Modul Berpikir Komputasional - Scratch

Tim Materi Berpikir Komputasional 2024/2025

Catatan

1. Modul ini dirancang untuk dapat menjadi pegangan pemrograman Berpikir Komputasional, sehingga banyak hal yang dipotong karena keluar dari konteks Berpikir Komputasional.
2. Anda dapat membuka modul ini saat latihan praktikum.
3. Anda sangat disarankan untuk mencoba menjalankan semua program modul ini di komputer Anda, supaya Anda dapat mengetahui keluaran dari program yang ada.
4. Anda sangat disarankan untuk bereksperimen dari program-program yang ada di modul ini supaya Anda mendapat gambaran lebih jelas mengenai apa yang program Anda lakukan.
5. Anda sangat disarankan membaca tutorial dari tempat lain dan mengeksplor sendiri bahasa yang Anda gunakan.

Contents

1 Pengenalan Singkat Scratch

1.1 Pendahuluan

Selamat datang di bab pertama modul praktikum ini! Pada bab ini, kita akan menjelajahi sejarah singkat Scratch, sebuah platform pemrograman yang menyenangkan dan mudah dipelajari.

1.2 Sejarah Singkat dari Scratch

Scratch adalah lingkungan pemrograman yang dirancang khusus untuk mempelajari dasar-dasar pemrograman. Ini dikembangkan oleh Lifelong Kindergarten Group di MIT (Massachusetts Institute of Technology). Tujuan utama Scratch adalah membantu orang-orang, terutama anak-anak dan pemula, belajar cara berpikir secara kreatif, memecahkan masalah, dan berkolaborasi melalui pemrograman.

Seiring berjalannya waktu, Scratch terus berkembang dan meningkat. Versi Scratch 2.0 dirilis pada tahun 2013 dengan peningkatan signifikan dalam antarmuka pengguna dan fitur-fitur baru. Scratch 2.0 memungkinkan pengguna untuk membuat proyek yang lebih kompleks dengan lebih banyak pilihan blok pemrograman.

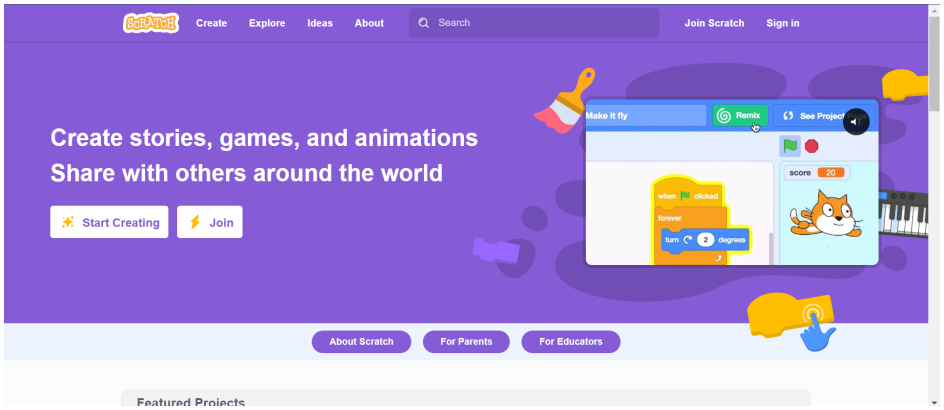
Pada tahun 2019, Scratch 3.0 diluncurkan dengan perubahan desain yang lebih modern dan penambahan fitur-fitur baru. Ini membuat Scratch lebih mudah digunakan dan menarik bagi pengguna baru.

Scratch digunakan oleh berbagai kalangan, termasuk anak-anak di sekolah dan rumah, guru, dan bahkan orang dewasa yang ingin belajar pemrograman. Dengan antarmuka yang ramah pengguna dan fitur-fitur yang intuitif, Scratch memungkinkan siapa saja untuk menciptakan karya kreatif mereka sendiri dengan menggunakan pemrograman.

