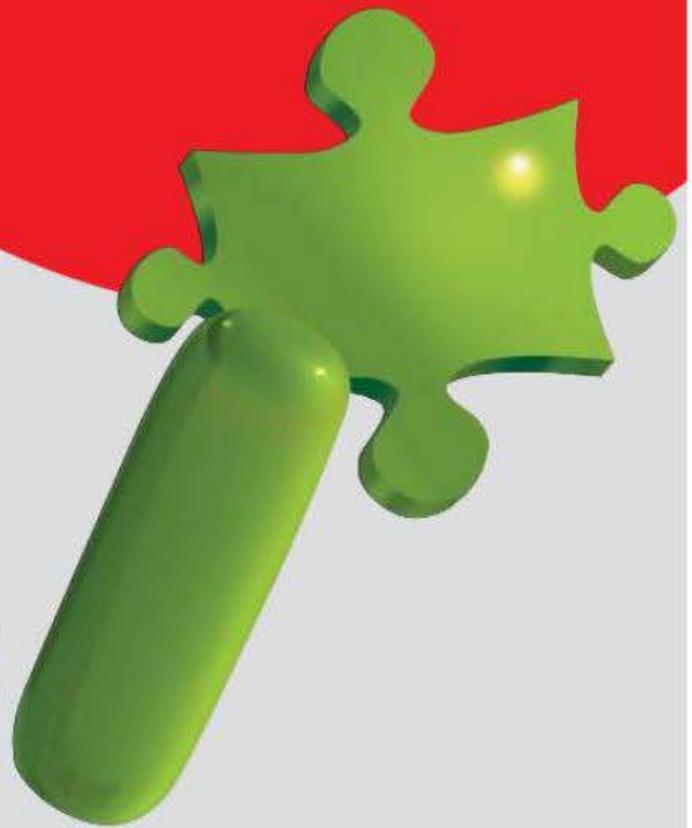


Membuat  
Aplikasi

# Android

Tanpa Coding  
dengan

APP Inventor





# **Membuat Aplikasi Android Tanpa Coding dengan App Inventor**

Sanksi Pelanggaran Pasal 72:  
Undang-Undang Nomor 19 Tahun 2002  
Tentang Hak Cipta

1. Barangsiapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 Ayat (1) atau Pasal 49 Ayat (1) dan Ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp 5.000.000.000,00 (lima milyar rupiah).
2. Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta atau hak terkait sebagai dimaksud pada Ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

# Membuat Aplikasi Android Tanpa Coding dengan **App Inventor**

**Wahana Komputer**

*PENERBIT PT ELEX MEDIA KOMPUTINDO*



*KOMPAS GRAMEDIA*

**Membuat Aplikasi Android Tanpa Coding dengan App Inventor**  
Wahana Komputer

© 2013, PT Elex Media Komputindo, Jakarta  
Hak cipta dilindungi undang-undang  
Diterbitkan pertama kali oleh  
Penerbit PT Elex Media Komputindo  
Kelompok Gramedia, Anggota IKAPI, Jakarta 2013

121130165 ISBN: 978-602-02-0480-2
--------------------------------------

*EISBN : 978-602-04-2507-8*

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta  
Isi di luar tanggung jawab percetakan

# PRAKATA

**J**ika Anda hendak belajar membuat aplikasi untuk platform Android, namun Anda masih awam dengan bahasa pemrograman yang dipakai, maka buku ini jawabannya. Buku ini akan mengajarkan kepada Anda tentang sebuah kemudahan dalam membuat aplikasi Android tanpa coding menggunakan App Inventor. Lalu, apa App Inventor itu?

App Inventor merupakan sebuah tool untuk membuat aplikasi android menggunakan visual block programming yang sangat mengagumkan. Mengapa disebut mengagumkan? ini karena dalam membuat sebuah aplikasi Android Anda tidak perlu mengetikkan satu pun code program. Anda hanya perlu melakukan drag drop atau menyusun block yang merupakan simbol perintah dan fungsi mirip dengan permainan puzzle.

Melalui buku ini penulis akan mengajarkan kepada Anda betapa mudahnya membuat aplikasi Android menggunakan App Inventor. Buku yang disajikan secara ringan dan berbobot ini, akan mengupas secara step by step pemrograman Android menggunakan App Inventor serta dilengkapi dengan contoh aplikasi yang dibuat.

## Struktur Penulisan Buku

Buku ini mengajarkan kepada Anda tentang pembuatan aplikasi Android menggunakan App Inventor, lengkap dengan contoh proyek-projek aplikasi. Agar mudah memahami materi yang dikupas, buku ini dibagi menjadi 5 bab yaitu sebagai berikut:

### Bab 1 Apa itu App Inventor?

Bab ini merupakan pengenalan tentang App Inventor sebelum Anda lebih jauh menggunakannya untuk membuat aplikasi Android. Pada bab ini akan dijelaskan tentang bahasa block untuk aplikasi mobile, arsitektur pemrograman dengan App Inventor, hingga istilah-istilah pemrograman menggunakan App Inventor.

## **Bab 2 Menggunakan IDE App Inventor**

Bab ini menjelaskan IDE App Inventor untuk merakit aplikasi Android menggunakan App Inventor. Pada bab ini dijelaskan tentang komponen desainer, block editor, sampai penggunaan emulator untuk menjalankan aplikasi Android yang sudah dibuat.

## **Bab 3 Membuat Aplikasi Pilih Gambar**

Pada bab ini Anda akan belajar membuat aplikasi pilih gambar menggunakan App Inventor. Pembahasan pada bab ini dimulai dengan membuat proyek dan menyusun interface, serta mengatur block program.

## **Bab 4 Membuat Aplikasi Slideshow**

Pada bab ini Anda akan belajar membuat aplikasi slidehshow. Aplikasi slideshow yang akan dibuat pada bab ini terdiri dari 3 macam, yaitu aplikasi slideshow otomatis, aplikasi slideshow manual, dan aplikasi slideshow random.

## **Bab 5 Membuat Aplikasi Web Browser**

Bab ini menjelaskan kepada Anda tentang pembuatan aplikasi Web Browser. Pembahasan dimulai dengan membuat proyek dan desain interface, dan dilanjutkan dengan menyusun block program.

## **Apa yang Harus Anda Kuasai?**

Agar dapat memahami materi yang disajikan, disarankan Anda terlebih dahulu mempunyai pengetahuan dasar tentang penggunaan aplikasi Android. Selain itu, diharapkan Anda sudah memiliki dasar pengetahuan tentang logika program.

## **Bagi Para Pembaca**

Walaupun dalam menyusun buku ini kami telah melakukan pengkajian dan penelitian yang mendalam serta berusaha untuk menyampaikan materi secara lengkap dan terstruktur, tentunya setiap karya tidaklah ada yang benar-benar sempurna sehingga mungkin buku ini kurang dapat memenuhi kebutuhan para

pembaca, atau mungkin para pembaca masih mengalami kesulitan atau masalah setelah mempelajari buku ini.

Untuk itu, Wahana Komputer membuka lebar-lebar kesempatan bagi para pembaca pada khususnya dan masyarakat pengguna komputer pada umumnya untuk melakukan konsultasi mengenai berbagai kesulitan yang dihadapi khususnya mengenai apa yang telah dijelaskan di dalam buku ini melalui situs web kami atau via pos, faks, ataupun email. Di samping itu Anda dapat memanfaatkan pula layanan pelatihan komputer kami melalui Lembaga Pendidikan Komputer Wahana untuk lebih mendalami materi-materi yang dipaparkan di dalam buku ini.

### **Wahana Komputer**

JL. MT Haryono 637 Semarang

Telp. (024) 8314727, 8413238 faks. (024) 8413964

email : [litbang@wahanakom.com](mailto:litbang@wahanakom.com)

web : [www.wahanakom.com](http://www.wahanakom.com)



# DAFTAR ISI

Kata Pengantar .....	v
<b>BAB 1 APA ITU APP INVENTOR?.....</b>	<b>1</b>
MEMULAI APP INVENTOR .....	2
MENGENAL BAHASA BLOCK UNTUK APLIKASI MOBILE .....	7
ARSITEKTUR .....	8
KOMPONEN .....	9
BEHAVIOR .....	10
EVENT .....	12
EVENT USER .....	12
EVENT INISIALISASI .....	13
EVENT TIMER .....	13
EVENT ANIMASI .....	13
EVENT EKSTERNAL .....	14
ISTILAH-ISTILAH PEMROGRAMAN .....	16
ALOKASI SLOT MEMORI .....	16
PROPERTIES .....	17
<b>BAB 2 MENGGUNAKAN IDE APP INVENTOR .....</b>	<b>21</b>
KOMPONEN DESIGNER .....	21
PALLETE .....	22
VIEWER .....	44
COMPONENTS .....	45
BLOCK EDITOR .....	49
EMULATOR .....	52
MEMBUAT APLIKASI “HALO DUNIA” .....	54

	MEMBUAT APLIKASI KLIK SPRITE BALL .....	64
<b>BAB 3</b>	<b>MEMBUAT APLIKASI PILIH GAMBAR .....</b>	<b>75</b>
	MEMBUAT PROJEK DAN USER INTERFACE .....	75
	MENYUSUN BLOCK PROGRAM .....	77
<b>BAB 4</b>	<b>MEMBUAT APLIKASI SLIDESHOW .....</b>	<b>85</b>
	MEMBUAT APLIKASI SLIDESHOW OTOMATIS .....	85
	MEMBUAT APLIKASI SLIDESHOW MANUAL .....	109
	MEMBUAT APLIKASI SLIDESHOW RANDOM .....	119
<b>BAB 5</b>	<b>MEMBUAT APLIKASI WEB BROWSER .....</b>	<b>123</b>
	MEMBUAT PROJECT DAN DESAIN WEB BROWSER .....	123
	MENYUSUN BLOCK PROGRAM .....	125

# 1

## APA ITU APP INVENTOR?

---

*Bab ini membahas:*

- ❖ Tentang App inventor.
- ❖ Bahasa Block Untuk Aplikasi Mobile.
- ❖ Arsitektur.
- ❖ Komponen.
- ❖ Behavior.
- ❖ Event.
- ❖ Istilah-Istilah Pemrograman.

---

**A**ndroid saat ini merupakan sistem operasi yang paling banyak digunakan pada ponsel mulai dari kelas *low end* hingga *high end*. Ini dikarenakan kebijakan yang diterapkannya sebagai *software* bebas dan terbuka (*open source*), sehingga siapapun boleh menggunakannya untuk menjalankan ponsel mereka.

Sebagai OS terbuka, pihak Google sebagai pengembang Android juga menyediakan aplikasi. Bahkan Google juga menyediakan aplikasi yang berbasis web untuk membuat aplikasi Android dengan nama App Inventor. Dengan aplikasi ini, membuat aplikasi Android menjadi lebih menarik seolah sedang memasang puzzle.



*Gambar 1.1. Logo dari MIT App Inventor*

Sebagai aplikasi pengembang, App Inventor merupakan bahasa pemrograman berbasis visual. Bahkan bisa dibilang hampir keseluruhan pembuatan aplikasi dilakukan hanya dengan men-*drag* dan men-*drop* saja. Untuk dapat mengakses dan membuat aplikasi melalui App Inventor, Anda cukup melakukan registrasi untuk memperoleh akun Google. Dan yang perlu Anda perhatikan, untuk memperoleh kompatibilitas yang baik ketika menggunakan App Inventor, disarankan Anda menggunakan browser Google Chrome.

Menurut Google, dengan App Inventor, seorang yang sangat awam dengan bahasa pemrograman pun bisa membuat aplikasi Android secara mudah dan cepat.

Secara umum AppInventor memiliki dua komponen utama, yaitu sebagai berikut:

- **Design View**, bagian ini memiliki semua komponen yang diperlukan untuk mendesain aplikasi.
- **Block Editor**, berfungsi untuk mengkombinasikan blok-blok sehingga bisa menjadi sebuah aplikasi Android.

## Memulai App Inventor

Sebelum bisa menggunakan App Inventor, Anda diwajibkan memiliki akun Gmail. Jika belum memiliki akun Gmail, Anda bisa melakukan pendaftaran dengan mengakses URL <http://www.gmail.com>. Setelahnya lakukan langkah-langkah sebagai berikut:

1. Buka browser (disarankan menggunakan Google Chrome) Anda kemudian login menggunakan akun Google Mail yang sudah Anda miliki.
2. Selanjutnya, akses URL <http://beta.appinventor.mit.edu/> sehingga ditampilkan halaman berikut:

## Google accounts

The application MIT AppInventor Experimental is requesting permission to access your Google Account.

Please select an account that you would like to use.

● edyandroid86@gmail.com

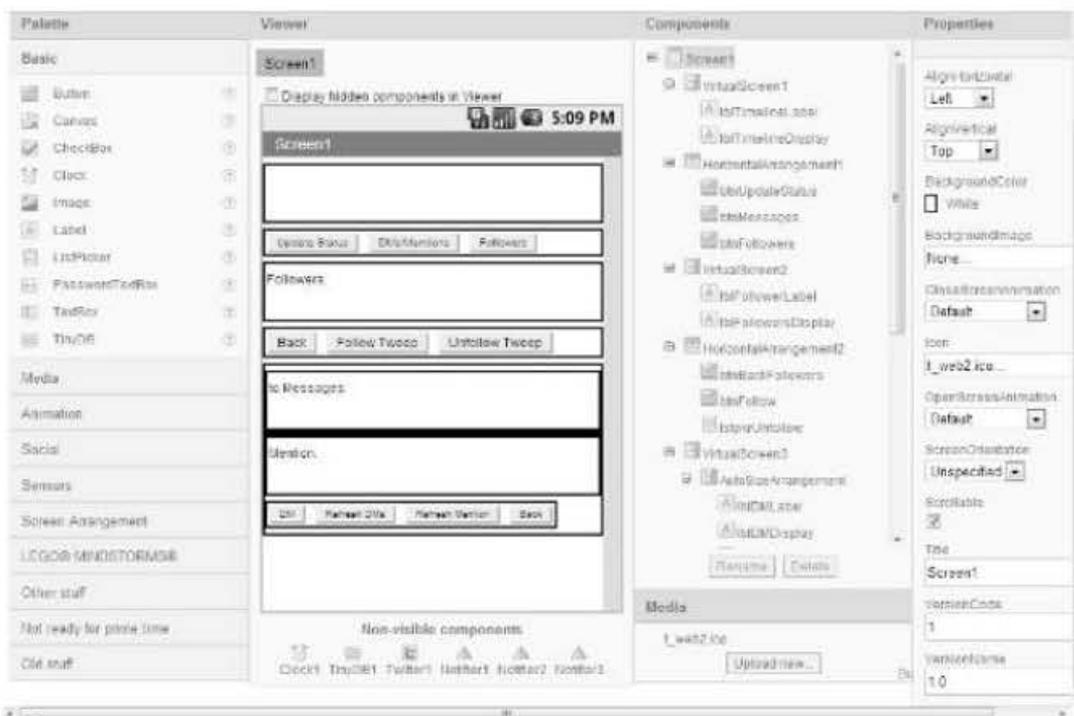
Google is not affiliated with the contents of MIT AppInventor Experimental or its owners. If you sign in, Google will use your account to help improve our products and services, but we will not use your password or any other personal information.

[Sign in to another account](#)

Remember this approval for the next 30 days

*Gambar 1.2. Permintaan izin mengakses akun Google*

3. Klik tombol **Allow/Izinkan**.
4. Sampai dengan langkah ini Anda sudah masuk ke jendela kerja utama App Inventor seperti yang ditunjukkan pada gambar dibawah ini.



*Gambar 1.3. Jendela kerja utama App Inventor*

5. Berikutnya, download Java 6 dari URL <http://www.java.com/en/> dan kemudian lakukan instalasi.



*Gambar 1.4. Download Java 6 (1.6)*

6. Setelah Java 6 terinstall di komputer Anda, kini download **App Inventor Setup for Windows** dengan mengakses URL <http://beta.appinventor.mit.edu/learn/setup/setupwindows.html> dan lakukan instalasi di komputer Anda.
7. Pada kotak dialog **Welcome to the AppInventor Setup Wizard**, klik **Next**.



*Gambar 1.5. Welcome to the AppInventor Setup Wizard*

8. Muncul **License Agreement** yang menjelaskan perjanjian lisensi dari produk ini. Klik tombol **I Agree** untuk menyetujui lisensi dan melanjutkan proses instalasi.



*Gambar 1.6. Klik pada I Agree untuk menyetujui lisensi*

9. Berikutnya tentukan lokasi instalasi. Standarnya berada di C:\Program Files\. Tapi bisa diubah dengan klik pada **Browse**.



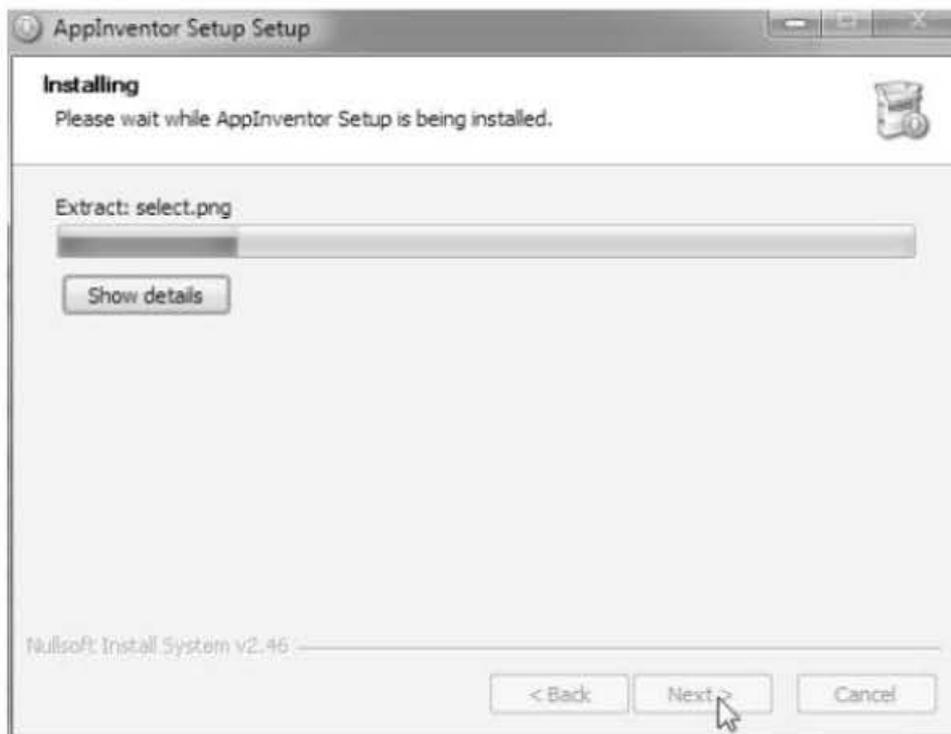
*Gambar 1.7. Memilih lokasi instalas di Choose Install Location*

10. Tentukan folder **Start Menu** yang akan menentukan lokasi Start Menu di kotak dialog **Choose Start Menu Folder**.



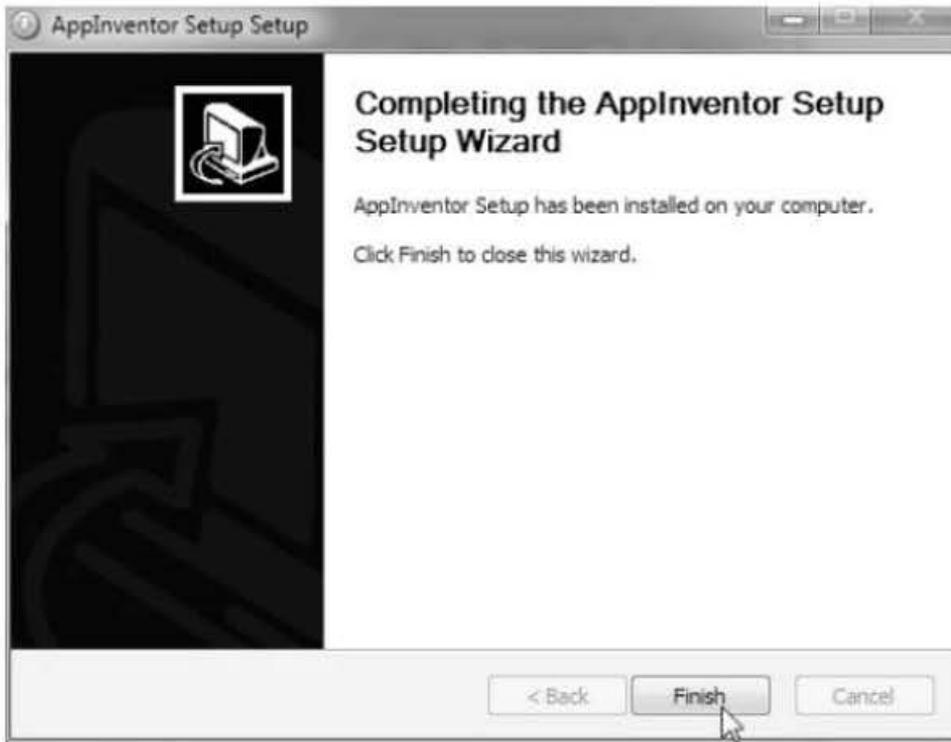
*Gambar 1.8. Choose Start Menu Folder*

11. Tunggu hingga proses instalasi selesai.



*Gambar 1.9. Proses instalasi berlangsung*

12. Pada kotak dialog **Completing the AppInventor Setup Wizard**, klik **Finish**.



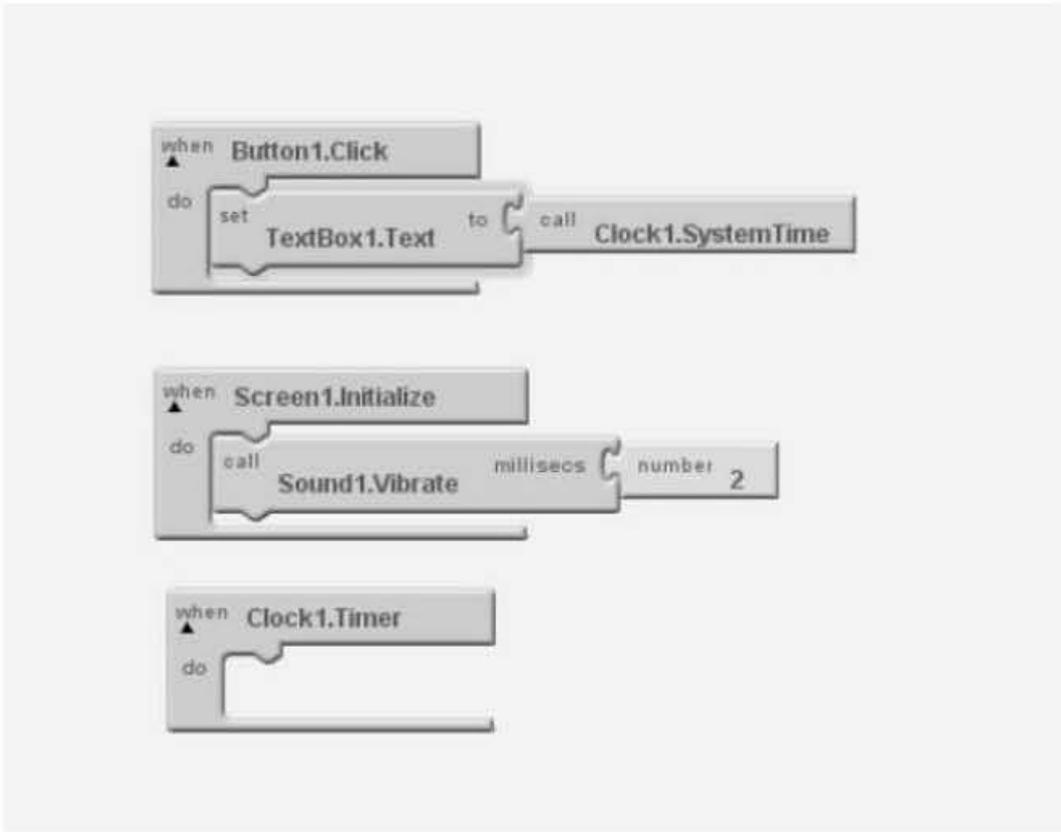
*Gambar 1.10. Proses instalasi telah selesai*

13. Jika Anda memiliki ponsel Android dan ingin menggunakannya untuk mencoba proyek yang Anda buat menggunakan App Inventor, Anda bisa melakukan instalasi dan setting driver ponsel Anda.

## **Mengenal Bahasa Block untuk Aplikasi Mobile**

Pada App Inventor, salah satu fitur utamanya adalah memudahkan seseorang membuat program tanpa harus mengetikkan kode sedikitpun. Ini bisa diakomodasi menggunakan bahasa block.

Menggunakan bahasa block, pengkodean dilakukan di belakang layar. Semuanya menggunakan antarmuka yang memudahkan bernama block/blok. Jadi blok ini merepresentasikan kode secara grafik. Cara Anda memasang block ini mirip dengan memasang puzzle gambar, tapi sebenarnya cara memasang block ini akan sesuai dengan kode yang dihasilkan.



Gambar 1.11. Contoh kode-kode dari Puzzle yang menentukan apa yang akan dilakukan

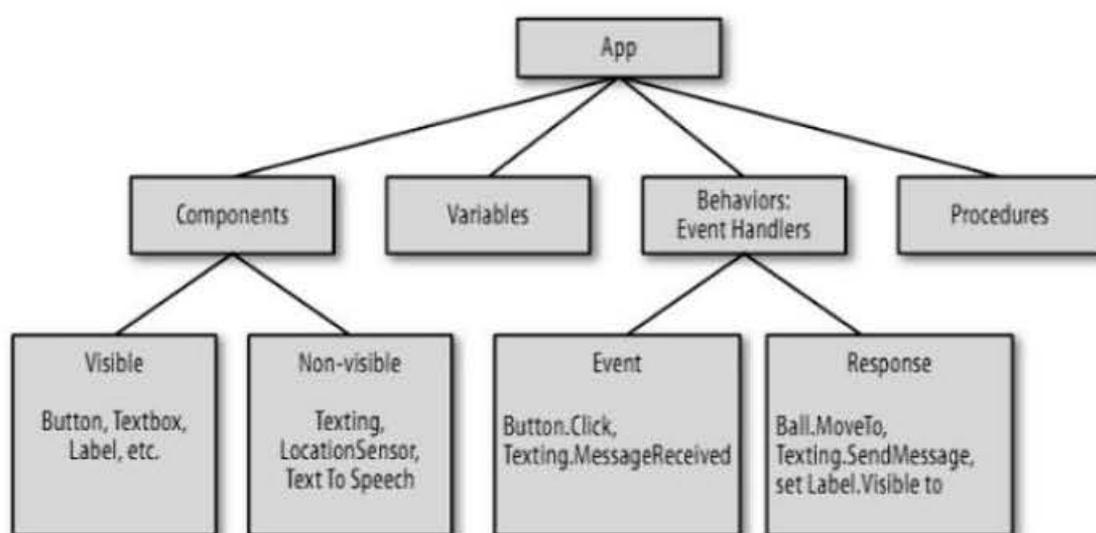
## Arsitektur

Apakah sebenarnya sebuah program itu? Banyak orang yang bisa menjelaskan arti sebuah program dari sisi pengguna, tapi jarang yang bisa menjelaskan apa arti sebuah program dari sisi programmer. Ini karena hanya programmer yang mengerti struktur dalam dari sebuah program.

Bagian internal sebuah program sebenarnya bisa dikategorikan menjadi dua, yaitu komponen dan perilaku/behavior-nya. Ini perlu dipahami agar Anda bisa meng-*create* program dengan lebih efektif.

Kedua bagian, komponen dan perilaku ini di App Inventor memiliki dua jendela tersendiri. Anda menggunakan Component Designer untuk membuat objek/komponennya dan menggunakan Block Editor untuk memprogram perilaku software terhadap event, baik event yang dihasilkan oleh user atau oleh event eksternal.

Berikut ini contoh arsitektur internal dari sebuah aplikasi di AppInventor:



**Gambar 1.12.** Arsitektur dari sebuah program di AppInventor

## Komponen

Komponen adalah bagian fisik dari aplikasi. Ada 2 jenis komponen di App Inventor, dan juga di Android pada umumnya, yaitu aplikasi yang terlihat/visible dan yang tidak terlihat/non-visible.

Komponen visible atau komponen yang terlihat adalah komponen-komponen yang bisa dilihat mata ketika sebuah aplikasi di-*launching*. Ini contohnya adalah button, textbox, label, dan screen. Ini sering disebut sebagai user interface.

Komponen yang non-visible adalah komponen yang tidak terlihat, jadi bukan termasuk bagian user interface. Tapi komponen ini menyediakan akses ke fungsi-fungsi *built-in* dari perangkat. Contoh dari komponen non-visible ini adalah Texting yang dipakai untuk mengirim dan memproses teks SMS. Kemudian komponen LocationSensor yang menentukan lokasi perangkat. Ada juga komponen TextToSpeech yang digunakan untuk membantu membaca teks.

Komponen-komponen non-visible ini merupakan komponen yang penting bagi program. Dan ini merupakan teknologi dari perangkat yang digunakan. Baik komponen visible dan non-visible memiliki sifat tertentu yang disebut properties.

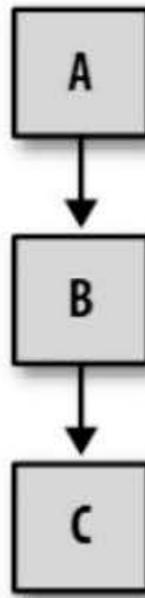
Properties adalah informasi mengenai komponen-komponen tertentu. Contohnya untuk komponen yang visible, memiliki properti seperti Width, Height, Color dan sebagainya. Properties ini akan menentukan bagaimana tampilan komponen yang akan Anda buat.

Properti ini bisa diedit di bagian **Component Designer** untuk menentukan tampilan awal dari komponen. Jika Anda mengubah properti ini, maka tampilan juga akan berubah. Misalnya kalau Anda mengubah Width dari 40 ke 50, maka tampilan objek akan menjadi lebih lebar, baik di designer atau di aplikasi.

## Behavior

Komponen aplikasi mudah untuk dipahami. Sebuah Textbox contohnya berfungsi untuk menerima inputan dari user, lalu Button gunanya untuk diklik. Sementara behavior atau perilaku dari aplikasi ini lebih susah dipahami karena umumnya bersifat kompleks.

Behavior ini menentukan bagaimana aplikasi akan merespon event, baik event yang dijalankan oleh user (seperti klik button), atau aplikasi eksternal (seperti adanya SMS ke ponsel). Nah, yang membuat pemrograman dikenal sulit adalah menentukan interaksi behavior ini. Namun untungnya AppInventor memudahkan Anda mengatur behavior karena Anda menggunakan blok-blok seperti puzzle untuk menentukan behavior tersebut.

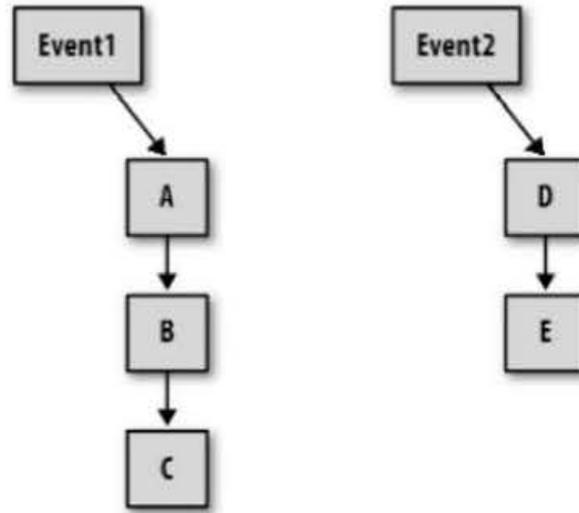


*Gambar 1.13. Flow dalam sebuah program konvensional*

Di awal-awal munculnya sejarah pemrograman, software sering kali dibandingkan sebagai kegiatan memasak. Dimana ketika memasak, Anda melakukan kegiatan linear, dari satu tahap ke tahap berikutnya. Step by step, langkah demi langkah secara teratur. Ini terjadi di dunia pemrograman berbasis console dimana ketika awal mula komputer, hampir semua komputer menggunakan antarmuka console.

Tapi ketika sudah muncul era GUI (graphical user interface) atau antarmuka grafis, maka pendekatan behavior ini tidak lagi menggunakan pendekatan resep step by step tapi menggunakan pendekatan event handler. Yaitu behavior akan dimulai kalau ada event. Jadi behavior ini akan bereaksi terhadap event, seperti tombol diklik atau lain sebagainya.

Misalnya ketika tombol diklik, SMS akan dikirimkan, atau file audio akan dimainkan. Jadi behavior ini terdiri dari set komponen yang berkorelasi terhadap event. Tetap saja ada step by step di langkah ini, hanya saja, urutan step by step ini hanya dijalankan ketika merespon sebuah event saja.



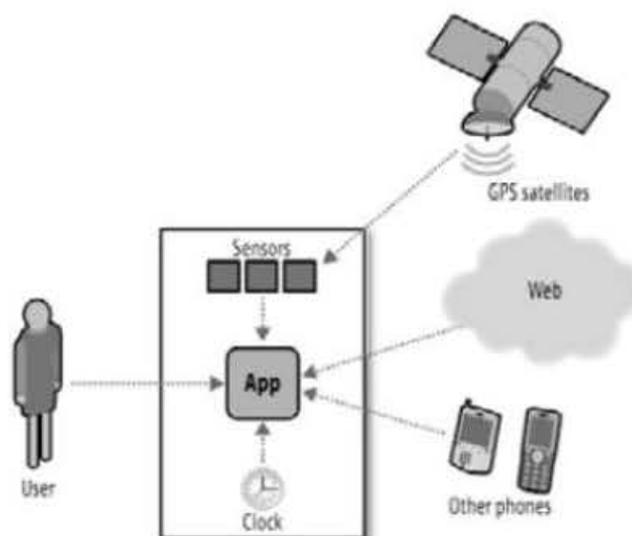
*Gambar 1.14. Contoh event yang memicu munculnya urutan step by step*

Ketika event terjadi, aplikasi akan bereaksi dengan memanggil beberapa fungsi. Fungsi adalah apa yang bisa Anda lakukan menggunakan komponen seperti mengirim SMS atau mengubah properti.

Ketika memanggil fungsi, Anda sebenarnya mengaktifkan fungsi tersebut. Jadi kalau programmer memanggil event, maka fungsi untuk event tersebut akan merespon event handler yang bersangkutan.

Ketika event terjadi, aplikasi akan bereaksi dengan memanggil urutan fungsi. Fungsi adalah apa yang Anda inginkan untuk beraksi. Operasi-operasi seperti mengirim SMS atau mengubah properti di label adalah sebuah fungsi.

Event kebanyakan diawali oleh end user, tapi lainnya tidak seperti itu. Ada yang ditrigger oleh perangkat ponsel, seperti perubahan orientasi sensor, atau GPS dari satelit.



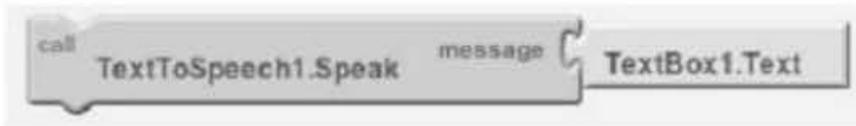
*Gambar 1.15. Aplikasi bisa merespon baik event internal atau eksternal*

Salah satu keunggulan pemrograman menggunakan AppInventor adalah karena aplikasi ini menggunakan paradigma event-response, sehingga Anda menggunakan blok-blok untuk mengatur apa yang akan dilakukan ketika event terjadi. Misalnya untuk membuat event yang jika diklik akan mengakses TextToSpeech dengan teks tertentu, cukup dibuat seperti block dibawah, tanpa harus mengetikkan banyak koding.



*Gambar 1.16. Event handler untuk aplikasi yang memakai TextToSpeech*

Dengan AppInventor, semua aktivitas yang terjadi merespon event yang muncul. Sehingga kalau tidak ada event melalui blok When-do, maka block tidak akan dieksekusi sama sekali.



*Gambar 1.17. Contoh blok yang mengambang tidak akan dieksekusi*

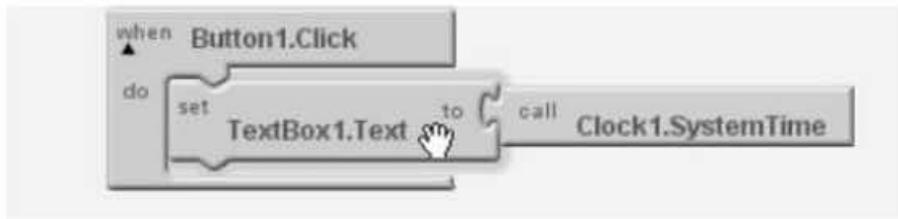
## Event

Event pada AppInventor memiliki beberapa tipe yang bisa dibagi menjadi beberapa kategori, yaitu sebagai berikut:

- Event yang dimulai oleh user, misalnya When user mengklik tombol button, do
- Event Inisialisasi: misalnya When Aplikasi launching, do
- Event Timer: misalnya When 10 milidetik lewat, do
- Event animasi: misalnya When dua objek bersentuhan, do
- Event eksternal: misalnya When ponsel menerima SMS, do

### Event User

Event pertama adalah event yang dimulai oleh user. Ini merupakan tipe event yang paling lazim. Contohnya event klik pada tombol/button yang akan memicu respon dari aplikasi. Untuk ponsel dengan tampilan grafis, contohnya adalah aplikasi yang merespon ke sentuhan dan gerakan.



*Gambar 1.18. Button click*

## Event Inisialisasi

Kemungkinan Anda perlu menjalankan fungsi tertentu, tepat ketika aplikasi dimulai. Dan bukannya saat merespon event-event tertentu. Anda bisa menggunakan method initialize yang memudahkan untuk meng-capture event initialization ini.

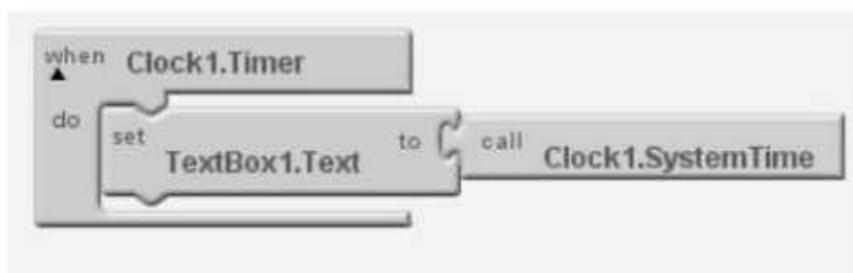
Bahasa AppInventor menganggap event launch ini sebagai sebuah event. Jika Anda ingin fungsi tertentu dilakukan tepat ketika aplikasi terbuka, Anda tinggal men-drag event seperti Screen1.Initalize dan meletakkan block lainnya di sana.



*Gambar 1.19. Contoh event Inisialisasi yang dipakai*

## Event Timer

Beberapa aktivitas dalam aplikasi, sering juga diinisialisasi oleh berlalunya waktu. Misalnya aplikasi alarm atau jam. Ini bisa diakomodasi menggunakan event Timer.



*Gambar 1.20. Contoh event timer bekerja*

## Event Animasi

Event animasi berkaitan dengan event objek grafis yang ada di dalam canvas.

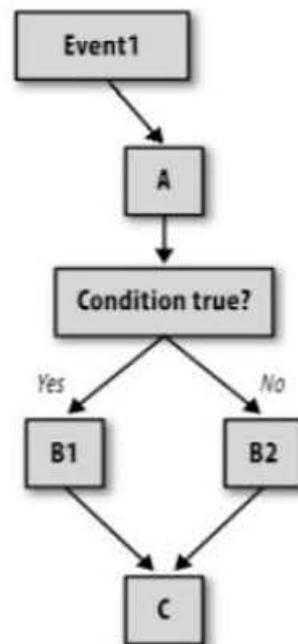
## Event Eksternal

Contoh ini adalah ketika ponsel Anda menerima informasi dari satelit GPS, atau ketika ponsel menerima teks. Maka event ini akan dipicu. Input eksternal tersebut akan mirip dengan event biasa seperti ketika user mengklik button. Jadi semua aplikasi pada prinsipnya adalah event handler, dimana tugasnya untuk menangani event. Beberapa event ini dipicu oleh input user, beberapa dipicu oleh timer/waktu, dan beberapa lagi dipicu oleh event eksternal. Tugas Anda sebagai seorang programmer adalah mengkonsepkan aplikasi dan mendesain apa yang akan dilakukan terhadap event tersebut.



