

Tutorial Mudah Membuat Aplikasi Android  
Dengan

# MIT APP INVENTOR (AI2)



Agus Suharto



Tutorial Mudah Membuat Aplikasi Android  
Dengan

# **MIT APP INVENTOR (AI2)**

**Agus Suharto**

**MUDAH MEMBUAT APLIKASI ANDROID  
DENGAN MIT APP INVENTOR (AI2)**

Indramayu © 2021, Penerbit Adab

Penulis: Agus Suharto

Editor: Abdul

Perancang Sampul: Nurul Musyafak

Layouter: Fitri Yanti

Diterbitkan oleh **Penerbit Adab**

CV. Adanu AbimataAnggota IKAPI: 354/JBA/2020

Jln. Jambal II No 49/A Pabean Udik Indramayu Jawa Barat

Kode Pos 45219 Telp: 081221151025

Surel: [penerbitadab@gmail.com](mailto:penerbitadab@gmail.com)

Web: <https://penerbitadab.id>

*Referensi / Non Fiksi / R/D*

xiv + 102 hlm. ; 15,5 x 23 cm

No ISBN: 978-623-6233-97-9

Cetakan Pertama, September 2021



**Hak Cipta dilindungi undang-undang.**

Dilarang memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apapun, secara elektronik maupun mekanis termasuk fotokopi, merekam, atau dengan teknik perekaman lainya tanpa izin tertulis dari penerbit.

*All right reserved*



# KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah saya panjatkan puja dan puji syukur kehadiran Allah swt yang senantiasa melimpahkan segala rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan buku dengan judul "Mudah membuat aplikasi android dengan MIT APP Inventor" (AI2) .

Buku ini disusun dari sumber web resmi MIT App Inventor pada bagian *Tutorials* dan e-book "*Educational App Development Toolkit for Teachers and Learners*" ditulis oleh "Ishan Sudeera Abeywardena" serta aplikasi sederhana yang dibuat oleh penulis. Pokok dari penulisan ini adalah memenuhi kebutuhan kalangan masyarakat umum, pelajar, mahasiswa/i dari berbagai program studi yang ingin belajar membuat aplikasi berbasis *mobile* android dengan mudah tanpa *coding*.

Teknik penyajian yang diangkat dilakukan berupa tutorial membuat aplikasi sederhana berbasis android yang pernah penulis buat. . .

Pembahasan modul ini dimulai dengan menjelaskan Pendahuluan MIT App Inventor, lalu dilanjutkan membuat aplikasi sederhana desain, *blocks*, sampai uji coba ke perangkat *gadget* android

Penyusun menyadari bahwa di dalam pembuatan buku ini masih banyak kekurangan, untuk itu penyusun sangat membuka saran dan kritik yang sifatnya membangun. Mudah-mudahan buku ini memberikan manfaat bagi kita semua.

Aamiin Ya Rabbal Alamin.

Jakarta, 20 September 2021

Penulis

# DAFTAR ISI

KATA PENGANTAR .....	iii
DAFTAR ISI .....	v
DAFTAR GAMBAR .....	vii
DAFTAR TABEL .....	xiv
 BAB I    PENGANTAR MIT APP INVENTOR.....	 1
1.1. PENDAHULUAN .....	2
1.2 Membuat Akun Aplikasi App Inventor 2 (AI2) .....	4
1.3 Lingkungan Pengembangan.....	7
1.4 <i>Setting</i> Perangkat untuk Pengembangan Aplikasi dan <i>Debugging</i> .....	8
 BAB II    TUTORIAL MEMBUAT APLIKASI MIT APP INVENTOR (PEMULA).....	  25
2.1 MEMBANGUN APLIKASI PERTAMA: HELLO CODII! .....	26
2.2 MEMBANGUN APLIKASI KEDUA: <i>TALK TO ME 1</i> ...	38
2.3 MEMBANGUN APLIKASI KETIGA: <i>TALK TO ME 2</i> ,	

<i>USER INPUT &amp; ACCELOROMETER SENSOR</i> .....	43
2.4 MEMBANGUN APLIKASI KE EMPAT:	
<i>BALLBOUNCE/Aplikasi game sederhana</i> .....	49
2.5 MEMBANGUN APLIKASI KE LIMA:	
DigitalDoodle/APLIKASI MENGGAMBAR .....	57
 BAB III TUTORIAL MEMBANGUN APLIKASI dengan	
BLOCK MATH, dan CONTROL.....	63
3.1 APLIKASI MENGHITUNG LUAS SEGITIGA .....	64
3.2 APLIKASI MENENTUKAN KELULUSAN NILAI	
AKHIR.....	80
3.3 APLIKASI KALKULATOR BMI (Body Mass Index) ....	88
 BAB IV MENDISTRIBUSIKAN, MENGUBAH PROJECT MIT	
APP INVENTOR MENJADI FILE APK ANDROID .....	95
4.1 EKSPOR FILE PROJECT APLIKASI KE	
KOMPUTER/LAPTOP (.AIA FILE).....	96
4.2 MEMBUAT APLIKASI .AIA MENJADI FILE .APK .....	98
 DAFTAR PUSTAKA .....	101
PROFIL PENULIS .....	102

# DAFTAR GAMBAR

Gambar 1. 1	<i>link halaman <a href="http://ai2.appinventor.mit.edu">http://ai2.appinventor.mit.edu</a>.....</i>	5
Gambar 1. 2	<i>pemilihan Akun google untuk akun authentication App Inventor.....</i>	5
Gambar 1. 3	<i>Halaman beranda MIT App Inventor.....</i>	6
Gambar 1. 4	<i>Halaman Welcome MIT App Inventor.....</i>	6
Gambar 1. 5	<i>tampilan jendela desainer dari AI2.....</i>	7
Gambar 1. 6	<i>tampilan jendela desainer dari AI2.....</i>	8
Gambar 1. 7	<i>Membangun apps dengan perangkat android dan koneksi Wif.....</i>	9
Gambar 1. 8	<i>Aplikasi MIT AI2 Companion pada play store.....</i>	10
Gambar 1. 9	<i>Halaman daftar Project Aplikasi MIT AI2.....</i>	11
Gambar 1. 10	<i>Halaman desainer Project Aplikasi MIT AI2.....</i>	11
Gambar 1. 11	<i>QRCode untuk pengujian Aplikasi MIT AI2.....</i>	12
Gambar 1. 12	<i>tampilan MIT AI2 pada ponsel untuk pengujian..</i>	12
Gambar 1. 13	<i>tampilan hasil uji pada ponsel Android.....</i>	13
Gambar 1. 14	<i>Membuat project pada laptop/Komputer dan menguji pada Emulator.....</i>	14
Gambar 1. 15	<i>file setup untuk mac os.....</i>	15
Gambar 1. 16	<i>Standar install setup untuk mac os.....</i>	15
Gambar 1. 17	<i>App inventor setup sukses dan telah di install.....</i>	16



Gambar 1. 18	<i>Pesan konfirmasi install MIT AI2.....</i>	17
Gambar 1. 19	<i>Ikon AI Start Windows .....</i>	20
Gambar 1. 20	<i>Tampilan AIStart .....</i>	21
Gambar 1. 21	<i>Koneksi Emulator dari AI2.....</i>	21
Gambar 1. 22	<i>Tampilan jendela selama proses start-up.....</i>	22
Gambar 1. 23	<i>Tampilan empat fase emulator mulai dari Start-Up.....</i>	22
Gambar 1. 24	<i>Membangun aplikasi menggunakan kabel USB...</i>	23
Gambar 2. 1	<i>Aplikasi Hello Codi .....</i>	27
Gambar 2. 2	<i>pemilihan akun google sebagai akun MIT AI2 .....</i>	27
Gambar 2. 3	<i>Memulai project MIT AI2.....</i>	28
Gambar 2. 4	<i>Penamaan Project MIT AI2.....</i>	28
Gambar 2. 5	<i>Komponen komponen pada Pallete.....</i>	29
Gambar 2. 6	<i>Pemilihan Komponen untuk aplikasi Hello Codi!</i>	30
Gambar 2. 7	<i>Pengaturan properties Text pada Komponen Button .....</i>	31
Gambar 2. 8	<i>Pengaturan properties Text, Font Size, BackGround Color, TextColor .....</i>	32
Gambar 2. 9	<i>Penyisipan komponen Sound ke jendela Viewer</i>	33
Gambar 2. 10	<i>Mengaktifkan jendela Block Editor.....</i>	33
Gambar 2. 11	<i>Pemilihan Block When Button1.Click Block When Button1.Click .....</i>	34
Gambar 2. 13	<i>Hasil penyisipan Blocks call sound.play ke dalam Block When Button1.Click .....</i>	35
Gambar 2. 14	<i>Meng koneksikan aplikasi ke perangkat android</i>	36
Gambar 2. 15	<i>aplikasi MIT AI2 android sebagai pengujian.....</i>	36
Gambar 2. 16	<i>aplikasi MIT AI2 android menscan QRCode dari project.....</i>	37

Gambar 2. 17	<i>Pogress uji aplikasi yang akan ditampilkan ke perangkat android.....</i>	37
Gambar 2. 18	<i>Hasil uji aplikasi ditampilkan ke perangkat android.....</i>	37
Gambar 2. 19	<i>Aplikasi Talk To Me 1.....</i>	38
Gambar 2. 20	<i>Penamaan project Aplikasi Talk To Me.....</i>	39
Gambar 2. 21	<i>Set Properties Text Aplikasi Talk To Me.....</i>	39
Gambar 2. 22	<i>menempatkan pallet Media TextToSpeech.....</i>	40
Gambar 2. 23	<i>pemilihan jendela blocks untuk pemrograman....</i>	40
Gambar 2. 24	<i>pemilihan blocks When Button1.click.....</i>	41
Gambar 2. 25	<i>pemilihan blocks Call TextToSpeech1.speak.....</i>	41
Gambar 2. 26	<i>pemilihan blocks Text untuk ditempatkan pada slot message.....</i>	42
Gambar 2. 27	<i>Pengisian Pesan Text "Congratulation You've made your frst app".....</i>	42
Gambar 2. 28	<i>Hasil pengujian ke perangkat android.....</i>	43
Gambar 2. 29	<i>Aplikasi Talk To Me 2.....</i>	43
Gambar 2. 30	<i>Penamaan Aplikasi Talk To Me 2.....</i>	44
Gambar 2. 31	<i>Penyisipan Pallet UI Button ke Viewer Desainer.....</i>	45
Gambar 2. 32	<i>Penyisipan Pallet AccelerometerSensor.....</i>	45
Gambar 2. 33	<i>Mengaktifkan Jendela Block untuk pemrograman.....</i>	46
Gambar 2. 34	<i>Pemilihan even When Button1.Click untuk pemrograman.....</i>	46
Gambar 2. 35	<i>Pemilihan even Call TextToSpeech1.Speech.....</i>	47
Gambar 2. 36	<i>Pemilihan Block TextBox1.text ke slot message....</i>	47
Gambar 2. 37	<i>WhenAccelerometer1.shaking(#2).....</i>	48
Gambar 2. 38	<i>Hasil akhir Blocks Aplikasi TalkToMe2.....</i>	49
Gambar 2. 39	<i>Aplikasi BallBounce.....</i>	49

Gambar 2. 40	<i>Penamaan Project Aplikasi BallBounce .....</i>	50
Gambar 2. 41	<i>Penyisipan Blocks Canvas ke dalam screen Viewer</i>	51
Gambar 2. 42	<i>Setting properties Scrollable di uncheck.....</i>	51
Gambar 2. 43	<i>Setting properties Canvas Height dan Width .....</i>	52
Gambar 2. 44	<i>Menyisipkan Drawing Ball.....</i>	52
Gambar 2. 45	<i>Mengaktifkan Jendela Block untuk pemrograman</i>	53
Gambar 2. 46	<i>Menyisipkan Block When Ball.Flung ke Viewer.....</i>	53
Gambar 2. 47	<i>Menyisipkan Blocks set Ball1.Speed to dan Set Ball1.Heading to.....</i>	54
Gambar 2. 48	<i>Menyisipkan Blocks variable Get dengan nilai speed dan heading .....</i>	54
Gambar 2. 49	<i>Menyisipkan When Ball1.EdgeReached .....</i>	55
Gambar 2. 50	<i>Call Ball1.Bounce Edge ke When Ball1.EdgeReached</i>	55
Gambar 2. 51	<i>Menyisipkan Variable dengan Get Edge.....</i>	56
Gambar 2. 52	<i>Block Keseluruhan untuk aplikasi Ball Bounce .....</i>	56
Gambar 2. 53	<i>Hasil pengujian aplikasi Ball Bounce pada perangkat ponsel .....</i>	57
Gambar 2. 54	<i>Aplikasi Digital Doodle.....</i>	58
Gambar 2. 55	<i>Aplikasi Digital Doodle.....</i>	58
Gambar 2. 56	<i>Setting properties Scrollable dengan uncheck .....</i>	59
Gambar 2. 57	<i>penyisipan Pallette Canvas ke jendela Viewer Screen.....</i>	59
Gambar 2. 58	<i>Setting properties Heigh dan Width canvas.....</i>	60
Gambar 2. 59	<i>Mengaktifkan Jendela Block untuk pemrograman</i>	60
Gambar 2. 60	<i>Menyisipkan Canvas dengan memilih Block When Canvas.Dragged.....</i>	61
Gambar 2. 61	<i>Menyisipkan Block Call Canvas.Drawline .....</i>	61

Gambar 2. 62	<i>Memberikan Nilai Block Call Canvas.Drawline dengan <math>X1 = \text{get PrevX}</math>, <math>Y1 = \text{get PrevY}</math>, <math>X2 = \text{CurrentX}</math>, <math>Y2 = \text{CurrentY}</math>.....</i>	62
Gambar 2. 63	<i>Hasil Akhir Block Aplikasi Digital Doodle.....</i>	62
Gambar 2. 64	<i>Hasil Pengujian ke Ponsel Android.....</i>	62
Gambar 3. 1	<i>Aplikasi Kalkulator Body Mass Index (BMI) .....</i>	64
Gambar 3. 2	<i>masuk ke MIT AI2 menggunakan akun Google....</i>	65
Gambar 3. 3	<i>Membuat Project Baru Aplik.....</i>	66
Gambar 3. 4	<i>Penamaan Project Baru LuasSegitiga .....</i>	66
Gambar 3. 5	<i>Setting properties Height dan Width.....</i>	66
Gambar 3. 6	<i>Menyisipkan label dan Setting properties Textnya</i>	67
Gambar 3. 7	<i>Menyisipkan komponen TextBox lalu Setting propertiesnya .....</i>	67
Gambar 3. 8	<i>Gambar Komponen Desain Aplikasi Luas Segitiga</i>	68
Gambar 3. 9	<i>Pemberian Block Initialize Global name.to ke Viewer.....</i>	70
Gambar 3. 10	<i>Penamaan Blocks Initialize Global name.to dengan Alas.....</i>	70
Gambar 3. 11	<i>Pemberian Nilai Block Initialize Global alas.to dengan decimal 0.....</i>	71
Gambar 3. 12	<i>Penamaan 3 Blocks Initialize Alas, tinggi,luas.....</i>	71
Gambar 3. 13	<i>Pemberian Blocks When BtnHitung.Click ke Viewer.....</i>	72
Gambar 3. 14	<i>Pemberian Nilai Blocks variables Alas di BtnHitung.Click.....</i>	72
Gambar 3. 15	<i>Pemberian Nilai Blocks variables Alas dengan TxtAlas.Text.....</i>	73
Gambar 3. 16	<i>Pemberian Nilai Blocks variables Alas, tinggi dan luas .....</i>	73

Gambar 3. 17	<i>Pemberian Nilai Blocks variables luas dengan rumus.....</i>	74
Gambar 3. 18	<i>Penambahan Blocks Number menjadi 3 komponen</i>	74
Gambar 3. 19	<i>Pemberian rumus Blocks variable luas segitiga ...</i>	75
Gambar 3. 20	<i>Pemberian Nilai Hasil perhitungan ditampilkan di TxtLuas.....</i>	75
Gambar 3. 21	<i>Pemberian Hasil perhitungan ditampilkan di TxtLuas dengan variable luas.....</i>	76
Gambar 3. 22	<i>Blocks keseluruhan tombol Hitung.....</i>	76
Gambar 3. 23	<i>Blocks keseluruhan tombol Clear.....</i>	77
Gambar 3. 24	<i>Blocks keseluruhan Aplikasi Luas Segitiga.....</i>	77
Gambar 3.25	<i>QrCode dan Code yang diberikan untuk pengujian</i>	78
Gambar 3. 25	<i>Aplikasi MIT AI2 di Ponsel android untuk pengujian</i>	78
Gambar 3. 26	<i>Aplikasi MIT AI2 di Ponsel android untuk pengujian dengan scan QR Code atau input Code .....</i>	79
Gambar 3. 27	<i>Gambar Progress pengujian pada layar web MIT AI2 setelah scan QR Code dari Ponsel .....</i>	79
Gambar 3. 28	<i>Hasil Pengujian pada layar ponsel android.....</i>	80
Gambar 3. 29	<i>Aplikasi Menentukan lulus_tidak lulus.....</i>	80
Gambar 3. 30	<i>Masuk ke akun MIT AI2 .....</i>	81
Gambar 3. 31	<i>Memulai Project baru.....</i>	82
Gambar 3. 32	<i>Memberi nama Project .....</i>	82
Gambar 3. 33	<i>Komponen Desain aplikasi yang dibutuhkan.....</i>	82
Gambar 3. 34	<i>Blocks Komponen BtnProses.....</i>	84
Gambar 3. 35	<i>Blocks Komponen If then Else disisipkan ke BtnProses .....</i>	85
Gambar 3. 36	<i>Block Komponen Compare Text disisipkan ke If...</i>	85

Gambar 3. 37	<i>Block Komponen TextBox1.text (Nilai) diberikan kondisi.....</i>	86
Gambar 3. 38	<i>Block Komponen TextBox2.text (Keterangan) diberikan Nilai Lulus berdasarkan kondisi IF &gt;60</i>	86
Gambar 3. 39	<i>Block Komponen TextBox2.text (Keterangan) Nilai "Tidak Lulus" .....</i>	86
Gambar 3. 40	<i>Block Komponen BtnClear (Tombol Clear).....</i>	87
Gambar 3. 41	<i>Block Komponen Keseluruhan Aplikasi.....</i>	87
Gambar 3. 42	<i>Aplikasi Kalkulator BMI .....</i>	88
Gambar 3. 43	<i>Rumus Angka BMI (Body Mass Index).....</i>	88
Gambar 3. 44	<i>Membuat Project baru .....</i>	89
Gambar 3. 45	<i>Masuk ke akun AI2 dengan Akun Google.....</i>	90
Gambar 3. 46	<i>Masuk ke akun AI2 dengan Akun Google.....</i>	90
Gambar 3. 47	<i>Penamaan Project Aplikasi Masuk ke akun AI2 dengan Akun Google.....</i>	90
Gambar 3. 48	<i>Komponen Desain Aplikasi KalkulatorBMI.....</i>	91
Gambar 3. 49	<i>Block Aplikasi KalkulatorBMI.....</i>	93
Gambar 3. 50	<i>Hasil Uji Aplikasi KalkulatorBMI di perangkat android.....</i>	94
Gambar 4. 1	<i>Ekspor File Project .AI2.....</i>	96
Gambar 4. 2	<i>Menyimpan File Project .AIA ke Komputer.....</i>	97
Gambar 4. 3	<i>Meng Import File Project .AIA ke Komputer.....</i>	97
Gambar 4. 4	<i>Memilih File Project .AIA ke akun ai2.....</i>	98
Gambar 4. 5	<i>Mengubah File Project .AIA ke File .apk.....</i>	98
Gambar 4. 6	<i>Progressbar menjadi File .apk.....</i>	99
Gambar 4. 7	<i>pemilihan Download File .apk .....</i>	99

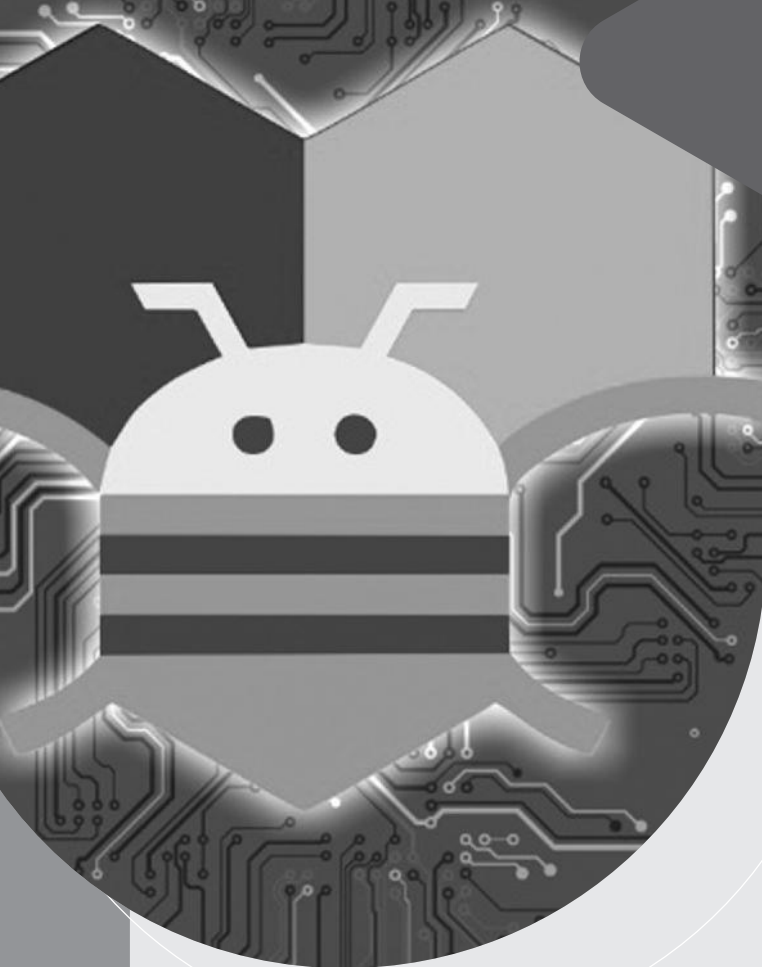


# DAFTAR TABEL

Tabel 3. 1	Desain Komponen dan Setting Properties nya .....	68
Tabel 3. 2	Komponen Desain dan setting propertiesnya.....	83
Tabel 3. 3	Angka BMI dan Statusnya.....	89
Tabel 3. 4	Komponen Desain dan setting propertiesnya.....	91

**BAB I**

.....



# **PENGANTAR MIT APP INVENTOR**

.....

.....



## 1.1. PENDAHULUAN

MIT App Inventor adalah sebuah *tools* pemrograman berbasis *blocks* yang memungkinkan para pemula untuk memulai pemrograman dan membangun aplikasi untuk perangkat *mobile* Android. Block disini adalah kumpulan atau code block berbentuk graphic seperti puzzle, dimana didalamnya terdapat komponen komponen **Logic, Control, Math, Text, Lists, Colors, Variables**, dan **Procedures**. Untuk para Pendatang baru ini dapat mengembangkan dan menjalankan aplikasi pertamanya pada MIT App Inventor dalam waktu kurang dari satu jam, dan dapat memprogram aplikasi yang lebih kompleks dalam waktu yang jauh lebih singkat dibandingkan dengan aplikasi berbasis teks dengan bahasa pemrograman tradisional.

