

MODUL AJAR KURIKULUM MERDEKA

FASE D KELAS IX

BAB 7 ALGORITMA DAN PEMROGRAMAN

INFORMASI UMUM

I. IDENTITAS MODUL

Nama Penyusun	: Mukhamad Fathoni, M.Pd.I.
Satuan Pendidikan	: MTs Nurul Huda Sukaraja
Kelas / Kelas	: IX (Sembilan)/D
Mata Pelajaran	: Informatika
Prediksi Alokasi Waktu	: 14 JP (40' x14)
Tahun Pelajaran	: 2024/2025

II. Tujuan Pembelajaran

Setelah mempelajari bab Algoritma dan Pemrograman, ini peserta didik mampu:

1. mengenali objek-objek dan memahami perintah atau instruksi dalam lingkungan pemrograman Scratch dan Blockly,
2. memahami modularisasi program dan model komputasinya,
3. mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan,
4. mengembangkan program visual yang mendukung literasi numerik dan sains maupun terwujudnya karya digital kreatif lainnya,
5. menerapkan aturan translasi konsep dari satu bahasa visual ke bahasa visual lainnya,
6. memahami Ozobot atau robot lainnya, apabila robot tersedia.

III. Kata Kunci

Pemrograman Visual; Algoritma; Pemrograman; Scratch; Blockly; Parameter; Fungsi; Prosedur; Sistem Bilangan; *Parity Bit*; Ozobot;

IV. Kaitan dengan Elemen Informatika dan Mata Pelajaran lain

Algoritma dan Pemrograman (AP) berkaitan dengan elemen lain Informatika, yaitu Sistem Komputer serta Teknologi Informasi dan Komunikasi. Pengetahuan ini juga digunakan pada Praktika Lintas Bidang Informatika dalam pengembangan artefak komputasional. Pengembangan artefak komputasional harus dikembangkan dengan praktik yang berkaitan dengan pengetahuan AP. AP akan membantu peserta didik dalam membentuk pola pikir dalam penyelesaian masalah. Algoritma dan Pemrograman (AP) juga bermanfaat ketika kalian mempelajari mata pelajaran Matematika, IPA,

V. Organisasi Pembelajaran

Tabel 7.1 Organisasi Pembelajaran Bab Algoritma dan Pemrograman

Materi	Lama Waktu (JP)	Tujuan Pembelajaran	Aktivitas
Scratch vs Blockly	2	<ol style="list-style-type: none"> Mengenali objek- objek dan memahami perintah atau instruksi dalam lingkungan pemrograman Scratch dan Blockly Menerapkan aturan translasi konsep dari satu bahasa visual ke bahasa visual lainnya 	Aktivitas AP-K9-01: Bilangan Prima
Literasi Numerik	2	<ol style="list-style-type: none"> Mengembangkan program visual sederhana berdasarkan contoh-contoh yang diberikan Mengembangkan program visual yang mendukung literasi numerik dan sains maupun terwujudnya karya digital kreatif lainnya 	Aktivitas AP-K9-02: Proyek Hitung Rata-rata dari N nilai Aktivitas AP-K9-03: Proyek Hitung Kriteria Ketuntasan Minimum dari N nilai
Modularisasi Program	4	<ol style="list-style-type: none"> Memahami modularisasi program dan model komputasinya Mengembangkan program visual sederhana berdasarkan contoh-contoh yang diberikan 	Aktivitas AP-K9-04: Modul Program: Fungsi Kuadrat Aktivitas AP-K9-05: Modul Program: Persamaan Linier Aktivitas AP-K9-06: Modul Program: Modifikasi Modul F1 Aktivitas AP-K9-07: Modul Program: Fungsi Kuadrat dan Plotting Aktivitas AP-K9-08: Modul Program: Hitung Volume Tabung

Literasi Sains	2	<ol style="list-style-type: none"> 1. Mengembangkan program visual sederhana berdasarkan contoh-contoh yang diberikan, 2. Mengembangkan program visual yang mendukung literasi numerik dan sains maupun terwujudnya karya digital kreatif lainnya, 	<p>Aktivitas AP-K9-09: Proyek Pertumbuhan 3 Hari</p> <p>Aktivitas AP-K9-10: Proyek Pertumbuhan N Hari</p> <p>Aktivitas AP-K9-11: Proyek Capaian Pertumbuhan Aktivitas AP-K9-22-U: Gelang Warna-Warni (Tambahan)</p>
Konverter Sistem Bilangan dan Parity Bit	6	<ol style="list-style-type: none"> 1. Memahami modularisasi program dan model komputasinya, 2. Mengembangkan program visual sederhana berdasarkan contoh-contoh yang diberikan 	<p>Aktivitas AP-K9-12: Konverter Bilangan Biner ke Desimal</p> <p>Aktivitas AP-K9-13: Konverter Bilangan Oktal ke Desimal</p> <p>Aktivitas AP-K9-14: Konverter Bilangan Desimal ke Biner</p> <p>Aktivitas AP-K9-15: Konverter Bilangan Desimal ke Oktal</p> <p>Aktivitas AP-K9-16-U: Pengantar Pesan Aktivitas AP-K9-17: Konverter Bilangan Desimal ke Biner dengan Parity Bit Genap</p>
			<p>Aktivitas AP-K9-18: Konverter Bilangan Biner dengan <i>Parity Bit</i> Genap ke Desimal</p> <p>Aktivitas AP-K9-19: Konverter Biner, Oktal, dan Desimal</p> <p>Aktivitas AP-K9-20: Poster Konverter Bilangan</p> <p>Aktivitas AP-K9-21: Poster Serunya Belajar Dua Bahasa</p>
Tambahan - Pemrograman dengan Ozobot	2	Memahami Ozobot atau robot lainnya, apabila robot tersedia	Aktivitas AP-K9-23: Memprogram Ozobot

VI. Pengalaman Belajar Bermakna, Proil Pelajar Pancasila, Berpikir

Komputasional, dan Praktik Inti

Tabel 7.2 Pengalaman Belajar Bermakna, Proil Pelajar Pancasila, Berikir Komputasional, dan Praktik Inti Bab Algoritma dan Pemrograman

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Core Praktik
Peserta didik mampu memahami perbedaan dan kesamaan Scratch dan Blockly.	Mandiri, Bernalar Kritis	Abstraksi, Dekomposisi, Algoritma, Pengenalan Pola	Memprogram dengan Scratch dan Blockly
Peserta didik mampu memahami modularisasi program.	Mandiri, Bernalar Kritis	Abstraksi, Dekomposisi, Algoritma	Implementasi modul program
Peserta didik mampu implementasi kasus menggunakan <i>function</i> dan <i>procedure</i> pada Blockly.	Mandiri, Bernalar Kritis, Kreatif	Abstraksi, Dekomposisi, Algoritma, Pengenalan Pola	Implementasi kasus menggunakan <i>function</i> dan <i>procedure</i> pada Blockly
	Mandiri, Bernalar Kritis, Kreatif	Abstraksi, Dekomposisi, Algoritma, Pengenalan Pola	
Peserta didik mampu memahami sistem bilangan.	Mandiri, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi, Pengenalan pola	Konversi sistem bilangan
Peserta didik mampu membuat program konverter sistem bilangan pada Blockly.	Mandiri, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi, Pengenalan pola	Mengimple-mentasikan konverter sistem bilangan pada Blockly
Peserta didik mampu memahami <i>parity bit</i> pada sistem bilangan biner.	Mandiri, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi, Pengenalan pola	Mengimple-mentasikan <i>parity bit</i> pada sistem bilangan biner
Peserta didik berkolaborasi dan berdiskusi implementasi beberapa konverter sistem bilangan menjadi program utuh.	Gotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi, Pengenalan pola	Berkolaborasi untuk mengim-plementasikan beberapa kon-verter sistem bi-langan menjadi program utuh
Peserta didik berkolaborasi dan berdiskusi membuat poster <i>wrap up</i> materi.	Gotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi, Pengenalan pola	Berkolaborasi untuk membuat poster <i>wrap up</i> materi

Peserta didik berkolaborasi dan berdiskusi melakukan pemrograman dengan Robot Ozobot.	Gotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi, Pengenalan pola	Eksplorasi pemrograman robot Ozobot
---	---	--	-------------------------------------

VII. Strategi Pembelajaran

Strategi pembelajaran Algoritma dan Pemrograman dapat dilakukan secara *plugged* dan *unplugged*. Kedua hal tersebut memerlukan kreativitas guru agar pembelajaran dapat berlangsung menarik.

1. Strategi Pembelajaran Perangkat Keras secara

Plugged

Materi Algoritma dan Pemrograman ini membentuk pola pikir penyelesaian masalah pada peserta didik. Aktivitas yang disediakan mencakup konsep dasar, contoh secara manual dan contoh menggunakan pemrograman visual. Guru juga dapat mencoba bersama peserta didik di kelas atau memberi contoh awal. Di Fase D (kelas VII--IX) masih digunakan pemrograman visual agar peserta didik dapat belajar dengan cara yang lebih menyenangkan. Guru juga dapat menjelaskan konsep pemrograman dengan lebih mudah karena berupa blok dan hasilnya dapat langsung dilihat.

Selain melakukan aktivitas dengan eksplorasi mandiri dibantu dengan langkah jelas dan pertanyaan-pertanyaan yang ada, peserta didik mulai belajar membuat model solusi, yaitu mendefinisikan *Input-Proses- Output* dan menganalisis persoalan. Di sini, aspek berpikir komputasional dipraktikkan, terutama saat melakukan dekomposisi, abstraksi serta memakai dan membentuk pola solusi berdasarkan solusi yang pernah dikerjakan. Oleh karena itu, rangkaian contoh soal harus memungkinkan peserta didik melakukan konstruksi pola. Rangkaian contoh harus dipilih dengan cermat, mempertimbangkan kemiripan dan kemajuan (*progression*) tingkat kesulitan persoalan. Jika guru akan mengganti kasus, hal ini perlu diperhatikan.

Selain aktivitas individu, peserta didik juga akan berkolaborasi dalam kelompok untuk melakukan beberapa aktivitas rangkuman (*wrap up*) sebagai kesimpulan atas pemahaman algoritma dan pemrograman yang sudah dipelajari. Peserta didik diajak dalam kelompok untuk membuat poster presentasi dan pengalaman dalam mempelajari materi ini.

2. Strategi Pembelajaran Perangkat Keras secara

Unplugged

Walaupun Algoritma dan Pemrograman berkaitan erat dengan perangkat lunak, pola berpikir pemrograman dapat dilakukan juga secara *unplugged*.

Peserta didik diajak untuk mengeksplorasi *method* dalam pemrograman visual dengan benda isik dan tidak menggunakan komputer. Atau, peserta didik juga dapat diajak untuk bermain peran sebagai program, hasil eksekusi program, dan tester seperti menjalankan program pada komputer. Guru dapat membuat blok program menjadi kartu-kartu blok yang dapat dimainkan bersama juga.

3. Aspek Kreatif Guru

Yang sangat perlu diperhatikan oleh para guru ialah bahwa pemrograman di tingkat SMP bukan bertujuan untuk menghasilkan “*programmer*”. Pemrograman di tingkat SMP bertujuan untuk membentuk pola pikir CT melalui karya kreatif. Paradigma yang dipilih ialah paradigma pemrograman visual karena memudahkan anak mengomposisi program tanpa disibukkan dengan kesalahan sintaks seperti dalam pemrograman tekstual.

KOMPETENSI INTI

Materi Algoritma dan Pemrograman disajikan dalam 8 kali pertemuan utama dan 1 pertemuan tambahan apabila guru/peserta didik/sekolah siap melaksanakannya.

Pertemuan 1: Scratch vs Blockly (2 JP)

I. Tujuan Pembelajaran:

1. Mengenali objek-objek dan memahami perintah atau instruksi dalam lingkungan pemrograman Scratch dan Blockly

2. Menerapkan aturan translasi konsep dari satu bahasa visual ke bahasa visual lainnya

Guru menjelaskan bahwa terdapat banyak bahasa pemrograman yang dapat digunakan.

Pada dasarnya, tiap bahasa pemrograman memiliki kemampuan yang sama, yaitu menjalankan instruksi yang diberikan dengan sintaks tertentu.

II. Apersepsi

Dalam kehidupan, seseorang perlu menguasai tidak hanya satu bahasa saja agar dapat berkomunikasi dengan banyak orang. Sebagai contoh, sebagai orang Indonesia, biasanya menguasai bahasa Indonesia, bahasa daerah, serta bahasa Inggris. Di dalam pemrograman, juga ada banyak sekali bahasa. Perbedaan antara satu bahasa dan bahasa lainnya antara lain terlihat dari tata bahasa (sintaks) yang tersedia dari setiap bahasa tersebut. Selain itu, peserta didik juga

dapat menggunakan bahasa pemrograman tertentu untuk tujuan tertentu pula. Hal ini mirip seperti ketika ingin menyapa seseorang. Jika tujuannya ialah menyapa teman yang tidak bisa berbahasa lain selain bahasa Korea, tentu akan menyapanya dengan “*annyeonghaseyo*”. Jika peserta didik ingin menyapa teman yang bisa berbahasa Inggris, akan menyapanya dengan “hallo”.

III. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser*
- b. Koneksi internet

IV. Kegiatan Inti

Kegiatan inti pada pertemuan ini adalah sebagai berikut.

1. Guru menjelaskan konsep-konsep yang berkaitan dengan perbedaan antara Scratch dan Blockly.
2. Peserta didik menjawab pertanyaan-pertanyaan yang ada pada bagian uji pemahaman.
3. Kegiatan berikutnya ialah mengerjakan Aktivitas AP-K9-01: Bilangan Prima. Setelah selesai, guru dapat menjelaskan makna dari tiap blok yang digunakan pada kedua aktivitas tersebut.

V. Jawaban Uji Pemahaman

1. Pemrograman berbasis *event* adalah sebuah paradigma pemrograman, di mana alur jalannya program ditentukan oleh suatu *event* misalnya *mouse* yang diklik, *keyboard* yang ditekan, dan lain-lain.
2. Dengan menguasai banyak bahasa pemrograman, peserta didik dapat menggunakan bahasa pemrograman untuk berbagai keperluan. Hal ini disebabkan karena seringkali, untuk suatu permasalahan tertentu bahasa pemrograman tertentu lebih baik digunakan dibandingkan dengan bahasa pemrograman lainnya.
3. Blok [length of], blok [hide variable]
4. Blok [function] dan blok [for each item in list]

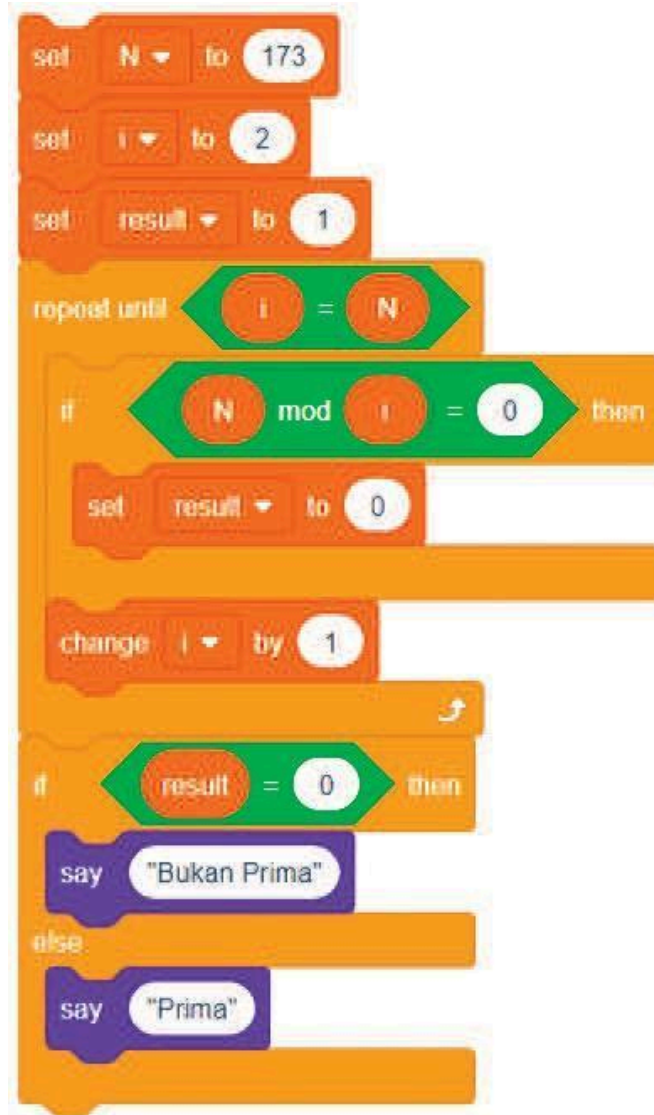
Jawaban Aktivitas AP-K9-01: Bilangan Prima

Sesuai dengan penjelasan yang ada pada Buku Siswa, akan ada dua buah variabel yang dibutuhkan, yaitu variabel N sebagai masukan, dan variabel

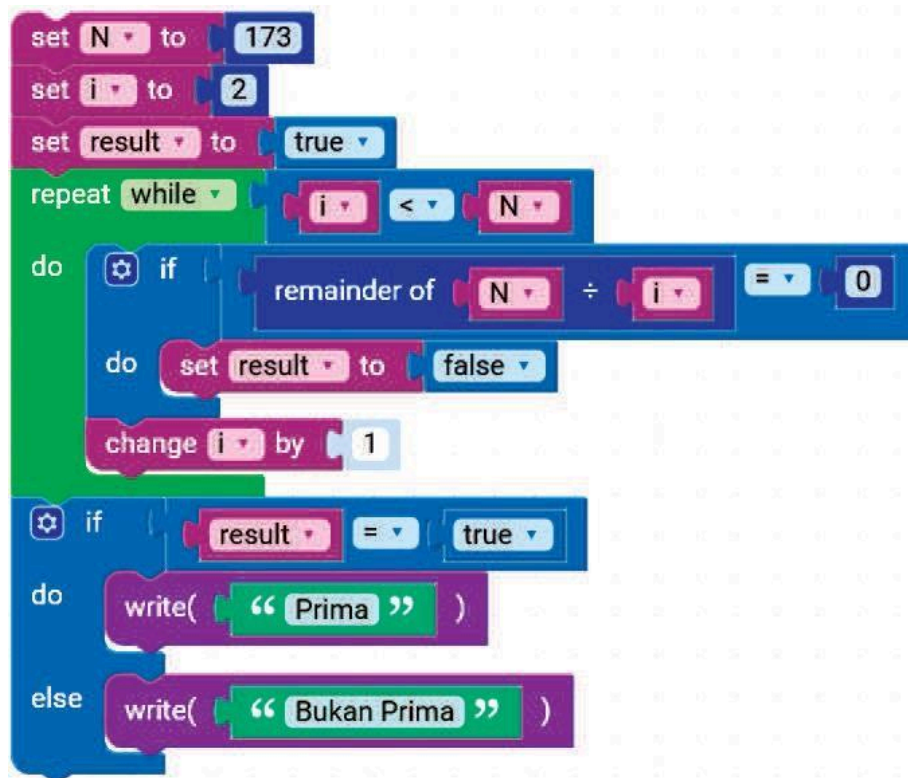
i. Sebenarnya, terdapat satu variabel lain yang dibutuhkan, yaitu variabel yang akan digunakan untuk menjadi penanda apakah N adalah bilangan prima atau bukan. Kita sebut saja variabel ini sebagai variabel result.

Gambar 1 adalah kode Scratch untuk aktivitas ini. Pada Gambar 7.1, nilai N adalah input,

angka 173 dapat diganti dengan angka berapa pun sebagai N yang akan diperiksa apakah prima atau bukan. Variabel i akan bernilai 2 sampai dengan N-1, itulah mengapa nilai awal dari variabel ini adalah 2, dan akan berubah-ubah di dalam pengulangan, sampai nilainya sama dengan N-1. Pengulangan akan berhenti ketika nilai $i = N$. Di dalam pengulangan, akan dilakukan pemeriksaan tiap nilai i, apakah ia habis membagi N atau tidak. Jika habis dibagi N, variabel result akan kita ganti nilainya menjadi 0. Nilai 0 pada variabel result akan menjadi penanda bahwa N bukan bilangan prima. Jika tidak ada i yang habis membagi N, artinya nilai result akan tetap bernilai 1. Ini menjadi penanda bahwa N adalah bilangan prima.



