

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/301546326>

Scratch: Pemrograman Visual untuk Semuanya

Article · June 2014

DOI: 10.31937/si.v5i1.218

CITATIONS

2

READS

3,387

1 author:



Seng Hansun

UNSW Sydney

104 PUBLICATIONS 216 CITATIONS

SEE PROFILE

Scratch

Pemrograman Visual untuk Semuanya

Seng Hansun

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

Diterima 31 Mei 2014

Disetujui 16 Juni 2014

Abstract—In this modern era, the need to have a more tinkerable, more meaningful, and more social programming language has arisen. One of the programming environments that catch up with this issue is Scratch.

In this paper, the author tries to explain Scratch as a visual programming language, its' basic concepts, and its' implementations. As an example, one interactive game that has been built using Scratch is shown at the end of this paper.

Index terms—game, Scratch, visual programming

I. PENDAHULUAN

Scratch merupakan suatu bahasa pemrograman baru yang memudahkan Anda dalam membuat cerita interaktif, *game* interaktif, dan animasi, serta membagikan karya kreasi Anda kepada orang lainnya melalui sarana internet [1]. Scratch 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.

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 [2]. Para pendidik ini tergabung dalam sebuah komunitas *online* yang disebut sebagai ScratchEd, yang diluncurkan pada Juli 2009, dan menjadi tempat berbagi kisah 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 disertasinya, Brennan [4] menyatakan bahwa dewasa ini kita hidup dalam budaya komputasional, yakni suatu budaya dimana manusia dikelilingi oleh sistem-sistem dan antarmuka komputasional, mulai dari jejaring sosial hingga infrastruktur perbankan, industri hiburan, dan sistem

transportasi. Beliau menggunakan pendekatan kualitatif untuk menelusuri pembelajaran dalam budaya komputasional, mempelajari bagaimana lingkungan bahasa pemrograman Scratch dan komunitas *online*-nya dapat diterapkan guna mendukung proses pembelajaran baik di dalam maupun di luar sekolah. Dalam penutup disertasinya, beliau meninjau kesempatan dan tantangan yang akan dihadapi oleh anak muda dan para pendidik dalam tiga perspektif, yakni sebagai seorang peneliti, seorang desainer, dan seorang pembelajar, yang mengarah pada kesimpulan pentingnya penerapan Scratch dalam budaya komputasional dewasa ini.

Pada makalah ini akan dijabarkan secara singkat seluk beluk Scratch sebagai bahasa pemrograman *visual*, pemanfaatan dan cara penerapannya. Selain itu, salah satu contoh permainan interaktif Ping Pong yang dikembangkan oleh kingminer29 – salah seorang Scratcher, istilah yang umum digunakan untuk merujuk pada *programmer* Scratch – juga ditampilkan di sini.

II. SCRATCH

Scratch merupakan bahasa pemrograman *visual* yang dikembangkan oleh Lifelong Kindergarten *research group* di MIT Media Lab [5]. Pengembangan Scratch didukung oleh pendanaan dari National Science Foundation, Microsoft, Intel Foundation, Nokia, dan MIT Media Lab *research consortia* [1]. Pada bab ini, akan dijelaskan perkembangan Scratch selama ini, antarmuka, dan blok pembangunan Scratch sebagai pengenalan dalam Scratch *programming*.

A. Perkembangan Scratch

Pengembangan Scratch dimulai sejak tahun 2003 yang dikepalai oleh Mitch Resnick, dan telah melalui beberapa versi perkembangan, sebagai berikut [6, 7].

1. Scratch 1.0

Dirilis pada 8 Januari 2007 dan merupakan versi Scratch pertama yang tersedia untuk umum. Beberapa perubahan dari versi *beta* meliputi *block* Next Costume, Next Background, Think(),

Think() for () Secs, Stop All Sounds, Forever If()); tombol baru untuk membuat *sprites*, kemampuan untuk berbagi *projects* secara *online*, perubahan ekstensi *file* dari *.scratch* menjadi *.sb*, dukungan beberapa bahasa, tombol Enter sebagai *shortcut* untuk Green Flag, dan kemampuan Import *project*.

