan ke area spesifik yang disebut dengan slot. Pemain memainkan *game* dengan menyeret dan menjatuhkan sejumlah perintah ke dalam sebuah slot dengan urutan yang mereka pilih, selama pemain memiliki slot yang kosong.

Untuk menyelesaikan sebuah level, pemain harus memindahkan robot mereka ke pintu teleportasi, mengaktifkan robot sehingga pemain dapat melanjutkan ke level berikutnya. Selanjutnya, pemain harus menyelesaikan setiap level dalam waktu yang tersedia. Ketika pemain maju ke level selanjutnya maka *grid* atau tempat bermain akan meluas dan permainan bertambah menjadi lebih kompleks. Di setiap level pemain juga menemukan item yang dapat ditangkap oleh robot itu. Item tersebut tersebar sepanjang *game* secara acak setiap kali pemain mulai bermain dari level satu ke level selanjutnya. Dengan demikian pendekatan semacam ini disajikan ke pemain lain yang bermain di level berbeda secara signifikan dari masalah yang disajikan kepada pemain lain yang bermain di level yang sama.

*Game* ini menghadiahkan pemain dengan fitur baru, seperti item baru, slot baru dan robot musuh yang harus dihindari. Ada tiga jenis tindakan yang dapat dilakukan pemain ketika mereka selesai merancang algoritma mereka. Yang

pertama, untuk menjalankan perintah dengan menekan tombol “run”. Saat perintah *run-time* di dalam slot sudah terkunci kemudian dieksekusi oleh robot sesuai aturan urutan pemain. Tindakan kedua adalah menghapus semua perintah yang jatuh ke dalam slot dengan menekan tombol “clear”. Fungsi ini memungkinkan pemain untuk membersihkan dan mendesain ulang solusi mereka. Jenis tindakan terakhir adalah *men-debug* solusi. Setiap kali pemain membuat suatu kesalahan ketika ada pergerakan robot yang keliru, maka mereka dapat menggunakan *mode debug*. *Mode debug* ini digunakan untuk mengidentifikasi potensi kesalahan. *Mode debug* sangat mendukung pemain selama pengembangan proses dan menawarkan bantuan dalam mengatasi suatu masalah.

Dengan fitur ini, *game* tidak hanya mendorong para pemain untuk mengembangkan praktik yang baik sari solusi *debugging*, tetapi juga mendorong mereka untuk berpikir kritis tentang solusi mereka, karena mereka harus menganalisis suatu masalah dan membangun algoritma untuk menyelesaikan mekanisme *game* yang menyediakan berbagai bentuk umpan balik secara otomatis sebagai reaksi terhadap tindakan pemain. Pemain menggunakan umpan balik ke aturan permainan atau *game* abstrak, *debug*, dan kembangkan strategi untuk meraih kemenangan. Dalam tahap menjalankan desain-*debug* ini, ada tiga aspek kunci dari pemikiran komputasi, analisis, abstraksi dan otomatisasi yang hadir dalam sebuah *game*.

Perintah pemrograman yang diintergrasikan kedalam *game* adalah representasi simbolis dari kontruksi awal pemrograman, yaitu urutan pemrograman, pemilihan (pengambilan keputusan), pengulangan dan metode. Untuk tujuan inilah, *game* ini dirancang dengan lima level dimana setiap level, pemain diperkenalkan ke tantangan yang baru. Di level 1, pemain menemukan bagaimana urutan pemrograman

bekerja dan mereka membangun algoritma mereka hanya dengan menyeret dan menjatuhkan tindakan perintah ke metode utama, yang berfungsi untuk mengendalikan robot. Di level 2 dan 3, pemain belajar bagaimana caranya untuk mengubah urutan pemrograman dan cara membuat suatu pola pemrograman dengan merancang dan memanggil kembali pemrograman yang telah ditentukan pengguna apabila difungsikan.

Di level 4, pemain belajar bagaimana menggunakan seleksi pada pemrograman. Robot pemain harus melewati musuh untuk melewati Level 4. Musuh dari robot tersebut tidak dapat meninggalkan lokasi permainan tetapi dapat menghentikan jalan dari robot pemain yang muncul secara acak. Untuk itulah pemain dituntut untuk melihat situasi dan bagaimana cara untuk melewati hal tersebut. Sementara itu juga, pemain dapat belajar untuk mengembangkan algoritma mereka dalam berpikir terhadap suatu masalah dan juga menyelesaikan bagaimana untuk menyeleksi solusi untuk permasalahan tersebut.

Di level 5, pemain mempraktekkan bagaimana mengombinasikan pengulangan dengan fungsi untuk menghindari serangkaian dinding. Karena dinding ini dirancang dalam formasi serupa, pemain dapat menggabungkan pengulangan dengan fungsi untuk mencapai kemenangan. Pada tingkat dasar, *game* ini adalah sebuah sistem dimana pemain harus menyesuaikan perilaku robot yang ada dengan menggunakan urutan pemrograman, kondisionan atau pengkondisian, pengulangan dan metode. Struktur *game* ini terdiri dari dua jenis aturan yaitu operasional dan berturut-turut. Aturan operasional adalah pedoman yang dibutuhkan pemain dalam rangka untuk memainkan *game*. Aturan ini dikirim melalui layar tutorial sebagai kotak dialog di awal setiap level. Di layar

tutorial tersebut menjelaskan fitur *game* dan bagaimana perintah pemrograman bekerja. Sedangkan aturan berturut-turut dirancang untuk menjadi struktur logis yang mendasari *game*. Hal ini merupakan prosedur yang tidak tertulis mengenai pengembangan strategi yang efisien untuk memenangkan *game*.

Literatur saat ini mendefinisikan abstrak aturan pada *game* sebagai cara yang bagus untuk menunjukkan ketrampilan berpikir secara CT. Oleh karena itu, peningkatan kerangka kerja akan dapat terus mendorong pemain untuk menemukan aturan berturut-turut yang mendasari sistem penilaian *game*. Sistem penilaian digunakan agar pemain paham tentang aturan permainan dan menyusun strategi untuk mengoptimalkan perilaku robot sesuai dengan aturan. Untuk mencapai tersebut, sistem penilaian menghitung skor pemain berdasarkan tiga kriteria, yaitu: barang koleksi atau item, slot dan perintah pemrograman. Mengumpulkan barang koleksi merupakan tantangan tambahan di *game* ini dan seringkali membutuhkan pola berulang yang dibuat karena jumlah slot dalam *game* terbatas. Barang koleksi atau item yang dikumpulkan lebih, semakin tinggi skor poin yang mereka dapatkan. Selain itu, semakin sedikit jumlah slot pemain yang digunakan untuk membangun algoritma, semakin tinggi pula skor mereka. Dengan demikian, solusi yang diinginkan terletak pada pembuatan pola yang berulang dengan slot sedikit mungkin, yang hanya dapat dicapai dengan menggunakan perintah pemrograman secara akurat. Yang terakhir, dengan sistem penilaian inilah dapat memotivasi pemain untuk mengontruksikan pemrograman sehingga dapat menciptakan strategi untuk menang. Perhitungan ini untuk setiap level dan mengukur seberapa baik pemain memahami cara membangun sebuah program untuk pekerjaan. Sebagai contoh, pemain dapat menyelesaikan level 2 dan level 3 tanpa menggunakan fungsi

tunggal. Namun, menciptakan solusi yang tidak efisien dan nantinya akan menghasilkan skor yang rendah.

Selanjutnya karena slot metode utama terbatas, tidak selalu memungkinkan untuk mengakumulasi semua barang atau item yang didapatkan juga akan mendapat skor yang rendah. Di sisi lain, pemain dapat mencapai skor yang tinggi ketika mereka menunjukkan pemahaman yang mendalam konsep pemrograman yang digunakan dalam sebuah *game*, seperti ketika mereka membuat fungsi *rekursif* atau *loop* yang dipadukan dengan fungsi. Oleh karena itu, membangun algoritma yang efisien menggambarkan *game* yang baik serta dapat mengembangkan kemampuan CT yang dibahas di atas. Akhirnya dengan hal tersebut mendorong mereka untuk berpikir strategi mana yang paling efisien. Akibatnya, pemain dapat menganalisis apakah solusi mereka cukup atau tidak untuk membuat strategi kemenangan dalam suatu *game*.

#### **D. Mengembangkan Keterampilan CT dan Belajar Pengantarpemrograman**

Dari beberapa bagian dari aktivitas *game* yang telah dideskripsikan di atas telah diketahui bahwa bagaimana siswa dapat mengembangkan keterampilan mereka dalam bermain *game*. Pada Tabel 3.1. di bawah ini dapat dilihat bagaimana definisi keterampilan dan karakter CT. Ilustrasi bagaimana kemampuan kognitif dapat dikembangkan melalui *game* dan menvalidasi keterampilan yang diuraikan. Selain itu, karena setiap konstruk pada pemrograman memiliki aksi yang sesuai dalam *game*. Berikut adalah visualisasi bagaimana pemrograman bekerja.



Tabel 4. Karakter CT pada Game Program Your Robot

Tugas	Kategori kemampuan yang diasosiasi CT	Aktivitas <i>Game</i>	Kategori pada Kemampuan Rasional
Identifikasi Masalah dan Dekomposisi	<i>Problem Solving</i> (Pemecahan Masalah)	Membantu Robot untuk mencapai pintu teleportasi. Mengaktifkan lampu robot ketika robot berada di depan teleportasi	CT mendeskripsikan penyelesaian masalah berdasarkan berbagai macam kegiatan belajar. Selain itu <i>game</i> merupakan kegiatan memecahkan masalah yang menyenangkan
Menciptakan pola yang efisien dan berulang	<i>Building Algorithms</i> (Membangun Algoritma)	Menciptakan algoritma solusi untuk melengkapu semua level dengan slot yang sedikit. Gunakan fungsi untuk menciptakan pola yang berulang.	CT sebagai eksekusi dari algoritma yang mana mengikuti beberapa level sampai level tertinggi tercapai.
Mempraktik-	<i>Debugging</i>	Tekan tombol	Debungging

kan <i>Mode Debug</i>		debug untuk melihat solusi aloritmamu untuk mendeteksi potensi kesalahan.	adalah komponen esensi dalam CT dan Kegiatan pemrograman.
Mempraktikkan <i>Mode Runtime</i>	<i>Simulation</i> (Simulasi)	Mengobservasi perpindahan robot selama waktu berjalan. Dapatkah kamu mengikuti solusi dari aloritmamu? Apakah kamu mengobservasi tingkah laku robot tersebut?	Ide dasar dari CT adalah mengembangkan model dan menstimulasikan masalah.
<i>Brainstorming</i>	<i>Socialization</i> (Sosialisasi)	Memeriksa strategi untuk mencapai kemenangan dan membandingkan solusi mereka denganmu. Apa saran yang akan kamu berikan	Perspektif sosial dari <i>Computational Thinking</i> adalah mendiskusikan komputasi ke dalam beberapa bagian dari informasi atau logika

		kepada dirimu sendiri dan mereka untuk mendapatkan skor yang lebih di dalam <i>game</i> ?	beberapa pemain selama proses debungging, simulasi, dan membangun algoritma
--	--	---	---

# BAB 4

## *COMPUTATIONAL THINKING*

### PADA SAINS

#### **A. Revolusi Komputer Memengaruhi Cara Berpikir Seseorang Tentang Sains, Eksperimen, dan Penelitian.**

Sebuah revolusi yang mendalam telah terjadi di seluruh bidang sains. Revolusi tersebut adalah revolusi komputasi. Revolusi komputasi telah mengubah ilmu pengetahuan dengan memungkinkan segala macam penemuan baru melalui teknologi informasi. Dalam sejarah sains dan teknologi, terdapat dua karakter yaitu pelaku eksperimen dan ahli teori. Pelaku eksperimen adalah mereka yang mengumpulkan data untuk mengungkapkan kapan hipotesis berhasil dan kapan tidak. Sedangkan ahli teori yaitu mereka yang mendesain model matematika untuk menjelaskan apa yang sudah diketahui dan menggunakan model apa untuk membuat prediksi tentang apa yang tidak diketahui. Kedua karakter tersebut berinteraksi satu sama lain karena hipotesis dapat berasal dari model, dan yang diketahui berasal dari model dan data sebelumnya. Eksperimen dan ahli teori telah aktif dalam sains jauh sebelum komputer muncul.

Ketika pemerintah mulai menugaskan proyek untuk membangun komputer elektronik pada 1940-an, para ilmuwan mulai mendiskusikan bagaimana mereka akan menggunakan mesin ini dan hampir semuanya ingin menghasilkan sesuatu. Para peneliti melihat ke komputer untuk analisis data, memilah-milah kumpulan data besar untuk pola statistik. Para ahli teori memandang mereka untuk menghitung persamaan model

matematika. Banyak model seperti itu dirumuskan sebagai persamaan diferensial, yang mempertimbangkan perubahan fungsi pada interval yang sangat kecil. Pertimbangkan misalnya fungsi umum  $f$  dari waktu ke waktu (disingkat  $f(t)$ ). Misalkan perbedaan  $f(t)$  dari waktu ke waktu menghasilkan persamaan lain, disingkat  $g(t)$ . Relasi ini ditulis sebagai  $\frac{d(f)}{dt} = g(t)$ . Kemudian dapat dihitung nilai perkiraan  $f(t)$  dalam serangkaian perubahan kecil dalam langkah waktu, disingkat  $\Delta t$ , dengan persamaan perbedaan  $f(t + \Delta t) \approx f(t) + \Delta t.g(t)$ . Perhitungan ini dapat dengan mudah diperluas ke beberapa dimensi ruang dengan persamaan perbedaan yang menggabungkan nilai-nilai pada node yang berdekatan dari sebuah *grid*. Dalam karyanya yang terkumpul, John von Neumann, *polymath* yang membantu merancang komputer program tersimpan pertama, mendeskripsikan algoritme untuk menyelesaikan sistem persamaan diferensial pada *grid* diskrit.

Menggunakan komputer untuk mempercepat pekerjaan tradisional para peneliti dan ahli teori adalah revolusi tersendiri. Namun ada hal yang lebih dari itu, yaitu ilmuwan yang menggunakan komputer mendapati diri mereka secara rutin merancang cara baru untuk memajukan sains. "Simulasi" adalah contoh utama. Dengan mensimulasikan aliran udara di sekitar sayap dengan jenis persamaan (disebut *Navier-Stokes*) yang dipecah di atas kisi yang mengelilingi pesawat yang disimulasikan, sebagian besar insinyur penerbangan menghilangkan kebutuhan akan terowongan angin dan uji penerbangan. Para astronom juga mensimulasikan tabrakan galaksi, dan ahli kimia mensimulasikan kerusakan pelindung panas pesawat luar angkasa saat memasuki atmosfer. "Simulasi" memungkinkan para ilmuwan untuk mencapai teori dan tidak bisa melakukan eksperimen. Hal ini menjadi cara

baru dalam melakukan sains. Ilmuwan menjadi desainer komputasi serta peneliti dan juga ahli teori.

Contoh penting lainnya tentang bagaimana komputer telah mengubah cara berpikir sains adalah paradigma baru yang memperlakukan proses fisik sebagai proses informasi, yang memungkinkan lebih banyak untuk dipelajari tentang proses fisik dengan mempelajari proses informasi. Ahli biologi telah membuat kemajuan signifikan dengan teknik ini, terutama dengan pengurutan dan pengeditan gen. Analisis data juga telah menemukan bahwa model pembelajaran mendalam memungkinkan mereka membuat prediksi proses yang sangat akurat di banyak bidang. Untuk kuantitas yang diprediksi, proses nyata berperilaku sebagai proses informasi. Kedua pendekatan ini sering digabungkan, seperti ketika proses informasi menyediakan simulasi untuk proses fisik yang dimodelkannya.

### **1. Asal usul Istilah Ilmu Komputasi**

Istilah ilmu komputasi, dan istilah pemikiran komputasi yang terkait, mulai digunakan secara luas selama tahun 1980-an. Pada tahun 1982, fisikawan teoretis Kenneth Wilson menerima Hadiah Nobel dalam bidang fisika karena mengembangkan model komputasi yang menghasilkan penemuan baru yang mengejutkan tentang perubahan fase dalam material. Dia merancang metode komputasi untuk mengevaluasi persamaan kelompok renormalisasi, dan menggunakannya untuk mengamati bagaimana suatu material berubah fase, seperti arah gaya magnet dalam ferrimagnet (di mana ion yang berdekatan memiliki muatan berlawanan tetapi tidak sama). Dia meluncurkan kampanye untuk mendapatkan pengakuan dan penghormatan terhadap ilmu komputasi. Dia berpendapat bahwa semua disiplin ilmu memiliki masalah yang sangat

sulit – “tantangan besar” – yang akan menghasilkan komputasi besar-besaran. Dia dan para visioner lainnya menggunakan istilah ilmu komputasi untuk cabang ilmu baru yang menggunakan komputasi sebagai metode utama mereka. Mereka melihat komputasi sebagai paradigma baru dalam sains, melengkapi paradigma teori dan eksperimen tradisional. Beberapa dari mereka menggunakan istilah *Computational Thinking* untuk proses berpikir dalam melakukan ilmu komputasi – mendesain, menguji, dan menggunakan model komputasi. Mereka meluncurkan gerakan politik untuk mengamankan pendanaan untuk penelitian ilmu komputasi, yang berpuncak pada Undang-Undang *High Performance Communicating and Computing* (HPCC) yang disahkan pada tahun 1991 oleh Kongres AS.

## **2. Pemikiran komputasi muncul dari dalam bidang ilmiah – itu tidak diimpor dari ilmu komputer.**

Sangat menarik bahwa ilmu komputasi dan pemikiran komputasi dalam sains muncul dari dalam bidang ilmiah – keduanya tidak diimpor dari ilmu komputer. Memang, para ilmuwan komputer lambat untuk mengikuti gerakan ini. Dari awal ilmu komputer di tahun 1940-an, ada cabang kecil tapi penting dari bidang ini yang mengkhususkan diri pada metode numerik dan perangkat lunak matematika. Ilmuwan komputer ini memiliki ketertarikan terbesar pada ilmu komputasi dan merupakan yang pertama menerimanya. Komputasi telah terbukti sangat produktif untuk kemajuan sains dan teknik sehingga hampir setiap bidang sains dan teknik telah mengembangkan cabang komputasi. Misalnya, pada tahun 2001 David Baltimore, peraih Nobel bidang Biologi, mengatakan bahwa biologi adalah ilmu informasi. Kemajuan terbaru dalam biologi telah melibatkan pemodelan DNA, pengurutan, dan

pengeditan. Kita bisa berharap tren ini terus berlanjut, dengan komputasi yang masuk lebih dalam ke setiap bidang, termasuk ilmu sosial dan humaniora. Banyak orang akan belajar menjadi desainer dan pemikir komputasi.

### **3. Apa itu Pemikiran Komputasi?**

Berpikir komputasi secara umum didefinisikan sebagai keterampilan mental yang memfasilitasi desain proses otomatis. Meskipun istilah ini berasal dari awal ilmu komputer di tahun 1950-an, istilah ini menjadi populer setelah tahun 2006 ketika para pendidik melakukan tugas membantu semua anak menjadi pengguna komputasi yang produktif sebagai bagian dari pendidikan STEM. Jika seseorang dapat mempelajari apa yang merupakan pemikiran komputasi sebagai keterampilan mental, orang tersebut mungkin dapat mengajak/memengaruhi banyak generasi muda untuk mempelajari sains dan mempercepat kemajuan sains.

Sebagian besar definisi yang dipublikasikan hingga saat ini dapat diparafrasekan sebagai berikut: "Pemikiran komputasi adalah proses berpikir yang terlibat dalam merumuskan masalah sehingga solusi mereka direpresentasikan sebagai langkah komputasi dan algoritme yang dapat dilakukan secara efektif oleh agen pemrosesan informasi." Definisi ini, bagaimanapun, penuh dengan ide-ide yang bermasalah. Pertimbangkan kata "merumuskan". Orang-orang secara teratur merumuskan permintaan agar mesin melakukan sesuatu untuk mereka tanpa harus memahami cara kerja komputasi atau cara mendesainnya. Istilah "agen informasi" juga bermasalah – ini dengan cepat membuka pintu ke keyakinan yang salah bahwa prosedur langkah demi langkah yang diikuti oleh manusia adalah algoritme. Banyak orang mengikuti prosedur "langkah demi langkah" yang tidak dapat direduksi menjadi algoritme



dan diotomatiskan oleh mesin. Definisi kabur ini mempersulit pendidik untuk mengetahui apa yang seharusnya mereka ajarkan dan bagaimana menilai apakah siswa telah mempelajarinya. Dan "proses berpikir" apa yang terlibat? Definisi yang diterbitkan mengatakan mereka termasuk membuat representasi digital, mengurutkan, memilih alternatif, mengulang loop, menjalankan tugas paralel, mengabstraksi, mendekomposisi, menguji, debugging, dan menggunakan kembali. Tapi ini bukanlah deskripsi yang lengkap. Untuk menjadi kontributor yang bermanfaat, seorang *programmer* juga perlu cukup memahami suatu bidang keilmuan agar dapat mengungkapkan masalah dan metode penyelesaian yang sesuai dengan bidang tersebut. Sebagai contoh, tim ilmuwan komputasi dinamika fluida mengundang ilmuwan komputer PhD untuk bekerja dengan mereka, hanya untuk menemukan bahwa ilmuwan komputer tidak cukup memahami dinamika fluida untuk menjadi berguna. Mereka tidak dapat berpikir dalam istilah dinamika fluida komputasi. Anggota tim yang lain akhirnya memperlakukan ilmuwan komputer seperti pemrogram daripada rekan kerja, mereka sangat kecewa. Tampaknya proses dari pemikiran komputasi harus mencakup praktisi terampil di bidang di mana komputasi akan digunakan.

Semua kesulitan ini menunjukkan bahwa kata "berpikir" bukanlah apa yang benar-benar diminati dan bukan kemampuan untuk merancang perhitungan. "Desain" lebih mencakup seluruh dimensi dari pengguna, menguji prototipe untuk melihat bagaimana pengguna bereaksi, dan membuat penawaran teknologi yang memperhatikan kepentingan pengguna. Oleh karena itu desain komputasi adalah istilah yang lebih akurat. Ini jelas merupakan seperangkat keterampilan, bukan sekumpulan pengetahuan mental tentang pemrograman.

#### 4. Apa itu Model Komputasi?

Aspek penting dari desain komputasi (atau pemikiran) adalah mesin yang akan melakukan langkah-langkah otomatis. Tetapi sebagian besar perancang komputasi tidak secara langsung mempertimbangkan perangkat keras mesin itu sendiri; sebagai gantinya mereka bekerja dengan model komputasi, yang merupakan mesin abstrak – pada dasarnya lapisan perangkat lunak di atas perangkat keras yang menerjemahkan program menjadi instruksi untuk perangkat keras. Desainer tidak peduli dengan pemetaan model ke mesin sebenarnya, karena itu adalah pekerjaan simulasi yang ditangani oleh teknisi perangkat lunak. Dalam ilmu komputasi, model yang paling banyak dibicarakan adalah mesin *Turing*, yang ditemukan pada tahun 1936 oleh pelopor komputasi yaitu Alan Turing. Modelnya terdiri dari barisan sel tersusun berupa pita yang dapat bergerak maju mundur, komponen aktif baca/tulis pita yang memiliki status perhitungan serta dapat mengubah/menulisi sel aktif yang ada di pita tadi, dan suatu kumpulan instruksi bagaimana komponen baca/tulis ini harus melakukan modifikasi terhadap sel aktif pada pita, serta bagaimana menggerakkan pita tersebut. Pada setiap langkah dalam komputasi, mesin ini akan dapat mengubah isi dari sel yang aktif, mengubah status dari komponen baca/tulis, dan mengubah posisi pita ke kiri atau ke kanan. Mesin *Turing* adalah model komputasi yang paling umum – apa pun yang menurut orang dapat dihitung secara wajar, dapat dihitung oleh mesin Turing. Tetapi mesin Turing terlalu primitif untuk merepresentasikan komputasi sehari-hari dengan mudah. Dengan setiap bahasa pemrograman baru, ilmuwan komputer menentukan mesin abstrak terkait yang mewakili entitas yang diprogram oleh bahasa tersebut. *Software* yang disebut *compiler*

kemudian menerjemahkan operasi bahasa pada mesin abstrak menjadi kode mesin pada *hardware* yang sebenarnya.

Model mesin Turing dan bahasa pemrograman semuanya bertujuan umum – mereka menangani apa pun yang dapat dihitung. Tapi sebenarnya manusia sering bekerja dengan model yang jauh lebih kuat dan yang sangat berguna. Salah satu yang paling umum adalah mesin keadaan hingga, yang terdiri dari rangkaian logika, satu set rangkaian sakelar *flip-flop* untuk merekam keadaan saat ini, dan jam yang detaknya memicu transisi keadaan. Mesin keadaan hingga memodelkan banyak pengontrol elektronik dan juru bahasa perintah sistem operasi. Jaringan saraf tiruan yang khas adalah model yang lebih sederhana. Ini adalah jaringan gerbang bebas lingkaran yang dimodelkan setelah neuron. Gerbang diatur dalam lapisan dari yang terhubung ke input yang terhubung ke output. Pola *bit* pada masukan melewati jaringan dan menghasilkan keluaran. Tidak ada keadaan untuk dicatat atau diingat. Setiap sinyal dari satu lapisan ke lapisan berikutnya memiliki bobot yang terkait. Jaringan dilatih oleh algoritme yang menyesuaikan bobot secara berulang hingga jaringan menjadi sangat baik dalam menghasilkan keluaran yang diinginkan. Beberapa orang menyebut pembelajaran mesin ini karena sirkuit terlatih (penyesuaian berat) memperoleh kemampuan untuk mengimplementasikan suatu fungsi dengan melihat banyak contoh. Ini juga disebut pembelajaran dalam karena lapisan dan bobot tersembunyi di sirkuit. Banyak kemajuan modern dalam kecerdasan buatan dan analitik data telah dicapai oleh sirkuit ini. Simulasi sirkuit ini sekarang memungkinkan jutaan node dan puluhan lapisan. Ketika keluar dari ilmu komputer, maka akan ditemukan sedikit orang yang berbicara tentang mesin Turing dan mesin keadaan hingga. Mereka berbicara alih-alih pembelajaran mesin dan simulasi model komputasi yang

relevan dengan bidang mereka. Di setiap bidang, desainer komputasi memprogram model atau mendesain model baru – atau keduanya.

## **5. Apakah Semua Masalah Dapat Diselesaikan dengan Komputasi?**

Masalah penting dengan model komputasi adalah kompleksitas – berapa lama waktu yang dibutuhkan untuk mendapatkan hasil? Berapa banyak penyimpanan yang dibutuhkan? Seringkali model komputasi yang akan memberikan jawaban yang tepat tidak mungkin, terlalu mahal, atau terlalu lambat. Desainer komputasi meniasati ini dengan heuristik – perkiraan cepat yang menghasilkan solusi perkiraan dekat dengan cepat. Validasi eksperimental seringkali merupakan satu-satunya cara untuk mendapatkan kepercayaan pada heuristik. Jaringan saraf tiruan untuk pengenalan wajah adalah heuristik. Tidak ada yang tahu algoritme pasti untuk mengenali wajah. Tapi kami tahu cara membangun jaringan saraf cepat yang dapat melakukannya dengan benar di sebagian besar waktu.

Komputasi telah berubah secara dramatis sejak saat pemodelan komputasi tumbuh. Pada 1980-an, sistem hosting untuk model tantangan besar adalah superkomputer. Saat ini sistem hosting adalah seluruh internet, yang sekarang lebih sering disebut *cloud*, yaitu sistem data yang terdistribusi secara masif dan sumber daya pemrosesan di seluruh dunia. Layanan *cloud* komersial memungkinkan seseorang memobilisasi penyimpanan dan daya pemrosesan yang dibutuhkan. Selain itu, tidak lagi dibatasi untuk berurusan dengan komputasi terbatas – komputasi yang memulai, menghitung, mengirimkan keluaran, dan berhenti.

Tetapi ada batasan penting tentang apa yang dapat dilakukan dengan semua daya komputasi. Salah satu batasannya adalah bahwa sebagian besar metode komputasi memiliki fokus yang tajam – metode tersebut sangat baik pada tugas tertentu yang dirancang untuknya, tetapi tidak untuk tugas yang tampaknya serupa. Kita seringkali dapat mengatasi batasan itu dengan desain baru yang menutup celah dalam desain lama. Pengenalan wajah adalah contohnya. Batasan lainnya adalah ada banyak masalah yang tidak bisa diselesaikan sama sekali dengan komputasi. Beberapa di antaranya yang murni teknis, seperti menentukan dengan inspeksi kapan program komputer akan berhenti atau memasuki *loop* tak terbatas. Banyak lainnya adalah masalah yang sangat kompleks yang menampilkan teknologi yang terkait dengan komunitas sosial dan tidak ada jawaban yang jelas – yang dikenal sebagai masalah jahat. Banyak masalah jahat disebabkan oleh efek gabungan dari miliaran orang yang menggunakan teknologi. Misalnya, produksi lebih dari satu miliar lemari es melepaskan cukup *fluorokarbon* untuk mengganggu perlindungan atmosfer bagian atas terhadap sinar matahari yang berlebihan. Jutaan mobil menghasilkan begitu banyak kabut asap sehingga beberapa kota menjadi tidak sehat. Satu-satunya solusi untuk masalah ini akan muncul dari kerjasama sosial di antara kelompok-kelompok yang sekarang menawarkan pendekatan yang saling bersaing dan bertentangan. Meskipun teknologi komputasi dapat membantu dengan memvisualisasikan efek skala besar dari tindakan individu, namun hanya tindakan sosial yang akan menyelesaikan masalah yang disebabkan tersebut.

Namun, ilmu komputasi adalah kekuatan yang kuat dalam sains. Ini menekankan "cara komputasi" dalam melakukan sains dan mengubah para praktisinya menjadi

desainer komputasi yang terampil (dan pemikir) di bidang sains mereka. Desainer komputasi menghabiskan banyak waktu mereka untuk menemukan, memprogram, dan memvalidasi model komputasi, yang merupakan mesin abstrak yang memecahkan masalah atau menjawab pertanyaan. Desainer komputasi harus menjadi pemikir komputasi serta praktisi di bidangnya sendiri. Desain komputasi akan menjadi sumber pekerjaan yang penting di masa depan.