Gambar 7.1: Kode Scratch untuk Aktivitas AP-K9-01



Gambar 7.2: Kode Blockly untuk Aktivitas AP-K9-01

Gambar 7.2 adalah kode Blockly untuk aktivitas ini. Perbedaannya dengan kode program dalam Scratch ialah sebagai berikut.

1. Variabel *result* bertipe *boolean*. Pada Scratch, tidak ada variabel bertipe ini, sehingga digunakan 0 dan 1.
2. Blok pengulangan yang digunakan pada Blockly adalah *repeat while*, sementara pada Scratch adalah *repeat until*.
3. Blok sisa hasil bagi yang digunakan pada Blockly adalah *remainder of*, sedangkan pada Scratch adalah *mod*.
4. Untuk menampilkan hasil pemeriksaan, Blockly menggunakan blok *write*. Pada Scratch, tidak ada perintah *write* sehingga digantikan dengan blok *say*.

Pertemuan 2: Literasi Numerik (2 jp)

I. Tujuan Pembelajaran:

1. Mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan
2. Mengembangkan program visual yang mendukung literasi numerik dan sains maupun terwujudnya karya digital kreatif lainnya

II. Apersepsi

Pemrograman dapat digunakan untuk membantu manusia dalam berbagai hal. Salah satu hal

yang dekat dan dapat memanfaatkan pemrograman adalah aktivitas hitung-menghitung. Pemrograman sangat membantu dalam melakukan perhitungan, khususnya perhitungan yang rumit dan besar dimana manusia sudah sangat sulit untuk melakukannya secara manual. Kali ini, peserta didik akan diajak untuk membuat program perhitungan yang sederhana yang sangat dekat dengan kehidupan peserta didik, yaitu terkait perhitungan nilai ujian.

Peserta didik dapat diajak untuk mengingat pelajaran Matematika tentang menghitung rata-rata. Apakah peserta didik masih ingat bagaimana cara menghitungnya, mulai dari rata-rata 2 bilangan, 3 bilangan, hingga sejumlah N (banyak) bilangan.

III. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser Chrome*
- b. Koneksi internet
- c. *Tools Coding with Chrome*

IV. Kegiatan Inti

Aktivitas AP-K9-02: Proyek Hitung Rata-Rata dari N Nilai

Setelah membahas terkait perhitungan rata-rata, guru dapat membahas persoalan pada aktivitas ini. Pembahasan dapat dimulai dari cerita pengantarnya, rincian *input output* dan contoh *input output*-nya. Dalam contoh, pesertadidik diajak untuk membahas bagaimana bisa mendapatkan *output* seperti yang dicontohkan dari input yang diberikan. Peserta didik harus diajak untuk mencari contoh input lain dan menghitung *output*-nya hingga peserta didik paham dengan yang diharapkan dari persoalan ini. Setelah paham dengan soalnya, peserta didik ditantang untuk menyusun alur programnya.

Catatan: Kunci jawaban berikut hanyalah salah satu contoh solusi untuk persoalan yang diberikan. Peserta didik mungkin akan memiliki program yang berbeda-beda sesuai kreasi masing-masing untuk menyelesaikan soal tersebut.

Secara umum, alur program terdiri atas skema/pola *Input-Proses- Output*.

- *Input*

Seperti yang dibahas dalam rincian soal, *input* terdiri atas 2 bagian. Dalam program Blockly nantinya, input ini diisikan ke dalam variabel yang tentunya harus dibuat terlebih dahulu. Nama variabel dapat menyesuaikan dengan nama yang sudah dinyatakan dalam rincian program. Proses input pertama yang menentukan banyaknya peserta didik dapat dilakukan dengan mudah dengan langsung mengisikannya ke dalam variabel N . Untuk input kedua yang mengisikan sejumlah N nilai, peserta didik dapat menggunakan blok [create_list_with] dengan banyak *item* sebanyak nilai yang diharapkan. Sampai di sini, program yang dibuat adalah program sekuensial tanpa percabangan dan perulangan (Gambar 7.3).



Gambar 7.3 Bagian input kode program Hitung Rata-Rata N Nilai.

-

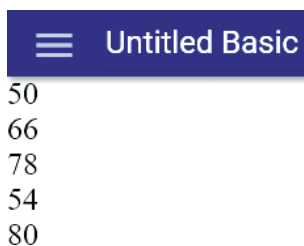
Proses

Setelah input disediakan, program dapat masuk ke proses perhitungan rata-rata. Namun sebelum itu, peserta didik perlu tahu bagaimana cara menelusuri variabel nilaiBhs yang ternyata memiliki banyak nilai (sebanyak N). Hal ini dapat dilakukan dengan perulangan, salah satunya menggunakan blok [count_with]. Jika belum paham penggunaannya, peserta didik dapat diajak untuk menggunakan blok ini untuk menuliskan isi dari variabel nilaiBhs satu per satu.



Gambar 7.4 Contoh kode program untuk menelusuri sebuah list.

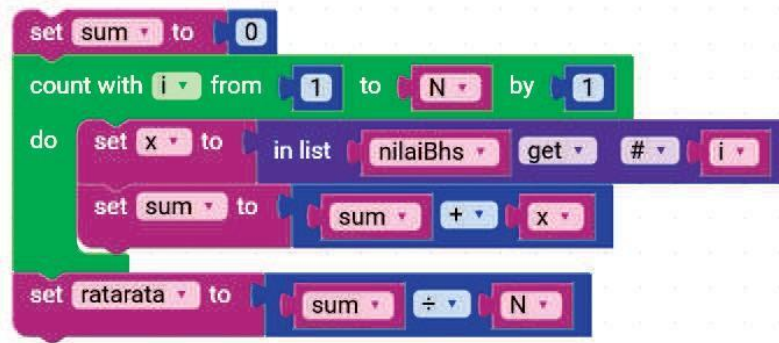
Program tersebut melakukan perulangan dengan berhitung satu sampai N (naik dengan selisih 1 sehingga hitungannya akan berupa 1, 2, 3 dan seterusnya). Perhatikan dalam perhitungan tersebut, digunakan variabel i sebagai variabel pengendali perulangan yang digunakan untuk menyimpan nilai hitungan (1, 2, 3 dan seterusnya). Selanjutnya, di dalam perulangan, disediakan variabel x untuk menyimpan satu nilai sehingga variabel nilaiBhs yang terdiri atas banyak nilai dapat diolah satu per satu, dalam hal ini untuk dicetak. Maka, saat dijalankan, hasilnya ditampilkan seperti diperlihatkan pada Gambar 7.5.



Gambar 7.5 Contoh keluaran dari kode program yang menelusuri sebuah list

Setelah nilai-nilai tersebut siap untuk diolah, perhitungan rata-rata bisa dilakukan.

Perhitungan rata-rata dilakukan dengan menjumlahkan semua nilai yang selanjutnya dibagi dengan banyaknya nilai. Oleh karena itu, terlebih dahulu, perlu dijumlahkan semua nilai yang telah dipersiapkan sebagai x. Dalam contoh ini, semua nilai ini diakumulasikan kedalam variabel sum. Variabel sum perlu diinisialisasi terlebih dahulu dengan mengosongkan nilainya sebelum perulangan dilakukan. Selanjutnya, setelah semua nilai diakumulasikan dalam perulangan, hasil penjumlahan ini dapat dibagi dengan banyaknya nilai (N) sebagai nilai rata-rata seperti diperlihatkan pada Gambar 7.6.



Gambar 7.6 Bagian proses kode program Hitung Rata-Rata N Nilai.

-

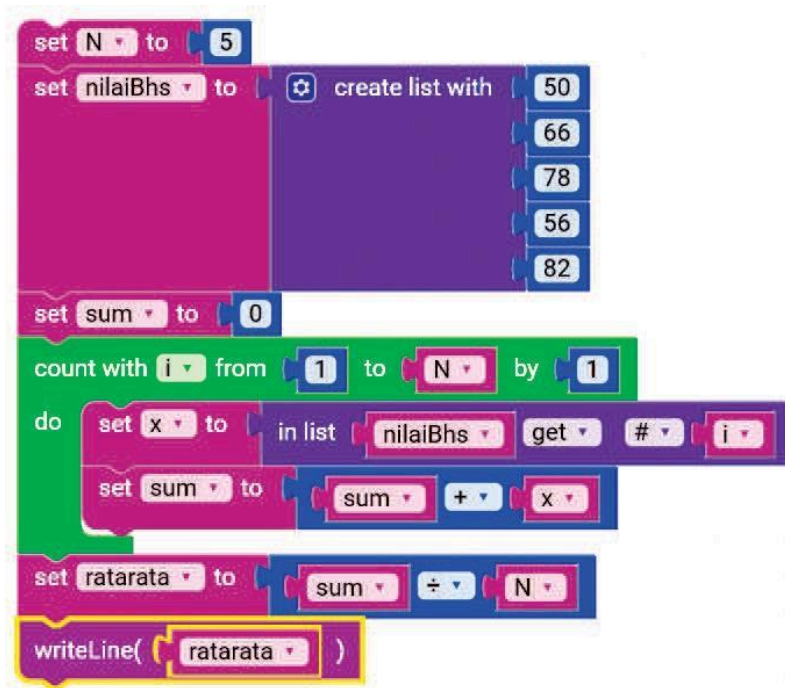
Output

Terakhir, variabel ratarata dapat dicetak untuk mengetahui nilai rata-rata N bilangan tersebut. Dalam contoh ini, nilai rata-rata yang diperoleh adalah 65,6. Perintah mencetak variabel dapat dilakukan dengan menggunakan blok [writeLine] yang akan mencetak variabel ratarata, seperti diperlihatkan pada Gambar 7.7.



Gambar 7.7 Bagian input kode program Hitung Rata-Rata N Nilai

Dengan demikian, didapatkan kode program utuh sebagai salah satu kunci jawaban untuk proyek ini seperti diperlihatkan pada Gambar 7.8.

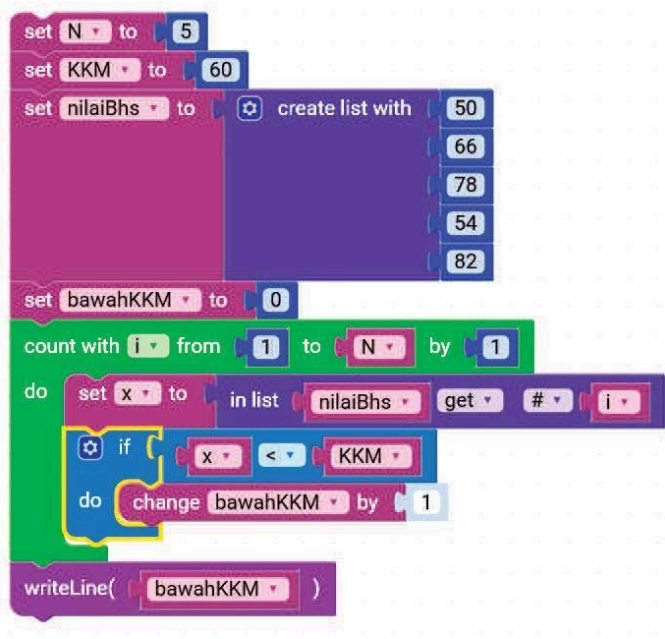


Gambar 7.8 Contoh kunci jawaban proyek AP-K9-02

Setelah program peserta didik berhasil menunjukkan *output* yang benar, guru dapat mengajak peserta didik untuk memodifikasi input seperti yang sudah dikumpulkan di awal saat pembahasan soal. Cari tahu apakah *output* yang dihasilkan program peserta didik sama dengan perhitungan manual yang dilakukan. Selanjutnya, guru dapat menunjukkan contoh program ini dengan menggunakan Blockly secara lebih interaktif pada link: https://blockly-demo.appspot.com/static/demos/code/index.html#_vzcpgb

Aktivitas AP-K9-03: Proyek Hitung Kriteria Ketuntasan Minimum dari N Nilai

Setiap kali menemui persoalan, guru dapat membahas deskripsi permasalahan, rincian program, contoh input dan pembahasan bagaimana mendapatkan *output*-nya. Selanjutnya, peserta didik diminta untuk memahami persoalan tersebut dengan mencari contoh kasus *input* lain dan kemudian menentukan *output* dari kasus tersebut. Dengan menghasilkan *input* *output*-nya sendiri, peserta didik dipastikan memahami persoalannya dengan lebih baik. Setelah itu, peserta didik diajak untuk mencari tahu persamaan dan perbedaan kasus ini dibandingkan dengan kasus pada aktivitas sebelumnya (AP-K9-02). Hal ini dilakukan agar kita dapat memodifikasi dengan tepat program yang sudah dibuat sebelumnya (Gambar 7.8). Persamaan atau kemiripan dari kedua kasus ini terletak pada input dan *output*-nya walaupun memiliki sedikit perbedaan. Maka, modifikasi input dapat dilakukan dengan menambahkan satu variabel yang menyimpan nilai KKM dan modifikasi *output* dilakukan dengan mengubah variabel yang akan dicetak. Persamaan lainnya terletak pada penggunaan perulangannya sehingga perulangan pada program sebelumnya dapat dipertahankan.



Gambar 7.9 Contoh kunci jawaban proyek AP-K9-03

Perbedaan kode program ini (Gambar 7.9) dengan kode program sebelumnya terletak pada proses perhitungannya. Jika pada program sebelumnya, rata-rata dihitung dengan mengakumulasi semua nilai, pada program ini, dicek semua nilai apakah kurang dari KKM atau tidak. Jika ya, program perlu menyimpan variabel untuk menghitung banyaknya nilai yang kurang dari KKM tersebut. Di sinilah, konsep percabangan digunakan. Banyaknya nilai yang kurang dari KKM disimpan dalam sebuah variable, misalnya *bawahKKM*, di mana setiap kita menemui nilai yang kurang dari KKM, variabel ini ditambah satu (*increment*). Perhatikan kunci jawaban di atas sebagai salah satu contoh program lengkap dari solusi persoalan ini. Dari program tersebut, akan didapatkan *output* bernilai 2, yang menunjukkan dua nilai yang kurang dari KKM (50 dan 54).

Pertemuan 3: Modularisasi Program (2 jp)

I. Tujuan Pembelajaran:

1. Memahami modularisasi program dan model komputasinya
2. Mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan

II. Apersepsi

Pada pertemuan ini, peserta didik akan membuat program dari materi yang dipelajari pada mata pelajaran Matematika, yaitu mengenai fungsi. Dalam pelajaran Matematika, peserta didik mengenal bahwa fungsi terdiri atas variabel-variabel dan nilai fungsi tersebut dipengaruhi oleh nilai dari variabel-variabel tersebut. Peserta didik akan mengimplementasikan konsep fungsi tersebut menjadi modul program. Nilai variabel pada

fungsi tersebut diatur pada modul utama. Dengan demikian, peserta didik dapat memahami juga bahwa modul untuk menghitung fungsi matematis saling berkaitan dengan modul utama dari program.

III. Pemantik

Guru dapat memberikan contoh fungsi matematis sederhana, misalnya dan meminta peserta didik menghitung hasil dari fungsi tersebut dengan nilai x dan y yang berbeda-beda. Dengan demikian, peserta didik mengingat kembali konsep dari fungsi dan variabel.

IV. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser Chrome*
- b. Koneksi internet
- c. *Tools Coding with Chrome*

V. Kegiatan Inti

Pada pertemuan ini, peserta didik diajak untuk mengalami terlebih dahulu pembuatan modul program dengan menyalin langkah-langkah pembuatan kode modul program. Setelah membuat modul program, peserta didik diminta untuk melakukan beberapa kali eksperimen terkait eksekusi modul program. Setelah mendapatkan pengalaman tersebut, peserta didik diarahkan untuk menarik kesimpulan berdasarkan pengalaman mereka.

Pada pertemuan ini, kegiatan yang dapat dilakukan di kelas adalah sebagai berikut.

1. Guru menjelaskan konsep modularisasi program.
2. Peserta didik diminta untuk mencoba kode program yang diberikan pada Aktivitas AP-K9-04: Modul Program: Fungsi Kuadrat.
3. Setelah itu, peserta didik diarahkan untuk mengerjakan AP-K9-05: Modul Program: Persamaan Linier.
4. Peserta didik diminta untuk mencoba kode program yang diberikan pada Aktivitas AP-K9-06: Modul Program: Modifikasi Modul F1.

Penjelasan Umum Mengenai Modul Program

Pada bagian ini, diberikan penjelasan konsep-konsep umum yang perlu diketahui oleh guru.

1. Konsep modularisasi program.

Mulai pada pertemuan ke-3 ini, peserta didik belajar bahwa sebuah program komputer dapat terdiri atas beberapa sub-program. Sub- subprogram ini saling terkait. Sub-program tersebut disebut modul. Dengan demikian, modul program dapat didefinisikan sebagai sekumpulan blok yang dikemas menjadi satu kesatuan kode yang

menjalankan tujuan tertentu.

Blok-blok program yang terdapat pada sebuah modul program hanya akan dieksekusi atau dijalankan jika modul program tersebut dipanggil oleh modul pemanggil. Setiap program, selalu terdapat satu sub-program utama yang menjadi titik awal eksekusi program. Sub- program ini biasanya disebut dengan modul **main** atau modul **utama**.

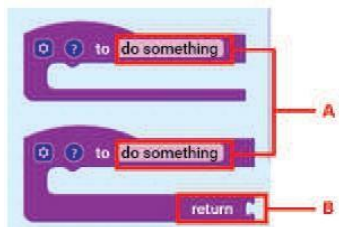
2. Penjelasan pembuatan modul program pada Blockly.

Untuk membuat modul program pada Blockly, dapat digunakan menu “Function” seperti yang tertera pada Gambar 7.10:



Gambar 7.10 - Menu Function

Ada dua bagian dari menu tersebut yang akan digunakan pada pertemuan ke-3 dan ke-4, yaitu bagian yang ditunjukkan pada Gambar 7.11 berikut ini.



Gambar 7.11 - Block untuk Membuat Modul Program

Gambar 7.11 menunjukkan dua blok program yang akan digunakan untuk membuat modul program. Setiap modul program, perlu diberi nama. Nama dari modul program pada umumnya menggambarkan apa yang akan dilakukan pada modul tersebut. Nama modul dapat dituliskan pada bagian “do something” seperti yang ditunjukkan oleh huruf A pada Gambar 7.11.

Terdapat dua jenis modul program, yaitu modul yang akan menghasilkan sebuah nilai hasil yang ‘diberitahukan’ kepada modul pemanggil dan modul yang tidak menghasilkan nilai hasil yang perlu diberitahukan kepada pemanggilnya. Modul yang memiliki nilai kembalian yang akan diberitahukan kepada modul pemanggilnya, disebut memiliki *return value*, ditunjukkan oleh huruf B pada Gambar 7.11. Penjelasan mengenai perbedaan kedua jenis modul ini akan dibahas lebih lanjut pada pertemuan ke-4.

Jawaban Aktivitas AP-K9-04: Modul Program: Fungsi Kuadrat

Aktivitas Individu

Aktivitas AP-K9-04: Modul Program: Fungsi Kuadrat

Tujuan: Pada aktivitas ini, kalian akan mengimplementasikan perhitungan persamaan kuadrat yang mempraktikkan pembuatan modul-modul program

Tingkat Kesulitan: 3

Syarat Pengetahuan: Fungsi Kuadrat

Saat ini, kalian akan mencoba membuat program yang terdiri atas dua buah modul. Berikut ini adalah langkah untuk membuat modul pada Blockly.

Sebelum memulai aktivitas ini, peserta didik harus memahami fungsi kuadrat yang diajarkan pada mata pelajaran Matematika.

Tingkat kesulitan aktivitas ini ialah 3 karena program yang dibuat sudah melibatkan lebih dari satu modul, tetapi tidak melibatkan percabangan atau *looping*.

