

MENINGKATKAN *ENGAGEMENT* SISWA DALAM PEMBELAJARAN DENGAN *CODING SCRATCH*

Gede Suweken

Jurusan Matematika, FMIPA, Undiksha

Email: gede.suweken@undiksha.ac.id

ABSTRACT

This article discusses about efforts that potentially can be used to improve students' learning engagement. Many teachers have difficulties in engaging students to their learning. Probably this is due to the generatin gap between teachers and their students. Students now are mainly of digital natives, while a lot of their teachers are of digital immigrants. This gap causes differences in methods, media, or tools used to learn. This article discusses one method to narrow this gap by integrating coding into the learning of mathematics where students' characteristics are taken into account. Since a lot of students as well as their teachers are not used to coding, coding language proposed is visual coding language, namely Scratch.

Keywords: *engagement, digital natives, digital immigrant, coding, visual coding language, Scratch.*

ABSTRAK

Artikel ini membahas tentang upaya-upaya yang bisa dilakukan dalam rangka meningkatkan keterlibatan siswa dalam pembelajaran. Banyak guru mengalami kesulitan dalam melibatkan siswa secara aktif dalam proses pembelajaran. Hal ini mungkin disebabkan karena *generation gap* antara siswa dan gurunya. Para siswa adalah *digital natives*, sementara gurunya *digital immigrant*. Perbedaan karakter ini menyebabkan perbedaan pilihan metode, media, maupun *tools* yang dipakai dalam pembelajaran. Artikel ini membahas salah satu cara dalam yang bisa digunakan dalam meningkatkan keterlibatan siswa dalam pembelajaran matematika dengan memperhitungkan karakteristik mereka, yaitu pengintegrasian *coding* ke dalam pembelajaran matematika. Mengingat guru maupun siswa sebagian besar tidak memiliki pengalaman dalam meng-*coding*, maka bahasa *coding* yang paling sesuai untuk digunakan adalah bahasa pemrograman visual, yakni Scratch.

Kata kunci: *keterlibatan, digital natives, digital immigrant, coding, bahasa visual, Scratch.*

PENDAHULUAN

Pendidik matematika baik nasional maupun internasional telah mengetahui bahwa pemecahan masalah (*problem solving*) adalah inti dari pembelajaran matematika. Betapa pentingnya membelajarkan siswa tentang *problem solving* mungkin belum banyak guru yang memahaminya dengan baik, dan betapa proses rutin hitung menghitung sebagian besar sudah bisa dilakukan komputer. Pemahaman inilah yang selalu harus dikumandangkan. Berikutnya, yang tidak kalah penting untuk dipikirkan adalah bagaimana pembelajaran harus dilaksanakan agar para siswa memiliki kemampuan *problem solving* tersebut. **Masalah utama pembelajaran saat ini**

adalah masalah keterlibatan (*engagement*) siswa dalam pembelajaran, khususnya siswa SMP. Hampir semua guru mengeluh tentang siswa yang tidak serius, tidak tekun, mau cepat dan jalan pintas dalam menyelesaikan masalah, tidak termotivasi, kemampuan mengingat yang sangat pendek, tapi jika main *game* bisa seharian berkonsentrasi tak lelah-lelah. Apa yang sebenarnya terjadi?

Dalam dunia yang semakin tergantung pada teknologi canggih dan senantiasa berubah ini, kemampuan membaca, menulis, dan menghitung tidak lagi mencukupi bagi anak didik kita agar bisa hidup dan bersaing pada persaingan global dewasa ini. Hal ini disebabkan karena masalah yang dihadapi

semakin kompleks, *ill structured*, dan sering tidak jelas dan berubah secara dinamik. Masalah-masalah dunia nyata yang semakin kompleks dan tidak terstruktur dengan baik dengan cara penyelesaian yang jelas, menyebabkan penguasaan kompetensi *calistung* tidak memadai lagi. Abad 21 memerlukan kompetensi 4C: *Creative and Critical thinking, Collaboration, and Communication*.

