

# Panduan pemula belajar Scratch

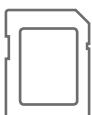
Bab 1

Scratch adalah sebuah lingkungan pemrograman visual. Dengan Scratch, kamu dapat membuat animasi, permainan dan gambar interaktif-mu sendiri. Dan, sambil kamu melakukan hal tersebut, kamu akan belajar beberapa prinsip penting teknik pemrograman tanpa harus mengetikkan kode pemrograman. Ini adalah salah satu cara terbaik untuk mulai belajar membuat program. Untuk mengetahui lebih jauh tentang Scratch, kunjungi alamat web **scratch.mit.edu**

Catatan:

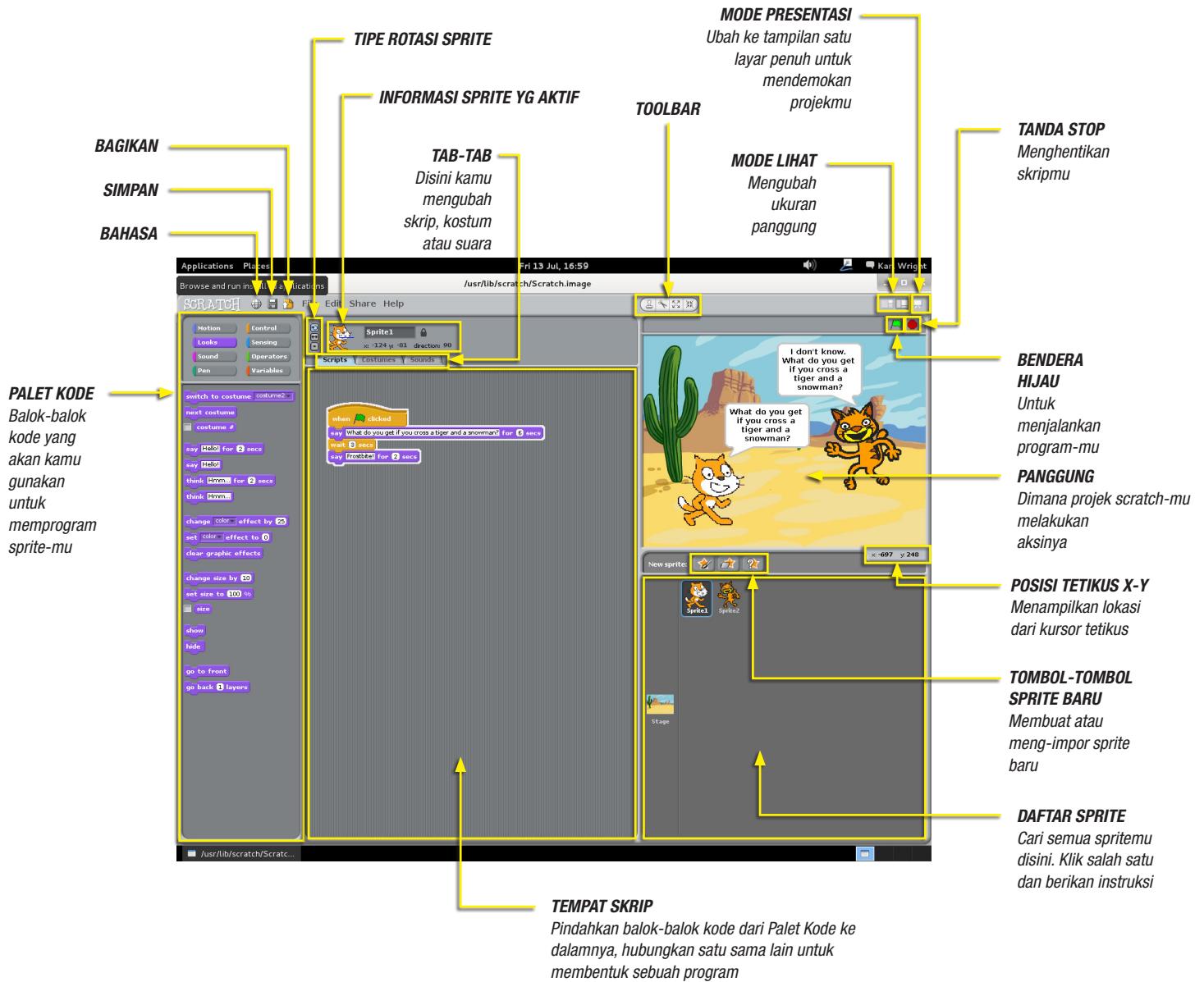
## Bagaimana menggunakan pedoman ini

Kami sudah membuat pedoman ini semudah mungkin untuk digunakan. Untuk membantumu melakukan latihan-latihan pada bab ini, kami telah mengumpulkan semua yang kamu butuhkan, seperti gambar latar belakang, kostum untuk sprite, efek suara dan contoh lengkap projek Scratch.



Semua berkas-berkas pendukung yang kamu butuhkan dapat ditemukan di dalam kartu SD belajar Raspberry Pi, di dalam folder /usr/share/scratch/RPiScratch. Setiap kali kamu melihat ada ikon gambar kartu SD di sebelah kiri tulisan, itu artinya kami merujuk ke sebuah file yang terdapat di dalam kartu SD Raspberry Pi-mu. Ayo lihatlah! Kamu juga dapat mengunduh berkas-berkas pendukung dari Google Drive di alamat <http://goo.gl/MpHUv>

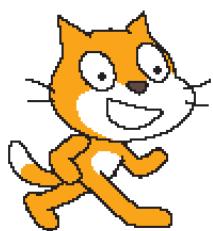
## Tampilan program Scratch



**TUJUAN LATIHAN:** Pada latihan ini, kamu akan belajar bagaimana menggunakan tampilan grafis (graphical user interface) Scratch (disingkat: GUI), serta bagaimana membuat karakter (sprite dan kostum) dan panggung (latar belakang) untuk projekmu, juga bagaimana menambahkan skrip.

**SUMBER:** Sprite “*kucing*” dan “*kucing\_romawi*”, dan latar belakang “*panggung\_romawi*”.

Pernahkah kamu ikut bermain sandiwara di sekolah? Jika kamu sudah pernah, kamu akan tahu untuk bermain sandiwara kamu membutuhkan panggung, aktor,



kostum dan sebuah skrip. Bayangkan Scratch mirip seperti sebuah sandiwara. Aktornya disebut dengan “**sprite**”.

Kamu dapat menghias aktormu dengan “**kostum**”, dan setiap sprite dapat memiliki lebih dari satu kostum. “**Panggung**” adalah area di layar dimana sprite-mu melakukan tugas yang kamu tulis untuk mereka.



Untuk membuat spritemu bergerak dan berkata, kamu harus memberikan instruksi. Kamu melakukannya dengan menuliskan “**skrip**” menggunakan balok-balok kode dari Palet Kode dan tab Semua Skrip di layar sebelah kiri.

Sudah cukup pengenalannya untuk saat ini; Mari kita mulai mencoba membuat program.

Jalankan Scratch dari menu Applications Raspberry Pi-mu. Kamu seharusnya saat ini melihat tampilan grafis Scratch, atau GUI (diucapkan “guuii”). Lihat sekitarmu dan centang kotak dibawah item yang kamu temukan:

- 1.** Panggung (layar besar berwarna putih)
- 2.** Sprite (petunjuk: seekor kucing)
- 3.** Dua kostum yang dapat dipakai kucingmu (klik tab Semua Kostum)
- 4.** Tab Semua Skrip

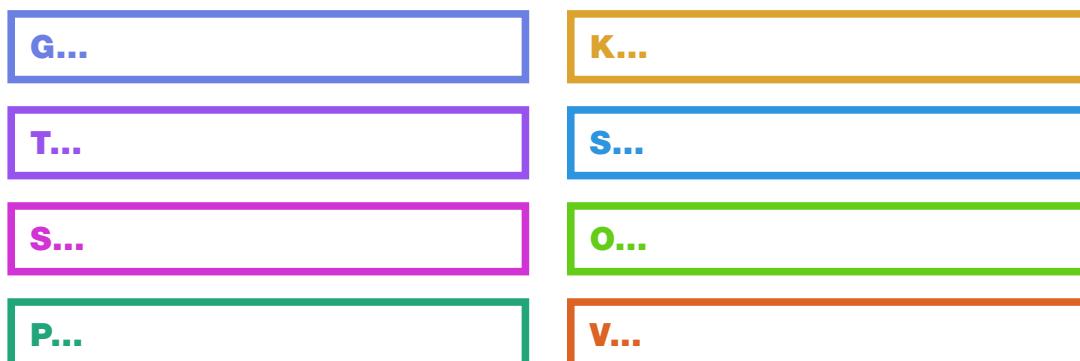
## Mari bersenang-senang dengan si kucing

Pertama-tama, kita buat si kucing mengatakan sesuatu. Kita akan mulai dengan berkata “Halo, Dunia!”. Ini adalah hal yang umumnya dilakukan oleh programer komputer saat belajar pertama kali (jangan tanya saya mengapa). Karena kamu saat ini baru mulai belajar membuat program, kamupun sebaiknya memulai dengan berkata “Halo, Dunia!” juga.

## Membuat si kucing berkata

Untuk membuat si kucing berkata “Halo, Dunia！”, kita akan menggunakan “**balok kode**”. Balok kode ini adalah kode yang banyak kegunaannya, setiap balok berisi sebuah instruksi yang akan diikuti oleh sprite-mu.

Ada delapan macam tipe balok kode. Balok kode ini dapat ditemukan di pojok kiri atas dari Scratch GUI. Mereka diberi warna yang berbeda sesuai kelompok fungsinya, jadi ingat-ingatlah warnanya. Cari tahu apa saja dan lengkapi namanya pada tabel dibawah:



**Sekarang, ikuti langkah sederhana berikut ini untuk membuat si kucing berkata:**



- 1** Klik sprite si kucing pada daftar Sprite (kanan bawah) untuk memastikan dia terpilih.

- 2** Klik tombol “Tampilan” pada Palet Kode untuk melihat balok-balok Tampilan.

- 3** Klik balok ber-label “berkata [Assalamualaikum!] selama [2] detik” dan pindahkan ke tab Semua Skrip.

- 4** Ganti “Assalamualaikum!” dengan “Halo, Dunia!”. Klik dua kali balok tersebut dan kucingmu akan berkata: “Hallo, Dunia!”.

Kita harus menjalankan sebuah program untuk melakukannya. Kamu dapat melakukannya dengan mengklik dua kali balok kodemu, jika kamu hanya memiliki satu balok saja. Tetapi apabila kita memiliki lebih dari satu balok, kita mungkin ingin membuat semua balok untuk jalan bersamaan. Kita dapat menggunakan “event bendera-hijau” untuk melakukannya.



#### **CARA MENCARI EVENT BENDERA-HIJAU:**

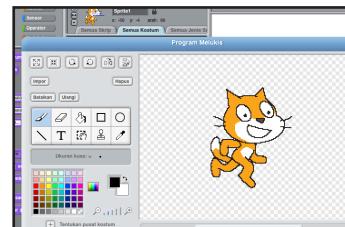
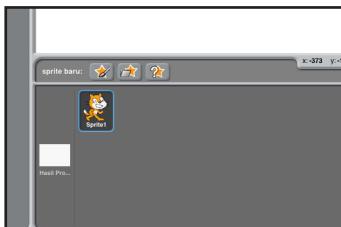
- 1.** Klik tombol Kontrol pada Palet Kode.
- 2.** Cari balok yang ber-label “ketika [gambar sebuah bendera hijau] diklik”.
- 3.** Pilih, kemudian pindahkan ke atas skrip yang telah kamu buat di tab Semua Skrip. Pastikan balok kodennya terpasang di tempat yang benar.

Kamu sekarang sudah siap menjalankan program Scratch-mu yang pertama secara benar. Cukup klik simbol bendera hijau di pojok kanan atas dari jendela Scratch, tepat diatas panggung, dan perhatikan si kucing melakukan kegiatannya.

#### **Giliranmu**

**PERTANYAAN:** Berapa lama si kucing berkata “Halo, Dunia!”? \_\_\_\_\_ detik

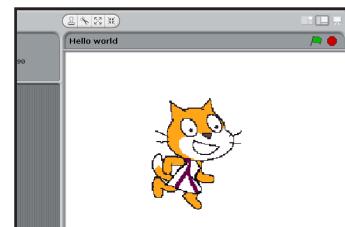
**TUGAS:** Coba ubah kodennya supaya si kucing berkata “Halo, Dunia!” selama 5 detik.



**1** Klik sprite-mu untuk memilih. Pada area Semua Skrip, klik tab Semua Kostum.

**2** Kita akan membuat kostum ketiga untuk si kucing, jadi klik tombol Perbanyak. Kostum kucing yang baru akan muncul.

**3** Pilih "kostum3" dan klik Edit. Ini akan membuka Program Melukis. Cobalah semua tombol dan alat untuk mengetahui kegunaannya.



