

# Pembuatan Aplikasi Pembelajaran Matematika untuk Android *Mobile* dengan Komunikasi *Device-Server*

Janice Elvina Fausta Leuw<sup>1</sup>, Justinus Andjarwirawan<sup>2</sup>, Adi Wibowo<sup>3</sup>  
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra  
Jl. Siwalankerto 121 – 131 Surabaya 60236  
Telp. (031) – 2983455, Fax. (031) - 8417658  
E-mail: jc\_lowlietelvinaleuw@yahoo.com<sup>1</sup>, justin@petra.ac.id<sup>2</sup>, adiw@petra.ac.id<sup>3</sup>

**ABSTRAK:** Belakangan ini, perkembangan Android semakin meningkat di pasaran. Pengguna android berasal dari berbagai kalangan. Tidak hanya orang dewasa saja yang mengenal *gadget* ini, melainkan anak-anak pun senang memainkan *gadget* ini. Adanya teknologi yang berkembang seperti ini menjadikan anak-anak semakin malas untuk belajar, mereka lebih memilih menghabiskan waktunya untuk bermain dengan *gadget* mereka.

Oleh karena itu, pada penelitian ini dibuat sebuah aplikasi yang dapat membantu pembelajaran anak-anak melalui *device* Android, khususnya pada pelajaran Matematika. Dengan adanya aplikasi ini, anak-anak akan diajak belajar sekaligus bermain, sehingga belajar akan menjadi lebih menarik bagi anak-anak. Aplikasi yang digunakan untuk mengimplementasikan penelitian ini yaitu aplikasi *Eclipse*. Pada aplikasi juga diterapkan penggunaan *Google Cloud Messaging* untuk membantu komunikasi antara dua *device* saat *multiplayer game* dimainkan.

Hasil pengujian aplikasi menunjukkan bahwa proses penggunaan aplikasi membutuhkan koneksi *internet* yang kuat, terutama saat mengimplementasikan *multiplayer game*. Aplikasi ini nantinya dapat dikembangkan sehingga dapat menambah pengetahuan anak-anak terhadap pelajaran lain selain matematika.

**Kata kunci:** *Game*, Matematika, *Android Device*, *Google Cloud Messaging*

**ABSTRACT:** Recently, Android development is increasing in the market. Android users come from any class. Not only adults who know these *gadgets*, but the kids were happy to play these *gadgets*. The existence of these emerging technologies make increasingly lazy kids to learn, they prefer to spend time playing with their *gadgets*.

Therefore, the final project is an application that can help children to learn through the Android device, especially in mathematics. With this application, children will be invited to learn and play, so that learning will be more attractive to them. Applications that are used to implement this thesis, namely *Eclipse*. The application also use *Google Cloud Messaging* to aid communication between two devices when playing *multiplayer games*.

The test results showed that the application requires a strong internet connection, especially when implement a *multiplayer game*. This application can be developed further to increase the children knowledge besides mathematics.

**Keywords:** *Math*, *Games*, *Android Device*, *Google Cloud Messaging*

## 1. PENDAHULUAN

Perkembangan teknologi handphone (HP) yang berbasis GSM (*Global System For Mobile*) sekarang ini sangat pesat. *Handphone* tidak hanya digunakan sebagai alat komunikasi saja, tetapi juga sebagai alat untuk memudahkan pengguna dalam kehidupan sehari-hari. Fasilitas yang terdapat pada HP antara lain: pengaksesan internet, e-mail, *organizer*, *game* dan sebagainya dapat langsung digunakan.

Wyatt Slooper (1999) mengemukakan bahwa dalam proses belajar, pelajar belajar: 10% dari apa yang dibaca, 20% dari apa yang didengar, 30% dari apa yang dilihat, 50% dari apa yang dilihat dan didengar, 70% dari apa yang dikatakan, 90% dari apa yang dikatakan dan dilakukan. Anak-anak pada umumnya lebih susah menerima pelajaran dibandingkan orang dewasa, mereka cenderung lebih senang bermain dari pada belajar. Sehingga banyak upaya yang dilakukan agar anak-anak (pelajar/siswa) senang belajar melalui inovasi-inovasi baru dalam pendidikan khususnya dalam inovasi pembelajaran di luar sekolah. Salah satunya adalah aplikasi *mobile game* untuk edukasi yang berbasis android.

