

PELATIHAN CODING BAGI GURU-GURU SMPN I SINGARAJA UNTUK MENINGKATKAN KETERLIBATAN SISWA DALAM PEMBELAJARAN

Gede Suweken¹, I Wayan Puja Astawa², I Nyoman Sukajaya³

^{1,2,3}Jurusan Matematika FMIPA UNDIKAHA

Email: gede.suweken@undiksha.ac.id

ABSTRACT

Teachers' main difficulties nowadays in delivering their learning process is the low students' engagement during the learning process. Teachers, as well as educational institutions, have done a lot of efforts in improving these students' low engagement, but the results are still not very encouraging. Based on students birth, our Junior High School Students are now mainly α -generation. Therefore they are mainly digital-natives. A lot of studies are also shown that characteristics gap between students, who are digital natives, and their teachers who are mainly digital immigrants, is probably the cause of this low students' engagement. So, this social responsibility activity is aimed at empowering teachers to be able to narrow this characteristics gap and in turn improve students' engagement. This is done by equipping teachers with the capability of the basics of coding so that they are able to integrate coding into their own teaching. Hopefully, this coding integration approach will be able to attract students' interest and improve their learning engagement.

Keywords: *characteristics gap, engagement, coding Scratch.*

ABSTRAK

Masalah utama pembelajaran saat ini adalah sulitnya melibatkan siswa ke dalam pembelajaran. Banyak upaya sebenarnya sudah dilakukan guru untuk meningkatkan keterlibatan ini. Namun demikian hasilnya belum terlalu signifikan, dimana-mana guru mengeluh sulitnya mengajak siswa berpikir, tidak tekun dan ingin cepat selesai, kurang konsentrasi, dan cepat lupa (daya ingatnya pendek). Dilihat dari tahun kelahiran siswa, kebanyakan siswa SMP saat ini adalah siswa generasi- α . Siswa-siswa ini adalah anak-anak yang *digital-native*. Banyak penelitian menunjukkan bahwa *characteristics gap* antara guru yang *digital immigrants* dan siswa yang *digital natives* adalah penyebab dari *engagement* siswa yang rendah ini. Untuk memperkecil gap tersebut, yang pada gilirannya akan meningkatkan *engagement* siswa dalam pembelajaran, maka kegiatan pengabdian ini diusulkan. Pada kegiatan ini, guru-guru akan dilatih memprogram computer atau *coding* dengan menggunakan bahasa pemrograman visual Scratch atau variannya. Targetnya, guru akan mampu mengintegrasikan coding ke dalam pembelajaran, dan dengan pengintegrasian ini keterlibatan siswa dapat ditingkatkan.

Kata kunci: *characteristics gap, engagement, coding Scratch.*

PENDAHULUAN

Pada *Ted Talk 2009*, Simon Sinek menyampaikan bagaimana seorang pemimpin mampu menginspirasi bawahannya, yaitu dengan bertanya "Mengapa (why?)." Ia menyatakan bahwa kebanyakan lembaga, institusi, atau perusahaan mengetahui apa dan bagaimana melakukan sesuatu sesuai bidang mereka masing-masing, namun pemimpin yang menginspirasi tidak hanya tahu apa dan bagaimana melakukan sesuatu tetapi juga tahu mengapa ia melakukan hal tersebut. Seseorang atau sebuah perusahaan akan sukses jika diawali

dengan bertanya "Why", yang tentu berkaitan dengan tujuan, keyakinan, atau alasan dari melakukan suatu kegiatan. "Why" akan mengarahkan proses untuk mencapai tujuan. Seorang guru, dapat mengimplementasikan model ini dengan bertanya "Mengapa saya memilih profesi guru?" Jawaban atas pertanyaan ini mungkin bervariasi, tetapi secara implisit pasti berisi keinginan "untuk membantu siswa sukses dan memberikan kontribusi pada kehidupan mereka nanti." Dengan memiliki alasan yang kuat "Mengapa (why)" seorang guru harus melakukan sesuatu, maka ia juga akan memiliki motivasi kuat untuk mencapainya.

