

# **KONTROLER GAME ANDROID BERBASIS *EVENT INJECTION***

Karya tulis

Disusun untuk Memenuhi Tugas Akhir  
sebagai Siswa SMA Pangudi Luhur Van Lith



Disusun oleh:

Reinard Rahardian A.S.

15163797

**SEKOLAH MENENGAH ATAS PANGUDI LUHUR VAN LITH**

**MUNTILAN**

**2017**

## **HALAMAN PENGESAHAN**

### **Kontroler Game Android Berbasis Event Injection**

Karya tulis

Disusun oleh:

Reinard Rahardian A.S.

15163797

Telah dipertahankan di depan panitia penguji

Pada tanggal \_\_ November 2017

Nama Lengkap:

	Penguji	Tanda tangan
Penguji 1	: Sr. Giovanni. CB	_____
Penguji 2	: Br. Agustinus Suparno. FIC	_____

Muntilan, \_\_ November 2017

Kepala sekolah

Br. Agustinus Giwal Santoso, FIC, MM.

No. G 11.353

## **PERNYATAAN KEASLIAN**

Saya menyatakan dengan sesungguhnya, bahwa karya tulis yang saya buat sebagai tugas akhir menjadi siswa SMA Pangudi Luhur Van Lith seluruhnya merupakan hasil karya saya sendiri. Adapun bagian-bagian tertentu dalam penulisan karya tulis yang saya kutip dari hasil karya orang lain telah dituliskan secara jelas dan sesuai dengan norma, kaidah serta etika akademis. Apabila dikemudian hari ditemukan seluruh atau sebagian karya tulis ini bukan karya saya sendiri atau adanya plagiat dalam bagian-bagian tertentu, saya bersedia menerima sanksi-sanksi dari SMA Pangudi Luhur Van Lith.

Muntilan, \_\_ November 2017

Yang membuat pernyataan,

Reinard Rahardian A.S.

## HALAMAN MOTTO

I hate programming

I hate programming

I hate programming

It works!

I love programming!

When talent works hard,

It beats everything.

First, solve the problem.

Then, Write the code.

With great power comes great responsibility.

Ikan hidup berenang melawan arus.

Jangan mau ikut arus, itu artinya sudah mati.

## HALAMAN PERSEMBAHAN

Karya tulis berjudul Kontroler Game Android Berbasis Event Injection ini penulis persembahkan untuk:

- Untuk keluarga ada dirumah:

Papa dan Mama: Yang sering terlewat dikabari, tapi selalu ada dan memberi dukungan.

Rafael: Si kecil yang jalanya didepan masih jauh.

- Untuk keluarga yang tidak ada dirumah:

Reuben: Yang lagi terdampar buat kuliah, tapi selalu memberi dukungan moral.

- Untuk keluarga yang tidak berasal dari rumah:

Keluarga kecilku di Van Lith: Yang bersedia menjadi sepenanggungan. Dalam segala jenis suka dan duka, termasuk mengerjakan karya tulis

- Pembimbing:

Bruder Agus: Yang bersedia memberi dispensasi terlambat sampai 5 hari, serta membimbing dalam pembuatan karya tulis.

- Semua pihak yang telah mendukung penulis dalam proses pembuatan

## KATA PENGANTAR

Puji syukur penulis haturkan pada Tuhan yang Maha Kuasa sebab hanya oleh kasih-karuniaNya, penulis dapan menyelesaikan karya tulis ini dengan baik. Karya tulis ini berjudul Kontroler Game Android Berbasis Event Injection. Judul ini dipilih dalam rangka mengikuti perkembangan teknologi yang sekarang gencar.

Dalam pembuatan karya tulis ini, penulis menemui banyak kesulitan dan halangan. Tapi melalui dukungan keluarga dan teman-teman penulis, penulis akhirnya mampu melewati segala kesulitan tersebut. Tanpa dukungan mereka, mustahil karya tulis ini dapat dibuat. Khususnya penulis berterima kasih kepada:

- Keluarga penulis, yaitu orangtua dan kakak-adik penulis
- Br. Agustinus Suparno selaku pembimbing, yang telah mendukung dan membimbing pembuatan karya tulis
- Semua teman-teman sepenanggungan di Van Lith
- Serta semua orang yang tidak dapat penulis sebut satu-persatu

Akhirnya, penulis hendak berterimakasih kepada semua orang yang bersedia meluangkan waktunya membaca karya tulis ini. Penulis berharap karya tulis ini dapat berguna.