Jawaban untuk eksperimen pada Tabel 7.3:

Tabel 7.3 Jawaban Eksperimen terkait Modul F1

No	Parameter				Hasil perhitungan manual	Hasil dari program	Sesuai? (ya/tidak)
	a	b	c	x			
1	1	1	1	1	3	3	Ya
2	3	10	5	5	133	133	Ya
3	10	20	70	10	1270	1270	Ya

Kesimpulan yang dapat diambil berdasarkan eksperimen ini ialah nilai yang dikembalikan oleh modul F1 bergantung pada nilai parameter input yang diterima oleh modul tersebut. Parameter dari sebuah modul dapat dikendalikan dari modul lain yang memanggil modul tersebut (dalam aktivitas ini, modul pemanggil modul F1 adalah modul utama).

Jawaban Aktivitas AP-K9-05: Modul Program: Persamaan Linier

Aktivitas Individu

Aktivitas AP-K9-05: Modul Program: Persamaan Linier

Tujuan: Mampu mengimplementasikan perhitungan persamaan linier yang mempraktikkan pembuatan modul-modul program

Tingkat Kesulitan: 3

Syarat Pengetahuan: Persamaan Linier

Pada aktivitas ini, kalian tidak diberi panduan berupa blok programnya. Silakan berlatih membuat blok program berdasarkan contoh-contoh yang sudah diberikan pada aktivitas-aktivitas sebelumnya.

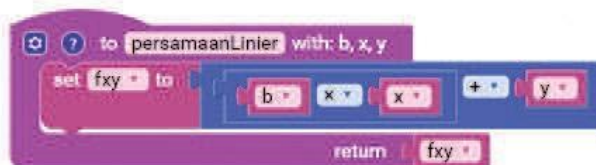
Salah satu bentuk persamaan linear ialah

$$f(x,y) = bx + y$$

Persamaan 2 - Contoh Persamaan Linier

Sebelum memulai aktivitas ini, peserta didik harus memahami persamaan linier yang diajarkan pada mata pelajaran Matematika. Tingkat kesulitan aktivitas ini ialah 3 karena program yang dibuat sudah melibatkan lebih dari satu modul, tetapi tidak melibatkan percabangan atau *looping*.

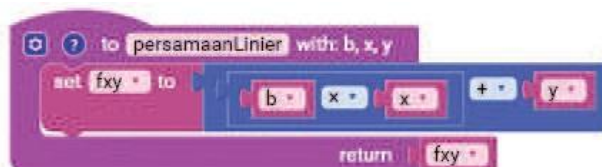
Modul persamaanLinier dan modul utama untuk persamaan :



utama

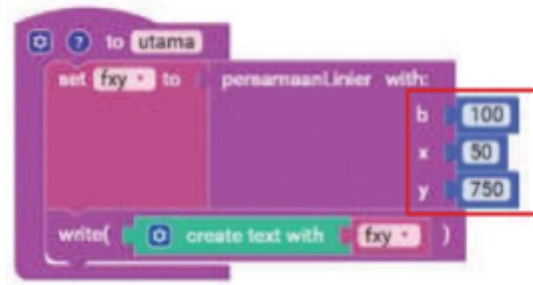
Penjelasan kode program

1. Modul persamaanLinier ini memiliki tiga buah parameter, sesuai banyaknya parameter pada persamaan linier yang diberikan pada soal.



2. Modul utama ini memanggil modul persamaanLinier. Saat melakukan pemanggilan modul, ditentukan juga nilai-nilai untuk parameter modul

persamaanLinier (ditandai dengan kotak merah). Nilai kembalian dari modul persamaanLinier diterima oleh variabel fxy pada modul utama dan ditampilkan ke layar.



Jawaban untuk eksperimen pada Tabel 7.4:

Tabel 7.4 Jawaban terkait Modul persamaanLinier

No	Parameter			Hasil perhitungan manual	Hasil dari program	Sesuai? (ya/tidak)
	b	x	y			
1	2	3	5	11	11	Ya
2	100	50	750	5750	5750	Ya
3	70	3	700	940	910	Ya

Panduan untuk menuntun aktivitas “Ayo, Kita Renungkan”

Pada bagian ini, diberikan tuntunan untuk refleksi peserta didik karena pada refleksi ini, peserta didik dituntut untuk mampu menarik abstraksi dari dua aktivitas yang telah dilakukan. Beberapa hal yang dapat disimpulkan dari aktivitas AP-K9-04 dan AP-K9-05 ialah seperti berikut.

1. Program dapat terdiri atas beberapa sub-program yang disebut modul.
2. Program yang melibatkan beberapa modul, selalu ada satu modul utama (*main*) yang berfungsi sebagai titik awal eksekusi program.
3. Sebuah modul program dapat memanggil modul lainnya.
4. Modul program yang tidak pernah dipanggil, tidak akan dieksekusi.

Jawaban Aktivitas AP-K9-06: Modul Program: Modifikasi Modul

F2

Aktivitas Individu


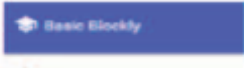
Aktivitas AP-K9-06: Modul Program: Modifikasi Modul

F2 Tujuan: Mampu memahami parameter modul program **Tingkat**

Kesulitan: 3

Syarat Pengetahuan: Fungsi Kuadrat

Pada bagian ini, kalian akan belajar lebih lanjut terkait dengan parameter modul program. Mari, kita buat sebuah modul lain, yaitu modul F2. Pada modul F2 tersebut, kita akan memodifikasi modul F1. Ikuti langkah-langkah berikut ini

Nama	Block	Keterangan
Buka <i>ile</i>		1. Untuk membuka <i>ile</i> program kalian, pilih menu yang pada bagian kiri atas layar Blockly.
Buka <i>ile</i>		2. Buka <i>ile</i> yang kalian buat pada Aktivitas AP-K9-04 dengan

Sebelum memulai aktivitas ini, peserta didik harus memahami fungsi kuadrat yang diajarkan pada mata pelajaran Matematika. Selain itu, peserta didik sudah mengerjakan Aktivitas AP-K9-04.

Tingkat kesulitan aktivitas ini ialah 3 karena program yang dibuat sudah melibatkan lebih dari satu modul, tidak melibatkan percabangan atau *looping*.

Jawaban untuk pertanyaan-pertanyaan terkait modul F2 dan modul utama:

1. Nilai variabel a, b, c, dan x tidak lagi dapat diatur dari modul utama.
2. Jika dikaitkan dengan Tabel 7.3, dapat disimpulkan bahwa modul yang tidak memiliki parameter input akan selalu menghasilkan hasil yang sama ketika modul tersebut dieksekusi.

Pertemuan 4: Modularisasi Program (2) (2 jp)

I. Tujuan Pembelajaran:

1. Memahami modularisasi program dan model komputasinya.
2. Mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan.

II. Apersepsi

Guru dapat mengajak peserta didik untuk melihat lagi gambar rumah yang tersusun dari banyak balok. Sebuah program komputer yang digunakan dalam kehidupan sehari-hari, biasanya sangat kompleks. Program tersebut terdiri atas banyak sekali modul penyusun. Pada pertemuan ini juga, peserta didik akan diperkenalkan pada jenis modul yang lain, yaitu *procedure*, modul yang tidak mengembalikan sebuah nilai kembalian kepada modul yang memanggilnya.

III. Pemantik

Guru dapat membahas cerita mengenai Ibu dan Andi yang ada pada Buku Siswa. Cerita tersebut menjadi pengantar untuk membedakan modul program yang memiliki nilai kembalian atau tidak.

IV. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser Chrome*
- b. Koneksi internet
- c. *Tools Coding with Chrome*

V. Kegiatan Inti

Berikut kegiatan yang akan dilakukan pada pertemuan ini:

1. Peserta didik diminta untuk mencoba kode program yang diberikan pada Aktivitas AP-K9-07: Modul Program: Fungsi Kuadrat dan *Plotting*.
2. Inti dari pertemuan ini dibahas melalui pertanyaan releksinya peserta didik. Guru dapat menuntun peserta didik untuk menemukan jawaban dari pertanyaan releksinya tersebut.
3. Sebagai materi pengayaan, peserta didik dapat mengerjakan Aktivitas AP-K9-07: Modul Program: Hitung Volume Tabung.

Penjelasan mengenai *procedure* dan *function*

Penjelasan mengenai *procedure* dan *function* dibahas dengan lengkap pada Buku Siswa melalui analogi percakapan antara Ibu dan Andi.

Jawaban untuk Ayo, Berpikir yang menampilkan percakapan Andi dan Ibu:

Percakapan tersebut mengilustrasikan *function* karena setelah mencuci apel-apel, Andi mengembalikan apel tersebut kepada ibunya. Andi dapat dianggap memberikan *return value* kepada ibunya.

Jawaban Aktivitas AP-K9-07: Modul Program: Fungsi Kuadrat dan *Plotting*

Aktivitas Individu

Aktivitas AP-K9-07: Modul Program: Fungsi Kuadrat dan *Plotting*

Tujuan: Mampu membuat program untuk melakukan pemetaan titik-titik hasil perhitungan fungsi kuadrat pada bidang dua dimensi

Tingkat Kesulitan: 4

Syarat Pengetahuan: Fungsi Kuadrat dan bidang Cartesian

Pada aktivitas ini, kalian akan membuat program yang terdiri atas tiga buah modul, yaitu modul untuk menghitung fungsi kuadrat, modul untuk melakukan pemetaan hasil fungsi kuadrat pada bidang dua dimensi, dan modul utama. Sebelum membuat dengan program, mari, buat pemetaan fungsi kuadrat tersebut secara manual.

Fungsi kuadrat yang akan digunakan adalah fungsi kuadrat yang sama dengan yang terdapat pada pertemuan ke-3, yaitu

Sebelum memulai aktivitas ini, peserta didik harus memahami fungsi kuadrat dan bidang Cartesian yang diajarkan pada mata pelajaran Matematika. Selain itu, peserta didik sudah mengerjakan Aktivitas AP- K9-04.

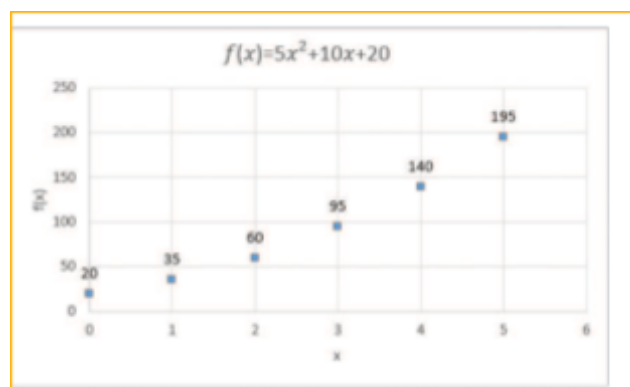
Tingkat kesulitan aktivitas ini ialah 4 karena program yang dibuat sudah melibatkan lebih dari satu modul dan melibatkan *looping*.

Jawaban untuk eksperimen pada Tabel 7.5, terkait fungsi kuadrat dengan $a=5$, $b=10$, dan $c=20$.

Tabel 7.5 - Jawaban Eksperimen terkait Fungsi Kuadrat

x	f(x)
0	20
1	35
2	60
3	95
4	140
5	195

Berikut adalah hasil pemetaan dengan $a=5$, $b=10$, dan $c=20$.



Tuntunan jawaban untuk pertanyaan releksasi peserta didik:

1. Baik *function* maupun *procedure*, keduanya akan mengeksekusi blok- blok program yang ada di dalamnya ketika modul tersebut dipanggil. Pada akhir bagian akhir eksekusi modul, *function* mengembalikan nilai kembalian ke modul yang memanggilnya, sedangkan *procedure* tidak memberikan nilai kembalian.
2. *Function* digunakan ketika modul pemanggilnya memerlukan informasi dari *function* tersebut. *Procedure* pada umumnya digunakan ketika program perlu melakukan suatu perintah saja, misalnya perintah untuk menampilkan teks ke layar.
3. Modul memerlukan parameter karena seringkali dalam sebuah percobaan saintifik, kita perlu melakukan percobaan yang berulang-ulang, dengan mengubah salah satu atau beberapa parameter. Misalnya, kita akan mengamati pertumbuhan tanaman. Model pertumbuhan tanaman diberikan dalam bentuk persamaan yang bergantung pada suhu dan kelembapan. Maka, model fungsi pertumbuhan menjadi: $F(t,h) =$ dengan t adalah temperatur rata-rata harian dan h adalah kelembapan rata-rata harian (*humidity*). Kita perlu mengubah nilai t dan/atau h untuk mencari nilai h dan t yang optimal. Di sinilah kita perlu menjalankan fungsi yang sama berulang kali dengan parameter h dan t yang berbeda-beda.
4. *Function* atau *procedure* yang tidak mengandung *function* atau *procedure* lain di dalamnya berarti tidak melakukan pemanggilan modul lain. Dengan demikian, modul tersebut adalah sekumpulan instruksi/perintah yang akan dijalankan jika ada modul lain yang memanggilnya. *Function* atau *procedure* yang di dalamnya memanggil *function* atau *procedure* lain dapat dianalogikan dengan amplop yang di dalamnya masih terdapat amplop lain. Untuk dapat membaca semua pesan yang ada pada amplop pertama, amplop yang terdapat di dalam amplop tersebut harus dibuka juga.

Aktivitas Individu

Aktivitas AP-K9-o8: Modul Program: Hitung Volume

Tabung Tujuan: Mampu membuat program menghitung volume tabung

Tingkat Kesulitan: 4

Syarat Pengetahuan: Konsep lingkaran, luas dan volume lingkaran

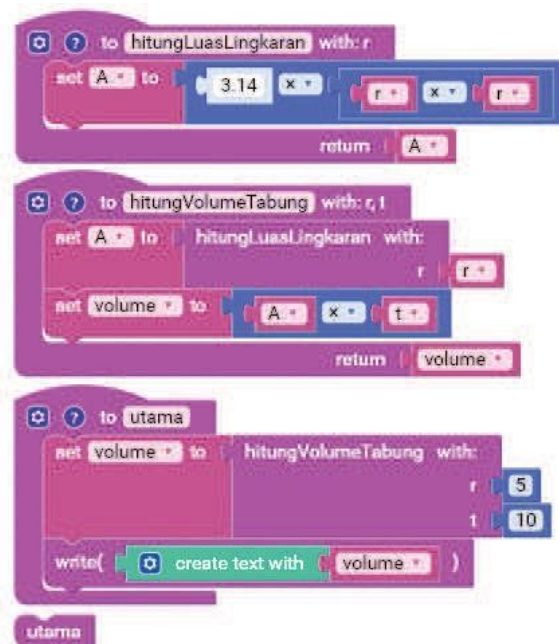
Pada aktivitas ini, kalian tidak diberi panduan berupa blok programnya. Silakan berlatih membuat blok program berdasarkan contoh-contoh yang sudah diberikan pada aktivitas-aktivitas sebelumnya.

Dari pelajaran Matematika, kalian telah mengetahui bahwa rumus umum untuk menghitung volume bangun ruang ialah menghitung hasil perkalian

Sebelum memulai aktivitas ini, peserta didik harus memahami konsep perhitungan volume tabung.

Tingkat kesulitan aktivitas ini ialah 4 karena program yang dibuat sudah melibatkan lebih dari satu modul, tetapi tidak melibatkan per-cabangan atau *looping*. Panduan yang diberikan berupa tuntunan langkah-langkah pembuatan modul, bukan contoh potongan blok program.

Panduan untuk modul-modul yang diperlukan sudah diberikan pada Buku Siswa. Berikut adalah jawaban untuk program penghitung volume tabung.

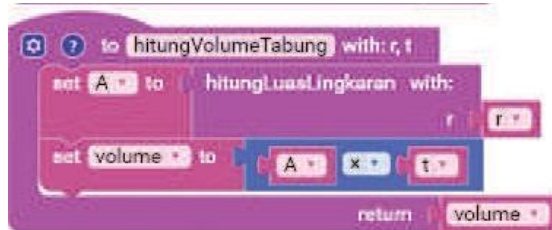


Penjelasan kode program.

1. Modul hitungLuasLingkaran berikut ini berguna untuk menghitung luas alas dari tabung, yaitu sebuah lingkaran. Rumus luas lingkaran adalah



Modul hitungVolumeTabung berikut ini memanggil modul hitungLuasAlas. Dengan demikian, hasil nilai kembalian dari modul hitungLuasAlas diterima oleh modul hitungVolumeTabung, kemudian dikalikan dengan tinggi tabung.



2. Modul utama ini memanggil modul hitungVolumeTabung. Nilai kembalian dari modul ini disimpan pada variabel volume dan ditampilkan ke layar.



Pertemuan 5: Literasi Sains (2 jp)

I. Tujuan Pembelajaran:

1. Mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan.
2. Mengembangkan program visual yang mendukung literasi numerik dan sains maupun terwujudnya karya digital kreatif lainnya.

II. Apersepsi

Seperti disampaikan sebelumnya bahwa pemrograman dapat digunakan untuk membantu dalam berbagai hal, termasuk dalam dunia sains. Dunia sains juga tidak terlepas dari perhitungan. Berbagai percobaan serta observasi sains yang dilakukan di laboratorium memiliki keterbatasan, misalnya keterbatasan bahan ataupun waktu. Oleh karena itu, percobaan

tersebut perlu disimulasikan sehingga diasumsikan bahwa ada bahan atau waktu yang tak terbatas. Bahan dan waktu tersebut dapat berkembang dengan bantuan perhitungan program

komputer. Kali ini, peserta didik diajak untuk melakukan perhitungan terkait sebuah percobaan sains.

Sebagai pemantik, peserta didik dapat diajak untuk berpikirkahwa semua yang ada di dunia ini akan berkembang, dari yang kecil seperti sel sampai alam semesta. Perkembangan ini ada yang konstan atau perkembangan dengan penambahan tetap, ada yang linear atau penambahan sesuai dengan kondisi, ada yang eksponensial atau penambahan jauh melebihi kondisi. Peserta didik diajak untuk mencari tahu apa saja contoh perkembangan tersebut, misalnya saldo tabungan, jarak tempuh kendaraan, tinggi air bak mandi, jumlah penduduk, hingga kasus COVID-19, baik di Indonesia maupun di dunia.

III. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser Chrome*
- b. Koneksi internet
- c. *Tools Coding with Chrome*

IV. Kegiatan Inti

Aktivitas AP-K9-09: Proyek Pertumbuhan 3 Hari

Perlu diingatkan bahwa untuk bisa memahami persoalan ini, peserta didik perlu membuat contoh input dan *output*-nya sendiri sehingga memahami bagaimana cara mendapatkan *output*-nya. Sebagai petunjuk, guru dapat menjelaskan terkait konsep perbandingan senilai dari dua nilai atau variabel. Jika salah satu variabel membesar, variabel yang lain ikut membesar (berbanding lurus). Jika perbesarannya ini memiliki rasio, nilai variabel yang diminta dapat dihitung. Perhatikan tabel perbandingan dua variabel yang berbanding lurus berikut.

Tabel 7.6 Perbandingan Dua Variabel yang Berbanding Lurus

var1	var2
3	6
7	?

Dari tabel perbandingan di atas, dengan informasi bahwa jika var1 (lama) bernilai 3, maka var2 (lama) bernilai 6. Dari informasi ini, kita dapat menghitung, jika var1 (baru) bernilai 7 maka var2 (baru) akan bernilai 14. Nilai var2 akan membesar dengan rasio 2 kali lipat dari var1. Nilai rasio diperoleh dari var2 yang lama yang dibagi dengan var1 yang lama. Rasio ini kemudian digunakan untuk menghitung nilai var2 yang baru, yaitu dikalikan dengan nilai var1 yang baru.

Untuk kasus Proyek Pertumbuhan 3 Hari, hanya terdapat 2 nilai, yaitu var1 lama dan var2 lama. var1 baru terdapat pada var1 lama karena dalam proses pertumbuhan nilai yang lama

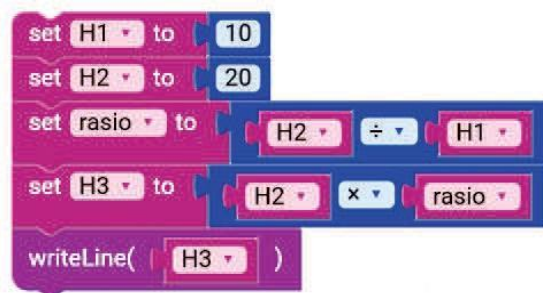
akan menjadi nilai baru untuk perhitungan berikutnya. Oleh karena itu, jika dibuat tabel perbandingan, contoh input *output* yang diberikan akan dapat dimodelkan sebagai berikut:

Tabel 7.7 Model Contoh Input dan Output

var1	var2
10	20
20	?