## **B. Berpikir Komputasi di Kelas Sains**

Pentingnya *Computational Thinking* (CT) sebagai tujuan pendidikan sains semakin diakui (Quinn, Schweingruber, Keller, 2012; Wilensky, Brady & Horn, 2014). Mengajar CT dalam konteks sains tidak hanya menyajikan kepada siswa tentang gambaran sains yang lebih otentik seperti yang dipraktikkan saat ini, tetapi juga meningkatkan akses ke mode berpikir yang kuat dan keterampilan yang dapat dipasarkan untuk banyak karier (Levy & Murname, 2004). Diperkirakan pada tahun 2020, satu dari setiap dua pekerjaan STEM akan ada di komputasi (ACMPATHWAYS Report 2013). Namun, siswa dari kelompok yang secara historis kurang terwakili di bidang STEM (seperti perempuan dan ras minoritas) cenderung tidak mendaftar di kelas ilmu komputer (Margolis, 2008; Margolis & Fisher, 2003) dan dengan demikian tidak secara tradisional terpapar dengan praktik CT. Memasukkan CT ke dalam kurikulum dapat meningkatkan akses untuk semua siswa, terutama mereka yang kurang terwakili di *Computer Science* (CS), dengan menanamkan praktik CT dalam mata pelajaran seperti biologi, kimia, dan fisika, yang diharapkan diambil oleh semua siswa sekolah menengah. Meskipun ini tidak memastikan bahwa siswa ini akan secara pribadi termotivasi untuk terlibat dalam kurikulum CT, namun

mereka setidaknya akan terpapar pada praktik CT dan diberi kesempatan untuk mempelajarinya. Untuk alasan yang diberikan di atas, kami percaya bahwa mengembangkan praktik CT dalam konteks mata pelajaran sains adalah usaha produktif. Namun, karakter praktik CT dalam disiplin ilmu belum dipahami dengan baik, begitu pula cara membuat kurikulum dan penilaian yang mengembangkan dan mengukur praktik tersebut (Grover & Pea, 2013). Untuk mengatasi kesenjangan ini, beberapa kelompok telah bekerja secara eksplisit mengkarakterisasi praktik inti CT sebagai tujuan pembelajaran yang spesifik dan menggunakannya untuk memandu pengembangan kurikulum dan penilaian sains. Salah satu langkah yang dilakukan yaitu dengan mengembangkan tujuan pembelajaran berdasarkan taksonomi teoritis CT di STEM. Taksonomi terdiri dari empat praktik CT: Praktik Data, Praktik Pemodelan dan Simulasi, Praktik Pemecahan Masalah Komputasi, dan Praktek Berpikir Sistem. Kemudian menerjemahkan elemen dari setiap untai taksonomi ke dalam tujuan pembelajaran melalui proses yang melibatkan wawancara dengan ilmuwan komputasi dan umpan balik dari guru sains sekolah menengah.

# BAB 5

## COMPUTATIONAL THINKING (CT) PADA MATEMATIKA

### A. Konsep Matematika dalam CT

Matematika merupakan inti dari banyak disiplin ilmu karena dapat melatih pikiran dalam representasi, logika, dan berbagai manipulasi data. Disamping itu matematika juga merupakan kemampuan penalaran, kreativitas, pemecahan masalah, dan keterampilan komunikasi yang efektif. Matematika bisa menjadi sulit untuk dipelajari jika minat dan pendekatan belajar-mengajar tidak menghubungkan teori dengan masalah dalam dunia nyata. Pengetahuan matematika sangat penting dalam kehidupan sehari-hari. Dari saat kita bangun di pagi hari hingga saat kita pergi tidur, kita bergantung pada pemahaman dan kemampuan kita untuk melakukan tugas-tugas sederhana berbasis matematika. Matematika mengasah kekuatan penalaran dan pemecahan masalah, yang mengarah pada kejelasan yang lebih baik dalam keterampilan komunikasi dan kemungkinan kreatif yang lebih tinggi (Chan, 2010; Snalune, 2015; Biswas, 2015; Kong, 2019).

Menurut Brennan dan Resnick's (2012), CT melibatkan tiga aspek: perspektif komputasi, praktik komputasi, dan konsep komputasi. Ketiga aspek tersebut diimplementasikan dengan cara berbeda untuk menyesuaikan dengan konteks dan kebutuhan yang berbeda. Dengan memungkinkan siswa untuk membuat, berbagi, dan mendiskusikan masalah berbasis kesenangan (*game*) untuk dipecahkan oleh orang lain, mereka akan mendapatkan kepercayaan pada pemahaman dan kreativitas mereka sendiri dalam pemecahan masalah.



Matematika terkait erat dengan CT karena melibatkan pengenalan pola, struktur masalah dan variabel yang dapat dipakai dengan nilai yang berbeda, seperti “dekomposisi”, untuk mengaktifkan modularitas dan pemecahan masalah yang lebih mudah; desain algoritma, karena penekanannya pada penalaran logis; dan “generalisasi”, yaitu dari banyak contoh perumusan prinsip. Dengan demikian, jika CT dapat digunakan untuk mencari yang terbaik cara memecahkan masalah, maka solusi yang lebih kreatif dapat ditemukan dan ditemukan untuk mengatasi kesulitan.

Selain itu, semakin banyak anak-anak menghabiskan banyak waktu di *internet* dan *video game*, bahkan ada konten-konten yang berisi kekerasan. Akademi Anak dan Remaja Amerika Psychiatry (2015) menunjukkan bahwa kecanduan *video game* dapat menyebabkan kesehatan mental dan masalah sosial pada anak-anak karena sebagian besar waktunya dihabiskan di dalam ruangan untuk bermain game *online/video*. Lebih jauh lagi, bermain *video game* selama berjam-jam bisa memengaruhi prestasi akademik siswa, karena siswa akan tidak fokus di kelas dan menjadi kurang berinteraksi dengan orang lain. Christopher (2015) menegaskan hal itu bahkan terjadi ketika pembelajaran matematika, mereka cenderung hanya membahas *video game*. Seiring berjalannya waktu, komunikasi sosial mereka akan terhenti dan mereka akan mengalami masalah dalam berkomunikasi dengan anggota masyarakat pada umumnya.

Mempertimbangkan manfaat pembelajaran Matematika dan keterampilan CT, sebaiknya perhatian anak-anak harus dialihkan dari *video game* yang tidak sehat ke *game* berbasis Matematika yang lebih mendidik. Dalam hal ini berfokus pada penggunaan matematika untuk mengajarkan keterampilan CT dengan cara yang menyenangkan. Studi sebelumnya oleh Lee,

Wong, dan Ee (2017) menggunakan gamifikasi berorientasi objek pendekatan untuk meningkatkan keterampilan matematika dan CT siswa. Pembelajaran yang berpusat pada siswa mendukung penggunaan pendekatan inkuiri seperti *game*. Selain itu, dalam 11 studi kasus Kong (2019) tentang pengembangan profesional guru, ia menemukan bahwa untuk skalabilitas praktik pengajaran, konten dan pengembangan guru model perlu dikembangkan dari perspektif kemitraan berbagai pemangku kepentingan. Selanjutnya, prinsip dapat berfungsi sebagai strategi *top-down* untuk diuji di ruang kelas, dimana pola sukses dari *e-learning* dapat diidentifikasi dalam jangka panjang untuk memperkaya pemahaman teoritis dan praktis siswa dan guru.

Media komputasi yang direkonstruksi oleh Abelson (2012) adalah salah satu karya awal pemrograman visual yang menyenangkan. Selanjutnya, Scratch yang menyenangkan muncul (berbentuk *game* yang mengasah kemampuan CT). Desain permainan membutuhkan pengenalan pola, dan variasi dari pola ini dan mekanika permainan yang melekat akan menimbulkan tantangan/pertanyaan menarik bagi para pemain. Dengan demikian, siswa akan mengutak-atik konsep Matematika dan berlatih dengan berbagai cara untuk menyempurnakan perspektif komputasi mereka.

Penerapan pemikiran komputasi (CT) dapat ditemukan di sebagian besar lembaga di beberapa negara. Wing (2006) mendefinisikan CT sebagai "sikap dan keterampilan yang berlaku secara universal yang membuat semua orang, tidak hanya ilmuwan komputer, akan bersemangat untuk belajar dan menggunakannya. Berkenaan dengan sikap, CT mendorong pola pikir terbuka yang menantang batasan dan mengejar penyempurnaan. Pola pikir terbuka sesuai dengan (a) merumuskan kembali masalah yang tampaknya sulit menjadi

satu solusi yang kita kenal, baik dengan reduksi, transformasi, atau simulasi; (b) memilih representasi yang tepat dari suatu masalah atau memodelkan aspek yang relevan dari suatu masalah; (c) berpikir secara rekursif; dan (d) memproses alternatif dan kendala secara bersamaan. Hal ini selanjutnya mengarah pada dua konsep ilmu komputer utama, yaitu dekomposisi dan abstraksi. Akibatnya, CT memungkinkan identifikasi pola data yang lebih baik, pemahaman dan pemecahan masalah yang lebih baik, sistem dan perilaku manusia, dan analisis yang lebih efektif.

## **B. CT dalam Meningkatkan Keterampilan Matematika**

Komputasi tidak hanya melibatkan pemrograman dan mempraktikkan keterampilan komputasi tetapi juga pemikiran rekursif, pencocokan model, dan penalaran komposisi. Dengan adanya komputasi, akan meningkatkan pemikiran algoritmik, pemikiran paralel, dan pemikiran praktis juga. Hasilnya, melalui CT, siswa dapat mengembangkan proses pemecahan masalah dan disposisi. Hal tersebut sangat penting untuk kemajuan sistem dan fungsi komputer, terutama analisis dan desain serta justifikasi terhadap masalah dan solusi (Togyer & Wing, 2017).

CT menunjukkan bagaimana pengetahuan dan keterampilan individu yang diperoleh dari mata pelajaran komputasi dapat membawa dampak positif bagi masyarakat. Dalam Ilmu Ekonomi, seseorang dapat menemukan pola siklus dalam naik turunnya perekonomian suatu negara; dalam Matematika, langkah-langkah untuk memecahkan masalah kalkulus; dalam Kimia, prinsip ionisasi. Oleh karena itu, orang yang memahami dan memiliki pengetahuan tentang CT dalam studinya akan melihat korelasi antara mata pelajaran teoritis

dan lingkungan (antara pembelajaran *indoor*/formal dengan pembelajaran *outdoor*/informal).

Abelson (2012) lebih lanjut menunjukkan bahwa agar CT dapat berkembang melampaui media komputasi yang dapat dibangun ulang, perlu dilengkapi dengan nilai komputasi yang mempromosikan akses terbuka. Hal ini akan menyebabkan popularitas gerakan *open-source*. Akibatnya, tujuan CT untuk membangun kemampuan dan kreativitas individu (Weintrop, Holbert, Horn, & Wilensky, 2016) dan gerakan *open-source* telah mempengaruhi banyak profesi dan kurikulum ilmu komputer di negara berkembang.

### C. Pemikiran Komputasi dan Matematika

Perkembangan komputasi sebagai bidang pengetahuan selama abad ke-20 dan ke-21 telah terkait erat dengan Matematika, terutama jika seseorang menganggapnya model teoritis pertama mereka, seperti *Turing Machine* misalnya, diciptakan untuk secara matematis menunjukkan kelayakan mengotomatiskan perhitungan numerik. Denning [2005] mengemukakan bahwa, selama perkembangan komputasi, kegiatannya didasarkan pada metode dan pengetahuan yang dipinjam dari tiga bidang ilmu yaitu Ilmu Pengetahuan Alam, Teknik dan Matematika. Metode berbasis eksperimen dari Ilmu Pengetahuan Alam digunakan untuk definisi algoritma heuristik, misalnya, sedangkan desain dan pengembangan perangkat lunak adalah aplikasi yang jelas dari metode rekayasa dan teknik. Representasi simbolik dan metode deduksi berbasis aksioma dari Matematika menentukan dasar untuk mendemonstrasikan kebenaran dan kinerja algoritma.

Pendidikan K-12 adalah pendidikan yang sangat kompleks, lingkungan yang sangat terpolitisasi di mana berbagai prioritas, ideologi, pedagogi, dan ontologi semuanya

bersaing untuk mendapatkan perhatian. Pemikiran komputasi diposisikan sebagai pendekatan praktis untuk pemecahan masalah dan dapat diterapkan secara luas pada lintas mata pelajaran.

[Barcelos dan Silveira 2013] melakukan analisis pedoman kurikulum untuk Matematika di pendidikan dasar di tiga negara (Amerika Serikat, Brasil dan Chili) dan membandingkan keterampilan yang disebutkan dalam pedoman ini kepada mereka yang hadir dalam berbagai definisi dan aplikasi CT. Membangun hubungan dan mengidentifikasi pola: keterampilan yang terkait situasi di mana seorang siswa hendaknya mengidentifikasi keteraturan dan menyimpulkan atau menetapkan aturan formasi. Dalam kasus Matematika, biasanya terkait dengan keteraturan numerik, tetapi juga terkait dengan keterampilan abstraksi inti untuk masalah pemecahan yang dijelaskan oleh Polya [2004]: analogi, generalisasi dan spesialisasi.

Membangun model deskriptif dan representatif: dengan menerapkan keterampilan ini, siswa harus mampu menggunakan bahasa matematika atau algoritmik untuk membangun model yang menjelaskan situasi umum, seperti untung dan rugi, kinematika, jajak pendapat, dan banyak lainnya. Dukungan komputasi untuk membangun dan menguji model (dengan menggunakan *spreadsheet*, perangkat lunak pembuatan bagan dan pemrograman alat bahasa) dapat membawa potensi untuk menerapkan CT pada mata pelajaran di sekolah.

CT menggambarkan proses kognitif yang berkaitan dengan abstraksi dan dekomposisi untuk memungkinkan pemecahan masalah menggunakan sumber daya komputasi dan strategi algoritmik. Ada beberapa kesamaan dalam keterampilan CT dengan yang berhubungan dengan Matematika. Polya,

masuk karya klasiknya tentang keterampilan pemecahan masalah, menunjukkan abstraksi yang didefinisikan sebagai kombinasi dari analogi, generalisasi dan spesialisasi, dan masalah keterampilan dekomposisi sangat penting bagi siswa Matematika untuk berhasil dalam memecahkan masalah [Polya 2004]. Banyak aktivitas CT yang terkait dengan keterampilan dan konten Matematika telah dilaporkan di literatur ilmiah dalam beberapa tahun terakhir, yang memotivasi perlunya analisis rinci hasil pendidikan diperoleh melalui inisiatif ini.

#### **D. Mendefinisikan Berpikir Komputasi untuk Matematika dan Ruang Kelas Sains**

Sains dan matematika menjadi upaya dalam komputasi. Fakta ini tercermin dalam Standar Sains Generasi Berikutnya yang baru-baru ini dirilis dan keputusan untuk memasukkan "pemikiran komputasi" sebagai praktik ilmiah inti. Dengan tambahan ini, dan peningkatan kehadiran komputasi dalam matematika dan konteks ilmiah, urgensi baru telah datang ke tantangan untuk mendefinisikan pemikiran komputasi dan memberikan landasan teoritis untuk bentuk apa yang harus diambil di sekolah dalam ruang kelas sains dan matematika. Definisi pemikiran komputasi untuk matematika dan sains dalam bentuk taksonomi yang terdiri dari empat kategori utama: praktik data, praktik pemodelan dan simulasi, praktik pemecahan masalah komputasi, dan praktik berpikir sistem.

Bagaimana pemikiran komputasi terkait dengan pemikiran matematika, pemikiran algoritmik, atau pemecahan masalah? Bagaimana kaitannya dengan bidang ilmu komputer? Untuk sejauh mana pemrograman komputer terlibat? Apakah berpikir komputasi selalu membutuhkan komputer?

## **1. Mengapa Membawa Pemikiran Komputasi ke Ruang Kelas Matematika dan Sains?**

Motivasi utama untuk memperkenalkan komputasi praktik berpikir ke dalam ruang kelas sains dan matematika adalah sifat disiplin ilmu ini yang berubah dengan cepat dipraktikkan di dunia profesional (Bailey dan Borwein 2011; Foster 2006; Henderson et al. 2007). Selama 20 tahun terakhir, hampir setiap bidang yang terkait dengan sains dan matematika telah melihat pertumbuhan tentang komputasi. Contohnya termasuk Bioinformatika, Statistik Komputasi, Kemometri, dan Neuroinformatika. Peningkatan ini menunjukkan pentingnya komputasi sehubungan dengan matematika, sains, dan bidang Sains, Teknologi, Teknik, dan Matematika (STEM) yang lebih luas telah diakui baik oleh mereka yang berada dalam komunitas pendidikan STEM dan organisasi pendidikan ilmu komputer (ACM/IEEE-CS Satgas Gabungan pada Kurikulum Komputasi 2013). Memasukkan praktik komputasi ke dalam ruang kelas matematika dan sains memberi siswa pandangan yang lebih realistis dan lebih mempersiapkan siswa untuk mengejar karir (Augustine 2005; Gardner 1983), dan membantu membekali siswa agar lebih paham tentang STEM warga di masa depan.

Selanjutnya, dari perspektif pedagogis, penggunaan alat dan keterampilan komputasi yang bijaksana dapat memperdalam pembelajaran matematika dan konten sains (Guzdial 1994; Eisenberg 2002; Dewan Riset Nasional 2011a, b; Kemerahan dan Wilson 1993; Repenning et al. 2010; Sengupta dkk. 2013; Sherin 2001; Wilensky 1995; Wilensky dkk. 2014; Wilensky dan Reisman 2006). Kebalikannya juga benar yaitu bahwa sains dan matematika memberikan makna konteks (dan serangkaian masalah) di mana komputasi pemikiran dapat diterapkan (Hambruch et al. 2009; Jona et al. 2014; Lin et al.

2009; Wilensky dkk. 2014). Ini berbeda secara nyata dari pengajaran pemikiran komputasi sebagai bagian dari kursus mandiri di mana tugas-tugas siswa diberikan cenderung terpisah dari masalah dunia nyata dan aplikasi. Penerapan dalam dunia nyata atau kehidupan sehari-hari merupakan hal penting dalam upaya memotivasi keberagaman dan partisipasi yang berarti dalam kegiatan komputasi dan ilmiah (Blikstein 2013; Chinn dan Malhotra 2002; Confrey 1993; Margolis dan Fisher 2003; Margolis 2008; Ryoo dkk. 2013). Hubungan timbal balik ini yaitu menggunakan komputasi untuk memperkaya pembelajaran matematika dan sains dan menggunakan konteks matematika dan sains untuk memperkaya pembelajaran komputasi.

## **2. Peran Tumbuh Komputasi dalam Matematika dan Sains**

Lanskap sains sedang berubah. Kemajuan terbaru di komputasi berkecepatan tinggi dan metode analitis telah menciptakan alat yang ampuh untuk memahami fenomena secara keseluruhan pada spektrum penyelidikan manusia. Di beberapa bidang ilmiah, seperti biologi molekuler dan kimia, telah muncul. Penghargaan Nobel Kimia tahun 1998 adalah diberikan kepada John A. Pople dan Walter Kohn untuk penghargaan karya inovatif mereka dalam pengembangan komputasi metode dalam kimia kuantum (Pople 2003; Kohn 2003). Penghargaan bergengsi seperti itu mendukung diterimanya komputasi sebagai alat yang valid dan ketat untuk menyelidiki fenomena kimiawi. Penerapan pendekatan statistik dan matematika yang mengandalkan komputasi, seperti Markov Chain Monte dengan Metode Carlo dan jaringan saraf tiruannya, telah terbukti membuka jalan baru untuk mengeksplorasi dan menghasilkan kemajuan dalam berbagai bidang seperti belajar asal muasal alam semesta dalam



astrofisika komputasi (Vogelsberger et al. 2014) untuk memahami kinetika pertumbuhan biji-bijian dalam ilmu material (Anderson et al. 1984; Srolovitz dkk. 1984). Wing (2006) menyatakan bahwa pendekatan pemikiran komputasi akan menjadi fundamental di semua disiplin ilmu dan kemajuan dalam komputasi akan memungkinkan para peneliti untuk melakukan dan membayangkan strategi pemecahan masalah baru dan menguji solusi yang baru di dunia maya dan dunia nyata.

Meskipun dalam bidang ilmu pasti seperti fisika dan teknik sudah berdiri sejak lama dan berdampingan secara interdisipliner dengan metodologi komputasi, pendekatan klasik untuk pemecahan masalah dalam biologi dan kimia secara historis menekankan deterministik sistem dengan kompleksitas rendah, sehingga mengabaikan sebagian besar masalah stokastik dan nonlinier. Bias kuat sebelumnya menuju studi sistem deterministik terutama satu kepraktisan, dengan istilah " nonlinier " menjadi hampir sinonim dengan " unsolvable ". Namun, alam memang begitu secara inheren nonlinier dan dapat ditandai dengan kekacauan perilaku, seperti yang dapat diamati dalam perubahan iklim (Dijkstra 2013; Manabe dan Stouffer 1988), penyebaran penyakit (Keeling dan Grenfell 1997; Olsen dan Schaffer 1990), tekanan ekologi (Lubchenco et al. 1991), dan evolusi (Lander dan Schork 1994; Turelli dan Barton 1994). Kedua matematis dan fisik, sistem deterministik/linier adalah pengecualian daripada aturannya. Dalam beberapa tahun terakhir, metode komputasi telah memperluas jangkauan fenomena nonlinier yang dapat dieksplorasi melalui penggunaan model matematika dan simulasi. Wolfram (2002) bahkan memproklamkan munculnya jenis ilmu baru berdasarkan eksperimen komputasinya menjadi pola yang muncul di alam, memperdebatkan eksplorasi

semacam itu tidak mungkin terjadi tanpa komputasi. Bidang ilmiah sedang mengalami kebangkitan dalam pendekatan eksperimental terutama karena ketersediaan komputer yang lebih kuat, aksesibilitas analitik metode baru, dan pengembangan model komputasi yang sangat rinci di mana beragam komponen dan mekanisme dapat digabungkan. Kemajuan ini dalam gilirannya, menciptakan kebutuhan yang berkembang untuk mendidik siswa dalam metode komputasi dan teknik untuk mendukung dengan cepat mengubah lanskap penelitian lintas matematika dan disiplin ilmu.

### **E. Berpikir Komputasi dan Penalaran Matematis**

Sejak sekolah dasar dan seterusnya, aritmatika anak-anak memiliki dua tahap yang sangat berbeda: memikirkan pertanyaan, dan kemudian mengerjakan jawabannya. Jumlah sesederhana  $23 + 39$  menuntut anak untuk dapat memecahkan kode simbol-simbol ini dengan cara yang berarti dan menentukan algoritma mana yang harus digunakan untuk menghitung jawabannya. Kemudian setelah itu, dapatkah anak tersebut melanjutkan bekerja keluar jawabannya. Ketika dihadapkan pada masalah kata, misalnya, berapa banyak uang kembalian yang akan saya terima dari selembar uang kertas lima puluh ribuan jika saya membeli tiga apel masing-masing seharga Rp. 10.000,-?. Kali ini menuntut tingkat abstraksi saat anak bergerak dari konteks tertentu untuk representasi matematisnya, dalam hal ini  $50.000 - 3 \times 10.000$ .

Secara lebih umum, kita mungkin melihat bahwa sebagian besar, bahkan mungkin semua, matematika mencerminkan dua tahap ini - berpikir tentang masalah dan kemudian memanipulasi simbol sesuai dengan aturan (yaitu versi yang lebih canggih untuk menyelesaikan masalah). Pandangan formalis matematika adalah bahwa matematika terdiri dari konsekuensi

aturan manipulasi string tertentu: misalnya geometri *Euclidean* dapat dianggap sebagai pernyataan yang dapat dibentuk dengan memanipulasi aksioma geometris sesuai dengan hukum inferensi. Namun, bahkan dalam paradigma formalis ini, matematika praktis dan berguna menuntut beberapa pemikiran tentang manipulasi *string* tertentu yang akan membawa kita menuju solusi dari masalah yang kita hadapi.

Pandangan matematika sebagai manipulasi simbol terletak pada dasar komputasi. Hingga tahun 1940-an, komputer diberikan kepada mereka yang dibayar untuk melakukan aritmatika di atas kertas atau kalkulator mekanis sesuai dengan aturan dan prosedur yang diberikan oleh manajer mereka. Turing mengungkapkan pandangan manipulasi simbol matematika ini dalam makalah seminarnya, 'Pada bilangan yang dapat dihitung, dengan mengaplikasikan ke *Entscheidungsproblem*', artinya mendefinisikan bilangan yang dapat dihitung sebagai yang dapat dituliskan oleh mesin dan menggeneralisasi ini ke fungsi yang dapat dihitung dan predikat yang dapat dihitung.

Dalam beberapa dekade sejak pekerjaan Turing dan Gereja, matematika, seperti banyak bidang lainnya, telah diubah oleh komputasi digital. Fase manipulasi simbol (atau pengerjaan) matematika yang dilakukan dalam sains, keuangan, ilmu sosial, seni dan setiap domain lainnya (selain pendidikan) dilakukan sekarang oleh komputer digital, bukan oleh orang, seperti yang dijelaskan oleh Conrad Wolfram dalam TED talk-nya. Memang banyak dari manipulasi simbol bahkan matematika murni sekarang sering dilakukan oleh komputer digital daripada ahli matematika, atau asisten penelitian mereka sendiri (lihat, misalnya, bukti Teorema Empat Warna yang dibantu komputer oleh Appel dan Haken.).

Fase pertama matematika - berpikir tentang masalah atau sistem - sebagian besar tetap tidak berubah. Pemecahan masalah yang kreatif dan imajinatif terletak di jantung matematika. Polya menyarankan empat prinsip pemecahan masalah: memahami masalah, menyusun rencana, melaksanakan rencana dan melihat ke belakang, ditambah sejumlah heuristik terkait. Wolfram berargumen dengan meyakinkan bahwa inilah yang seharusnya menjadi fokus pendidikan matematika, mengingat bahwa komputasi aktual sekarang dilakukan oleh mesin, dan menyarankan heliks empat tahapnya sendiri untuk pemecahan masalah: mendefinisikan pertanyaan, menerjemahkan ke matematika, menjawab dan menafsirkan komputer. Ada kesamaan di sini dengan proses pengembangan dalam rekayasa perangkat lunak: spesifikasi, desain, implementasi, dan pengujian.

Salah jika menganggap matematika sekolah terbatas pada penghitungan manual. Pemecahan masalah dan penalaran matematika merupakan bagian penting dari pendidikan matematika. Misalnya, Kurikulum Nasional Bahasa Inggris bertujuan untuk memastikan bahwa semua siswa bernalar secara matematis dengan mengikuti garis penyelidikan, menduga hubungan dan generalisasi, dan mengembangkan argumen, membenaran atau pembuktian menggunakan bahasa matematika; dan dapat memecahkan masalah dengan mengaplikasikan matematika pada berbagai masalah rutin dan non rutin dengan kecanggihan yang semakin meningkat, termasuk memecah masalah menjadi serangkaian langkah yang lebih sederhana dan tekun dalam mencari solusi.

Mengingat hubungan yang kuat antara matematika dan ilmu komputer, maka akan mengejutkan jika jenis penalaran matematika yang terlibat dalam memahami masalah dan merencanakan solusi mereka tidak dicerminkan oleh pemikiran serupa dalam menentukan sistem dan merancang solusi di

bidang komputasi. Seperti halnya matematika dapat dilihat sebagai pemikiran yang diikuti dengan manipulasi simbol, sehingga pemrograman dapat dilihat sebagai algoritma plus kode. Sebelum pemrograman mulai mengerjakan solusi pengkodean, mereka harus benar-benar memahami masalahnya dan memiliki rencana yang jelas (algoritma) tentang cara menyelesaikannya.

Istilah 'pemikiran komputasi' telah diciptakan untuk menggambarkan proses pemikiran yang terlibat dalam merumuskan masalah dan solusinya sehingga solusi tersebut direpresentasikan dalam bentuk yang dapat dilakukan secara efektif oleh agen pengolah informasi (Sayap, 2010). Meskipun belum ada konsensus universal mengenai bahan-bahan yang tepat dari pemikiran komputasi, kepentingannya dalam pendidikan komputasi diterima secara luas. Ini dilihat sebagai 'benang emas' yang berjalan pertama kali melalui Kurikulum Nasional Bahasa Inggris untuk Komputasi. Pendidikan komputasi berkualitas tinggi membekali siswa untuk menggunakan pemikiran komputasi dan kreativitas untuk memahami dan mengubah dunia. Komputasi memiliki hubungan yang dalam dengan matematika, sains dan desain dan teknologi, dan memberikan wawasan tentang sistem alami dan buatan.

Berdasarkan karya Brennan dan Resnick di mana pemikiran komputasi dieksplorasi sebagai konsep, praktik, dan perspektif, program pengembangan profesional berkelanjutan *Computing At School 'Barefoot Computing'* untuk guru sekolah dasar mengidentifikasi enam konsep dan lima pendekatan untuk pemikiran komputasi (*qv Computing At School's QuickStart Computing handbook*). Konsep tersebut memberikan pendekatan terpadu untuk pemecahan masalah baik dalam matematika maupun komputasi, dengan sejumlah contoh aktivitas yang dihasilkan untuk Barefoot Computing yang menghubungkannya dengan topik dalam kurikulum matematika Inggris.

# BAB 6

## COMPUTATIONAL THINKING

### DI BIDANG PENDIDIKAN

#### A. Pemikiran Komputasi Menjadi Keterampilan Dasar bagi Siswa

Karena pemikiran komputasi menjadi keterampilan mendasar untuk abad ke-21, maka guru sebaiknya berpegang pada prinsip-prinsip komputasi. Dengan memberikan informasi yang relevan tentang pemikiran komputasi, maka siswa akan lebih menyukai pelajaran komputer dan mereka akan lebih cenderung untuk mengintegrasikan prinsip-prinsip komputasi dalam kehidupan mereka di masa depan. Wing beranggapan bahwa CT merupakan keterampilan dasar dan pemikiran analitis untuk semua orang, tidak hanya untuk ilmuwan komputer. Dia menggambarkan CT adalah kegiatan "memecahkan masalah, merancang sistem, dan memahami perilaku manusia, dengan menggambar di atas konsep dasar ilmu komputer". Potensi pemikiran komputasi yang belum dimanfaatkan untuk pendidikan, menurut Wing, "Untuk membaca, menulis, dan berhitung, kita harus menambahkan pemikiran komputasi ke kemampuan analitik setiap anak." Sebuah Laporan tentang pemikiran komputasi oleh *National Research Council* (NRC) mengajukan gagasan serupa, bahwa CT adalah keterampilan kognitif yang "diharapkan untuk dimiliki oleh hampir setiap orang". Demikian pula, Bundy menyarankan bahwa konsep berpikir komputasi telah digunakan dalam disiplin ilmu yang lain sebagai proses pemecahan masalah, dan bahwa kemampuan untuk berpikir secara komputasi sangat penting untuk setiap disiplin ilmu. Laporan NRC juga

menyoroti “(1) bahwa siswa dapat belajar strategi berpikir seperti berpikir komputasi ketika mereka mempelajari suatu disiplin ilmu, (2) bahwa guru dan kurikulum dapat memodelkan strategi ini untuk siswa, dan (3) bahwa pedoman yang tepat dapat memungkinkan siswa untuk belajar menggunakan strategi ini secara mandiri”. Sebenarnya dengan mendidik guru tentang pemikiran komputasi akan memiliki dampak yang signifikan, karena seperti mempersiapkan pendidik masa depan untuk mempresentasikan mata pelajaran mereka menggunakan gagasan dari pemikiran komputasi, siswa akan memiliki eksposur yang lebih besar untuk komputasi secara umum.