**4** Setelah kamu nyaman, gambar pakaian pada kostum dan klik OK. Aku berikan jubah ke spriteku supaya terlihat seperti Kaisar Romawi.

**5** Kemudian, pilih tab Semua Skrip, klik tombol Tampilan dan pilih balok "ganti dengan kostum [ ]".

**6** Pindahkan kebawah tab Semua Skrip dan gunakan menu drop-down utk memilih "kostum3". Klik dua kali balok ini, si kucing akan berganti

Sekarang kamu memiliki tiga balok kode pada tab Semua Skrip, satu utk bendera hijau, kedua utk berkata "Halo, Dunia!" dan terakhir utk ganti kostum. Kamu dpt menyatukan balok ganti kostum dengan memindahkannya tepat dibawah balok berkata "Halo, Dunia!". Jika terlihat garis putih, kedua balok kode akan menyatu. Dua atau lebih balok kode menyatu bersama disebut sebagai sebuah "skrip".

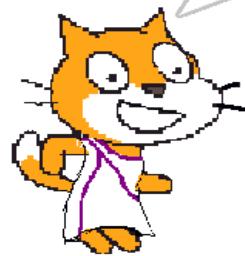
## Giliranmu

**PERTANYAAN:** Sekarang kita memiliki skrip dengan tiga balok kode, apa yang terjadi ketika kamu klik dua kali?

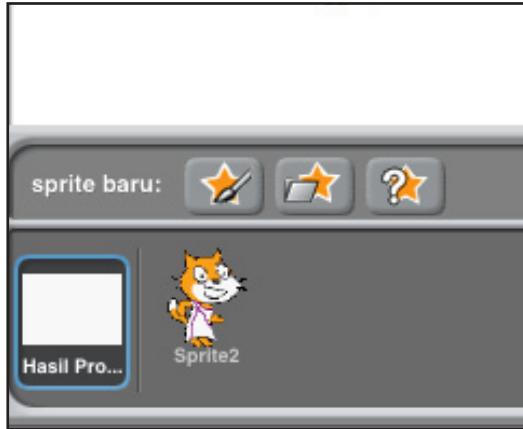
**TUGAS:** Coba kamu atur empat balok kode untuk membuat si kucing berubah ke kostum jubah, mengatakan "Halo, Dunia!", kemudian kembali lagi ke kostum yang normal.

Teman2, rakyat romawi, aku butuh sebuah panggung untuk pentas!

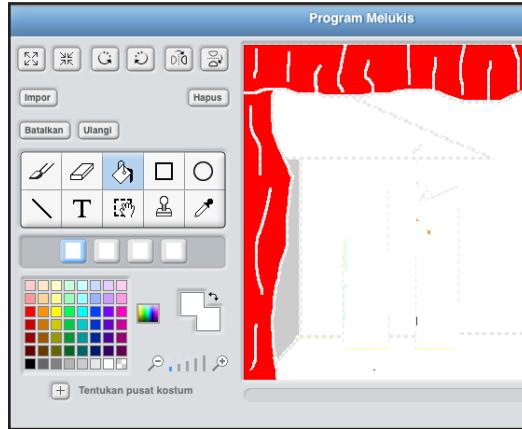
Kucing yang dikanan: sepertinya dia tersesat di dalam hujan badai. Kita harus memberinya panggung untuk dia pentas.



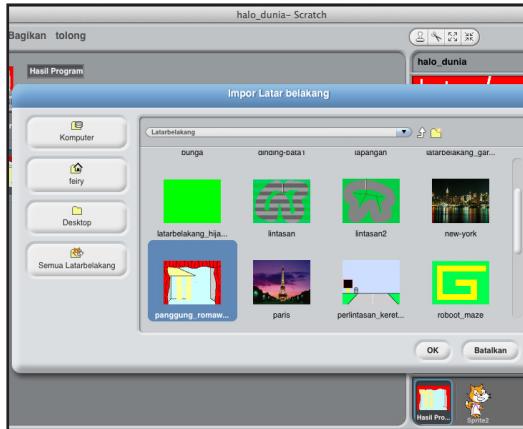
Saatnya memberi panggung pada si kucing. Kita bisa bermalasan dan impor saja latar belakangnya, tetapi anggap saja kita sedang bersemangat dan ingin menggambar sendiri.



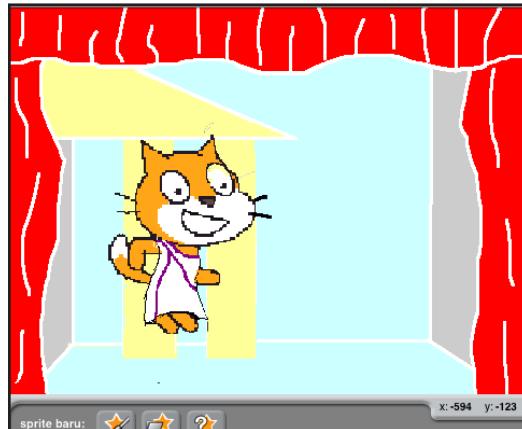
- 1** Klik panggung pada daftar Sprite (pojok kanan bawah layar). Lalu klik tab Semua Latarbelakang dan klik tombol Edit.



- 2** Seperti sebelumnya, Program Melukis akan terbuka. Gambarlah panggung untuk sprite-mu. Setelah selesai, klik tombol OK. Kamu dapat mengubahnya lebih lanjut dengan klik tombol Edit.



- 3** Selain itu, kamu dapat meng-impor latar belakang yang sudah jadi. Pilih Panggung, kemudian Semua Latarbelakang lalu klik tombol Impor.



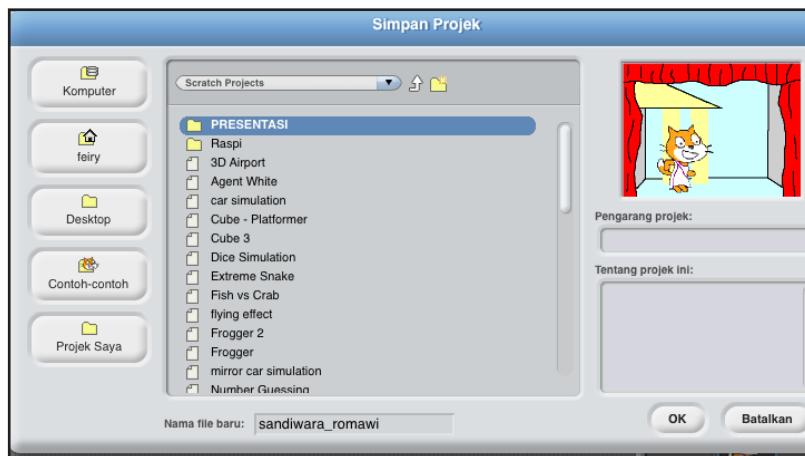
- 4** Lihatlah semua latar belakang yang tersedia sebelum kamu memilih latar belakang yang kamu inginkan. Kami pilih "panggung\_romawi". Pilih latar belakang dengan meng-kliknya, kemudian klik tombol OK.

## Simpan pekerjaanmu

Catatan:

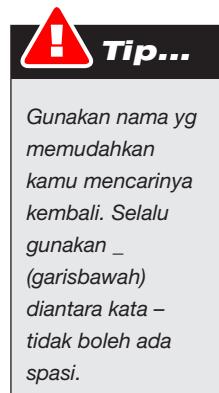
Ini saat yang tepat untuk menyimpan pekerjaanmu. Alangkah baiknya apabila kamu menyimpannya setiap 10 menit, sehingga kamu dapat yakin tidak akan kehilangan hasil kerja kerasmu. Saat mengerjakan projek besar, simpan di dua tempat berbeda, sehingga kamu punya cadangannya.

Untuk menyimpan projek, klik File, kemudian Simpan – jendela Simpan Projek akan muncul.



Secara default, scratch akan menyimpan projekmu pada folder Scratch Projects. Ini adalah tempat yang umum, jadi kamu tinggal ketikkan nama filenya, dibawah. aku memberi nama fileku “sandiwara\_romawi”, pilihlah nama yang berbeda untuk projekmu atau kamu akan menimpa pekerjaanku ke fileku! Klik tombol OK untuk menyimpan.

Wow! Sudah banyak sekali yang dikerjakan untuk pelajaran pertama. Bermainlah dengan Scratch – cobalah dengan kotak-kotak kode yang berbeda untuk mengetahui apa fungsinya. Dan kembalilah setelah kamu cukup beristirahat dan ingin mencoba Pelajaran 2.



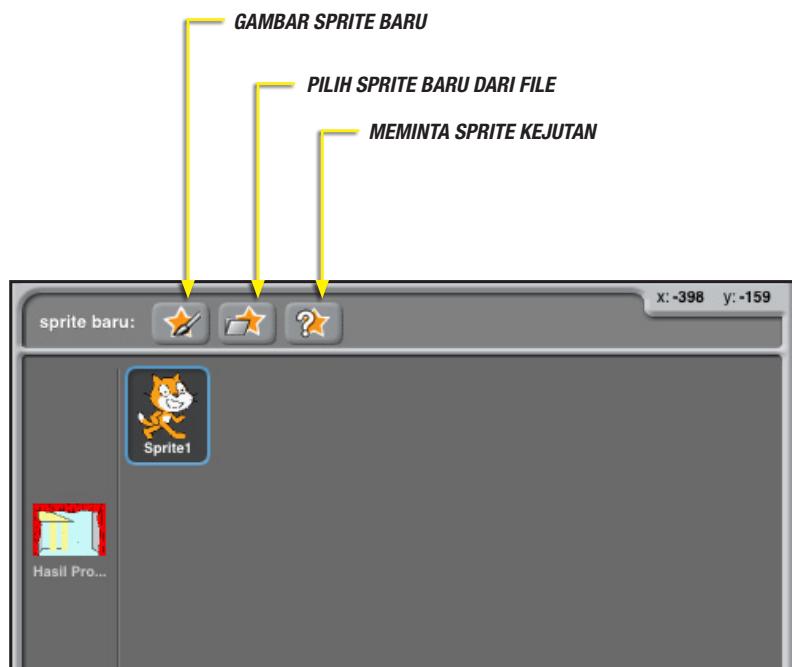
*Gunakan nama yg  
memudahkan  
kamu mencarinya  
kembali. Selalu  
gunakan \_  
(garisbawah)  
diantara kata –  
tidak boleh ada  
spasi.*



**TUJUAN LATIHAN:** Pada latihan ini, kamu akan belajar bagaimana menggerakkan sprite di panggung secara terkontrol dan bagaimana membuat lelucon.

**SUMBER:** Sprite "kucing\_romawi" dan latar belakang "panggung\_romawi".

Si kucing merasa kesepian, jadi sebaiknya kita membuat beberapa karakter sebagai teman bermain. Kamu dapat gambar sprite-mu sendiri atau meng-impor dari folder Costumes. Gunakan tombol sprite baru untuk melakukannya.



Pada bagian sebelah kanan program, tepat dibawah panggung dan diatas Daftar Sprite, kamu akan menemukan tiga tombol: tombol-tombol Sprite Baru. Ini yang akan kita gunakan.

Aku akan menambahkan bocah penjelajah waktu ke panggungku. Supaya tetap sederhana, dan kita bisa fokus ke pembuatan program, kita akan impor saja spritenya.

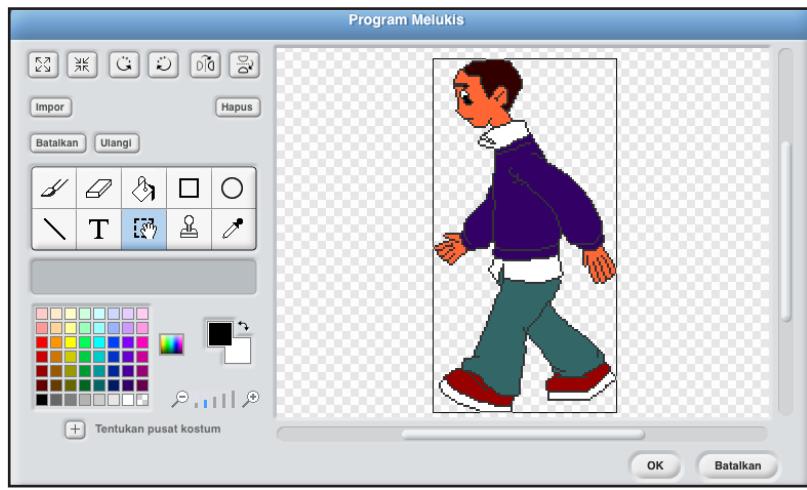
Klik tombol tengah dari Sprite Baru dan impor "boy4-walking-c". Tetapi tunggu

**Gunakan tombol impor sprite untuk mencari dan meng-impor sprite "boy4-walking-c".**



*Catatan:*

**Dari tab Semua Kostum, Klik Edit dan gunakan tombol Balikkan gambar secara horizontal.**



sebentar: dia menghadap ke arah yang salah! Tidak masalah. Pergi ke tab Semua Kostum dan klik tombol Edit. Gunakan tombol Balikkan gambar secara horizontal untuk membuat si bocah menghadap ke kiri.