Gambar 1.21. Contoh event handler dalam sebuah program

Sebuah event handler tidak harus berupa hal-hal yang linear, seperti melakukan sesuatu, menampilkan teks atau lainnya. Tapi event handler juga bisa meminta input dari user, misalnya menggunakan pencabangan atau kondisional.



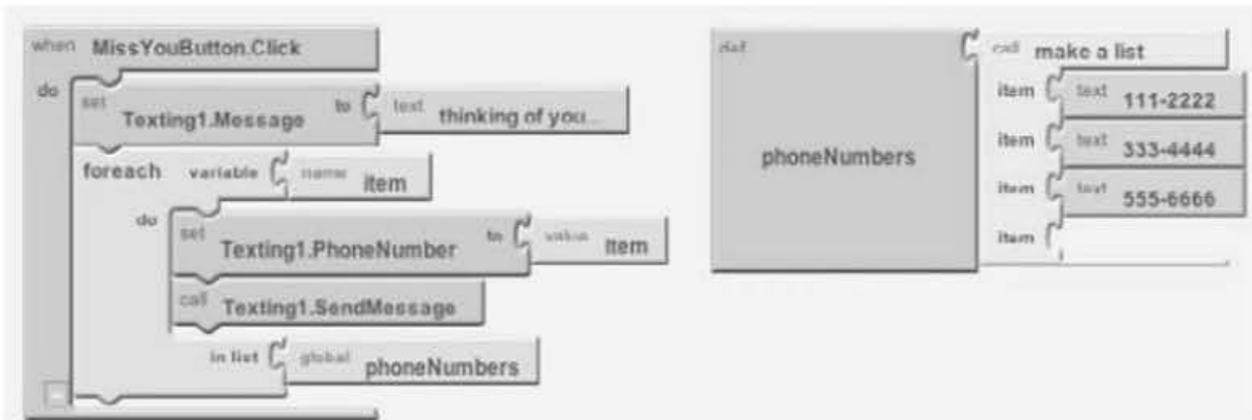
Gambar 1.22. Contoh event handler yang menangani event dengan kondisional

Pada tes kondisional diatas berisi data boolean, apakah Yes atau True. Misalnya menanyakan apakah nilai mencapai 1000? Atau apakah SMS sudah diterima? Tes boolean ini juga bisa berisi operator yang lebih kompleks, misalnya menggunakan operator lebih besar, lebih kecil, atau operator logika (and, or, not atau lainnya). Nanti Anda akan menggunakan banyak hal seputar kondisional saat membuat program.



**Gambar 1.23. Contoh kondisional**

Event handler juga bisa mengulang blok. App Inventor menyediakan blok-blok seperti Foreach dan While Do untuk mengulangi blok-blok kode. Semua blok di dalam for each dilakukan sekali untuk setiap item di list. Misalnya kalau Anda mau mengirim pesan SMS ke banyak nomor. Anda bisa menggunakan pengulangan ini.



**Gambar 1.24. Pengulangan menggunakan ForEach**

Blok di dalam blok For each akan selalu diulangi sesuai dengan jumlah list yang ada di dalamnya.

Event Handler juga bisa mengingat apa yang dimasukkan. Karena sering kali program perlu untuk mengingat informasi. Informasi ini bisa dimasukkan di slot memori yang bernama variabel. Anda bisa mendefinisikan variabel di editor blok.

Variabel ini mirip dengan properti sebuah komponen, tapi tidak dikaitkan dengan komponen tertentu. Misalnya Anda bisa membuat variabel "nilai" yang akan menyimpan nilai tertentu. Variabel akan menyimpan data secara temporer ketika aplikasi berjalan. Ketika Anda menutup aplikasi, maka data yang ada di variabel akan hilang dan tidak akan ada lagi.

Bisa juga aplikasi membutuhkan data yang bisa tetap tersimpan, bahkan ketika aplikasi dimatikan. Ini semuanya bisa dilakukan dengan database.

Karena data di database akan selalu tersimpan, walaupun program sudah ditutup.

Event handler juga bisa berkomunikasi dengan web. Ini bisa dilakukan kalau aplikasi mengirimkan request web API (application programming interface) untuk berhubungan dengan data di web. Aplikasi yang berhubungan dengan web ini bisa disebut sebagai aplikasi yang web-enabled.

Contoh aplikasi yang bisa berkomunikasi dengan web adalah Twitter. Anda bisa menuliskan aplikasi yang me-request dan menampilkan tweet dari friend dan juga mengupdate status twitter Anda.

Seorang programmer haruslah melihat aplikasinya dari dua sisi, yaitu dari perspektif end user dan dari perspektif programmer. Setelah Anda menulis beberapa program, maka Anda akan mengkaitkan antara tampilan GUI dengan kode di dalamnya.

## Istilah-Istilah Pemrograman

Berikut ini prinsip dari software engineering yang perlu Anda perhatikan dalam membuat sebuah aplikasi, termasuk aplikasi Android menggunakan App Inventor:

1. Ketika memprogram, pertimbangkan prospektif dari user saat membuat program.
2. Bangunlah prototipe awal yang lebih sederhana, kemudian sempurnakan sedikit demi sedikit.
3. Tes kode Anda sedikit demi sedikit. Jangan lebih dari beberapa block sekaligus.
4. Bagi-bagi program Anda menjadi blok-blok kecil.
5. Beri komentar pada blok-blok untuk memudahkan memahaminya.
6. Tracking blok-blok kode menggunakan pensil untuk memudahkan memahami cara kerjanya.

Trik-trik pemrograman di atas akan memudahkan Anda, dan menghindarkan Anda dari rasa frustrasi dan menghemat waktu Anda.

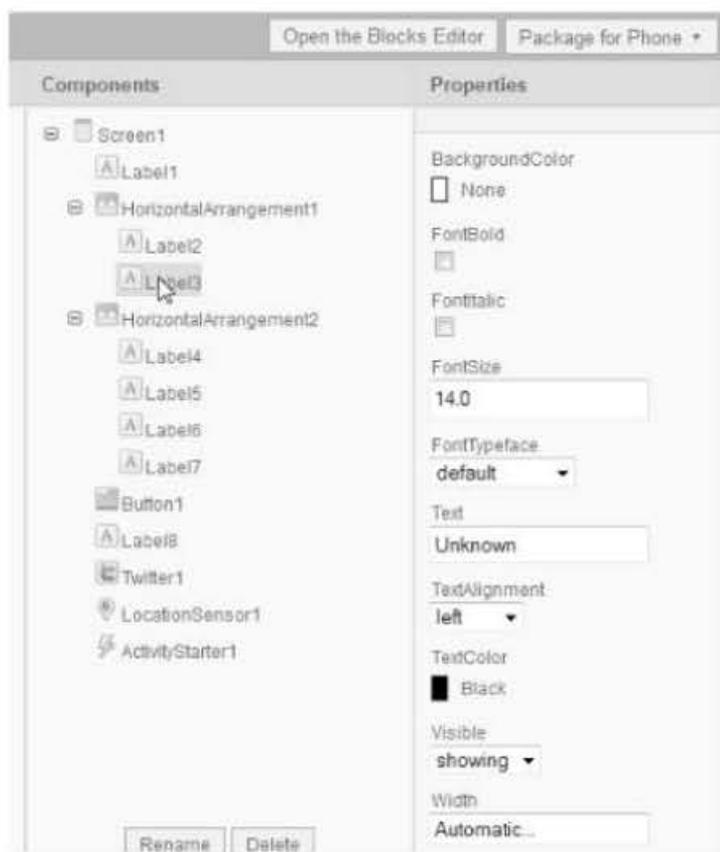
## Alokasi Slot Memori

Memori aplikasi terdiri dari beberapa slot yang dinamai. Beberapa slot memori ini akan otomatis terbuat ketika Anda men-drag komponen ke aplikasi. Slot ini disebut properties.

Anda juga bisa meminta alokasi slot di memori, tapi yang tidak dikaitkan dengan komponen tertentu. Ini disebut variabel atau jika properti biasanya terlihat di aplikasi, variabel bisa dianggap sebagai memori yang tersembunyi dan tidak terlihat pengaruhnya di antarmuka program.

## Properties

Sebuah komponen, seperti button, textbox dan canvas merupakan bagian dari user interface. Properti adalah nilai yang menentukan bagaimana bentuk tampilan dari user interface tersebut. Nilai yang disimpan di slot memori untuk properties ini bisa diubah di bagian Component Designer.



*Gambar 1.25. Mengubah Properties dari sebuah objek*

Sebuah komponen (misalnya canvas) memiliki beberapa properti, seperti BackgroundColor, PaintColor, Visible dan sebagainya. Nilai ini bisa berupa boolean, string dan lain sebagainya. Contohnya properti Visible bernilai boolean true atau false dan menentukan apakah komponen terlihat atau tidak. Ketika Anda mengubah property dari Component Designer, maka Anda sama saja dengan menentukan bagaimana tampilan aplikasi tersebut ketika di-launching.

Sama seperti properties, variabel juga memakan slot memori namun tidak diasosiasikan dengan komponen tertentu. Anda memerlukan variabel ketika program Anda membutuhkan tempat untuk menyimpan nilai tertentu. Misalnya username atau password. Variabel ini nantinya bisa dipanggil untuk ditampilkan di bagian lain. Jika properties dibuat secara otomatis ketika Anda men-drag komponen ke Component Designer, Anda bisa mendefinisikan variabel baru dengan men-drag blok **def variable**.

Anda bisa memberikan nama variabel dengan mengklik teks "variable" di blok. Dan Anda bisa menentukan nilai awal dengan men-drag blok **number**, **text**, **color**, atau **make a list** kemudian memasukkannya ke dalam puzzle yang ada. Berikut ini contoh membuat variabel:

1. Drag blok **def variable** dari folder **Definition** di blok **Built in**.



*Gambar 1.26. Def Variable*

2. Ubah nama variabel dengan mengklik teks **variable** dan mengetikkan nama variabel, misalnya "score".
3. Set nilai awal dari variabel ke angka dengan men-drag **block number** dan memasangkan ke definisi variabel.



*Gambar 1.27. Mengeset value ke angka*

4. Ubah nilai awal dari nilai default "123" ke angka 0.



*Gambar 1.28. Mengubah nilai default*

Ketika Anda mendefinisikan variabel, sama saja Anda memberitahukan aplikasi untuk mengeset slot memori guna menyimpan nilai tertentu. Dan nilai ini tidak akan terlihat oleh user.

Ketika mendefinisikan variabel, AppInventor akan membuat dua blok untuk ini. Dimana keduanya akan ditampilkan di **My Definitions**.



*Gambar 1.29. My Definitions*

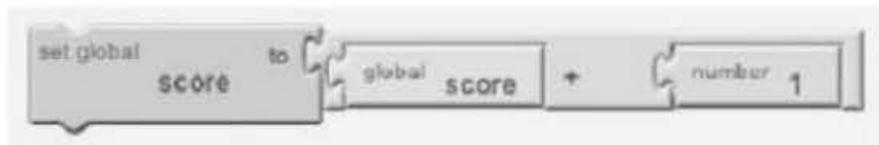
Block set global berguna untuk memodifikasi nilai yang tersimpan di variabel. Misalnya kalau Anda mengeset nilai 5 di variabel, maka term "global" di blok "set global score to" mengacu bahwa variabel bisa dipakai di semua event handler dari program secara global. Beberapa bahasa pemrograman memungkinkan Anda mengeset beberapa variabel ke local, tapi di sini tidak bisa.



*Gambar 1.30. Menggunakan global score untuk memanggil score*

### Mensetting Variabel ke Ekspresi

Sering kali Anda perlu mengoperasikan sebuah variabel, misalnya menambahkan variabel dengan nilai tertentu ketika user mengklik tombol tambah. Untuk itu Anda bisa menggunakan blok Set Global. Misalnya berikut ini contoh menambahkan variabel score dengan angka 1.



*Gambar 1.31. Menambahkan variabel score dengan angka 1*

Jika Anda bisa memahami block ini, maka Anda sudah siap untuk menjadi programmer menggunakan App Inventor.



# 2

## MENGGUNAKAN IDE APP INVENTOR

---

---

*Bab ini membahas:*

- ❖ Komponen Designer.
  - ❖ Block Editor.
  - ❖ Emulator.
  - ❖ Membuat Program Pertama Halo Dunia.
  - ❖ Membuat Program Sprite Ball.
- 

**S**ebelum membuat program, Anda perlu mengenali IDE (integrated development environment) App Inventor tempat Anda membuat program. IDE ini adalah tempat untuk Anda bekerja. Disinilah Anda melakukan pekerjaan pembuatan program. Ada tiga bagian dari IDE yang ada, yaitu komponen Designer, Block Editor, dan yang ketiga adalah Android Virtual Device emulator.

### Komponen Designer

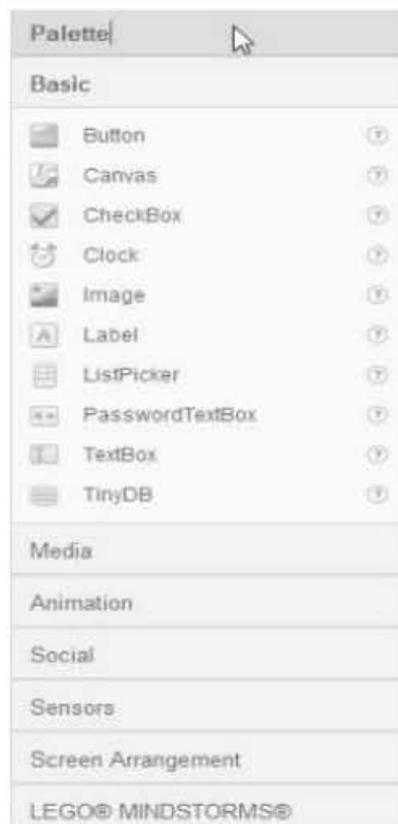
Komponen designer adalah tempat dimana Anda mendesain komponen-komponen antarmuka aplikasi menggunakan App Inventor. Untuk mengaksesnya, silakan buka <http://beta.appinventor.mit.edu/> dan loginlah dengan akun email Anda. Antarmuka pertama yang Anda lihat seperti berikut ini:



*Gambar 2.1. Jendela kerja App Inventor di web browser*

## Palette

Palette adalah bagian dari jendela kerja App Inventor yang menampilkan komponen-komponen. Ada beberapa kategori komponen, dari mulai Basic hingga Lego Mindstorm. Jenis palette pertama adalah **Basic**. Di sini Anda bisa menggunakan beberapa komponen seperti **Button** hingga **Database**.



*Gambar 2.2. Palette Basic*

Beberapa komponen di kelompok **Basic** ini antara lain:

- **Button**

Button atau tombol ini memiliki kemampuan untuk mendeteksi event klik. Anda juga bisa mengatur atribut tampilan, seperti apakah bisa diklik (enabled), dan teks di dalamnya.



*Gambar 2.3. Button di tampilan designer*



*Gambar 2.4. Button ditampilan emulator*

- **Canvas**

Sebuah panel kotak yang sensitif terhadap *touch*, dimana di dalamnya, Anda bisa menggambar sesuatu dan akan muncul bekas di dalamnya. Anda bisa mengatur `BackgroundColor`, `PaintColor`, `BackgroundImage`, `Width`, dan `Height` dari canvas. Baik menggunakan designer, atau di Blocks Editor. Nilai `Width` dan `Height` ini memiliki satuan pixel dan harus bernilai positif.



*Gambar 2.5. Canvas di tampilan viewer*



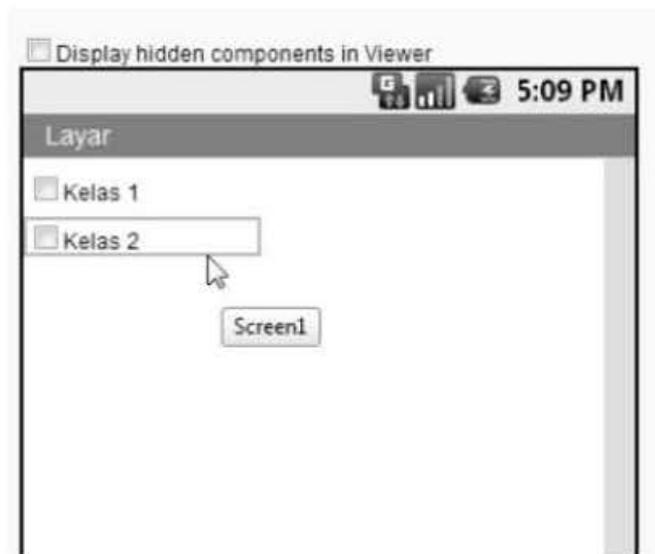
*Gambar 2.6. Canvas di tampilan program*

Semua lokasi di canvas bisa dispesifikasikan dengan nilai (x, y). Dimana X adalah jarak pixel dari ujung kiri canvas. Dan Y adalah jarak pixel dari atas canvas.

Ada banyak event di canvas, misalnya ketika canvas sudah di-touch atau belum. Atau sprite (ImageSprite atau Ball) sudah di-drag. Ada juga method untuk menggambar titik, garis dan lingkaran.

- **CheckBox**

Checkbox akan memicu event jika user mengklik pada checkbox tersebut. Ada banyak properti yang bisa mempengaruhi tampilan dari checkbox yang bisa diset di **Designer** dan **Block Editor**.



*Gambar 2.7. Contoh tampilan checkbox di viewer*



*Gambar 2.8. Checkbox di tampilan layar*

- **Clock**

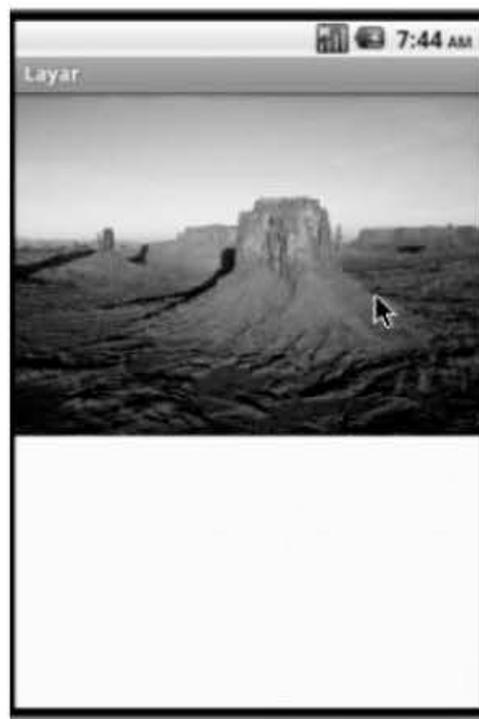
Komponen non-visible yang menyediakan clock untuk ponsel, timer, dan penghitungan waktu.

- **Image**

Komponen untuk menampilkan gambar. Gambar yang akan ditampilkan serta aspek penampilan gambar lainnya bisa diatur di Designer atau di Block Editor.



*Gambar 2.9. Image dengan gambar, di tampilan viewer*



*Gambar 2.10. Image menampilkan gambar, di tampilan viewer*

- **Label**

Fungsinya menampilkan sebagian dari teks, yang ditentukan melalui properti Text. Properti ini bisa diset di Designer atau Block Editor, seperti misalnya pengaturan ukuran font untuk teks di label.

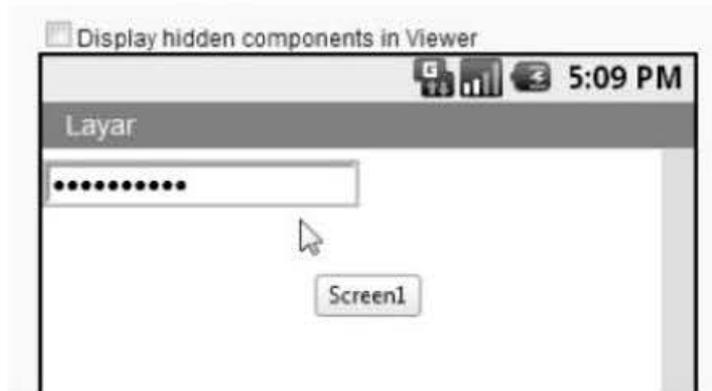
- **ListPicker**

Komponen yang ketika diklik, akan menampilkan daftar teks yang bisa dipilih oleh user. Teks ini bisa ditentukan menggunakan Designer atau Block Editor dengan cara mengatur properti `ElementsFromString` dengan dipisahkan tanda koma (contohnya `choice 1, choice 2, choice 3`) atau dengan menentukan properti `Elements` ke List di Blocks editor.

Ini juga punya properti lain, seperti untuk mengatur tampilan (`TextAlignment`, `BackgroundColor`, dst.) dan apakah ini bisa diklik atau tidak (`Enabled`).

- **PasswordTextBox**

Sebuah kotak untuk memasukkan password. Ini sama seperti komponen `TextBox`, hanya saja tidak menampilkan karakter yang diketikkan oleh user. Nilai dari teks di box bisa ditentukan atau diset menggunakan properti `Text`.



*Gambar 2.11. Sebuah password textbox di tampilan viewer*



*Gambar 2.12. Tampilan password textbox saat masih kosong*



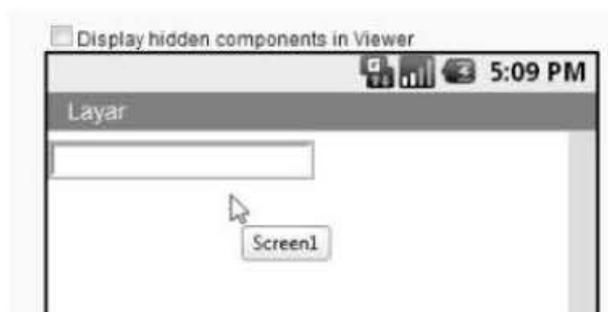
*Gambar 2.13. Tampilan password setelah diisikan*

- **TextBox**

Komponen yang memungkinkan user untuk memasukkan teks, nilai ini bisa dimasukkan dari properti Text. Jika kosong, properti Hint akan menyediakan panduan apa yang harus diketikkan oleh user.

TextBox punya properti MultiLine yang menentukan apakah teks bisa terdiri dari banyak baris atau hanya satu baris saja. Kalau untuk satu baris, keyboard akan close secara otomatis ketika ueser mengklik key Done. Sementara untuk menutup keyboard bagi textbox yang multiline, user harus menggunakan method HideKeyboard atau Back.

Ada properti NumbersOnly yang membatasi keyboard hanya menampilkan masukan nomor saja. Dan properti Enabled menentukan apakah textbox bisa dipakai atau tidak. Textbox biasanya dilengkapi dengan button, dimana kalau user mengklik button, entri teks akan langsung diproses.



*Gambar 2.14. Textbox dalam tampilan viewer*

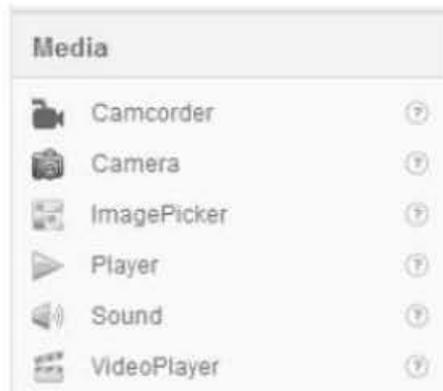


*Gambar 2.15. Textbox diisikan teks pada emulator program*

- **TinyDB**

Merupakan komponen non-visible yang menyimpan nilai secara permanen di ponsel.

Komponen lainnya di Palette dikelompokkan di grup **Media**. Komponen-komponen di grup **Media** ini berkaitan dengan pengaksesan media dengan berbagai jenisnya.



*Gambar 2.16. Palette Media*

Beberapa komponen Media yang penting adalah sebagai berikut:

- **Camcorder**

Merupakan komponen untuk merekam video menggunakan camcorder dari perangkat. Setelah video direkam, maka nama file di ponsel berisi clip akan tersedia sebagai argumen untuk event `AfterRecording`.

Nama file tersebut bisa digunakan, misalnya untuk properti `source` dari komponen `VideoPlayer`.

- **Camera**

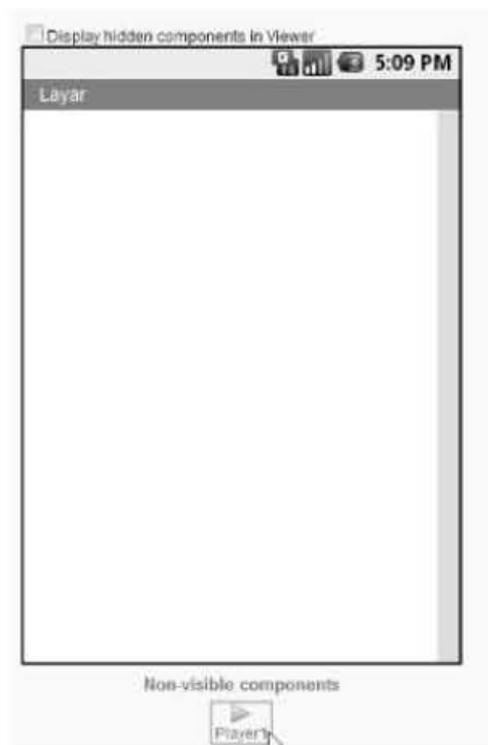
Komponen untuk mengambil gambar menggunakan kamera dari perangkat. Setelah gambar diambil, nama dari file di ponsel di file gambar yang diambil bisa dipakai sebagai argumen untuk event AfterPicture. Nama file ini bisa dipakai untuk mengeset properti Picture dari komponen Image.

- **ImagePicker**

Tombol khusus dimana ketika user men-tap pada image picker, image gallery dari Android akan muncul dan user bisa memilih image. Setelah user mengambil image, maka diambil properti ImagePath diset ke nama image tersebut.

- **Player**

Komponen multimedia yang akan memainkan audio atau video dan mengontrol vibrasi dari ponsel. Nama field multimedia bisa ditentukan di properti Source. Dimana ini bisa diset di Designer atau di Blocks Editor. Panjang waktu dari vibrasi ditentukan di block editor dengan satuan milidetik.



*Gambar 2.17. Tampilan player*

Jika Anda hanya ingin memainkan file sound dan bervibrasi, tanpa memakai video, maka komponen ini sangat cocok untuk file suara yang panjang, seperti lagu, sementara komponen Sound lebih efisien untuk file yang pendek, seperti sound effect.

- **Sound**

Komponen multimedia untuk memainkan file dan juga bisa memvibrasi selama beberapa detik yang waktunya bisa ditentukan

di Blocks Editor. Nama dari file suara yang bisa ditentukan baik di Designer atau di Blocks Editor. Daftar file multimedia yang didukung oleh Core dari Android bisa dilihat di tabel berikut ini:

*Tabel 2.1 Daftar file multimedia yang didukung oleh Core dari Android*

Tipe	Format	Encoder	Decoder	Detail	Format yang didukung
Audio	AAC LC	-	-	Support for mono/stereo/5.0/5.1 content with standard sampling rates from 8 to 48 kHz.	3GPP (.3gp) • MPEG-4 (.mp4, .m4a) • ADTS raw AAC (.aac, decode in Android 3.1+, encode in Android 4.0+, ADIF not supported) • MPEG-TS (.ts, not seekable, Android 3.0+)
	HE-AACv1 (AAC+)	Android 4.1+	-		
	HE-AACv2 (enhanced AAC+)	-	-	Support for stereo/5.0/5.1 content with standard sampling rates from 8 to 48 kHz	
	AAC ELD (enhanced low delay AAC)	Android 4.1+	Android 4.1+	Support for mono/stereo content with standard sampling rates from 16 to 48 kHz	
	AMR-NB	-	-	4.75 to 12.2 kbps sampled @ 8kHz	3GPP (.3gp)
	AMR-WB	-	-	9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz	3GPP (.3gp)
Audio	FLAC		• (Android 3.1+)	Mono/Stereo (no multichannel). Sample rates up to 48 kHz (but up to 44.1 kHz is recommended on devices with 44.1 kHz output, as the 48 to 44.1 kHz downsampler does not include a low-pass filter). 16-bit recommended; no dither applied for 24-bit.	FLAC (.flac) only

Type	Format	Encoder	Decoder	Detail	Format yang didukung
	MP3		•	Mono/Stereo 8-320Kbps constant (CBR) or variable bit-rate (VBR)	MP3 (.mp3)
	MIDI		•	MIDI Type 0 and 1. DLS Version 1 and 2. XMF and Mobile XMF. Support for ringtone formats RTTTL/RTX, OTA, and iMelody	<ul style="list-style-type: none"> <li>• Type 0 and 1 (.mid, .xmf, .mxmf)</li> <li>• RTTTL/RTX (.rtttl, .rtx)</li> <li>• OTA (.ota)</li> <li>• iMelody (.imy)</li> </ul>
	Vorbis		•		<ul style="list-style-type: none"> <li>• Ogg (.ogg)</li> <li>• Matroska (.mkv, Android 4.0+)</li> </ul>
	PCM/WAVE	• (Android 4.1+)	•	8- and 16-bit linear PCM (rates up to limit of hardware). Sampling rates for raw PCM recordings at 8000, 16000 and 44100 Hz.	WAVE (.wav)
Image	JPEG	•	•	Base+progressive	JPEG (.jpg)
	GIF		•		GIF (.gif)
	PNG	•	•		PNG (.png)
	BMP		•		BMP (.bmp)
	WEBP	• (Android 4.0+)	• (Android 4.0+)		WebP (.webp)
Video	H.263	•	•		<ul style="list-style-type: none"> <li>• 3GPP (.3gp)</li> <li>• MPEG-4 (.mp4)</li> </ul>
	H.264 AVC	• (Android 3.0+)	•	Baseline Profile (BP)	<ul style="list-style-type: none"> <li>• 3GPP (.3gp)</li> <li>• MPEG-4 (.mp4)</li> <li>• MPEG-TS (.ts, AAC audio only, not seekable, Android 3.0+)</li> </ul>
	MPEG-4 SP		•		3GPP (.3gp)
	VP8		• (Android	Streamable only in Android 4.0 and above	• WebM (.webm)

Tipe	Format	Encoder	Decoder	Detail	Format yang didukung
			2.3.3+)		• Matroska (.mkv, Android 4.0+)

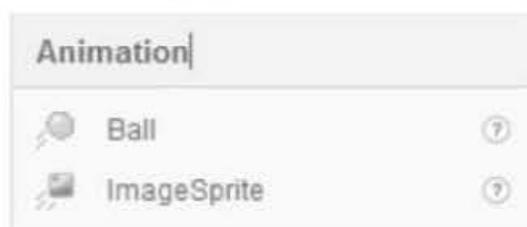
- **VideoPlayer**

Ini adalah komponen multimedia yang bisa memainkan video. Ketika aplikasi ini dijalankan, VideoPlayer akan ditampilkan dalam bentuk kotak di layar. Jika user mengklik pada kotak tersebut, akan muncul kontrol navigasi yang akan mengontrol Play/Pause, Skip Ahead dan Skip Backward di video. Aplikasi ini juga bisa mengontrol perilaku dengan memanggil Start, Pause dan SeekTo.

File video yang didukung bisa berupa format Windows Media Video (.wmv), 3GPP (.3gp), atau MPEG-4 (.mp4). App Inventor untuk Android hanya membatasi video dibawah 1 MB dan membatasi ukuran total dari aplikasi ke 5 MB. Tidak semua yang tersedia untuk media (video, audio dan sound).

Jika file media Anda terlalu besar, Anda akan memperoleh error ketika mempaket atau menginstal aplikasi. Jadi kemungkinan Anda perlu mengurangi jumlah atau ukuran file. Anda bisa menggunakan alat bantu video editor seperti Movie Maker untuk mengurangi ukuran video Anda.

Grup komponen ketiga di Pallette adalah **Animation**. Di sini ada 2 komponen, yaitu **Ball** dan **ImageSprite**. Berikut ini penjelasannya:



*Gambar 2.18. Pallette Animation*

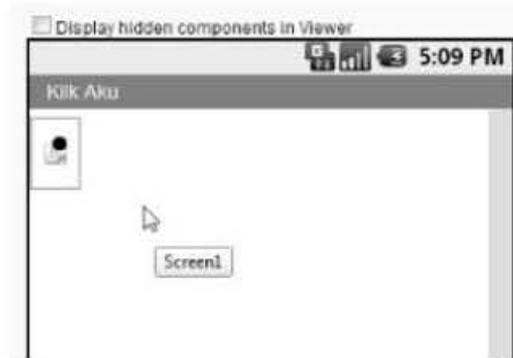
- **Ball**

Merupakan sebuah sprite yang bisa diletakkan di canvas dimana ini bisa bereaksi terhadap sentuhan dan drag. Ini juga dapat berinteraksi dengan ImageSprites serta Ball lainnya.

Misalnya untuk menggerakkan ball 4 piksel ke atas tiap 500 millisecond

(setengah detik), Anda dapat mengeset properti ke 5 piksel, dan interval ke 500. Properti heading bisa diarahkan ke 90 dan properti Enabled ke True. Properti-properti ini bisa diubah kapanpun.

Perbedaan antara ball dan image sprite adalah image prite bisa diubah tampilannya menggunakan file image, sementara tampilan ball hanya bisa diubah menggunakan PaintColor dan properti Radius-nya.



*Gambar 2.19. Contoh ball di canvas*



*Gambar 2.20. Ball di tampilan emulator program*

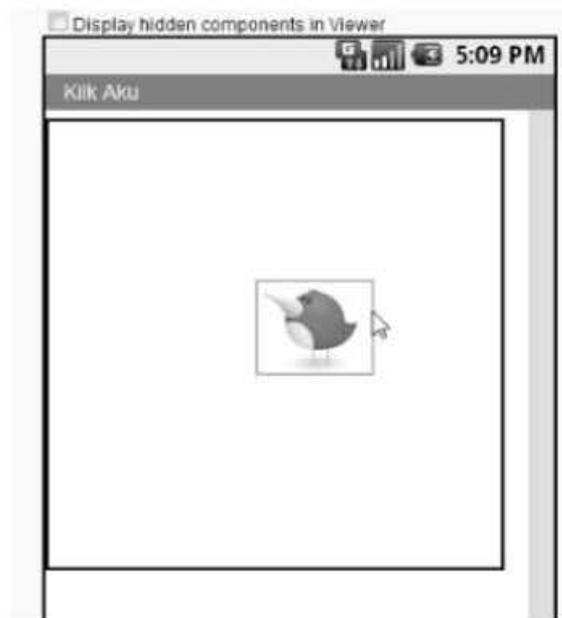


*Gambar 2.21. Ball ini bisa berpindah secara otomatis*

- ImageSprite

Sebuah sprite yang bisa diletakkan di canvas dimana bisa bereaksi terhadap sentuhan dan drag. Serta berinteraksi dengan sprite lainnya. Tampilannya tergantung pada image yang ditentukan di properti Picture. Kecuali kalau set properti-nya False.

Misalnya kalau mau mengubah ImageSprite 10 piksel ke kiri tiap 1 detik, Anda bisa mengeset properti speed ke 10 piksel, dan interval ke 1000 milidetik. Heading properti bisa diset ke 180 derajat dan set enabled ke True.



*Gambar 2.22. Image sprite*



*Gambar 2.23. Sprite gambar berputar sesuai pengaturan properti*

Grup komponen keempat di Pallette adalah **Social**. Di sini terdapat komponen yang berkaitan dengan hubungan dengan orang lain via telepon, sms atau internet.



*Gambar 2.24. Pallet Social*

Beberapa komponen di dalamnya adalah sebagai berikut:

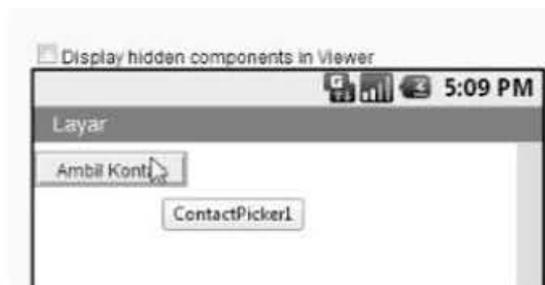
- **ContactPicker.**

Sebuah button yang ketika diklik akan menampilkan daftar kontak yang bisa dipilih. Setelah user memilih, maka properti ini akan diset berkaitan dengan informasi dari Chosen Contact, yaitu:

- **ContactName:** nama dari kontak.
- **EmailAddress:** alamat email dari kontak.
- **Picture:** nama dari file yang berisi image kontak. Ini bisa dipakai sebagai nilai properti **Picture** untuk **Image** atau **ImageSprite**.

Ada properti lain yang mempengaruhi tampilan button ini, seperti **TextAlignment**, **BackgroundColor**, dan sebagainya. Ini bisa diakses asalkan diaktifkan (**Enabled**).

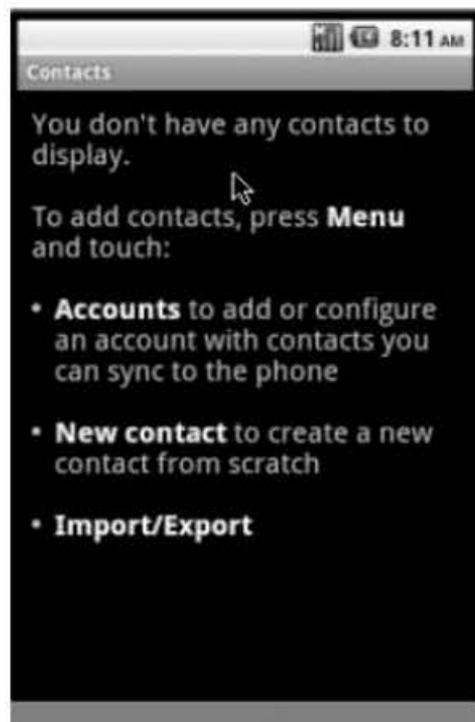
Contact Picker tidak didukung di semua ponsel. Jika ini tidak bisa dijalankan, maka komponen akan menampilkan notifikasi. Perilaku error ini bisa ditimpa menggunakan event **Screen.ErrorOccurred**.



*Gambar 2.25. ContactPicker di tampilan viewer*



*Gambar 2.26. Tampilan contact picker sebelum diklik*

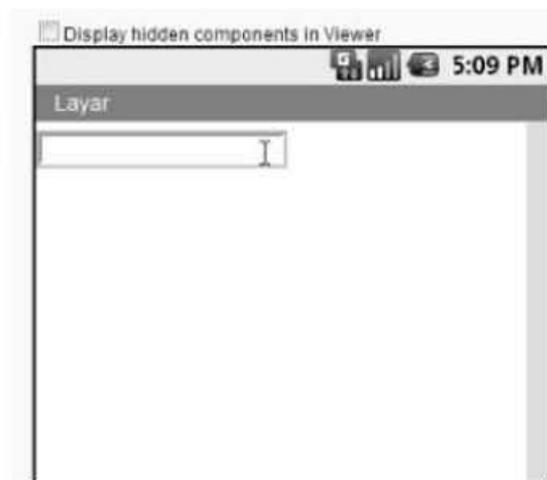


*Gambar 2.27. Tampilan setelah diklik*

#### ▪ EmailPicker

Sebuah textbox yang bisa digunakan user untuk mengisikan alamat email dan nantinya akan diberi fasilitas autocomplete. Nilai yang diambil akan disimpan di properti Text. Jika nilai Text masih kosong, maka akan ditampilkan di textbox sebagai petunjuk untuk user.

EmailPicker juga punya properti yang bisa mempengaruhi tampilan dari textbox, seperti TextAlignment, BackgroundColor, dan lainnya yang bisa diakses, ketika perangkat ini dijalankan (Enabled).



*Gambar 2.28. Tampilan Email picker di preview*



*Gambar 2.29. Mengisikan email picker di emulator*

- **PhoneCall**

Komponen tak terlihat yang berguna untuk melakukan panggilan telepon ke nomor tertentu di properti `PhoneNumber`. Ini bisa diset di Designer atau di Blocks Editor. Komponen ini punya method `MakePhoneCall` yang akan mengaktifkan program untuk melakukan panggilan telepon.

Komponen ini sering dipakai beriringan dengan komponen `ContactPicker` yang memungkinkan Anda memilih contact yang tersimpan di ponsel dan mengeset properti `PhoneNumber` ke nomor telepon dari kontak.

Misalnya untuk menelpon nomor 021-555-1212, Anda bisa mengeset `PhoneNumber` ke Text dengan digit tertentu (misalnya '0215551212').

