



Scratch Coding for Kids: Upaya Memperkenalkan Mathematical Thinking dan Computational Thinking pada Siswa Sekolah Dasar

Muhammad Zuhair Zahid^{a,*}, Nuriana Rachmani Dewi^a, Tri Sri Noor Asih^a,
Endang Retno Winarti^a, Tri Utami Kusuma Putri^a, Bambang Eko Susilo^a

^a Universitas Negeri Semarang, Kampus Sekaran Gunungpati, Semarang, 50229, Indonesia

* Alamat Surel: zuhairzahid@mail.unnes.ac.id

Abstrak

Computational thinking disebut-sebut sebagai salah satu kemampuan dasar yang melengkapi kemampuan membaca, menulis, dan aritmatika. Pemerintah Indonesia melalui Kemendikbud berencana mengintegrasikan *computational thinking* dalam kurikulum pendidikan dasar. Namun, di sisi lain, pengetahuan guru dan siswa tentang *computational thinking* masih sangat minim. Pengabdian ini bertujuan, (1) mensosialisasikan *Scratch Coding for Kids* sebagai sarana memperkenalkan *mathematical thinking* dan *computational thinking* sekolah dasar, dan (2) memberikan pengalaman langsung pada siswa dan guru sekolah dasar dalam mengenal konsep pemrograman sederhana yang terkait dengan konsep-konsep matematika. Metode pelaksanaan pengabdian meliputi: (1) tahap pendahuluan dan perizinan, (2) tahap sosialisasi, (3) tahap pelaksanaan yang terdiri dari dua kegiatan utama: seminar tentang *mathematical thinking* dan *computational thinking*, dan kegiatan praktik penggunaan *Scratch Coding for Kids* bagi siswa dan guru, dan (4) tahap evaluasi akhir.

Kata kunci:

PISA 2021, *computational thinking*, literasi matematis, penalaran matematis.

© 2021 Dipublikasikan oleh Jurusan Matematika, Universitas Negeri Semarang

1. Pendahuluan

Kementerian Pendidikan dan Kebudayaan (Kemendikbud) melalui Kepala Pusat Kurikulum dan Pembelajaran Awaluddin Tjalla pada acara *Grow with Google* pada 18 Februari 2020 menyatakan bahwa *computational thinking* sebagai salah satu kompetensi baru yang akan masuk dalam sistem pembelajaran anak Indonesia. Hal yang melatarbelakangi kebijakan ini adalah upaya pemerintah mempersiapkan generasi muda yang melek literasi digital.

Penyempurnaan Kurikulum 2013 melalui Permendikbud Nomor 35, 36, dan 37 Tahun 2018 dalam pertimbangannya menyebutkan pentingnya informatika untuk diintegrasikan ke struktur kurikulum 2013 di tingkat SMP dan SMA. Pada Permendikbud tersebut, secara resmi dimuat istilah *computational thinking* sebagai salah satu Kompetensi Dasar yang dipelajari dalam kurikulum Indonesia. Usaha untuk memperkenalkan *computational thinking* juga dilakukan oleh *Google* melalui *Google Educator Group* dan Bebras Indonesia melalui Bebras Challenge (“Bebras Indonesia,” 2016).

Scratch merupakan salah satu bahasa pemrograman berbasis visual yang menggunakan blok kode yang berbentuk seperti *puzzle*. *Scratch* memudahkan seseorang yang ingin mempelajari prinsip-prinsip pemrograman melalui proyek pembuatan animasi dan permainan yang menyenangkan. Dengan konsep drag and drop, programmer pemula yang bahkan belum memiliki dasar pemrograman dapat mempelajari prinsip pemrograman tanpa harus khawatir dengan penulisan sintaks. *Scratch* dikembangkan oleh MIT MEDIA

To cite this article:

Zahir, M. Z., Dewi, N. R., Asih, T. S. N., Winarti, E. R., Putri, T. U. K., & Susilo, B. E. (2021). *Scratch Coding for Kids: Upaya Memperkenalkan Mathematical Thinking dan Computational Thinking pada Siswa Sekolah Dasar*. PRISMA, Prosiding Seminar Nasional Matematika 4, 476-486

LAB dari *Massachusetts Institute of Technology* yang dirancang untuk pembelajaran anak dari umur 8 sampai 16 tahun.

