



MODUL

UVU MODUL

Pengenalan Dasar MIT
App Inventor 2



iMECHATEUS 2021

Daftar Isi

Daftar Isi	2
i. Penjelasan Singkat	3
ii. Aplikasi yang dapat dibuat	3
iii. Keunggulan	3
iv. Kekurangan	3
1.Pengenalan Aplikasi	4
1.1 Designer Area	4
1.1.1 Pallette	4
1.1.2 Viewer	5
1.1.3 Components and Properties	5
1.1.4 Tab Bar	6
1.2 Blocks Area	7
1.1.2 Blocks	7
1.1.3 Viewer	8
2. Menghubungkan Dengan Perangkat	9
2.1 Menghubungkan dengan smartphone	9
2.2 Menghubungkan dengan Arduino	10
3. Percobaan	14
3.1 Percobaan 1	14
3.2 Percobaan 2	16

Penjelasan Singkat

App inventor adalah aplikasi web Open Source yang dikembangkan oleh Google, dan sekarang dikelola oleh Massachusetts Institute of Technology (MIT) yang memungkinkan orang awam untuk melakukan pemrograman komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi android.

Aplikasi yang dapat dibuat dengan MIT App Inventor

- A. Aplikasi sederhana
- B. Beberapa aplikasi sederhana menggunakan sensor;
 - 1. Text to Speech
 - 2. Speech to Text
 - 3. Barcode Reader
- C. Aplikasi yang dapat diintegrasikan dengan perangkat luar seperti mikrokontroler

Keunggulan MIT App Inventor

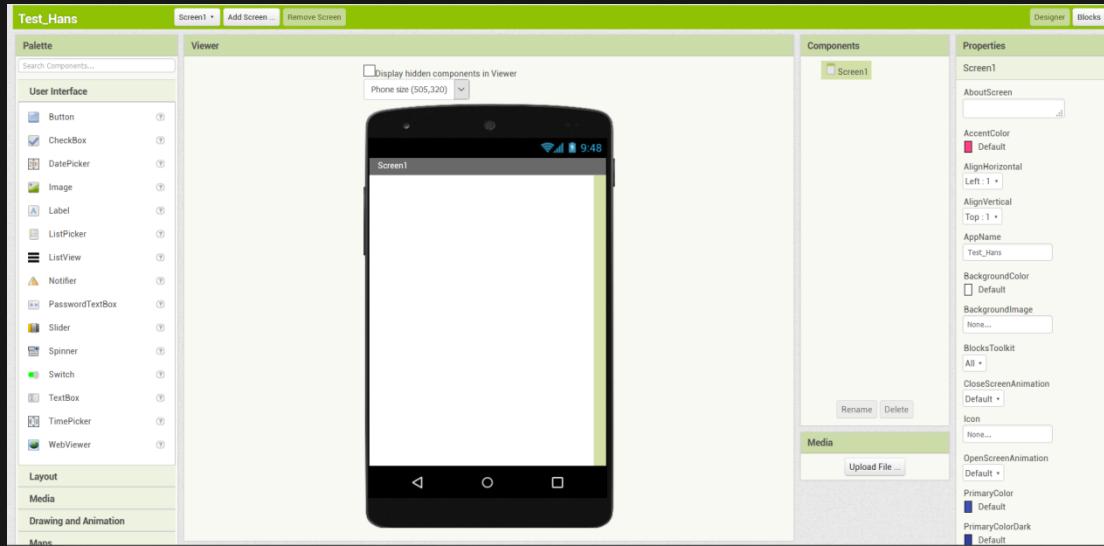
- A. Berbasis Goole Blockly, pemrograman visual dengan tujuan untuk mempermudah semua orang untuk melakukan pemograman tanpa mengetahui syntax program.
- B. Komponen dan Blok event sudah tersedia dengan lengkap dan user hanya perlu melakukan drag and drop.

Kekurangan MIT App Inventor

- A. Kurang fleksibel apabila dibandingkan dengan native programing.
- B. Pada saat menggunakan sensor dalam aplikasi, terkadang tidak didukung oleh beberapa device.

