

## Topik CT dan Projek Kreatif: Projek *Programming* dengan Scratch

### Materi Pengayaan

Durasi : 2 Pertemuan (lihat petunjuk penggunaan modul)

Alokasi waktu yang diberikan pada tabel ini adalah alokasi waktu pada keadaan ideal.

Dosen dapat menyesuaikan alokasi waktu dengan situasi kelas.

Capaian Pembelajaran	Alokasi Waktu
CPMK-04 Pembelajaran berbasis Studi Kasus Tematik untuk penerapan CT dengan menjalankan siklus lengkap <i>problem solving</i> dengan 4 fondasi CT mulai memahami permasalahan, menganalisis, menemukan akar persoalan, mengusulkan alternatif solusi, memilih solusi “terbaik” (efektif, efisien, optimal). <b>Sub-CPMK</b> CPMK-04.05 Menghasilkan karya kreatif sebagai ekspresi solusi berbasis CT dari sebuah problem (P1, KU1, KK1, KK3).	[TM: 2x(2x50”)] [BT: 2x(2x60”); BM: 2x(2x60”)]

Anda telah mempelajari bahwa CT adalah suatu cara berpikir yang dapat membantu menyelesaikan suatu permasalahan. Solusi yang disusun untuk menyelesaikan permasalahan ini dapat diwujudkan dalam berbagai bentuk. Salah satunya adalah dalam bentuk program komputer. Solusi persoalan yang diwujudkan dalam bentuk program komputer dapat digunakan secara berulang-ulang, oleh banyak orang dan proses komputasi yang dilakukan oleh komputer cenderung lebih cepat jika dibandingkan dengan dilakukan oleh manusia dan jika data banyak tidak menimbulkan kesalahan karena lelah atau jenuh mengulang hal yang sama. Kelebihan-kelebihan tersebut yang menyebabkan beberapa negara melatih peserta didiknya untuk membuat program sederhana sejak usia SD.

Pada Petunjuk Penggunaan Modul (Gambar 1) disebutkan bahwa terdapat dua cara untuk menuliskan program, yaitu dengan pemrograman visual dan pemrograman

tekstual. Pemrograman visual pada umumnya digunakan oleh peserta didik usia SD atau SMP ketika memulai belajar membuat program karena pemrograman visual cenderung mudah dipahami. Dengan alasan tersebut dan dengan pertimbangan bahwa mata kuliah ini dapat diambil oleh guru non-informatika, maka metode pemrograman yang digunakan pada topik ini adalah metode pemrograman visual.

Untuk memperkaya pengetahuan dan keterampilan mahasiswa, pada bagian lampiran ini, mahasiswa akan mempelajari bagaimana CT dapat membantu dalam membuat suatu solusi permasalahan dalam bentuk program komputer. Meskipun diletakkan pada lampiran dan dijadikan sebagai pengayaan, topik ini adalah topik yang penting. Bagi guru dengan latar belakang pendidikan di bidang Informatika, topik ini dapat digunakan sebagai penyegaran kembali materi pemrograman yang dahulu pernah dipelajari. Dengan demikian diharapkan agar para guru berlatar belakang pendidikan di bidang Informatika dapat melihat hubungan antara CT dengan pemrograman dengan lebih jelas. Bagi guru dengan latar belakang pendidikan selain Informatika, topik ini dapat digunakan sebagai materi untuk pengenalan konsep-konsep dasar pemrograman. Setelahnya, para guru diharapkan dapat memahami bahwa program komputer yang baik adalah cerminan dari keterampilan CT yang baik pula.

Untuk mendukung tujuan yang disebutkan di atas, di akhir topik ini mahasiswa akan diminta untuk membuat suatu karya kreatif dalam bentuk pemrograman Scratch. Ada beberapa proyek yang perlu dikerjakan.

- Pembuatan proyek permainan tebak angka sederhana yang disajikan pada lembar kerja bagian 01.04.
- Pembuatan karya kreatif yang berkaitan dengan bidang mata pelajaran yang akan diajar oleh para mahasiswa yang merupakan para calon guru. Alasan perlu dibuatnya karya kreatif yang berkaitan dengan bidang mata pelajaran ini adalah karena nantinya solusi dari suatu persoalan di bidang mata pelajaran, tidak lagi hanya berupa jawaban tertulis, namun bisa saja dalam bentuk program komputer.