Penulis sadar bahwa penulisan karya tulis ini masih sangat jauh dari sempurna, maka penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Muntilan, 7 November 2017

Penulis

## ABSTRAK

*Rahardian Andykaputra Setiadji, Reinard. 2017. Kontroler Game Android Berbasis Event Injection. Karya Tulis. SMA Pangudi Luhur Van Lith.*

*Videogame* telah menjadi sebuah cabang perkembangan teknologi informasi, segera setelah teknologi itu sendiri berkembang, diawali oleh *Tennis for Two*. Kini, gaming, atau aktivitas bermain game, bahkan telah menjadi suatu gaya hidup tersendiri. Terdapat berbagai konsol yang memungkinkan sebuah game untuk dijalankan, salah satunya adalah smartphone. Masalahnya, konsol smartphone tidak memberikan kebebasan kontrol seluas konsol lain karena keterbatasan jari yang bisa mengakses kontrol.

Dalam usaha mengatasi masalah tersebut, maka penelitian ini diadakan. Digunakan prinsip *event injection* dimana pada layar smartphone diberikan suatu injeksi interaksi pengguna berupa tekanan pada layar di koordinat yang telah ditentukan, beserta suatu alat khusus yang memberikan tambahan 3 tombol. Kode yang bertanggungjawab terhadap injeksi tersebut disertakan melalui modifikasi berkas pemasang aplikasi dari aplikasi yang hendak digunakan.

Dari hasil penelitian dan rekayasa, ditemukan bahwa penggunaan alat tersebut meningkatkan performa permainan, yaitu dengan memberikan situasi permainan yang mendekati situasi dengan adanya penggunaan kontroler *Playstation*.

## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
PERNYATAAN KEASLIAN.....	ii
HALAMAN MOTTO.....	iii
HALAMAN PERSEMBAHAN.....	iv
KATA PENGANTAR.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	vii
PENDUHLULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	2
C. Pembatasan masalah.....	2
D. Tujuan penelitian.....	2
E. Manfaat penelitian.....	3
DASAR TEORI.....	4
A. Gaming.....	4
B. Perangkat Game.....	5
C. Smartphone.....	6
D. Java.....	6
E. Kode Sumber.....	6
F. Fungsi (Subroutine).....	7
G. Android Studio.....	7
H. Apktool.....	7
I. Gradle.....	8
J. ART/Android Runtime.....	9
K. Smali.....	9
L. Touchscreen.....	9
M. Android OS.....	9



N. Service.....	10
O. Activity.....	10
P. Bluetooth.....	10
Q. Bluetooth-Library.....	11
R. Android Input Events.....	11
S. Android Instrumentation.....	11
T. Komunikasi serial.....	11
U. Arduino Nano.....	12
V. HC-05 Bluetooth Module.....	12
METODOLOGI PENELITIAN.....	13
A. Jenis penelitian.....	13
B. Kelengkapan penelitian.....	13
C. Metode penelitian.....	13
HASIL PENELITIAN.....	17
KESIMPULAN DAN SARAN.....	19
A. Kesimpulan.....	19
B. Saran.....	19
DAFTAR PUSTAKA.....	20
LAMPIRAN.....	22
A. Dokumentasi.....	22
B. Kode sumber untuk unit kontroler.....	32
C. Kode sumber untuk penerima <i>bluetooth</i> .....	32
BIOGRAFI PENULIS.....	37

# BAB I

## PENDUHLUAN

### A. Latar Belakang

*Videogame* telah menjadi sebuah cabang perkembangan teknologi informasi, segera setelah teknologi itu sendiri berkembang, diawali oleh *Tennis for Two* (Flatow, 1993) hingga kini, gaming, atau aktivitas bermain game, bahkan telah menjadi suatu gaya hidup tersendiri.

Dalam perkembangannya telah banyak muncul genre game, seperti *First Person Shooter (FPS)*, *Third Person Shooter (TPS)*, *Role-Play Game (RPG)*, *Real-Time Strategy (RTS)* *Multiplayer Online Battle Arena (MOBA)*, *Indie*, dan sebagainya. *Console* gaming pun sangat beragam, mulai dari *console* terdedikasi seperti Xbox dan Playstation, hingga yang tersedia sebagai program tambahan dari gawai seperti laptop dan smartphone. Akan tetapi, tidak semua *console* memungkinkan kebebasan kontrol yang sama. Kontroler untuk komputer tersedia dalam varian yang dikhususkan untuk gaming, bahkan dijual mouse gaming dengan 20 tombol (Logitech G700) yang setiap tombolnya dapat diprogram.

