**Permainan “*The Amazing Race*” Berbasis Arduino dengan Modul Bluetooth HC-05 beserta Aplikasi Android Menggunakan MIT App Inventor 2 (v.1.0)**

**Oleh : Saniyaayu Sekarpramesi**

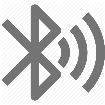
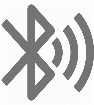
**Pendahuluan**

Terinspirasi dari *reality game show* populer bertajuk “*The Amazing Race*”*,* proyek ini memanfaatkan skalabilitas konsep *Internet of Things*—dimana konektivitas antar perangkat dapat terjadi secara *real-time* dan simultan. Terdapat dua jenis komponen utama untuk proyek ini, yaitu Checkpoint (*Arduino dengan sensor bluetooth yang berperan sebagai pemancar data permainan*), serta Player Device (*smartphone Android sebagai penerima data permainan*).

Dalam suatu contoh setting permainan, terdapat beberapa Checkpoint yang berisikan game berbeda-beda. Terdapat sebuah aplikasi utama yang terpasang dalam setiap Player Device yang memuat semua game. Hanya satu game yang dapat aktif dalam satu waktu pada Player Device tersebut, tergantung dengan checkpoint terdekat yang mengaktifkannya.

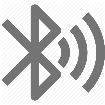
Berikut adalah ilustrasi arsitektur sederhana permainan lengkap ini :

Game5 : Aktif





Game3 : Aktif



*Gambar 1 : Checkpoint-checkpoint dengan game yang berbeda-beda*

**Tujuan**

Proyek ini menunjukkan komunikasi antara perangkat Android dan Arduino tanpa menggunakan koneksi internet (menggunakan bluetooth).

**Alat dan Komponen**

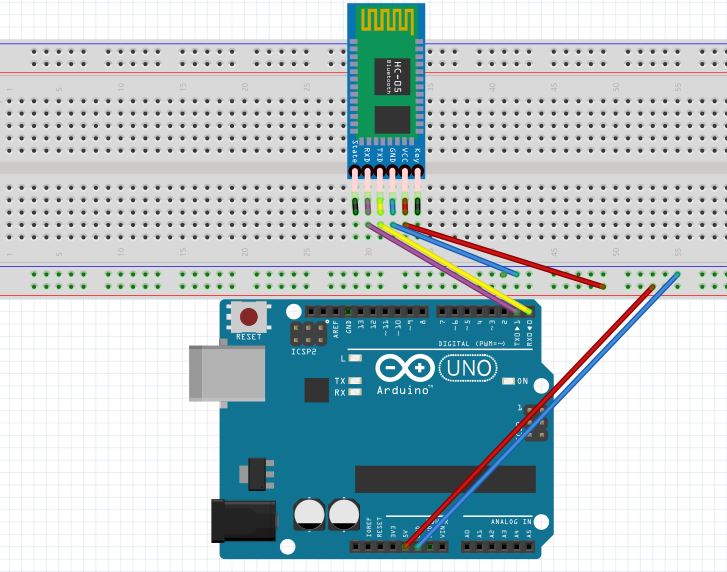
Checkpoint :

1. Arduino Uno (1x)
2. Kabel Jumper Arduino (6x)
3. Modul Bluetooth HC-05 (1x)
4. Kabel Daya Arduino (1x)
5. Breadboard (1x)

Player Device :

1. *Smartphone* bersistem operasi Android

**Penyusunan Komponen dan Perangkat Lunak**

1. **Rangkaian Listrik untuk Checkpoint**

*Gambar 2 : Rangkaian untuk Checkpoint*

Susunlah kabel dan komponen seperti diagram diatas. Hubungkan Pin GND Arduino pada pin kutub negatif (-) *breadboard*, sedangkan Pin 5V Arduino pada kutub positif(+) *breadboard*. Sesuai gambar, modul bluetooth HC-05 dipasangkan pada breadboard. Hubungkan kabel sesuai label yang tertera pada modul bluetooth—VCC dan GND mengikuti Arduino. Perlu diingat bahwa pin RXD(*receive*) pada modul bluetooth dipasangkan pada pin TXD(*transmit*) Arduino, dan sebaliknya pula.