Mengingat pentingnya *computational thinking* dalam dunia pendidikan, para pengabdian memandang penting memperkenalkan *computational thinking* kepada siswa-siswa sekolah dasar di Kecamatan Gunungpati sekaligus sebagai upaya persiapan siswa saat nanti *computational thinking* benar-benar diaplikasikan dalam kurikulum sekolah dasar. Para pengabdian menggunakan *Scratch* sebagai media memperkenalkan *computational thinking* mengingat *Scratch* adalah salah satu *tool* visual programming yang cukup populer saat ini.

Masalah yang dapat diidentifikasi adalah, (1) guru belum mengenal *mathematical thinking* dan *computational thinking* dengan baik, (2) proses pembelajaran yang ada belum mempertimbangkan *mathematical thinking* dan *computational thinking*, dan (3) guru dan siswa belum mengenal *Scratch* sebagai satu *tool* yang dapat dimanfaatkan untuk mengenal *mathematical thinking* dan *computational thinking*.

Dari masalah yang telah diidentifikasi, dirumuskan dua masalah yang akan dipecahkan dalam kegiatan pengabdian ini, yakni: (1) guru dan siswa belum mengenal *mathematical thinking* dan *computational thinking*, dan (2) guru dan siswa belum mengenal *Scratch* sebagai satu *tool* yang dapat dimanfaatkan untuk mengenal *mathematical thinking* dan *computational thinking*.

Kegiatan pengabdian ini memiliki dua tujuan, yakni: (1) mensosialisasikan *Scratch Coding for Kids* sebagai sarana memperkenalkan *mathematical thinking* dan *computational thinking* sekolah dasar, dan (2) memberikan pengalaman langsung pada siswa dan guru sekolah dasar dalam mengenal konsep pemrograman sederhana yang terkait dengan konsep-konsep matematika.

2. Pembahasan

2.1. Mengetahui *Scratch* dan *Computational Thinking*

Mathematical thinking jauh lebih dari sekedar kemampuan seseorang melakukan aritmatika atau memecahkan masalah aljabar. *Mathematical thinking* adalah cara menyeluruh untuk melihat berbagai hal sampai ke esensi, apakah substansi yang dikandungnya adalah substansi numerik, struktural atau logis dan kemudian menganalisis pola yang mendasarinya. Mempelajari matematika berarti mempelajari pola. Mengajar matematika berarti mengajarkan abstraksi dan pola yang terjadi setiap saat dan terjadi secara umum. Sangat penting untuk membuat siswa mampu melihat struktur-struktur dasar dari suatu substansi, apakah itu dalam masalah matematika, dalam kehidupan bermasyarakat, maupun di alam sekitar, yang menjadi salah satu alasan mengapa belajar matematika sangat bermanfaat (Vaillancourt, 2016).

Mengembangkan pemikiran matematis adalah tentang mengembangkan kebiasaan pikiran: mendefinisikan, mensistematisasikan, mengabstraksikan, membuat koneksi, mengembangkan cara-cara baru untuk menggambarkan situasi dan membuat prediksi, menciptakan, menduga-duga, dan bereksperimen (Cuoco *et al.*, 1996). Mengembangkan pemikiran matematis juga dapat dipahami dengan memasukkan imajinasi dan pengungkapan, spesialisasi dan generalisasi, dugaan dan meyakinkan, pemfokusan dan defocusing, pembuatan akal, melihat ke belakang, melepaskan diri (Mason *et al.*, 2011), dan mengartikulasikan (Lee, 2006). Fokus pada menggambarkan pemikiran matematika dalam hal proses daripada konten mengarah pada konsep literasi matematika.

Sementara itu, *computational thinking* diperkenalkan oleh Seymour Papert dan mulai kembali dikenal ketika Jeanette Wing menyebutnya sebagai salah satu kemampuan dasar yang melengkapi kemampuan membaca, menulis, dan aritmatika (Khine, 2018; Wing, 2006). Kolumnis-kolumnis bidang pendidikan secara radikal bahkan menyebutkan bahwa *computational thinking* harus masuk sebagai salah satu kemampuan wajib di abad 21 (Grover, 2018; Riddell, 2018).