2. Scratch 1.1

Dirilis secara resmi pada Mei 2007, dengan penambahan fitur baru berupa *installer* Windows yang baru, *block* Repeat Until (), bentuk Cap Block diperkenalkan, satu *sprite* dapat memainkan beberapa *notes* sekaligus, menambahkan *tags* untuk *projects* yang di-*sharing*, dan sebagainya.

3. Scratch 1.2

Merupakan rilis terakhir untuk Windows 98 dan ME, yang dirilis pada 7 Desember 2007. Beberapa fitur baru yang diperkenalkan adalah *block* Rest for () beats, kompresi gambar yang lebih baik, perbaikan beberapa *sample projects*, dan perubahan antarmuka di Paint Editor.

4. Scratch 1.3

Dirilis pada 2 September 2008, dengan beberapa fitur baru seperti Variables dapat menangani *strings*, Lists, Comments, penggunaan *fonts* dari komputer *user*, dan *block* If () Then, Repeat, Forever secara otomatis melingkupi *block stacks* lainnya.

5. Scratch 1.4

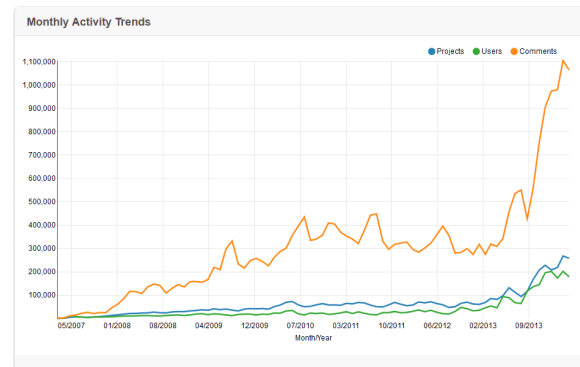
Dirilis secara resmi pada 19 Juli 2009, dengan fitur-fitur tambahan seperti *blocks* baru untuk penanganan *string*, *block* Ask () and Wait, *block* () Contains (), pemanfaatan *webcam* untuk mengambil gambar, dan sebagainya.

6. Scratch 2.0

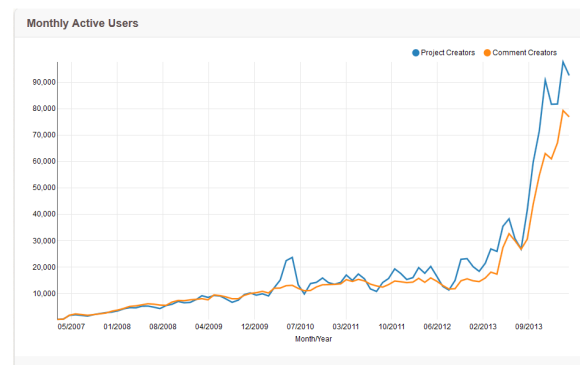
Scratch 2.0 dikembangkan selama beberapa tahun hingga versi *beta*-nya diperkenalkan pada 28 Januari 2013. Versi resminya dirilis pada 9 Mei 2013, dengan beberapa fitur baru seperti UI yang baru, Procedures, Cloning, Cloud data, Vector graphics, Sound editor, dan sebagainya.

Sejak diperkenalkan kepada umum pada 2007, terdapat 5.619.530 *projects* yang telah dibagikan, 3.387.470 *user* yang teregister, 27.229.307 komentar yang di-*post*, dan 440.857 *studios* yang dibangun [8]. Grafik 1 dan 2 menunjukkan statistik aktivitas *user* sebagaimana yang direkam dalam situs resmi Scratch, yang memperlihatkan jumlah *projects*, *users*, dan komentar yang diberikan.

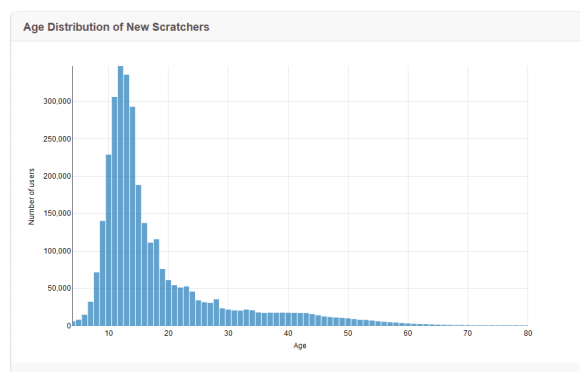
Sementara itu, dari gambar 3 terlihat bahwa distribusi umur Scratcher baru paling banyak berasal dari anak-anak remaja dengan rentang usia 10-20 tahun. Hal ini memperlihatkan bahwa Scratch berhasil menarik minat para anak remaja untuk mempelajari bahasa pemrograman ini. Distribusi penyebaran Scratchers di dunia secara luas juga diperlihatkan pada Gambar 4 dengan intensitas jumlahnya masing-masing.