Sudah banyak usaha yang dilakukan pemerintah dan guru matematika untuk mewujudkan harapan-harapan di atas. Kurikulum sudah disempurnakan sejak tahun 2013 menjadi K-13 untuk mempersiapkan anak didik kita agar memiliki kompetensi untuk memecahkan masalah yang semakin kompleks, memiliki karakter yang baik dalam menghadapi dunia yang terus berubah, dan memiliki literasi dalam menerapkan kompetensi inti untuk kegiatan sehari-hari. (Harun Harosid, 2017). Guru juga sudah mengalami penataran, *upgrading*, PLPG, dan berbagai pelatihan lainnya untuk meningkatkan kemampuannya dalam membelajarkan siswa secara berkualitas. Tetapi hasilnya 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 *computational thinking and coding* yang diwacanakan oleh menteri Pendidikan dan Kebudayaan, yang ide utamanya adalah inovasi pada sisi guru dan siswa, maka salah satu pendekatan inovatif yang memadukan harapan menteri, peningkatan *engagement* siswa, serta peningkatan kemampuan *problem solving* danantisipasi masa depan yang semakin *computerized*, adalah pengintegrasian *computational thinking and coding* ke dalam pembelajaran matematika. Di luar negeri *computational thinking and coding* sudah

diperkenalkan sejak tahun 2010, sementara di sini sampai tahun 2020 baru terbatas wacana. Anak didik kita sudah ketinggalan 10 tahun dari dari saudara-saudara mereka diluar sana. Bagaimana mereka nantinya bisa bersaing?

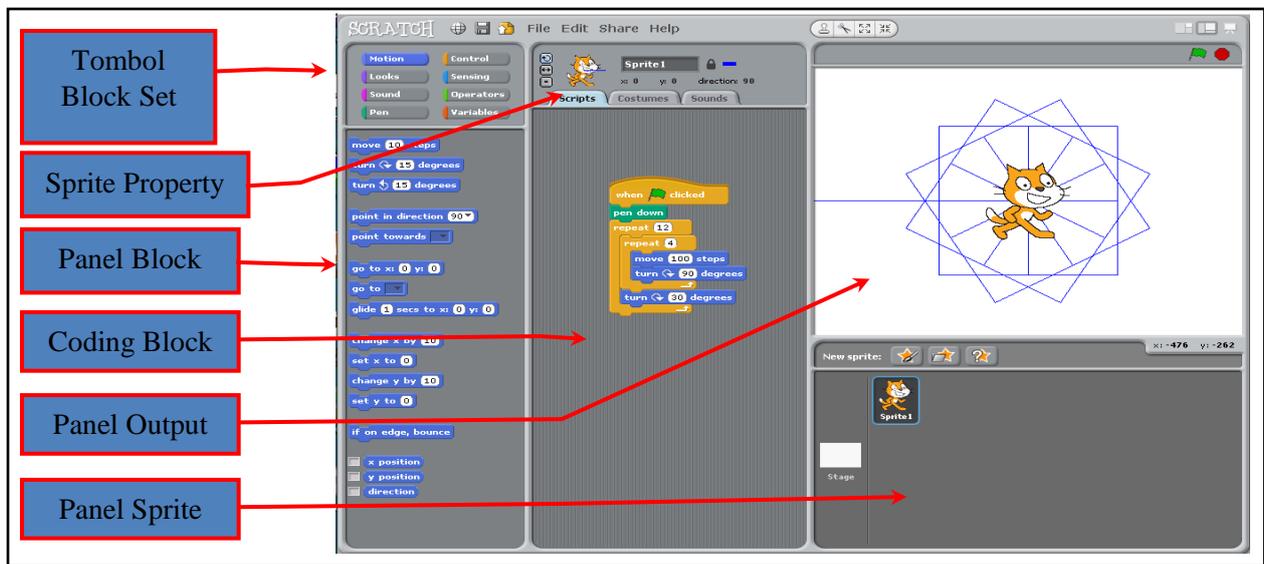
Masalah tunggal yang ingin dipecahkan melalui pengabdian ini adalah: “Bagaimana meningkatkan kompetensi guru dalam meningkatkan *engagement* siswa dalam pembelajaran matematika melalui pengembangan materi *coding* Scratch.”

METODE

1. Bahasa *coding* Scratch

Kegiatan yang digunakan untuk memecahkan masalah di atas adalah dengan menyelenggarakan pelatihan pengembangan materi dan *coding*-nya untuk pembelajaran matematika.” Kegiatan ini dipilih karena guru-guru matematika belum ada yang tahu bagaimana membuat *coding* dari materi-materi pembelajaran matematika yang mereka ajarkan, disamping juga memilih materi yang relevan untuk dibuatkan *coding*-nya.