MIT App Inventor berjalan pada layanan Web pada awalnya dikembangkan oleh Profesor Hal Abelson dan tim dari Google *Education* ketika itu profoser Abelson sedang cuti panjang Google Inc. Selanjutnya dikelola oleh staf Pusat Pembelajaran Seluler MIT kolaborasi dengan Massachusetts Institute of Technology (MIT ) program studi Ilmu Komputer dan Artif cial Intelligence Laboratory (CSAIL)

MIT mendukung komunitas di seluruh dunia dengan lebih dari satu juta pengunjung unik setiap bulannya yang berasal dari 195 negara dan telah membuat hampir 30 juta aplikasi pada tahun 2021, MIT App Inventor mengubah cara dunia membuat aplikasi dan cara anak-anak belajar tentang komputasi.

Tim dari MIT App Inventor men *support* lima tujuan utamanya yaitu:

- *Sustaining and enhancing the tool:*  
Komitmen jangka panjang untuk mempertahankan dan meningkatkan MIT App Inventor sebagai layanan gratis mutakhir untuk pengguna akhir. Untuk akhir ini,tim App Inventor terus meningkatkan *tools*, menambahkan fitur baru, debugging dan meningkatkan kinerjanya.

- *Building enterprise enhancements:*  
Bekerja sama dengan instansi pemerintah dan perusahaan swasta untuk mendukung aplikasi unik dengan mengembangkan atau meningkatkan alat f tur khusus MIT App Inventor dalam menanggapi kebutuhan mitra.
- *Building capacity:*  
Berusaha memperluas kapasitas komputasi pendidikan formal dan informal untuk orang dewasa dan remaja di seluruh dunia. Pada pelaksanaannya, terlibat secara aktif dalam mengembangkan dan menyebarkan sumber daya dan materi pelatihan untuk mendukung ketertarikan pada pemrograman di wilayah nya.
- *Promoting computer science education:*  
Berkomitmen untuk menarik perhatian pada ilmu komputer dan pemikiran komputasi di dunia pendidikan. Dari perspektif kebijakan, berpartisipasi secara aktif dalam percakapan lokal dan nasional tentang standar komputasi pendidikan; dari perspektif kesadaran, berpartisipasi dan mendukung skala besar kampanye yang bertujuan untuk meningkatkan kesadaran di antara audiens baru.
- *Conducting and supporting community research:*  
Sarjana dan pascasarjana mahasiswa di MIT dan institusi yang bekerja sama secara aktif terlibat dalam melakukan dan menerbitkan penelitian sambil mengembangkan, menguji, dan mengevaluasi penggunaan Aplikasi MIT App Inventor di seluruh dunia.

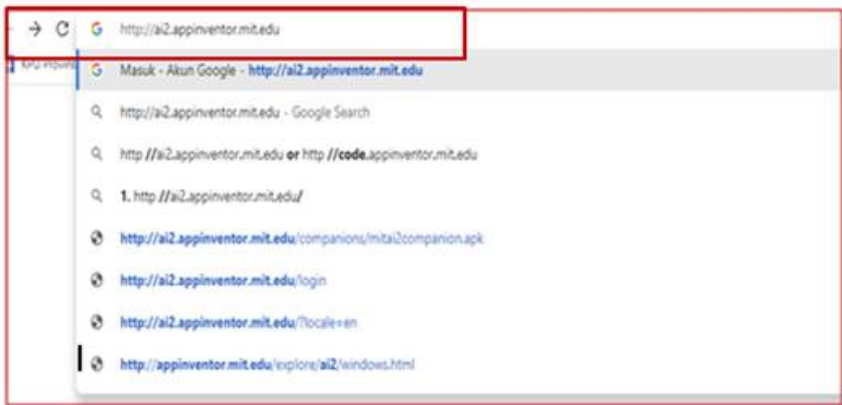
MIT App Inventor Sebagai *tools* yang *open source* berupaya membuat pemrograman dan pembuatan aplikasi yang dapat diakses oleh berbagai khalayak, dan ini telah menarik perhatian mereka diantaranya:

- **Pendidikan formal dan informal** telah menggunakan MIT App Inventor dan memperkenalkan pemrograman berbasis *block* ini untuk para siswa dan mahasiswa Ilmu Komputer, anggota klub sains. Banyak pendidik juga sudah mulai menggunakan MIT App Inventor untuk mengembangkan aplikasi guna mendukung tujuan instruksional mereka sendiri.
- **Pemerintah dan sukarelawan sipil** telah memanfaatkan MIT App Inventor agar dapat mengembangkan aplikasi hiper-lokal dalam menanggapi bencana alam dan kebutuhan berbasis masyarakat.
- **Desainer dan manajer produk** telah melihat potensi MIT App Inventor untuk mendukung proses desain berulang melalui pembuatan prototipe, pengujian, dan iterasi yang cepat.
- **Peneliti** menggunakan MIT App Inventor untuk membuat aplikasi khusus untuk memenuhi data mereka dalam persyaratan pengumpulan dan analisis untuk mendukung penelitian mereka di berbagai bidang kedokteran hingga ilmu sosial.
- **Penghobi** dan pengusaha yang memiliki ide dan ingin mengubah menjadi aplikasi dengan cepat tanpa biaya atau kurva pembelajaran yang dibutuhkan oleh pembuatan aplikasi yang lebih tradisional.

## 1.2 Membuat Akun Aplikasi App Inventor 2 (AI2)

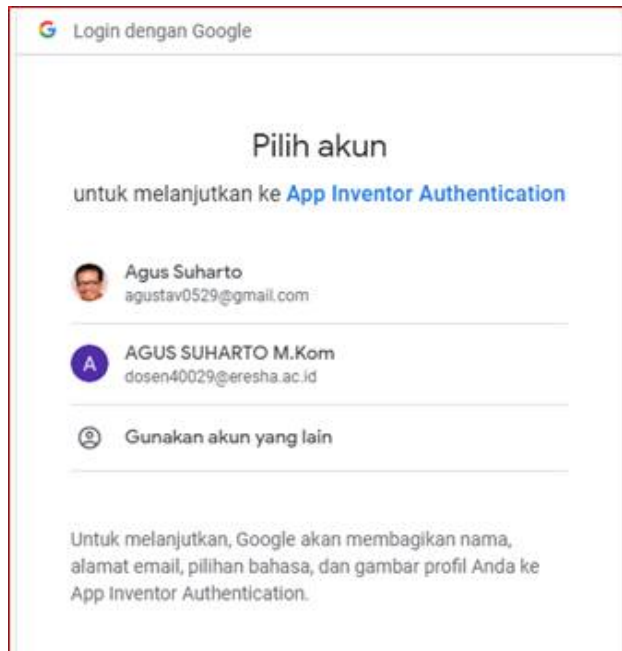
Untuk memulainya, Anda perlu membuat akun App Inventor. App inventor menggunakan Kredensial Google untuk autentikasi, jadi jika anda belum punya akun google anda harus membuat akun Gmail terlebih dahulu.

Jika sudah mempunyai Akun Gmail, kunjungi <http://ai2.appinventor.mit.edu> untuk membuat akun AI2 yang akan kita miliki. Anda bisa Gunakan alamat Gmail dan kata sandi untuk masuk. kemudian nantinya akan diarahkan ke akun Google halaman izin. Untuk mengakses platform AI2.



Gambar 1.1 link halaman <http://ai2.appinventor.mit.edu>

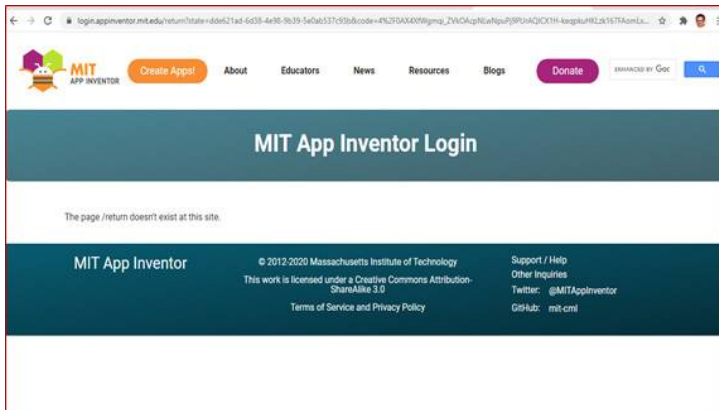
Kemudian nantinya akan diarahkan ke akun Google halaman izin.



Gambar 1.2 pemilihan Akun google untuk akun authentication App Inventor

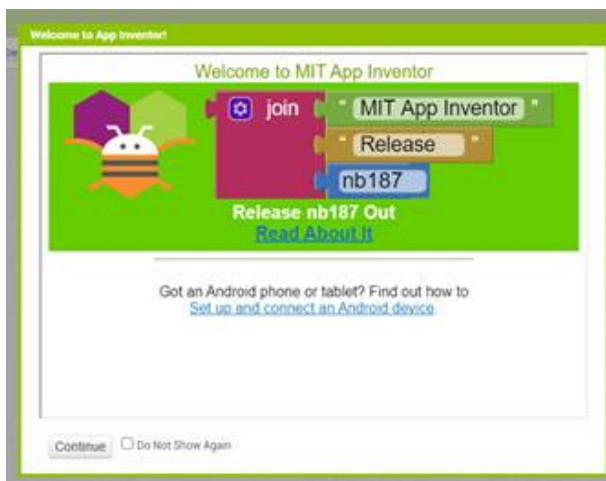
Untuk mengakses platform AI2 Anda perlu mengklik *allow* .

Selanjutnya jika berhasil masuk ke halaman beranda app inventor seperti pada gambar 1.3 dibawah ini,



Gambar 1. 3 Halaman beranda MIT App Inventor.

Untuk membuat project baru dengan AI2, klik tombol *create apps* selanjutnya jika baru pertama kali menggunakan AI2 maka akan muncul tampilan halaman *welcome to app inventor*



Gambar 1. 4 Halaman Welcome MIT App Inventor.

## 1.3 Lingkungan Pengembangan

Pengembangan aplikasi pada AppInventor2 (AI2) terdapat dua tampilan yaitu:

1. Tampilan **Design** (Gambar 1.5) memberikan kita desain visual representasi dari aplikasi yang sedang dibangun.

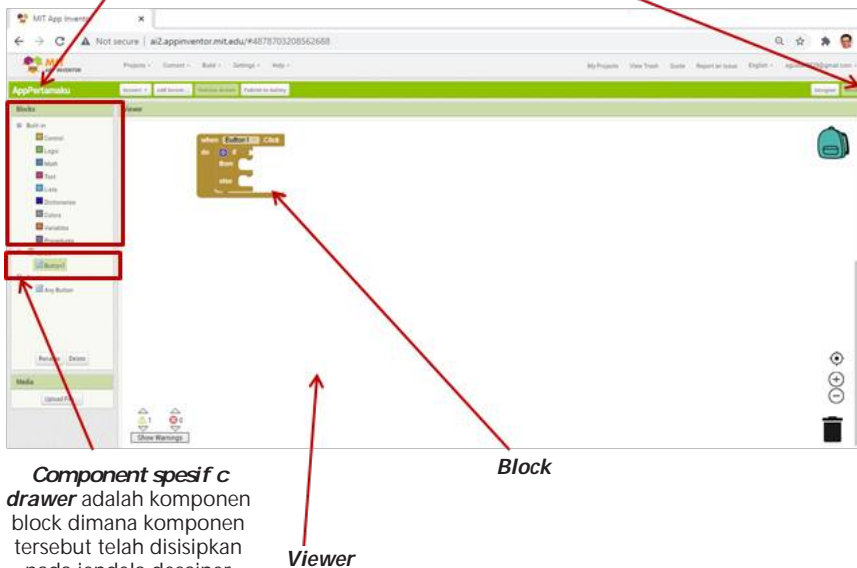
Tampilan **Blocks** (Gambar 1.6) menyediakan ruang untuk membangun logika aplikasi menggunakan berbagai komponen logika yang disebut dengan *Block*. Pada bab selanjutnya kita akan mempelajari bagaimana cara menggunakan *Block* ini dalam sebuah Activity.



Gambar 1.5 tampilan jendela desainer dari AI2

**Built-in drawer** adalah komponen block pada umumnya dimana memungkinkan kita untuk menambah block dengan men drag ke jendela viewer

**Tombol Block** digunakan untuk mengaktifkan dan me non aktif jendela block



Gambar 1. 6 tampilan jendela desainer dari AI2

## 1.4 Setting Perangkat untuk Pengembangan Aplikasi dan Debugging

Persiapan dalam membuat aplikasi dengan App Inventor hanya dalam hitungan menit. Jendela Desainer dan berjalan sepenuhnya pada browser (cloud).

Untuk melihat aplikasi kita pada perangkat saat kita membangunnya (juga disebut pengujian langsung), ada tiga opsi untuk men set pengujian langsung langkah-langkah nya seperti di bawah ini:

**Opsi 1:** Jika Anda menggunakan perangkat android dan memiliki Wif koneksi Internet, Anda dapat mulai membuat aplikasi tanpa mengunduh perangkat lunak apa pun ke komputer. Kita cukup instal **MIT A12 Companion** ke gadget

android melalui play store android opsi ini sangat di rekomendasikan . (Lihat pada bagian **1.4.1.**)

- Opsi 2:** Jika Anda tidak memiliki perangkat Android, Anda harus menginstal perangkat lunak di komputer Anda sehingga Anda dapat menggunakan emulator Android di layar. (Lihat bagian 1.4.2.)
- Opsi 3:** Jika Anda tidak memiliki Wif koneksi Internet, Anda harus menginstal perangkat lunak di komputer Anda sehingga Anda dapat terhubung ke perangkat Android Anda melalui USB. Opsi Koneksi USB bisa jadi rumit, terutama di Windows. Gunakan ini sebagai upaya terakhir. (Lihat bagian 2.4.3.)



Gambar 1. 7 Membangun apps dengan perangkat android dan koneksi Wifi

### 1.4.1 Opsi 1: Membangun Aplikasi dengan Perangkat Android dan Koneksi Wi-Fi

Membuat aplikasi android dengan perangkat Android, dan koneksi Wi-Fi serta PC dan laptop sebagai media desain dan block adalah cara termudah untuk menguji aplikasi Anda (Gambar 1.7).

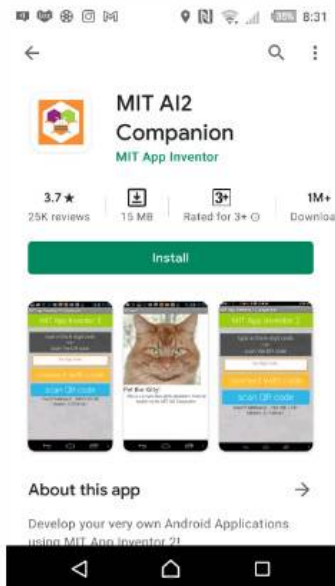
Dengan kombinasi ini, kita dapat menggunakan App Inventor tanpa mengunduh apa pun ke PC/Laptop karena aplikasi yang kita kembangkan berada di situs web AI2: <https://ai2.appinventor.mit.edu>.



Untuk melakukan pengujiannya secara real time ke perangkat Android kita, cukup instal di ponsel atau tablet Android aplikasi **MIT AI2 Companion** pada playstore . berikut Langkah-langkah panduan untuk install ke perangkat ponsel atau tablet android tersebut.

### Langkah 1:

Unduh dan instal melalui playstore MIT AI2 Companion

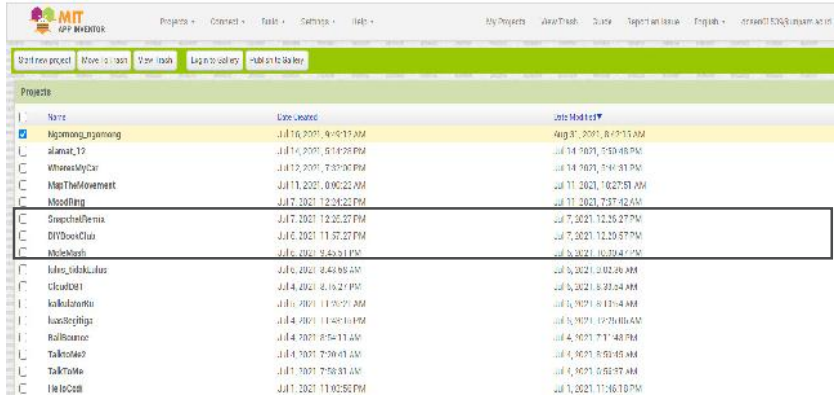


Gambar 1. 8 Aplikasi MIT AI2 Companion pada play store

### Langkah 2:

Hubungkan komputer dan perangkat ponsel ke Jaringan Wi-Fi yang sama

Buka proyek App Inventor yang pernah dibuat atau buat baru melalui **Start New Project**, jika project sudah ada pilih project nya dengan mencheck nama project yang akan di ujikan menggunakan MIT AI2 Companion pada ponsel .



Name	Date Created	Date Modified
Ngomong ngomong	Jul 16, 2021, 9:49:11 AM	Aug 8, 2021, 8:47:15 AM
alamat_12	Jul 16, 2021, 01:42:28 PM	Jul 14, 2021, 7:53:48 PM
WhereasMyCar	Jul 12, 2021, 7:51:00 PM	Jul 14, 2021, 7:41:31 PM
Map The Movement	Jul 11, 2021, 09:02:22 AM	Jul 11, 2021, 1:02:51 AM
Modelling	Jul 7, 2021, 12:26:22 PM	Jul 11, 2021, 7:57:42 AM
SimplestThema	Jul 7, 2021, 12:26:37 PM	Jul 7, 2021, 12:35:27 PM
DIYBoosClas	Jul 6, 2021, 11:57:27 PM	Jul 7, 2021, 12:39:57 PM
Modellari	Jul 6, 2021, 9:43:31 PM	Jul 6, 2021, 10:39:47 PM
Info_nisakLulus	Jul 6, 2021, 8:49:05 AM	Jul 6, 2021, 10:02:36 AM
Cloud001	Jul 4, 2021, 8:19:24 PM	Jul 6, 2021, 8:22:54 AM
Kalkulator	Jul 4, 2021, 11:50:29 AM	Jul 5, 2021, 8:13:04 AM
KasusRefleksi	Jul 4, 2021, 11:48:10 AM	Jul 5, 2021, 12:20:00 AM
RailBounce	Jul 4, 2021, 8:54:11 AM	Jul 4, 2021, 7:11:48 PM
TalktoMe2	Jul 4, 2021, 7:50:41 AM	Jul 4, 2021, 8:53:45 AM
TalkToMe	Jul 1, 2021, 7:58:31 AM	Jul 4, 2021, 8:53:45 AM
HelelCodi	Jul 1, 2021, 11:02:52 PM	Jul 3, 2021, 11:46:10 PM

Gambar 1. 9 Halaman daftar Project Aplikasi MIT AI2

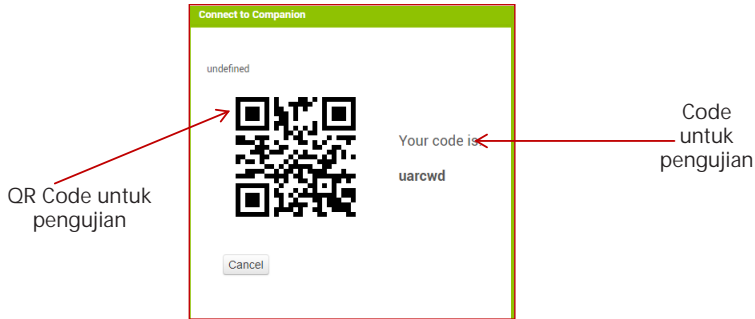
### Langkah 3:

Setelah project terbuka klik menu **Connect, AI Companion**



Gambar 1. 10 Halaman desainer Project Aplikasi MIT AI2

Selanjutnya muncul QR Code dan Kode untuk di scan atau di input pada saat pengujian di ponsel (pilih salah satu)



Gambar 1. 11 QRCode untuk pengujian Aplikasi MIT AI2

#### Langkah 4:

Buka aplikasi **MIT AI2 Companion** yang sudah di install pada ponsel Untuk pengujian nya pada ponsel terdapat 2 cara, yaitu melalui Melalui **Connet With Code** dengan mengetikan kode yang ditampilkan seperti pada gambar 1.9 lalu Melalui **Scan QR Code** .



Gambar 1. 12 tampilan MIT AI2 pada ponsel untuk pengujian

Jika project berjalan baik maka dalam hitungan detik akan muncul hasil uji aplikasi android pada ponsel android kita



Gambar 1. 13 tampilan hasil uji pada ponsel Android

### 1.4.2 Opsi 2: Membangun Aplikasi dengan Emulator

Jika Anda tidak memiliki ponsel atau tablet Android, kita masih dapat menggunakan App Inventor. Dengan menginstall aplikasi emulator.



Gambar 1. 14 *Membuat project pada laptop/Komputer dan menguji pada Emulator*

Untuk menggunakan emulator, kita harus terlebih dahulu menginstal beberapa perangkat lunak di komputer/laptop (ini bukan diperlukan untuk opsi jika punya Wi-Fi). Ikuti petunjuk di bawah ini untuk sistem operasi yang kita miliki

#### **Langkah 1.** Install perangkat lunak pengaturan App Inventor

- **Petunjuk untuk pengguna Mac OS X:**

Untuk mendapatkan emulator Android pada Mac kita, unduh dan

1. instal *setup package*. Yang sudah di download.
2. Klik ganda file yang sudah diunduh untuk memulai penginstalan. (File bernama AppInventor\_Setup\_v\_X.X.dmg di mana XX adalah nomor versi.) Jika Anda mendapatkan pesan bahwa Anda tidak dapat menginstal aplikasi dari pengembang yang tidak dikenal, kontrol-klik tombol ikon aplikasi, lalu pilih Buka dari menu pintasan.



Gambar 1. 15 file setup untuk mac os

3. Klik *continue*
4. Baca petunjuk dan menerima perjanjian lisensi perangkat lunak.
5. Pada layar Instal Standar, klik *Install*. Jangan ubah lokasi pemasangan



Gambar 1. 16 Standar install setup untuk mac os

Jika diminta, masukkan kata sandi Anda untuk mengonfirmasi bahwa Anda benar-benar ingin menginstal perangkat lunak. Klik **Ok**.

6. Penginstalan mengonfirmasi bahwa paket App Inventor Setup telah diinstal.



Gambar 1.17 App inventor setup sukses dan telah di install

- **Petunjuk untuk pengguna Windows**

Menginstal perangkat lunak Windows untuk App Inventor Setup memiliki dua bagian:

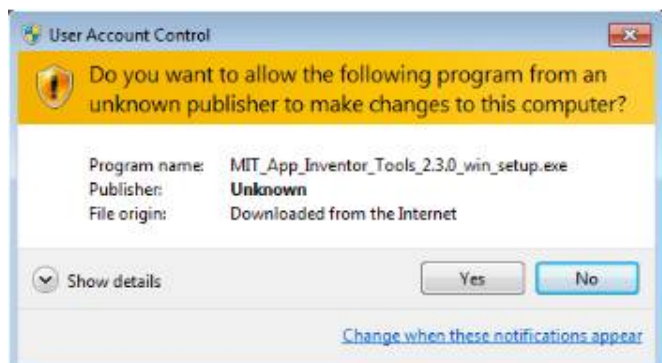
1. Install paket perangkat lunak App Inventor Setup. Langkah ini sama untuk semua perangkat Android, dan sama untuk Windows XP, Vista, Windows 7, 8.1, dan 10.
2. Jika kita memilih untuk menggunakan kabel USB untuk menghubungkan ke perangkat, maka kita harus menginstal driver Windows untuk ponsel Android Anda.