Untuk studi kasus pada penelitian ini di angkat salah satu mata pelajaran dari Sekolah Dasar yaitu mata pelajaran Matematika. Dimana aplikasi ini mengajarkan penjumlahan, perkalian, dan pengurangan. Didalam penelitian ini dirancang dan dibuat suatu pemodelan dari perhitungan matematika tersebut ke dalam suatu aplikasi yang dikemas dalam bentuk permainan (*game*), yang berbasis *mobile phone*. *Edu game* ini dibangun dengan menggunakan JAVA STANDARD EDITION (Java SE), yang menghasilkan aplikasi *game* untuk Android Mobile.

## 2. LANDASAN TEORI

### 2.1 Pengertian Android

Android adalah sistem operasi yang berbasis linux untuk mobile device misalnya smartphones dan computer tablet, yang dikembangkan oleh google dalam hubungannya dengan Open Handset Alliance. Android smartphone pertama yang terjual pada tahun 2009, dan sejak itu android berkembang menjadi sistem operasi smartphone terbesar.

Android pada awalnya dikembangkan oleh Android Inc, yang didukung secara finansial oleh google dan kemudian dibeli pada tahun 2005. Android mulai didistribusikan pada tahun 2007

melalui pengumuman oleh pendiri Open Handset Alliance. Open Handset Alliance adalah sebuah konsorsium yang terdiri dari 86 perusahaan *hardware*, *software* dan telekomunikasi yang ditujukan untuk memajukan open standards untuk mobile device [1]. Google merilis kode Android sebagai open-source di bawah lisensi *Apache*. Android Open Source Project (AOSP) bertanggung jawab atas pemeliharaan dan pengembangan lebih lanjut dari Android. Google dan beberapa anggota dari Open Handset Alliance berkolaborasi atas pengembangan dan perilsan Android. Android OS terdiri dari 12 juta baris kode yang ditulis dalam C / C++ / Java dan XML. [2]

## 2.2 Accelerometer

*Accelerometer* adalah sebuah sensor yang dapat mengukur percepatan, mendeteksi, dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi. Sensor *accelerometer* mengukur percepatan dari 3 sumbu gerakan akibat gerakan benda yang melekat padanya. Pengukuran percepatan selalu di pengaruhi oleh gaya grafitasi bumi :  $a_d = -g - \sum F/m$ . Dimana,  $a_d$  merupakan percepatan yang di terapkan pada *device*/perangkat,  $g$  adalah gaya grafitasi,  $F$  adalah gaya yang bekerja pada perangkat, dan  $m$  adalah massa perangkat. Tanda  $\sum$  merupakan jumlah dari  $x$ -,  $y$ - dan  $Z$ axis.

*Accelerometer* memiliki 3 axis pada sumbu X,Y dan Z. Dengan adanya tiga axis *accelerometer* ini memungkinkan perubahan arah layar Android *device* ataupun objek yang ada di dalam *device* secara otomatis, menjadi *vertical* maupun *horizontal* sesuai dengan arah layar tersebut digerakkan. [3] [4]

## 2.3 Google Cloud Messaging

*Google Cloud Messaging* untuk android adalah layanan yang memungkinkan pengiriman data dari *server* ke perangkat Android. Data berupa pesan yang disampaikan melalui notifikasi pada *device*. Layanan ini menyediakan mekanisme yang sederhana, dan ringan bahwa *server* dapat berhubungan secara langsung dengan *device*. Layanan GCM menangani semua aspek antrian pesan dan mengirimkan aplikasi pada target perangkat Android. Berikut adalah karakteristik utama dari *Google Cloud Messaging* (GCM) :