Dari perbandingan tersebut, kita perlu menghitung nilai rasio. Dalam kasus ini, dihitung dari nilai H2 yang dibagi dengan nilai H1. Selanjutnya, nilai yang ditanya (H3) dihitung dari nilai H2 sebelumnya dikalikan dengan nilai rasio yang telah dihitung. Sampai bagian petunjuk ini, peserta didik dapat ditantang kembali untuk menyelesaikan programnya (bagi yang belum selesai). Programnya dapat dibuat dengan skema/pola dasar *Input-Proses-Output*. Alur yang dimiliki juga hanya alur sekuensial tanpa percabangan dan perulangan. Walaupun pada soal diminta untuk menggunakan konsep modularitas, pada fase ini, peserta didik masih dapat dibebaskan untuk membuat program tanpa modul.

Setelah 5-10 menit mencoba, guru dapat menunjukkan contoh kode program pada Gambar 7.12 sebagai salah satu alternatif solusi dari persoalan ini.



Gambar 7.12 Contoh jawaban program Pertumbuhan 3 Hari

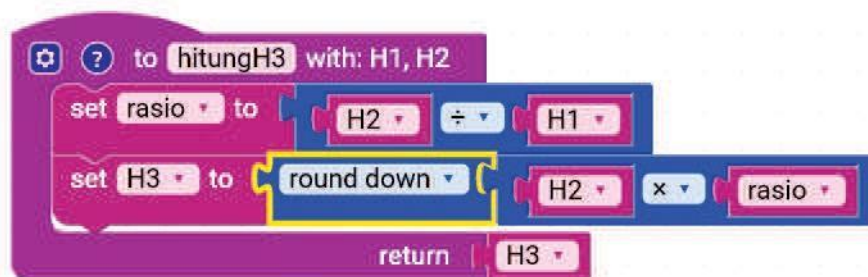
Seperti disampaikan sebelumnya bahwa contoh program di atas masih merupakan program sekuensial tunggal yang belum dimodularisasi. Bagaimana cara mengubah program tersebut agar menjadi program yang modular? Kode program tersebut dapat dipecah ke dalam beberapa bagian di mana salah satu (atau dapat juga lebih dari satu) bagiannya nanti dipisah ke dalam sebuah modul yang dapat dipergunakan berkali-kali.

Perhatikan skema dasar program di atas (*Input-Proses-Output*). Bagian *Input* terdapat pada dua baris pertama, bagian *Proses* terdapat pada dua baris di tengah, dan bagian *Output* terdapat pada satu baris terakhir. Dari bagian-bagian ini, kita dapat memisahkan bagian blok *Proses* (walaupun demikian, peserta didik juga dapat memisahkan bagian atau *Output*-nya). Bagian *Proses* atau perhitungan nilai H3 ini dipisah karena biasanya proses menjadi sebuah bagian yang nantinya akan dapat digunakan oleh program yang lain. Jika dua blok pada bagian *Proses* pada program di atas dipisah ke dalam sebuah modul, potongan kode

program tersebut dapat dipindah ke dalam blok *functions* [to_do_something]. Guru dapat mendiskusikan parameter apa yang digunakan, atau blok mana yang digunakan: blok dengan *return* atau tanpa *return*.

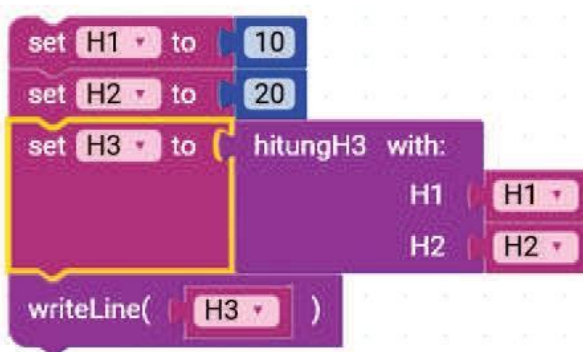
Peserta didik dapat menggunakan blok dengan *return* karena yang diperlukan dari blok proses atau perhitungan nilai H3 ini ialah nilai H3-nya. Oleh karena itu, diperlukan blok [to_do_something] yang memiliki *return* untuk mengembalikan nilai H3 yang telah dihitung. Selanjutnya, sebagai parameter, peserta didik dapat menggunakan parameter dari variabel H1

dan H2 sebagai input untuk *function* ini. Oleh karena itu, salah satu kunci jawaban modul untuk proyek ini diperlihatkan pada Gambar 7.13 berikut. Blok [round_down] ditambahkan untuk membulatkan perhitungan karena banyaknya organisme bakteri dihitung utuh (tidak ada pecahan). Tidak lupa untuk mengganti nama blok [to_do_something] tersebut dengan nama yang relevan, misalnya hitungH3 seperti diperlihatkan pada contoh kunci jawaban proyek Gambar 7.13.



Gambar 7.13 Contoh kunci jawaban modul untuk proyek AP-K9-09

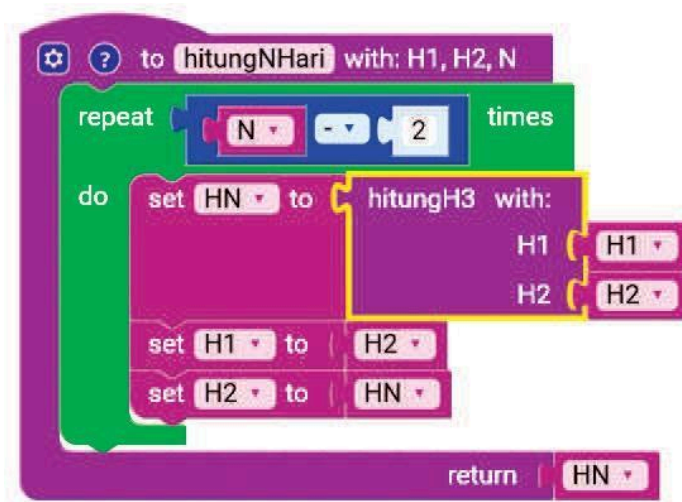
Setelah modul untuk perhitungan H3 selesai dibuat, fungsi tersebut dapat digunakan dan dipanggil pada program utama sebelumnya. Perintah ini mengganti dua baris pada bagian Proses, cukup dengan satu baris perintah set H3 yang menggunakan/memanggil fungsi hitungH3 dengan menggunakan variabel H1 dan H2 sebagai parameternya. Oleh karena itu, salah satu kunci jawaban untuk proyek ini dicontohkan kode program pada Gambar 7.14.



Gambar 7.14 Contoh kunci jawaban program utama untuk proyek AP-K9-09

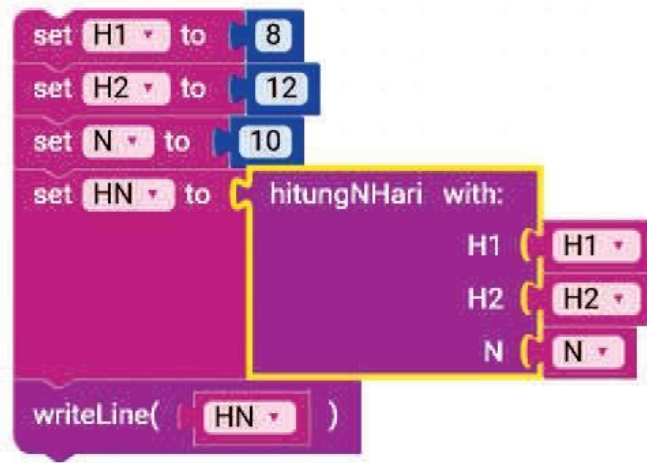
Aktivitas AP-K9-10: Proyek Pertumbuhan N Hari

Proyek ini ialah lanjutan dari proyek sebelumnya (AP-K9-09). Oleh karena itu, peserta didik dapat memodifikasi program sebelumnya agar sesuai dengan kasus permasalahan pada kasus ini. Perbedaannya terletak pada proses perhitungannya di mana pada persoalan ini, peserta didik diminta untuk menghitung perkembangan secara berulang. Oleh karena itu, kita dapat membuat satu lagi *function* untuk melakukan perulangan ini. Perulangan ini menggunakan perulangan blok *repeat* untuk menghitung perkembangan organisme bakteri sebanyak hari yang diminta. Dalam hal ini, *repeat* dilakukan sebanyak $N-2$ karena soal sudah memberikan banyaknya organisme bakteri dari hari pertama dan hari kedua. Dengan demikian, jika diminta banyaknya organisme hari ketiga, kita cukup melakukan perhitungan selama satu kali saja. Dalam perulangan ini, perhitungan banyaknya organisme pada hari berikutnya (*hitungH3*) seperti yang dilakukan pada Gambar 7.14 dilakukan kembali di dalam perulangan yang berada di dalam *function* ini. Oleh karena itu, kita memerlukan tiga parameter, yaitu H_1 , H_2 , dan N . Modul ini ditunjukkan pada Gambar 7.15.



Gambar 7.15 Contoh kunci jawaban modul untuk proyek AP-K9-10

Selanjutnya, setelah modul dibuat, fungsi tersebut dipanggil pada program utama yang dimodifikasi dari kode program Gambar 7.14, di mana kali ini yang diminta ialah banyaknya organisme bakteri pada hari ke- N , yang dalam contoh kunci jawaban Gambar 7.16 diberi nama variabel HN . Oleh karena itu, kita juga perlu membaca/menyediakan variabel yang menyimpan banyaknya hari yang diminta, yaitu variabel N . Terakhir, nilai variabel HN ini dicetak sehingga jika Titi memiliki organisme pada hari pertama (H_1) berjumlah 8, pada hari kedua (H_2) berjumlah 10, dan Titi diminta mengamati hingga hari ke-10, banyaknya organisme pada hari ke-10 berjumlah 276.



Gambar 7.16 Contoh kunci jawaban program utama untuk proyek AP-K9-10

Catatan Tambahan: Peserta didik dapat mengamati pertumbuhan yang terjadi setiap hari dengan menambahkan blok [writeLine] dan variabel HN pada modul hitungHN di akhir perintah di dalam blok [repeat]. Proses ini dapat dilakukan sebagai salah satu cara *debugging* untuk mengecek apakah program sudah benar atau belum.

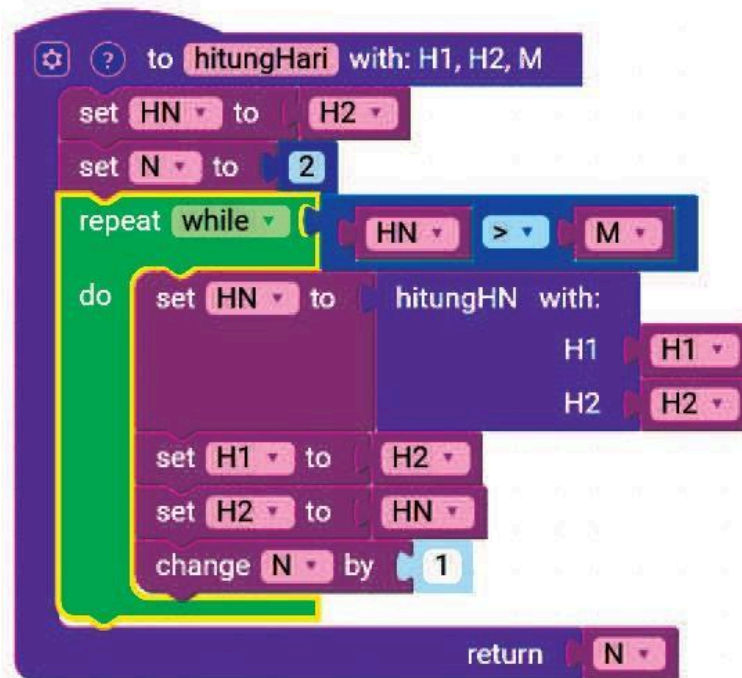
Aktivitas AP-K9-11: Proyek Capaian Pertumbuhan

Pada kasus sebelumnya (AP-K9-10), peserta didik diminta untuk menghitung banyaknya organisme bakteri pada hari yang diminta. Pada kasus ini, peserta didik diminta untuk menghitung banyaknya hari yang diperlukan agar banyaknya organisme bakteri mencapai nilai yang diminta. Dua kasus ini perlu diselesaikan dengan perulangan. Namun demikian, terdapat perbedaan penggunaan perulangan pada kedua kasus tersebut. Pada kasus sebelumnya, banyaknya perulangan dapat diketahui dengan pasti sehingga perulangan dapat dilakukan menggunakan blok [repeat] sebanyak N kali. Namun pada kasus ini, berapa kali perulangan harus dilakukan? Berapa kali program harus mengulang agar banyaknya organisme bakteri mencapai nilai yang diminta? Blok perulangan mana yang harus digunakan?

Jika menemui kasus di mana kita tidak mengetahui secara pasti banyaknya perulangan yang harus dilakukan, perulangan dapat dilakukan dengan menentukan kondisi perulangannya. Kode program yang melakukan perulangan dengan kondisi seperti ini dapat menggunakan blok [repeat_while]. Kondisi perulangan diletakkan pada blok [repeat_while] ini. Peserta didik diajak berdiskusi, “Perulangan pada kode program Gambar 7.15 dilakukan selama apa?” Peserta didik diajak menelusuri kembali contoh input *output* yang diberikan atau contoh input *output* yang dibuat sendiri.

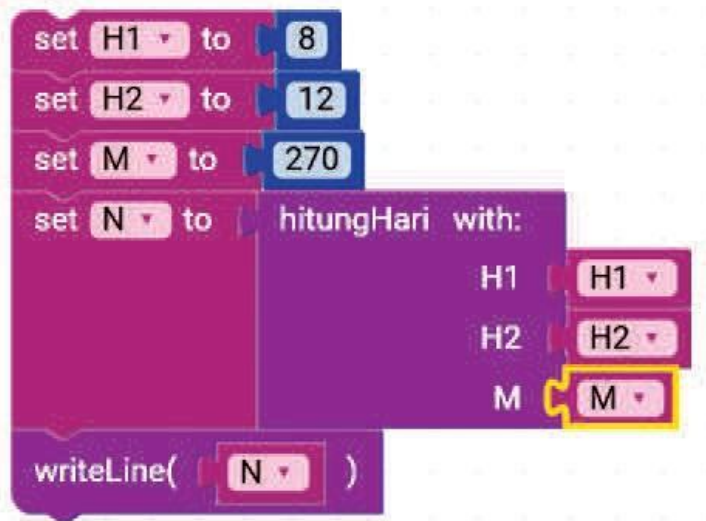
Perulangan dilakukan dengan kondisi: selama banyaknya bakteri yang dihitung pada hari tertentu (hari ke-N) masih kurang dari atau masih sama dengan nilai yang diminta (M). Hal ini dapat ditulis dengan kondisi: $(HN \leq M)$. Setelah itu, blok [repeat_while] ini dapat

diisi dengan perintah- perintah yang serupa dengan perintah di dalam perulangan pada kode program Gambar 7.15. Tidak lupa, setiap melakukan perulangan, perlu dihitung banyaknya perulangan ini, yang juga menyatakan banyaknya hari yang diperlukan untuk mencapai banyaknya organisme bakteri tersebut. Informasi tersebut dapat digunakan sebagai petunjuk untuk meminta peserta didik melengkapi kode programnya.



Gambar 7.17 Contoh kunci jawaban modul untuk proyek AP-K9-11

Setelah 5-10 menit mencoba, guru dapat menunjukkan contoh kode program pada Gambar 7.17 sebagai salah satu alternatif modul untuk persoalan ini dan Gambar 7.18 sebagai program utamanya. Pada modul hitungHari tersebut, beberapa variabel perlu diinisialisasi atau diberi nilai di awal. Variabel HN (banyaknya organisme bakteri) perlu diinisialisasi dengan nilai H2 karena akan dicek dalam perulangan [repeat_while]. Jika tidak diberi nilai terlebih dahulu, bagaimana jika nilai HN terisi nol atau sembarang? Kondisi dalam [repeat_while] sesuai dengan kondisi di hari kedua di mana perhitungan akan dilakukan. Kemudian, variabel N (hari ke berapa) diinisialisasi dengan nilai 2 untuk menunjukkan kondisi awal perhitungan, yaitu setelah hari kedua. Dengan demikian, jika program utama pada Gambar 7.18 dijalankan, akan menghasilkan keluaran N bernilai 10.



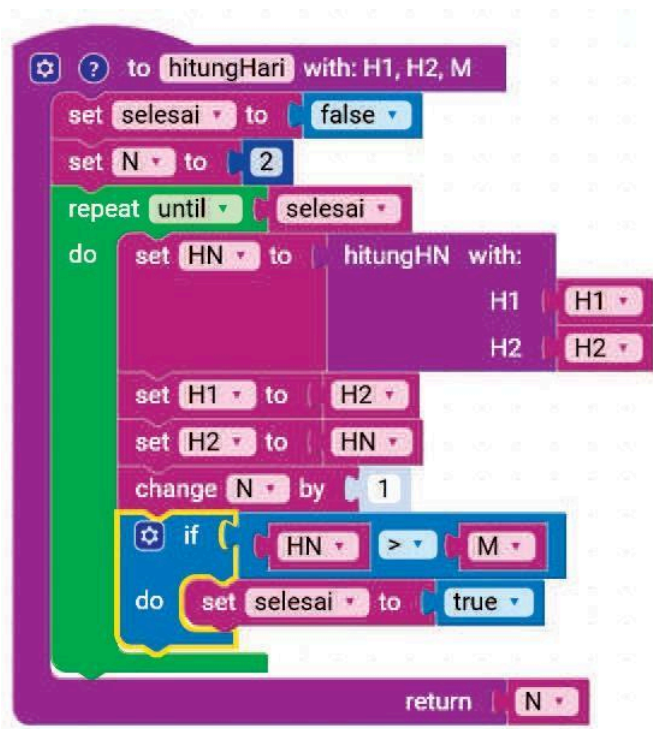
```

set H1 to 8
set H2 to 12
set M to 270
set N to hitungHari with:
  H1 H1
  H2 H2
  M M
writeLine( N )

```

Gambar 7.18 Contoh kunci jawaban program utama untuk proyek AP-K9-11

Sebagai alternatif, kode program pada Gambar 7.17 dapat diubah agar mengandung percabangan seperti diperlihatkan dalam kode program pada Gambar 7.19. Pada program ini, blok perulangan yang digunakan ialah blok [repeat_until]. Perulangan dilakukan hingga kondisi selesai, yaitu jika banyaknya organisme bakteri (HN) sudah melebihi nilai yang diminta (M). Kondisi percabangan ini diletakkan di akhir perulangan [repeat_until]. Dengan demikian, inisialisasi juga melibatkan variabel selesai yang diisi dengan nilai false untuk memastikan agar program masuk ke dalam perulangan [repeat_until].



```

to hitungHari with: H1, H2, M
  set selesai to false
  set N to 2
  repeat until selesai
  do
    set HN to hitungHN with:
      H1 H1
      H2 H2
    set H1 to H2
    set H2 to HN
    change N by 1
    if HN > M
    do
      set selesai to true
  return N

```

Gambar 7.19 Alternatif kunci jawaban modul untuk proyek AP-K9-11

Aktivitas AP-K9-22-U: Gelang Warna-Warni (Tambahan)

I. Tujuan Pembelajaran:

Memahami konsep modularisasi melalui permainan membuat gelang manik-manik yang berwarna-warni.

II. Apersepsi

Aktivitas ini merupakan aktivitas tambahan yang dapat dilakukan guru bersama peserta didiknya apabila ada waktu yang mencukupi. Aktivitas ini ditempatkan di bagian akhir Buku Siswa.

