

MODUL PELATIHAN

Pembuatan Aplikasi



Menggunakan



Disusun Oleh:

Riza Arifudin
Ardhi Prabowo

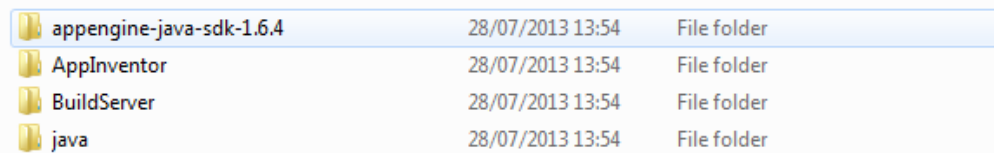


MEMULAI APP INVENTOR OFFLINE (APP INVENTOR PERSONAL SERVER)

App Inventor pada awalnya muncul merupakan versi online, kemudian sekarang sudah ada versi yang Offline. Perbedaan dari keduanya secara singkat padat dan jelas adalah untuk yang versi online apabila ingin mengcompile alias mempaket aplikasi maka tetap harus terkoneksi internet. Sedangkan dengan menggunakan App inventor offline tanpa terkoneksi ke internetpun kita bisa membuat aplikasi android secara mandiri sekaligus bisa mempaket atau mengcompile-nya langsung. Itulah sebabnya disebut juga App Inventor Personal Server, karena sudah include server compilernya, benar-benar sangat menyenangkan buat yang ingin belajar App Inventor tetapi terhambat dengan koneksi internetnya.

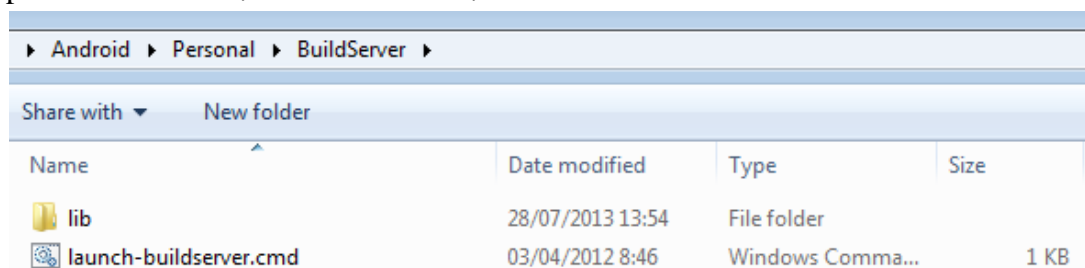
App Inventor true offline ini dipackage oleh **Gary Frederick** (Jefferson Software) dari source yang di rilis oleh MIT. Disini kita akan memakai versi windows, masih versi test tetapi sudah bagus dan sebanding dengan versi online-nya. OK sekarang waktunya untuk mencobanya :

1. Download file Personal.zip dari
<http://sourceforge.net/projects/ai4a-configs/files/Personal%20server/>
2. Menurut keterangan (Readme) sudah di test pada Windows 7, Saat di coba pada Windows XP SP3 dengan JDK 1.6, Block Editor tidak bisa terkoneksi ke Emulator. Dan ketika memakai Java JDK 1.7 bisa lancar, jadi pastikan apabila tidak bisa lancar menggunakan java JDK 1.6 maka pakailah Java JDK 1.7 download dari <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1637583.html>
3. Install terlebih dulu JDK 1.7
4. Install AppInventor_Setup_Installer_v_1_2.exe yang bisa di download <http://beta.appinventor.mit.edu/learn/setup/setupwindows.html>
5. Ekstrak file **Personal.zip** , bebas ditempatkan difolder mana, maka kita akan mendapatkan empat (4) buah folder yaitu :
 - a. appengine-java-sdk-1.6.4
 - b. AppInventor
 - c. BuildServer
 - d. Java



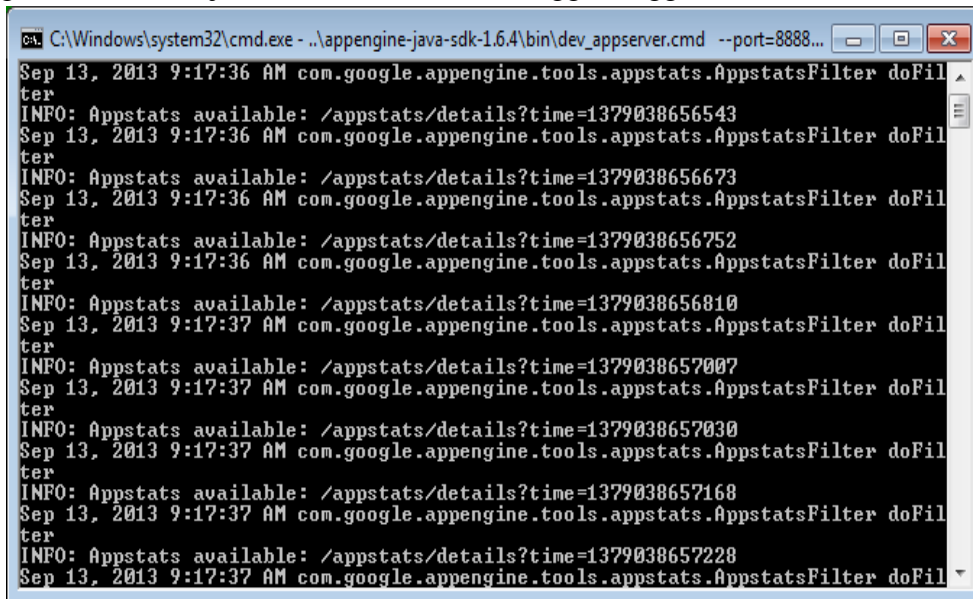
| | | |
|--------------------------|------------------|-------------|
| appengine-java-sdk-1.6.4 | 28/07/2013 13:54 | File folder |
| AppInventor | 28/07/2013 13:54 | File folder |
| BuildServer | 28/07/2013 13:54 | File folder |
| java | 28/07/2013 13:54 | File folder |

6. Untuk mencobanya, jalankan BuildServer AppInventor terlebih dulu, yaitu klik dua kali pada **Personal\BuildServer\launch-buildserver.cmd**



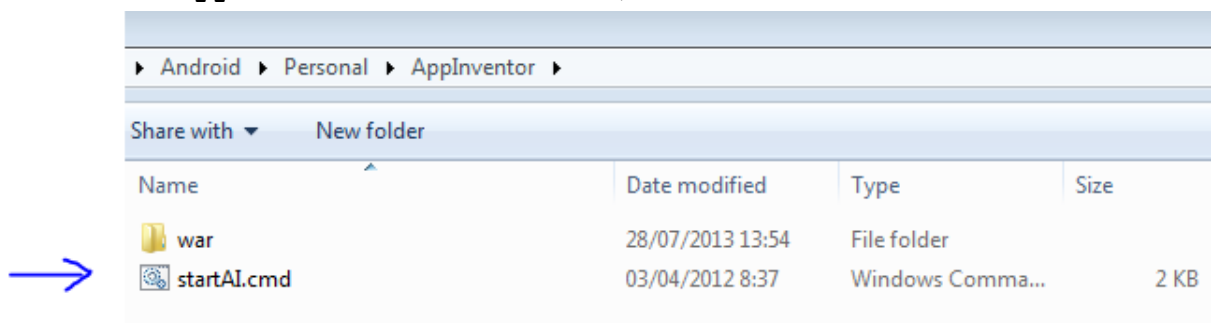
| Name | Date modified | Type | Size |
|------------------------|------------------|------------------|------|
| lib | 28/07/2013 13:54 | File folder | |
| launch-buildserver.cmd | 03/04/2012 8:46 | Windows Comma... | 1 KB |