Disana juga terdapat tombol-tombol untuk membuat spritemu membesar, mengecil, berputar berlawanan arah jarum jam, berputar searah jarum jam, demikian juga membalikkan gambar secara horizontal dan vertikal. Cobalah semuanya. Aku juga menggunakan tombol perkecil untuk mengecilkan spriteku.

## Buat spritemu membuat lelucon

Mari buat spritemu membuat lelucon satu sama lain. Untuk melakukannya kamu dapat menggunakan balok berkata pada kategori Tampilan.

Kamu dapat memulai dengan lelucon sederhana 'tok tok' untuk memulai.

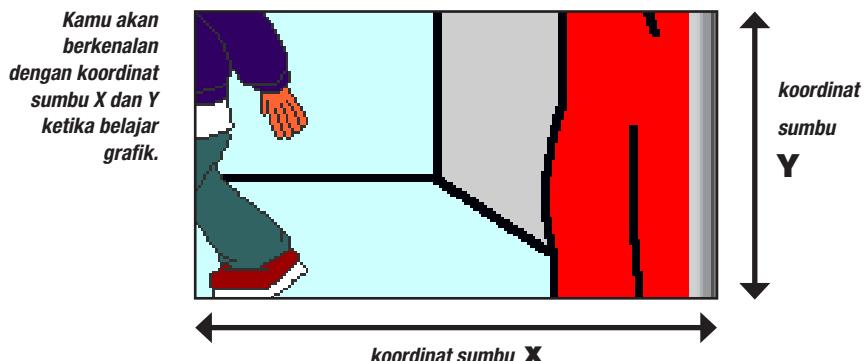
Tapi tunggu! Apakah kamu menemukan kedua spritemu bicara berbarengan.

Untuk memperbaikinya, dari balok Kontrol tambahkan balok "tunggu [1] detik" pada sprite kedua, diatas balok "berkata".

## Merubah posisi sprite

Catatan:

Oke, kita sudah membuat lelucon. Tapi sandiwaranya terlihat kaku, jadi mari kita buat karakter kita bergerak. Tugas pertama adalah memindahkan kedua karakter kita ke titik awal masing-masing. Pada sandiwaraku, si kucing akan datang dari arah kiri dan si bocah datang dari arah kanan.

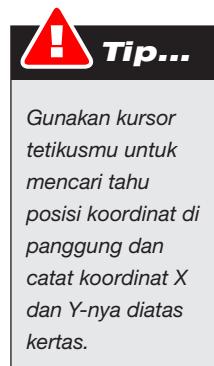


Koordinat dari setiap titik di panggung akan terlihat di bagian pojok kanan bawah panggung. Gerakkan kursor tetikusmu di dalam panggung dan perhatikan angka X dan Y nya berubah.

**TUGAS:** Gunakan tetikus-mu untuk menemukan titik tengah panggung. Pindahkan kursor tetikusmu sampai dia berada tepat di titik x: 0 y: 0. Sekarang mari kita pindahkan posisi kedua sprite kita.

1. Pilih sprite si kucing, pada Palet Kode, klik kategori Gerakan kemudian klik balok berlabel “pindah ke x: [0] y: [0]”.
2. Ubah nilainya menjadi x: -240 y: -80. Ini akan membuat si kucing pindah ke posisi sebelah kiri panggung.
3. Kemudian letakkan balok “tunggu selama [1] detik” (dari Kontrol) ke dalam skripmu. Supaya kamu bisa melihat kucingmu sebelum bergerak.
4. Sekarang tambahkan balok kedua “pindah ke x: [0] y: [0]”. Gunakan tetikusmu untuk mencari tahu nilai x coordinate dari posisi agak di sebelah kiri titik tengah panggung, posisi yang kita inginkan si kucing bergerak.

Ulangi langkah diatas untuk sprite kedua, letakkan pada posisi agak di sebelah kanan titik tengah panggung. Idealnya, kedua sprite harus bergerak dari pinggir panggung menuju ke tengah layar bertatapan muka, dipisahkan pada jarak yg pendek.



```
ketika [diklik] 
  pindah ke x: 220 y: -90
  tunggu selama 1 detik
  pindah ke x: 60 y: -90
  tunggu selama 2 detik
  berkata [Mengapa bangsa romawi membuat jalannya lurus?] selama 2 detik
```

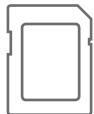
Kode kami untuk sprite bocah adalah seperti ini

```
ketika [diklik]
  pindah ke x: -240 y: -80
  tunggu selama 3 detik
  pindah ke x: -70 y: -80
  tunggu selama 2 detik
  berkata [Supaya tentara kita tidak berjalan melengkung!] selama 2 detik
```

Dan ini adalah kode kami untuk sprite si kucing. Apakah kode milikmu sama dengan ini?

Nah sekarang kamu harus buat kedua spritemu membuat lelucon. Ingatlah untuk memberikan jeda pendek setiap kali mereka berkata, kalau tidak mereka akan bicara berbarengan. Silahkan lihat gambar rekaman layar untuk melihat kode kami (dan lawakan kami yang sangat lucu).

## Giliranmu



**TUGAS:** Sekarang tambahkan kode untuk menggerakkan karaktermu ke sebelah kanan panggung dan setelah jeda pendek kembali ke titik tengah panggung.

**Selamat! Kamu telah mengerti cara menggerakkan sprite di panggung. Mengapa kamu tidak coba untuk menambahkan beberapa karakter baru lagi dan buat mereka membuat lawakan?**

*Jika kamu menghadapi kesulitan, kamu dapat membuka sampel kode, "sandiwara\_romawi.sb", untuk melihat bagaimana programnya dipasangkan. Silahkan untuk mengubah-ubah dan bereksperimen, karena itu adalah cara belajar yang sangat baik.*

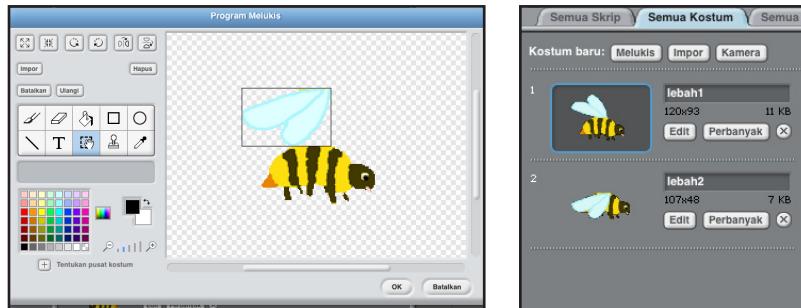
## Pelajaran 1.3: Animasi (perulangan)

**TUJUAN BELAJAR:** Pada latihan ini, kamu akan belajar bagaimana menggunakan perulangan untuk membuat animasi sederhana.

**SUMBER:** Sprite "lebah", "bunga\_betina" dan "bunga\_jantan", dan latar belakang "taman\_bunga".

Dengan karakter yang dapat dibuat animasi, Scratch sangat bagus sebagai alat untuk bercerita. Aku harus mengerjakan projek sains sekolah tentang penyerbukan, aku memutuskan menggunakan Scratch untuk bercerita tentang penyerbukan dengan menggunakan gambar bergerak. Kamu dapat membantuku dengan mengikuti perintah berikut untuk membuat animasi lebah terbang.

**Salin lebahmu, kemudian edit "lebah2" ubah sayapnya supaya menghadap kebawah.**



Pertama-tama, buka file “bee1” dari folder “Animals” didalam galeri Scratch. Kemudian, impor latar belakang “flower\_bed”, yang ini ada pada folder “Nature” pada galeri Scratch. Hapus sprite kucing; kita tidak gunakan pada projek ini.

Salin “bee1”, lalu edit “bee2” menggunakan Alat Pengatur Pilihan dan Membalikkan gambar secara horizontal, untuk membuat sayapnya menghadap ke bawah. Bersama-sama, kedua kostum – “bee1” dan “bee2” – akan menjadi animasi lebah terbang.

**Ini adalah kode untuk membuat lebahmu terbang. Untuk menjalankan kodeku aku tidak menggunakan ketika bendera hijau di klik tetapi aku akan menggunakan kotak kode “ketika Sprite1 diklik” dari kategori Kontrol. Kodenya akan berjalan ketika aku mengklik si lebah.**



Kita membutuhkan beberapa kode untuk membuat si lebah kelihatan terbang. Kita melakukannya dengan menggantikan kostum berulang-ulang, membuat si lebah terlihat seperti mengepakan sayapnya. Bersamaan dengan itu kita juga akan menggerakkan si lebah maju ke depan.

Sekarang, buatlah skripmu sendiri untuk membuat si lebah terbang. Kamu membutuhkan balok kode dari kategori Kontrol, Tampilan dan Gerakan. Jika kamu mengalami kesulitan, lihatlah rekaman gambar kode kami. Kamu akan menemukan lebih jauh di dalam pelajaran.

#### **Ini adalah langkah-langkah yang harus diikuti:**

1. Mulai dengan kostum “bee1”.
2. Tambahkan balok “tunggu selama [0.2] detik”, sehingga penonton sempat melihat kostumnya.
3. Gerakkan si lebah sebanyak 10 step, sebelum diganti dengan kostum “bee2”.
4. Tambahkan balok “tunggu selama [0.2] detik” lagi, supaya penonton sempat melihat kostum ke dua.
5. Gerakkan si lebah sebanyak 10 step lagi.

Tetapi kita harus melakukannya lebih dari satu kali. Supaya si lebah bisa terbang sejauh ukuran panggung, kita harus mengulanginya sebanyak 20 kali.

**Jangan panik!** Kamu menggunakan komputer. Komputer itu sangat hebat untuk mengerjakan pekerjaan yang berulang-ulang. Mereka dapat melakukan ini dengan sangat akurat dan tidak pernah bosan, capek atau kesal.

Yang kita butuhkan adalah perulangan.

Catatan:



Ini yang kita gunakan untuk mem-program komputer untuk mengulangi sesuatu berulang-ulang. Kamu akan menemukan balok ulangi (“ulangi [10]”) pada kategori Kontrol.

Balok ulangi ini terlihat sedikit berbeda karena ada tempat kosong ditengahnya agar kita dapat meletakkan kode yang ingin diulangi didalamnya. Pindahkan balok “ulangi [10]” dibawah balok “ketika Sprite1 diklik”. Scratch akan otomatis memasukkan semua kode dibawahnya ke dalam balok perulangan.

**Gunakan iterasi perulangan, kamu dapat membuat sprite melakukan aksi yang sama berulang-ulang. Sehingga si lebah menepakkan sayapnya keatas dan kebawah berulang ulang.**



Ini adalah kode yang kita butuhkan untuk menggerakkan si lebah agar dia terbang dari ujung ke ujung panggung.

## Giliranmu

**PERTANYAAN:** Coba tebak mengapa aku menambah jumlah perulangan dari 10 menjadi 20?

**TUGAS:** Beberapa sprite Scratch sudah memiliki dua kostum. Perhatikan sprite kucing. Gunakan kedua kostumnya dan kode yang mirip dengan contoh untuk membuatnya bergerak.



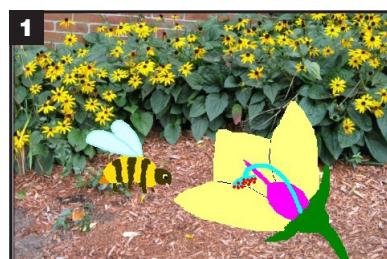
**Tip...**

Jika lebahmu terbang ke arah yg salah, cek spritemu pastikan dia menghadap ke arah yg benar. Dengan sprite terpilih, lihat bar diatas tab Semua Skrip. Arah “depan” dari sprite terlihat dari indikator garis berwarna biru. Lebah di sebelah kiri akan bergerak 90° terhadap vertikal dan yg sebelah kanan akan bergerak 60° terhadap vertikal. Kamu dpt memutar garisnya utk memudahkan arah sprite.

**Berikut adalah gambar cerita dari projek penyerbukan saya. Saya tambahkan dua kostum lagi pada sprite lebah saya untuk menunjukkan si lebah membawa serbuk sari dan saya juga telah menggambar sebuah sprite bunga dengan beberapa benang sari berwarna biru.**

Tapi aku tidak mau berhenti disini. Aku ingin si lebah mengunjungi bunga kedua – bunga betina – dari layar berikutnya, jadi aku menyalin keempat kostum lebah dan membalikkannya ke arah horizontal.

