

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/350383897>

Computational Thinking Learning and Teaching Guide for Primary and Secondary Schools in Indonesia

Book · December 2020

CITATIONS

3

READS

47,415

6 authors, including:



[Adi Mulyanto](#)

Bandung Institute of Technology

16 PUBLICATIONS 37 CITATIONS

[SEE PROFILE](#)



[Ginar Santika Niwanputri](#)

The University of Sheffield

31 PUBLICATIONS 40 CITATIONS

[SEE PROFILE](#)




[Yasya Rusyda](#)

Bandung Institute of Technology

3 PUBLICATIONS 8 CITATIONS

[SEE PROFILE](#)



PEMBELAJARAN COMPUTATIONAL THINKING PADA PENDIDIKAN DASAR DAN MENENGAH

Institut Teknologi Bandung
2020



Pusat Artificial Intelligence ITB



**LEMBAGA PENELITIAN
DAN PENGABDIAN
KEPADA MASYARAKAT**
INSTITUT TEKNOLOGI BANDUNG

Kata Pengantar

Puji dan syukur penulis panjatkan kepada Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan modul yang berjudul “Pembelajaran *Computational Thinking* pada Pendidikan Dasar dan Menengah” sebagai bahan acuan dari kegiatan Program Pengabdian Masyarakat (PPM) ITB 2020 yang berjudul “Pengenalan dan Pendampingan Penerapan *Computational Thinking* pada Pendidikan Dasar dan Menengah di Kota Bandung”. Kegiatan ini merupakan awal dari pengenalan dan pendampingan guru-guru sekolah di Bandung untuk dapat mengenal dan memahami konsep *Computational Thinking* serta bagaimana penerapannya. Kegiatan *workshop* yang telah dilaksanakan pada tanggal 17 Oktober 2020 dan 24 Oktober 2020 sebagai wujud dari pelaksanaan kegiatan PPM ITB ini tidak akan terlaksana tanpa dukungan dari Pusat *Artificial Intelligence* ITB serta Lembaga Penelitian dan Pengabdian kepada Masyarakat ITB.

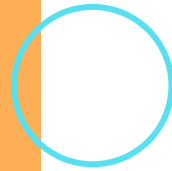
Penulis memohon maaf sebesar-besarnya atas ketidaksempurnaan yang ada pada modul ini. Penulis pun berharap bahwa modul ini dapat memberi manfaat bagi seluruh pihak yang membacanya, baik sebagai sumber pengetahuan maupun sebagai referensi dalam pengenalan *Computational Thinking*. Penulis juga sangat terbuka atas kritik dan saran agar modul ini dapat disempurnakan di masa mendatang dengan mengirimkan surel ke itb@bebras.or.id.

Bandung, 8 November 2020

Tim Penyusun Materi

Daftar Isi

<i>Kata Pengantar</i>	<i>ii</i>
<i>Daftar Isi</i>	<i>iii</i>
<i>A. Pengenalan Computational Thinking</i>	<i>1</i>
<i>A.1. Computational Thinking</i>	<i>2</i>
<i>A.2. Penerapan Computational Thinking Dalam Bidang Lain</i>	<i>6</i>
<i>B. Pembelajaran Computational Thinking</i>	<i>9</i>
<i>B.1. Code.org</i>	<i>10</i>
<i>B.2. CS Unplugged</i>	<i>12</i>
<i>B.3. STEM Family</i>	<i>14</i>
<i>B.4. Ozobot</i>	<i>16</i>
<i>B.5. Bebras</i>	<i>17</i>
<i>C. Latihan Soal Computational Thinking</i>	<i>18</i>
<i>C.1. Mari Berlatih!</i>	<i>19</i>
<i>C.2. Soal Tingkat Sekolah Dasar</i>	<i>20</i>
<i>C.3. Soal Tingkat Sekolah Menengah Pertama</i>	<i>30</i>
<i>C.4. Soal Tingkat Sekolah Menengah Atas</i>	<i>41</i>
<i>D. Topik Tambahan</i>	<i>54</i>
<i>D.1. Membuat Soal Computational</i>	<i>55</i>
<i>Thinking yang Baik</i>	<i>55</i>
<i>D.2. The Science of Bebras Task</i>	<i>62</i>
<i>DAFTAR PUSTAKA</i>	<i>67</i>



A. Pengenalan *Computational Thinking*

A.1. Computational Thinking

Computational thinking (CT) merupakan istilah yang merujuk pada ide-ide dan konsep utama yang ada pada bidang Informatika dan *Computer Science* (Bocconi et al., 2016). Pada tahun 2006, Wing mengenalkan istilah *computational thinking* yang melibatkan kemampuan memecahkan masalah, mendesain sistem, dan memahami perilaku manusia dengan menggambarkan konsep dasarnya ke *computer science*. Menurutnya, *computational thinking* terdiri dari berbagai *mental tools* yang mencerminkan luasnya bidang *computer science*. *Computational thinking* merupakan **proses berpikir yang diperlukan dalam memformulasikan masalah dan solusinya, sehingga solusi tersebut dapat menjadi agen pemroses informasi yang efektif dalam menyelesaikan masalah** (Wing, 2010).

Berdasarkan materi pembelajaran *online* BBC Bitesize (n.d.), *computational thinking* memungkinkan kita memahami masalah yang kompleks dan mengembangkan solusi yang mungkin. Kemudian masalah dan solusi tersebut bisa disajikan dengan cara yang dipahami oleh komputer, manusia, atau keduanya. Berdasarkan materi pembelajaran tersebut, *computational thinking* mencakup dekomposisi, abstraksi, berpikir dan merumuskan algoritma, dan pembentukan pola solusi untuk persoalan-persoalan sejenis. Kemampuan *computational thinking* adalah kemampuan yang perlu diasah melalui latihan-latihan, dan merupakan salah satu pengetahuan dasar untuk kemampuan penyelesaian persoalan tingkat tinggi yang dibutuhkan insan abad ke-21. *Computational Thinking* menjadi salah satu kemampuan yang penting untuk diasah sejak usia dini karena pada era informasi, era industri 4.0 atau *society* 5.0. manusia hidup di dunia nyata, dan sekaligus di dunia digital yang dikelilingi dengan IoT (*Internet of Things*), *Big Data*, dan *Artificial Intelligence*.

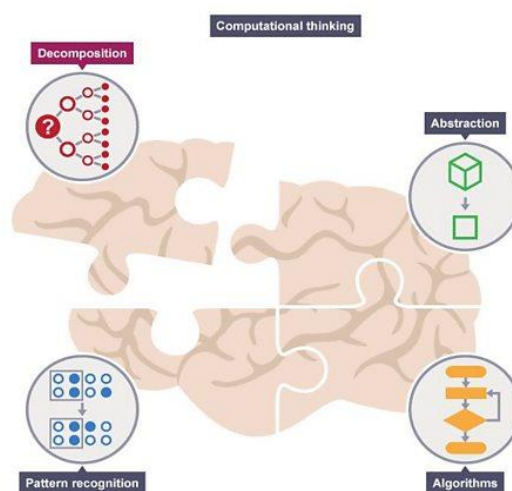
Computational thinking adalah *problem solving*, yakni mendefinisikan akar masalah beserta solusinya, lalu memikirkan apakah solusi tersebut akan lebih efektif jika melibatkan komputer. *Computational thinking* merupakan proses *problem-solving* yang terdiri dari beberapa karakteristik sebagai berikut (CSTA & ISTE, 2009 seperti yang dikutip oleh Bocconi et al., 2016):

1. Merumuskan masalah yang dapat membuat seseorang menggunakan komputer dan *tools* lainnya untuk membantu mencari solusinya
2. Mengelola dan menganalisis data secara logis
3. Merepresentasikan data melalui abstraksi seperti model dan simulasi

4. Mengotomatisasi solusi melalui *algorithmic thinking* (serangkaian urutan langkah-langkah)
5. Mengidentifikasi, menganalisis, dan mengimplementasi seluruh solusi yang mungkin untuk mencapai langkah yang paling efisien dan efektif dari *resource* yang ada
6. Menggeneralisasi dan mentransfer proses *problem-solving* untuk berbagai permasalahan

Terdapat beberapa hal yang akan lebih cepat diselesaikan oleh manusia dan ada yang akan lebih cepat diselesaikan dengan komputer, kita sebagai manusia adalah yang menentukan. Dalam praktiknya, terdapat sekumpulan kemampuan yang merupakan bagian dari *computational thinking*. Google for Education membahas bagaimana pendekatan *computational thinking* harus dilakukan. Google for Education memiliki empat langkah dasar yang terdiri dari **dekomposisi, pengenalan pola, abstraksi, dan perancangan algoritma** (Kidd, Lonnie R, & Morris, Jr., 2017). Berikut ini adalah pengertian dari keempat langkah dasar tersebut (BBC Bitesize, n.d.).

1. **Abstraksi** adalah proses yang berfokus pada informasi yang penting dan mengabaikan detail yang tidak relevan
2. **Algoritma** adalah mengembangkan solusi langkah demi langkah atau aturan yang harus diikuti untuk menyelesaikan masalah
3. **Dekomposisi** adalah memecah masalah atau sistem yang rumit menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola
4. **Pengenalan pola** adalah mencari kesamaan di antara dan di dalam masalah.

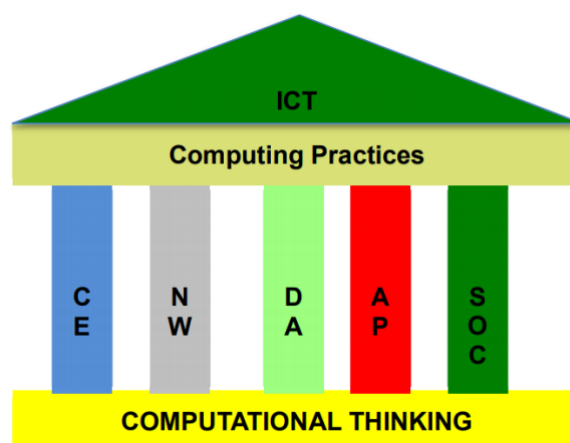


Gambar 1. Empat langkah dasar *computational thinking*
(Sumber: <https://www.bbc.co.uk/bitesize/topics/z7tp34j>)

Keempat hal tersebut bukanlah urutan dan tidak juga harus dimiliki semua. Kemampuan dan keterampilan berpikir komputasional tersebut ditunjang dengan beberapa sikap sebagai berikut (Tim Olimpiade Komputer Indonesia, 2018).

- a) Yakin dan percaya diri dalam menghadapi dan mengelola kompleksitas.
- b) Gigih dan tekun bekerja dalam menghadapi persoalan yang sulit.
- c) Toleran terhadap ambiguitas.
- d) Kemampuan untuk menangani “*open ended problems*”.
- e) Kemampuan berkomunikasi dan bekerja sama dalam tim untuk mencapai suatu tujuan atau menghasilkan solusi.

Pada negara maju, *Computer Science*, atau yang biasa juga disebut dengan Informatika, sudah diajarkan sejak usia dini di tingkat pendidikan dasar. Materi dan kegiatan yang dirancang mengacu pada kerangka kurikulum yang disusun oleh persatuan guru-guru, asosiasi profesi informatika, perusahaan terkemuka di bidang informatika dan TIK, serta organisasi-organisasi nirlaba yang peduli terhadap perlunya edukasi tentang informatika sejak usia dini. Dalam kerangka kurikulum Informatika tersebut, terdapat lima bidang pengetahuan, yakni Sistem Komputer (CE), Jaringan Komputer (NW), Analisis Data (DA), Algoritma dan Pemrograman (AP), dan Aspek Sosial dari pemanfaatan Informatika (SOC). Selain kalimat bidang pengetahuan tersebut, didefinisikan pula praktik-praktik komputasi untuk mengemas pengetahuan dan mempraktikkannya, yakni pembinaan menumbuhkan budaya komputasi, menciptakan artefak, berkolaborasi untuk mewujudkan suatu produk TIK, menguji dan memperbaiki/menyempurnakan artefak TIK, mengenali dan mendefinisikan permasalahan komputasi, berkomunikasi tentang komputasi, dan mengembangkan serta menggunakan abstraksi (Tim Olimpiade Komputer Indonesia, 2018).



Gambar 2 Kerangka kurikulum Informatika
(Sumber: <https://aren.cs.ui.ac.id/kikd/xindex.php>)

Salah satu opini yang populer terkait computational thinking adalah bahwa hal tersebut sama dengan *coding*, namun hal tersebut kurang tepat. Sebuah bahasa pemrograman yang komputer dapat digunakan untuk menyelesaikan sebuah tugas atau serangkaian instruksi adalah *coding*, dan *computational thinking* merupakan proses *problem solving* di baliknya (Grover, 2018). Apabila dikaitkan dengan kemampuan manusia, kemampuan yang dibutuhkan untuk memerintah komputer untuk melakukan suatu hal adalah yang disebut dengan programming, sedangkan proses berpikir dibalik *programming* adalah *computational thinking* (Digital Promise, 2017). Terdapat beberapa miskonsepsi yang dipahami oleh orang-orang terkait apa yang dimaksud dengan *computational thinking*. Berikut ini adalah beberapa kesalahan terkait pemahaman tentang *computational thinking*.

- a) *Computational thinking* adalah konseptual, bukan sebuah *programming skill*.
- b) *Computational thinking* merupakan fundamental atau dasar dari berpikir, bukan merupakan kemampuan menghafal. Semua soal yang bisa dihafal dan dijawab ulang bukanlah computational thinking.
- c) *Computational thinking* adalah cara manusia berpikir, bukan komputer. Cara berpikir manusia berbeda dengan komputer karena komputer sangat detail dan terstruktur langkah demi langkah, sedangkan manusia lebih abstrak.
- d) *Computational thinking* melengkapi dan mengkombinasikan matematika dengan *engineering thinking*.
- e) *Computer science* secara inheren mengacu pada *engineering thinking*.
- f) *Computational thinking* merupakan ide, bukanlah artefak.

A.2. Penerapan *Computational Thinking* Dalam Bidang Lain

Pemikiran komputasi memiliki sejarah yang panjang dalam ilmu komputer, yakni sejak tahun 1950-an dan 1960-an. Dengan berkembangnya teknologi yang ada pada saat ini, komputasi kini memungkinkan dan mendorong banyak teknologi yang mempengaruhi banyak aspek di kehidupan kita. Kemampuan *computational thinking* (CT) telah menjadi keterampilan prasyarat untuk banyak upaya di abad ke-21 ini (Wing, 2008). **Penerapan *Computational Thinking* tidak hanya terpaku pada kegiatan yang berhubungan dengan komputer, namun juga dapat berhubungan dengan subjek studi lain.** Berikut ini merupakan beberapa subjek studi yang telah terintegrasi dengan konsep *computational thinking* (Mgova, 2018).

1. STEM

Pemikiran komputasi dapat ditanamkan dalam mata pelajaran Sains, Teknologi, Teknik, dan Matematika (STEM), namun tidak semua guru kompeten dan tidak semua murid minat akan hal tersebut. Oleh karena itu, pembawaan pemikiran komputasi di kelas STEM harus memberikan pandangan realistis di bidang STEM kepada siswa. Salah satu contoh implementasinya adalah C3stem, yaitu sebuah kerangka kerja (*framework*) yang dirancang untuk mengatasi masalah keterampilan pemecahan masalah di bidang STEM. Kerangka kerja tersebut bertujuan untuk menyediakan lingkungan belajar yang menarik yang memungkinkan siswa untuk memecahkan masalah dunia nyata yang ada di bidang STEM. Kerangka c3stem mendukung siswa untuk menganalisis arus lalu lintas dengan kinematika kendaraan dan perilaku mengemudi dasar. Dalam studi ini, para siswa menerapkan beberapa konsep seperti kecepatan dan akselerasi dari fisika dengan mengerjakan tantangan berbasis realitas dari domain lalu lintas.

