

# Rancang Bangun Permainan Interaktif dengan Scratch

Seng Hansun

Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia  
hansun@umn.ac.id

Diterima 31 Mei 2014

Disetujui 16 Juni 2014

**Abstract**—Scratch is a programming language and an online community where people can program and share interactive media, such as stories, games, and animation. In this paper, the author tries to use Scratch in designing and building an interactive game, called Swimming Fish. The project has been done completely and shared to all people through Scratch website.

**Index terms**—game, interactive, project, Scratch, swimming fish

## I. PENDAHULUAN

Scratch merupakan suatu bahasa pemrograman *visual* yang dikembangkan oleh Lifelong Kindergarten *research group* di MIT Media Lab dengan dukungan dana dari berbagai pihak, baik pemerintah maupun swasta [1]. Scratch dapat digunakan untuk membuat cerita interaktif, *game* interaktif, dan animasi, serta dapat dibagikan kepada orang lainnya melalui sarana internet [2].

Scratch didesain dengan tujuan pembelajaran dan pemahaman. Beragam pendidik telah bergabung dan mendukung pengembangan Scratch sejak 2007, baik dalam lingkungan formal maupun non formal, seperti K-12 *classroom teachers*, peneliti Ilmu Komputer, pustakawan, pendidik sejarah museum, dan para orang tua. Para pendidik ini tergabung dalam sebuah komunitas *online* yang disebut sebagai ScratchEd, yang diluncurkan pada Juli 2009, dan menjadi tempat berbagi pengalaman, pertukaran sumber data dan pertanyaan, serta mencari orang. Semua orang dapat bergabung dalam komunitas *online* tersebut secara gratis melalui tautan <http://scratched.media.mit.edu/> [3].

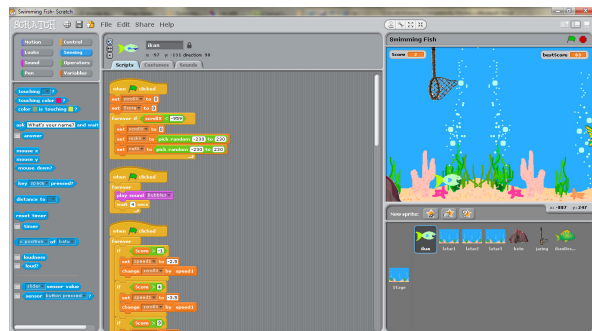
Dalam makalah ini, akan dijelaskan langkah-langkah penerapan bahasa pemrograman *visual* Scratch dalam membuat sebuah permainan interaktif, dengan nama *project* ‘Swimming Fish’. Permainan ini dibuat oleh penulis yang merupakan seorang pemula dalam Scratch *programming* dan telah dipublikasikan secara *online* di situs resmi Scratch dengan tautan <http://scratch.mit.edu/projects/22760956/?fromexplore=true> [4]. Melalui praktik secara langsung, penulis dapat menyimpulkan bahwa Scratch dengan bantuan antarmuka visual

yang interaktif, dapat membantu *programmer* dalam membuat aplikasi-aplikasi interaktif, terutama para pemula dalam konsep pemrograman.

## II. MEMBANGUN GAME INTERAKTIF DENGAN SCRATCH

Scratch sebagai lingkungan pemrograman visual dapat membantu anak-anak muda untuk berpikir secara kreatif, menalar secara sistematis, dan bekerja secara kolaboratif, yang merupakan keahlian mendasar yang dibutuhkan oleh semua orang saat ini.

Pada bab ini akan dijelaskan langkah-langkah pembuatan sebuah permainan interaktif dengan memanfaatkan Scratch versi 2.0 *offline*. Untuk versi *online*, cara penerapannya juga sama, namun langsung dikerjakan pada situs resmi Scratch dengan syarat bergabung menjadi anggotanya terlebih dahulu. Berikut adalah tampilan antarmuka utama Scratch 2.0 versi *offline*.



Gambar 1. Antarmuka lingkungan Scratch

### A. Sprites dan Costumes

Dalam pembuatan sebuah *project* di Scratch, dikenal istilah *sprites* dan *costumes*. *Sprites* merujuk pada obyek-obyek yang digunakan dalam *project*, sementara *costumes* adalah atribut kostum dari suatu *sprite*, sehingga *sprite* tersebut dapat menggunakan beberapa *costumes* yang berbeda.