Anda juga bisa menggunakan tanda minus atau tanda kurung (misalnya (021)-555-1212') tapi akan diabaikan oleh Android.

- **PhoneNumberPicker**

Sebuah button yang jika diklik akan menampilkan daftar dari nomor kontak dari ponsel untuk dipilih. Setelah user memilih, properti akan diset ke informasi mengenai kontak.

- **ContactName:** nama kontak.
- **PhoneNumber:** nomor telepon kontak.
- **EmailAddress:** alamat email dari kontak.
- **Picture:** nama dari file yang berisi image kontak. Ini bisa dipakai sebagai nilai properti **Picture** untuk komponen **Image** atau **ImageSprite**.

- **Texting**

Komponen yang akan mengirimkan pesan teks ketika method **SendMessage** dipanggil. Ini akan mengirimkan pesan yang ada di properti **Message** ke nomor ponsel di properti **PhoneNumber**.

Komponen ini juga akan menerima pesan teks, kecuali jika properti properti **ReceivingEnabled** bernilai **False**. Ketika ada pesan masuk, event **MessageReceived** akan diangkat dan menyediakan pesan dan nomor yang akan dikirim.

Jika properti **GoogleVoiceEnabled** diaktifkan, pesan ini bisa dikirim atau diterima menggunakan Wifi. Ini mensyaratkan Anda harus memiliki akun Google Voice. Dan juga Google Voice terinstal di ponsel Anda. Ini hanya bisa dipakai di ponsel yang mendukung Android 2.0 ke atas. Komponen ini sering digunakan dengan komponen **ContactPicker** yang memungkinkan Anda memilih kontak dari telepon.

Grup komponen kelima di **Palette** adalah **Sensors**. Pada grup ini berisi beberapa sensor-sensor yang ada di Android, seperti yang ditampilkan pada gambar berikut:



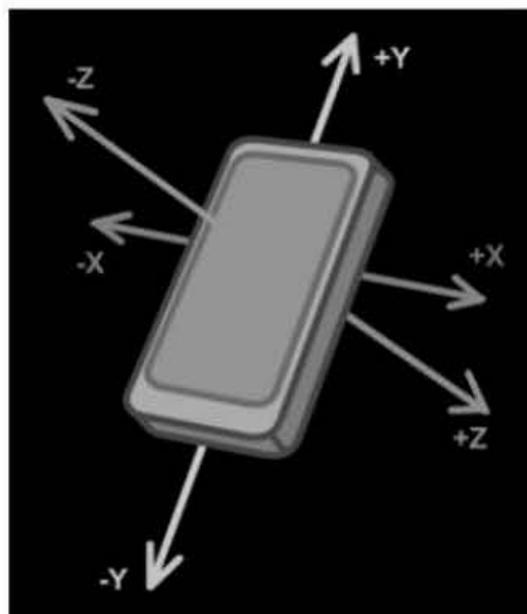
*Gambar 2.30. Palette Sensors*

Berikut merupakan penjelasan masing-masing komponen:

- **AccelerometerSensor**

Komponen tak terlihat yang bisa mendeteksi guncangan dan mengatur kecepatan dalam arah tiga dimensi menggunakan unit Standar internasional ( $m/s^2$ ). Komponen-komponen di AccelerometerSensor ini adalah:

- **xAccel**: nilainya 0 ketika ponsel diletakkan di permukaan yang rata. Dan positif ketika ponsel miring ke kanan (bagian kirinya diangkat) dan negatif ketika ponsel diangkat ke kiri (sisi kanannya diangkat).
- **yAccel**: nilainya 0 ketika ponsel diletakkan di permukaan yang rata. Nilainya positif ketika bagian bawahnya diangkat dan negatif ketika bagian atasnya diangkat.
- **zAccel**: nilainya  $-9.8$  (gravitasi bumi dalam satuan meter/detik), ketika perangkat diset paralel di bumi dengan arah atas menghadap ke bawah, dan nilainya 0 ketika sejajar dengan ground ketika perangkat menghadap ke atas. Dan  $+9.8$  ketika menghadap ke bawah. Nilai ini bisa ber-efek dengan mengakselerasinya searah atau berkaitan dengan gravitasi.



*Gambar 2.31. Sumbu-sumbu di Accelerometer*

- **LocationSensor**

Komponen tak terlihat, yang fungsinya untuk menyediakan informasi lokasi, termasuk longitude, latitude, altitude (jika didukung oleh perangkat) dan alamat. Komponen ini juga bisa melakukan geocoding, mengkonversi alamat ke latitude yang dilakukan oleh method `LatitudeFromAddress` dan longitude dengan method `LongitudeFromAddress`.

Agar bisa berfungsi, komponen harus mengeset properti `Enabled` ke `True` dan perangkat harus memiliki sensor lokasi melalui wifi atau satelit GPS (kalau ada).

- **OrientationSensor**

Komponen visible yang menyediakan informasi mengenai orientasi perangkat fisik dalam tiga dimensi. Komponen yang terdapat di OrientationSensor antara lain:

- **Roll:** 0 derajat ketika perangkat datar, dan akan bertambah ke 90 derajat jika diangkat ke sisi kiri, dan menurun ke -90 derajat ketika perangkat diangkat ke sisi kanannya.
- **Pitch:** 0 derajat ketika perangkat datar, naik ke 90 derajat ketika perangkat diangkat sehingga bagian atasnya ada di bawah, dan 180 derajat ketika dikebawahkan. Sementara jika bagian bawahnya mengarah ke bawah, maka pitch akan menurun ke -90 derajat. Dan akan menurun ke -180 ketika diputar secara penuh.
- **Azimuth:** 0 derajat ketika bagian atas menghadap utara, 90 derajat ketika mengarah ke timur, 190 derajat ketika mengarah ke selatan, dan 270 derajat ketika mengarah ke barat.

Grup komponen keenam Palette adalah **Screen Arrangement** yang berisi komponen-komponen untuk mengatur layout program.



*Gambar 2.32. Screen Arrangement*

Beberapa komponen yang ada di grup ini antara lain:

- **HorizontalArrangement**

Elemen pemformatan yang berguna untuk meletakkan komponen dari kiri ke kanan. Jika Anda ingin meletakkan sebuah komponen di atas lainnya, gunakan Vertical Alignment.

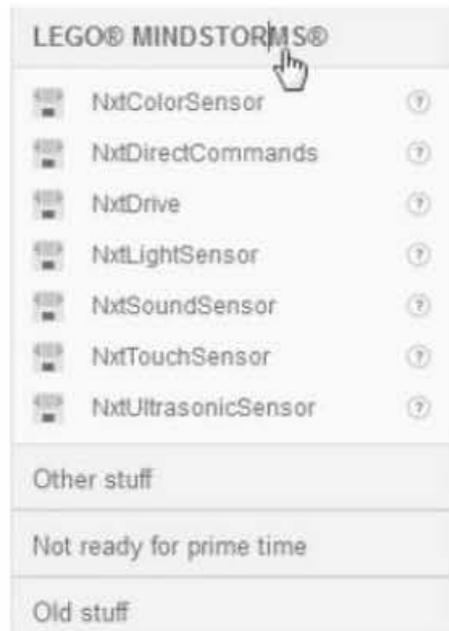
- **TableArrangement**

Elemen pemformatan untuk meletakkan komponen yang diatur dalam bentuk tabel.

- **VerticalArrangement**

Elemen pemformatan untuk menampilkan komponen secara vertikal. Komponen pertama diletakkan di atas, dan komponen berikutnya di bawahnya, dan seterusnya.

Grup komponen ke tujuh di Pallette adalah **Lego Mindstorm** seperti yang bisa Anda lihat di gambar berikut:



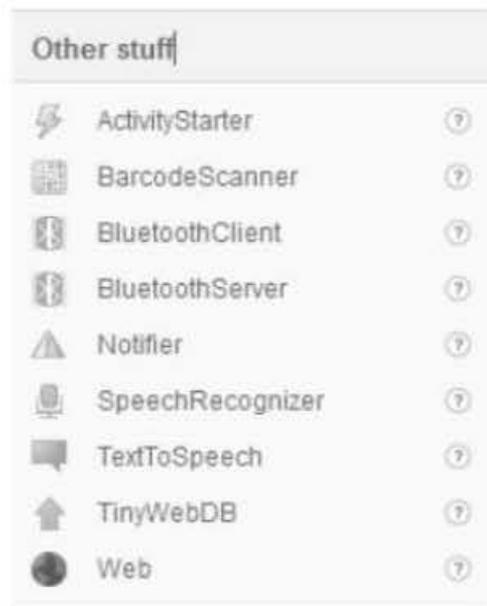
*Gambar 2.33. Pallette Lego Mindstorm*

Lego Mindstorm adalah komponen kit yang berisi software yang bisa dikombinasikan dengan hardware Lego untuk membuat robot yang kecil dan bisa dikostumisasi.

Hardware dan software dari Lego Mindstorm ini berasal dari lab Mindstorms Robotics Invention System yang dibuat oleh MIT Media Lab. Brick yang dipakai diprogram menggunakan logo Brick. Bahasa pemrograman visual yang dipakai disebut LEGOsheets, karena dikembangkan oleh University of Colorado pada tahun 1994 berbasis kepada AgentSheets. Sistem kit dari Mindstorms Robotics Invention ini terdiri dari dua motor, dua sensor sentuh dan satu sensor cahaya.

Mindstorm kit ini dijual dan digunakan sebagai tool edukasi. Awalnya dikembangkan sebagai kerjasama antara Lego dan MIT Media Laboratory. Versi pendidikannya disebut Lego Mindstorms for Schools, dan memiliki bahasa pemrograman ROBOLAB GUI-based programming software, yang dikembangkan di Universitas Tufts dengan engine LabVIEW.

Grup komponen berikutnya di **Pallette** adalah **Other Stuff** yang berisi beberapa komponen lain dari kategori sebelumnya, komponen ini penting pula untuk Anda dipahami.



*Gambar 2.34. Pallette Other stuff*

Beberapa komponen di grup ini adalah sebagai berikut:

- **ActivityStarter**

Ini adalah komponen yang bisa meluncurkan activity menggunakan method StartActivity. Beberapa aktivitas yang bisa dijalankan antara lain:

- Menjalankan AppInventor lain untuk aplikasi Android.
- Memulai aplikasi kamera.
- Melakukan pencarian web.
- Membuka browser untuk membuka webpage tertentu.
- Membuka aplikasi map ke lokasi tertentu.
- Anda bahkan bisa menjalankan aktivitas mengembalikan data teks.

- **BarcodeScanner**

Ini adalah komponen yang akan membaca barcode untuk Anda.

- **BluetoothClient**

Merupakan komponen Bluetooth client.

- **BluetoothServer**

Merupakan komponen bluetooth server.

- **Notifier**

Merupakan komponen yang akan menciptakan pesan alert, dialog pop up dan entri log.

- **SpeechRecognizer**

Merupakan komponen yang menggunakan Voice Recognition untuk mengkonversi dari nada bicara ke teks.

- **TextToSpeech**

Merupakan komponen yang menggunakan TextToSpeech untuk mengubah teks menjadi suara.

- **TinyWebDB**

Merupakan komponen yang tidak terlihat yang berkomunikasi dengan web service untuk menyimpan dan mengambil informasi.

- **Web**

Merupakan komponen tidak terlihat yang menyediakan fungsi untuk request HTTP Get dan POST.

Dan grup komponen terakhir di Pallete adalah **Not ready for prime time**. Ini adalah tempat komponen yang masih dalam tahap beta dan belum sempurna pengembangannya oleh tim MIT.



*Gambar 2.35. Pallete Not ready for prime time*

## Viewer

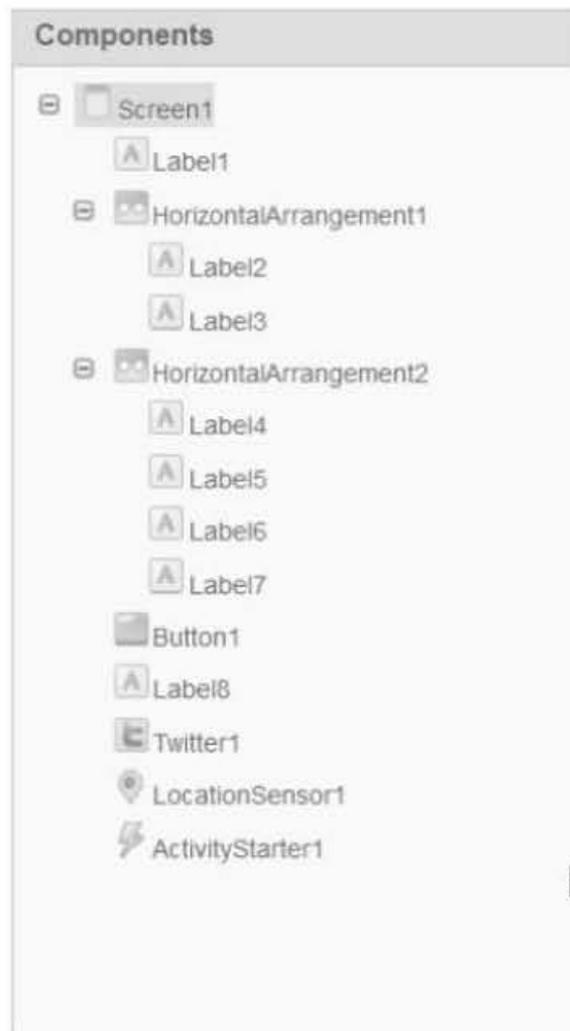
Komponen Designer berikutnya yang penting adalah Viewer, ini gunanya untuk menampilkan contoh tampilan program Android Anda di tahap desain. Tampilan di tahap run biasanya lebih realistis dari tampilan di jendela Viewer ini.



*Gambar 2.36. Tampilan Viewer*

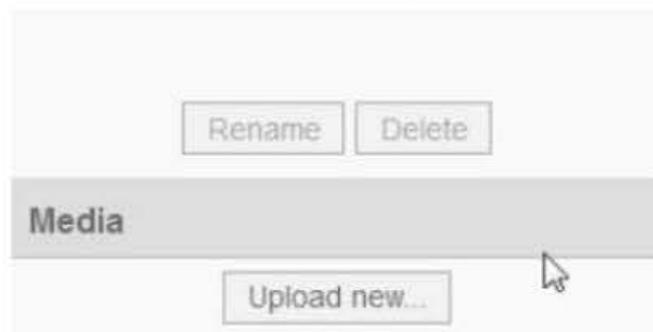
## Components

Komponen ini gunanya untuk menampilkan semua komponen yang sudah dimasukkan ke form. Semuanya yang ada di screen akan ada di bawah screen yang bersangkutan. Anda bisa memilih salah satu komponen untuk melihat properties-nya.



*Gambar 2.37. Bagian Components*

Di bagian bawah **Components** terdapat **Media**. Ini adalah tempat bagi Anda untuk mengupload komponen baru.



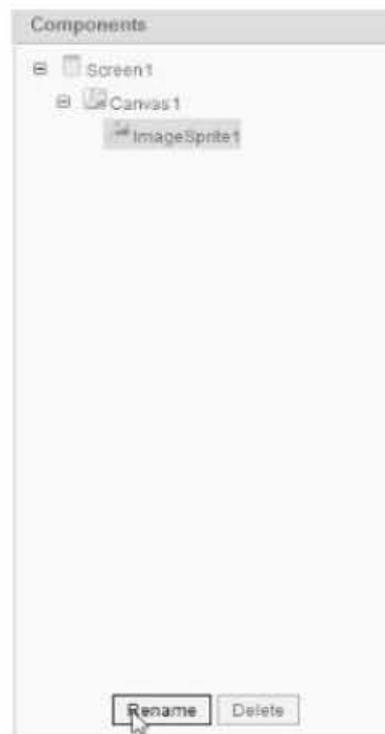
*Gambar 2.38. Upload komponen di Media*

Kalau Anda mengklik pada komponen tertentu, terlihat properties yang berkaitan dengan komponen tersebut. Di panel **Properties**, Anda bisa mengedit properti-properti komponen yang bersangkutan.



*Gambar 2.39. Bagian Properties*

Anda bisa mengganti nama sebuah komponen. Caranya dengan klik pada nama komponen tersebut, kemudian klik tombol **Rename** di bagian bawahnya.



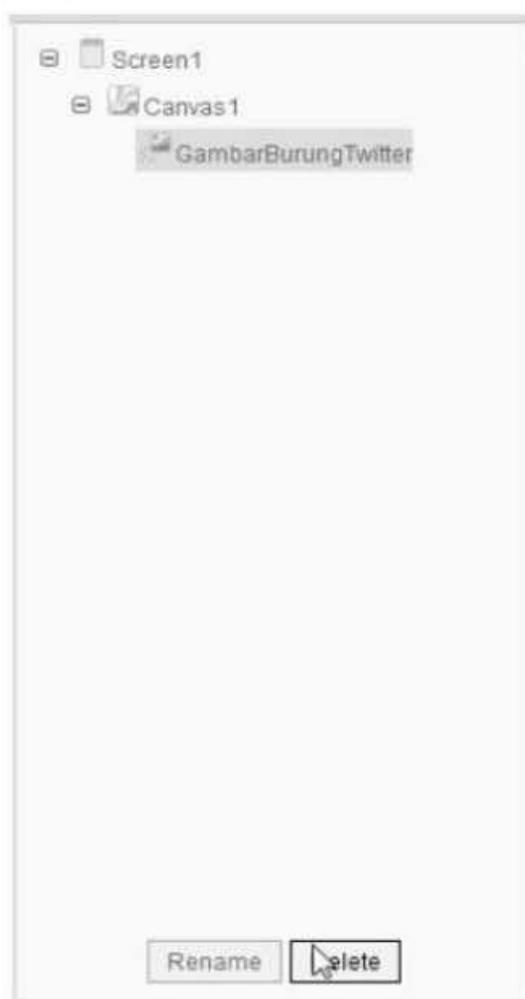
*Gambar 2.40. Tombol Rename untuk mengubah nama komponen*

Kemudian isikan nama baru di textbox **New name** di kotak **Rename Component**.



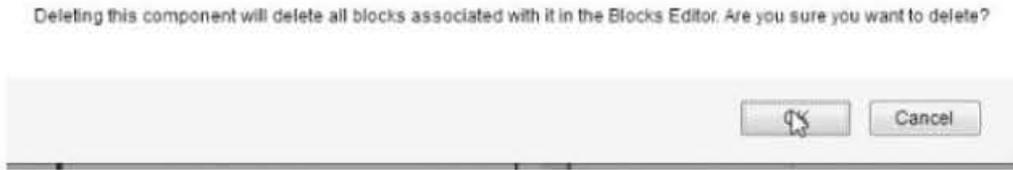
*Gambar 2.41. Mengisikan nama baru di Rename Component*

Nama di jendela **Components** akan langsung berubah sesuai dengan yang Anda tuliskan. Anda juga bisa menghapus komponen dengan klik pada komponen tersebut, kemudian klik **Delete** untuk menghapusnya.



*Gambar 2.42. Klik Delete untuk menghapus komponen*

Muncul konfirmasi, klik **OK** maka komponen akan langsung terhapus dan hilang dari panel **Components**.



*Gambar 2.43. Konfirmasi untuk menghapus komponen*

## Block Editor

Block editor adalah komponen yang berfungsi untuk memasang blok-blok puzzle. Karena App Inventor ini merupakan pemrograman berbasis block, ini identik dengan tampilan kode dimana Anda bisa mengatur flow program dari aplikasi yang Anda buat. Untuk membuka Block Editor ini, klik pada **Open the Blocks Editor** dari tampilan **Designer**.



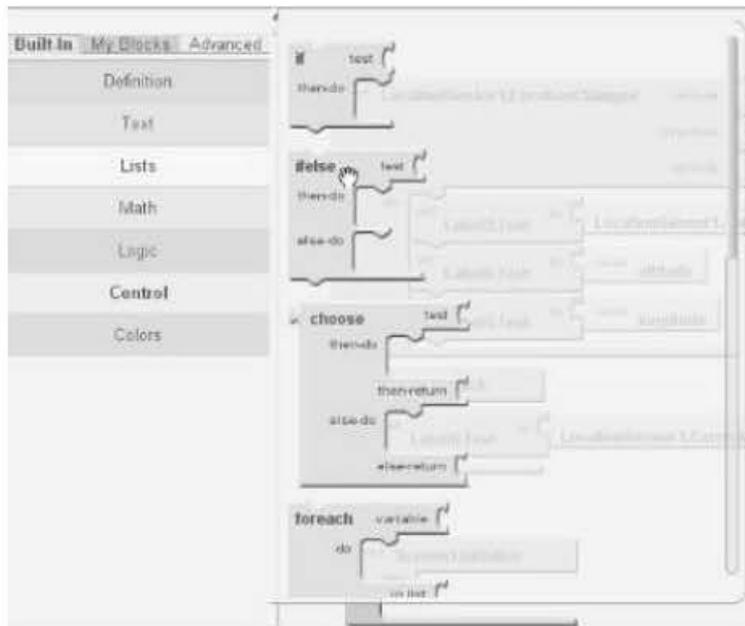
*Gambar 2.44. Klik pada Open the Blocks Editor*

Maka muncul jendela **Block Editor** yang dibuat menggunakan Java. Karena diawal pembahasan Anda sudah menginstall Java, maka Block Editor ini akan langsung diaktifkan. Berikut ini tampilan dari Block Editor:



*Gambar 2.45. Contoh tampilan block editor dengan adanya blok di dalamnya*

Pada bagian kiri dari jendela **Block Editor** terdapat 3 tab, yang pertama adalah **Built-In**. pada tab ini Anda bisa melihat berbagai kategori block yang built in dan penting untuk pemrograman, seperti berbagai blok untuk mengatur flow, lalu control dan sebagainya.



*Gambar 2.46. Pengaturan built in untuk pemrograman*

Tab kedua adalah **My Blocks**, Anda bisa melihat berbagai block yang bersumber dari komponen-komponen Anda.



*Gambar 2.47. Berbagai block yang bersumber dari komponen-komponen*

Misalnya untuk **Button1**, Anda bisa melihat berbagai blok berkaitan dengan event Button1 tersebut. Dari mulai event terhadap button, hingga method atau nilai properti dari komponen tersebut.



*Gambar 2.48. Pengetikkan button yang diinginkan*

Dan tab ketiga adalah **Advanced**. Pada tab ini terdapat blok-blok lain yang penting, seperti untuk activity starter, dan semua jenis komponen secara umum.



*Gambar 2.49. Block-block lain yang penting*

# Emulator

Bagian ketiga dari IDE yang penting adalah emulator. Untuk menjalankan emulator, Anda bisa klik pada tombol **New emulator** di **Block Editor**.



*Gambar 2.50. Klik pada New Emulator di Block Editor*

Muncul pemberitahuan bahwa emulator sedang dijalankan, loading-nya bisa memakan waktu yang lama sesuai dengan spesifikasi komputer Anda.



Screenshot saved

*Gambar 2.51. Emulator sedang dimulai*

Kalau tampilan emulator sudah muncul, terlihat seperti ini:



*Gambar 2.52. Tampilan emulator sudah muncul*

Tinggal di *un-lock* dan Anda bisa melihat tampilan emulatornya.



*Gambar 2.53. Tampilan emulator setelah di-unlock*

Jika Anda ingin mengubah orientasi layar, tekan tombol **Ctrl+F11**.

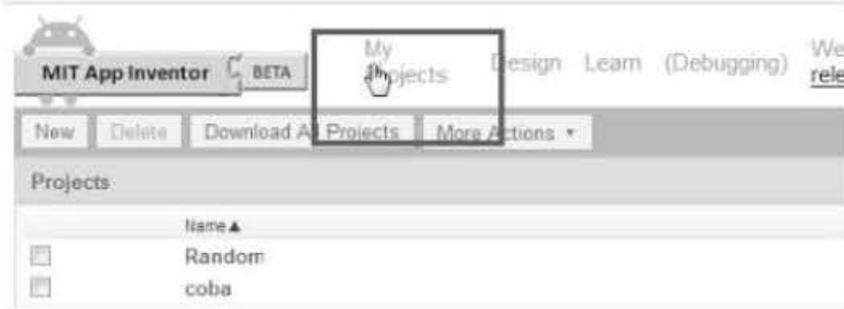


*Gambar 2.54. Emulator berubah orientasi setelah diklik Ctrl+F11*

# Membuat Aplikasi “Halo Dunia”

Untuk melihat bagaimana alur atau *step by step* apa saja yang perlu dilakukan untuk membuat aplikasi Android di App Inventor, ikuti tahapan pembuatan aplikasi Halo Dunia berikut ini:

1. Klik pada **My Projects** di bagian atas.



*Gambar 2.55. Klik pada tulisan My Projects*

2. Isikan nama proyek baru yaitu “HaloDunia” di bagian **Project Name**.



*Gambar 2.56. Pengisian nama proyek baru di Project Name*

3. Klik **OK**, maka proyek baru tersebut akan tersimpan dan bisa Anda lihat terdapat di list **Projects**.



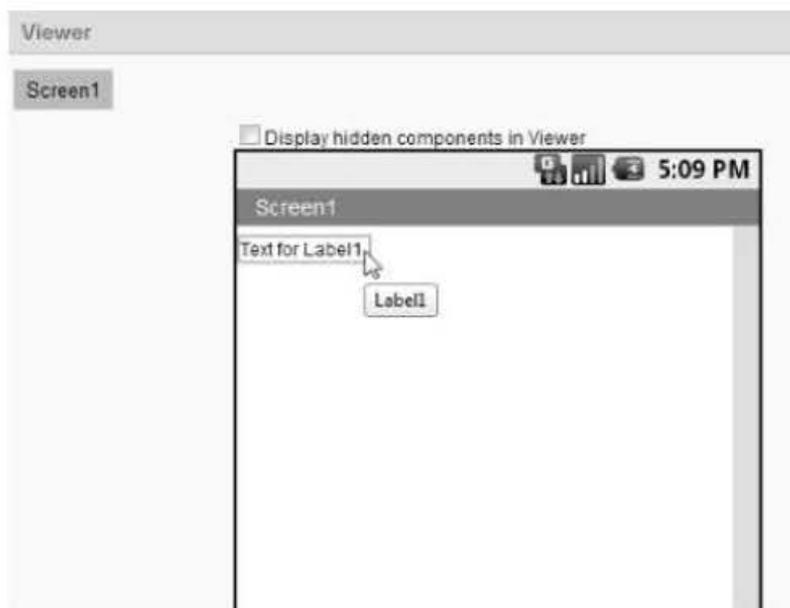
*Gambar 2.57. Nama proyek akan ada di list Projects*

4. Tampilan halaman **Viewer** menunjukkan screen yang masih kosong.



*Gambar 2.58. Tampilan halaman Viewer menunjukkan screen yang masih kosong*

5. Masukkan objek **Label**, maka di layar akan muncul label.



*Gambar 2.59. Klik pada label untuk memasukkan label*

6. Ganti properti **Text** dari label ini dengan tulisan "...Klik Tombol..."



*Gambar 2.60. Penggantian properti text dari label*

7. Kemudian masukkan **Button** ke screen.



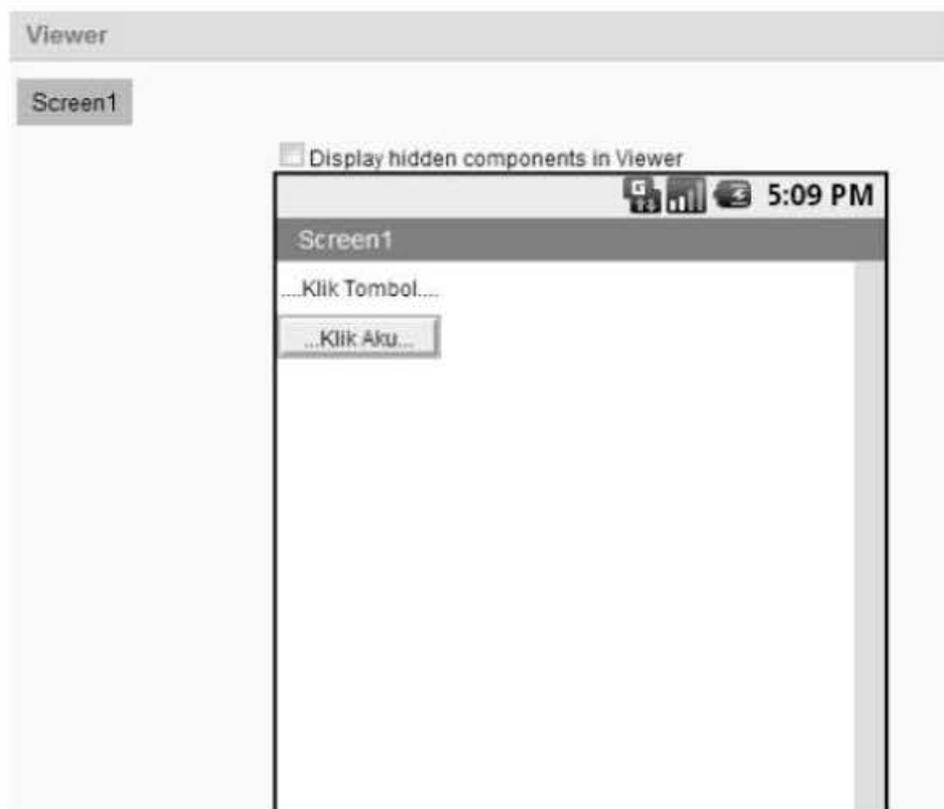
*Gambar 2.61. Memasukkan button ke screen*

8. Ganti properti **Text** button ini menjadi "...Klik Aku...".



*Gambar 2.62. Penggantian properti text dari button*

9. Maka penggantian properti text tersebut akan langsung terlihat di screen.



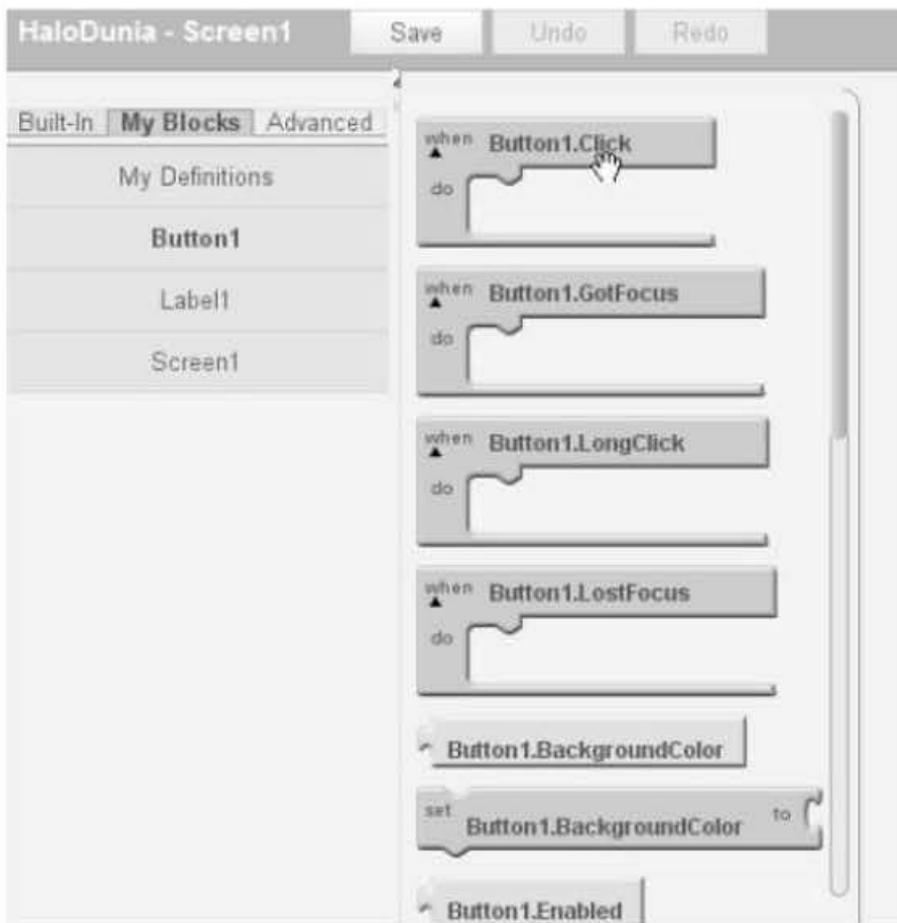
*Gambar 2.63. Penggantian properti text terlihat di screen*

10. Kemudian beralih ke **Block Editor**, klik pada **My Blocks** → **Button1**.



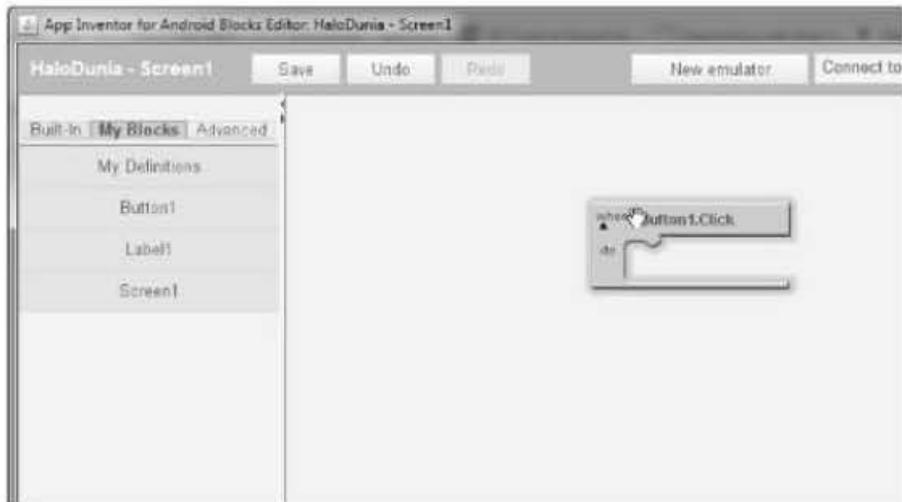
*Gambar 2.64. Klik pada Button1*

11. Klik pada event blok **When Button1.Click do**



*Gambar 2.65. Klik pada blok When Button1.Click do*

12. Maka block tersebut akan terlihat di jendela **Block Editor**.



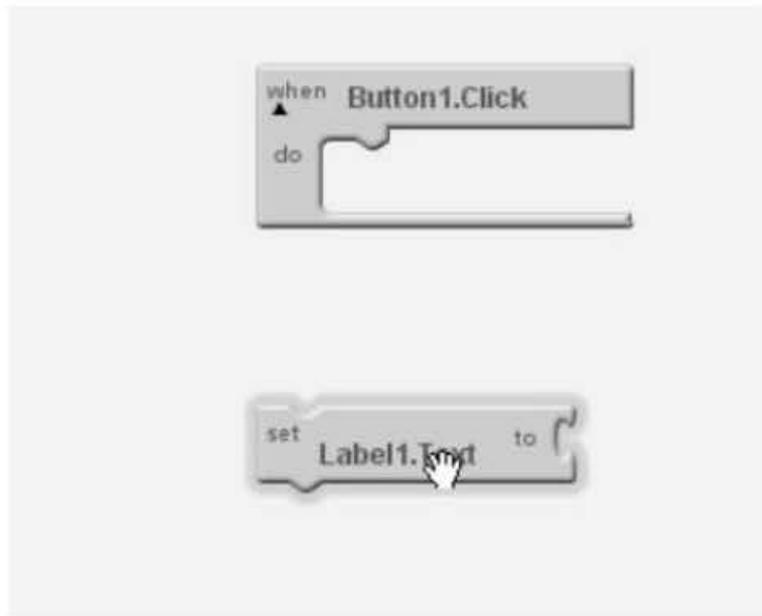
*Gambar 2.66. Block editor*

13. Pada contoh ini, jika tombol diklik, maka label1.text akan berisi tulisan "Halo Dunia". Maka klik pada **My Blocks** → **Label1** dan klik pada block method **Set Label1.Text to**.



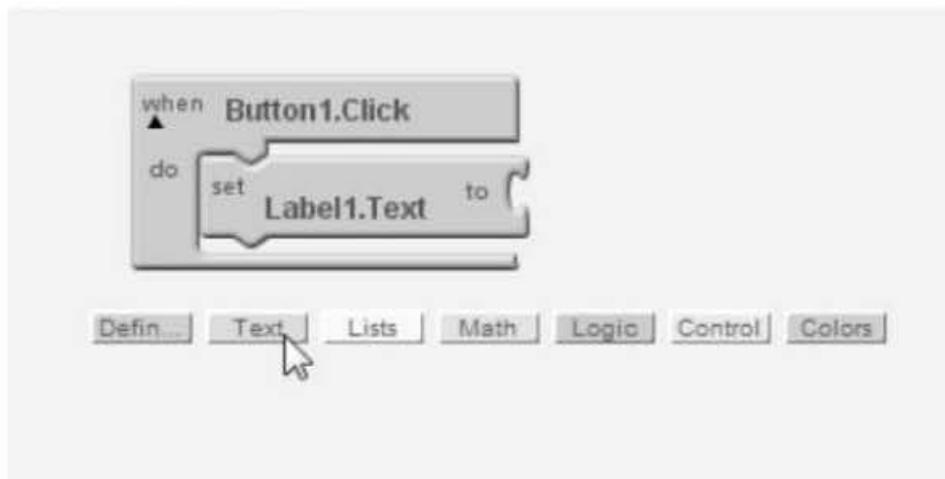
*Gambar 2.67. Klik pada blok Set Label1.Text to*

14. Maka ada dua block yang sudah masuk seperti berikut ini:



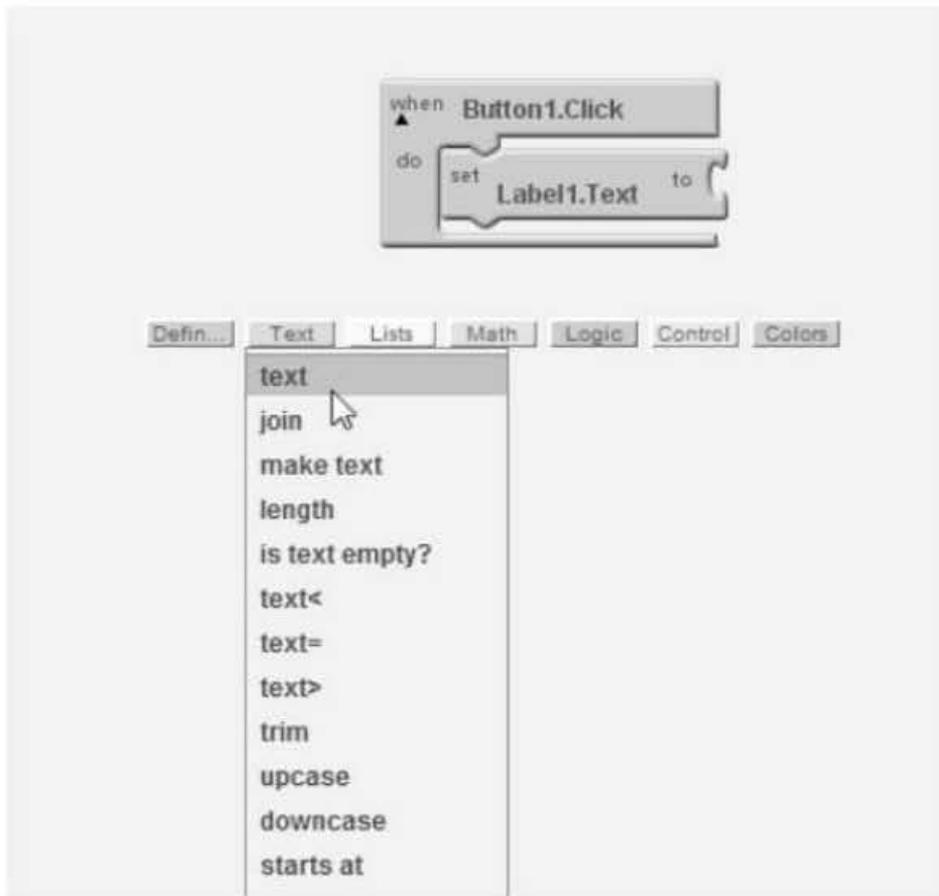
*Gambar 2.68. Dua block yang sudah masuk*

15. Geser blok **Label1.Text** ke tengah **When button1.click do** sehingga menjadi seperti berikut:



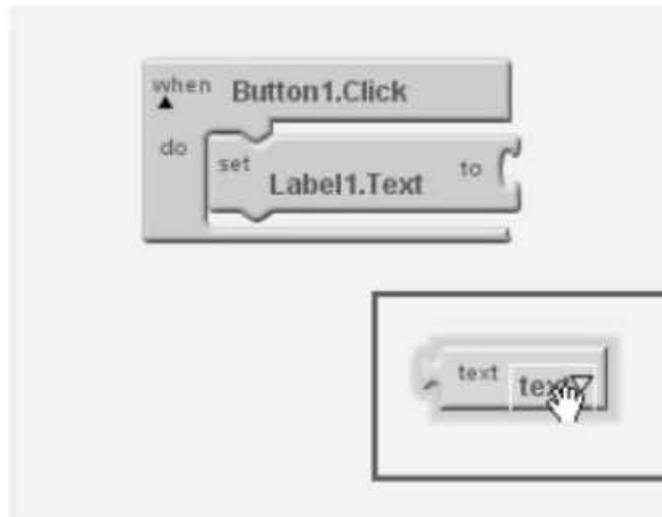
*Gambar 2.69. Blok set Label1.text sudah diletakkan di dalam event When button1.click*

16. Untuk mengisi teks di **label1.text**, klik pada sembarang tempat di area **Designer**, kemudian klik pada kotak **Text**, dan pilih menu **Text**.