Membuat program berarti menuliskan beberapa baris intruksi. Makin besar program makin banyak baris program ditulis. Makin banyak baris program ditulis, makin sulit untuk memahami program tersebut. Jika sebuah program sulit untuk dipahami, sulit pula untuk dikembangkan dan diperbaiki jika terdapat kesalahan. Oleh karena itu, program perlu disederhanakan. Beberapa kode program dapat dikumpulkan dan diabstraksikan sehingga jika kita membaca program utama, akan terlihat sederhana dan mudah dipahami.

Sebagai pemantik, peserta didik dapat diajak untuk mengerjakan bagian tantangan aktivitas ini. Kegiatan tantangan dapat dilakukan melalui gambar atau demo dari guru menggunakan manik-maniknya. Setelah itu, peserta didik dapat mulai praktik membuat gelang dengan sarana yang disiapkan oleh guru.

III. Kebutuhan Sarana dan Prasarana

Setiap anak dapat disiapkan sarana sebagai berikut.



1. Benang, bisa berupa benang jahit, benang nilon, atau tali apa pun yang berukuran kecil (panjang 30 cm)
2. Manik warna-warni (5 merah, 5 kuning, 5 hijau, 5 biru, 5 abu-abu; atau warna lain yang diinginkan)



Jika kesulitan mendapatkan manik-manik, bisa menggunakan kancing baju warna-warni atau benda apapun yang bisa dimasukkan ke dalam benang.



3. Alat tulis: pensil atau pulpen
4. Lembar Kerja dalam buku ini.

IV. Kegiatan Inti

Peserta didik diminta untuk membuat kreasi manik-manik yang nantinya dapat berupa gelang, kalung atau gantungan kunci. Aktivitas ini terinspirasi dari video pembelajaran pada kegiatan Unplugged Code.org dengan alamat <https://code.org/curriculum/course3/4/Teacher> dan dengan video pembelajaran <https://www.youtube.com/watch?v=Rb5DNYhLb7I>.



sumber gambar: <https://www.youtube.com/watch?v=Rb5DNYhLb7I>

Dalam video tersebut, peserta didik diminta untuk membuat hiasan gantungan (*suncatcher*) dari berbagai macam manik-manik. Dalam kegiatan ini, peserta didik diminta untuk membuat gelang, dengan variasi warna. Namun demikian, kreasi beragam manik-manik juga dapat dibuat sesuai kreativitas dari guru.



sumber gambar: <https://id.pinterest.com/pin/659495939167371364/>
<https://giftsncrystals.com.au/product/glass-seahorse-suncatcher-2/> <https://en.wikipedia.org/wiki/Suncatcher>

Jawaban (salah satu alternatif) dari amplop BuatGelangBunga (Lembar Kerja 1)

Lembar Kerja 1								
	<table border="1"> <thead> <tr> <th>BuatGelangBunga</th> </tr> </thead> <tbody> <tr> <td>Lakukan BuatGelangSatu</td> </tr> <tr> <td>Rangkai Manik-Manik MERAH</td> </tr> <tr> <td>Rangkai Manik-Manik ABU-ABU</td> </tr> <tr> <td>Rangkai Manik-Manik HIJAU</td> </tr> <tr> <td>Rangkai Manik-Manik BIRU</td> </tr> <tr> <td>Lakukan BuatGelangDua</td> </tr> </tbody> </table>	BuatGelangBunga	Lakukan BuatGelangSatu	Rangkai Manik-Manik MERAH	Rangkai Manik-Manik ABU-ABU	Rangkai Manik-Manik HIJAU	Rangkai Manik-Manik BIRU	Lakukan BuatGelangDua
BuatGelangBunga								
Lakukan BuatGelangSatu								
Rangkai Manik-Manik MERAH								
Rangkai Manik-Manik ABU-ABU								
Rangkai Manik-Manik HIJAU								
Rangkai Manik-Manik BIRU								
Lakukan BuatGelangDua								
<table border="1"> <thead> <tr> <th>BuatGelangSatu</th> </tr> </thead> <tbody> <tr> <td>Rangkai Manik-Manik ABU-ABU</td> </tr> <tr> <td>Rangkai Manik-Manik BIRU</td> </tr> </tbody> </table>	BuatGelangSatu	Rangkai Manik-Manik ABU-ABU	Rangkai Manik-Manik BIRU					
BuatGelangSatu								
Rangkai Manik-Manik ABU-ABU								
Rangkai Manik-Manik BIRU								

Rangkai Manik-Manik KUNING

BuatGelangDua

Rangkai Manik-Manik KUNING

Rangkai Manik-Manik MERAH

Rangkai Manik-Manik HIJAU

Atau, jika peserta didik mengamati pola dari Gelang Bunga, ada beberapa manik-manik yang memiliki kesamaan dan dapat dibuat satu amplop tersendiri untuk pola yang sama tersebut, yaitu pola warna BIRU- KUNING-MERAH. Setelah itu, amplop yang lain dapat diisi dengan warna yang lain.

Jawaban (alternatif lain) dari amplop BuatGelangBunga (Lembar Kerja 1)

Lembar Kerja 1	
BuatGelangBunga	
Rangkai Manik-Manik ABU-ABU	
Lakukan BuatGelangSatu	
Lakukan BuatGelangDua	
Lakukan BuatGelangSatu	
Rangkai Manik-Manik HIJAU	

BuatGelangSatu
Rangkai Manik-Manik BIRU
Rangkai Manik-Manik KUNING
Rangkai Manik-Manik MERAH

BuatGelangDua
Rangkai Manik-Manik ABU-ABU
Rangkai Manik-Manik HIJAU

Jawaban dari amplop BuatGelangDian



Setelah itu, peserta didik dapat secara bebas membuat kumpulan instruksi untuk membuat gelang warna-warni dan meminta pasangannya untuk membuat gelang berdasarkan instruksi tersebut. Guru dapat memantau peserta didik jika menemui kesulitan.

Jawaban dari Evaluasi

1. A. BuatGelangAsri dan C. BuatGelangCici
Hasilnya dari dua amplop ini menjadi tujuan utama dari Asri dan Cici.
2. B. BuatGelangBeta dan D. BuatGelangDua
Sebagai bagian dari program utama, dapat digunakan, diduplikasi atau tidak digunakan. BuatGelangBeta digunakan di amplop BuatGelangTeman sedangkan BuatGelangDua digunakan di amplop BuatGelangCici dan BuatGelangDian.
3. C. Gelang C
Gelang ini diawali dengan manik merah, kemudian dilanjutkan dengan Kreasi3 yang diawali dengan manik kuning.
4. D. Kode Program D
Semua jawaban dapat menghasilkan manik-manik yang serupa di awal. Namun, jawaban D tidak mampu menghentikan 'program' karena terus berputar di prosedur Kreasi1 dan Kreasi2. Ini merupakan salah satu contoh *ininite loop*, atau perulangan tiada henti. Hal ini dapat terjadi saat kita salah melakukan pemanggilan prosedur/fungsi.

Kegiatan Alternatif

Jika tidak bisa menyediakan bahan dan peralatannya, kegiatan ini bisa digantikan dengan kegiatan menggambar.

Pertemuan 6: Konverter Sistem Bilangan (2 jp)

I. Tujuan Pembelajaran:

1. Memahami modularisasi program dan model komputasinya
2. Mengembangkan program visual sederhana berdasarkan contoh-contoh yang diberikan

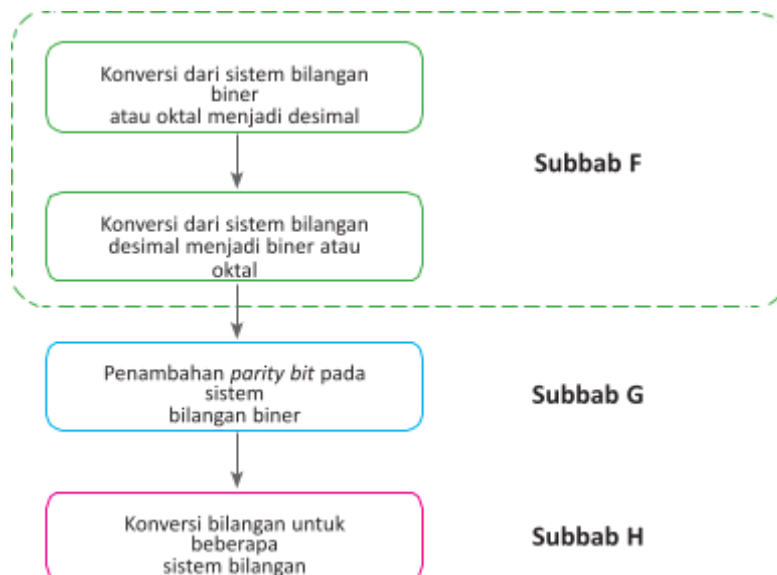
II. Apersepsi

Saat pertama belajar menghitung, alat bantu yang digunakan biasanya ialah jari tangan.

Karena ada 10 jari tangan, sistem bilangan yang paling umum digunakan oleh manusia adalah sistem bilangan desimal di mana digit terendah adalah 0 dan digit tertinggi adalah 9. Komputer berbeda dengan manusia. Komputer hanya mengenal 2 jenis digit, yaitu 0 dan 1. Sistem bilangan yang digit tertingginya 1 disebut sistem bilangan biner. Selain sistem bilangan desimal dan sistem bilangan biner, masih ada sistem bilangan-sistem bilangan lainnya yang sering digunakan juga. Semua sistem bilangan ini memiliki konsep yang sama hanya berbeda pada basis yang digunakan.

III. Pemanasan

Guru dapat mengingatkan peserta didik tentang materi bilangan biner yang pernah dipelajari di kelas VII dan kelas VIII. Guru juga dapat memberi contoh sebuah bilangan yang sama, tetapi dari 2 sistem bilangan yang berbeda. Contoh ini bisa didapat di Buku Siswa. Setelah itu, guru dapat menjelaskan target program yang akan dibuat serta tahapan penyelesaiannya. Tahapan penyelesaian dapat dilihat di diagram berikut ini:



Guru menjelaskan alur kegiatan yang akan mereka lakukan untuk membuat program konverter bilangan di subbab H. Berikan pengertian kepada peserta didik bahwa untuk membuat program yang kompleks, mereka perlu mulai dari program yang lebih sederhana. Konsep ini dinamakan dekomposisi.

IV. Kebutuhan Sarana dan Prasarana

- Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser*.
- Koneksi internet.

V. Kegiatan Inti

Kegiatan inti pada pertemuan ini adalah sebagai berikut.

- Guru menjelaskan makna bilangan biner di sistem bilangan desimal. Guru mengajarkan

peserta didik bagaimana melakukan konversi dari bilangan biner menjadi bilangan desimal.

2. Setelah memberikan contoh, guru dapat meminta peserta didik mengisi tabel untuk mengubah bilangan biner 1011 menjadi bilangan desimal. Guru juga dapat memberi beberapa latihan lagi kepada peserta didik untuk memastikan apakah peserta didik sudah paham bagaimana melakukan konversi bilangan biner menjadi bilangan desimal.
3. Jika peserta didik sudah paham bagaimana melakukan konversi dari bilangan biner menjadi desimal, ajaklah peserta didik menerapkan konsep tersebut untuk melakukan konversi dari bilangan oktal menjadi desimal dengan mengisi tabel konversi bilangan oktal 1011. Mintalah satu atau dua orang peserta didik untuk menjelaskan proses konversi yang dilakukan. Guru dapat memberikan beberapa latihan kepada peserta didik untuk memastikan peserta didik sudah paham bagaimana melakukan konversi bilangan oktal menjadi desimal. Saat peserta didik menggunakan konsep konversi bilangan biner menjadi desimal untuk melakukan konversi bilangan oktal menjadi desimal, peserta didik tersebut sedang melakukan abstraksi.
4. Agar peserta didik dapat menguji apakah konversi yang mereka lakukan sudah benar secara mandiri, guru mengenalkan konverter *online* yang sudah banyak tersedia. Salah satu konverter *online* tersedia pada link <https://www.rapidtables.com/convert/number/index.html>.
5. Kegiatan berikutnya ialah mengerjakan Aktivitas AP-K9-12: Konverter Bilangan Biner ke Desimal dan Aktivitas AP-K9-13: Konverter Bilangan Oktal ke Desimal secara mandiri. Dalam pembuatan program, guru perlu mengingatkan peserta didik untuk menguji program yang dibuat. Mintalah peserta didik untuk membuat beberapa contoh masukan dan keluaran untuk menguji program mereka. Ingatkan peserta didik bahwa penilaian tidak hanya dari kebenaran keluaran program, tetapi juga struktur program yang dibuat. Peserta didik harus membuat *function* sesuai instruksi dari masing-masing aktivitas.
6. Selanjutnya, guru memberikan penjelasan mengenai konversi dari sistem bilangan desimal ke sistem bilangan biner dan oktal, sesuai dengan penjelasan yang ada pada Buku Siswa. Guru juga dapat memberikan beberapa latihan sebelum peserta didik melakukan aktivitas individu.
7. Kegiatan berikutnya adalah mengerjakan Aktivitas AP-K9-14: Konverter Bilangan Desimal ke Biner dan Aktivitas AP-K9-15: Konverter Bilangan Desimal ke Oktal. Dalam pembuatan program, guru perlu mengingatkan peserta didik untuk memeriksa program yang dibuat. Mintalah peserta didik untuk membuat beberapa contoh masukan dan keluaran untuk menguji keluaran dari program yang mereka buat. Selain memastikan keluaran program sudah benar, peserta didik juga harus membuat *function* sesuai dengan

instruksi dari masing-masing aktivitas.

8. Setelah semua aktivitas dikerjakan, peserta didik diingatkan untuk mengisi refleksi pada jurnal mereka.
9. Pada pertemuan ini, disediakan tantangan untuk peserta didik yaitu membuat konversi desimal ke heksadesimal dan sebaliknya. Guru dapat menyesuaikan waktu dan kemampuan peserta didik untuk tantangan ini. Tantangan juga dapat diberikan sebagai nilai tambahan untuk peserta didik yang dapat menyelesaikannya.

Jawaban tabel konversi bilangan biner 1011 menjadi bilangan desimal

Tabel 7.8 Jawaban Tabel Konversi Bilangan Biner 1011 Menjadi Bilangan Desimal

Bilangan Biner 1011			
Digit pada bilangan basis 2	Posisi	Perpangkatan dari posisi	Arti
1	0		
1	1		
0	2		
1	3		
Bilangan desimal hasil konversi 1011 adalah			

Jawaban tabel konversi bilangan oktal 1011 menjadi bilangan desimal

Tabel 7.9 Jawaban Tabel Konversi Bilangan Oktal 1011 Menjadi Bilangan Desimal

Bilangan Oktal 1011			
Digit pada bilangan basis 8	Posisi	Perpangkatan dari posisi	Arti
1	0		
1	1		
0	2		
1	3		
Bilangan desimal hasil konversi 1011 adalah			

Jawaban Aktivitas AP-K9-12: Konverter Bilangan Biner ke Desimal

Pada algoritma yang diberikan di Buku Siswa, perulangan pada poin A dilakukan sampai seluruh digit dari variabel bilBiner selesai diproses. Semua digit pada bilBiner selesai diproses ditandai dengan variabel bilBiner berisi 0 yang artinya tidak ada lagi digit yang belum dikonversi ke desimal. Ekspresi pengulangan:

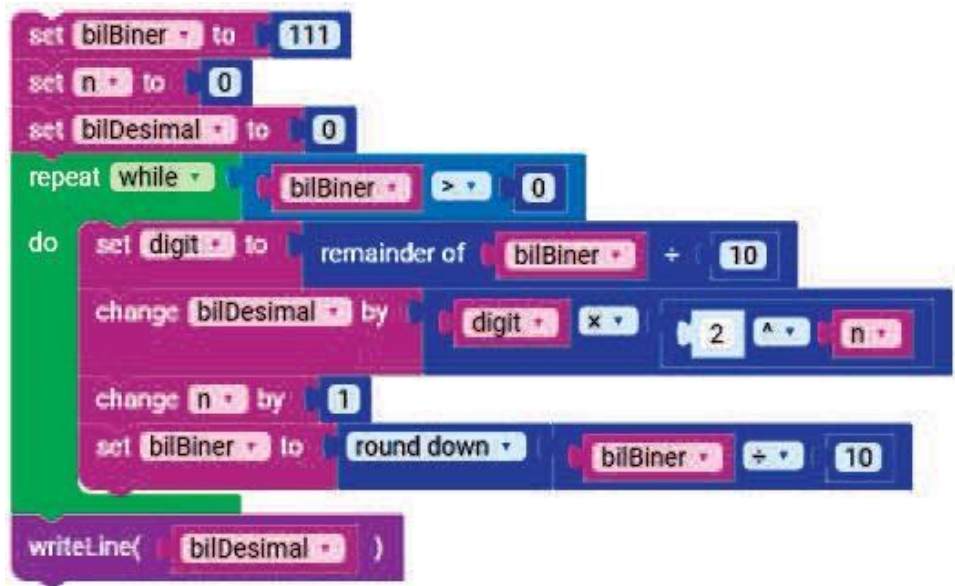
repeat while (bilBiner > 0) Ekspresi tersebut bisa juga

diganti dengan *repeat until*:

repeat until (bilBiner = 0)

Blok kode untuk melakukan konversi bilangan biner menjadi desimal (isi dari

variabel bilBiner dapat diganti-ganti bergantung pada bilangan yang ingin dikonversi).



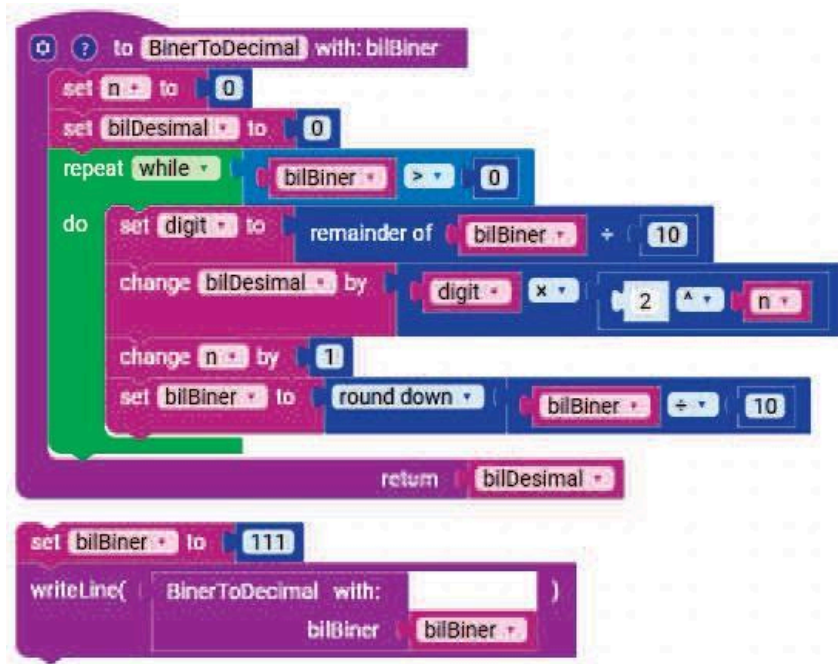
Gambar 7.20 Contoh kunci jawaban Aktivitas AP-K9-12

Urutan blok kode peserta didik tidak harus sama persis dengan kunci jawaban pada Gambar 7.20. Beberapa blok bisa saja tertukar urutannya. Peserta didik juga mungkin membuat perintah yang berbeda dengan kunci jawaban. Ingatkan peserta didik untuk menuliskan hasil pengujian mereka di lembar kerja.

Catatan:

- Ingatkan peserta didik untuk membuat nama variabel yang sesuai dengan maknanya.
- Pada Buku Siswa, diberikan saran untuk penamaan *ile*, tetapi guru bisa mengubah format penamaan *ile* sesuai dengan kondisi kelas.

Modifikasi program dengan membungkus semua blok untuk melakukan konversi bilangan biner ke desimal dalam sebuah fungsi.



Gambar 7.21 Contoh modifikasi kunci jawaban Aktivitas AP-K9-12

Mengapa perlu membungkus bagian program di dalam *function*?