Dalam *project* ini digunakan tujuh *sprites*, yang dapat dijabarkan seperti berikut.

#### 1) *Sprite* 1: ikan

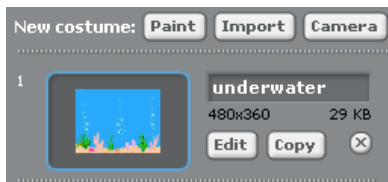
*Sprite* ini merupakan *sprite* utama, yakni berupa seekor ikan kecil yang memiliki dua *costumes* berbeda. Saat berenang mengambang (menekan tombol panah 'UP'), ikan akan terus-menerus berubah *costume*. *Sprite* ini dapat ditemukan dalam *file sprites* yang telah disediakan oleh Scratch.



Gambar 2. *Sprite* ikan dan *costume*-nya

## 2) *Sprite* 2: latar1

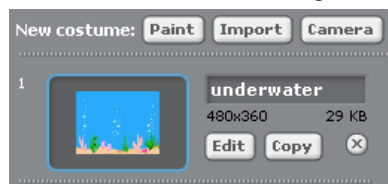
*Sprite* latar1 merupakan *sprite* yang digunakan sebagai latar permainan, yakni berupa ilustrasi kehidupan bawah air. *Sprite* ini juga telah disediakan oleh Scratch.



Gambar 3. *Sprite* latar1

## 3) *Sprite* 3: latar2

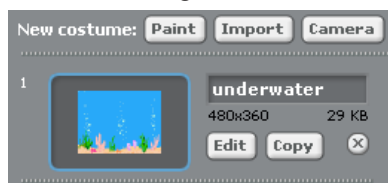
*Sprite* ini merupakan *copy* dari *sprite* latar1, yang akan digunakan sebagai latar permainan kedua. Perbedaannya hanyalah bahwa *sprite* latar2 ini merupakan cerminan *horizontal* dari *sprite* latar1.



Gambar 4. *Sprite* latar2

## 4) *Sprite* 4: latar3

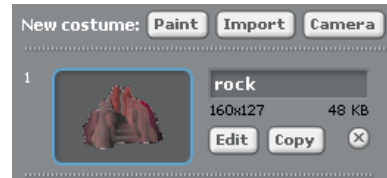
*Sprite* ini merupakan *copy* dari *sprite* latar1, tanpa ada perbedaan sama sekali. Bersama dengan *sprite* latar1 dan latar2, *sprite* ini akan digunakan sebagai latar permainan yang saling berkaitan dan diatur untuk memperlihatkan efek bergerak ke kiri.



Gambar 5. *Sprite* latar3

## 5) *Sprite* 5: batu

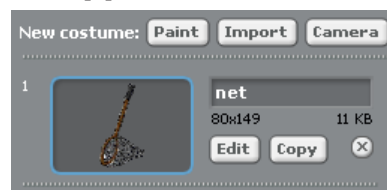
*Sprite* ini berupa gambar sebuah batu, yang juga telah disediakan oleh Scratch. *Sprite* ini akan digunakan sebagai rintangan yang harus dihindari oleh *sprite* ikan.



Gambar 6. *Sprite* batu

## 6) *Sprite* 6: jaring

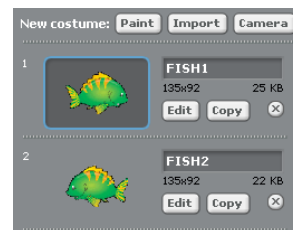
*Sprite* ini juga akan digunakan sebagai rintangan bagi *sprite* ikan, yakni berupa gambar sebuah jaring ikan. Gambar *sprite* ini diambil dari sumber luar dengan tautan <http://www.qacps.k12.md.us/ces/clipart/Carson%20Dellosa%20Clipart/Carson%20Dellosa%20Learning%20Themes/Images/Color%20Images/Ocean/> [5].