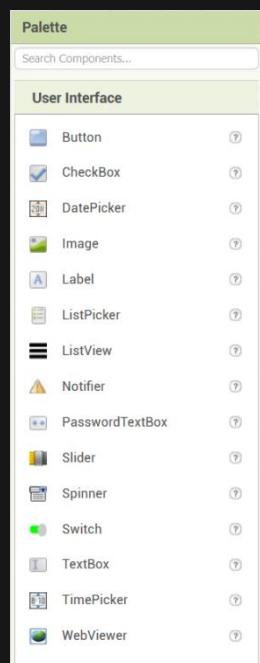
1. Pengenalan Aplikasi

1.1 Designer Area



1.1.1 Palette

Kelompok komponen-komponen yang akan digunakan



1.1.2 Viewer

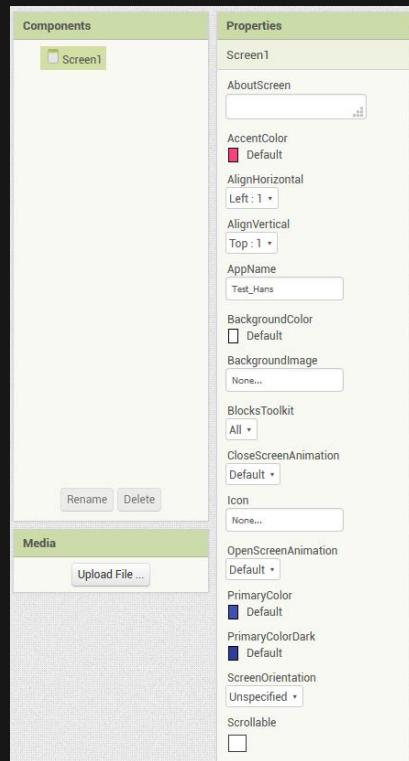
Merupakan workspace atau area kerja, tempat user drag and drop komponen dari **Palette**



1.1.3 Components & Properties

Components merupakan item-item dari **Palette** yang kita gunakan di dalam project

Properties merupakan detail properties dari tiap **Palette** item yang kita gunakan seperti; memberikan label, warna, dll



1.1.4 Tab Bar

Test_Hans => Nama project yang sedang dikerjakan

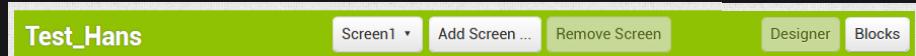
Tab Screen digunakan untuk mengatur **Viewer/Workspace** untuk menampilkan screen yang ingin digunakan

Add Screen digunakan untuk menambah screen

Remove Screen digunakan untuk menghapus screen, hanya akan aktif apabila kita berada selain di **Screen 1**

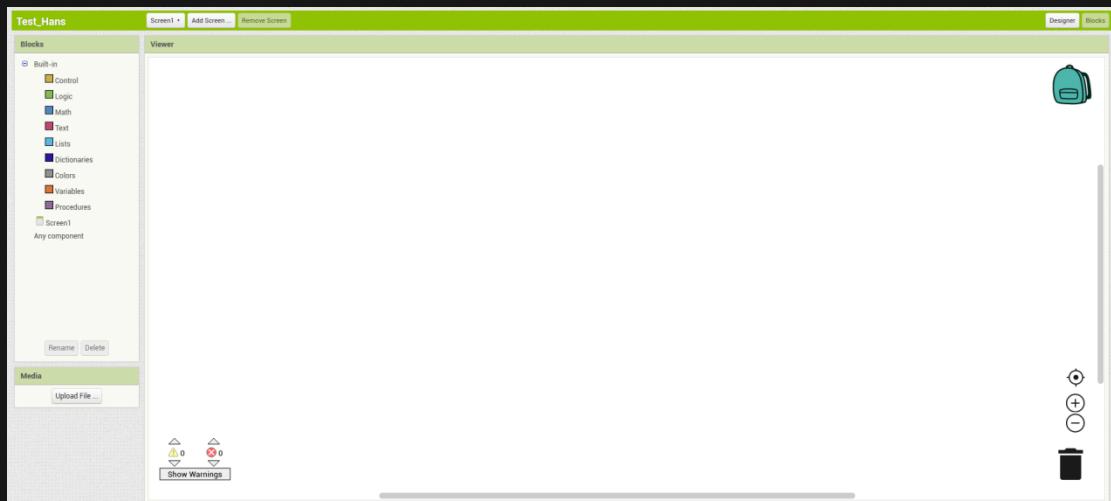
Designer digunakan untuk mendesain tampilan aplikasi dengan berbagai komponen dan layout yang disediakan sesuai dengan keinginan (halaman saat ini)