Agar peserta didik merasakan manfaat dari modifikasi program, guru dapat membahas target dari model komputasi ini. Target akhir dari model komputasi ialah membuat konverter dari berbagai sistem bilangan. Pengguna dimungkinkan untuk memilih sistem bilangan asal dan tujuan. Untuk membuat program tersebut, dibutuhkan modul-modul yang lebih sederhana untuk melakukan konversi dari sebuah sistem bilangan ke sistem bilangan lainnya, salah satunya *function* BinerToDecimal.

Jawaban untuk pertanyaan pada aktivitas.



1. Variabel *n* diisi dengan 0 karena variabel *n* digunakan sebagai pangkat dari 2 tergantung digit yang sedang diproses. Perpangkatan 2 dimulai dari 0 (posisi digit paling kanan). Jika variabel *n* tidak diisi dengan 0, hasil dari program adalah undefined karena pada bagian variabel *n* belum memiliki nilai. Blockly tidak mengisi secara otomatis variabel yang dibuat sehingga pengguna perlu mengisi variabel dengan nilai yang diinginkan.



2. Variabel *bilDesimal* digunakan untuk menampung nilai desimal dari digit-digit

bilangan biner yang sudah diproses. Pengisian variabel bilDesimal adalah dengan menjumlahkan hasil pemrosesan digit pada bilangan biner ke variabel bilDesimal. Agar nilai awal variabel bilDesimal tidak mengubah hasil penjumlahan dari pemrosesan digit- digit pada bilangan biner, variabel desimal diisi dengan identitas dari penjumlahan yaitu 0. Pada kasus variabel bilDesimal, sebenarnya tanpa memberi nilai awal pada variabel ini, program masih bisa berjalan dengan benar karena perintah akan mengisi variabel bilDesimal dengan 0 secara otomatis jika ditemukan variabel bilDesimal masih belum memiliki nilai.

Catatan: walau tanpa memberi nilai awal untuk variabel bilDesimal program tetap berjalan dengan benar, biasakan peserta didik untuk memberi nilai awal variabel yang diisi dengan cara menambahkan

nilai ke variabel tersebut. Hal ini dikarenakan tidak semua bahasa pemrograman mengisi sebuah variabel dengan nilai 0 secara otomatis.

3. Tujuan dari pembulatan ke bawah variabel bilBiner yang dibagi 10 adalah untuk menghilangkan digit satuan dari variabel bilBiner. Jika tidak dibulatkan, variabel bilBiner tidak lagi berupa bilangan bulat setelah dibagi 10.

Contoh: $1101/10 = 110,1$.

Agar 1 di belakang koma hilang bilangan 110,1 dibulatkan ke bawah sehingga menjadi 110.

Pada kasus konversi bilangan biner menjadi desimal, penggunaan blok [round] dan [round down] akan menghasilkan hasil yang sama. Hal ini disebabkan hanya ada digit 0 dan 1 sehingga digit berapapun yang ada di belakang koma akan menyebabkan pembulatan ke bawah walau blok yang digunakan adalah blok [round]. Namun, penggunaan blok [round] akan bermasalah ketika konversi bilangan oktal ke desimal atau sebaliknya (desimal ke biner atau desimal ke oktal). Hal ini dikarenakan ada digit-digit di atas 5 sehingga pembulatan menyebabkan bilangan bertambah 1.

Contoh: 12,7 dibulatkan menjadi 13.

4. Agar pada akhir program, bilangan bilBiner dapat ditampilkan lagi, nilai bilangan bilBiner perlu disimpan di variabel lain sebelum variabel bilBiner diproses. Dengan demikian, sekalipun nilai variabel bilBiner berubah, tetapi nilai awal bilBiner tetap disimpan dalam variabel lain tersebut.



Jawaban Aktivitas AP-K9-13: Konverter Bilangan Oktal ke Desimal

Untuk membuat konverter bilangan oktal ke desimal, peserta didik dapat memodifikasi program yang sudah dibuat sebelumnya. Pastikan peserta didik sudah

menuliskan blok-blok yang perlu diubah dan apa perubahannya sebelum peserta didik membuat program. Hal ini untuk mendeteksi apakah peserta didik paham algoritma konversi. Jika peserta

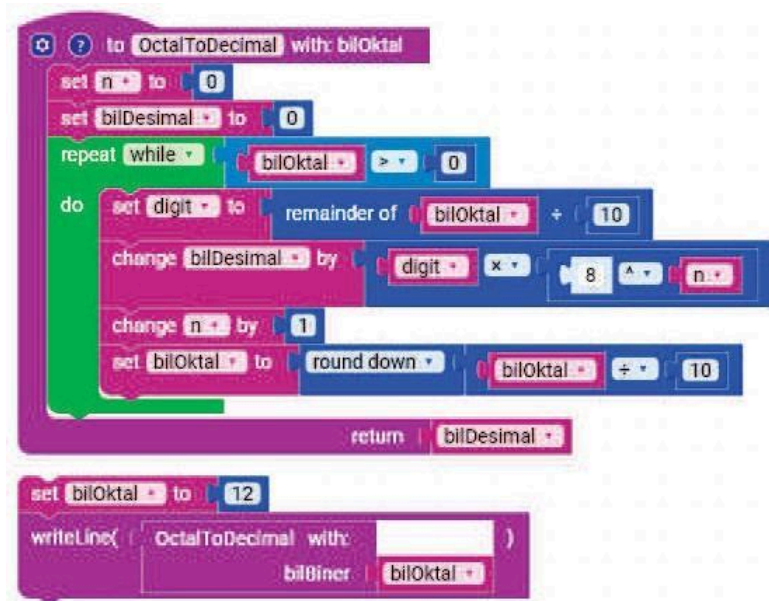
didik mampu menentukan perubahan apa saja yang perlu dilakukan, peserta didik tersebut memahami jalannya algoritma konversi yang diberikan.

Tabel 7.10 Jawaban Aktivitas AP-K9-12B: Konverter Bilangan Oktal ke Desimal

No	Blok pada AP-K9-12b	Perubahan yang Dilakukan
1	Nama variabel bilBiner	Nama variabel bilOktal
2	Nama <i>function</i> BinerToDecimal	Nama <i>function</i> OctalToDecimal
3		

Perubahan yang paling utama ialah perubahan pada nomor 3. Dengan melakukan perubahan pada nomor 3, sekalipun nama variabel dan *function* tidak berubah, program tetap melakukan konversi bilangan oktal menjadi desimal. Namun, ingatkan peserta didik untuk tetap mengubah nama variabel dan nama *function*, agar penamaan pada program tidak membingungkan.

Blok untuk program konversi bilangan oktal menjadi desimal dapat dilihat pada Gambar 7.22.



```

to OctalToDecimal with: bilOktal
  set n to 0
  set bilDesimal to 0
  repeat while bilOktal > 0
  do
    set digit to remainder of bilOktal / 10
    change bilDesimal by digit * 8 ^ n
    change n by 1
    set bilOktal to round down bilOktal / 10
  return bilDesimal

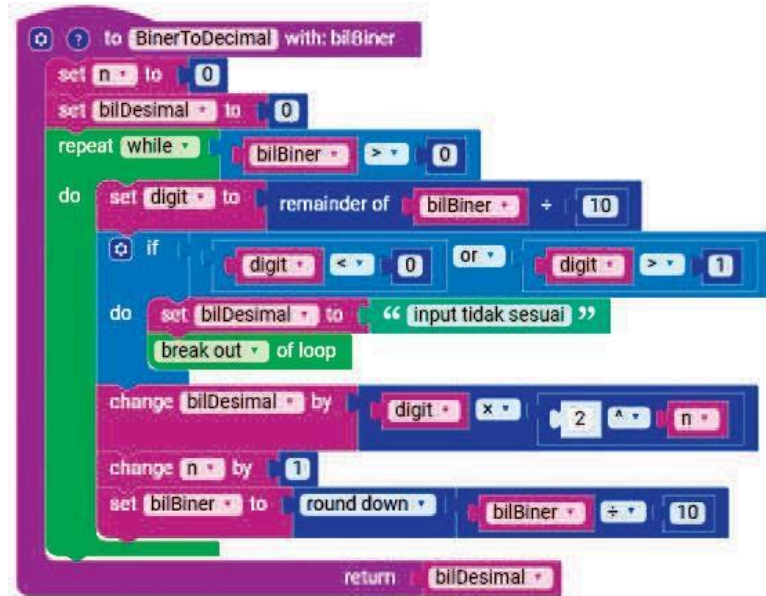
set bilOktal to 12
writeLine( "OctalToDecimal with: " + bilBiner + " " + bilOktal )
  
```

Gambar 7.22 Contoh kunci jawaban Aktivitas AP-K9-13

Jawaban pengayaan

Jika ada masukan yang tidak sesuai dengan sistem bilangan yang seharusnya, program akan tetap mengeluarkan hasil, tetapi tidak sesuai dengan yang seharusnya. Jika program tidak memeriksa apakah inputnya sesuai atau tidak, kesalahan tidak terdeteksi dan terasa seperti tidak ada yang salah.

Ide perbaikan: memeriksa apakah setiap digit pada masukan sesuai dengan sistem bilangan yang sedang digunakan. Apabila ada digit yang tidak benar, program akan mengembalikan tulisan “input tidak sesuai“. Berikut ini merupakan contoh modifikasi pada fungsi BinerToDecimal. Pada contoh ini, jika ditemukan digit selain 0 dan 1, fungsi akan mengembalikan tulisan “input tidak sesuai“.



Sebagai contoh, jika pada program utama bilBiner diisi dengan 12, keluarannya adalah input tidak sesuai.



Jawaban untuk tabel konversi bilangan 33 pada sistem bilangan desimal ke sistem bilangan biner

Tabel 7.11 Jawaban Konversi Bilangan Desimal ke Biner

Bilangan Desimal 33				
Digit Pada Bilangan Basis 10	Operasi Pembagian dengan Basis	Bilangan Bulat Hasil Pembagian	Sisa Hasil Bagi	Hasil
33	33/2	16	1	1
16	16/2	8	0	01
8	8/2	4	0	001
4	4/2	2	0	0001
2	2/2	1	0	00001
1	1/2	0	1	100001

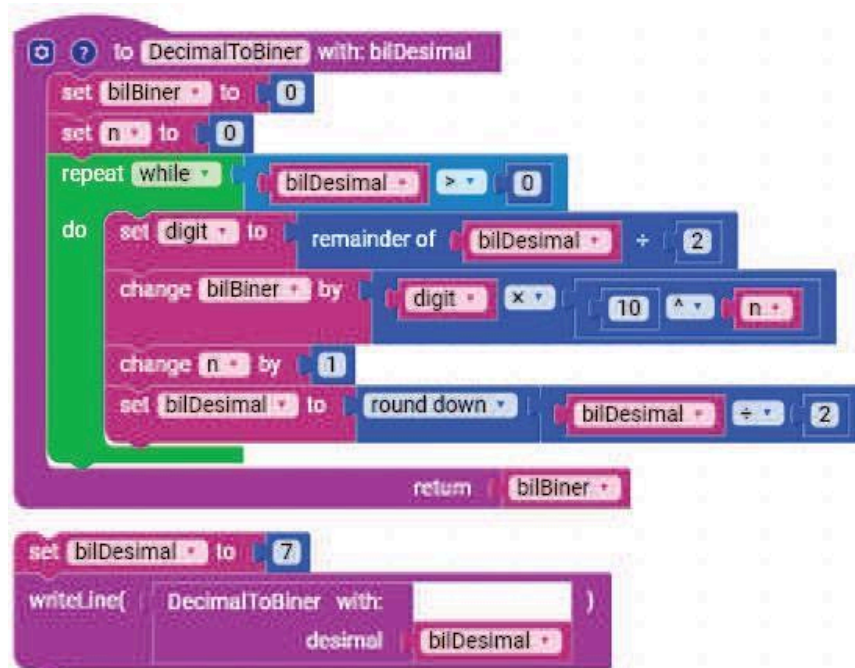
Jawaban untuk tabel konversi bilangan 33 pada sistem bilangan desimal ke sistem bilangan oktal

Tabel 7.12 Jawaban Konversi Bilangan Desimal ke Oktal

Bilangan Desimal 33				
Digit Pada Bilangan Basis 10	Operasi Pembagian dengan Basis	Bilangan Bulat Hasil Pembagian	Sisa Hasil Bagi	Hasil
33	33/8	4	1	1
4	4/8	0	4	41
Bilangan oktal hasil konversi 33 adalah				41

Jawaban Aktivitas AP-K9-14: Konverter Bilangan Desimal ke Biner

Implementasi algoritma yang diberikan di Buku Siswa:



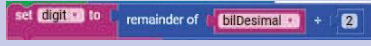



Sekarang peserta didik sudah memiliki konverter DecimalToBiner dan juga BinerToDecimal. Untuk memeriksa kebenaran program mereka, peserta didik tidak perlu lagi menggunakan konverter *online*. Mereka bisa mencocokkan hasil dari DecimalToBiner dengan program mereka BinerToDecimal. Apakah hasilnya sesuai? Jika ya, program mereka kemungkinan sudah benar.

Jawaban Aktivitas AP-K9-15: Konverter Bilangan Desimal ke Oktal

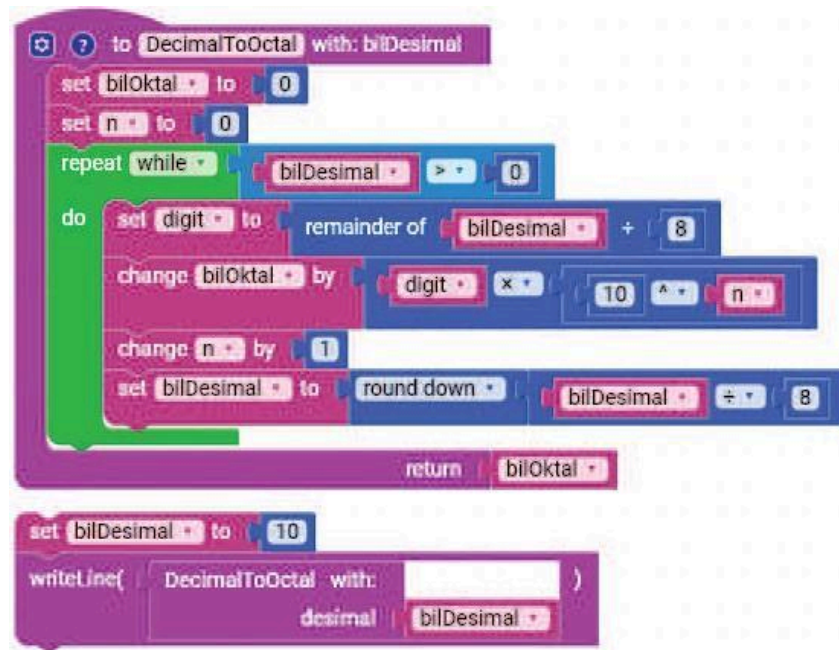
Untuk membuat konverter bilangan desimal ke oktal, peserta didik dapat memodifikasi program yang sudah dibuat sebelumnya (konverter bilangan desimal

ke biner). Pastikan peserta didik sudah menuliskan blok-blok yang perlu diubah dan apa perubahannya sebelum peserta didik membuat program. Hal ini untuk mendeteksi apakah peserta didik paham algoritma konversi. Jika peserta didik mampu menentukan perubahan apa saja yang perlu dilakukan, peserta didik tersebut memahami jalannya algoritma konversi yang diberikan.

Tabel 7.13 Jawaban Aktivitas AP-K9-14: Konverter Bilangan Desimal ke Oktal

No	Blok pada AP-K9-14	Perubahan yang Dilakukan
1	Nama variabel bilBiner	Nama variabel bilOktal
2	Nama <i>function</i> DecimalToBiner	Nama <i>function</i> DecimalToOctal
3		
4		

Blok untuk konversi bilangan desimal ke oktal dapat dilihat pada Gambar 7.23.



Gambar 7.23 Contoh kunci jawaban Aktivitas AP-K9-15

Ingatkan peserta didik untuk menguji kebenaran dari program yang sudah mereka buat. Ajak peserta didik untuk memeriksa benar atau tidaknya program mereka dengan mencocokkan hasil DecimalToOctal dengan OctalToDecimal.

Jawaban Tantangan

- a. Konversi 4000 dari desimal menjadi heksadesimal:

Tabel 7.14 Jawaban Tantangan Konversi Desimal ke Heksadesimal

Bilangan Desimal 4000				
Digit Pada Bilangan Basis 10	Operasi Pembagian dengan Basis	Bilangan Bulat Hasil Pembagian	Sisa Hasil Bagi	Hasil
4000	4000/16	250	0	0
250	250/16	15	10	A0
15	15/10	0	15	FA0
Bilangan heksadesimal hasil konversi 4000				FA0

b. Konversi 3A41 dari heksadesimal menjadi desimal:

Tabel 7.15 Jawaban Tantangan Konversi Heksaesimal ke Desimal

Bilangan Heksadesimal 3A41			
Digit pada bilangan basis 16	Posisi	Perpangkatan dari posisi	Arti
1	0		
4	1		
A	2		
3	3		
Bilangan desimal hasil konversi 3A41			

Kode untuk mengubah bilangan heksadesimal menjadi bilangan desimal

```
to HexaToDecimal with: bilHeksa
  set n to 0
  set bilDesimal to 0
  set length to length of bilHeksa
  count with i from length to 1 by 1
  do
    set digit to in text bilHeksa get letter # i
    if digit = "F"
      do set digit to 15
    else if digit = "E"
      do set digit to 14
    else if digit = "D"
      do set digit to 13
    else if digit = "C"
      do set digit to 12
    else if digit = "B"
      do set digit to 11
    else if digit = "A"
      do set digit to 10
    change bilDesimal by digit * 16 ^ n
    change n by 1
  return bilDesimal
```

```
set bilHeksa to "3A41"
writeLine( / HexaToDecimal with:
           heksa bilHeksa )
```

Kode untuk mengubah bilangan desimal menjadi heksadesimal

```

to DecimalToHexa with: bilDesimal
  set n to 0
  set bilHeksa to ""
  repeat while bilDesimal > 0
  do
    set digit to remainder of bilDesimal + 16
    if digit = 15
    do
      set item to "F"
      to item append text bilHeksa
      set bilHeksa to item
    else if digit = 14
    do
      set item to "E"
      to item append text bilHeksa
      set bilHeksa to item
    else if digit = 13
    do
      set item to "D"
      to item append text bilHeksa
      set bilHeksa to item
    else if digit = 12
    do
      set item to "C"
      to item append text bilHeksa
      set bilHeksa to item
    else if digit = 11
    do
      set item to "B"
      to item append text bilHeksa
      set bilHeksa to item
    else if digit = 10
    do
      set item to "A"
      to item append text bilHeksa
      set bilHeksa to item
    else
      set item to digit
      to item append text bilHeksa
      set bilHeksa to item
    set bilDesimal to round down bilDesimal + 16
  return bilHeksa

```

```

set bilDesimal to 5039
writeLine( DecimalToHexa with:
           decimal bilDesimal )

```

Pertemuan 7: Penambahan *Parity Bit* pada Sistem Bilangan Biner (2 jp)

I. Tujuan Pembelajaran:

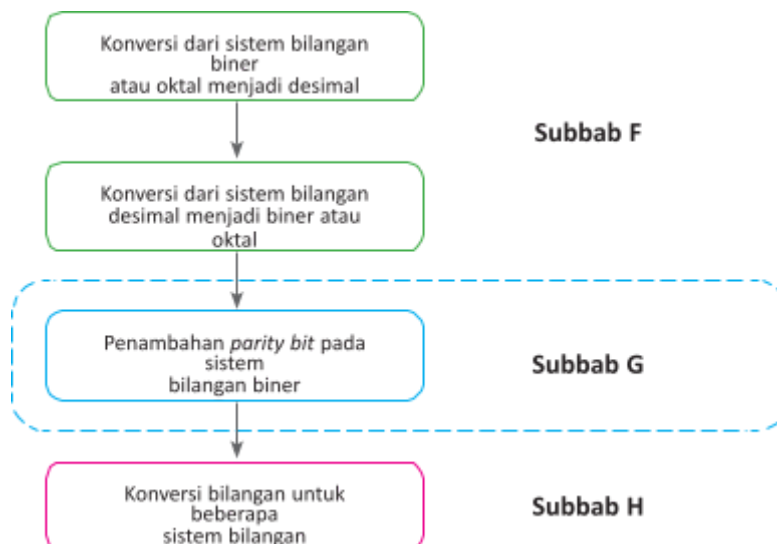
1. Memahami modularisasi program dan model komputasinya
2. Mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan

II. Apersepsi

Dalam kehidupan sehari-hari, sering terjadi kesalahan dalam penyampaian pesan sekalipun pengirim pesan sudah menyampaikan pesan dengan benar. Mungkin saja pengirim pesan mengalami kendala atau ketika disampaikan langsung pun mungkin saja penerima pesan salah mendengar. Dalam pengiriman pesan melalui jaringan komputer juga mungkin terjadi kesalahan. Mungkin saja ada kendala jaringan atau ada orang yang sengaja mengubah pesan yang sedang dikirimkan. Untuk dapat melacak apakah terjadi kesalahan pada pesan yang dikirim, salah satu metodenya ialah dengan menambahkan *parity bit*.