Berkaitkan dengan pembelajaran, guru matematika pada umumnya telah mengetahui bahwa pemecahan masalah (*problem solving*) adalah inti dari pembelajaran matematika. Yang perlu dipahami dan disadari juga adalah “Mengapa guru harus membelajarkan siswanya tentang *problem solving*.” Pentingnya pembelajaran *problem solving* mungkin belum banyak guru yang memahaminya, dan karena itulah pembelajaran matematika sebagian besar masih didominasi rutinitas, hapalan, drill, hitung-hitungan, tak kontekstual, tak bermakna, dan tak memotivasi.

Kesadaran pentingnya *problem solving* harus senantiasa dikumandangkan, karena kompetensi inilah *core* dari pembelajaran matematika, materi matematika boleh berubah setiap saat, tetapi cara menghadapinya tetap sama, yakni kemampuan memecahkan masalah. Yang juga tidak kalah penting untuk dipikirkan adalah bagaimana pembelajaran harus dilaksanakan sehingga siswa termotivasi untuk ikut terlibat dalam proses *problem solving* tersebut. Masalah utama pembelajaran saat ini adalah masalah keterlibatan (*engagement*) siswa dalam pembelajaran. Banyak guru mengeluh tentang siswa yang tidak serius, tidak tekun, mau cepat dan jalan pintas dalam menyelesaikan masalah, tidak termotivasi, dan *daya ingatnya* pendek. Namun, terlepas dari itu semua, jika main *game* bisa seharian dengan berkonsentrasi penuh. Apa yang sebenarnya terjadi?

Dalam dunia yang berubah cepat dan tergantung pada teknologi, terutama TIK, kita semakin sadar bahwa kemampuan membaca, menulis, dan menghitung tidak lagi mencukupi bagi siswa-siswa kita untuk bisa hidup dan bersaing pada era global saat ini. Hal ini disebabkan karena masalah yang kita hadapi semakin kompleks, *ill structured*, tidak jelas dan berubah dengan cepat (*volatile*). Kemampuan calistung tentu sangat tidak memadai untuk menghadapi masalah-masalah seperti ini. Abad 21 memerlukan kompetensi 4C: *Creative and Critical thinking, Collaboration, and Communication* plus kemampuan memprogram

(*coding*) computer sebagai alat bantu bagi otak kita yang kemampuannya terbatas ini.

Sudah banyak usaha yang dilakukan pemerintah dan guru untuk mengatasi kendala tidak termotivasinya siswa dan untuk mewujudkan harapan-harapan di atas. Pengimplementasian Kurikulum 2013 sebenarnya dimaksudkan untuk mempersiapkan siswa agar memiliki kompetensi pemecahan masalah yang baik, karakter kuat dalam menghadapi dunia yang terus berubah, dan kompeten menerapkan yang telah dipelajari untuk kegiatan sehari-hari. (Harun Harosid, 2017). Dengan nafas yang sama, Kurikulum Merdeka diperkenalkan, yang ide utamanya adalah berkembangnya inovasi baik dari pihak guru maupun siswa.

Pada pihak guru, juga sudah dilakukan berbagai upaya peningkatan kompetensi melalui penataran, *upgrading*, *plpg*, dan yang lainnya. Tetapi sayang, semua upaya ini masih tetap belum memuaskan. Pada tahun 2018, hasil PISA anak-anak kita melorot lagi. Hasil yang memprihatinkan ini tentu harus dicarikan solusinya.

Sehubungan dengan ide merdeka belajar dan pembelajaran *coding* yang sudah lama diwacanakan, maka salah satu pendekatan inovatif yang memadukan peningkatan *engagement* siswa, peningkatan kemampuan *problem solving*, dan antisipasi masa depan yang semakin *computerized* adalah pengintegrasian *computational thinking and coding* ke dalam pembelajaran.