Blocks digunakan untuk berpindah ke area pemrograman block



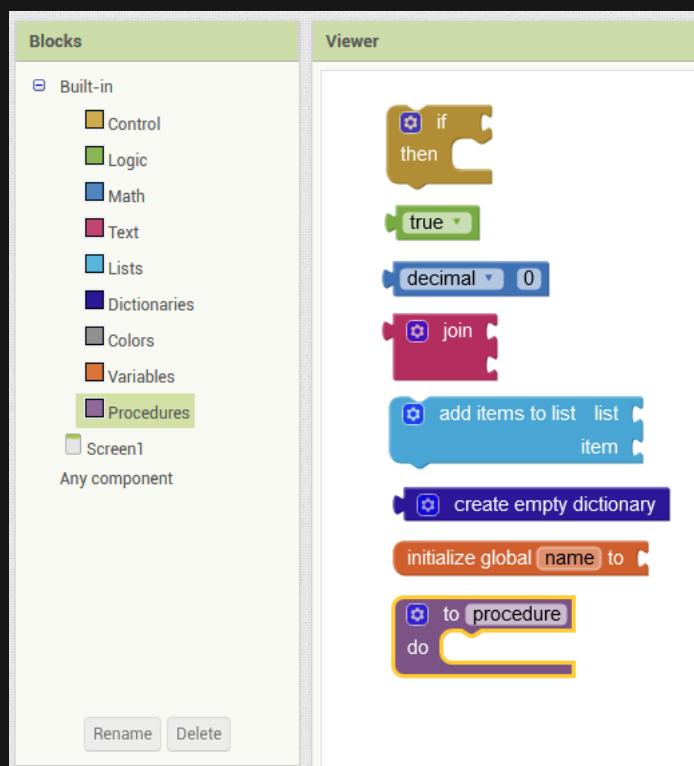
1.2 Blocks Area

Digunakan untuk melakukan event handling atau melakukan pemrograman pada setiap komponen yang sudah kita desain melalui area **Designer**



1.2.1 Blocks

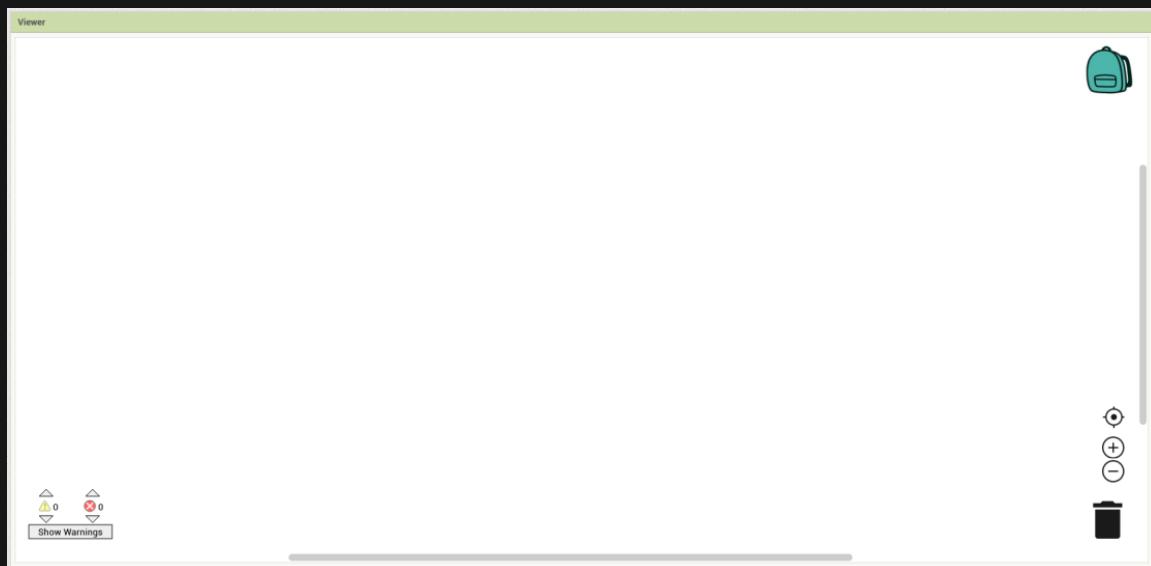
Merupakan komponen-komponen untuk melakukan event handling, komponen ini berbasis Google Blockly



iMECHATEUS 2021

1.2.2 Viewer

Berfungsi sebagai workspace atau area bekerja.

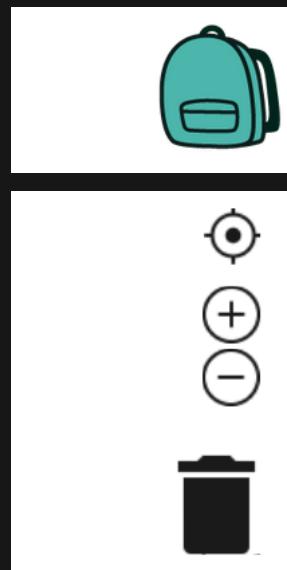


Backpack merupakan tools yang digunakan untuk melakukan copy-paste, namun dapat digunakan untuk copy-paste antar screen

Crosshair berguna untuk mengubah posisi **Block** berada ditengah **Viewer**

Zoom berguna untuk memperbesar dan memperkecil **Block** pada **Viewer**

Trash digunakan untuk menghapus **Block** yang tidak digunakan dengan car drag and drop **Block** ke **Trash**