2. Etika

Pemikiran komputasi tidak hanya berlaku di bidang STEM atau ilmu komputer, tetapi juga berlaku dalam studi etika. Seoane-Pardo mengembangkan “*moral machine*” dan pemrograman pengambilan keputusan di kelas etika untuk memungkinkan siswa membuat keputusan logis dalam situasi ekstrem ketika bekerja dalam pemrograman mobil. Berdasarkan alasan berbagai pendekatan etis, siswa diminta untuk mengembangkan skema pengambilan keputusan yang lebih etis agar mesin merespons. Dengan mengembangkan skema logis, secara tidak langsung siswa diharuskan untuk berpikir secara komputasi.

3. Sains

Pada tahun 2010, terdapat sebuah lokakarya (*workshop*) yang diadakan tiga hari untuk siswa sekolah menengah atas dan guru sains dengan judul “*computational thinking for sciences*”. Tujuan dari lokakarya tersebut adalah untuk memperdalam pemahaman yang terhubung antara ilmu komputer, ilmu alam, dan matematika. Topik yang diajarkan dalam lokakarya tersebut adalah berpikir komputasi, simulasi, biologi, kimia fisika, probabilitas, matematika, karir dalam ilmu komputasi dan rencana pelajaran untuk guru.

Selain itu, para ilmuwan juga sudah membuat kursus yang dirancang untuk memperkenalkan *computational thinking* dalam sains. Kursus tersebut dirancang oleh fakultas ilmu komputer bekerja sama dengan departemen fisika, kimia, dan bioinformatika. Siswa diperkenalkan dengan konsep alat komputasi dari Fisika dan Bioinformatika menggunakan Python.

4. Sains Biologi

Pada tahun 2009 terdapat kursus yang dikembangkan untuk mengajarkan Bioinformatika kepada mahasiswa biologi. Kursus tersebut dimaksudkan untuk memperkuat keterampilan komputasi kuantitatif dan kritis untuk siswa biologi. Selain itu, pada tahun 2014 pun terdapat kursus berjudul “*Computational Approaches for Life Scientist*”. Kursus tersebut dirancang untuk menjembatani kesenjangan antara pemikiran komputasi dan siswa ilmu sains.

5. Jurnalistik

Pada tahun 2010, terdapat sebuah program jurnalisme interaktif yang dirancang untuk siswa sekolah menengah dan guru-guru mereka selama sekolah musim panas dan setelah sekolah. Siswa dan guru dari konselor seni, teknologi, dan bimbingan bekerja bersama dalam melakukan wawancara dan mengembangkan cerita baru yang kemudian dikumpulkan dalam bentuk teks, video, dan animasi prosedural dalam Scratch sebagai paket cerita.

6. Musik

Pada tahun 2010, terdapat kursus “*sound thinking*” yang dikembangkan untuk mendorong pemikiran komputasi kepada siswa seni. Siswa dalam kursus ini pertama kali diekspos ke halaman web yang dibuat menggunakan HTML dan JavaScript API dan menanamkan file musik ke halaman web mereka dibuat. Siswa bermigrasi dari menggunakan HTML dan JavaScript ke awal yang tampaknya kuat dalam mendorong konsep pemikiran komputasi siswa. Menggunakan Scratch untuk membuat musik, siswa dapat belajar banyak konsep komputasi termasuk *loop*, inisialisasi, variabel, mengubah variabel secara algoritmik, modularisasi dan pemrosesan acara.

Selain beberapa subjek studi yang telah terintegrasi dengan konsep *computational thinking* yang telah dijelaskan sebelumnya, berikut ini merupakan beberapa contoh aktivitas penerapan *Computational Thinking* dalam beberapa bidang pelajaran (Grover, 2018).

1. Seni Bahasa

- a. Penggunaan logika untuk menyusun cerita yang campur aduk dalam urutan yang benar
- b. Identifikasi pola untuk berbagai jenis kalimat dan aturan tata bahasa
- c. Penggunaan logika tingkat pertama untuk sampai pada kesimpulan berdasarkan fakta yang diberikan
- d. Pembangunan jejaring sosial untuk menganalisis cerita
- e. Program cerita dengan jalur alternatif ("Pilih petualangan Anda sendiri")

2. Matematika

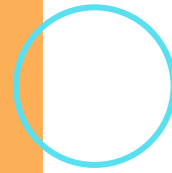
- a. Memodelkan fungsi dalam aljabar melalui program (bandingkan dengan fungsi dalam program)
- b. Menuliskan sebuah algoritma (atau urutan langkah-langkah yang tepat) tentang bagaimana melakukan perkalian matriks atau bagaimana menyelesaikan persamaan kuadrat
- c. Menggunakan dekomposisi untuk menyelesaikan masalah kata
- d. Generalisasi (sebagai representasi aljabar) dengan mengidentifikasi pola

3. Sains

- a. Lakukan klasifikasi spesies dengan logika eksplisit "*If-Then*"
- b. Membangun model komputasi dari fenomena fisik
- c. Membuat (program) model komputasi dan simulasi untuk mempelajari dan menginterogasi fenomena dibandingkan bermain atau memanipulasi simulasi perangkat lunak *pre-developed* dari fenomena ilmiah

4. Sains Sosial

- a. Mempelajari data dan mengidentifikasi pola/tren dalam perang dan peristiwa bersejarah lainnya
- b. Membuat visualisasi dari pola dan tren tersebut
- c. Membuat simulasi untuk mempelajari hubungan dalam fenomena ilmu sosial seperti pendidikan dan kesehatan wanita
- d. Membuat model untuk sistem sosial, jejaring sosial, atau pilihan sosial.

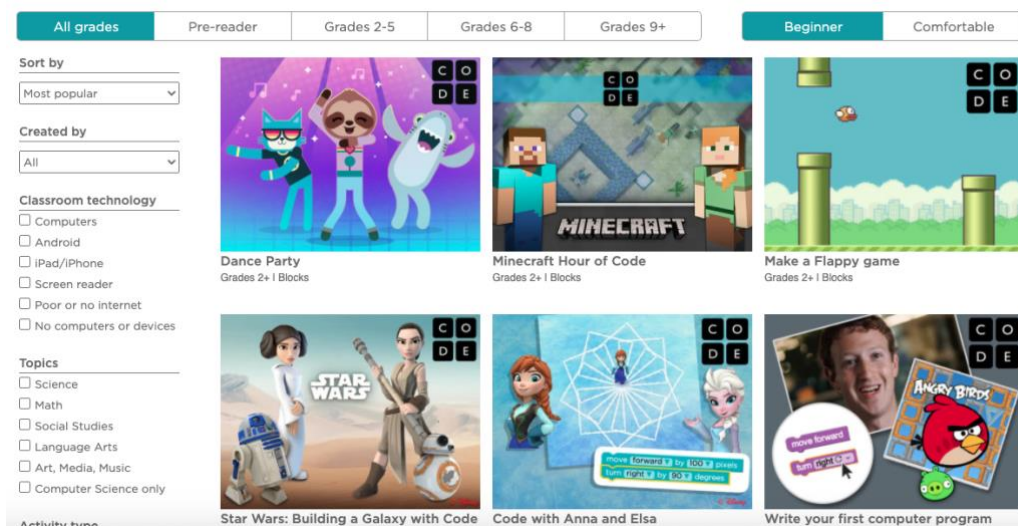


B. Pembelajaran *Computational Thinking*

B.1. Code.org

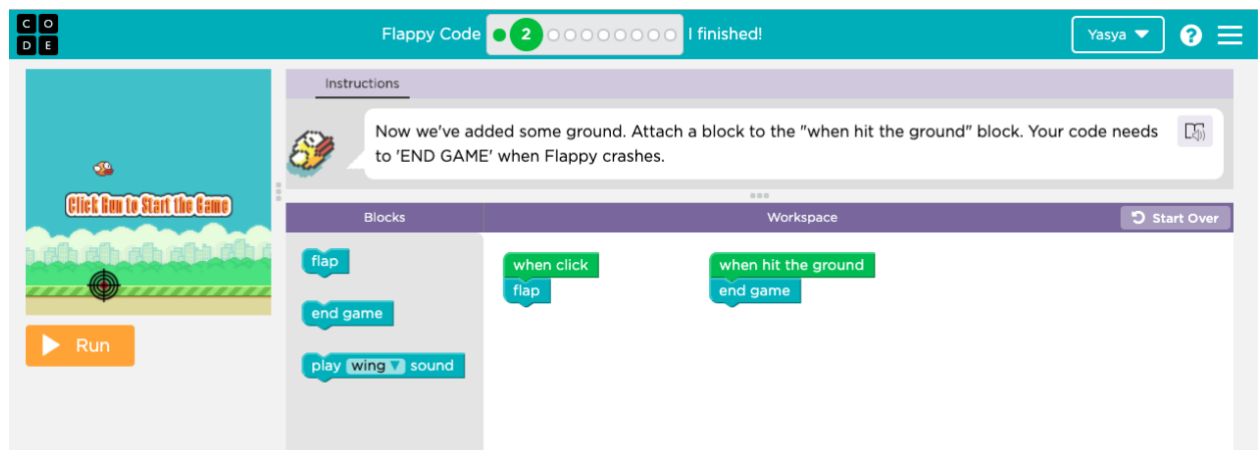
Code.org® merupakan lembaga nonprofit yang didedikasikan untuk memperluas akses menuju ilmu komputer di sekolah dan meningkatkan partisipasi wanita muda dan siswa lain dari kelompok yang kurang terwakili di bidang ilmu komputer. Visi lembaga tersebut yaitu siswa di setiap sekolah memiliki kesempatan untuk mempelajari ilmu komputer sebagai bagian dari pendidikan inti K-12 mereka. Sebagai penyedia kurikulum ilmu komputer K-12 terkemuka di distrik sekolah terbesar di Amerika Serikat, Code.org, juga membuat kampanye Hour of Code tahunan, yang telah melibatkan lebih dari 15% dari siswa di dunia. Code.org didukung oleh banyak pendonor diantaranya yaitu Microsoft, Facebook, Amazon, Infosys Foundation, Google dan banyak lagi. (code.org)

Pada code.org, pengguna bisa mendaftar sebagai guru dan juga sebagai siswa. Mereka bisa belajar atau mengajar berdasarkan katalog yang ada. Guru bisa melihat katalog yang tersedia mencakup berbagai rentang usia.



Gambar 3 Halaman daftar katalog code.org
(Sumber: code.org)

Pembelajaran *Computational Thinking* pada situs web ini dilakukan dengan pembelajaran pemrograman sederhana yang interaktif. Pemrograman dilakukan dengan melakukan penyusunan langkah-langkah yang dibutuhkan sesuai instruksi pada masing-masing permainan. Ada terdapat banyak variasi permainan pada situs web ini. Salah satu contohnya yaitu pembelajaran pembuatan permainan *flappy bird* dengan menyusun langkah-langkah sesuai instruksi.



Gambar 4 Contoh pembelajaran dengan membuat permainan flappy bird
(Sumber: <https://studio.code.org/flappy/1>)


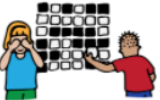




B.2. CS Unplugged

CS Unplugged adalah situs web yang berisi kumpulan kegiatan belajar gratis yang mengajarkan konsep dasar ilmu komputer melalui permainan dan teka-teki menarik dengan menggunakan peralatan sederhana. Materi pelajaran pada CS Unplugged memiliki keterkaitan yang erat dengan *computational thinking*, yakni siswa akan belajar cara menggambarkan masalah, mengidentifikasi rincian penting yang diperlukan untuk menyelesaikan masalah tersebut, memecah masalah menjadi langkah-langkah yang kecil dan logis, menggunakan langkah-langkah tersebut untuk membuat proses yang memecahkan masalah, dan kemudian mengevaluasi proses tersebut (CS Unplugged, n.d.).

Konsep-konsep *computational thinking* pada materi pelajaran di CS Unplugged terhubung satu sama lain dan saling mendukung, namun tidak semua aspek *computational thinking* akan terjadi di setiap unit atau materi pelajaran. Pada setiap unit dan pelajaran terdapat penjelasan terkait koneksi penting antara kegiatan yang dilakukan dalam pelajaran dengan konsep *computational thinking*. Berikut ini adalah enam macam topik utama yang terdapat di CS Unplugged. (CS Unplugged, 2020).

Topics

Open a topic to see all related unit plans, lessons, curriculum integrations, and programming challenges.

Binary numbers Ages 5 to 10 6 lessons 7 curriculum integrations 24 programming challenges		Error detection and correction Ages 5 to 10 3 lessons 5 curriculum integrations 24 programming challenges	
Image Representation Ages 5 to 10 1 lessons		Kidbots Ages 5 to 10 4 lessons 4 curriculum integrations 50 programming challenges	
Searching algorithms Ages 5 to 10 6 lessons 4 curriculum integrations		Sorting networks Ages 5 to 14 4 lessons 2 curriculum integrations	

Gambar 5 Enam topik utama yang ada di dalam CS Unplugged
(Sumber: <https://csunplugged.org/en/topics/>)

Di dalam materi dalam CS Unplugged akan ditampilkan capaian pembelajaran dari materi tersebut serta penjelasan lengkap mengenai letak penerapan *computational thinking* di dalam materi permainan tersebut. Seluruh penjelasan terkait cara mengajar, contoh bermain, termasuk aset tambahan (*printables*) pun disediakan di dalam halaman ini (CS Unplugged, n.d.).

Home / Topics / Error detection and correction / Error detection and correction / Parity magic

Parity magic

- Duration: 30 minutes
- Ages 8 to 10: Lesson 1

▼ Learning outcomes

Students will be able to:

- Describe the steps it takes to work out how to find the card that is turned over.
Computational Thinking: Decomposition
- Describe what are rows and what are columns.
Mathematics: Numeracy
- Discuss why changing one card will change the state of the row and column which it is in.
Mathematics: Numeracy
- Explain how knowing what odd and evens numbers are means you can solve the parity problem.
Mathematics: Numeracy
- Explain that each card is a bit and that the cards can represent data.
Computational Thinking: Algorithmic Thinking
- Explain why they chose each extra bit (card) when setting up a parity column/row.
Computational Thinking: Generalising and Patterns

Printables

Parity Cards
Two pages of parity cards per pair of students.

Classroom resources

- A set of 36 parity cards
- Whiteboard

Programming challenges

[View related programming challenges](#)

Key questions

Gambar 6 Capaian pembelajaran untuk aktivitas "Parity Magic"

(Sumber: <https://csunplugged.org/en/topics/error-detection-and-correction/unit-plan/parity-magic-junior/>)

Lesson starter

► See teaching this in action

► Notes on resources

- Teacher to class: "I've just learnt a magic trick I want to show you".
- Teacher to class: "So who will be my assistant?"
- Teacher to student: Hand the cards to the student and ask them to make a grid of 5 by 5 cards (calling it "5 rows of 5 cards might be clearer for younger students"). "Don't make any patterns - try to make it as random as possible". To speed up the setup you could have two students do this task; encourage them to leave a small gap between the cards, and not be too fussy about making them straight.
- Point out that the cards represent bits (binary digits). If they haven't studied binary numbers, you may need to just point out that this is the way everything is represented on computers - the cards here could represent a file, a message, a web page or even a program.
- Teacher to class: "I'm going to make this a little bit harder by adding another row and another column".