Walaupun dalam ranah spesifikasi bidang akademik, *computational thinking* masuk dalam pembahasan ilmu komputer (*computer science*), namun di banyak negara, *computational thinking* sudah terintegrasi dalam kurikulum, bahkan sejak mulai pendidikan dasar, seperti Inggris yang sejak 2012 memperkenalkan *computational thinking* dalam kurikulum. Negara-negara Uni-Eropa mulai memasukkan *computational thinking* pada sekitar tahun 2016 dan 2017 (Bocconi *et al.*, 2016). Di Amerika Serikat, bentuk integrasi *computational thinking* dimulai saat munculnya gerakan *Computer Science for All* yang diinisiasi oleh Presiden Obama pada tahun 2016 (“CS for All,” n.d.). Negara-negara maju di Asia juga tidak mau ketinggalan. Mereka mengenalkan *computational thinking* dengan caranya masing-masing. Singapura, Jepang, Hong Kong, China, Taiwan, dan Malaysia adalah contoh negara-negara yang memilih

mengintegrasikan konsep pemrograman komputer dalam kurikulum pendidikan dasar (Ling *et al.*, 2018; Seow *et al.*, 2019; So *et al.*, 2020).

Scratch (<http://scratch.mit.edu>) adalah bahasa pemrograman komputer untuk anak-anak, dengan antarmuka pengguna *drag-and-drop* grafis. *Scratch* adalah turunan dari Logo, dikembangkan di MIT Media Lab. *Scratch* terus berkembang di dunia pendidikan sebagai satu upaya memperkenalkan ilmu komputer untuk jurusan selain ilmu komputer, yang juga sangat terkenal sebagai media memperkenalkan ilmu komputer di usia dini (Harvey & Mönig, 2010).

Banyaknya program yang dibuat menggunakan *Scratch* saat ini sudah mencapai lebih dari 14 juta proyek. *Scratch* adalah bahasa pemrograman berbasis blok di mana pengguna memanipulasi blok kode dalam suatu program. Bahasa berbasis blok telah ada sejak tahun delapan puluhan (Glinert, 1986), tetapi baru-baru ini saja diadaptasi sebagai alat untuk mengajarkan pemrograman. Selain *Scratch*, *Alice* (Conway *et al.*, 1994), *Blockly* dan *App Inventor* (Wolber *et al.*, 2011) adalah contoh bahasa pemrograman berbasis blok yang ditujukan untuk programmer pemula. Beberapa penelitian telah menunjukkan bahwa bahasa berbasis blok sangat sesuai digunakan sebagai alat untuk mengajarkan pemrograman (Cooper *et al.*, 2003; Meerbaum-Salant *et al.*, 2013; Moskal *et al.*, 2004; Price & Barnes, 2015).

Scratch tetap memiliki kelemahan sebagai bahasa pemrograman salah satunya adalah ia tidak mendukung prosedur sehingga *Scratch* belum dapat digunakan untuk menyampaikan fenomena rekursi yang merupakan salah satu ide sentral ilmu komputer. *Scratch* juga kurang mendukung untuk struktur data. Namun kelemahan ini bukanlah nampaknya memang disengaja oleh para perancang *Scratch*, karena mereka menghindari hal-hal yang terlalu tinggi untuk dipelajari oleh anak-anak (Harvey & Mönig, 2010).

2.2. Metode Pelaksanaan Pengabdian

Berangkat dari masalah yang dihadapi oleh guru, maka alternatif pemecahan masalah dalam kegiatan ini adalah pengenalan *mathematical thinking* dan *computational thinking* menggunakan *Scratch Coding for Kids*.

Partisipasi mitra dalam hal ini guru dan siswa SD Negeri Jatirejo tidak hanya pasif menerima pengetahuan saja, tetapi aktif melakukan sesuatu. Kegiatan disusun sedemikian sehingga guru dan siswa SD Negeri Jatirejo ikut merasakan pengalaman langsung melakukan pemrograman sederhana menggunakan *Scratch*.