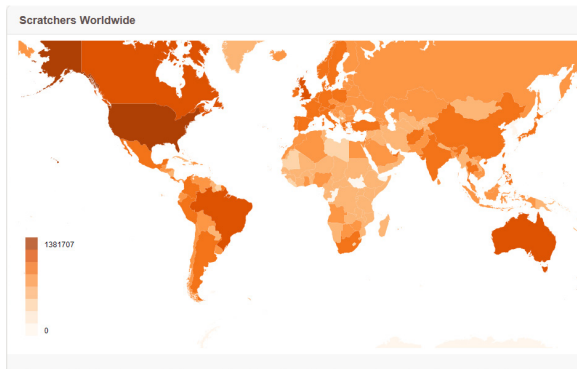
Gambar 1. Tren aktivitas situs Scratch [8]



Gambar 2. User aktif [8]



Gambar 3. Distribusi umur Scratcher baru [8]



Gambar 4. Distribusi Scratchers dunia [8]

B. Antarmuka Scratch

Sebelum mempelajari Scratch, perlu diketahui antarmuka bahasa pemrograman *visual* Scratch terlebih dahulu. Beberapa hal yang perlu diketahui, antara lain [9]:

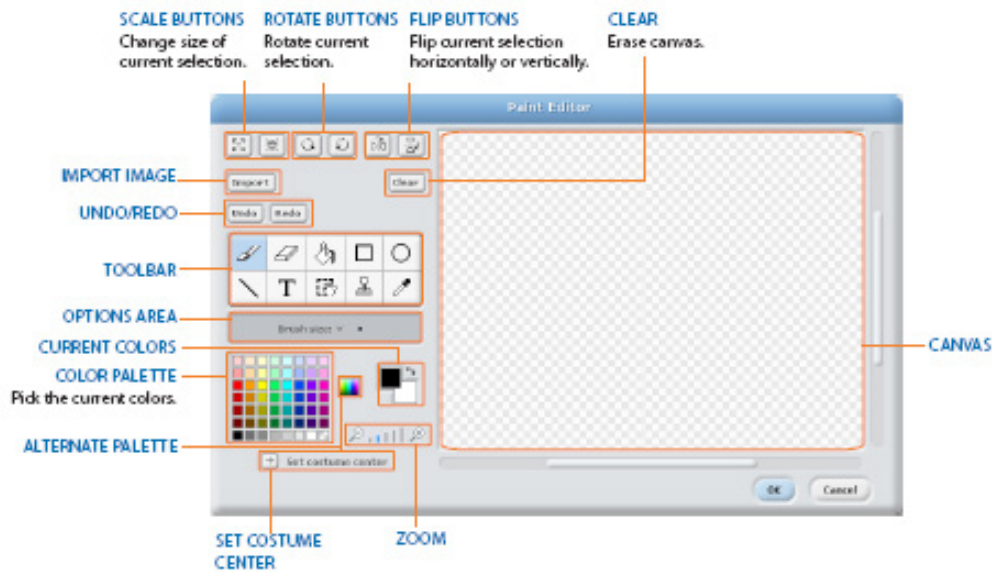
1. Stage
2. New Sprites
3. Sprite List

4. Blocks Palette dan Scripts Area
5. Costumes
6. Sounds
7. Current Sprite Info
8. Rotation Style
9. Toolbar
10. Menu
11. Green Flag
12. Paint Editor

Gambar 5 di bawah memperlihatkan antarmuka Scratch beserta komponen-komponennya seperti yang disebutkan di atas. Sementara gambar 6 memperlihatkan antarmuka Paint Editor Scratch yang dapat dimanfaatkan untuk mengubah *costumes* dan *backgrounds* yang digunakan dalam *project* Scratch.