1. **Arduino Sketch Sederhana untuk Checkpoint**

Arduino yang terpasang, selain berfungsi sebagai pemancar sinyal bluetooth juga berfungsi sebagai penentu ID game—mengingat bahwa banyaknya game yang direncanakan untuk keseluruhan permainan ini. Berikut adalah program sederhana untuk sebuah Checkpoint :

void setup()

{

Serial.begin(9600);

}

void loop() //Data akan dikirim secara berulang-ulang

{

int value = 1; // Merupakan ID game yang ingin kita

kirimkan (dapat diubah tergantung pada ID game yang kita inginkan)

Serial.println(value); //Mencetak nilai ID

delay(1000); //Jeda waktu pengiriman nilai

}

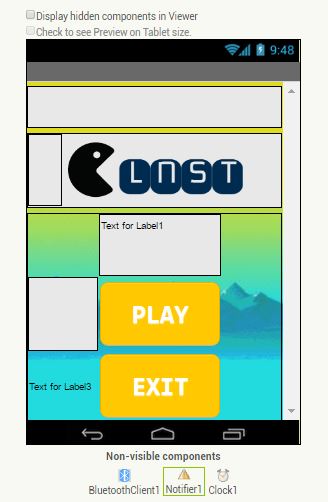
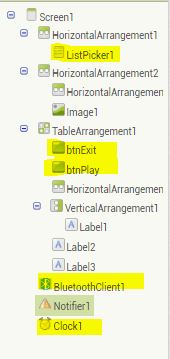
Untuk mengubah ID Game, ubahlah nilai *value*. Dalam contoh ini, nilai *value* yang akan kita gunakan adalah 1 atau 2.

1. **Membuat Aplikasi Utama untuk Player Device**

**Bagian *Designer***

Untuk aplikasi utama, kita akan menggunakan App Inventor 2 untuk membangun sebuah aplikasi android siap pakai. Sebagai contoh, kita asumsikan hanya terdapat 2 buah Checkpoint--berarti, hanya terdapat 2 buah game pada aplikasi utama. Siapkan 3 *screen* pada proyek App Inventor2; satu sebagai layar utama, sisanya untuk permainan yang ingin kita pakai. Tetapi sebagai contoh, hanya digunakan 1 buah Checkpoint. Adanya 2 buah game hanya untuk menunjukkan bahwa Checkpoint dapat diberi input game dengan nilai yang berbeda-beda[1].

Berikut adalah tampilan visual untuk layar utama :

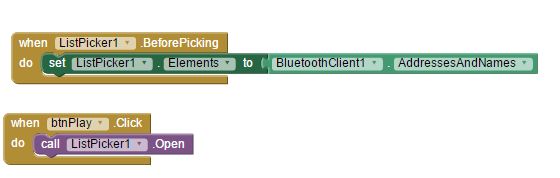


*Gambar 3.1(kiri) :Tampilan Visual Layar Utama dan Gambar 3.2(kanan): Komponen yang terdapat di layar utama.*

Untuk layar utama, komponen yang terpenting adalah komponen yang telah di *highlight* kuning pada Gambar 3.2, yaitu : *btnExit, btnPlay, BluetoothClient1,Clock1,* dan *ListPicker1*[2].

**Bagian *Blocks***

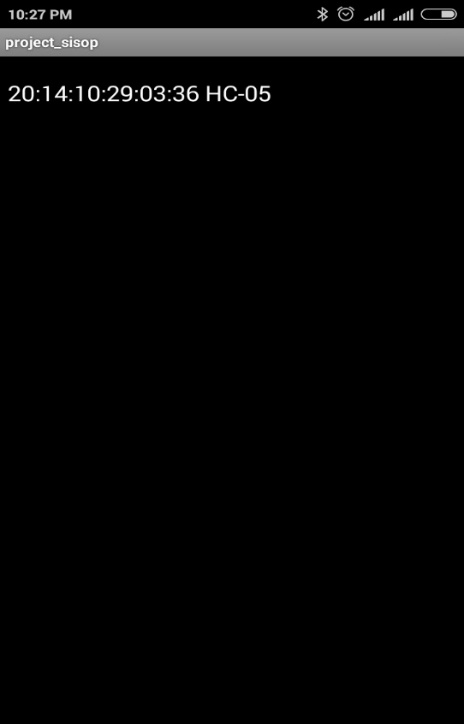
**Mengetahui *MAC Address*  Modul Bluetooth**

Setelah menyusun komponen, kita akan masuk ke bagian *blocks* untuk layar utama. Karena modul bluetooth yang kita gunakan untuk sebuah Checkpoint tidak akan berubah, kita memerlukan *MAC address* dari modul bluetooth tersebut agar Player Device pasti akan terhubung pada Checkpoint tersebut. Karena tidak ada informasi *MAC address* modul bluetooth tertera pada detail produk, maka kita perlu mengecek manual dengan bantuan *ListPicker1* dari komponen visual yang sudah kita bahas di awal.