Khalayak sasaran dalam kegiatan pengabdian ini adalah guru dan siswa SD Negeri Jatirejo. Letak SD Negeri Jatirejo di Desa Jatirejo yang notabene merupakan desa binaan FMIPA Universitas Negeri Semarang membuat pemilihan SD Negeri Jatirejo sebagai mitra kegiatan ini sangat relevan. Peta lokasi dapat dilihat Gambar 2 dan 3 di subbab 10.

Pihak yang terkait dengan kegiatan ini adalah (1) *sekolah* yang bertugas memfasilitasi tempat dan perlengkapan lainnya pada pelaksanaan kegiatan pengabdian, serta berkewajiban mendorong guru-guru untuk mengikuti kegiatan pengabdian, dan (2) *Fakultas MIPA Unnes* yang bertugas memberikan fasilitas SDM untuk fasilitasi kegiatan pengabdian melalui tim pengabdi.

Pelaksanaan teknis kegiatan pengabdian ini terdiri dari empat tahap:

- tahap pendahuluan dan perizinan; pada tahap ini, tim pengabdi melakukan komunikasi dengan pimpinan SD Negeri Jatirejo dan mengurus kelengkapan administrasi.
- tahap sosialisasi; pengabdi melakukan sosialisasi detail kegiatan kepada guru-guru yang terlibat, sekaligus meminta masukan dan saran terkait kegiatan yang direncanakan.
- tahap pelaksanaan; pada tahap pelaksanaan, tim pengabdi datang ke sekolah dan melaksanakan paling tidak dua kegiatan utama:
 - a. memberikan seminar tentang *mathematical thinking* dan *computational thinking*.
 - b. memandu kegiatan praktik penggunaan *Scratch Coding for Kids* bagi siswa dan guru.
- tahap evaluasi akhir; hasil isian angket dan lembar observasi digunakan untuk menganalisis dan mengevaluasi kegiatan.

Untuk mengetahui tingkat keberhasilan kegiatan ini, evaluasi akan dilaksanakan dengan dua jenis, yaitu angket dan lembar observasi. Angket digunakan untuk melihat apakah ada peningkatan pengetahuan guru dan siswa mengenai konsep dasar *mathematical thinking* dan *computational thinking*. Lembar observasi digunakan untuk melihat kemampuan guru dan siswa dalam melakukan *coding* sederhana menggunakan *Scratch*.

M. Zuhair Zahid sebagai ketua tim pengabdian merupakan dosen Jurusan Matematika UNNES yang *concern* dengan integrasi *computational thinking* dalam pembelajaran matematika. Nuriana R. D. N., Tri Sri N. A., dan Endang R. W. adalah dosen-dosen Jurusan Matematika yang memiliki kepakaran di bidang *mathematical thinking*. Dari sini dapat disimpulkan bahwa tim pengabdian memiliki kepakaran di bidang yang sesuai dengan kegiatan yang diusulkan.

2.3. Hasil pelaksanaan kegiatan pengabdian

Kegiatan dibuka oleh Kepala Sekolah SDN Jatirejo, Bapak Sumardani. Beliau menyambut kegiatan ini dengan hangat dan mempersilahkan tim pengabdian melaksanakan kegiatan sampai selesai.



Gambar 1. Kepala sekolah membuka kegiatan pengabdian

Materi pengabdian kepada masyarakat ini dibagi menjadi dua sesi. Pada sesi pertama, Dr. Bambang Eko Susilo, M.Pd., dan Dr. Nuriana R. D. N., M.Pd. memberikan materi perangkat pembelajaran matematika yang paling sesuai untuk sekolah dasar. Di sini para pengabdian menyampaikan cara memanfaatkan situs-situs yang menyediakan perangkat pembelajaran secara gratis demi memperlancar kebutuhan administratif guru tentang perangkat pembelajaran. Tercatat situs yang paling mudah diakses dan menyediakan *database* perangkat yang paling banyak adalah situs dikdasmen Kemendikbud, lebih tepatnya di laman <http://www.datadikdasmen.com/2018/03/lengkap-rpp-silabus-prota-promes.html>. Di laman ini, guru diberikan berbagai macam perangkat yang dapat diakses secara gratis, antara lain RPP, Silabus, Prota, Promes, Pemetaan KI-KD, Jurnal, KKM, Buku Guru dan Buku Siswa Kelas 1,2,3,4,5 dan 6 SD/MI Kurikulum 2013 dan Kurikulum 2006 (KTSP) Tahun Pelajaran 2020-2021.