**Catatan: Sangat penting untuk menghargai kekayaan intelektual. Karena itu karya kreatif yang dibangun harus merupakan karya kreatif yang tidak melanggar HAKI (misalnya tidak menyalin karya orang lain dan diakui sebagai karya yang dibuat sendiri) karena saat ini ada banyak sekali contoh program scratch yang dengan mudah dapat di-copy/paste.**

# 01. Eksplorasi Konsep

## Jenis Aktivitas: Aktivitas Individual

### 01.01 Pendekatan Pembelajaran

Pada modul ini, Anda secara individual akan mempelajari bagaimana membuat solusi dari suatu permasalahan dalam bentuk program komputer. Pendekatan pembelajaran yang akan digunakan adalah pendekatan *flipped learning*. Karena beragamnya latar belakang mahasiswa dan dosen, maka pada modul ini, mahasiswa dan dosen dapat memilih aktivitas yang berbeda, yang secara umum adalah sebagai berikut:

1. Jika mahasiswa sudah terbiasa memprogram Scratch, maka mahasiswa dapat langsung mengerjakan proyek, kemudian menggali CT di dalam kegiatan memprogram.
2. Jika mahasiswa belum pernah atau tidak/belum terbiasa memprogram, maka mahasiswa pertama-tama perlu mempelajari secara lebih detail tentang kegiatan memprogram dan mempraktekkan kegiatan memprogram. Tujuannya adalah agar mahasiswa dapat merasakan dan mempunyai pengalaman memprogram. Sebagai permulaan, Anda dapat membaca buku Informatika Kelas VII, pada Bab Algoritma dan Pemrograman (Wijanto et al., 2021) dan tutorial penggunaan Scratch (*Get Started in Scratch*).

Bagian 01 dan 02 dari modul ini dipersiapkan bagi calon guru yang belum biasa memprogram dengan Scratch. Modul 03, 04, 05, 06, dan 07 dipersiapkan bagi seluruh calon guru.

### 01.02 Paparan Konsep

#### 01.02. 01. Miskonsepsi: CT adalah Koding (Liem & Natali, 2021)

Pemrograman dan CT adalah dua elemen pada informatika yang komplementer dan tidak terpisahkan satu sama lainnya. Pemrograman adalah suatu kegiatan untuk menghasilkan suatu solusi dari sebuah persoalan atau menghasilkan suatu karya digital. Baris-baris kode yang dituliskan pada kegiatan ini merupakan

cerminan dari pola pikir dari orang yang menulis kode. Karena itulah sebuah program yang baik merupakan cerminan dari keterampilan CT yang baik pula. Program yang baik dapat dihasilkan dari: dekomposisi yang baik, abstraksi yang sesuai, menggunakan pola program yang teruji, serta menggunakan strategi algoritmik yang tepat. Dengan demikian, kita dapat menyimpulkan bahwa CT dan pemrograman memiliki kaitan yang erat. Perbedaannya adalah bahwa CT berkaitan dengan konsep berpikir sementara pemrograman adalah suatu keterampilan untuk menulis program dalam suatu bahasa yang dapat dipahami baik oleh komputer maupun oleh manusia yang mengaksesnya.

Terdapat dua buah paradigma dalam pemrograman, yaitu pemrograman visual dan pemrograman prosedural. Paradigma pemrograman berbeda dengan bahasa pemrograman. Paradigma pemrograman adalah sudut pandang pemrogram dalam menghasilkan program komputer. Bahasa pemrograman adalah alat untuk mengekspresikan program. Pada modul ini, kita akan menggunakan paradigma pemrograman visual dengan menggunakan bahasa pemrograman Scratch.

Selain paradigma dalam pemrograman, perlu juga diperhatikan bahwa pembelajaran pemrograman dapat dilakukan secara *unplugged* dan secara *plugged*. *Unplugged* artinya tidak menggunakan perangkat komputer, sementara *plugged* artinya menggunakan perangkat komputer. Keduanya akan dipelajari pada modul ini.

Bagi mahasiswa yang sudah biasa memprogram, Anda dapat mulai membuat sebuah karya inovatif berupa proyek yang ada pada Ruang Kolaborasi. Sementara itu, bagi mahasiswa yang belum pernah memprogram, Anda perlu merasakan dan mengalami terlebih dahulu aktivitas memprogram.

#### 01.02.01. Mempelajari Scratch dengan Mempraktikkan Membuat Program Sederhana (Bagi Mahasiswa yang Belum Terbiasa Memprogram dengan Scratch).