Dalam kasus ini, *console* smartphone termasuk tidak fleksibel. Seorang pemain hanya akan dapat memakai 2 tombol (jempol kiri dan kanan). Tidak jarang sebuah game butuh lebih dari itu, misalnya genre FPS butuh minimal 3, yaitu joystick untuk bergerak, satu lagi untuk mengarahkan pandangan, serta satu lagi untuk menyerang (tembak), sehingga diperlukan peralatan tambahan. Kontroler tambahan untuk *Android* sebenarnya telah

beredar luas, tetapi akan membutuhkan dukungan terprogram dari game yang hendak dimainkan. Melalui penelitian ini akan dibuat suatu kontroler tambahan yang bisa bekerja dengan semua game, melalui metode event injection.

#### **B. Rumusan Masalah**

- a. Bagaimana membuat suatu alat periferan gaming yang dapat mengatasi masalah jeda reaksi pada game Android?
- b. Seberapa besar dampak yang akan ditimbulkan dari penggunaan alat yang akan dibuat?

#### **C. Pembatasan masalah**

Penelitian ini hanya akan difokuskan pada pembuatan periferan gaming dengan spesifikasi:

- a. Ringan
- b. Terjangkau
- c. Hanya memberi tambahan 3 tombol, yaitu tombol L1, R1, dan R2 seperti yang ada pada kontroler Playstation dan Xbox
- d. Sistem kerja touch event injection
- e. Khusus *console* smartphone berbasis *Android*

#### **D. Tujuan penelitian**

Penelitian ini ditujukan untuk pembuatan sebuah periferan gaming untuk *console Android* yang bisa kompatibel dengan sebagian game, terutama game dengan genre *First Person Shooter*.

#### **E. Manfaat penelitian**

Alat yang dihasilkan dari penelitian ini akan bermanfaat untuk menghilangkan jeda reaksi yang disebabkan harus memindah jari dari satu kontrol ke kontrol lainya terutama dalam game tertentu yang tidak memberikan dukungan terprogram untuk kontroler terdedikasi.

## **BAB II**

### **DASAR TEORI**

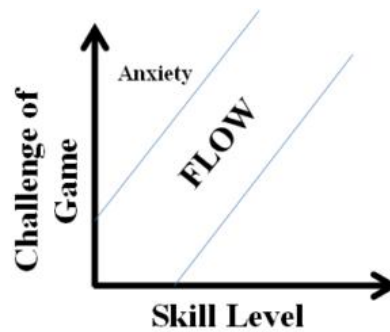
#### **A. Gaming**

“Gaming is the playing of games for stakes.” (US Legal Inc., 2016). Sekarang ini, gaming sudah menjadi sesuatu yang lebih dari sekedar sebuah hiburan belaka. Sekarang ini gaming telah berubah menjadi suatu gaya hidup, dan suatu bagian yang sulit dipisahkan terutama dari anak muda. Bahkan ada yang telah menjadikannya mata pencaharian, memanfaatkan acara yang mirip turnamen olahraga. Beberapa tim gaming profesional yaitu *Natus Vincere* dan *Team Liquid (DotA)*.

Ada 3 faktor yang menyebabkan seseorang terus-menerus ingin bermain game. Yang pertama adalah pencapaian (Achievement), yaitu keinginan untuk mencapai suatu prestasi dalam game. Yang kedua adalah sosial (Social), yaitu keinginan untuk bersosialisasi dengan orang lain dalam game melalui komunikasi dan kerjasama tim. Ketiga, faktor pendalaman (Immersion) yang merupakan keinginan untuk kabur dari dunia nyata, dan hidup sebagai karakter buatan didalam game (Nick, 2007).

Menurut Ryan dan Deci, perilaku manusia dipengaruhi oleh kebutuhan akan kompetensi, otonomi dan relevansi. Ketiga hal ini ada dalam game. Kompetensi dicapai melalui “Flow” yaitu kecocokan antara kemampuan dan tantangan. Bila tantangan lebih mudah dari kemampuan akan membuat bosan, sedangkan bila lebih sulit akan membuat frustrasi. Otonomi hadir melalui kontrol penuh pemain atas game yang dimainkan. Dan

relevansi terpenuhi melalui interaksi dan komunitas pemain, dimana selama masih ada orang lain yang memainkan, game akan terasa relevan.



## B. Perangkat Game

*“Gaming device is any device which is used directly or indirectly in conducting a game. It is designed for playing a game for money or a prize. Gaming device is used solely for the purposes of amusement or entertainment or gambling. Gaming device can be mechanical, electromechanical or electronic equipment.”* (US Legal Inc., 2016) Perangkat game adalah alat apapun yang digunakan secara langsung maupun tidak dalam menjalankan sebuah game, untuk mendapat uang ataupun hadiah, dan hanya untuk kegunaan pemuasan kesenangan, hiburan ataupun berjudi. Perangkat game bisa bersifat mekanikal, elektromekanikal, maupun elektronik.