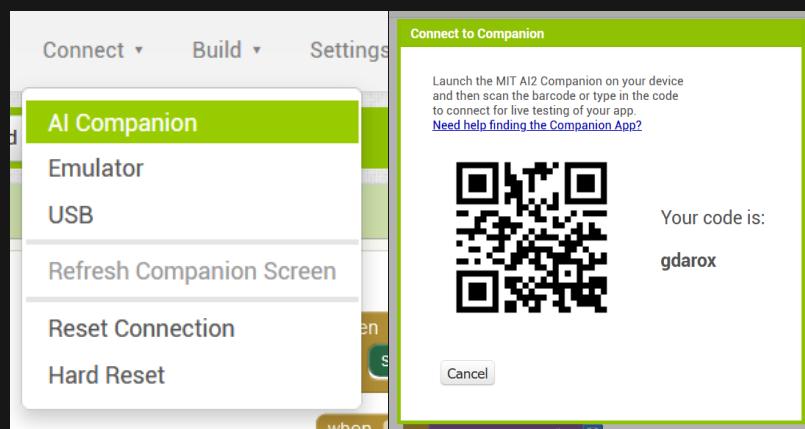
Emulator digunakan untuk pengujian terhadap aplikasi yang kita buat, dengan cara klik **Connect - Emulator**



2. Menghubungkan dengan perangkat

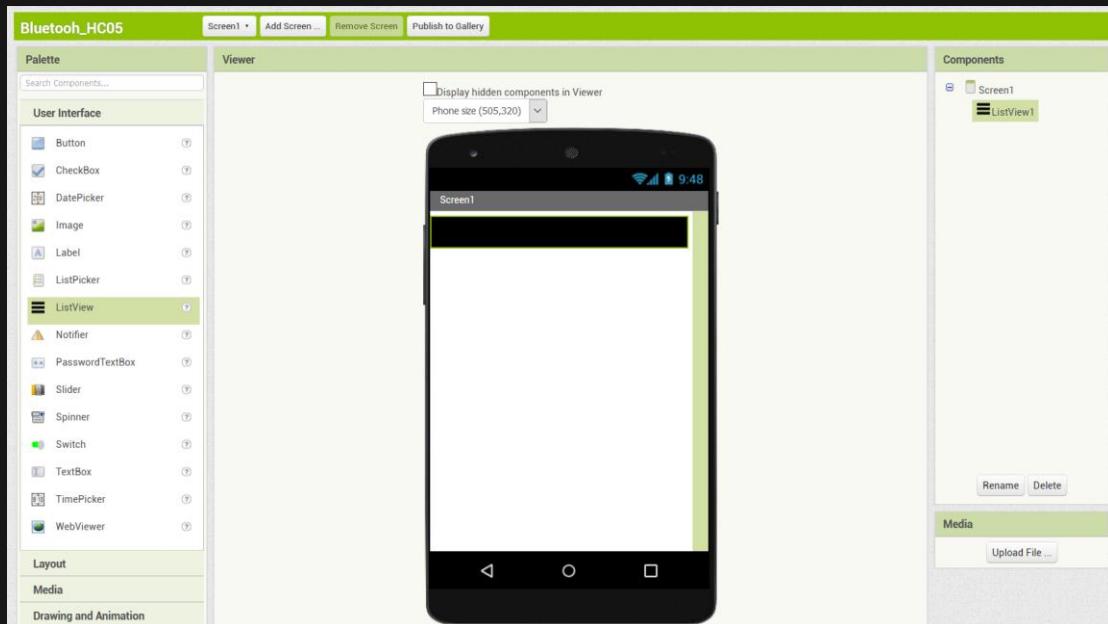
2.1 Menghubungkan dengan Smartphone

Klik tombol Connect, lalu pilih AI Companion, lalu scan barcode yang muncul / masukan code atau pilih emulator jika ingin menggunakan emulator