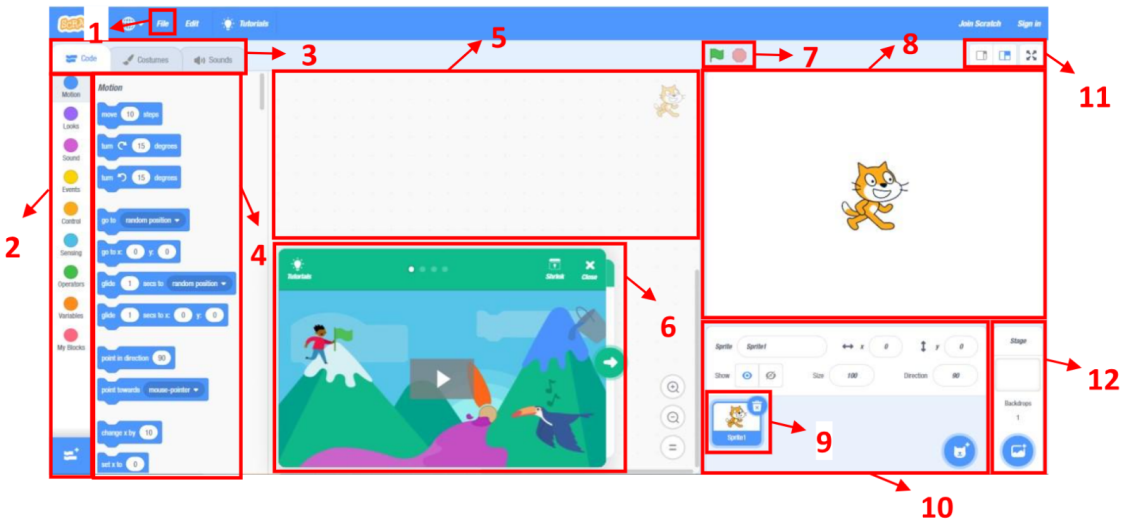
1.3 Pengenalan Area Kerja dan Fitur pada Scratch

Scratch memiliki berbagai fitur yang memungkinkan kita untuk membuat animasi, permainan, dan karya kreatif lainnya dengan mudah.

Untuk memulai sebuah project, kamu harus terlebih dahulu membuka http://scratch.mit.edu/, lalu klik tombol Start Creating.



Setelah itu halaman baru akan terbuka, fitur pada halaman baru kira-kira terlihat seperti ini:



Keterangan:

1. **File**: dapat digunakan untuk menyimpan hasil proyek pengguna dan juga dapat digunakan untuk mengunggah hasil proyek yang sebelumnya telah dikerjakan.
2. **Code tab**: berisi kategori kode interaktif yang dapat digunakan.
3. **Code – costumes – sounds tab**: tab code digunakan untuk menunjukkan kategori kode, tab costumes digunakan untuk menyunting kostum pada sprite, dan tab sounds untuk menyunting suara pada sprite.
4. **Code puzzle tab**: berisi potongan kode yang dapat di drag-and-drop dan dikelompokkan berdasarkan warna sesuai kategori kode tersebut.
5. **Editor area**: merupakan tempat untuk meletakkan sekuens kode pada proyek. Terlihat sprite yang sedang dikerjakan muncul di bagian kanan atas editor area.
6. **Tutorial tab**: berisi tutorial Scratch yang membantu pengguna dalam mengerjakan proyek. Tab ini dapat digeser, dikecilkan (shrink) ataupun ditutup.
7. **Start and stop button**: tombol hijau merupakan tombol untuk memulai jalannya program, sementara tombol merah merupakan tombol untuk menghentikan semua perintah.
8. **Results area**: merupakan area dimana program yang berasal dari kode di editor area dijalankan secara visual.
9. **Sprite**: merupakan karakter atau objek yang digunakan dalam proyek pengguna. Tanda biru di sekitar kotak sprite berarti sprite tersebut sedang dipilih.
10. **Sprite configuration**: bertujuan untuk menyunting pengaturan yang ada pada sprite seperti nama, posisi, ukuran, arah, visibilitas, dan terdapat pula tombol biru yang berfungsi untuk menambahkan sprite.
11. **Deploy area configuration**: bertujuan untuk memperkecil, memperbesar, dan membuat full screen dari deploy area tersebut.
12. **Stage tab**: berfungsi sebagai daerah pengaturan background pada program. Tombol biru digunakan menambahkan atau mengganti background pada program.