Beberapa contoh kontroler terdedikasi atau khusus memainkan game adalah kontroller *PS* dan *Xbox*. Kontroler lain umumnya bukan merupakan kontroler khusus memainkan game, melainkan perangkat umum yang dapat digunakan untuk memainkan game, misalnya keyboard dan mouse, walaupun perangkat tersebut juga banyak yang dirancang khusus memainkan game.

Contohnya adalah tambahan tombol di mouse yang dapat diprogram, jarak antar tombol keyboard yang diperpendek, dan penyalinan belakang keyboard (*Backlight*).

### **C. Smartphone**

Tidak ada batas resmi tentang telepon genggam apa yang dapat dikategorikan sebagai smartphone, karena ini merupakan kosakata tidak resmi. Tapi pada umumnya publik menganggap smartphone memiliki fitur yang mendekati sebuah komputer, memiliki fungsi dasar telepon genggam (Telepon dan SMS), portabel atau mudah dibawa kemanapun, umumnya memiliki antarmuka touchscreen, dan mengusung sistem operasi populer seperti *Android*, *iOS*, dan *BlackBerry OS*.

### **D. Java**

*Java* adalah bahasa pemrograman dan serambi, atau bagian dari kombinasi perangkat keras dan kerangka perangkat lunak untuk mengeksekusi suatu *software* atau program. *Java* memiliki keunggulan *Write once, run anywhere* dimana setiap aplikasi java dapat dijalankan dimana ada serambi yang dimaksud, yaitu *Java Virtual Machine*. Sekarang *java* merupakan bahasa pemrograman dan serambi utama dari sistem operasi Android.

### **E. Kode Sumber**

Kode sumber, atau lebih dikenal dengan *source code* adalah serangkaian teks berisi instruksi yang mudah dibaca, dipahami, dan direvisi manusia, untuk kemudian diterjemahkan melalui proses yang

disebut kompilasi menjadi bahasa mesin dalam program yang dapat langsung dibaca eksekutor program.

#### **F. Fungsi (Subroutine)**

Dalam pemrograman, fungsi merupakan sekumpulan instruksi yang tidak berada dalam urutan eksekusi dari instruksi utama aplikasi, tetapi sewaktu-waktu dapat dipanggil untuk dieksekusi dari instruksi lain, bisa melalui rangkaian instruksi utama, dari fungsi lainnya, maupun dari fungsi yang sama, yang dikenal sebagai instruksi rekursif atau memanggil dirinya sendiri.

#### **G. Android Studio**

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment (IDE)* untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA, yang merupakan sebuah IDE *Java*. Android Studio menawarkan beberapa fitur yang memudahkan pengembangan, misalnya emulator untuk mencoba aplikasi secara langsung, penganalisis kode untuk menemukan kesalahan, integrasi dengan sistem kontrol versi *GitHub*, dan sistem pembangunan aplikasi *Gradle*. *Android Studio* bekerja dengan berkas kerja *Android Studio Project*, yang memuat berkas penting dalam pembuatan aplikasi.

#### **H. Apktool**

*Apktool* merupakan alat yang digunakan untuk mendekompile suatu aplikasi android menjadi kode sumbernya, serta mengembalikannya menjadi bentuk aplikasi setelah diubah, untuk kepentingan penambahan fitur



dan dukungan bagi jenis *smartphone* tertentu, walau dalam prakteknya banyak didapati penyalahgunaan untuk kegunaan ilegal, seperti melakukan peretasan aplikasi berbayar atau penambahan kode yang berfungsi untuk mencuri informasi. Dalam penelitian ini akan digunakan fitur dekompilasi yang memiliki sintaks pemanggilan sebagai berikut:

```
apktool d <aplikasitarget.apk> -o <./direktoritarget>
```

Maksudnya adalah: (apktool:) Gunakan program *apktool* untuk (d:) mendekompilasi (<aplikasitarget.apk>:) berkas “aplikasitarget.apk” dan (-o <./direktoritarget>:) simpan hasil keluaranya di folder “direktoritarget”

Dalam penelitian ini akan digunakan juga fitur rekompilasi dengan sintaks sebagai berikut:

```
apktool b <./direktoritarget>
```

Maksudnya adalah: (apktool:) Gunakan program *apktool* untuk (b:) mengkompilasi ulang (<./direktoritarget>:) berkas yang ada di direktori “direktoritarget.”

## I. Gradle

*Gradle* adalah sebuah *build system*, yaitu sistem otomatisasi proses materi asal seperti *source code* dan *resource* seperti gambar maupun teks, menjadi program yang siap digunakan melalui peranan sebuah *gradle script*. *Android Studio* menggunakan *Gradle* sebagai *build system*nya, dengan tujuan agar pengguna dapat melakukan *build project* hanya melalui *command line*.