Berpikir komputasi dalam pendidikan secara signifikan berpotensi untuk memajukan keterampilan memecahkan masalah siswa. Namun, literatur tentang penerapan pemikiran komputasi dalam pengaturan kurikulum masih relatif jarang. Langkah penting untuk berhasil mengintegrasikan pemikiran komputasi ke dalam kurikulum adalah untuk mempersiapkan guru masa depan untuk mengajarkannya. Hal tersebut didukung oleh Barr dan Stephenson (2011) yang menyatakan bahwa pengembangan profesional guru dan pendidikan CT adalah elemen penting untuk keberhasilan implementasi CT dalam pendidikan. Namun, Bower dan Falkner (2015) menemukan bahwa CT tidak selalu memiliki pemahaman yang kuat tentang apa yang melibatkan CT, dan mereka mungkin tidak memiliki ide yang jelas tentang bagaimana mengembangkan CT di ruang kelas mereka di masa depan. Juga, dalam studi Bower dan Falkner (2015), CT mengungkapkan kebutuhan mereka untuk mengembangkan "kemampuan pedagogis berpikir komputasi, memahami kurikulum, ide pelajaran, strategi untuk implementasi, penerapan ke contoh dunia nyata".

Wing (2006) menyarankan, "Untuk membaca, menulis, dan berhitung, kita harus menambahkan CT ke kemampuan analitis setiap anak". Barr dan Stephenson (2011) menyatakan bahwa karena siswa saat ini hidup - dan akan terus hidup - di dunia yang sangat dipengaruhi oleh komputasi, "Tidak lagi cukup untuk menunggu sampai siswa di perguruan tinggi untuk memperkenalkan konsep-konsep ini". Seperti yang ditulis Bower dan Falkner (2015), "Kita perlu memastikan bahwa sistem pendidikan kita tidak hanya menyediakan dasar-dasar literasi digital - keakraban dengan alat dan pendekatan untuk berinteraksi dengan teknologi - tetapi juga proses CT diperlukan untuk memahami praktik ilmiah yang mendukung teknologi".

Gadanidis (2017) mengemukakan bahwa CT mengemukakan lima unsur yang mendukung pendidikan matematika dasar, yaitu: agensi, akses, abstraksi, otomatisasi, dan audiensi. Selanjutnya, menurut Yadav, Mayfield, Zhou, Hambrusch dan Korb (2014), "Penting bahwa kita mengembangkan pemahaman guru tentang pemikiran komputasi dalam konteks materi pelajaran yang mereka ajarkan"; jika tidak, "guru hanya dapat memperoleh 'CT' pemahaman abstrak," dan "pengetahuan mereka akan tetap lambat dan mereka tidak akan dapat memasukkannya ke dalam pengajaran mereka". Berdasarkan pendapat-pendapat para peneliti CT tersebut dapat disimpulkan bahwa memang sebaiknya CT dimasukkan ke dalam kurikulum pelajaran sekolah sehingga dapat memberikan bekal kepada siswa tentang komputasi yang berguna bagi pekerjaan mereka di masa depan. Oleh karena itu, guru atau calon guru memang perlu untuk memahami dan mendalami komputasi.

Pendidikan komputasi semakin penting secara global untuk pemahaman yang lebih besar tentang pengembangan konseptual yang dapat menginformasikan pedagogi. Perhatian



yang semakin meningkat telah diberikan pada perolehan keterampilan abad ke-21 dan kompetensi digital oleh anak-anak. Sesuai dengan kebutuhan ini, pemikiran komputasi dan pengkodean, dalam beberapa tahun terakhir, telah menjadi bagian yang terintegrasi pada kurikulum sekolah di banyak negara. Misalnya Estonia, Israel, Finlandia, dan Inggris Raya hanyalah beberapa contoh dari upaya pemerintah yang terus berkembang untuk mengintegrasikan pengkodean sebagai literasi baru dan untuk mendukung siswa dalam tugas-tugas pemecahan masalah yang kreatif (Hubwieser, Armoni, Giannakos, & Mittermeir, 2014). Demikian pula, organisasi seperti "code.org", "codeacademy.com" menawarkan lingkungan belajar yang bermanfaat untuk mempromosikan aktivitas coding. Selain itu, ACM, Asosiasi Guru Ilmu Komputer, Inisiatif Matematika dan Sains Nasional, dan Kerangka Kerja Ilmu Komputer K-12 memberikan pedoman untuk menginformasikan dan membangun komunitas untuk pengajaran ilmu komputer. Sementara semakin banyak orang percaya bahwa keterampilan pengkodean sama pentingnya dengan matematika dan menulis (Horizon, 2015), ada kebutuhan mendesak akan bukti tentang desain kegiatan belajar yang efektif dan menarik untuk anak-anak (Guzdial, 2017, Lye dan Koh, 2014).

Sejak kemunculan pertama tentang pemrograman Logo Papert yang diperkenalkan pada 1960-an, banyak alat pemrograman lain telah muncul. Saat ini, ada banyak software ramah anak yang menawarkan pengalaman yang intuitif dan menyenangkan saat membuat kode. Contoh software tersebut adalah Scratch, Alice dan Kodu. Dengan berpartisipasi dalam kegiatan pengkodean, anak-anak dihadapkan pada pemikiran komputasi (Wing, 2006), yang melibatkan, namun tidak terbatas pada, pemikiran kritis, pemecahan masalah dan kreativitas.

Kegiatan ini terlihat baik dalam pengaturan formal dan informal dan dicirikan oleh desain, teknologi dan pendekatan yang berbeda (Papavlasopoulou, Giannakos, & Jaccheri, 2017). Penelitian-penelitian yang sudah ada menunjukkan bahwa pendekatan yang berbeda dapat menggabungkan fabrikasi fisik dan pengkodean (Kafai & Vasudevan, 2015), sementara yang lain, seperti Buechley, Eisenberg, Catchen, dan Crockett (2008), menggunakan LilyPad Arduino untuk membuat pengkodean menarik bagi anak perempuan. Dengan menggunakan lingkungan berbasis logo dan papan tulis interaktif mulai usia taman kanak-kanak, anak mengembangkan konsep matematika dan interaksi sosial, sekaligus menikmati kegiatan belajar (Fessakis, Gouli, & Mavroudi, 2013). Manfaat yang berbeda muncul dari kegiatan pengkodean landasan pada konstruksionisme (Papert, 1980); anak-anak diberi kesempatan untuk meningkatkan pemahaman mereka tentang konsep pemrograman, untuk mempromosikan kolaborasi dengan teman-teman, dan untuk mengubah sikap mereka terhadap pengkodean (Fessakis et al., 2013, Maloney et al., 2008, Papavlasopoulou et al., 2019).

Meskipun penelitian berkembang dan banyak kemungkinan yang ditawarkan oleh lingkungan belajar untuk merancang kegiatan pengkodean berbasis konstruksionisme untuk anak-anak (Kafai & Burke, 2015), ada relatif sedikit penelitian yang berfokus pada masalah gender dalam membuat dan kegiatan pengkodean untuk anak-anak (Papavlasopoulou, Giannakos, dkk., 2017). Terdapat perbedaan gender dalam sains, teknologi, teknik, dan matematika (STEM), dengan perempuan lebih sedikit terwakili di bidang ilmu komputer (Cheryan, Ziegler, Montoya, & Jiang, 2017). Dalam hal minat, kesenjangan gender dimulai sejak sekolah dasar (Ceci & Williams, 2010). Di antara berbagai faktor yang berdampak

pada wanita yang mengikuti jalur ilmu komputer adalah kurangnya pengalaman pendidikan yang positif di masa kecil mereka (Adya & Kaiser, 2005), ketakutan mereka untuk terlibat dalam kursus pengkodean yang sangat teknis, dan stereotip dan kesalahpahaman seputar karir dalam komputasi (Teague, 2002). Minat anak perempuan dalam ilmu komputer sejak usia muda mungkin memudar karena pendekatan pedagogis yang tidak sesuai gender atau tidak sesuai (Schulte & Knobelsdorf, 2007). Berkenaan dengan pengurangan kesenjangan gender dalam partisipasi dan untuk menarik lebih banyak anak perempuan ke komputasi, beberapa studi tentang lokakarya pengkodean telah berfokus pada perbedaan dalam kompetensi anak perempuan dibandingkan dengan anak laki-laki (Kalelioğlu, 2015), sementara yang lain telah mengeksplorasi peningkatan efikasi diri anak perempuan minat, sikap dan kepercayaan diri (Cheryan et al., 2017, Çakır et al., 2017). Untuk mendapatkan pengetahuan tentang bagaimana merancang kegiatan pengkodean, perlu menggunakan metode obyektif baru untuk menyelidiki keberadaan perbedaan gender dalam aspek-aspek seperti kinerja pembelajaran dalam kegiatan pengkodean dan untuk menemukan perbedaan utama antara praktik anak laki-laki dan perempuan (Papavlasopoulou et al., 2018, Papavlasopoulou, Sharma et al., 2017).

## **B. Konstruksionisme dan Kerangka Berpikir Komputasi**

Teori konstruksionisme Papert berpendapat bahwa pengalaman belajar lebih kuat ketika siswa secara aktif terlibat dalam proses pembelajaran dengan membuat proyek mereka sendiri (Papert, 1980). Dengan pengalaman mengembangkan sebuah proyek, anak-anak membangun pengetahuan mereka sebelumnya dan menemukan pengetahuan baru tanpa menerimanya secara pasif. Elemen inti dalam konstruksionisme

adalah "objek untuk dipikirkan". Hal inilah yang akan memberikan kesempatan kepada siswa untuk berinteraksi dan mendukung pemikirannya sendiri. Namun konstruksionisme lebih seperti sintesis karakteristik yang akan menghasilkan pembelajaran yang efektif. Dengan demikian, bersama dengan elemen inti untuk merangsang pemikiran individu, perlu adanya keterlibatan aktif siswa, elemen yang bermakna secara sosial dan interaksi sosial (Kafai, 2006).

Pemrograman permainan komputer merupakan strategi pendidikan pembelajaran konstruksionis. Selama proses pembuatan *game*, anak-anak berusaha mencapai suatu tujuan dan menguasai cara belajar dan berpikirnya sendiri. Penggunaan pemrograman menawarkan kemungkinan untuk membuat *artefak* yang bermakna secara sosial, berkomunikasi dengan orang lain dan memiliki pengalaman yang menyenangkan dan menarik (Robertson & Howells, 2008). Banyak penelitian telah menggunakan konstruksionisme sebagai dukungan untuk kegiatan pengkodean baik dalam pengaturan formal maupun informal untuk mempromosikan pengkodean, pemecahan masalah, pemikiran kritis dan keterampilan kolaboratif (Papavlasopoulou, Giannakos, et al., 2017).

Gagasan Papert terhadap komputasi yaitu "anak-anak yang menggunakan bahasa pemrograman Logo mengembangkan pemikiran algoritmik" (Papert, 1980). Namun, istilah "pemikiran komputasi" dipopulerkan oleh Wing (2006), yang berpendapat bahwa "pemikiran komputasi mewakili sikap dan keterampilan yang berlaku secara universal setiap orang, tidak hanya ilmuwan komputer, akan bersemangat untuk belajar dan menggunakan". Sejak itu, berbagai upaya untuk mendefinisikan pemikiran komputasi telah muncul, dengan tujuan mendukung pentingnya penelitian tentang membuat

pemikiran komputasi menjadi literasi abad ke-21 yang dapat diakses oleh semua individu (Guzdial, 2008). Sehubungan dengan Scratch, kerangka kerja Brennan dan Resnick (2012) menyarankan tiga dimensi utama untuk menggambarkan pemikiran komputasi: konsep komputasi (konsep yang digunakan pengguna saat mereka memprogram, seperti paralelisme dan variabel); praktik komputasi (praktik yang dikembangkan pengguna, seperti abstraksi dan debugging); dan perspektif komputasi (perspektif yang dikembangkan pengguna untuk komputasi, diri mereka sendiri, dan dunia di sekitar mereka). Kerangka pemikiran komputasi Brennan dan Resnick memungkinkan peneliti untuk memantau aktivitas pengkodean dan untuk memahami bagaimana anak-anak menggunakan konstruksi yang berbeda dan menangani konsep, bagaimana mereka fokus pada pembelajaran dan mengadopsi praktik berpikir yang berbeda, dan, akhirnya, bagaimana perspektif mereka berkembang dalam hubungannya, untuk diri mereka sendiri, orang lain, dan dunia teknologi.

Berpikir Komputasi sederhananya adalah proses berpikir yang terlibat dalam merumuskan masalah dan menghasilkan berbagai solusi dengan cara yang dapat dipahami oleh manusia atau komputer. Mengembangkan pengetahuan dan disposisi yang diperlukan untuk dipahami dan dibuat dengan pemikiran komputasi sekarang menjadi keharusan abad ke-21. Dengan triliunan microchip jaringan yang mengubah hidup kita, pendidikan menjadi sangat penting bagi masa depan kita. Para siswa saat ini membutuhkan lebih dari sekedar “kode” di kelas; mereka perlu menggunakan pemikiran komputasi secara berkelanjutan untuk menginspirasi keingintahuan, imajinasi, permainan, penemuan, dan kreasi. Untuk maju di abad ke-21, tidak cukup bagi kami untuk mengajari anak-anak membuat kode. Kami harus fokus pada

proses berpikir di balik pengkodean yang akan memiliki konsekuensi luas dan dampak luas pada semua aspek kehidupan dan masyarakat.

### **C. Spekulasi CT dalam Pendidikan di Masa Yang Akan Datang**

Landasan pemikiran komputasi adalah proses berulang yang didasarkan pada tiga tahap: Perumusan masalah (abstraksi); Ekspresi solusi (otomatisasi); Eksekusi dan evaluasi solusi (analisis). Berpikir komputasi digunakan untuk membuat program. Namun juga dapat digunakan untuk memecahkan berbagai masalah lintas disiplin ilmu. Sebelum kita menggunakan komputer untuk memecahkan masalah, kita harus memahami masalahnya sendiri dan mengidentifikasi cara untuk menyelesaikan masalah tersebut. Teknik berpikir komputasi membantu tugas-tugas ini. Jadi, pertanyaan besar - pertanyaan yang mendesak dan mendesak, bagi kita di industri pendidikan untuk menjawabnya adalah ini: jika A.i. diatur untuk melakukan semua hal luar biasa ini yang dulu kita pikir hanya bisa dilakukan oleh manusia, lalu apa tujuan pendidikan dan apa yang seharusnya menjadi fokus sekolah di masa depan? Dan bagaimana dengan robotnya? Apa itu Robotika? Sederhananya, robotika adalah cabang dari A.i. yang berkaitan dengan penggunaan praktis robot. Ini adalah cabang teknologi yang melihat perpaduan antara teknik, komputasi, matematika, dan sains. Penulis fiksi ilmiah, Isaac Asimov dikreditkan dengan menggunakan kata robotika untuk pertama kalinya dalam sebuah cerita pendek tahun 1940.

Bagaimana itu akan mengubah cara siswa belajar? Robotika sekarang merupakan teknologi informasi, dan seperti semua teknologi informasi, robotika tumbuh dengan kecepatan yang eksponensial. Robotika adalah bagian penting dari

pendidikan STEAM yang sudah menjadi fokus utama di banyak sekolah progresif. Fokus baru pada robotika dan STEAM akan membantu membina siswa inovatif yang kreatif. Saat dunia menjadi semakin digital, coding dan pemrograman (yang merupakan bagian integral dari robotika) akan semakin menjadi elemen inti dari apa yang dipelajari siswa. Robotika juga membantu memperkenalkan elemen bermain ke dalam pembelajaran. Gerakan Kreator adalah bagian penting dari mempromosikan inovasi di sekolah. Robotika memainkan peran penting dalam gerakan pembuat yang memungkinkan siswa untuk belajar melalui kolaborasi dan dengan berfokus pada pemecahan masalah praktis yang dihadapi masyarakat manusia. Lab Media MIT melihat pengkodean sebagai literasi baru. Siswa akan semakin diharapkan untuk membuat dan mendesain dengan menggunakan keterampilan pemrograman. Apa yang harus kita lihat di sekolah dalam waktu dekat?

- Peningkatan fokus pada pengkodean dan pemrograman
- Lebih banyak lab STEM / STEAM
- Penekanan pada ruang Maker
- Kurang fokus pada konten subjek dan lebih fokus pada pemecahan masalah dunia nyata
- Peningkatan penekanan pada pengkodean untuk printer 3D
- Robotika dalam waktu dekat akan menjadi sama pentingnya dan ada di mana-mana seperti halnya TIK dalam beberapa tahun terakhir.
- Guru akan fokus pada peningkatan kemampuan diri mereka sendiri (bahkan guru non-STEM).

Dalam mata pelajaran apa, robotika akan paling berguna? Terutama dalam mata pelajaran STEM (dari usia yang sangat muda - bahkan mungkin siswa tahap Foundation). Tetapi tergantung pada kreativitas dan imajinasi guru dan rencana pelajaran, itu bisa digunakan dalam Seni dan

Humaniora juga. Robotika dapat digunakan dengan dua cara: 1) sebagai mata pelajaran sendiri 2) Sebagai alat untuk mengajar mata pelajaran lain

Bagaimana robotika dapat membantu anak-anak dengan kebutuhan khusus? Memperkenalkan elemen permainan berguna saat menangani siswa berkebutuhan khusus. Sumber daya seperti kit Lego Mindstorm membuat pembelajaran menjadi menyenangkan, tetapi juga mengajarkan siswa untuk berpikir kritis, berkolaborasi, memecahkan masalah, dan kreativitas. Akankah itu berdampak pada cara anak-anak berinteraksi dan berpotensi menghambat komunikasi? Siswa masa depan tidak dapat diharapkan untuk berkomunikasi seperti yang kita lakukan saat ini. Sulit untuk mengatakan dengan pasti apakah robotika akan mempromosikan atau menghalangi komunikasi sosial. Tetapi dapat kita katakan dengan pasti, anak-anak masa depan akan berkomunikasi secara berbeda (seperti halnya semua generasi dibandingkan dengan pendahulunya: generasi setelah penemuan papirus berbeda dengan generasi yang membaca buku, yang pada gilirannya berbeda dengan generasi yang dibaca melalui CD ROM, Kindles, tablet, dan ponsel.) Bagaimana robotika akan berkembang dari waktu ke waktu - dalam 10 tahun ke depan atau lebih? Robotika dan A.i. akan semakin melengkapi satu sama lain. Jumlah pekerjaan yang terus meningkat akan diambil alih oleh A.i. atau diotomatiskan oleh robot. Ini tidak akan menjadi kasus kita v/s mereka. Manusia akan secara kognitif melakukan sinkronisasi dengan robot mereka untuk membentuk Humanity 2.0. Perusahaan baru Elon Musk, Neuralink, bertujuan untuk membuat antarmuka langsung antara manusia dan komputer. Hal ini berpotensi menimbulkan konsekuensi dan konsekuensi yang sangat besar bagi masa depan spesies kita. Ray Kurzweil, Pendiri Singularity University



dan Direktur Teknik di Google, memprediksikan bahwa dalam dekade mendatang kita akan memiliki daya komputasi yang triliunan kali lebih kuat daripada otak manusia. Kekuatan komputasi ini, dikombinasikan dengan A.i. dan Robotika akan mengubah masyarakat manusia dengan cara yang belum bisa kita bayangkan. (Sama seperti pencipta internet tidak dapat membayangkan dunia YouTube, Facebook, Twitter, dan Wi-Fi di mana-mana). Akankah robotika menjadi bagian penting dari pembelajaran? Pendidikan robotika akan terbukti penting dalam mempromosikan inovasi. Robotika, ditambah dengan Virtual Reality dan Augmented Reality akan memainkan peran yang terus meningkat di semua industri, terutama pendidikan, kedokteran, peperangan, dan eksplorasi ruang angkasa. Robotika terhubung dengan pemikiran kritis, kolaborasi, dan pemecahan masalah - keterampilan penting abad ke-21 yang akan berlaku untuk semua industri.

# BAB 7

## COMPUTATIONAL THINKING PADA PENDIDIKAN MATEMATIKA

### A. Pendekatan *Computational Thinking*

*Computational Thinking* (CT) dapat digambarkan sebagai pendekatan analitis yang berbagi metode dan prosedur berpikir matematis, serta cara lain yang biasanya diterapkan orang dalam pemecahan masalah, seperti pemikiran logis. Pendekatan ini dapat mempengaruhi atau dapat digunakan dalam beberapa bidang penelitian dan ditujukan untuk siapa saja, di mana saja, karena CT dianggap sebagai pendekatan intrinsik dan karakternya yang umum dapat diterapkan di luar ilmu komputer. Karakteristik utama kemampuan CT adalah:

- Merumuskan masalah sedemikian rupa sehingga memungkinkan orang untuk menggunakan komputer dan/atau alat lain untuk membantu memecahkan masalah.
- Mengorganisir dan menganalisis data secara logis;
- Merepresentasikan data melalui abstraksi seperti model dan simulasi;
- Mengotomatiskan solusi melalui pemikiran algoritmik (serangkaian langkah yang dipesan);
- Mengidentifikasi, menganalisis, dan menerapkan solusi yang mungkin dengan tujuan mencapai yang paling efisien dan kombinasi efektif dari langkah-langkah dan sumber daya;
- Menggeneralisasi dan mentransfer pemecahan masalah ini memproses berbagai macam masalah.

Konsep dasar CT berasal dari bagaimana seseorang memahami dan merencanakan solusi untuk masalah, pemetaannya dan menghubungkan beberapa konteks untuk menghasilkan solusi yang layak. Dalam literatur, ada dua pendekatan CT yang disorot. Pertama yaitu pengajaran dalam disiplin ilmu atau kursus dalam komputasi (misalnya pemrograman, robotika, dan *game development*) dengan atau tanpa menggunakan komputer; dan yang kedua yaitu pengenalan konsep ilmu komputer yang dikombinasikan dengan disiplin ilmu yang dimulai dari pendidikan dasar. Pendekatan pertama bertujuan untuk mengembangkan CT menggunakan mekanisme seperti pemrograman (algoritma dan prosedur) dan komputasi *unplugged*. Opsi kedua ini menyangkut praktik membuat orang memahami konsep ilmu komputer (misalnya protokol jaringan dan nomor *biner*) dengan memainkan elemen seperti permainan dan teka-teki tanpa menggunakan komputer, itu adalah solusi *unplugged*. Mengikuti pendekatan pertama membutuhkan pengenalan disiplin khusus untuk pembelajaran/pengajaran konsep komputasi dikurikulum sekolah, apakah ada yang mungkin tidak bisa dilakukan.

## **B. *Computational Thinking* dalam Pendidikan Matematika**

Guna mendukung perkembangan CT dalam pendidikan dasar tanpa membutuhkan disiplin ilmu komputasi khusus, Barr dan Stephenson menyarankan beberapa keterampilan yang dianggap sebagai inti dari CT. Keterampilan ini harus dirangsang dalam konteks lain pada disiplin ilmu pendidikan dasar, seperti matematika, sains dan membaca. Keterampilan-keterampilan tersebut diantaranya yaitu pengumpulan data, analisis data, representasi data, dekomposisi, abstraksi, algoritma, otomatisasi, paralelisasi dan simulasi. keterampilan

CT dapat dikembangkan pada sejumlah pendukung pendidikan dan pendidik berusaha untuk mengalihkan fokus dalam pendidikan matematika dari manipulasi simbolik prosedural ke keterampilan, seperti pemikiran komputasi, yang lebih mempersiapkan siswa untuk masa depan pekerjaan. Karena perubahan dalam pendidikan membutuhkan waktu, bagaimana guru matematika dapat memaparkan siswa pada keterampilan sekarang ini? Buku ini akan mencoba membagikan beberapa alat dan tip untuk membawa pemikiran komputasi ke dalam kelas matematika. Para siswa matematika tidak membutuhkan disiplin ilmu tertentu dalam komputasi tetapi melalui latihan menjawab pertanyaan yang sesuai dengan CT dan pemecahan masalah.

CT dalam pendidikan matematika diperlukan untuk membantu siswa dalam menyadari bahwa matematika melibatkan tidak hanya menemukan jawaban yang tepat untuk suatu masalah, tetapi juga pemahaman tentang masalah dan yang ada tidak ada solusi tunggal untuk itu. Dalam pengertian ini, metodologi pemecahan masalah bertujuan agar siswa dapat berkembang, dengan cara ini pemahaman matematika dan pentingnya bagi kehidupan. Pemecahan masalah tidak hanya menerapkan matematika konseptual semata, melainkan untuk mempelajari nilai sebenarnya dari matematika dalam kehidupan sehari-hari.

Dalam konteks ini CT muncul sebagai pendekatan terhadap pemecahan masalah. Tujuan utamanya adalah untuk merangsang keterampilan komputasi itu akan membantu siswa dari segala usia untuk merangsang keterampilan lain yang ada dibutuhkan untuk meningkatkan kemampuan mereka dalam memecahkan masalah. Wing menggambarkan CT sebagai ilmu yang dibangun di atas kekuatan batasan ilmu komputer. CT adalah seperangkat keterampilan pemecahan masalah yang

ditujukan untuk potensi mesin, tidak sepenuhnya bergantung padanya, tetapi menggunakannya sebagai alat kerja untuk mempraktikkan cara mereka memikirkan dan merumuskan solusi untuk masalah. CT bisa dilihat sebagai cara untuk memahami dan menghitung solusi, memungkinkan kemajuan signifikan dalam berbagai situasi masalah. Contoh penerapan ini adalah kemajuan dalam biologi, kimia, dan fisika. Wing juga mencirikan esensi CT sebagai abstraksi, elegan, dan aljabar yang mudah. Dalam abstraksi pada CT, setiap langkah untuk memecahkan masalah didukung oleh serangkaian solusi yang lebih kecil yang diselesaikan. Semua formalisasi ini diatur dalam lapisan abstraksi, dan komputasi mengotomatiskan lapisan abstraksi ini, kemudian memperoleh solusi. Oleh karena itu, CT itu tanpa alat dan komputasi instrumental, karena, seperti yang dikatakan sebelumnya, ini adalah bentuk organisasi pemikiran manusia. Memperhatikan penerapan CT dalam matematika, Barcelos dkk menyajikan pemetaan sistematis yang berfokus pada penerapan komputasi pada beberapa disiplin ilmu telah dievaluasi.

Menurut *International Society for Technology in Education* (ISTE), pemikiran komputasi adalah proses pemecahan masalah yang melibatkan identifikasi pola, pembuatan abstraksi, pengembangan algoritma, dan perumusan prosedur untuk komputer yang akan digunakan untuk menemukan solusi. Menulis kode untuk memecahkan masalah membutuhkan pemikiran komputasi. Siswa sering kali perlu melihat suatu masalah dengan cara yang berbeda saat membuat algoritma untuk menemukan solusinya. Saat memecahkan masalah kode, mereka menemukan nuansa matematika yang mungkin terlewat. Saat mereka bekerja, mereka mengembangkan kemampuan berpikir logis, menemukan pola, dan abstraksi yang lebih kuat. Banyak situs gratis yang dapat ditemukan

secara online untuk memandu Anda dan siswa Anda langkah demi langkah, terlepas dari tingkat kenyamanan Anda dengan *coding*. Beberapa situs tersebut diantaranya yaitu *CSandMath*, *Python*, *STEMCoding*, *Hour of Code* dan *Polyup*. Yang pertama yaitu *CSandMath*, situs web ini menyediakan pelajaran matematika untuk SD, SMP, dan SMA. Dalam setiap pelajaran, siswa menyelidiki dan menerapkan konsep matematika dengan menulis kode di *Scratch*, bahasa pemrograman blok yang mudah dipelajari untuk segala usia. Petunjuk, tantangan dan dukungan guru disertakan.

Yang kedua yaitu *Python*, adalah bahasa pemrograman yang gratis, populer, kuat, dan mudah dipelajari. Bagi mereka yang baru mengenal Python, buku *Math Adventures with Python: An Illustrated Guide to Exploring Math with Code* oleh Peter Farrell adalah bacaan yang baik untuk memulai. Buku tersebut menjelaskan cara mengunduh dan menginstal *Python* serta menyajikan aktivitas dan tantangan yang membangun keterampilan pemrograman sambil meminta siswa untuk mengeksplorasi konsep matematika. Misalkan dalam mata pelajaran Geometri, siswa dapat menjelajahi properti poligon dan transformasi menggunakan alat gambar *Turtle Python*.

Selanjutnya, *STEMCoding*; Proyek dari *The Ohio State University* ini menyediakan aktivitas dengan video yang memandu siswa dalam menulis kode untuk memecahkan masalah menggunakan matematika. Tidak diperlukan pengalaman pengkodean sebelumnya; panduan dan dukungan guru disediakan. *Hour of Code* - Inisiatif nasional oleh Pekan Pendidikan Ilmu Komputer dan Code.org memperkenalkan jutaan siswa pada satu jam ilmu komputer dan pemrograman komputer. Kegiatan pengkodean selama satu jam diurutkan berdasarkan tingkat kelas, dan kursus tambahan tersedia untuk siswa dan guru. Terakhir, *Polyup* adalah taman bermain

pemikiran komputasi. Siswa dari semua tingkat kelas bekerja dengan kecepatan mereka sendiri melalui aktivitas yang disebut "mesin," mempraktikkan pemikiran komputasi dan keterampilan berhitung saat mereka "memodifikasi mesin" untuk memecahkan masalah. Ini adalah permainan yang bagus untuk dimainkan sebagai aktivitas *Hour of Code*, untuk saat-saat istirahat di ruang kelas, atau untuk mempraktikkan konsep tertentu seperti urutan atau fungsi.