1.4 Pengenalan Kategori Kode

Kategori kode ini dapat dilihat pada bagian paling sebelah kiri dari daerah kerja atau berada pada nomor 2 (Code tab) pada bagian pengenalan daerah kerja. Kode ini terdiri dari beberapa kategori yaitu:

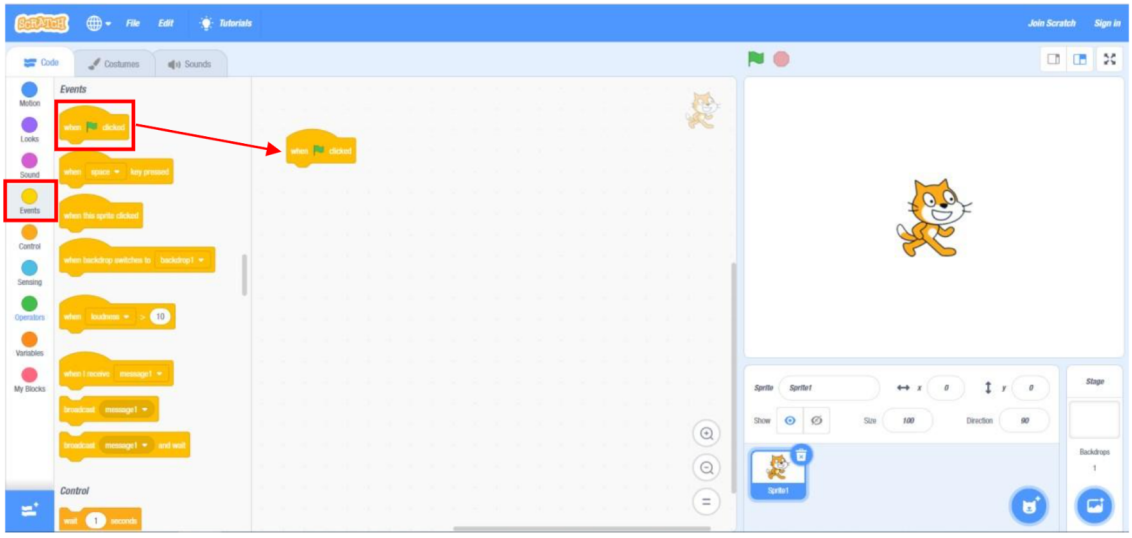
Keterangan:

1. **Motion**: merupakan kumpulan kode yang memiliki fungsi untuk menggerakan sprite. Bekerja dengan cara mengubah posisi dari sprite di deploy area.
2. **Looks**: berfungsi untuk mengubah atau menambah visualisasi pada sprite dan lingkungan di sekitarnya seperti menambahkan bubble text halo, mengubah warna, mengubah kostum, mengubah ukuran, mengubah backdrop, dll.
3. **Sound**: berfungsi untuk menambahkan suara pada sprite. Dapat digunakan untuk mengubah volume dan pitch pada suara sprite.
4. **Events**: berfungsi untuk menambahkan pemicu awal agar sebuah kode bisa berjalan, seperti when green flag clicked, when x key pressed, dll. Tombol pemicu dapat diubah sesuai keinginan pengguna.
5. **Control**: digunakan untuk mengontrol dan mengatur suatu sekuens kode seperti tunggu x detik, repeat x, forever, if x then y else z, wait until, repeat until, dll.
6. **Sensing**: merupakan fungsi yang dapat berinteraksi dengan pengguna dengan cara menghasilkan kode yang mengharuskan adanya input pengguna. Contohnya saat mouse menyentuh bagian tertentu, menanyakan nama, mendeteksi aktivitas tertentu dan mengeluarkan output, dll.
7. **Operators**: merupakan fungsi yang digunakan untuk melakukan operasi matematika pada program.
8. **Variables**: digunakan untuk mengatur variabel yang telah pengguna buat secara custom.
9. **My Blocks**: berfungsi untuk membuat block custom sesuai dengan kode keinginan pengguna.
10. **Add Extension**: berfungsi untuk menambah accessories yang berada di luar fungsi Scratch.