Sebelum mulai melakukan praktek membuat program sederhana, Anda wajib mempelajari dan mencoba program Scratch. Salah satunya adalah melalui Buku Informatika kelas VII (Wijanto et al., 2021), pada Bab Algoritma dan Pemrograman. Dengan demikian, Anda akan dapat memahami istilah-istilah pada Scratch dan dapat mengerjakan bagian ini dengan lebih mudah.

Bahasa Scratch dipilih karena merupakan salah satu bahasa pemrograman yang cocok digunakan bagi orang-orang yang masih pemula dalam memprogram,

khususnya bagi anak-anak. Itulah mengapa Scratch memiliki antarmuka visual yang sangat sederhana namun menarik. Selain itu, Scratch dipilih karena mendukung pengenalan keterampilan CT dan *problem solving* melalui berbagai fitur yang ia miliki. Kita akan membuat program yang sangat sederhana yang akan membuat *sprite* bergerak mengelilingi suatu bidang berukuran 100 x 100 satuan langkah. Kita akan menggunakan CT untuk membuat program ini. Pada contoh ini, program dimodifikasi berkali-kali dengan tujuan agar penggunaan fondasi-fondasi CT pada program dapat terlihat. Karena CT adalah pola pikir, seharusnya CT diterapkan dahulu sebelum membangun sebuah program.

### Dekomposisi

Untuk dapat bergerak/berkeliling, kita dapat mendekomposisinya menjadi tiga bagian, yaitu:

1. Gerakkan *sprite* ke depan
2. Belok kanan atau berputar 90 derajat
3. Diam selama beberapa waktu. Langkah ke-3 ini ditambahkan agar pergerakan *sprite* pada Scratch tidak terlalu cepat dan bisa dilihat peralihan gerakannya.

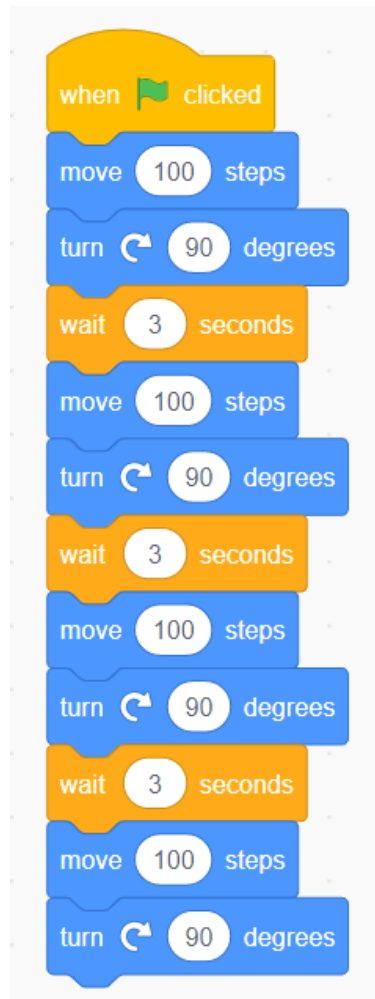
### Algoritma

Berdasarkan dekomposisi tersebut, kita dapat menyusun langkah penyelesaiannya.

Berikut ini adalah **salah satu langkah penyelesaiannya**:

1. Gerakkan *sprite* 100 langkah ke depan.
2. Belok kanan sebesar 90 derajat.
3. Diam selama 3 detik.
4. Gerakkan *sprite* 100 langkah ke depan.
5. Belok kanan sebesar 90 derajat.
6. Diam selama 3 detik.
7. Gerakkan *sprite* 100 langkah ke depan.
8. Belok kanan sebesar 90 derajat.
9. Diam selama 3 detik.
10. Gerakkan *sprite* 100 langkah ke depan.
11. Belok kanan sebesar 90 derajat.
12. Diam selama 3 detik.

Apabila langkah-langkah di atas kita terjemahkan menjadi program Scratch, maka program kita akan menjadi seperti Gambar B.1.