Selain situs-situs tersebut diatas ada juga perangkat lunak atau software yang mendukung pemikiran komputasi atau biasa disebut *software* komputasi teknis. Perangkat lunak komputasi teknis ini digunakan oleh akademisi dan industri untuk memodelkan proses, menganalisis dan memvisualisasikan data, dan melakukan komputasi. Perangkat lunak yang gratis diantaranya *Maxima*, *SageMath*, dan *GeoGebra CAS Calculator*. Sedangkan yang berlisensi diantaranya *Mathematica*, *Maple*, dan *MATLAB*. Untuk menggunakan perangkat lunak komputasi teknis, siswa merumuskan dan mengetik perintah dalam sintaks bahasa pemrograman tertentu. Untuk memplot fungsi dalam *Mathematica*, misalnya, siswa harus memahami struktur dan sintaks perintah `Plot []` dan berbagai opsinya. Jika perintah tidak bekerja dengan benar, siswa belajar untuk memecahkan masalah dan memperbaiki masalah.

### **C. Bagaimana Meluangkan Waktu untuk Berpikir Komputasi**

CT dapat diberikan dalam rangka "menyegarkan" pembelajaran di kelas. Perhatikan bahwa penting untuk memberi siswa waktu untuk merasa nyaman dengan perangkat lunak atau *platform* pengkodean dan untuk memecahkan masalah. Rencanakan dua atau tiga periode kelas per aktivitas, tergantung pada kerumitannya. Dengan alat berpikir

komputasi, dapat melatih kesabaran dan ketekunan siswa saat mereka mempelajari sintaksis dan memecahkan masalah matematika. Mereka merumuskan algoritma saat mereka mencari dan menemukan pola dalam konsep matematika. Mereka merasakan pencapaian saat mereka merasa lebih nyaman dengan sintaksis dan matematika. Dan mereka mengembangkan keterampilan hebat untuk ditambahkan ke *resume* mereka saat mereka mengejar magang atau bayangan pekerjaan sebagai persiapan untuk masa depan pekerjaan.

Pemikiran komputasi (CT) dapat mengambil berbagai bentuk, dapat berupa pemrograman komputer berbasis layar, dapat digunakan untuk mengontrol sirkuit digital dan robot, dan secara lebih umum dapat menjadi desain solusi algoritmik untuk masalah yang dapat dilakukan oleh komputer. Pendidikan telah menjadi *focal point* untuk pengajaran CT. Namun, tidak jelas bagaimana CT dapat atau harus sesuai dengan pendidikan. Sehingga muncul banyak pertanyaan yang sampai sekarang belum terjawab. Bagaimana pendidik matematika memberi ruang dalam kurikulum yang sudah penuh? Apakah mereka mengajar CT sebagai mata pelajaran tersendiri atau terintegrasi dengan mata pelajaran lain? Haruskah mereka mengikuti pendekatan Papert dengan mengintegrasikan CT dalam pendidikan matematika? Bagaimana mereka menangani CT dalam pendidikan guru?

Wing (2006) menyarankan, "Untuk membaca, menulis, dan berhitung, kita harus menambahkan CT untuk kemampuan analitis setiap anak". Barr dan Stephenson (2011) menyatakan bahwa karena siswa saat ini hidup - dan akan terus hidup - di dunia yang sangat dipengaruhi oleh komputasi, "Tidak lagi cukup menunggu sampai siswa di perguruan tinggi untuk memperkenalkan konsep-konsep ini. Seperti yang ditulis Bower dan Falkner (2015), "Kami perlu memastikan bahwa sistem



pendidikan kami tidak hanya menyediakan dasar-dasar literasi digital - keakraban dengan alat dan pendekatan untuk berinteraksi dengan teknologi - tetapi juga proses CT yang diperlukan untuk memahami praktik ilmiah yang mendukung teknologi”.

Gadanidis (2017) mengemukakan bahwa CT menawarkan lima kemampuan yang mendukung pendidikan matematika dasar yaitu agensi, akses, abstraksi, otomatisasi, dan audiens. Lebih lanjut, menurut Yadav, Mayfield, Zhou, Hambruch dan Korb (2014), “Penting bagi kami untuk mengembangkan pemahaman guru tentang pemikiran komputasi dalam konteks materi pelajaran yang mereka ajarkan”; jika tidak, "guru hanya dapat memperoleh pemahaman 'abstrak' tentang CT," dan "pengetahuan mereka akan tetap inert dan mereka tidak akan dapat memasukkannya ke dalam pengajaran mereka".

#### **D. Pembelajaran Matematika melalui Berpikir Komputasi**

Berpikir Komputasi merupakan terminologi yang mencakup rangkaian kompleks, proses penalaran yang diadakan untuk menyatakan dan menyelesaikan masalah melalui alat komputasi. Kemampuan mensistematisasi masalah dan menyelesaikannya dengan cara-cara ini sedang dilakukan dianggap keterampilan yang harus dikembangkan oleh semua siswa, bersama dengan Bahasa, Matematika dan Ilmu pengetahuan. Menimbang bahwa Ilmu Komputer memiliki banyak akarnya pada Matematika, itu masuk akal untuk merenungkan jika dan bagaimana pembelajaran Matematika dapat dipengaruhi dengan menawarkan kegiatan terkait dengan Berpikir Komputasi kepada siswa. Pada bukti-bukti yang dilaporkan dari pembelajaran Matematika dalam kegiatan yang ditujukan untuk mengembangkan keterampilan Berpikir

Komputasi. Empat puluh dua artikel yang disajikan didaktik kegiatan bersama dengan desain eksperimental untuk mengevaluasi hasil belajar yang diterbitkan dari 2006 hingga 2017 dianalisis. Mayoritas kegiatan yang diidentifikasi menggunakan alat perangkat lunak atau perangkat keras untuk pengembangannya.

#### **E. Mengapa CT dalam Pendidikan Matematika?**

Gagasan mengintegrasikan CT dalam pendidikan matematika bukanlah hal baru. Itu adalah bagian penting dari karya Papert (1980) dengan Logo, yang dikembangkan sebagai lingkungan belajar matematika. Papert (1980) mengungkapkan dalam bukunya 1980 *Mindstorms: Children, Computers, and Idea Powerfull* bahwa Logo "adalah untuk belajar matematika apa yang hidup di Perancis adalah untuk belajar bahasa Prancis." Lebih lanjut, mendukung gagasan bahwa matematika adalah bahasa untuk berkomunikasi dengan komputer, Papert (1980) menulis bahwa "belajar berkomunikasi dengan komputer dapat mengubah cara pembelajaran lainnya terjadi", dan juga "dimungkinkan untuk merancang komputer sehingga pembelajaran untuk berkomunikasi dengan mereka dapat menjadi proses alami". Namun, kurikulum CT di seluruh dunia tampaknya memandang CT sebagai tujuan kurikulumnya sendiri, daripada terintegrasi untuk mendukung dan meningkatkan pembelajaran bidang mata pelajaran yang ada, seperti halnya dengan Logo dan matematika. Namun, seperti dicatat oleh Gadanidis et al (2016), "ada hubungan alami antara CT dan matematika – tidak hanya dalam struktur logis atau dalam kemampuan untuk memodelkan hubungan matematika yang ditawarkan CT (Wing, 2006), tetapi juga dalam integrasi CT memberikan novel, pendekatan kreatif untuk pemecahan

masalah matematika, dan meningkatkan jangkauan matematika yang dapat digunakan siswa di semua tingkatan”.

Pemikiran komputasi (CT) dapat diaplikasikan ke dalam berbagai bentuk: Dapat berupa pemrograman komputer berbasis layar, dapat digunakan untuk mengontrol sirkuit digital dan robot, dan secara lebih umum dapat menjadi desain solusi algoritmik untuk masalah yang dapat dilakukan oleh sebuah komputer.

# BAB 8

## HASIL-HASIL PENELITIAN *COMPUTATIONAL THINKING* (CT)

Promraksa, S., Sangaroon, K., & Inprasitha, M. (2014). Characteristics of *Computational Thinking* about the Estimation of the Students in Mathematics Classroom Applying Lesson Study and Open Approach. *Journal of Education and Learning*, 3 (3), 56–66. <https://doi.org/10.5539/jel.v3n3p56>

Penelitian ini bertujuan untuk mempelajari dan menganalisis karakteristik dari berpikir komputasi tentang estimasi pada siswa kelas matematika dengan menggunakan lesson study dan pendekatan terbuka. Penelitian ini merupakan penelitian kualitatif, yang melihat tingkat berpikir dan proses berpikir tentang estimasi pada siswa kelas matematika yang diajar dengan lesson study dan pendekatan open yang akan ditunjukkan dengan 1) menganalisis proses berpikir tentang estimasi yang berfokus pada observasi, analisis videotape, didukung oleh wawancara dan analisis deskripsi. 2) menilai tingkatan berpikir tentang estimasi pada siswa kelas matematika yang diajar dengan lesson study dan pendekatan open. 3) menciptakan teori tentang estimasi pada siswa kelas matematika yang diajar dengan lesson study dan pendekatan open. Penelitian dibagi menjadi 2 tahap, pertama mempelajari konteks, kedua menciptakan teori tentang tingkatan berpikir dan menganalisis karakteristik dari berpikir komputasi dengan menggunakan 2 rencana pembelajaran yang dikembangkan

menurut lesson study dan open approach yaitu aktivitas nomor 1 “berapa banyak buku catatan yang akan kita dapatkan?” dan aktivitas 2. Berapa banyak kereta yang diperlukan?

Hasil dari penelitian ini adalah siswa dapat menunjukkan karakteristik dari berpikir komputasi tentang estimasi menurut 4 tahap dari pendekatan terbuka yang sekaligus mendapatkan kerangka teori dari tingkatan berpikir komputasi tentang estimasi pada siswa kelas matematika yang menerapkan lesson study dan pendekatan terbuka. Tingkatan berpikir siswa kelas 4 tahun ajaran 2011 di sekolah Choomchon Banchonnabot, menunjukkan bahwa: karakteristik dari berpikir komputasi tentang estimasi pada siswa kelas matematika yang menerapkan lesson study pendekatan terbuka masuk dalam semua level kerangka teori tingkatan berpikir tentang estimasi pada siswa kelas matematika yang menerapkan lesson study pendekatan terbuka, dan siswa menggunakan reformulasi dari estimasi terlebih dahulu sebelum komputasi dan membulatkan setelah komputasi.

Saran dari penelitian ini adalah sebaiknya guru memikirkan jawaban yang beragam terkait dengan permasalahan dan dihubungkan dengan kehidupan sehari-hari. Tingkatan berpikir tentang estimasi pada siswa kelas matematika yang menerapkan lesson study pendekatan terbuka dapat diterapkan untuk tipe tingkatan berpikir yang lainnya.

Son, J., & Lee, J. (2016). Pre- service Teachers ' Understanding of Fraction Multiplication, Representational Knowledge, and Computational Skills. *Mathematics Teacher Education and Development*, 18 (2), 5–28.

Penelitian ini membahas tentang bagaimana karakteristik profil kompetensi calon guru matematika pada bahasan perkalian pecahan dengan menguji pemahaman calon

guru matematika tentang perkalian pecahan dalam tiga konteks yang berbeda.

Cara meneliti adalah dengan penelitian kualitatif. Analisis data melibatkan empat proses: (1) pembacaan awal setiap respons calon guru matematika; (2) mengidentifikasi kebenaran tanggapan; (3) mengeksplorasi jenis model, penjelasan, dan strategi yang digunakan; dan (4) menafsirkan data secara kuantitatif dan kualitatif (Creswell, 1994). Dalam penelitian ini, penulis mengadakan suatu kursus selama satu semester (14 minggu) yang diikuti oleh calon guru matematika sejumlah 60 calon guru dengan kisaran usia 22 - 27 tahun (54 wanita dan 6 pria) tahun ajaran 2012/2013. Materi yang disampaikan dalam kursus tersebut yaitu pendalaman pemahaman tentang bilangan dan operasi pada sekolah dasar dan menengah. Instrumen yang digunakan dalam penelitian ini adalah soal-soal tugas yang terdiri dari tiga pertanyaan yang membutuhkan waktu sekitar 15 sampai 20 menit. Adapun tiga soal tersebut antara lain, 1) bagaimana Anda dapat menyelesaikan soal cerita berikut? Tiga perempat siswa dari kelas tujuh pergi ke pertandingan sepak bola. Dari siswa-siswa yang pergi ke permainan, sepertiga pergi dengan naik mobil. Berapa jumlah siswa kelas tujuh yang pergi ke permainan dengan naik mobil? 2) Bayangkan bahwa Anda sedang mengajarkan perkalian pecahan kepada siswa kelas lima. Bagaimana Anda menggunakan soal cerita di atas? Agar bermakna bagi siswa, Anda dapat menggunakan representasi grafis seperti gambar, menggambar, dll. Jelaskan masalah tersebut dengan menggunakan beberapa representasi. Konsep apa yang Anda pikirkan agar siswa mampu menyelesaikan masalah tersebut? Siswa sudah mengetahui bahwa  $\frac{3}{4}$  berarti "tiga bagian dari 4 bagian". Jelaskan dan sebutkan sebanyak mungkin. 3) Setiap orang memiliki pendekatan yang berbeda

untuk memecahkan masalah yang melibatkan perkalian pecahan. Bagaimana Anda memecahkan masalah di bawah ini? Selesaikan masalah. Dan tuliskan jawaban Anda dengan berbagai representasi seperti representasi verbal, gambar, pecahan, batang, dll.  $\frac{3}{4} \cdot \frac{2}{3} = ?$  Pertanyaan 1) dan 3) dirancang untuk menilai pengetahuan konten peserta (yaitu, pengetahuan konten khusus). Pertanyaan 2) dikembangkan untuk menilai strategi pedagogis mereka (yaitu, pengetahuan tentang konten dan pengajaran) dengan meminta mereka memberikan pendekatan pengajaran yang mungkin digunakan untuk mengajarkan masalah kata yang melibatkan perkalian pecahan. Masing-masing peserta memberikan tanggapan tertulis. Hasil jawaban dari para calon guru tersebut dianalisis

Hasil penelitian ini mengungkap bahwa ada campuran kompetensi yang berbeda, mulai calon guru yang tidak dapat menyelesaikan masalah tertentu dalam konteks apapun, ada juga yang mampu secara fleksibel menggambarkan pemahaman tentang perkalian pecahan dalam tiga konteks. Sebagian besar calon guru yang mengenali permasalahan dalam bentuk kata-kata sebagai perkalian pecahan mampu menjelaskan pemikiran mereka dengan menggunakan representasi grafis. Namun, berbagai jenis kesalahan yang dibuat oleh calon guru dalam merepresentasikan permasalahan kata ke dalam representasi grafis dan menerjemahkannya ke bentuk perkalian yang benar.

Saran/ rekomendasi adalah temuan ini menawarkan deskriptor tentang bagaimana calon guru memahami perkalian pecahan dalam konteks yang berbeda dan memberikan informasi untuk rancangan intervensi dalam pendidikan guru. Salah satu tujuannya adalah untuk mendukung pengenalan konektivitas perkalian pecahan dalam konteks yang berbeda.

Sriwongchai, A., Jantharajit, N., & Chookhampaeng, S. (2015). Developing the Mathematics Learning Management Model for Improving Creative Thinking in Thailand. *International Education Studies*, 8(11), 77–87. <https://doi.org/10.5539/ies.v8n11p77>

Yang diteliti adalah bagaimana mengembangkan model manajemen pembelajaran matematika untuk meningkatkan pemikiran kreatif siswa di Thailand. Tujuan penelitian ini adalah: 1) Mengetahui kondisi saat ini dan permasalahan siswa sekolah menengah yang relevan dalam mengembangkan model manajemen pembelajaran matematika untuk meningkatkan pemikiran kreatif, 2) Mengevaluasi keefektifan model tentang: a) efisiensi proses pembelajaran, b) perbandingan pretest dan posttest pada pemikiran kreatif dan prestasi siswa, dan c) perbandingan pemikiran dan prestasi kreatif antara kelompok eksperimen dan kelompok kontrol.

Cara meneliti adalah dengan membuat dan mengimplementasikan model pembelajaran pada siswa kelas 8 sekolah menengah di Thailand dan dibandingkan dengan kelas control yang diajar dengan pendekatan tradisional (konvensional). Penelitian ini merupakan penelitian pengembangan (R&D).

Hasil penelitian adalah keefektifan model berdasarkan skor prestasi adalah 76,25%, dan berdasarkan pemikiran kreatif sebesar 61,67%. Rata-rata posttest dalam prestasi belajar dan kemampuan berpikir kreatif kelompok eksperimen lebih tinggi daripada pretest, dan kelompok eksperimen menunjukkan pemikiran kreatif yang lebih tinggi daripada kelompok kontrol pada tingkat signifikansi 0,01.

Saran tidak disampaikan pada artikel.



Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, 3(2), 154–171. <https://doi.org/10.1007/s40751-017-0031-2>

Penelitian ini bertujuan untuk mengembangkan kerangka pedagogis dari berpikir komputasi yang dikembangkan dari teori konstruksi dan teori sosial-konstruktivisme. CTPF (*Computational Thinking Pedagogies Framework*) mencakup empat hal, yaitu unplugged (pelepasan), tinkering (megutak-atik), making (membuat), dan remixing (mengolah kembali).

Penelitian ini merupakan penelitian fundamental untuk menemukan teori baru. Keempat pengalaman pedagogi yang diajukan didasarkan pada teori pembelajaran konstruktivisme. Keempat pengalaman pedagogis tersebut secara parsial mencerminkan sebuah rangkaian perkembangan, atau zona pembelajaran proksimal, dimana setiap pengalaman pedagogis dapat mencerminkan tingkat permintaan kognitif yang semakin menantang daripada pengalaman pedagogis sebelumnya bagi siswa.

Hasil penelitian ini adalah dalam pelaksanaannya di pendidikan matematika, CTPF tidak harus berurutan. CTPF yang diusulkan dimaksudkan untuk menyediakan gambaran pendahuluan untuk pembelajaran bagi siswa dengan mempertimbangkan bagaimana cara mengajarkan berpikir komputasi. Di antara semua pengalaman yang diajukan, para guru idealnya harus membangun pelajaran dan pembelajaran sedemikian rupa sehingga mendukung berbagai jenis kegiatan, sehingga orang dengan berbagai minat dan gaya belajar dapat saling berinteraksi. CTPF yang diusulkan mungkin sangat

berguna bagi guru dan siswa yang memiliki pemahaman berpikir komputasi terbatas. CTPF yang diusulkan juga bisa menjadi struktur yang berguna untuk mendukung pengembangan guru. Pendekatan sekuensial terhadap CTPF, dimulai dengan pengalaman yang tidak dilepas dan bekerja menuju pengalaman remix, mungkin sangat berguna bagi para pemula. Namun, pendekatan sekuensial tidak penting; Sebaliknya, memastikan paparan keempat pengalaman di CTPF mungkin lebih optimal dalam hal mendukung pengembangan berpikir komputasi.

Saran / rekomendasi adalah mengingat sifat awal pernyataan tentang CTPF, ada banyak peluang untuk penelitian lebih lanjut. Selain studi terfokus yang berkaitan dengan matematika dan berpikir komputasi, penelitian lintas disiplin yang berkaitan dengan masing-masing dari keempat pengalaman tersebut diperlukan untuk memahami implikasi pengajaran dan pembelajaran di luar aplikasi tradisional dalam ilmu komputer. Penelitian diperlukan untuk mengeksplorasi CTPF secara penuh dan kegunaannya untuk pembelajaran berpikir komputasi. Penelitian semacam itu bisa menggali kegunaan kerangka kerja dalam kaitannya dengan pengembangan pengetahuan berpikir komputasi tingkat guru dan siswa dan pengembangan profesional berpikir komputasi guru.

Sanford, J. F., & Naidu, J. T. (2016). *Computational Thinking Concepts for Grade School. Contemporary Issues in Education Research, 9* (1), 23–32

Penelitian ini bertujuan untuk mengembangkan berpikir komputasi siswa dalam matematika melalui media spreadsheet.

Penelitian ini merupakan penelitian terapan yang menggunakan demonstrasi suatu media pembelajaran (*spreadsheet*). Konsep berpikir komputasi harus tersedia untuk memperkaya kehidupan sehari-hari di masyarakat modern. Pemikiran komputasional adalah sebuah augmentasi terhadap pemecahan masalah dan karenanya dapat dikenalkan sejak siswa belajar kata. Masalah kata matematik menggambarkan situasi dan meminta siswa untuk membentuk solusi matematis. Solusinya benar-benar muncul dari apa yang bisa disebut model matematis. Sebagai contoh, "Jika Fred memiliki 6 apel dan Jane memiliki 3 apel, berapa banyak apel yang dimiliki Fred dan Jane?" Anggapannya adalah penggunaan kalkulator tidak disarankan karena siswa harus bisa melakukan aritmatika mental dasar. Demikian pula, alat bantu elektronik dan solusi komputer harus diperkenalkan hanya setelah anak benar-benar menguasai aritmatika mental yang dibutuhkan untuk solusi terhadap masalah sederhana semacam itu. Dalam paper ini, dimunculkan beberapa permasalahan matematika, misalnya permasalahan aljabar.

Hasil penelitian ini adalah perlu diusulkan sebuah pengantar untuk "Berpikir Komputasi" pada tahap awal di sekolah dasar. Karena perlunya berpikir komputasi untuk angkatan kerja di dunia sekarang ini dan harus diajarkan bersamaan dengan bidang inti lainnya seperti membaca, menulis dan matematika di sekolah dasar. Beberapa contoh yang melibatkan aljabar dasar untuk memperdebatkan pentingnya pemikiran komputasi di mana penggunaan *spreadsheet* untuk memecahkan masalah. Dengan kata lain, seorang siswa dapat melihat skenario yang berbeda serta menanyakan pertanyaan "bagaimana jika". Ini akan membantu siswa untuk bekerja di tempat kerja besok.

Saran/ rekomendasi dari penelitian ini adalah berpikir komputasi sangat diperlukan siswa karena membantu siswa dalam menyelesaikan masalah di dunia kerja nantinya.

Djambong, T., & Freiman, V. (2016). Task-based assessment of students' *Computational Thinking* skills developed through visual programming or tangible coding environments. In *Proceedings of the 13th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2016* (pp. 41–51)

Penelitian ini bertujuan untuk memahami artikulasi antara tugas berpikir komputasi di satu sisi, kemampuan siswa mentarget, dan tipe-tipe permasalahan yang mereka selesaikan.

Penelitian ini merupakan studi kasus percontohan skala kecil dengan dua kelompok siswa untuk kelas dasar (kelas 6) dan kelas menengah (kelas 9). Sementara para siswa mengerjakan unit kurikuler selama 5 minggu dengan teknologi yang berbasis komputer dan pemrograman robotika, kami menilai kemampuan berpikir komputasi mereka dengan 23 tugas yang diberikan sebagai pra dan pasca tes. Metode pengumpulan data adalah pre-test dan posttest yang dianalisis dengan metode deskriptif kuantitatif.

Hasil penelitian adalah ditemukan perbedaan antara jenis keterampilan yang dinilai, kemudahan tugas, dan kelompok usia, yang membuat sulit untuk mencapai stabilitas kesimpulan. Perlu adanya penelitian lanjutan yang menggunakan sample yang lebih besar.

Saran/ rekomendasi dari penelitian ini adalah bahwa kebutuhan akan penilaian yang lebih lama dan lebih canggih sebagai perspektif penelitian selanjutnya untuk membangun bukti empiris yang lebih kuat tentang kemungkinan hubungan antara variabel terkait.

Angeli Charoula; Voogt, J. F. A. (2016). A K-6 *Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Educational Technology & Society, 19*(3), 47–57

Penelitian ini membahas tentang perspektif merancang kurikulum berpikir komputasi yang berfokus pada masalah dunia nyata, bagaimana kerangka pengetahuan konten pedagogis teknologi yang menjelaskan secara terperinci pengetahuan bahwa guru harus dapat mengajarkan berpikir komputasi. CSTA dan ISTE, bekerja sama dengan para pemimpin dari pendidikan tinggi, industri, dan pendidikan K-12, mengembangkan definisi operasional berpikir komputasi sebagai proses pemecahan masalah yang mencakup elemen berikut: (a) perumusan masalah dengan cara yang memungkinkan kita menggunakan komputer dan alat lainnya untuk membantu menyelesaikannya; (b) Mengorganisasikan dan menganalisis data secara logis; (c) Mewakili data melalui abstraksi, seperti, model dan simulasi; (d) Mengotomatisasi solusi melalui pemikiran algoritmik (yaitu serangkaian langkah yang diperintahkan); (e) Mengidentifikasi, menganalisis, dan menerapkan solusi yang mungkin dengan tujuan mencapai kombinasi langkah dan sumber daya yang paling efisien dan efektif; dan (f) Generalisasi dan transfer proses pemecahan masalah ini ke berbagai masalah.

Metode yang digunakan dalam penelitian ini desain research. Dalam menerapkan perubahan kurikulum menggunakan pendekatan desain holistik. Pendekatan disain holistik bertujuan untuk menghilangkan kompartementalisasi dan fragmentasi dengan memusatkan perhatian pada keseluruhan tugas pembelajaran yang kompleks dan otentik, tanpa menghilangkan unsur individu yang membentuk keseluruhan kompleks. Dengan demikian, dengan pendekatan

ini, jika diterapkan dengan benar oleh guru, anak belajar berpikir secara komputasi untuk memecahkan suatu masalah, dan juga mempelajari semua pengetahuan penyusun dan interkoneksi lainnya (teoritis dan atau praktis) yang berhubungan langsung dengan tugas berpikir komputasi. Kami mendukung pendekatan disain holistik untuk mengajarkan berpikir komputasi dan menekankan di sini dua langkah perancangan dalam prosesnya, yaitu, (a) perancangan masalah dalam menyelesaikan tugas dengan fokus pada masalah kehidupan nyata, dan (b) urutan tugas pemecahan masalah dari yang sederhana sampai yang kompleks.

Pengetahuan pedagogis untuk berpikir komputasi mencakup pengetahuan pedagogis umum yang berlaku untuk semua domain konten lainnya (yaitu, penggunaan pertanyaan untuk mempromosikan pemahaman, penggunaan contoh, penjelasan, demonstrasi), di samping pengetahuan tentang praktik pedagogi spesifik subjek yang berkaitan dengan berpikir komputasi. Pengetahuan pedagogis untuk berpikir komputasi didefinisikan dalam kaitannya dengan taktik pengajaran adalah sebagai berikut: (a) model yang bagaimana untuk memecahkan masalah atau memikirkan masalah dengan cara iteratif dan inkremental, (b) menyajikan atau menjelaskan solusi untuk masalah dalam rangkaian serangkaian langkah, (c) pengambilan keputusan model berdasarkan kondisi, (d) melakukan sesuatu berdasarkan (dan memperluas) apa yang orang lain atau yang telah Anda lakukan (menggunakan kembali dan remix), (e) menunjukkan bagaimana masalah yang kompleks dapat didekomposisi menjadi masalah yang lebih sederhana dan mengembangkan solusi secara bertahap, (f) tunjukkan bagaimana merancang model sebelum menulis program komputer untuk memecahkan masalah, dan (g) mencoba hal-hal

saat Anda pergi dan membuat revisi berdasarkan apa yang terjadi.

Kesimpulannya, dalam makalah ini penulis mempresentasikan kerangka kurikulum berpikir komputasi untuk merancang kurikulum K-6, area penelitian yang masih dalam tahap awal, mendeskripsikan pedoman desain untuk memberlakukan kerangka kurikulum, dan mendefinisikan TPCK (Technological Pedagogical Content Knowledge) dari berpikir komputasi sebagai badan pengetahuan bahwa guru harus mampu mengajar kurikulum di K-6. Selain itu, penulis memberikan contoh kursus persiapan guru yang dirancang khusus untuk mempromosikan TPCK dalam berpikir komputasi guru.

Saran/rekomendasi dari tulisan ini adalah perlu adanya penelitian lebih jauh tentang keefektifan kerangka kerja yang diusulkan ini dalam berbagai konteks. Diharapkan dengan mengadopsi ilmu computer secara bertahap pada mata pelajaran di seluruh dunia, pengetahuan dan basis penelitian ini mengenai isu-isu yang dibahas dalam tulisan ini akan berkembang secara pesat dalam beberapa tahun mendatang.

Hu, C. (2011). *Computational Thinking – What It Might Mean and What We Might Do about It*. In *ITiCSE '11: Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 223–227). <https://doi.org/10.1145/1999747.1999811>

Penelitian ini bertujuan untuk mendeskripsikan definisi dari berpikir komputasi dan hubungannya dari berpikir untuk mengetahui paradigma berpikir.

Dalam penelitian ini, peneliti menganalisis tentang perbedaan persepsi dari berpikir komputasi, perbedaan cara

berpikir dalam hubungan untuk menghitung, berpikir matematis dalam kaitannya menghitung, apakah berpikir komputasi merupakan sebuah campuran, mungkin?

Hasil dari analisis adalah gagasan berpikir komputasional muncul pada waktunya untuk meningkatkan tingkat urgensi dalam mempromosikan pendidikan komputasi di seluruh spektrum pendidikan K-16. Melakukan mempengaruhi cara kita berpikir. Paradigma berpikir tidak banyak berarti bagi siapa saja yang tidak memiliki banyak pengalaman dalam melakukan berbagai hal yang dapat membantu paradigma tersebut. Kami mempromosikan pendidikan STEM dengan meminta siswa memecahkan masalah STEM, bukan dengan menganjurkan pemikiran ilmiah, teknik, atau matematis. Ada inheren C (Computing) di STEM. Belajar STEM tanpa belajar komputasi pada dasarnya tidak memadai. Belajar komputasi sambil memecahkan masalah STEM, di sisi lain, pasti akan menumbuhkan kemampuan berpikir komputasi seseorang tidak peduli bagaimana pengertiannya didefinisikan.

Faber, H. H., Wierdsma, M. D. M., Doornbos, R. P., Van Der Ven, J. S., & De Vette, K. (2017). Teaching *Computational Thinking* to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network*, 12, 13–24.

Artikel ini berfokus pada sebuah kursus pengantar dalam berpikir komputasi pada siswa pada tahun terakhir di sekolah dasar. Kursus ini terdiri dari enam pelajaran 90 menit yang diajarkan seminggu sekali selama enam minggu di 26 sekolah di utara Belanda. Pelajaran ini dirancang bagi siswa untuk mempelajari konsep pemrograman tanpa memerlukan komputer atau tablet. Makalah ini menjelaskan tentang proses



perancangan dan evaluasi untuk pelajaran 'unplugged' ini dalam berpikir komputasi.