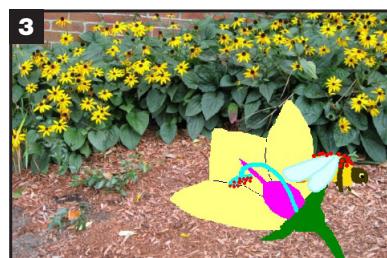
Aku juga menyalin, membalikkan dan meng-edit bunga jantan untuk membuat sprite bunga betina. Aku sudah menyiapkan dua kostum. satu menampilkan kepala putik tanpa serbuk sari dan yang satunya dengan serbuk sari. Mari kita lihat hasil animasinya.



**1.** Lebah terbang ke bunga jantan.



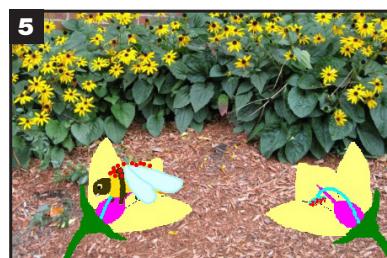
**2.** Lebah berhenti untuk mengambil madu dan mengambil serbuk sari.



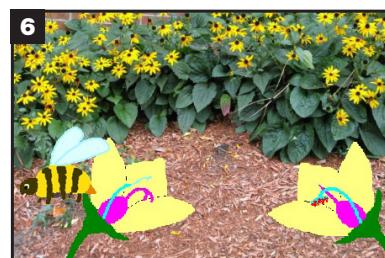
**3.** Lebah lalu terbang membawa serbuk sari dari benang sari bunga.



**4.** Lebah terbang ke arah bunga betina.



**5.** Lebah lalu mengambil madu dan meninggalkan serbuk sari.



**6.** Lebah terbang dan meninggalkan serbuk sari pada kepala putik.

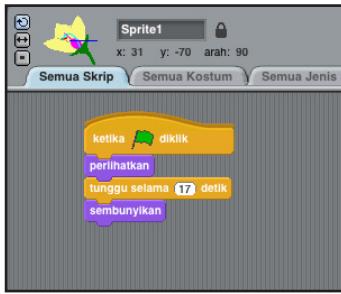
Aku juga memutuskan hanya ingin ada satu bunga di satu layar. Jadi, aku harus menambahkan skrip untuk membuat bungaku hilang dan muncul pada titik yang diinginkan pada animasi. Aku menggunakan balok “tampilkan” dan “sembunyikan” pada kategori Tampilan untuk kebutuhan ini.

Berikut adalah kode untuk ketiga sprite di projek ini.

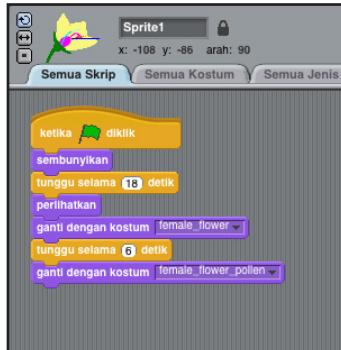
#### Kode untuk lebah



#### Kode untuk bunga jantan

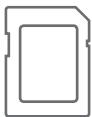


#### Kode untuk bunga betina



Karena aku memiliki tiga sprite dengan skrip masing-masing, aku menggunakan balok event “ketika [bendera hijau] diklik” untuk menjalankan semuanya secara bersamaan.

**Wow! Ini projek yang sangat rumit, tetapi jika kamu memecahkan animasinya menjadi banyak adegan-adegan kecil akan membuat perencanaan dan pembuatan programnya menjadi lebih mudah.**



Untuk melihat keseluruhan projek seperti apa setelah selesai, buka RPiScratch/Projects/pollination.

**Tip...**  
Impor sprite yang ada dan latar belakang dari folder gambar Scratch. Ini akan meningkat banyak waktumu..

**TUJUAN BELAJAR:** Kamu akan belajar bagaimana menggunakan variabel untuk menyimpan data. Kamu juga akan belajar menggunakan operator untuk melakukan penjumlahan sederhana.

**SUMBER:** Sprite standar “cat” pada latar belakang putih standar.



#### Apakah kamu merasa matematika sulit?

Dapatkah kamu membayangkan bagaimana rasanya apabila kamu bisa melakukan jutaan perhitungan dalam hitungan detik dan selalu benar? Bahkan perhitungan yang paling sulit sekalipun?

Komputer itu hebat sekali dalam matematika. Kenyataannya, matematika adalah kemampuan terbaik dari sebuah komputer. Kita dapat mem-program kucing Scratch untuk mengerjakan tugas matematika. Si kucing akan menanyakan beberapa angka dan kemudian mengerjakan perhitungannya. Jadi, bagaimana kita menyimpan angka-angka ke dalam program untuk digunakan oleh si kucing?

Ketika kita memasukkan angka (menyimpan angka ke dalam komputer), komputer harus memiliki tempat untuk menyimpan angka-angka tersebut. Beda orang dapat memasukkan angka yang berbeda pula, sehingga angka-angka ini akan selalu berbeda setiap saat.

Ketika mem-program, kita menyimpan angka kedalam sebuah “**variabel**”. Salah satu cara untuk membayangkan sebuah variabel adalah bayangan variabel adalah sebuah kotak, atau tempat, dimana kita dapat menyimpan angka, huruf atau kata-kata.

Kita dapat memiliki lebih dari satu variabel di dalam program, sehingga kita harus memberikan nama yang berbeda kepada masing-masing variabel. Nama variabel bisa sesederhana hanya berupa satu huruf (atau serumit apapun yang kamu suka!). Sebagai contoh, jika kita menyimpan sebuah angka, kita dapat menyebutnya variabel “n”.



Catatan:



**Tip...**

Berikan setiap variabel dengan nama sesuai dengan nilai yang disimpan.

Contoh,  
Jika kamu membuat game dan menggunakan variable untuk menyimpan score, nama yang bagus untuk variabel itu adalah "game\_score".

**Pada diagram, kita menyimpan angka "7" ke dalam variabel "n". Sehingga kita dapat mengatakan "n = 7".**

Pada contoh diatas, kita membuat variabel bernama "n" dan menyimpan angka "7" ke dalamnya. Pada Scratch, kamu akan melakukannya dengan dua langkah: pertama membuat variabel "n", kemudian menggunakan balok kode pada kategori Variabel untuk men-set nilainya dengan angka "7".



Jika kamu ingin menggunakan nama variabel yang terdiri dari dua kata, kamu harus memisahkan kedua kata dengan garis bawah (karakter "\_"), bukan dengan spasi.

## Menggunakan variabel di dalam Scratch

Catatan:

Ok, sekarang kita akan membuat dan menggunakan variabel. Klik Variabel pada Kotak Palet dan buat variabel dengan nama “game\_score”.



1 Klik tombol Variabel pada Kotak Palet, lalu klik tombol “Buat variabel”. Ini akan membuka jendela dialog “Nama variabel?”. Masukkan nama “game\_score” untuk nama variabelmu dan klik OK.

2 Tarik “Set [game\_score] sebesar [0]” ke dalam tab Semua Skrip. Lalu, dari Kotak Palet Kontrol, tarik kotak kode “ketika Sprite1 diklik”. Gabungkan keduanya utk membentuk skrip.



3 Nilai default dari variabel baru adalah “0”. Pilih kotak kode “set [game\_score] sebesar [0]” dan ganti nilainya menjadi “100”.

4 Terakhir, dari Tampilan, Tarik kotak kode “berkata [Assalamualaikum!] selama [2] detik” ganti dengan “berkata [Horee, Aku dapat 100 point!] selama [2] detik”.

Pada contoh ini, kita set score kucing menjadi 100 dengan mengklik sprite kucing. Tetapi ada banyak lagi cara untuk men-set nilai variabel. Cara yang kamu pilih bergantung dari bagaimana kamu akan menggunakan variabel tersebut.

## Masukkan angka-angka

Gurumu telah menyiapkan empat perhitungan sebagai berikut:

$$\begin{aligned} 56 + 39 &= \\ 87 - 42 &= \\ 16 \times 9 &= \\ 240 \div 6 &= \end{aligned}$$

Kami akan menunjukkan bagaimana membuat si Kucing Matematika melakukan pekerjaan rumah untukmu. Ayo kita mulai dengan soal pertama,  $56 + 39$ . Jika kamu kesulitan, lihat gambar skrip lengkap yang ada di akhir latihan ini.

Catatan:



Pertama, buat tiga variabel, dan beri nama dua variabel pertama dengan "first\_number" dan "second\_number". Kamu akan menggunakan dua variabel ini untuk menyimpan kedua angka. Beri nama variabel ketiga dengan "result". Disini kamu menyimpan hasil perhitungan.



Untuk memasukkan angka pertama kamu membutuhkan balok dari Palet Sensor. Tarik balok "tanya [Siapa namamu?] dan tunggu" ke tab Semua Skrip. Ganti tulisannya dengan "Berapa nilai angka pertama?".

Jawaban dari yang dimasukkan oleh pemain kemudian dimasukkan ke variabel, yang disimpan untuk digunakan pada perhitungan, dengan mengkombinasikan balok set "variabel" dari Kotak Palet Variabel dan balok "jawaban" dari Kotak Palet Sensor. Mari kita lihat bagaimana caranya.



Klik Kotak Palet Variabel, pilih balok "set [first\_number] sebesar [0]", tarik ke tab Semua Skrip dan tempelkan ke balok sebelumnya.



Klik Kotak Palet Sensor, pilih balok dengan nama "jawaban" dan tarik ke angka "0" dari balok sebelumnya. Skripmu harusnya terlihat seperti rekaman gambar diatas.

Catatan:



Sekarang program mengetahui nilai dari angka pertama. Untuk memberitahu nilai angka kedua, ulangi langkah-langkah diatas tetapi ingatlah untuk mengganti “first\_number” menjadi “second\_number” pada kotak pilihan dari balok set variabel.

### Sekarang bagian yang sangat cerdas

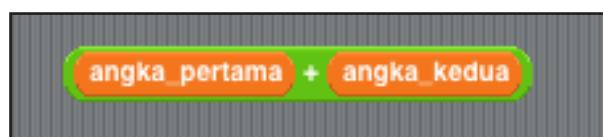
Untuk melakukan perhitungan dengan Scratch, kamu membutuhkan sesuatu yang bernama “operator”. Itu hanyalah istilah keren dari tanda matematika +, -, × dan ÷. Ya, tebakanmu benar, kita dapat menemukannya pada Kotak Palet hijau Operator.

Klik Operator:  
operator “tambah”  
“[ ] + [ ]” berada  
pada baris pertama  
dalam daftar Kotak  
Palet Operator.



Kita akan menggunakan operator “tambah”, jadi tarik ke dalam tab Semua Skrip. Kita akan menggunakanannya untuk menjumlah variabel “first\_number” dengan “second\_number”.

Dari Variabel, tarik variabel “first\_number” dan masukkan ke dalam kotak kosong di sebelah kiri pada operator “tambah”. Letakkan “second\_number” pada kotak kosong kedua (lihat rekaman gambar dibawah).



Catatan:

Kita belum selesai. Kita telah memberitahukan program untuk menambahkan angka-angka kita, tetapi kita harus menyimpan hasilnya ke salah satu variabel kita.

Dari Variabel pada Kotak Palet, tarik “set [first\_number] sebesar [0]” ke tab Semua Skrip. Ganti “first\_number” dengan “result” dan tarik operator “add” tadi ke dalam angka “0”. Balokmu harusnya terlihat seperti ini:



Terakhir, kita juga harus memberitahukan Scratch untuk menampilkan hasil perhitungan. Apabila tidak, dia hanya akan menyimpannya untuk dirinya sendiri (dan itu tidak baik untuk kita).

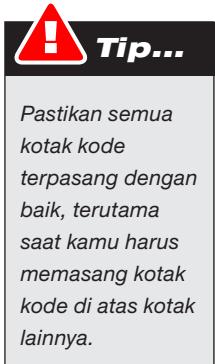
Klik Tampilan pada Kotak Palet. Pilih “berkata [Assalamualaikum!]” dan tarik kedalam tab Semua Skrip. Dari Variabel, tarik variable “result” dan masukkan ke dalam “Assalamualaikum!”. Gambar dibawah akan menunjukkan yang kami



maksud.

Dan selesailah sudah! Kami sudah memasukkan semua kode dibawah. Kami sudah menambahkan beberapa balok kode lagi, untuk membuat si kucing sedikit

*Gunakan rekaman gambar ini untuk membuat keseluruhan skrip. Dan ingat, setiap kali kamu ingin si kucing melakukan perhitungan kamu harus mengklik bendera hijau.*



cerewet tetapi pada dasarnya sama dengan skrip yang kita buat diatas.