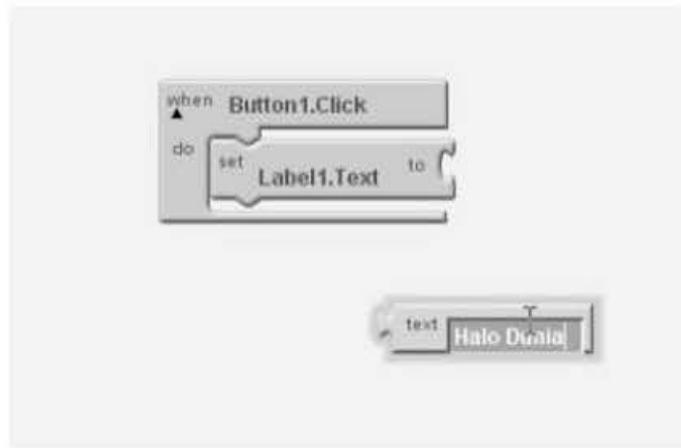
*Gambar 2.70. Klik pada menu Text*

17. Maka muncul blok baru seperti berikut ini:



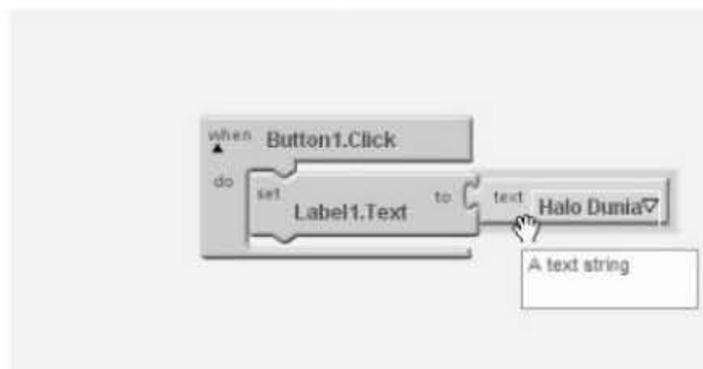
*Gambar 2.71. Muncul blok text*

18. Ganti teks dari **Text** dengan tulisan "Halo Dunia".



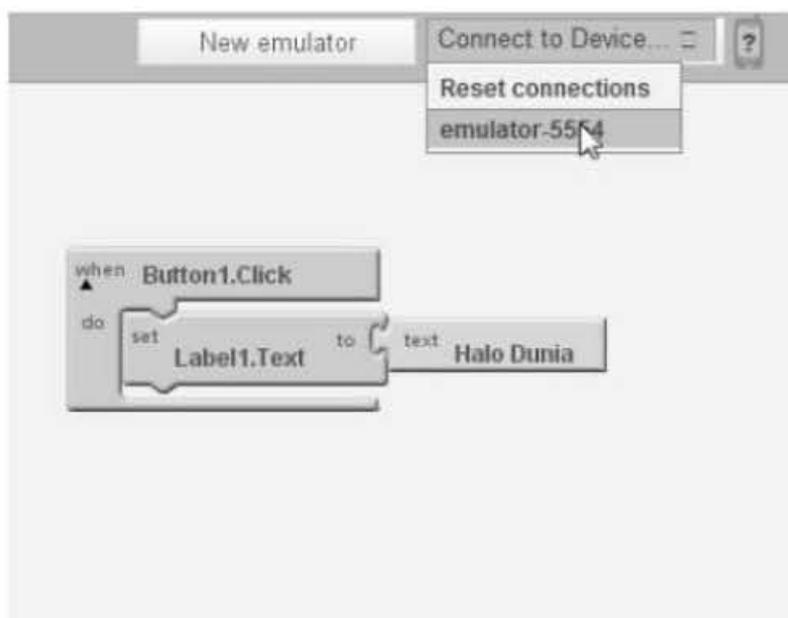
Gambar 2.72. Penggantian teks dengan tulisan Halo Dunia

19. Pasangkan blok text tersebut pada method Set `label1.text` to.



Gambar 2.73. Pemasangan teks

20. Hubungkan ke emulator dengan mengklik **Connect to device** → **Nama emulator** yang sedang berjalan.



Gambar 2.74. Menu untuk koneksi ke device

21. Hasilnya, aplikasi yang Anda buat muncul di emulator. Awalnya pada label terdapat teks “klik tombol”.



Gambar 2.75. Di label ada tulisan untuk klik tombol

22. Kalau tombol sudah diklik, tampilan di label akan berubah menjadi “Halo Dunia”.



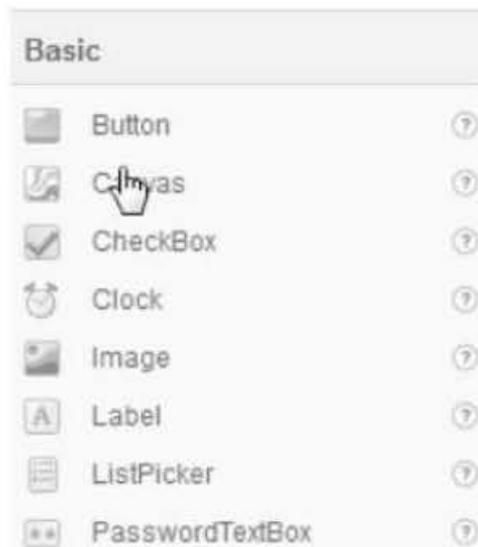
Gambar 2.76. Tampilan di label sudah berubah

## Membuat Aplikasi Klik Sprite Ball

Sekarang akan diberikan contoh membuat aplikasi menggunakan canvas dan sprite ball di atasnya. Anda akan mencoba membuat aplikasi yang akan menambahkan skor tiap kali Anda mengklik bola di canvas dan kemudian menampilkan skor tersebut di bagian textbox di bawahnya.

Berikut ini langkah-langkah pembuatan aplikasinya:

1. Buat proyek baru, atau bisa pakai proyek yang sudah ada.
2. Tambahkan komponen **Canvas** pada grup **Basic** di Palette.



*Gambar 2.77. Klik pada Canvas*

3. Anda bisa mengatur **Width** dan **Height** sesuai dengan keinginan Anda.



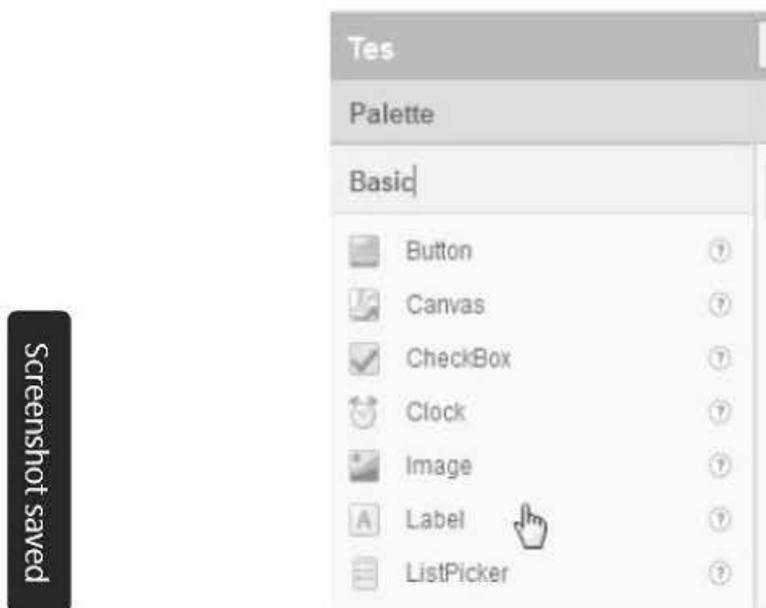
*Gambar 2.78. Pengaturan Height dan Width*

4. Kemudian tambahkan komponen **Ball** dengan mengklik **Animation** → **Ball**.



Gambar 2.79. Klik pada komponen Animation → Ball

5. Dan tambahkan **Label** sebanyak 2 buah.



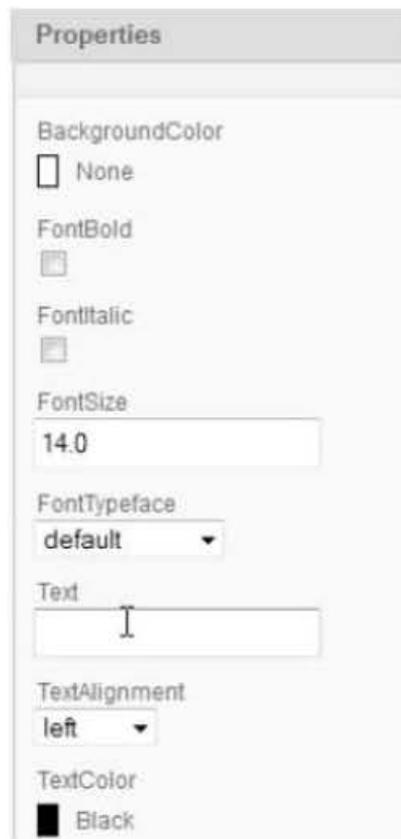
Gambar 2.80. Klik pada Basic → Label untuk memasukkan komponen label

6. Tambahkan komponen **Textbox**, dan beri properti **Text** dengan "Skor".



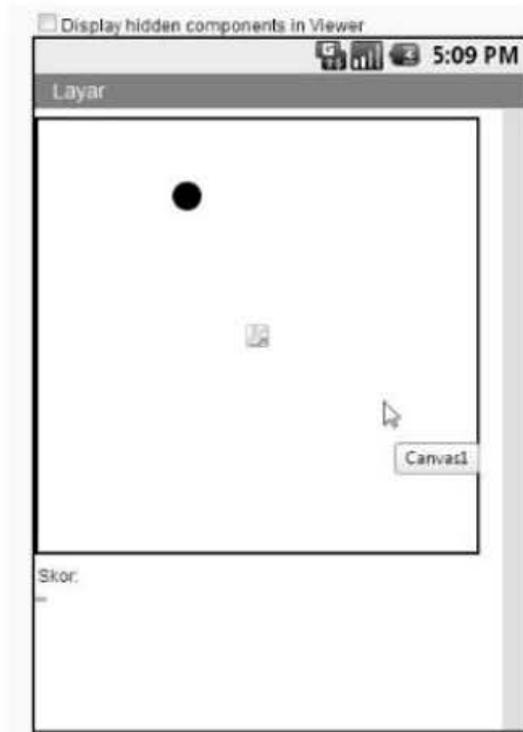
*Gambar 2.81. Pemberian properti Text*

7. Tambahkan lagi komponen **Textbox** namun kali ini kosongkan properti **Text**-nya.



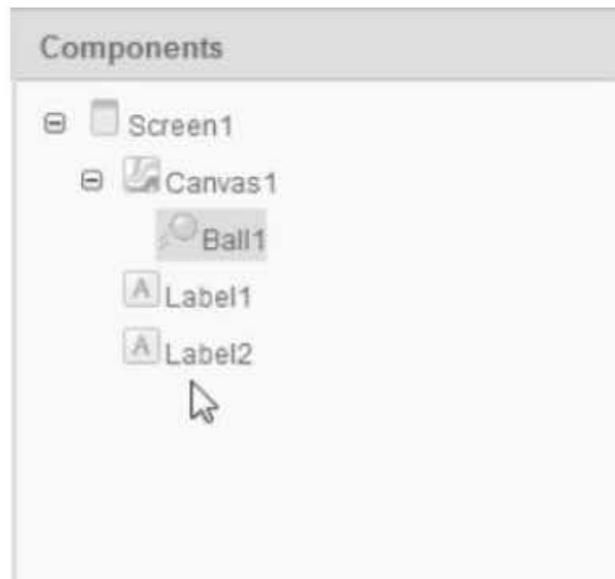
*Gambar 2.82. Mengkosongkan textbox kedua*

8. Tampilan dari layar terlihat seperti berikut ini:



*Gambar 2.83. Tampilan komponen-komponen di layar*

9. Di panel **Components**, terdapat komponen-komponen yang sudah Anda masukkan.



*Gambar 2.84. Komponen-komponen yang ada di dokumen*

10. Pertama buat dulu set variabel dengan nama "nilai", Anda bisa klik pada sembarang tempat, kemudian klik **Def** → **Variable** dan beri nama nilai as Numeric.
11. Berikutnya set variabel ini dengan nilai nol, caranya dikaitkan dengan event **Screen1.Initialize**.



*Gambar 2.85. Klik pada Screen1.Initialize*

12. Klik pada **My Blocks** → **My Definitions** → **Set global nilai**.



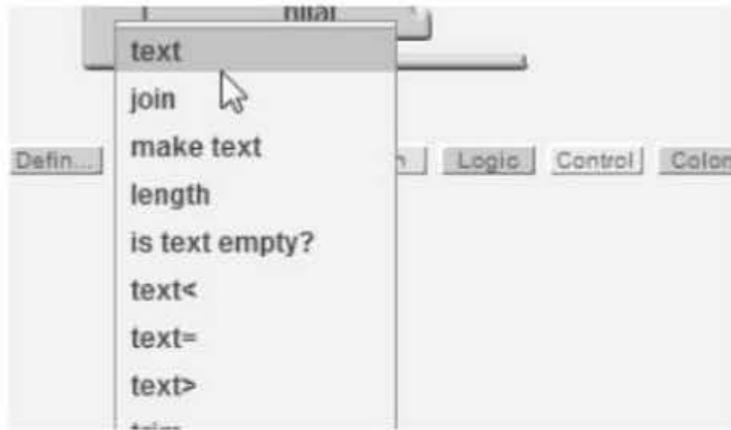
*Gambar 2.86. Klik pada Set global nilai*

13. Pasangkan blok **Set Global nilai to** di bawah **When Screen Initialize do**.



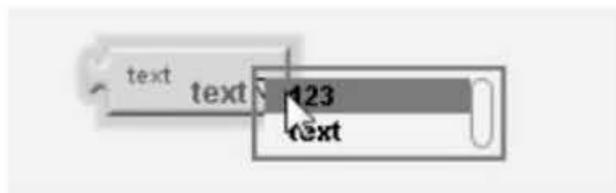
*Gambar 2.87. Memasukkan Set Global Nilai ke When Screen Initialize Do*

14. Kemudian klik di sembarang tempat, dan klik **Text** → **Text**.



Gambar 2.88. Memasukkan text

- Ubah tipenya menjadi numerik.



Gambar 2.89. Mengubah tipe ke numerik

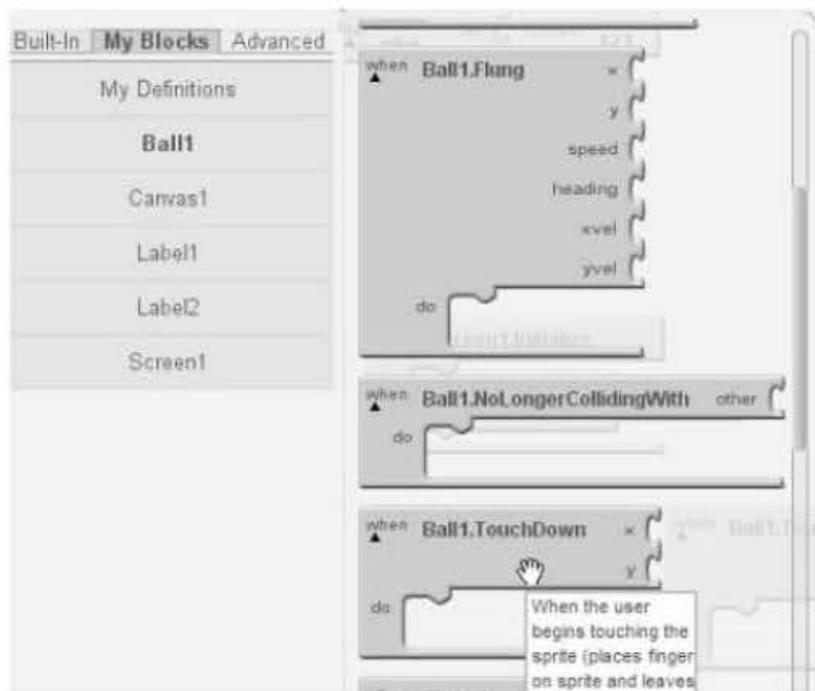
- Kemudian set nilai awal variabel nilai ini ke angka 0.

Screenshot saved



Gambar 2.90. Pengesetan nilai awal ke angka nol

- Pada contoh ini kode dimasukkan di event **touchdown** dari ball, yaitu ketika ball diklik. Karena itu klik **My Blocks** → **Ball1** → **Ball1.Touchdown**.



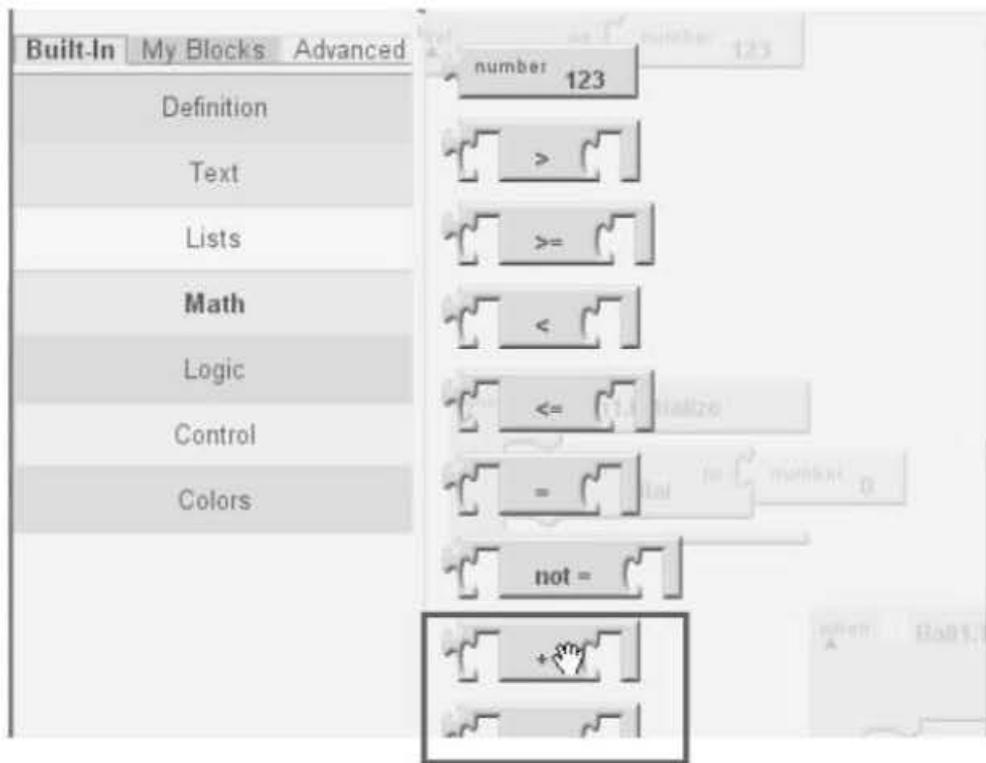
*Gambar 2.91. Ball1.Touchdown*

18. Pada contoh ini setiap kali ada event klik, maka global nilai akan diset, karena itu masukkan blok **Set global nilai** di bawah **When ball1.Touchdown**.



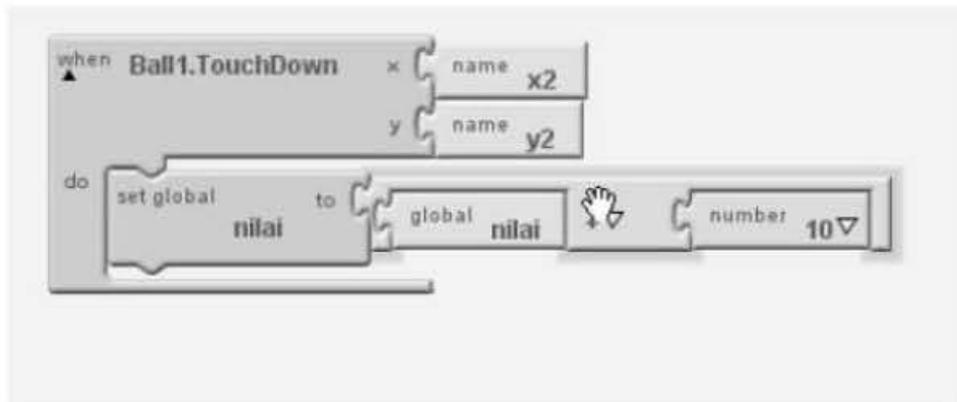
*Gambar 2.92. Set global nilai ke nilai tertentu*

19. Selanjutnya klik pada **Built in** → **Math** → **Penambahan**.



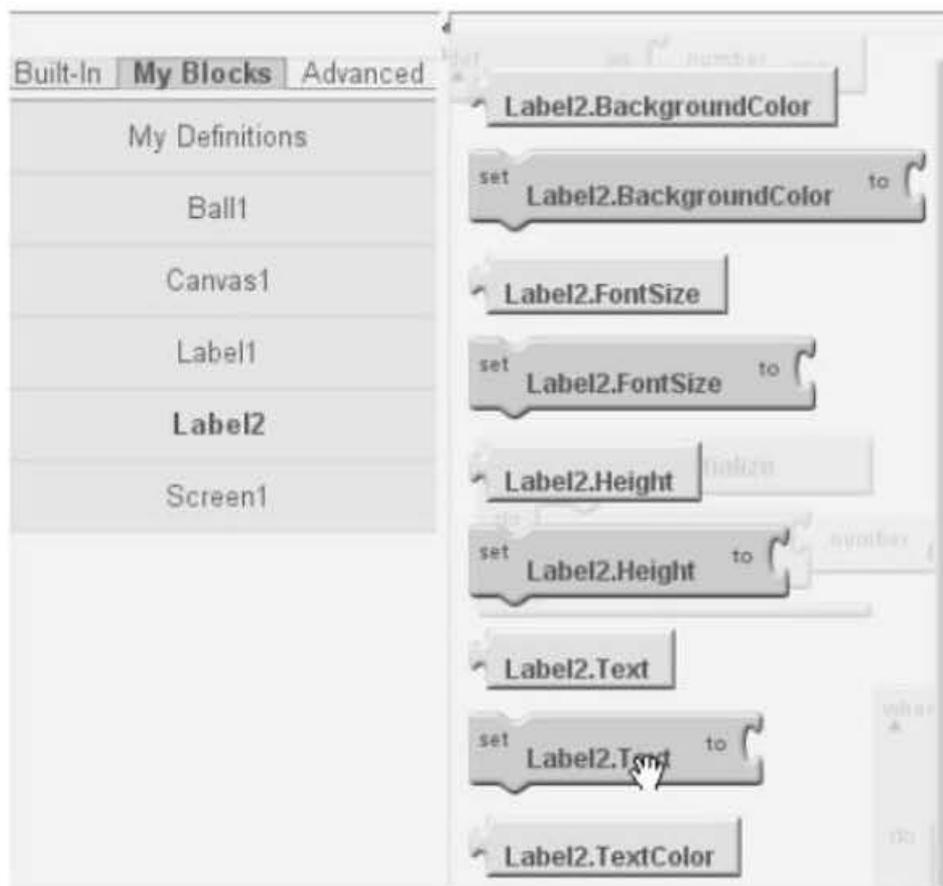
Gambar 2.93. Klik pada Built in → Math → Penambahan untuk operasi penambahan

20. Kemudian set agar global nilai diinkremenkan dengan nilai 10.



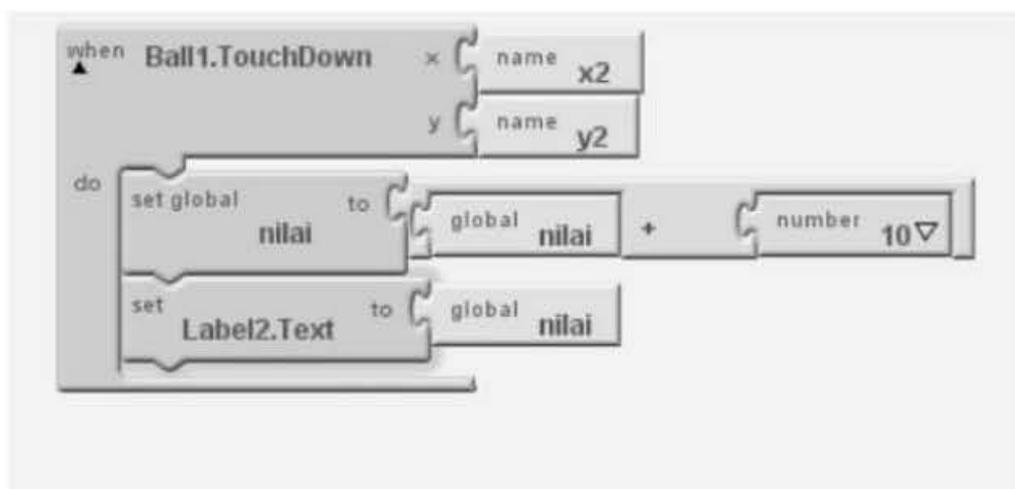
Gambar 2.94. Nilai variabel nilai ditambahkan dengan 10

21. Kemudian set Label2.Text ke global nilai, caranya masukkan dulu My Blocks → Label2 → Set Label2.Text.



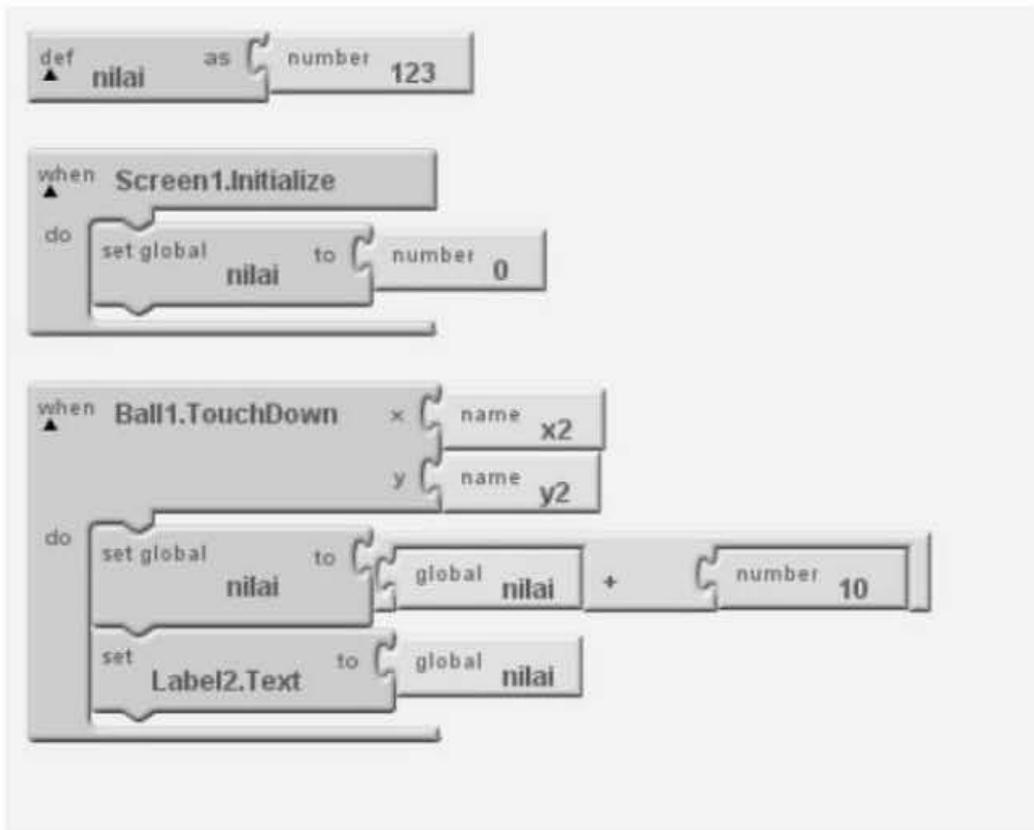
*Gambar 2.95. Memasukkan blok set label2.text*

22. Masukkan blok **Label2.Text** ke variabel **Nilai**.



*Gambar 2.96. Memasukkan blok Label2.Text ke variabel nilai*

23. Jadinya, blok lengkap seperti berikut ini:



*Gambar 2.97. Memasukkan blok lengkap*

24. Jalankan aplikasi, maka pertama skor masih kosong. Klik pada animasi bola yang berjalan.

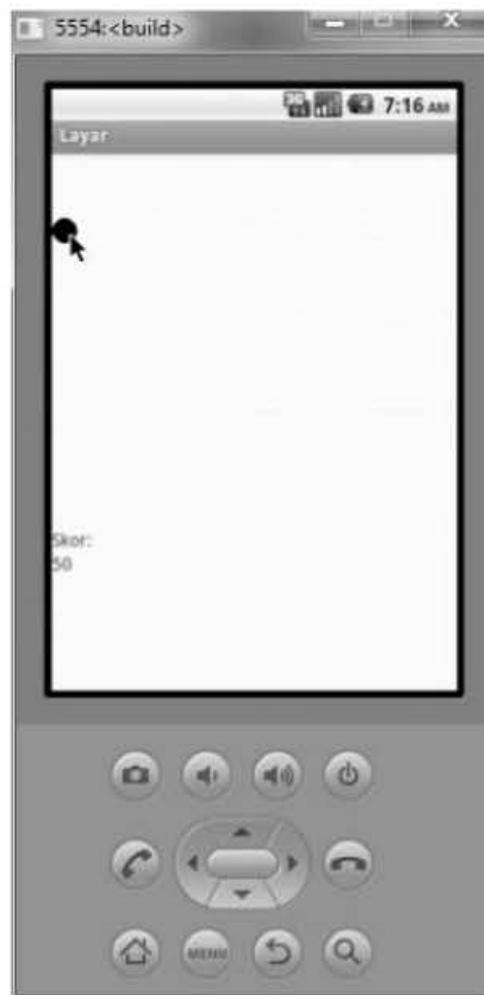


*Gambar 2.98. Kondisi awal skor masih kosong*

25. Kalau diklik, nilai akan bertambah 10 dan trus sesuai dengan kelipatan 10.



*Gambar 2.99. Setelah diklik, nilai bertambah 10*



*Gambar 2.100. Nilai akan terus bertambah*

# 3

## MEMBUAT APLIKASI PILIH GAMBAR

*Bab ini membahas:*

- ❖ Membuat Proyek.
- ❖ Menyusun Block Program

Pada bab ketiga ini, Anda akan diberikan contoh membuat aplikasi pilih gambar. Pada aplikasi pilih gambar ini, akan disediakan sebuah button untuk memilih sebuah gambar yang kemudian gambar tersebut akan ditampilkan.

### Membuat Proyek dan User Interface

Untuk membuat aplikasi pilih gambar menggunakan App Inventor ini, Anda buat sebuah Project baru, kemudian buat pula user interface sesuai dengan langkah-langkah dibawah ini:

1. Klik pada **My Project** kemudian klik tombol **New**.
2. Setelah muncul kotak **New App Inventor for Android Project** isikan nama projek "PilihGambar" di kolom **Project name**.



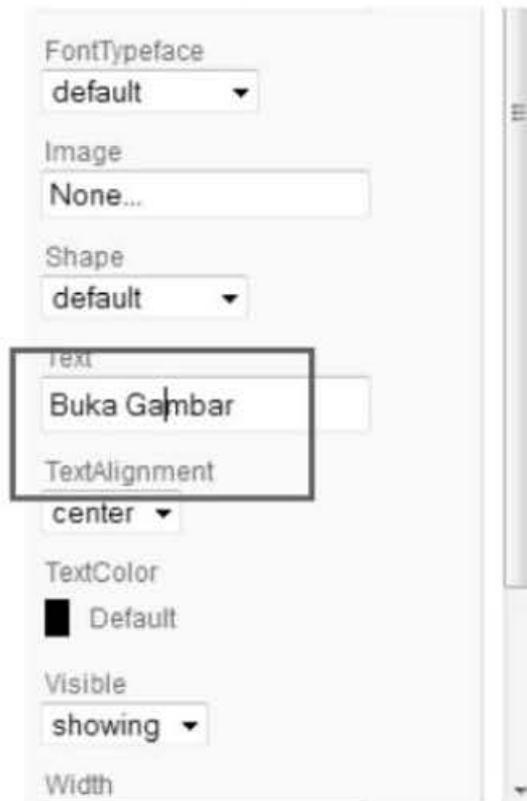
*Gambar 3.1. Mengisi nama projek untuk aplikasi PilihGambar*

3. Setelah project terbuat klik **Media** → **ImagePicker** untuk memasukkan komponen **ImagePicker**.



*Gambar 3.2. Klik pada ImagePicker*

4. Pada properti **Text** ubah menjadi "Buka Gambar".



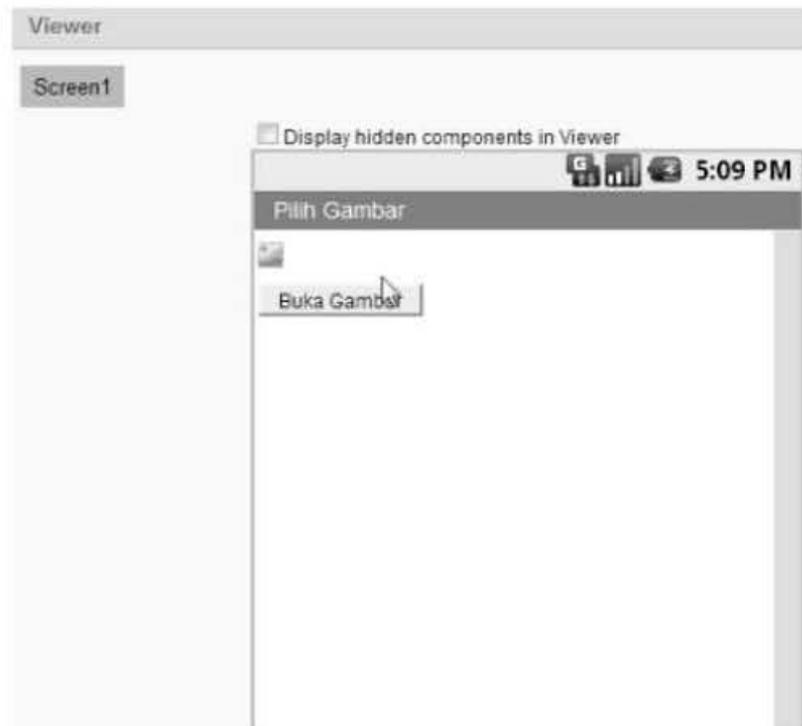
*Gambar 3.3. Mengganti properti Text*

5. Kemudian klik pada **Basic** → **Image** untuk memasukkan komponen **Image**.



*Gambar 3.4. Klik pada Image untuk memasukkan komponen Image*

6. Hasilnya, komponen **Image** dan **Button** akan terlihat di panel **Viewer** berikut ini:

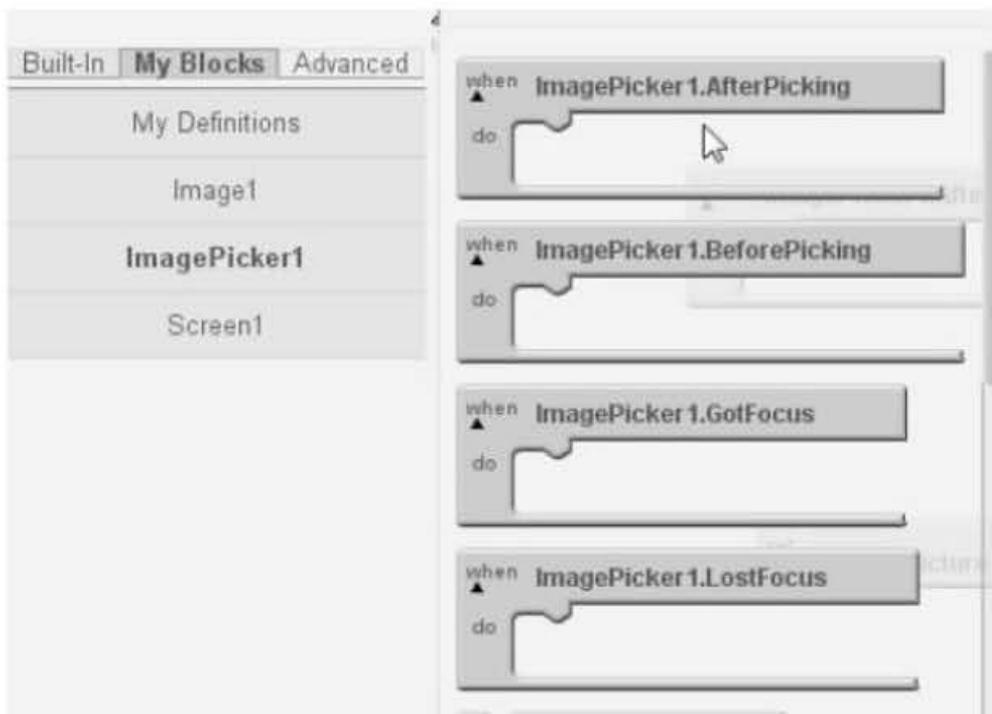


*Gambar 3.5. Image dan Button akan di panel Viewer*

## Menyusun Block Program

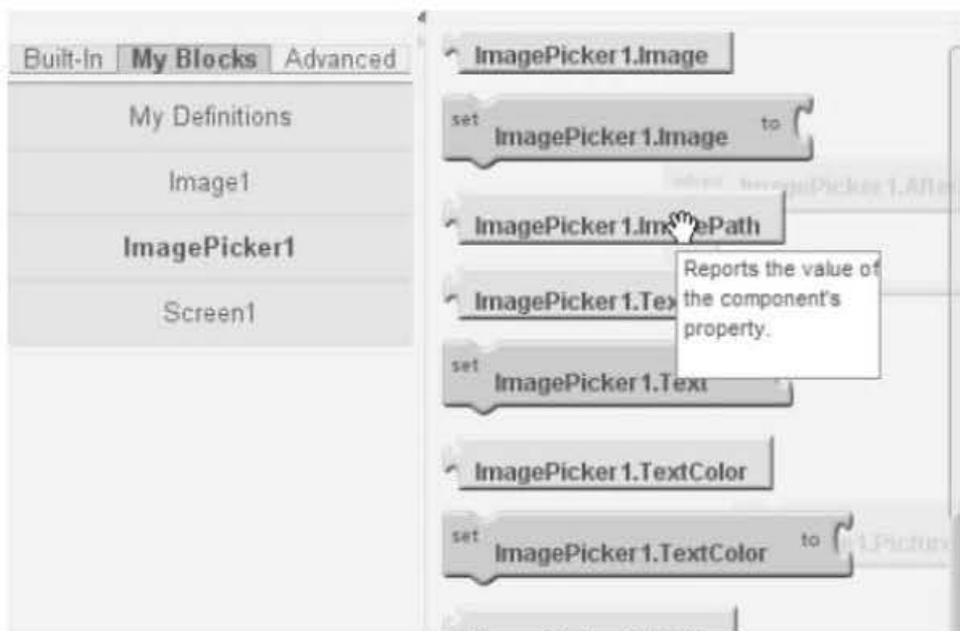
Setelah membuat user interface, saatnya Anda membuat block yang akan mengatur bagaimana flow dari program ini berjalan. Berikut ini langkah-langkahnya:

1. Klik pada **My Blocks** → **ImagePicker1**, kemudian klik pada event **When ImagePicker1.AfterPicking**.



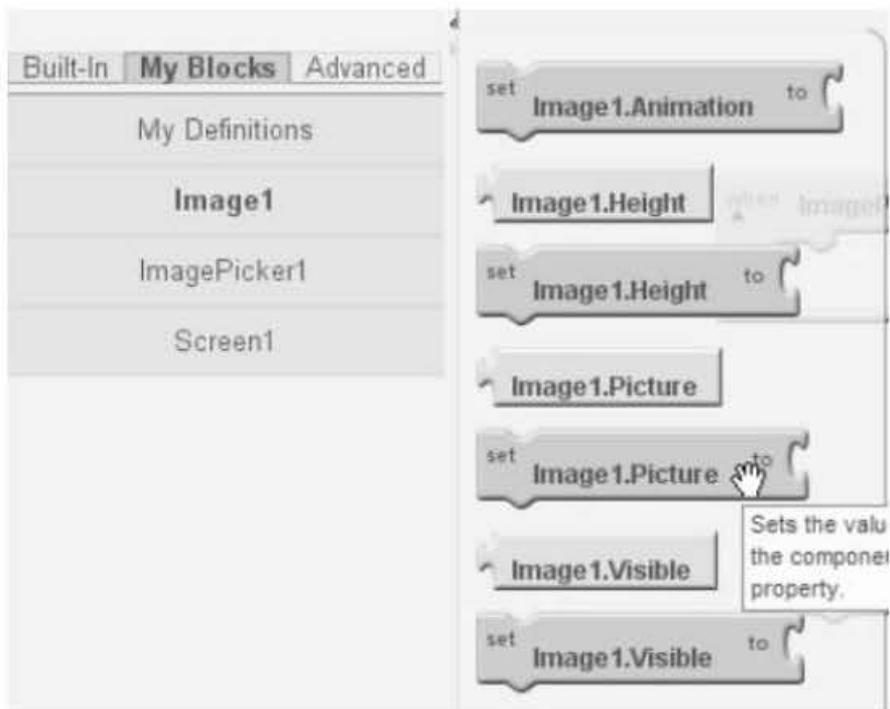
*Gambar 3.6. When ImagePicker1.AfterPicking*

2. Kemudian scroll ke bawah dan ambil blok **ImagePicker1.ImagePath**.



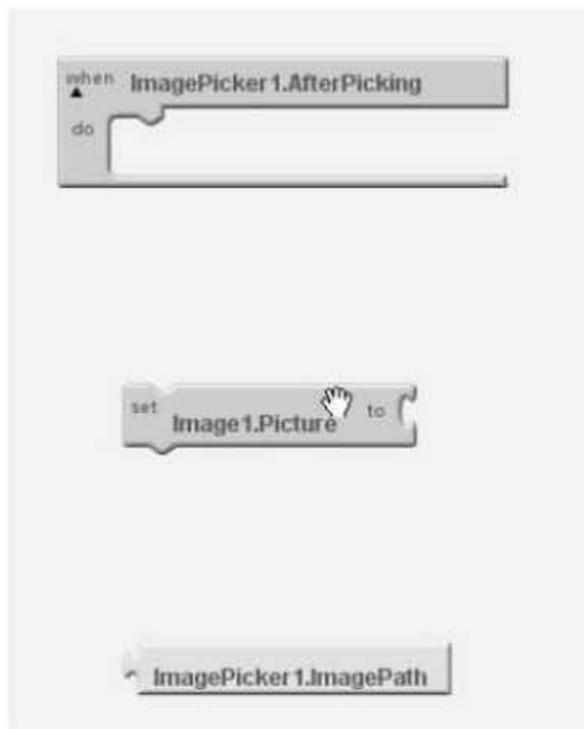
*Gambar 3.7. Klik pada ImagePicker1.ImagePath*

3. Beralih ke **My Blocks** → **Image1** kemudian klik pada method **SetImage1.Picture to**.



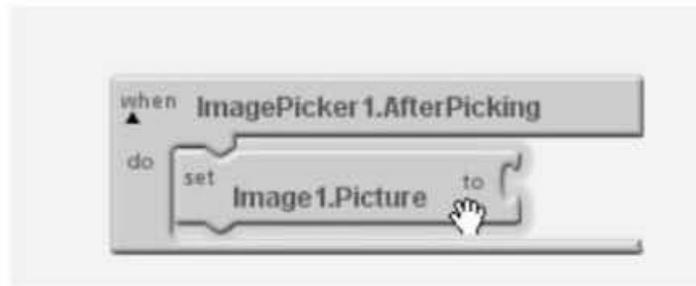
*Gambar 3.8. Set Image1.Picture to*

4. Maka akan terdapat 3 blok di dalam **Designer Block**.



*Gambar 3.9. Terdapat tiga blok di Designer Block*

5. Masukkan method **Set Image1.Picture** ke event **When ImagePicker1.AfterPicking**.



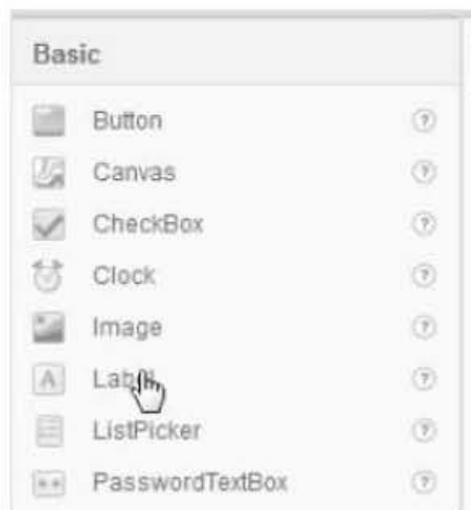
*Gambar 3.10. Memasukkan Set Image1.Picture ke event ImagePicker1.AfterPicking*

6. Kemudian pasangkan ImagePicker1.ImagePath ke Set Image1.Picture.



*Gambar 3.11. Memasangkan ImagePicker1.ImagePath ke Set Image1.Picture*

7. Agar *path* atau lokasi direktori dari image yang dibuka nanti bisa diperlihatkan, maka tambahkan sebuah komponen **Label** dengan mengklik Pallete **Basic** → **Label**.



