



UNIVERSITAS
INDONESIA
Veritas, Probitas, Justitia

FAKULTAS
ILMU
KOMPUTER

29
NOV
2018

BROWN-BAG SESSION (BBS) #4

Computational Thinking

Konsep dan Strategi Pengembangan Soft Skills di Era Disrupsi

Viewpoint | Jeannette M. Wing

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.



Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? What can computers do better than humans? More fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the close R, what is appreciably increasing about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science tries to build theoretical underpinnings to answer such questions precisely.

Setting the difficulty of a problem accurately for the underlying power of the machine—the computing device that will run the solution. We start consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use heuristics to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It is

moderating something in anticipation of multiple users or predicting and caching in anticipation of future use.

Computational thinking is thinking in terms of prevention, protection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction. It is learning to avoid race conditions when synchronizing meetings with one another.

Computational thinking is using heuristic reasoning to discover a solution. It is planning, learning, and scheduling in the presence of uncertainty. It is search, search, and more search, resulting in a list

of solutions; when nondeterminism and garbage collection take on the meanings used by computer scientists; and when trees are drawn upside down.

We have witnessed the influence of computational thinking on other disciplines. For example, machine learning has transformed statistics. Statistical learning is being used for problems on a scale, in terms of both data size and dimension, unimaginable only a few years ago. Statistics departments in all kinds of organizations are hiring computer scientists. Schools of computer science are embracing existing or starting up new statistics departments. Computer scientists' recent interest in biology is driven by their belief that biologists can benefit

Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction.

Web pages, a strategy for winning a game, or a counterexample. Computational thinking is using massive amounts of data to speed up computation. It is making trade-offs between time and space and between processing power and storage capacity.

Consider these everyday examples. When your daughter goes to school in the morning, she puts in her backpack the things she needs for the day: that's prefetching and caching. When your son loses his mittens, you suggest he retraced his steps: that's backtracking. As when you do you stop setting the table and how you record a party, that's online algorithms. Which line do you stand in at the supermarket, that's performance modeling for multi-server systems. Why does your telephone still work during a power outage? that's independence of failure and redundancy in design. How do Completely Automated Public Turing Tests to Tell Computers and Humans Apart, or CAPTCHAs, authenticate humans, that's exploiting the difficulty of solving hard AI problems to fool computing agents.

Computational thinking will have become ingrained in everyone's lives when words like algorithm and precondition are part of everyone's vocabulary.

from computational thinking. Computer science's contribution to biology goes beyond the ability to search through vast amounts of sequence data looking for patterns. The hope is that data structures and algorithms—our computational abstractions and methods—can represent the structure of proteins in ways that elucidate their functions. Computational biology is changing the way biologists think. Similarly, computational game theory is changing the way economists think, microcomputing, the way climate scientists think, and quantum computing, the way physicists think.

This kind of thinking will be part of the skill set of not only other scientists but of everyone else. Ubiquitous computing is today's computational thinking is to tomorrow. Ubiquitous computing was yesterday's dream that became today's reality; computational thinking is tomorrow's reality.

WHAT IS, AND WON'T BE
Computer science is the study of computation—what can be computed and how to compute it. Computational thinking is the study of computation that has the following characteristics:

Pembicara : Dr. R. Yugo K. Isal & Suryana Setiawan, PhD

Lokasi: Ruang Rapat Besar Gedung A Lt 2 - Fasilkom UI

Waktu: 09.00 - 12.00 WIB

Computational Thinking saat ini mulai banyak dibicarakan kalangan pendidik. Tidak hanya yang berasal dari bidang Ilmu Komputer, tetapi juga mereka yang berasal dari bidang lain. Hal ini tidak mengherankan mengingat Computational Thinking merupakan kemampuan yang seharusnya dimiliki setiap orang berkaitan dengan Problem Solving. Kemampuan yang sangat dibutuhkan oleh generasi di Revolusi Industri 4.0.

BBS #4 Lab Digital Library & Distance Learning akan mengangkat topik Computational Thinking dengan target peserta para dosen dari bidang Science, Technology, Engineering, and Mathematics (STEM). Dosen dari luar STEM yang tertarik, silakan berpartisipasi dengan menyatakan minat dan rencana ke depan berkaitan dengan topik ini.

INFORMASI

Email: harrybs@cs.ui.ac.id

Office Phone: (021) 7863419 ext. 3200

Peserta acara ini tidak dipungut biaya

Terdapat fasilitas Snack, Makan Siang, dan Sertifikat

Registrasi ditunggu hingga 27 November 2018, Jam 13.00 WIB



DL2 LAB

<http://dl2.cs.ui.ac.id>

Formulir Registrasi: http://bit.ly/BBS_04