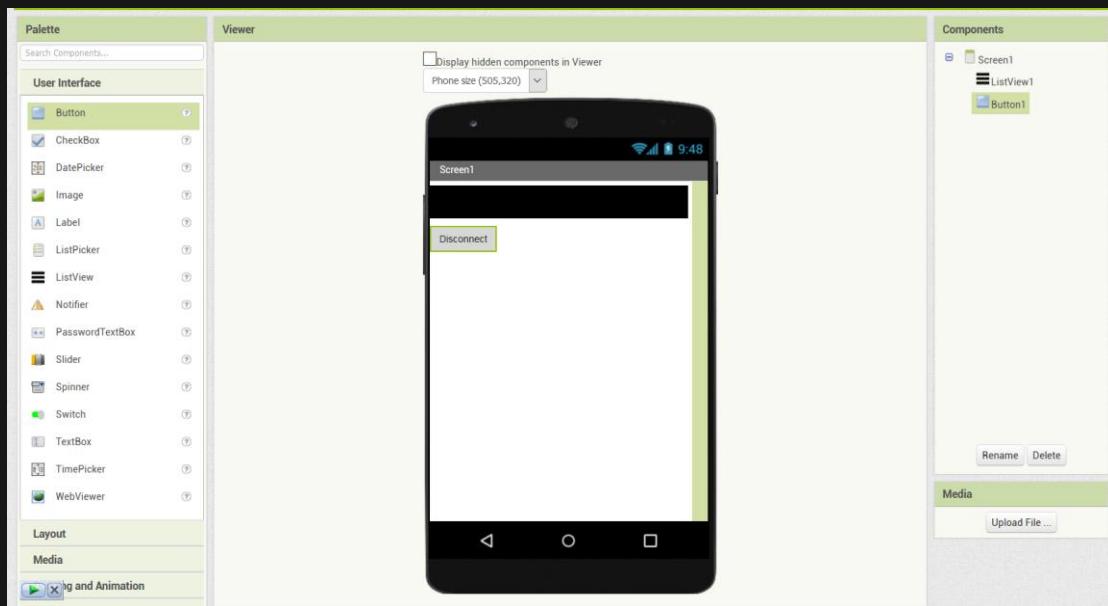


2.2 Menghubungkan dengan Arduino

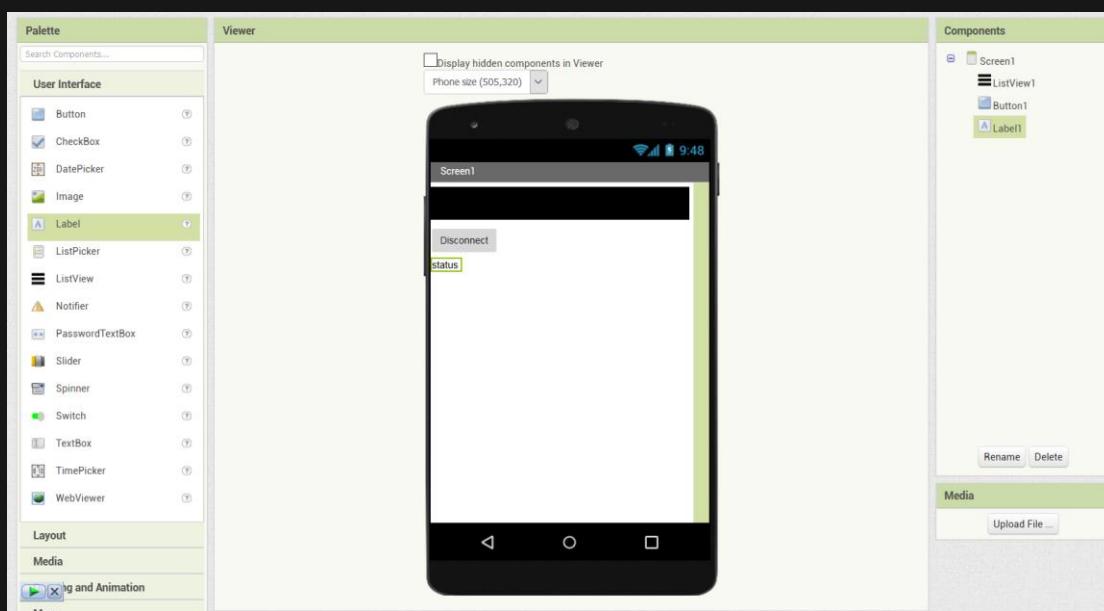
1. Masukkan ListView ke viewer



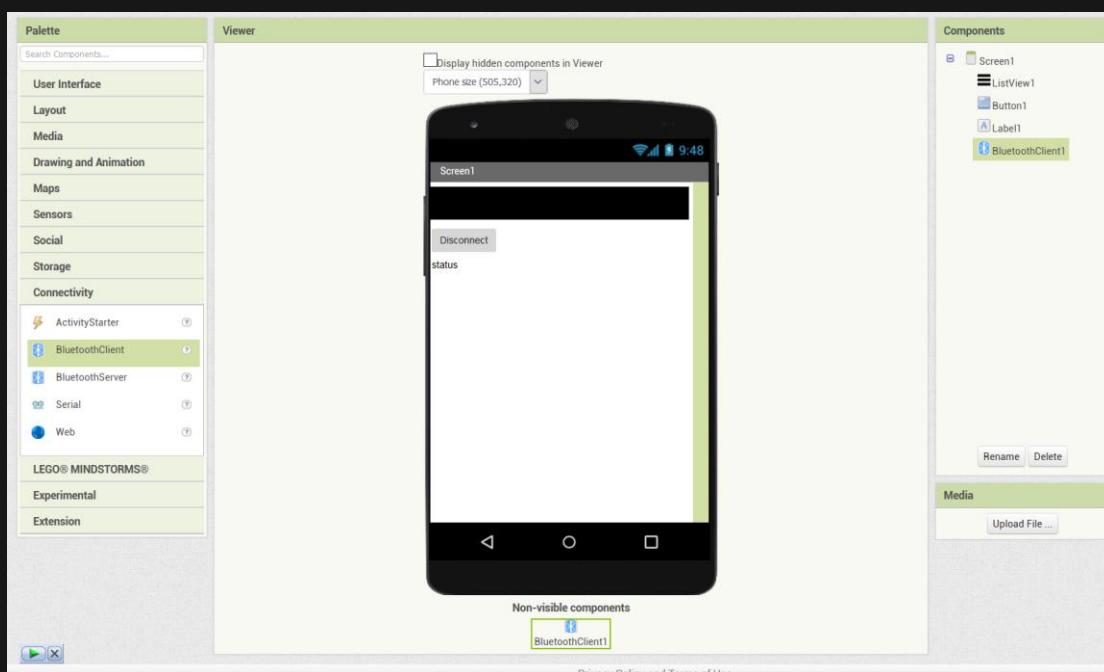
2. Masukkan Button ke viewer, lalu ubah menjadi “Disconnect”



3. Masukkan **Label** ke viewer dan ubah menjadi “Status”

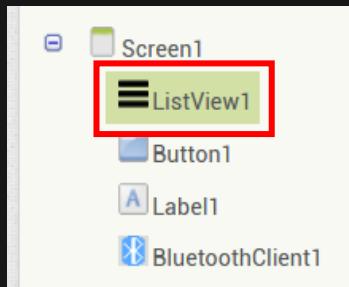


4. Masukkan **BluetoothClient1** ke viewer



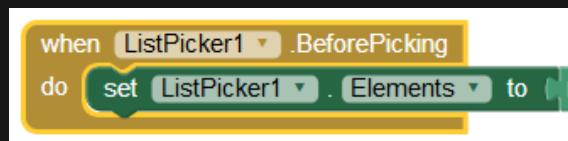
Kemudian pindah ke **Blocks**