## **J. ART/Android Runtime**

*Android runtime* merupakan eksekutor dari program yang ada di aplikasi android. ART menerima format berkas *dalvik executable format/.dex*. Generasi sebelumnya dari ART adalah *Dalvik Runtime Environment* yang digunakan sampai *Android KitKat 4.4*. Format *dalvik executable* sendiri merupakan hasil konversi dari berkas *class*, sehingga dapat disebut bahwa ART merupakan turunan dari *Java Virtual Machine*.

## **K. Smali**

*Smali* merupakan terjemahan *human-readable* dari berkas *.dex*. Dalam pekerjaan *reverse-engineering* suatu aplikasi, berkas *.dex* yang merupakan hasil kompilasi dari *source code java* mustahil didekompilasi kembali menjadi kode sumber aslinya yang berbahasa *java*, karena perbedaan struktur. Maka dari itu, berkas *.dex* harus dimodifikasi langsung. Dekompilasi format *.dex* ke *.smali* bertujuan untuk mempermudah pekerjaan ini.

## **L. Touchscreen**

*Touchscreen* merupakan suatu komponen elektronik yang merangkap sebagai komponen keluaran dan masukan, dimana fungsi keluaran dimiliki sebagai layar dan fungsi masukan dimiliki sebagai penunjuk sentuhan.

## **M. Android OS**

*Android OS* adalah sistem operasi berbasis linux yang didesain untuk gawai portabel. Sekarang ini *Android* secara resmi didukung oleh Google dan menerima pembaruan secara konstan. Dari pertama dirilis hingga kini,

*Android* telah mengalami beberapa kali pembaruan besar yang namanya berdasarkan manisan dengan huruf awalurut abjad; *Alpha, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jellybean, Kitkat, Lollipop, Marshmallow, Nougat*, hingga versi terbaru adalah 8.0.0 *Oreo*, yang dikeluarkan dengan perangkat mengembang (*Standard Development Kit/SDK*) Level 26.

#### **N. Service**

Sebuah komponen dari aplikasi yang dibuat untuk operasi latar belakang yang berlangsung lama dan tidak membutuhkan antarmuka.

#### **O. Activity**

Sebuah komponen dari aplikasi yang dibuat untuk mempermudah interaksi pengguna dengan aplikasi melalui sebuah antarmuka. Umumnya *Activity* adalah komponen yang menerima *input events*.

#### **P. Bluetooth**

*"Bluetooth® is a low-power wireless connectivity technology used to stream audio, transfer data and broadcast information between devices. There are two flavors of Bluetooth technology, Basic Rate/Enhanced Data Rate (BR/EDR) and Low Energy (LE)."*(Bluetooth SIG, Inc. 2017) Sama seperti teknologi *wireless* yang kebanyakan ada sekarang, *Bluetooth* berbasis gelombang radio. Bluetooth dapat digunakan sebagai media protokol komunikasi serial.

#### **Q. Bluetooth-Library**

Merupakan *java library* yang berisi API untuk pemrograman interaksi *bluetooth*. Dalam *project Android Studio*, *library* ini disertakan melalui penyertaan kode berikut di *gradle dependencies*:

```
compile 'me.aflak.libraries:bluetooth:1.2.4'
```

#### **R. Android Input Events**

Dalam *Android*, segala jenis interaksi pengguna disebut “*Input Event*”. Antarmuka yang digunakan untuk merekam *Input Event* adalah *Event Listeners*. (Android Developers, 2017) Bentuk *Event* yang mungkin terjadi dibagi menjadi dua kelompok besar, yaitu sentuhan dan interaksi keyboard. Sentuhan sendiri dibagi menjadi banyak, misalnya sentuhan, pelepasan, dan sentuhan lama.

#### **S. Android Instrumentation**

(Android Developers, 2017) *Android Instrumentation* adalah seperangkat *Application Programming Interface (API)* yang berisi berbagai peralatan untuk pengukuran teliti dari berbagai hal yang sedang berjalan dalam *Android OS*, serta melakukan simulasi *input*. API ini digunakan melalui penyertaan kode berikut:

```
import android.app.Instrumentation.*;
```

#### **T. Komunikasi serial**

Komunikasi serial adalah bentuk penyederhanaan dari komunikasi paralel yang membutuhkan banyak pin. Pada dasarnya komunikasi digital menggunakan kode biner (0 dan 1) yang dikode dengan perbedaan tegangan, misalnya +5v berarti 1 dan 0v berarti 0. Pada komunikasi serial, hanya

digunakan 2 pin, yaitu Rx dan Tx, pin keluar dan masuk, dimana pada komunikasi paralel, bisa digunakan sampai  $2^{3+n}$  pin.