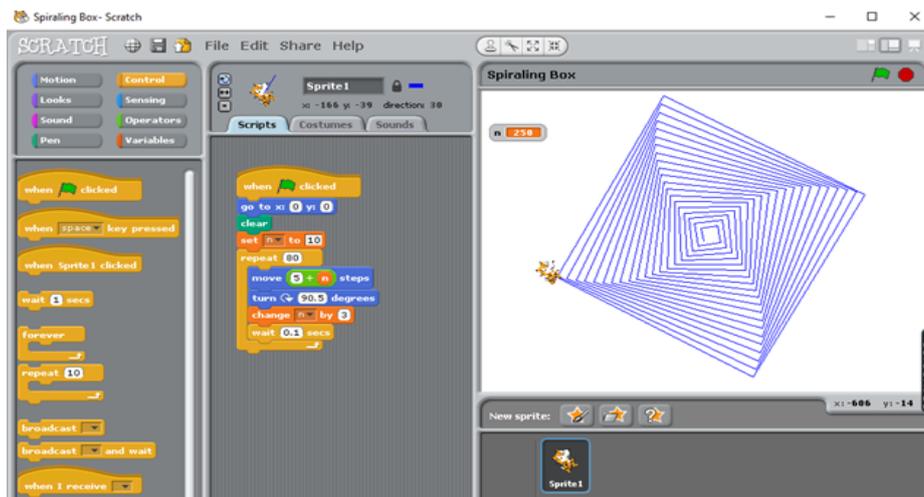
Mengapa *coding* harus dibelajarkan kepada siswa? Steve Job pernah mengatakan bahwa “*Everyone in this country should learn how to program a computer ... because it teaches you how to think.*” (youtu.be/nKIu9yen5nc). Namun sayang, proses berpikir mungkin sudah lama hilang dari pembelajaran kita, terutama matematika. Guru sering mengeluh tentang keengganan siswa diajak berpikir, untuk berusaha memahami konsep matematika yang dibelajarkan. Kelemahan siswa ini merupakan salah satu karakteristik dari generasi Z yang memenuhi

kelas-kelas kita saat ini (Prensky, 2009). Lalu bagaimana caranya agar mereka mau berpikir? Lagi-lagi cara yang paling relevan adalah dengan menyuruh mereka membuat program komputer (*coding*) untuk menyelesaikan suatu masalah. Heidi William (2017) menyatakan beberapa alasan lain *coding* relevan dibelajarkan:

- *It's about teaching perseverance.*
- *It's about teaching students how to think and reason (computational thinking).*
- *It's about creativity and expression.*
- *It's another way to demonstrate content knowledge (just like creating a PowerPoint or display board).*

• *It's a way to see math in action.*

Beberapa orang mungkin berdalih bahwa coding itu sulit. Itu tidak terjadi lagi saat ini. Dengan Bahasa pemrograman visual, memprogram menjadi tidak jauh berbeda dengan bermain puzzle. Dengan Bahasa pemrograman visual, seperti Scratch, siswa hanya perlu fokus pada pemecahan masalahnya, pada penalaran, *algoritma*, dan hanya perlu focus pada pemecahan masalah, pada penalaran, algoritma, dan pada berpikir kritis dan kreatif. Berikut adalah contoh, bagaimana Scratch digunakan untuk membuat beberapa bangun geometri. Pada Gambar 1 di bawah ini, diperlihatkan salah satu coding yang mungkin untuk membuat persegi yang sedikit diplintir (*twisted*).



Gambar 1: Spiral Persegi.

Seperti yang terlihat pada gambar di atas, aktifitas pemrograman kini tidak jauh berbeda seperti bermain puzzle. Siswa SD tentu akan termotivasi mengerjakan puzzle, selebihnya tergantung pada masalah yang harus dipecahkan dengan “bermain puzzle” tersebut.