► Teaching observations

Table of contents

- Parity magic
- Key questions
- Lesson starter**
- Lesson activities
- Applying what we have just learnt
- Lesson reflection
- Computational Thinking

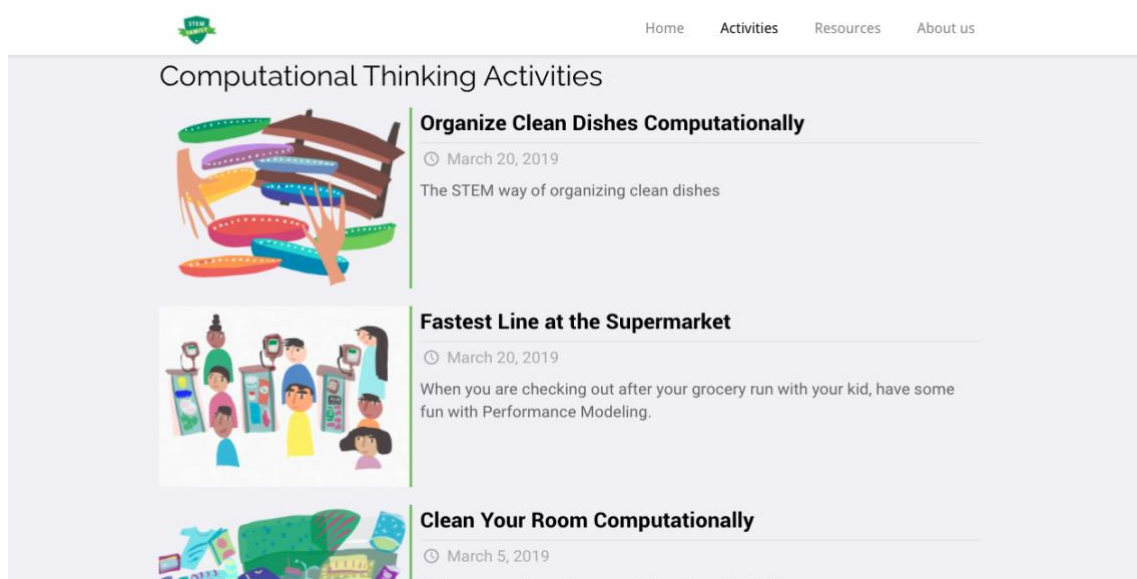
Gambar 7 Panduan aktivitas pada CS Unplugged

(Sumber: <https://csunplugged.org/en/topics/error-detection-and-correction/unit-plan/parity-magic-junior/>)

B.3. STEM Family

STEM Family merupakan website yang menyajikan permainan, strategi, mainan, dan acara terbaik yang harus diketahui orang dewasa untuk memberikan pendidikan yang terbaik kepada anak-anak mereka. Kreator STEM Family merancang kontennya sendiri, dan menyoroti karya hebat orang lain.

STEM Family membuat kegiatan belajar cara berpikir pada STEM yaitu *design thinking*, *systems thinking*, dan *computational thinking* menjadi *screen-free* (tidak belajar dengan gadget tetapi di dunia nyata), *lightweight* (mudah dilakukan), *affordable* (kontennya gratis dan beberapa berbayar murah), *playful* (menyenangkan dan interaktif).



Gambar 8 Aktivitas yang ada pada STEM Family
(Sumber: <https://www.stem.family/activities/computational-thinking-activities/>)

Kegiatan-kegiatan di dalam STEM Family menggunakan studi kasus dari kehidupan sehari-hari. Pada website ini disajikan pula rincian mengenai aktivitasnya seperti judul aktivitas, estimasi durasi aktivitas, capaian pembelajaran yang sesuai dengan tujuan dari pembelajaran STEM, target usia, dan langkah-langkah cara melakukan aktivitasnya yang didukung dengan gambar.

The STEM way of tidying up


Make a computational game out of the drudgery

How long?
🕒 20-25 minutes

What's the Learning Outcome?

- 💡 Debugging
- 💡 Sequencing
- 💡 Algorithm
- 💡 Efficiency

What age is it for?
👤 5-8 year olds



Bekir Mugayitoglu
*Doctorate in Education
Originally from Turkey, then Pittsburgh, now California*

I got my doctorate in educating kids how to code, and how to think computationally so they can thrive in STEM. I have been researching how Offline Activities – where kids aren't in front of a screen, but are playing in the real world – can help kids get core concepts of coding.

Gambar 9 Rincian aktivitas pada STEM Family
(Sumber: <https://www.stem.family/2019/03/05/clean-your-room-computationally-2-2/>)

B.4. Ozobot

Ozobot adalah sebuah robot edukasi ciptaan perusahaan robotik, Evo, yang diprogram untuk membantu anak-anak dalam kegiatan belajar, seperti mengulas kode, dan juga bersosialisasi. Ozobot memiliki sensor yang dapat berinteraksi dengan lingkungan, cahaya, penguas suara, dan bahkan mampu berkomunikasi. Program ozobot dapat diatur menggunakan *remote control* Evo yang tersedia pada aplikasi Ozobot. Dalam aplikasi tersebut pula seorang pengguna dapat menemukan pengguna Ozobot lainnya. Ozobot juga dapat membentuk *team up* yang berfungsi untuk meningkatkan jaringan sosial. Desain Ozobot dapat dikreasikan sendiri oleh pengguna dalam warna hitam dan putih dengan kisaran biaya sebesar \$100 (Salatiga City, n.d.).

Robot Ozobot dapat mengikuti jejak garis yang digambar oleh pengguna dengan menggunakan spidol tebal, berwarna, atau yang setara. Robot Ozobot kini telah tersedia di iOS dan Android. Pengguna dapat menyesuaikan apa yang Robot Ozobot lakukan dengan mengubah warna garis. Beberapa pilihan tindakan yang saat ini dapat dilakukan adalah termasuk membuat Ozobot berputar dalam lingkaran atau mempercepat jalan.



Gambar 10. Robot Ozobot
(Sumber: <https://ozobot.com/>)

B.5. Bebras

Bebras pertama kali digelar di Lithuania (www.bebas.org), merupakan aktivitas ekstra kurikuler yang mendedukasi kemampuan problem solving dalam informatika dengan jumlah peserta terbanyak di dunia. Siswa peserta akan mengikuti kompetisi bebras di bawah supervisi guru, yang dapat mengintegrasikan tantangan tersebut dalam aktivitas mengajar guru. Kompetisi ini dilakukan setiap tahun secara online melalui komputer.

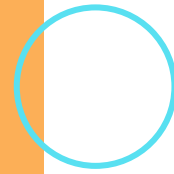


Gambar 11 Logo Bebras Indonesia
(Sumber: <http://bebras.or.id/v3/>)

Yang dilombakan dalam kompetisi adalah sekumpulan soal yang disebut Bebras task. Bebras task disajikan dalam bentuk uraian persoalan yang dilengkapi dengan gambar yang menarik, sehingga siswa dapat lebih mudah memahami soal. Soal-soal tersebut dapat dijawab tanpa perlu belajar informatika terlebih dahulu, tapi soal tersebut sebetulnya terkait pada konsep tertentu dalam informatika dan *computational thinking*. Bebras task sekaligus menunjukkan aspek informatika dan *computational thinking*.

Soal Bebras berperan penting bagi siswa (peserta kompetisi) maupun guru (sebagai penyusun soal). Siswa “didorong” untuk berpikir tentang informatika, sedangkan guru harus berpikir tentang kaitan kehidupan sehari-hari dengan ilmu komputer. Soal yang kreatif dan menarik adalah tantangan utama dalam penyelenggaraan kompetisi Bebras.

Penyusun soal Bebras berusaha memilih soal yang menarik untuk memotivasi siswa dalam mengidentifikasi persoalan informatika dan berpikir lebih dalam tentang teknologi. Mereka juga ingin menyajikan sebanyak mungkin topik informatika dan pembelajaran komputer. Di bidang informatika, masih ada masalah silabus. Bahkan di sekolah-sekolah di beberapa negara, sampai saat ini belum ada kesepakatan bersama materi apa yang harus dimasukkan dalam silabus informatika yang terpadu, dengan memanfaatkan teknologi informasi.



C. Latihan Soal

Computational Thinking

C.1. Mari Berlatih!

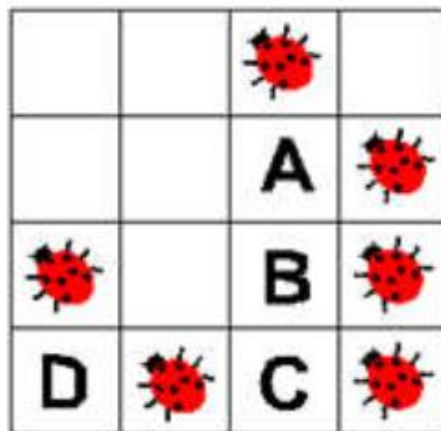
Latihan soal di bawah ini merupakan beberapa soal latihan untuk *Bebras challenge* untuk masing-masing tingkat pendidikan. *Bebras challenge* atau Tantangan Bebras adalah latihan penyelesaian persoalan dalam bentuk soal menarik dan lucu, yang merepresentasikan konsep-konsep informatika yang "tersembunyi" dalam kehidupan sehari-hari, dapat dengan mudah dimengerti oleh anak. Setiap soal harus dapat dijawab dalam waktu kurang lebih 3 menit, berupa soal pendek, yang pada umumnya muat dalam 1 layar. Soal dapat dijawab secara on line atau tanpa komputer. Jika *online*, dapat dikerjakan tanpa perlu menggunakan software lain.

Tantangan Bebras diadakan setiap tahun pada pekan bebras, yaitu minggu kedua bulan November, menyajikan tantangan *computational thinking*, yang harus diselesaikan dalam waktu singkat. Tantangan disajikan dalam bentuk soal menarik, tanpa menggunakan peristilahan informatika karena diperuntukkan bagi siswa mulai berumur 5 tahun s.d. 18 tahun. Tantangan Bebras sangat berguna untuk membentuk kemampuan berpikir dalam suatu semangat "kompetisi".

C.2. Soal Tingkat Sekolah Dasar

Kode Soal : I-2016-SK-10
Sumber : Tantangan Bebras 2016
Judul Soal : Kepik-Kepik
Soal :

Ada beberapa ekor kepik dalam grid yang terdiri dari 16 sel. Sebuah sel dikatakan bertetangga dengan sel lainnya jika sisi atau sudut/pojok keduanya bersebelahan. Ini berarti setiap sel dapat mempunyai sampai dengan 8 tetangga.



Gambar 12 Sel kepik
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan :
Pada gambar di atas, sel manakah yang dikelilingi oleh paling banyak kepik?

Pilihan Jawaban :

- A. A
- B. B
- C. C
- D. D

Pembahasan :

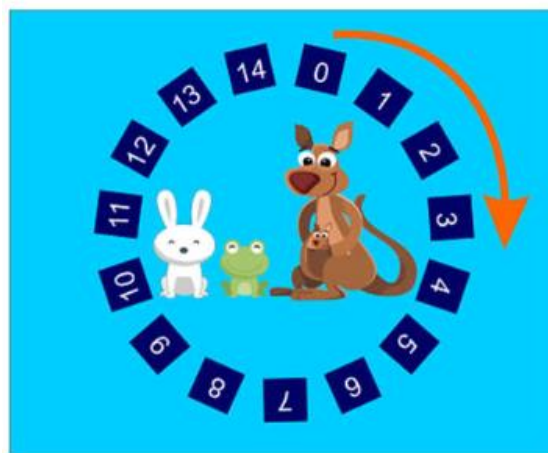
Jawaban yang benar adalah sel B. Kita dapat menemukan dengan menghitung (mencacah) jumlah kepik yang mengelilingi setiap huruf.

Grid adalah struktur yang terdiri dari baris dan kolom. Pada soal ini, kita mencari hubungan antara satu kotak (sel) dengan kotak (sel) tetangga. Salah satu contoh pemakaian grid adalah untuk merepresentasikan sepotong gambar yang terdiri dari piksel, di

mana informasi diperoleh dari kotak tetangga. Tergantung pada definisi "kotak/sel tetangga", pengertiannya bisa berbeda secara signifikan.

Kode Soal : I-2016-IR-01
Sumber : Tantangan Bebras 2016
Judul Soal : Lomba Melompat
Soal :

Pada suatu pagi yang cerah tiga sekawan, seekor kelinci, seekor kodok, dan seekor kangguru, bertanding dalam suatu lomba melompat (*hopping race*). Lintasannya merupakan keliling suatu lingkaran dengan 15 posisi langkah. Posisi langkah itu dinomori dari 0 sampai dengan 14. Setiap kali siapapun yang mencapai atau melalui posisi 14, posisi berikutnya adalah posisi 0.



Gambar 13 Lintasan melompat
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Berang-berang yang kini menjadi wasitnya, akan meniupkan peluit setiap detik selama perlombaan. Pada setiap meniupan peluit, kelinci dapat melompat tepat 3 posisi berikutnya; kodok melompat tepat 2 posisi berikutnya, dan kangguru melompat tepat 5 posisi berikutnya. Di awal lomba semua pemain memulai pada posisi 0.

Tantangan :

Setelah empat kali meniupan peluit, berapa banyak posisi yang telah dilompatinya masing-masing? Tuliskan nomor posisi terakhir, berturut-turut posisi kelinci, posisi kodok dan posisi kangguru (pisahkan dengan hanya sebuah tanda koma, misalnya 3,2,5).

Pilihan Jawaban : Isian

Pembahasan :

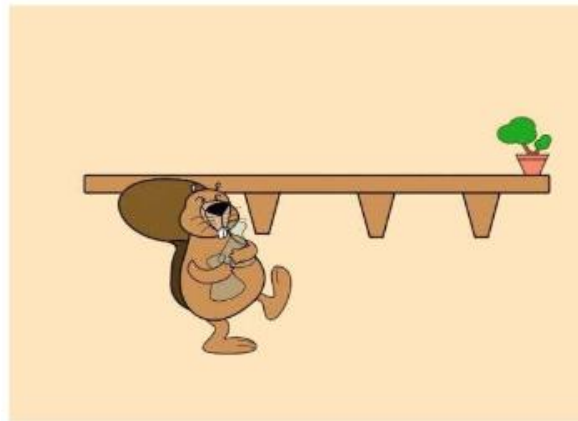
Jawaban yang benar: 12, 8, 5. Setelah terdengar empat kali bunyi peluit, berikut ini adalah jumlah langkah yang telah dilakukan oleh masing-masing pemain, diikuti posisi terakhirnya: Kelinci 12 lompatan, posisi 12 Kodok 8 lompatan, posisi 8 Kangguru 20 lompatan, posisi 5.

Setiap kali mendengar peluit, kelinci, kodok, kangguru melompat sesuai nilai langkahnya yang sudah terdefinisi. Kodok melompat 2 langkah per lompatan, dan dengan melakukan 4 kali, si kodok melompat 8 langkah. Kelinci melompat 3 langkah per lompatan, sehingga saat 4 kali melakukannya dia akan melangkah 12 langkah. Kangguru melompat 5 langkah per lompatan, sehingga ia melangkah sejauh 20 lompatan.

Kita perlu menghitung jumlah langkah yang sudah dilakukan setiap kali setelah peluit ditiup, dan posisi yang dicapai berdasarkan jumlah langkah tersebut yang dicapai masing-masing pemain. Soal ini melatih konsep perkalian (atau penjumlahan yang berulang) dan melakukan operasi modulo (kembali ke 0) untuk menentukan posisi.

Kode Soal : I-2016-US-01
Sumber : Tantangan Bebras 2016
Judul Soal : .Botol
Soal :

Bebras ingin meletakkan 5 botol di atas rak. Ia ingin agar botol-botol itu dibariskan dari yang paling kecil bagian tengahnya hingga yang terbesar dari kiri ke kanan. Susunlah urutan dari botol-botol tersebut.



Gambar 14 Rak Bebras
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan :

Susunlah botol-botol berikut ini di atas rak, terurut menurut bagian tengah (atau leher) dari botol, mulai dari yang paling kecil ke paling besar.