Metode yang digunakan dalam penelitian ini adalah metode penelitian desain pendidikan yang berfokus pada penciptaan dan peningkatan desain pendidikan untuk mendapatkan lebih banyak wawasan tentang proses pembelajaran untuk topik tertentu dan merumuskan prinsip-prinsip untuk desain pendidikan masa depan mengenai topik tersebut. Dengan mengumpulkan umpan balik dari guru setelah menguji setiap revisi desain, keefektifan desain dapat ditingkatkan. Dalam penelitian ini, kami mengembangkan sebuah desain untuk serangkaian materi pelajaran yang ditujukan untuk pengajaran pemikiran komputasi. Setelah menggunakan materi pelajaran untuk mengajarkan pemikiran komputasi, umpan balik yang dikumpulkan dari guru digunakan untuk memperbaiki materi pelajaran. Dalam penelitian eksploratif pendahuluan ini, prototipe desain dikembangkan dan dievaluasi. Dalam sebuah studi lanjutan, revisi materi pelajaran akan dievaluasi dan diperbaiki. Analisis data evaluasi dapat mengungkapkan prinsip desain. Prinsip desain ini dapat digunakan oleh orang lain untuk membuat materi pelajaran baru untuk mengajarkan pemikiran komputasi. Prinsip desain menyatakan elemen materi pelajaran mana yang menghasilkan reaksi positif baik pada siswa dan guru. Dengan memperbaiki setiap revisi desain, dengan cara siklis di mana setiap desain diuji dan dievaluasi, lebih banyak prinsip desain dapat diperoleh. Prinsip desain adalah hasil utama dari penelitian ini. Namun, materi pelajaran yang dikembangkan selama proses merupakan hasil sekunder dari penelitian ini. Materi pelajaran dan prinsip desain digunakan untuk menjawab pertanyaan penelitian. Sementara prinsip desain dapat memiliki efek yang lebih luas, misalnya saat merancang

materi pelajaran baru, materi pelajaran aktual yang dikembangkan selama studi ini dapat digunakan secara langsung oleh seorang guru sekolah dasar untuk mengajarkan pemikiran komputasi. Umpan balik yang diperlukan untuk memperbaiki desain terdiri dari wawancara kelompok terarah dengan para guru. Pertanyaan berkaitan dengan aspek praktis dari materi pelajaran, seperti waktu yang dibutuhkan untuk menyelesaikan pelajaran dan bagaimana reaksi para siswa. Selanjutnya, kami bertanya apa yang disukai dan disukai oleh guru tentang materi pelajaran, dan jika mereka dapat memberikan saran tentang bagaimana memperbaiki materi. Pelajaran yang diciptakan diberikan kepada 26 sekolah di wilayah utara provinsi Groningen di Belanda. Secara total, materi pelajaran digunakan untuk mengajar 411 siswa, dan kami merekrut 15 guru untuk memberikan pelajaran dan memberi umpan balik selama wawancara kelompok terarah.

Saran/ rekomendasi dari penelitian ini adalah artikel ini diakhiri dengan prinsip-prinsip desain untuk perancangan pelajaran dalam berpikir komputasi, dan membahas kemungkinan arahan untuk penelitian selanjutnya.

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking. *Journal of Educational Computing Research*, 53(4), 562–590. <https://doi.org/10.1177/0735633115608444>

Penelitian ini bertujuan untuk merancang kerangka kerja *Three-Dimensional Integrated Assessment* (TDIA). TDIA memiliki dua tujuan: satu adalah untuk mengintegrasikan tiga dimensi (*directionality, openness, dan process*) ke dalam rancangan tugas penilaian yang efektif; dan yang lainnya menilai secara

komprehensif tiga dimensi CT termasuk konsep, praktik, dan perspektif komputasi.

Pembuatan kerangka kerja penilaian dari berpikir komputasi ini berfokus pada tiga dimensi dalam desain tugas penilaian: directionality (maju atau mundur), keterbukaan (tiga jenis keterbukaan), dan proses (*self-report*). Berdasarkan dimensi ini, kami merancang enam tugas sebagai berikut:

1. Tugas tertutup, yang merupakan tugas yang tidak diselesaikan dengan hasil yang didefinisikan dan proses yang didefinisikan semata-mata;
2. Tugas maju semiopen, yang merupakan tugas yang tidak selesai dengan hasil yang didefinisikan semata-mata dan proses yang tidak selesai atau terbuka;
3. Tugas tertutup yang tertutup, yaitu tugas pemecahan masalah dengan hasil yang didefinisikan dan proses yang didefinisikan semata-mata;
4. Tugas pembalikan semiopen, yang merupakan tugas pemecahan masalah dengan hasil yang didefinisikan semata-mata dan proses yang tidak selesai atau terbuka;
5. Tugas terbuka dengan laporan desain kreatif, yang merupakan tugas kreatif dengan hasil terbuka dan proses terbuka; dan
6. Tugas terbuka tanpa laporan desain kreatif, yang merupakan tugas kreatif dengan hasil terbuka dan proses terbuka.

Untuk lebih memastikan penerapan masing-masing tugas dan kelebihan dan kekurangannya, dilakukan percobaan uji coba pada akhir semester musim gugur di tahun 2014 di sekolah dasar selama 3 minggu.

Penelitian ini mengembangkan sebuah kurikulum untuk menguji kerangka kerja penilaian berpikir komputasi, yaitu kurikulum berbasis sekolah "belajar bercerita dengan

pemrograman" berdasarkan kerangka kerja 3D CT. Kurikulum ini menggunakan bahasa pemrograman 3D Alice2.4 yang dikembangkan di Carnegie Mellon. Subjek dalam penelitian ini adalah 144 siswa kelas enam yang berasal dari sekolah dasar di Changshu City of China (disebut "Sekolah C" dalam artikel ini), yang dipilih secara acak 4 dari 8 kelas. Para siswa ini telah mengikuti kursus LOGO dan kursus awal masing-masing selama satu semester di kelas 5 dan mengikuti kurikulum Alice selama satu semester di kelas 6 yang diajarkan oleh guru yang sama. Materi uji coba mencakup enam tugas dalam proyek pemrograman semi-final, laporan refleksi, laporan desain kreatif, rubrik, dan skema pengkodean.

Hasil dari penerapan tugas tersebut antara lain menunjukkan bahwa (a) tugas sebaliknya tidak lebih unggul daripada tugas ke depan; (b) tugas semiopen dan tugas terbuka lebih efektif daripada tugas tertutup, dan tugas semiopen memiliki tingkat kesulitan dan diskriminasi yang lebih tinggi daripada yang lain; (c) laporan sendiri menyediakan fungsi yang bermanfaat untuk diagnosis dan bimbingan pembelajaran; (d) skor tidak memiliki perbedaan yang signifikan antara anak sekolah dan siswi di enam tugas; dan (e) kesulitan enam tugas diskriminasi yang lebih tinggi daripada yang lain. Untuk menerapkannya secara efektif, saran guru berikut untuk merancang tugas komputasi diusulkan: memotivasi minat dan antusiasme siswa, menggabungkan artefak semifinal, yang melibatkan diagnosis dan panduan pembelajaran, dan termasuk beberapa jenis tugas dalam penilaian.

Dalam penelitian ini, laporan refleksi dan laporan desain kreatif digunakan sebagai pendekatan untuk evaluasi proses saja. Mereka memiliki potensi untuk diubah menjadi nilai dalam evaluasi sumatif di masa depan. Tugas terbuka dapat digunakan secara teoritis untuk menilai semua perspektif

komputasi, namun sebenarnya hanya aspek kreativitas dan ekspresi yang dinilai dalam penelitian ini karena siswa diminta menyelesaikan tugas terbuka secara independen. Jika mereka diminta bekerja dalam kelompok, perspektif komputasi mereka seperti berkomunikasi dan berkolaborasi, memahami, dan mempertanyakan dapat dinilai lebih lanjut. Kita akan lebih jauh mengeksplorasi kemungkinan ini dalam penelitian selanjutnya.

Saran/rekomendasi dari penelitian ini adalah perlu adanya penelitian lebih lanjut yang lebih luas, misalnya perlu dilihat bagaimana berpikir komputasi dari siswa yang menyelesaikan tugas dengan berkelompok dan juga bagaimana untuk kurikulum yang lain.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining *Computational Thinking* for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25 (1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>

Penelitian ini merupakan penelitian kajian literatur untuk mengajukan definisi tentang berpikir komputasi untuk matematika dan sains dalam bentuk taksonomi yang terdiri dari empat kategori utama, yaitu: praktik data, pemodelan dan simulasi, praktik pemecahan masalah komputasi, dan praktik berpikir sistem.

Metode yang digunakan dalam penelitian ini adalah pemanfaatan literatur berpikir komputasi yang ada, wawancara dengan matematikawan dan ilmuwan, dan contoh dari berpikir komputasi yang baik. Untuk mengembangkan taksonomi praktik berpikir komputasi untuk matematika dan sains, digunakan beberapa sumber untuk mengidentifikasi praktik karakteristik yang paling penting untuk memenuhi kebutuhan

siswa dan untuk mencerminkan karya profesional di berbagai bidang matematika dan sains. Dalam proses ini, peneliti bekerja sama dengan para peneliti STEM lainnya, para guru dan pengembang kurikulum. Tiga sumber untuk menciptakan dan memvalidasi taksonomi ini adalah: (1) kegiatan pendidikan teladan yang melibatkan berpikir komputasi dalam matematika dan sains, (2) persediaan konsep dan dokumen standar yang ada, dan (3) wawancara dengan ahli matematika dan ilmuwan.

Hasil dari penelitian ini adalah manfaat utama dengan menanamkan berpikir komputasi kepada siswa dalam konteks ini adalah: (1) membangun hubungan timbal balik untuk belajar antara pemikiran komputasi dan matematika dan domain sains, (2) membahas masalah praktis untuk menjangkau semua siswa, dan memiliki guru yang unggul, dan (3) ia membawa pendidikan sains dan matematika lebih sesuai dengan praktik profesional saat ini di bidang ini.

Aspek dari keterampilan berpikir komputasi:

1. Kemampuan untuk mengatasi masalah terbuka
2. Ketekunan dalam bekerja melalui tantangan
3. Ketahanan dalam menghadapi kompleksitas
4. Mewakili ide dengan cara yang berarti secara komputasi
5. Memecah masalah besar menjadi masalah yang lebih kecil
6. Menciptakan abstraksi untuk aspek masalah yang ada
7. Membingkai ulang masalah menjadi masalah yang dapat dikenali
8. Menilai kekuatan / kelemahan dari representasi data / sistem representasi
9. Membangkitkan solusi algoritmik
10. Mengakui dan mengatasi ambiguitas dalam algoritma

Saran/rekomendasi dari penelitian ini adalah butuh dukungan dari berbagai pihak dalam memasukkan berpikir komputasi ke dalam kurikulum.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing *Computational Thinking* to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. *Technology, Knowledge and Learning*, 22(3), 1–21. <https://doi.org/10.1007/s10758-017-9328-x>

Penelitian ini bertujuan untuk melihat berpikir komputasi di dalam matematika dan pemrograman yang ada di kelas dasar. Penelitian ini juga meneliti dampak dari pengambilan perspektif komputasi melalui berbagai tingkat aktivitas yang terkandung di dalamnya, yaitu tinggi dan rendah terhadap prestasi belajar matematika dan pemrograman siswa.

Penelitian ini merupakan penelitian kuantitatif yang menggunakan pendekatan intervensi yang dirancang sebagai percobaan faktorial  $2 \times 2$  dengan dua variabel independen, menghasilkan empat kelompok eksperimen. Menguji efek dari empat kondisi dari kombinasi kedua variabel independen tersebut membantu menemukan bukti tentang bagaimana berbagai jenis kondisi pembelajaran menghasilkan tingkat pemahaman yang berbeda dalam matematika dan pemrograman. Subjek dalam penelitian ini adalah 66 siswa TK dan kelas satu (36 laki-laki, 30 perempuan) terdaftar dalam program pengkabelan afterschool di dua sekolah negeri tetangga New York City. Peserta dalam kelompok Praktik Perspektif Komputasi (CPP) melakukan peran pemrogram, yang melibatkan (a) memerintahkan pengganti dan (b) memberikan langkah-langkah prosedural untuk memecahkan masalah yang dikombinasikan dengan gerakan tubuh dan tangan. Peserta tanpa CPP menggunakan gerakan tubuh atau tangan untuk memecahkan masalah tanpa berkomunikasi dengan rekan pengganti.

Penelitian ini dilakukan selama lima sesi, 1 jam per sesi, di dua sekolah dasar negeri New York City sebagai bagian dari kurikulum di luar sekolah mereka. Adapun kelima sesi tersebut antara lain:

- a. Sesi 1 (Pre-tes), dilakukan pada minggu pertama untuk mengukur kemampuan siswa sebelum diperkenalkan program.
- b. Sesi 2 (Intervensi Perwujudan 1), dilakukan pada minggu ke-2. Pembagian kelas ke dalam empat kondisi.
- c. Sesi 3 (Intervensi Perwujudan 2), tahapan setelah intervensi pertama
- d. Sesi 4 (Scratch Jr. Programming), Setelah menyelesaikan 2 minggu kegiatan yang diwujudkan dengan menggunakan keterampilan estimasi, penambahan, dan pengurangan jumlah baris, siswa diminta untuk melakukan tugas transfer dengan menggunakan pemrograman
- e. Sesi 5 (Uji Tertunda), setelah kembali ke program setelah 1 minggu liburan, siswa menyelesaikan tes tertunda pada estimasi jumlah baris dan jumlah rasa.

Hasil dari penelitian menunjukkan bahwa kegiatan siswa yang melibatkan berpikir komputasi dalam memecahkan masalah matematika meningkatkan pemahaman dan keterampilan pemrograman matematika seperti yang ditunjukkan dalam Stracth Jr. Selain itu, berpikir komputasi secara signifikan dapat meningkatkan pemahaman siswa tentang konsep pemrograman.

Saran/rekomendasi dari penelitian ini adalah bagaimana membuat proses berpikir komputasi lebih konkret dan relevan dalam konteks kurikulum standar, khususnya matematika.



Gadanidis, G., & Gadanidis, G. (2017). Artificial intelligence, *Computational Thinking*, and mathematics education. <https://doi.org/10.1108/IJILT-09-2016-0048>

Tujuan dari penelitian ini adalah untuk menginvestigasi kesamaan dari kecerdasan buatan (AI), berpikir komputasi (CT), dan pendidikan Matematika (ME) untuk siswa kelas 8. Fokus dari ketiga komponen diatas adalah agensi, pemodelan fenomena dan konsep abstrak di luar contoh spesifik.

Metode yang digunakan dalam penelitian ini merupakan kerangka teoritis yang mengadopsi perspektif budaya asocio dimana pengetahuan dibangun dalam interaksi dengan orang lain. Juga mengacu pada keragaman teknologi yang ada di sekitar.

Hasil penelitian ini adalah ada kesamaan antara AI, CT dan ME yang penting untuk dipertimbangkan dalam pendidikan matematika. Orisinalitas /nilai dari penelitian adalah memandang pendidikan matematika dari sudut pandang disiplin ilmu lain dan memahami pentingnya agensi, pemodelan dan abstraksi di dalam pendidikan matematika serta menyediakan konteks dan alat baru untuk digunakannya di dalam kelas.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining *Computational Thinking* for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>

Penelitian ini bertujuan untuk mengembangkan pemahaman berpikir komputasi khususnya dalam penerapan pada bidang matematika dan sains. Dalam artikel ini disajikan sebuah usulan tentang definisi dari berpikir komputasi untuk

matematika dan sains dalam bentuk taksonomi yang terdiri dari empat kategori, yaitu (1) data, (2) pemodelan dan simulasi, (3) menyelesaikan permasalahan komputasi, dan (4) sistem berpikir.

Membawa berpikir komputasi dan praktiknya ke dalam matematika dan sains akan memberikan pebelajar pandangan realistik yang lebih yang sesuai bidangnya, memperbaiki persiapan siswa dalam mengejar karir dalam bidangnya. Dilihat dari segi pedagogis, pemikiran yang menggunakan komputasi dan keterampilan dapat memperdalam konten matematika dan sains dalam pembelajaran.

Penelitian ini merupakan penelitian pengembangan yang terdiri dari 5 tahap. Tahap pertama yaitu kajian pustaka. Pada tahap ini peneliti membuat sekumpulan konsep awal dari berpikir kritis, yaitu kemampuan untuk menyelesaikan permasalahan terbuka, ketekunan dalam mengerjakan soal, kepercayaan diri dalam menghadapi masalah yang kompleks, merepresentasikan kembali ide-ide dalam cara komputasi yang berarti, memecah masalah yang kompleks menjadi lebih sederhana, membuat abstraksi dari permasalahan, meringkai masalah kompleks menjadi masalah yang lebih dikenal, menilai kekuatan/kelemahan dari representasi data/system, menggeneralisasikan solusi algoritmik, dan memahami dan membahas ambiguitas dalam algoritma. Tahap kedua yaitu membuat taksonomi dengan cara mengumpulkan dan mengklasifikasikan jenis-jenis kegiatan yang didesain (berdasarkan tahap pertama) sebagai pengenalan berpikir komputasi ke matematika dan sains pada pendidikan tinggi. Tahap ketiga yaitu, merevisi dan mengelompokkan taksonomi. Pada tahap ini peneliti merevisi dan mengelompokkan kegiatan-kegiatan yang dihasilkan pada tahap kedua. Tahap keempat yaitu review eksternal. Pada tahap ini dilakukan

review dari luar yaitu dari para guru dan peneliti matematika. Tahap kelima yaitu tahap tambahan. Pada tahap ini proses dari penggolongan dari tahap-tahap sebelumnya didiskusikan dengan beberapa ahli (15 ahli STEM) di bidang matematika dan sains, yang dalam penelitian ini adalah ahli biokimia, fisikawan, ahli mesin, astrofisika, ahli computer, dan ahli biomedis.

Hasil dari penelitian ini adalah taksonomi dari berpikir komputasi terdiri dari 4 kategori, yaitu

- (1) data dan informasi, kegiatan yang termasuk dalam kategori ini adalah mengumpulkan data, membuat data, memanipulasi data, menganalisis data, memvisualisasi data,
- (2) pemodelan dan simulasi, kegiatan yang terlibat dalam kategori ini adalah menggunakan model komputasi untuk memahami konsep, menggunakan model komputasi untuk menemukan dan menguji solusi, menilai model komputasi, mendesain model komputasi, mengkonstruksi model komputasi,
- (3) Menyelesaikan permasalahan komputasi, kegiatan yang termasuk dalam kategori ini adalah menyiapkan permasalahan untuk solusi komputasi, memprogram komputer, memilih alat komputasi yang efektif, menilai perbedaan pendekatan/solusi dari suatu masalah, membuat abstraksi komputasi, penyelesaian masalah dan pembersihan eror.
- (4) berpikir system, kegiatan yang terlibat adalah menginvestigasi system kompleks sebagai keseluruhan, memahami hubungan dari system, berpikir dalam tingkatan, mengkomunikasikan informasi tentang system, mendefinisikan system dan mengatur kompleksitas.

Tiga keuntungan utama dari melibatkan berpikir komputasi ke dalam matematika dan sains adalah pertama,

berpikir komputasi dapat membangun hubungan timbal-balik antara berpikir komputasi dan matematika dan sains dalam pembelajaran. Kedua, berpikir komputasi membahas masalah praktis dalam menjangkau semua siswa dan membuat guru mahir. Ketiga, berpikir komputasi membawa pendidikan sains dan matematika lebih sesuai praktik profesional saat ini.

Saran/rekomendasi dari penelitian ini adalah masih banyak pekerjaan yang harus dilakukan untuk mengatasi tantangan dalam mendidik populasi yang cerdas secara teknologis dan ilmiah dan mempersiapkan generasi penerus ilmuwan kelas dunia.

Sung, W., Ahn, J., & Black, J. (2017). *The Design of Embodied Activities Promoting Computational Thinking and Mathematics Learning in Early-childhood ...*

Penelitian ini membahas tentang bagaimana membuat proses berpikir komputasi lebih konkrit dan relevan dengan konteks kurikulum standar, khususnya matematika, sehingga siswa dapat menerapkan kemampuan berpikir komputasi dalam bidang STEM seperti pemrograman dan robotic. Berpikir komputasi dalam penelitian ini didefinisikan sebagai sikap yang dapat diaplikasikan secara umum dan keterampilan yang setiap orang akan sangat bersemangat untuk menggunakannya. Kegiatan pembelajaran dalam penelitian ini menambahkan konkrit dan analitis sebagai contoh dari penalaran yang inheren dalam berpikir komputasi untuk menyelesaikan masalah matematika. Desain penelitian ini mengadopsi wujud perspektif untuk mendesain kegiatan yang mengkombinasikan matematika dan komputasi dalam praktiknya.

Metode yang digunakan dalam penelitian ini adalah kuantitatif. Penelitian ini menguji tentang pengaruh dari

kegiatan dengan berpikir komputasi sebagai strategi instruksional dalam perkembangan anak pada kemampuan matematika dan pemrograman awal melalui pengalaman dari berpikir komputasi. Subjek dalam penelitian ini adalah 59 siswa TK dan siswa kelas 1 yang terdiri dari 33 laki-laki dan 26 perempuan.

Penelitian ini menghasilkan temuan dan implikasi menarik bagi penelitian selanjutnya. Analisis utama menunjukkan signifikansi efek utama untuk tingkat yang lebih tinggi pada materi number sense, kemampuan aritmatika, dan kemampuan efisiensi program.

Doleck, T., Bazalais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between *Computational Thinking* skills and academic performance. *Journal of Computers in Education*, (August). <https://doi.org/10.1007/s40692-017-0090-9>

Penelitian ini bertujuan untuk menguji perbedaan antara berpikir komputasi dan hasil akademik dan spesifikasi dari komponen-komponen berpikir komputasi yang beragam yang berhubungan dengan hasil akademik. Dalam penelitian ini, peneliti menyebutkan bahwa berpikir komputasi dapat diidentifikasi sebagai berpikir algoritmik, kekerjasamaan, kekreatifan, berpikir kritis dan menyelesaikan masalah.

Metode yang digunakan dalam penelitian ini adalah metode kuantitatif. Dimana subjeknya adalah calon mahasiswa di Kanada sebanyak 104 orang yang terdiri dari 54 wanita dan 50 pria yang rata-rata uasiannya adalah 17,9 tahun. Subjek mengisi angket dala dua sesi. Sesi pertama disebut sesi pendahuluan, yang meliputi informasi demografis dan akademik. Sesi kedua disebut sesi kompetensi berpikir

komputasi. Dalam sesi kedua, kompetensi berpikir komputasi diukur dengan skala berpikir komputasi. Skala tersebut terdiri dari 29 item dan dibagi ke dalam 5 ranah, yaitu: berpikir algoritmik, kekerjasamaan, kekreatifan, berpikir kritis, dan penyelesaian masalah. Skala yang digunakan adalah skala Likert dengan skor dari 1= tidak pernah sampai 5= selalu. Hasil akademik diukur dengan menggunakan CEGEP R-Score. Demografis dan prestasi (usia, jenis kelamin, GPA di SMA) digunakan sebagai variabel kontrol.

Hasil dari penelitian ini adalah bahwa kekerjasamaan mempunyai kaitan yang negative dengan hasil akademik, berdasarkan usia, jenis kelamin dan prestasi akademik sebelumnya. Sebaliknya, tidak ditemukan hubungan yang signifikan antara kemampuan berpikir komputasi dan hasil akademik lainnya. Kurangnya hubungan ini menunjukkan bahwa diperlukan penyelarasan kurikulum yang baik yang berguna bagi perkembangan berpikir komputasi siswa dengan keterampilan lainnya.

Psycharis, S. (2016). Inquiry Based-Computational Experiment, Acquisition of Threshold Concepts and Argumentation in Science and Mathematics Education. *Educational Technology & Society*, 19(3), 282–293

Tujuan dari penelitian ini untuk mengeksplorasi pengaruh dari pendekatan pembelajaran IBSE pada mahasiswa yang meliputi (a) argumentasi, (b) keterlibatan penggunaan dari indikator pemodelan, dan (c) akuisisi dari kepastian konsep awal fisika dan matematika.

Metode yang digunakan dalam penelitian ini adalah metode kuantitatif. Statistik uji yang digunakan adalah uji-t dan analisis regresi.

Hasil dari penelitian ini adalah selama intervensi Teaching Observation Protocol digunakan berdasarkan desain dan implementasi Lesson (yaitu, misalnya jika strategi dan aktivitas instruksional menghargai pengetahuan dan prakiraan siswa sebelumnya yang melekat di dalamnya, jika pelajaran ini mendorong siswa untuk mencari dan menilai cara-cara penyelidikan alternatif atau penyelesaian masalah). Dalam penelitian ini satu model instruksi untuk pengembangan Skema Pedagogi berbasis Skema dijelaskan. Pendekatan instruksional ini diusulkan sebagai pendekatan untuk memahami dan mempromosikan sebuah metodologi menuju pengembangan (CT) dalam model berbasis inquiry di bidang Fisika dan Matematika.

Inti dari model instruksi adalah pengembangan model simulasi berdasarkan alat EJS dan ini bisa terjadi dapat membantu guru karena mereka bisa mengembangkan cara berpikir yang lebih canggih tentang bagaimana teknologi bisa dilakukan mengubah praktik mengajar mereka. Akhirnya, terlepas dari hasil optimis hasil penelitian ini, kami anggap bahwa pengembangan skenario berbasis Komputasi Model-Inquiry, bukanlah tugas yang mudah. Akibatnya, intensif, Upaya sistematis dan terkoordinasi perlu direncanakan dan dilaksanakan dalam layanan pra-layanan dan in-service program pendidikan untuk mengembangkan kegiatan kurikulum berdasarkan model pengajaran dan penelitian masa depan harus dilakukan untuk memvalidasi, memodifikasi, atau memperbaiki kerangka kerja yang diusulkan dalam makalah penelitian ini. Akhirnya, penelitian menunjukkan bahwa guru dan banyak anggota staf enggan menyarankan (TC) konsep, atau menggambarkannya dari konsep inti (Wilson et al., 2010). Studi penelitian ini dapat memberikan kerangka kerja bagi pendidik untuk menangani dengan konsep ambang batas

dengan cara yang lebih sistematis dengan mengembangkan proses pembelajaran yang mengintegrasikan fitur inquiry dan model komputasi.

Ambrosio, A. P., & Franco, A. (2014). *Exploring Core Cognitive Skills of Computational Thinking*.

Makalah ini menyajikan sebuah studi eksplorasi yang dikembangkan untuk memilih tes penilaian psikologis yang dapat digunakan untuk mengidentifikasi dan mengukur proses kognisi Berpikir Komputasional, yang terkait dengan komponen pemrograman, sehingga strategi dapat dikembangkan untuk mempromosikannya. Meski masih inovatif dan tidak banyak disebarluaskan, *Computational Thinking* dianggap sebagai keterampilan kritis bagi siswa di abad ke-21. Ini melibatkan banyak keterampilan, namun kemampuan pemrograman tampaknya menjadi aspek inti karena mereka mendorong pengembangan cara berpikir baru yang merupakan kunci pemecahan masalah yang memerlukan kombinasi kekuatan mental dan kemampuan komputasi manusia.

Penelitian ini merupakan penelitian pengembangan. Setelah tinjauan literatur, mengidentifikasi empat proses kognitif sentral yang tersirat dalam pemrograman, oleh karena itu penting bagi Pemikiran Komputasi, dan karenanya memilih seperangkat empat tes yang diberikan kepada sampel dari 12 siswa pemrogram pengantar.

Hasil penelitian ini menunjukkan bahwa penalaran spasial dan kecerdasan umum adalah dimensi penting untuk pemrograman pengantar, juga berkorelasi dengan keberhasilan akademis siswa di bidang ini. Namun, penalaran aritmetika dan perhatian terhadap tes detail tidak berkorelasi. Berdasarkan hasil ini, arahan untuk penelitian masa depan telah



didefinisikan untuk secara efektif mengidentifikasi dan mengembangkan proses kognitif inti pemrograman, ergo, untuk membantu mengembangkan Pemikiran Komputasi.

Atmatzidou, S., & Demetriadis, S. (2014). How to Support Students ' *Computational Thinking* Skills in Educational Robotics Activities. In *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education* (pp. 43–50).

Penelitian ini menggambarkan implementasi dan evaluasi pengembangan kemampuan berpikir komputasi dalam kegiatan Pendidikan Robotika untuk sekolah teknik sekunder, yang berfokus pada keterampilan dasar CT, yaitu: abstraksi, generalisasi, algoritma, modularitas, dekomposisi dan pemecahan masalah.

Penelitian ini merupakan penelitian penerapan suatu alat ( Lego Mindstorms NXT 2.0), dalam penelitian ini ada pra-dan pasca kuesioner dan serangkaian wawancara dengan pemikiran. Adapun model kemampuan berpikir komputasi antara lain:

#### 1. Abstraksi

Definisi: abstraksi adalah proses penciptaan suatu bentuk sederhana dari bentuk kompleks dengan menghilangkan hal-hal yang tidak relevan, menemukan pola yang relevan, dan memisahkan ide/ gagasan dari hal-hal nyata.

Pedoman pengembangan:

- a. Memisahkan informasi penting dari informasi yang berlebihan yang tidak terpakai
- b. Menganalisis dan menspesifikasi tingkah laku umum/kebiasaan atau struktur pemrograman antara skrip yang berbeda

- c. Mengidentifikasi abstraksi antara lingkungan pemrograman yang berbeda.

## 2. Generalisasi

Definisi: generalisasi adalah mentransfer proses pemecahan masalah ke permasalahan yang lebih bervariasi.

Pedoman pengembangan :

- a. Memperluas solusi yang ada dalam masalah yang diberikan untuk menutupi lebih banyak kemungkinan/kasus.
- b. Menggunakan variabel dalam solusi.

## 3. Algoritma

Definisi: algoritma adalah praktek menulis langkah demi langkah, spesifik dan jelas, intruksi untuk melakukan proses.

Pedoman pengembangan:

- a. Menggunakan kata-kata eksplisit dalam langkah-langkah algoritma
- b. Kemungkinan algoritma yang berbeda untuk masalah yang sama
- c. Usaha untuk menemukan algoritma yang paling efektif.

## 4. Modularitas

Definisi: modularitas adalah pengembangan dari proses otonom, yang merangkum seperangkat perintah yang sering digunakan yang menunjukkan fungsi tertentu dan mungkin digunakan dalam masalah yang sama atau berbeda.

Pedoman pengembangan: Kembangkan bagian kode otonom untuk digunakan untuk masalah yang sama atau berbeda.

## 5. Dekomposisi

Definisi: dekomposisi adalah proses memecah masalah menjadi bagian yang lebih kecil yang mungkin diselesaikan dengan lebih mudah.

Pedoman pengembangan: pecah masalah menjadi lebih kecil/ tunggal yang lebih mudah untuk diselesaikan.