Gambar 5. Antarmuka lingkungan Scratch [9]



Gambar 6. Antarmuka Paint Editor untuk membuat atau meng-edit costumes dan backgrounds [9]

C. Scratch Blocks

Scratch *blocks* diorganisasi ke dalam delapan kategori warna berbeda, yakni Motion, Looks, Sound, Pen, Control, Sensing, Operators, dan Variables [9].



Gambar 7. Kategori Blocks

1. Motion

Block motion berisikan blok-blok yang dapat digunakan untuk menggerakkan *sprites* dalam *project user*.

Motion	
	Moves sprite forward or backward.
	Rotates sprite clockwise.
	Rotates sprite counter-clockwise.
	Points sprite in the specified direction. (0=up, 90=right, 180=down, 270=left)

	Points sprite toward mouse-pointer or another sprite.
	Moves sprite to specified x and y position on Stage.
	Moves sprite to the location of the mouse-pointer or another sprite.
	Moves sprite smoothly to a specified position over specified length of time.
	Changes sprite's x-position by specified amount.
	Sets sprite's x-position to specified value.
	Changes sprite's y-position by specified amount.
	Sets sprite's y-position to specified value.
	Turns sprite in opposite direction when sprite touches edge of Stage.
	Reports sprite's x-position. (Ranges from -240 to 240)
	Reports sprite's y-position. (Ranges from -180 to 180)
	Reports sprite's direction. (0=up, 90=right, 180=down, 270=left)

Selain itu, terdapat *motor blocks* yang dapat ditampilkan jika opsi 'Show Motor Blocks' dalam menu Edit dipilih.

	Turns motor on for specified amount of time.
	Turns motor on.
	Turns motor off.
	Sets power of motor and turns motor on. (Ranges from 0 to 100)
	Sets or changes direction of motor, but does not turn motor on. (this way = clockwise; that way = counterclockwise; reverse = change direction)

2. Looks

Block ini dapat digunakan untuk mengganti *costume sprites*, *background stage*, ukuran *sprite*, dan sebagainya.

Looks	
	Changes sprite's appearance by switching to different costume.
	Changes sprite's costume to next costume in the costume list. (if at end of the costume list, jumps back to first costume.)
	Reports sprite's current costume number.
	Changes Stage's appearance by switching to a different background.

	Changes Stage's background to next background in the background list.
	Reports Stage's current background number.
	Displays sprite's speech bubble for specified amount of time.
	Displays sprite's speech bubble. (You can remove speech bubble by running this block without any text.)
	Displays sprite's thought bubble for specified amount of time.
	Displays sprite's thought bubble.
	Changes a visual effect on a sprite by specified amount. (Use pull-down menu to choose effect.)
	Sets a visual effect to a given number. (Most visual effects range from 0 to 100.)
	Clears all graphic effects for a sprite.
	Changes sprite's size by specified amount.
	Sets sprite's size to specified % of original size.
	Reports sprite's size, as % of original size.
	Makes sprite appear on the Stage.
	Makes sprite disappear from the Stage. (When sprite is hidden, other sprites cannot detect it with touching? block.)
	Moves sprite in front of all other sprites.
	Moves sprite back a specified number of layers, so that it can be hidden behind other sprites.

3. Sound

Blok-blok yang mengatur penggunaan efek suara dapat ditemukan dalam *sound block* ini.

	Starts playing a sound, selected from pull-down menu, and immediately goes on to the next block even as sound is still playing.
	Plays a sound and waits until the sound is finished playing before continuing with next block.
	Stops playing all sounds.
	Plays a drum sound, selected from pull-down menu, for specified number of beats.
	Plays a musical note (higher numbers for higher pitches) for specified number of beats.
	Rests (plays nothing) for specified number of beats.
	Sets the type of instrument that the sprite uses for play note blocks. (Each sprite has its own instrument.)
	Changes sprite's sound volume by specified amount. Volume ranges from 0 to 100.
	Sets sprite's sound volume to specified value.
	Reports sprite's sound volume.
	Changes sprite's tempo by specified amount.
	Sets sprite's tempo to specified value in beats per minute.
	Reports sprite's tempo in beats per minute.