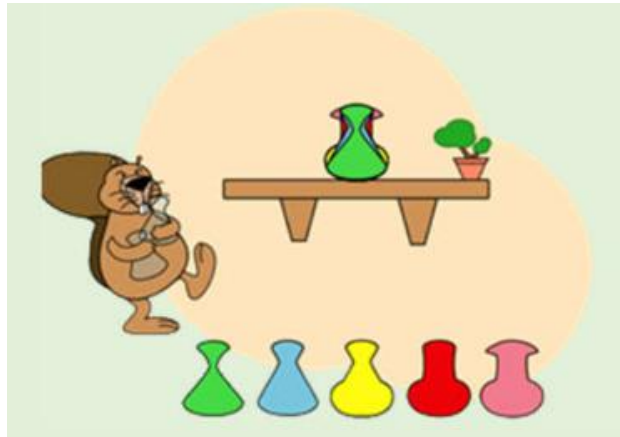
Gambar 15 Botol Bebras
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Pilihan Jawaban : Isian

Pembahasan :

Urutan yang benar adalah: Hijau, Biru, Kuning, Merah, Pink.

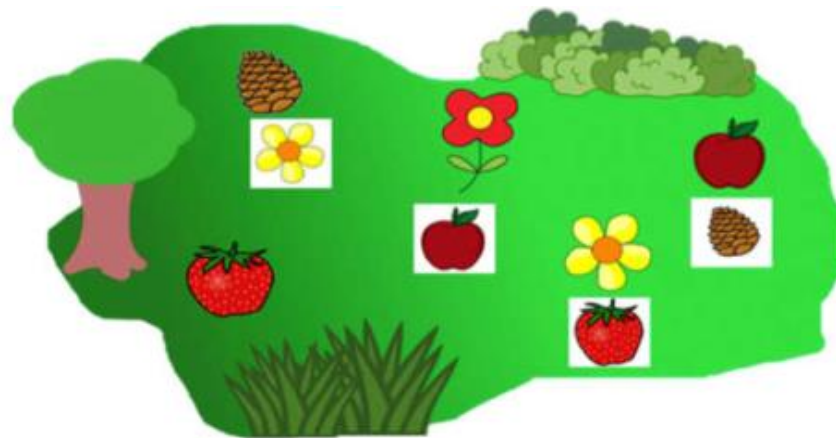
Kita dapat mencoba dengan berbagai cara, namun ada satu indikator apakah meletakkan dua buah botol sudah dalam urutan yang benar. Jika sebuah botol “hilang” atau tidak terlihat karena tertutup saat meletakkan botol lain sesudahnya, maka kita tahu botol tersebut seharusnya diletakkan didepan. Kita dapat mencoba meletakkan botol-botol tersebut satu per satu hingga soal ini terselesaikan. Botol yang paling kecil harus diletakkan paling depan.



Gambar 16 Jawaban urutan botol
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

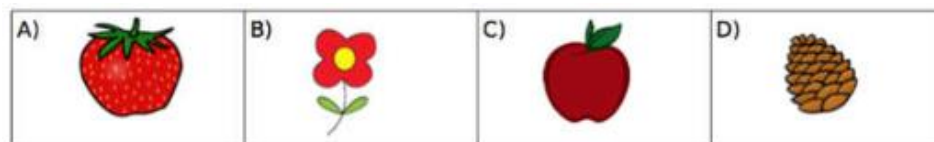
Kode Soal : I-2016-HU-02
Sumber : Tantangan Bebras 2016
Judul Soal : Resep Rahasia
Soal :

Keluarga Bebras sedang mempersiapkan acara Festival Makanan, dan mereka ingin memanggang kue kering; Kati akan membuat kue. Dia sangat memperhatikan urutan membuat kue dengan memasukkan bahan kue dengan urutan yang benar. Ketika dia berjalan ke taman, dia melihat ada secarik kertas pada setiap bahan kue yang akan digunakan. Gambar pada kertas menjelaskan bahan kue yang harus ditambahkan pada urutan berikutnya. Hanya ada satu bahan kue yang tidak memiliki kertas. Ilustrasi taman seperti di bawah ini:



Gambar 17 Bahan-bahan kue
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan : Bahan kue manakah yang harus dimasukkan pertama kali?
Pilihan Jawaban :



Pembahasan :

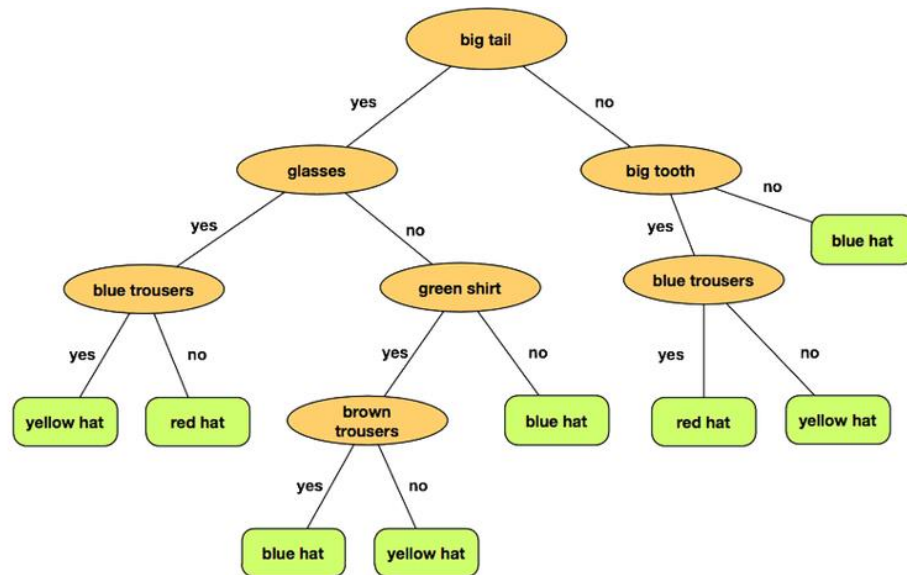
Jawaban yang benar adalah B.

Jika Kati memulai dari B maka dia akan menambahkan kelima bahan kue dengan urutan yang benar. Bahan kue pertama yang harus ditambahkan adalah yang tidak dirujuk oleh bahan lain. Jika dia memulai dari stroberi, maka dia tidak dapat melanjutkan ke tahap berikutnya karena tidak ada petunjuk bahan kue selanjutnya pada kertas. Jika dia memulai

dari apel maka salah, karena bahan kue bunga merah akan terlewatkan. Jika dia memulai dari buah pohon cemara juga salah, karena bahan kue bunga merah dan apel akan terlewatkan.

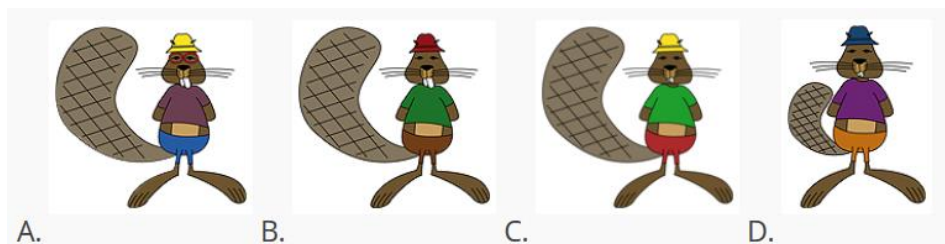
Kode Soal : -
Sumber : Latihan Tantangan Bebras
Judul Soal : Dress code untuk Berang-berang
Soal :

Berang-berang mempunyai sistem aturan berpakaian yang kompleks untuk menentukan penampilannya, yaitu kombinasi dari pakaian. Manfaatkan gambar yang diberikan untuk menentukan aturan berpakaian yang benar.



Gambar 18 Aturan berpakaian berang-berang
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan :
Berang-berang yang mana yang tidak berpakaian seperti aturan berpakaian?
Pilihan Jawaban :



Pembahasan :
Berang-berang kedua (jawaban B) berpakaian tidak sesuai dengan aturan berpakaian. Dia seharusnya memakai topi biru dan bukannya topi merah.

Ini merupakan sebuah contoh pemakaian pohon keputusan dan pengenalan pola. Sebuah pohon keputusan adalah sebuah gambar “pohon” yang menggunakan metode percabangan

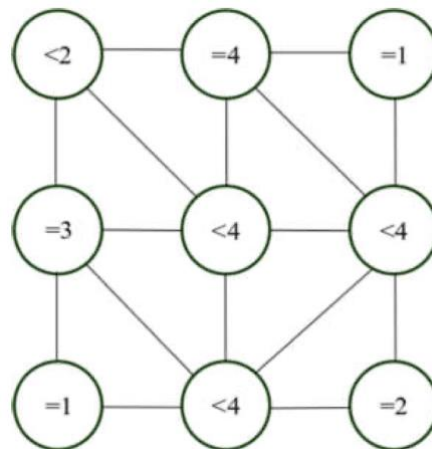
untuk mengilustrasikan setiap kemungkinan hasil suatu keputusan. Gambar yang diberikan disebut pohon karena ada simpul paling atas yang disebut sebagai akar, dengan cabang-cabang yang menghubungkan simpul lainnya. Pada setiap simpul, Anda harus memutuskan ke arah mana Anda ingin pergi dalam pohon, Anda tidak bisa naik lagi.

C.3. Soal Tingkat Sekolah Menengah Pertama

Kode Soal : I-2018-CA-05
Sumber : Tantangan Bebras 2018
Judul Soal : Menghubungkan Lingkaran
Soal :

Tugas anda adalah mewarnai lingkaran-lingkaran pada gambar berikut.

Lingkaran-lingkaran tersebut dihubungkan dengan lingkaran tetangganya (yang terhubung langsung dengan garis). Terdapat 9 lingkaran dan 16 hubungan antar dua buah lingkaran. Angka yang dituliskan dalam lingkaran menunjukkan jumlah tetangga yang harus diwarnai. Misalnya sebuah lingkaran dengan tulisan “=3”, artinya 3 dari 4 tetangganya harus diwarnai. Sebuah lingkaran dengan tulisan “<4” artinya lingkaran tetangga yang harus diwarnai kurang dari 4.



Gambar 19 Lingkaran yang harus terhubung
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

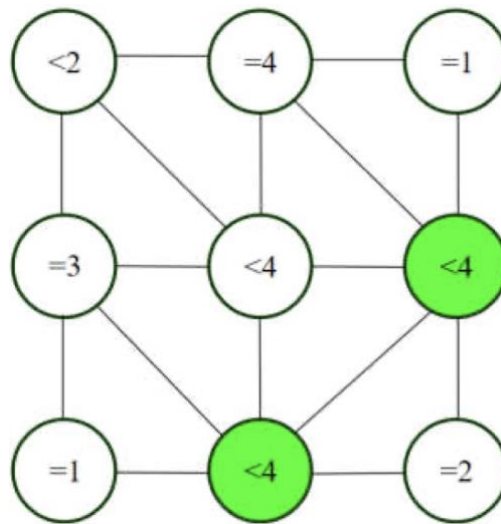
Tantangan : Berapa banyak lingkaran yang harus anda warnai?

Pilihan Jawaban:

- A. 4
- B. 5
- C. 6
- D. 7

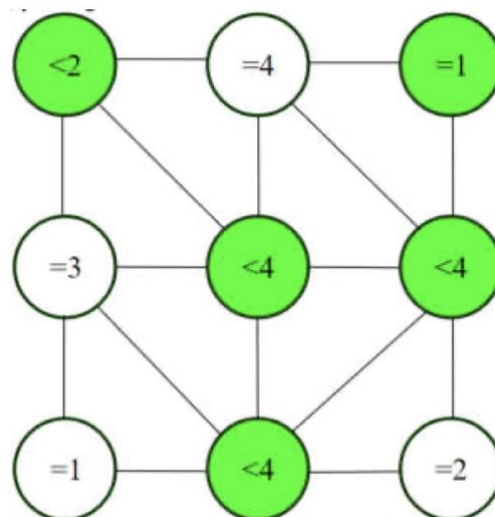
Pembahasan:

Dengan mencermati lingkaran paling kanan bawah yang berisi “=2”, kita tahu bahwa tetangganya harus diisi untuk memperoleh:



Gambar 20 Langkah menjawab
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Dengan mencermati lingkaran yang isinya “=4”, 4 tetangganya harus diisi sehingga



Gambar 21 Lanjutan langkah menjawab
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Dengan demikian, semua lingkaran sudah memenuhi kriteria. Mencermati setiap lingkaran sisanya, kita lihat bahwa lingkaran-lingkaran tersebut tidak dapat diwarnai. Secara lebih spesifik:

Jika lingkaran “=1” diwarnai, maka lingkaran “=3” menjadi salah

Jika lingkaran “=2” diwarnai, maka lingkaran “<4” di atasnya menjadi salah

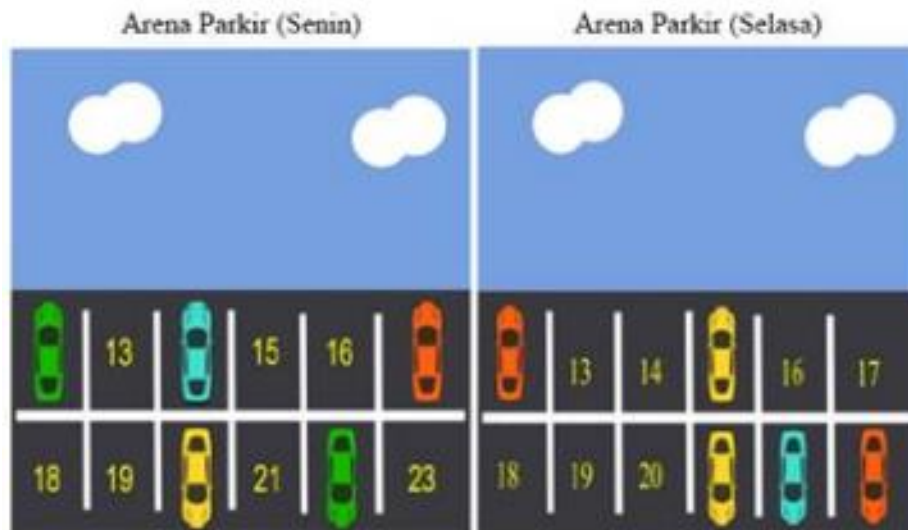
Jika lingkaran " $=3$ " diwarnai, maka lingkaran " <2 " di atasnya menjadi salah

Jika lingkaran " $=4$ " diwarnai, maka lingkaran " $=1$ " di kanannya menjadi salah

Catatan: kita juga bisa mulai dari lingkaran " $=4$ " dan akan mendapatkan hasil yang sama dengan penalaran yang sama.

Kode Soal : I-2017-CA-01
Sumber : Tantangan Bebras 2017
Judul Soal : Memindahkan Dadu
Soal :

Terdapat 12 tempat untuk parkir mobil di area parkir. Setiap tempat diberi nomor. Gambar di bawah ini menunjukkan kondisi area parkir pada hari Senin dan pada hari Selasa.



Gambar 22 Parkiran mobil
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Mobil yang parkir (pada hari Senin maupun Selasa) pasti akan parkir di tempat yang tergambar.

Tantangan : Berapa banyak tempat parkir yang tidak pernah terisi mobil pada hari Senin maupun hari Selasa?

Pilihan Jawaban :

- ☐ 3
- ☐ 5
- ☐ 4
- ☐ 6

Jawaban :

Jawaban yang tepat adalah 4.

Kita dapat melihat tempat mana yang dipakai dengan menempatkan mobil-mobil dari kedua hari di arena parkir secara bersamaan.



Gambar 23 Jawaban parkir mobil
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Kemudian kita dapat menghitung banyaknya tempat kosong untuk menentukan 4 tempat kosong pada hari Senin dan Selasa.

Ini Informatika!

Semua data dapat dipandang sebagai rangkaian nol dan satu. Setiap nol atau satu disebut sebagai bit dan rangkaian bit disebut kode biner, representasi biner atau bilangan biner.

Pada kasus sini, kita dapat memodelkan tempat bersisi mobil sebagai satu (1) dan tempat kosong sebagai nol (0); sehingga satu tempat parkir direpresentasikan dengan satu bit. Kita mendapatkan rangkaian bit jika kita memandang bahwa tempat parkir berurutan. Misalnya kita mulai dari baris atas kemudian baru baris bawah untuk memperoleh 101001001010 untuk arena parkir pada hari Senin dan 100100000111 untuk arena parkir pada hari Selasa.