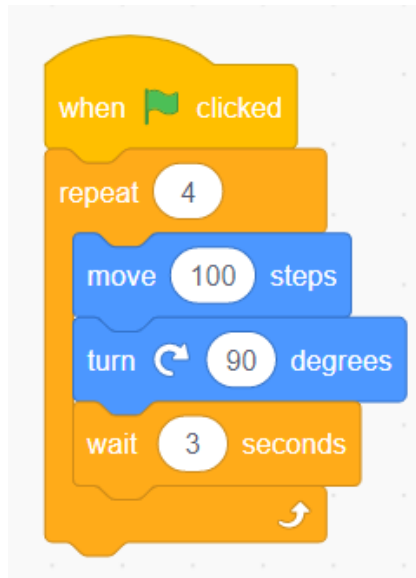
Gambar B.1: Contoh Program Scratch untuk Membuat Sprite Bergerak Berkeliling

#### Pengenalan Pola

Pada program di atas, kita dapat melihat ada pola yang dilakukan sebanyak empat kali, yang masing-masing adalah:

- a. Gerakkan *sprite* 100 langkah ke depan.
- b. Belok kanan sebesar 90 derajat.
- c. Diam selama 3 detik.

Berdasarkan hal tersebut, kita dapat memodifikasi program kita sehingga menjadi seperti Gambar B.2.



Gambar B.2: Contoh Penggunaan Pengenalan Pola Pada Pemrograman Scratch untuk Kasus Membuat *sprite* Bergerak Berkeliling

### Abstraksi

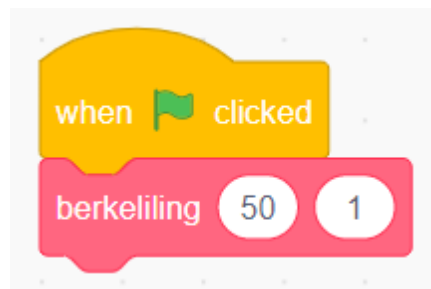
Bagaimana program yang dibuat dapat bersifat lebih general? Misalnya, kita ingin agar gerakan yang dilakukan bisa berapapun, tidak hanya 100 langkah. Begitu juga dengan waktu diamnya, ingin kita percepat atau perlambat, bukan sebesar 3 detik saja. Di sinilah fondasi abstraksi dapat berperan.

Ingat kembali bahwa abstraksi adalah eliminasi detail-detail atau bagian-bagian yang tidak relevan pada suatu persoalan, dan membangun suatu solusi general untuk persoalan sejenis. Pada pemrograman, hal ini dapat dilakukan salah satunya dengan membuat sebuah fungsi. Pada contoh ini, fungsi tersebut diberi nama `berkeliling`. Karena banyak langkah dan waktu diamnya dapat berubah-ubah, maka fungsi yang kita buat akan memiliki dua buah parameter. Pada contoh implementasi di Gambar B.3, banyak langkah diimplementasikan sebagai variabel `banyakLangkah` sedangkan waktu diam sebagai variabel `waktuDiam`.



Gambar B.3: Contoh Penggunaan Abstraksi Pada Pemrograman Scratch untuk Kasus Membuat Sprite Bergerak Berkeliling

Dengan adanya fungsi `berkeliling`, kita dapat membuat sprite melangkah sebanyak apapun. Gambar B.4 adalah contoh implementasi jika kita ingin agar sprite bergerak 50 langkah dengan waktu diam sebanyak 1 detik.



Gambar B.4: Contoh Penggunaan Fungsi Pada Pemrograman Scratch

#### 01.02.02. Belajar Membuat Program Secara *Unplugged*

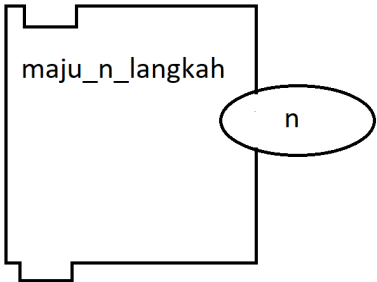

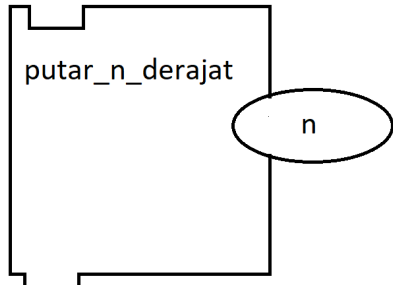
Seperti yang pernah dibahas sebelumnya, pemrograman dapat dipelajari secara *unplugged*. Caranya adalah dengan membuat blok-blok dalam bentuk fisik, misalnya berupa kartu. Untuk contoh pada kasus yang sebelumnya, yaitu program untuk bergerak berkeliling, kita dapat membuat blok menggunakan kartu, seperti pada Tabel B.1.