*CATATAN: AI2 tidak berfungsi dengan Internet Explorer. Untuk pengguna Windows, sebaiknya gunakan Chrome atau Firefox sebagai browser Anda untuk digunakan dengan App Inventor.*

Menginstal paket perangkat lunak pada Setting AI2 harus melakukan penginstalan dari akun yang memiliki hak administrator. Untuk menginstal melalui akun non-administrator saat ini belum didukung.

Jika kita telah menginstal versi sebelumnya dari alat penyiapan AI2, Kita harus menghapus instalannya sebelum menginstal versi terbaru. Ikuti petunjuk pada cara Memperbarui Perangkat Lunak *setup* App Inventor.

1. Download file penginstalan .
2. Temukan file MIT\_Appinventor\_Tools\_2.3.0 (~ 80 MB) di file Unduhan atau Desktop Anda. Lokasi unduhan di komputer Anda tergantung pada bagaimana browser Anda dikonfigurasi.
3. Buka file nya.
4. Klik melalui langkah-langkah penginstal. Jangan ubah lokasi penginstalan tetapi catat direktori penginstalan, karena kita mungkin memerlukannya untuk memeriksa driver nanti. Direktori akan berbeda tergantung pada versi Windows Anda dan apakah Anda masuk sebagai administrator atau tidak.
5. Kita mungkin ditanya apakah ingin mengizinkan program dari penerbit yang tidak dikenal, untuk membuat perubahan pada perangkat komputer/laptop ini. Klik yes.



Gambar 1. 18 Pesan konfirmasi install MIT AI2



- **Petunjuk ins all untuk pengguna GNU/Linux:**

Anda memerlukan hak sudo untuk melakukan instalasi.

Catatan: Program pengaturan adalah perangkat lunak 32-bit. Jika kita memiliki sistem 64-bit, Anda mungkin perlu menginstal *library* agar mesin kita menjalankan perangkat lunak 32-bit. Salah satu cara untuk melakukannya adalah dengan menjalankan perintah `Sudo apt-get install lib32z1`, tetapi ini mungkin tidak bekerja pada semua distribusi GNU/Linux, dan Anda mungkin perlu melakukan penyelidikan untuk sistem khusus Anda. Jika sebelumnya Anda telah menginstal perangkat lunak pengaturan App Inventor, Anda harus menghapus file tersebut sebelum menginstal perangkat lunak baru:

**`sudo rm-rf /usr/google/appinventor`**

**`sudo rm-rf ~/.appinventor`**

### **Ins ruksi untuk sis em yang dapat mengins al paket Debian**

Gunakan instruksi ini untuk sistem yang dapat menginstal paket Debian (mis. Debian atau Ubuntu):

*Catatan: Jika sebelumnya Anda menginstal paket penyiapan untuk App Inventor Classic, Anda harus menghapusnya, karena dapat mengganggu penginstalan baru. Hapus paket dengan `Sudo apt-get remove appinventor-setup`.*

1. Unduh paket Debian penginstal Appinventor Setup. File bernama `appinventor2-setup_2.3_all.deb`. Ini adalah file penginstal paket Debian. Tempat itu akan berakhir di komputer kita tergantung pada bagaimana browser Anda dikonfigurasi. Biasanya, akan masuk ke folder Unduhan Anda.
2. Jika sistem Anda dapat menginstal paket hanya dengan mengklik file paket, lakukan itu.

3. Jika sistem Anda tidak mendukung penginstal paket yang dapat diklik, navigasikan ke direktori tempat file berada dan jalankan perintah:

**`sudo dpkg--install appinventor2-setup_2.3_all.deb`**

Dengan salah satu metode tersebut, Anda mungkin perlu memastikan bahwa file deb serta direktori yang ada di dalamnya dapat dibaca dunia dan dieksekusi dunia. Pada beberapa sistem, sudo tidak memiliki hak default untuk membaca dan mengeksekusi semua file.

4. Perangkat lunak akan diinstal di bawah/usr/google/appinventor.
5. Kita mungkin juga perlu mengonfigurasi sistem untuk mendeteksi perangkat kita. Lihat petunjuk pengembang Android pada *setup* . Ikuti instruksi di bawah langkah "siapkan sistem Anda untuk mendeteksi perangkat Anda" dalam butir di bawah "Jika Anda mengembangkan di Ubuntu Linux".

### **Instruksi Untuk Sistem GNU/Linux Lainnya**

1. Download file kompresi tar Appinventor file Setup. file Ini bernama appinventor2-setup\_2.3.tar.gz. yang terkompresi Gzip.
2. Instal file menggunakan metode yang sesuai dengan sistem operasi Anda. Anda harus memeriksa apakah direktori commands-for-Appinventor berada di bawah/usr/google/appinventor.

### **Memulai aiStarter**

Program aiStarter mengelola komunikasi antara browser Web dan perangkat Android. Itu harus berjalan setiap kali orang menggunakan emulator atau kabel USB; ini tidak perlu dijalankan ketika menggunakan pendamping *wireless*. Setiap kali masuk untuk menggunakan App Inventor dengan emulator atau USB, harus memulai aiStarter. perintah Ini dapat

dilakukan dengan `usr/google/appinventor/commands-for-Appinventor/aiStarter`, Anda mungkin ingin mengatur agar perintah ini dijalankan secara otomatis setiap kali ketika masuk, atau ketika sistem dimulai. Cara yang tepat untuk melakukan ini tergantung pada distribusi GNU/Linux yang Anda gunakan. Lihat dokumentasi untuk distribusi Anda.

### Menemukan direktori Setup

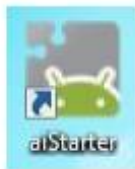
Dalam kebanyakan kasus, App Inventor harus dapat menemukan sendiri perangkat lunak Setup yang diinstal. Jika memang menanyakan di mana perangkat lunak itu berada, jalur direktori yang harus Anda masukkan adalah `usr/google/appinventor/commands-for-Appinventor`.

### Langkah 2. Meluncurkan aiStarter (hanya Windows dan GNU/Linux)

Menggunakan emulator atau kabel USB memerlukan penggunaan program bernama aiStarter. program ini mengizinkan browser untuk berkomunikasi dengan emulator atau kabel USB.

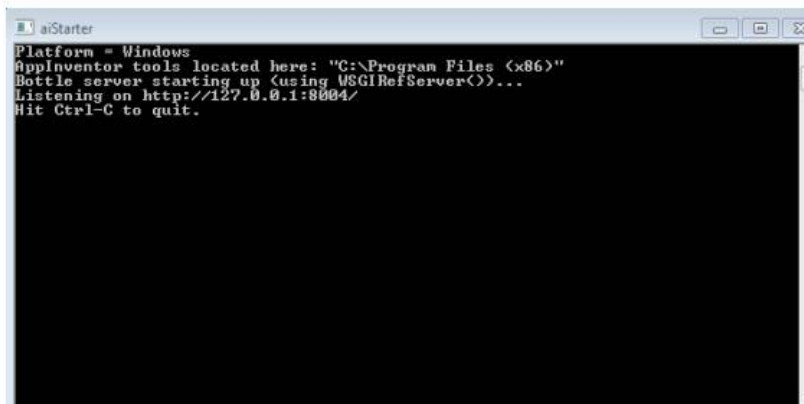
Pada Mac, aiStarter akan mulai secara otomatis saat kita masuk ke akun kita dan akan berjalan tidak terlihat di latar belakang.

Dengan Windows, akan ada pintasan ke aiStarter dari Desktop Anda, dari Start menu, dari Semua Program atau dari Folder Startup. Jika Anda ingin menggunakan emulator dengan App Inventor, Anda harus meluncurkan aiStarter secara manual di komputer Anda saat Anda masukin. Anda dapat memulai aiStarter dengan mengklik ikon (Gambar 1.8) di Desktop atau dari menu.



Gambar 1. 19 Ikon AI Start Windows

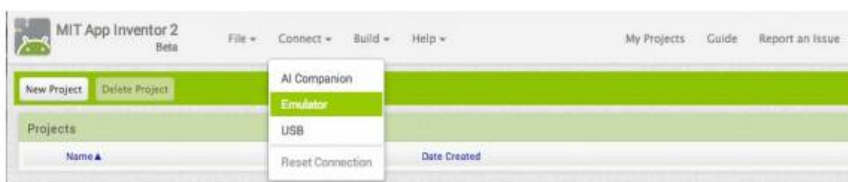
Jika berhasil maka jendela yang akan tampil adalah seperti dibawah ini



Gambar 1. 20 Tampilan AIStart

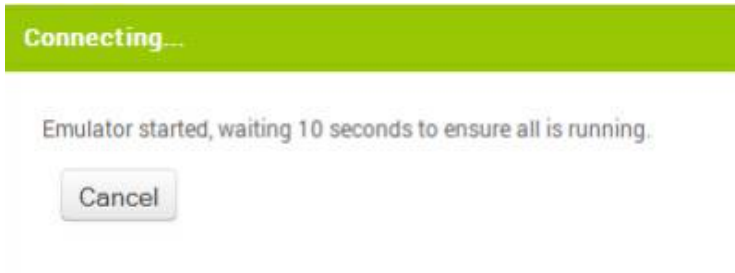
**Langkah 3.** Membuka proyek AI2 dan dan menghubungkan ke emulator

Buka App Inventor dan buka proyek (atau buat yang baru: Project > New Project dan beri nama proyek Anda). Kemudian, dari menu App Inventor (pada App Inventor perangkat lunak berbasis cloud di <http://ai2.appinventor.mit.edu>), buka menu Connect dan klik opsi Emulator seperti yang ditunjukkan pada Gambar 1.19.



Gambar 1. 21 Koneksi Emulator dari AI2

Kita akan mendapatkan pemberitahuan yang mengatakan bahwa emulator terhubung. Untuk memulai emulator dapat memakan waktu yang cukup lama . Kita mungkin saja melihat layar pembaruan seperti Gambar 1.20 selama start-up.



Gambar 1. 22 Tampilan jendela selama proses start-up

Empat fase start-up emulator ditunjukkan pada Gambar 1.21. Emulator awalnya akan muncul dengan layar hitam kosong (1).

Emulator siap ketika Anda melihat warna latar belakang (2).

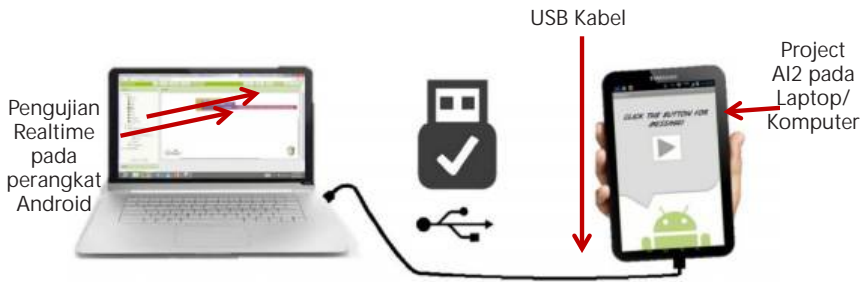
Setelah latar belakang muncul, tunggu hingga ponsel yang diemulasi selesai menyiapkan kartu SD-nya: akan ada pemberitahuan di bagian atas layar ponsel saat kartunya sedang disiapkan. Setelah terhubung, emulator akan meluncurkan dan menampilkan aplikasi yang telah kita buka di App Inventor.



Gambar 1. 23 Tampilan empat fase emulator mulai dari Start-Up

### 1.4.3 Opsi 3 Membangun Aplikasi dengan Perangkat Android dan Kabel USB

Beberapa firewall sekolah dan organisasi kemungkinan diblokir jenis koneksi Wi-Fi nya . Sehingga untuk membangun aplikasi dengan AI2, dapat menggunakan kabel USB seperti yang ditunjukkan pada Gambar 1.22



Gambar 1. 24 Membangun aplikasi menggunakan kabel USB

Saat Anda menggunakan App Inventor dengan ponsel atau tablet, perangkat kita berkomunikasi dengan Perangkat lunak App Inventor yang berjalan di jendela browser. Komunikasi ini dikelola oleh Aplikasi AI2 companion yang berjalan di perangkat android seperti pada sebelumnya.

Berikut langkah-langkah menggunakan App Inventor dengan kabel USB.

### **Langkah 1:** Unduh dan instal perangkat lunak pengaturan App Inventor

Untuk terhubung dengan USB, Anda harus terlebih dahulu menginstal perangkat lunak pengaturan App Inventor di komputer/laptop.

- Petunjuk untuk Mac OS X: <http://appinventor.mit.edu/explore/ai2/mac.html>
- Petunjuk untuk Windows: <http://appinventor.mit.edu/explore/ai2/windows.html>
- Petunjuk untuk GNU/Linux: <http://appinventor.mit.edu/explore/ai2/linux.html>

(Pada langkah ini sama dengan langkah 1 pada bagian 1.4.2)

**Langkah 2:** Unduh dan instal MIT AI2 App Companion di ponsel Anda pada play store  
(Pada Langkah ini sama dengan langkah 1 pada bagian 1.4.1)

**Langkah 3.** Luncurkan aiStarter (hanya Windows & GNU/Linux).  
(Pada Langkah ini sama dengan langkah 2 pada bagian 1.4.1).

**Langkah 4:** Siapkan perangkat Anda untuk USB (aktifkan USB Debugging) di perangkat Android Anda, buka Pengaturan sistem/Setting lalu pilih Developer options/Opsi pengembang dan aktifkan developer option/Opsi pengembang, pastikan USB debugging diizinkan.

Pada sebagian besar perangkat yang menjalankan Android 3.2 atau lebih lama, Anda dapat menemukan opsi ini di bawah **setting**/Pengaturan > **application**/Aplikasi > **developer**/Pengembangan.

Pada Android 4.0 dan yang lebih baru, itu disembunyikan secara default. Untuk membuatnya, buka **Setting** > **About phone** > dan ketuk **Build Number** tujuh kali.

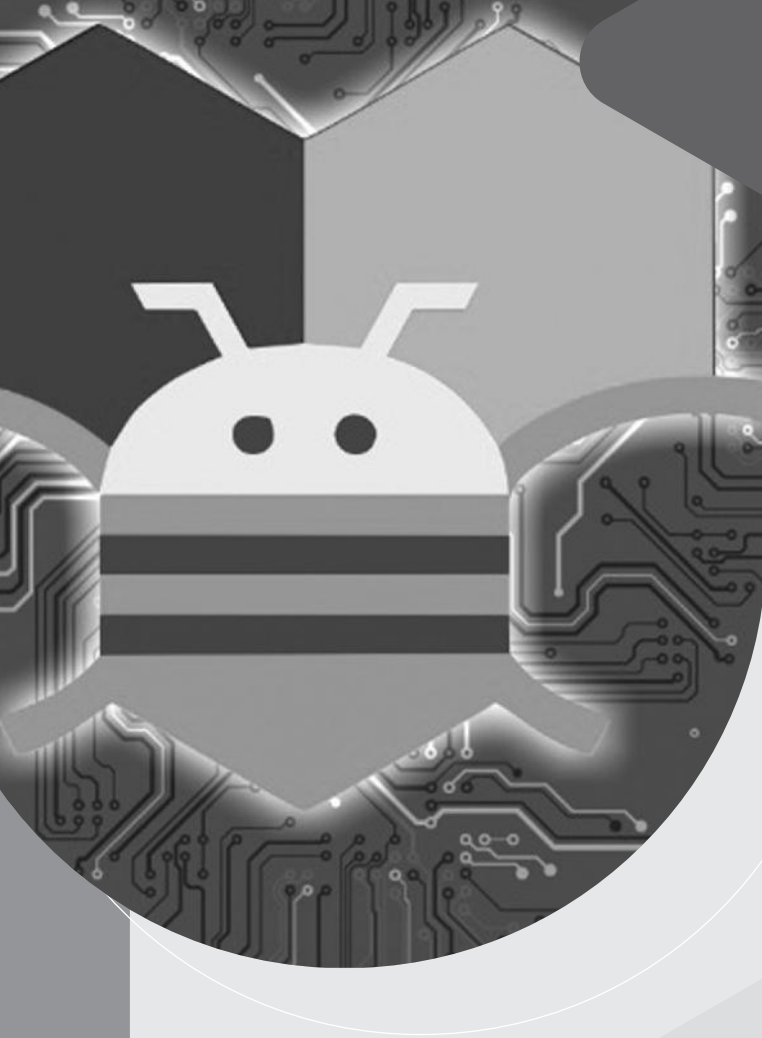
Kembali ke layar sebelumnya dan kita akan menemukan menu **developer option**/opsi Pengembang, termasuk **USB Debugging**.

**Langkah 5:** Hubungkan komputer/laptop dan perangkat Anda, dan autentikasi jika perlu

Hubungkan perangkat Android Anda ke komputer menggunakan kabel USB. Pastikan perangkat terhubung sebagai perangkat penyimpanan massal (bukan perangkat media) Anda mungkin harus membuka Finder (di Mac) atau My Computer(di Windows) dan lepaskan drive apa pun yang terpasang saat Anda menghubungkan perangkat Android.

## BAB II

.....



# TUTORIAL MEMBUAT APLIKASI MIT APP INVENTOR (PEMULA)

.....

.....



App Inventor adalah layanan berbasis cloud gratis yang memungkinkan kita membuat aplikasi seluler sendiri menggunakan bahasa pemrograman berbasis *block*. Kita bisa mengakses App Inventor menggunakan browser web (Chrome, Firefox, Safari).

Tutorial membuat aplikasi pemula ini, akan mempelajari dasar-dasar aplikasi pemrograman untuk perangkat Android.

Untuk menjalankan aplikasi kita (rekomendasi adalah pada bab 1 bagian 1.4.1 opsi pertama yaitu Membangun Aplikasi dengan Perangkat Android dan Koneksi Wi-Fi

Yang kita perlukan adalah:

1. Komputer Mac atau Windows (lihat persyaratan sistem)
2. Koneksi internet/wi-f
3. Perangkat seluler Android (ponsel atau tablet) untuk pengujian
4. Aplikasi AI2 companion yang sudah ter install pada perangkat pengujian

## 2.1 MEMBANGUN APLIKASI PERTAMA: HELLO CODI!

Setelah kita menyiapkan komputer atau laptop serta perangkat android, dan setelah mempelajari cara kerja desainer dan Editor block, aplikasi pertama kita adalah membangun aplikasi HelloCodi.

**Hello Codi** adalah aplikasi sederhana yang dapat kita buat dalam waktu yang sangat singkat. Kita membuat tombol dimana terdapat image/gambar lebah Codi pada tombol tersebut, lalu kita memprogramnya dengan *block* sehingga ketika tombol image lebah codi diklik/ditekan (Touch The Screen) ponsel akan mengeluarkan suara mendengung.



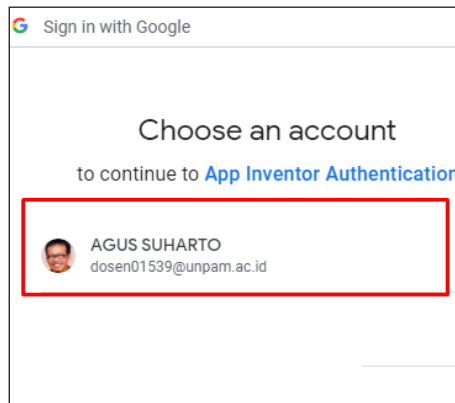
Gambar 2.1 Aplikasi Hello Codi

### 2.1.1 Langkah untuk memulai membangun aplikasi Hello Codi!,:

1. Masuk ke web mit app inventor pada browser ketik::<https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik *Create Apps*

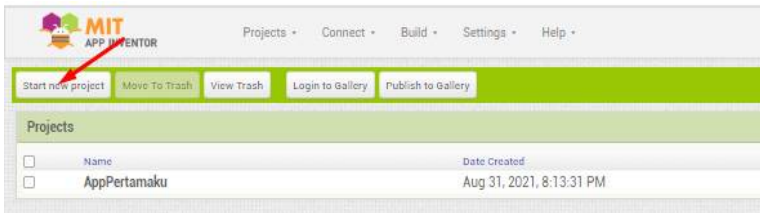


3. Selanjutnya pilih akun google untuk masuk sebagai akun mit AI2



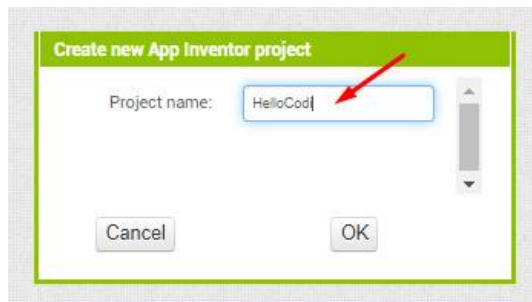
Gambar 2.2 pemilihan akun google sebagai akun MIT AI2

4. Pada halaman *project* klik *start new project* untuk membuat project baru



Gambar 2. 3 Memulai project MIT AI2

5. Pada jendela penamaan *project* beri nama *projectnya* dengan aturan penamaan file, tidak boleh ada spasi jika terdapat 2 suku kata, tidak boleh ada karakter simbol, tidak boleh didahului angka pada awal nama.



Gambar 2. 4 Penamaan Project MIT AI2

Untuk aplikasi pertama kita beri nama **HelloCodi**,selanjutnya klik **OK**

6. Pada aplikasi pertama kita memerlukan file gambar *Codi the bee*. Unduh file-file ini ke komputer/laptop sumber gambar bisa kita download dengan mengklik tautan berikut:  
<http://appinventor.mit.edu/explore/sites/all/files/ai2tutorials/helloCodi/codi.jpg>

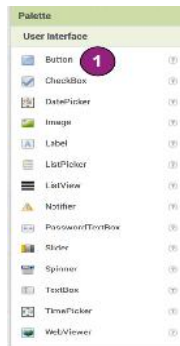
Lalu suara lebah codi dapat diunduh pada tautan dibawah ini:  
<http://appinventor.mit.edu/explore/sites/all/files/ai2tutorials/helloCodi/codi.jpg>

- Gambar codi: codi.jpg (Klik kanan dan Simpan)
- Suara lebah: Suara Lebah.mp3 (Klik kanan dan Simpan) setelah mengeklik tautan, klik kanan pada gambar dan pilih "Simpan Sebagai (Save AS)". Simpan file ke desktop atau folder unduhan Anda, atau di mana pun Anda dapat menemukannya dengan mudah nanti.

### 2.1.2 Komponen untuk Desain aplikasi Hello Codi!

Komponen App Inventor terletak di sisi kiri Jendela Desainer di bawah judul *Palette*. Komponen adalah elemen dasar yang kita gunakan untuk membuat aplikasi di ponsel Android. . contoh komponen seperti komponen **Label**, yang hanya menampilkan teks di layar, atau komponen **Button**/Tombol (kiri #1) yang Anda ketuk untuk memulai tindakan.