#### **U. Arduino Nano**

*Arduino Nano* adalah salah satu varian dari *Arduino*, yaitu peralatan *microcontroller* untuk hobi robotika. Versi *nano* memiliki ukuran yang relatif kecil dibanding varian lainnya. Dalam komunikasi, seri *Arduino* mendukung komunikasi serial.

#### **V. HC-05 Bluetooth Module**

HC-05 *Bluetooth Module* adalah perangkat *bluetooth* yang dapat diintegrasikan dengan berbagai peralatan dengan komunikasi serial, dengan mentransmisikan data serial yang diterimanya melalui protokol *bluetooth*. HC-05 dapat bertindak sebagai *master* maupun *slave*, yaitu menunggu kontak dari luar maupun membuka sebuah kontak dengan perangkat lain.

## BAB III

### METODOLOGI PENELITIAN

#### A. Jenis penelitian

Penelitian yang digunakan adalah dengan membuat sebuah alat yang merupakan aplikasi dari konsep yang telah dijabarkan, maka penelitian ini dikategorikan sebagai penelitian rekayasa.

#### B. Kelengkapan penelitian

Penelitian dilaksanakan dengan alat berikut:

a. Laptop, OS 4.9.0-kali4-amd64 (Linux, dari *output* `uname -r`), Processor *Intel i3 4030u*, RAM 2GB. Daftar aplikasi yang akan digunakan:

- i. *Gradle*
- ii. *Android Studio*
- iii. *Apktool*
- iv. Editor teks (*Sublime text 3*), dengan *plugin Arduino IDE*.
- v. *Android Debug Bridge (ADB)*

b. *Arduino Nano* dengan kelengkapan tambahan berikut:

- i. Modul *bluetooth HC-05*
- ii. *3 Lever button*

c. Smartphone *Android* sebagai media uji, digunakan *Hisense Pureshot+*

#### C. Metode penelitian

Rekayasa aplikasi dilaksanakan dengan langkah-langkah umum sebagai berikut, dan dapat berbeda-beda untuk setiap aplikasi:

- a. Diprogram sebuah aplikasi yang memiliki satu *main activity* sebagai pembawa dan satu *service* yang berisi kode utama penerima *bluetooth input* dan *event injector* dalam suatu *project* Android Studio baru. Aplikasi ini akan dinamakan “BTSerial”. Langkahnya sebagai berikut:
  - i. Dibuka *Android Studio*, dan pilih *Create a new project*.
  - ii. Diberi nama aplikasi “BTSerial”, dan nama paket utama “com.retorikal.btserial” Selebihnya digunakan konfigurasi standar yang disediakan.
  - iii. Mulai tanpa membuat aktifitas baru.
- b. Dilakukan pemrograman sebanyak yang diperlukan.
- c. Dibangun sebuah berkas aplikasi dengan mengeksekusi perintah berikut di direktori asal dari *project Android Studio* yang telah dibuat:  

```
./gradlew assembleDebug
```

Artinya adalah: (*./gradlew:*) jalankan aplikasi *gradlew* di direktori tersebut untuk (*assembleDebug:*) membangun kode sumber yang tersedia menjadi aplikasi versi *debug*. Aplikasi akan dibuat di direktori *./app/build/outputs/apk/app-debug.apk*.
- d. Disediakan satu aplikasi target, yang akan diberikan dukungan untuk alat yang akan dibuat.
- e. Keduanya *didecompile* dengan *apktool*.
  - i. Untuk BTSerial, digunakan perintah berikut:  

```
apktool d app-debug.apk -o ./source
```
  - ii. Untuk aplikasi yang hendak dimodifikasi, digunakan perintah berikut:  

```
apktool d <aplikasitarget.apk> -o ./target
```

- f. Disalin semua berkas *.smali* yang ada di direktori */smali/com/retorikal/* hasil dekompilasi *BTSerial*, dan merupakan hasil dekompilasi dari *service* penerima *bluetooth input* dan *event injector*, yaitu yang mengandung nama "*BTService*".
- g. Dibuat direktori */com/retorikal/* di dalam direktori */smali/* aplikasi target, lalu tempel semua berkas yang telah disalin.
- h. Disalin direktori */smali/me/* dari *BTSerial*, dan ditempel di dalam direktori */smali/* dari aplikasi target.
- i. Diperiksa berkas *AndroidManifest.xml* di direktori asal (/) hasil dekompilasi aplikasi target;
  - i. Ditentukan *Main Activity*, melalui keberadaan label berikut di dalam salah satu label *<activity>*:
 

```
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
```
  - ii. Ditambahkan label berikut di dalam label *<manifest>*, untuk mendeklarasikan bahwa aplikasi akan menggunakan *hardware support bluetooth*:
 

```
<uses-permission android:name="
    android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="
    android.permission.BLUETOOTH"/>
```
  - iii. Didaftarkan penggunaan *service BTService* yang ikut selesai seiring aplikasi, dengan penambahan label berikut di dalam label *<application>*, yang ada di dalam label *<manifest>*:



```

<service android:exported="false"
        android:name="com.retorikal.btserial.BTService"
        android:stopWithTask="true"/>