Tapi tunggu sebentar: Jika kamu ingin mencoba membuat si kucing melakukan semua pekerjaan rumah kamu akan menyadari ada satu masalah besar.

Si kucing hanya bisa mengerjakan penjumlahan! Jangan panik – dengan melakukan sedikit perubahan akan menyelesaikan masalah kita.

Catatan:

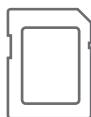
Klik Operator pada Kotak Palet sekali lagi dan kamu akan melihat ada balok operator yang lain juga: untuk pengurangan, perkalian dan pembagian.



Jika kamu ingin si kucing melakukan pengurangan cukup tukar kotak operator “penjumlahan” pada skrip diatas dengan kotak operator “pengurangan”, dan seterusnya sampai si kucing selesai mengerjakan semua pekerjaan rumahmu.



## Giliranmu



**TUGAS:** Program si kucing matematika untuk menyelesaikan semua tugas pada lembar pekerjaan rumah – kamu harus mengganti operator setiap kali si kucing melakukan perhitungan yang diinginkan.

Untuk melihat contoh yang sudah benar, buka RPiScratch/Projects/kucing\_matematika

**TUJUAN BELAJAR:** Pada latihan ini, kamu akan belajar bagaimana memasukkan data dan membuat program-mu mengambil keputusan berdasarkan data tsb.

**SUMBER:** Sprite “cat” dan “bluedog”; latar belakang “brick-wall1”, “sydney”, “paris” dan “new\_york”.

Kecerdasan buatan adalah sebuah bidang ilmu pengetahuan komputer dimana orang mencoba membuat program komputer yang memiliki kecerdasan. Ideya adalah membuat komputer terlihat berpikir seperti manusia. Ini sebenarnya cukup sulit, seperti yang kamu bayangkan, jadi disini kami hanya memberikan contoh kecil tentang bagaimana kamu dapat membuat program-mu sedikit cerdas.

Si kucing tentu berpikir dia cerdas, jadi mari kita beri kesempatan padanya untuk menunjukkan betapa cerdasnya dia. Untuk itu, kita akan menggunakan beberapa lagi masukan dan keluaran, bersama dengan sesuatu yang disebut “pernyataan bersyarat”. Kedengarannya rumit sekali, tetapi sebenarnya tidak.

Sebagai contoh, Aku telah membuat dua sprite: si kucing dan si anjing. Si kucing akan menanyakan beberapa pertanyaan kepada si anjing, jadi kita membutuhkan beberapa variabel untuk menyimpan jawabannya.

Buat variabel berikut ini (kami akan menjelaskannya sebentar lagi):

**age**  
**country**  
**holiday**  
**name**

Sebelum kita mulai, kamu harus mengimpor sprite “bluedog” dan latar belakang “brick-wall1”, “sydney”, “paris” dan “new\_york”. Kamu akan membutuhkannya nanti.

### Apakah si anjing ingin berlibur?



Pertama, kita harus mencari tahu apakah si anjing ingin berlibur. Pilih sprite kucing dan buat skrip seperti rekaman gambar diatas untuk membantu si kucing mencari tahu.

Setiap kali kita menggunakan program ini variabel country akan di set dengan sebuah huruf. Tetapi ketika kita menggunakan program berikutnya, kita ingin variabel country isinya kosong. Jadi kita “kosongkan” dengan mengesetnya dengan nilai “0”, siap untuk pemain berikutnya.

**PERTANYAAN:** Pelajari skripnya. Dapatkah kamu mengetahui apa yang akan dilakukan pada saat dijalankan?

Pada saat si kucing bertanya, jawaban si anjing akan disimpan ke variabel “holiday” dan “age”. Skrip akan menggunakan kedua variabel ini nanti.

Sekarang pada bagian yang pintar: pernyataan bersyarat. Kita akan mencari tahu apakah si anjing ingin berlibur. Saatnya untuk berpikir logika: si anjing akan menjawab “yes” atau “no”.

Kita harus mencari cara untuk memberi tahu si kucing apa yang diinginkan si anjing dan menginstruksikan si kucing untuk bereaksi terhadap jawaban tersebut.



Dari Kontrol pada Palet Kode, pilih balok “jika”.

Kamu juga butuh balok operator pada Palet Kode Operator untuk mengetes masukan. Tarik oprator “sama dengan” “[ ] = [ ]” dari Palet Kode.

Dari Variablel pada Palet Kode, tarik dan letakkan variabel “holiday” ke bagian kiri dari operator dan ketikkan “no” pada bagian kanan.

Kemudian, kita tarik dan letakkan ke dalam balok “jika”.

Sekarang, dari Tampilan letakkan balok “berkata [Assalamualaikum!] selama [2] detik” ke dalam balok “jika” – sehingga balok “jika” melingkupinya. Ganti “Assalamualaikum!” dengan “OK, no holiday for you!”.

*Kode untuk  
mengetahui  
apakah si  
anjing ingin  
berlibur*



Tapi tunggu sebentar, bagaimana jika si anjing ingin berlibur? Balok “jika” kita tidak memperbolehkan untuk memasukkan kemungkinan jawaban “yes”.

Jika si anjing ingin berlibur sesuatu yang lain harus terjadi. Kita harus mengganti balok “jika” dengan balok “jika/selain itu”.

*Kotak kode  
“jika/selain itu”*



*Catatan:*

Ambil skrip kecil yang sebelumnya berada didalam balok “if” pindahkan ke dalam bagian “jika” dari balok “jika/selain itu”. Hasilnya harusnya seperti rekaman berikut.



### **Apakah si anjing sudah cukup umurnya untuk berlibur?**

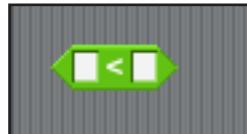
Kita juga harus mencari tahu apakah si anjing sudah cukup umurnya untuk berlibur sendiri. Kita akan membutuhkan balok “jika/selain itu” kedua. Tarik ke dalam tab Semua Skrip, tapi jangan pasangkan ke skripmu dulu – biarkan dia berdiri sendiri.

Kita akan menanyakan si anjing umurnya berapa saat ini. Tergantung dari jawabannya, ada dua kemungkinan hasil:

- 1.** Umurnya kurang dari 10 tahun, dan dia tidak boleh berlibur sendirian.
- 2.** Dia berumur diatas 10 tahun, dan dia boleh pergi berlibur.

Jika umur si anjing kurang dari 10 tahun, kita ingin si kucing berkata, “Sorry dog, you’re too young to go on holiday.”

*Operator “lebih kecil dari”.*



Gunakan operator “lebih kecil dari” (<) , juga balok pada Variabel dan Tampilan, buat skrip yang dapat kamu lihat dari rekaman gambar di halaman berikutnya.

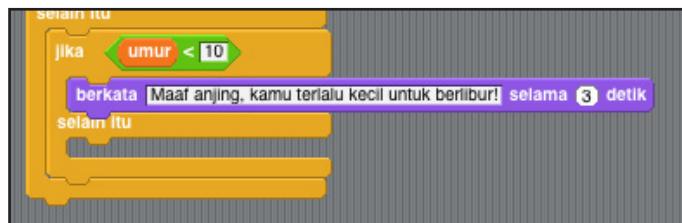
Letakkan skrip ini didalam “jika” dari balok “jika/selain itu” yang kedua.

Sebelum kita lanjut ke langkah berikutnya, ambil balok “jika/selain itu” yang kedua

Catatan:



dan letakkan di dalam "selain itu" dari balok "jika/selain itu" yang pertama: yang digunakan untuk mengetahui apakah si anjing ingin berlibur.  
Skripmu saat ini seharusnya terlihat seperti diatas.



## Giliranmu

**PERTANYAAN:** Apa yang akan dikatakan si kucing jika si anjing berumur 9?

**PERTANYAAN:** Apa yang akan dikatakan si kucing jika si anjing berumur 12?

**PERTANYAAN:** Apa yang akan dikatakan si kucing jika si anjing berumur 10?

## Si anjing ingin berlibur kemana?

Supaya adil dengan si anjing, kita harus memberikan pilihan kemana dia ingin berlibur. Kita akan memberikan tiga kemungkinan tujuan liburan: Australia, Perancis dan Amerika. Supaya tetap sederhana, kita tidak meminta si anjing untuk memasukkan negaranya sendiri, cuma huruf pertamanya saja.

Gunakan informasi dari rekaman gambar dibawah, tambahkan balok untuk menanyakan si anjing ingin berlibur kemana. Simpan jawabannya ke variabel "country".

Kamu akan butuh balok dari Tampilan, Sensor dan Variabel. Letakkan didalam "selain itu" dari balok "jika/selain itu" yang kedua.

Terakhir, lengkapi skripmu dengan balok "tunggu selama [ ] detik" dan "hentikan



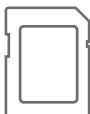
semua skrip” dari Kontrol. Set nilai “tunggu selama [ ] detik” menjadi “4” dan letakkan kedua balok di paling akhir, setelah kedua “jika/selain itu”.

Catatan:



Tujuan yang dipilih si anjing akan disimpan ke variabel “country”. Ini penting, karena variabel ini akan digunakan oleh program untuk menentukan keluaran dari skrip.

## Skrip untuk panggung



Untuk melengkapi program kita membutuhkan latar belakang. Jika kamu belum mengimpor dari RPiScratch\Resources\Backgrounds, sekarang saatnya.



Pilih panggung dari Daftar Sprite dan buat skrip yang dapat kamu lihat diatas.

Disini, kita telah membuat tiga “jika berkait” pernyataan bersyarat, yang akan memilih latar belakang berdasarkan pilihan si anjing. Kami menggunakan istilah “**berkait**” jika satu pernyataan bersyarat diletakkan didalam pernyataan bersyarat lainnya.

Setiap kali kita menggunakan program, panggung harus diset dengan latar belakang “brick-wall1”. Pernyataan bersyarat harus terus mengecek sampai si anjing menjawab, jadi aku meletakkannya didalam “jalankan tanpa henti”.



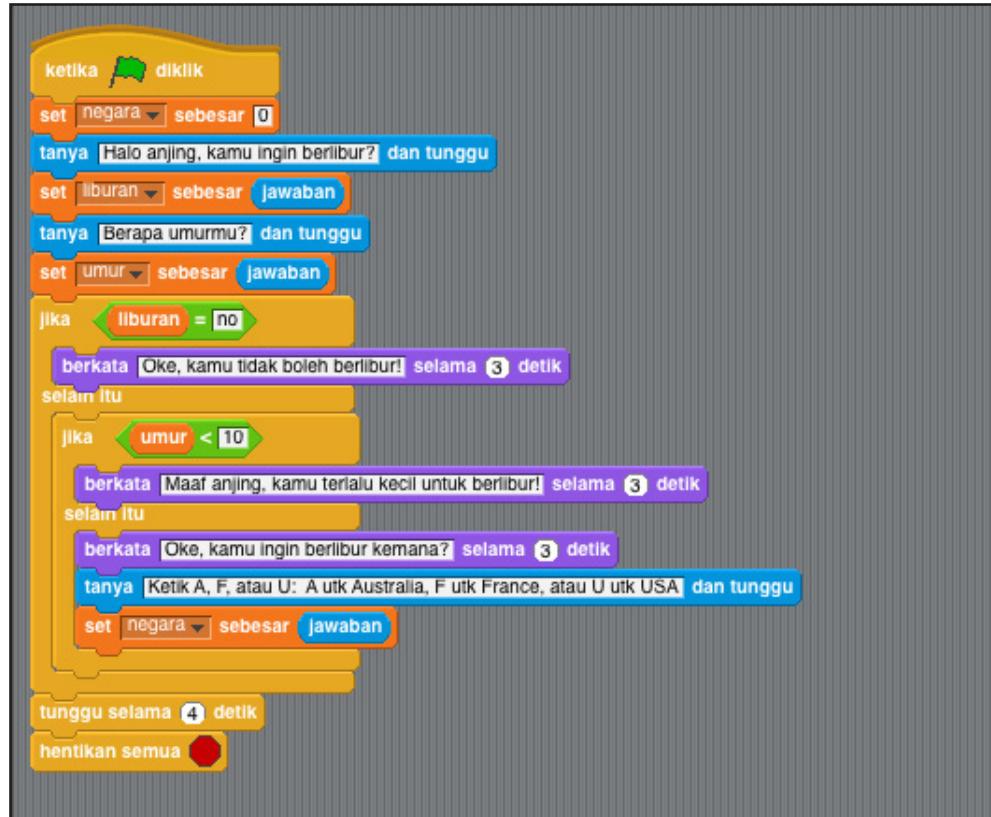
**Tip...**

Users do not always do what they are supposed to do. If a user provides an unexpected input, this can cause an “error” in the program, which will often cause the program to “crash”. Programs that have errors in them are said to contain “bugs”. An important task for a programmer is testing their program to make sure it is bug-free.