- GCM memungkinkan pihak ke-3, yaitu aplikasi *server* untuk mengirim pesan ke aplikasi Android.
- Sebuah aplikasi Android pada perangkat Android tidak perlu berjalan (*running*) untuk menerima pesan. Sistem akan membangunkan aplikasi Android melalui siaran *Intent* bila pesan tiba, asalkan aplikasi Android telah mengaktifkan *permission* untuk *broadcast receiver*.
- GCM tidak menyediakan *user interface* atau penanganan lainnya untuk data pesan. GCM hanya menyampaikan data pesan mentah yang diterimanya langsung ke aplikasi Android.
- GCM membutuhkan perangkat Android 2.2 atau perangkat lain yang lebih tinggi yang juga memiliki aplikasi *Google Play Store*, atau sebuah emulator yang menjalankan Android 2.2 dengan *Google API*
- GCM menggunakan koneksi yang ada untuk layanan Google. Untuk perangkat pre-3.0, hal ini mengharuskan pengguna untuk mengatur akun Google pada perangkat *device*-nya.[5]

## 3. DESAIN SISTEM

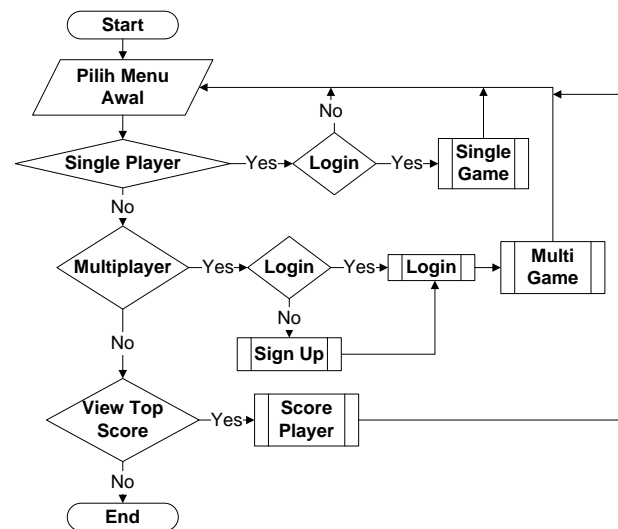
### 3.1 Desain Interkoneksi Client-Server Android

Sebelum membuat sistem jalannya aplikasi pembelajaran, koneksi *client server* pada Android perlu di desain, agar alur jalannya *game* dapat diimplementasikan secara optimal. Android membutuhkan operator *network* untuk menghubungkannya dengan internet, setelah terhubung dengan internet barulah android dapat berkomunikasi dengan suatu *server*. Pada aplikasi ini, Android bertindak sebagai *Client*.

### 3.2 Flowchart

#### 3.2.1 Desain Umum Sistem Client

Pada saat aplikasi dijalankan, terdapat halaman awal yang menampilkan beberapa menu pilihan, yaitu menu *Single player*, menu *Multiplayer* dan menu *Score*. Saat memilih menu *single player*, maka *user* akan diminta untuk melakukan *login* terlebih dahulu. Data *login* dari *user* berguna untuk menyimpan data *player* saat *player* akan meninggalkan aplikasi. Setelah melakukan *login* barulah *user* dapat masuk pada halaman *single game* dan dapat langsung memainkan *game* di dalamnya. Apabila memilih menu *multiplayer*, maka *user* akan masuk pada halaman *login* terlebih dahulu sebelum memilih lawan mainnya. Apabila *user* belum memiliki *account* yang terdaftar, maka *user* harus melakukan *sign up* terlebih dahulu. Menu *View Score* pada menu ini menampilkan *score player* yang paling tinggi ke rendah berdasarkan harian, mingguan, bulanan dan tahunan. Desain umum sistem *client* dapat pada Gambar 1.