4. Pen

Mengatur penggunaan *pen* dan *stamps* di *project*.

	Clears all pen marks and stamps from the Stage.
	Puts down sprite's pen, so it will draw as it moves.
	Pulls up sprite's pen, so it won't draw as it moves.
	Sets pen's color, based on choice from color picker. Picking the color also changes the pen shade.
	Changes pen's color by specified amount.
	Sets pen's color to specified value. (pen_color=0 at red end of rainbow, pen_color=100 at blue end of rainbow. Ranges from 0 to 200 to go around the color wheel.)
	Changes pen's shade by specified amount.
	Sets pen's shade to specified amount. (pen_shade=0 is very dark, pen_shade=100 is very light. Default is 50, unless set with color picker.)
	Changes pen's thickness.
	Sets pen's thickness.
	Stamps sprite's image onto the Stage.

5. Control

Blok ini berisikan sejumlah blok utama *script* agar dijalankan dan blok struktur kendali.

	Runs script below when green flag is clicked.
	Runs script below when specified key is pressed.
	Runs script below when sprite is clicked.
	Waits specified number of seconds, then continues with next block.
	Runs the blocks inside over and over.
	Runs the blocks inside a specified number of times.
	Sends a message to all sprites, then continues with the next block without waiting for the triggered scripts.
	Sends a message to all sprites, triggering them to do something, and waits until they all finish before continuing with next block.
	Runs script below when it receives specified broadcast message.
	Continually checks whether condition is true; whenever it is, runs the blocks inside.
	If condition is true, runs the blocks inside.
	If condition is true, runs the blocks inside the if portion; if not, runs the blocks inside the else portion.
	Waits until condition is true, then runs the blocks below.
	Checks to see if condition is false; if so, runs blocks inside and checks condition again. If condition is true, goes on to the blocks that follow.
	Stops the script.
	Stops all scripts in all sprites.

6. Sensing

Terkait hal-hal yang dapat dikerjakan *sprites*.

	Reports true if sprite is touching specified sprite, edge, or mouse-pointer. (Select from pull-down menu.)
	Reports true if sprite is touching specified color. (Click on color patch, then use eyedropper to select color.)
	Reports true if first color (within sprite) is touching second color (in background or another sprite). (Click on color patch, then use eyedropper to select color.)

	Asks a question on the screen and stores keyboard input in the answer variable. Causes the program to wait until the Enter key is pressed or check mark is clicked.
	Reports keyboard input from the most recent use of ask and wait . Shared by all sprites (global).
	Reports the x-position of mouse-pointer.
	Reports the y-position of mouse-pointer.
	Reports true if mouse button is pressed.
	Reports true if specified key is pressed.
	Reports distance from the specified sprite or mouse-pointer.
	Sets the timer to zero.
	Reports the value of the timer in seconds. (The timer is always running)
	Reports a property or variable of another sprite.
	Reports the volume (from 1 to 100) of sounds detected by the computer microphone.
	Reports true if computer microphone detects a sound volume greater than 30 (on scale of 1 to 100).
	Reports the value of specified sensor. To use this block, you need a sensor connected to your computer. You can use this with a Scratch sensor board (http://www.playfulinvention.com/picoboard.html) or with LEGO® WeDo™ (http://www.legoeducation.com).
	Reports true if specified sensor is pressed. To use this block, you need to have a Scratch sensor board connected to your computer. (See http://www.playfulinvention.com/picoboard.html)

7. Operators

Berisi berbagai operator aritmatika yang dapat digunakan dalam *project*.

Operators	
	Adds two numbers.
	Subtracts second number from first number.
	Multiplies two numbers.
	Divides first number by second number.
	Picks a random integer within the specified range.
	Reports true if first value is less than second.
	Reports true if two values are equal.
	Reports true if first value is greater than second.
	Reports true if both conditions are true.
	Reports true if either condition is true.
	Reports true if condition is false; reports false if condition is true.
	Concatenates (combines) strings.
	Reports the number of letters in a string.
	Reports the letter at the specified position in a string.
	Reports result of selected function (abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^) applied to specified number.
	Reports remainder from division of first number by second number.
	Reports closest integer to a number.