Sehingga akan muncul jendela command line, tunggu hingga selesai

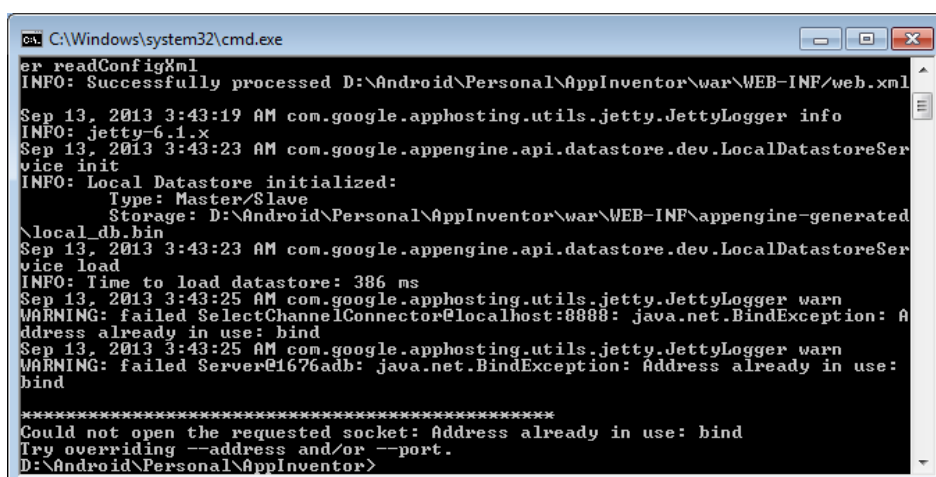


```
C:\Windows\system32\cmd.exe - ..\appengine-java-sdk-1.6.4\bin\dev_appserver.cmd --port=8888...
Sep 13, 2013 9:17:36 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038656543
Sep 13, 2013 9:17:36 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038656673
Sep 13, 2013 9:17:36 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038656752
Sep 13, 2013 9:17:36 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038656810
Sep 13, 2013 9:17:37 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038657007
Sep 13, 2013 9:17:37 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038657030
Sep 13, 2013 9:17:37 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038657168
Sep 13, 2013 9:17:37 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
INFO: Appstats available: /appstats/details?time=1379038657228
Sep 13, 2013 9:17:37 AM com.google.appengine.tools.appstats.AppstatsFilter doFilter
```

7. Jalankan Sever AppInventor, yaitu dari folder AppInventor klik dua kali pada **Personal\AppData\startAI.cmd**,



ini juga akan membuka jendela command line sampai proses selesai dan selanjutnya menghilang.



```
C:\Windows\system32\cmd.exe
er readConfigXml
INFO: Successfully processed D:\Android\Personal\AppData\war\WEB-INF\web.xml
Sep 13, 2013 3:43:19 AM com.google.apphosting.utils.jetty.JettyLogger info
INFO: jetty-6.1.x
Sep 13, 2013 3:43:23 AM com.google.appengine.api.datastore.dev.LocalDatastoreService init
INFO: Local Datastore initialized:
      Type: Master/Slave
      Storage: D:\Android\Personal\AppData\war\WEB-INF\appengine-generated\local_db.bin
Sep 13, 2013 3:43:23 AM com.google.appengine.api.datastore.dev.LocalDatastoreService load
INFO: Time to load datastore: 386 ms
Sep 13, 2013 3:43:25 AM com.google.apphosting.utils.jetty.JettyLogger warn
WARNING: failed SelectChannelConnector@localhost:8888: java.net.BindException: Address already in use: bind
Sep 13, 2013 3:43:25 AM com.google.apphosting.utils.jetty.JettyLogger warn
WARNING: failed Server@1676adb: java.net.BindException: Address already in use: bind

*****
Could not open the requested socket: Address already in use: bind
Try overriding --address and/or --port.
D:\Android\Personal\AppData>
```

8. Buka browser, dan masukkan alamat **localhost:8888**, maka akan terbuka jendela seperti berikut ini.

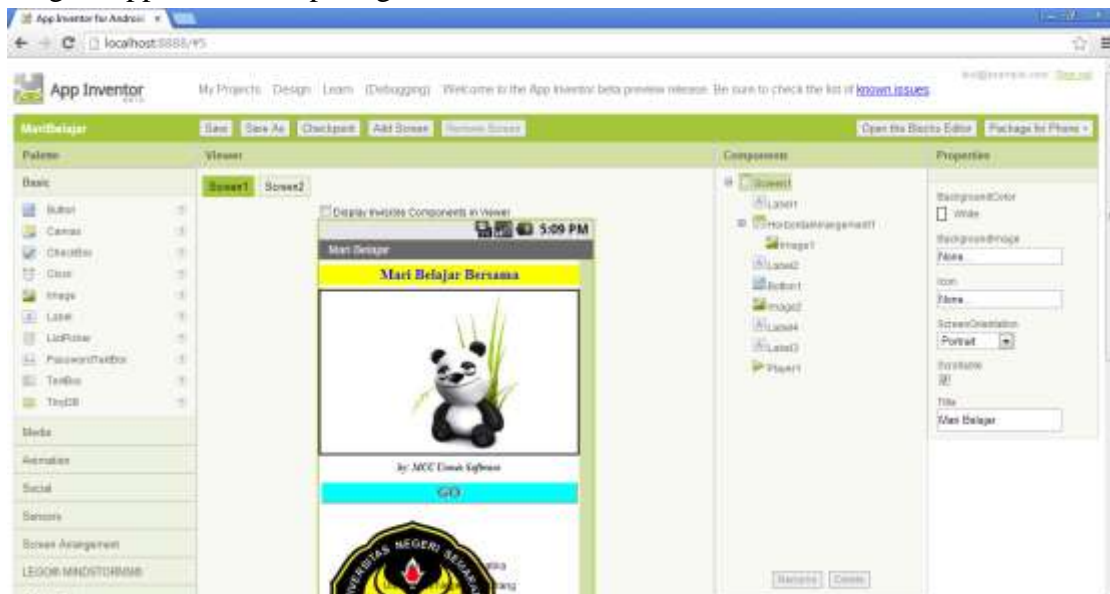
localhost:8888/_ah/login?continue=http%3A%2F%2F

Not logged in

Email:

Sign in as Administrator

Selanjutnya di cek pada **sign in as Administrator** kemudian di klik pada **Log in** sehingga muncul jendela desainer sudah bisa digunakan untuk membuat aplikasi android dengan App Inventor seperti gambar berikut ini.

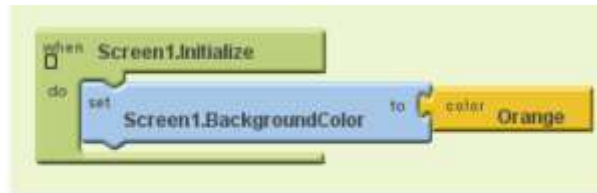


9. Alamat lain yang bisa dicoba adalah halaman admin yaitu pada **localhost:8888/_ah/admin**, silahkan di explore halaman admin ini.