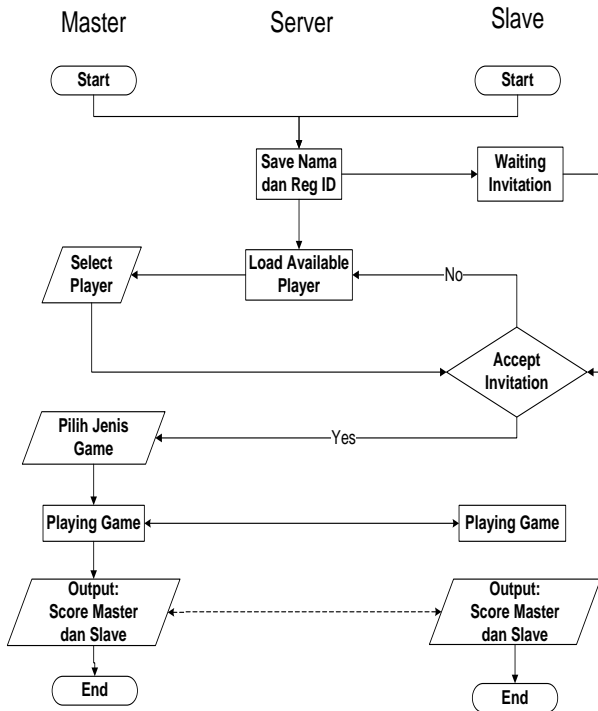


Gambar 1. Flowchart Client Secara Umum

#### 3.2.2 Desain Sistem Multiplayer Game

*Multiplayer game* memiliki dua *client* yang saling berkomunikasi dalam permainannya, dimana salah satu *Client* merupakan *master*, atau *player* yang memulai suatu permainan dengan meng-*invite* lawan mainnya, sedangkan *Client* satunya merupakan *player* yang menerima *invite* dari *Client* lain yang biasa disebut dengan *slave*. Pada ada Gambar 2 dapat dilihat proses bekerjanya *master* dan *slave* dengan melibatkan *server*. Sebelum memulai permainan *multiplayer user* harus memutuskan

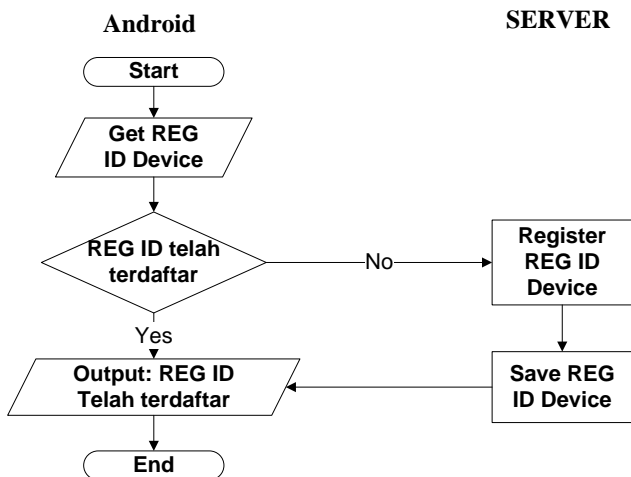
terlebih dahulu ingin menjadi *Master* yang melakukan *invite* atau *Slave* yang menerima *invite*.



Gambar 2. Document Flow Master-Slave Pada Multiplayer Game

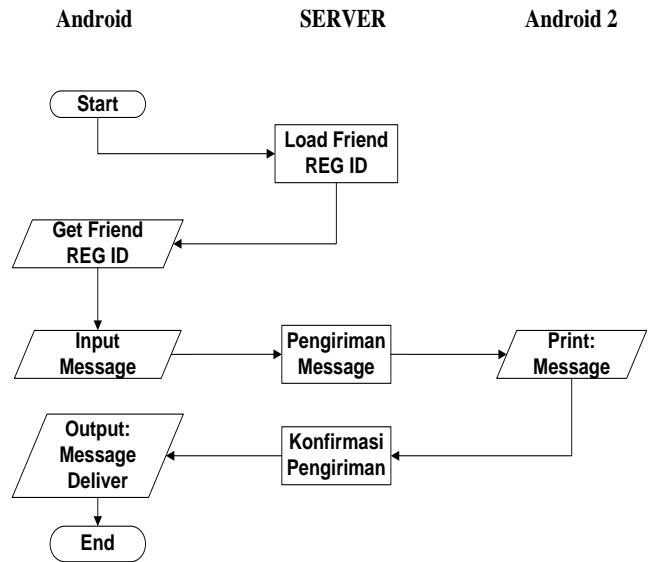
### 3.2.3 Desain Sistem Google Cloud Messaging

Google Cloud Messaging digunakan untuk membangun komunikasi antara dua *device* saat *multiplayer game* dimainkan. Agar dapat menggunakan Google Cloud Messaging, setiap *device* harus melakukan registrasi ke *server* GCM terlebih dahulu. Pada Gambar 3 dapat dilihat proses registrasi *device* terhadap GCM, dan pada Gambar 4 dapat dilihat proses bekerjanya GCM untuk berkomunikasi.