Komponen lain **canvas** yang dapat menampung gambar statis atau animasi, sensor Accelerometer yang berfungsi seperti pengontrol Wii dan mendeteksi saat Anda menggerakkan atau menggoyangkan ponsel, komponen yang mengirim pesan teks, komponen yang memutar musik dan video, komponen yang mendapatkan informasi dari situs Web, dan sebagainya.



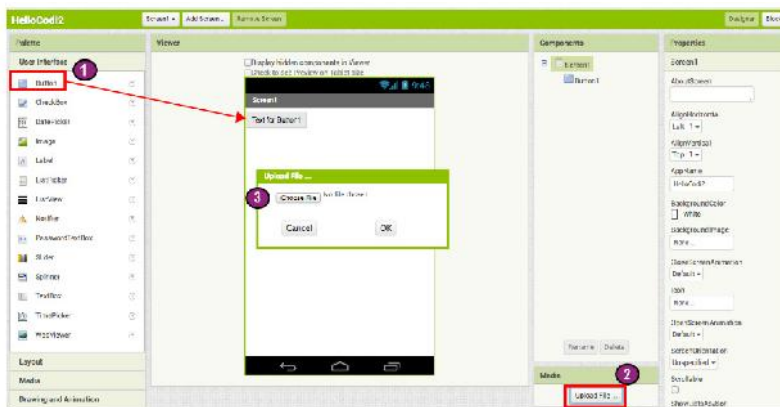
Gambar 2. 5 Komponen komponen pada Palette

## Langkah-langkah untuk memilih komponen aplikasi *Hello Codi* dan mengatur properti

HelloCodi membutuhkan komponen Button yang menampilkan gambar lebah yang kita download sebelumnya. Langkahnya:

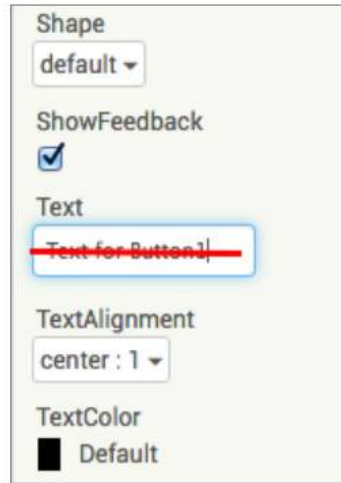
**Langkah 1a.** Dari palet Antarmuka/UI Pengguna, seret dan lepas komponen Tombol/Button ke Screen1(#1).

**Langkah 1b.** Untuk memberi tombol gambar lebah, di panel Properties, di bawah Gambar, klik teks "None..." dan klik "**Upload File...**" (#2). Sebuah jendela akan muncul untuk memilih file gambar. Klik "**Browse**" dan kemudian arahkan ke lokasi file **codi.jpg** yang Anda unduh sebelumnya (#3). Klik file **codi.jpg**, klik "**Open**", lalu klik "**OK**".



Gambar 2. 6 Pemilihan Komponen untuk aplikasi Hello Codi!

**Langkah 2.** Ubah Tombol/Button pada properti Teks: Hapus "Teks untuk Button1", biarkan properti teks Tombol kosong sehingga tidak ada tulisan di atas gambar lebah.



Gambar 2.7 Pengaturan *properties Text* pada Komponen Button

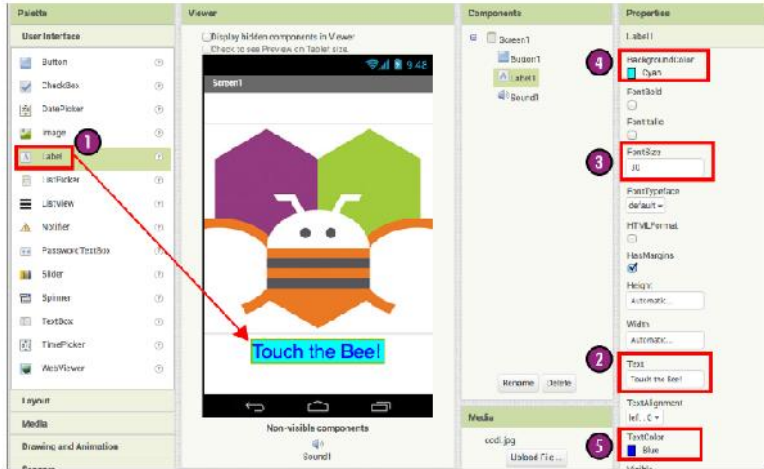
### Langkah 3.

Dari *pallet User Interface*, seret dan lepas komponen Label ke **viewer** (#1), letakkan di bawah gambar lebah. Ini akan muncul di bawah daftar komponen kita sebagai Label1.

Di bawah panel Properties, ubah:

- [2] set *properties* Teks Label1 "***Touch the Bee***". akan melihat perubahan teks di Designer dan di perangkat Anda.
- [3] set *properties* FontSize label1 Ukuran menjadi 30.
- [4] set BackgroundColor dari Label1 dengan mengklik kotak. Lalu kita dapat mengubahnya ke warna apa pun yang kita suka.
- [5] set *properties* TextColor label1 ke warna apa pun yang kita suka. (Catatan: jika BackgroundColor dan TextColor sama, kita tidak akan dapat membaca teks!)

Di sini, warna latar belakang diatur ke aqua dan warna teks diatur ke biru.



Gambar 2.8 Pengaturan properties Text, Font Size, Background Color, TextColor

#### Langkah 4.

Di bawah Palet, klik pada **Media Drawer** dan seret komponen **Sound** dan letakkan di Viewer (#1). Di mana pun kita melepaskannya, akan muncul di area di bagian bawah Viewer yang ditandai Komponen tidak terlihat.

Di bawah panel Media, Klik **Upload File...** (#2) Telusuri ke lokasi file **Bee-Sound.mp3** yang kita download sebelumnya dan upload ke project ini (#3).

Di bawah panel Properties, lihat bahwa properti Source saat ini None.... Klik kata None... untuk mengubah Source komponen Sound1 menjadi Bee-Sound.mp3 (#4).



Gambar 2. 9 Penyisipan komponen Sound ke jendela Viewer

### 2.1.3 Pemrograman dengan *Block Editor*

Sejauh ini kita telah mengatur layar dan komponen aplikasi kita pada *Designer*, yang ada di jendela browser web. Untuk mulai memprogram aplikasi, kita harus membuka **Block Editor**. Klik tombol **Block** di kanan atas layar .



Gambar 2. 10 Mengaktifkan jendela Block Editor

Setelah Editor **Block** tampil langkah berikutnya kita akan memprogram aplikasi kita dengan *block*.

#### Memaikan Suara

##### Langkah 1.

Di sisi kiri **Block Editor**, klik komponen Button1 untuk membukanya. Seret dan lepas blok **When Button1.Click** di area kerja (area terbuka di sebelah kanan).





Gambar 2. 11 Pemilihan Block When Button1.Click

Block kuning disebut **block event handler**. **Block event handler** menentukan bagaimana perangkat seluler harus merespons kejadian tertentu: tombol telah ditekan, telepon diguncang, pengguna menyeret jarinya di atas kanvas, dll. ketika Button1.

### Langkah 2a.

Klik **drawer Sound1** dan seret **blok Sound1.Play** dan hubungkan ke bagian “do” dari block **When Button1.Click**. Block terhubung bersama seperti potongan puzzle dan kita dapat mendengar suara klik saat terhubung.



Gambar 2. 12 Penyisipan Blocks call sound.play ke dalam Block When Button1.Click

**Block** ungu disebut *block* perintah, yang ditempatkan di badan *event handler*. Ketika *event handler* dieksekusi, *Block* ini menjalankan urutan perintah di tubuhnya dan menentukan tindakan yang akan dilakukan (misalnya, memutar suara, menekan Button1)

**Block** kita akan terlihat seperti pada gambar dibawah ini:



Gambar 2. 13 Hasil penyisipan Blocks call sound.play ke dalam Block When Button1.Click

Sekarang kita dapat melihat *block* perintah yang ada di **event handler**. Kumpulan block ini berarti; "Ketika Button1 di klik ", Sound1 akan diputar."

#### 2.1.4 Menguji aplikasi helloCodi yang sudah dibangun

Setelah mendesain dan memberikan *code block* pada aplikasi, selanjutnya melakukan pengujian di perangkat android baik ponsel atau tablet kita.

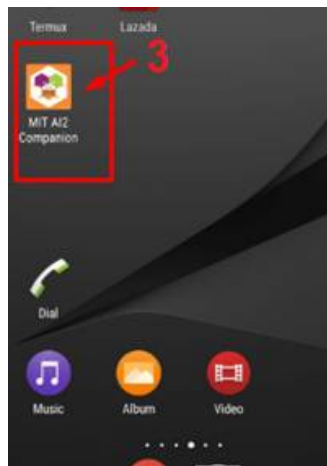
Langkah:

1. Pada jendela project baik *designer* maupun *block*, Klik menu **Connect–AI Companion (#1)**
2. Selanjutnya kita akan mendapatkan **QRCode dan Code** yang akan digunakan untuk dimasukan pada ponsel/tablet pengujian



Gambar 2.14 Meng koneksi aplikasi ke perangkat android

3. Buka aplikasi MIT AI2 Companion pada Ponsel/Tablet android yang digunakan sebagai media pengujian aplikasi



Gambar 2.15 aplikasi MIT AI2 android sebagai pengujian

4. Selanjutnya MIT AI2 Companion akan meminta isian kode QR Code yang didapatkan pada langkah 2, bisa kita ketikan atau scan QRCode tersebut sehingga otomatis code akan muncul pada kotak isian



Gambar 2. 16 aplikasi MIT AI2 android menscan QRCode dari project

5. Selanjutnya pada project ada tampilan progress dan akan mengirimkan hasil pengujian ke perangkat android kita:



Gambar 2. 17 Pogress uji aplikasi yang akan ditampilkan ke perangkat android

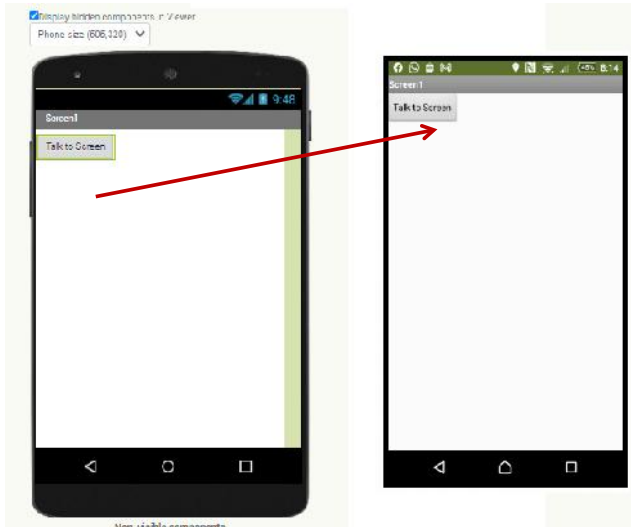
6. Jika tidak ada masalah maka hasil akhir pengujian ditampilkan di perangkat ponsel android.



Gambar 2. 18 Hasil uji aplikasi ditampilkan ke perangkat android

Lakukan pengujian selanjutnya dengan menyentuh gambar lebah codi, jika mendengar suara mendengung Selamat aplikasi pertama anda sukses dan berhasil

## 2.2 MEMBANGUN APLIKASI KEDUA: *TALK TO ME 1*



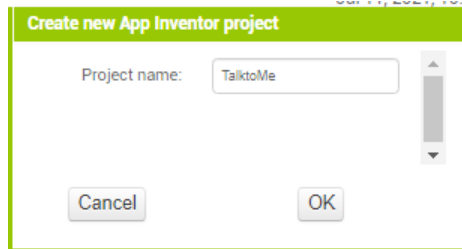
Gambar 2. 19 Aplikasi Talk To Me 1

Aplikasi kedua kita adalah Talk To me, aplikasi sederhana ini terdapat Tombol/Button dan sound, tujuan aplikasi ini ketika aplikasi berjalan di perangkat android Tombol di tekan akan memunculkan pesan suara sesuai *message* yang diketikan pada **block**

### 2.2.1 Langkah untuk memulai aplikasi *TalktoMe*:

Lakukan langkah 1,2,3,4 untuk membuat project baru seperti pada bagian **2.1.1 Membangun aplikasi Hello Codi!**, tetapi jika jendela project masih terbuka lakukan langkah 4 saja yaitu **Start New Project** .

5. Pada Jendela penamaan *project name* ketikan: TalkToMe seperti contoh pada gambar dibawah



Gambar 2. 20 Penamaan project Aplikasi Talk To Me

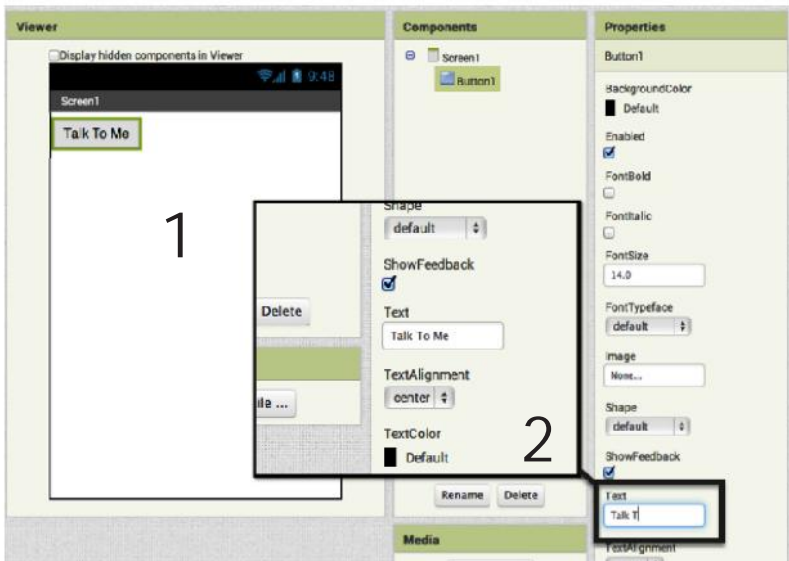
## 2.2.2 Komponen untuk Desain aplikasi TalkToMe

### Langkah 1:

Dari Pallette seret dan lepaskan Tombol/Button ke Screen1

### Langkah 2:

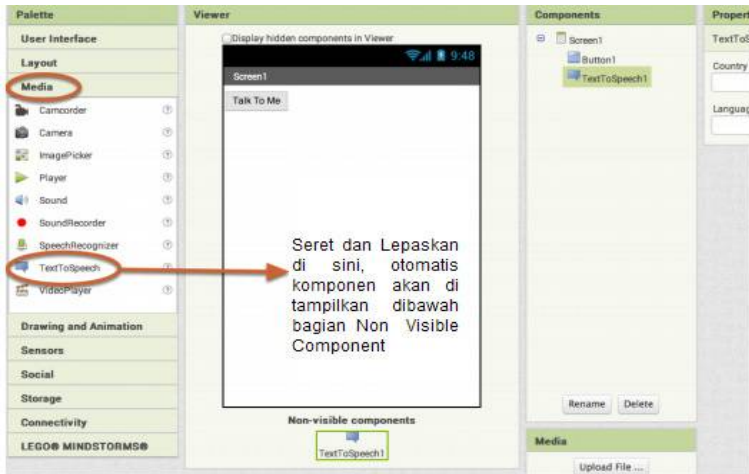
Pada panel properti, ubah teks untuk Tombol. di bawah properti Text, ketik "Talk To Me". Perhatikan bahwa teks pada tombol aplikasi apakah sudah berubah.



Gambar 2. 21 Set Properties Text Aplikasi Talk To Me

### Langkah 3 :

Buka Media pada **Palette** pilih komponen **TextToSpeech**. Seret dan lepaskan (*Drag*) ke Viewer. Perhatikan bahwa komponen tersebut di bawah "**Non-Visible Component**" karena komponen tersebut tidak ditampilkan di antarmuka pengguna aplikasi. Tetapi komponen tersebut tersedia dan digunakan untuk aplikasi.



Gambar 2. 22 menempatkan pallet Media TextToSpeech

### 2.2.3 Pemrograman dengan Block Editor

Saatnya memberikan program untuk aplikasi kita dengan *block*, Klik tombol "**Block**" untuk pindah ke Editor *Block*.



Gambar 2. 23 pemilihan jendela blocks untuk pemrograman

Setelah Editor **Block** tampil berikutnya adalah:

### Langkah 1:

Pilih pada Button1 drawer(#1). lalu Klik dan tahan “**when Button1.Click do (#2)**” Seret ke Viewer (#3) dan letakkan di sana. Blok ini akan digunakan ketika tombol di klik pada aplikasi Ini disebut “**Event Handler**”



Gambar 2. 24 pemilihan blocks When Button1.click

### Langkah 2:

Klik pada Drawer TextToSpeech(#1). Klik dan tahan **call TextToSpeech1.Speak** (#2) blok. Seret ke Viewer dan letakkan di sana(#3). Ini adalah blok yang akan memunculkan pesan suara. Karena berada di dalam **Button.Click**, dan akan dijalankan ketika tombol di aplikasi kita di klik.



Gambar 2. 25 pemilihan blocks Call TextToSpeech1.speak



### Langkah 3:

Selanjutnya kita akan memberikan blok **TextToSpeech.Speak** pesan suara apa yang ditampilkan. Klik pada **drawer Text**, seret blok teks dan sisipkan ke soket berlabel **"message"**.



Gambar 2. 26 pemilihan blocks Text untuk ditempatkan pada slot message

### Langkah 4:

Klik pada **block Text** dan ketik " **Congratulations! You've made your first app.** "

(Atau ketikan kalimat apapun yang kita inginkan)



Gambar 2. 27 Pengisian Pesan Text "Congratulation You've made your first app"

## 2.2.4 Menguji aplikasi TalkToMe yang sudah dibangun:

Lakukan pengujian ke perangkat ponsel/tab android kita ikuti langkah 3 pada bagian **Menguji aplikasi HelloCodi** sebelumnya. Jika tidak ada masalah maka hasil akhir pengujian ditampilkan di perangkat ponsel atau android



Gambar 2. 28 Hasil pengujian ke perangkat android

Sentuh/klik pada tombol "***Talk to Screen***" dan jika terdengar pesan suara "***Congratulations! You've made your first app***", Selamat aplikasi anda berhasil.

## 2.3 MEMBANGUN APLIKASI KETIGA: *TALK TO ME 2, USER INPUT & ACCELOROMETER SENSOR*



Gambar 2. 29 Aplikasi Talk To Me 2

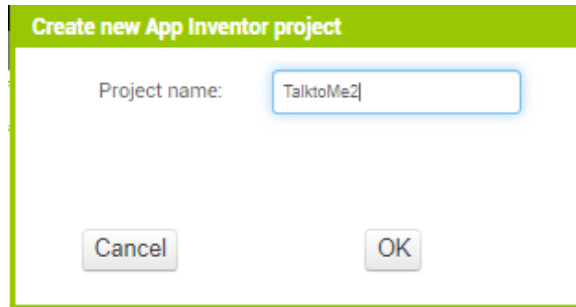
Tutorial kali ini adalah lanjutan dari aplikasi TalkToMe 1 dimana pada aplikasi menambah komponen Textbox sebagai isian teks

yang akan di konversi menjadi suara dan sensor **Accelrometer**, sensor tersebut merespons ketika ponsel diguncang maka akan menampilkan pesan suara yang kita inginkan.

### 2.3.1 Langkah untuk memulai aplikasi *Talk to me 2*:

Lakukan langkah 1,2,3,4 seperti pada bagian **2.1.1 Membangun aplikasi Hello Codi!**, tetapi jika jendela project masih terbuka lakukan langkah 4 saja klik tombol **Start new Project**.

5. Pada Jendela penamaan *project name* ketikan: TalkToMe2 atau nama lain yang kita inginkan, contoh pada gambar dibawah:



Gambar 2. 30 Penamaan Aplikasi Talk To Me 2

Klik **OK** untuk selanjutnya masuk ke antar muka desain

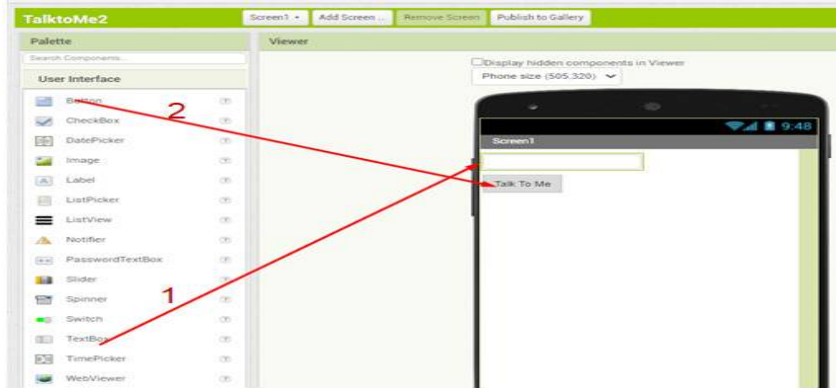
### 2.3.2 Komponen untuk Desain aplikasi TalkToMe2

#### Langkah 1:

Dari Antarmuka Pengguna pallete, seret **TextBox** dan letakkan pada viewer desainer .

#### Langkah 2:

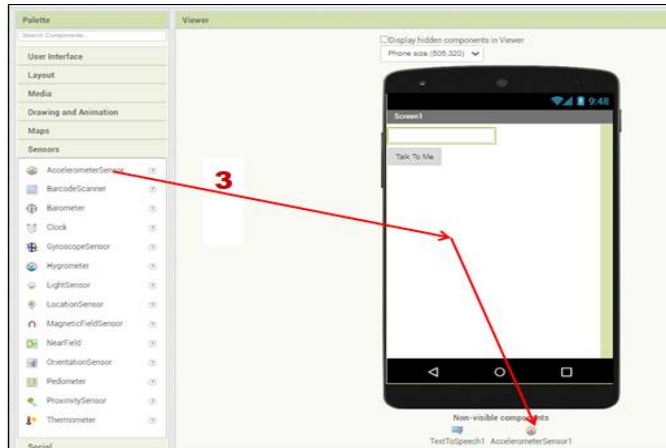
Selanjutnya seret **Button** dan letakkan diatas TextBox yang sudah ada pada viewer desainer . kemudian set properties text pada Button ketik **TalkToMe**



Gambar 2. 31 Penyisipan Pallete UI Button ke Viewer Desainer

### Langkah 3:

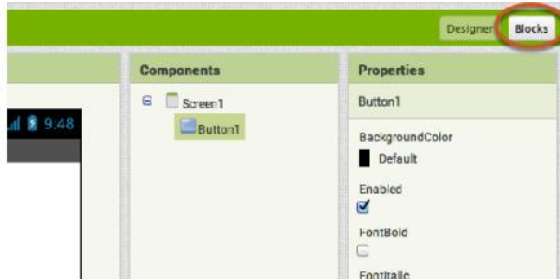
Pada Drawer **Sensors**, seret komponen **AccelerometerSensor** dan jatuhkan ke Viewer (komponen ini tidak terlihat, sehingga turun ke bagian bawah layar. **Non-Visible Component** )



Gambar 2. 32 Penyisipan Pallete AccelerometerSensor

### 2.3.3 Pemrograman dengan Block Editor

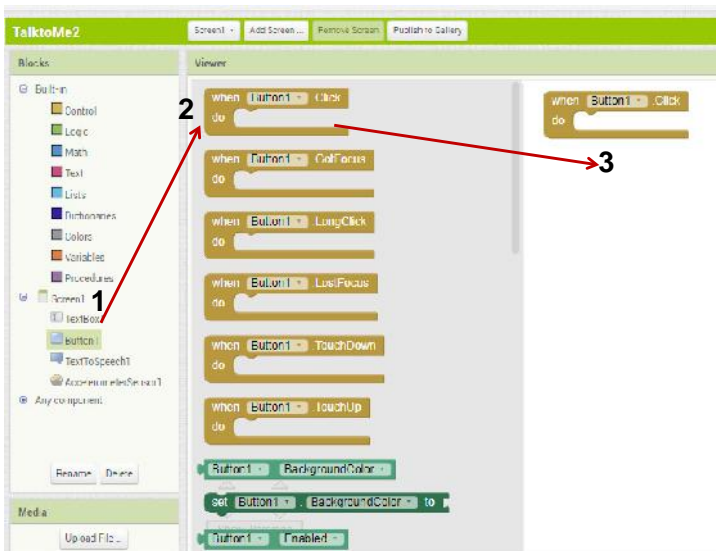
Saatnya memberikan program untuk aplikasi Talktome 2 dengan *block*, Klik tombol "*Block*" untuk pindah ke Editor *Block*.