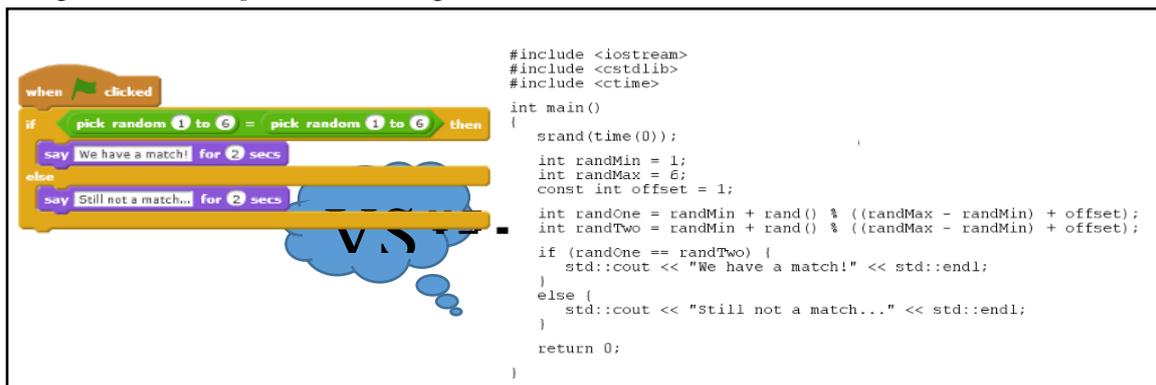
Untuk mempermudah guru dan pada gilirannya juga siswa dalam belajar dan mengembangkan *coding* dari suatu permasalahan, maka bahasan *coding* yang dipilih adalah Scratch. Bahasa *coding* ini sangat mudah dipelajari karena memang diperuntukkan untuk anak-anak (siswa). Namun walaupun demikian motto yang dipakai Scratch adalah “*low floor high ceiling*” yang maknanya adalah bahwa bahasa *coding* ini menuntut pengetahuan awal serendah mungkin, walaupun masalah-masalah yang bisa diatasinya tetap tinggi. Berikut adalah tampilan Scratch. Dengan Scratch, maka siswa dan guru bisa focus pada pengembangan solusi *coding* dari masalah yang dihadapi bukan pada sintaks *coding*-nya.



Gambar 1: Tampilan Scratch 1.4

Pada gambar di atas dapat dilihat bahwa untuk membuat *coding* dari permasalahan yang dihadapi, siswa tidak perlu mengetikkan apapun, yang diperlukan hanyalah menyeret komponen-komponen puzzle yang relevan untuk menyelesaikan masalah dari *panel block* ke *block coding*. Solusi *visual block-coding* seperti ini sangat mengurangi kendala yang biasa dihadapi pada pemrograman (*coding*) terutama bagi siswa

kelas rendah atau para pemula. *Text-based coding* sering membuat orang takut meng-*coding* karena rentan terhadap kesalahan ketik, istilah-istilah yang tidak bermakna, dan penggunaan tanda-tanda baca tertentu yang absurd. Bandingkan dua coding di bawah ini hanya untuk membandingkan apakah dua bilangan random yang dibangkitkan komputer sama atau tidak.



Gambar 2: Visual-Based VS Textual-Based

Dapat dilihat betapa tidak bermaknanya coding yang dihasilkan melalui *text-based computer language programming*, yang mana sangat berbeda dari coding yang dihasilkan oleh *Visual-Based Language*, seperti Scratch.

Materi pelatihan meliputi:

1. *Computational Thinking (CT)*,
2. Scratch dalam pengembangan CT,
3. Pengenalan *sprite* dan *backdrop*,
4. Pengenalan menu Scratch,
5. Membuat *block code*,
6. Media matematika dengan Scratch,