Gambar 3. Flowchart Registrasi GCM

Data *device* yang telah ter-registrasi akan akan disimpan pada *database* GCM sehingga untuk berkomunikasi satu sama lain hanya perlu me-load data *device* dari *server* GCM. Data yang disimpan pada *database* yaitu berupa *reg id* dari tiap *device*.

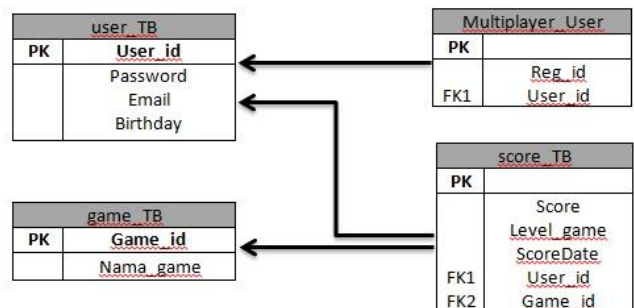


Gambar 4. Flowchart GCM Send Message

Sebelum mengirimkan pesan, *device* harus tahu terlebih dahulu *reg id* dari *device* tujuannya. Pada Gambar 4 dapat dilihat Android *device* 1 me-load dan mengambil *reg id* dari Android *device* 2. Setelah mengetahui *reg id* dari *device* tujuannya, Android *device* 1 dapat mengirimkan pesan kepada Android *device* 2.

### 3.3 Desain Database

Dalam pembuatan aplikasi pembelajaran ini, digunakan *database* untuk menyimpan data *player* berupa, data individu dan *score* yang diperoleh dari permainan. Aplikasi menggunakan *database* MySQL. Desain *database* dapat dilihat pada Gambar 5.



Gambar 5. Desain Tabel Penyimpanan Data Player

## 4. PENGUJIAN SISTEM

### 4.1 Spesifikasi Perangkat Pengujian

Pengujian aplikasi ini menggunakan 4 perangkat keras. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

1. Samsung *Galaxy Tab P1000*
  - a. *Dimensions* 190.1 x 120.5 x 12 mm (7.48 x 4.74 x 0.47 in)
  - b. *Size* 600 x 1024 pixels, 7.0 inches
  - c. Android OS, 2.2 (*Froyo*)
2. Samsung *Galaxy Tab 27.0 P3100*
  - a. *Dimensions* 193.7 x 122.4 x 10.5 mm (7.63 x 4.82 x 0.41 in)
  - b. *Size* 600 x 1024 pixels, 7.0 inches
  - c. Android OS, 4.0.4 (*Ice Cream Sandwich*)
3. Samsung *Galaxy Note N7000*
  - a. *Dimensions* 146.9 x 83 x 9.7 mm (5.78 x 3.27 x 0.38 in)
  - b. *Size* 800 x 1280 pixels, 5.3 inches
  - c. Android OS, 4.0.4 (*Ice Cream Sandwich*)
4. Sony *Xperia U*
  - a. *Dimensions* 112 x 54 x 12 mm (4.41 x 2.13 x 0.47 in)
  - b. *Size* 480 x 854 pixels, 3.5 inches
  - c. Android OS, 4.0.4 (*Ice Cream Sandwich*)

### 4.2 Pengujian Single Player Game

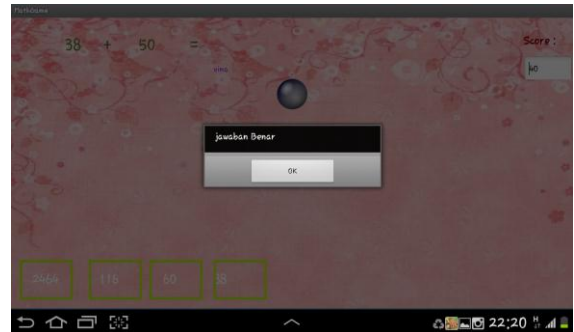
Pada *single game* terdapat tiga jenis permainan, yaitu *Drag and Drop Game*, *Accelerometer Game*, dan *Mini Game*.