Gambar 2. 33 Mengaktifkan Jendela Block untuk pemrograman

### Langkah 1:

Pilih pada drawer **Button1** (#1) . lalu Klik dan tahan "**when Button1.Click do** (#2)" Seret ke Viewer (#3) dan letakkan di sana. Blok ini akan digunakan ketika tombol di klik pada aplikasi Ini disebut "**Event Handler**"



Gambar 2. 34 Pemilihan even When Button1.Click untuk pemrograman

### Langkah 2:

Klik pada Drawer TextToSpeech(#1). Klik dan tahan **call TextToSpeech1.Speak** (#2) blok. Seret ke Viewer dan sisipkan

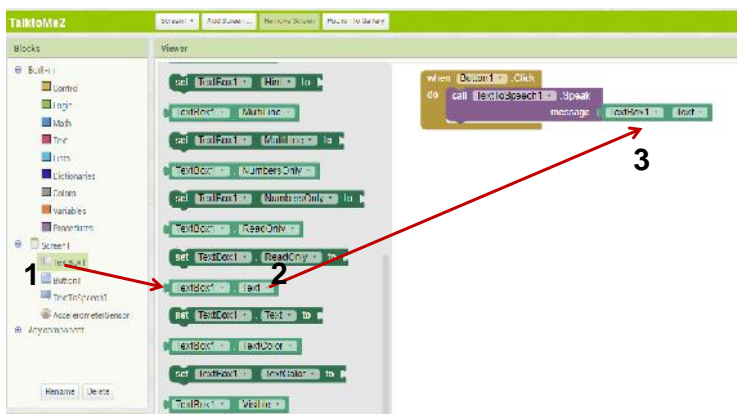
ke Slot **whenButton1.Click** (#3) . Ini adalah blok ketika tombol di klik akan akan memunculkan pesan suara.



Gambar 2. 35 Pemilihan even Call TextToSpeech1.Speech

### Langkah 3:

Klik pada Drawer **Textbox1**(#1). Klik dan tahan (#2) block **Textbox1.Text**. Seret ke Viewer (#3) dan sisipkan ke slot TextToSpeech1. Ini adalah blok ketika tombol di klik akan akan memunculkan pesan suara dari apapun yang diketik pada TextBox1 (Kotak Isian).



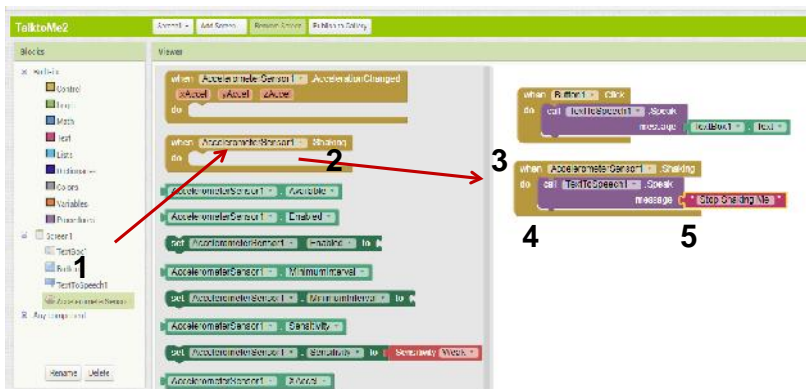
Gambar 2. 36 Pemilihan Block TextBox1.text ke slot message

Pada Langkah ini kita bisa menguji ke ponsel android apakah sudah berjalan sesuai yang diharapkan, ketika kotak isian textbox1 di input, kemudian tekan tombol(Button1), jika mengeluarkan pesan suara maka aplikasi berhasil, jika ingin melanjutkan maka ikuti tahapan berikut

#### Langkah 4:

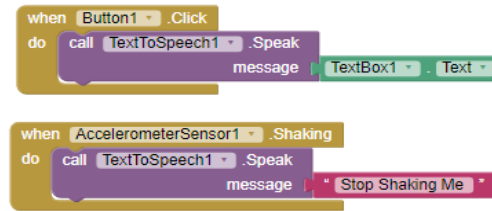
Klik pada Drawer **Accelerometer1(#1)**. Klik dan tahan **blocks WhenAccelerometer1.shaking(#2)**. Seret ke Viewer (#3) dan sisipkan CallTextToSpeech1.speak ke slot "do" (#4) seperti pada langkah 2 sebelumnya, selanjutnya sisipkan Drawer **Text**, ketikkan **"Stop Shaking Me" (#5)** "

**Block** ini ketika dijalankan dan ponsel di guncang akan menampilkan pesan suara **"Stop Shaking Me"**



Gambar 2. 37 *WhenAccelerometer1.shaking(#2)*

Sehingga hasil akhir dari **Block** pada aplikasi kita terlihat seperti gambar dibawah ini

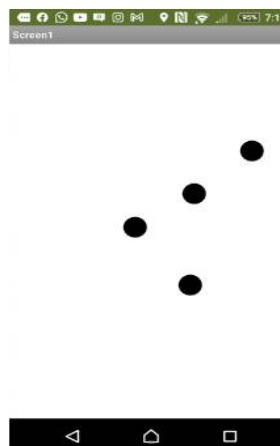


Gambar 2. 38 Hasil akhir Blocks Aplikasi TalkToMe2

#### 2.3.4 Menguji aplikasi TalkToMe2 ke perangkat android ponsel/Tablet:

Lakukan pengujian ke perangkat ponsel/tab android kita ikuti langkah 3 pada bagian **Menguji aplikasi HelloCodi** sebelumnya. Jika tidak ada masalah maka hasil akhir pengujian ditampilkan di perangkat ponsel atau android lakukan pengujian dengan mengetikan kata/kalimat pada kotak isian/Textbox tekan Tombol, juga gerakkan ponsel jika mendengar suara kata/kalimat yang diketikan maka Selamat aplikasi anda berhasil,,

### 2.4 MEMBANGUN APLIKASI KE EMPAT: *BALLBOUNCE/Aplikasi game sederhana*



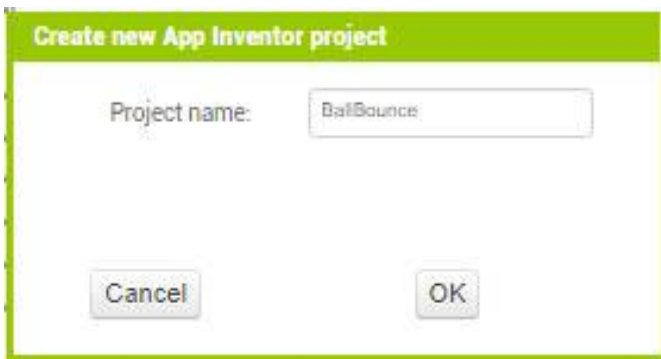
Gambar 2. 39 Aplikasi BallBounce



Pada tutorial kali ini, kita akan belajar tentang aplikasi animasi sederhana dengan membuat Bola (sprite) memantul di layar dan ketika berbenturan dengan batas layar akan memantul kembali sesuai arah bola .

#### 2.4.1 Langkah untuk Memulai aplikasi *BallBounce*:

Lakukan langkah 1,2,3,4 seperti pada bagian **2.1.1 Membangun aplikasi Hello Codi!**, tetapi jika jendela project masih terbuka lakukan langkah 4 saja, dengan meng klik **Start New Project** .



Gambar 2. 40 Penamaan Project Aplikasi *BallBounce*

5. Pada Jendela penamaan *project name* ketikan: BallBounce atau nama lain yang kita inginkan, contoh pada gambar dibawah:
6. Klik **OK** untuk selanjutnya masuk ke antar muka desain

#### 2.4.2 Komponen untuk Desain aplikasi *BallBounce*

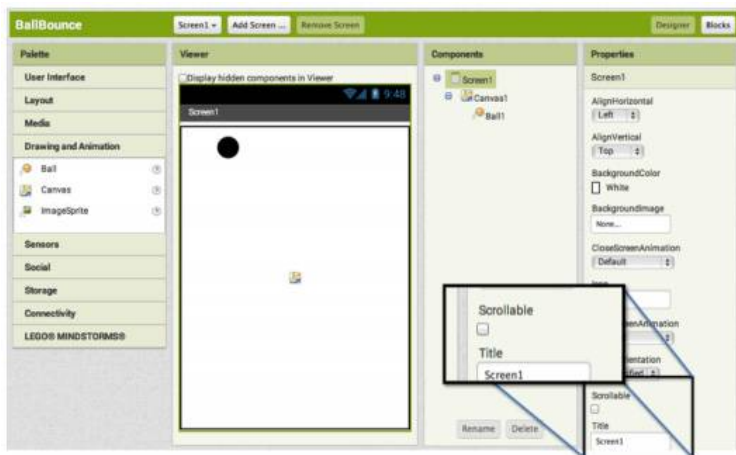
##### Langkah 1:

Dari **Drawer Drawing and Animation**, seret keluar komponen **Canvas** dan jatuhkan ke **viewer**.



Gambar 2. 41 Penyisipan Blocks Canvas ke dalam screen Viewer

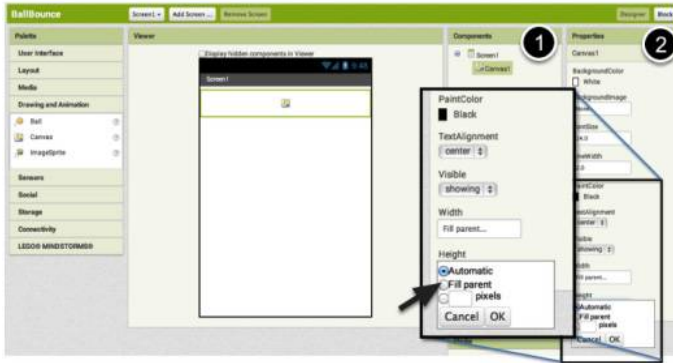
Pengaturan default untuk App Inventor adalah **Screen** aplikasi Anda akan “dapat digulir”/**Scrollable**, yang berarti antarmuka pengguna dapat melampaui batas layar dan pengguna dapat menggulir ke bawah dengan menggesekkan jari (seperti menggulir di halaman web). Saat kita menggunakan Kanvas, kita harus mematikan Setting “Scrollable” (**UNCHECK THE BOX**) agar layar tidak bergulir. Ini akan memungkinkan kita untuk membuat Kanvas mengisi seluruh layar.



Gambar 2. 42 Setting properties Scrollable di uncheck

## Langkah 2:

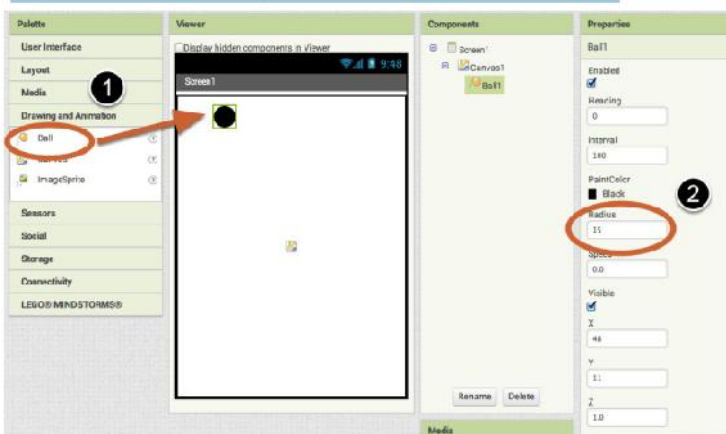
Pastikan komponen Canvas dipilih (#1) sehingga propertinya muncul di Panel Properties (#2). Di bagian bawah, atur properti **Height** ke "Fill Parent". Lakukan hal yang sama dengan Properti **Width**



Gambar 2. 43 Setting properties Canvas Height dan Width

## Langkah 3:

Sekarang setelah kita memiliki Canvas, kita dapat menambahkan Ball Sprite. Ini juga dapat ditemukan di *Drawing and Animation*. Drag Drop komponen Ball ke Canvas (#1). Jika kita ingin agar bola muncul, Kita dapat mengubah properti Radius di panel Properties (#2).



Gambar 2. 44 Menyisipkan Drawing Ball

### 2.4.3 Pemrograman dengan Block Editor aplikasi *Digital Doodle*

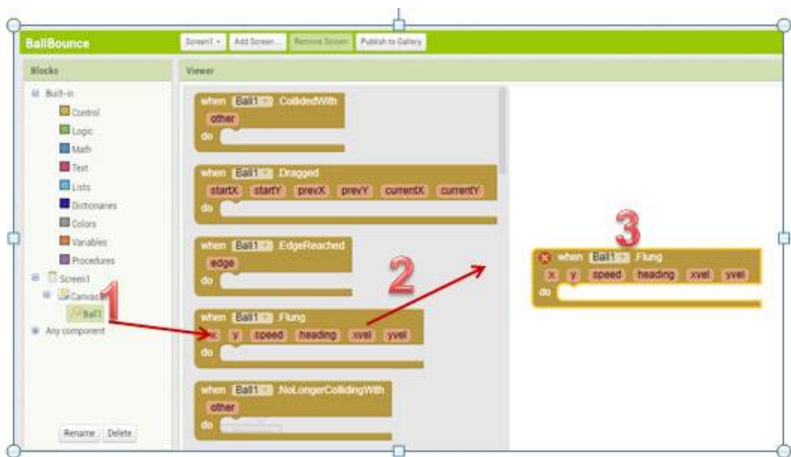


Gambar 2. 45 Mengaktifkan Jendela Block untuk pemrograman

Saatnya memberikan program untuk aplikasi **BallBounce** dengan **block**, Klik tombol "**Block**" untuk pindah ke **Editor Block**

#### Langkah 1:

Buka Drawer Ball1 [#1] untuk melihat **Block Ball1** Pilih blok **WhenBall1.Flung** [#2] dan drag-and-drop ke ruang kerja Viewer[#3] . **Flung** mengacu pada pengguna membuat "Fling Gestur " dengan jarinya untuk "melempar" Ball.

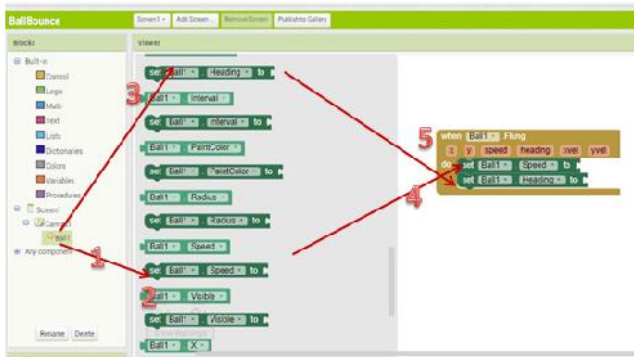


Gambar 2. 46 Menyisipkan Block When Ball.Flung ke Viewer

#### Langkah 2:

Buka Kembali Drawer Ball1 dan *scroll* ke bawah [#1] dalam daftar block pilih **set Ball1.Speed to** [#2] dan **Set Ball1.Heading to**

[#3], selanjutnya sisipkan **set Ball1.Speed to** [#4] dan **Set Ball1.Heading to** [#5] kedalam slot **When.Ball1.Flung** viewer, dan sisipkan seperti pada gambar dibawah ini:



Gambar 2. 47 Menyisipkan Blocks *set Ball1.Speed to* dan *Set Ball1.Heading to*

### Langkah 3:

Buka Kembali Drawer **Variable** [#1] dalam daftar block pilih **Block Get** [#2] sisipkan ke slot **set Ball1.Speed to** beri nilai Get dengan "**Speed**" [#3], sisipkan juga **Block "Get"** Sisipkan ke slot **Set Ball1.Heading to** beri nilai "Get" dengan "**Heading** [#4]" hasilnya seperti pada gambar dibawah ini:



Gambar 2. 48 Menyisipkan Blocks *variable Get* dengan nilai *speed* dan *heading*

**Langkah 4:**

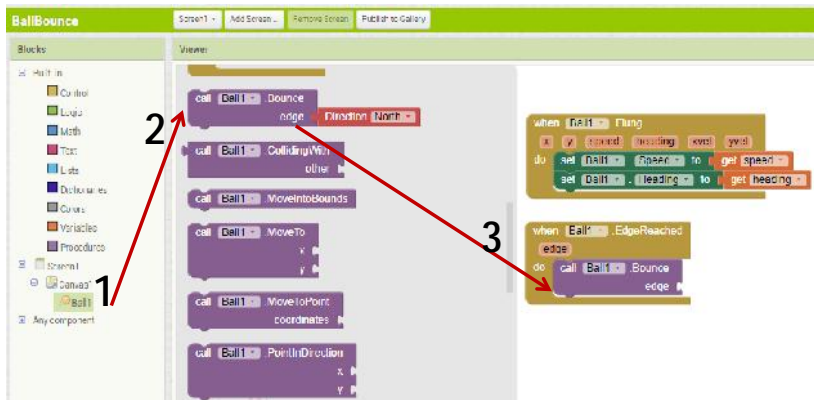
Buka Kembali Drawer **Ball1** [#1] dan *scroll* ke bawah dalam daftar block pilih lalu pilih **When Ball1.EdgeReached** [#2] *seret/drag* lalu tempatkan/drop jendela viewer [#3]



Gambar 2. 49 Menyisipkan When Ball1.EdgeReached

**Langkah 5:**

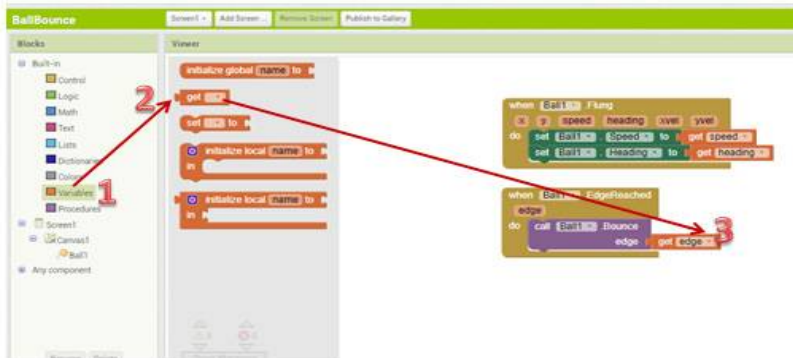
Buka Kembali Drawer **Ball1** [#1] dan *scroll* ke bawah dalam daftar block pilih lalu pilih **Call Ball1.Bounce Edge** [#2] *seret* lalu tempatkan pada slot **When Ball1.EdgeReached** do pada jendela viewer [#3] hapus bawaan block **Direction.North** hasilnya seperti pada gambar dibawah



Gambar 2. 50 Call Ball1.Bounce Edge ke When Ball1.EdgeReached

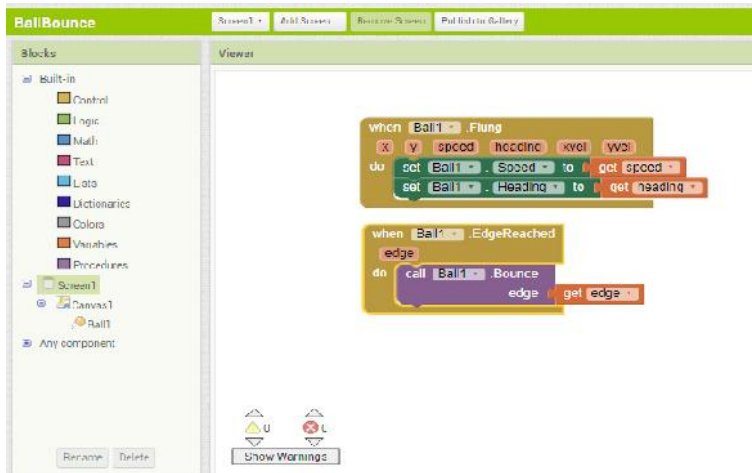
### Langkah 6:

Buka Kembali Drawer **Variable** [#1] dalam daftar block pilih **Block Get** [#2] sisipkan ke slot **Call Ball1.edge** lalu beri nilai Get dengan **"Edge"** [#3], seperti pada gambar dibawah ini:



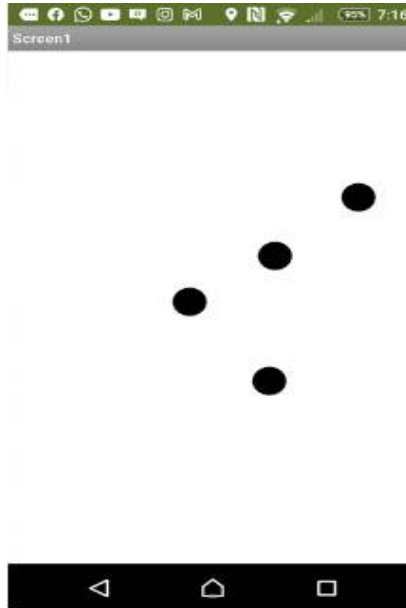
Gambar 2. 51 Menyisipkan Variable dengan Get Edge

Langkah Block untuk aplikasi BallBounce selesai secara keseluruhan hasil Block pada aplikasi kita seperti pada gambar dibawah ini:



Gambar 2. 52 Block Keseluruhan untuk aplikasi Ball Bounce

Lakukan pengujian ke perangkat ponsel/tab android kita ikuti langkah 3 yang sudah ada pada bagian **Menguji aplikasi HelloCodi** sebelumnya .



Gambar 2. 53 Hasil pengujian aplikasi Ball Bounce pada perangkat ponsel

Jika tidak ada masalah maka akan ditampilkan ke ponsel/Tablet android sebuah bola hitam, jika disentuh akan bergerak dan jika bertemu batas layar akan memantul, lalu bergerak kembali sesuai arah bola . Selamat anda telah berhasil membuat aplikasi sederhana ke empat.

## 2.5 MEMBANGUN APLIKASI KE LIMA: DigitalDoodle/APLIKASI MENGGAMBAR

Tutorial kelima kita adalah membuat aplikasi bagaimana menggambar garis garis di layar saat pengguna menyeret jarinya.