## Skrip Lengkap

Fuihh, tadi itu rumit sekali. Tetapi mudah-mudahan kamu tidak mengalami kesulitan. Supaya lebih mudah, berikut adalah skrip lengkap dari si kucing dan panggung.

### Kode untuk si kucing

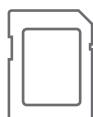


### Kode untuk panggung



Catatan:

Untuk melihat sampel dari pelajaran ini, buka  
RPiScratch/Projects/anjing\_berlibur.



**PERTANYAAN:** Pelajari kode untuk si kucing dan panggung diatas. Apakah kamu mengerti?

**TUGAS:** Ubah-ubah kodennya. Apakah kamu bisa tambah pertanyaan lagi.

## Akhir kata

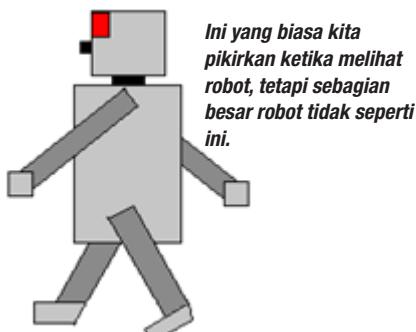
Pada kenyataannya, si kucing tidak cerdas sama sekali. Komputer tidak memiliki kecerdasan – program yang berjalan didalamnya yang membuatnya terlihat cerdas. Itu sebabnya kita menyebutnya dengan “kecerdasan buatan”. Satu-satunya yang cerdas adalah si programer, yang memprogram si kucing, itu adalah kamu, untuk menyelesaikan pelajaran 1.5.

## Pelajaran 1.6: Kontrol

**TUJUAN BELAJAR:** Pada latihan ini, kamu akan belajar bagaimana membuat program kontrol yang memberi respon berbeda-beda tergantung dari masukan.

**SUMBER:** Sprite “robot\_atas” dan latar belakang “latarbelakang\_hijau”.

## Si Robot

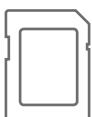


Kamu mungkin sudah terbiasa dengan robot di film: manusia besi yang berdenting dan bersuara metal. Ini sebenarnya ide kuno tentang sebuah robot. Ini berasal dari cerita di tahun 1920. Aku yakin bahkan gurumu masih kecil saat itu.

Pemikiran yang lebih modern tentang robot adalah segala sesuatu yang dapat dikontrol dengan komputer. Ini dikenal dengan “teknik kontrol”. Alat ini bisa berupa pesawat, mesin cucui, mesin bubut, mesin welding, a level-crossing barrier, mesin jahit, mobil yang bisa menyetir sendiri atau apapun yang dapat kamu bayangkan.

Tapi mari mulai dengan yang biasa, robot jadul. Kita tidak punya robot yang asli, atau semua kabel dan sirkuit yang dibutuhkan untuk mengontrol. Jadi, kita akan menggunakan robot yang hidup di panggung Scratch. Apa yang akan kamu dapat dari melakukan ini adalah bagaimana membuat robotmu merespon masukan.

Kita akan memprogram robot kita sehingga kita dapat mengontrolnya dengan tombol “panah keatas”. Supaya mudah melihat robot kita bergerak, kita harus dapat melihatnya dari atas.



Impor sprite “robot\_atas” dari RPiScratch/Resources/Costumes.

Impor juga background “latarbelakang\_hijau” dari RPiScratch/Resources/Backgrounds.

*Skrip ini akan membuat robot kita bergerak sesuai perintah kita, tetapi hanya satu arah.*



*Catatan:*

Perhatikan rekaman gambar diatas. Kita menggunakan event “ketika tombol [ruang] ditekan” dari Kontrol untuk menjalankan program. Sprite robot bergerak ke tengah panggung dengan balok “pindah ke x: [0] y: [0]” dari Gerakan (lihat “Menggerakkan sprite”, pada Pelajaran 1.2).

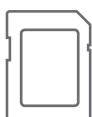
Keseluruhan kode diletakkan di dalam perulangan “ulangi sampai” (dari Kontrol). Didalam perulangan, terdapat “jika” (pernyataan bersyarat) yang mengecek jika tombol “panah keatas” ditekan. Ini telah diset dengan event “apakah tombol [panah keatas] ditekan?” (dari Sensor).

Balok “apakah menyentuh [ ]?” dari Sensor di set dengan “batas”. Itu artinya jika si robot menyentuh batas panggung, event tersebut akan menghentikan skrip. Sebuah “event” adalah sesuatu yang terjadi didalam program.

**Pada skrip ini, ada dua kemungkinan yang terjadi:**

1. Tombol “panah keatas” ditekan – pada kasus ini, si robot akan berjalan keatas sebanyak 20 langkah, berganti-ganti kostum saat itu terjadi.
2. Tombol “panah keatas” belum ditekan – pada kasus ini, si robot tidak akan melakukan apa-apa.

Kelihatannya tidak terlalu berguna. Mari kita lihat apakah bisa dibuat lebih bagus.



*Untuk melihat contoh skrip ini, buka RPiScratch/Projects/robot\_v1.*



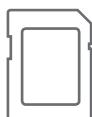
**SUMBER:** Sprite “robot\_control” dan latar belakang “robot\_maze”.

Kami telah memasukkan kode yang lebih lengkap, “Robot\_v2” (lihat rekaman gambar diatas). Pada versi ini, robot dapat bergerak kekanan, kekiri, keatas dan kebawah.

**TUGAS:** Sekarang giliranmu untuk membuat program. Buatlah program untuk membuat si robot mengikuti jejak kuning pada latar belakang “robot\_maze”.

**TUGAS:** Buka kodenya dan ubah agar si robot bergerak lebih cepat.

Untuk melihat contoh skrip ini, buka RPiScratch/Projects/robot\_v2.



**Tip...**

Ini adalah kode yang berguna untuk mengontrol karakter di dalam permainan.

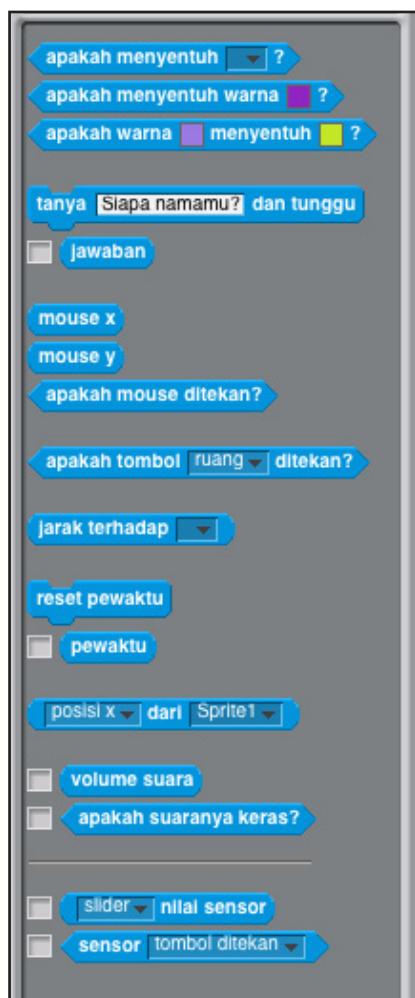
## Kendaraan yang mengikuti garis (LFV)

Catatan:

**SUMBER:** Sprite “yellow\_car” dan latar belakang “line\_background”.

Robot itu seperti berasal dari film fiksi ilmiah kuno dan membosankan. Mari kita lihat contoh yang lebih moderen. Ilmuwan saat ini sedang berkesperimen dengan kendaraan yang dapat menyetir sendiri. Salah satu caranya dengan memprogram kendaraan untuk mengikuti garis yang dicat di permukaan jalan. Kendaraan ini dikenal sebagai “line-following vehicles” (kendaraan yang mengikuti garis), atau disingkat LFV.

Pada latihan ini, kita akan melihat lebih dalam balok-balok kode pada kategori Sensor.



*Balok kode  
Sensor mampu  
membuat  
program-mu  
untuk merasakan  
perubahan di  
lingkungannya,  
baik itu memberi  
respon terhadap  
masukan dari  
emain atau  
event di dalam  
program.*

Tugas kita adalah membuat sebuah mobil yang mengikuti garis, persis seperti yang dilakukan insinyur sungguhan merancang LFVs untuk digunakan di jalan. Alat utama kita adalah balok kode “apakah warna [ ] menyentuh [ ]?” dari Sensor.



Balok kode ini mampu membuat program mendeteksi kapan sebuah sprite dengan warna tertentu menyentuh warna yang lain: contoh, seperti pada rekaman gambar diatas, warna merah menyentuh warna hitam.

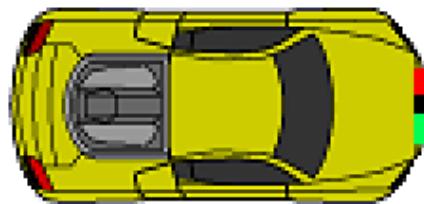
Notes:

Saat program mendeteksi dua warna saling bersentuhan, kita dapat memberitahukan bagaimana cara merespon kejadian tersebut. Event warna menjadi pemicu, membuat program melakukan sesuatu.

Untuk mengeset warna di balokmu, klik salah satu kotak kecil berwarna. Kursor tetikusmu akan berubah menjadi “tetes-mata”. Gunakan tetes-mata untuk mengklik warna yang ingin kamu gunakan didalam balok.



Tetapi bagaimana kita dapat menggunakan kemampuan mendeteksi warna untuk membantu kendaraan LFV kita mengikuti garis di jalan? Ya, mari kita mulai mengimpor “*line\_background*” dan sprite “*yellow\_car*”.



Perhatikan mobil baik-baik, kamu akan melihat kotak berwarna hijau di bemper mobil. Kita akan menggunakan kotak hijau tersebut sebagai “sensor”. Buatlah skrip yang terlihat pada rekaman gambar berikut.



Sekarang, klik bendera warna hijau dan perhatikan apa yang terjadi. Wow, semuanya berjalan dengan baik selama garisnya melengkung ke kiri tetapi begitu garisnya melengkung ke kanan mobilnya salah arah. Apa yang terjadi?

Masalahnya mobil butuh dua buah sensor. Saat ini, jika garis bengkok ke kanan, garis hitam akan menyentuh kotak warna hijau – program akan mendeteksi ini dan memberitahu mobil untuk belok ke kanan.



If you can't stop your car for long enough to sample a colour, hit the red “stop scripts” button at the top-right of the screen.

Sekarang, kita butuh kotak warna merah pada bemper mobil sebagai sensor merah. Sehingga program bisa mendeteksi ketika garis belok kekiri dan memberitahu mobil untuk belok ke kiri juga.

Berikut adalah skrip untuk melengkapi LFV program-mu:



Seperti yang kamu lihat, program akan berjalan dengan event bendera hijau. Setelah itu:

- Balok kode “go to” digunakan untuk meletakkan mobil pada garis.
- Kita harus menggunakan balok “jalankan tanpa henti” untuk membuat mobil terus berjalan.
- Mobil bergerak 0.5 langkah (pada setiap siklus perulangan).
- Balok “jika” digunakan untuk mengecek apakah kotak bemper hijau menyentuh garis hitam. Jika menyentuh, mobil akan berbelok satu derajat ke kanan.
- Balok “jika” kedua digunakan untuk mengecek apakah kotak bemper warna merah menyentuh garis hitam. Jika menyentuh, mobil akan berbelok satu derajat ke kiri.



Untuk contoh versi lengkap dari projek ini, buka  
RPiScratch/Projects/lfv.

**Tip...**

Jika mobilmu berhenti pada sudut yang ganjil, dan kamu tidak dapat meluruskannya, kamu dapat mengimpor sprite “yellow\_car” kembali, tarik dan letakkan skrip lengkapmu ke sprite baru kemudian hapus yang lama

**Tip...**

Jalankan program pada mode layar penuh – akan terlihat sangat mengesankan. Klik tombol di pojok kanan atas jendela Scratch, tepat diatas bendera hijau dan lingkaran merah.