Hasilnya menunjukkan bahwa para siswa menjadi terbiasa dengan konsep CT, dan mengintegrasikannya ke tingkat yang memuaskan dalam proses pemecahan masalah dalam aktivitas ER.

Saran/ rekomendasi adalah perlu adanya penelitian lebih lanjut guna memperbaiki model yang diusulkan untuk mendukung pengembangan keterampilan CT, dengan fokus pada: (a) memperkaya lembar kerja dengan kegiatan yang ditargetkan yang membimbing dan mendukung siswa (b) meningkatkan jumlah sesi, dan (c) membuat Penelitian lebih luas tentang penilaian.

Bilbao, J., García, O., Rebollar, C., Bravo, E., & Varela, C. (2016). Skills, attitudes and concepts of the *Computational Thinking* essential tool for doing scientific. In *Economics and Education* (pp. 82–87).

Tujuan dari penelitian ini adalah untuk mengetahui kemampuan, sikap dan konsep dari berpikir komputasi.

Penelitian ini merupakan penelitian deskriptif. Berpikir komputasi meliputi pemecahan masalah, mendesain system, dan memahami tingkah laku manusia, dengan menggambarkan konsep dasar pada komputer sains. Karakteristik utama dari berpikir komputasi meliputi:

- a. Menganalisis dan mengorganisasikan data secara logis
- b. Memodelkan, mengabstraksikan data dan mensimulasikannya

- c. Merumuskan masalah yang mungkin dibantu computer
- d. Mengidentifikasi, menguji, dan mengimplementasi solusi yang mungkin
- e. Mengotomatisasi solusi dengan berpikir algoritmik
- f. Menggeneralisasi dan menerapkan proses tersebut ke berbagai masalah

Konsep pemikiran komputasi yang tersirat dalam definisi operasional adalah sebagai berikut:

1. Pengumpulan data: proses pengumpulan informasi yang tepat
2. Analisis Data: membuat rasa data, menemukan pola, dan membuat kesimpulan
3. Representasi data: menggambarkan kembali dan mengorganisasi data dalam gambar yang tepat, grafik, kata-kata, atau gambar.
4. Dekomposisi masalah: memecah-mecah tugas menjadi lebih kecil, mengatur bagian.
5. Abstraksi: mengurangi kompleksitas untuk mendefinisikan ide pokok.
6. Algoritma dan prosedur: Serangkaian langkah yang diperintahkan yang diambil untuk memecahkan suatu masalah atau mencapai beberapa akhir.
7. Otomatisasi: Memiliki komputer atau mesin melakukan tugas yang berulang dan membosankan.
8. Simulasi: representasi atau model dari proses. Simulasi juga meliputi menjalankan eksperimen menggunakan model.
9. Paralelisasi: mengorganisir sumber-sumber untuk mengatur sumber daya untuk sekaligus menjalankan tugas untuk mencapai tujuan bersama.

Hasil dari penelitian ini adalah salah satu keuntungan utama mengembangkan berpikir komputasi adalah

memungkinkan transformasi masyarakat konsumen teknologi belaka di salah satu pengembang potensial teknologi ini.

Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing *Computational Thinking* into High School Mathematics and Science Classrooms. In *Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) 2016, Volume 2* (pp. 705–712)

Tujuan dari penelitian ini adalah untuk menyelidiki pendekatan untuk menanamkan konten berpikir komputasi ke dalam matematika dan sains SMA yang ddikipersyaratkan.

Pengumpulan data dalam penelitian ini dilakukan selama 3 tahun untuk menyelidiki keefektifan dari strategi menanamkan CT pada matematika dan sains. Selama proyek berlangsung, 58 guru mengikuti lokakarya pengembangan profesional dari 38 sekolah. Data yang dihadirkan berasal dari 11 ruang kelas di kota Midwestern yang berpartisipasi dalam tahun ketiga proyek tersebut. Sebagai bagian dari penelitian ini, penilaian keterampilan pra / post attitudinal dan CT diberikan bersamaan dengan observasi kelas dan wawancara guru. Survei sikap dimodelkan setelah upaya serupa lainnya untuk mengukur sikap siswa dalam konteks STEM dan komputer. Penilaian keterampilan pra / pasca dirancang sebagai bagian dari proyek yang lebih besar dan dirancang untuk menilai kemampuan siswa dalam menerapkan praktik CT, berlawanan dengan pengetahuan konten dari domain ilmiah atau matematika tertentu. Penilaian dilakukan secara online dan meminta siswa untuk menggunakan berbagai alat komputasi (termasuk visualisasi data interaktif, model komputasi dan simulasi, dan widget pengelolaan data dinamis) untuk menjawab pertanyaan pilihan terbuka dan pilihan ganda yang

berkaitan dengan keempat CT dalam kategori matematika dan sains.

Hasil dari penelitian ini adalah Sebagai metodologi komputasi, peralatan, dan praktik terus mendorong penemuan ilmiah dan matematika, semakin penting bagi peserta didik untuk memahami bagaimana menafsirkan, dan membangun, temuan yang bergantung pada teknologi semacam itu. Hal ini penting tidak hanya bagi siswa yang tertarik untuk mengejar karir di bidang matematika atau sains, namun bagi semua peserta didik agar dapat berpartisipasi dalam masyarakat secara ilmiah dan matematis, warga negara yang terpelajar. Selama tiga tahun terakhir, telah ditempuh pendekatan untuk mengenalkan pelajar SMA kepada praktik pemikiran komputasi kritis ini dengan merancang pelajaran yang disempurnakan oleh CT yang sesuai dengan kurikulum matematika dan sains yang ada. Dengan karya ini, menunjukkan bahwa pendekatan ini efektif untuk menjangkau khalayak yang beragam dan mudah diadopsi oleh guru in-service. Selanjutnya, penyajian data yang menunjukkan efek dosis, menunjukkan bahwa semakin banyak CT dalam aktivitas matematika dan sains yang dihadapi peserta, semakin baik kinerjanya pada penilaian praktik CT. Temuan ini menunjukkan bahwa menciptakan lebih banyak aktivitas, dan menemukan lebih banyak cara untuk meningkatkan rencana pelajaran yang ada dengan praktik berpikir komputasi, selanjutnya akan meningkatkan kemampuan belajar CT dalam kemampuan matematika dan sains.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *AERA* (pp. 1–25).

Penelitian ini merupakan penelitian merancang kegiatan pembelajaran berbasis media interaktif yang mendukung pengembangan berpikir komputasi.

Tahap pertama penelitian ini adalah mendeskripsikan dimensi kunci dari kerangka berpikir komputasi yaitu: konsep komputasi, praktik komputasi, dan perspektif komputasi. Tahap kedua mendeskripsikan pendekatan yang digunakan untuk mengembangkan menilai dimensi-dimensi ini, termasuk analisis portofolio proyek, wawancara berbasis artefak, dan skenario desain. Tahap terakhir merupakan serangkaian saran untuk menilai pembelajaran yang terjadi saat orang muda terlibat dalam pemrograman.

Hasil penelitian ini adalah Scratch - lingkungan pemrograman yang memungkinkan kaum muda untuk membuat cerita, permainan, dan simulasi interaktif mereka sendiri, dan kemudian membagikan ciptaan tersebut di komunitas online dengan pemrogram muda lainnya dari seluruh dunia

Voskoglou, M. G. (2013). Problem Solving, Fuzzy Logic and Computational Thinking. *Egyptian Computer Science Journal ECS*, 37(1), 131–145.

Penelitian ini bertujuan untuk mengembangkan model fuzzy untuk proses berpikir komputasi dengan mewakili tahap utama dari proses pemodelan sebagai subhimpunan fuzzy dari satu set *label linguistic* yang mencirikan kinerja para modelling dalam masing-masing tahap ini. Selain itu penelitian ini juga menerapkan metode “centroid” untuk mendapatkan ukuran keterampilan berpikir komputasi seseorang.

Dalam penelitian ini disajikan dua kelas eksperimen dengan menggambarkan penggunaan model fuzzy dalam

praktik. Berpikir komputasi adalah metode pemecahan masalah baru yang dinamai untuk penggunaan ekstensif teknik sains komputer. Ini mensintesis pemikiran kritis dan pengetahuan yang ada dan menerapkannya untuk memecahkan masalah teknologi dunia nyata yang kompleks. Berpikir pemodelan merupakan inti dari CT, karena ia mensintesis semua komponen lain dari CT (pemikiran abstrak, logis dan konstruktif) untuk pemecahan masalah yang sesuai.

Hasil penelitian ini memberi indikasi kuat bahwa penggunaan komputer sebagai alat untuk pemecahan masalah meningkatkan kemampuan siswa dalam memecahkan masalah dunia nyata yang melibatkan pemodelan matematika.

- L. Junsay, M. (2016). Reflective Learning and Prospective Teachers' Conceptual Understanding, Critical Thinking, Problem Solving, and Mathematical Communication Skills. *Research in Pedagogy*, 6(2), 43–58. <https://doi.org/10.17810/2015.34>

Tujuan dari penelitian ini adalah untuk mengeksplorasi efek pembelajaran reflektif terhadap pemahaman konseptual para guru, pemikiran kritis, pemecahan masalah, dan kemampuan komunikasi matematis dan hubungan variabel-variabel ini.

Penelitian ini merupakan penelitian eksperimental semu pre-test - post-test yang melibatkan 60 calon guru dari dua kelas matematika dasar sebuah institusi pendidikan tinggi. Ada dua perlakuan yang digunakan dalam penelitian ini: pendekatan ceramah-diskusi dan pendekatan pembelajaran reflektif. Ceramah-diskusi melibatkan lima tahap: pendahuluan, presentasi, pemahaman, pemantauan, integrasi, dan peninjauan dan penutupan. Pendekatan pembelajaran reflektif melibatkan proses inisialisasi, eksplorasi, dan penghubungan. Uji coba peneliti yang telah divalidasi dan reliabel terhadap pemahaman



konseptual, pemikiran kritis, pemecahan masalah, dan keterampilan komunikasi matematis disajikan sebagai instrument.

Hasil penelitian menunjukkan bahwa jumlah calon guru yang terpapar pembelajaran reflektif dan pendekatan diskusi ceramah di bawah 50% dari nilai sempurna sebelum dan bahkan setelah intervensi. Lebih jauh lagi, menunjukkan bahwa pemahaman konseptual para guru, pemikiran kritis, pemecahan masalah dan kemampuan komunikasi matematika meningkat secara signifikan dalam kelompok belajar reflektif, dan dalam kelompok diskusi ceramah. Namun, keuntungan rata-rata kedua kelompok berbeda secara signifikan dalam mendukung kelompok belajar reflektif dalam semua keterampilan yang disebutkan di atas kecuali dalam pemikiran kritis. Selanjutnya, pemahaman konseptual para guru, pemikiran kritis, pemecahan masalah, dan kemampuan komunikasi matematis semuanya diketahui secara signifikan terkait dalam kelompok belajar reflektif. Namun, dalam kelompok diskusi ceramah, hanya pasangan berikut yang berhubungan secara signifikan: pemahaman konseptual dan kemampuan komunikasi matematis; pemahaman konseptual dan keterampilan pemecahan masalah; dan pemecahan masalah dan kemampuan komunikasi matematis.

Loong, Esther Yook-Kin 2014, A primary teacher's developing understanding of mathematical reasoning, in MERGA 2014: Curriculum in focus: research guided practice: Proceedings of the Mathematics Education Research Group of Australasia 2014 annual conference, MERGA, Adelaide, S. A., pp. 706-709.

Penelitian ini bertujuan untuk mengkaji dampak pelajaran demonstrasi tentang pengetahuan pedagogis guru

tentang penalaran sebagai kemampuan matematika yang ada di dalam kurikulum Matematika Australia (berdasarkan proyek penelitian pembelajaran professional yang menggunakan pelajaran demonstrasi).

Metode yang digunakan dalam penelitian ini adalah studi kasus. Pertumbuhan pengetahuan guru ini dianalisis dengan menggunakan kerangka fenomenografi yang dikembangkan untuk mengevaluasi perkembangan guru dalam penalaran matematis. Jenis-jenis penalaran ada empat, yaitu penalaran induktif, deduktif, adaptif dan inferensi.

Hasil dari penelitian ini menunjukkan bahwa demonstrasi dan pelajaran percobaan selanjutnya memberikan kontribusi terhadap pertumbuhannya.

Anderson, N. D. (2016). A Call for *Computational Thinking* in Undergraduate Psychology. *Psychology Learning and Teaching*, 15(3), 226–234. <https://doi.org/10.1177/1475725716659252>

Penelitian ini bertujuan untuk mendefinisikan *Computational Thinking* (CT) di bidang ilmu psikologi.

Metode penelitian ini adalah penelitian deskriptif. Peneliti mendeskripsikan CT menjadi 5 komponen utama, yaitu:

- a. **Dekomposisi (penguraian/pemecahan):** merupakan proses menyederhanakan masalah yang kompleks menjadi sub-masalah yang lebih kecil yang kemudian dipecahkan secara individu atau diuraikan lebih jauh tergantung pada kompleksitas sub masalah tersebut. peneliti akan memecahkan masalah ini menjadi serangkaian isu " yang lebih kecil 'yang bisa didekati secara sistematis. Isu seperti jenis desain penelitian yang akan digunakan, demografi subjek yang akan diuji, dan jenis rangsangan yang akan ada disajikan, semuanya merupakan contoh sub-masalah yang

bisa didekomposisi dari keseluruhan pertanyaan penelitian. Semakin rinci peneliti bisa dengan dekomposisi proses, semakin baik untuk merancang serangkaian langkah yang diformulasikan dengan baik untuk solusi tercapai.

- b. **Pengenalan pola:** langkah lanjutan setelah dekomposisi yaitu mencari pola informasi sepanjang masalah tersebut. Pola, dalam konteks ini, mencerminkan jenis apa pun informasi yang berulang. Pola itu bisa bersifat fisik (yang bisa menggambarkan beberapa rangsangan yang mungkin dirancang), atau mungkin lebih konseptual di alam. Dalam penelitian, untuk contohnya, akan ada pola yang menggambarkan apa yang harus disampaikan dari persidangan ke pengadilan, dan peneliti perlu mengidentifikasi dengan tepat apa pola itu. Sekali pola sudah diidentifikasi, pemecah masalah kemudian dapat mengembangkan solusi untuk satu iterasi pola. Setelah solusi yang telah dikembangkan, kemudian dapat diterapkan untuk semua iterasi dari pola secara otomatis. Pada akhirnya, ini harus membuat proses pemecahan masalah lebih banyak efisien.
- c. **Abstraksi:** Langkah ketiga dari pemikiran komputasi melibatkan menghasilkan representasi abstrak dari pola itu setelah diidentifikasi. Untuk tujuan ini, pemecah masalah perlu disaring keluar dari rincian pola sehingga representasi pola umum dapat digunakan untuk memecahkan masalah. Misalnya, sekali pola untuk generasi percobaan telah diidentifikasi, peneliti perlu menggeneralisasi pola itu sehingga bisa dijadikan "cetak biru" untuk apa yang harus terjadi di setiap persidangan. Pada tahap ini, masalah sekarang telah diwakili secara simbolis, dan representasi simbolis itu dapat

diterapkan pada situasi dimana pola diamati. Pola abstrak yang dihasilkan dari tahap komputasi ini berpikirlah menjadi model yang mewakili semua iterasi dari pola.

- d. **Desain algoritma:** Begitu representasi abstrak dari pola telah dihasilkan, pemecah masalah saat itu merancang sebuah algoritma untuk menghasilkan solusi iteratif untuk masalah ini. Algoritma adalah serangkaian instruksi eksplisit yang perlu diikuti untuk mencapai beberapa tujuan. Sebuah contoh dari sebuah algoritma dalam penelitian psikologis akan menjadi kontrol arus kejadian itu tentukan apa yang terjadi selama percobaan. Algoritma selalu dimulai dengan titik awal yang ditentukan dan titik akhir yang ditentukan, dan pemecah masalah menentukan rangkaian langkah yang perlu dilakukan diikuti untuk mendapatkan dari awal sampai akhir.
- e. **Evaluasi:** Langkah terakhir dalam proses berpikir komputasional adalah evaluasi algoritma. Sekali algoritma telah dihasilkan, sangat penting untuk memastikan bahwa algoritma berisi semua langkah yang diperlukan yang diperlukan untuk memecahkan masalah. Jika sebuah algoritma dirancang untuk eksperimen, misalnya, peneliti harus berjalan melalui percobaan langkah demi langkah untuk memastikan bahwa semua proses yang diperlukan disertakan. Masalah di algoritma mungkin membuat produk menjadi tidak valid; Oleh karena itu, sangat penting bahwa solver masalah menguji algoritma secara metodis dan sistematis. Untuk mencapai ini, pemecah masalah harus memastikan bahwa algoritma ini komprehensif (yaitu, berisi semua langkah yang diperlukan untuk solusi) dan efisien (yaitu, tidak mengandung langkah-langkah yang tidak perlu). Di Ilmu komputer, tahap pemrograman ini disebut sebagai "debugging" panggung. Hanya setelah

tahap ini selesai, pemecah masalah harus memastikan bahwa mereka telah mengembangkan solusi komprehensif untuk masalah yang dihadapi.

Penerapan berpikir komputasional dalam kelas psikologi adalah pendekatan berpikir komputasi akan menguntungkan setiap jenis penyelesaian masalah terbuka. Salah satu contoh paling umum dari masalah terbuka dalam psikologi ditemukan di dalamnya desain penelitian. Jenis pertanyaan penelitian yang biasanya ditanyakan dalam psikologi adalah open-ended, sehingga pemikiran komputasi menyediakan kerangka kerja untuk mengembangkan solusi pertanyaan penelitian ini. Manfaat tambahan dalam menggunakan pendekatan berpikir komputasi di Indonesia adalah bahwa banyak eksperimen sekarang dilakukan dengan menggunakan komputer. Jika pendekatan berpikir komputasi digunakan untuk mengembangkan solusi penelitian, kemudian solusi tersebut dilakukan sifat diformulasikan dengan cara yang memudahkan mereka untuk diimplementasikan oleh komputer. Pemikiran komputasional dapat dengan mudah diperkenalkan dalam kursus mana pun siswa berada, diperlukan untuk merancang paradigma eksperimen mereka sendiri untuk menjawab pertanyaan penelitian terbuka. Jenis tugas ini khas dalam banyak metode penelitian, atau tingkat atas kursus laboratorium atau seminar (misalnya, Saville, 2008). Misalnya, instruktur mungkin bertanya kelas mereka merancang eksperimen untuk menentukan apakah respons emosional dapat dipikirkan melalui paparan rangsangan emosional lainnya. Ini adalah masalah terbuka, dan tujuannya bagi siswa adalah merancang solusi untuk masalah (yaitu merancang eksperimen).

Hasil penelitian ini adalah salah satu contoh proses psikologi yang akan mendapat manfaat dari pemikiran

komputasi misalnya dalam menentukan rangsangan akan dipaparkan di setiap persidangan, merancang jenis tertentu rangsangan, atau bahkan reorganisasi data untuk analisis statistik, hanya beberapa contoh saja proses yang bisa dioptimalkan melalui pemikiran komputasi. Pemikiran komputasional, secara alami, adalah keterampilan yang mendorong kecerdikan yang bisa digunakan untuk mengembangkan baru dan solusi kreatif untuk masalah psikologi. Di luar penerapan praktis pemikiran komputasi dalam psikologi, siswa akan melakukan manfaat lebih umum dengan mengembangkan keterampilan ini juga. Berpikir komputasional dengan cepat mendapatkan daya tarik sebagai keterampilan dasar yang akan dibutuhkan di masa depan, sama seperti membaca, menulis, dan aritmatika (Lu & Fletcher, 2009; Wing, 2006). Bagi siswa yang berminat dalam mengejar karir dalam penelitian, banyak yang akan diminta untuk belajar bagaimana memprogram beberapa orang titik dalam karir akademis mereka. Mengajar para siswa dengan dasar-dasar berpikir komputasi sangat penting untuk mengembangkan keterampilan pemrograman yang lebih umum. Di luar akademisi, pemikiran komputasi adalah keterampilan yang tak ternilai untuk berbagai karir, dari yang dibutuhkan pemecahan masalah kreatif (misalnya konsultan) kepada mereka yang melibatkan data besar (misalnya pasar analisis penelitian). Berpikir komputasional adalah keterampilan yang akan bermanfaat bagi siswa kita dari lintasan karir mereka.

Bower, M., Wood, L. N., Howe, C., & Lister, R. (2017). Improving the *Computational Thinking* Pedagogical Capabilities of School Teachers. *Australian Journal of Teacher Education*, 42(3), 53–72.

Penelitian ini bertujuan untuk mengembangkan kemampuan pedagogis berpikir komputasi guru, untuk menentukan bagaimana guru bisa didukung pengembangan profesional dengan sebaik-baiknya dan penyediaan sumber daya. Serta membahas isu-isu strategis dengan menggambarkan serangkaian lokakarya berpikir komputasi untuk guru pendidikan dasar kelas 8 di Universitas di Australia.

Penelitian ini merupakan penelitian mix method antara penelitian kualitatif dan kuantitatif. Subjek dalam penelitian ini adalah 91 dan 75 guru yang mengikuti pelatihan. Pengumpulan data dilakukan dengan melakukan survey pra dan pasca pelatihan. Analisis data, pertanyaan terbuka dari survei pra dan pasca lokakarya dianalisis dengan menggunakan teknik analisis tematik kualitatif untuk mengidentifikasi 1) konsep yang berkaitan dengan pemikiran komputasi (konten, pedagogi, dan teknologi), 2) kesulitan dalam pembelajaran pedagogi baru, dan 3) mendukung guru yang dibutuhkan. Sedangkan, pertanyaan tertutup (yaitu, data kuantitatif) guru pra dan pasca-survei lokakarya dianalisis dengan Statistik SPSS 22.0. Uji t sampel berpasangan digunakan untuk menentukan apakah ada perbedaan yang signifikan secara statistik antara alat pra dan pasca lokakarya tentang persepsi guru tentang pentingnya anak mengembangkan kemampuan berpikir komputasi dan tingkat kepercayaan tentang pengembangan kemampuan siswa.

Hasil penelitian adalah pemahaman pemikiran komputasi guru, kemampuan pedagogis, pengetahuan dan kepercayaan teknologi dapat ditingkatkan dalam waktu yang relatif singkat melalui pembelajaran profesional yang ditargetkan.

Sanford, J. F., & Naidu, J. T. (2017). Mathematical Modeling And Computational Thinking. *Contemporary Issues in Education Research-Second Quarter 2017*, 10(2), 159–168.

Penelitian ini menyajikan beberapa contoh pemodelan matematika menggunakan spreadsheet pada tingkat lanjutan seperti sekolah menengah atas atau awal.

Penelitian ini merupakan penelitian pengembangan model yang disajikan di untuk menunjukkan penerapan spreadsheet. Diasumsikan pada tingkat SMA dan perguruan tinggi sudah mengenal spreadsheet. Sebagai contoh permasalahan adalah perusahaan manufaktur berencana mengenalkan produk baru. Ada berbagai faktor yang terlibat dalam proses ini, dan daftar faktor penting muncul di bawah ini, yaitu:

- Perkiraan volume penjualan paling sedikit pada awal umur produk
- Bahan yang terlibat dan biayanya
- Metode dan biaya produksi
- Saluran dan biaya distribusi
- Pajak dan biaya perawatan

Hasil penelitian adalah berbagai permasalahan di dunia bersifat analitik dan pada umumnya digambarkan dengan model matematis. Sangat diharapkan bahwa konsep pemikiran komputasi diperkenalkan di awal pengalaman pendidikan. Penggunaan spreadsheet sebagai tambahan pemecahan masalah yang biasanya menjadi bagian dari kurikulum sekolah menengah dan awal dapat mengajarkan konsep berharga pemodelan matematika dan berpikir komputasi.

Yasar, O., Maliekal, J., Veronesi, P., & Little, L. (2017). The essence of *Computational Thinking* and tools to promote it. In *American*



*Society for Engineering Education*. Retrieved from  
<https://www.asee.org/public/conferences/78/papers/17618/vi>  
ew

Penelitian ini bertujuan untuk menghubungkan kognisi dengan proses komputasi dasar, khususnya pemodelan dan simulasi.

Penelitian ini merupakan penelitian kuantitatif dengan subjek 15 sekolah menengah.

Hasil penelitian ini adalah menunjukkan dampak yang sama terhadap pembelajaran siswa. Sebelum mengajarkan keterampilan berpikir komputasi secara elektronik, sebaiknya mengajarkan kebiasaan kognitif yang relevan kepada siswa. Karena hal ini tidak hanya akan meningkatkan kemampuan berpikir kritis tetapi juga memberikan motivasi dan kesiapan mereka untuk belajar berpikir komputasi secara elektronik.

Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. (2015). Empowering K – 12 Students With Disabilities to Learn *Computational Thinking* and Computer Programming. *TEACHING Exceptional Children*, 48(1), 45–53. <https://doi.org/ISSN: 0040-0599>

Artikel ini membahas tentang pemberdayaan siswa kelas 12 yang mengalami disabilitas dalam mempelajari berpikir komputasi dan pemrograman komputer.

Artikel ini merupakan suatu deskripsi tentang bagaimana berpikir komputasi diajarkan di kelas 12?, Strategi yang dapat meningkatkan akses dan keterlibatan siswa dalam pendidikan komputasi.

Ada beberapa strategi bagi pendidik khusus yang dapat meningkatkan kesempatan siswa dengan disabilitas untuk berhasil dalam pendidikan komputasi. Praktik pembelajaran

komputasi harus disesuaikan dengan kebutuhan masing-masing siswa untuk mengembangkan pengalaman komputasi yang bermakna, menarik, dan mudah dilakukan oleh siswa penyandang disabilitas.

Hartnett, J. (2015). Teaching Computation in Primary School without Traditional Written Algorithms. In *Proceedings of the 38th annual conference of the Mathematics Education Research Group of Australasia* (pp. 285–292).

Penelitian ini bertujuan untuk melaporkan sebuah proyek di satu sekolah di mana staf setuju untuk pengajaran algoritma tradisional dan menggantinya dengan strategi komputasi.

Penelitian ini termasuk penelitian mix method. Subjek dalam penelitian ini adalah siswa sekolah dasar pada tahun 1 sampai 7. Teknik analisis data menggunakan kualitatif dan kuantitatif. Data hasil observasi di analisis dengan kualitatif. Analisis kuantitatif menggunakan NAPLAN data. Setelah analisis data dilakukan refleksi terhadap guru dan orang tua.

Hasil dari penelitian ini menunjukkan bahwa siswa dapat berhasil dalam matematika tanpa algoritma tulis tradisional (perhitungan prosedural) sebagai bagian dari program matematika sekolah tetapi diganti dengan menggunakan strategi komputasi (*computational thinking*).

Korucu, A. T. (2017). Examination of the *Computational Thinking Skills* of Students. *Journal of Learning and Teaching in Digital Age*, 2(1), 11–19. Retrieved from <http://joltida.org/index.php/joltida/article/view/25>

Penelitian ini bertujuan untuk mengetahui kemampuan berpikir komputasi siswa sekolah menengah dalam variabel yang berbeda.

Metode yang digunakan dalam penelitian ini adalah metode kuantitatif. Adapun subjek dalam penelitian ini sebanyak 160 siswa sekolah menengah yang melanjutkan pendidikan di berbagai tingkat di Konya. Pengumpulan data menggunakan “skala keterampilan berpikir komputasi” yang dikembangkan oleh Korkmaz, Çakır dan Özden (2015). Instrumen ini terdiri dari 22 butir dan merupakan skala likert 5 titik. Reliabilitas dihitung dengan rumus Cronbach Alpha dan didapat sebesar 0,8 serta memenuhi uji kevalidan untuk mengukur tingkat keterampilan berpikir komputasi siswa.

Hasil penelitian menunjukkan bahwa ada perbedaan keterampilan berpikir komputasi siswa sekolah menengah berdasarkan tingkatan kelas dan durasi kepemilikan teknologi mobile mereka, tetapi tidak ada perbedaan berdasarkan jenis kelamin, durasi penggunaan internet mingguan.

Durak, H. Y., & Saritepeci, M. (2017). Analysis of the relation between *Computational Thinking* skills and various variables with the structural equation model. *Computers & Education*, 116,91–202. <https://doi.org/10.1016/j.compedu.2017.09.004>

Penelitian ini bertujuan untuk mengetahui sejauh mana kemampuan berpikir komputasi siswa berdasarkan berbagai variabel dan untuk menghasilkan model yang menjelaskan dan memprediksi hubungan antara kemampuan berpikir komputasi dan berbagai model tersebut.

Penelitian ini merupakan penelitian pengembangan. Subjek dalam penelitian ini adalah siswa di kelas 5-12 tahun akademik 2015/2016 sebanyak 156 siswa di berbagai sekolah di

Ankara. Model yang digunakan dalam penelitian ini adalah model skrining relational. Adapun pengumpulan data dilakukan dengan dua instrument yang berbeda, yaitu formulir data diri dan skala kemampuan berpikir komputasi. Analisis data menggunakan model persamaan structural (Structure Equation Models/SEM) sehingga menghasilkan model yang menjelaskan dan memprediksi hubungan antara kemampuan berpikir komputasi dan berbagai variabel.

Hasil penelitian menunjukkan bahwa kemampuan berpikir komputasi siswa sangat dipengaruhi oleh gaya berpikir, keberhasilan akademik di kelas matematika dan sikap terhadap kelas matematika.

Saran atau rekomendasi adalah untuk studi di masa depan, direkomendasikan bahwa kurikulum dikaji berdasarkan cara berpikir siswa dalam studi desain lingkungan belajar, yang akan mengajarkan keterampilan berpikir komputasi, dengan mempertimbangkan model yang valid, dan efek dari kurikulum ini diteliti.

Morreale, P., Goski, C., & Jimenez, L. (2012). Measuring the Impact of *Computational Thinking* Workshops in High School Teachers. *JCSC*, 27(6), 151–157.

Yang diteliti dalam penelitian ini adalah dampak dari workshop berpikir komputasi yang diberikan kepada guru sekolah menengah atas di Amerika Serikat. Hal yang melatar belakangi penelitian ini karena banyak Negara bagian Amerika Serikat, matematika dibutuhkan di kurikulum sekolah menengah, sedangkan ilmu komputer tidak. Selama workshop, guru sekolah menengah atas diberikan ilustrasi bagaimana mengembangkan pemikiran komputasi dan keterampilan memecahkan masalah.