MENGENAL APP INVENTOR LEBIH DALAM

Antarmuka

Apakah yang menarik pada App Inventor?, yang menarik dari App Inventor adalah, kita tidak lagi menemukan kode-kode program seperti halnya ketika menggunakan tools developing lain. Kode-kode itu sudah dibungkus kedalam objek visual, yang dinamakan Blocks. Kita tinggal memahami fungsi dari Blocks tersebut, kemudian bagaimana dia berinteraksi dengan Blocks lain, dan terakhir merangkainya persis seperti merangkai puzzle untuk membuat suatu bentuk atau gambar. Coba lihat pada gambar, seperti potongan puzzle bukan?, sangat mudah, dan menyenangkan, disini bisa kita anggap kita melewati fase untuk belajar mengenai bahasa pemrograman, dan berfokus pada belajar kreatifitas dan logika.



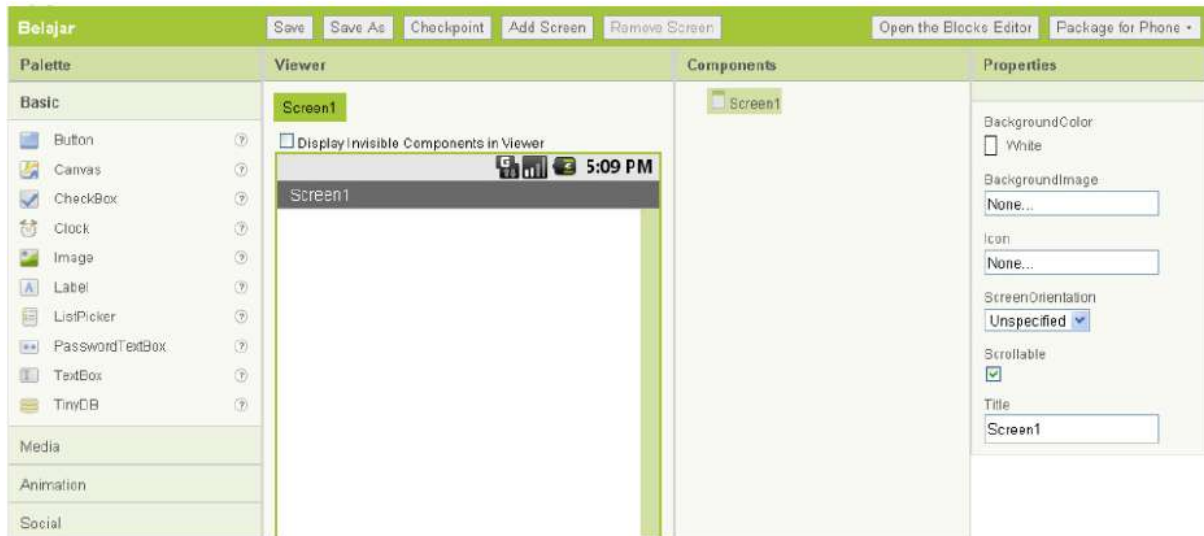
Gb 1. Contoh Blocks Mengubah Warna Screen

Berlatih membuat aplikasi adalah cara yang paling efektif untuk belajar membuat aplikasi dengan App Inventor. Kita tidak harus tahu dulu seluruhnya, cukup beberapa hal penting yang akan kita gunakan pada proses belajar ini. Seperti misalnya workspace apa saja yang akan kita gunakan, atau darimana kita memperoleh komponen dan blocks yang kita butuhkan. Untuk itu kita akan melihat sebenarnya apa saja yang kita butuhkan, dan apa yang penting untuk kita pahami terlebih dahulu.

Mari sekali lagi kita lihat, untuk membuat aplikasi dengan App Inventor ada tiga langkah utama yaitu :

1. Kita membuat UI dan memasukkan komponen-komponen App-Inventor yang akan kita pakai pada aplikasi dengan jendela Desainer kita
 2. Membuat komponen itu berfungsi yaitu dengan mengambil Block dari komponen, dan menyusunnya di Blocks Editor
 3. Mencoba aplikasi pada Emulator atau handset Android.
- A. **Desainer** : Membuat UI (*User Interface*) dan memasukkan komponen-komponen App-Inventor yang akan kita pakai pada aplikasi dengan jendela Desainer kita. Disini terdapat bagian Palette, Viewer, Components dan Properties. Selain itu juga terdapat beberapa menu penting yang akan kita gunakan.
- a. Pallette : Berisi seluruh komponen yg bisa kita pakai untuk membuat aplikasi
 - b. Viewer : Untuk menempatkan Komponen dan mendesain seperti apa tampilan atau UI dari aplikasi
 - c. Components : Berisi komponen yg telah kita ambil dan akan gunakan pada aplikasi
 - d. Properties : Disini kita bisa mengubah properti dari masing-masing komponen, misal warna layar.
 - e. Add/Remove Screen : Menambahkan atau menghapus screen, ini fitur baru pada App Inventor, yaitu dukungan Multiscreen.
 - f. Open The Blocks Editor : untuk mengaktifkan jendela Blocks Editor, sekali diaktifkan apabila Blocks Editor

- g. Package for Phone : Ketika aplikasi selesai dibuat, dan kita ingin mencobanya di handset android, maka bisa menggunakan menu ini.



Gb 2. Jendela Desainer

- B. **Blocks Editor** : adalah tempat untuk mengambil blocks (*kode program yg telah dijadikan objek visual*), dan menyusun Blocks-Blocks tersebut sehingga berfungsi sesuai dengan yang kita inginkan. Ada lima menu penting disini yaitu :
- a. Built-In : Blocks dasar dari komponen App Inventor
 - b. My Blocks : Blocks dari komponen yang kita gunakan untuk aplikasi, yaitu yang dimasukkan pada Desainer
 - c. Advance : Blocks pelengkap dari yang ada pada My Blocks
 - d. New Emulator : Menu untuk mengaktifkan Emulator
 - e. Connect To Devices : Menu untuk mengkoneksikan projek aplikasi dengan Emulator atau HP Android, yaitu ketika kita ingin mencoba Aplikasi yang sedang dibangun. Dan setelah sekali dikoneksikan maka ketika terjadi perubahan pada aplikasi yang sedang dibangun, misal komponen diganti, blocks diganti, secara otomatis akan dikomunikasikan ke Emulator atau HP Android.



Gb 3. Blocks Editor

C. Emulator Handset Android

Apabila kita ingin mencoba aplikasi android yang telah dibuat, kita bisa memakai Emulator, hampir semua yang bisa dilakukan oleh Handset Android bisa dilakukan oleh Emulator, kecuali menelepon, berkirim SMS, dan GPS. Jadi kita tidak usah kuatir, ketika membuat aplikasi Android dengan App Inventor, maka kita bisa menggunakan Emulator sebelum mencobanya di Handset atau HP Android.



Gb 4. Emulator

Apabila kita sudah memahami workspace dari App Inventor ini, maka tinggal kita mencobanya. Ada banyak contoh dari aplikasi App Inventor ini di internet. Dengan melihat komponen yang digunakan dan dipasang pada Desainer, atau melihat Blocks yang tersusun pada Blocks Editor, kita akan segera bisa membuat dan mencobanya sendiri. Jika ingin cepat menguasainya, kuncinya sekali lagi adalah ber latihan dan berlatih, pikirkan kita akan membuat aplikasi apa, kemudian cobalah untuk berlatih membuatnya.