Apa yang sebenarnya dibelajarkan dalam *coding*? Yang dibelajarkan melalui *coding* adalah *Computational Thinking (CT)* yang tidak lain adalah *problem solving* plus *coding*. CT akan membelajarkan siswa tentang:

- Logically organizing and analyzing data

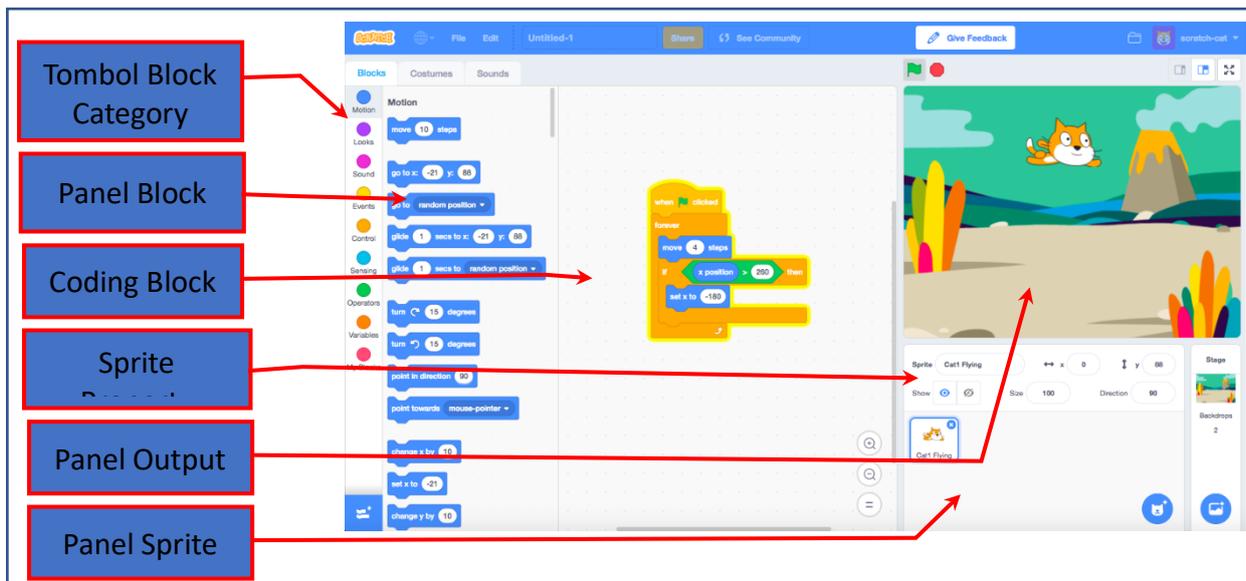
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking (a series of ordered steps),
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources,
- Generalizing and transferring this problem-solving process to a wide variety of problems, and

- Formulating problems in a way that enables us to use a computer and other tools to help solve them. (ISTE & CSTA, 2011)

Pada tahun 1991, Seymour Papert mengagas sebuah paradig pembelajaran yang disebut Konstruksionisme (*Costructionism*). Berbeda dengan konstruktivisme, konstruksionisme tidak berhenti hanya pada pentingnya siswa mengkonstruksi konsep, tetapi juga pada pentingnya siswa mengkonstruksi produk yang bermakna bagi dirinya atau orang lain dari konsep yang sudah dikuasai tersebut. Menurut aliran ini, belajar menjadi “*learning by making*.” Papert tertarik pada bagaimana siswa

terlibat pada sebuah komunikasi dengan artifak (baik itu buatan sendiri atau orang lain), dan bagaimana komunikasi ini meningkatkan belajar *self-directed*-nya, yang pada akhirnya memfasilitasi konstruksi dari suatu pengetahuan yang baru.