Metode yang digunakan dalam penelitian ini adalah metode survei. Subjek dalam penelitian ini adalah guru-guru sekolah menengah di Amerika Serikat. Sebelum dan sesudah workshop guru SMA tersebut mengikuti survei. Hasilnya dianalisis untuk mengetahui apakah ada perubahan dalam persepsi guru tentang berpikir komputasi, dan apakah peserta workshop pertama (musim gugur) yang mengikuti workshop kedua (musim panas) sudah menerapkan materi workshop di kelasnya.

Hasil penelitian ini memberikan kontribusi untuk memahami persepsi pemikiran komputasi dan ilmu komputer di kalangan guru sekolah menengah, serta identifikasi alat dan sumber terbaik yang kemungkinan besar akan digunakan oleh guru SMA dan yang dapat digunakan untuk mengimplementasikan berpikir komputasi dalam standar kurikulum inti, termasuk matematika.

Korkmaz, O., Cakir, R., & Ozden, M. Y. (2017). Computers in Human Behavior A validity and reliability study of the *Computational Thinking scales* ( CTS ). *Computers in Human Behavior*,72,558–569. <https://doi.org/10.1016/j.chb.2017.01.005>

Yang diteliti adalah validitas dan reliabilitas alat yang digunakan untuk mengukur berpikir komputasi.

Penelitian ini merupakan penelitian pengembangan. Subjek dalam penelitian ini adalah 726 mahasiswa di Universitas Amasya, Turki untuk aplikasi pertama dan 580 siswa untuk aplikasi kedua. Validitas dan reliabilitas skala telah dipelajari dengan melakukan analisis faktor eksploratori, analisis faktor konfirmatori, item analisis kekhasan, koefisien konsistensi internal dan analisis keteguhan. Angket yang dihasilkan terdiri dari 8 butir kemampuan komunikasi, 20 butir

berpikir algoritmik, 12 butir berpikir kritis, 8 butir pembelajaran kooperatif, 13 butir kreativitas dan 13 butir pemecahan masalah. Angket tersebut menggunakan skala 5 dengan skor 1 (tidak pernah), 2 (jarang), 3 (kadang-kadang), 4 (sering) dan 5 (selalu).

Hasil dari analisis yang dilakukan, telah disimpulkan bahwa skala adalah alat ukur yang valid dan andal yang dapat mengukur kemampuan berpikir komputasi siswa. Sebagai tambahan; individu usia digital diharapkan memiliki kemampuan berpikir komputasi, dan pada tingkat mana mereka memiliki keterampilan ini.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015).

*Computational Thinking* in compulsory education : Towards an agenda for research and practice. *Educ Inf Technol*, 20, 715–728. <https://doi.org/10.1007/s10639-015-9412-6>

Penelitian ini bertujuan untuk menguraikan lebih jauh tentang *Computational Thinking* dan menyajikan contoh tentang apa yang perlu diajarkan dan bagaimana cara mengajarkannya.

Metode dalam penelitian ini adalah metode penelitian deskriptif kualitatif. Pertama, memposisikan *Computational Thinking* dalam karya Papert dengan LOGO. Kemudian mendiskusikan tantangan dalam mendefinisikan *Computational Thinking* dan membahas aspek inti dan periferan sebuah definisi. Setelah itu menawarkan contoh bagaimana *Computational Thinking* dapat ditangani baik dalam setting pendidikan formal maupun informal. Pada bagian kesimpulan dan pembahasan, sebuah agenda penelitian dan praktik disajikan.

Liu, H. P., Perera, S. M., & Klein, J. W. (2017). Using Model-Based Learning to Promote *Computational Thinking* Education. In *Emerging Research, Practice, and Policy on Computational Thinking, Educational Communications and Technology: Issues and*

*Innovations* (pp. 153–172). <https://doi.org/10.1007/978-3-319-52691-1>

Artikel ini membahas tentang program pembelajaran yang bisa digunakan untuk mengembangkan *Computational Thinking* di semua tingkatan kelas mulai dari K6 sampai K16. Mengingat pesatnya teknologi digital di abad ini, *Computational Thinking* menjadi keterampilan yang mendasar untuk memberdayakan masyarakat Amerika di generasi berikutnya. Oleh karena itu, pendidikan CT di semua disiplin ilmu dan tingkat kelas sedang dianjurkan oleh pemerintah. Namun, kelengkapan di sekolah-sekolah dan universitas untuk mempersiapkan siswa dengan kemampuan dan pengetahuan CT yang dibutuhkan saat ini belum memadai. Dalam artikel ini penulis mengusulkan beberapa program pembelajaran berbasis model untuk mengembangkan pembelajaran CT aktif bagi siswa dari berbagai kelompok usia.

Metode yang digunakan adalah kajian pustaka dan studi kasus tentang pendidikan computational thinking.

Hasil yang dicapai adalah penulis mengusulkan beberapa program pembelajaran berbasis model yang telah penulis telusuri sejak tahun 2012 untuk mempromosikan pembelajaran CT aktif bagi siswa dari berbagai kelompok usia. Sebagian besar program dirancang untuk memanfaatkan program pendidikan out-of-school dan proyek penelitian tim langsung untuk memajukan pendidikan CT dari siswa K6 sampai K16. Dalam konteks CT, program yang diusulkan ada menekankan pada pengembangan kemampuan pemecahan masalah siswa melalui pembelajaran berbasis masalah (PBL) di mana siswa belajar berpikir komputasional dengan menyelesaikan proyek tim. Penulis juga menggambarkan bagaimana universitas kecil dan sekolah K-12 dapat

mengembangkan pendidikan CT secara efektif dengan membentuk koalisi, memanfaatkan teknologi cyberlearning yang sedang berkembang, dan berbagi sumber daya pendidikan.

Rambally, G. (2017). Integrating *Computational Thinking* in Discrete Structures. In *Emerging Research, Practice, and Policy on Computational Thinking, Educational Communications and Technology: Issues and Innovations* (pp. 99–119). Springer International Publishing. <https://doi.org/10.1007/978-3-319-52691-1>

Penelitian ini bertujuan untuk mengintegrasikan *Computational Thinking* ke dalam materi matematika diskrit dengan menggunakan strategi pedagogi pembelajaran berbasis masalah dan kerangka teoritis APOS.

Metode yang digunakan dalam penelitian ini adalah metode penelitian kuantitatif. Adapun subjek dalam penelitian ini adalah 33 siswa yang mana sebanyak 12 siswa sebagai kelompok kontrol (8 siswa jurusan teknologi informatika dan 4 siswa jurusan matematika) dan 11 siswa sebagai kelas eksperimen (8 siswa jurusan teknologi informasi dan 3 siswa jurusan matematika). Pada akhir semester semua siswa tersebut mengikuti tes kemampuan yang meliputi, tracing algorithms, designing algorithmic solutions, detecting algorithmic errors, applying abstractions and heuristics, reformulating problems, and choosing appropriate representations. Uji statistic yang digunakan adalah uji beda rerata (yaitu uji-t) dengan tingkat signifikansi 1,33.

Hasil dari penelitian menunjukkan bahwa mengintegrasikan pemikiran komputasi dalam pembelajaran struktur diskrit dapat secara efektif dan signifikan mempengaruhi pemahaman siswa terhadap berbagai konsep



CT. CT bisa menjadi alat penting dalam memecahkan masalah matematika yang efektif dalam pemahaman konseptual matematika.

Kong, S. (2016). A framework of curriculum design for *Computational Thinking* development in K-12 education. *Journal of Computers in Education*, 3(4), 377–394. <https://doi.org/10.1007/s40692-016-0076-z>

Penelitian ini bertujuan untuk mengusulkan/ mengembangkan kerangka tujuh prinsip untuk mendukung perancangan kurikulum *Computational Thinking* dalam K-12.

Metode yang digunakan adalah desain research. Dengan menelaah hasil rancangan dalam penelitian yang dilakukan oleh Brennan pada tahun 2016, yang mana membagi *Computational Thinking* menjadi tiga bagian yaitu *Computational Thinking concept*, *Computational Thinking practice* dan *Computational Thinking perspective*.

Hasil dari pengembangan ini adalah tujuh prinsip kerangka *Computational Thinking* yaitu tiga prinsip pertama berisi tentang keterampilan dan perspektif *Computational Thinking* dalam kurikulum melalui lingkungan pemrograman yang mendorong pemerolehan pengetahuan computational thinking. Empat prinsip berikutnya merupakan strategi desain untuk mengembangkan computational thinking, menyediakan tugas komputasi yang rumit secara bertahap pada semua tingkat kurikulum untuk mengembangkan keterampilan computational thinking, meninjau setiap tingkat kurikulum dengan memproduksi sampel proyek akhir untuk memastikan cakupan pengetahuan CT yang komprehensif; rancang tugas komputasi yang menarik bagi siswa; dan menetapkan kriteria penilaian yang tepat untuk proyek akhir dan menunjukkan hasilnya untuk meningkatkan kreativitas siswa.

Rekomendasi dari penelitian ini adalah sebaiknya perlu adanya penelitian lebih lanjut terkait dengan merancang, menerapkan, dan mengevaluasi kurikulum *Computational Thinking* yang didukung oleh tujuh prinsip ini di K-12.

Bagley, S., & Rabin, J. M. (2016). Students ' Use of *Computational Thinking* in Linear Algebra. *Int. J. Res. Undergrad. Math. Ed.*, 2, 83–104. <https://doi.org/10.1007/s40753-015-0022-x>

Yang diteliti dalam penelitian ini adalah bagaimana cara berpikir siswa saat disajikan masalah aljabar linear yang baru. Tujuan dari penelitian ini adalah untuk mengeksplorasi bagaimana siswa menggunakan dan mengkoordinasikan tiga mode pemikiran, yang kita sebut komputasi, abstrak, dan geometrik, mengikuti kerangka kerja serupa yang diusulkan oleh Hillel (2000) dan Sierpinska (2000).

Metode yang digunakan dalam penelitian ini adalah metode penelitian kualitatif. Subjek dalam penelitian ini mahasiswa yang menempuh mata kuliah Aljabar Linear di Amerika.

Hasil penelitian ini adalah mahasiswa menggunakan cara berfikir komputasi dalam berbagai cara produktif dan reflektif yang mengejutkan. Makalah ini membahas strategi solusi yang digunakan siswa untuk memecahkan masalah, dengan menekankan penggunaan mode berpikir komputasi mereka.

Benkli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2017). Introducing *Computational Thinking* through hands-on projects using R with applications to calculus, probability and data analysis. *International Journal of Mathematical Education in Science and Technology*, 48(3), 393–427. <https://doi.org/10.1080/0020739X.2016.1254296>

Tujuan dari makalah ini adalah untuk mempromosikan pemikiran komputasi antara matematika, teknik, sains dan teknologi, melalui percobaan komputer langsung. Kegiatan ini memiliki potensi untuk belajar memahami, belajar dan belajar dengan teknologi, dan mereka terlibat dalam pemikiran komputasi melalui simulasi, visualisasi dan analisis data. Kami menyajikan sembilan eksperimen komputer dan menyarankan beberapa hal lagi, dengan aplikasi untuk kalkulus, probabilitas dan tata letak, yang terkait dengan pemikiran melalui simulasi, visualisasi dan analisis data. Kami menggunakan program statistik bebas (open-source). Tujuan adalah untuk memberi rasa dari apa yang ditawarkan R daripada menyajikan komprehensif tentang bahasa. Perhotelan, pengalaman komputer interaktif dapat dengan mudah diintegrasikan ke dalam kelas cerdas. Selanjutnya, kegiatan ini cenderung membuat siswa termotivasi dan secara efektif melakukan pengembangan kemampuan belajar, pemecahan masalah dan mengembangkan pemikiran orang tua untuk memahami konsep matematika matematis.

Adler, R. F., & Kim, H. (2017). Enhancing future K-8 teachers' *Computational Thinking* skills through modeling and simulations. *Educ Inf Technol*.

Tujuan dari penelitian ini adalah memodifikasi struktur metode sains yang ada untuk calon guru untuk memasukkan *Computational Thinking* melalui pemodelan dan simulasi.

Metode penelitian yang digunakan adalah metode deskriptif kualitatif, dengan subjek adalah calon guru.

Hasil penelitian adalah calon guru mengetahui materi secara efektif, merasa bahwa latihan CT akan bermanfaat dalam

pendidikan K-8., dan berencana untuk memasukkan CT ke dalam kelas masa depan mereka.

Cho, Y., & Lee, Y. (2017). Possibility of Improving *Computational Thinking Through Activity Based Learning*. *Journal of Theoretical and Applied Information Technology*, 95(18), 4385–4393.

Tujuan dari penelitian ini adalah untuk mengembangkan *Computational Thinking* pada siswa kelas dasar, karena mengingat pentingnya *Computational Thinking* bagi generasi penerus dalam mempersiapkan pendidikan di masa mendatang.

Metode yang digunakan dalam penelitian ini adalah metode terapan, dimana peneliti menerapkan penggunaan robot yang diberi nama “ladybug”. Kegiatan pembelajaran didasarkan pada permainan dengan strategi tertentu dimana robot ladybug melakukan kegiatan sehari-hari yang berkaitan dengan pembelajaran. Subjek dalam penelitian ini adalah siswa yang berumur 5-6 tahun.

Hasil penelitian menunjukkan bahwa siswa yang berpartisipasi dalam program belajar *Computational Thinking* menggunakan strategi belajar berbasis kegiatan sehari-hari mempunyai minat yang tinggi di kelas, dan kelas ini dapat membantu mengembangkan berpikir komputasi siswa.

Choi, J., Lee, Y., & Lee, E. (2017). Puzzle Based Algorithm Learning for Cultivating Computational Thinking. *Wireless Personal Communications*, 93(1), 131–145.  
<https://doi.org/10.1007/s11277-016-3679-9>

Tujuan dari penelitian ini adalah untuk menginvestigasi dampak dari pembelajaran puzzle berbasis algoritma dalam kemampuan berpikir komputasi pebelajar.

Metode yang digunakan dalam penelitian ini adalah penelitian terapan, dimana peneliti menggunakan/menerapkan suatu program yaitu puzzle yang di dalamnya ada algoritma-algoritma di dalam pembelajaran.

Hasil penelitian menunjukkan bahwa puzzle berbasis algoritma efektif untuk mengembangkan kemampuan berpikir komputasi pebelajar. Peneliti menyarankan bahwa pembelajaran dengan puzzle berbasis algoritma baik digunakan dalam mengembangkan kemampuan berpikir komputasi pebelajar.

Tsai, M., & Tsai, C. (2017). Applying online externally-facilitated regulated learning and *Computational Thinking* to improve students ' learning. *Universal Access in the Information Society*, (168). <https://doi.org/10.1007/s10209-017-0542-z>

Tujuan penelitian ini adalah untuk mengeksplorasi dampak dari pembelajaran regulasi dengan fasilitas tambahan dan berpikir komputasi dalam mengembangkan kemampuan komputasi siswa.

Metode yang digunakan dalam penelitian ini adalah metode penelitian kuantitatif. Subjek dalam penelitian ini adalah siswa dari 4 kelas semester satu.

Hasil penelitian menunjukkan bahwa kemampuan komputasi siswa dapat dikembangkan dengan kondisi secara tersimultan menerapkan pembelajaran regulasi dan berpikir komputasi. Temuan yang idperoleh adalah siswa dapat menunjukkan implikasi/dampak dari guru dan pendidik online.

Witherspoon, E. B., Higashi, R. M., & Schunn, C. D. (2017). Developing *Computational Thinking* through a Virtual Robotics

Programming Curriculum. *ACM Trans. Comput. Educ.*, 18(1), 1–20.

Tujuan dari penelitian ini adalah mengukur penilaian berpikir komputasi yang diberikan kepada siswa sekolah menengah yang berpartisipasi dalam kurikulum program robotic.

Metode yang digunakan dalam penelitian ini adalah metode penelitian kuantitatif.

Hasil penelitian menunjukkan bahwa skor pre dan post test siswa mempunyai hubungan yang signifikan. Siswa dengan program skafolding dalam kurikulum ini mendukung pengembangan pengetahuan berpikir komputasi dan kemampuan dalam mengasosiasikan dengan menumbuhkan kegiatan penyelesaian masalah pada tugas komputasi non-robotic.

Lockwood, E., & Asay, A. (2016). Algorithmic thinking : An initial characterization of *Computational Thinking* in mathematics.

Tujuan dari penelitian ini adalah menginvestigasi beberapa berpikir dalam matematika.

Metode penelitian yang digunakan adalah metode penelitian kualitatif. Subjek dalam penelitian ini adalah 5 matematikawan yang berperan di bidang komputasi dalam pekerjaannya dan topik pengembangan wawancara adalah berpikir algoritmik.

Hasil dari penelitian menunjukkan bahwa nilai matematikawan dalam berpikir, dan definisi dari berpikir algoritmik dan beberapa aplikasi yang menggambarkan berpikir tersebut.

Swaid, S. I. (2015). Bringing *Computational Thinking* to STEM education. *Procedia Manufacturing*, 3(Ahfe), 3657–3662. <https://doi.org/10.1016/j.promfg.2015.07.761>

Artikel ini melaporkan bahwa sebuah project bernama HBCU-UP II yang bekerja membawa berpikir komputasi ke dalam bidang ilmu STEM.

Metode penelitian yang digunakan adalah desain research.

Hasil dari penelitian ini adalah kerangka kerja, implementasi dan luaran. Project ini berkontribusi dalam upaya membangun berpikir komputasi sebagai sikap yang dapat diterapkan secara umum yang menggunakan pembelajaran STEM, pendidikan, dan kurikulum.

Shute, V. J., Sun, C., & Asbell-clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>

Tujuan dari penelitian ini adalah untuk melihat/mereview pertumbuhan dari berpikir komputasi dalam pendidikan.

Metode yang digunakan dalam penelitian ini adalah kajian literature.

Hasil penelitian menunjukkan bahwa kategori berpikir komputasi dibagi menjadi 6 komponen utama, yaitu dekomposisi, abstraksi, desain algoritma, debugging, iterasi, dan generalisasi.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Computers & Education Supporting all learners in school-wide computational thinking : A cross-case qualitative analysis. *Computers & Education*, 82, 263–279.

Tujuan dari penelitian ini adalah untuk menginvestigasi bagaimana guru sekolah dasar dengan pengalaman pengetahuan computer yang terbatas dalam kebutuhan tinggi dari berpikir komputasi dalam pembelajarannya.

Metode yang digunakan dalam penelitian ini adalah analisis kasus silang (cross-case analysis).

Hasil penelitian menunjukkan bahwa memasukkan berpikir komputasi ke dalam pembelajaran merupakan suatu tantangan dan keuntungan untuk mengembangkan teknologi baru.





# DAFTAR PUSTAKA

- Angeli, C. V. (2016). A K-6 *Computational Thinking* Curriculum Framework: Implications for Teacher Knowledge STEM+C *Computational Thinking* workshop View project E-QUAL View project. Diambil kembali dari <https://www.researchgate.net/publication/30>
- Back, J. (2019, Desember 9). *Integrating Computational Thinking into Math Classes*. Diambil kembali dari [www.gettingsmart.com](http://www.gettingsmart.com): <https://www.gettingsmart.com/2019/12/integrating-computational-thinking-into-math-classes/>
- Barcelos, T. &. (2018). Mathematics Learning through *Computational Thinking* Activities: A Systematic Literature Review . *JOURNAL OF UNIVERSAL COMPUTER SCIENCE*, 24. 815.
- Barcelos, T. S., Munoz, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through *Computational Thinking* Activities: A Systematic Literature Review. *Journal of Universal Computer Science*, 815-845. Diambil kembali dari [http://jucs.org/jucs\\_24\\_7/mathematics\\_learning\\_through\\_computational/jucs\\_24\\_07\\_0815\\_0845\\_barcelos.pdf](http://jucs.org/jucs_24_7/mathematics_learning_through_computational/jucs_24_07_0815_0845_barcelos.pdf)
- Cendros, R. (2018, Juli 5). *Why CT in Math Education?* Diambil kembali dari [ctmath.ca](http://ctmath.ca): <http://ctmath.ca/uncategorized/why-ct-in-math-education/>
- DeJong, G. (2004). Explanation-based learning. . *Computer Science Handbook, Second Edition*, 49(3), 68-1-68-18. <https://doi.org/10.1201/b16812-43>.
- Denning, P. J. (1942). *Computational Thinking*. Cambridge.
- Gadanidis, G. C. (2017). *Computational Thinking* in mathematics teacher education. *Contemporary Issues in Technology and Teacher Education*, 17(4). Diambil kembali dari

<https://citejournal.org/volume-17/issue-4-17/mathematics/computational-thinking-in-mathematics-teacher-education/>

- Gadanidis, G. H. (2016). Berpikir Komputasi, Siswa Kelas 1 dan Teorema Binomial. Pengalaman Digital dalam Pendidikan Matematika. doi 10.1007 / s40751-016-0019-3.
- Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). *Computational Thinking in Mathematics Teacher Education. Contemporary Issues in Technology and Teacher Education*, 17(4). Diambil kembali dari <https://citejournal.org/volume-17/issue-4-17/mathematics/computational-thinking-in-mathematics-teacher-education>
- Guzdial, M. (2008). Education Paving the Way for Computational Thinking. Diambil kembali dari <https://doi.org/10.1145/1378704.1378713>
- Lee, C.-S. d.-Y. (2019, Mei 3). *Mathematics Learning: Perceptions Toward the Design of a Website Based on a Fun Computational Thinking-Based Knowledge Management Framework*. Diambil kembali dari [link.springer.com](http://link.springer.com): [https://link.springer.com/chapter/10.1007/978-981-13-6528-7\\_11](https://link.springer.com/chapter/10.1007/978-981-13-6528-7_11)
- Namukasa, I. P. (2017). Tools for Integrating *Computational Thinking* and Mathematics in the Middle Grades.
- Papert, S. (1980). *Mindstorms: Anak-anak, komputer, dan ide-ide kuat*. . New York: NY: Basic Books, Inc.
- Tedre, M. &. (2016). The Long Quest for Computational Thinking. The Long Quest for Computational Thinking. 120–129. Diambil kembali dari <https://doi.org/10.1145/2999541.2999542>
- Voogt, J. F. (2015). *Computational Thinking* in compulsory education: Towards an agenda for research and practice. Education and

Information Technologies. 20(4), 715–728.  
doi:<https://doi.org/10.1007/s10639-015-9412-6>

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining *Computational Thinking* for Mathematics and Science. *J Sci Educ Technol*.

Wing, J. (2006). Berpikir komputasi. *Komunikasi ACM*, 49 (3), 33-36.

Wing, J. M. (2008). *Computational Thinking* and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 366(1881), 3717–3725. . doi:<https://doi.org/10.1098/rsta.2008.0118>

Yadav, A. &. (2011). Introducing *Computational Thinking* in Education Courses. 465-470. 10.1145/1953163.1953297.

Adler, R. F., & Kim, H. (2017). Enhancing future K-8 teachers ' *Computational Thinking* skills through modeling and simulations. *Journal Education and Information Technologies*, 23(4), 1501–1514.

Allsop, Y. (2019). Assessing *Computational Thinking* process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30–55.  
<https://doi.org/10.1016/j.ijcci.2018.10.004>

Ambrosio, A. P., Almeida, L. da S., Macedo, J., & Franco, A. (2014). Keywords: POP-II.A. novices, POP-V.A. cognitive theories, POP-VI.F. exploratory. *PPIG*, 1–10.

Anderson, N. D. (2016). A Call for *Computational Thinking* in Undergraduate Psychology. *Psychology Learning and Teaching*, 15(3), 226–234. <https://doi.org/10.1177/1475725716659252>

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 *Computational Thinking* Curriculum Framework: Implications for Teacher Knowledge. *Educational Technology & Society*, 19(3), 47–57.

Armoni, M. (2013). On Teaching Abstraction in Computer Science to

- Novices. *Journal of Computers in Mathematics and Science Teaching*, 32(3), 265–284.
- Astrachan, O., & Briggs, A. (2012). The CS principles project: A new introductory computing course for everyone. *ACM Inroads*, 3(2), 38–42. <https://doi.org/10.1145/2189835.2189849>
- Atmatzidou, S., & Demetriadis, S. (2014). How to Support Students' Computational Thinking Skills in Educational Robotics Activities. *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, 43–50.
- Bagley, S., & Rabin, J. M. (2016). Students' Use of Computational Thinking in Linear Algebra. *International Journal of Research in Undergraduate Mathematics Education*, 2(1), 83–104. <https://doi.org/10.1007/s40753-015-0022-x>
- Barcelos, T. S., & Silveira, I. F. (2016). Computational Thinking and mathematics: Possible relationships revealed by an analysis of national curriculum guidelines. *Leadership and Personnel Management: Concepts, Methodologies, Tools, and Applications*, 2(1994), 853–865. <https://doi.org/10.4018/978-1-4666-9624-2.ch037>
- Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning and Leading with Technology*, 38(6), 20–23.
- Belanger, C., Christenson, H., & Lopac, K. (2018). *Confidence and Common Challenges: The Effects of Teaching Computational Thinking to Students Ages 10-16*.
- Bell, T., Andreae, P., & Robins, A. (2012). Computer science in NZ high schools: The first year of the new standards. *SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 343–348. <https://doi.org/10.1145/2157136.2157240>

- Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2017). Introducing *Computational Thinking* through hands-on projects using R with applications to calculus, probability and data analysis. *International Journal of Mathematical Education in Science and Technology*, 48(3), 393–427. <https://doi.org/10.1080/0020739X.2016.1254296>
- Bengtsson, D. (2003). The Distributed Nature of Pattern Generalization. *PNA*, 9(3), 165–191.
- Bermejo, P., Bermúdez, A., Carrión, C., Díaz, G., Fernández, G., & García, I. (2017). An Empirical Study on the Integration of *Computational Thinking* Principles in Primary and Secondary Schools of Castilla-La Mancha. *EDULEARN17 Proceedings*, 1, 1788–1797. <https://doi.org/10.21125/edulearn.2017.1380>
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). *Computational Thinking* and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, 72, 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Bilbao, J., García, O., Rebollar, C., Bravo, E., & Varela, C. (2016). Skills, attitudes and concepts of the *Computational Thinking* essential tool for doing scientific. *Economics and Education*, 82–87.
- Bower, M., Wood, L. N., Howe, C., & Lister, R. (2017). Improving the *Computational Thinking* Pedagogical Capabilities of School Teachers. *Australian Journal of Teacher Education*, 42(3), 53–72.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *AERA*, 1–25.
- Brinda, T., Puhlmann, H., & Schulte, C. (2009). Bridging ICT and CS - Educational standards for computer science in lower secondary education. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, (July), 288–292. <https://doi.org/10.1145/1562877.1562965>

- Calderón, J. F., & Ebers, J. (2017). Problock : a tool for *Computational Thinking* development using problem-based learning. *2017 36th International Conference of the Chilean Computer Science Society (SCCC)*, 1–5. IEEE.
- Callejo, M. L., & Zapatera, A. (2017). Prospective primary teachers' noticing of students' understanding of pattern generalization. *Journal of Mathematics Teacher Education*, 20(4), 309–333. <https://doi.org/10.1007/s10857-016-9343-1>
- Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *Journal of Mathematical Behavior*, 47(November 2016), 70–80. <https://doi.org/10.1016/j.jmathb.2017.06.004>
- Cho, Y., & Lee, Y. (2017). Possibility of Improving *Computational Thinking* Through Activity Based Learning. *Journal of Theoretical and Applied Information Technology*, 95(18), 4385–4393.
- Choi, J., Lee, Y., & Lee, E. (2017). Puzzle Based Algorithm Learning for Cultivating Computational Thinking. *Wireless Personal Communications*, 93(1), 131–145. <https://doi.org/10.1007/s11277-016-3679-9>
- Corradini, I., Lodi, M., & Nardelli, E. (2017). Conceptions and Misconceptions about *Computational Thinking* among Italian Primary School Teachers. *Proceedings of the 2017 ACM Conference on International Computing Education Research - ICER '17*, 136–144. <https://doi.org/10.1145/3105726.3106194>
- Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing *Computational Thinking* in the classroom: a framework. In *Computing at School*.
- Czerkawski, B. C., & Lyman, E. W. (2015). Exploring Issues About *Computational Thinking* in Higher Education. *TechTrends*, 59(2), 57–65. <https://doi.org/10.1007/s11528-015-0840-3>
- Dagiene, V., & Stupuriene, G. (2016). Informatics Concepts and *Computational Thinking* in K-12 Education: A Lithuanian

- Perspective. *Journal of Information Processing*, 24(4), 732–739.  
<https://doi.org/10.2197/ipsjjip.24.732>
- Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7), 13–18.  
<https://doi.org/10.1145/1272516.1272529>
- Denning, P. J. (2010). The great principles of computing. *American Scientist*, 98(5), 369–372.  
<https://doi.org/10.1515/9781400839544.82>
- diSessa, A. A. (2018). Computational Literacy and “The Big Picture” Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, 20(1), 3–31.  
<https://doi.org/10.1080/10986065.2018.1403544>
- Djambong, T., & Freiman, V. (2016). Task-based assessment of students’ *Computational Thinking* skills developed through visual programming or tangible coding environments. *Proceedings of the 13th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2016*, (Celda), 41–51.
- Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between *Computational Thinking* skills and academic performance. *Journal of Computers in Education*, 4(4), 355–369.  
<https://doi.org/10.1007/s40692-017-0090-9>
- Duncan, C., Bell, T., & Atlas, J. (2017). What do the teachers think? Introducing *Computational Thinking* in the primary school curriculum. *Proceedings of the Nineteenth Australasian Computing Education Conference on - ACE '17*, 65–74.  
<https://doi.org/10.1145/3013499.3013506>
- Durak, H. Y., & Saritepeci, M. (2017). Analysis of the relation between *Computational Thinking* skills and various variables with the structural equation model. *Computers & Education*, 116, 191–202.  
<https://doi.org/10.1016/j.compedu.2017.09.004>