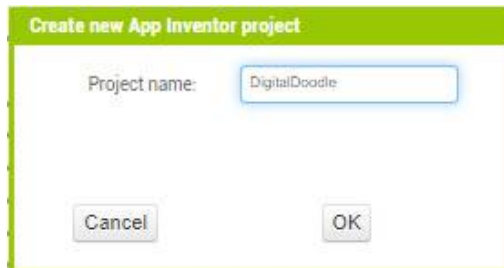


Gambar 2. 54 Aplikasi Digital Doodle

### 2.5.1 Langkah untuk memulai aplikasi *DigitalDoodle*:

Lakukan langkah 1,2,3,4 untuk membuat project seperti pada bagian **2.1.1 Membangun aplikasi Hello Codi!**, tetapi jika jendela project masih terbuka lakukan langkah 4 saja **yaitu start new project** .

5. Pada Jendela penamaan *project name* ketikan: **DigitalDoodle** atau nama lain yang kita inginkan, contoh pada gambar dibawah .



Gambar 2. 55 Aplikasi Digital Doodle

Klik **OK** untuk selanjutnya masuk ke antar muka desain

## 2.5.2 Komponen Desain dan Setting properties untuk aplikasi DigitalDoodle

### Langkah 1:

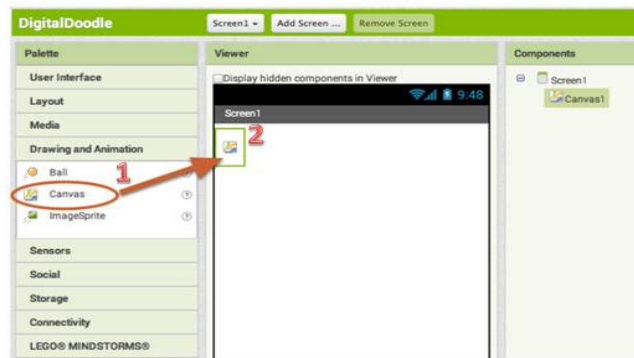


Gambar 2. 56 Setting properties Scrollable dengan unchecked

Pengaturan default untuk App Inventor adalah layar aplikasi kita akan “Scrollable”, berarti antarmuka pengguna dapat melampaui batas layar dan pengguna dapat menggulir ke bawah dengan menggesekkan jari (seperti menggulir di halaman web). Saat kita menggunakan Kanvas, kita harus mematikan Setting “Scrollable” dengan (**UNCHECK THE BOX**) agar layar tidak bergulir/*Scroolable*.

### Langkah 2:

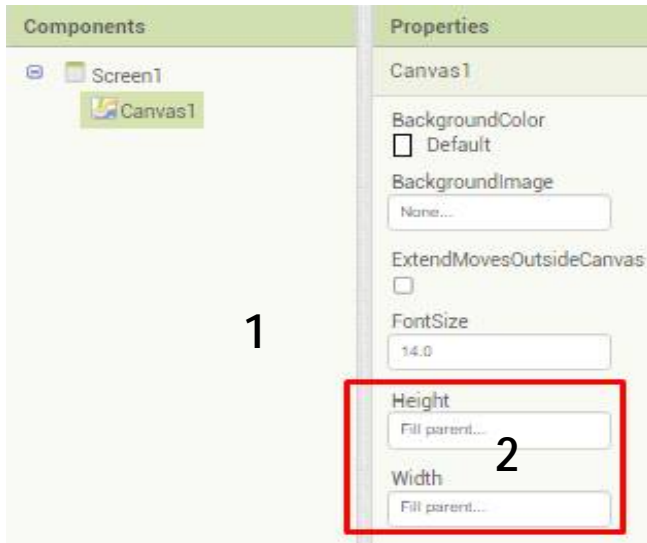
Dari **Drawing and Animation drawer**], drag Komponen **Canvas** [#1] lalu lepaskan ke jendela Viewer [#2]



Gambar 2. 57 penyisipan Palette Canvas ke jendela Viewer Screen

### Langkah 3:

Set Properties komponen Canvas (#1). Pada bagian bawah, atur properti **Height** ke "**Fill Parent**". Lakukan hal yang sama dengan **Width** ke "**Fill Parent**"



Gambar 2. 58 Setting properties Height dan Width canvas

Aplikasi ini hanya butuh Canvas untuk komponen desainnya . selanjutnya masuk ke Editor **Block** untuk memprogram aplikasi.

### 2.5.3 Pemrograman dengan *Block Editor*

Saatnya memberikan program untuk aplikasi **DigitalDoodle** dengan **block**, Klik tombol "**Block**" untuk pindah ke **Editor Block**



Gambar 2. 59 Mengaktifkan Jendela Block untuk pemrograman

**Langkah 1:**

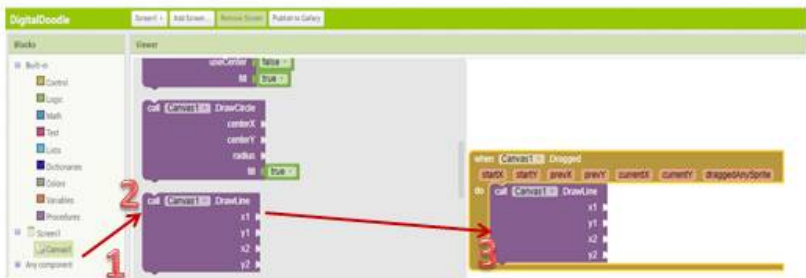
Pada *Drawer Canvas1* [#1], Drag **When Canvas1.Dragged** [#2] lalu lepas ke jendela Viewer [#3]



Gambar 2. 60 Menyisipkan Canvas dengan memilih Block **When Canvas.Dragged**

**Langkah 2:**

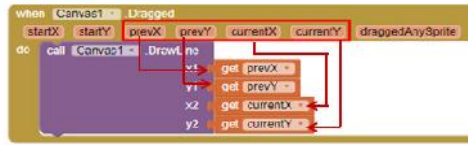
Masih Pada *Drawer Canvas1* [#1], pilih Block **Call Canvas.DrawLine** [#2] lalu Drag jendela Viewer Sisipkan ke Slot do **When Canvas1.Dragged** [#3]



Gambar 2. 61 Menyisipkan Block **Call Canvas.Drawline**

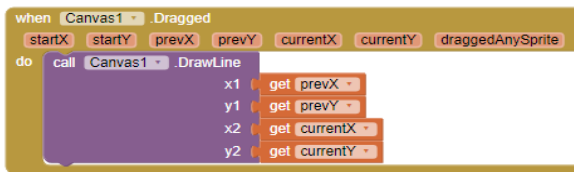
**Langkah 3:**

Berikan Nilai masing masing **X1 = get PrevX**, **Y1 = get PrevY**, **X2 = CurrentX**, **Y2= CurrentY**, pengambilan nilai pada masing masing tombol di bagian **When Canvas1.Dragged** Seperti nampak pada gambar dibawah



Gambar 2. 62 Memberikan Nilai Block Call Canvas.Drawline dengan  $X1 = \text{get PrevX}$ ,  $Y1 = \text{get PrevY}$ ,  $X2 = \text{CurrentX}$ ,  $Y2 = \text{CurrentY}$

Hasil akhir dari Program dengan **Block** aplikasi **DigitalDoodle** seperti gambar dibawah ini



Gambar 2. 63 Hasil Akhir Block Aplikasi Digital Doodle

#### 2.5.4 Menguji aplikasi DigitalDoodle ke perangkat android ponsel/Tablet:

Lakukan pengujian ke perangkat ponsel/tab android kita ikuti langkah 3 pada bagian **Menguji aplikasi HelloCodi** sebelumnya.

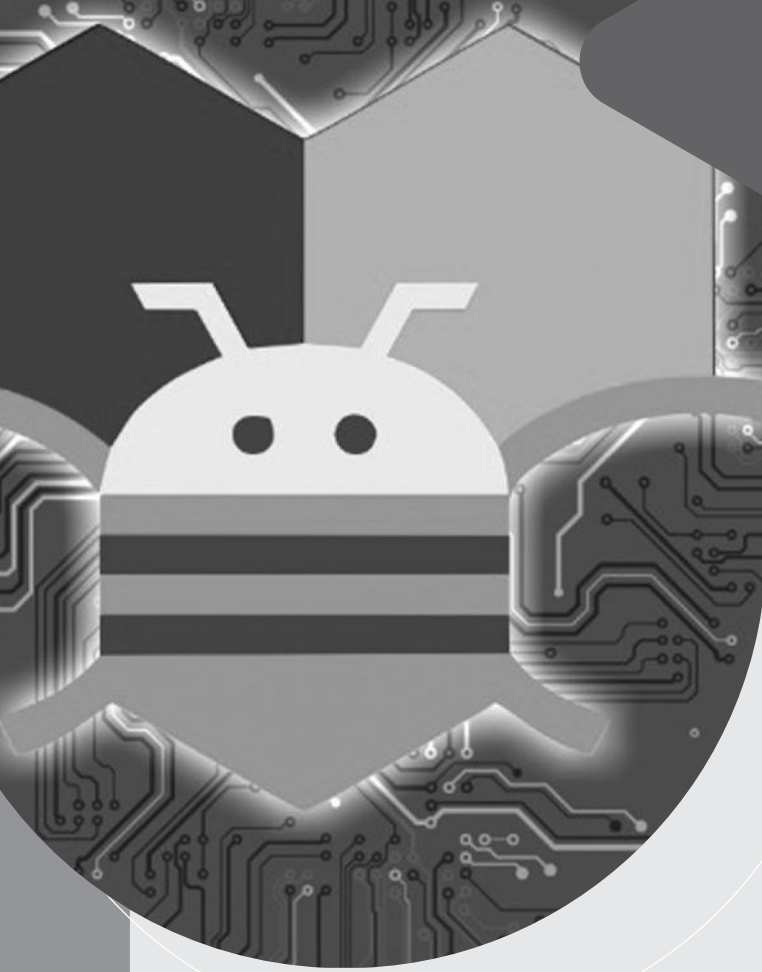


Gambar 2. 64 Hasil Pengujian ke Ponsel Android

Jika tidak ada masalah maka hasil akhir pengujian ditampilkan di perangkat android lakukan pengujian kembali dengan menulis ke layar dengan jari "Aplikasi digital Doodle" .

# BAB III

.....



## TUTORIAL MEMBANGUN APLIKASI DENGAN BLOCK MATH, DAN CONTROL

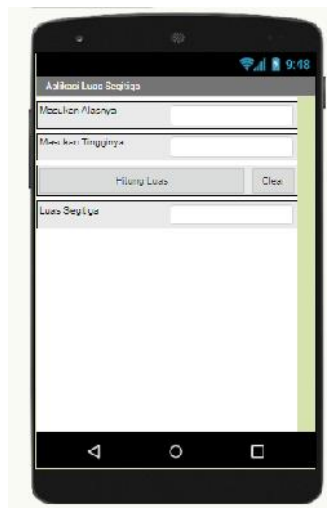
.....

.....

Setelah kita pelajari bagaimana membangun aplikasi sederhana pada bab sebelumnya Tutorial kali ini adalah membangun aplikasi dengan tambahan *blocks* **Math**, **Control Logic**, **Variable**, aplikasi yang akan kita buat diantaranya yaitu:

1. Aplikasi Menghitung Luas Segitiga
2. Aplikasi menentukan kelulusan dari nilai akhir
3. Aplikasi BMI (*Body Mass Index*)

### 3.1 APLIKASI MENGHITUNG LUAS SEGITIGA



Gambar 3. 1 Aplikasi Kalkulator Body Mass Index (BMI)

Pada Aplikasi luas segitiga terdapat 2 kotak isian Alas dan Tinggi untuk menerima input lalu 1 kotak output untuk luasnya. Alas, tinggi dan ;luas ketiganya menggunakan komponen Textbox. untuk menampilkan hasil dari rumus perhitungan luas segitiga dengan komponen *Button*

Algoritma deskriptif aplikasi menghitung luas segitiga:

1. Deklarasi/inisialisasi Variable ALAS,TINGGI, LUAS
2. Berikan Nilai awal ALAS dengan TextBox ALAS

3. Berikan Nilai awal TINGGI dengan TextBox Tinggi
4. Hitung Variable LUAS= 0,5 X (ALAS X TINGGI)
5. Tampilkan Hasil dari Variable LUAS pada isian TextBox Luas

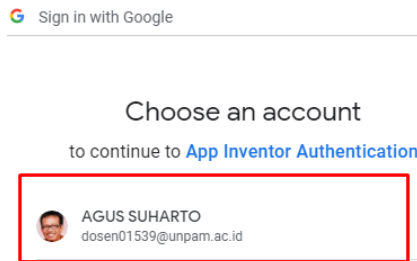
### 3.1.1 Langkah untuk memulai membuat aplikasi menghitung luas segitiga

#### Langkah 1:

1. Masuk ke web mit app inventor pada browser ketik::<https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik *Create Apps*  
Selanjutnya pilih akun google untuk masuk sebagai akun mit AI2



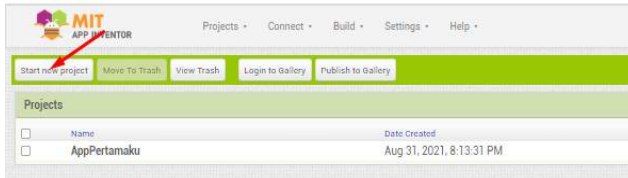
3. Pada halaman *project* klik *start new project* untuk membuat project baru  
*asi*



Gambar 3. 2 masuk ke MIT AI2 menggunakan akun Google

4. Pada jendela penamaan *project* beri nama *projectnya* tentunya ikuti aturan penamaan file misalnya tidak boleh ada spasi jika terdapat 2 suku kata, tidak boleh ada karakter simbol, tidak boleh didahului angka pada digit pertama nama. Berikan nama project: LuasSegitigaApp seperti pada contoh gambar dibawah ini:





Gambar 3. 3 Membuat Project Baru Aplik

5. Klik Ok untuk masuk jendela desainer



Gambar 3. 4 Penamaan Project Baru LuasSegitiga

### 3.1.2 Komponen Desain dan Setting properties untuk aplikasi Luas Segitiga

#### Langkah 1:

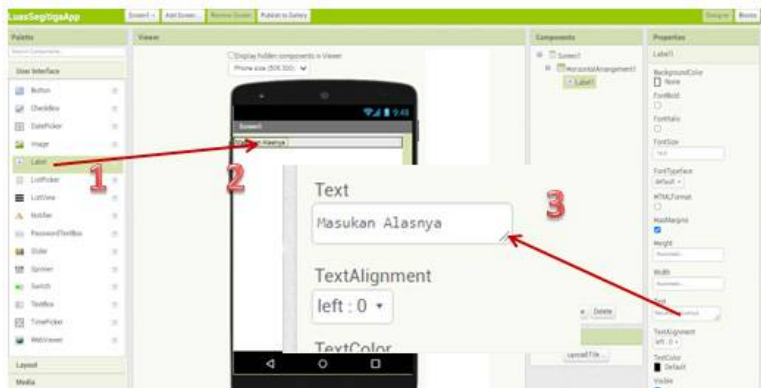
Klik **Drawer Layout**, pilih komponen **Horizontal Arrangement** [#1] untuk batas layout komponen **label** dan **textbox** Alas, Drag komponen tersebut lalu lepaskan ke jendela viewer [#2], setting properties **Height: automatic** dan **Width: Fill parent** [#3]



Gambar 3. 5 Setting properties Height dan Width

## Langkah 2:

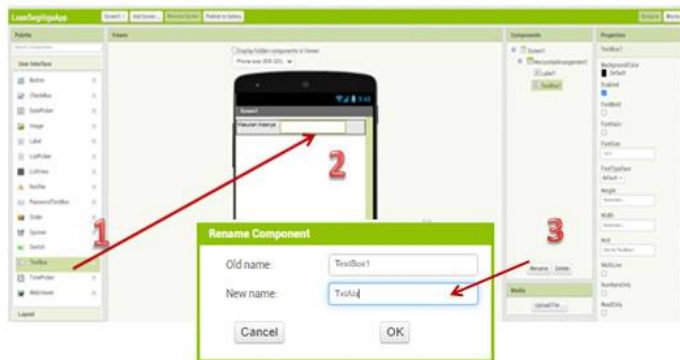
Klik **Drawer User Interface** pilih Komponen label [#1] lalu *drag* tempatkan didalam komponen **Horizontal Arrangement** [#2] selanjutnya *setting* properties text: "Masukan Alas nya:"[#3].



Gambar 3. 6 Menyisipkan label dan Setting properties Textnya

## Langkah 3:

Masih pada **Drawer User Interface** pilih Komponen **Textbox** [#1] lalu *drag* tempatkan didalam komponen **Horizontal Arrangement** [#2] *setting* properties **Name** dengan meng klik tombol **rename** [#3], lalu pada kotak isian **New Name** ketikan **TxtAlas** [#3] Penamaan ini untuk memberikan nama variable Textbox nya .



Gambar 3. 7 Menyisipkan komponen TextBox lalu Setting properties nya

#### Langkah 4:

Lengkapi Desain aplikasi selanjutnya lakukan langkah yang sama pada bagian sebelumnya dan sesuaikan kebutuhan aplikasi seperti pada gambar dibawah:



Gambar 3. 8 Gambar Komponen Desain Aplikasi Luas Segitiga

Pada jendela properties set masing maing components seperti pada tabel dibawah ini:

Tabel 3. 1 Desain Komponen dan Setting Properties nya

No	Pallete	Component Name	Properties
1	LayOut: Horizontal Alignment	Horizontal Allignment1	Height: Automatic Width: Fill Parent
2	User Interface: Label	Label1	Height: Automatic Width: Fill Parent Text: Masukan Alasnya

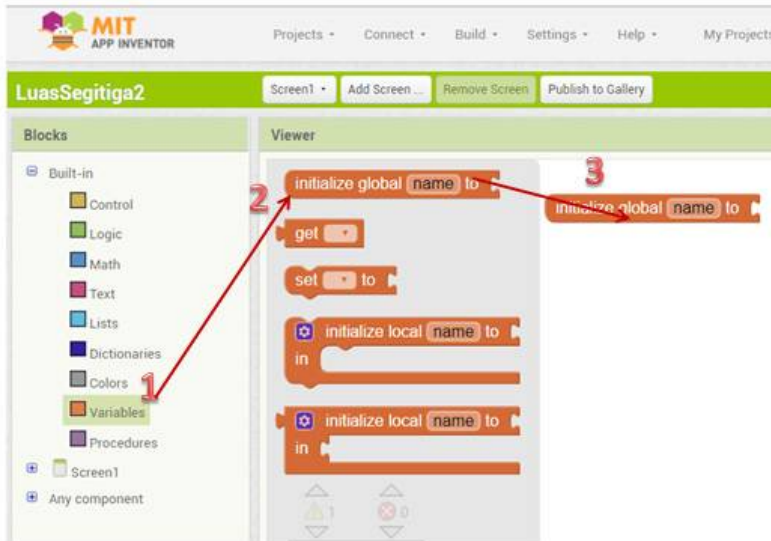
3	User Interface: TextBox	TxtAlas	TxtAlignment: Right
4	LayOut: Horizontal Allignment	Horizontal Allignment2	Height: Automatic Width: Fill Parent
5	User Interface: Label	Label2	Height: Automatic Width: Fill Parent Text: Masukan Tingginya
6	User Interface: TextBox	TxtTinggi	TxtAlignment: Right
7	LayOut: Horizontal Allignment	Horizontal Allignment4	Height: Automatic Width: Fill Parent
8	User Interface: Button	BtnHitung	Height: Automatic Width: Fill Parent Text: Hitung Luas Center: 1
9	User Interface: Button	BtnClear	Height: Automatic Width: Fill Parent Text: Clear Center: 1
10	LayOut: Horizontal Allignment	Horizontal Allignment3	Height: Automatic Width: Fill Parent
11	User Interface: Label	Label4	Height: Automatic Width: Fill Parent Text: Luas Segitiga
12	User Interface: TextBox	TxtLuas	TxtAlignment: Right Text: Kosongkan

### 3.1.3 Pemrograman dengan *Block Editor*

Pada langkah selanjutnya memberikan program untuk aplikasi **Luas Segitiga** dengan **blocks**, Klik tombol "**Blocks**" untuk pindah ke **Editor Block**

### Langkah 1:

Deklarasi/Inisiasi 3 Variabel alas, tinggi, luas, caranya adalah Pada **Blocks Built-in** pilih Blocks **Variables** [#1], lalu pilih Initialize Global name To [#2], drag lalu drop ke jendela **Viewer** bagian kanan [#3].



Gambar 3. 9 Pemberian Block Initialize Global name.to ke Viewer

### Langkah 2:

Ganti nama *initiale global name* dengan mengetikan **alas** pada kotak name .



Gambar 3. 10 Penamaan Blocks Initialize Global name.to dengan Alas

### Langkah 3:

Berikan nilai awal variabel alas dengan decimal 0,

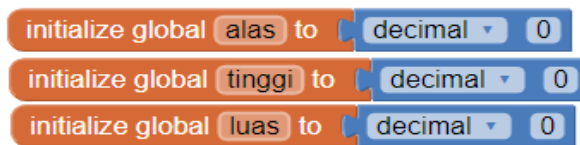
Pada **Blocks Built-in** pilih Blocks **Math** [#1], lalu pilih **decimal 0** [#2], drag lalu drop sisipkan ke slot setelah **to** [#3]



Gambar 3.11 Pemberian Nilai Block Initialize Global *alas.to* dengan *decimal 0*

#### Langkah 4:

Untuk variabel tinggi, luas dan memberikan nilai nya ulangi langkah yang sama seperti bagian sebelumnya hasilnya seperti nampak pada gambar 3.11



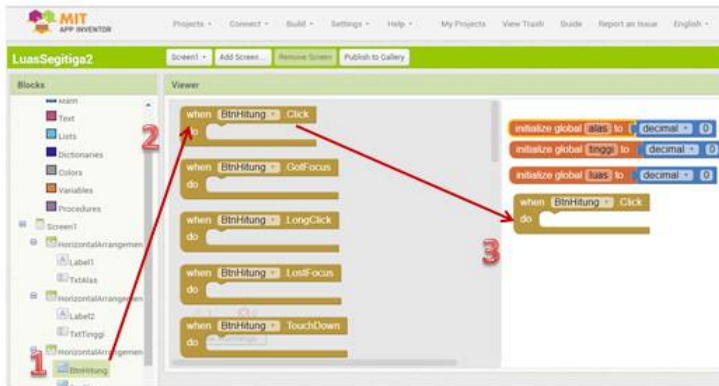
Gambar 3.12 Penamaan 3 Blocks Initialize Alas, tinggi, luas

#### Langkah 5:

Selanjutnya memberikan event klik pada Tombol Hitung Luas, dimana setelah memberikan nilai isian kotak input alas dan tinggi akan melaksanakan perhitungan luas segitiga,

Berikut Langkah nya: pada jendela *Blocks-Horizontal Arrangement4*, pilih komponen *BtnHitung* [#1], lalu pilih *When BtnHitung.click*, [#2]

drag lalu drop ke jendela viewer [#3] seperti ditunjukkan pada gambar dibawah ini



Gambar 3. 13 Pemberian Blocks When BtnHitung.Click ke Viewer

#### Langkah 6:

Menyisipkan variabel alas yang sudah di inisialisasi kedalam blocks **When BtnHitung Click** langkahnya pada **Blocks** pilih **Variable** [#1] lalu pilih **Set To** [#2], drag lalu drop tempatkan ke slot setelah **Do** [#3] setelah itu klik **panah** bawah **sebelum To** lalu pilih **Global Alas** [#4]

Seperti gambar dibawah ini:



Gambar 3. 14 Pemberian Nilai Blocks variables Alas di BtnHitung.Click

Berikan Nilai Variable Alas nya dengan komponen TxtAlas  
 Pada Block pilih TxtAlas [#1] dibawah **HorizontalArrangement1**,  
 lalu pilih **TxtLuas.Text** [#2] tempatkan pada slot setelah **to** [#3]  
 seperti pada gambar dibawah ini:

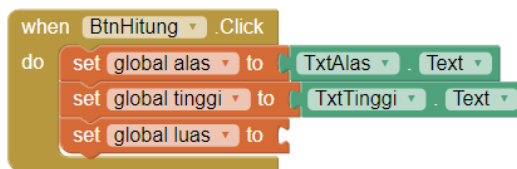


Gambar 3. 15 Pemberian Nilai Blocks variables Alas dengan TxtAlas.Text

### Langkah 7:

Menyisipkan variabel Tinggi dan Luas yang sudah di inialisasi kedalam blocks **When BtnHitung Click** langkahnya sama seperti Gambar 5 sebelumnya.