7. Gamifikasi dengan Scratch, dan
8. Presentasi produk.

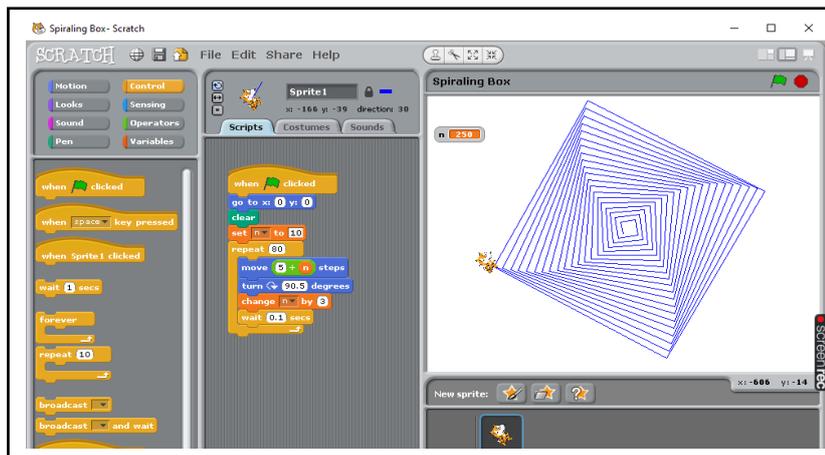
Adapun materi matematika yang potensial diintegrasikan ke *coding* Scratch meliputi Aljabar, Geometri, dan Bilangan. Pendekatan Pembelajaran Berbasis Projek (PjBL) sangat sesuai disertai dengan pengembangan produk berupa coding yang dikembangkan dengan Scratch. Sebagai dapat dilihat pada Gambar 1. Gambar tersebut sebenarnya menunjukkan sebuah persegi, yang kemudian bisa diulang-ulang

pengkonstruksianya setelah dirotasi sebesar sudut tertentu. Pada proses ini siswa akan mengimplementasikan beberapa komponen dari CT, seperti dekomposisi (analisis) masalah, pengembangan algoritma, dan abstraksi. Jika siswa melakukan kesalahan, maka dengan segera bisa dilihat hasilnya yang tidak sesuai harapan. Jadi *feedback* yang sifatnya segera benar-benar terjadi di sini. Dengan terpenuhinya balikan yang segera ini, maka

siswa akan segera dapat memperbaiki kesalahannya, untuk mencapai yang diharapkan.

2. Beberapa contoh

Berikut ditampilkan contoh *coding* yang dikembangkan melalui Scratch. Dengan contoh tersebut akan diperlihatkan bagaimana komponen-komponen CT terimplementasikan.

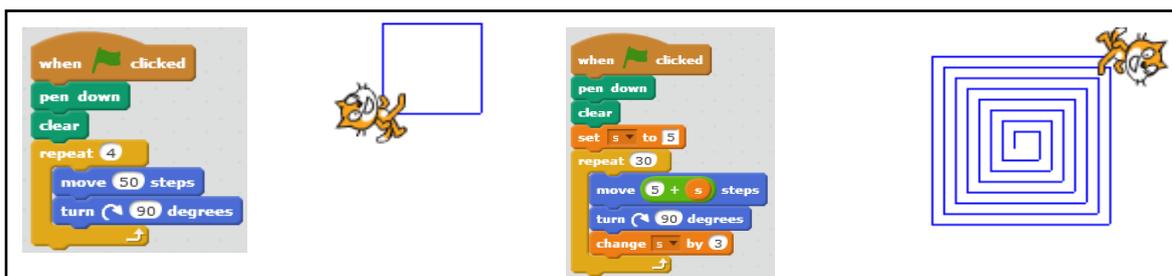


Gambar 3: *Spiraling Box*.

Computational Thinking (CT) pada dasarnya adalah “*the thought processes involved in formulating a problem and expressing its solution in a way that a computer —human or machine— can effectively carry out.*” (Janette Wing, 2006). *CT* terdiri dari komponen konsep dan proses. Komponen konsep meliputi: (i) Logika, (ii) Algoritma, (iii) Dekomposisi, (iv) Pola, (v) Abstraksi, dan (vi) Evaluasi. Sementara komponen proses meliputi (i) *Thinking*, (ii) *Creating*, (iii) *Debugging*, (iv) *Persevering*, dan (v)

Collaborating.

Pada gambar di atas dengan jelas terlihat adanya pola. Dilihat lebih cermat lagi, ternyata gambar tersebut hanyalah persegi yang diputar dan diperbesar sedikit demi sedikit (Dekomposisi). Selanjutnya adalah bagaimana algoritma yang menghasilkan gambar di atas. Dalam proses ini, komponen Abstraksi akan terjadi. Proses berpikir Logis dan Evaluasi akan senantiasa terjadi dalam rangka menghasilkan visual yang diinginkan secara bertahap.