Gambar 7. *Sprite* jaring

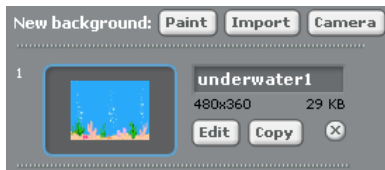
## 7) *Sprite* 7: ikanBesar

*Sprite* ikanBesar ini akan menjadi salah satu penghalang utama bagi *sprite* ikan, dimana *sprite* ikanBesar ini akan bergerak lebih dinamis di dalam air. Terdapat dua *costumes* untuk *sprite* ini, yang tidak lain hanya berbeda pada arah hadap ikanBesar tersebut. *Sprite* ini juga diambil dari sumber luar [5].



Gambar 8. *Sprite* ikanBesar

Selain ketujuh *sprites* yang dipersiapkan di atas, pada panggung (*stage*) juga perlu diganti latarnya, dari latar putih menjadi latar *underwater* yang sama seperti *sprite* latar1. Tujuannya untuk menyamarkan perbedaan warna saat *sprites* latar bergerak satu dengan lainnya.



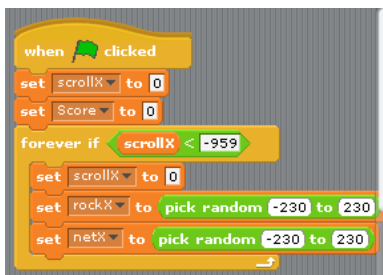
Gambar 9. Stage background

## B. Scripts

Setelah mempersiapkan latar *stage* dan *sprites* yang dibutuhkan, maka selanjutnya logika-logika pemrograman perlu diterapkan. Dalam Scratch, tiap *sprite* dapat memiliki logika pemrogramannya sendiri, yang dapat diatur dalam *Scripts Area*. Logika-logika atau *scripts* tersebut dapat dibangun dengan mudah menggunakan fasilitas *drag-and-drop block-block* yang telah disediakan dalam *Block Palette* ke dalam *Script Area* sebuah *sprite*. Sebagai referensi kegunaan masing-masing *block* dapat ditemukan pada makalah yang ditulis oleh Hansun [3] atau pada Reference Guide Scratch [6]. Berikut pengorganisasian *scripts* untuk tiap *sprite* berbeda yang digunakan dalam *project* ini.

### 1) Script ikan

*Sprite* ikan sebagai *sprite* utama memiliki *script* yang paling banyak, dengan tujuannya masing-masing. *Script* pertama yang dibutuhkan diperlihatkan pada gambar 10 berikut.



Gambar 10. Script 1 ikan

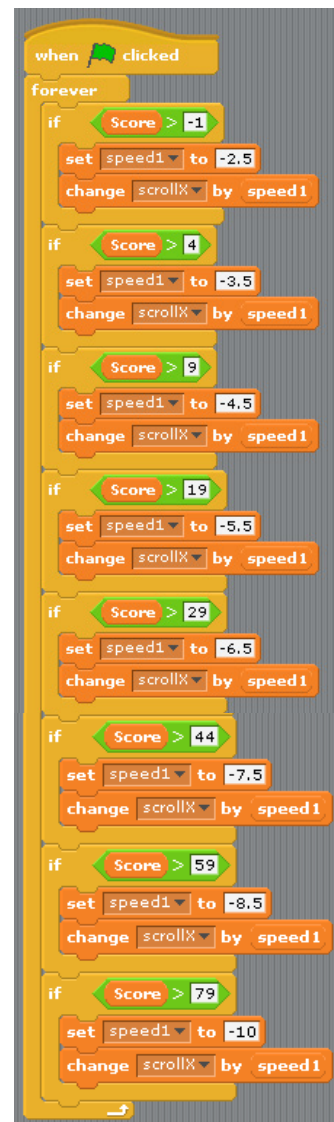
*Script* di atas digunakan untuk menginisialisasi variabel *scrollX* dan *Score* menjadi 0. Kemudian dilakukan pengecekan jika *scrollX* < -959, maka dilakukan inisialisasi ulang untuk nilai variabel *scrollX*, dan inisialisasi nilai variabel *rockX* dan *netX* secara acak. Variabel *Score* merupakan nilai skor yang diperoleh pemain dalam satu kali permainan, sementara variabel *scrollX* akan digunakan untuk menggeser *sprites* latar1, latar2, dan latar3 sepanjang sumbu X negatif (ke arah kiri). Variabel *rockX* dan *netX* akan menentukan posisi kemunculan rintangan batu dan jaring secara acak.