*Gambar 3.3 : Pemanggilan ListPicker untuk mengetahui perangkat yang aktif beserta MAC Addressnya.*

Fungsi *when ListPicker1.BeforePicking* akan menyimpan alamat dan nama perangkat bluetooth yang aktif pada sebuah *list* yang kosong. *ListPicker1* akan dipanggil ketika tombol *Play* ditekan.

Setelah di *build* dan terpasang di perangkat android, tekan tombol Play dan daftar perangkat bluetooth yang aktif akan muncul.

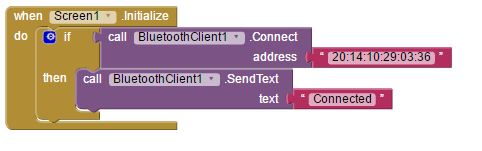


*Gambar 3.4 : Daftar perangkat bluetooth yang aktif*

Sekarang kita tahu *MAC address* untuk modul bluetooth kita, yaitu “20:14:10:29:03:36”. Hapuslah seluruh *blocks* sebelumnya yang berjudul fungsi *when ListPicker1.Before Picking* serta *when btnPlay.Click* yang berfungsi untuk menyimpan data pada *ListPicker1* serta fungsi pemanggilannya.

**Menyetel Aplikasi Agar Langsung Tersambung Pada Modul Bluetooth**

Kita tentunya tidak ingin repot-repot memilih perangkat seperti pada *ListPicker*  sebelumnya. Maka itu, kita harus menyetel *MAC Address* secara *hard-coded* (paten, tidak dapat diubah) pada aplikasi. Pada kasus ini, kita hanya menggunakan satu modul bluetooth—maka hanya satu *MAC Address* yang dipasang dalam aplikasi ini. Konfigurasi ini memungkinkan Player Device untuk langsung terhubung pada Checkpoint [3].



*Gambar 3.5 : Konfigurasi untuk setelan otomatis pada modul bluetooth*

Fungsi ini akan langsung terpanggil disaat aplikasi dimulai. Aplikasi akan mencocokkan *MAC Address*, dan akan langsung memberikan notifikasi “*Connected*”ketika berhasil terhubung.

**Pemanggilan Game Berdasarkan ID**

Berhubung banyaknya Checkpoint dan Game dalam permainan ini yang direncanakan, maka diperlukan sebuah ID untuk membedakan game yang akan dipanggil sesuai dengan Checkpointnya. Arduino akan mengirimkan sinyal berulang-ulang (lihat bagian Arduino Sketch untuk Checkpoint), maka itu kita memerlukan komponen *Clock1*  untuk terus menerus menerima data.

Pertama, kita perlu membuat variable untuk menampung ID game, yang akan kita sebut *gameMode*.



*Gambar 3.6 : Inisialisasi variabel gameMode untuk menampung ID game.*



*Gambar 3.7 : Komponen Clock1 untuk memroses data secara iteratif.*

Kemudian, fungsi *Timer* pada komponen *Clock1* akan memeriksa apakah bluetooth sudah terkoneksi. Bila terkoneksi, maka data yang dikirimkan oleh Arduino lewat bluetooth akan dipanggil dan diterima. Data yang diterima akan disimpan pada variabel *gameMode*.

Selanjutnya, bila ingin mulai bermain, kita tentu ingin langsung terhubung pada permainan yang sesuai dengan Checkpoint.



*Gambar 3.8 : Fungsi untuk tombol Play*

Ketika tombol *Play* ditekan saat aplikasi berjalan, maka langsung terjadi pengecekan terhadap ID game yang dipasang pada checkpoint. Pada contoh diatas, untuk *gameMode* bernilai 1, maka akan memulai game 1 yang dibuat pada *Screen 3*[2]—untuk *gameMode* bernilai 2 maka akan memulai game2 dibuat pada *Screen4*.