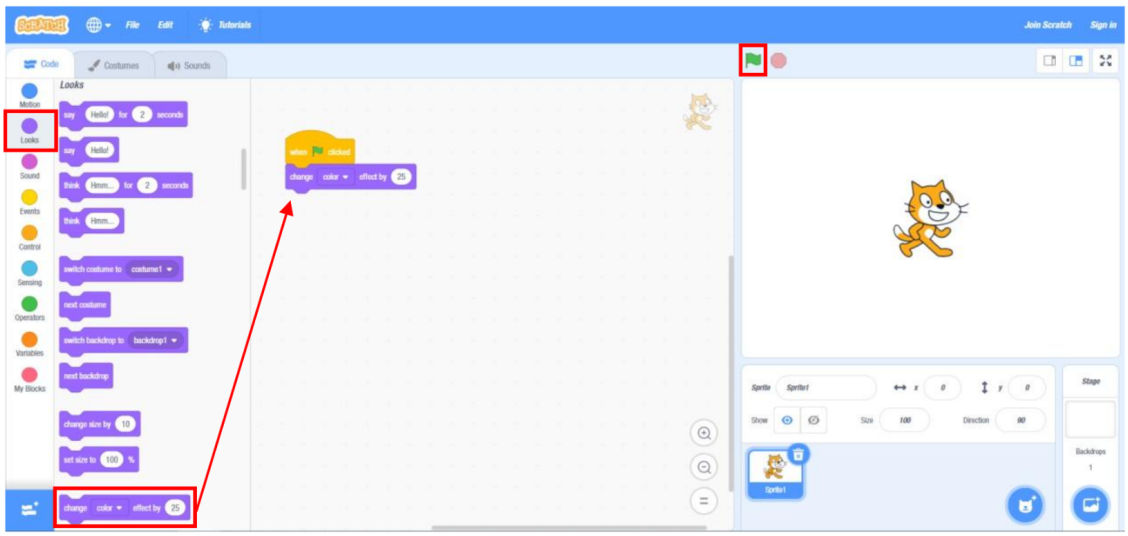
2 Memulai Sebuah Program

Untuk memulai sebuah program, pertama kamu harus memilih dahulu sprite yang akan gunakan, secara default, sprite kucing Scratch akan dipilih.

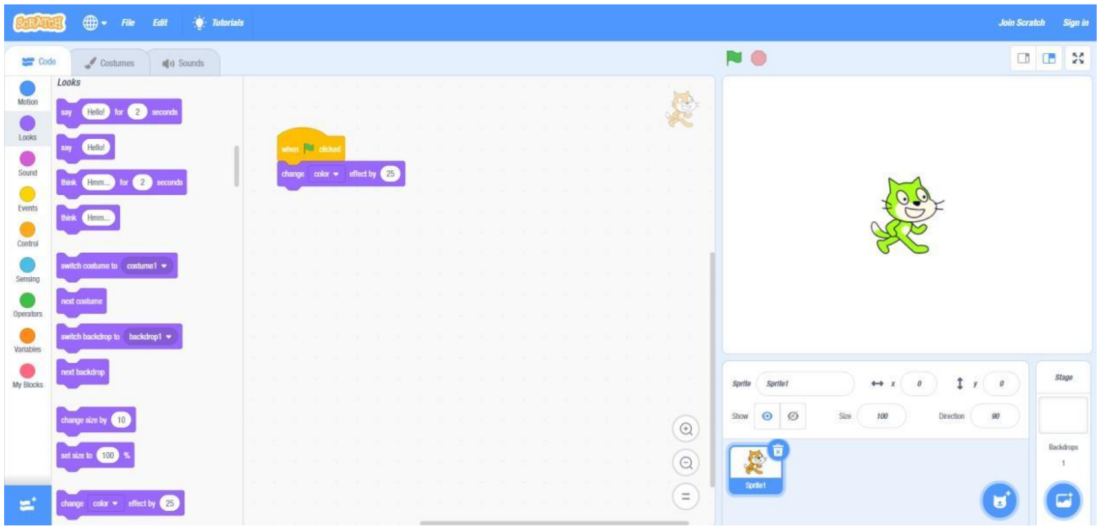
Selanjutnya, kode yang diinginkan dapat diseret dari daerah code puzzle tab. Pada umumnya diletakkan terlebih dahulu potongan yang digunakan untuk memulai sekuens. Pada kasus ini kita akan menggunakan perintah "when green flag clicked". Pertama, klik terlebih dahulu tombol "Event", lalu drag-and-drop perintah yang akan digunakan (lihat Gambar di bawah ini).