Gambar 11. Script 2 ikan

Selanjutnya, untuk memainkan musik latar digunakan *script* seperti yang diperlihatkan pada gambar 11. Musik yang dimainkan adalah 'Bubbles' yang tersedia di dalam Scratch.

Gambar 12 berikutnya memperlihatkan *script* yang digunakan untuk mengatur perubahan kecepatan gerak latar berdasarkan skor yang diperoleh pemain. Semakin tinggi skor yang diperoleh pemain, maka latar akan bergerak makin cepat, sehingga dapat menambah kedinamisan permainan.



Gambar 12. Script 3 ikan

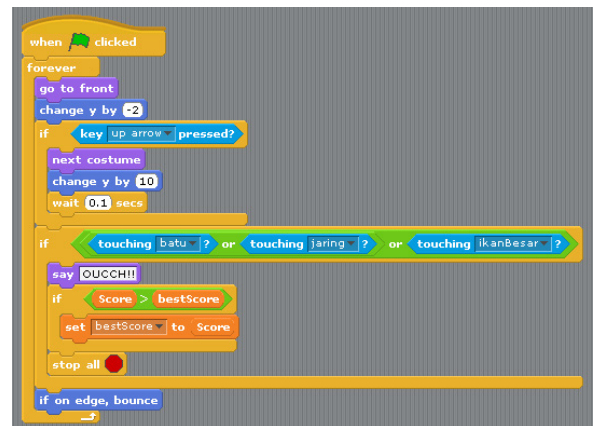
Dari gambar 12 terlihat bahwa terdapat total delapan tingkat kecepatan yang berubah mulai dari awal permainan hingga akhir permainan, yakni mencapai skor maksimum. *Range* perubahan kecepatannya diatur dari perubahan nilai variabel scrollX, yakni dimulai dari -2.5 hingga -10.

*Script* selanjutnya adalah *script* yang digunakan untuk memberikan *feedback* kepada pemain, yakni berupa ucapan si ikan saat berhasil mencapai perolehan skor tertentu. Seperti yang dapat dilihat pada *script* gambar 13 di halaman berikutnya, nilai maksimum skor yang dapat diperoleh pemain adalah 100, dan pada saat mencapai skor maksimum tersebut permainan akan berhenti.



Gambar 13. *Script* 4 ikan

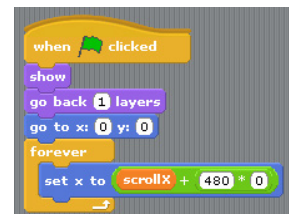
*Script* terakhir untuk *sprite* ikan diperlihatkan pada gambar 14 di bawah. *Script* ini akan membuat *sprite* ikan secara otomatis tenggelam ke dasar lautan dengan perubahan nilai *y* sebanyak -2 satuan. Pemain dapat mempertahankan ikan agar tetap mengambang di air dengan menekan tombol arah panah ke atas 'UP', dan secara otomatis *sprite* ikan akan berubah *costume*. Selain itu, jika *sprite* ikan menyentuh obyek-obyek rintangan, seperti batu, jaring, dan ikan besar; maka permainan akan berakhir dengan ditandai dipanggilnya *block* 'stop all'. Skor yang diperoleh pemain akan dibandingkan dengan skor terbaik yang tersimpan dalam permainan. Jika skornya lebih baik dari permainan-permainan sebelumnya, maka skor tersebut akan diganti dan diperlihatkan di laman utama (*stage*) permainan.



Gambar 14. *Script* 5 ikan

## 2) *Script* latar1

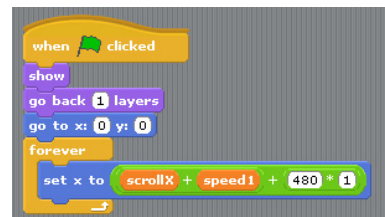
*Sprite* latar1 hanya memiliki satu *script* sederhana yang digunakan untuk mengatur pergerakan latar1 ke kiri sesuai nilai variabel scrollX yang didapatkan.