Gambar 2. Materi dari Dr. Bambang Eko Susilo, M.Pd.

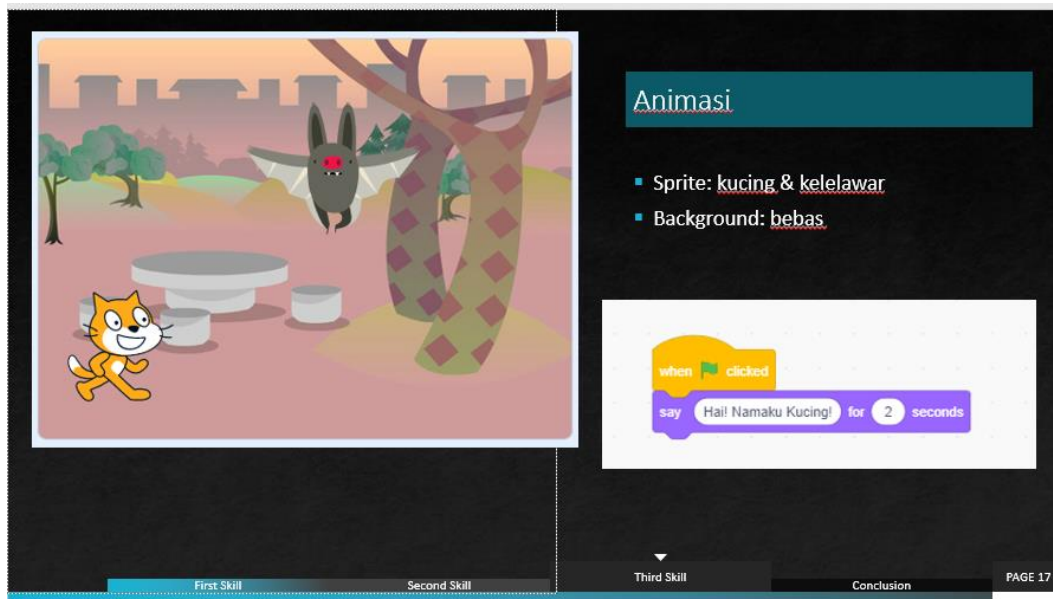


Gambar 3. Materi dari Dr. Nuriana R. D. N., M.Pd.