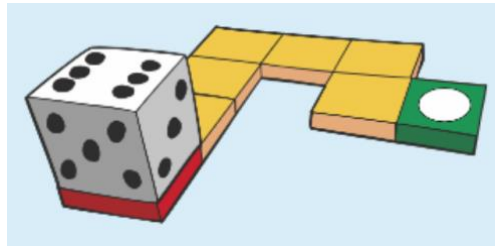
Soal ini meminta anda untuk menentukan pasangan bit mana dari kedua belas posisi pada kedua representasi biner yang keduanya bernilai nol (0).

Dalam logika, pernyataan: mencari tempat yang kosong pada (hari Senin dan hari Selasa) dapat dinegasi menjadi tempat yang isi (pada hari Senin atau hari Selasa).

Dalam hal ini, kita menentukan bit mana saja bernilai 1, dan operasi logik yang disebut OR. Perhatikan bagaimana kita dapat menghitung jawaban yang benar dengan melihat bahwa 101001001010 OR 100100000111 memberikan 10110100111. Ini memberikan hasil bilangan biner yang mempunyai 4 buah nol di dalamnya. Jadi, ada 4 tempat kosong pada hari Senin maupun juga kosong pada hari Selasa.

Kode Soal : I-2017-MY-05
Sumber : Tantangan Bebras Penggalang 2017
Judul Soal : Memindahkan Dadu
Soal :

Jack si berang-berang menggulirkan sebuah dadu sepanjang jalan tanpa pengeseran. Untuk memindahkan dadu dari satu petak ke petak berikutnya, Jack memutar dadu sepanjang pinggir yang ada di perbatasan antara dua petak. Dia melakukannya 7 kali sampai dadu mencapai petak berisi bulatan putih di sebelah kanan.



Gambar 24 Lintasan dadu
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Perhatikan bahwa banyaknya titik di sisi kebalikan sebuah dadu selalu 7 (1 berlawanan dengan 6; 2 berlawanan dengan 5; 3 berlawanan dengan 4). Pada mulanya, sisi dengan 1 titik (berlawanan dengan sisi 6) ada di dasar dadu, seperti ditunjukkan pada gambar. Setelah memutar dadu sekali ke petak kedua, sisi dengan 2 titik (berlawanan dengan 5) akan berada di dasar dadu.

Tantangan : Sisi dadu dengan berapa titik ada di dasar dadu saat dadu mencapai petak hijau di ujung?

Pilihan Jawaban :

- a. 4
- b. 5
- c. 2
- d. 6

Pembahasan :

Jawaban yang benar adalah a) 4. Salah satu cara untuk memecahkan tantangan ini adalah dengan mengamati posisi semua sisi dadu saat dadu itu berputar. Tapi sebaliknya, kita dapat mengurangi kompleksitasnya dengan mengamati 3 sisi saja, karena 3 sisi yang lainnya bisa ditentukan berdasarkan sisi yang berlawanan dengannya.

Misalnya, pertama-tama kita dapat mencatat posisi awal dadu dengan sisi 3, 5, dan 6 yang berada di posisi kanan, depan dan atas. Kita terus mengamati posisi ketiga sisi ini saat dadu

berputar, selangkah demi selangkah sampai ke petak terakhir. Setelah dadu selesai bergerak, kita bisa tahu sisi mana yang berada di bawah.

		Posisi ●●●	Posisi ●●●●●	Posisi ●●●●●●
Langkah	Arah bergerak	Kanan	Depan	Atas
1	↑	Kanan	Atas	Belakang
2	↑	Kanan	Belakang	Bawah
3	↑	Kanan	Bawah	Depan
4	→	Bawah	Kiri	Depan
5	→	Kiri	Atas	Depan
6	↓	Kiri	Depan	Bawah
7	→	Atas	Depan	Kiri

Jadi sisi yang menghadap ke bawah di petak terakhir adalah 4, karena sisi 3 berada di sebelah atas.

Ini Informatika!

Ini adalah contoh pemecahan masalah dengan mengamati hanya sebagian informasi di setiap tahap. Metode ini dapat mengurangi kerumitan dan penggunaan memori. Komputer sering diprogram seperti ini. Salah satu alternatif dari metode ini adalah melacak jumlah titik di bagian bawah setelah setiap langkah. Tetapi jauh lebih sulit. Untuk beberapa putaran pertama, itu masih mudah. Tetapi setelah beberapa langkah, kita akan mengalami kesulitan karena beberapa angka tidak dapat dilihat. Selanjutnya, jika kita memprogram dua metode ini di komputer, kita akan melihat bahwa metode pertama, seperti yang ditunjukkan dalam jawaban, lebih cepat.

Kode Soal : 2016-IR-01a
Sumber : Tantangan Bebras Penggalang 2016
Judul Soal : Mengecat Menjadi Gelap
Soal :

Kombinasi kartu A dan kartu B, menghasilkan kartu C.



Gambar 25 Kombinasi kartu
 (Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan : Berapa banyak sel gelap dari kombinasi kartu D dan kartu E berikut?



Pilihan Jawaban :

Tuliskan bilangannya sebagai jawaban.

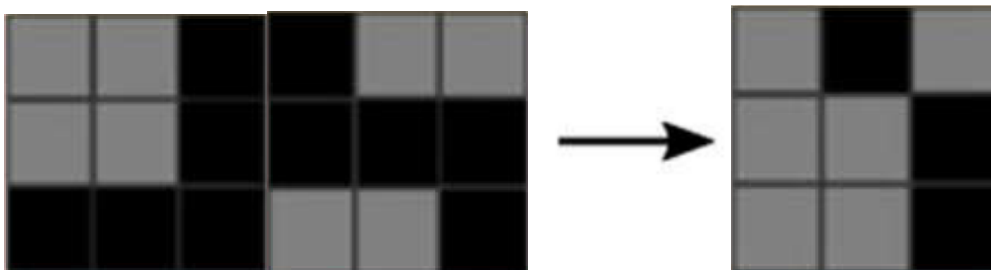
Pembahasan :

Jawaban yang benar adalah 3.

Aturan untuk melakukan kombinasi kartu D dan E adalah sebagai berikut:

- Jika warna sel kartu yang bersesuaian sama maka warna yang dihasilkan hitam
- Selainnya, warna yang dihasilkan adalah putih

Berikut ini adalah hasil kombinasi kartu D dan E:

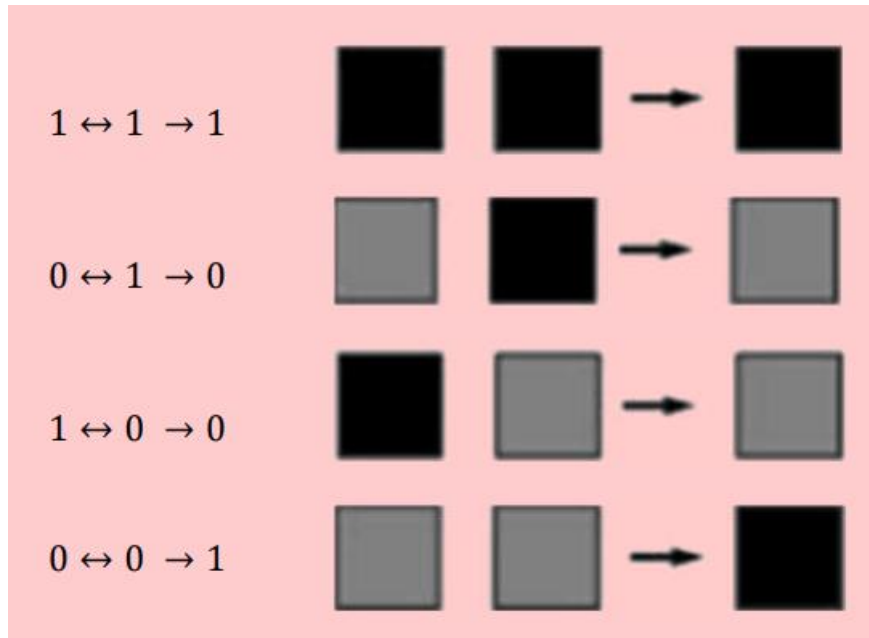


Gambar 26 Jawaban kombinasi kartu
 (Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Jawaban yang benar adalah 3.

Ini Informatika!

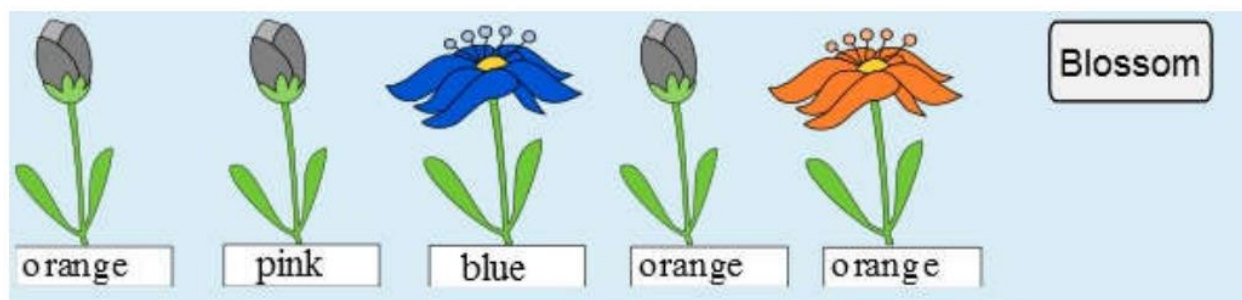
Sirkuit Boolean adalah salah satu model komputasi matematika. Ekuivalensi adalah salah satu operasi Boolean. Jika sel yang berwarna putih bernilai 0 atau SALAH dan sel yang berwarna hitam bernilai 1 atau BENAR, maka operasi ini dapat dijelaskan sebagai berikut:



Gambar 27 Pembahasan kombinasi kartu
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

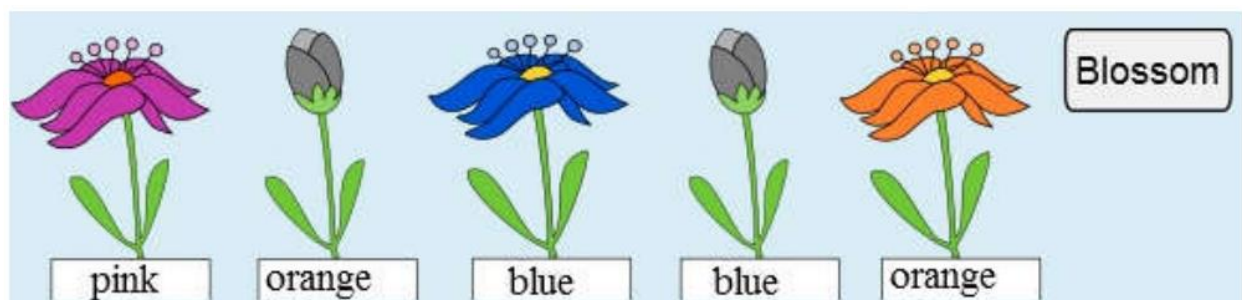
Kode Soal : 2016-SK-04
Sumber : Tantangan Bebras Penggalang 2016
Judul Soal : Warna-warni Bungabungaku
Soal :

Jane sedang bermain suatu permainan komputer. Secara rahasia komputer memilih warna-warna untuk lima kuntum bunga. Warna pilihan yang tersedia adalah biru, oranye, dan pink. Pilihan warna-warna itu tidak berubah selama satu permainan. Jane harus menebak warna-warna itu. Lalu Jane menebaknya dan oleh komputer kuntum-kuntum yang warnanya berhasil ditebak segera ditampilkan mengembang, dan yang belum berhasil ditebak tetap ditampilkan sebagai kuntum, seperti terlihat pada gambar berikut.



Gambar 28 Kondisi tebakan warna bunga
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Itu adalah hasil tebakan pertama. Berikutnya, Jane mendapat kesempatan untuk menebak kedua kalinya, dan hasil tebakannya ditampilkan komputer sebagai pada gambar berikut.



Gambar 29 Kondisi ke-2 hasil tebakan bunga
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan :

Warna-warna apakah yang telah dipilih komputer untuk kuntum-kuntum bunga tersebut?

Pilihan Jawaban :

- A. pink, biru, biru, oranye, oranye
- B. pink, biru, biru, pink, oranye
- C. pink, pink, biru, pink, oranye
- D. biru, pink, biru, oranye, oranye

Pembahasan :

Jawaban yang benar adalah B (pink, biru, biru, pink, oranye).

Setelah tiga kali tebakan ada tiga warna kuntum bunga. Dengan demikian kita sudah melihat warna yang sudah dipilih oleh komputer untuk bunga yang pertama, ketiga dan kelima. Warna bunga yang pertama adalah merah muda, maka pilihan D tidak benar. Pada tebakan yang pertama, Jane menebak bunga yang kedua pink, dan bunga tidak mengembang. Selanjutnya dia menebak oranye dan bunga juga tidak mengembang. Karena hanya ada tiga pilihan warna bunga, maka warna bunga yang kedua pasti biru. Tebakan ini menyebabkan pilihan C dan D salah. Hal yang sama juga jika Jane menebak bunga ke empat berwarna oranye atau biru, dan bunga tetap tidak mengembang. Oleh karena itu, warna bunga ke empat pasti pink. Karena C salah, maka jawaban yang benar adalah B.

Ini Informatika!

Memahami konsekuensi dari sebuah kejadian yang terjadi atau tidak adalah sangat penting untuk menyelesaikan masalah. Soal ini adalah contoh kasus sederhana dari permainan Mastermind. Setelah melakukan tebakan, pemain mendapatkan informasi yang lebih lengkap. Jika pada setiap tebakan pemain memilih warna yang berbeda untuk bunga yang belum mengembang, maka pada tebakan ketiga pemain pasti dapat menebak warna bunga dengan benar.

C.4. Soal Tingkat Sekolah Menengah Atas

Kode Soal : 2016-JP-01
Sumber : Tantangan Bebras Penegak 2016
Judul Soal : Ramuan Ajaib
Soal :

Taro si berang-berang menemukan lima jenis ramuan ajaib yang efeknya adalah sebagai berikut:

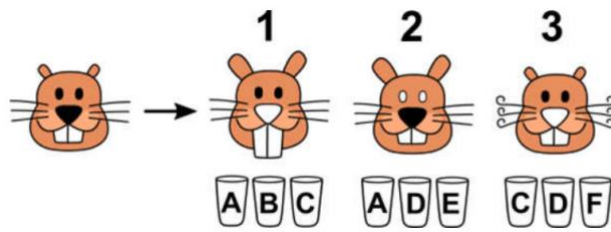
- Ramuan pertama membuat telinga bertambah panjang
- Ramuan lainnya membuat gigi bertambah panjang
- Ramuan lainnya membuat kumis menjadi keriting
- Ramuan lainnya membuat hidung menjadi putih
- Ramuan terakhir membuat mata menjadi putih Taro menaruh setiap macam ramuan ajaib tersebut dalam sebuah gelas,
- dan ada sebuah gelas yang berisi air.



Gambar 30 Gelas ramuan
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Keenam gelas tersebut diberi label A sampai dengan F. Malangnya, ia lupa mencatat gelas mana yang mengandung ramuan ajaib apa. Maka, ia mengadakan percobaan sebagai berikut untuk mengidentifikasi jenis ramuan ajaib pada setiap gelas.

- Percobaan 1
Jika ia mengambil ramuan pada gelas A,B dan C, maka efeknya adalah pada Gambar 1
- Percobaan 2
Jika ia mengambil ramuan pada gelas A,D dan E, maka efeknya adalah pada Gambar 2
- Percobaan 3
Jika ia mengambil ramuan pada gelas C, D dan F, maka efeknya adalah pada Gambar 3



Gambar 31 Hasil meminum ramuan
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan : Gelas mana yang berisi air?

Pilihan Jawaban :

- A
- B
- C
- D
- E
- F

Pembahasan:

Jawaban d. D.