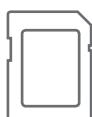
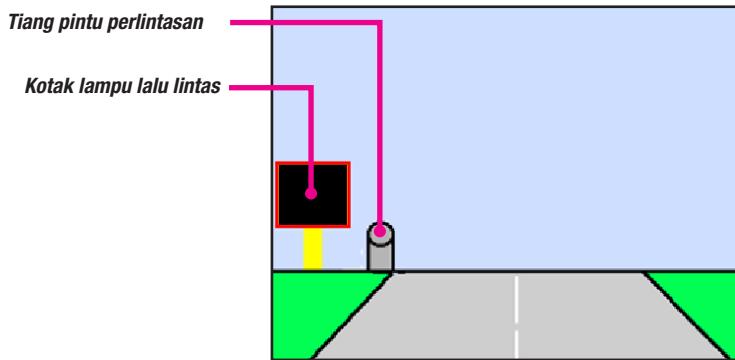
## Pintu Perlintasan Kereta Api

**SUMBER:** Sprite “lampa” dan “palang”, latar belakang “perlintasan\_keretaapi” dan suara “perlintasan\_keretaapi\_a”.

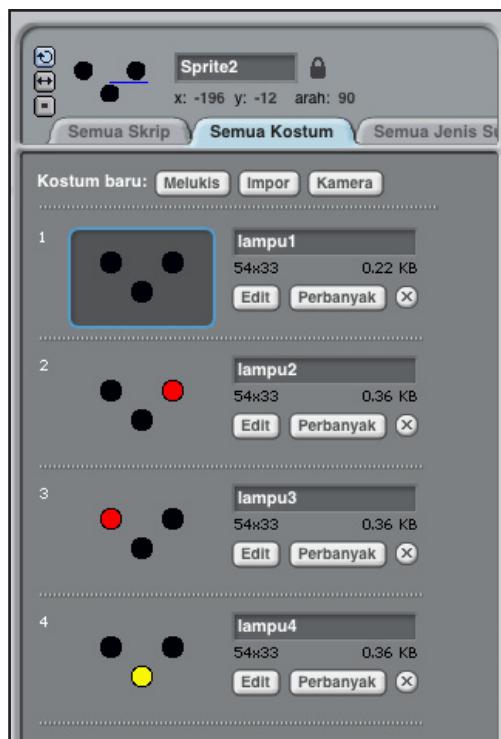
Sistem kontrol komputer dapat mengamankan kita. Pada pintu perlintasan kereta api kita harus memastikan mobil dan kereta api tidak saling bertabrakan. Ini dilakukan dengan memasang sensor pada rel kereta yang akan mendeteksi kereta api yang melintas.

Komputer kemudian merespon kejadian (event) ini dengan memberhentikan lalu lintas yang mendekati pintu perlintasan dan menutup pintu. Program kontrol ini dapat dibuat dengan Scratch. Tetapi sebelum kita mulai, kita harus mengimpor sprite dan latar belakang yang dibutuhkan.

Pertama, impor latar belakang “*perlintasan\_keretaapi*”. Seperti yang kamu lihat pada gambar dibawah, latar belakang memperlihatkan jalan menuju ke perlintasan, kotak lampu lalu lintas dan tiang pintu perlintasan.



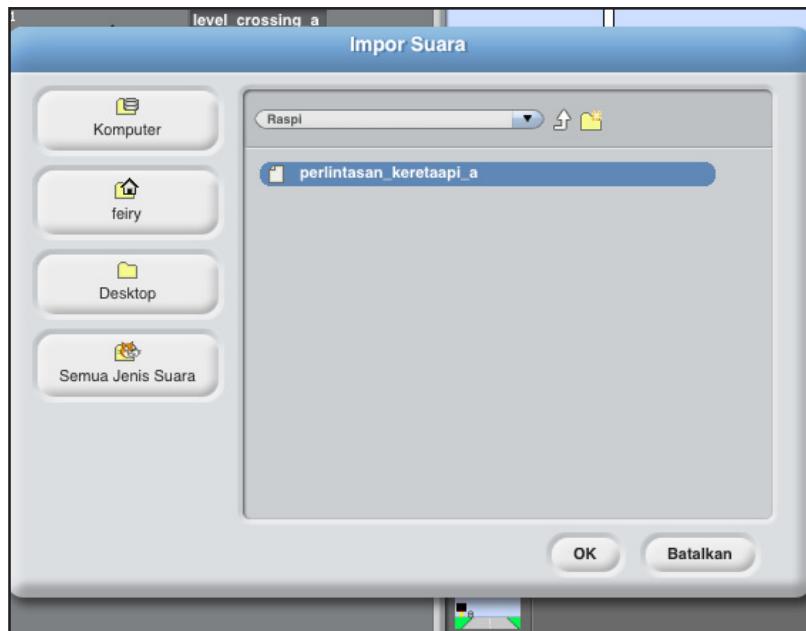
Sekarang kita butuh sprite kita. Impor “*lampu*” dan “*palang*” dari folder RPiScratch/Resources/Costumes.



Perhatikan gambar diatas, kamu akan melihat sprite “*lampu*” memiliki empat kostum yang berbeda. Dengan mengganti kostumnya dengan cepat, kita dapat membuat lampunya terlihat seperti berkedip-kedip.

Yang terakhir yang kita butuhkan sebelum memulai adalah mengimpor suara alarm pintu perlintasan kereta api. Untuk itu, klik tab Semua Jenis Suara, cari suara “*perlintasan\_keretaapi\_a*” kemudian gunakan fungsi impor, sama seperti yang kamu lakukan pada latar belakang.

**Impor efek suaramu dari tab Semua Jenis Suara. Kamu dapat temukan di sebelah kanan dari Semua Skrip dan Semua Kostum.**



Catatan:

Sekarang, mari buat skripnya. Sebelum memulai, pilih sprite “lights” pada Daftar Sprite.

Pertama-tama: kita ingin memulai dalam keadaan lampu padam. Jadi kita gunakan event bendera hijau untuk mengubah kostum ke “costume1”, yang menampilkan tiga lampu hitam.



Pada saat kereta mendekati pintu perlintasan, aku menggunakan balok “ketika tombol [ ] ditekan” dari Kontrol untuk mendeteksinya, ubah nilai menjadi “t”. Event ini akan memicu lampu lalu lintas, alarm dan pintu perlintasan. Tetapi pertama-tama, mari kita mengerjakan alarm terlebih dahulu.

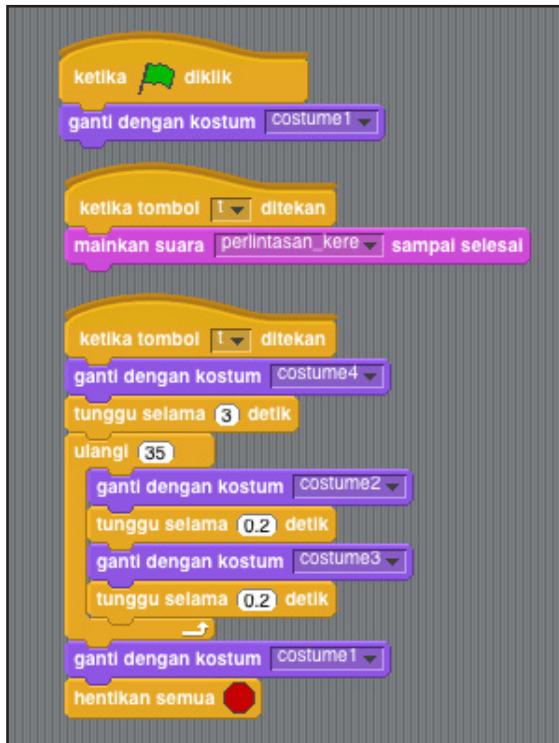


Buat skrip seperti yang kamu lihat pada rekaman gambar diatas. Ini akan membuat alarm berbunyi apabila tombol “t” ditekan.

Catatan:



Untuk membuat lampu merah berkedip-kedip kita harus mengganti-ganti dua kostum lampu merah berulang-ulang. Kita menggunakan balok “ulangi” untuk kebutuhan ini. Buat skrip seperti terlihat pada gambar diatas. Kamu harus bereksperimen dengan angka perulangan sampai kamu dapat waktu yang pas.



Gambar diatas adalah program lengkap dari lampu lalu lintas, dengan semua balok terpasang pada tempatnya. Perhatikan kamu memiliki tiga skrip yang terpisah, tidak saling bersentuhan.

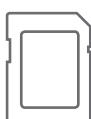
Sekarang untuk pintu perlintasan: tarik dan pindahkan sampai sampai pintunya berada tepat diatas tiang pintu perlintasan, dengan bagian yang tebal menempel tepat diatas gardu. Setiap kali program dimulai, pintu perlintasan harus menghadap keatas. Gunakan balok “menghadap kearah [90]” (dari Gerakan) untuk melakukannya.



Untuk skrip pintu perlintasan, Aku menggunakan balok “ketika tombol [] ditekan” dari Kontrol, nilainya diset dengan “t”. Ingat kembali – skrip lampu juga dimulai dengan tekan tombol “t”. Dengan cara ini, lampu lalu lintas dan pintu perlintasan akan tersinkronisasi: artinya keduanya akan mulai pada saat yang bersamaan.

Untuk menurunkan pintu perlintasan, kita akan memutar pintu secara bertahap dengan menggunakan balok “ulangi”. Eksperimenlah dengan jumlah perulangan dan sudut putar untuk menggerakan pintu perlintasan dengan tepat.

Setelah waktu tertentu, yaitu 13 detik, kita menaikkan kembali pintu perlintasan dengan menggunakan balok perulangan. Kereta telah melintas, semua mobil telah aman. Selamat!



*Untuk contoh lengkap projek ini, buka RPiScratch/Projects/perlintasan\_keretaapi.*

## Giliranmu

**TUGAS:** Sebagai fitur tambahan dari program ini, buat sprite kereta api yang melintasi layar ketika pintu perlintasan turun.

## Pelajaran 1.7: Permainan Scratch

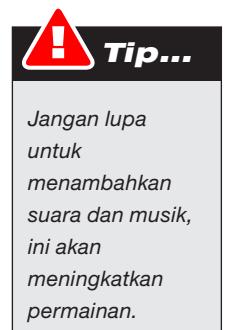
**TUJUAN BELAJAR:** Pada latihan ini, kamu akan belajar beberapa teknik yang digunakan pada pembuatan program permainan (game) dengan memainkan beberapa permainan Scratch.

Beberapa hal yang paling menyenangkan yang bisa kamu dapat dari Scratch adalah melalui pembuatan program permainan (game). Kamu dapat membuat permainan apapun yang kamu suka. Kami sudah memasukkan dua contoh game untuk kamu pelajari, dan ada banyak lagi permainan dan contoh yang lain pada folder Scratch Projek.

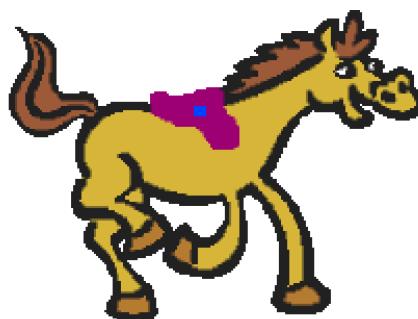
Titik awal dari setiap permainan baru adalah tema atau ide, dan tujuan dari permainan. Ketika kamu memiliki sebuah ide permainan, pikirkan mengenai “cerita” yang ingin kamu sampaikan dan karakter-karakter permainan, pemain, benda-benda dll. itu yang kamu butuhkan. Ini akan menjadi sprite-spritemu.

Kamu dapat menemukan gambar-gambar bagus untuk sprite dan latar belakang dengan mengimpor dari folder Scratch Media atau dengan mencari gambar dari Google, simpan dan masukkan ke dalam Scratch. Jika kamu ingin mendapatkan ide-ide bagaimana cara memulai dan apa yang dapat kamu lakukan, perhatikan permainan-permainan buatan orang lain. Cara terbaik untuk belajar bagaimana membuat program adalah dengan memeriksa skrip orang lain untuk melihat bagaimana mereka membuat hal yang keren tersebut dilakukan.

Terakhir, kamu akan membutuhkan latar belakang untuk panggungmu untuk membuat ruang, level dan adegan pada permainanmu, supaya karakter-karakter ciptaanmu punya tempat untuk bergerak.



### Permainan Kuda Poni



**SUMBER:** Sprite: “horse1-b” dan “girl4-sitting” dan latar belakang “lapangan”.

**TUJUAN PERMAINAN:** Seekor kuda poni bergerak di sebuah lapangan, bergerak ke segala arah secara acak. Dengan menggunakan tetikus, pemain mengontrol anak perempuan. Untuk mendapat nilai, kamu harus meletakkan si anak perempuan pada kuda poni.

Ada empat tugas yang harus dilakukan:

- 1.** Menggerakan kuda poni.
- 2.** Membuat perubahan acak pada arah gerakan kuda poni.
- 3.** Menggerakan si anak perempuan dengan tetikus.
- 4.** Menambahkan score setiap kali si anak perempuan berhasil diletakkan diatas kuda poni.

Ayo kita mulai!

## Menggerakan kuda poni

### 1. Menggerakan kuda poni



Pilih sprite kuda poni dan tarik event bendera hijau ke dalam tab Semua Skrip. Kita akan menggunakan event bendera hijau untuk menjalankan semua skrip yang ada pada permainan ini secara bersamaan.