1. **Koneksi Arduino dan Android**

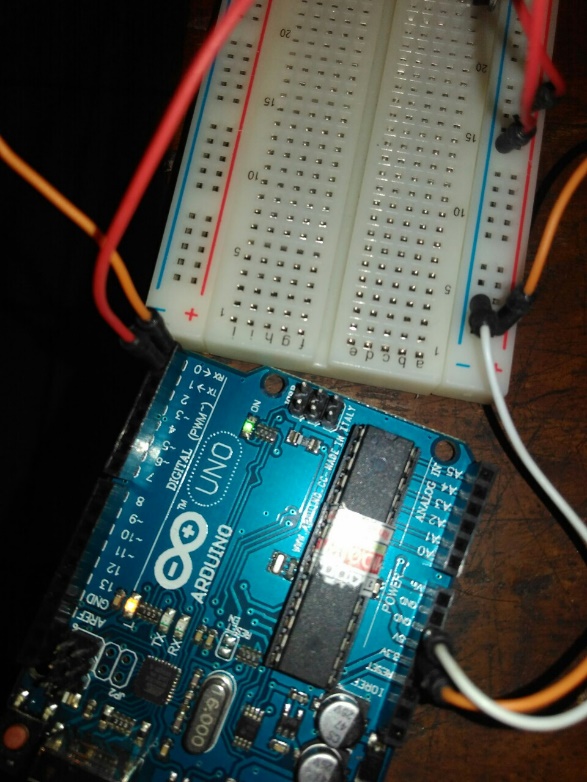
Setelah rangkaian dan aplikasi terpasang, kita siap untuk menjalankan permainan. Hubungkan sambungan daya Arduino pada laptop/komputer, jalankan aplikasi Arduino yang berisi *sketch* sederhana yang sudah dijelaskan pada bagian sebelumnya.

Perlu diingat bahwa sebelum mengunggah *sketch*, kabel Jumper untuk RX dan TX harus dalam kondisi dilepas. Bila kabel RX dan TX sudah terpasang pada Arduino saat *sketch* diunggah, maka Arduino akan mencoba untuk mengunggah data—yang kondisi ini tidak mungkin karena Arduino masih dalam keadaan kosong.



*Gambar 4.1 : Saat mengunggah, jangan lupa dilepas dahulu kabel untuk TX dan RX.*

Setelah selesai mengunggah *sketch*, pasang kabel untuk TX dan RX. Ingat, kabel TX pada bluetooth dipasang pada RX Arduino, dan sebaliknya.



*Gambar 4.2 : Kabel terpasang setelah mengunggah*

Setelah selesai, jalankan aplikasi pada perangkat Android, dan game akan muncul sesuai nilai *value* yang kita inisialisasi pada *sketch* Arduino.

**Kesimpulan dan Catatan**

Karena ini merupakan versi awal(v.1.0) dari pengembangan proyek ini, maka ada beberapa kekurangan yang direncanakan untuk diperbaiki dan dikembangkan oleh penulis.

[1] Saat panduan ini ditulis, percobaan mengaktifkan dua Checkpoint secara simultan belum dilakukan. Selain itu, pengaktifan Checkpoint masih bergantung pada koneksi laptop untuk sumber daya serta pengunggahan *sketch* Arduino.

[2] Panduan ini menitikberatkan pada koneksi antar Android dan Arduino melalui bluetooth, maka desain tidak perlu mengikuti panduan.

[3] Memang setelah *MAC Address* disetel secara manual pada aplikasi, perangkat Android dapat langsung terhubung. Namun, diperlukan *pairing device* secara manual oleh perangkat Android masing-masing. Ini dapat menyulitkan ketika Checkpoint yang terdapat dalam permainan berjumlah banyak.

[4] Untuk saat ini, hanya satu perangkat Android yang dapat terhubung pada suatu Checkpoint dalam satu waktu. Ini diakibatkan oleh keterbatasan modul bluetooth HC-05.

Semua *source code* dan *file* untuk proyek ini dapat dilihat dan diunduh lewat : <https://github.com/sekarpramesi/amazing-race-internet-of-things>

Penulis dapat dihubungi via e-mail : [niyasekar@gmail.com](mailto:niyasekar@gmail.com)