Solusi 1:

- Pada percobaan pertama, gelas A, B dan C tidak ada yang berisi air, karena ada tiga perubahan pada berang-berang.
- Pada percobaan kedua, gelas D atau E berisi air atau ramuan yang membuat hidung menjadi putih. Hal ini dikarenakan dari percobaan pertama A bukan berisi air.
- Pada percobaan ketiga, gelas D dan F berisi air murni atau ramuan yang membuat kumis keriting. Hal ini dikarenakan dari percobaan pertama C bukan berisi air murni.
- Oleh karena itu dapat disimpulkan bahwa D adalah air murni.

Solusi 2:

- Percobaan 1 menyebabkan tiga perubahan,
- Percobaan 2 dan 3 menyebabkan dua perubahan.

Oleh karena itu tidak ada air murni pada percobaan 1 dan ada satu gelas pada percobaan 2 dan 3 yang berisi air.

Gelas yang sama pada percobaan 2 dan 3 adalah D, maka dengan demikian gelas D adalah air murni.

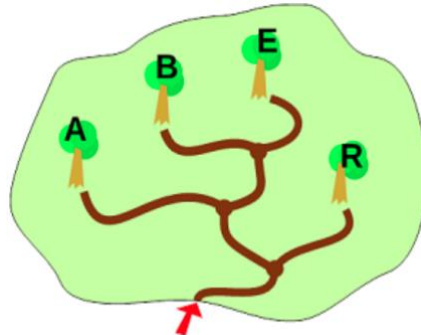
Penjelasan:

Pada soal ini kita memiliki beberapa fakta yang dapat digunakan untuk menghasilkan informasi baru dengan cara melakukan **ekstrapolasi**. Hal ini dapat dilakukan dengan menggunakan penalaran **logika** (logical reasoning). Logika memegang peranan penting di dalam ilmu komputer.

Satuan terkecil dalam komputer adalah **bit** yang memiliki nilai 1 (benar) atau 0 (salah). Semua informasi di dalam komputer disimpan dalam bentuk deretan kombinasi bit. Komputer menggunakan logika untuk mengambil keputusan dan keputusan tersebut didasarkan pada apakah bit bernilai benar (1) atau salah (0).

Soal ini juga menjelaskan **teori dasar himpunan**. Kita mencari sebuah elemen dalam himpunan yang tidak digunakan dalam percobaan 1, artinya elemen yang merupakan komplement dari A, B, C. Setelah itu mencari elemen yang merupakan irisan (elemen yang sama) pada percobaan 2 dan 3

Kode Soal : I-2017-BE-03a
Sumber : Tantangan Bebras Penegak 2017
Judul Soal : Kiri kanan
Soal :



Gambar 32 Pohon pengkodean kata
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Berang-berang menciptakan sistem pengkodean kata yang disebut kode belang-berang, dengan memakai peta di atas:

- Setiap pohon di taman diberi nama dengan satu huruf.
- Kode untuk setiap huruf ditemukan dengan cara mencapai pohon tersebut dengan berbelok kiri (L) dan kanan (R).
- Kode untuk setiap huruf selalu dimulai dari pintu masuk taman (bertanda panah).

Contoh-contoh :

- Contoh 1:
Kode untuk A adalah LL karena untuk mencapai pohon A dari pintu masuk taman kamu harus berbelok kiri dua kali.
- Contoh 2:
Kode untuk kata BAR adalah LRLLLR.

Tantangan: Berapa banyak huruf dalam kode belang-berang tersebut untuk kata BEAR?
(Tuliskan _____ angkanya!)

Pilihan Jawaban: Isian

Pembahasan: Jawaban yang benar adalah 9.

Tabel berikut ini berisi kode kiri/kanan untuk semua huruf:

B	E	A	R
LRL	LRR	LL	R

Sehingga BEAR akan dikode menjadi LRLLRLLR yang terdiri dari 9 huruf.

Penjelasan:

Jika suatu komputer mengganti huruf L dalam kode kiri/kanan dengan 0 dan huruf R dengan 1, maka kode kiri/kanan menjadi apa yang disebut kode biner. Peta untuk taman menjadi suatu struktur data komputer yang disebut pohon biner. Hal ini berarti rute yang panjang dan rumit dapat disimpan oleh komputer dengan sangat mudah menggunakan ruang yang sangat kecil. Hal menarik mengenai kode ini adalah bahwa tidak diperlukan koma atau pemisah. Cobalah mengubah kode (decode) jawaban untuk melihat bahwa kamu tidak perlu spasi untuk menunjukkan kode untuk setiap huruf cukup dan mulai dengan kode baru. Jenis kode seperti ini disebut prefix code. Hal ini berarti kode menjadi lebih singkat.

Kode Soal : I-2017-IR-07

Sumber : Tantangan Bebras Penegak 2017

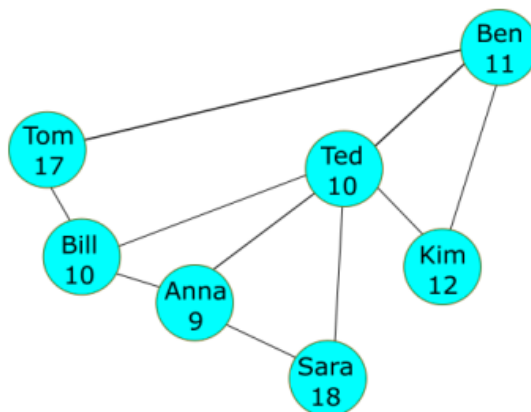
Judul Soal :

Soal :

Ada tujuh (7) siswa yang gemar membaca buku dan mereka membentuk klub untuk berbagi buku. Jika ada satu buku baru diperoleh (dan dibaca) seorang siswa, kemudian ia akan meneruskan meminjamkan ke anggota klub lainnya dengan cara berikut. Tidak setiap siswa menjadi sahabat siswa lainnya, maka seorang siswa hanya meneruskan meminjamkan buku ke siswa yang bersahabat dengannya. Jika seorang siswa mempunyai beberapa sahabat, maka sahabat yang paling muda yang akan dipinjam terlebihi dulu, yang belum pernah meminjam buku itu. Kalau semua sahabatnya sudah pernah meminjamnya, maka ia akan mengembalikan ke siswa yang sebelumnya meminjamkan buku itu kepadanya.

Diagram berikut menunjukkan tujuh siswa idan garis-garis menunjukkan hubungan "sahabat" itu. Setiap simpul berisi informasi nama dan umur.

Ben selesai membaca sebuah buku baru dan ingin berbagi dengan semua anggota klub dan selain Ben belum ada yang pernah membacanya. Siapa yang akan menjadi pembaca terakhir dari buku tersebut?



Gambar 33 Simpul teman

(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tantangan: Ben selesai membaca sebuah buku baru dan ingin berbagi dengan semua anggota klub dan selain Ben

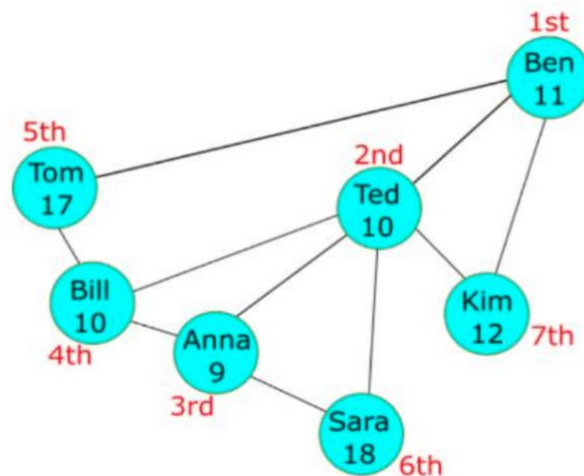
Pilihan Jawaban :

- a. Bill
- b. Sara
- c. Tom
- d. Kim

Pembahasan :

Jawaban yang benar adalah Kim

Ben meminjamkan bukunya ke Ted. Kemudian Ted memberikannya ke Anna. Anna mengoper bukunya ke Bill. Kemudian Bill meminjamkannya ke Tom. Tom mengembalikan bukunya ke Bill. Bill memberikannya kembali ke Anna. Anna meminjamkannya ke Sara. Sara kemudian mengembalikan bukunya ke Anna. Anna mengembalikannya ke Ted. Ted kemudian memberikan buku itu ke Kim.



Gambar 34 Hasil jawaban peminjaman buku
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Penjelasan :

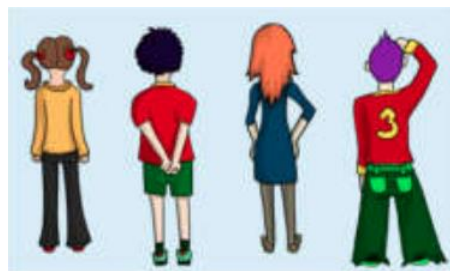
Kebanyakan dari data yang digunakan di bidang informatika saling berhubungan. Misalnya kehidupan sosial kita yang terbentuk dari jaringan elemen (orang) dan koneksi (persahabatan). Contoh lainnya adalah jaringan transportasi yang terdiri dari elemen (kota) dan koneksi (jalanan). Atau juga rantai makanan yang terdiri dari elemen (makhluk hidup) dan koneksi (produsen–konsumen). Jadi banyak segi kehidupan yang dapat dipandang sebagai “network” (jejaring).

Untuk menggambarkan jejaring, Ilmuwan/Ahli Informatika menggunakan graf sebagai struktur data, yang menggambarkan rangkaian dari hubungan. Untuk menganalisis graf tersebut, ada

bermacam-macam algoritma yang dapat dipakai mengevaluasi berbagai hal. Salah satu algoritma yang terkenal adalah Depth-First Search. Algoritma tersebut digunakan untuk mencari struktur data di grafik dalam urutan tertentu. Ini sangat berguna untuk menghitung komponen dalam grafik, misalnya ada berapa komponen grafik yang berhubungan. Depth-First dapat digunakan untuk mencari elemen yang terhubung dari suatu titik pangkal. Kalau elemen yang dicari sudah ditemukan, proses yang sama dapat diulangi untuk elemen yang lain. Cara melacak dalam sistem peminjaman buku dalam tantangan ini mirip dengan sistem pencarian Depth First.

Kode Soal : I-2018-HR-05
Sumber : Tantangan Bebras Penegak 2018
Judul Soal : Siapa Berbohong?
Soal :

Pada suatu hari yang cerah, Maya, David, Iva, dan Marko bermain sepak bola. Malangnya, salah satu melempar bola dan memecahkan kaca kelas. Bu Guru ingin tahu siapa yang menyebabkan kaca jendela tsb pecah. Bu Guru mengenal dengan baik bahwa tiga di antara anak tersebut tidak pernah bohong. Tapi ia tidak yakin siapa yang bersalah.



Gambar 35 Maya, David, Iva, dan Marko
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Anak-anak tersebut berkata secara berurutan :

- Marko : Bukan saya yang memecahkan kaca
- Iva : Marko atau David yang memecahkan kaca
- Maya : David yang memecahkan kaca
- David : bukan saya, Maya bohong!

Tantangan : Siapa yang memecahkan kaca jendela?

Pilihan Jawaban :

- a. David
- b. Marko
- c. Maya
- d. Iva

Pembahasan :

Jawaban yang benar adalah David.

Hal pertama yang kita temukan adalah bahwa pernyataan dari Maya dan David tidak bisa keduanya benar atau keduanya berbohong. Karena itu, salah satu dari mereka mengatakan yang sebenarnya, dan yang satunya berbohong. Ada dua cara yang berbeda, sama benarnya,

yang bisa kita gunakan. Penting untuk mengetahui bahwa kedua pendekatan ini bisa dipakai:

----- Pendekatan 1 -----

- a. Jika Maya mengatakan yang sebenarnya, maka hanya David yang berbohong.
- b. Jika David mengatakan yang sebenarnya, maka Maya dan salah satu dari Iva atau Marko berbohong, tetapi hanya ada satu pembohong di kelompok.

Ada dua kemungkinan (a) dan (b) bahwa David yang memecahkan jendela.

----- Pendekatan 2 -----

Atau, secara lebih umum, kita dapat menyelesaikan masalah dengan cara berikut:

- a. Jika Maya berbohong ketika mengatakan bahwa “David memecahkan jendela”, maka itu berarti yang lain harus mengatakan yang sebenarnya. (Kita tahu itu karena guru Ana mengenal murid-muridnya dan dia tahu itu mereka bertiga selalu mengatakan yang sebenarnya.) Dalam hal itu, Marko mengatakan yang sebenarnya ketika dia mengatakan bahwa dia tidak memecahkan jendela, dan sesuai dengan pernyataan Iva berarti bahwa David yang memecahkan jendela. Tapi, itu bertentangan dengan pernyataan David, yang artinya bahwa ini bukan jawaban yang benar.
- b. Jika David berbohong, maka itu berarti yang lain harus mengatakan yang sebenarnya. Kalau begitu, Marko tidak memecahkan jendela. Iva menyatakan bahwa David memecahkan jendela, dan Maya mengatakan hal yang sama, jadi ini bisa jadi adalah jawaban yang benar.

Ada dua kemungkinan (a) dan (b) bahwa David memecahkan jendela.

Penjelasan:

Dasar teori aljabar logis, yang mendasari semua pemrograman komputer, diciptakan pada 1854 oleh George Boole (1815-1864). Elemen dasar aljabar logis adalah pernyataan logis. Setiap pernyataan dapat dengan tegas ditentukan apakah itu benar atau salah.

Pernyataan benar: benar, T atau 1

Pernyataan salah: salah, F atau 0

Pernyataan-pernyataan tersebut dapat digabungkan secara bersama dalam istilah logis di mana pernyataan disebut operandi dan jenis operasi yang dilakukan antara operandi

memberitahu kita operatornya. Operasi logika dasar terdiri dari satu atau dua operandi dan satu operator sedangkan operasi logis kompleks terdiri dari logika dasar atau dasar operasi. Tabel status (tabel kebenaran) menyatakan hubungan antar operandi yang tergantung pada logika operasi. Masing-masing, berdasarkan keaslian semua pernyataan yang termasuk dalam operasi, menentukan keasliannya. Karena komputer terbuat dari sirkuit elektronik yang hanya membedakan dua kondisi stabil, Prinsip Aljabar Boolean (operasi, operan dan aturan hubungan logis) dapat diterapkan pada konstruksi dan analisis pekerjaan komputer digital.

Kode Soal : I-2018-ID-02ab
Sumber : Tantangan Bebras Penegak 2018
Judul Soal : Tugas Satu Jam
Soal :

Berang-berang si robot dapat melakukan banyak tugas. Setiap tugas membutuhkan 1, 2, 3, atau lebih jam kerja.

Dalam satu jam, si robot hanya dapat mengerjakan satu tugas.

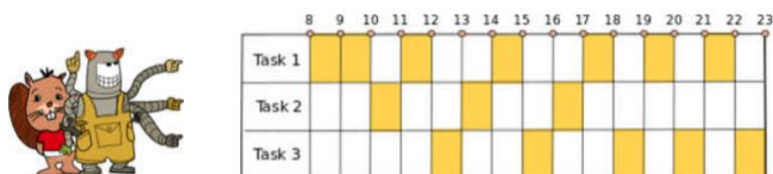
Pada akhir setiap jam, dia mengecek apakah ada sebuah tugas baru:

- Jika ya, maka si robot harus mulai mengerjakan tugas baru tsb.
- Jika tidak, si robot melanjutkan mengerjakan tugas yang paling lama tidak dikerjakannya.

Berikut ini, contoh sebuah jadwal kerja si robot dalam sehari.

- Pada pukul 8:00, ada tugas yang membutuhkan 7 jam
- Pada Pukul 10:00, datang tugas yang membutuhkan 3 jam
- Pada Pukul 12:00, datang tugas yang membutuhkan 5 jam

Pada tabel, warna kuning menunjukkan tugas tersebut sedang dikerjakan, warna putih menunjukkan tugas tersebut ditunda.