Papert sering mengkonstraskan konstruksionisme dan instruksionisme. Instruksionisme memusatkan perhatian pada cara mengajarkan sesuatu, sementara konstruksionisme memusatkan perhatiannya pada membelajarkan siswa agar mampu mengkonstruksi sesuatu. Keduanya penting, tetapi peningkatan kualitas pembelajaran



Gambar 2: Tampilan Scratch 3.

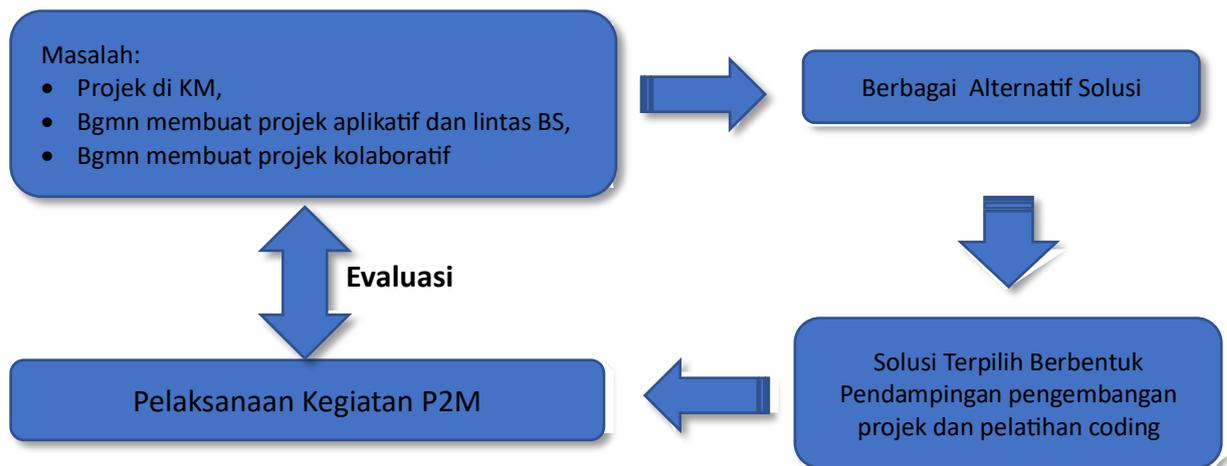
diyakini terjadi melalui konstruksionisme, bukan instruksionisme. Tantangan yang dihadapi pendidik adalah bagaimana mengembangkan alat dan lingkungan yang melibatkan siswa dalam proses konstruksi dan eksperimentasi secara intensif.

Scratch adalah salah satu *software* pemrograman yang sesuai untuk siswa SD atau SMP untuk

mengimplementasikan paham konstruksionisme. Berikut adalah tampilan *Scratch*

METODE

Secara ringkas, metode yang digunakan dalam P2M ini dapat digambarkan sebagai berikut:



Gambar 3: Diagram Kerangka Pemecahan Masalah.

Kegiatan P2M diselenggarakan secara luring di SMPN 1 Singaraja dengan rincian:

1. Ekspositori tentang cara mendisain atau mendapatkan projek yang berkaitan dengan konsep matematika,
2. Pembeajaran dan pelatihan mengcoding dengan *Scratch* melalui PBL,
3. Latihan dan diskusi pembuatan *Scratch coding* dengan materi kajian Geometri,
4. Latihan dan diskusi pembuatan coding *Scratch* dengan materi Aljabar,
5. Penugasan
6. Latihan dan diskusi pembuatan *Scratch coding* dengan materi kajian Aljabar, hasil dan pembahasan

HASIL DAN PEMBAHASAN

Pelaksanaan P2M ini dilakukan secara luring selama 1 hari pada tanggal 16 September 2024. Kegiatan kemudian dilanjutkan dengan pemberian tugas terbimbing dengan pembimbingan melalui grup WA. Kegiatan tatap

muka diisi dengan teori tentang cara meningkatkan *engagement* dalam pembelajaran terutama bagi siswa yang *digital natives*, *learning by making*, *Computational Thinking (CT)*, pengenalan *Scratch*, dan contoh pembuatan coding *Scratch*. Sementara tugas mandiri diserahkan topiknya kepada masing-masing guru.