8. Variables

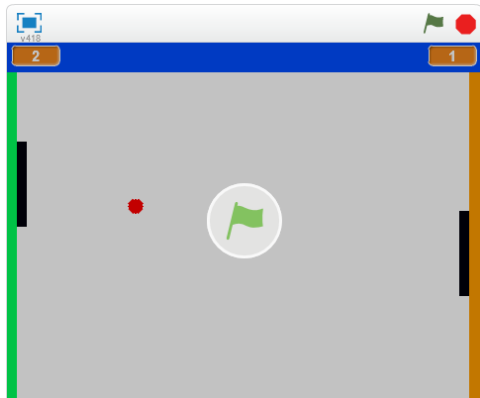
Berisi blok-blok yang dapat digunakan untuk membuat variabel dan *list*, maupun menghapusnya.

Variables	
	Click to create and name a new variable. When you create a variable for the first time, the variable blocks will appear. You choose whether the variable is for all sprites (global) or just for one sprite (local).
	Deletes all blocks associated with a variable.
	Reports value of the variable.
	Changes the variable by specified amount. If you have more than one variable, use the pull-down menu to select the variable name.
	Sets the variable to specified value.
	Shows the variable monitor on the stage.
	Hides the variable monitor so it is not visible on the stage.
	Click to create and name a new list. When you create a list for the first time, the list blocks will appear. You can choose whether the list is for all sprites (global) or just for one sprite (local).
	Deletes the blocks associated with the list.
	Reports all the items in the list. (The items are separated by spaces. However, if the items are individual letters or digits, spaces are omitted.)
	Adds the specified item to the end of the list. The item can be a number or a string of letters and other characters.
	Deletes one or all items from a list. You can specify which item to delete by choosing from the pull-down menu or typing in a number. Choosing "first" deletes the first item in the list. Choosing "all" deletes everything from the list. Deleting decreases the length of the list.
	Inserts an item at the specified position in the list. You can specify where to insert the item by choosing from the pull-down menu or typing in a number. Choosing "any" inserts at a random place in the list. Choosing "last" adds the item to the end of the list. The length of the list increases by 1.
	Replaces an item in the list with the specified value. You can specify which item to replace by choosing from the pull-down menu or typing in a number. Choosing "any" replaces a random item in the list. The length of the list does not change.
	Reports the item at the specified position in the list. You can specify which item to report by choosing from the pull-down menu or typing in a number. Choosing "any" reports a random item in the list.
	Reports how many items are in the list.
	Reports true if the list contains the specified item. The item must match exactly to report true.

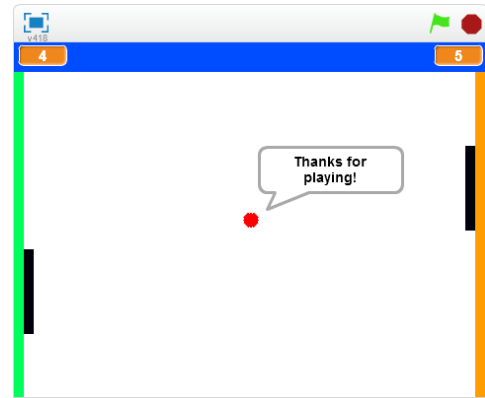
III. PING PONG GAME

Sebagaimana yang telah dijelaskan dalam bab Pendahuluan, Scratch dapat digunakan untuk membuat game interaktif. Menurut Sadigh [10], terdapat lima elemen utama dari suatu permainan komputer, yakni *graphics* (grafik), *sound* (suara), *interface* (antarmuka), *gameplay* (alur permainan), dan *story* (kisah). Scratch dengan grafik 2D dan efek suara yang didukungnya, antarmuka yang interaktif, serta *gameplay* dan *story* yang dapat dengan mudah diatur oleh *programmer*-nya menggunakan blok-blok yang disediakan, dapat menjawab seluruh kebutuhan elemen dalam permainan komputer tersebut.