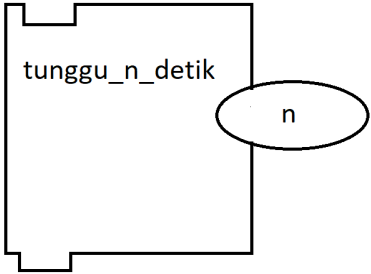
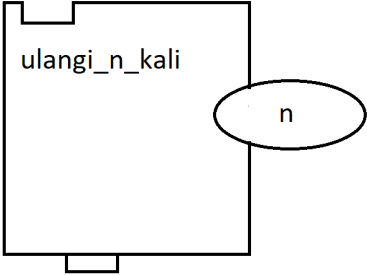
Setiap jenis blok pada Tabel B.1 dapat dicetak dalam beberapa salinan sesuai kebutuhan. Anda juga dapat membuat kartu-kartu lainnya untuk jenis blok lainnya

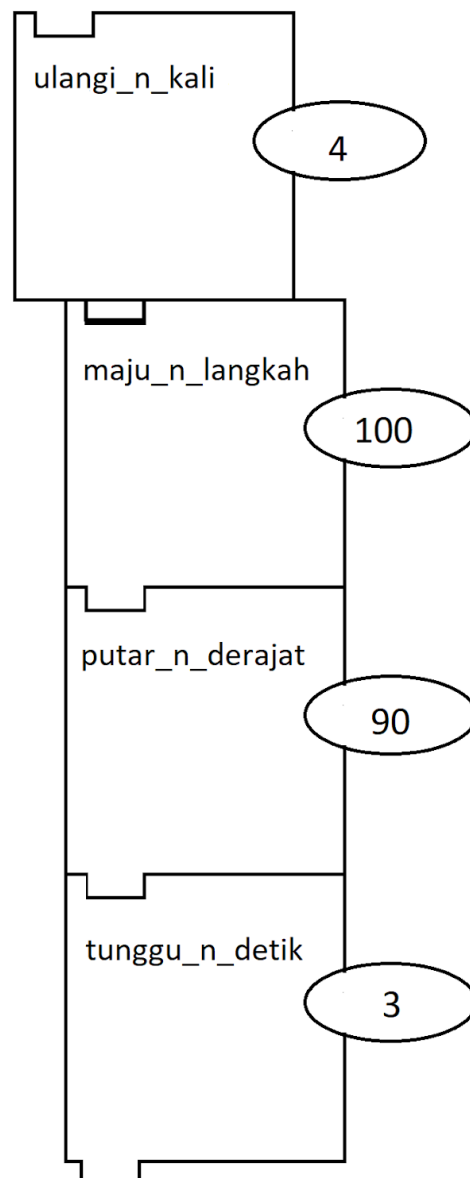


yang ada di Scratch. Gambar B.5 adalah contoh penggunaan kartu-kartu di atas untuk *problem* gerak berkeliling pada contoh di bagian 01.02.01.

*Tabel B.1: Contoh Pembuatan Blok Untuk Mempelajari Pemrograman Secara Unplugged*

No	Nama Blok	Notasi	Keterangan
1	maju_n_langkah	 <p>The diagram shows a Scratch 'move' block with the text 'maju_n_langkah' inside and an oval input field containing the variable 'n'.</p>	<p>Blok ini digunakan sebagai instruksi untuk maju sebanyak <math>n</math> langkah. Nilai <math>n</math> dapat diberikan dengan menggunakan kartu <code>bilangan_bulat_n</code>.</p>
2	bilangan_bulat_n	 <p>The diagram shows a Scratch 'say' block, which is an oval shape.</p>	<p>Blok ini digunakan sebagai instruksi untuk menyimpan sebuah data yang bertipe numerik. Nilai <math>n</math> yang akan digunakan dapat dituliskan dengan spidol atau pena.</p>
3	putar_n_derajat	 <p>The diagram shows a Scratch 'turn' block with the text 'putar_n_derajat' inside and an oval input field containing the variable 'n'.</p>	<p>Blok ini digunakan sebagai instruksi untuk berputar sebanyak <math>n</math> derajat. Nilai <math>n</math> dapat diberikan dengan menggunakan kartu</p>

No	Nama Blok	Notasi	Keterangan
			bilangan_bulat_n .
4	tunggu_n_detik		Blok ini digunakan sebagai instruksi untuk diam atau menunggu selama n detik. Nilai n dapat diberikan dengan menggunakan kartu bilangan_bulat_n.
5	ulangi_n_kali		Blok ini digunakan untuk mengulangi satu atau beberapa instruksi sebanyak n kali. Nilai n dapat diberikan dengan menggunakan kartu bilangan_bulat_n.