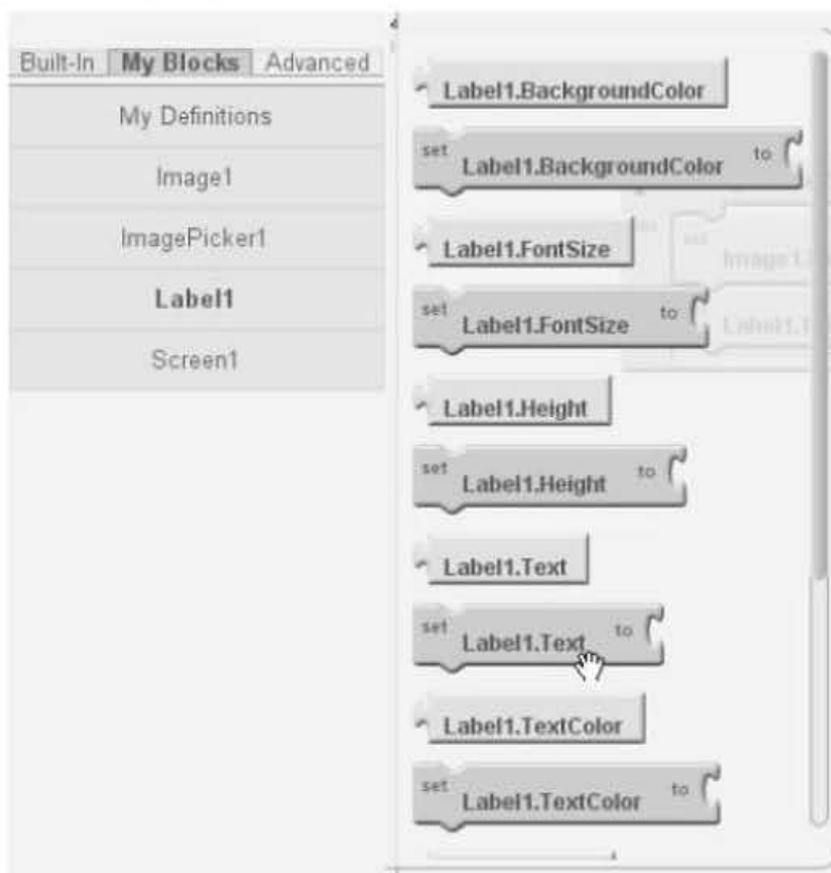
*Gambar 3.12. Klik pada Basic → Label*

8. Kosongkan properti **Text** dari **Label** tersebut.



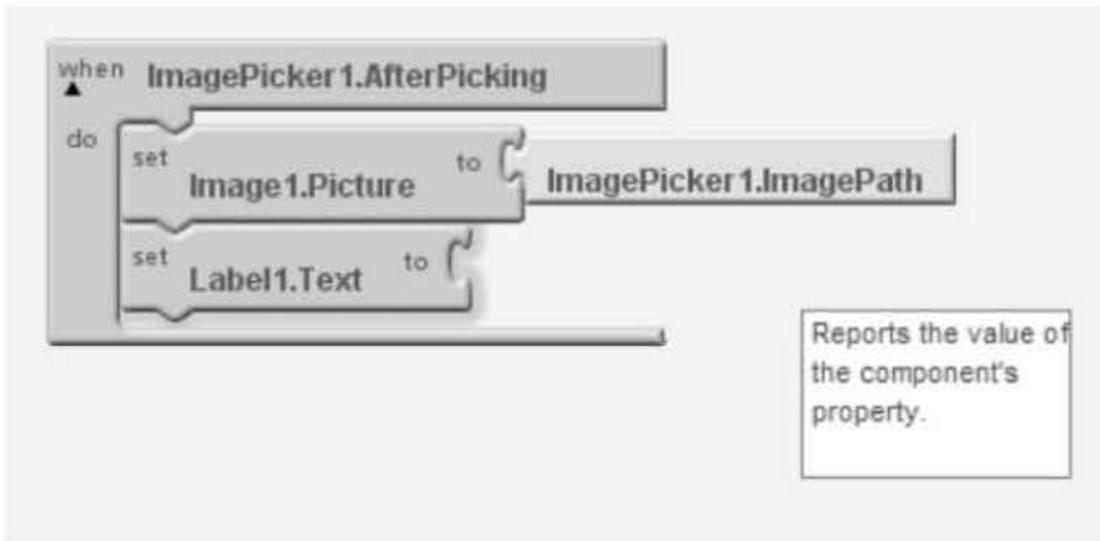
Gambar 3.13. Mengkosongkan properti Text dari Label

9. Kemudian beralih ke **Block Designer**, klik pada **My Blocks** → **Label1 Set Label1.Text**.



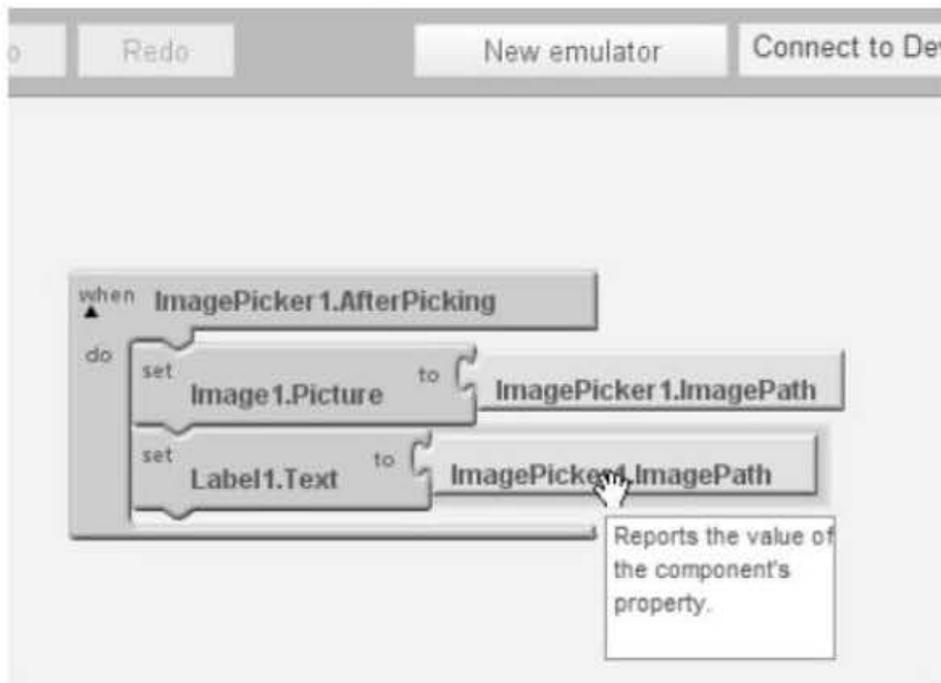
Gambar 3.14. Pengesetan ke My Blocks → Label 1

10. Dan letakkan block tersebut di bawah **Set Image1.Picture**, artinya setelah **Image1** menampilkan image, path dari image tersebut juga ditampilkan di label.



*Gambar 3.15. Meletakkan path ke Label1.Text*

11. Salin nilai **ImagePicker.ImagePath** dan letakkan di **set Label1.Text**.



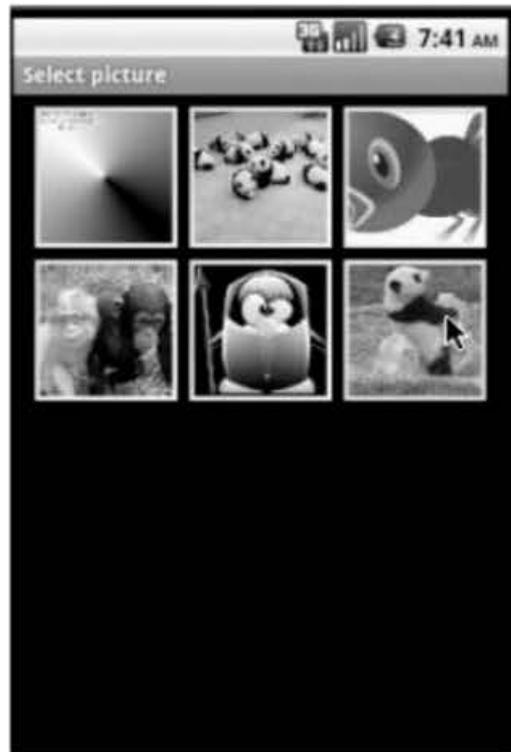
*Gambar 3.16. Penyalinan nilai ImagePicker1.ImagePath*

12. Dan jika sudah selesai, jalankan aplikasi. Klik pada **Buka Gambar**.



*Gambar 3.17. Klik pada Buka Gambar*

13. Pilih gambar sesuai dengan keinginan Anda.



*Gambar 3.18. Memilih gambar yang ada*

14. Maka gambar yang terpilih akan ditampilkan, dan Anda bisa melihat *path* dari file gambar tersebut dibagian bawah.



*Gambar 3.19. Path dari gambar terlihat di Label*

# 4

## MEMBUAT APLIKASI SLIDESHOW

*Bab ini membahas:*

- ❖ Membuat Aplikasi Slideshow Otomatis.
- ❖ Membuat Aplikasi Slideshow Manual.
- ❖ Membuat Aplikasi Slideshow Random.

Pada bab ini Anda akan membuat sebuah aplikasi slideshow menggunakan App Inventor. Aplikasi yang akan dibuat ini berguna untuk menampilkan gambar tertentu di media. Terdapat 3 jenis aplikasi yang akan dibuat yaitu, aplikasi slideshow otomatis, aplikasi slideshow manual, serta aplikasi slideshow random.

### Membuat Aplikasi Slideshow Otomatis

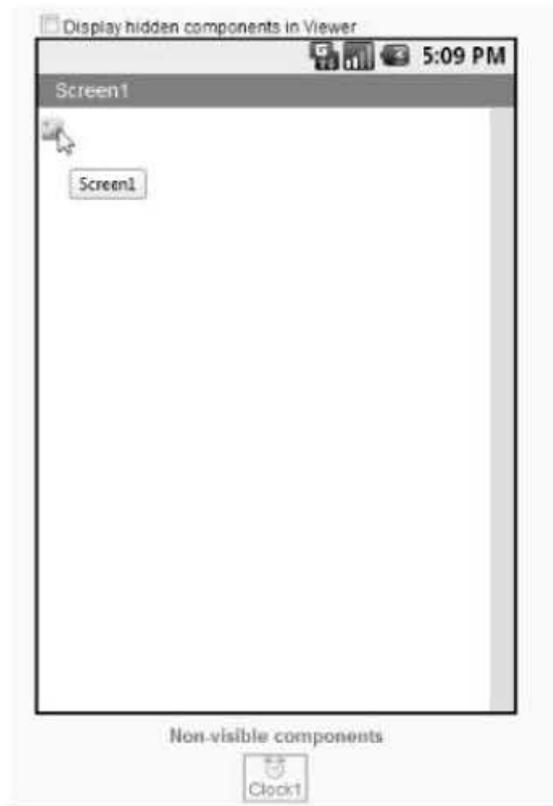
Pada proyek pertama ini, Anda akan membuat sebuah program yang akan menampilkan gambar yang ada di media secara otomatis menggunakan komponen Timer, sehingga user tidak perlu melakukan tindakan apapun. Tahapan membuat aplikasi slideshow otomatis ini seperti berikut:

1. Klik pada **My Projects** → **New** untuk membuat project baru.
2. Buat nama proyek "SlideShow" dan klik **OK** untuk menyimpan project.



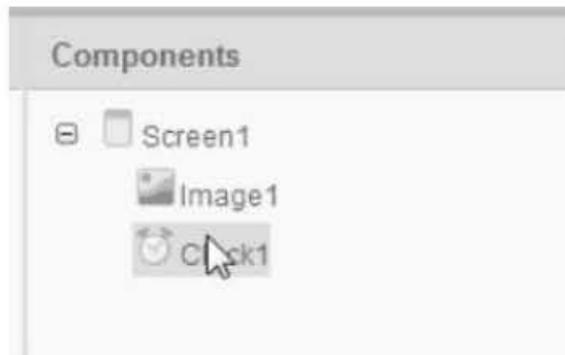
*Gambar 4.1. Mengisi nama proyek SlideShow*

3. Masukkan komponen **Image** ke dalam Screen, dan masukkan juga komponen **Clock1** ke dalamnya.



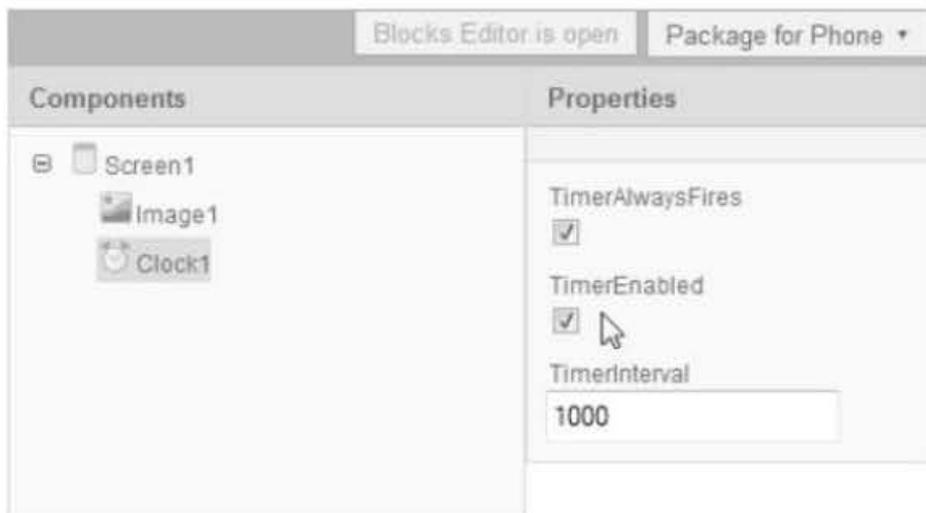
*Gambar 4.2. Memasukkan komponen Image dan Clock*

4. Di panel **Components**, pastikan komponen **Image** dan **Clock** sudah ada, biarkan dengan nama standarnya, yaitu **Image1** dan **Clock1**.



*Gambar 4.3. Image1 dan Clock1 sudah ada di komponen*

5. Untuk komponen **Clock1**, pastikan **TimerAlwaysFires** dan **TimerEnabled** dalam posisi tercentang. Kemudian tentukan **Interval** waktu Clock dipicu dalam satuan milidetik.



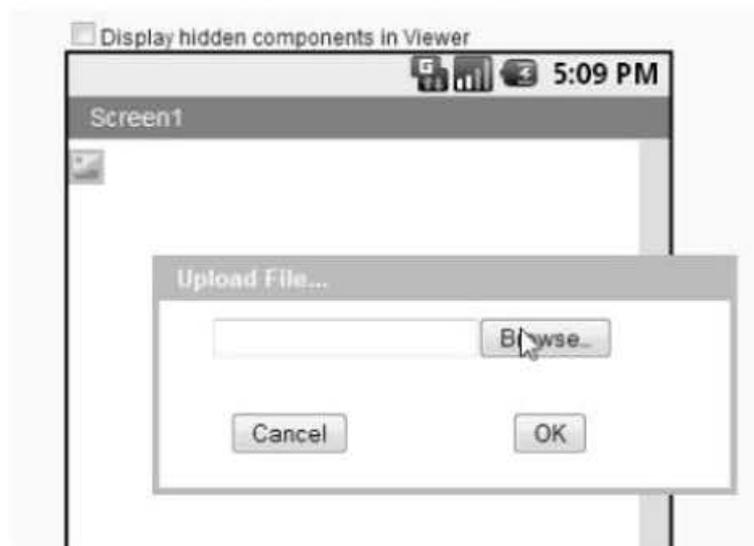
*Gambar 4.4. Pengaturan Properties dari Clock*

6. Beralih ke **Media**, pilih media yang akan dilihat atau di-slideshowkan, dalam hal ini adalah gambar. Klik tombol **Upload new**.



*Gambar 4.5. Klik pada Upload new untuk mengupload file gambar*

7. Klik **Browse** di kotak **Upload File** yang muncul.



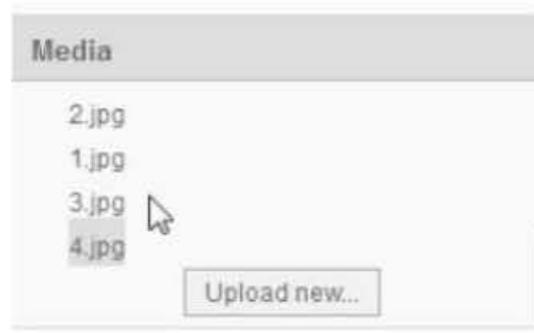
*Gambar 4.6. Klik pada Browse untuk memilih file gambar*

8. Pilih file-file gambar yang akan digunakan sebagai image slideshow, kemudian klik tombol **Open**.



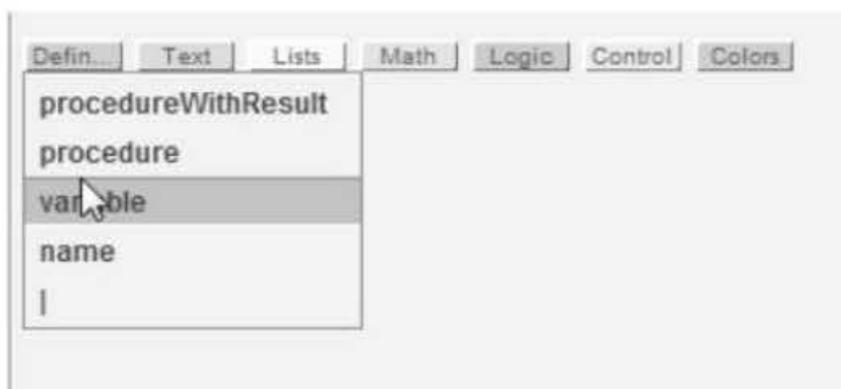
*Gambar 4.7. Pilih file-file gambar yang akan diupload*

- Untuk contoh slideshow ini digunakan 4 gambar. Berikutnya ubah nama file gambar Anda menjadi "1.jpg", "2.jpg", "3.jpg", dan "4.jpg".



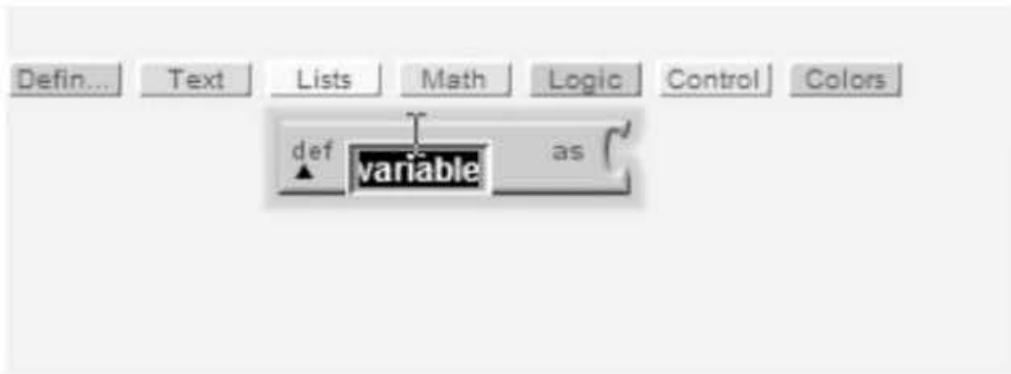
*Gambar 4.8. Memberikan nama untuk gambar yang akan dbuat slideshow*

- Sekarang beralih ke **Block Designer** untuk mengatur *logik* dari program. Buat dulu variabel dengan klik pada sembarang tempat di variabel, kemudian klik pada **Define** → **Variable**.



*Gambar 4.9. Define Variable*

- Beri nama variabel dengan klik pada teks **Variable** yang ada.



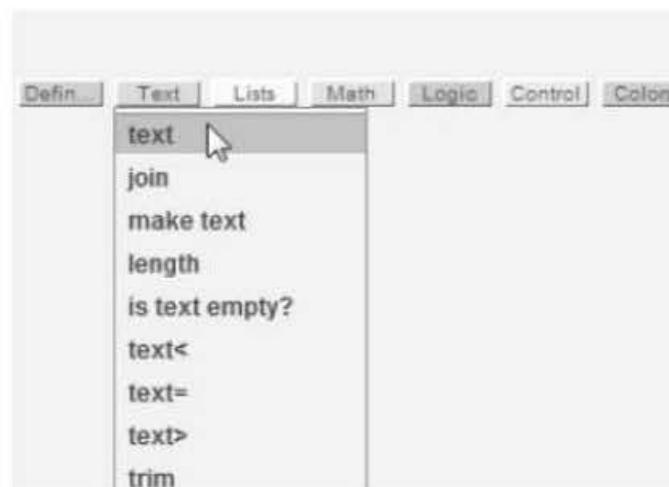
*Gambar 4.10. Klik pada nama Variable untuk memberi nama*

12. Ubah namanya menjadi “nomor” untuk mendefinisikan variable nomor.



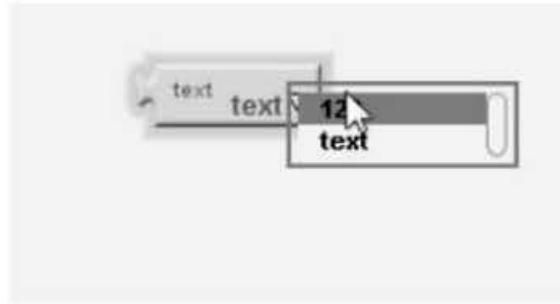
*Gambar 4.11. Mengganti nama menjadi “nomor” untuk variabel*

13. Variable tersebut perlu didefinisikan jenisnya, apakah teks atau numeric. Karena itu klik pada sembarang tempat dan klik **Text** → **Text**.



*Gambar 4.12. Menentukan jenis variable text*

14. Berikutnya tentukan tipe data untuk variabel nomor ini. Pada contoh ini, variabel nomor dibuat dengan tipe numurik, karena itu klik tulisan “text” kemudian ganti dengan jenis numerik dengan memilih angka 123.



Gambar 4.13. Memilih tipe numeric

15. Kalau sudah, maka blok yang menyatakan jenis variabel tersebut akan bertuliskan **number** dari sebelumnya **text**.



Gambar 4.14. Blok text berubah jadi number

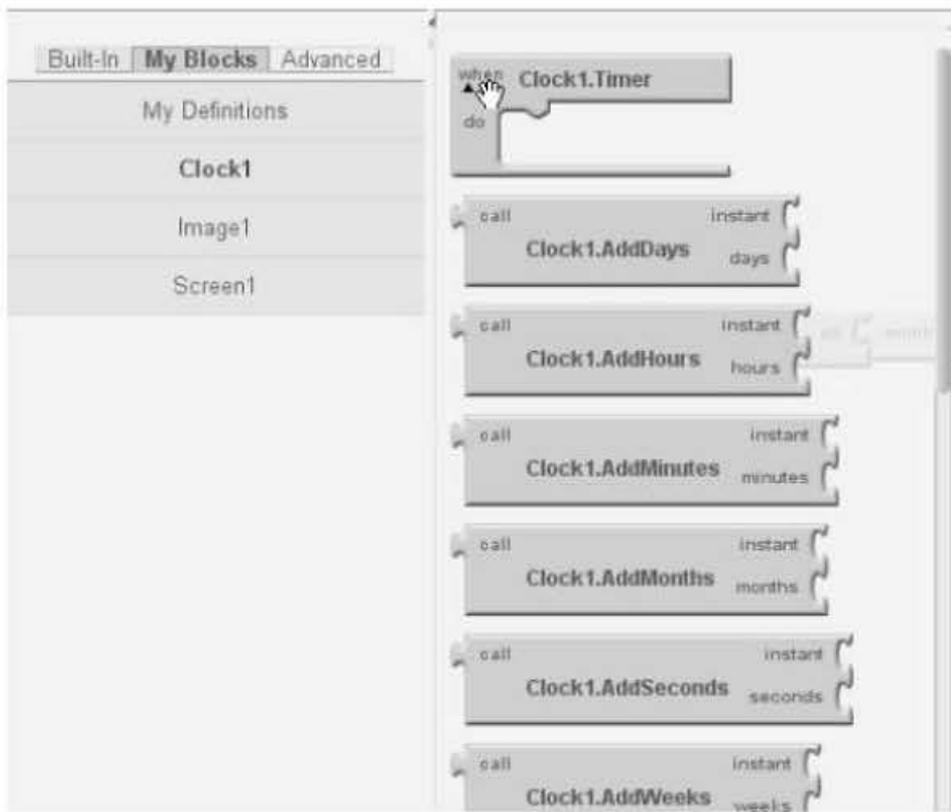
16. Kemudian pasangkan number tersebut ke blok def nomor as. Ini artinya variable nomor berubah menjadi numerik.



Gambar 4.15. Variable nomor akan berubah menjadi numerik

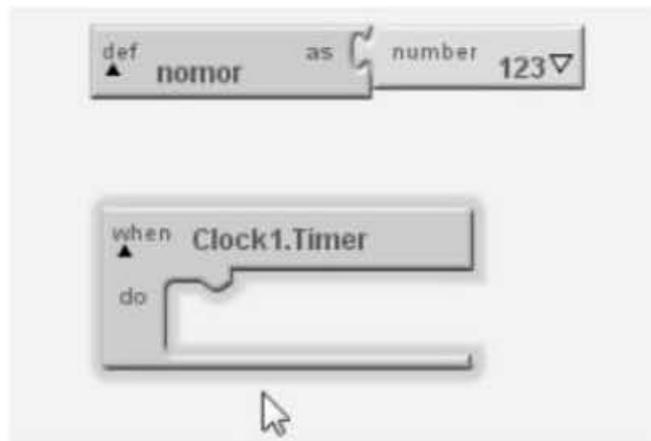
17. Sekarang buat event yang digunakan untuk memicu kode dengan klik **My Blocks** → **Clock1**. Kemudian pilih event **When Clock1.Timer do**.

Screenshot saved



*Gambar 4.16. Klik pada blok event Clock1.Timer*

18. Drag blok tersebut ke designer sehingga sejauh ini pengaturan blok menjadi seperti berikut ini:



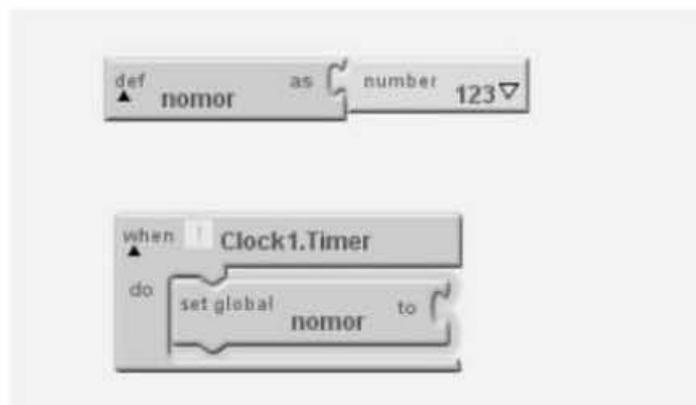
*Gambar 4.17. Blok Clock1.Timer sudah ditambahkan*

19. Pada contoh ini akan dibuat jika event timer muncul, variable nomor diset ke nilai terakhirnya. Karena itu klik pada **My Blocks** → **My Definitions** → **Set Global nomor to**.



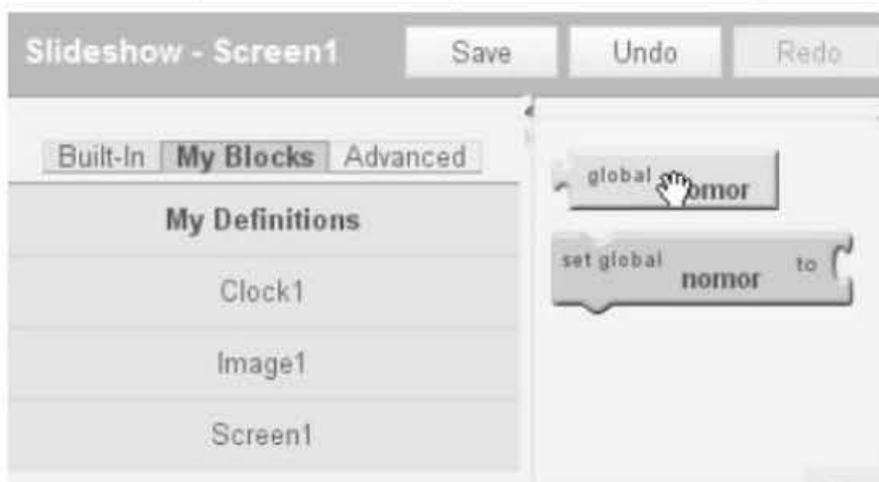
*Gambar 4.18. Klik pada blok untuk mengeset variable Global Nomor*

20. Masukkan blok tersebut ke event **When Clock1.Timer do.**



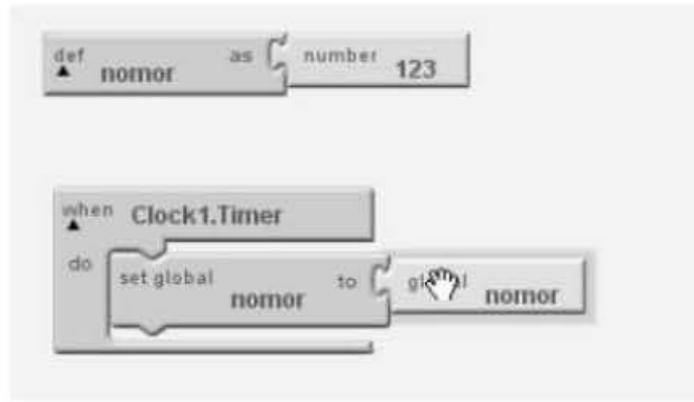
*Gambar 4.19. When Clock1.Timer do Set variable Global ke nilai dari variable global terakhir*

21. Kemudian klik **My Blocks** → **My Definitions** → **global nomor.**



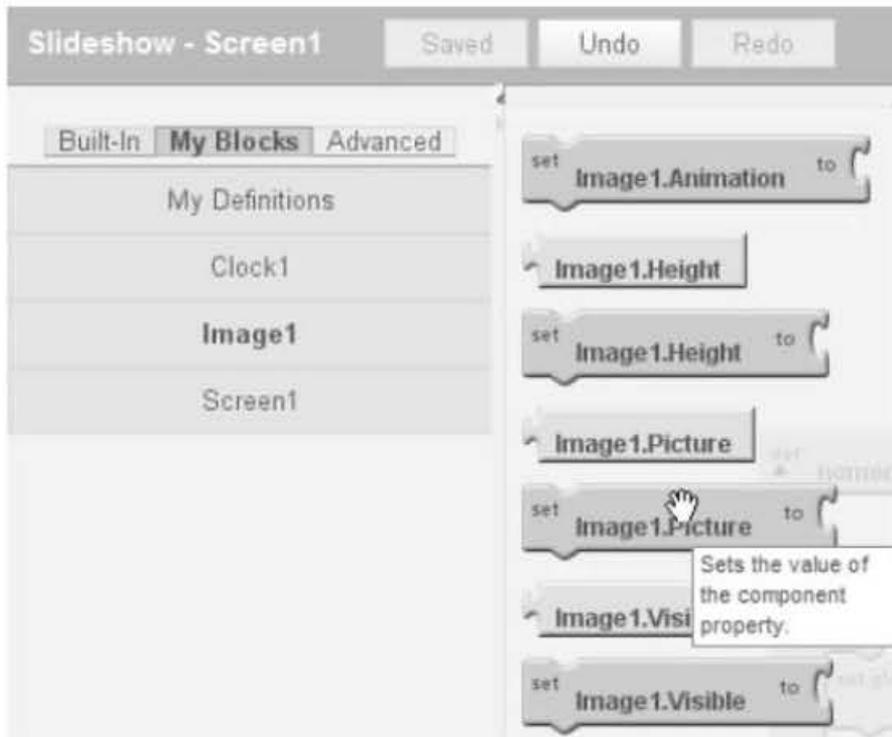
*Gambar 4.20. Klik pada nilai untuk variable nomor*

22. Letakkan blok nilai variabel **nomor** ke **set global nomor**.



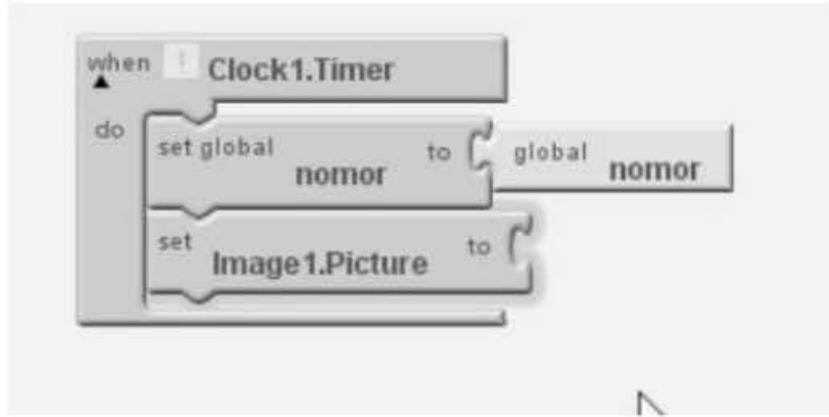
*Gambar 4.21. Mengeset variable nomor ke nilai terakhir variable ini*

23. Berikutnya akan diatur bagaimana aplikasi bisa menampilkan gambar. Klik pada **My Blocks** → **Image1** → **Set Image1.Picture to**.