#### 4.2.1 Pengujian Terhadap *Drag And Drop Game*



Gambar 6. Melakukan *Drag Bola*

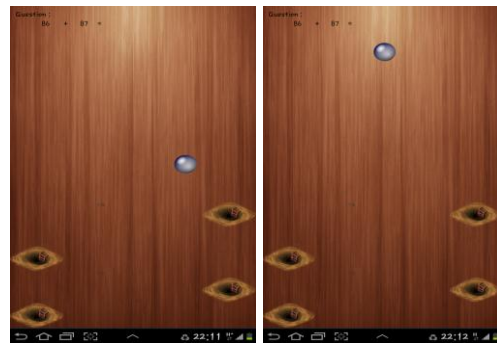
Saat bola di-*drop* pada bingkai jawaban yang benar, maka akan muncul *message dialog* seperti pada Gambar 7, kemudian *score player* akan otomatis bertambah. Jika bola di-*drop* pada bingkai jawaban yang salah maka akan muncul *message dialog* seperti pada Gambar 8.



Gambar 7. Jawaban *Drag And Drop* Benar

#### 4.2.2 Pengujian Terhadap *Accelerometer Game*

Pada *Accelerometer Game* terdapat soal dan lubang-lubang jawaban seperti yang terlihat pada Gambar 8. Untuk menjawab pertanyaan yang terdapat pada *game*, *user* harus memasukkan bola dengan menggelindingkannya ke lubang yang benar. Apabila bola masuk ke lubang yang benar, maka *player* akan mendapatkan *score* yang secara otomatis akan diinput kedalam *database*. Sebaliknya jika bola masuk ke lubang yang salah, maka *player* tidak akan mendapatkan *score*.

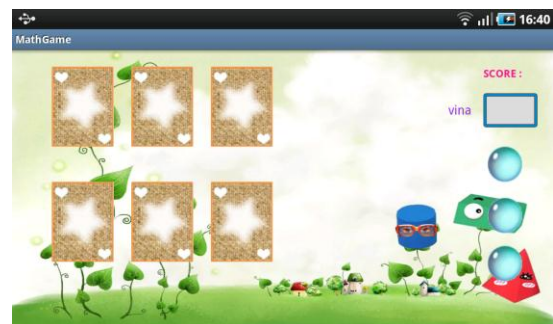


Gambar 8. Tampilan *Accelerometer Game*

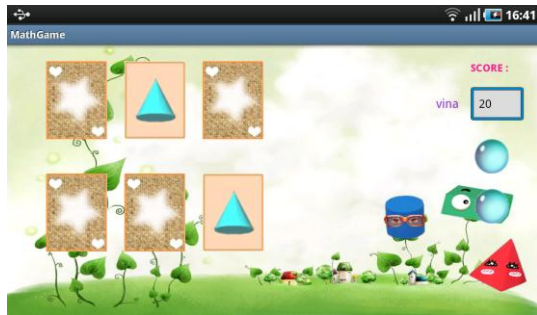
kartu akan tertutup kembali serta kesempatan menebak *player* akan berkurang.