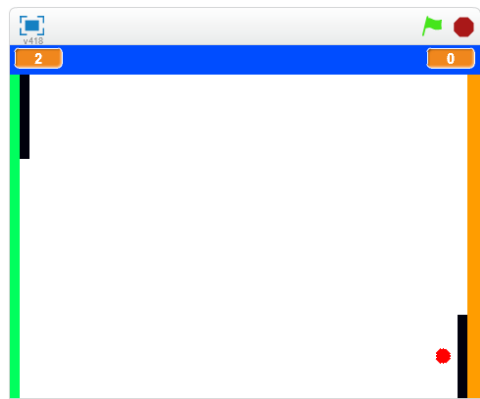
Berikut disajikan salah satu contoh permainan interaktif populer, yakni Ping Pong *game*, yang dibuat dengan menggunakan Scratch. Permainan ini dibuat oleh salah seorang Scratcher dengan nama alias kingminer29, dan dapat dimainkan secara langsung melalui tautan <http://scratch.mit.edu/projects/22791282?fromexplore=true> [11].



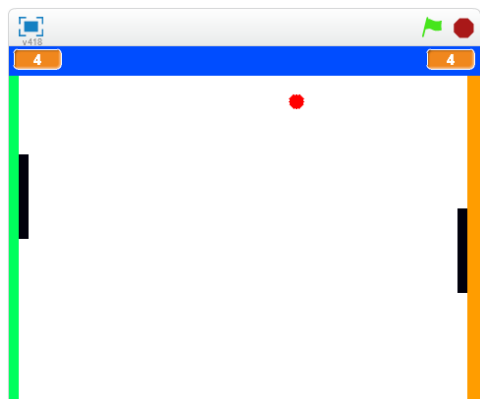
Gambar 8. Laman awal permainan ping pong



Gambar 11. Permainan berakhir saat salah seorang pemain memperoleh skor 5



Gambar 9. Permainan ping pong antara dua pemain



Gambar 10. Bola ping pong memantul sepanjang dinding meja permainan

Selain permainan interaktif, Scratch juga dapat digunakan untuk membuat beragam aplikasi lainnya, seperti animasi, seni, musik, cerita, dan sebagainya. Kumpulan *projects* yang telah dihasilkan dapat dilihat pada situs resmi Scratch secara langsung [12].

IV. SIMPULAN

Scratch sebagai bahasa pemrograman *visual* yang populer menawarkan sejumlah kemudahan dan fitur yang menarik bagi para pemula dalam mempelajari cara pemrograman. 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 [13].

Dalam makalah ini, sejarah perkembangan, antarmuka dan blok pembangun sebagai konsep dasar yang perlu diketahui oleh para pemula Scratch telah dijelaskan. Salah satu contoh permainan interaktif yang dikembangkan dengan menggunakan Scratch juga telah diperlihatkan. Dengan demikian, diharapkan dapat memberikan gambaran kepada para pembaca mengenai cara pemanfaatan dan pentingnya Scratch dalam budaya komputasional yang dihadapi oleh manusia masa kini.

DAFTAR PUSTAKA

- [1] Getting Started with SCRATCH Version 1.4, <http://scratch.mit.edu>, Lifelong Kindergarten Group, MIT Media Lab.
- [2] <http://scratch.mit.edu/educators/>, diakses 27 Mei 2014.
- [3] <http://scratched.media.mit.edu/>, diakses 27 Mei 2014.
- [4] Brennan, K.A., 2013, Best of Both Worlds: Issues of Structure and Agency in Computational Creation, In and Out of School, Dissertation, School of Architecture and Planning, Massachusetts Institute of Technology.
- [5] Chiang, J., Shall We Learn Scratch Programming, <http://shallwelearn.com>.
- [6] http://wiki.scratch.mit.edu/wiki/Development_of_

Scratch_1.0, diakses 27 Mei 2014.

- [7] http://wiki.scratch.mit.edu/wiki/Scratch_Versions, diakses 27 Mei 2014.
- [8] <http://scratch.mit.edu/statistics/>, diakses 27 Mei 2014.
- [9] Reference Guide: Scratch Version 1.4, <http://scratch.mit.edu>, Lifelong Kindergarten Group, MIT Media Lab.
- [10] Sadigh, M., 2002, How to Design a Computer Game?, Final Project, University of Washington.
- [11] <http://scratch.mit.edu/projects/22791282/?fromexplore=true>, diakses 28 Mei 2014.
- [12] <http://scratch.mit.edu/explore>, diakses 28 Mei 2014.
- [13] 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.