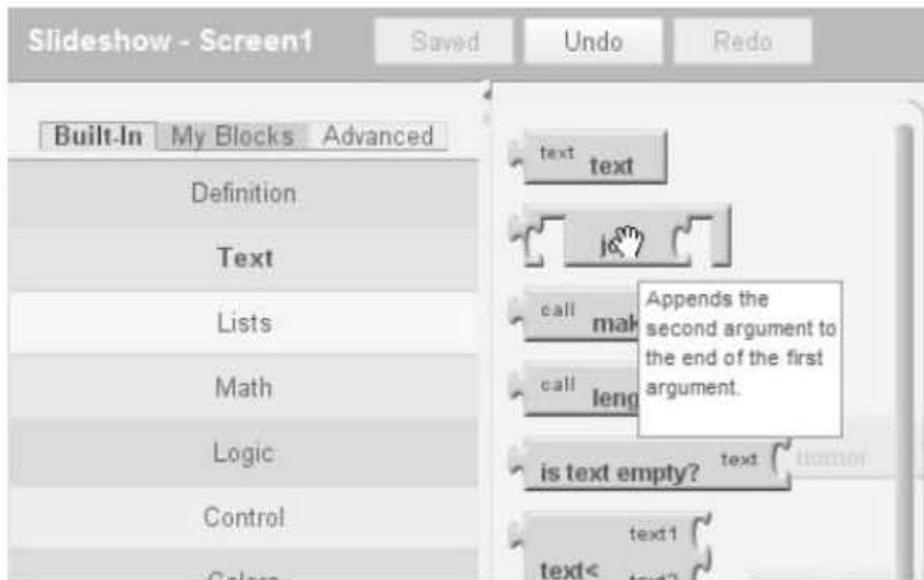
*Gambar 4.22. My Blocks → Image1 → Set Image1.Picture to*

24. Drag dan letakkan di **Set Global nomor**.



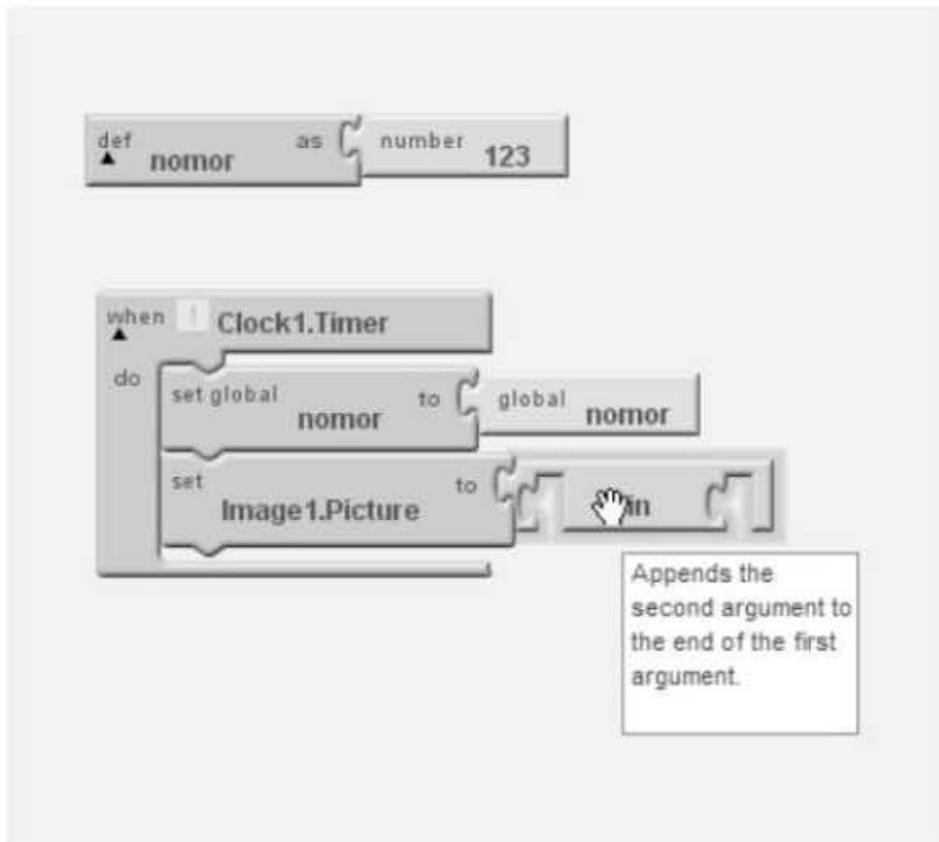
*Gambar 4.23. Meletakkan action Set Image1.Picture di bawah action Set Global Nomor*

25. Pada contoh ini dibuat aturan **Image1** membuka gambar dengan nama file nomor+".jpg". Jadi kalau nilai nomor= 1, maka akan dibuka file 1.jpg. Karena itu klik pada **Built In** → **Text** → **Join**.



*Gambar 4.24. Klik pada blok untuk Join Text*

26. Pasangkan block tersebut ke **Set Image1.Picture**.



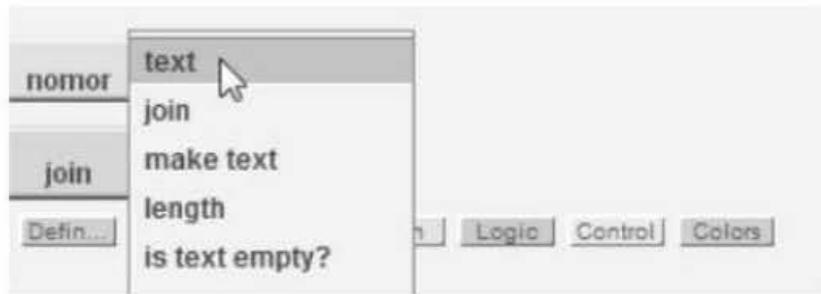
Gambar 4.25. Memasang blok Join ke Image1.Picture

27. Yang akan di-join pertama, adalah variabel nomor, karena itu klik My Blocks → My Definitions → Global nomor.



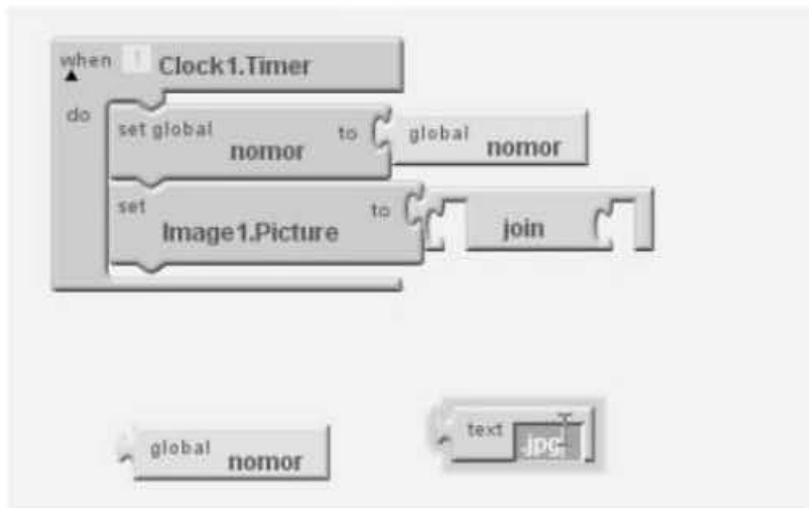
Gambar 4.26. Klik pada variabel Nomor

28. Kemudian yang akan di-join kedua adalah ekstensi dari file gambar ".jpg". Karena itu klik pada sembarang tempat, dan klik Text → Text.



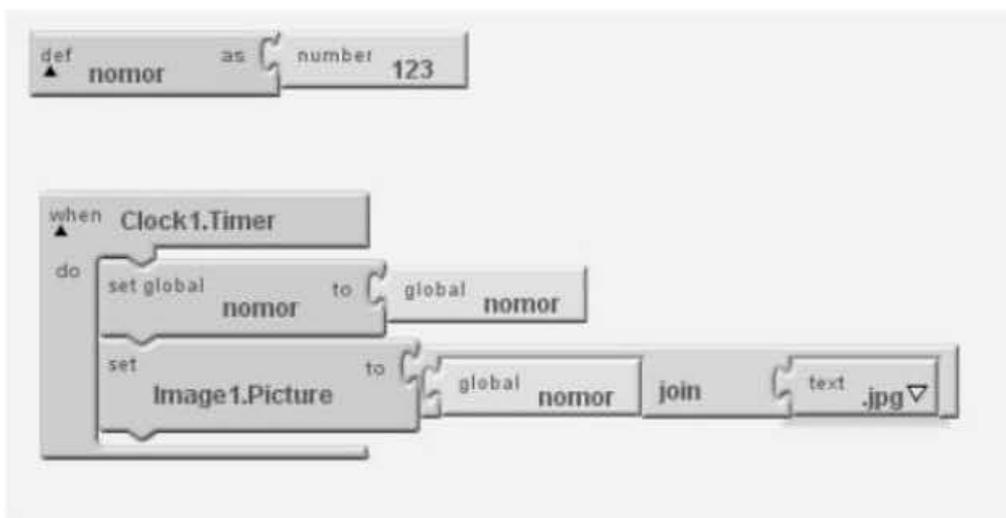
Gambar 4.27. Klik pada Text → Text untuk membuat nilai text

29. Kemudian isikan karakter “.jpg” di blok text tersebut.



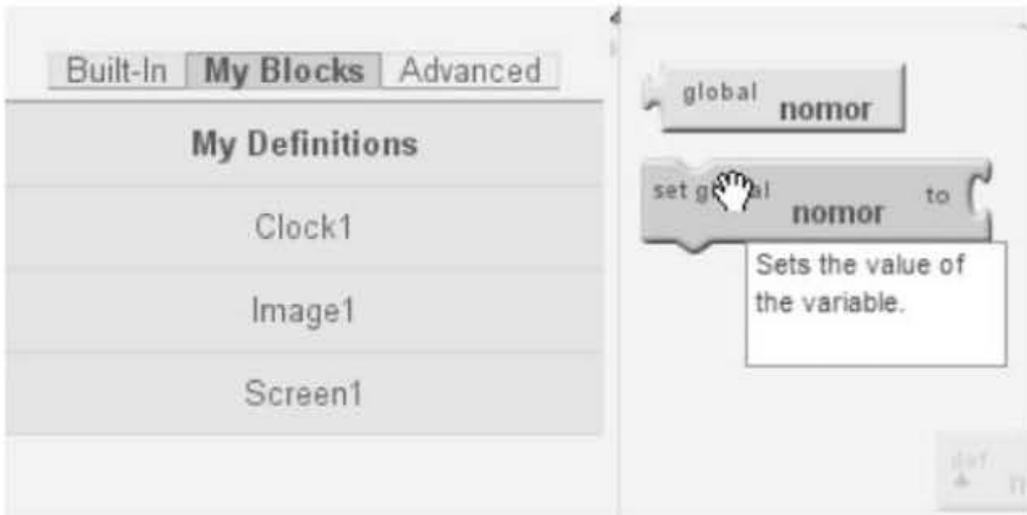
Gambar 4.28. Mengisi karakter “.jpg”

30. Kemudian masukkan ke join dan set ke Image1.Picture seperti berikut ini:



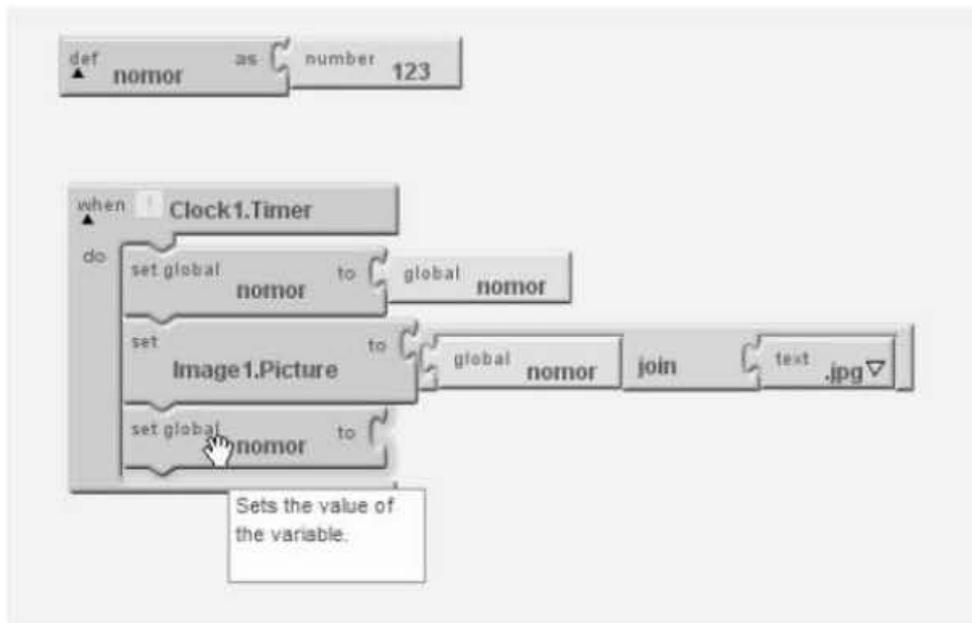
Gambar 4.29. Memasukkan ke join

31. Pada aplikasi setiap kali habis menampilkan, nilai nomor akan ditambahkan/diinkremenkan. Karena itu di bawah **Set Image1.Picture**, Anda akan menginkremenkan. Klik pada **My Blocks** → **My Definitions** → **Set Global nomor to**.



*Gambar 4.30. Set Global nomor to*

32. Masukkan di bawah **Set Image1.Picture** masih di event **When Clock1.Timer do**.



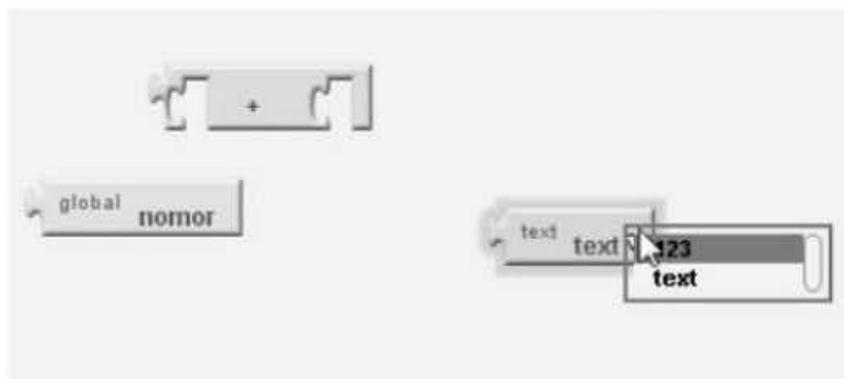
*Gambar 4.31. Memasukkan Set Global nomor ke bawah set Image1.Picture*

33. Kemudian klik **Built In** → **Math** → block penambahan.



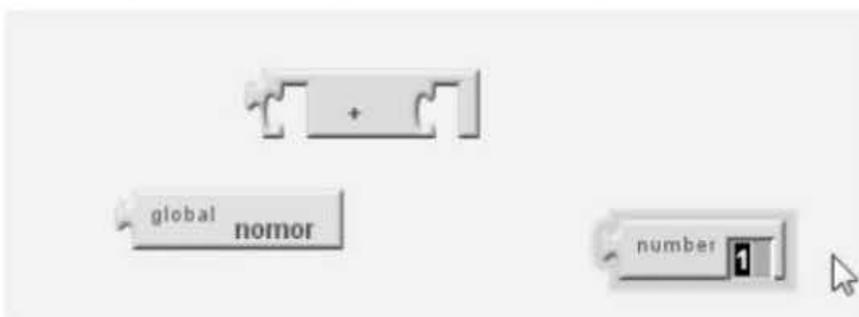
*Gambar 4.32. Menambahkan block penambahan*

34. Kemudian masukkan block **text**, kemudian ganti ke jenis numerik.



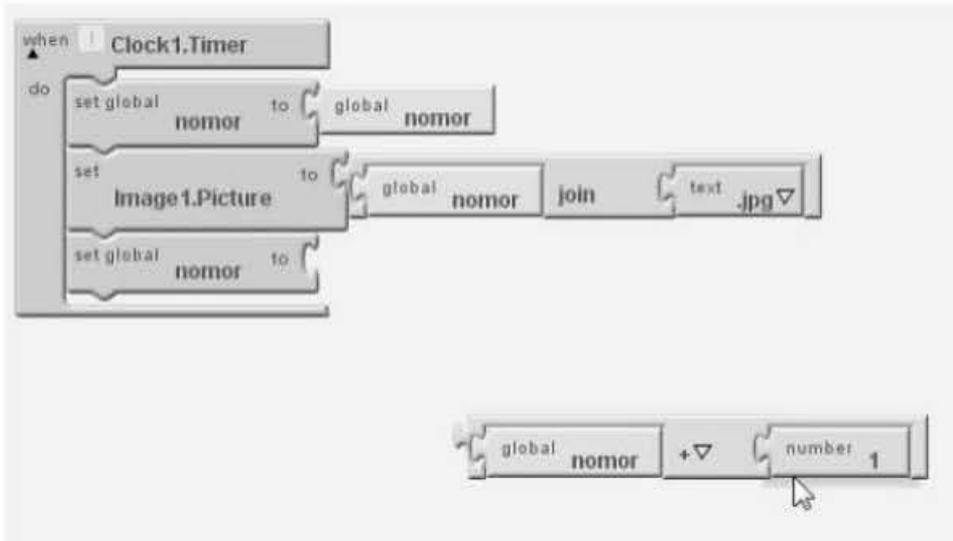
*Gambar 4.33. Mengganti nilai text menjadi bernilai numerik*

35. Kalau sudah jadi **number**, set dengan angka 1.



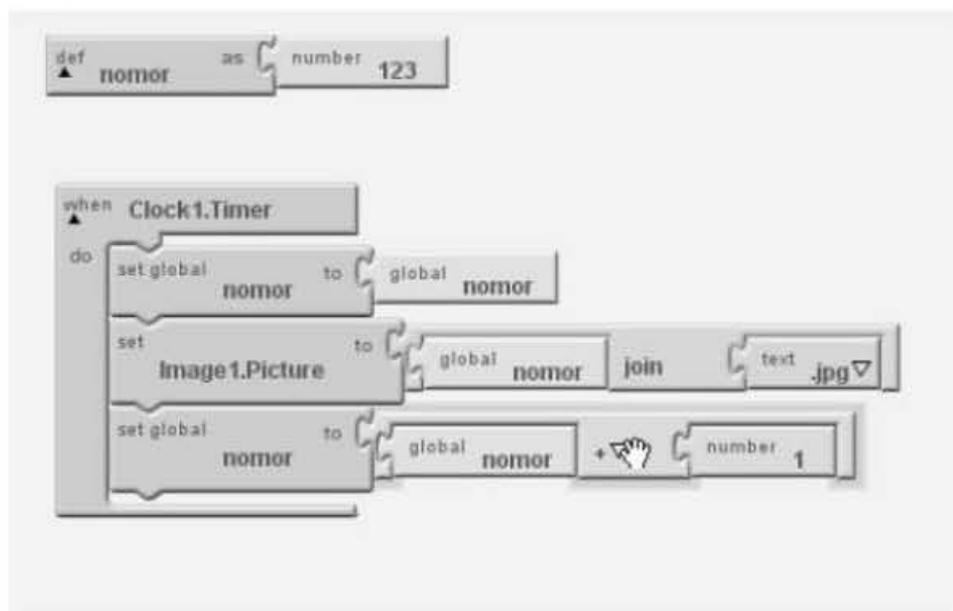
*Gambar 4.34. Set menjadi 1*

36. Maka blok untuk menginkremenkan variabel nomor sudah jadi, tinggal pasangkan saja ke **set global nomor to**.



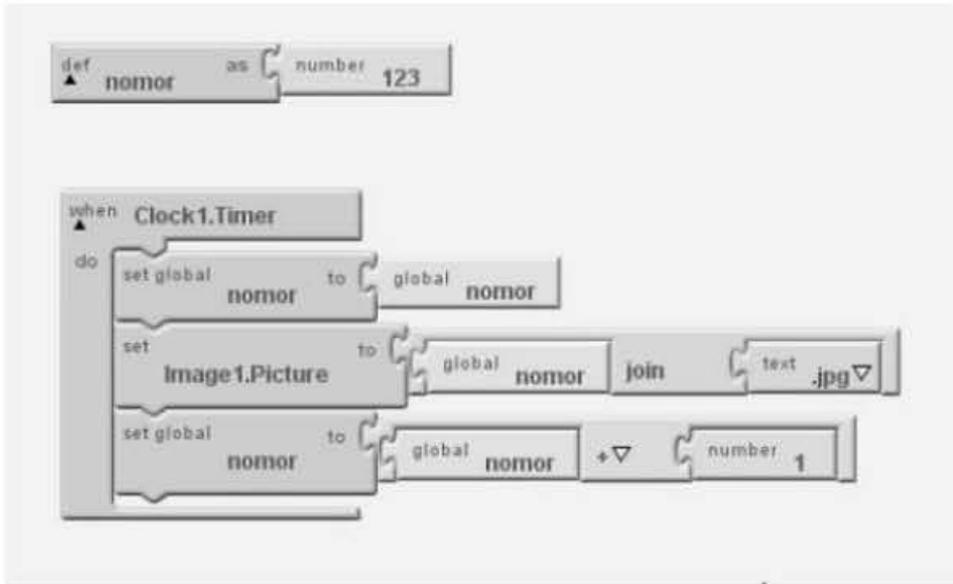
*Gambar 4.35. Blok untuk menginkremenkan variabel nomor*

37. Pasangkan blok untuk inkremen tersebut seperti berikut ini:



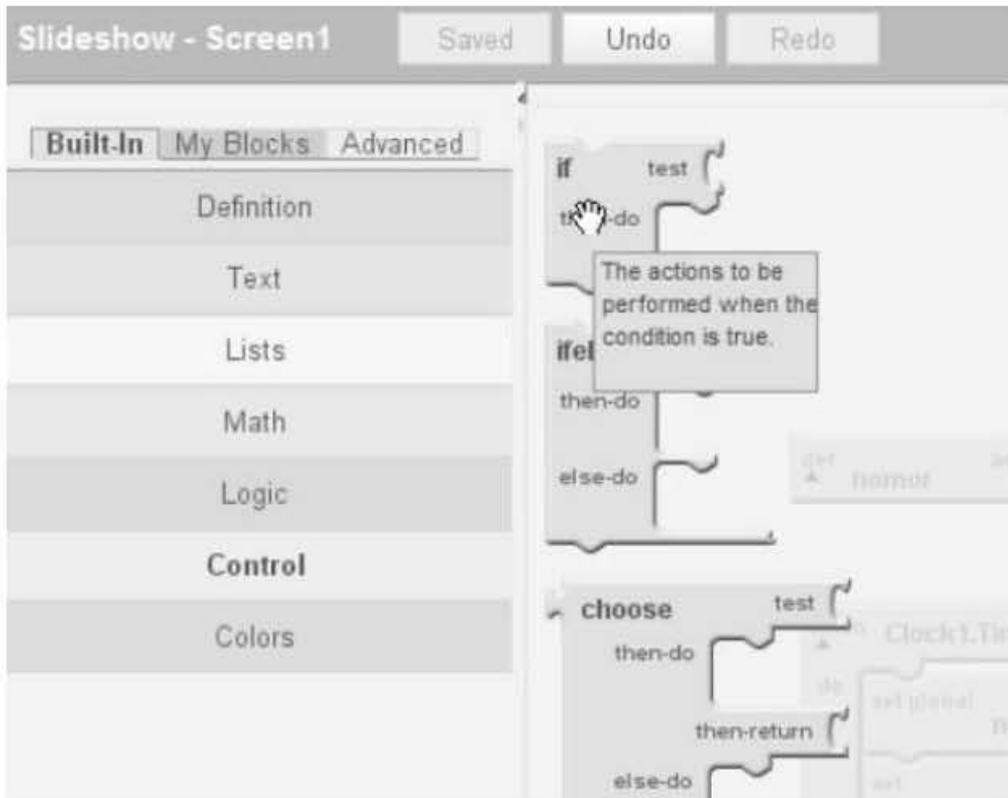
*Gambar 4.36. Memasangkan blok untuk inkremen*

38. Kalau sudah dipasangkan, terlihat tampilan seperti berikut:



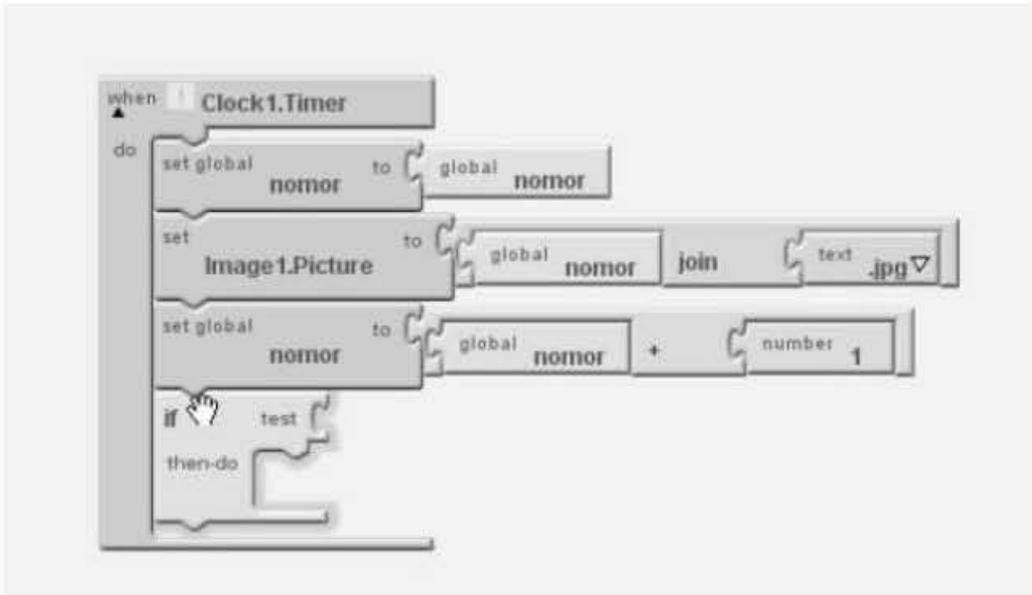
Gambar 4.37. Blok untuk inkremen sudah dipasang

- 39. Berikutnya, akan dibuat jika nomor sudah angka 5, maka dikembalikan ke 1. Ini karena file-file gambar hanya dari 1.jpg hingga 4.jpg.
- 40. Untuk melakukan ini, gunakan logic If then, dengan klik pada blok Built in → Control → If test.



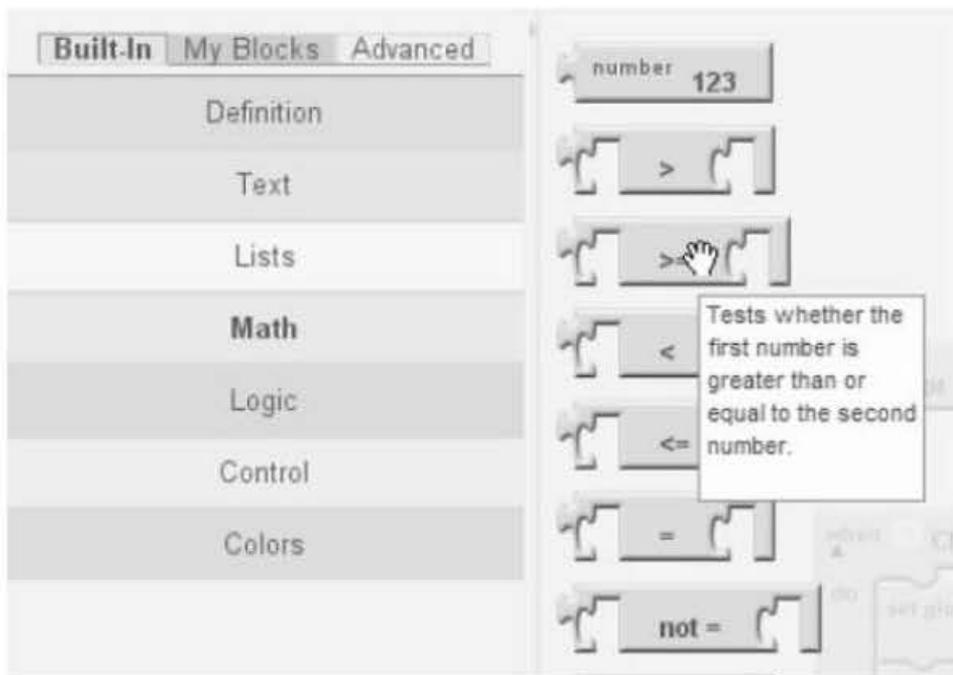
Gambar 4.38. Klik pada blok untuk Built in

- 41. Pasangkan di bawah set global nomor to.



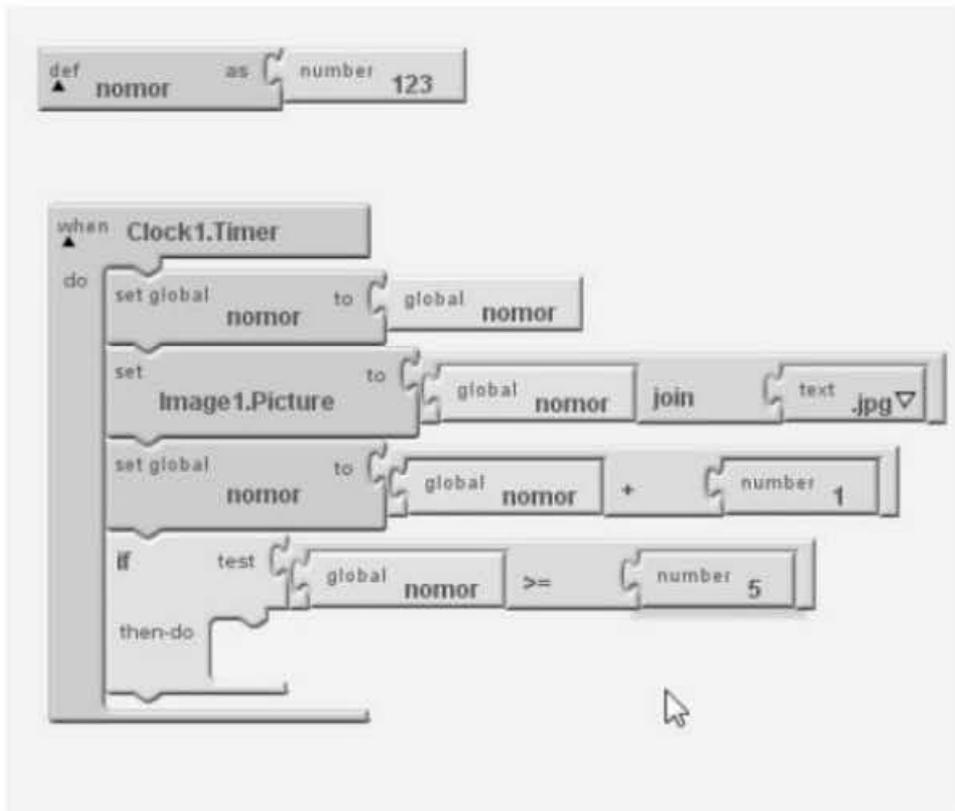
*Gambar 4.39. Blok untuk logika dipasangkan di bawah Set global nomor to*

42. Kemudian klik **Built in** → **Math** dan pilih blok kondisi lebih besar sama dengan.



*Gambar 4.40. Klik pada blok untuk kondisi lebih besar atau sama dengan*

43. Pasangkan blok kondisi yang menyatakan jika global nomor lebih besar sama dengan lima.



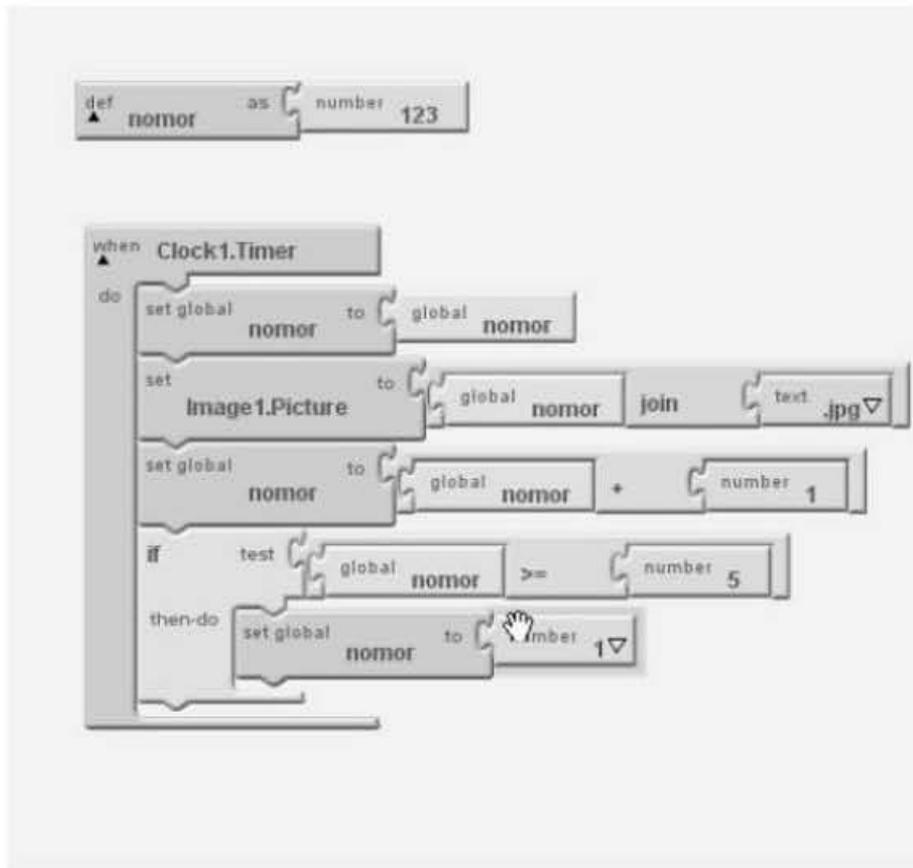
*Gambar 4.41. Mengisi persyaratan jika global nomor lebih dari atau sama dengan 5*

44. Kemudian set global nomor dengan klik pada **My Blocks** → **My Definitions** → **Set global nomor**.



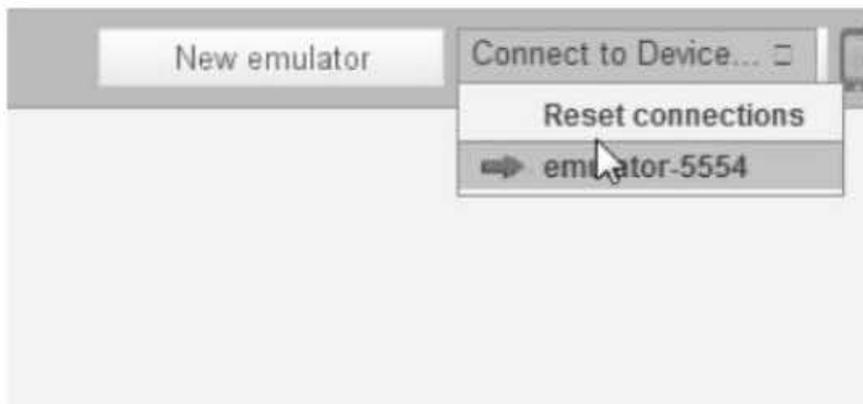
*Gambar 4.42. Set global nomor*

45. Maka operasi yang dilakukan kalau kondisi sudah selesai adalah dengan mengeset nomor untuk kembali ke 1.



*Gambar 4.43. Nomor kembali ke satu*

46. Untuk melihat apakah logika dari program sudah berjalan atau belum, klik pada **Connect to device** → nama emulator.



*Gambar 4.44. Menu untuk menghubungkan ke emulator*

47. Ketika program slideshow berjalan, otomatis gambar akan dijalankan secara otomatis. Karena semuanya diletakkan di event timer.



*Gambar 4.45. Gambar pertama sedang ditampilkan*

48. Ketika event timer berjalan otomatis, file gambar dengan nama berikutnya akan ditampilkan.



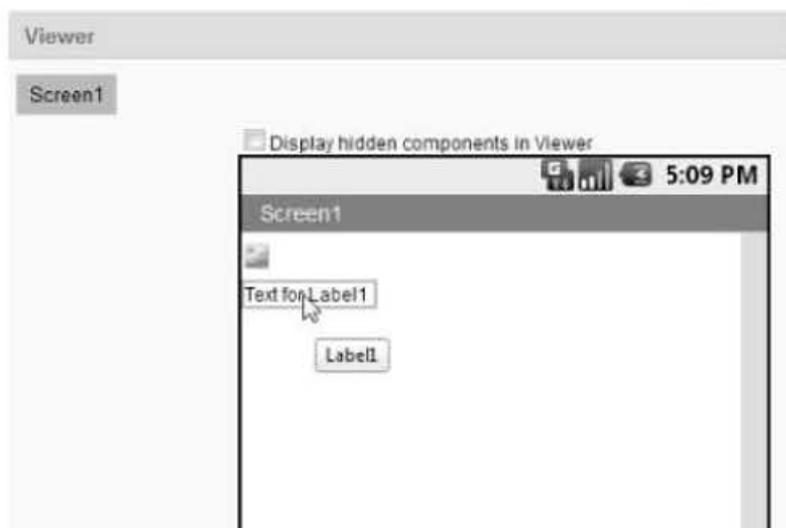
*Gambar 4.46. File gambar dengan nama berikutnya ditampilkan*

49. Hingga sampai file terakhir yaitu 4.jpg maka akan kembali ke gambar pertama.



*Gambar 4.47. File gambar berikutnya sudah ditampilkan karena inkremen*

50. Sekarang kembali ke tampilan Designer. Anda akan menambahkan komponen **Label** untuk menampilkan nama file gambar.



*Gambar 4.48. Memasukkan label ke designer*

51. Hilangkan properti **Text** dari **Label** tersebut, dan beralih ke Block Editor. Klik pada **Label1** dan klik pada **Set Label1.Text**.



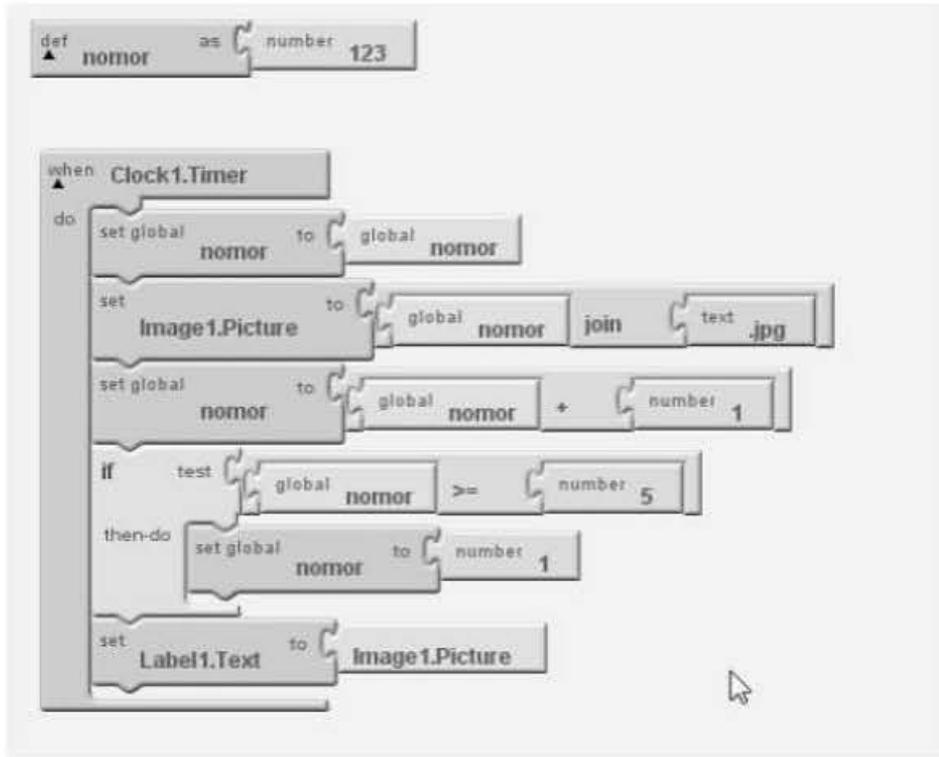
Gambar 4.49. Klik pada blok Set Label1.Text

52. Kemudian masukkan juga My Blocks → Image1 → Image1.Picture.



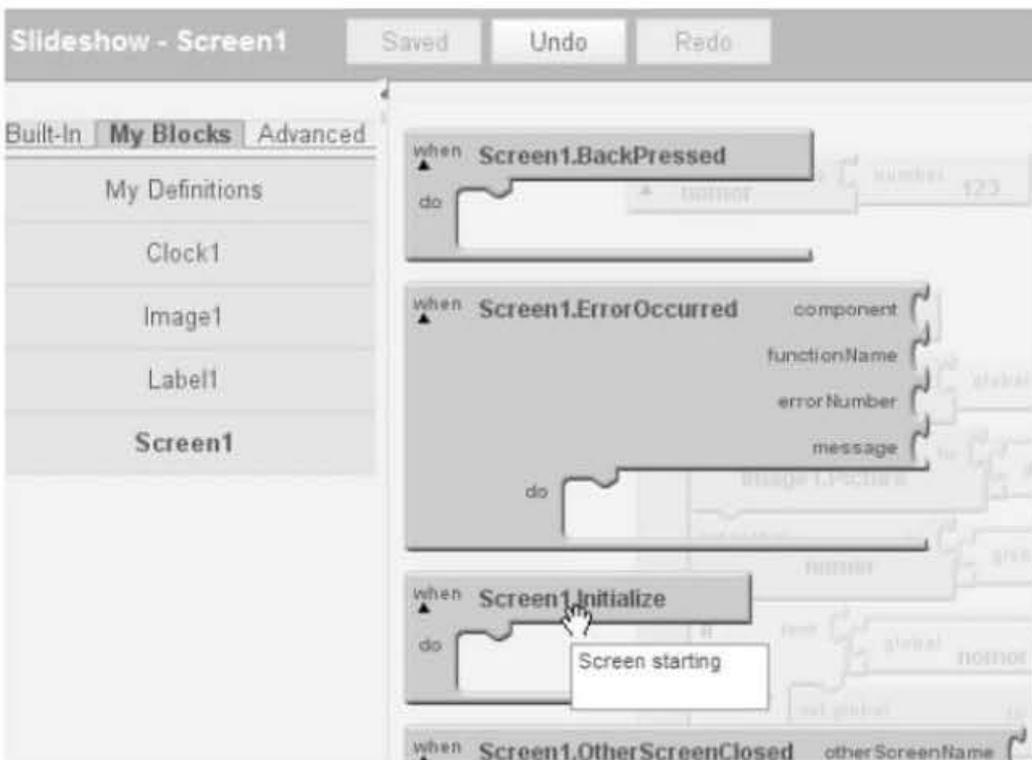
Gambar 4.50. Klik pada blok Image1.Picture

53. Kemudian letakkan Set Label1.Text ke Image1.Picture di bawah tes If then sebelumnya.



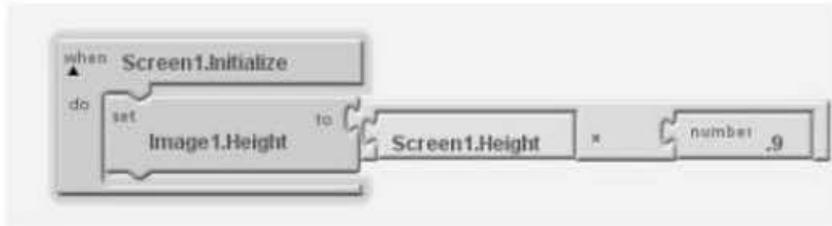
Gambar 4.51. Memasukkan Set Label1.Text ke Image1.Picture

54. Agar label tersebut tidak tertutup oleh gambar, atur ukurannya dengan klik pada **My Blocks** → **Screen1** → **Screen Initialize**.