Sehingga hasilnya seperti gambar berikut:



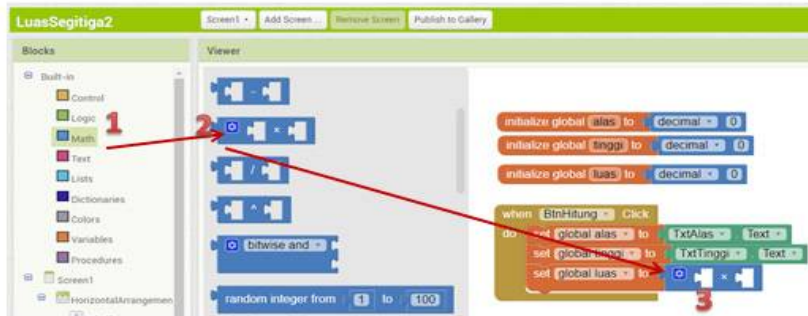
Gambar 3. 16 Pemberian Nilai Blocks variables Alas, tinggi dan luas

### Langkah 8:

Untuk variabel **Luas** akan diberikan rumus luas segitiga menggunakan **Blocks Math**, caranya:

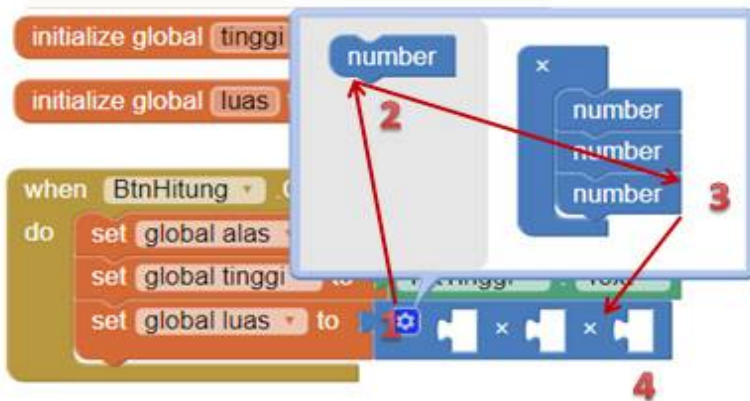


Pada **Blocks Built-in**, pilih **Math** [#1] lalu pilih **Blocks** perkalian [#2], drag lalu tempatkan setelah **Global Luas** to [#3]



Gambar 3. 17 Pemberian Nilai Blocks variables luas dengan rumus

Sebelum memberikan rumus siapkan 3 number untuk masing masing nilai (**0,5xalasxtinggi**) langkahnya klik simbol setting [#1], drag number di sisi kiri atas [#2], drag lalu drop dibawah number ke 2 [#3], sehingga hasilnya seperti gambar dibawah [#4]



Gambar 3. 18 Penambahan Blocks Number menjadi 3 komponen

Isikan masing masing *number* dengan rumus luas segitiga, number 1 sisipkan block **math** ketikan angka 0.5, number kedua

mengambil variabel alas ( **get global Alas** ) dan number ketiga mengambil nilai variable tinggi (**get global tinggi**), hasil nya seperti gambar berikut ini:



Gambar 3. 19 Pemberian rumus Blocks variable luas segitiga

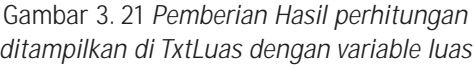
### Langkah 9:

Selanjutnya adalah menampilkan hasil perhitungan rumus luas ke dalam komponen TxtLuas, langkahnya adalah sebagai berikut: Pada jendela **Blocks** pilih komponen **TxtLuas**[#1], lalu pilih blocks **Set txtluas.Text.to**[#2] tempatkan dibawah blocks **set.global luas.to** [#3] seperti gambar dibawah ini:



Gambar 3. 20 Pemberian Nilai Hasil perhitungan ditampilkan di TxtLuas

Berikan nilai komponen **TxtLuas.text** dengan mengambil nilai variable [#1] **Get** [#2] lalu tempatkan setelah set txtluas.Text to[#3], pada get klik panah pilih **global luas** hasil blocksnya seperti gambar dibawah ini:



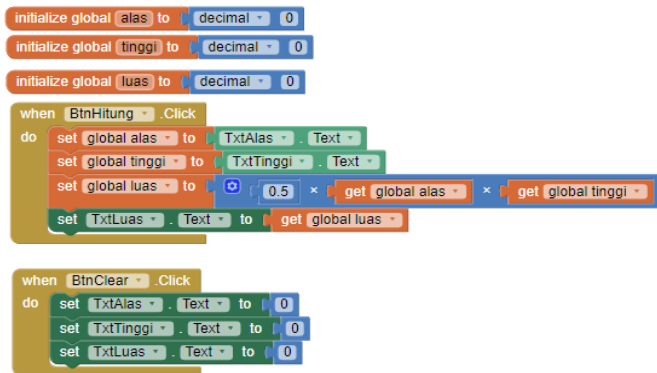
Keseluruhan dari *blocks* inialisasi dan tombol hitung luas adalah seperti ditunjukan gambar 3.21 di bawah ini:

Selanjutnya adalah memberikan Blocks untuk tombol **Clear**, Tombol ini adalah untuk mengosongkan semua komponen textbox TxtAlas, TxtTinggi, TxtLuas, setelah melakukan isian alas dan tinggi, dan menampilkan hasil ke TxtLuas,

Untuk langkah penyisipan *blocks component* adalah sama seperti langkah penyisipan tombol **Hitung luas**, **Blocks** untuk Tombol **Clear** adalah sebagai berikut:

Gambar 3. 23 *Blocks keseluruhan tombol Clear*

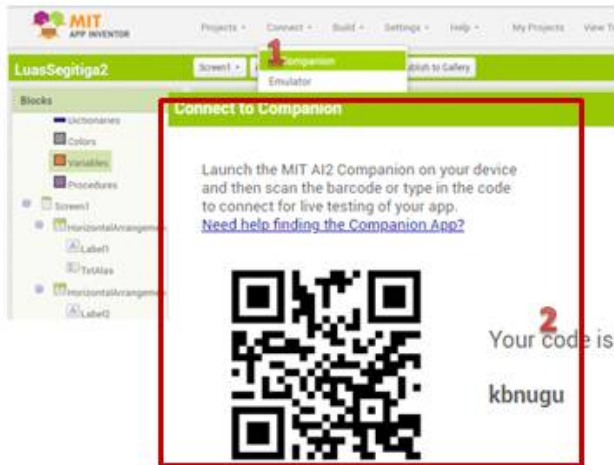
Sampai disini proses pembentukan **Blocks** keseluruhan adalah sebagai berikut:

Gambar 3. 24 *Blocks keseluruhan Aplikasi Luas Segitiga*

### 3.1.4 Menguji aplikasi Luas Segitiga ke perangkat android ponsel/Tablet:

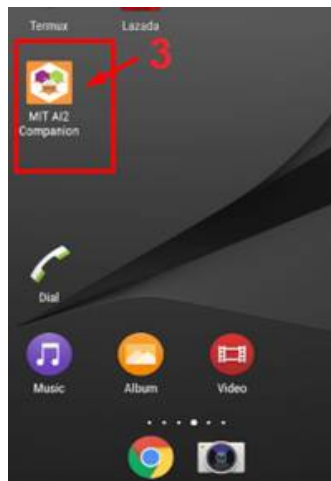
Setelah proses desain dan *block* selesai selanjutnya adalah menguji aplikasi kita ke perangkat android . langkah langkah nya sama seperti bagian pembuatan aplikasi sebelumnya ta[i marilah kita ulangi tahapan tahapannya:

1. Pada jendela project baik *designer* maupun *block*, Klik menu **Connect–AI Companion (#1)**
2. Selanjutnya **akan ditampilkan jendela QRcode** dan Code yang diberikan yang akan digunakan untuk dimasukan pada ponsel/tablet pengujian [#2].



Gambar 3.25 QrCode dan Code yang diberikan untuk pengujian

3. Buka aplikasi MIT AI2 Companion pada Ponsel/Tablet android yang digunakan sebagai media pengujian aplikasi



Gambar 3. 25 Aplikasi MIT AI2 di Ponsel android untuk pengujian

Pada jendela MIT AI2 Companion akan meminta isian kode QR Code yang didapatkan pada langkah 2, bisa kita ketikan atau scan **QRCode** tersebut sehingga otomatis code akan muncul pada kotak isian



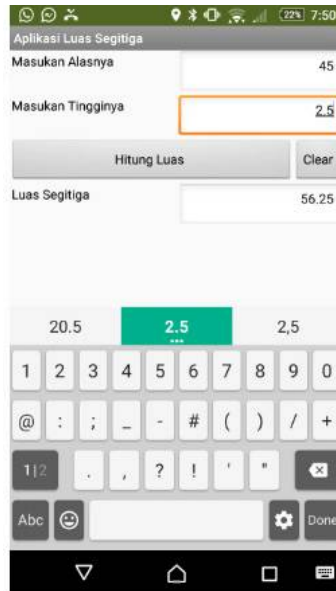
Gambar 3. 26 Aplikasi MIT AI2 di Ponsel android untuk pengujian dengan scan QR Code atau input Code

4. Selanjutnya pada project ada tampilan progress dan akan mengirimkan hasil pengujian ke perangkat android kita:



Gambar 3. 27 Gambar Progress pengujian pada layar web MIT AI2 setelah scan QR Code dari Ponsel

5. Jika tidak ada masalah maka hasil akhir pengujian ditampilkan di perangkat ponsel android. Seperti pada gambar dibawah ini:



Gambar 3. 28 Hasil Pengujian pada layar ponsel android

## 3.2 APLIKASI MENENTUKAN KELULUSAN NILAI AKHIR



Gambar 3. 29 Aplikasi Menentukan lulus\_tidak lulus

Aplikasi sederhana ke dua pada Bab 3 ini adalah menentukan kelulusan nilai akhir, dimana kita akan menambahkan *blocks Control* sebagai pengendali dari sebuah syarat menggunakan If..Else,

Algoritma dari aplikasi adalah sebagai berikut:

1. Masukan Nilai Akhir
2. Klik tombol **Proses**
3. Jika Nilai Akhir  $\geq 65$  maka keterangan = "Lulus", selain itu keterangan="Tidak Lulus"
4. Tampilkan hasil kondisi pada kotak Keterangan
5. Klik Tombol *Clear* untuk membersihkan dan mulai lagi input data baru.

### 3.2.1. Langkah untuk memulai membuat aplikasi menghitung luas segitiga

#### Langkah 1:

1. Masuk ke web mit app inventor pada browser ketik::<https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik *Create Apps*



Selanjutnya pilih akun google untuk masuk sebagai akun mit AI2

Sign in with Google

Choose an account

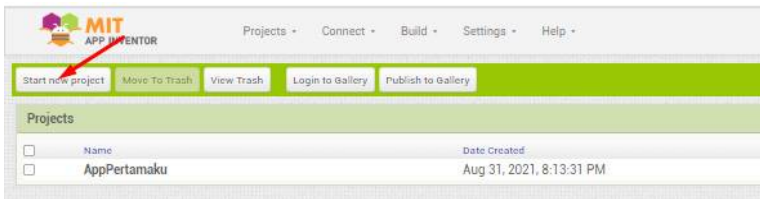
to continue to App Inventor Authentication



Gambar 3. 30 Masuk ke akun MIT AI2

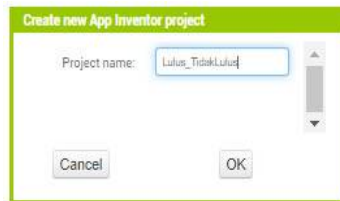


3. Pada halaman *project* klik *start new project* untuk membuat project baru



Gambar 3. 31 Memulai Project baru

Pada jendela penamaan *project* beri nama *projectnya*: Lulus\_ TidakLulus seperti pada contoh gambar dibawah ini:



Gambar 3. 32 Memberi nama Project

### 3.2.2 Komponen Desain dan Setting properties untuk aplikasi

Desain dan Setting properties pada aplikasi ini tidak dijelaskan langkah demi langkah per komponen karena di asumsikan sudah mengetahui bagaimana cara memilih komponennya untuk gambar desain di sajikan pada gambar dibawah ini



Gambar 3. 33 Komponen Desain aplikasi yang dibutuhkan

Setting properties seperti pada tabel berikut ini:

Tabel 3. 2 Komponen Desain dan setting propertiesnya

No	Pallete	Component Name	Properties
1	<i>LayOut: Horizontal Alignment</i>	<i>Horizontal Allignment1</i>	<i>Height: Automatic Width: Fill Parent</i>
2	<i>User Interface: Label</i>	<i>Label2</i>	<i>Height: Automatic Width: Fill Parent Text: Nilai Akhir</i>
3	<i>User Interface: TextBox</i>	<i>TextBox1</i>	<i>TextAlignment: Right Hint: Kosongkan</i>
4	<i>LayOut: Horizontal Alignment</i>	<i>Horizontal Allignment2</i>	<i>Height: Automatic Width: Fill Parent</i>
5	<i>User Interface: Label</i>	<i>Label3</i>	<i>Height: Automatic Width: Fill Parent Text: Keterangan</i>
6	<i>User Interface: TextBox</i>	<i>Textbox3</i>	<i>TextAlignment: Right Hint: Kosongkan</i>
7	<i>LayOut: Horizontal Alignment</i>	<i>Horizontal Allignment4</i>	<i>Height: Automatic Width: Fill Parent</i>
8	<i>User Interface: Button</i>	<i>BtnProses</i>	<i>Height: Automatic Width: Fill Parent Text: Proses Center: 1</i>
9	<i>User Interface: Button</i>	<i>BtnClear</i>	<i>Height: Automatic Width: Fill Parent Text: Clear Center: 1</i>

### 3.2.3 Pemrograman dengan Block Editor

Pada langkah selanjutnya memberikan program dengan **blocks**, Klik tombol "**Blocks**" untuk pindah ke **Editor Block**

**Langkah 1:** pada langkah ini tidak di jelaskan bagaimana cara meletakkan **blocks** ke jendela Viewer diasumsikan sudah mengetahui bagaimana langkah demi langkah nya.

Di aplikasi ini kita coba tanpa inisialisasi Variable, tetapi variable-nya menggunakan nama komponen langsung yaitu Textbox1 untuk Nilai dan Texbox2 untuk keterangan sehingga langsung saja menyisipkan Block untuk Even Proses

Buat event ketika tombol proses di klik, pada **Blocks** pilih BtnProses(#1), lalu pilih **When BtnProses.Click** [#2] drag lalu letakan pada jendela Viewer [#3]



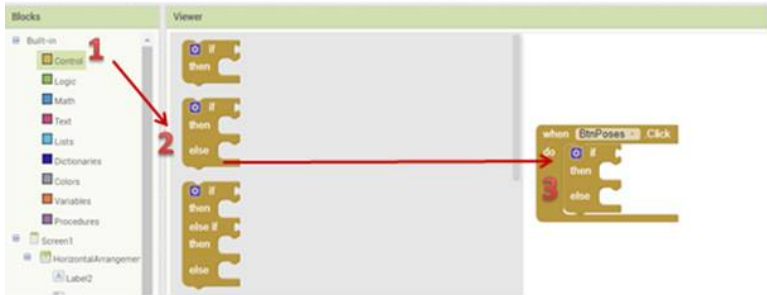
Gambar 3. 34 Blocks Komponen BtnProses

#### Langkah 2:

Langkah ini mengontrol Nilai yang telah di input lalu mengisikan pernyataan keterangan menggunakan If...Then ..Else alur

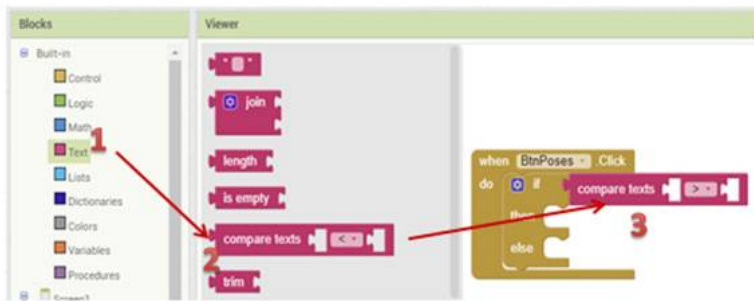
kondisinya adalah Jika Nilai  $\geq 60$  Keterangan="Lulus", Selain itu Keterangan ="Tidak Lulus "

1. Pada jendela *Blocks* pilih Control[#1], lalu pilih block *If then Else* [#2] drag lalu drop ke dalam **when BtnProses.Click** [#3]



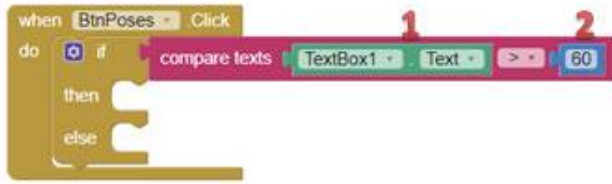
Gambar 3. 35 *Blocks* Komponen *If then Else* disisipkan ke *BtnProses*

2. Pilih Block *Text* [#1] lalu pilih *Compare texts* [#2] drag lalu masukan kedalam slot setelah *If* [#3], pilih simbol Greater (>) [#4]



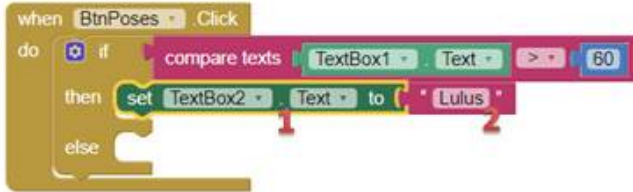
Gambar 3. 36 *Block* Komponen *Compare Text* disisipkan ke *If*

3. Sisipkan *TextBox1.text* [#1], untuk dibandingkan dengan angka > (lebih besar) 60 mengambil dari komponen *math* [#2]



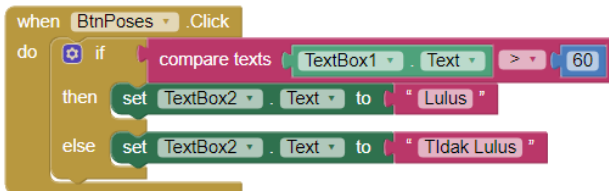
Gambar 3. 37 *Block Komponen TextBox1.text (Nilai) diberikan kondisi*

4. Selanjutnya mengisikan pernyataan Keterangan atau TextBox2.text [#1] setelah *then* dengan Blocks Text "Lulus" [#2] jika TextBox1.Text >60 (True)



Gambar 3. 38 *Block Komponen TextBox2.text (Keterangan ) diberikan Nilai Lulus berdasarkan kondisi IF >60*

5. Isikan Keterangan atau TextBox2.text [#1] setelah *else* dengan Blocks Text "Tidak Lulus" [#2] jika TextBox1.Text <60 (False)

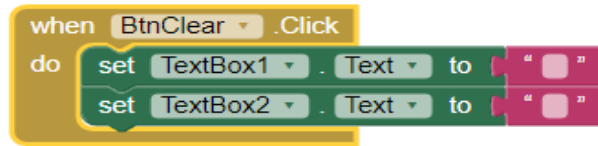


Gambar 3. 39 *Block Komponen TextBox2.text (Keterangan) Nilai "Tidak Lulus"*

Sampai disini untuk pengisian **Blocks when BtnProses**. **Click** selesai kemudian dilanjutkan dengan **Blocks BtnClear** (Tombol *Clear*)

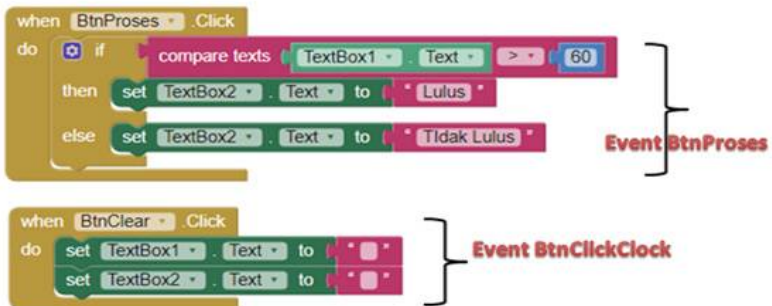
**Langkah 4:**

**Blocks BtnClear** adalah membersihkan komponen **TextBox1 (Nilai)** dan komponen **TextBox2(Keterangan)**, dan *user* bisa meng input ulang untuk proses selanjutnya, pada langkah ini tidak dibahas karena sama dengan bagian langkah sebelumnya, hasil dari **Blocks BtnClear** seperti pada gambar dibawah ini:



Gambar 3. 40 Block Komponen BtnClear (Tombol Clear )

Sehingga untuk Blocks keseluruhan aplikasi ini seperti nampak pada gambar dibawah ini:



Gambar 3. 41 Block Komponen Keseluruhan Aplikasi

### 3.2.4 Menguji aplikasi Lulus\_Tidak Lulus ke perangkat android ponsel/ Tablet:

Setelah proses desain dan *blocks* selesai selanjutnya adalah menguji aplikasi kita ke perangkat android . untuk langkah langkah nya sama seperti bagian pembuatan aplikasi sebelumnya.