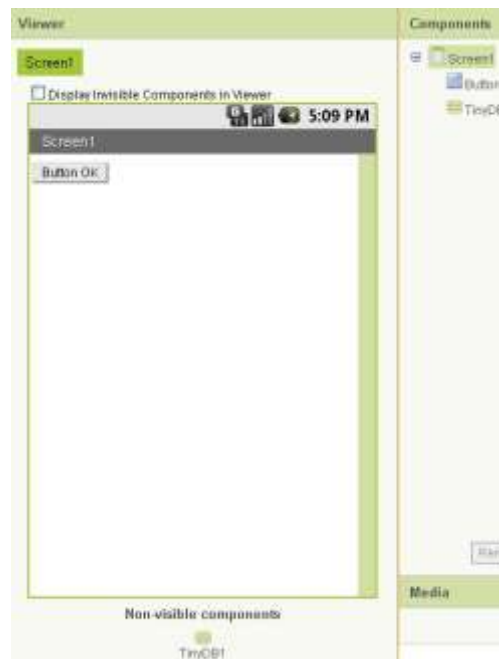
Buat yang belum pernah mengenal App Inventor, App Inventor adalah satu Tool webbased (Cloud) untuk membuat Aplikasi pada platform Android yang ditujukan untuk semua orang. Memakainya kita bisa membuat aplikasi android tanpa kode sama sekali.

Jendela Desainer

Setelah sebelumnya kita mengenal workspace pada App Inventor, selanjutnya kita akan mengenal beberapa hal penting secara detail pada Jendela Desainer.

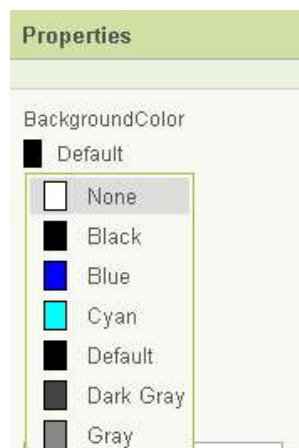
1. Pada jendela Desainer ketika kita memasukkan komponen kedalam Viewer maka ada komponen yang akan masuk kedalam layar Viewer (*Visible Component*) dan ada juga komponen yang secara otomatis akan masuk dibawah layar viewer (*Non Visual Component*). Komponen yang termasuk dalam *Visible Component* berarti akan terlihat pada layar aplikasi, sedangkan komponen *Non-visible Component* tidak akan terlihat pada

layar ketika aplikasi dijalankan. Pada gambar, tombol **Button OK** (*Button*) termasuk Visible Component, sedangkan **TinyDB1** merupakan Non-visible Component.



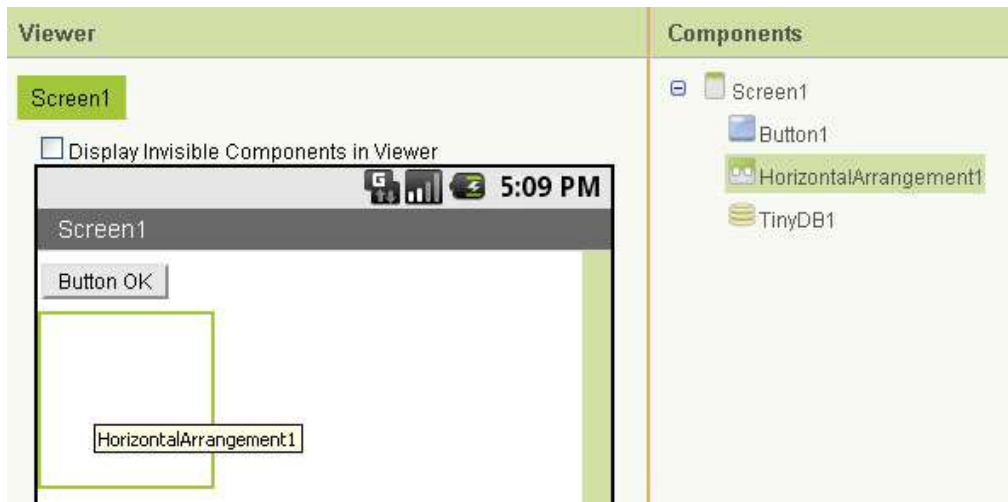
Gb1. Visible-Non-Visible

2. Jika ingin mengubah properti dari suatu komponen, maka kita bisa klik pada komponen tersebut, dan ubah melalui jendela properties. Misalnya mengubah background warna dari tombol (*button*).



Gb2. Properties

3. Untuk mengatur tata-letak komponen pada layar, maka digunakan komponen *ScreenArrangement*, terdapat tiga macam pengaturan yaitu :
 - a. *HorisontalArrangement* : Untuk mengatur komponen secara mendatar
 - b. *TableArrangement* : Untuk mengatur komponen pada sebuah tabel
 - c. *VerticalArrangement* : Untuk mengatur komponen secara tegaklurusKita bisa mengatur lebar dan tinggi dari masing-masing propertisnya, sedangkan khusus untuk *TableArrangement* bisa diatur jumlah kolomnya.



Gb3. ScreenArrangement

4. Terdapat komponen spesial untuk membuat aplikasi yang ditujukan khusus untuk robot Lego NXT. Hampir seluruh fungsi robot bisa dikendalikan, dengan komponen-komponen ini. Yaitu beberapa sensor, dan Direct Command (*Perintah*), maupun untuk mengendalikan Drive (*Arah*).



Gb.3 Lego_NXT

5. Pada menu Package for Phone, terdapat tiga macam pilihan yaitu :
 - a. Show Barcode : Untuk membuat barcode dari aplikasi yang dibuat
 - b. Download to This Computer : Mendownload aplikasi yang dibuat ke dalam komputer, kemudian bisa dicopy ke HP Android untuk dicoba.
 - c. Download to Connected Computer : Apabila kita mengkoneksikan HP Android semenjak awal, maka bisa langsung mencoba menjalankan aplikasi di HP.



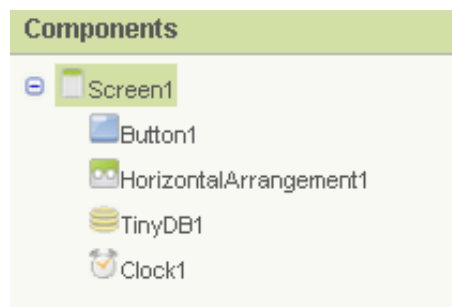
Gb.4 Package_Aplikasi

Blocks Editor

Sebelumnya kita telah mengenal mengenai workspace yang pada App Inventor, dan membahas secara detail Jendela Desainer. Dan workspace lainnya yang akan sering kita gunakan adalah Blocks Editor, yaitu sebuah tempat untuk mengambil dan menyusun blocks (*kode program yg telah dijadikan objek visual*) dari komponen yang kita pasang pada Jendela Desainer. Disini kita akan menyusun Blocks-Blocks tersebut sehingga aplikasi akan berfungsi sesuai dengan yang kita inginkan.