1. Klik **ListView** pada **Screen1**

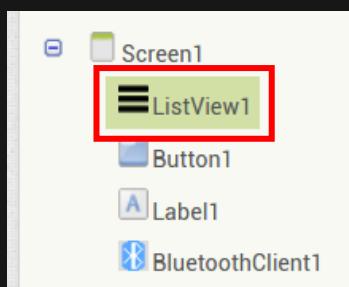


2. Masukkan *codeblock* berikut

Codeblock ini berfungsi untuk menentukan tugas ListView1 saat default

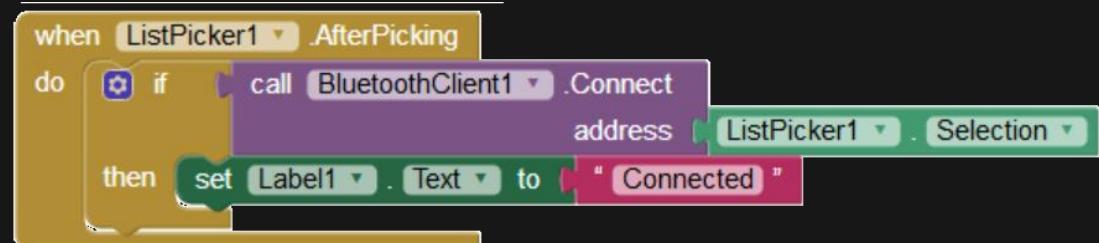


3. Klik **BluetoothClient1** pada **Screen1**, dan masukkan *codeblock* berikut



4. Masukkan *codeblock* berikut

Codeblock ini berfungsi untuk menentukan tugas ListView1 pada saat kita memilih Bluetooth yang akan disambungkan