### 3.3 APLIKASI KALKULATOR BMI (Body Mass Index)



Gambar 3. 42 Aplikasi Kalkulator BMI

Body Mass Index (BMI) atau Indeks Massa Tubuh (IMT) adalah angka yang menjadi penilaian standar untuk menentukan apakah berat badan kita tergolong Status Kurang, Normal, Berlebih, obesitas tingkat 1 atau Obesitas tingkat 2. Sebagai acuan menentukan status adalah mendapatkan angka BMI nya dengan rumus seperti dibawah ini:

$$\text{BMI} = \frac{\text{Berat Badan}}{(\text{Tinggi Badan})^2}$$

Gambar 3. 43 Rumus Angka BMI (Body Mass Index)

Keterangan:

- Berat Badan dalam Kilogram
- Tinggi Badan Dalam Meter

Hasil dari Angka BMI menentukan status berat badan mengacu pada tabel dibawah ini:

Tabel 3. 3 Angka BMI dan Statusnya

BMI	STATUS BERAT BADAN
<18.5	Kekurangan Berat badan
18,5–22,9	Normal
23-24,9	Kelebihan Berat badan
25-29,9	Obesitas Tingkat 1
>30	Obesitas Tingkat 2

Algoritma dari aplikasi Kalkulator BMI sederhana ini adalah:

1. Inisialisasi variabel Berat Badan
2. Inisialisasi variable Tinggi
3. Inisialisasi variable Angka BMI
4. Masukkan Angka Berat Badan dalam Kilogram
5. Masukkan Angka Tinggi dalam meter
6. Hitung rumus BMI = Berat Badan/(Tinggi \* Tinggi )
7. Tekan Tombol Proses
8. Tampilkan angka BMI nya
9. Tampilkan Status BMI nya
10. Tekan Tombol Bersihkan untuk mulai menghitung lagi

### 3.3.1 Langkah untuk memulai membuat aplikasi menghitung luas segitiga

#### Langkah 1:

1. Masuk ke web mit app inventor pada browser ketik::<https://appinventor.mit.edu/>
2. Setelah halaman app inventor muncul klik *Create Apps*



Gambar 3. 44 Membuat Project baru

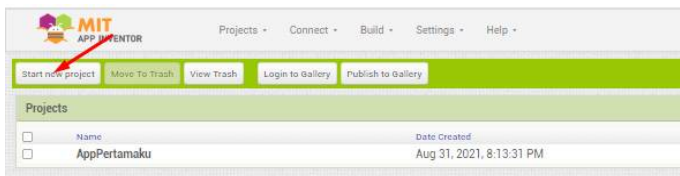


- Selanjutnya pilih akun google untuk masuk sebagai akun mit AI2



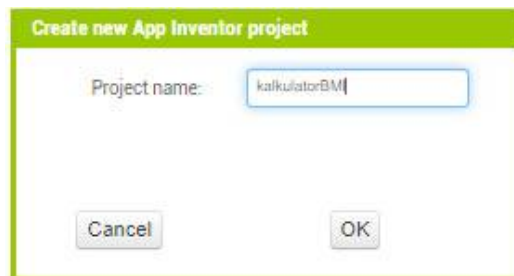
Gambar 3. 45 Masuk ke akun AI2 dengan Akun Google

- Pada halaman *project* klik *start new project* untuk membuat project baru



Gambar 3. 46 Masuk ke akun AI2 dengan Akun Google

- Pada jendela penamaan *project* beri nama *projectnya*: kalkulatorBMI seperti pada contoh gambar dibawah ini:



Gambar 3. 47 Penamaan Project Aplikasi Masuk ke akun AI2 dengan Akun Google

Klik Ok untuk melanjutkan

### 3.3.2 Komponen Desain dan Setting properties untuk aplikasi

Desain dan Setting properties pada aplikasi KalkulatorBMI ini tidak dijelaskan langkah demi langkah per komponen karena di asumsikan sudah mengetahui bagaimana cara memilih komponennya. Desain di sajikan terlihat pada gambar dibawah ini



Gambar 3. 48 *Komponen Desain Aplikasi KalkulatorBMI*

Setting properties desain seperti pada tabel berikut:

Tabel 3. 4 *Komponen Desain dan setting propertiesnya*

No	Palette	Component Name	Properties
1		Screen1	<i>Title: Kalkulator BMI</i>
2	<i>LayOut: Horizontal Allignment</i>	<i>Horizontal Allignment1</i>	<i>Height: Automatic Width: Fill Parent</i>

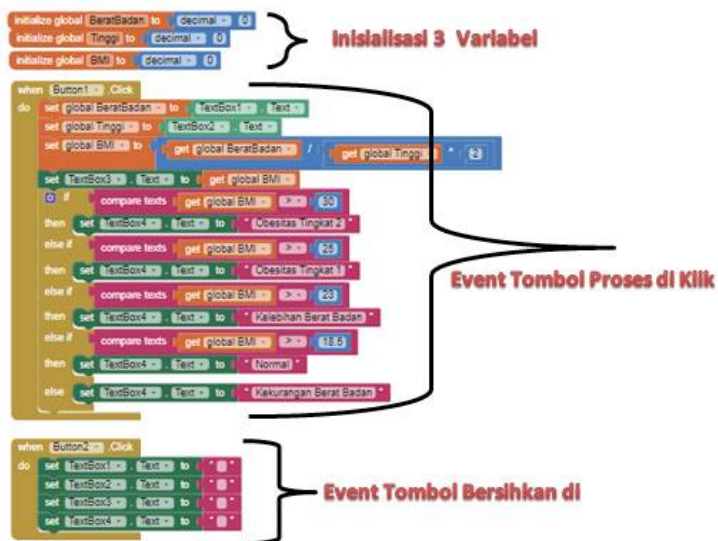
3	User Interface: Label	Label1	Height: Automatic Width: Fill Parent Text: Masukan Berat (Kg)
4	User Interface: TextBox	TextBox1	TextAlignment: Right Hint: Kosongkan
5	LayOut: Horizontal Allignment	Horizontal Allignment2	Height: Automatic Width: Fill Parent
6	User Interface: Label	Label2	Height: Automatic Width: Fill Parent Text: Masukan Tinggi (Meter)
6	User Interface: TextBox	Textbox2	TextAlignment: Right Hint: Kosongkan
7	LayOut: Horizontal Allignment	Horizontal Allignment5	Height: Automatic Width: Fill Parent
8	User Interface: Button	BtnProses	Height: Automatic Width: Fill Parent Text: Proses Center: 1
9	User Interface: Button	BtnClear	Height: Automatic Width: Fill Parent Text: Bersihkan Center: 1
10	LayOut: Horizontal Allignment	Horizontal Allignment3	Height: Automatic Width: Fill Parent
11	User Interface: Label	Label3	Height: Automatic Width: Fill Parent Text: Angka BMI Anda:
12	User Interface: TextBox	Textbox3	TextAlignment: Right Hint: Kosongkan
13	LayOut: Horizontal Allignment	Horizontal Allignment4	Height: Automatic Width: Fill Parent
14	User Interface: Label	Label4	Height: Automatic Width: Fill Parent Text: Status Berat badan:
15	User Interface: TextBox	Textbox4	TextAlignment: Right Hint: Kosongkan

### 3.3.3 Pemrograman dengan Block Editor

Pada langkah selanjutnya memberikan program dengan **blocks**, Klik tombol "**Blocks**" untuk pindah ke **Editor Block**

Pada langkah membuat **Blocks** aplikasi kalkulatorBMI tidak di jelaskan bagaimana cara meletakkan **blocks** ke jendela Viewer diasumsikan sudah mengetahui bagaimana langkah demi langkah nya serta pemilihan komponen blocksnya karena sudah di bahas pada Bab 3.1 dan Bab 3.2.

Berikut hasil akhir dari **Blocks** aplikasi kalkulatorBMI:



Gambar 3. 49 Block Aplikasi KalkulatorBMI

### 3.3.4 Menguji aplikasi Lulus\_Tidak Lulus ke perangkat android ponsel/ Tablet:

Setelah proses desain dan **blocks** selesai selanjutnya adalah menguji aplikasi kita ke perangkat android . untuk langkah langkah nya sama seperti bagian pembuatan aplikasi sebelumnya. Hasil Ujicoba aplikasi KalkulatorBMI pada ponsel android jika berjalan baik nampak pada gambar:

KALKULATOR BMI

Masukan Berat badan (Kg) 75

Masukan Tinggi (Meter) 1.50

Proses Bersihkan

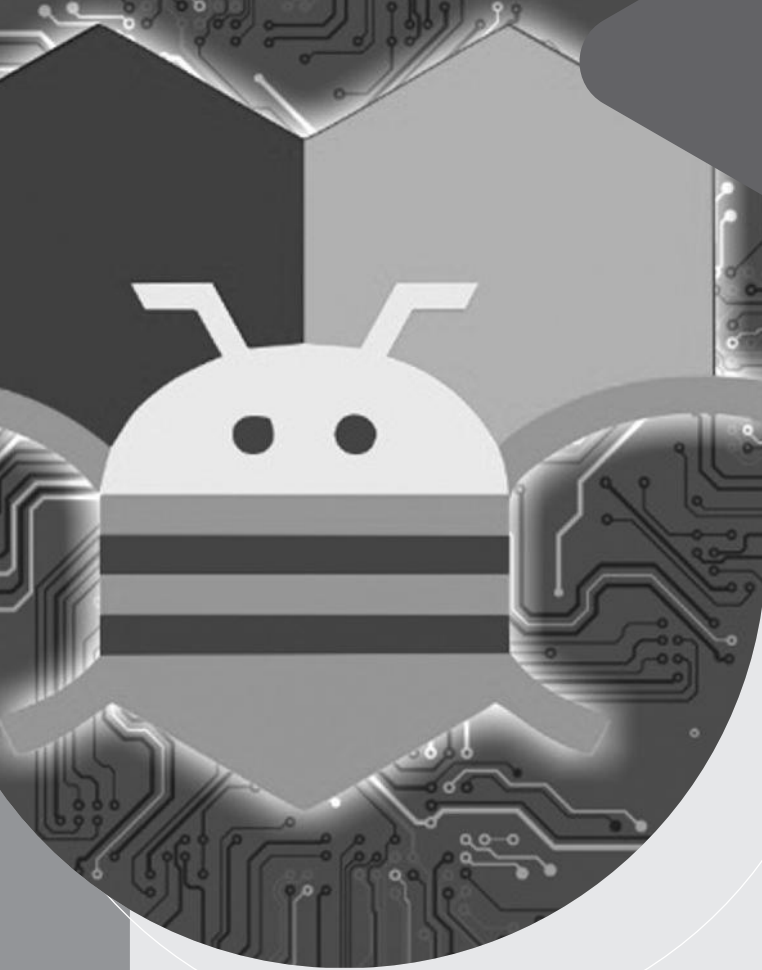
angka BMI anda 33.33333

status Berat Badan Obesitas Tingkat 2

Gambar 3. 50 Hasil Uji Aplikasi KalkulatorBMI di perangkat android

## BAB IV

.....



# **MENDISTRIBUSIKAN, MENGUBAH PROJECT MIT APP INVENTOR MENJADI FILE APK ANDROID**

.....

.....

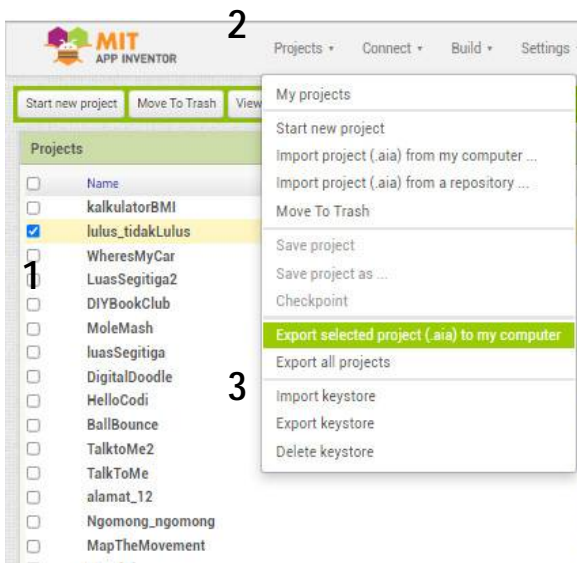
Setelah kita membuat aplikasi sederhana pada proyek AI2, pada pengujian ke ponsel atau gadget android yang dilakukan belum ter install ke gadget android, lalu bagaimana untuk mengubahnya menjadi .apk, selain itu kita juga dapat meng ekspor project aplikasi kita (.ai2) untuk dapat di *share* ke akun orang lain, juga bisa distribusikan aplikasi kita di Google Play Store.

## 4.1 EKSPOR FILE PROJECT APLIKASI KE KOMPUTER/LAPTOP (.AIA FILE)

Untuk Meng ekspor File **.AIA** pastikan daftar semua proyek yang ada tampil di jendela project, atau pilih menu **Project > My Projects** [#1] . Setelah semua project terlihat Selanjutnya Pilih **Project** [#2] yang ingin kita ekspor dengan mencentang kotak di sebelahnya.

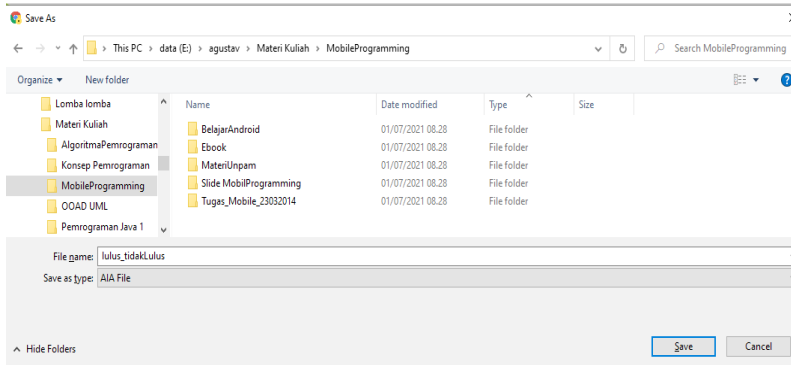
Selanjutnya pilih menu **Project > Export Select Project (.aia) to My Computer** [#3] untuk mengekspor kode sumber nya (*Blocks*) .

Seperti pada gambar 4.1



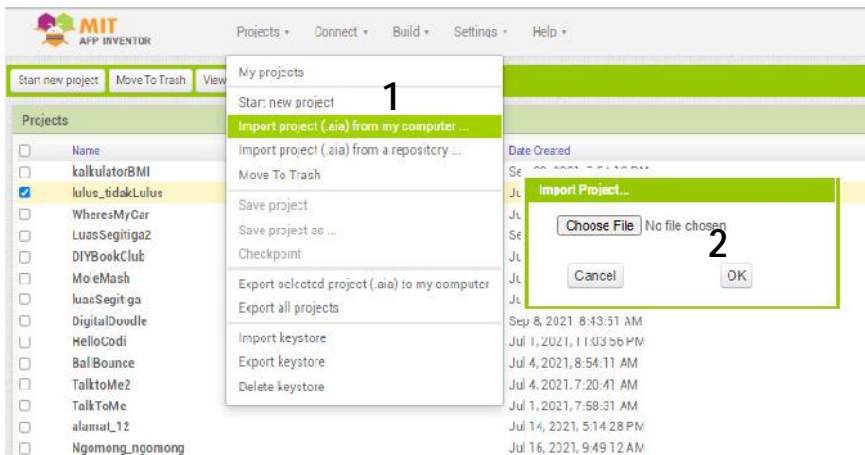
Gambar 4.1 Ekspor File Project .AIA

tempatkan file **.AIA** ke folder komputer yang di inginkan lalu klik **Save** seperti pada gambar 4.2 dibawah ini



Gambar 4. 2 Menyimpan File Project .AIA ke Komputer

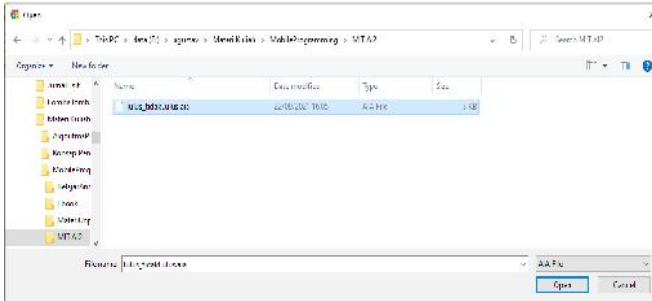
Jika kita ingin mengirimkannya ke teman atau membuka dari akun MIT AI2 lain, dapat membukanya **dengan Project > Import project (.aia) from my computer** [#1], klik **choose** untuk memilih File .AIA seperti yang ditunjukkan pada Gambar 4.3.



Gambar 4. 3 Meng Import File Project .AIA ke Komputer



Selanjutnya klik file .AIA nya



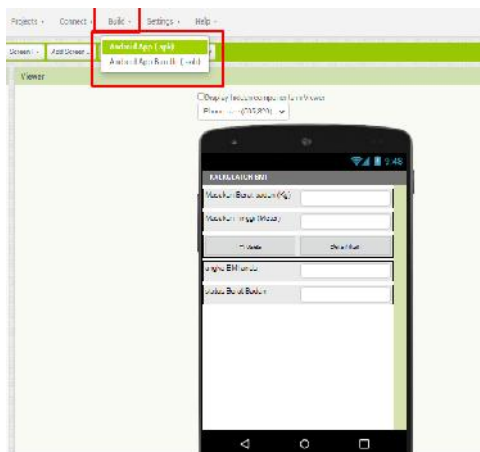
Gambar 4. 4 Memilih File Project .AIA ke akun ai2

## 4.2 MEMBUAT APLIKASI .AIA MENJADI FILE .APK

Seperti di jelaskan sebelumnya saat pengujian aplikasi sudah dapat berjalan di perangkat android, tetapi belum ter install, setelah selesai pengujian dan tidak terkoneksi lagi maka aplikasi yang sudah berjalan baik di perangkat android tersebut akan hilang, maka untuk itu dibutuhkan file .apk nya.

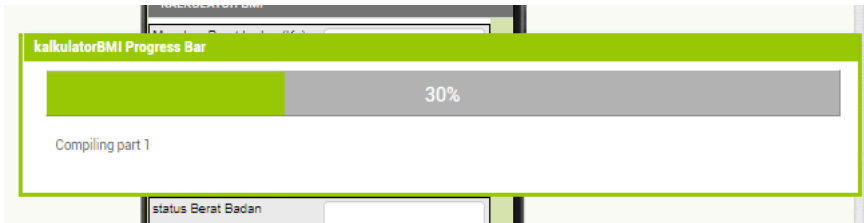
Berikut adalah langkahnya:

Buka projectnya yang akan dijadikan .apk lalu pilih menu **Build** [#1] pada toolbar App Inventor, Pilih menu **Android App (.apk)** [#2], seperti terlihat pada gambar 4.5



Gambar 4. 5 Mengubah File Project .AIA ke File .apk

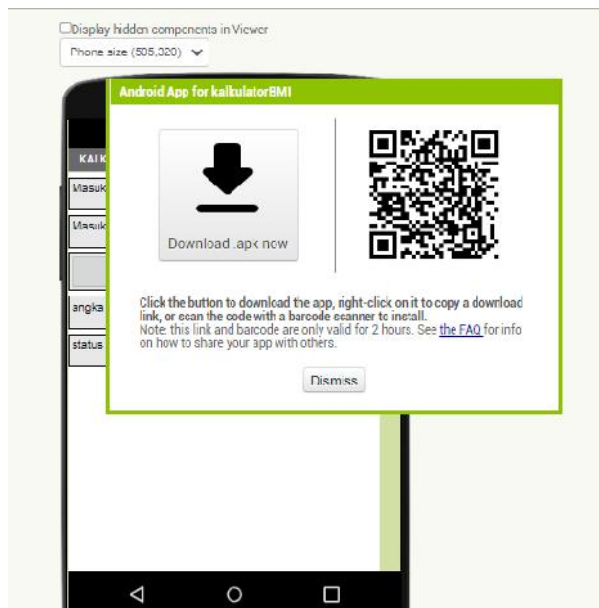
Selanjutnya akan muncul tampilan *progress bar*, **tunggu sampai progress 100%**



Gambar 4. 6 Progressbar menjadi File .apk

Jika tampilan *progress* sudah selesai 100% akan ada 2 pilihan untuk meng unggah file **.apk**

1. Melalui tombol *download* dan file .apk akan di tempatkan ke media penyimpanan laptop/Komputer
2. Melalui Scan QRCode file .apk hasil download di tempatkan ke perangkat android sekaligus meng install ke perangkatnya .



Gambar 4. 7 pemilihan Download File .apk

Proses pembentukan file AIA menjadi .apk selesai. Jika kita ingin mendistribusikannya lebih luas, kita dapat mengunggahnya ke situs web yang dapat diakses oleh kita dan teman kita. juga bisa kita distribusikan aplikasi Anda melalui Google Play Store

Catatan:

1. Ketika kita menginstal aplikasi yang merupakan file .apk perlu mengubah pengaturan pada perangkat android untuk memungkinkan penginstalan aplikasi *non-market*.
2. Untuk menemukan pengaturan ini pada versi Android sebelum 4.0, buka Setting *Settings*/Pengaturan > *Applications*/Aplikasi kemudian centang kotak di sebelah *Unknown Sources*/Sumber Tidak Dikenal
3. Untuk perangkat yang menjalankan Android 4.0 keatas ; buka *System settings*/Pengaturan sistem > *Security or Settings*/Keamanan atau Pengaturan > *Security & Screen lock*/Kunci layar, centang kotak di sebelah *Unknown source*/Sumber tidak dikenal dan konfirmasi pilihan Anda.



## DAFTAR PUSTAKA

Abeywardena, Ishan Sudeera, (2015). *EDUCATIONAL APP DEVELOPMENT TOOLKIT FOR TEACHERS AND LEARNERS*, *Commonwealth of Learning*, Burnaby, British Columbia Canada V5H 4M2.

<http://appinventor.mit.edu/explore/library>, diakses pada 20 agustus 2020

<http://appinventor.mit.edu/explore/ai2/tutorials>, diakses pada 1 Juli 2020

## PROFIL PENULIS

**Agus Suharto**, lahir di Bandung, 29 Mei 1965 . Meraih Ijazah Sarjana S1 Manajemen Informatika tahun 1992, Universitas Gunadarma. Kemudian meraih ijazah S2, gelar Magister Komputer ( M.Kom) STMIK Eresha Jakarta tahun 2011, Saat ini mengajar sebagai Dosen Universitas Indonesia program vokasi Akuntansi sejak tahun 2004, dan Dosen tetap Universitas Pamulang program studi Sistem Informasi sejak tahun 2016, kompetensi mata kuliah mengampu Database, Pemrograman, Rekayasa Perangkat Lunak, selain mengajar juga sebagai praktisi tenaga ahli programmer di beberapa Konsultan IT.



MIT App Inventor adalah sebuah *tools* pemrograman berbasis *blocks* yang memungkinkan para pemula untuk memulai pemrograman dan membangun aplikasi untuk perangkat *mobile* Android. *Block* disini adalah kumpulan atau *code block* berbentuk *graphic* seperti *puzzle* yang berwarna warni, dimana didalamnya terdapat komponen komponen **Logic, Control, Math, Text, Lists, Colors, Variables**, dan **Procedures**. Untuk para Pendatang baru dapat mengembangkan dan menjalankan aplikasi pertamanya pada MIT App Inventor dalam waktu kurang dari satu jam, dan dapat memprogram aplikasi yang lebih kompleks dalam waktu yang jauh lebih singkat dibandingkan dengan aplikasi berbasis teks dengan bahasa pemrograman tradisional. Buku ini berupa tutorial bagaimana membuat aplikasi untuk pemula bersumber dari tutorial web resmi <http://ai2.appinventor.mit.edu/> dan aplikasi yang dibuat oleh penulis sendiri. Buku ini layak dimiliki untuk semua kalangan SD, SMP, SMA, Mahasiswa, para pendidik berbagai jurusan, serta masyarakat umum yang ingin belajar bagaimana membuat aplikasi android tanpa harus *coding*, dengan tutorial langkah demi langkah yang mudah, dan nantinya sebagai referensi ke tingkat mahir atau sebagai profesi pengembang aplikasi berbasis android.



 Penerbit Adab  
 @penerbitadab  
 [www.PenerbitAdab.id](http://www.PenerbitAdab.id)

Pabean Udik - Indramayu - Jawa Barat  
Telp. 081221151025 | [penerbitadab@gmail.com](mailto:penerbitadab@gmail.com)

KOMPUTER

ISBN 978-623-6233-97-9