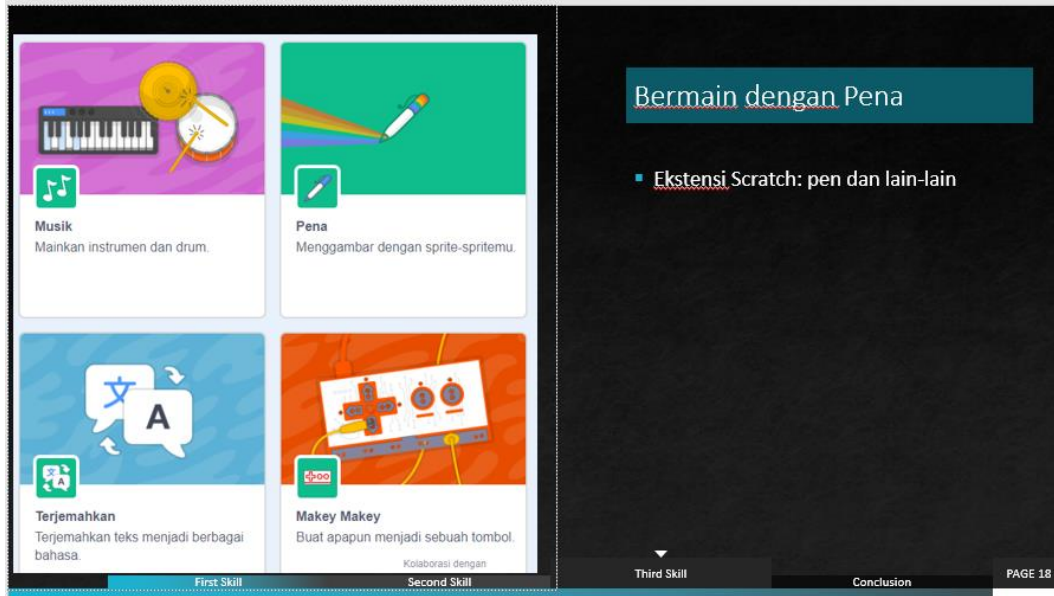
Gambar 4. Website dikkasmen

Sesi kedua berisi materi ini berupa materi pengenalan Scratch. Di sesi ini pengabdian Muhammad Zuhair Zahid mengenalkan *Scratch* kepada para guru sekaligus mengajak para guru untuk praktik menggunakan *Scratch* untuk membuat animasi-animasi sederhana yang dapat digunakan dalam pembelajaran. Animasi pertama yang dikenalkan pada para peserta merujuk pada penggunaan blok kategori "looks" yang dapat digunakan untuk membuat animasi *story telling* antar-*sprite*.



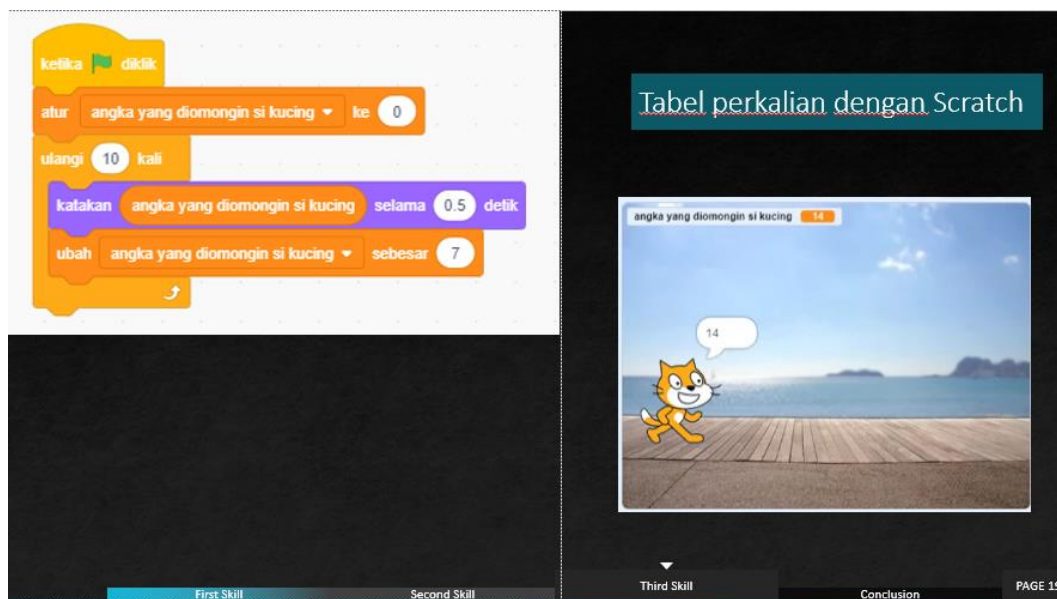
Gambar 5. Animasi *story telling* antar-*sprite*

Animasi kedua yang dibahas adalah animasi drawing yang memanfaatkan blok *add on* "pen" pada *Scratch*. Animasi yang dibuat adalah animasi gambar segi empat yang sekaligus bisa digunakan untuk mengajarkan materi sudut, sifat persegi, dan konsep *looping* dalam pemrograman.



Gambar 6. Animasi *story telling* antar-*sprite*

Hal terakhir yang dibahas adalah penggunaan looping untuk mengajarkan tabel perkalian pada siswa. Di animasi ini siswa bisa belajar tabel perkalian sekaligus mempelajari konsep perulangan dalam pemrograman.



Gambar 7. Animasi *story telling* antar-*sprite*



Gambar 8. Pengabdian memberikan materi *Scratch*

2.4. Pembahasan

Peserta diberikan materi tentang hal-hal yang berkaitan dengan perangkat pembelajaran, *mathematical thinking*, *computational thinking*, dan *Scratch*.