Setelah potongan pemulai sekuens diletakkan, kamu dapat menggunakan potongan apa pun untuk langkah berikutnya sesuai keinginan kamu. Sebagai contoh, kita akan menggunakan perintah "change color by effect 25". Pertama, klik tombol "Looks", lalu selanjutnya drag-and-drop perintah yang akan digunakan menuju bagian bawah potongan pertama. Setelah kamu memasukkan potongan tersebut, klik bendera hijau untuk menjalankan program (lihat Gambar di bawah ini).



Berikut merupakan tampilan program setelah dijalankan.

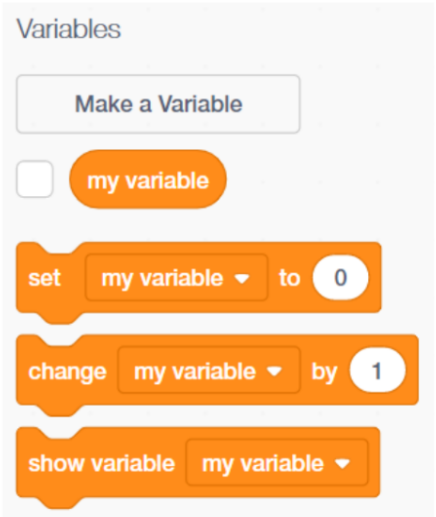


3 Input dan Output

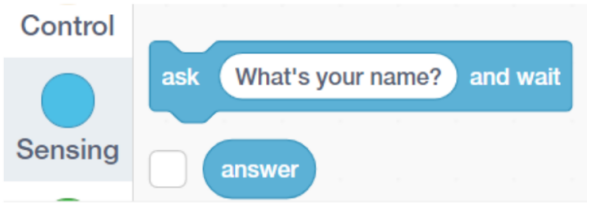
Input adalah cara untuk memasukkan data atau informasi ke dalam program. Di Scratch, terdapat beberapa cara untuk mendapatkan input dari pengguna. Sedangkan output adalah hasil atau respons yang ditampilkan oleh program kepada pengguna.

Input dan Output adalah konsep penting dalam pemrograman, dan Scratch menyediakan berbagai cara untuk berinteraksi dengan pengguna dan menghasilkan output yang menarik. Dengan menggunakan blok-blok pemrograman, sensor, dan variabel, kamu dapat menciptakan program-program yang interaktif dan responsif dalam proyek Scratch kamu.

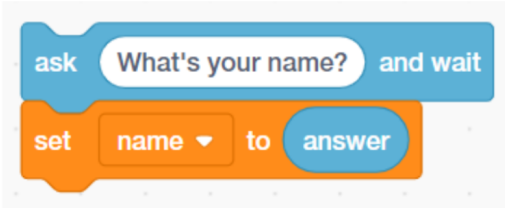
Untuk menerima input, kita membutuhkan variabel untuk menyimpan data yang diinputkan. Untuk membuat variabel, gunakan “Make a Variable”. Variabel yang dibuat akan muncul di bawahnya.



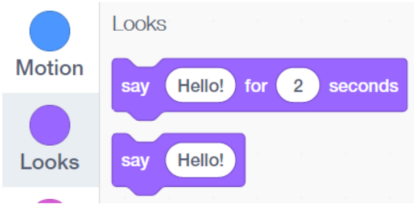
Untuk menerima input, gunakan kode “Ask (...) and wait”. Input yang diberikan akan disimpan sementara dalam variabel “answer”.