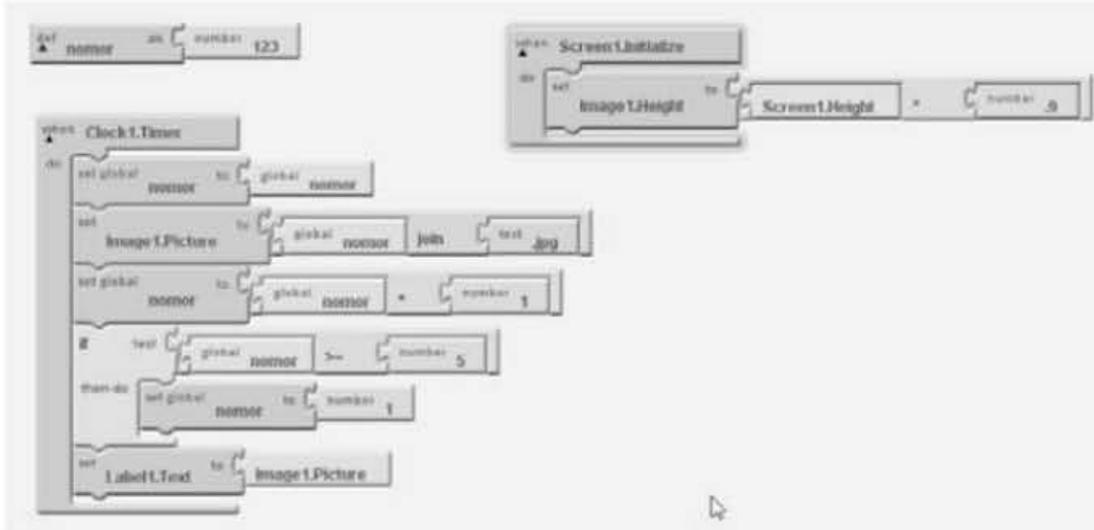
Gambar 4.52. Set padaScreen Initialize

55. Kemudian set **Image1.Height** ke **90%** dari **Screen1.Height**.



**Gambar 4.53.** Menentukan height dari Image1 ke 90% dari height Screen1

56. Maka keseluruhan blok sampai sejauh ini seperti berikut:



**Gambar 4.54.** Keseluruhan blok yang sudah dibuat

57. Jalankan kembali aplikasi, maka nama file akan terlihat, misalnya file 1.jpg seperti berikut:



**Gambar 4.55.** File pertama 1.jpg sudah terlihat, nama file terlihat di label

58. Gambar terus dijalankan secara otomatis, kalau gambar berubah, maka nama file juga akan terlihat berubah.



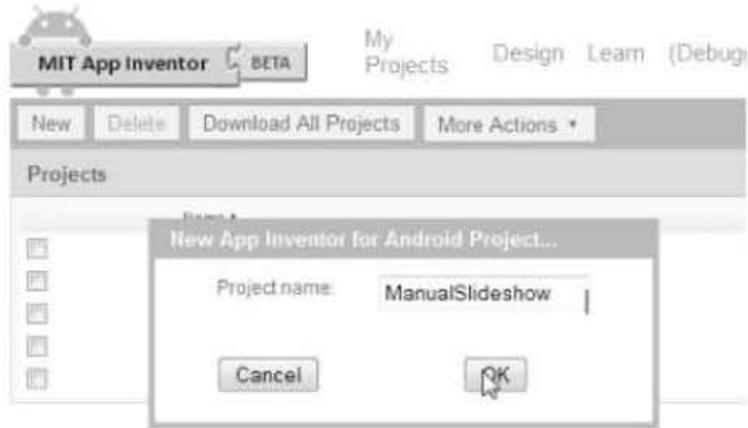
*Gambar 4.56. Nama file berubah karena event timer dari Clock*

## Membuat Aplikasi Slideshow Manual

Kalau di aplikasi sebelumnya Anda membuat aplikasi yang event pemindahan gambarnya dilakukan secara otomatis. Maka sekarang Anda akan membuat event slide show dimana gambar akan dipindahkan ke gambar selanjutnya menggunakan tombol **Next** dan **Previous** atau secara manual.

Untuk membuatnya lakukan langkah-langkah sebagai berikut:

1. Klik pada **My Projects** → **New** untuk membuat proyek baru.
2. Isikan nama project "ManualSlideShow" untuk membuat proyek slideshow manual ini.



*Gambar 4.57. Membuat projek manual*

3. Kemudian masukkan komponen **Image**, serta masukkan **HorizontalArrangement** dengan mengklik **Screen Arrangement** → **HorizontalArrangement**.
4. Masukkan dua **Button** ke dalam **HorizontalArrangement**.



*Gambar 4.58. Memasukkan dua buah button ke Screen Arrangement*

5. Kemudian isikan properti **Text** masing-masing **Button** tersebut dengan `<|` serta `|>`.



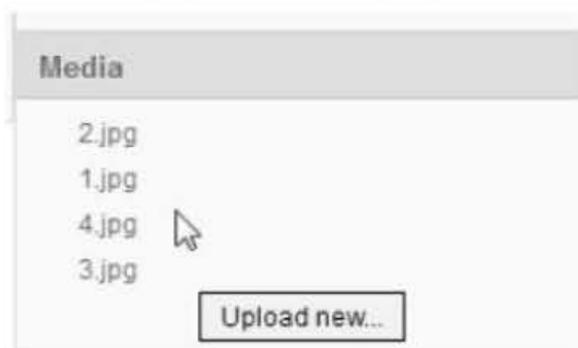
*Gambar 4.59. Mengisi properti Text dari button*

6. Hasilnya, **Button** menjadi seperti berikut ini:



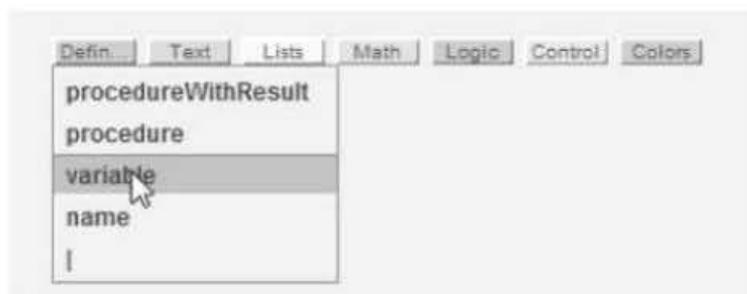
Gambar 4.60. Mengatur properti button untuk tombol previous dan next

7. Sama seperti sebelumnya, di **Media**, upload file 1.jpg hingga 4.jpg.



Gambar 4.61. Upload file 1.jpg hingga 4.jpg

8. Definisikan variabel dengan klik pada sembarang tempat, lalu klik **Define** → **Variable**.



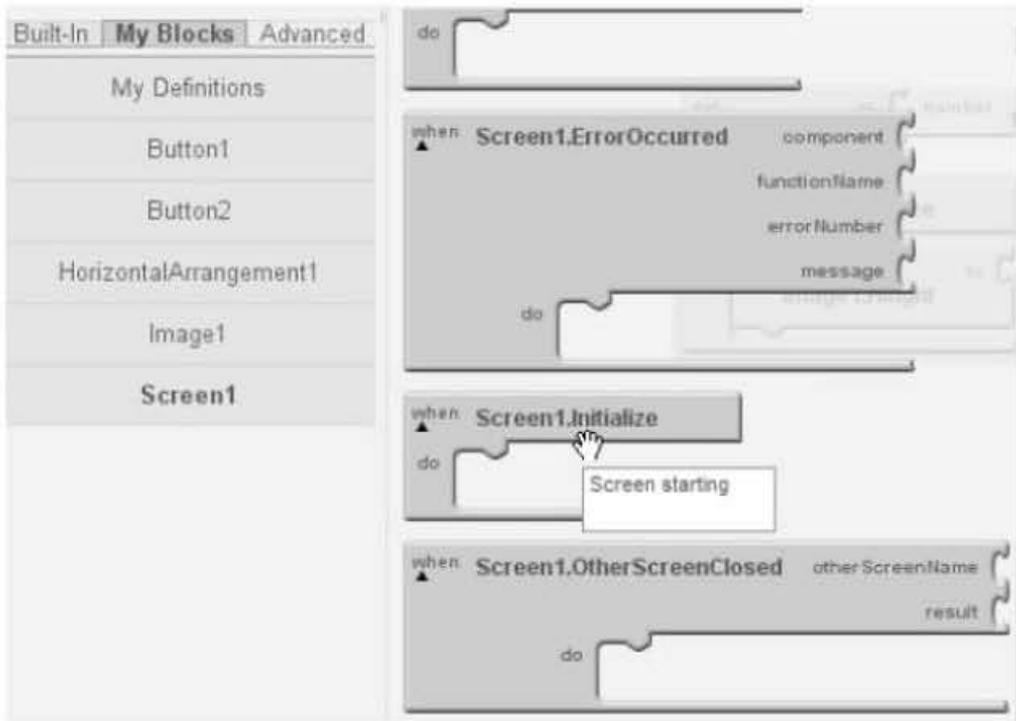
Gambar 4.62. Klik pada Define variable

9. Ganti nama variable dengan "nomor", dan ganti tipe-nya menjadi **number**.



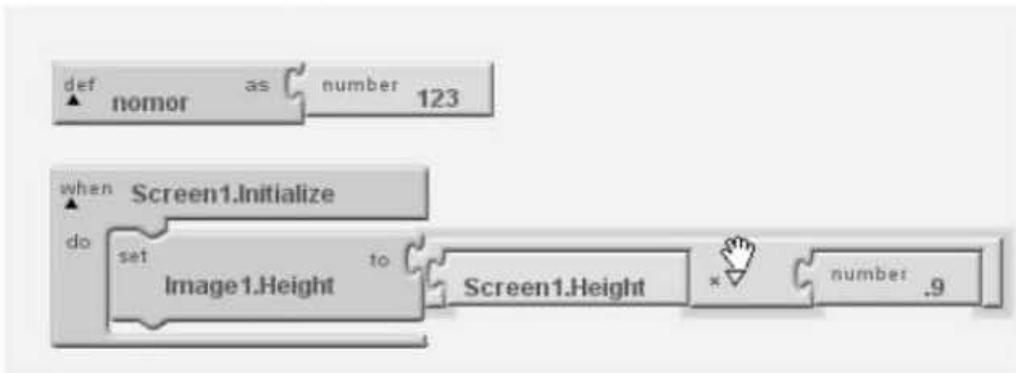
Gambar 4.63. Mengubah jenis variable menjadi number

10. Pada project ini dibuat agar button terlihat, karena itu masukkan dulu screen initialize agar ketika layar dijalankan, ketinggian image bisa diatur. Klik pada **My Blocks** → **Screen1** → **Screen1.Initialize**.



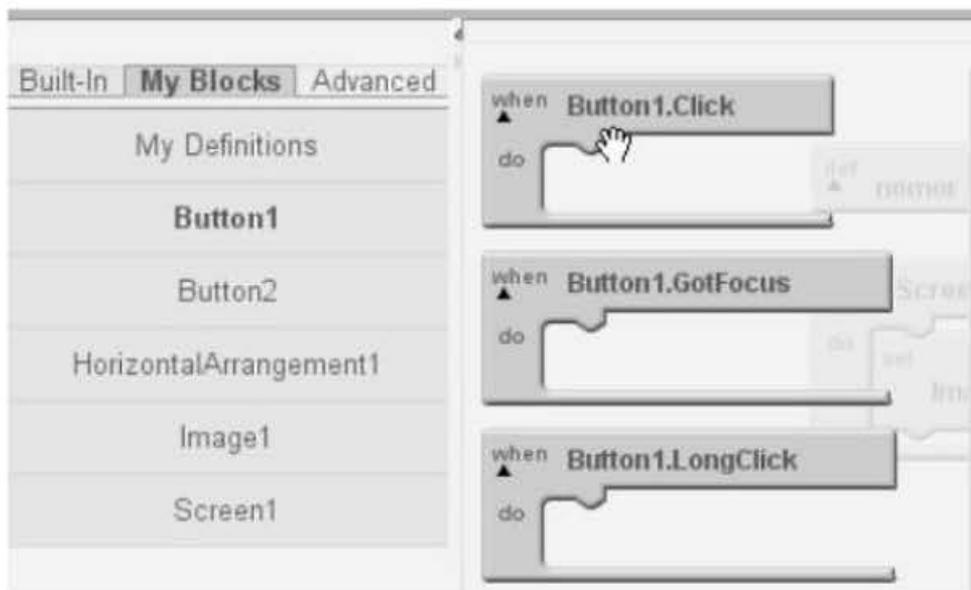
Gambar 4.64. Klik pada blok Screen Initialize

11. Kemudian set **Image1.Height** menjadi **90%** dari layar.



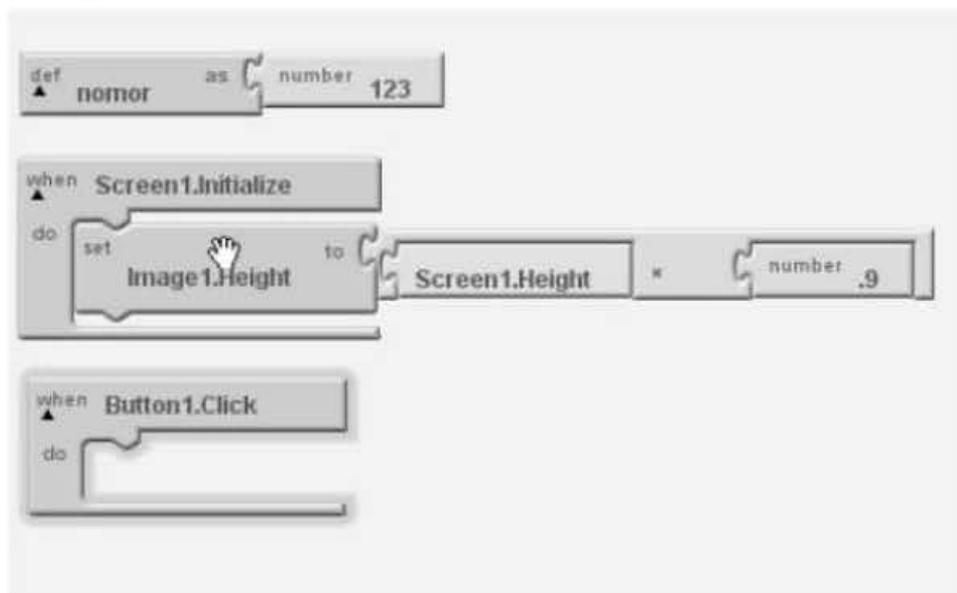
Gambar 4.65. Pengesetan Image1.Height menjadi 90% dari layar

12. Kemudian masukkan event handler untuk **Button1.Click** yang akan menangani event untuk tombol *Previous*.



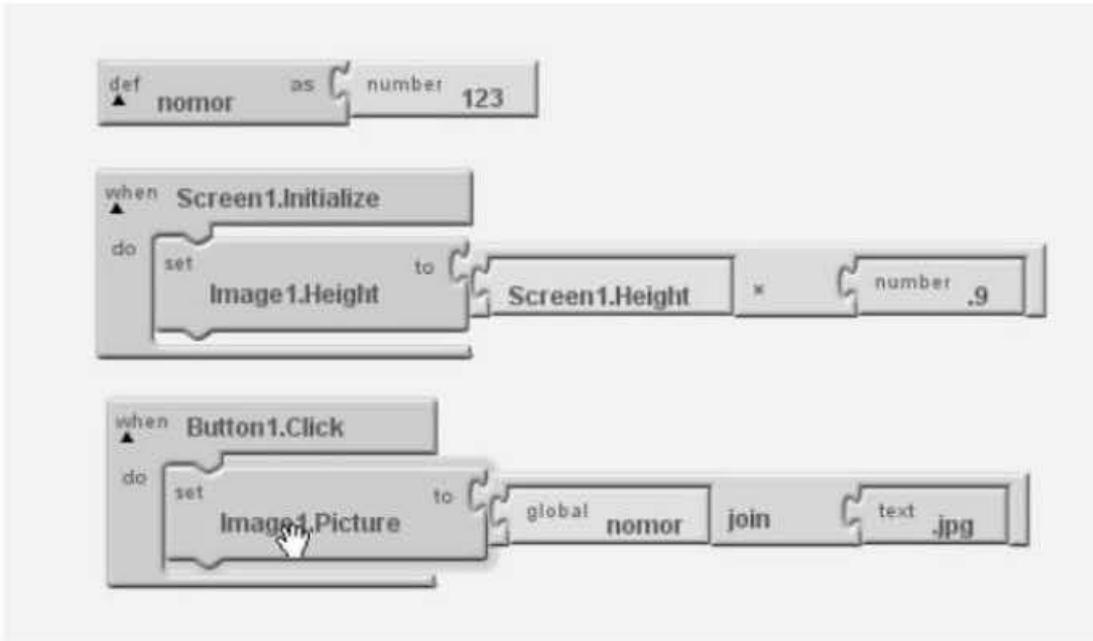
*Gambar 4.66. Klik pada Event handler Button1.Click*

13. Dan tempatkan di bawah `screen1.initialize`.



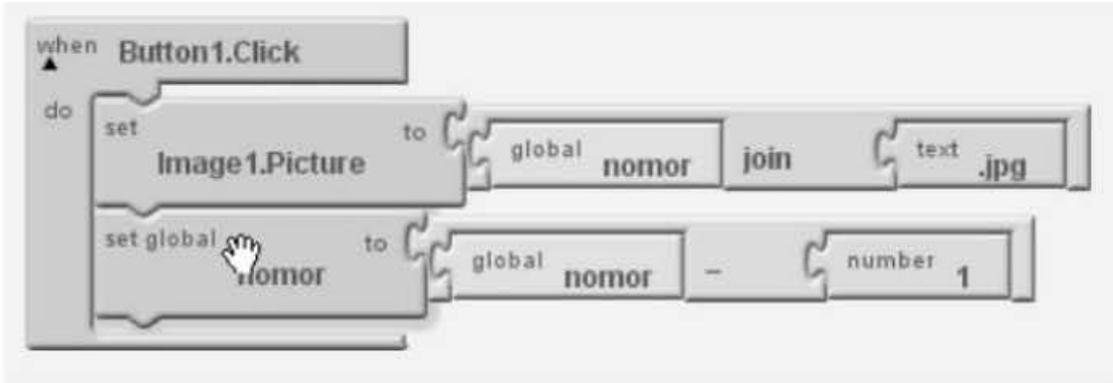
*Gambar 4.67. Memasukkan button1.click event handler yang masih kosong*

14. Kemudian tentukan `image1.picture` join antara nomor dengan teks ".jpg".



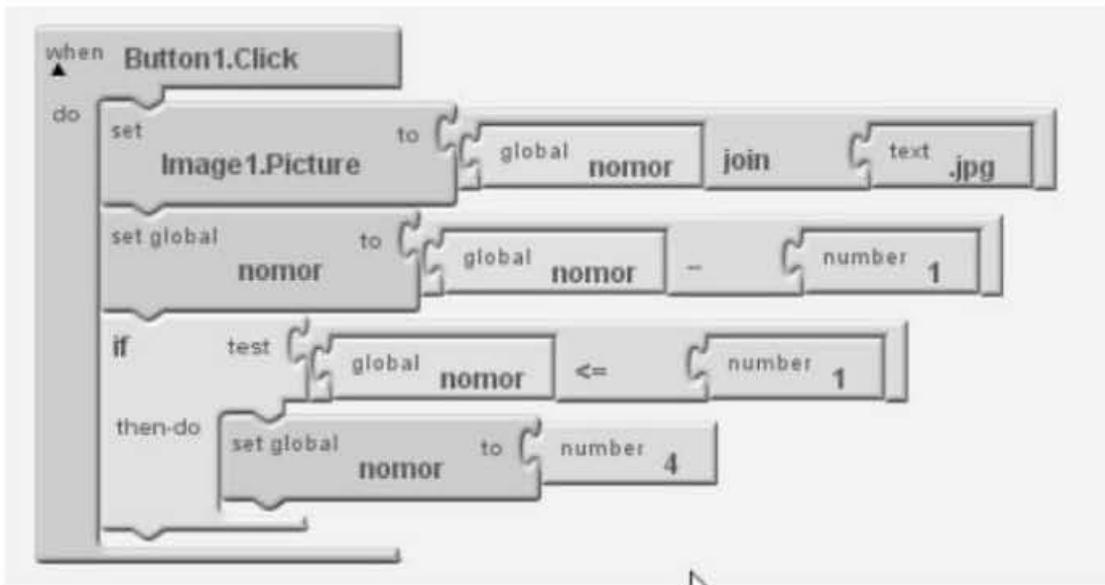
*Gambar 4.68. Image1.Picture ke join antara nomor dengan teks .jpg*

15. Karena ini merupakan tombol previous, maka set nomor berkurang dengan mengurangi variabel nomor dengan angka 1.



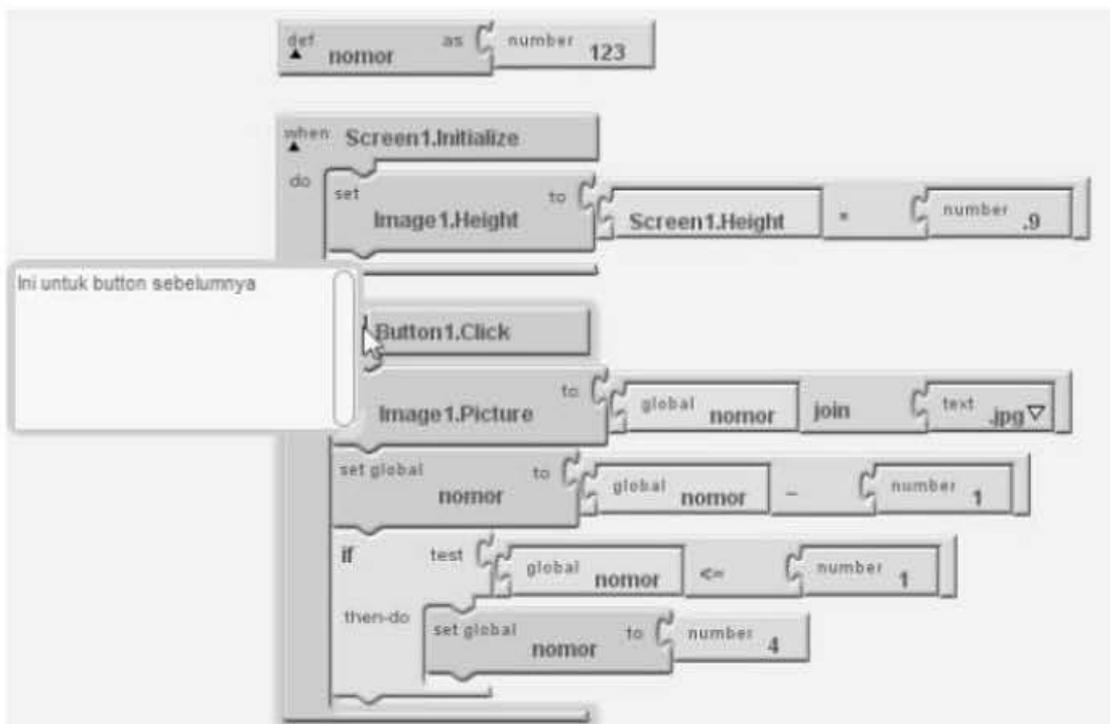
*Gambar 4.69. Mengurangkan variabel nomor dengan angka 1*

16. Setelah itu berikan tes logika If then dimana kalau nomor kurang dari 1, maka nomor dijadikan menjadi 4.



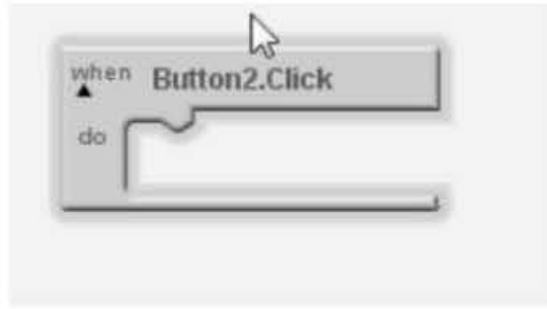
**Gambar 4.70.** Kalau nomor kurang dari 1, maka nomor dijadikan menjadi 4

17. Untuk memudahkan pembacaan kode, beri komentar di blok `button1.click` yang menjelaskan bahwa ini untuk button *previous*.



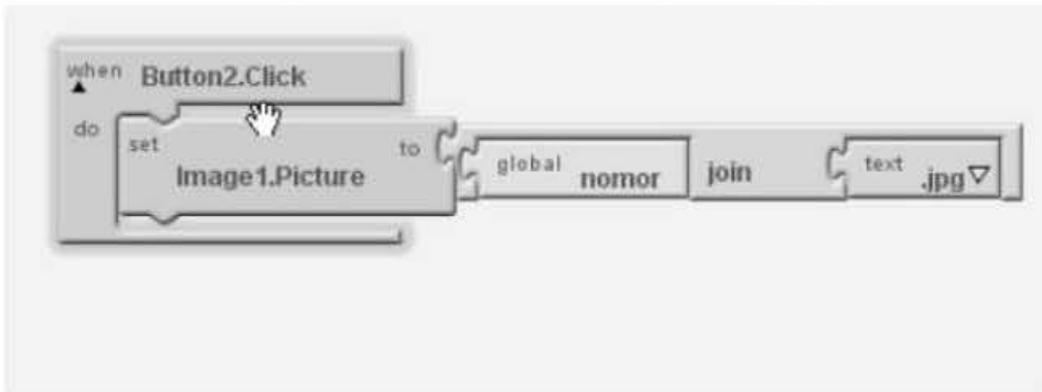
**Gambar 4.71.** Block untuk button *previous*

18. Sampai disini Anda sudah selesai membuat tombol *Previous*. Berikutnya pilih `Button2.Click` untuk membuat tombol *Next*..



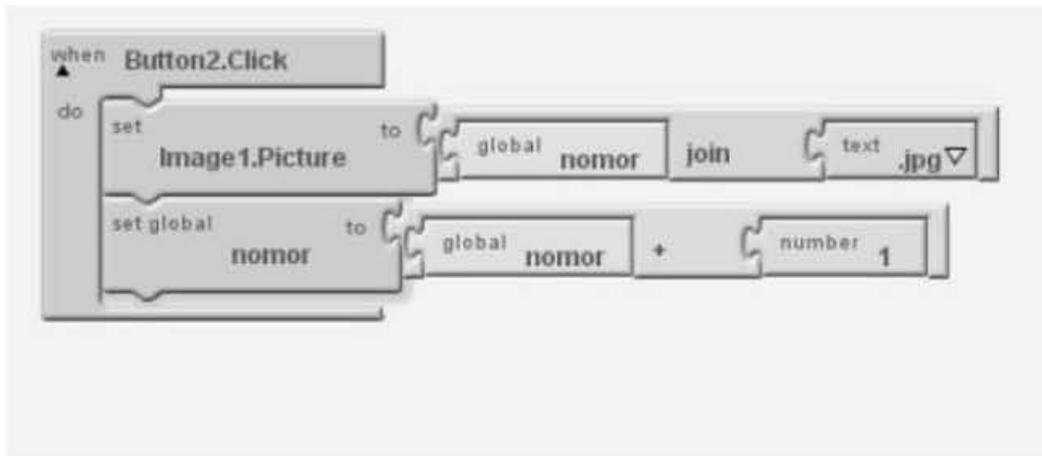
Gambar 4.72. Blok untuk event handler Button2.Click

19. Pertama, set `image1.pictures` ke nomor join ".jpg".



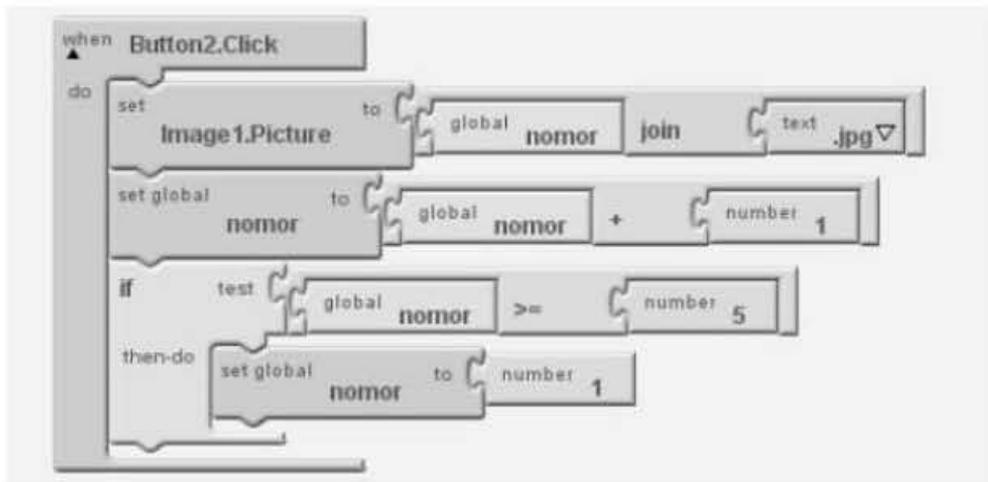
Gambar 4.73. Set Image1.Pictures

20. Kemudian set nomor diinkremenkan dengan penambahan 1.



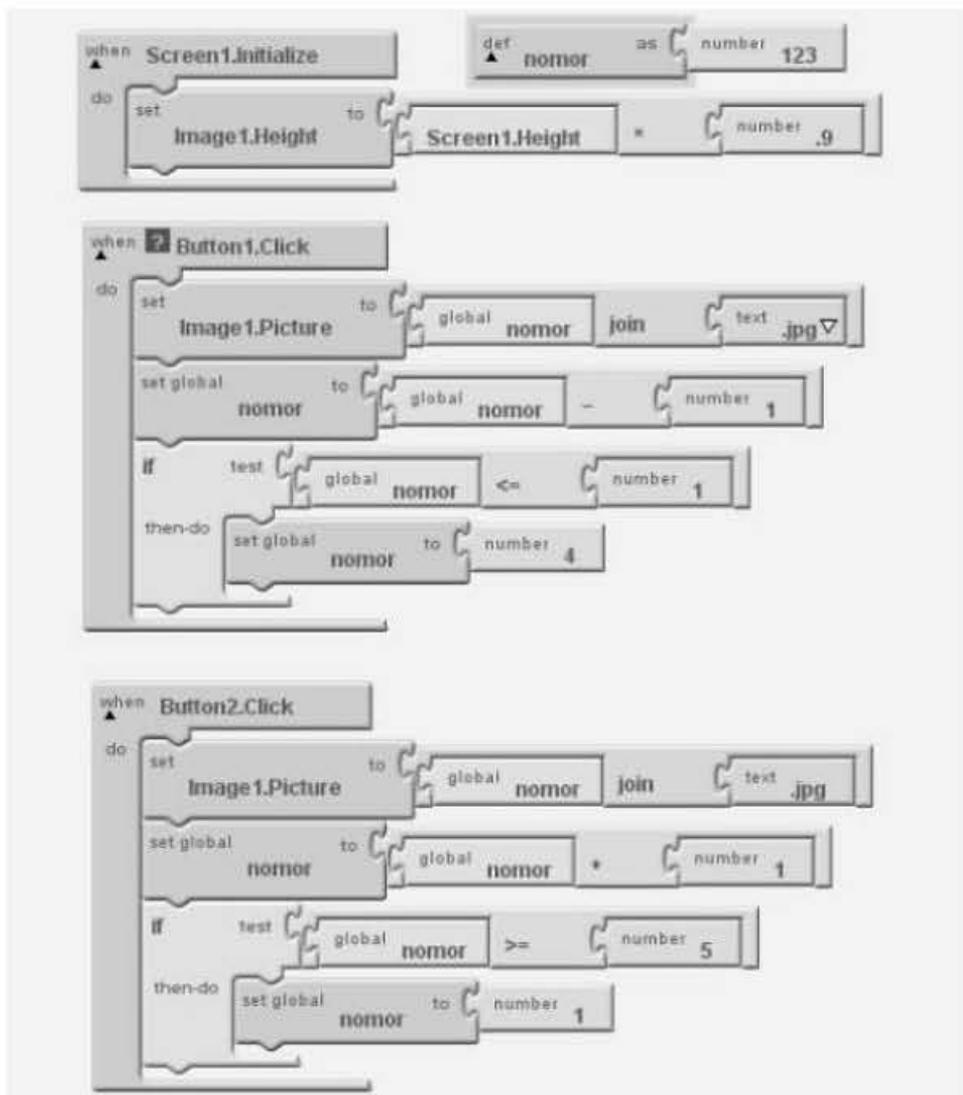
Gambar 4.74. Menginkremenkan nomor ditambahkan dengan satu

21. Terakhir masukkan tes logika If then dimana jika nomor lebih dari lima, maka nomor menjadi kembali menjadi 1.



*Gambar 4.75. Nomor menjadi satu*

22. Hasil akhir, semua susunan block menjadi seperti berikut:



*Gambar 4.76. Kode untuk membuat slideshow manual*

23. Jalankan aplikasi, lalu klik pada **Next**.



*Gambar 4.77. Klik Next untuk menuju gambar berikutnya*

24. Maka gambar yang dibuka adalah gambar berikutnya.



*Gambar 4.78. File gambar yang terbuka adalah gambar berikutnya*

25. Kalau sudah sampai gambar terakhir, lalu diklik **Next** maka akan kembali ke gambar pertama.

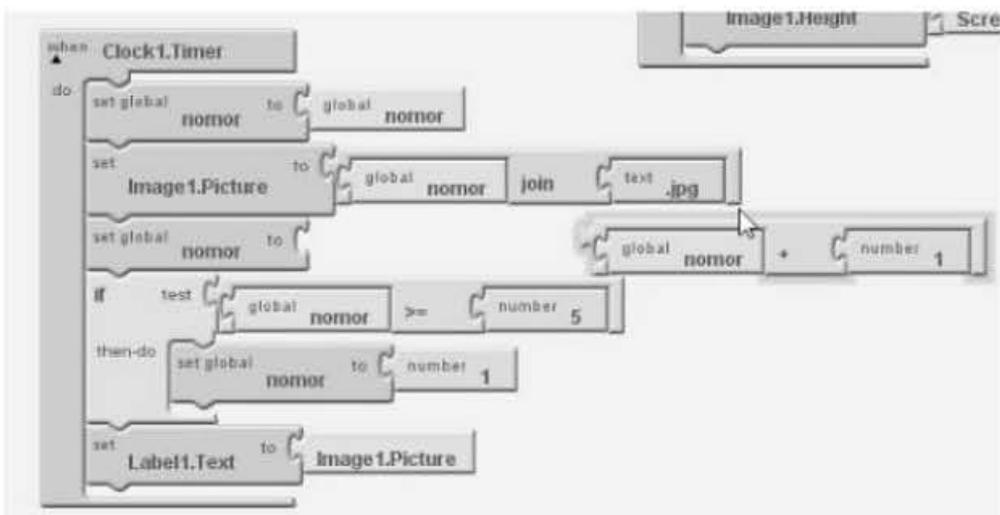


*Gambar 4.79. Kalau sudah digambar terakhir akan kembali ke gambar pertama*

## Membuat Aplikasi Slideshow Random

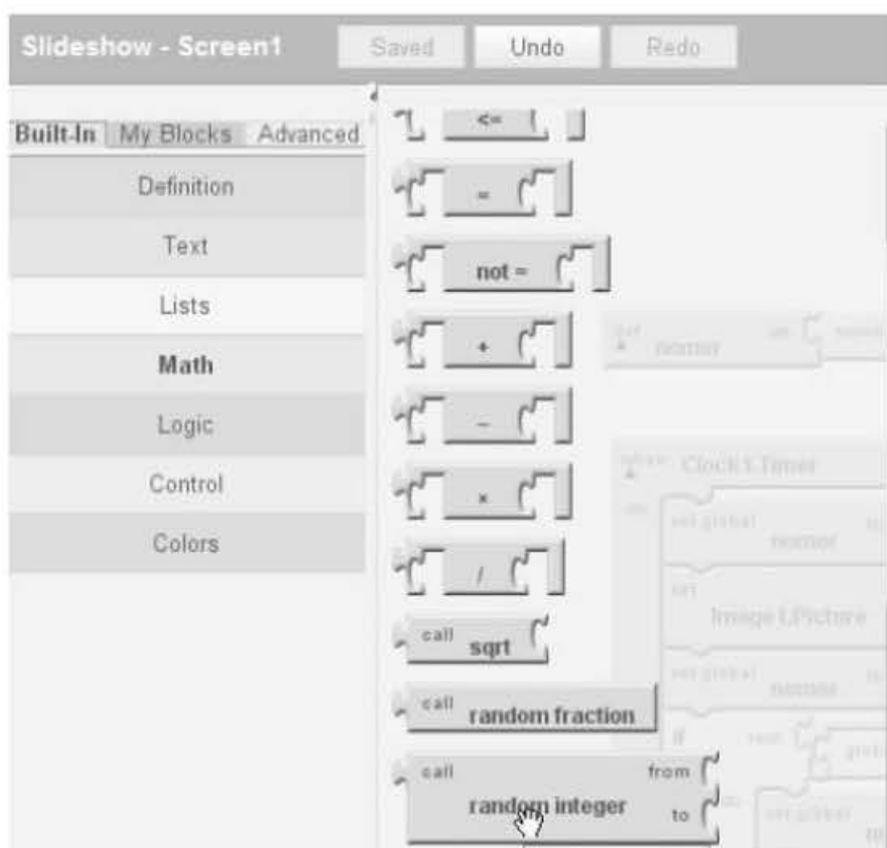
Aplikasi slideshow ketiga yang dibuat adalah membuat aplikasi yang menampilkan gambar secara acak/random, jadi bukan berdasarkan penambahan nilai, atau pengurangan nilai. Berikut langkah pembuatannya:

1. Di sini Anda hanya akan mengedit aplikasi di sub bab pertama yaitu pembahasan tentang "Membuat Aplikasi Slide Show Otomatis".
2. Lepas blok inkremen **set global nomor**.