Pada hari pertama, berbagai teori tentang karakteristik siswa *digital natives* (Prensky, 2009) disampaikan. Juga dibahas tentang paradigma pembelajaran *learning by making* dari Seymour Papert (Papert dan Harel, 2002). Selanjutnya dibahas tentang bahasa pemrograman (coding) berbasis visual yang relatif mudah digunakan dibandingkan dengan bahasa pemrograman berbasis text. Dengan *visual-based coding* siswa tidak lagi dibebani oleh sintaks program melainkan fokus pada logika dan algoritma penyelesaian masalah.

Selanjutnya materi yang disampaikan adalah berbagai perintah *Scratch* dan

pemanfaatannya sebagai alat untuk membuat coding. Secara rinci materi yang dibahas pada bagian ini adalah sebagai berikut:

1. Computational Thinking (CT),
2. Scratch dalam pengembangan CT
3. Pengenalan Menu Scratch,
4. Pengenalan sprite dan background (backdrop stage),
5. Pengenalan *Block Code*,
6. Media pembelajaran dengan Scratch,
7. Gamifikasi dengan Scratch, dan
8. Presentasi.

Terlihat guru-guru sangat antusias mengikuti kegiatan, seperti pada dokumentasi berikut:



Gambar 4: (a) Pembukaan. (b)

Berikut adalah coding yang dihasilkan oleh guru:

(a)

(b)

Jalankan coding ini, dan amati bahwa hasilnya btm. sesuai harapan. Gunakan 3 puzzle berikut yang sesuai, lalu masukkan ke lubang yang sesuai agar menghasilkan bunga dg. jumlah kelopak sesuai denoan yang dinout diawal.

Saya bisa buat bunga Dik.

Mau bungan dg. jumlah kelopak = ...?

Gambar 4: Beberapa coding yang dihasilkan guru.

SIMPULAN

Melalui kegiatan P2M ini, guru-guru SMPN I Singaraja telah mendapatkan pengetahuan tambahan tentang cara memprogram computer dengan menggunakan Bahasa pemrograman visual Scratch. Yang lebih mengesankan, kegiatan ini diikuti bukan hanya oleh guru matematika, melainkan juga oleh guru-guru bidang studi yang lain.

DAFTAR RUJUKAN

- Calao, Luis Alberto, et.al. Developing Mathematical Thinking with Scratch, An Experiment with 6th Grade Students. © Springer International Publishing Switzerland 2015, DOI: 10.1007/978-3-319-24258-32.
- Calder, Nigel. 2010. Using Scratch: An Integrated Problem Solving Approach to Mathematical Thinking. APCM 15(4)2010.
- Computing At School. 2014. Computing in the national curriculum. A guide for secondary teachers. UK: NAACE.
- Papert, Seymour. 1999. Logo Phylosophy and Implementation. Logo computer System Inc.
- Papert, Seymour and Idit Harel. 2002. Situating Constructionism. Digital Nations: MIT Media Lab.
- Prensky, Marc. 2001. Digital Natives, Digital Immigrant. (From On the Horizon (MCB University Press, Vol. 9 No. 5, October 2001)
- Prensky, Marc. 2009. Teaching Digital Natives. Partnering for Real Learning. UK: Center for Excellence in Media Praticce.
- Resnick, Mitchel, John Maloney, and Natalie Rusk. 2009. Scratch: Programming for All. Communication of the ACM, November 2009. DOI: 10.1145/1592761.1592779
- Small, Marian. 2017. Teaching Mathematical Thinking. New York: Teachers Colleges Press.
- William, Heidi. 2017. No Fear Coding. Computational Thinking Across The K5 Curriculum. USA: ISTE.