#### 4.2.3 Pengujian Terhadap *Mini Game*

*Mini Game* merupakan permainan tambahan dari aplikasi pembelajaran ini. Cara memainkan *game* ini yaitu dengan menebak pasangan dari masing-masing kartu. Seperti yang terlihat pada Gambar 9 dan Gambar 10, terdapat tiga bola biru di sebelah kanan layar *game*. Bola-bola ini merupakan *live* atau kesempatan *player* untuk menebak kartu. Untuk membuka kartu-kartu yang tertutup, *player* harus men-*touch* kartu tersebut. Apabila kartu yang terbuka merupakan dua gambar yang berbeda, maka kedua



Gambar 9. Tampilan *Mini Game*



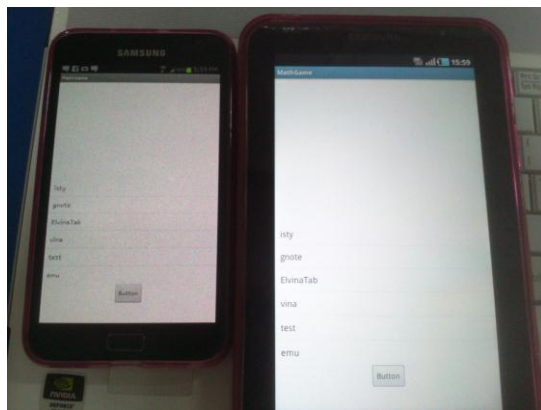
Gambar 10. Tampilan Mini Game Saat Kartu Terbuka

### 4.3 Pengujian Multiplayer Game

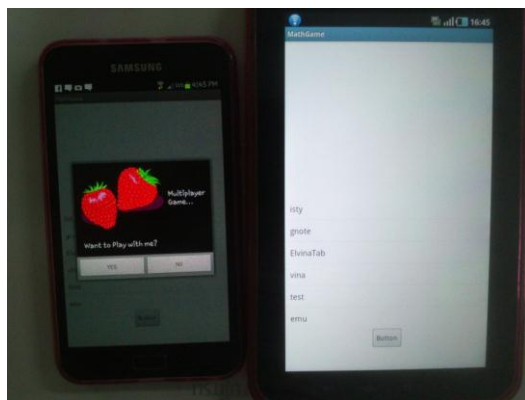
Pada permainan *multiplayer game* terdapat dua jenis *game* yang disediakan yaitu, *Drag and Drop Game* dan *Mini Game*. Untuk memulai permainan pada *multiplayer game*, *user* harus meng-*invite* lawan mainnya terlebih dahulu.

#### 4.3.1 Pengujian Invite Player

Setelah *user* berhasil melakukan *login* pada *multiplayer game*, *user* akan masuk pada halaman *list player*. Setelah masuk pada halaman ini, *user* harus memutuskan ingin menjadi *master* atau *slave*. Apabila ingin menjadi *master*, *user* harus meng-*invite* lawan mainnya untuk memulai permainannya, dan jika *user* memutuskan ingin menjadi *slave*, maka *user* hanya perlu diam dan menunggu *invite* dari *master*.



Gambar 11. List Available Player



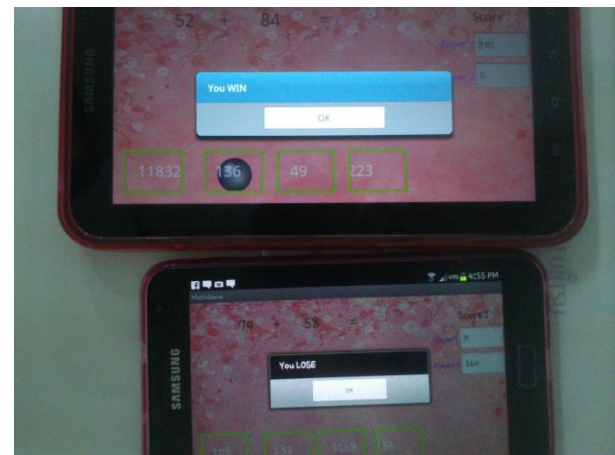
Gambar 12. Invite Player

#### 4.3.2 Pengujian Drag And Drop Game Multiplayer

Pada *multiplayer game*, *player* dapat melihat *score*-nya sendiri dan juga *score* dari lawan mainnya. Soal matematika yang ditampilkan pada kedua *device* sama, sehingga tingkat kesulitan dari kedua *player* setara.