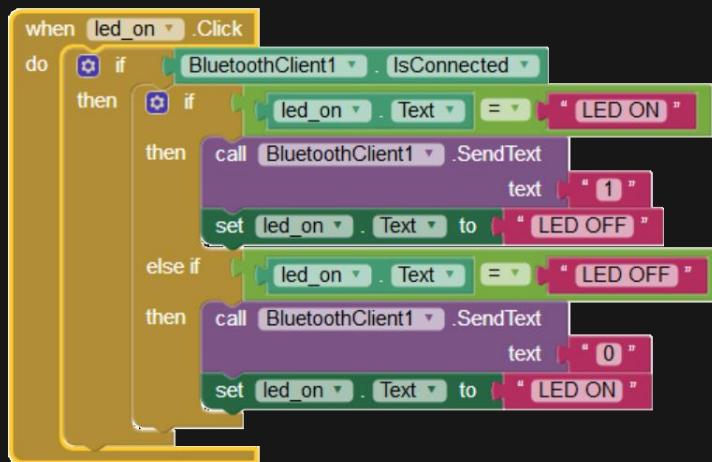


5. Masukkan codeblock berikut ke viewer.

Codeblock ini berfungsi pada saat kita memutuskan hubungan Bluetooth



Contoh penggunaan MIT AI2 dan sensor HC-05



3. Percobaan

3.1 Percobaan 1

Membuat aplikasi sederhana, apabila tombol “teknik?” ditekan, akan muncul kalimat “we are one”

Component;

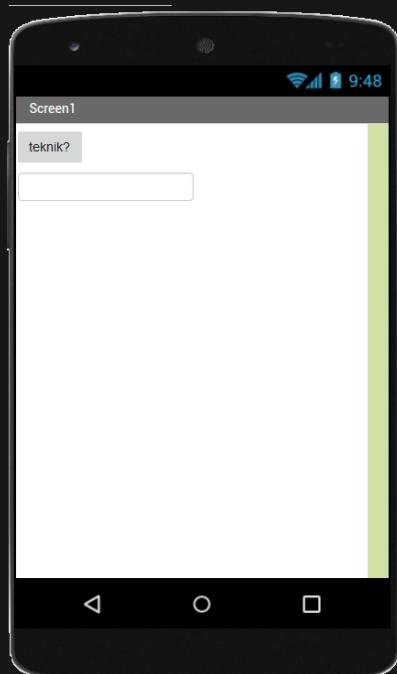
1. Button
2. TextBox

Block;

1. when *Button1* . Click
2. set *Button1* . Text to
3. “ ”

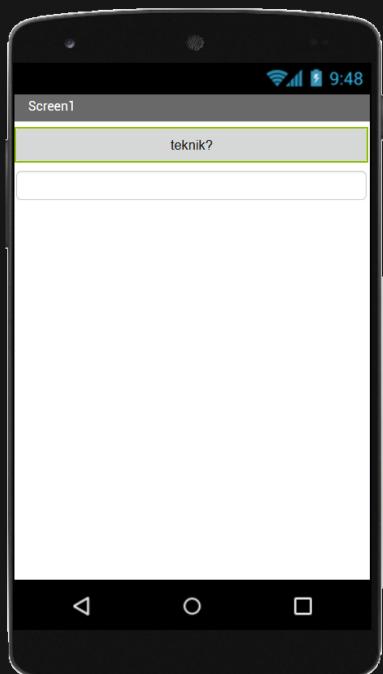
Desain Aplikasi

1. Drag and Drop **Button** ke viewer
2. Drag and Drop **TextBox** ke viewer



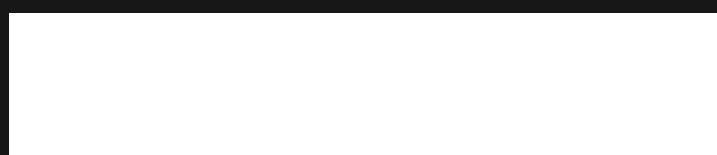
Merubah ukuran Content

1. Pilih Button1 pada Components
2. Ubah width menjadi *fill parent* untuk merubah ukuran menjadi sebesar ukuran perangkat, atau masukkan angka pada kolom *pixels* untuk merubah ukuran sesuai pixel yang diinginkan
3. Ubah teks menjadi “teknik?”
4. Lakukan hal yang sama pada TextBox



Kemudian pindah ke **Blocks**

1. Klik Button1 pada Screen1, drag blocks *when Button1 . Click*, dan drop ke viewer
2. Klik Button1 pada Screen1, drag and drop *set Button1 . Text to*, dan sambungkan ke *when Button1 . Click*
3. Klik Text pada Built-in, drag block “__”, drop ke viewer, dan sambungkan ke *set Button1 . Text to*
4. Ketik “we are one” pada block “__”