Setelah menerima materi dari pengabdian, diskusi kemudian terjadi antara pengabdian dengan terkait materi perangkat pembelajaran. Peserta dan pengabdian mengeksplorasi rancangan perangkat yang dijadikan sebagai pemantik diskusi. Pada saat peserta diberikan kesempatan untuk mengajukan pertanyaan, peserta terlihat sangat antusias untuk menanyakan hal-hal yang berkaitan dengan perangkat pembelajaran. Hal-hal yang muncul menjadi diskusi yang menarik adalah bagaimana membuat suatu RPP yang spesifik digunakan pada masa pandemi. RPP masa pandemi memiliki karakteristik yang berbeda jauh dengan RPP reguler mengingat pelaksanaan pembelajaran yang meminimalisir tatap muka. Terjadi diskusi yang menarik antara para pengabdian dan para peserta tentang pembuatan skema pembelajaran yang mengadopsi kondisi di masa pandemi. Dalam pendampingan pembahasan rancangan perangkat, setiap peserta ikut “urun rembuk”.

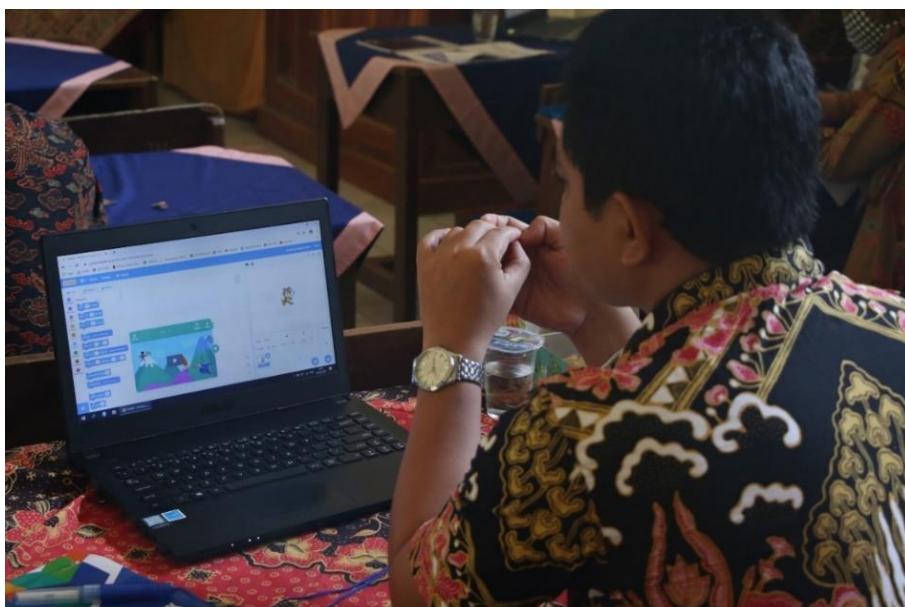


Gambar 9. Sesi diskusi dengan para peserta



Gambar 10. Sesi diskusi dengan para peserta

Terkait dengan materi *Scratch*, peserta terlihat antusias mengikuti dan mempraktekkan beberapa kode yang diberikan oleh pengabdian. Pengabdian sengaja menyiapkan beberapa buah tablet dan koneksi internet via *hotspot* portabel untuk memfasilitasi guru dalam mempraktekkan *coding Scratch*. Peserta juga saling mendiskusikan ide mereka tentang animasi dan pemrograman menggunakan *Scratch*. Setelah selesai dilaksanakan evaluasi dan refleksi yang dipandu oleh tim pengabdian pada masyarakat, disampaikan dalam evaluasi dan refleksi diperoleh kesimpulan bahwa beberapa hal yang disampaikan dalam sharing terkait beberapa kekurangan dalam materi yang diberikan, terutama karena program pengabdian ini belum berorientasi pada suatu media pembelajaran spesifik yang dibuat menggunakan *Scratch*. Di akhir kegiatan seluruh peserta menyepakati bahwa jika dalam waktu ke depan diselenggarakan kegiatan semacam program ini mereka bersedia untuk mengikuti kembali, karena kebermanfaatannya kegiatan ini.