Gambar 36 Tabel jadwal robot
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Tugas-1 selesai pada Pk 22:00, Tugas-2 selesai pada Pk 17:00, dan Tugas-3 selesai pada 23:00.

Tantangan:

Jika si robot menerima empat tugas sebagai berikut:

Tugas-1: pada pk 8:00 membutuhkan 5 jam

Tugas-2: pada pk 11:00 membutuhkan 3 jam

Tugas-3: pada pk 14:00 membutuhkan 5 jam

Tugas-4: pada pk 17:00 membutuhkan 2 jam

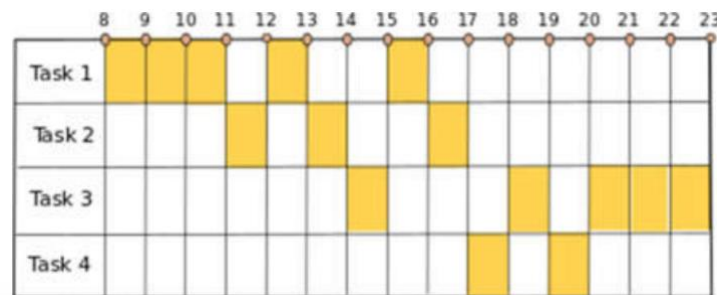
Pada jam berapa tugas akan selesai?

Pilihan Jawaban:

Isikan jawab dengan angka berupa bilangan bulat antara 0 sampai dengan 23

Pembahasan:

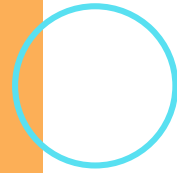
Pukul 20, dengan tabel sebagai berikut



Gambar 37 Tabel jawaban jadwal robot
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Penjelasan:

Solusinya bisa dicari dengan hanya menggunakan simulasi dan menggambarkan grafik garis waktu sesuai dengan aturan/prosedur yang diberikan. Pertanyaan ini mencoba untuk mengekspos proses manajemen 52 aktual berdasarkan Penjadwalan Round Robin dan penggunaan Gantt's Chart (kegiatan versus garis waktu).



D. Topik Tambahan

D.1. Membuat Soal *Computational*

Thinking yang Baik

(Sumber: Workshop Bebras 2019 oleh University of Waterloo, Canada)

Dalam berlatih *computational thinking* menggunakan latihan soal, diperlukan soal yang baik. Meskipun sudah banyak latihan soal yang tersedia, kita juga perlu mengetahui bagaimana cara membuat soal *computational thinking* yang baik karena :

- Mencari *task* yang baik adalah hal yang sulit dan merupakan proses yang tiada akhir
- Mendaur ulang (atau menyesuaikan tujuan kembali) bisa berguna dalam pencarian *task* yang baik
- Mengubah beberapa aspek pada *task* bisa mengubah kesulitan *task* secara signifikan.

Agar lebih jelas, berikut terdapat penjelasan mengenai hal-hal tersebut dengan contoh yang ada pada soal Bebras.

Contoh-Contoh Soal

1. Soal Bebras Kanada: "Parking Lots" (Tempat Parkir)

Soal:

Terdapat 12 tempat untuk mobil pada sebuah tempat parkir. Terdapat gambar yang menunjukkan tempat mana saja yang digunakan pada hari Senin dan tempat yang digunakan pada hari Selasa.

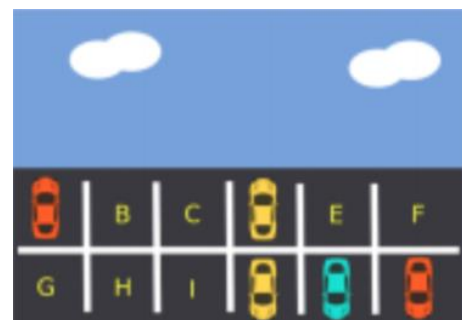
Parkiran Hari Senin



Gambar 38 Parkiran mobil

(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Parkiran Hari Selasa



Gambar 39 Parkiran hari selasa

(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Berapa tempat parkir yang tidak ditempati sama sekali pada hari Senin dan Selasa ?

- a. 3
- b. 4
- c. 5
- d. 6

2. Soal untuk CCC Junior 2 “*Occupy Parking*” (Menempati Tempat Parkir)

Deskripsi Permasalahan:

Anda mengawasi sebuah parkiran yang memiliki N buah tempat parkir.

Kemarin, Anda mencatat tempat parkir mana saja yang ditempati oleh mobil dan tempat parkir mana saja yang kosong.

Hari ini, Anda mencatat informasi yang sama.

Ada berapa tempat parkir yang sama-sama ditempati baik pada hari ini dan kemarin?

Spesifikasi input:

Baris pertama dari masukan merupakan bilangan bulat N ($1 \leq N \leq 100$). Baris kedua dan ketiga dari input berisi N buah karakter pada masing-masing baris. Baris kedua dari input merupakan informasi mengenai parkiran kemarin dan baris ketiga merupakan informasi mengenai tempat parkir hari ini. Masing-masing karakter bisa berupa “c” untuk menandakan bahwa tempat terisi atau “.” untuk menandakan bahwa tempat tersebut kosong.

Spesifikasi Output:

Output berupa angka yang merupakan jumlah tempat parkir yang sama-sama ditempati kemarin dan hari ini

Contoh input 1:

```
5
CC..C
.CC..
```

Contoh output:

1

Penjelasan:

Hanya tempat parkir kedua dari kiri yang sama-sama terisi hari ini dan kemarin

Ide kunci mengenai contoh-contoh tersebut,

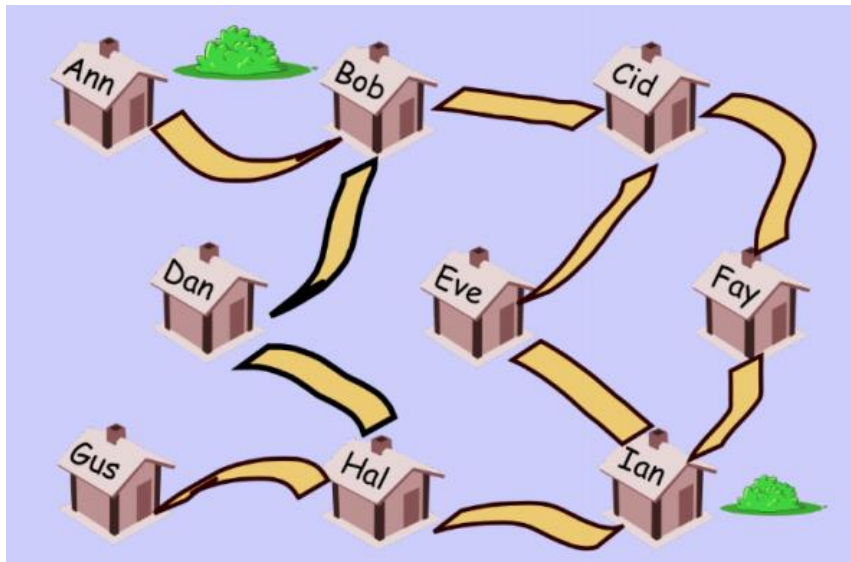
- **Generalisasi** (atau parameterisasi), yaitu kedua soal secara general merupakan soal yang sama hanya saja terdapat perbedaan parameter atau variabel yang digunakan. Perhatikan bahwa angka “5” pada Bebras Task menjadi “N” pada *computing task* (Soal CCC Junior 2). Perbedaan parameter tersebut membuat tingkat kesulitan pada kedua soal berbeda.
- Pada Buku Pembahasan Soal Bebras (bisa diakses di bebras.or.id), bagian “**It’s Informatics**” bisa digunakan untuk petunjuk bagaimana mengembangkan permasalahan agar lebih computational. Bagian tersebut menjelaskan abstraksi dari sebuah soal dengan menggunakan konsep informatika. Konsep informatika tersebut bisa digunakan sebagai dasar untuk membuat soal sejenis.
- **Tidak semua** soal Bebras digeneralisasi dengan cara yang sama seperti pada contoh sebelumnya. Hal ini bergantung kepada aspek *computational* yang digunakan pada soal. Aspek *computational* pada sebuah soal bisa saja menjadi sangat sulit sehingga sulit untuk menggeneralisasi soal. Dibutuhkan analisis yang seksama mengenai kesulitan dari permasalahan yang lebih umum.

Dalam menganalisis kesulitan sebuah soal, dalam bidang informatika terdapat teori kompleksitas komputasi. Kompleksitas komputasi terdiri dari permasalahan P (Polynomial-time algorithm) dan NP (Non Polynomial algorithm). Secara umum permasalahan P lebih mudah dikerjakan karena dapat diselesaikan dalam waktu polinom, sedangkan NP lebih rumit. Biasanya permasalahan keputusan merupakan permasalahan NO. NP sendiri terbagi lagi menjadi NP-Hard dan NP-Complete. NP-Complete merupakan permasalahan NP yang menambahkan peringatan apakah sebuah masalah NP bisa direduksi waktunya dan bisa disimulasikan untuk permasalahan yang lain. Anda tidak perlu benar-benar memahami konsep ini. Hal yang perlu diperhatikan yaitu bahwa soal pada Bebras menanyakan sebuah permasalahan yang spesifik sehingga tidak perlu memikirkan apakah algoritma dari sebuah soal bisa digunakan oleh soal yang lain karena soal pada Bebras menanyakan hal-hal yang spesifik yang bisa dipecahkan dengan percobaan.

Contoh soal:
“Firefighter”

Soal:

Walikota Beaverville sedang mencari sukarelawan pemadam kebakaran. Sebuah peta yang menunjukkan kemungkinan rumah sukarelawan dan bagaimana mereka bisa saling terhubung dengan jalan ditunjukkan pada gambar di bawah ini. Walikota ingin memastikan bahwa setiap rumah di kota merupakan rumah dari sukarelawan atau terhubung oleh sebuah jalan ke rumah sukarelawan.



Gambar 40 Kota Beaverville
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Pertanyaan:

Berapa jumlah minimal sukarelawan yang dibutuhkan oleh walikota?

- a. 1
- b. 2
- c. 3
- d. 4

Perhatikan bahwa permasalahan tersebut merupakan NP-complete: algoritma pada masalah jenis ini kemungkinan besar membutuhkan waktu eksponensial

Kita tidak perlu memikirkan permasalahannya merupakan NP-complete! Kita selalu menanyakan mengenai sebuah permasalahan yang spesifik, dibandingkan algoritma general

Contoh lain:





Soal Return Back:


Terdapat robot lebah yang memiliki 4 tombol panah pada punggungnya.

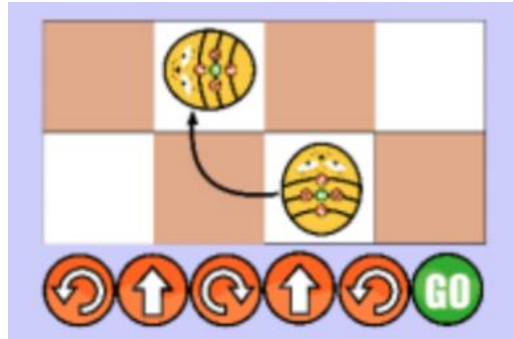


Gambar 41 Robot yang memiliki panah
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Lebah tersebut bergerak pada lantai kotak-kotak berdasarkan urutan input dengan menekan kombinasi tombol berikut:

-  Bergerak maju ke kotak berikutnya
-  hadap kiri 90° pada kotak yang sama
-  hadap kanan 90° pada kotak yang sama
-  bergerak mundur ke kotak di belakang lebah

Tombol  pada bagian punggung lebah berfungsi untuk menjalankan urutan gerakan. Contoh tombol yang ditekan dan pergerakan lebah terdapat pada gambar di bawah ini.



Gambar 42 Contoh langkah pergerakan lebah
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Lebah akan mengingat tombol-tombol yang ditekan sehingga menekan tombol “GO” kembali akan mengakibatkan pergerakan yang sama. Perlu dicatat bahwa bagaimanapun urutannya, jika menekan tombol GO berkali-kali, lebah akan kembali ke posisi awal dan menghadap arah yang sama atau tidak pernah kembali.

Terdapat sebuah angka “buzz number” yaitu jika kita bisa memasukkan urutan pergerakan yang membuat lebah kembali ke posisi awal dan arah original pada pertama kali.

Pertanyaan:

Berapa jumlah maksimum buzz number yang mungkin terjadi?

- a. 2
- b. 4
- c. 8
- d. Tidak ada jumlah maksimum

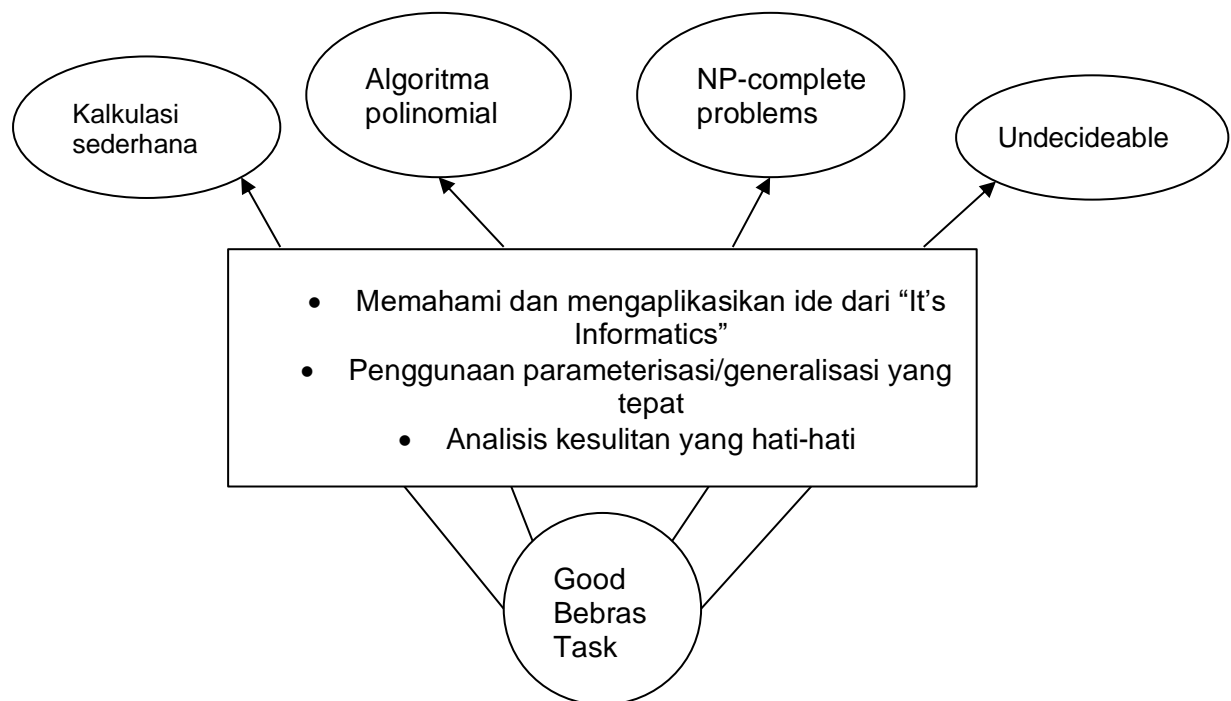
Penjelasan:

Difficulty:

Permasalahan bahkan bukan merupakan NP-complete, permasalahan pada hal ini merupakan *uncomputable* (tidak dapat ditentukan), karena permasalahan ini sama seperti Halting Problem, yang merupakan bagian dari Hilbert’s Entscheidungsproblem