Gambar B.5: Contoh Blok Unplugged untuk Membuat Objek Bergerak Berkeliling

Sebagai pengganti komputer yang mengeksekusi rangkaian instruksi di atas, Anda kelak dapat memilih satu orang peserta didik Anda untuk mengeksekusi tiap instruksi yang diberikan.

#### 01.02.03. Asesmen Keterampilan CT melalui Pemrograman Scratch

Pada makalah “*Computational thinking in programming with Scratch in primary schools: A systematic review*” (Fagerlund et al., 2021) dijelaskan berbagai studi yang telah dilakukan oleh berbagai peneliti yang telah membangun berbagai *framework* untuk memberi asesmen pada keterampilan CT melalui pemrograman. (Moreno-León

et al., 2015) menyusun rubrik penilaian seperti yang ditunjukkan pada Tabel B.2 (Moreno-León et al., 2015).

*Tabel B.2: Indikator Level Kompetensi CT Melalui Pemrograman Scratch*

Konsep CT	Indikator Level Kompetensi		
	Dasar	Sedang	Tinggi
Abstraksi dan Dekomposisi <i>Problem</i>	Penggunaan lebih dari satu <i>script</i> dan satu sprite.	Pembuatan blok/fungsi.	Cloning (pembuatan salinan atau duplikat dari suatu sprites. Duplikat ini akan mewarisi sifat dari induknya)
Paralelisme	Penggunaan blok “when green flag clicked” untuk memulai dua buah <i>script</i> .	Penggunaan blok “when key pressed”, atau “when this sprite clicked” untuk memulai dua buah <i>script</i> .	Penggunaan blok “when I receive message”, “create clone” atau “when backdrop change to” untuk memulai dua buah script.
<i>Logical thinking</i>	Penggunaan blok “if”.	Penggunaan blok “if-else”.	Penggunaan operator logika
Sinkronisasi	Penggunaan blok “wait”.	Penggunaan blok “broadcast”, “when I receive message”, “stop all”, “stop program” atau “stop programs sprite”.	Penggunaan blok “wait until”, “when the backdrop change to” atau “broadcast and wait”.

Konsep CT	Indikator Level Kompetensi		
	Dasar	Sedang	Tinggi
Flow Control	Penggunaan blok-blok sekuensial.	Penggunaan blok “repeat” atau “forever”.	Penggunaan blok “repeat until”.
Interaktivitas dengan Pengguna	Penggunaan blok “green flag”.	Penggunaan blok “key pressed”, “sprite clicked”, “ask and wait” atau blok-blok yang melibatkan interaksi melalui mouse.	Penggunaan blok-blok untuk audio atau video.
Representasi Data	Penggunaan sprite modifiers.	Penggunaan operator-operator untuk variabel.	Pengoperasian <i>list</i> .

### 01.03. Lembar Kerja Reflektif Individual (Pertanyaan Diskusi di Kelas)

Tuliskan hal baru apa sajakah yang Anda dapatkan dari mempelajari CT melalui Pemrograman dengan Scratch!

--

### 01.04. Lembar Kerja Mahasiswa (Dikumpulkan dan Dinilai)

Secara individual, mahasiswa mengerjakan sebuah projek sederhana, yaitu membuat sebuah permainan menebak angka. Sebelum memulai membuat projek ini, ingatlah bahwa mengambil karya orang lain dalam bentuk kode program termasuk ke dalam tindakan plagiarisme.

Pada permainan ini, pemain akan menebak sebuah angka acak dari 1 hingga 100. Lalu, sprite akan memberitahukan apakah tebakan benar, terlalu kecil, atau

terlalu besar. Di awal permainan, akan ditampilkan interaksi seperti yang ditunjukkan pada Gambar B.6. Tampilan ini akan terus menerus ditampilkan selama tebakan dari pemain belum tepat.



*Gambar B.6: Tampilan Awal Permainan Tebak Angka*

Gambar B.7 adalah respon dari permainan jika tebakan yang dimasukkan oleh pemain lebih kecil daripada angka yang seharusnya. Setelah itu, program akan menampilkan kembali tampilan awal seperti semula.