Gambar 15. *Script* latar1

## 3) *Script* latar2

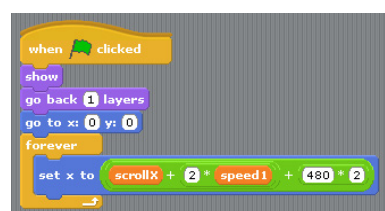
*Sprite* latar2 juga hanya memiliki satu *script* sederhana dengan tujuan yang sama seperti pada *script* latar1, dengan perbedaan lokasi nilai *x* *sprite* latar2 tersebut.



Gambar 16. *Script* latar2

## 4) *Script* latar3

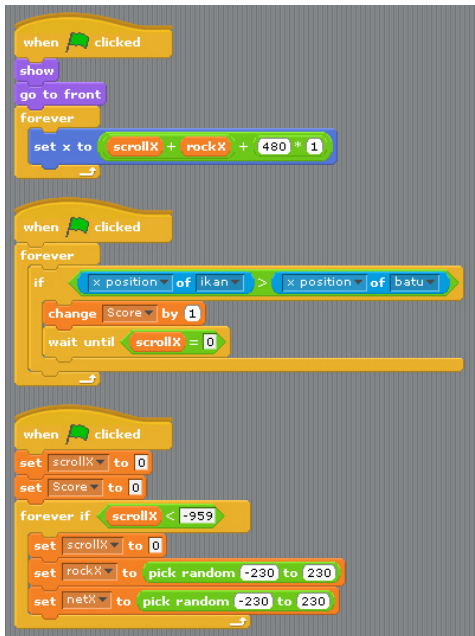
*Sprite* latar3 memiliki satu *script* yang sama seperti *sprite* latar1 dan latar2 sebelumnya, dengan perbedaan hanya pada lokasi nilai *x* *sprite* latar3 ini.



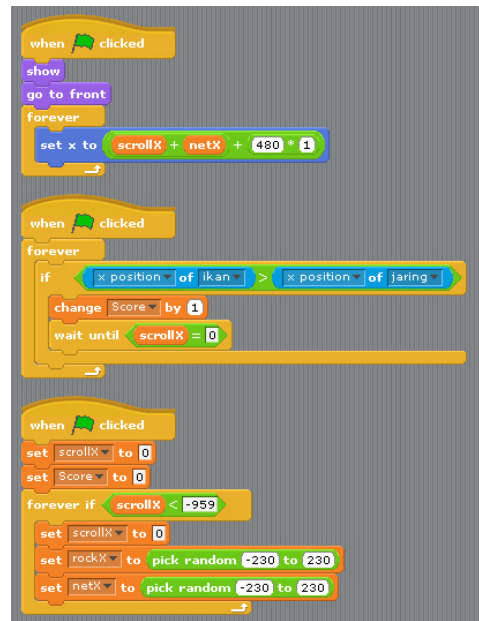
Gambar 17. *Script* latar3

5) *Script batu*

Pada gambar 18 terlihat tiga *scripts* berbeda yang diterapkan pada *sprite* batu. *Script* pertama digunakan untuk mengatur kemunculan batu di latar2 secara acak. *Script* kedua digunakan untuk mengubah nilai skor pemain sebanyak 1 satuan saat *sprite* ikan berhasil melalui *sprite* batu dengan mengecek posisi x dari kedua *sprite* tersebut. *Script* terakhir merupakan *script* yang sama seperti *script* 1 *sprite* ikan yang telah dijelaskan sebelumnya. *Script* ini digunakan untuk menginisialisasi kembali nilai variabel *scrollX* dan *Score* menjadi 0, serta mengatur nilai variabel *rockX* dan *netX* untuk dipilih secara acak antara -230 hingga 230.

Gambar 18. *Script* batu6) *Script jaring*

*Sprite* jaring memiliki *scripts* yang hampir serupa dengan *scripts* pada *sprite* batu. Kemunculan jaring secara acak akan diatur oleh *script* pertama, sementara pada *script* kedua skor pemain akan bertambah 1 satuan saat ikan berhasil melalui jaring tersebut. Tidak lupa diinisialisasi juga beberapa variabel yang diperlukan pada *script* ketiga.