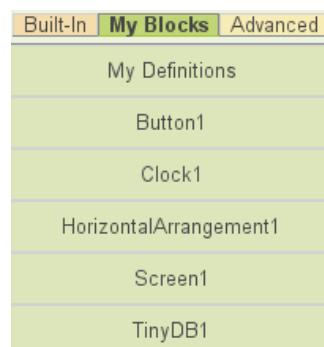
Kita akan lihat, dalam contoh projek berikut ini :

1. Pertama masukkan komponen **Button1**, **HorizontalArrangement1**, **TinyDB1** dan **Clock1** kedalam workspace Desainer, sehingga akan terlihat seperti gambar dibawah ini :



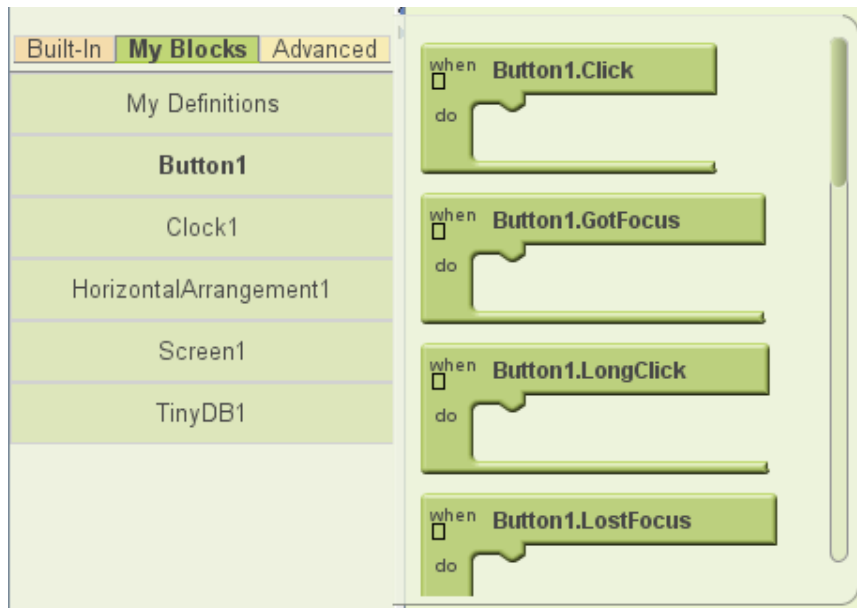
Gb 5. Komponen pada Jendela Desainer

2. Kemudian kita lihat pada Blocks Editor, yaitu pada bagian **My Blocks**. Disitu kita akan melihat Blocks komponen sudah masuk, yaitu Blocks komponen **Button1**, **HorizontalArrangement1**, **TinyDB1** dan **Clock1**. Apabila belum membuka Blocks Editor, silahkan klik tombol Open Block Editor pada jendela Desainer.



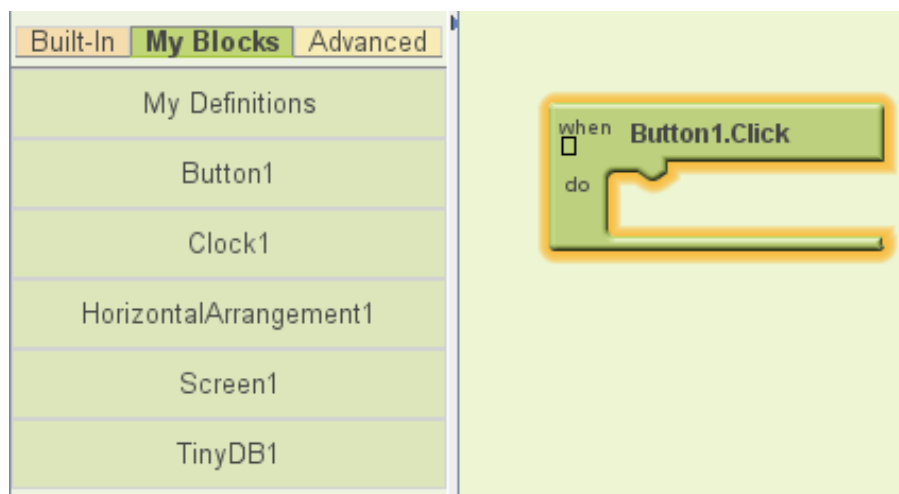
Gb6. Blocks Komponen pada Block Editor

3. **My Blocks** adalah tempat dimana Blocks komponen yang kita pasang pada jendela Desainer berada, sedangkan **Built-In** adalah tempat semua blocks dasar yang ada pada App Inventor. **Advance Blocks** pelengkap dari yang ada pada My Blocks. Jika kita ingin menyusun Blocks yang terkait dengan komponen yang kita gunakan, maka klik nama komponen pada My Blocks, sehingga akan muncul blocks didalamnya.



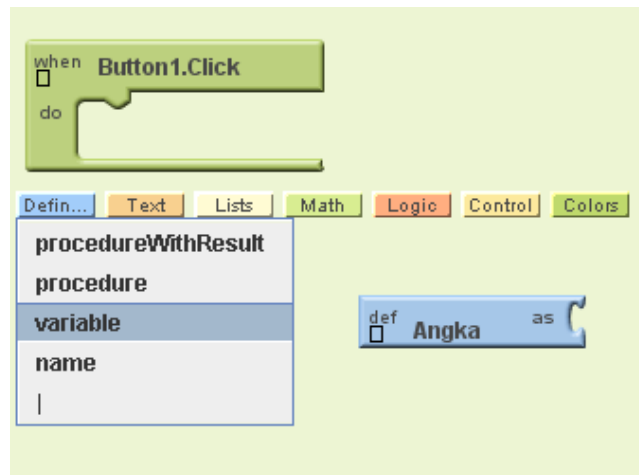
Gb7. Blocks Komponen Button

- Pilih pada Block yang akan kita pakai, kemudian drag kedalam workspace Blocks Editor, misalnya untuk Button, Blocks yang sering akan kita pakai adalah **Blocks Button1.Click**. Yaitu Block yang berfungsi handle event apabila Button atau tombol di klik, sedangkan nama **Button1** merujuk pada komponen Button yang kita pakai, apabila kita ganti (*rename*) nama komponennya menjadi **Button10** maka nama Blocks akan menjadi **Button10.Click**.



Gb8. Blocks Button1

- Pada My Blocks terdapat satu Blocks yang akan selalu ada yaitu **My Definitions**, akan tetapi apabila kita belum membuat satu variable tertentu maka Blocks ini akan kosong. Misalnya disini kita akan membuat **Blocks Variabel** dengan nama **Angka**. Caranya bisa dari **Built-In > Definitions > def variable as**, atau **klik pada workspace > Def > variable**



Gb9. Membuat Blocks Variable

6. Sekarang klik pada **My Definitions**, maka akan terlihat Blocks yang kita buat sendiri, yaitu **Blocks Angka**. Pointnya adalah disini kita membuat Blocks yang dibutuhkan oleh aplikasi kita, namun tidak tersedia secara dasar, ataupun bukan merupakan Blocks dari komponen. Misalnya variable atau penampung suatu nilai.



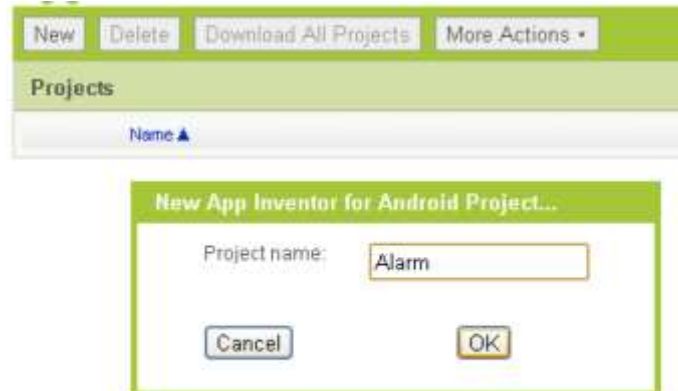
Gb10. Blocks My Definitios

Langkah-langkah diatas adalah, hal-hal penting yang harus diketahui dan dipahami pada Blocks Editor. Selain itu silahkan bereksperimen sendiri, dan mencoba Blocks yang ada pada setiap komponen. Yang perlu diketahui adalah, nama dari setiap Blocks mencerminkan fungsi dari Blocks tersebut. Kemudian, hal lain yang harus juga diperhatikan adalah bahwa Blocks ini menentukan fungsi yang kita inginkan pada suatu aplikasi, jadi yang berperan disini adalah Logika dan Kreatifitas kita.