Gambar 11. Peserta membuat animasi menggunakan *Scratch*



Gambar 12. Peserta membuat animasi menggunakan *Scratch*

3. Simpulan

Secara umum pelaksanaan pengabdian sudah berjalan dengan baik, ada rencana tindak lanjut baik dari tim pengabdian secara terbuka mempersilahkan para peserta untuk menjalin komunikasi dengan perguruan tinggi dalam hal ini Jurusan Matematika FMIPA Universitas Negeri Semarang dalam mencari solusi dari masalah pembelajaran matematika yang ditemui dalam kelas.

Daftar Pustaka

- Bebras Indonesia. (2016). Retrieved from <http://bebras.or.id/v3/>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). *Developing Computational Thinking in compulsory education, Implications for policy and practice*. Luxembourg: Publications Office of the European Union. <https://doi.org/10.2791/792158>
- Conway, M., Pausch, R., Gossweiler, R., & Burnette, T. (1994). Alice: a rapid prototyping system for building virtual environments. In *Conference Companion on Human Factors in Computing Systems* (pp. 295–296).
- Cooper, S., Dann, W., & Pausch, R. (2003). Teaching objects-first in introductory computer science. *ACM SIGCSE Bulletin*, 35(1), 191–195.
- CS for All. (n.d.). Retrieved from <https://www.csforall.org/>
- Cuoco, A., Goldenberg, E. P., & Mark, J. (1996). Habits of mind: An organizing principle for mathematics curricula. *The Journal of Mathematical Behavior*, 15(4), 375–402.
- Glinert, E. P. (1986). Towards second generation interactive graphical programming environments. In *Proceedings of IEEE Workshop on Visual Language*. IEEE CS Press, Silver Spring, MD (pp. 61–70).
- Grover, S. (2018). The 5th ‘C’ of 21st Century Skills? Try Computational Thinking (Not Coding). Retrieved from <https://www.edsurge.com/news/2018-02-25-the-5th-c-of-21st-century-skills-try-computational-thinking-not-coding>
- Harvey, B., & Mönig, J. (2010). Bringing “no ceiling” to Scratch: Can one language serve kids and computer scientists. *Proc. Constructionism*, 1–10.
- Khine, M. S. (2018). Strategies for Developing Computational Thinking. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines* (pp. 3–10). Springer, Cham.

- Lee, C. (2006). *Language for learning mathematics: assessment for learning in practice: Assessment for learning in practice*. McGraw-Hill Education (UK).
- Ling, U. L., Saibin, T. C., Naharu, N., Labadin, J., & Aziz, N. A. (2018). An evaluation tool to measure computational thinking skills: pilot investigation. *National Academy of Managerial Staff of Culture and Arts Herald*, 1, 606–614.
- Mason, J., Burton, L., & Stacey, K. (2011). *Thinking mathematically*. Pearson Higher Ed.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239–264.
- Moskal, B., Lurie, D., & Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education* (pp. 75–79).
- Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. In *Proceedings of the eleventh annual international conference on international computing education research* (pp. 91–99).
- Riddell, R. (2018). Should the 4 Cs of 21st century skills make room for one more? Retrieved from <https://www.educationdive.com/news/should-the-4-cs-of-21st-century-skills-make-room-for-one-more/517878/>
- Seow, P., Looi, C.-K., How, M.-L., Wadhwa, B., & Wu, L.-K. (2019). Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore. In S.-C. Kong & H. Abelson (Eds.), *Computational Thinking Education* (pp. 345–361). Singapore: Springer Singapore.
- So, H.-J., Jong, M. S.-Y., & Liu, C.-C. (2020). Computational Thinking Education in the Asian Pacific Region. *The Asia-Pacific Education Researcher*, 29(1), 1–8. <https://doi.org/10.1007/s40299-019-00494-w>
- Vaillancourt, D. (2016). What Is Mathematical Thinking? Retrieved from <https://drvcourt.wordpress.com/2016/07/08/what-is-mathematical-thinking/>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.
- Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2011). *App Inventor*. “O’Reilly Media, Inc.”