Sebagai contoh, di bawah ini adalah program yang menerima input nama dan menyimpannya dalam variabel bernama “name”.



Untuk menampilkan output, gunakan kode “say (...)”

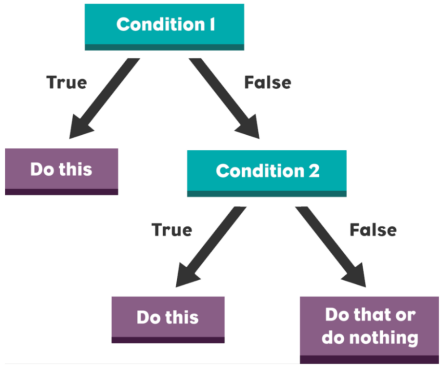


4 Percabangan pada Scratch

Percabangan adalah salah satu konsep dasar dalam pemrograman yang digunakan untuk mengambil keputusan berdasarkan situasi tertentu. Dalam Scratch, kita dapat menggunakan blok ”if” dan ”else” untuk mengimplementasikan percabangan.

Percabangan memungkinkan program kita untuk menjalankan perintah yang berbeda tergantung pada suatu kondisi. Kondisi tersebut bisa benar (true) atau salah (false).

Kita dapat memvisualisasikan sebuah percabangan pada suatu pemrograman dalam gambar di bawah ini.



5 Perulangan pada Scratch

Perulangan adalah konsep yang digunakan dalam pemrograman untuk menjalankan serangkaian instruksi berulang kali. Dalam Scratch, kita dapat menggunakan dua jenis perulangan: ”Repeat” dan ”Repeat until”.

1. **Repeat (...)**:
   Blok ”Repeat” digunakan ketika kita tahu berapa kali pengulangan harus dilakukan. Kita dapat menentukan jumlah pengulangan dengan mengisi kotak kosong di dalam blok ”Repeat”. Misalnya, jika kita ingin menjalankan serangkaian instruksi sebanyak 5 kali, kita dapat memasukkan angka 5 di dalam blok ”Repeat”.
2. **Repeat until (ekspresi logika)**:
   Blok ”Repeat until” digunakan ketika kita tidak tahu berapa kali pengulangan harus dilakukan, tetapi kita mengetahui kondisi kapan pengulangan harus berhenti. Kondisi berhenti ditentukan oleh suatu ekspresi logika. Pengulangan akan berlanjut hingga kondisi tersebut terpenuhi atau bernilai true.

Dalam kedua jenis pengulangan ini, aksi atau instruksi yang diulang dapat ditempatkan di dalam blok pengulangan. Misalnya, kita dapat memindahkan sprite ke suatu posisi, mengubah warna, atau melakukan aksi lainnya dalam setiap pengulangan.

Pengulangan sangat berguna dalam berbagai situasi, misalnya untuk melakukan perhitungan matematika berulang kali, mengontrol pergerakan sprite dalam game, atau mengulang interaksi dengan pengguna.

6 Pelajari Lebih Lanjut

Untuk mendalami materi-materi dalam modul ini dan mengembangkan pemahaman lebih lanjut tentang Scratch, berikut adalah beberapa rekomendasi:

1. **Dokumentasi Scratch**: Jelajahi dokumentasi resmi Scratch di website mereka untuk memperdalam pemahaman tentang konsep dasar Scratch, jenis blok kode, dan pengembangan proyek yang lebih rumit.
2. **Sejarah dan Asal-usul**: Cari literatur atau artikel untuk mendalami sejarah dan asal-usul Scratch serta pengaruhnya dalam pemrograman pendidikan.
3. **Tutorial dan Video**: Gunakan tutorial dan video tutorial tentang Scratch untuk memahami lebih baik fitur-fitur dan area kerja dalam Scratch.
4. **Latihan Praktis**: Buatlah proyek kecil yang menggabungkan berbagai kategori kode untuk menguasai setiap kategori. Kamu juga dapat mengeksplorasi proyek-proyek dari komunitas Scratch.

Dengan memanfaatkan sumber-sumber ini, kamu akan dapat mendalami Scratch dan meningkatkan keterampilan pemrograman kamu dengan lebih baik.