Contoh:

Ok, sekarang kita akan langsung mencoba membuat Aplikasi Alarm, agar kita bisa cepat memahami bagaimana membuat aplikasi dengan App Inventor. Aplikasi ini berfungsi seperti Alarm tetapi dengan keunikan untuk bisa mematikan bunyi dari Alarm kita harus menghitung sebuah perkalian. Tujuannya adalah, ketika Alarm difungsikan sebagai Alarm untuk membangunkan kita dipagi hari, kita akan dipaksa untuk bangun dengan berpikir dan menghitung, asyik bukan?

1. Jalankan AppInventor kemudian login. Klik pada menu **New** > Beri nama aplikasi misal disini Alarm, kemudian klik OK sehingga akan terlihat nama aplikasi Alarm pada jendela projek, selanjutnya klik pada nama aplikasi tersebut



GB 1 . New Projek

2. Akan terbuka workspace Komponen Desainer, disini kita akan memasukkan seluruh komponen yang dibutuhkan untuk membangun aplikasi Alarm ini. Untuk memasukkan kita tinggal mengklik komponen dari **Pallette** disisi kiri, jangan dilepas kemudian menariknya ke layar workspace, sedangkan untuk mengeset propertinya, silahkan dipilih pada jendela **Properties** di sisi sebelah kanan.
3. Kita menggunakan komponen **HorizontalArrangement** untuk menempatkan komponen lainnya agar lebih mudah diatur. Cara mendapatkannya dari **Pallette** > **Screen Arrangement** > **HorizontalArrangement**, kemudian drag ke workspace screen. Kemudian kita juga menempatkan beberapa buah Label, Textbox, juga sebuah tombol.
4. Komponen sound kita ambil dari **Pallette** > **Media** > **Sound** , komponen Sound ini untuk menangani file suara, yaitu dengan menambahkan (*add*) file **Sound** dari properti **Source** > **add** > dan kemudian browse file dari komputer. Kita bisa memilih file sound berformat **mp3** atau **wav**, yang perlu kita perhatikan adalah, semakin besar file sound semakin besar juga nanti aplikasi kita, jadi disini perlu dipertimbangkan tentang hal itu. Sedangkan komponen **Timer** akan kita gunakan untuk membuat file **sound** agar selalu aktif berbunyi.

| No | Nama Komponen | Property | Tempat |
|----|--|---|--|
| 1 | HorizontalArrangement : HorizontalArrangement1 s/d 5 | | Pallette > Screen Arrangement > HorizontalArrangement |
| 2 | Label : LabelInfo, LabelAngkaSatu, LabelAngkaDua, LabelPerkalian, LabelInfoHitung, LabelInfoPenilaianJawaban, LabelPenilaianJawaban | Ubah jenis, font size juga pada Text | Pallette > Basic > Label |
| 3 | Button : ButtonJawab | Ubah font size, Bold | Pallette > Basic > Basic |
| 4 | TextBox : TextboxJawaban | Ubah font size, | Pallette > Basic > Basic |
| 5 | Sound : Sound1 | Source > Add Sound | Pallette > Media > Basic |
| 6 | Clock : Clock1 | | Pallette > Media > Basic |



GB2. Komponen Desainer

5. Agar memudahkan kita ketika membuat event di **Blocks Editor**, silahkan rename nama komponen pada jendela **Components** yaitu pada menu **Rename**. Apabila kita salah atau ingin mengganti komponen yang dimasukkan, silahkan hapus dengan menu **Delete** disamping menu **Rename**, sebelumnya pilih terlebih dahulu komponen yang akan dihapus dengan mengkliknya.
6. Setelah selesai memasukkan semua komponen dan mengeset propertinya, klik pada menu **Open Blocks Editor** disamping kanan atas. Tunggu beberapa saat hingga terbuka jendela **Blocks Editor**, kemudian lihat pada drawer **My Blocks**, ini adalah tempat event fungsi dari masing-masing komponen. Ketika kita klik pada salahsatu komponen, maka akan muncul event fungsi apa saja yang ada pada komponen tersebut. Agar menghemat waktu aktifkan juga **Emulator**, dengan cara klik pada menu **New Emulator**, dan biarkan proses pengaktifan Emulator selesai. Setelah **Emulator** aktif dan untuk mengkoneksikan projek kita dengan Emulator maupun Handset Android kita bisa memilih menu **Connect to Device**.



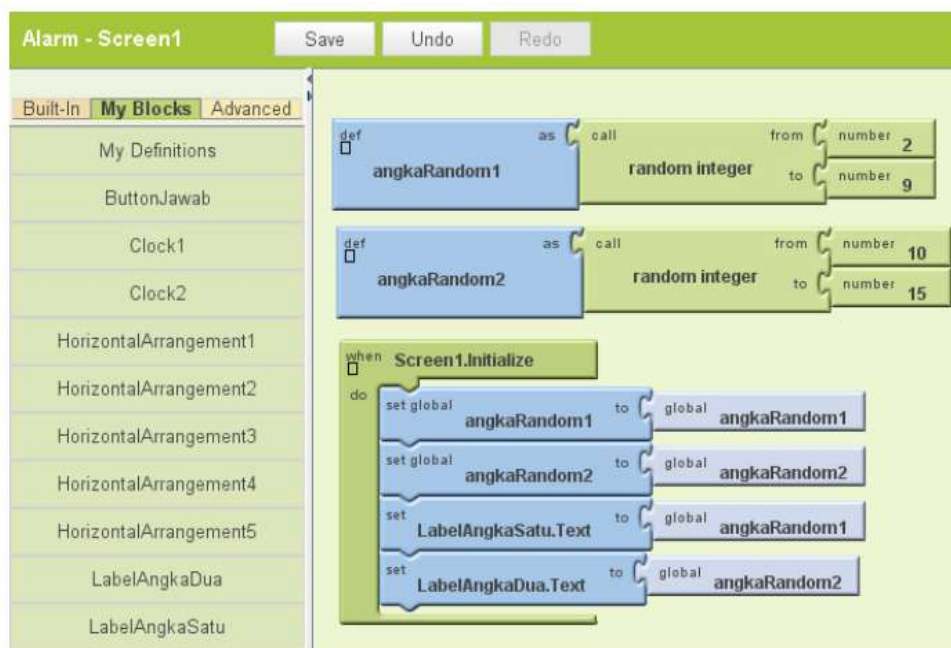
GB3. Block Editor

7. Sekarang waktunya untuk memulai memprogram aplikasi kita dengan membuat event fungsi, pertamanya kita harus memahami pola kerja aplikasi secara logika terlebih dahulu. Untuk Alarm ini karena kita ingin membuat perhitungan perkalian maka kita membutuhkan dua buah angka acak (*Random*). Buat terlebih dahulu sebuah penampung var iabel, untuk menampung angka tersebut, kita bisa namakan bebas misal disini **angkaRandom1** dan **angkaRandom2**, untuk membuatnya klik pada **workspace > defin > variable** (atau bisa dari drawer **Buil-In > Definition > Variable**) drag ke layar, klik pada blocks dan rename teks “variabel” menjadi **angkaRandom1**.