- Eisenberg, M. (2010). Bead games, or, getting started in *Computational Thinking* without a computer. *International Journal of Computers for Mathematical Learning*, 15(2), 161-166. <https://doi.org/10.1007/s10758-010-9167-5>
- El Mouhayar, R. (2018). Trends of progression of student level of reasoning and generalization in numerical and figural reasoning approaches in pattern generalization. *Educational Studies in Mathematics*, 99(1), 89-107. <https://doi.org/10.1007/s10649-018-9821-8>
- English, L. (2018). On MTL 's Second Milestone: Exploring *Computational Thinking* and Mathematics Learning. *Mathematical Thinking and Learning*, 20(1), 1-2. <https://doi.org/10.1080/10986065.2018.1405615>
- Faber, H. H., Wierdsma, M. D. M., Doornbos, R. P., Van Der Ven, J. S., & De Vette, K. (2017). Teaching *Computational Thinking* to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network*, 12, 13-24.
- Folk, R., Lee, G., Michalenko, A., Peel, A., & Pontelli, E. (2015). GK-12 DISSECT: Incorporating *Computational Thinking* with K-12 science without computer access. *Proceedings - Frontiers in Education Conference, FIE, 2014*. <https://doi.org/10.1109/FIE.2015.7344238>
- Gadanidis, G. (2017a). Artificial intelligence, *Computational Thinking*, and mathematics education. *The International Journal of Information and Learning Technology*, 34(2), 133-139. <https://doi.org/10.1108/IJILT-09-2016-0048>
- Gadanidis, G. (2017b). Five Affordances of *Computational Thinking* to support Elementary Mathematics Education. *Jl. of Computers in Mathematics and Science Teaching*, 36(2), 143-151.
- Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). *Computational Thinking* in Mathematics Teacher Education. *Contemporary Issues in Technology and Teacher Education*, 17(4), 458-477. [https://doi.org/10.1007/978-3-319-52691-1\\_13](https://doi.org/10.1007/978-3-319-52691-1_13)

- Gadanidis, G., Hughes, J. M., Minniti, L., & White, B. J. G. (2017). Computational Thinking, Grade 1 Students and the Binomial Theorem. *Digital Experiences in Mathematics Education*, 3(2), 77–96. <https://doi.org/10.1007/s40751-016-0019-3>
- García-Peñalvo, F. J. (2012). Computational Thinking. *IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje (IEEE RITA)*, 13(1), 17–19. <https://doi.org/https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25. <https://doi.org/10.1145/1378704.1378713>
- Hartnett, J. (2015). Teaching Computation in Primary School without Traditional Written Algorithms. *Proceedings of the 38th Annual Conference of the Mathematics Education Research Group of Australasia*, 285–292.
- Hashemi, N., Mohd Salleh Abu, Kashefi, H., & Rahimi, K. (2013). Generalization in the Learning of Mathematics. *2nd International Seminar on Quality and Affordable Education (ISQAE 2013)*, (Isqae), 208–215.
- Henrion, M., Fischer, G. W., & Mullin, T. (1993). Divide and conquer? effects of decomposition on the accuracy and calibration of subjective probability distributions. *Organizational Behavior and Human Decision Processes*, 55(2), 207–227. <https://doi.org/10.1006/obhd.1993.1031>
- Higgins, H. J., & Wiest, L. R. (2006). Individual Interviews as Insight into Children’s Computational Thinking. *Australian Primary Mathematics Classroom*, 11(1), 25–29.
- Howland, K., & Good, J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers and Education*, 80, 224–240. <https://doi.org/10.1016/j.compedu.2014.08.014>

- Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Hu, C. (2011). *Computational Thinking – What It Might Mean and What We Might Do about It*. *ITiCSE '11: Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 223–227. <https://doi.org/10.1145/1999747.1999811>
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015b). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education*, 82, 263–279. <https://doi.org/10.1016/j.compedu.2014.11.022>
- Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. (2015). Empowering K – 12 Students With Disabilities to Learn *Computational Thinking* and Computer Programming. *TEACHING Exceptional Children*, 48(1), 45–53. <https://doi.org/ISSN:0040-0599>
- Iyer, S. (2019). *Computational Thinking Education*. In *Computational Thinking Education*. <https://doi.org/10.1007/978-981-13-6528-7>
- Jacob, S. R., & Warschauer, M. (2018). *Computational Thinking and Literacy*. *Journal of Computer Science Integration*, 1(1). <https://doi.org/10.26716/jcsi.2018.01.1.1>
- Jamil, H. M. (2017). Visual *Computational Thinking* Using Patch. *ICWL*, 10473, 208–214. <https://doi.org/10.1007/978-3-319-66733-1>
- Juškevičiene, A., & Dagiene, V. (2018). *Computational Thinking relationship with digital competence*. *Informatics in Education*, 17(2), 265–284. <https://doi.org/10.15388/infedu.2018.14>
- Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational What? Relating *Computational Thinking* to Teaching. *TechTrends*, 62(6), 574–584. <https://doi.org/10.1007/s11528-018-0290-9>

- Kong, S. (2016). A framework of curriculum design for *Computational Thinking* development in K-12 education. *Journal of Computers in Education*, 3(4), 377–394. <https://doi.org/10.1007/s40692-016-0076-z>
- Koning, J. L., Faber, H. H., & Wierdsma, M. D. M. (2011). Introducing *Computational Thinking* to 5 and 6 year old students in Dutch primary schools; an educational design research study. *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, 189–190. <https://doi.org/10.1088/1751-8113/44/8/085201>
- Korkmaz, O., Cakir, R., & Ozden, M. Y. (2017). Computers in Human Behavior A validity and reliability study of the *Computational Thinking* scales ( CTS ). *Computers in Human Behavior*, 72, 558–569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Korucu, A. T. (2017). Examination of the *Computational Thinking* Skills of Students. *Journal of Learning and Teaching in Digital Age*, 2(1), 11–19. Retrieved from <http://joltida.org/index.php/joltida/article/view/25>
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, 3(2), 154–171. <https://doi.org/10.1007/s40751-017-0031-2>
- Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 36–42. <https://doi.org/10.1145/1232743.1232745>
- Krauss, J., & Prottzman, K. (2017). *Computational Thinking and Coding for Every Student: The Teacher's Getting-Started Guide Teachers*. United States of America: SAGE.
- Kynigos, C. (2007). Using half-baked microworlds to challenge teacher educators' knowing. *International Journal of Computers for Mathematical Learning*, 12(2), 87–111.

<https://doi.org/10.1007/s10758-007-9114-2>

- Lavigne, H., Orr, J., & Wolsky, M. (2018). Exploring *Computational Thinking* in Preschool Math Learning Environments. *Society for Information Technology & Teacher Education International Conference*, 38–43.
- Lee, T. Y., Mauriello, M. L., Ahn, J., & Bederson, B. B. (2014). CTArcade: *Computational Thinking* with games in school age children. *International Journal of Child-Computer Interaction*, 2(1), 26–33. <https://doi.org/10.1016/j.ijcci.2014.06.003>
- Leonard, J., Mitchell, M., Barnes-johnson, J., Unertl, A., Outka-hill, J., Robinson, R., & Hester-croff, C. (2017). Preparing Teachers to Engage Rural Students in *Computational Thinking* Through Robotics, Game Design, and Culturally Responsive Teaching. *Journal of Teacher Education*, 69(4), 386–407. <https://doi.org/10.1177/0022487117732317>
- Lewandowski, G., Bouvier, D. J., McCartney, R., Sanders, K., & Simon, B. (2007). Commonsense computing (episode 3): Concurrency and concert tickets. *Third International Computing Education Research Workshop, ICER'07*, (episode 3), 133–144. <https://doi.org/10.1145/1288580.1288598>
- Liu, H. P., Perera, S. M., & Klein, J. W. (2017). Using Model-Based Learning to Promote *Computational Thinking* Education. In *Emerging Research, Practice, and Policy on Computational Thinking, Educational Communications and Technology: Issues and Innovations* (pp. 153–172). <https://doi.org/10.1007/978-3-319-52691-1>
- LLiu, Y., Ma, Z., & Qian, Y. (2019). Developing Chinese Elementary School Students' Computational Thinking: A Convergent Cognition Perspective. *CompEd 2019 - Proceedings of the ACM Conference on Global Computing Education*, 42, 238. <https://doi.org/10.1145/3300115.3312514>
- Lockwood, E., & Asay, A. (2016). Algorithmic thinking : An initial characterization of *Computational Thinking* in mathematics. *The*

- Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (38th, Tucson, AZ, Nov 3-6, 2016), (November).*
- Lu, J. J., & Fletcher, G. H. L. (2009). Thinking About Computational Thinking Categories and Subject Descriptors. *Sigcse*, 260–264. <https://doi.org/10.1145/1539024.1508959>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of Computational Thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Maharani, S., Kholid, M. N., Pradana, L. N., & Nusantara, T. (2019). Problem Solving in the Context of Computational Thinking. *Infinity Journal of Mathematics Education*, 8(2), 109–116.
- Maharani, S., Nusantara, T., As'ari, A. R., & Qohar, A. (2019). How The Students Computational Thinking Ability On Algebraic? *International Journal of Scientific & Technology Research*, 8(9), 419–423.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference - ITiCSE-WGR '14*, (23), 1–29. <https://doi.org/10.1145/2713609.2713610>
- Masfingatin, T., & Maharani, S. (2019). Computational Thinking : Students On Proving Geometry Theorem. *International Journal of Scientific & Technology Research*, 8(09), 2216–2223.
- Matsumoto, P. S., & Cao, J. (2017). The Development of Computational Thinking in a High School Chemistry Course. *Journal of Chemical Education*, 94(9), 1217–1224. <https://doi.org/10.1021/acs.jchemed.6b00973>
- Melhuish, K., Thanheiser, E., & Guyot, L. (2018). Elementary school

- teachers' noticing of essential mathematical reasoning forms: justification and generalization. In *Journal of Mathematics Teacher Education*. <https://doi.org/10.1007/s10857-018-9408-4>
- Mgova, Z. (2018). *Computational Thinking Skills in Education Curriculum*. Retrieved from [http://epublications.uef.fi/pub/urn\\_nbn\\_fi\\_uef-20180343/urn\\_nbn\\_fi\\_uef-20180343.pdf](http://epublications.uef.fi/pub/urn_nbn_fi_uef-20180343/urn_nbn_fi_uef-20180343.pdf)
- Morreale, P., Goski, C., & Jimenez, L. (2012). Measuring the Impact of *Computational Thinking Workshops in High School Teachers*. *Journal of Computing Sciences in Colleges*, 27(6), 151–157.
- Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to develop *Computational Thinking* in early Childhood Education. *Media Education Research Journal*, 27(59), 63–72. <https://doi.org/10.3916/C59-2019-06>
- Niemel, P., Partanen, T., Harsu, M., Leppänen, L., & Ihantola, P. (2017). *Computational Thinking* as an emergent learning trajectory of mathematics. *ACM International Conference Proceeding Series*, (September 2018), 70–79. <https://doi.org/10.1145/3141880.3141885>
- Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing *Computational Thinking* into High School Mathematics and Science Classrooms. *Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) 2016, Volume 2*, 705–712.
- Palts, T., & Pedaste, M. (2017a). Tasks for assessing skills of computational thinking. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, Part F1286(372)*, 367. <https://doi.org/10.1145/3059009.3072999>
- Palts, T., & Pedaste, M. (2017b). Tasks for Assessing Skills of Computational Thinking. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '17*, (November), 367–367. <https://doi.org/10.1145/3059009.3072999>

- Peel, A., & Friedrichsen, P. (2018). Algorithms, Abstractions, and Iterations: Teaching *Computational Thinking* Using Protein Synthesis Translation Computational Thinking : Algorithms CT is an essential skill for navigating today ' s complex technological world . *The American Biology Teacher*, 80(1), 21–28.
- Pei, C. (Yu), Weintrop, D., & Wilensky, U. (2018). Cultivating *Computational Thinking* Practices and Mathematical Habits of Mind in Lattice Land. *Mathematical Thinking and Learning*, 20(1), 75–89. <https://doi.org/10.1080/10986065.2018.1403543>
- Peters-Burton, E. E., Cleary, T. J., & Kitsantas, A. (2015). The development of *Computational Thinking* in the context of science and engineering practices: A self-regulated learning approach. *Proceedings of the 12th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2015*, (Celda), 257–261.
- Pollock, L., Pan, Y., Yilmaz-ozden, S., & Antonio, S. (2017). *Infusing Computational Thinking in Teacher Preparation : Examining Preservice Teacher*.
- Portelance, D. J., & Bers, M. U. (2015). Code and tell: Assessing Young Children's Learning of *Computational Thinking* Using Peer Video Interviews with ScratchJr. *Proceedings of the 14th International Conference on Interaction Design and Children*, 271–274. <https://doi.org/10.1145/2771839.2771894>
- Promraksa, S., Sangaroon, K., & Inprasitha, M. (2014). Characteristics of *Computational Thinking* about the Estimation of the Students in Mathematics Classroom Applying Lesson Study and Open Approach. *Journal of Education and Learning*, 3(3), 56–66. <https://doi.org/10.5539/jel.v3n3p56>
- Psycharis, S. (2016). Inquiry Based-Computational Experiment, Acquisition of Threshold Concepts and Argumentation in Science and Mathematics Education. *Educational Technology & Society*, 19(3), 282–293.



- Radford, L., Socas, M., Zazkis, R., & Liljedahl, P. (2011). Generalization of Patterns: The tension between algebraic thinking and algebraic notation. *Proceedings of the 35th Conference of the International Group for the Psychology of Mathematics Education*, 1(2007), 379–402. <https://doi.org/10.1023/A:1020291317178>
- Rambally, G. (2017). Integrating *Computational Thinking* in Discrete Structures. In *Emerging Research, Practice, and Policy on Computational Thinking, Educational Communications and Technology: Issues and Innovations* (pp. 99–119). <https://doi.org/10.1007/978-3-319-52691-1>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the *Computational Thinking Test*. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Sagastizábal, C. (2012). Divide to conquer: Decomposition methods for energy optimization. *Mathematical Programming*, 134(1), 187–222. <https://doi.org/10.1007/s10107-012-0570-7>
- Sanford, J. F., & Naidu, J. T. (2016). *Computational Thinking Concepts for Grade School*. *Contemporary Issues in Education Research*, 9(1), 23–32.
- Sanford, J. F., & Naidu, J. T. (2017). Mathematical Modeling And Computational Thinking. *Contemporary Issues in Education Research-Second Quarter 2017*, 10(2), 159–168.
- Selby, C. C., & Woollard, J. (2013). Refining an Understanding of Computational Thinking. *Special Interest Group on Computer Science Education (SIGCSE)*, 1–23.

- Selby, C., & Woollard, J. (2013). Computational Thinking: The Developing Definition. *ITiCSE Conference 2013*, 5–8.
- Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a Phenomenology of Computational Thinking in K-12 STEM. In *Computational Thinking in STEM Discipline: Foundations and Research Highlights*. Springer.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating Computational Thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351–380. <https://doi.org/10.1007/s10639-012-9240-x>
- Sentance, S., & Csizmadia, A. (2015). Teachers ' perspectives on successful strategies for teaching Computing in school. *IFIP TCS*, (June), 1–11. Retrieved from <http://community.computingschool.org.uk/files/6303/original.pdf>
- Serafini, G. (2011). Teaching programming at primary schools: Visions, experiences, and long-term research prospects. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7013 LNCS, 143–154. [https://doi.org/10.1007/978-3-642-24722-4\\_13](https://doi.org/10.1007/978-3-642-24722-4_13)
- Shute, V. J., Sun, C., & Asbell-clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Son, J., & Lee, J. (2016). Pre- service Teachers ' Understanding of Fraction Multiplication, Representational Knowledge, and Computational Skills. *Mathematics Teacher Education and Development*, 18(2), 5–28.
- Sung, W., Ahn, J., & Black, J. (2017a). The Design of Embodied Activities Promoting Computational Thinking and Mathematics Learning in Early-childhood ... *The Annual Conference of the American Educational Research Association*, (April).

- Sung, W., Ahn, J., & Black, J. B. (2017b). Introducing *Computational Thinking* to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. *Technology, Knowledge and Learning*, 22(3), 1–21. <https://doi.org/10.1007/s10758-017-9328-x>
- Swaid, S. I. (2015). Bringing *Computational Thinking* to STEM education. *Procedia Manufacturing*, 3(Ahfe), 3657–3662. <https://doi.org/10.1016/j.promfg.2015.07.761>
- Swanson, S. R., & Re-, T. (2013). Work in Progress: Integrating *Computational Thinking* in STEM Education through a Project-based Learning Approach Work-in-Progress: Integrating *Computational Thinking* in STEM Education through a Project-based Learning Approach. *American Society for Engineering Education, Proceedings of ASEE Annual Conference*.
- Tsai, M., & Tsai, C. (2017). Applying online externally-facilitated regulated learning and *Computational Thinking* to improve students ' learning. *Universal Access in the Information Society*, 16(8). <https://doi.org/10.1007/s10209-017-0542-z>
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). *Computational Thinking* in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. <https://doi.org/10.1007/s10639-015-9412-6>
- Voskoglou, M. G. (2013). Problem Solving, Fuzzy Logic and Computational Thinking. *Egyptian Computer Science Journal ECS*, 37(1), 131–145.
- Weintrop, D. (2018). Defining, Designing, and Documenting *Computational Thinking* Across K-12 Education. *ICLS 2018, (July)*, 2008–2010. Retrieved from [https://drive.google.com/open?id=1\\_ZNTSsxlyQuk5e-GT\\_rzk7jk-MUKka77](https://drive.google.com/open?id=1_ZNTSsxlyQuk5e-GT_rzk7jk-MUKka77)
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., &

- Wilensky, U. (2015). Defining *Computational Thinking* for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. <https://doi.org/10.1007/s10956-015-9581-5>
- Werner, L., Denner, J., & Campe, S. (2014). Children programming games: A strategy for measuring computational learning. *ACM Transactions on Computing Education*, 14(4). <https://doi.org/10.1145/2677091>
- Wilkerson, M. H., & Fenwick, M. (2016). Using mathematics and computational thinking. *Helping Students Make Sense of the World Using Next Generation Science and Engineering Practices*, 181-204.
- Wilson, C., & Guzdial, M. (2010). How to make progress in computing education. *Communications of the ACM*, 53(5), 35. <https://doi.org/10.1145/1735223.1735235>
- Wing, J. M. (2006). Computational Thinking. *Communication of The ACM*, 49(3), 33-35.
- Wing, J. M. (2008). *Computational Thinking* and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wing, J. M. (2010). Computational Thinking: What and Why? *The Link - The Magazine of the Varnegie Mellon University School of Computer Science*, (March 2006), 1-6. Retrieved from <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Wing, J. M. (2011). Computational thinking. *Communications of the ACM*, 3-3. <https://doi.org/10.1109/vlcc.2011.6070404>
- Witherspoon, E. B., Higashi, R. M., & Schunn, C. D. (2017). Developing *Computational Thinking* through a Virtual Robotics Programming Curriculum. *ACM Trans. Comput. Educ.*, 18(1), 1-20.
- Yadav, A., Gretter, S., Good, J., & Mclean, T. (2017). *Computational*

- Thinking in Teacher Education*. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 205–220). <https://doi.org/10.1007/978-3-319-52691-1>
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). *Computational Thinking in elementary and secondary teacher education*. *ACM Transactions on Computing Education*, 14(1). <https://doi.org/10.1145/2576872>
- Yadav, A., Stephenson, C., & Hong, H. (2017). *Computational Thinking for teacher education*. *Communications of the ACM*, 60(4), 55–62. <https://doi.org/10.1145/2994591>
- Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). *Introducing Computational Thinking in education courses*. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*, 465–470. <https://doi.org/10.1145/1953163.1953297>
- Yağcı, M. (2019). A valid and reliable tool for examining *Computational Thinking* skills. *Education and Information Technologies*, 24(1), 929–951. <https://doi.org/10.1007/s10639-018-9801-8>
- Yasar, O., Maliekal, J., Veronesi, P., & Little, L. (2017). The essence of *Computational Thinking* and tools to promote it. *American Society for Engineering Education*, (September). Retrieved from <https://www.asee.org/public/conferences/78/papers/17618/view>
- Yinnan, Z., & Chaosheng, L. (2012). Training for *Computational Thinking* Capability on Programming Language Teaching. *The 7th International Conference on Computer Science & Education*, (Iccse), 1804–1809.
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking. *Journal of Educational Computing Research*, 53(4), 562–590. <https://doi.org/10.1177/0735633115608444>

# TENTANG PENULIS



**Swasti Maharani**, perempuan yang lahir di Ngawi pada 10 Juni 1989 ini berprofesi sebagai salah satu dosen di Universitas PGRI Madiun. Menyelesaikan program sarjana bidang pendidikan matematika di IKIP PGRI Madiun tahun 2011. Program magister bidang pendidikan matematika di Universitas Sebelas Maret Surakarta tahun 2013, dan sekarang sedang menempuh program doktoral bidang pendidikan matematika di Universitas Negeri Malang.

Kiprahnya di dunia publikasi ilmiah dimulai dari tahun 2015. Menjadi editor dan reviewer di beberapa jurnal matematika dan pendidikan matematika di Indonesia. Buku-buku yang telah ditulis diantaranya “Analisis numerik berbasis group investigation untuk meningkatkan kemampuan berpikir kritis”, “Implementasi *Digital Objek Identifier* (DOI) pada Website OJS”, “Mengembangkan HOTS (*High Order Thinking Skills*) Melalui Matematika”, “Optimalisasi Layanan Crossref pada Jurnal Ilmiah”, dan “Ragam Soal Matematis Untuk Mengembangkan Disposisi Berpikir Kritis”.

Dapat dihubungi melalui e-mail [swasti.mathedu@unipma.ac.id](mailto:swasti.mathedu@unipma.ac.id)



**Toto Nusantara**, Laki-laki kelahiran Malang 30 Nopember 1967 saat ini berprofesi sebagai dosen Jurusan Matematika FMIPA Universitas Negeri Malang. Gelar doktor dalam bidang MIPA diraih di ITB Tahun 2001, dengan konsentrasi pada matematika terapan. Sedangkan gelar magister matematika, konsentrasi analisis ditempuh tahun 1992 dan selesai tahun 1994. Latar belakang pendidikan diperoleh saat kuliah S1 Pendidikan Matematika di IKIP Malang tahun 1986 - 1990. Dunia pendidikan terasah berkat bimbingan S1 Pendidikan Matematika yang ditugaskan penulis sejak diangkat jadi dosen di Jurusan

Matematika tahun 1990, sedangkan membimbing S2 Pendidikan Matematika sejak lulus Magister tahun 1994, serta membimbing S3 Pendidikan Matematika sejak berdiri tahun 2009. Mandat kewenangan sebagai promotor tahun 2008.

Buku-buku yang telah ditulis diantaranya adalah “Sistem Dinamik Linear” diterbitkan tahun 2012. Buku “Peramalan Artificial Neural Network Berbasis GUI & NNTTool Matlab” diterbitkan tahun 2019, merupakan produk penelitian Kerjasama dengan mitra peneliti dari Universitas Muhammadiyah Lombok. Buku “Metode tak Baku” sedang proses persiapan untuk terbit. Selain, publikasi dalam bentuk buku, puluhan artikel dari publikasi bereputasi telah dihasilkan oleh penulis dan tim. Detil lengkap dapat dilihat di biografi <https://scholar.google.co.id/citations?user=XShTjHUAAAAJ&hl=id> ataupun

<https://www.scopus.com/authid/detail.uri?authorId=55337998100> .

Korespondensi kepada penulis dapat menghubungi e-mail [toto.nusantara.fimipa@um.ac.id](mailto:toto.nusantara.fimipa@um.ac.id)



**Abdur Rahman As'ari** adalah nama lelaki yang dilahirkan di suatu desa kecil, tepatnya di desa Sumber Kembar, kecamatan Pakuniran, Kabupaten Probolinggo. Dilahirkan pada tanggal 01 bulan Maret tahun 1962, As'ari (begitu nama panggilannya) bersekolah di sekolah-sekolah negeri. Pendidikan Dasar ditempuhnya di Kabupaten Probolinggo sendiri, yaitu di SDN 1 Bucorkulon kecamatan Pakuniran, dan SMPN 1 Kraksaan. Pendidikan selanjutnya ditempuh di luar kabupaten Probolinggo.

Pendidikan jenjang menengah ditempuhnya di SMA PPSP IKIP MALANG (hanya dalam waktu 2 tahun meskipun belum ada program akselerasi resmi dari pemerintah). Pendidikan tinggi ditempuh sebagian besar di Malang, yaitu S1 Pendidikan Matematika IKIP MALANG, S2 Pendidikan Matematika IKIP MALANG, dan S3 Teknologi Pembelajaran di Universitas Negeri Malang. As'ari pernah juga kuliah dan lulus dengan gelar M.A. di jurusan Early and Middle Childhood Education di The Ohio State University, USA.

Kiprahnya di bidang pendidikan dimulai ketika masih berstatus sebagai mahasiswa jenjang S1. Pada tahun 1981, As'ari sudah diminta membantu menjadi pengajar di SPG (Sekolah Pendidikan Guru) yang berlokasi di Jl. Yogyakarta Malang (sekarang Jl. Veteran), dan di SMA PPSP IKIP MALANG. Pada tahun 1985 sampai sekarang, As'ari mengabdikan diri menjadi dosen di Jurusan Matematika IKIP MALANG yang sekarang berubah menjadi Universitas Negeri Malang.

Di awal-awal karirnya, As'ari pernah membantu menjadi dosen tidak tetap di Jurusan Pendidikan Matematika di Fakultas Tarbiyah IAIN Malang, serta di beberapa perguruan tinggi swasta, antara lain: IKIP Budi Utomo, Universitas Muhammadiyah, dan di Universitas Islam Malang. Di Universitas Islam Malang (UNISMA) bahkan As'ari pernah menjabat sebagai ketua jurusan.

Di tingkat nasional, As'ari pernah membantu menjadi konsultan di Direktorat Jenderal Pendidikan Dasar dan Menengah (DITJEN DIKDasmen), Departemen Pendidikan Nasional atau Departemen Pendidikan dan Kebudayaan. As'ari membantu pengelolaan beberapa proyek antara lain Proyek CALISTUNG (Baca Tulis Hitung), BASIC EDUCATION PROJECT, dan PROYEK IMBAL SWADAYA. As'ari membantu mulai dari tahun 1996 sampai tahun 2003, dan lebih banyak waktunya tinggal di Jakarta. Selanjutnya, sejak tahun 2003 sampai 2012, As'ari kembali tinggal di Malang. Akan tetapi, meskipun lebih banyak berada di Malang, tetapi kiprahnya di dunia pendidikan justru banyak dimanfaatkan oleh USAID (United State Agency for International Development). As'ari diminta membantu pengelolaan beberapa proyek yaitu Managing Basic Education Project (2003 - 2007), Decentralized Basic Education Project (2007 - 2011), PRIORITAS (2011 - 2012). Di sela-sela waktu, bahkan As'ari juga diminta membantu menjadi konsultan di AusAID, UNICEF, UNESCO, dan ADB.

Di bidang organisasi, As'ari pernah menjadi Wakil Presiden III (Wakil Presiden Bidang Pendidikan) Himpunan Matematika Indonesia atau IndoMS selama dua periode kepemimpinan, yaitu tahun 2002-2004 dan tahun 2004-2006. Akhir-akhir ini, As'ari juga menjadi anggota



dewan pembina IMES (Indonesian Mathematics Educators Society), serta menjadi pembina organisasi guru matematika Indonesia, yaitu Matematika Nusantara wilayah Jawa Timur.

Beberapa buku telah berhasil diterbitkan. Buku tentang “Ragam Tugas untuk Mengembangkan Disposisi Berpikir Kritis”, dan “Mengembangkan Kemampuan Berpikir Tingkat Tinggi (HOTS) dalam Pembelajaran Matematika”, telah dipublikasikan oleh Penerbit Universitas Negeri Malang. As’ari juga berhasil membantu pemerintah dalam menyediakan buku teks untuk siswa dan guru dalam rangka penerapan kurikulum 2013. As’ari berhasil menulis beberapa buku yang diterbitkan oleh Pusat Kurikulum dan Perbukuan Kementerian Pendidikan dan Kebudayaan (KEMDIKBUD), yaitu Buku Siwa Matematika Kelas 7, Kelas 8, dan Kelas 12. As’ari juga berhasil menulis Buku Guru Matematika Kelas 7, Kelas 8, dan Kelas 12. Terakhir, untuk keperluan Peningkatan Keprofesional Berkelanjutan (PKB) bagi guru-guru SMP, As’ari juga menuliskan beberapa modul yang diterbitkan oleh P4TK Matematika Yogyakarta.



**Abd. Qohar.** lahir di Bojonegoro tanggal 21 Maret 1968. Menempuh Pendidikan Sarjana (S1) di Jurusan matematika Universitas Gadjah Mada, lulus tahun 1990. Kemudian melanjutkan kuliah program Magister (S2) di Teknik Informatika ITS dan lulus tahun 2001. Selanjutnya pada tahun 2010 berhasil lulus pada program doktor (S3) Pendidikan Matematika di Universitas Pendidikan Indonesia.

Pernah mengajar di Universitas Darul Ulum Jombang, Universitas Merdeka Malang dan Universitas Terbuka. Selain mengajar di perguruan tinggi, juga pernah bertugas sebagai expert pada program Teachers Quality Improvement Program (TEQIP), kerjasama UM dan PERTAMINA pada tahun 2011-2015. Saat ini aktif mengajar di program studi S1, S2 dan S3 pendidikan Matematika Universitas Negeri Malang. Sudah banyak karya ilmiah yang dihasilkan pada jurnal-jurnal terakreditasi nasional maupun yang terindex Scopus, antara lain: “Development of Android Based Instructional Media of

Algebraic Tiles for Quadratic Equation”, “Android-Based Mathematics Learning Games That are Interesting for Junior High School Students”, “Improving mathematical communication ability and self regulation learning of yunior high students by using reciprocal teaching”.



**BUATBUKU.COM**

----- BUAT AJA DULU-----

# Computational THINKING

*(Pemecahan Masalah di Abad ke-21)*

**A**pa itu Computational Thinking (CT)? CT dapat digunakan untuk memecahkan masalah pada skala yang rumit secara algoritmik, dan sering digunakan untuk mewujudkan peningkatan efisiensi yang besar.

Buku *Computational Thinking: Pemecahan Masalah di Abad ke-21* merupakan sebuah buku teks yang berisi tentang konsep-konsep Computational Thinking (CT) di beberapa disiplin ilmu.

Buku ini mengulas tentang sejarah munculnya CT, tokoh-tokoh yang mengawalinya, dan hal apa saja yang melatarbelakangi kemunculan CT. Pengertian CT juga disajikan dalam Bab 2. Selain itu ulasan tentang CT di beberapa disiplin ilmu juga disajikan secara runtut dan rinci di dalam buku ini. Mulai dari CT dalam bidang Teknologi Informasi beserta contoh-contohnya, CT dalam bidang Sains, CT dalam bidang Matematika, CT dalam bidang pendidikan, dan yang terakhir CT dalam bidang Pendidikan Matematika. Yang tak kalah menarik dari isi buku ini adalah di akhir bab, penulis memaparkan sekitar 50 artikel hasil-hasil penelitian CT dari berbagai sumber di seluruh dunia.

Buku ini ditujukan kepada para mahasiswa, dosen, peneliti, dan akademisi lain yang ingin memperdalam pengetahuan tentang CT.