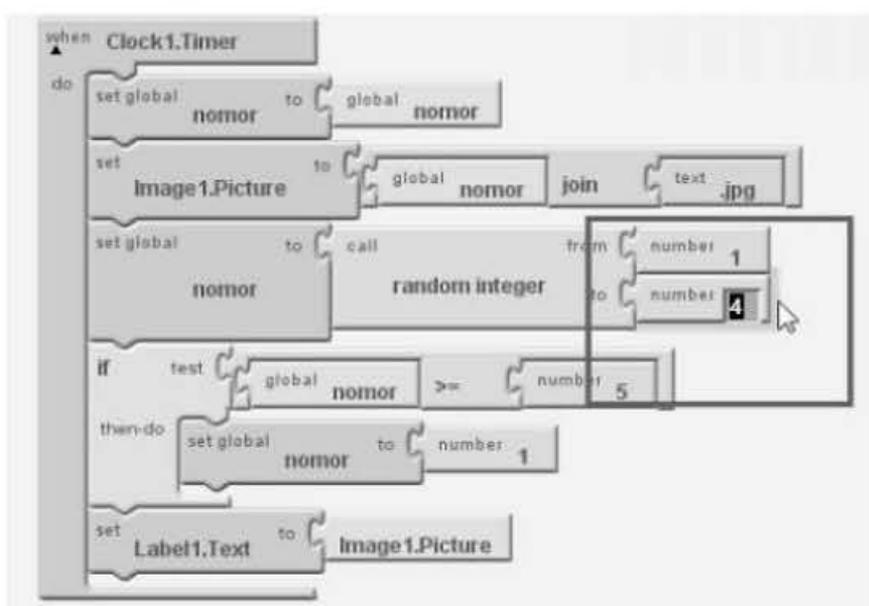
*Gambar 4.80. Melepas blok set inkremen global nomor*

3. Kemudian ganti dengan **Random Integer** dengan mengklik **Built in** → **Math** → **Random Integer**.



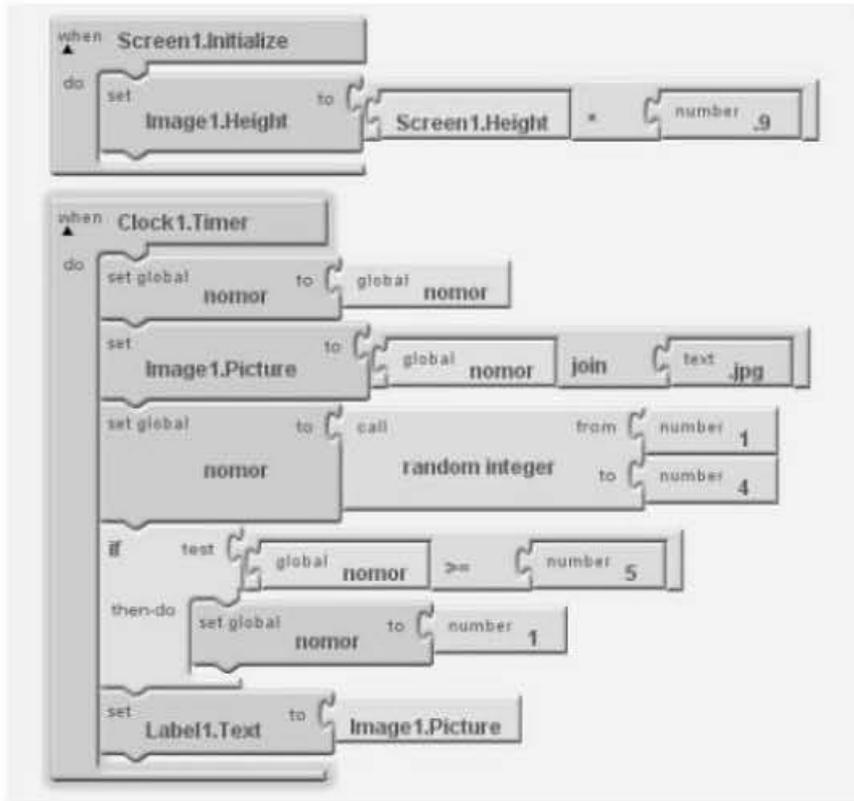
Gambar 4.81. Built in → random integer

- Set random integer dari 1 sampai 4.



Gambar 4.82. Pengesetan random integer

- Maka block program menjadi seperti berikut:



Gambar 4.83. Keseluruhan block program

6. Ketika aplikasi dijalankan, maka akan menampilkan gambar secara acak.



Gambar 4.84. Tampilan pertama

7. Gambar tidak akan berganti secara urut, melainkan secara acak.



*Gambar 4.85. Menampilkan gambar acak berikutnya*

# 5

## MEMBUAT APLIKASI WEB BROWSER

---

*Bab ini membahas:*

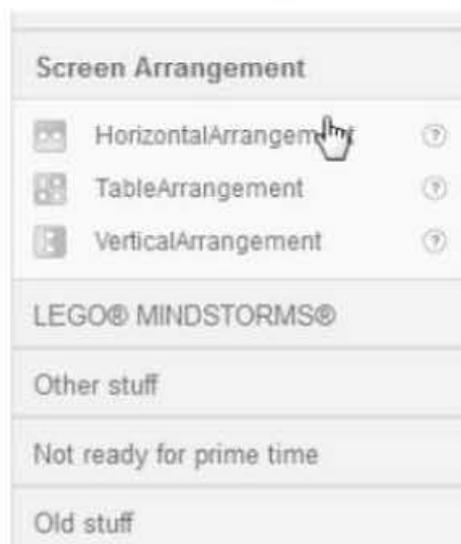
- ❖ Membuat Desain.
  - ❖ Menyusun Block Program.
- 

Pada bab terakhir ini Anda akan diberikan sebuah contoh project aplikasi web browser. Web browser yang nantinya Anda buat bisa Anda gunakan untuk mengakses website tertentu. Aplikasi ini menggunakan komponen web dari platform Android.

### Membuat Project dan Desain Web Browser

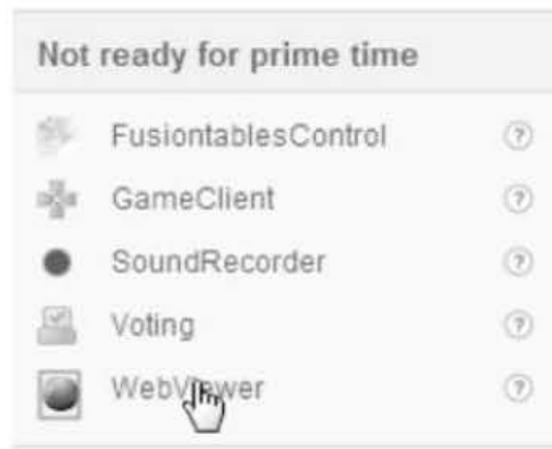
Sama seperti pembahasan sebelumnya, siapkan sebuah project baru dan buat desain aplikasi menggunakan langkah-langkah berikut:

1. Klik **My Project** → **New**.
2. Tambahkan **Horizontal Arrangement** dengan mengklik **Screen Arrangement** → **Horizontal Arrangement**.



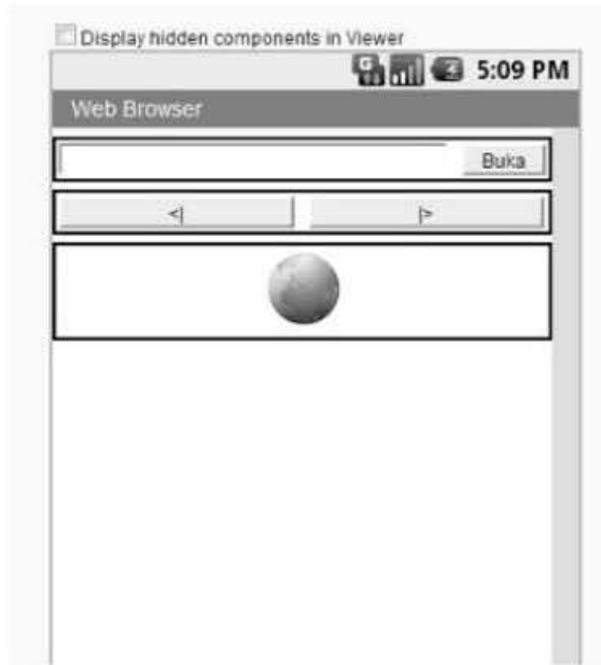
*Gambar 5.1. Horizontal Arrangement*

3. Atur **Width** dari **Horizontal Arrangement1** menjadi **Fill Parrent**.
4. Di dalam **Horizontal Arrangement1** tambahkan sebuah **Textbox** dan sebuah **Button**.
5. Ubah nama **Textbox** di dalam **Horizontal Arrangement1** menjadi "URLTextbox" dan atur **Width** menjadi **Fill Parrent**.
6. Ubah nama **Button** di dalam **Horizontal Arrangement1** menjadi "btnBuka", dan **Text** menjadi "Buka", serta atur **Width** menjadi **Fill Parrent**.
7. Tambahkan lagi **Horizontal Arrangement** dan tempatkan dibawah **Horizontal Arrangement1**. Jadi sekarang Anda memiliki 2 buah **Horizontal Arrangement** yaitu **Horizontal Arrangement1** dan **Horizontal Arrangement2**.
8. Atur pula **Width** dari **Horizontal Arrangement2** menjadi **Fill Parrent**.
9. Masukkan 2 buah **Button** ke dalam menjadi **Horizontal Arrangement2**.
10. Atur **Width** kedua **Button** tersebut menjadi **Fill Parrent**.
11. Ubah nama masing-masing button menjadi "btnSebelumnya" dan "btnSetelahnya" dan **Text** menjadi <| dan |>.
12. Masukkan komponen **WebView** dengan mengklik **Not ready for prime time** → **WebView**.
13. Dan terakhir tambahkan komponen **Notifier**.



*Gambar 5.2. Menambahkan WevViewer*

14. Desain yang dibuat terlihat seperti pada gambar berikut ini:



*Gambar 5.3. Tampilan komponen*

15. Pada panel **Components**, terlihat komponen-komponen yang sudah Anda masukkan.



*Gambar 5.4. Komponen-komponen yang sudah dimasukkan*

## Menyusun Block Program

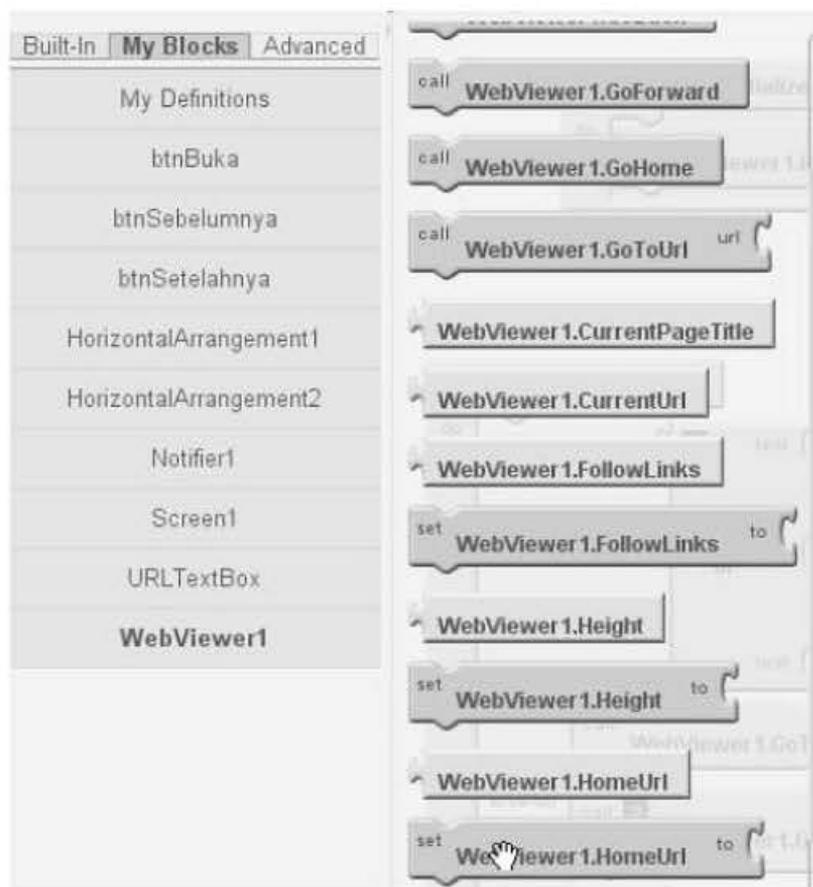
Setelah desain Anda buat, saatnya Anda menambahkan blok untuk program web ini. Lakukan langkah-langkah berikut ini:

1. Pertama klik pada **My Blocks** → **Screen1** → **Screen1.Initialize**.



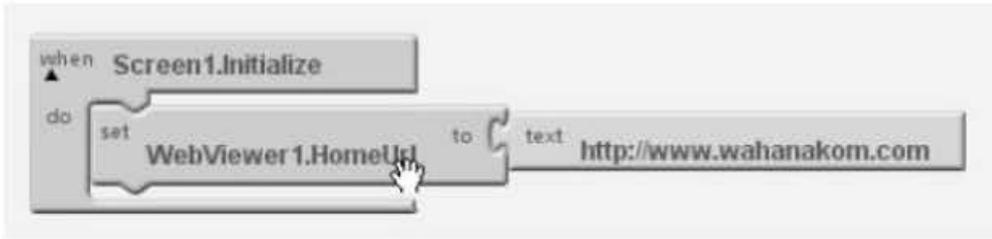
*Gambar 5.5. Klik pada blok Screen Initialize untuk event handler*

2. Pada aplikasi ini akan dibuat sebuah URL default, karena itu klik block **WebView1.HomeURL**.



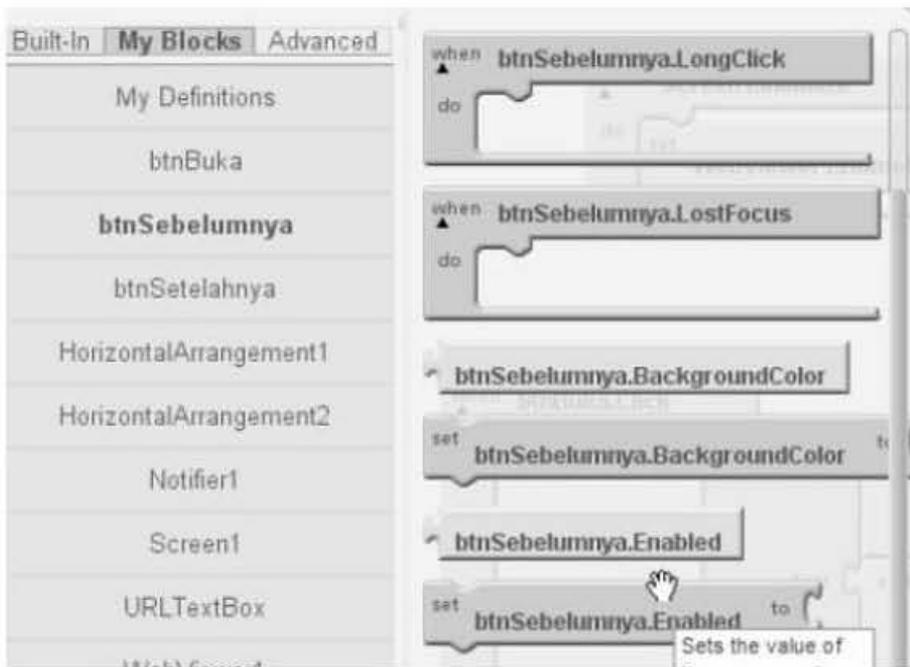
*Gambar 5.6. Blok WebView1.HomeURL*

3. Pasangkan blok **WebView1.HomeURL** dan set ke nilai teks dengan nama *http://www.wahanakom.com*.



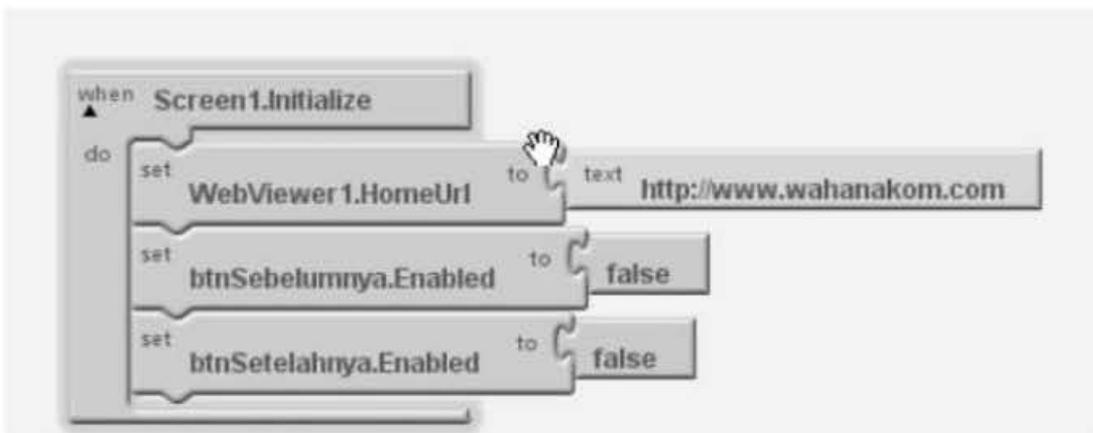
Gambar 5.7. Mengisikan variabel teks untuk WebViewer1.HomeURL

4. Ketika dijalankan pertama kali, akan dibuat agar tombol *Sebelum* dan *Sesudah* dinonaktifkan. Maka, pasang block `set btnSebelumnya.Enabled` to ke `Screen1.Initialize` dan pasang block `False`.



Gambar 5.8. Klik pada btnSebelumnya.Enabled

5. Begitu juga dengan `btnSesudahnya.Enabled` to pasang ke `Screen1.Initialize` dan atur menjadi `False`.



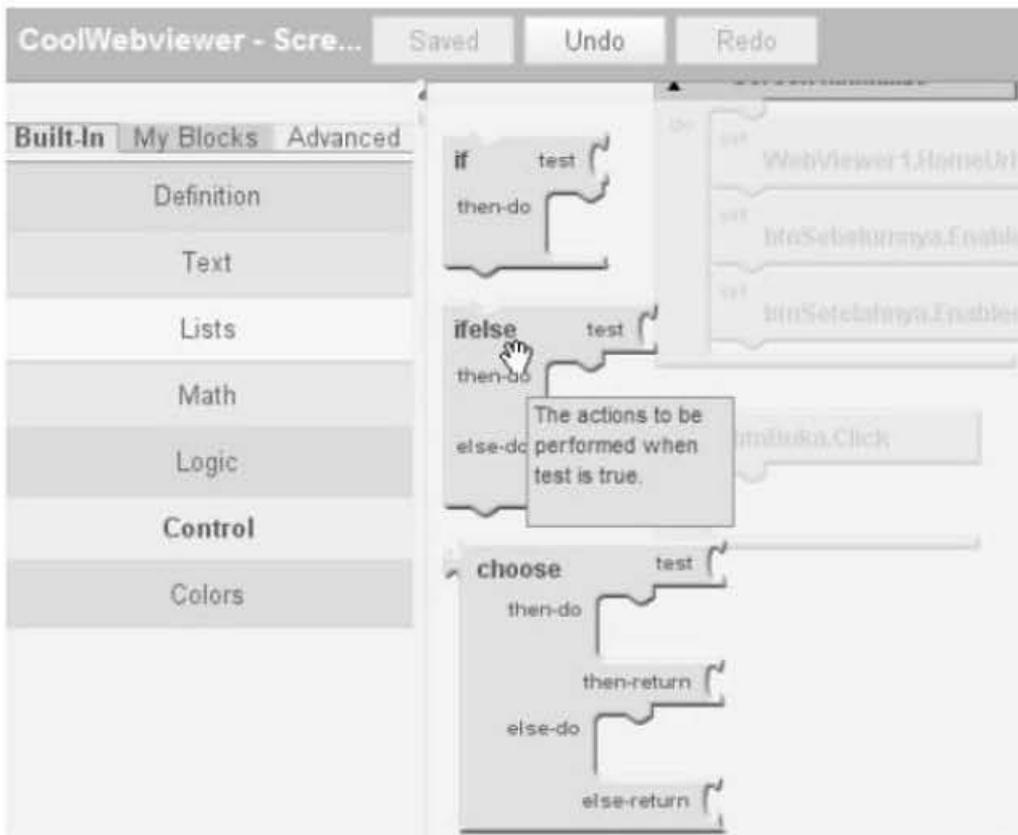
Gambar 5.9. Block untuk menonaktifkan button

6. Akan dibuat pula jika btnBuka diklik, maka website akan dibuka, karena itu klik pada blok **My Blocks** → **btnBuka** dan pilih event handler **btnBuka.Click**.



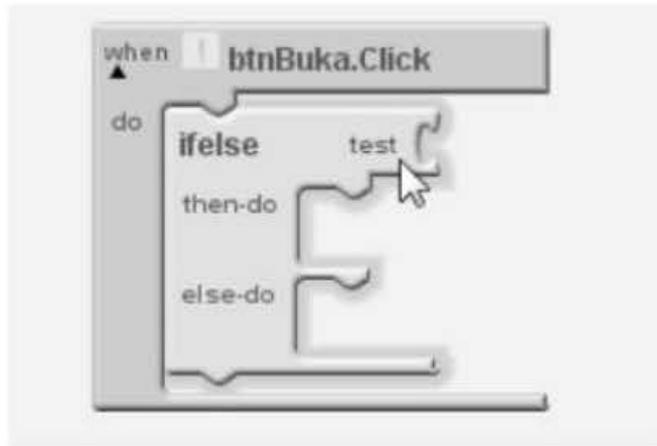
*Gambar 5.10. Pemilihan Event handler btnBuka.click*

7. Akan diberikan pula kondisi apabila belum ada http:// akan ditambahkan http:// karena itu klik **Built in** → **Control** → **If Else**.



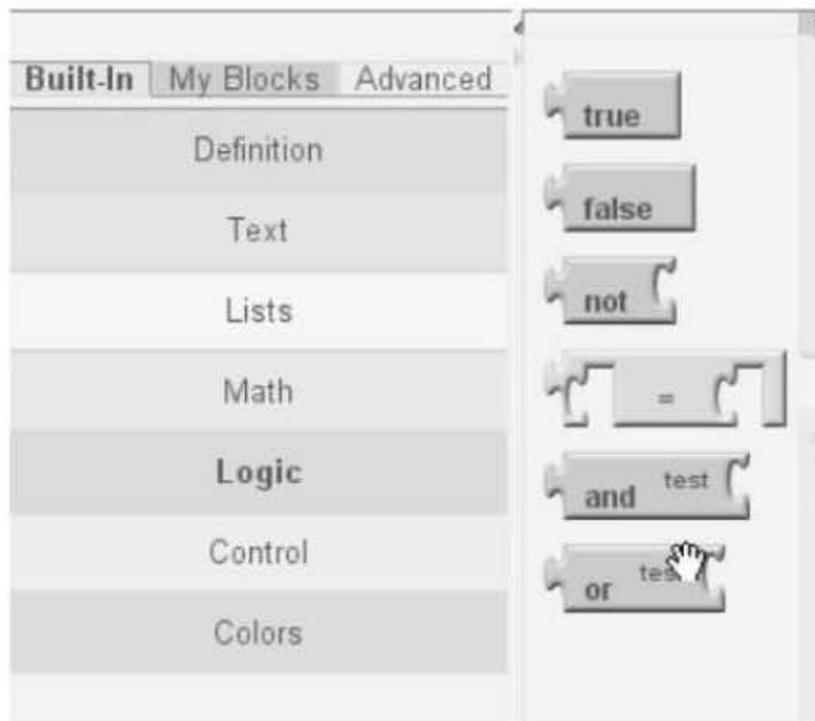
*Gambar 5.11. Menu untuk If Else*

8. Masukkan blok **If Else** tersebut ke **btnBuka.click**.



*Gambar 5.12. Klik If Else ke btnBuka.Click*

9. Kemudian gunakan operator **OR**.



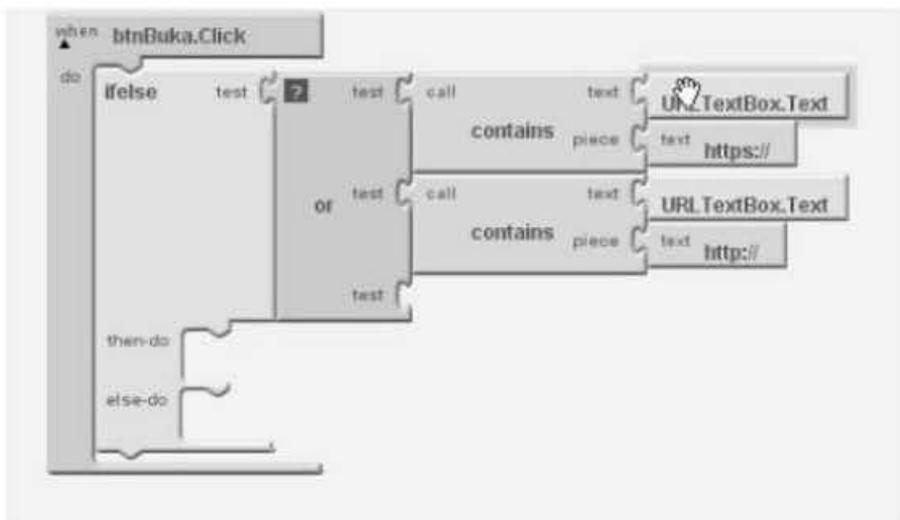
*Gambar 5.13. Memilih blok untuk operator OR*

10. Dibuat pula jika terdapat suatu contain/mengandung teks tertentu, maka akan diambil tindakan tertentu. Klik pada **Built in** → **Text** → **Contains**.



*Gambar 5.14. Built in → Text → Contains*

11. Dibuat pula jika terdapat teks mengandung https atau http, maka akan diambil tindakan tertentu.



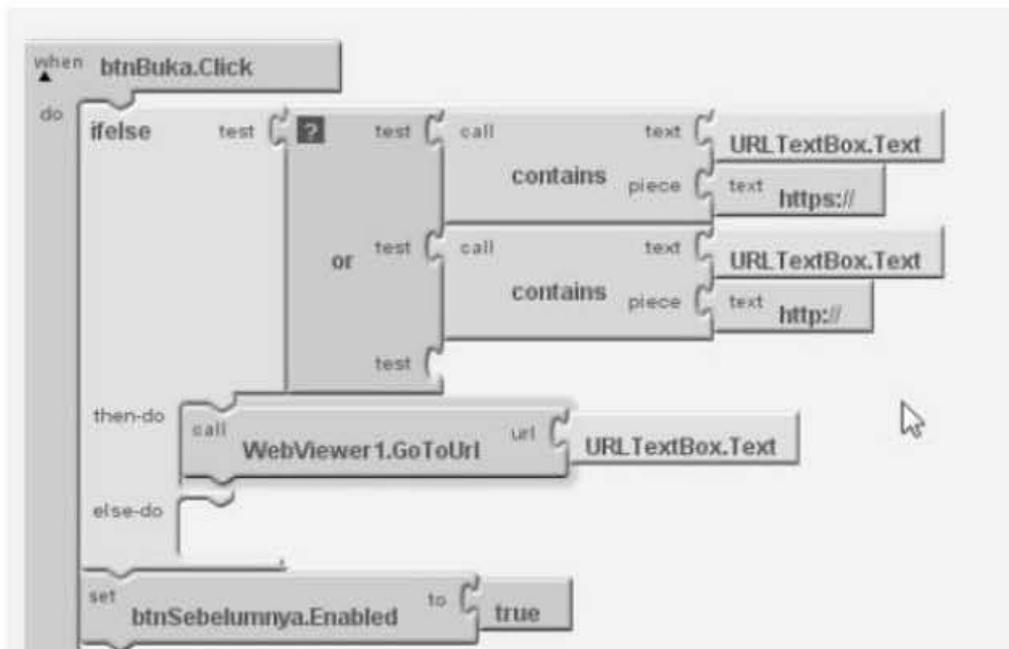
*Gambar 5.15. Pengujian apabila teks mengandung karakter http atau https*

12. Agar web browser ini bisa membuka URL yang Anda tentuka, klik pada **My Blocks → WebViewer1 → WebViewer1.GoToURL**.



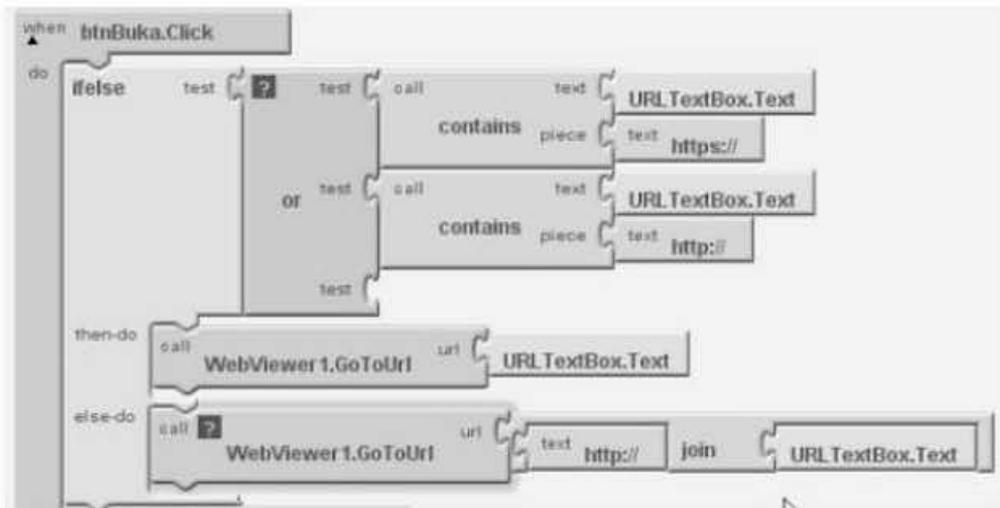
Gambar 5.16. *WebView1.GoToURL*

13. Set agar WebView langsung membuka url di `URLTextbox.Text`.



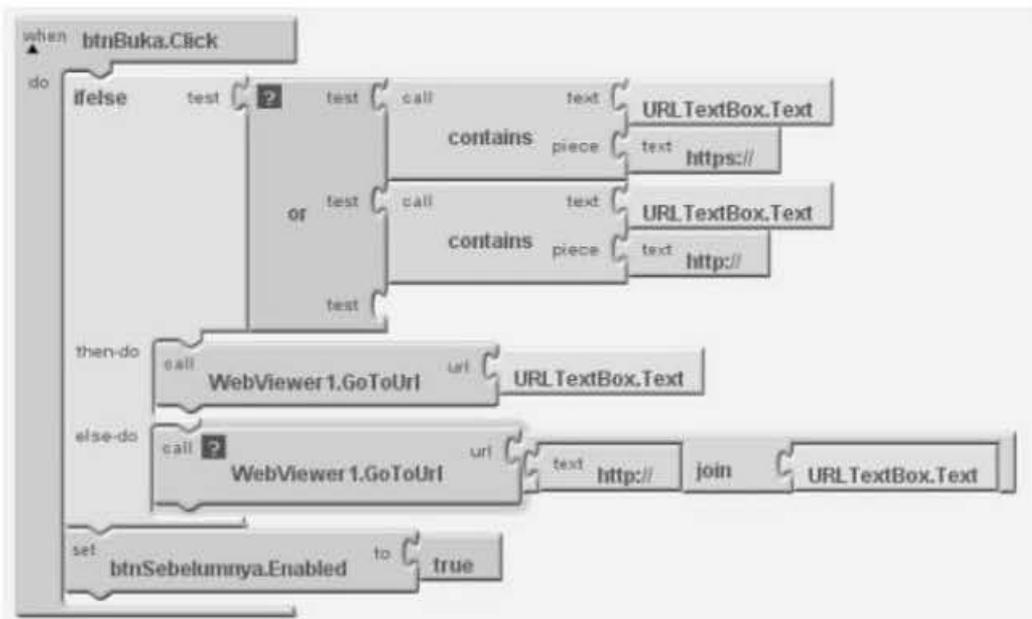
Gambar 5.17. *Set untuk langsung membuka di bagian Then do*

14. Sementara kalau user tidak memberikan karakter "http" atau "https" maka akan ditambahkan http di bagian depannya menggunakan perintah **Join**.



*Gambar 5.18. Menambahkan jika false untuk men-join dengan http*

- Setelah itu set **btnSebelumnya** menjadi **True**. Ini agar user bisa kembali ke halaman sebelumnya/back.



*Gambar 5.19. Block untuk kembali ke halaman sebelumnya*

- Kalau dijalankan, maka terlihat halaman web akan dibuka dengan HomeURL.



*Gambar 5.20. Halaman web di Home URL*

17. Anda bisa mengisi halaman yang akan dibuka, pada textbox yang ada, kemudian klik **Buka**.



*Gambar 5.21. Klik pada Buka untuk membuka*

18. Jika Anda ingin memberikan action ke event handler **btnSetelahnya** dan **btnSebelumnya**. Pertama set **btnSebelumnya** menjadi **True** dengan contoh seperti berikut ini:



Gambar 5.22. Pengesetan btnSebelumnya menjadi True

19. Kemudian untuk action setelahnya, klik pada **WebView1.GoForward**.



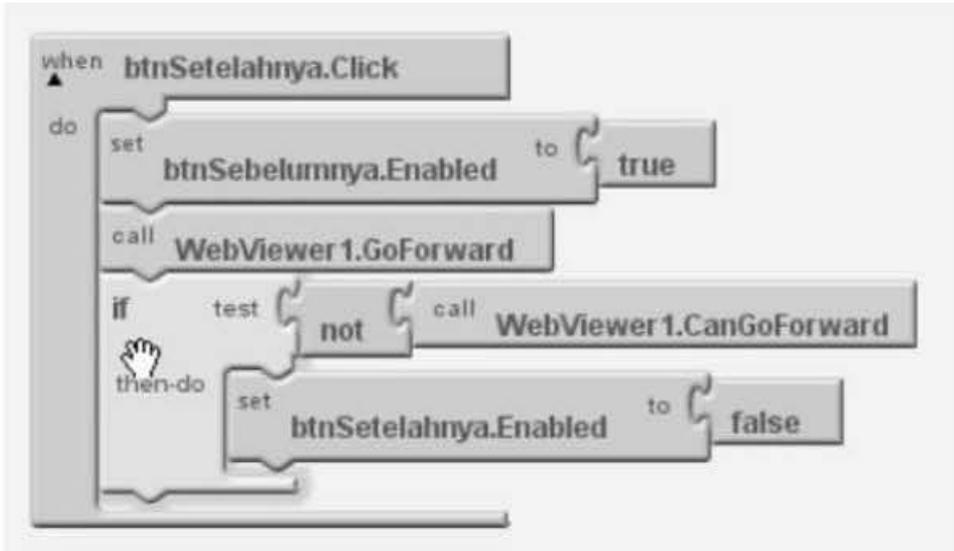
Gambar 5.23. Klik pada WebView1,GoForward

20. Pasangkan dibawah **btnSebelumnya.Enabled = True**.



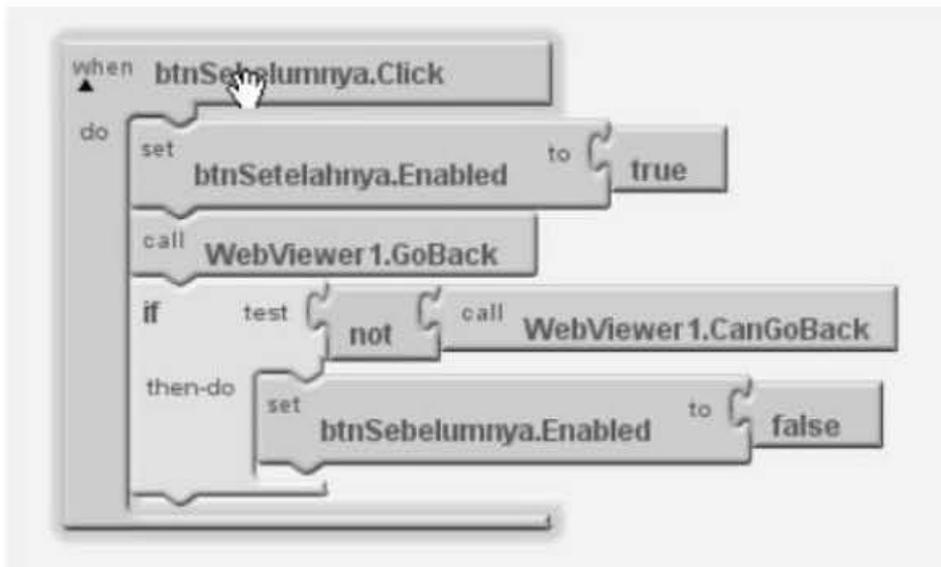
Gambar 5.24. Mengisikan action WebView1.GoForward

21. Setelah itu gunakan blok If then untuk memeriksa apakah **WebView1.CanGoForward**. jika ya, maka set **btnSetelahnya** menjadi tidak aktif (nilai **enabled = false**).



*Gambar 5.25. Pengesetan btnSetelahnya menjadi tidak aktif*

22. Untuk **btnSebelumnya**, kebalikannya, Anda tinggal memanggil action **WebView1.GoBack**.



*Gambar 5.26. Block-block kode untuk btnSetelahnya.Click*

23. Maka kalau Anda menjalankan program, lalu membuka halaman setelahnya, **btnSebelumnya** akan aktif.



*Gambar 5.27. BtnSebelumnya sudah diaktifkan dan siap untuk digunakan*

# PENUTUP

**S**etelah membaca buku ini diharapkan pembaca telah memahami penggunaan App Inventor untuk membuat aplikasi Android. Banyak hal yang sudah disampaikan, mulai dari mengenal App Inventor, menggunakan IDE App Inventor, Membuat Aplikasi Pilih Gambar, Membuat Aplikasi Slideshow, hingga membuat aplikasi Web Browser.

Penulis berharap Anda dapat memahami apa yang telah disampaikan dengan baik dan benar. Bila Anda mengalami kesulitan dalam mempelajari buku ini, Anda dapat menyampaikan pertanyaan kepada penulis melalui alamat yang berada pada bagian Prakata. Selamat Belajar..!!!



# Membuat Aplikasi Android Tanpa Coding dengan APP Inventor



Jika Anda hendak belajar membuat aplikasi untuk platform Android, namun Anda masih awam dengan bahasa pemrograman yang dipakai, maka buku ini jawabannya. Buku ini akan mengajarkan kepada Anda tentang sebuah kemudahan dalam membuat aplikasi Android tanpa coding menggunakan App Inventor. Lalu, apa App Inventor itu?

App Inventor merupakan sebuah tool untuk membuat aplikasi Android menggunakan visual block programming yang sangat mengagumkan. Mengapa disebut mengagumkan? ini karena dalam membuat sebuah aplikasi Android Anda tidak perlu mengetikkan satu pun code program.

Anda hanya perlu melakukan drag drop atau menyusun block yang merupakan simbol perintah dan fungsi mirip dengan permainan puzzle.

Melalui buku ini penulis akan mengajarkan kepada Anda betapa mudahnya membuat aplikasi Android menggunakan App Inventor. Buku yang disajikan secara ringan dan berbobot ini, akan mengupas secara step by step pemrograman Android menggunakan App Inventor serta dilengkapi dengan contoh aplikasi yang dibuat. Lebih lengkap, buku ini membahas:

- Bab 1. Apa itu App Inventor?
- Bab 2. Menggunakan IDE App Inventor
- Bab 3. Membauat Aplikasi Pilih Gambar
- Bab 4. Membuat Aplikasi Slideshow
- Bab 5. Membuat Aplikasi Web Browser

PT ELEX MEDIA KOMPUTINDO  
Kompas Gramedia Building  
Jl. Palmerah Barat 29-37, Jakarta 10270  
Telp. (021) 53650110-53650111, Ext 3214  
Webpage: <http://www.elexmedia.co.id>

<b>Kelompok</b>
<b>Sistem Operasi</b>
<b>Ketrampilan</b>
<input checked="" type="checkbox"/> Tingkat Pemula
<input checked="" type="checkbox"/> Tingkat Menengah
<input type="checkbox"/> Tingkat Mahir
<b>Jenis Buku</b>
<input checked="" type="checkbox"/> Referensi
<input checked="" type="checkbox"/> Tutorial
<input type="checkbox"/> Latihan