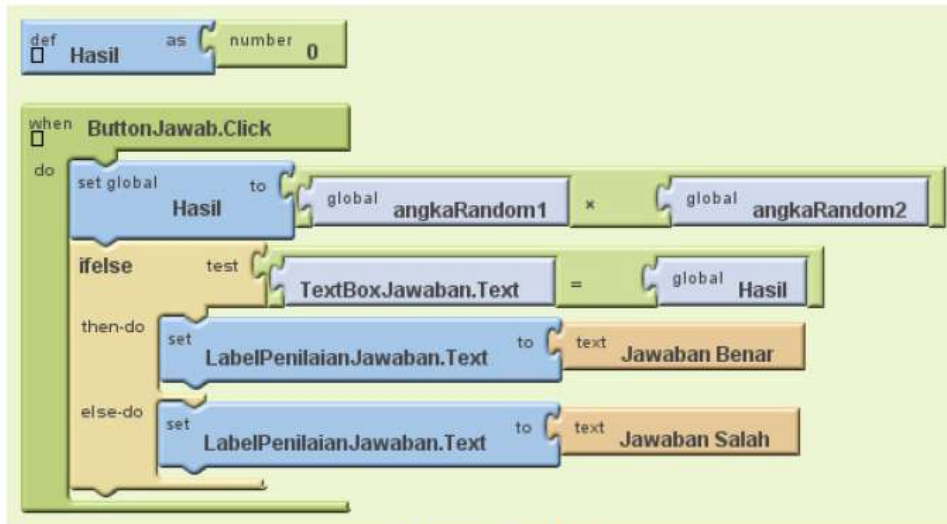
GB4 Angka Random

8. Kemudian ambil sebuah block **Random Integer** untuk membuat batas angka random kita, dari drawer **Built-In > Math > Random Integer**. Kemudian isi slot **From** dan **To** dengan angka yang kita inginkan, misal disini range angka acaknya adalah antara 2 sd 9 untuk **angkaRandom1**, dan 10 sd 15 untuk **angkaRandom2**.
9. Kemudian, kita membutuhkan block event yang berfungsi mengeksekusi block-block event didalamnya, pada saat aplikasi dijalankan, yaitu block **Initialize**, karena disini kita menggunakan layar dgn nama **Screen1**, maka nama bloknya adalah **Screen1.Initialize**, yang bisa ambil dari drawer **My Blocks > Screen1.Initialize**. Disini kita akan menggunakan block **Screen1.Initialize** untuk menampilkannya angka random yang kita dapatkan tadi, yaitu dengan memasukkannya ke label **LabelAngkaSatu** dan **LabelAngkaDua**. Sampai tahap ini kita sudah bisa mencobanya pada emulator, yaitu dengan memilih menu **Connect to Device**.



GB5 Initialize Random

10. Selanjutnya adalah menangani event perkalian itu sendiri, kita akan membutuhkan hasil dari operasi perkalian antara **angkaRandom1** dan **angkaRandom2**. Dan kemudian membandingkannya dengan angka jawaban yang kita masukkan pada **TextBoxJawaban**, dan menempatkan block kondisi untuk menilai apakah jawaban benar atau tidak.
11. Disini kita membutuhkan Block Event ketika tombol **ButtonJawab** diklik, yaitu **ButtonJawab.Click**. Block ini kita ambil dari **My Blocks > ButtonJawab > ButtonJawab.Click**, kemudian block untuk perkalian angka dari **Built-In > Math > X Block operasi perkalian**. Buat terlebih dulu block penampung variabel, kita namakan **Hasil**, yang digunakan untuk menampung hasil perkalian. Masukkan block **angkaRandom1** dan **angkaRandom2** untuk dikalikan, kemudian hasilnya disimpan pada variabel **Hasil** yang telah kita buat sebelumnya.



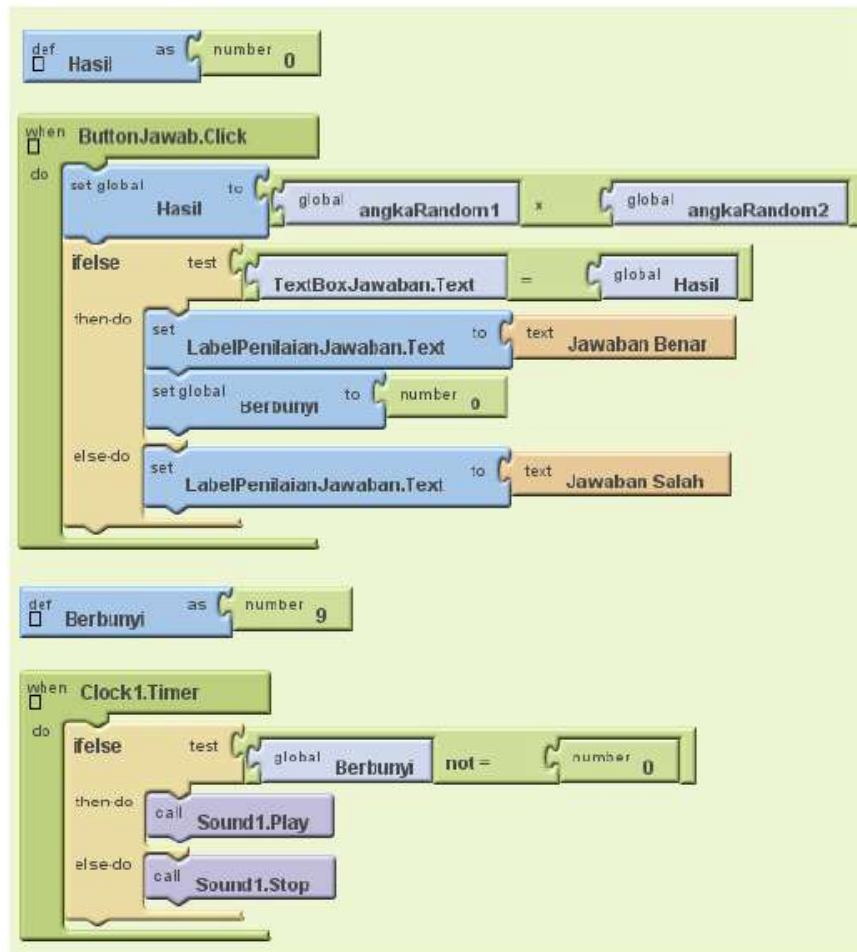
GB6 Button Block

12. Kemudian ambil **block kondisional If else**, dari **Built-In > Control > if else**, ini kita akan gunakan untuk membandingkan antara angka hasil perkalian dengan jawaban. Yaitu Block **Hasil** dengan angka dari **TextBoxJawaban**, apabila sama set **LabelPenilaianJawaban** dengan Block teks "**Jawaban Benar**", apabila tidak sama set **LabelPenilaianJawaban** dengan Block teks "**Jawaban Salah**".
13. Sampai disini kita bisa mencobanya, pada emulator. Ketik jawaban pada **Textbox**, kemudian klik pada tombol **Klik Jawab**. Maka apabila jawaban kita benar, maka akan muncul teks "**Jawaban Benar**", begitu juga dengan sebaliknya.