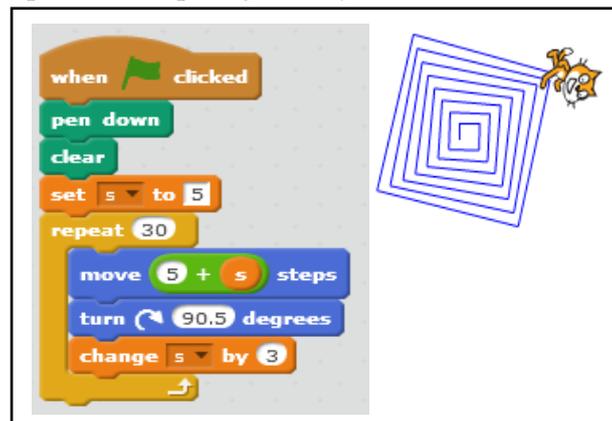


Gambar 4: Proses Abstraksi.

Visualisasi di sebelah kiri menunjukkan cara membuat persegi. Pada *coding* di sebelah kanan, variable s mulai digunakan dengan nilai awal 5. Ketika masuk loop, nilai s ini meningkat 3. Jika ini dilakukan secara berulang-ulang, maka setiap kali variable s yang merupakan sisi persegi akan memanjang 3, sehingga terjadilah spiral tegak seperti pada gambar di sebelah kanan. Ini merupakan proses abstraksi, dimana proses pembuatan persegi

dianggap sebuah objek saja, yang kemudian diperbesar terus dengan mengubah panjang sisinya s .

Tampaknya hasil yang diperoleh pada gambar di sebelah kanan tersebut terlalu biasa-biasa. Bagaimana jika kita ubah putarannya sedikit misalnya menjadi 90,5? Ini tentu akan membuat persegi menjadi sedikit berubah. Hasil yang kita peroleh menjadi cukup menarik, yakni:



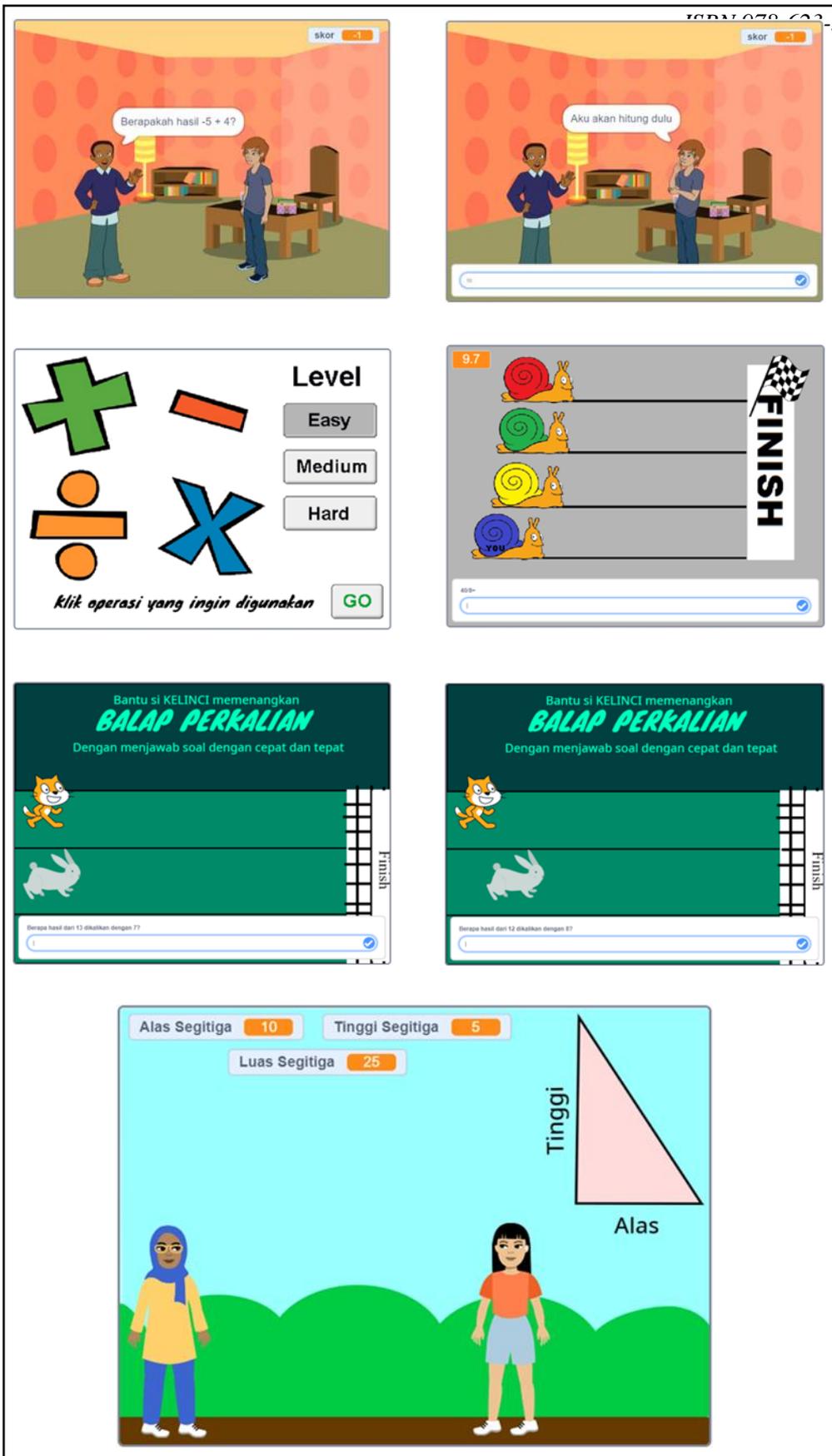
Gambar 5: *Twisted Square*.