III. Pemanasan

Guru dapat mengulang materi pada pertemuan sebelumnya untuk mengingatkan peserta didik mengenai sistem bilangan. Kemudian, guru mengingatkan peserta didik tentang posisi mereka saat ini dalam tahapan pembuatan program konverter bilangan. Dari diagram di bawah, saat ini, peserta didik sedang dalam tahap untuk menambahkan *parity bit* pada sistem bilangan biner.



Agar peserta didik mengerti pentingnya penambahan *parity bit*, guru dapat menjelaskan mengenai proses pengiriman data secara sederhana dan kesalahan yang mungkin terjadi.

IV. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser*.

b. Koneksi internet.

V. Kegiatan Inti

Kegiatan inti pada pertemuan ini adalah sebagai berikut:

1. Guru menjelaskan apa itu *parity bit* dan fungsinya. Peserta didik diberi waktu untuk mengerjakan latihan yang sudah tersedia di Buku Siswa agar peserta didik paham mengenai *parity bit* genap dan *parity bit* ganjil.
2. Peserta didik diarahkan untuk melakukan kegiatan *unplugged* Aktivitas AP-K9-16-U: Pengantar Pesan. Melalui kegiatan ini, ajaklah peserta didik untuk mendapat jawaban dari pertanyaan yang terdapat pada Buku Siswa. Setelah setiap kelompok mencoba, ajak peserta didik berdiskusi hasil dari kegiatan yang mereka lakukan.
3. Kegiatan berikutnya ialah mengerjakan Aktivitas AP-K9-17: Konverter Bilangan Desimal ke Biner dengan *Parity Bit* Genap dan Aktivitas AP-K9-18: Konverter Bilangan Biner dengan *Parity Bit* Genap ke Desimal. Selama mengerjakan aktivitas, ingatkan peserta didik untuk memahami algoritma yang diberikan dan tidak hanya sekedar mengubah algoritma menjadi kode.
4. Setelah selesai mengerjakan aktivitas, jika masih tersedia waktu peserta didik bisa diminta untuk mencoba membuat konverter desimal ke biner dan sebaliknya tetapi dengan *parity bit* ganjil. Hal ini untuk memastikan apakah peserta didik paham algoritma yang mereka kerjakan sehingga dapat memodifikasi kode blok untuk melakukan kegiatan yang berbeda.
5. Setelah semua aktivitas dikerjakan, peserta didik diingatkan untuk mengisi refleksi pada jurnal mereka.

Jawaban Aktivitas AP-K9-17: Konverter Bilangan Desimal ke Biner dengan Parity Bit Genap dan Aktivitas AP-K9-18:

Konverter Bilangan Biner dengan *Parity Bit* Genap ke Desimal

1. Bilangan biner dengan *parity bit* genap untuk setiap bilangan desimal berikut ialah:
 - a. Desimal: 10, Biner (*parity bit* genap): 10100
 - b. Desimal: 13, Biner (*parity bit* genap): 11011
2. Apakah bilangan biner yang sudah ditambahkan oleh *parity bit* genap merupakan bilangan yang benar?
 - a. 111 : Tidak, karena banyaknya bit satu pada bilangan ini adalah ganjil.
 - b. Ya. Hasil konversi bilangan tersebut ke bilangan desimal adalah 6.

Jawaban dari Ayo Renungkan

1. Ya, saat bit yang berubah sebanyak genap.

Misal: bilangan yang sesungguhnya adalah 1100 pada sistem bilangan biner dengan *parity bit* genap. Karena ada kesalahan dalam

pengiriman, bilangan tersebut berubah menjadi 1010. Ada 2 bit yang berubah, akan tetapi 1010 tetap dianggap bilangan yang benar pada sistem bilangan biner dengan *parity bit* genap.

2. Karena posisi dan banyaknya bit yang berubah tidak diketahui.

Melalui kegiatan *unplugged*, peserta didik dituntun untuk mencari pola kesalahan-kesalahan seperti apa yang tidak terdeteksi dengan menggunakan *parity bit*. Mungkin ada juga peserta didik yang langsung mengetahui jawabannya sebelum melakukan kegiatan. Mintalah mereka untuk tetap melakukan kegiatan dan membantu teman mereka untuk menemukan polanya.

Jawaban Aktivitas AP-K9-17: Konverter Bilangan Desimal ke Biner dengan *Parity Bit* Genap

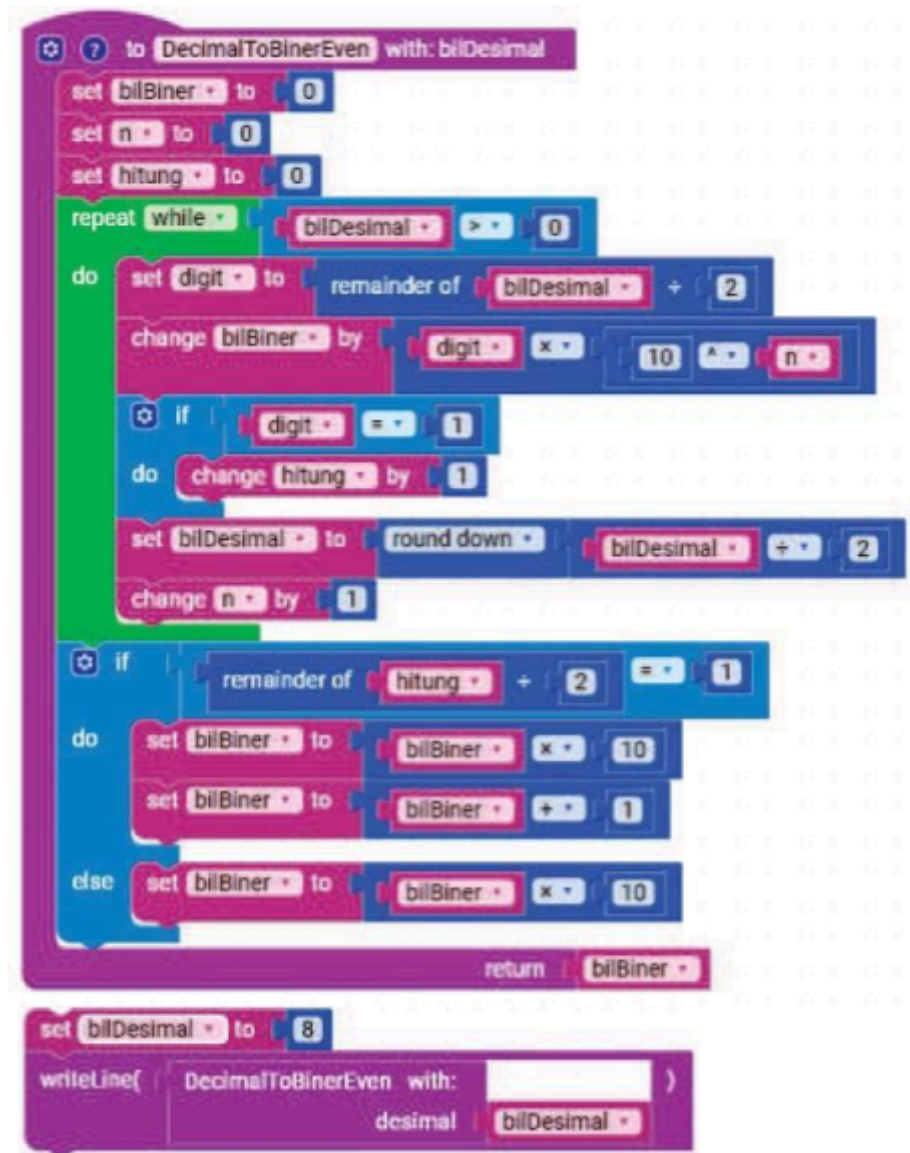
Untuk menambahkan sebuah digit di akhir bilangan, bilangan yang akan ditambah dikali dengan 10 terlebih dahulu untuk memberi tempat satuan ke digit yang akan ditambahkan.

Contoh: 1 ingin ditambahkan di akhir bilangan 10.

Langkah:

Bilangan yang ingin ditambahkan dikali dengan 10: Setelah dikali 10, tambahkan digit yang baru:

Blok untuk program konversi bilangan desimal ke biner dengan *parity bit* genap dapat dilihat pada Gambar 7.24.



Gambar 7.24 Contoh kunci jawaban Aktivitas AP-K9-17

Jawaban Pengayaan

Variabel hitung dapat diubah dengan menambahkan variabel digit yang ditemukan ke variabel hitung. Karena saat digit adalah 1, variabel hitung ditambah 1, sedangkan saat digit adalah 0, variabel hitung ditambah 0. Jadi, tanpa harus melakukan pemeriksaan terhadap isi variabel digit, variabel hitung bisa langsung ditambahkan dengan digit. Untuk jawaban aktivitas-

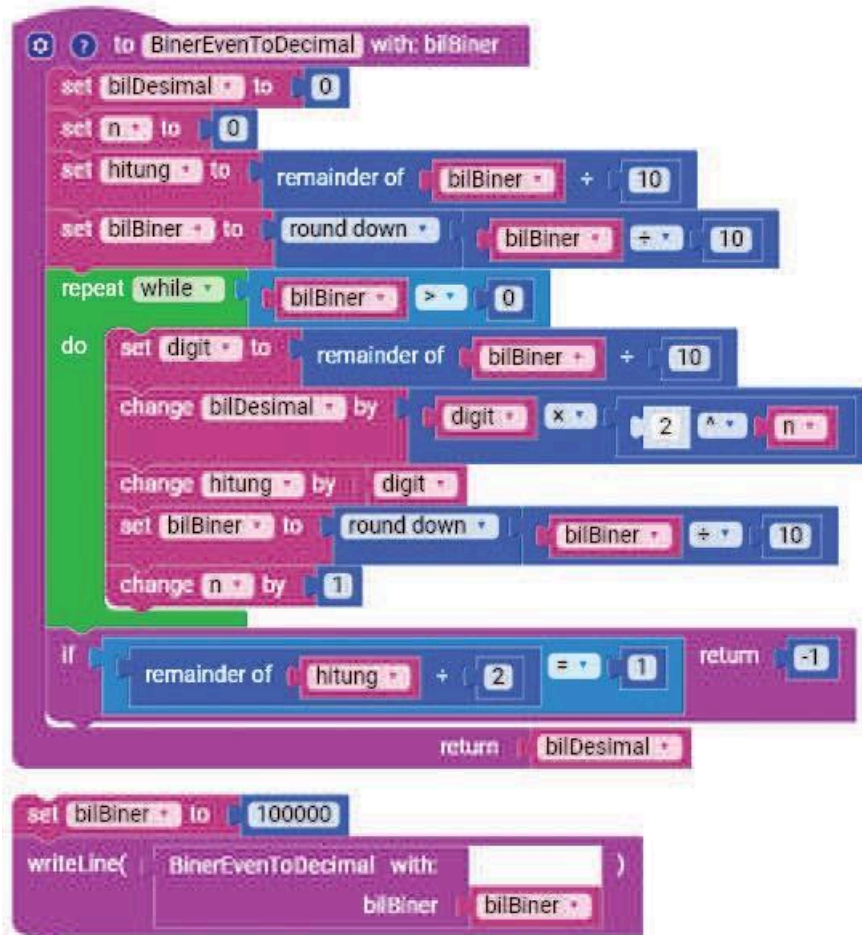
aktivitas berikutnya, variabel hitung diubah dengan cara ditambahkan dengan variabel digit.

Jawaban Aktivitas AP-K9-18: Konverter Bilangan Biner dengan

Parity Bit Genap ke Desimal

Blok untuk program konversi bilangan biner dengan *parity bit* genap ke desimal

dapat dilihat pada Gambar 7.25.



Gambar 7.25 Contoh kunci jawaban Aktivitas AP-K9-18

Setelah membuat program, ingatkan peserta didik untuk mengisi relesi pada lembar kerja mereka. Jika ada waktu, ajak peserta didik memodiikasi program yang mereka buat menjadi *parity bit* ganjil.

Pertemuan 8: *Wrap Up* Konverter Bilangan (2 jp)

I. Tujuan Pembelajaran:

1. Memahami modularisasi program dan model komputasinya
2. Mengembangkan program visual sederhana berdasarkan contoh- contoh yang diberikan

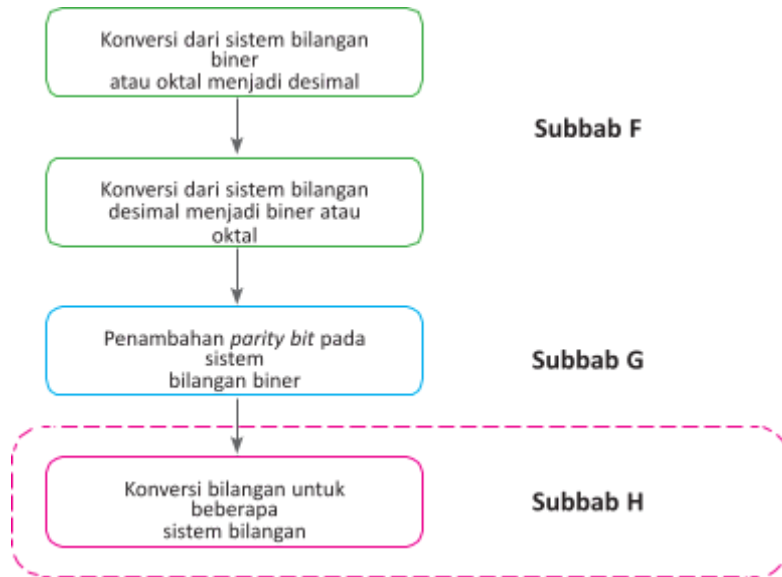
II. Apersepsi

Guru bisa menanyakan pengalaman peserta didik ketika bepergian ke luar kota. Misalnya seorang peserta didik yang berada di kota A ingin pergi ke kota C. Akan tetapi, tidak ada transportasi yang dapat mengantarnya langsung dari kota A ke kota C. Apakah artinya peserta didik tersebut tidak dapat pergi ke kota C? Tidak ada transportasi yang dapat mengantar langsung tidak selalu berarti peserta didik tersebut tidak dapat pergi ke kota tujuan. Orang bisa mencari alternatif seperti misalnya transportasi yang mengantar dari A ke B, kemudian

dari B ke C. Dalam penggunaan fungsi, terkadang orang perlu menggunakan beberapa fungsi untuk mencapai tujuan.

III. Pemanasan

Guru bisa membahas materi pada subbab-subbab sebelumnya, serta *function* yang sudah pernah dibuat sebelumnya untuk mengingatkan peserta didik tentang materi yang sudah dipelajari. Ingatkan peserta didik tahapan-tahapan yang sudah mereka lalui dan apa yang akan mereka kerjakan sekarang dengan diagram di bawah ini.



IV. Kebutuhan Sarana dan Prasarana

- Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser*.
- Koneksi internet.

V. Kegiatan Inti

Kegiatan inti pada pertemuan ini adalah sebagai berikut:

- Guru mengajak peserta didik untuk membuat kelompok yang berisi 2 orang. Di dalam kelompok, peserta didik membuat konverter bilangan untuk menyelesaikan Aktivitas AP-K9-19: Konverter Biner, Oktal, dan Desimal
- Kegiatan berikutnya ialah setiap kelompok membuat 2 buah poster untuk menyelesaikan Aktivitas AP-K9-20: Poster Konverter Bilangan dan Aktivitas AP-K9-21: Poster Serunya Belajar Dua Bahasa.

Jawaban Aktivitas AP-K9-19: Konverter Biner, Oktal, dan Desimal

Dari 4 *function* yang sudah pernah dibuat sebelumnya, untuk membuat konverter antar sistem bilangan, dapat digunakan dengan aturan berikut.

Tabel 7.16 Jawaban Aktivitas AP-K9-19: Konversi Biner, Oktal, dan Desimal

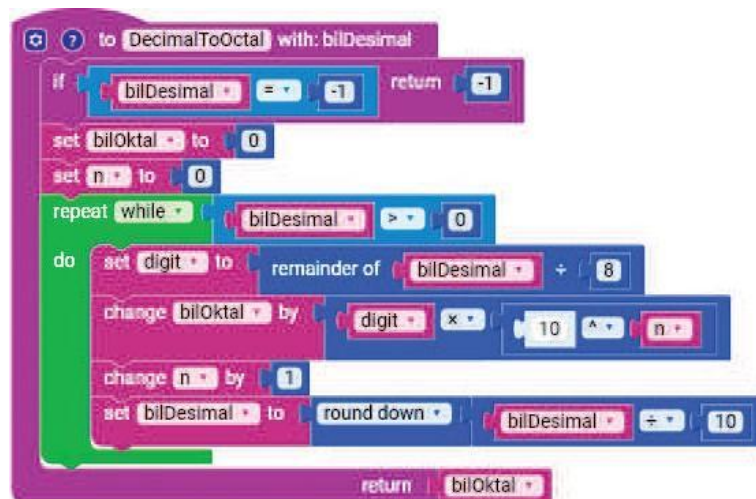
Awal	Tujuan	Urutan Fungsi
Biner	Oktal	BinerEvenToDecimal – DecimalToOctal
Biner	Desimal	BinerEvenToDecimal
Oktal	Biner	OctalToDecimal – DecimalToBinerEven
Oktal	Desimal	OctalToDecimal
Desimal	Biner	DecimalToBinerEven
Desimal	Oktal	DecimalToOctal

Isi dari setiap *function* tidak berubah kecuali *function* DecimalToOctal. Ada kemungkinan parameter untuk *function* tersebut ialah -1. Hal ini dapat terjadi saat pengguna melakukan konversi dari biner ke oktal. Untuk melakukan konversi biner ke oktal, urutan *function*:

BinerEvenToDecimal kemudian DecimalToOctal

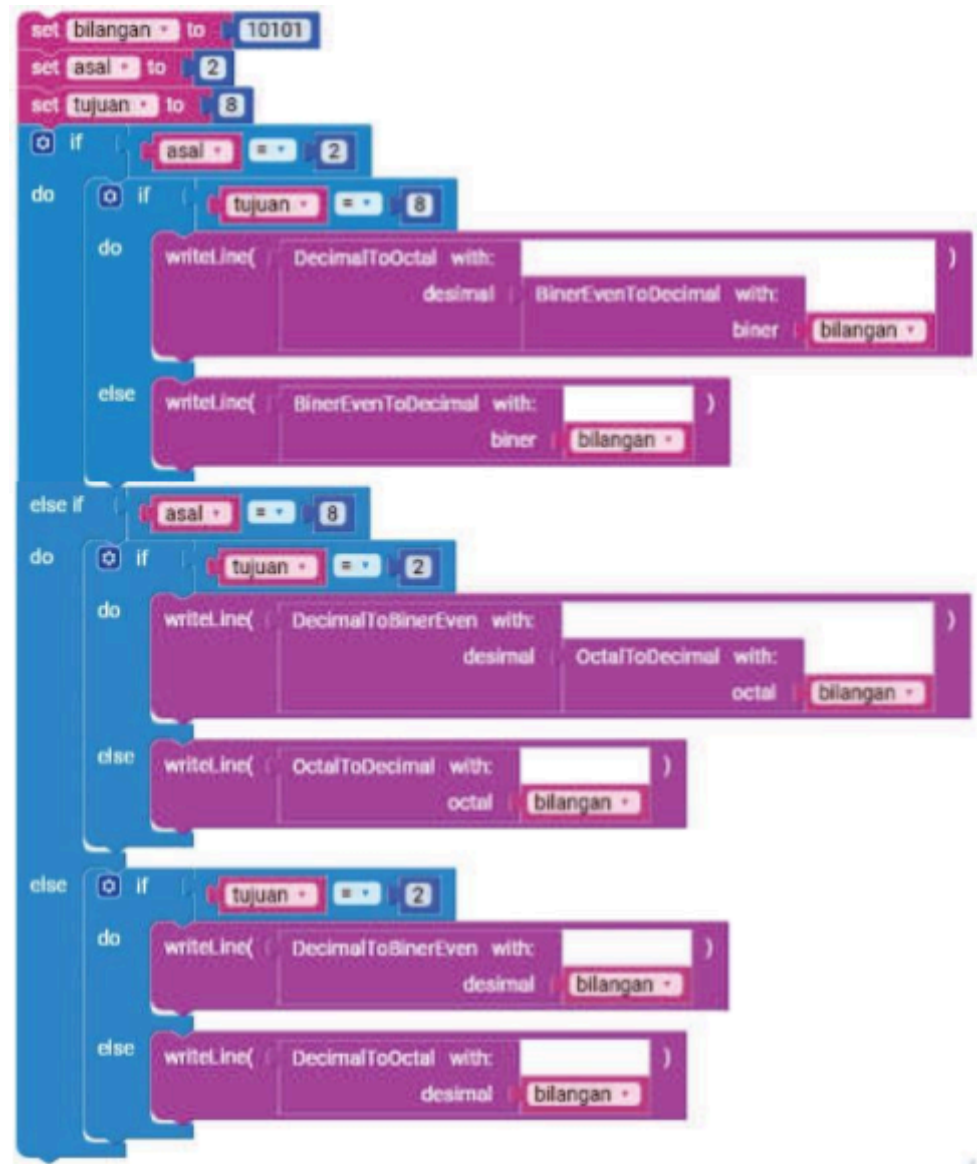
Jika banyaknya bit biner yang dimasukkan adalah 1 (ganjil), *function* BinerEvenToDecimal akan mengembalikan -1. Nilai -1 ini akan menjadi parameter untuk *function* DecimalToOctal. Jika *function* DecimalToOctal tidak menangani kasus di saat masukannya adalah -1, kembalian dari *function* tersebut akan menjadi 0.