Gambar 19. *Script* jaring7) *Script ikanBesar*

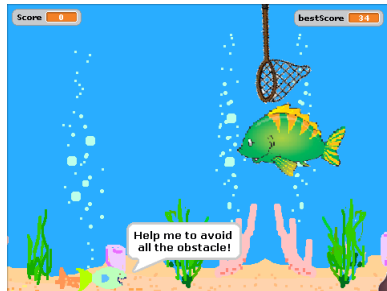
*Sprite* ikanBesar merupakan salah satu rintangan utama bagi *sprite* ikan. *Sprite* ikanBesar ini akan bergerak secara dinamis dari kanan ke kiri dan sebaliknya, yang diatur dalam *script* pertama *sprite* ini. *Script* kedua digunakan untuk menginisialisasi kembali nilai variabel-variabel yang dibutuhkan oleh *sprite* ini.

Gambar 20. *Script* ikanBesar

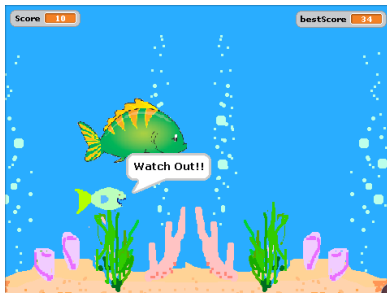
## III. BERBAGI SECARA ONLINE

Setelah berhasil membangun suatu aplikasi, dalam hal ini permainan interaktif, Scratcher

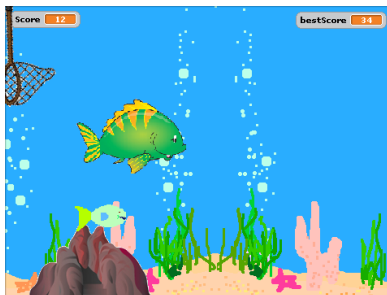
dapat mempublikasikan aplikasinya dalam situs resmi Scratch maupun situs-situs lainnya. *Project 'Swimming Fish'* ini juga telah dipublikasi dan dibagikan secara *online* melalui tautan <http://scratch.mit.edu/projects/22760956/?fromexplore=true> [4]. Berikut beberapa *screenshots* tampilan hasil akhir permainan yang dikembangkan.



Gambar 21. Ilustrasi awal permainan



Gambar 22. Si ikan berusaha menghindari rintangan-rintangan yang ada



Gambar 23. Permainan berakhir saat ikan menyentuh salah satu rintangan dalam permainan

#### IV. SIMPULAN

Scratch sebagai bahasa pemrograman *visual* yang populer menawarkan sejumlah kemudahan dan fitur yang menarik bagi para pemula dalam mempelajari cara pemrograman [3]. Bahkan sejumlah sekolah menengah atas dan universitas (termasuk Harvard dan University of California, Berkeley) juga telah menggunakan Scratch sebagai langkah pertama dalam memperkenalkan konsep pemrograman kepada para peserta didiknya [7].

Dalam makalah ini, penulis telah mencoba untuk membangun sebuah permainan interaktif yang diberi nama 'Swimming Fish'. Dengan berbagai fitur yang ditawarkan oleh Scratch, para pemula dalam konsep pemrograman tidak perlu menguasai sintaks bahasa pemrograman tertentu, namun cukup memahami logika pemrograman yang ingin diterapkan dalam aplikasi yang dibangun.

#### DAFTAR PUSTAKA

- [1] Chiang, J., Shall We Learn Scratch Programming, <http://shallwelearn.com>.
- [2] Getting Started with SCRATCH Version 1.4, <http://scratch.mit.edu>, Lifelong Kindergarten Group, MIT Media Lab.
- [3] Hansun, S., Scratch: Pemrograman Visual untuk Semuanya, belum terbit.
- [4] <http://scratch.mit.edu/projects/22760956/?fromexplore=true>, diakses 28 Mei 2014.
- [5] <http://www.qacps.k12.md.us/ces/clipart/Carson%20Dellosa%20Clipart/Carson%20Dellosa%20Learning%20Themes/Images/Color%20Images/Ocean/>, diakses 29 Mei 2014.
- [6] Reference Guide: Scratch Version 1.4, <http://scratch.mit.edu>, Lifelong Kindergarten Group, MIT Media Lab.
- [7] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y., Scratch: Programming for All, Communications of the ACM, Vol. 52, No. 11, November 2009.