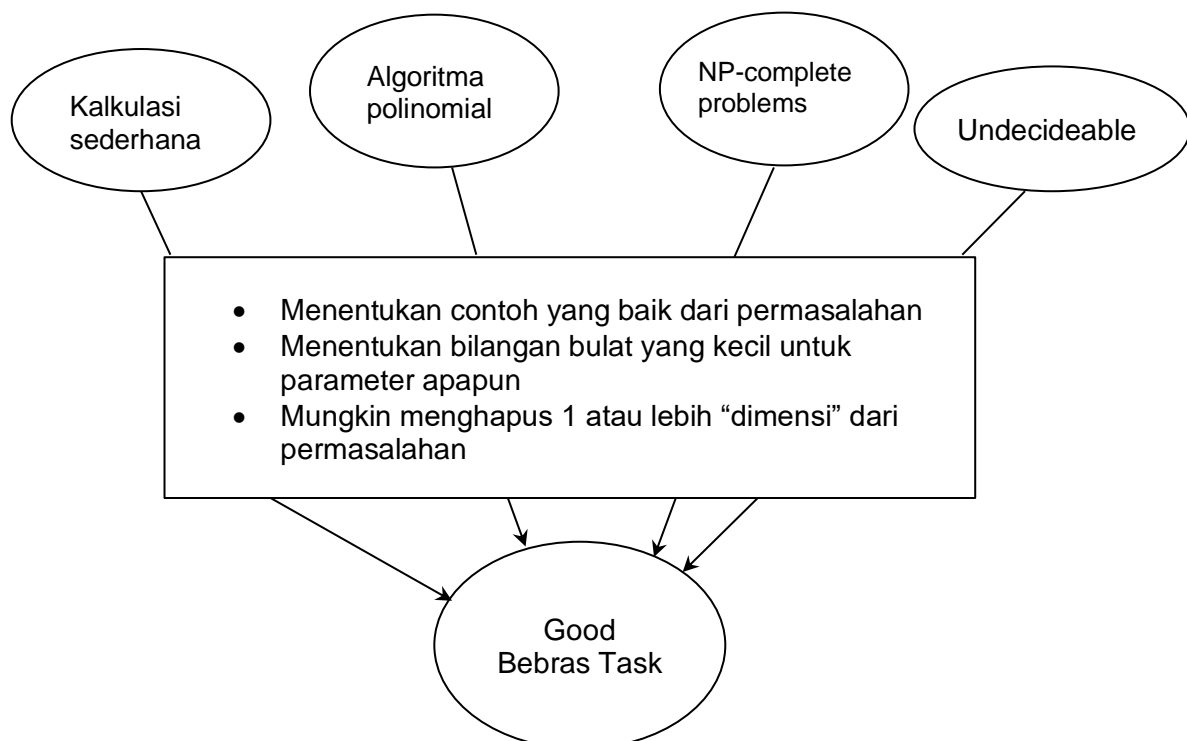
Bagaimanapun, Bebras task merupakan sebuah instance yang spesifik, yang bisa dipecahkan dengan trial-and-error/ad-hoc/hand

Jadi, berdasarkan penjelasan sebelumnya, soal Bebras sebenarnya bisa menjadi berbagai jenis soal *computational* yang berbeda.



Gambar 43 Soal bebras bisa menjadi berbagai jenis soal berbeda

Berlaku juga kebalikannya yaitu semua soal pada dasarnya bisa dimodifikasi menjadi soal Bebras.



Gambar 44 Soal apa saja bisa dimodifikasi menjadi soal Bebras

D.2. The Science of Bebras Task

(Sumber: Workshop Bebras 2019 oleh University of Waterloo, Canada)

Warm-up Problem:

Seperti apa bentuk gambar yang hilang pada pola berikut?



Gambar 45 Pola menara
(Sumber: Pretti, J.P. 2019)

Jawaban:

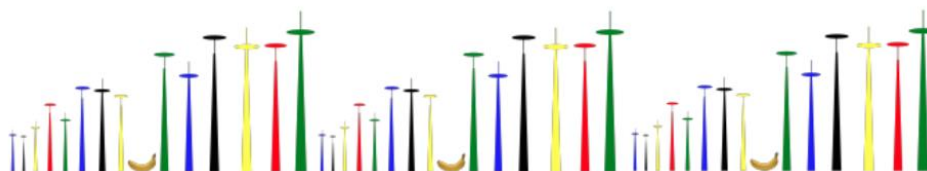
Pisang!



Gambar 46 Pisang
(Sumber: Pretti, J.P. 2019)

Mengapa pisang??

Penjelasan:



Gambar 47 Pisang yang disusun di dalam pola
(Sumber: Pretti, J.P. 2019)

Perhatikan gambar! Gambar menara masih belum cukup jelas polanya, sehingga jika dilanjutkan bisa saja benda tersebut berupa pisang. Membuat permasalahan ini menjadi

multiple choice tidak menyelesaikan permasalahan. Pisang bisa digantikan oleh *tower* apapun pada gambar.

Key point:

Permasalahan utamanya yaitu kita jangan menilai jawaban salah atau benar. Permasalahan ini merupakan latihan yang baik untuk bisa didiskusikan di kelas.

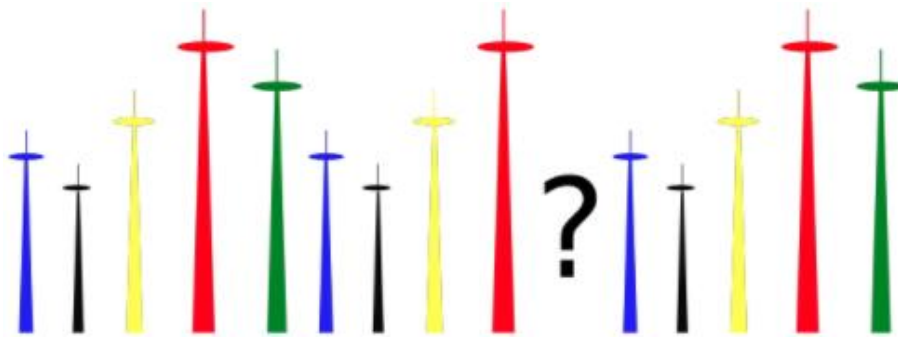
Mengapa hal ini penting?

1. Kebanyakan orang akan setuju dengan jawaban yang benar. Anak merupakan sosok yang sangat kreatif. Oleh karena itu, perlu dilakukan diskusi dengan anak mengenai seluruh jawaban yang ia kemukakan. Meskipun jawaban mereka berbeda dengan jawaban yang diharapkan, bisa saja terdapat alasan yang menarik dari jawaban mereka.
2. Kita harus bergantung ke akal pikiran. Komputer tidak memiliki akal pikiran, presisi pada komputerlah yang sangat penting. Oleh karena itu dalam menemani siswa belajar berpikir komputasi tetap diajak berdiskusi agar mereka bisa menggunakan mengemukakan isi pikiran mereka tidak hanya terpaku pada jawaban presisi.
3. Kemampuan mengidentifikasi pola merupakan kemampuan yang penting.
4. Anak-anak membutuhkan usaha yang lebih jika terdapat banyak kata-kata. Gambar yang akurat merupakan hal yang baik namun tidak bisa menggambarkan cerita keseluruhan
5. Tipe pertanyaan seperti ini muncul pada tes yang resmi. Bebras seharusnya bisa menjadi standar yang baik (gold standard)

Not all Pattern Question are Wrong

Soal sebelumnya merupakan contoh soal pola yang kurang baik. Contoh berikut merupakan contoh soal pengenalan pola yang lebih baik.

Pada gambar ini, sebuah pola diulangi lebih dari sekali. Seperti apa bentuk gambar yang hilang?



Gambar 48 Contoh gambar pola yang benar
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Examples are Examples

Dalam soal pengenalan pola, biasanya penulis soal akan memberikan contoh gambar atau data yang menunjukkan bagaimana sebuah pola terbentuk. Sebenarnya soal yang baik merupakan soal yang bisa dijawab tanpa contoh. Penulis boleh saja memberikan contoh untuk memperjelas soalnya. Namun, dalam memberikan contoh, perlu representasi yang menggambarkan aturan keseluruhan soal. Jangan sampai menimbulkan kebingungan untuk pembaca. Berikut merupakan contoh soal pengenalan pola yang kurang baik.

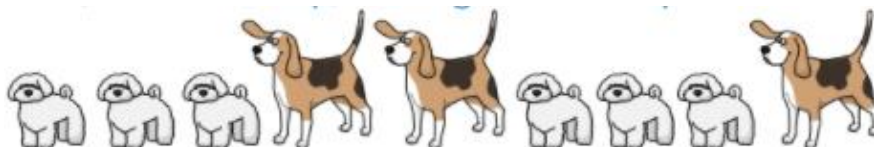
Contoh 1:

Sekumpulan anjing berbaris seperti berikut:



Gambar 49 Kumpulan anjing berbaris
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Kemudian, setelah satu kali pertukaran, sekumpulan anjing berbaris seperti berikut:



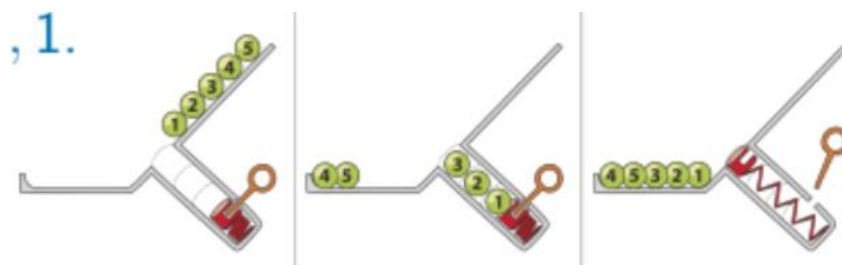
Gambar 50 Pertukaran anjing berbaris
(Sumber: <http://bebras.or.id/v3/pembahasan-soal/>)

Berapa jumlah minimal pertukaran yang dibutuhkan sehingga setiap anjing kecil berbaris di sebelah kiri anjing besar?

Permasalahan: Hanya ada satu contoh pada soal tersebut. Tidak ada contoh bagaimana pertukaran kedua harus dilakukan. Ada banyak kemungkinan yang bisa terjadi di pola kedua. Oleh karena itu, soal tersebut tidak menggambarkan suatu pola tertentu karena ketidaklengkapannya memberi contoh. Pembaca soal akan kebingungan aturan atau pola yang berlaku pada soal tersebut untuk pertukaran berikutnya karena memang tidak tergambarkan secara jelas.

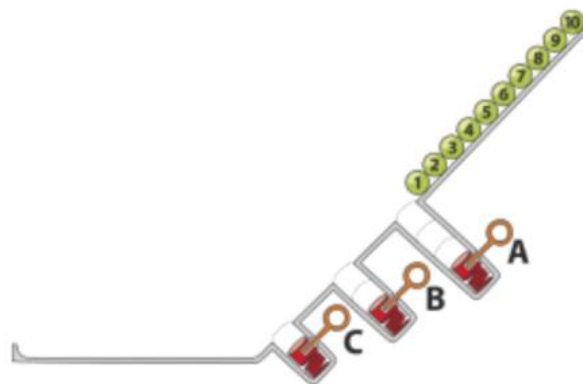
Contoh 2:

Jika bola-bola menggelinding pada lintasan berikut maka bola-bola tersebut akan terurut menjadi 4,5,3,2,1



Gambar 51 Lintasan bola
(Sumber: Pretti, J.P 2019)

Bagaimana urutan akhir dari bola-bola yang melintasi lintasan berikut?



Gambar 52 Lintasan bola dengan banyak lubang
(Sumber: Pretti, J.P 2019)

Permasalahan: Pada contoh hanya ada 1 lubang, tidak menjelaskan pola jika terdapat lebih dari 1 lubang. Pembaca tidak akan mengetahui lubang mana yang akan pertama kali mengeluarkan bola. Pada contoh hanya terdapat 1 lubang sehingga jelas sudah pasti lubang A yang melemparkan bola. Sedangkan pada soal terdapat 3 lubang yang membuat pembaca harus menerka-nerka lubang mana yang akan melemparkan bolanya terlebih dahulu, A, B, atau C.

Mengapa hal tersebut penting?

Seperangkat contoh yang terbatas tidak mendefinisikan serangkaian kasus yang tidak terbatas. Hal ini merupakan bentuk lain dari “pattern problem”.

Hal penting:

- Interaksi dengan contoh soal bisa memberikan definisi implisit
- Dalam menjawab soal, umpan balik yang diberikan untuk anak secara langsung bisa membuat anak mencoba hingga “benar”. Oleh karena itu berikan deskripsi soal yang jelas, terutama dalam pengenalan pola sehingga anak bisa lebih mudah mengidentifikasi ketika ia melakukan kesalahan dalam cara berpikir.

Task harus benar!

Soal yang digunakan untuk latihan *computational thinking* harus merupakan soal yang sudah benar dan dapat diselesaikan. Terdapat beberapa catatan untuk mendapatkan soal yang benar:

- Sebuah permasalahan pada soal pada umumnya bisa diperbaiki
- Tentukan pola pada soal. Pola pada soal harus bisa dikenali oleh pengajar, jika benar-benar tidak ada pola yang terbentuk soal tersebut perlu dievaluasi kembali.
- Periksa kembali soal yang dibuat, apakah pertanyaan bisa dijawab tanpa contoh

DAFTAR PUSTAKA

- BBC Bitesize. (2020). Computational thinking - KS3 Computing - BBC Bitesize. Retrieved June 2020, from BBC Bitesize: <https://www.bbc.co.uk/bitesize/topics/z7tp34j>
- Bebras Indonesia. (2016). Bebras Indonesia Challenge 2016 Kelompok Penegak. NBO Bebras Indonesia. Diakses dari bebras.or.id.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education*. JRC Science for Policy Report.
- CS Unplugged. (2020). Topics. Retrieved from CS Unplugged: <https://csunplugged.org/en/topics/>
- CS Unplugged. (n.d.). Computational Thinking and CS Unplugged. Retrieved from CS Unplugged: <https://csunplugged.org/en/computational-thinking/>
- Digital Promise. (2017). Computational Thinking for a Computation World. Digital Promise. Retrieved November 2019, from Computational Thinking for a Computation World
- Grover, S. (2018, February 25). The 5th 'C' of 21st Century Skills? Try Computational Thinking (Not Coding). Retrieved December 2019, from EdSurge: <https://www.edsurge.com/news/2018-02-25-the-5th-c-of-21st-century-skills-try-computational-thinking-not-coding>
- Introduction to Computational Thinking. (n.d.). *BBC Bitesize KS3 Materials*. Diakses dari <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>.
- Kidd, T., Lonnie R, & Morris, Jr. (2017). Handbook of Research on Instructional Systems and Educational Technology. United States of America: IGI Global.
- Mgova, Z. (2018). *Computational Thinking Skills In Education Curriculum*. Joensuu: UNIVERSITY OF EASTERN FINLAND,.
- Pretti, J.P. (2019) The Science of Bebras Task. Bebras Workshop 2019. Center for Education in Mathematics and Computing. University of Waterloo: Canada.
- Salatiga City. (n.d.). OZOBOT, ROBOT MUNGIL YANG MEMUDAHKAN ANDA DALAM BERAKTIVITAS. Retrieved from Salatiga City: <https://www.salatigacity.com/ozobot-robot-mungil-yang-memudahkan-anda-dalam-beraktivitas/>
- Tim Olimpiade Komputer Indonesia. (2017). Tantangan Bebras Indonesia 2017: Bahan Belajar Computational Thinking Tingkat SMA. Diakses dari bebras.or.id.
- Tim Olimpiade Komputer Indonesia. (2018). Tantangan Bebras Indonesia 2018: Bahan Belajar Computational Thinking Tingkat SMA. NBO Bebras Indonesia. Diakses dari bebras.or.id.

- Tim Olimpiade Komputer Indonesia. (2018). Tantangan Bebras Indonesia 2018: Bahan Belajar Computational Thinking Tingkat SMP. Diakses dari bebras.or.id.
- Vasiga, Troy. (2019). *How to Make a Task More Computational*. Faculty of Mathematics. University of Waterloo: Canada.
- Wing, J. (2010). Computational Thinking: What and Why? *Communications of the ACM*, CACM, 49.