```

j. Analisis rantai panggilan *Main Activity*, dengan menelusuri *activity* yang dipanggil, hingga mencapai sebuah *activity* yang memiliki fungsi `onCreate()`.

k. Dimasukkan kode yang bertujuan untuk memulai *BTService* sewaktu aplikasi dimulai, dengan menuliskan kode tersebut dibawah “`invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V`”:

```

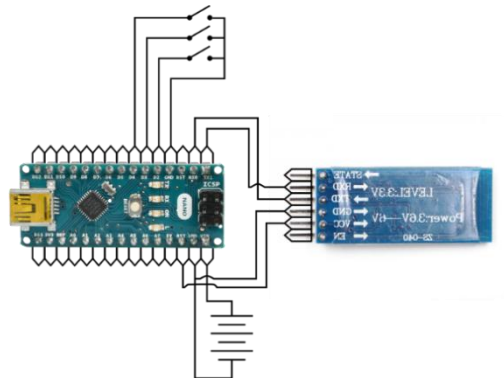
invoke-static {p0}, Lcom/retorikal/btserial/BTService;-
    >start(Landroid/content/Context;)V

```

l. Dikompilasi ulang aplikasi hasil dekompilasi yang telah dimodifikasi, dengan perintah berikut:

```
apktool b ./target
```

Untuk mendukung hasil tersebut dibuat juga perangkat keras kontroler yang mengimpelementasikan fitur injeksi tersebut, dibuat menggunakan *Arduino Nano* dan modul *bluetooth* HC-05. Perangkat tersebut ditempelkan dengan penjepit *monopod* agar dapat terpasang pada *smartphone*. Skema alat tersebut adalah sebagai berikut:



## BAB IV

### HASIL PENELITIAN

Dalam tes penelitian digunakan dua aplikasi contoh: yaitu *Sniper 3D* dan *Pixel Fury*, yang mana keduanya tersedia di *Google Play Store*, tapi untuk mendapatkan berkas pemasang *.apk*, pengunduhan dilakukan melalui *apkpure.com*. Aplikasi hasil modifikasi memiliki keluaran sebagai berikut:

1. Muncul notifikasi “Implements “Injektor” by Ganthet”
2. Muncul notifikasi “Connect to: HC-05” saat berusaha menyambung dengan perangkat keras yang ada.
3. Muncul notifikasi “Connected!”, alat siap digunakan.

Pada pengujian dengan aplikasi *Pixel Fury*, perangkat ini diketahui memiliki sifat yang kurang mendukung permainan. Sifat yang dimaksud adalah tidak adanya fitur sentuhan panjang, dimana dalam game *FPS*, penggunaan senjata *full-automatic* seperti *rifle*, *sub-machine gun*, dan *machine gun* dilakukan dengan sentuhan panjang, sementara senjata kelas tersebut tidak diperuntukkan bagi penggunaan mode semi-otomatis. Dengan kelemahan tersebut, perangkat ini masih dapat berguna untuk senjata dengan mode penembakan *semi-automatic*, misalnya kelas *sniper* maupun *bazooka*. Dalam kasus sebuah sasaran yang bergerak cepat, perintah tembak yang berbeda tombol dapat mengeliminasi jeda reaksi dari perpindahan jari, maka jari yang mengendalikan *joystick* arah hadap dapat terus mengarahkan bidiknya kepada target.

Pada pengujian menggunakan aplikasi *Sniper 3D*, didapati penurunan performa permainan. Hal ini disebabkan oleh jarak tekan tombol pada alat yang

relatif jauh dibanding dengan menekan tombol perintah tembak yang asli di kanan bawah layar, sehingga terdapat jeda lain. Selain itu, permainan ini tidak menuntut mobilitas tinggi, dimana pemain hanya diam di satu titik dan menembak target yang telah ditentukan. Hal ini membebaskan satu jari dari kewajiban mengendalikan navigasi, maka jari tersebut dapat digunakan untuk mengendalikan penembakan, maka alat ini tidak cocok untuk digunakan dalam permainan seperti ini.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

- a. Untuk mengatasi masalah jeda reaksi, dapat dibuat sebuah alat yang memberikan tombol tambahan untuk mencegah jeda yang dibutuhkan untuk memindah jari dari satu tombol kontrol ke tombol kontrol lainnya.
- b. Alat ini berfungsi paling baik saat karakter diharuskan melakukan navigasi mengelilingi peta permainan, tetapi alat yang dibuat belum diprogram untuk menginjeksikan *input* berupa sentuhan panjang. Oleh karena itu alat ini lebih cocok untuk digunakan dalam mode *First Person Shooter* dimana karakter memegang senjata yang tidak menembak secara otomatis, seperti *sniper* dan *bazooka*.