Kita telah menggunakan balok “jalankan tanpa henti” untuk membuat kuda poni kita bergerak terus menerus (ingat kendaraan LFV). Untuk setiap siklus perulangan, si kuda poni bergerak tiga langkah. Jika dia menyentuh batas dia akan memantul ke arah yang berlawanan; ini untuk mencegah si kuda poni berhenti.

### 2. Random changes in direction



Jika si kuda poni tidak pernah berubah arah kecuali ketika dia menyentuh batas, permainan akan terlihat membosankan. Jadi kita menggunakan balok “jalankan tanpa henti” kedua untuk membuat si kuda poni merubah arahnya secara acak dari waktu ke waktu.

Balok “ambil nilai acak antara [ ] hingga [ ]” dari Operator telah di set untuk mengambil nilai acak dari 1 hingga 4. Jika angka yang terambil adalah 1, si kuda poni akan berputar 180 derajat.

Ini diset untuk dilakukan setiap 0.5 detik dengan menggunakan balok “tunggu”.

Catatan:

### 3. Menggerakan si anak perempuan dengan tetikus



Pilih sprite anak perempuan dari Daftar Sprite. Dengan menggunakan balok “jalankan tanpa henti”, dan juga balok-balok pada Gerakan dan Sensor:

- Set koordinat x dari sprite anak perempuan ke posisi x dari kursor tetikus.
- Set koordinat y dari sprite anak perempuan ke posisi y dari kursor tetikus.

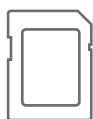
### 4. Tambahkan score setiap kali anak perempuan berhasil diletakkan diatas kuda poni.



Pada skrip ini, kamu dapat melihat beberapa hal berikut:

- Kita telah membuat variabel dan memberi nama dengan “score” (bayangkan!).
- Ketika permainan dimulai score di set dengan nilai “0”.
- Perulangan “jalankan tanpa henti jika” mengecek apakah warna merah dari baju anak perempuan menyentuh titik kecil warna biru yang ada di pelana kuda poni.
- Setiap kali warna merah menyentuh warna biru score diset dengan “score + 1”. Tunggu sebentar! Setiap kali warna merah menyentuh titik biru, “score” berubah menjadi “score + 1”. Jika kamu pintar matematika, kamu pasti bertanya-tanya! Bagaimana mungkin sesuatu ditambah dengan dirinya sendiri tambah satu?

Jangan khawatir; programer sering melakukan ini. Ingat variabel bukanlah sebuah angka; variabel adalah sebuah tempat yang dapat menyimpan angka. Kita dapat melakukan perhitungan terhadap angka yang disimpannya dan menyimpan kembali hasil perhitungan ke dalam tempat yang sama.



Untuk melihat contoh lengkap dari permainan kuda poni, buka RPiScratch/Projects/kuda\_poni..

**Tip...**

Ingat, hentikan semua gerakan supaya kamu dapat mengambil sampel warna baju atasan si anak perempuan dengan alat tetes mata, klik tombol merah “hentikan semua skrip”. Jika kamu tidak dapat menemukan si anak perempuan, cari di pojok kiri bawah layar. Mungkin dia bersembunyi disana ketika kamu menarik tetikusmu ke dalam tab Semua Skrip.

**SUMBER:** Sprite “mobil\_merah”, “mobil\_ungu” dan latar belakang “lintasan”.

**TUJUAN PERMAINAN:** Permainan dengan dua pemain. Setiap pemain mengontrol sebuah mobil balap mengelilingi lintasan. Jika mobil keluar dari lintasan, mobilnya akan hancur. Yang pertama sampai di garis finish adalah pemenangnya.

Ada lima tugas untuk mem-program setiap mobil:

1. Membariskan mobil-mobil pada garis start.
2. Menggerakkan mobil-mobil.
3. Hancurkan mobil jika keluar dari lintasan.
4. Memberi kemenangan pada mobil pertama yang mencapai garis finish.
5. Oh ya – pemain harus dapat mengontrol mobil mereka!

Dimulai dengan event bendera hijau. Ketika bendera diklik, mobil harus:

- diganti dengan “costume1”
- dipindahkan ke posisi koordinat x: 35 y: 130
- dihadapkan ke arah kanan (arah 90 derajat)

Untuk melakukan semua itu, kamu akan membutuhkan, bersama dengan ketika bendera hijau diklik, balok-balok dari Tampilan dan Gerakan. Untuk menggerakkan mobil, kita menggunakan teman lama kita yaitu balok “jalankan tanpa henti”. Mobil bergerak satu langkah setiap siklus.

Jika mobil keluar lintasan, dia akan menyentuh rumput yang berwarna hijau.

### 1. Menggerakkan mobil merah



Kejadian (event) ini akan membuat balok “jika” merusak mobil dan menghentikan skrip dari mobil tersebut.

Pada saat mobil melewati garis finish, dia akan menyentuh garis putih. Kejadian (event) ini akan menyebabkan balok “jika” kedua berkata “I win!” dan menghentikan semua skrip yang menjalankan kedua mobil.

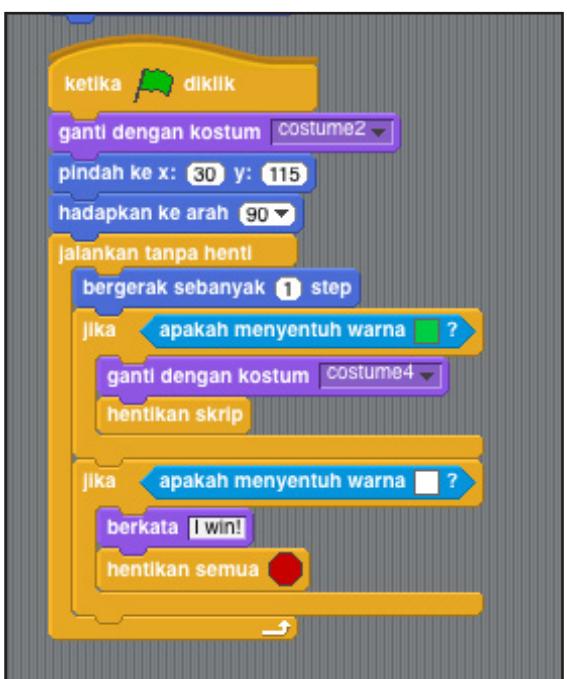
**2.  
Mengendalikan  
mobil merah**



Oke, jadi kita dapat menggerakkan mobil merah. Tetapi kita harus bisa mengendalikan mobil tersebut. Untuk mengendalikan mobil kita membutuhkan dua balok “ketika tombol [ ] ditekan” (dari Kontrol).

Seperti yang kamu duga, Tombol “panah kanan” untuk membelokkan mobil ke kanan dan tombol “paha kiri” untuk menggerakkan mobil ke kiri.

**3. Lakukan cara  
yang sama untuk  
mobil ungu**



Kode yang serupa digunakan untuk menggerakkan mobil ungu. Cara termudah melakukannya adalah melengkapi semua kode mobil merah kemudian disalin jadi dua. Jangan lupa untuk mengganti mobil salinan kedua dengan kostum warna ungu.

Perubahan yang harus kamu lakukan terhadap mobil ungu adalah:

Mobil ungu harus ditaruh disebelah mobil merah pada posisi x: 30 y: 115.



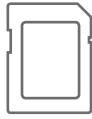
Kamu dapat menggunakan tombol “x” untuk mengerakkan mobil ke kanan dan tombol “z” untuk mengerakkan mobil ke kiri.

Catatan:



**Tip...**

Ketika kamu membuat permainan, jangan terlalu berambisi di saat awal. Programlah ide awalmu dan pastikan itu bekerja dengan baik. Kemudian kamu kembangkan terus sedikit demi sedikit, tambah karakter baru, event baru dan level baru. Sebaiknya kamu simpan setiap versi baru dengan nama yang baru, sehingga kamu dapat menelusuri perubahan yang kamu lakukan. Kamu bisa saja mengalami masalah dan ingin kembali ke versi yang lama dari permainanmu.



Untuk melihat contoh lengkap dari permainan ini, buka  
[RPiScratch/Projects/game\\_balap\\_v1](#).

Catatan:

## Giliranmu

**TUGAS:** Tambahkan suara mobil balapan supaya tambah seru. Kamu harus merekam atau membuat file suaramu sendiri.

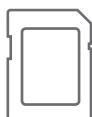
**TASK:** Cobalah untuk melakukan perbaikan berikut pada permainan:

Pada permainan mobil balap yang baru kamu buat, mobil berkata “I win!” ketika mencapai garis finish. Tetapi tulisannya cepat sekali hilangnya, sehingga susah dibaca. Tetapi, jika kita mengganti skripnya supaya tulisannya tetap ada di layar selama, 2 detik, ini akan memungkinkan mobil kedua juga mencapai garis finish pada waktu tersebut dan juga mengatakan “I win!”. Jadinya kita tidak tahu siapa yang menang.

Cobalah buat versi baru dari permainan, simpan dengan nama “racing\_game\_v2”, kali ini buat mobil yang kalah berhenti di lintasan saat mobil yang menang menyentuh garis finish. Untuk melakukan ini, kamu butuh menggunakan variabel.

Beri nama variabel ini dengan “have\_winner”. Nilai default dari “have\_winner” diset menjadi “0”. Ketika salah satu mobil memang, saat menyentuh garis finish putih, ubah nilai variabel “have\_winner” menjadi “1”. Ini akan menjadi pemicu untuk menghentikan kedua mobil berhenti.

Terakhir, ubah ketika mobil menyentuh rumput tidak menyebabkan mobil rusak melainkan membuat mobil melambat, sehingga pemain memiliki kesempatan untuk mengembalikan mobil ke lintasan balapan.



Untuk melihat contoh lengkap permainan ini, buka  
[RPiScratch/Projects/game\\_balap\\_v2](#)

## Berikutnya apa?

Selamat! Kamu sekarang adalah seorang Scratch Programmer. Mudah-mudahan, ini hanya awal bagimu, dan kamu akan terinspirasi untuk membuat permainanmu sendiri dengan bahasa Scratch. Ada banyak lagi ketrampilan dan teknik pemrogramman yang harus dipelajari.

Untungnya, salah satu yang menyenangkan dari Scratch adalah ada banyak sekali bantuan dan nasehat tentang scratch di internet. Berikut adalah beberapa tautan dari beberapa tutorial online untuk kamu eksplorasi:

[http://learnscratch.org](http://learnsscratch.org)  
<http://www.scratch.ie>  
<http://scratch.redware.com/index.php>  
<http://blogs.wsd1.org/etr/?p=395>  
<http://scratched.media.mit.edu>  
<http://morpheus.cc/ict/scratch/default.htm>

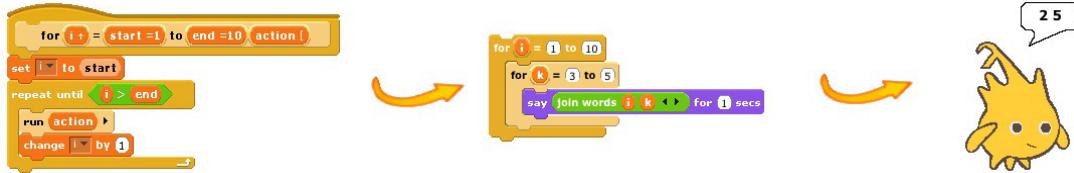
Dan mengapa kamu tidak beritahuakn gurumu tentang Scratch? Mungkin kamu dan teman-teman yang sama-sama suka dari sekolahmu bahkan membuat Club Scratch. Dan ketika kamu mendapat projek, kamu dapat menanyakan ke gurumu apakah boleh dikerjakan menggunakan Scratch. Yang terpenting adalah

*Kesulitan dan tidak tau bagaimana memecahkan nya? Jangan diam saja. Ada banyak forum yang bersahabat pada website Scratch website dimana banyak pemakai Scratch yang siap membantumu.*



## Buat balok-balokmu sendiri

Ketika kamu sudah menjadi programer Scratch yang mahir, kamu akan mengalami kejadian dimana tidak ada skrip balok yang dapat dipakai untuk mengerjakan yang kamu inginkan. Atau kamu ingin memperbaiki program-mu dengan membuat



balok-balok yang lebih baik. Kamu bisa mendapat bantuan dari BYOB pada alamat <http://byob.berkeley.edu>

Terima kasih sudah membaca petunjuk ini. Aku harap ini membantumu dengan memberikan waktu yang menyenangkan bersama dengan Raspberry Pi-mu.

Sekarang saatnya untuk lanjut ke bab berikutnya, dimana kamu akan belajar bagaimana membuat program dengan bahasa Python. Sangat menarik!