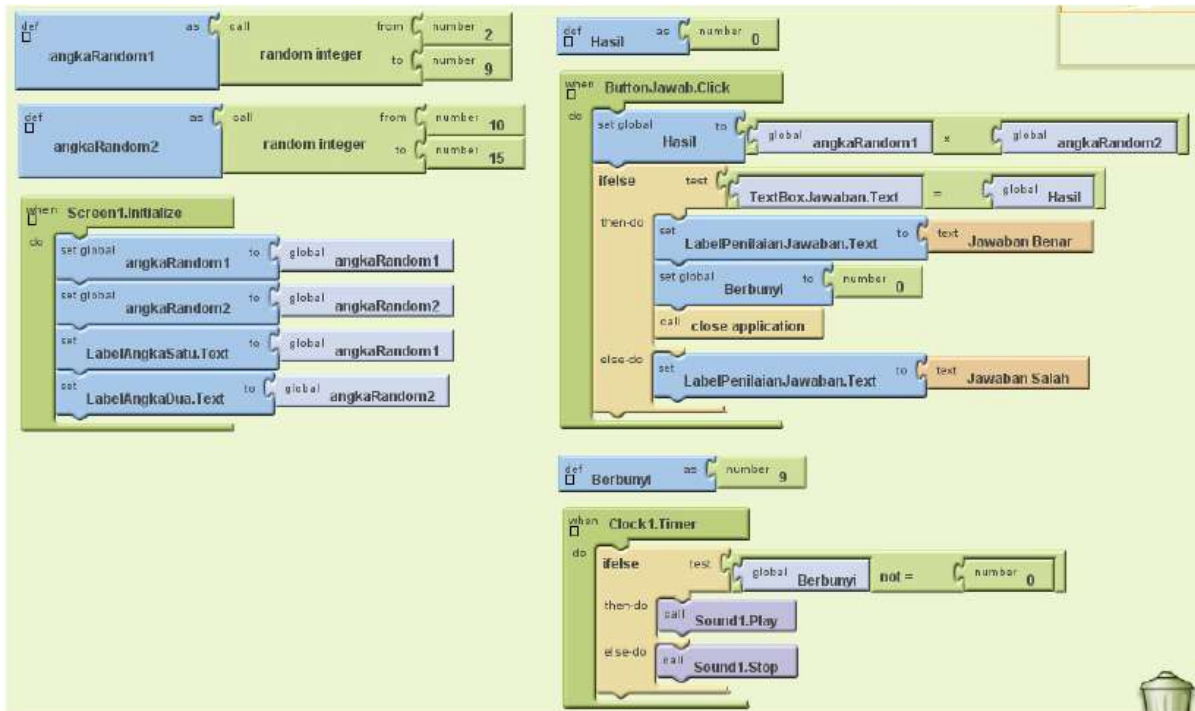
GB7. Test Aplikasi

14. Tahap selanjutnya kita akan menambahkan event bunyi alarm, yaitu dengan menggunakan blocks komponen **Clock1.Timer** dan **Sound1**. Disini kita akan menyalakan terus menerus suara. Dari komponen **Sound1**, yang sudah kita masukkan kedalam propertinya file suara. Idenya adalah, suara akan terus berbunyi selama kita salah menjawab perhitungan perkalian. Dan dalam implementasinya kita menggunakan block **block kondisional If else**, untuk membandingkan kondisi yang kita buat, yaitu dengan membuat sebuah variabel dengan nilai 9 (*sembilan*) dan membuat selama nilai dari variabel **Berbunyi = 9** , maka suara akan terus terdengar. Sedangkan untuk mematikannya adalah kita menambahkan satu block yang akan mengeset variabel **Berbunyi = 0** , yaitu ketika kita mengklik tombol **Klik Jawaban**



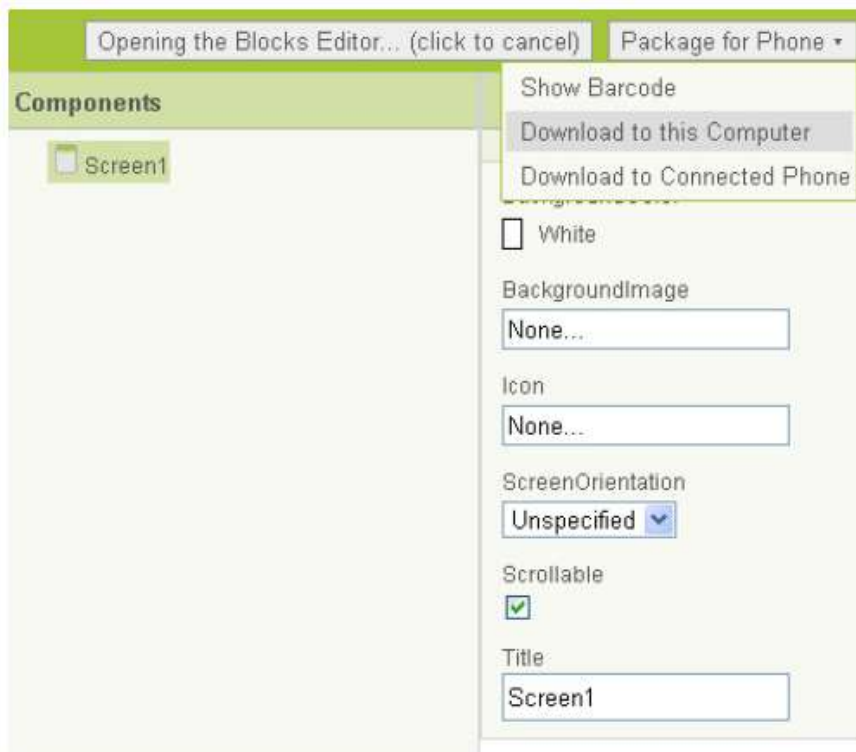
GB8. Block Timer dan Suara

15. Terakhir kita menambahkan block close application, yang kita ambil dari drawer **Built-In > Control > close application** . Block ini merupakan event untuk menutup aplikasi, dan kita pasang pada bagian block kondisi **ButtonJawab.Click**, sehingga aplikasi akan langsung tertutup ketika kita mengklik tombol **Klik Jawaban**, dan jawaban kita benar.



GB9. Block Lengkap

16. Untuk menggunakan aplikasi ini pada HP Android, Download terlebih dari menu **Package for Phone** pada **Desainer Komponen**, pilih **Download to This Computer**, sehingga aplikasi akan terdownload ke komputer kita, kemudian copy ke HP Android. Pilihan **Download to Conected Phone** bisa kita pilih ketika kita mengkoneksikan HP Android kita, dan mengaturnya pada mode **USB debugging connected**.



GB10. Package for Phone

17. Untuk menggunakannya sebagai alarm, kita harus mengeset waktu kapan aplikasi alarm ini akan berfungsi, dengan mengaturnya melalui aplikasi **Clock > Add Alarm > Set Waktu > Application** > pilih aplikasi **Alarm** pada **App to launch > Done** . Ini berarti aplikasi Alarm yang kita buat masih harus menggunakan aplikasi lain untuk memanggilnya, karena disini kita belum membuat event fungsi untuk mengeset waktu yang secara otomatis akan memanggil aplikasi.

Aplikasi yang kita buat barusan, terasa sederhana dan mudah membuatnya, ini karena App Inventor memang didesain untuk memudahkan siapapun membuat aplikasi android.

Referensi:

- Mulyadi, 2011, *Membuat Aplikasi Android & Android App Inventor*, Yogyakarta: Multimedia Center Publishing.
- Tabloid PC Plus 2012