Nah, untuk mengatasi masukan -1, fungsi DecimalToOctal perlu ditambahkan perintah mengembalikan -1 jika masukannya adalah -1 seperti terlihat pada Gambar 7.26.



Gambar 7.26 Contoh modifikasi fungsi DecimalToOctal

Blok program pada bagian program utama yang memanggil fungsi- fungsi sesuai dengan permintaan pengguna ditunjukkan dalam Gambar 7.27.



Gambar 7.27 Contoh pemanggilan program utama

Setelah membuat program, ingatkan peserta didik untuk menguji program yang sudah mereka buat. Minta mereka menuliskan hasil pengujian mereka di lembar kerja dalam sebuah tabel dengan format yang diberikan di Buku Siswa. Guru juga bisa membantu memberikan kasus- kasus uji agar peserta didik bisa mendeteksi apabila program yang mereka buat masih memiliki kesalahan.

Aktivitas AP-K9-20

Dalam pembuatan poster, guru membimbing peserta didik untuk menyajikan penjelasan program konverter bilangan yang sudah mereka buat. Pastikan isi dari poster jelas dan lengkap sesuai dengan petunjuk di Buku Siswa.

Jika ada peserta didik yang tidak tahu bagaimana membuat poster digital, guru

dapat mengajarkan *tools* yang sederhana untuk membuat poster. *Tools* paling sederhana adalah menggunakan Microsoft Word. Guru bisa mencontohkan membuat poster dengan *shape*, *text box*, dan gambar di Microsoft Word.

Aktivitas ini juga dapat dibuat menjadi aktivitas *unplugged*. Contohnya, poster dapat dibuat di kertas manila/kertas gambar berukuran A2. Peserta didik dapat berkreasi menggunakan spidol, pensil warna, atau krayon atau kertas lipat.

Untuk penilaian poster, guru dapat menilai dari segi berikut.

1. Kelengkapan dan kejelasan isi poster: apakah peserta didik menyajikan isi dengan lengkap dan jelas. Agar poster enak dibaca, poster tidak boleh memiliki banyak isi. Dari sini, peserta didik belajar untuk merangkum materi yang penting dan menyajikan dengan ringkas, tetapi lengkap.
2. Struktur poster: apakah alur poster enak untuk dibaca.
3. Desain poster: apakah peserta didik membuat desain yang enak dilihat. Desain dinilai bukan dari seberapa rumit pembuatannya, tetapi dari pemilihan posisi, warna, dan font. Dari sini, peserta didik belajar untuk memilih perpaduan warna dan *font* yang membuat pembaca dapat melihat dengan nyaman.

Aktivitas AP-K9-21

Setelah belajar bahasa Scratch dan Blockly, peserta didik diajak untuk merefleksikan pengalaman belajar mereka dan menyajikannya dalam sebuah poster. Bimbing peserta didik untuk menyajikan refleksi seperti petunjuk pada Buku Siswa melalui poster mereka.

Proses selanjutnya sama dengan Aktivitas AP-K9-20 di atas.

Hasil dari poster dapat disajikan secara fisik apabila kondisinya memungkinkan. Jika tidak memungkinkan, guru bisa memasang poster hasil karya peserta didik secara digital dengan menyimpannya di *drive*, *jamboard*, atau aplikasi lain yang bisa diakses oleh semua peserta didik. Dari kegiatan ini, diharapkan peserta didik dapat menikmati proses belajar pemrograman visual dan memiliki dokumentasi karya mereka (program konversi bilangan).

Pertemuan 9: Tambahan - Pemrograman dengan Ozobot (2 jp)

I. Tujuan Pembelajaran:

Memahami ozobot atau robot lainnya, apabila robot tersedia

II. Apersepsi

Sama seperti di buku kelas VIII, modul ini ialah modul tambahan apabila sekolah dan peserta didik memang memiliki robot Ozobot atau robot *line follower* lainnya. Jika sekolah memiliki robot Ozobot ini, modul ini siap pakai untuk dijadikan panduan dalam eksplorasi. Peserta

didik juga dapat diajak berperan sebagai robot Ozobot sehingga menjadi aktivitas *unplugged*. Robot Ozobot ini tidak hanya dapat bergerak mengikuti garis begitu saja. Sama seperti robot lainnya, robot Ozobot ini dapat dikendalikan dengan program yang dibuat oleh manusia. Guru dapat memberi contoh ilmu yang terdapat robot di dalamnya.

III. Pemantik

Diberikan video simulasi Robot Ozobot.

- <https://ozobot.com/educate>
- <https://ozobot.com/create/challenges>
- <https://ozobot.com/educate/lessons>

IV. Kebutuhan Sarana dan Prasarana

- a. Komputer/ponsel pintar yang ter-*install* sistem operasi dan *browser*
- b. Koneksi internet
- c. Robot Ozobot

V. Kegiatan Inti

Guru mengingatkan kembali bahwa di kelas VIII, peserta didik telah mempelajari bagaimana cara kerja Ozobot secara manual dan bagaimana merawat dan melakukan kalibrasi robot Ozobot. Di materi kelas IX ini, peserta didik akan mempelajari bagaimana mengatur Ozobot melalui program. Guru menjelaskan bahwa gerakan robot Ozobot dapat dikendalikan melalui sebuah program yang dibuat sendiri. Untuk itu, diperlukan seperangkat komputer atau laptop yang terhubung dengan internet.

Apabila sekolah tidak dapat menyediakan robot Ozobot, guru dapat mencoba implementasi secara *unplugged*. Peserta didik dibagi menjadi 3 orang per kelompok dan masing-masing akan memiliki peran yang berbeda, yaitu:

- (1) menjadi program yang akan memberi perintah pada robot
- (2) menjadi robot yang akan menerima perintah dari program
- (3) menjadi tester yang akan memastikan apakah robot bergerak sesuai perintah

Kelompok peserta didik ini dapat saling bekerja sama untuk menjalankan aktivitas yang ada dan apabila memungkinkan, lembar kerja yang tadinya berupa garis di kertas dapat dilakukan secara nyata dengan membuat pola sejenis di ruang kelas.

Apabila sekolah dapat menyediakan robot Ozobot, hal ini dapat dilakukan langsung menggunakan robotnya.

Pembuatan program untuk kendali robot Ozobot menggunakan OzoBlockly yang dapat diakses melalui link: <https://OzoBlockly.com/editor?lang=en&robot=bit&mode=2>. Guru akan menjelaskan langkah per langkah dan peserta didik langsung mencobanya pada robot

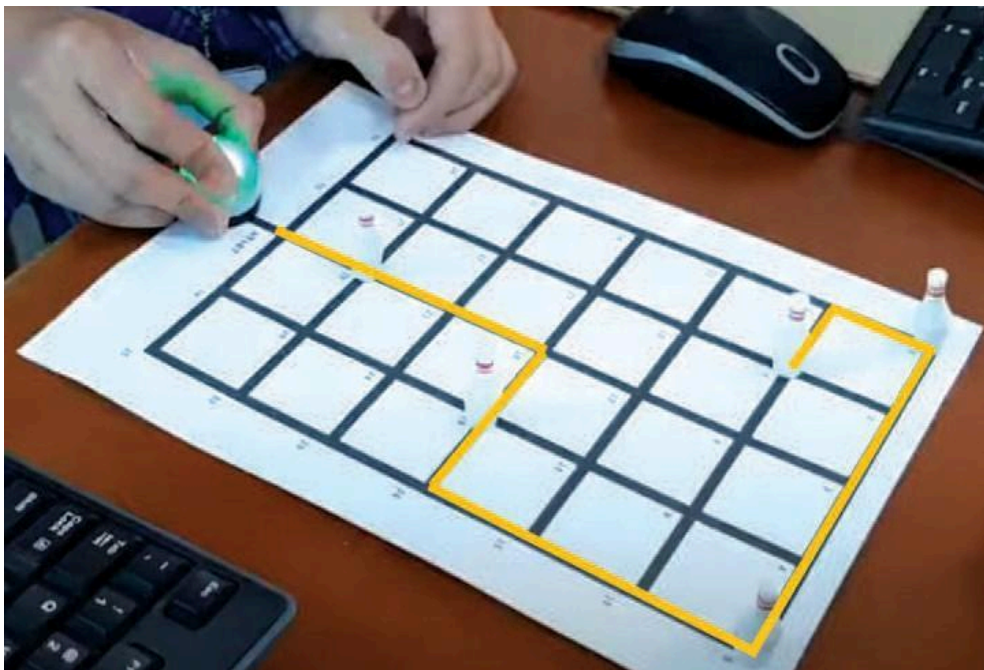
Ozobot. Yang akan dipelajari pada materi ini adalah:

1. Navigasi tanpa Garis
2. Permainan Lampu pada Ozobot
3. Pengaturan Waktu pada Ozobot
4. Pengulangan Blok
5. Navigasi dengan Garis
6. Blok Percabangan dengan Garis

Setiap materi ada pada Aktivitas AP-K9-23: Memprogram Ozobot. Guru mengajak peserta didik untuk mencoba sesuai panduan pada aktivitas.

Ada tantangan yang diberikan pada peserta didik untuk menjatuhkan semua benda kecil yang ada di titik-titik persimpangan secepat mungkin (berarti dengan langkah sesedikit mungkin) dengan menggunakan robot Ozobot. Ada beberapa aturan yang ditetapkan, yaitu bahwa semua garis hanya dapat dilewati satu kali saja.

Salah satu contoh jalur yang dapat diikuti dapat dilihat pada Gambar 7.28.



Gambar 7.28 Contoh kunci jawaban jalur robot Ozobot

Metode Pembelajaran Alternatif

Pembelajaran pada bab ini hampir semuanya membutuhkan sarana dan prasarana komputer. Akan tetapi, guru dapat berkreasi dengan memanfaatkan materi *unplugged* pada bahan pengayaan. Atau, guru juga dapat membuat blok-blok kode menjadi kartu-kartu yang dapat dimainkan oleh peserta didik.

Aktivitas pembuatan poster juga dapat dibuat sebagai aktivitas *unplugged* sesuai penjelasan pada panduan pembelajarannya. Materi tambahan terkait robot Ozobot dapat menggunakan robot *line follower* lainnya atau melakukannya secara *unplugged* seperti yang telah dijelaskan pada panduan pembelajarannya.

Pengayaan dan Remedial

Pengayaan

Aktivitas pembelajaran bisa dikembangkan dengan mempelajari materi dari situs-situs yang memiliki reputasi bagus, seperti:

1. Kegiatan unplugged, <https://csunplugged.org/en/> atau <https://code.org/curriculum/unplugged>
2. Kurikulum plugged Scratch, <https://csirst.withgoogle.com/s/en/home>
3. Materi Blockly, <https://developers.google.com/blockly/guides/overview>
4. <https://Ozobot.com/create/challenges>
5. <https://Ozobot.com/educate/lessons>
6. <https://Ozobot.com/create/Ozoblockly>

Remedial

Aktivitas pembelajaran pada kelompok yang membutuhkan remedial dapat dikembangkan dengan melakukan pendampingan kepada peserta didik untuk setiap aktivitas yang ada. Guru dapat juga memberikan tips- tips khusus untuk memudahkan pemahaman materi. Tutorial sebaya juga dapat dilakukan dengan mengajak berdiskusi peserta didik yang telah memahami materi. Penjelasan dalam bentuk video tutorial yang dapat diakses oleh peserta didik juga akan sangat membantu, sehingga peserta didik dapat mempelajari materi pembelajaran berulang-ulang secara mandiri hingga memahaminya.

Asesmen dan Rubrik Penilaian

Formatif:

Penilaian formatif dilakukan tiap minggu dari aktivitas yang ada, seperti Aktivitas AP-K9-01 sampai Aktivitas AP-K9-23.

Tabel 7.17 Kriteria Penilaian Asesmen Bab Algoritma dan Pemrograman

Komponen Penilaian	A = Baik Sekali	B = Baik	C = Cukup	D = Kurang
Kemampuan menyelesaikan kasus menggunakan	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil

kan Scratch dan Blockly				diterapkan
Kemampuan pemahaman modularisasi program	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil diterapkan
Kemampuan implementasi <i>function</i> dan <i>procedure</i> pada Blockly	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil diterapkan
Kemampuan pemahaman sistem bilangan	≥ 80% soal yang diminta berhasil diselesaikan	60%-79% soal yang diminta berhasil diselesaikan	40%-59% soal yang diminta berhasil diselesaikan	Kurang dari 40% soal yang diminta berhasil diselesaikan
Kemampuan implementasi konversi sistem bilangan pada Blockly	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil diterapkan
Kemampuan pemahaman <i>parity bit</i> pada sistem bilangan biner	≥ 80% soal yang diminta berhasil diselesaikan	60%-79% soal yang diminta berhasil diselesaikan	40%-59% soal yang diminta berhasil diselesaikan	Kurang dari 40% soal yang diminta berhasil diselesaikan
Kemampuan implemen-tasi <i>parity bit</i> pada sistem bilangan biner mengguna-kan Blockly	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil diterapkan

Komponen Penilaian	A = Baik Sekali	B = Baik	C = Cukup	D = Kurang
Kemampuan implementasi konversi sistem bilangan	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil diterapkan

menjadi satu program utuh				
Kemampuan membuat poster presentasi dan pengalaman	≥ 80% itur yang diminta berhasil diterapkan	60%-79% itur yang diminta berhasil diterapkan	40%-59% itur yang diminta berhasil diterapkan	Kurang dari 40% itur yang diminta berhasil diterapkan
Kemampuan melakukan pengamatan saat mencoba Ozobot	Semua latihan dilakukan. Dapat memperhatikan cara pemakaian robot dengan luar biasa, di luar ekspektasi	Semua latihan dikerjakan. Tapi tidak mencoba membuat sesuatu yang baru untuk dicobakan pada robot	Tidak semua latihan selesai. Berusaha menyelesaikan latihan sampai waktu habis	Hanya mencoba satu latihan saja atau tidak sama sekali. Tidak mengikuti petunjuk
Kemampuan navigasi Ozobot tanpa Garis	Dapat memberikan irama kepada robot. Robot mampu mencapai tujuan dengan efisien	Seluruh gerakan dasar robot dapat diprogram ke dalam robot. Robot mampu mencapai tujuan	Robot hanya melakukan 2 atau 3 gerakan dasar saja. Robot tidak mencapai tujuan karena terhalang objek	Tidak mencoba. Sengaja membuat robot menabrak objek yang berpotensi merusak robot
Kemampuan penerapan Fungsi Pengulangan	Robot mampu menyelesaikan semua soal pengulangan dengan sangat efisien tanpa melakukan kesalahan berarti	Robot mampu menyelesaikan semua soal pengulangan yang diberikan	Robot hanya berhasil melakukan 1 fungsi pengulangan sesuai contoh. Tidak mencoba menyelesaikan soal lain	Tidak mencoba. Sengaja membuat robot menabrak objek yang berpotensi merusak robot

Komponen Penilaian	A = Baik Sekali	B = Baik	C = Cukup	D = Kurang
Kemampuan penerapan Fungsi	Robot mampu menyelesaikan semua	Robot mampu menyelesaikan semua	Robot hanya berhasil melakukan 1 fungsi	Tidak mencoba. Sengaja

Percabangan	rintangan cabang dengan sangat eisien tanpa melakukan kesalahan berarti	rintangan percabangan yang diberikan	percabangan sesuai contoh. Tidak mencoba menyelesaikan soal lain	membuat robot menabrak objek yang berpotensi merusak robot
-------------	---	--------------------------------------	--	--

Sumatif:

Sumatif dilakukan dengan asesmen melalui soal, seperti contoh pada uji kompetensi.

Jawaban Uji Kompetensi

>>>> Soal Esai <<<<

1. Uji Kompetensi 1 – Soal 1

```

to kabisat with: tahun
  if remainder of (tahun + 400) = 0
  do set jenis to "kabisat"
  else if remainder of (tahun + 100) = 0
  do set jenis to "bukan kabisat"
  else if remainder of (tahun + 4) = 0
  do set jenis to "kabisat"
  else set jenis to "bukan kabisat"
  return jenis

```

Interaksi Guru dan Orang Tua/Wali

Orang tua/wali hendaknya selalu aktif dalam mengawasi anaknya ketika melakukan aktivitas *online*. Guru dapat berinteraksi dengan memberikan informasi dan tips bagi orang tua dalam penggunaan *tools* pemrograman visual yang ada. Sehingga orang tua juga dapat membantu anaknya ketika mengalami kesulitan.

Refleksi Guru

Tabel 7.17 Refleksi Guru Bab Algoritma dan Pemrograman

No	Aspek	Pertanyaan
1	Tujuan pembelajaran	Apakah tujuan pembelajaran sudah tercapai?
2	Proses kegiatan belajar mengajar	Apakah kegiatan belajar mengajar yang dilakukan sudah berhasil dengan baik?
3	Materi/konten pelajaran	Apakah ketepatan, kedalaman dan keluasan materi yang saya sampaikan sudah cukup untuk mencapai Tujuan Pembelajaran?
4	Kondisi peserta didik	Apakah semua peserta didik dapat mengikuti aktivitas pembelajaran dengan baik?
5	Kesulitan belajar peserta didik	Apakah ada peserta didik yang mengalami kesulitan dalam belajar materi ini?
6	Minat belajar	Apakah ada peserta didik yang memiliki minat belajar lebih dan berkeinginan mengeksplorasi lebih dari materi ini?
7	Efektivitas metode pembelajaran	Seberapa efektifkah metode pembelajaran yang sudah saya gunakan?
8	Variasi pembelajaran	Apakah saya memiliki ide lain untuk mengembangkan materi ini ?

Mengetahui,
Kepala madrasah,

Nur Khamid, S.Pd.
NIP. –

Sukaraja, 2 Januari 2025
Guru mata pelajaran,

Mukhamad Fathoni, M.Pd.I.
NIP. 198002162005011003