3. Subjek P2M

Subjek yang dilibatkan dalam Pengabdian ini adalah guru-guru matematika yang menjadi anggota dari MGMP Matematika Kabupaten Buleleng. Guru yang dilibatkan ada sebanyak 25 orang dengan durasi pelatihan selama 1 bulan dari 27 Juni 2022 sampai dengan 27 Juli 2022. Pelatihan berupa penyampaian materi dan pemberian tugas dilaksanakan setiap hari Sabtu, yang kemudian

diikuti dengan kerja mandiri dan diskusi melalui group WA pada hari-hari selain Sabtu. Pada Sabtu terakhir, guru-guru diberikan kesempatan untuk mempresentasikan produk-produk yang sudah mereka kembangkan sesuai pilihan mereka masing-masing terkait materi matematika yang diajarkan.

Berikut ini adalah beberapa produk Scratch hasil kreasi guru adalah sebagai berikut:



Gambar 6: Contoh karya guru.

SIMPULAN DAN SARAN

Kegiatan pengabdian ini sudah berlangsung dengan baik dan lancar. Kehadiran peserta selama pengabdian tidak pernah berkurang. Produk yang dihasilkan juga sudah baik, walaupun masih sangat sederhana. Ada Guru yang menyarankan menggunakan kegiatan coding untuk mengisi pelajaran penguatan Pancasila dengan mengganti soal-soalnya dengan soal-soal yang berkaitan dengan Pancasila. Dengan cara seperti ini tentu saja *Computational Thinking* dan *Coding* menjadi kegiatan yang lintas mata pelajaran.

Mengingat produk yang dihasilkan masih sangat sederhana, dan itu disebabkan karena kurangnya waktu untuk penyelenggaraan kegiatan ini, maka disarankan untuk melanjutkan kegiatan ini. Kami sendiri selaku pengabdian, tetap membuka grup WA yang dibuat sebelumnya sebagai tempat diskusi, namun jika ada fasilitas yang lebih baik tentu kegiatannya akan lebih terarah.

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.
- Computing At School. 2014. *Computing in the national curriculum. A guide for secondary teachers*. UK: NAACE.
- Jones, Simon Peyton. 2009. *Computing at School: the state of the nation. A report of the Computing at School Working Group For the UK Computing Research Committee*. UK.
- Papert, Seymour and Idit Harel. 2002. *Situating Constructionism*. Digial 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
- Sweigart, Al. 2016. *Scratch Programming Playground*. San Fracisco: No Starc Press.
- William, Heidi. 2017. *No Fear Coding. Computational Thinking Across The K5 Curriculum*. USA: ISTE.