*Gambar B.7: Respon Program Jika Tebakan Terlalu Kecil*

Pada Gambar B.8 dapat dilihat respon permainan jika tebakan pemain terlalu besar. Setelah itu, program akan menampilkan kembali tampilan awal seperti semula.



*Gambar B.8: Respon Program Jika Tebakan Terlalu Besar*

Jika tebakan tepat, maka respon dari permainan adalah seperti Gambar B.9. Setelah itu, program akan menampilkan kembali tampilan awal seperti semula.



*Gambar B.9: Respon Program Jika Tebakan Tepat*

Isilah lembar kerja berikut ini!

<b>Nama:</b>	
<b>NIM:</b>	
1. Mengacu ke contoh pada bagian 01.02.01, jelaskanlah bagaimana Anda menggunakan fondasi CT untuk	<b><u>Dekomposisi:</u></b>  <b><u>Algoritma:</u></b>

memprogram permainan tebak angka di atas! Anda bisa melengkapi dengan konsep CT lainnya dari Tabel B.2!	<u><b>Pengenalan</b></u> <u><b>Pola:</b></u>  <u><b>Abstraksi:</b></u>
2. Tuliskan atau tampilkanlah program Scratch yang telah Anda buat!	

## 02. Mulai dari Diri

### Jenis Aktivitas: Aktivitas Individual

Setelah mempelajari keterkaitan CT dan pemrograman, dan membuat program sederhana dengan Scratch, Anda selanjutnya perlu menuliskan kendala-kendala apa saja yang dihadapi pada saat Anda mempelajari hal-hal tersebut.

#### 02.01 Latar Belakang

Pemrograman komputer mulai masuk ke dalam kurikulum pendidikan dasar dan menengah. Dengan mempelajari pemrograman komputer, sebenarnya peserta didik diajak untuk dapat membuat solusi dari suatu *problem*, dengan cara membuat program komputer. Untuk dapat membuat program komputer yang menghasilkan solusi yang efektif, efisien, dan optimal, diperlukan keterampilan CT yang baik.

Jika selama ini solusi dari suatu permasalahan di bidang mata pelajaran terbatas pada solusi tertulis yang bukan berupa program, maka ada kemungkinan bahwa kedepannya, solusi yang dihasilkan oleh peserta didik dapat berupa program komputer. Oleh karena itu, penting bagi guru untuk juga dapat memahami sedikit konsep pemrograman sebelum mulai mengajar peserta didik di kelasnya masing-masing.



## 02.02 Pertanyaan Reflektif Individual (Pertanyaan Diskusi di Kelas)

Berdasarkan apa yang telah Anda pelajari tentang pemrograman dengan Scratch, serta bagaimana melakukan asesmen CT melalui pemrograman, tuliskanlah kendala-kendala apa yang Anda hadapi. Jelaskan pula bagaimana strategi Anda untuk mengatasi kendala tersebut!

Pertanyaan Reflektif	Respon
Apa saja kendala yang Anda hadapi ketika mempelajari pemrograman dengan Scratch?	
Apa saja kendala yang Anda hadapi ketika mempelajari asesmen CT melalui pemrograman?	
Tuliskan strategi Anda dalam mengatasi kendala-kendala yang telah Anda sebutkan di atas!	

## 03. Ruang Kolaborasi

### Jenis Aktivitas: Aktivitas Kelompok

Aktivitas ini akan dinilai. Hasil dari aktivitas ini yang berupa program, lembar kerja dan presentasi akan dikumpulkan bersama dengan bahan untuk presentasi pada demonstrasi kontekstual.

### Mekanisme Pembagian kelompok

Di Ruang Kolaborasi, mahasiswa secara kelompok akan membuat sebuah karya kreatif dalam bentuk program Scratch. Kelompok yang dibuat terdiri dari 2 - 3 orang. Jika memungkinkan, bentuklah kelompok yang sesuai jenjang. Akan lebih baik juga apabila anggota kelompok berasal dari bidang mata pelajaran yang sama, sehingga karya kreatif yang dihasilkan dapat sesuai dengan bidang mata pelajaran. Jika tidak memungkinkan, kelompok dapat terdiri dari anggota yang mengajar di berbagai jenjang dan berbagai mata pelajaran.