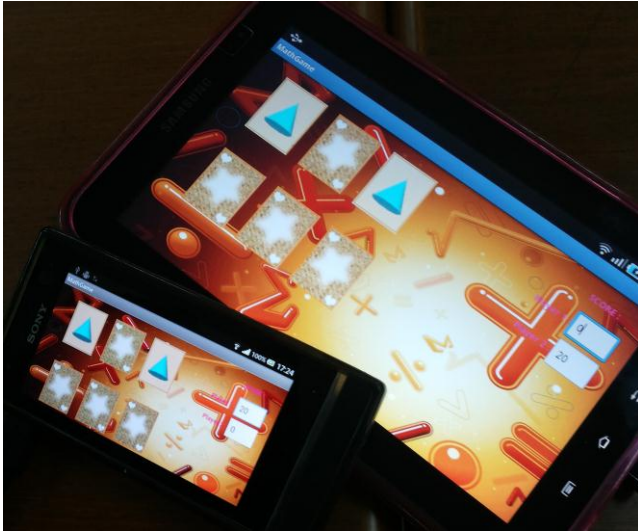
Gambar 13. Tampilan Drag And Drop Multiplayer Game



Gambar 14. Game Drag And Drop Multiplayer Berakhir

#### 4.3.3 Pengujian Mini Game Multiplayer

*Mini Game Multiplayer* dimainkan dengan cara *turn base* atau secara bergantian antara *player*. Setelah giliran *player* 1 berakhir, maka *player* 2 akan mendapatkan pemberitahuan berupa *message dialog* bahwa sekarang merupakan gilirannya bermain. Sebaliknya, jika giliran *player* 2 telah berakhir, maka giliran main akan berganti pada *player* 1. Kartu yang telah dibuka pada *turn* sebelumnya, akan dapat dilihat oleh masing-masing *player*. Dapat dilihat proses permainan *Multi Game* pada Gambar 15.



Gambar 15. Tampilan Mini Game Multiplayer

player yang di-*invite* tidak akan menerima *request* yang telah dikirim.

## 6. DAFTAR PUSTAKA

- [1] DiMarzio, Jerome (2008). *Android A Programmer's Guide*
- [2] Deitel, Paul; Deitel, Harvey; Deitel, Abbey; Morgano, Michael (2012). *Android for Programmers And App-Driven Approach*
- [3] NVIDIA (2010). *Tegra Android Accelerometer Whitepaper.*
- [4] Chen, Gang (2011). *How to Convert 3-Axis Direction and Swap X-Y Axis of Accelerometer Data within Android™ Driver.*
- [5] Nerieri, Francesco (2012). *Cloud Messaging in Android.*

## 4.4 Pengujian Operator Network

Untuk menggunakan aplikasi ini, dibutuhkan koneksi *internet* yang kuat, terutama saat memainkan *multiplayer game*. Koneksi *internet* yang kuat akan memperlancar komunikasi antar *device*. Oleh karena itu dilakukan pengujian terhadap operator *network*.

Dari pengujian yang telah dilakukan, hasil yang diperoleh menunjukkan bahwa dengan menggunakan jaringan Wifi UK.Petra, komunikasi antar *device* lebih lancar.

Operator	Jumlah Percobaan	Percobaan yang Berhasil
XL	10	5
Simpati	10	6
Axis	10	8
Wifi UK.Petra	10	10

Tabel 1. Pengujian Operator

## 5. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan pada aplikasi, dapat disimpulkan beberapa hal sebagai berikut:

1. Pengiriman pesan melalui GCM sebagai pusat pertukaran data antar *master* dan *slave* berhasil dilakukan.
2. Untuk menggunakan aplikasi ini sangat dibutuhkan adanya koneksi *internet* yang kuat, terutama saat mengaplikasikan *Multiplayer Game*, karena *multiplayer game* melibatkan *Google Cloud Messaging* (GCM) dalam komunikasi antar *device*. Kedua *device* yang saling berhubungan harus sama-sama memiliki koneksi *internet* yang baik, sehingga komunikasi datanya dapat berjalan dengan lancar.
3. Proses *invite player* pada *Multiplayer Game* harus berlangsung di dalam halaman *List Player*. Jika tidak, maka