iMECHATEUS 2021

3.2 Percobaan 2

Membuat aplikasi untuk menggambar sederhana dengan 4 warna

Component

1. Button
2. Canvas
3. Horizontal Arrangement

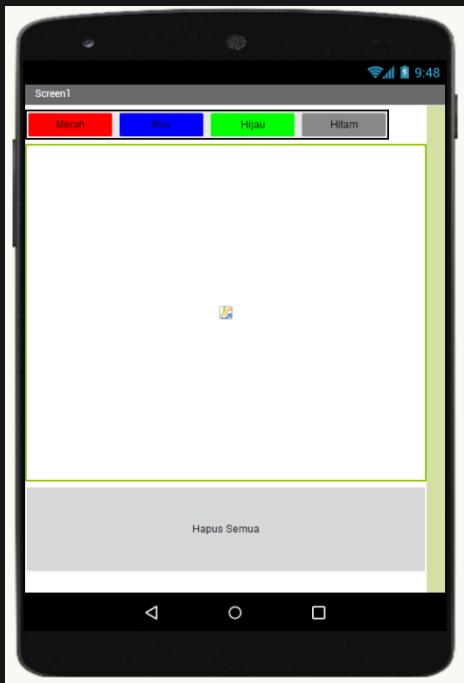
Blocks

1. when *Button* . Click
2. when *Canvas* . Touched
3. when *Canvas* . Dragged
4. set *Canvas* . *BackgroundColor* . to
5. call *Canvas* . Clear
6. call *Canvas* . DrawCircle
7. call *Canvas* . Dragged
8. call *Canvas* . DrawLine
9. get

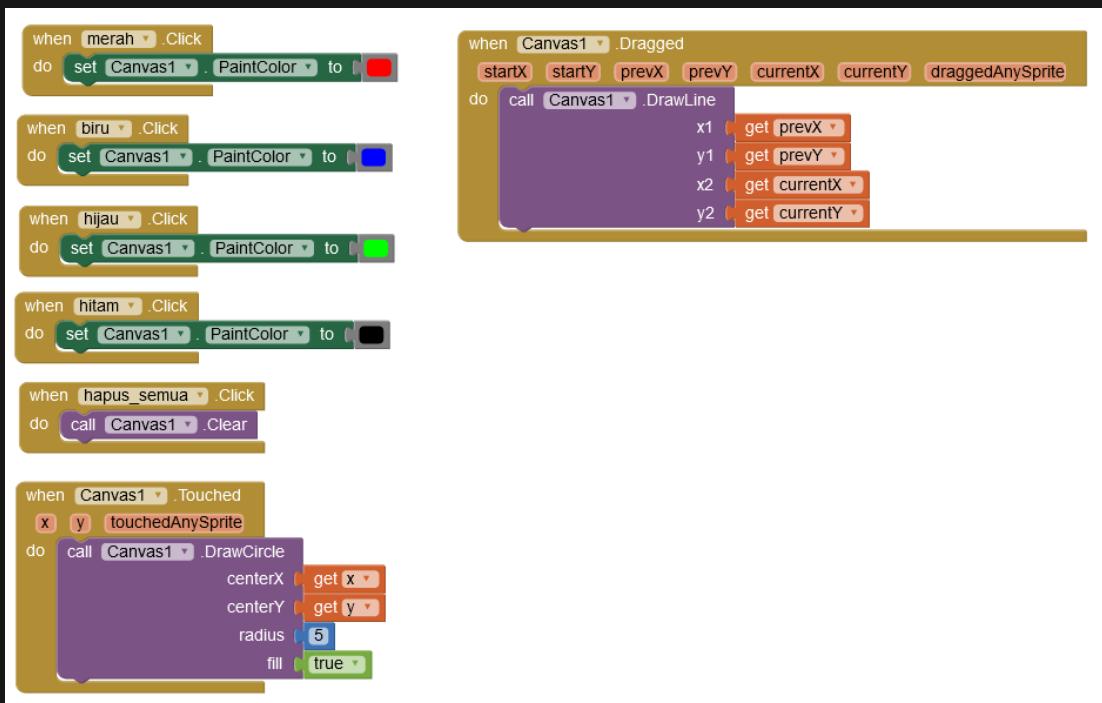
Desain Aplikasi

1. Drag and drop **Button** pada User Interface ke viewer
2. Ubah teks dan warna sesuai warna
3. Ubah ukuran button dan bentuk sesuai keinginan
4. Drag and drop **Horizontal Arrangement** pada Layout ke viewer
5. Drag and drop button yang tadi ke dalam kotak **Horizontal Arrangement**
agar button menjadi sebaris
6. Drag and drop **Canvas** pada Drawing and Animation ke viewer
7. Ubah ukuran canvas selebar perangkat pada viewer dan panjang sesuai keinginan
8. Drag and drop **Button** ke viewer, ubah teks menjadi “hapus semua”
9. Ubah ukuran button dan bentuk sesuai keinginan





Kemudian pindah ke Blocks



*Notes

Untuk set *Canvas1 . PaintColor* to, pilih set *Canvas1 . BackgroundColor* to, lalu ubah *BackgroundColor* menjadi *PaintColor*

Radius dapat dimasukkan angka sesuai besar titik yang diinginkan