Karya kreatif yang dapat dihasilkan dapat beraneka rupa. Berikut ini beberapa ide yang dapat dipilih:

- Untuk bidang matematika, fisika, kimia, akuntansi, dapat dibuat sebuah program “kalkulator” yang dapat menghitung sesuatu. Di bidang fisika, misalnya dapat dibuat program yang dapat menghitung besarnya gaya berdasarkan massa dan percepatan.
- Untuk bidang lain seperti Bahasa Indonesia, Agama, Sejarah, Bahasa Inggris, atau Kesenian, dapat dibuat program yang isinya berupa animasi atau *story*. Pelajari Buku Informatika kelas VII (Wijanto et al., 2021), makalah (Fagerlund et al., 2021), (Moreno-León et al., 2015), dan tutorial Scratch yang dapat diakses melalui *link* <https://scratch.mit.edu/projects/editor/?tutorial=getStarted> agar Anda dapat lebih memahami bagaimana membuat animasi atau *story* dengan menggunakan Scratch.

Anda bebas membuat karya kreatif apapun sesuai dengan kreativitas Anda, tidak hanya mengimplementasikan ide yang diberikan.

### Lembar Kerja Kelompok

Isilah lembar kerja kelompok berikut ini!

<b>Nomor Kelompok</b>	:
<b>Anggota Kelompok</b>	: 1. _____  2. _____  3. _____
<b>Judul Program yang Dibuat</b>	
<b>Tuliskan secara singkat tentang program yang Anda buat. Apa yang menjadi masukan, bagaimana masukan akan diproses, dan jelaskan juga keluaran dari program Anda.</b>	

Contoh: program yang dibuat adalah program untuk menghitung besarnya gaya. Masukan dari program adalah massa benda dan percepatannya. Keluarannya berupa nilai gaya yang dituliskan pada <i>buble chat</i> dari <i>sprite</i> yang digunakan. Gaya dihitung berdasarkan rumus $F = m.a$	
Tampilkanlah program Scratch yang telah Anda buat!	

## 04. Demonstrasi Kontekstual

### Jenis Aktivitas: Aktivitas Kelompok

Mahasiswa mempresentasikan hasil karya kreatif yang telah dibuat secara berkelompok. Di dalam presentasinya, mahasiswa perlu menampilkan lembar kerja yang sudah diisi pada Ruang Kolaborasi serta menjelaskan fondasi CT yang digunakan dalam pembuatan tugas karya kreatif berupa program Scratch.

## 05. Elaborasi Pemahaman

Tuliskanlah hal-hal yang belum Anda pahami tentang CT dan pemrograman. Setelah itu, diskusikanlah bersama dengan dosen Anda!

Nama	
NIM	
Hal-hal yang belum dipahami terkait dengan CT dan <i>programming</i> :	

## 06. Koneksi Antar Materi

### Jenis Aktivitas: Aktivitas Individual (pertanyaan diskusi di kelas)

Setelah mempelajari CT serta pemrograman dengan Scratch, tuliskanlah bagaimana keterkaitan antara CT dengan pemrograman Scratch! Selanjutnya, tuliskan juga bagaimana pemrograman dengan Scratch akan dapat membantu proses pembelajaran Anda di kelas sesuai dengan bidang mata pelajaran yang Anda ajar! Tuliskan jawaban dari pertanyaan tersebut pada lembar kerja berikut ini!

<b>Nama</b>	
<b>NIM</b>	
Apa keterkaitan CT dengan pemrograman Scratch?	
Bagaimana pemrograman dengan Scratch dapat membantu proses pembelajaran di kelas, sesuai dengan bidang mata pelajaran yang Anda ajar?	

## 07. Aksi Nyata

### 07.01. Pertanyaan Reflektif (menjadi bagian portofolio)

#### Jenis Aktivitas: Aktivitas Individual

Jawablah pertanyaan-pertanyaan reflektif berikut!

1. Pengalaman apa saja yang Anda dapatkan dari mempelajari kaitan CT dengan pemrograman?

--

2. Bagaimana perasaan Anda pada saat mempelajari kaitan CT dengan pemrograman?

--

## 07.02. Portofolio Topik CT dan Pemrograman

Hal-hal yang menjadi bagian dari portofolio untuk topik CT dan proyek STEM adalah sebagai berikut:

1. Hasil Lembar Kerja 01.03, 01.04 dan 02.02.
2. Hasil kerja pada Ruang Kolaborasi.
3. Rancangan rubrik penilaian CT dalam pemrograman yang disusun pada Demonstrasi Kontekstual.
4. Hasil refleksi yang diisikan pada Aksi Nyata.