#### **B. Saran**

Hasil penelitian ini telah menunjukkan akan adanya kemungkinan dibuatnya kontroler game android yang berfungsi dengan hampir semua game melalui metode *event injection*. Hasil dari karya tulis ini dapat dikembangkan lagi, misalnya dengan pengayaan fitur dan penggunaan tombol yang jarak tekanya pendek, dan pemberian fitur injeksi sentuhan panjang.

## **BAB VI**

### **DAFTAR PUSTAKA**

- Flatow, Ira. 1993. They All Laughed... from light bulbs to lasers: the fascinating stories behind the great inventions that have changed our lives. HarperCollins
- Yee, Nick. 2007. Motivations for Play in Online Games. *CyberPsychology & Behavior* 9.6 : 772-775.
- Ryan, R. M., & Deci, E. L. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1), 68.
- Syaputra, Rizki. 2015. Membuat aplikasi android dengan Android Studio. IMA Studio
- US Legal, Inc. 2016. Gaming Device Law and Legal Definition.  
<https://definitions.uslegal.com/g/gaming-device/> (Diakses 06:15/4/10/2017)
- Android Developers, 2017. Input Events.  
<https://developer.Android.com/guide/topics/ui/ui-events.html> (Diakses 21:50/16/10/2017)
- Android Developers, 2017. Instrumentation.  
<https://developer.android.com/reference/android/app/Instrumentation.html> (Diakses 18:13/11/10/2017)
- Android Developers, 2017. Activity.  
<https://developer.android.com/reference/android/app/Activity.html> (Diakses 22:34/03/11/2017)

Android Developers, 2017. Service.

<https://developer.android.com/reference/android/app/Service.html> (Diakses 22:30/03/11/2017)

Tumleson, Connor. 2017. Apktool - A tool for reverse engineering 3rd party, closed, binary Android apps. <https://ibotpeaches.github.io/Apktool/> (Diakses 23:07/3/11/2017)

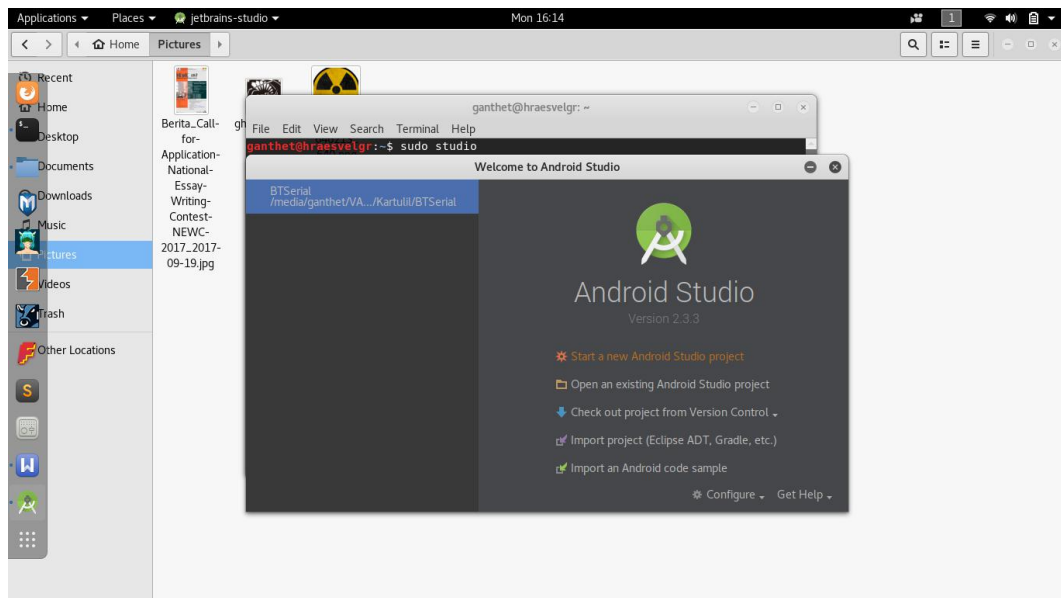
Aflak, Omar. 2016. Bluetooth-Library. <https://github.com/omaflak/Bluetooth-Library> (Diakses 18:22/11/1/2017)

Bluetooth SIG, Inc. 2017. How it works. <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works> (Diakses 21:50/16/10/2017)

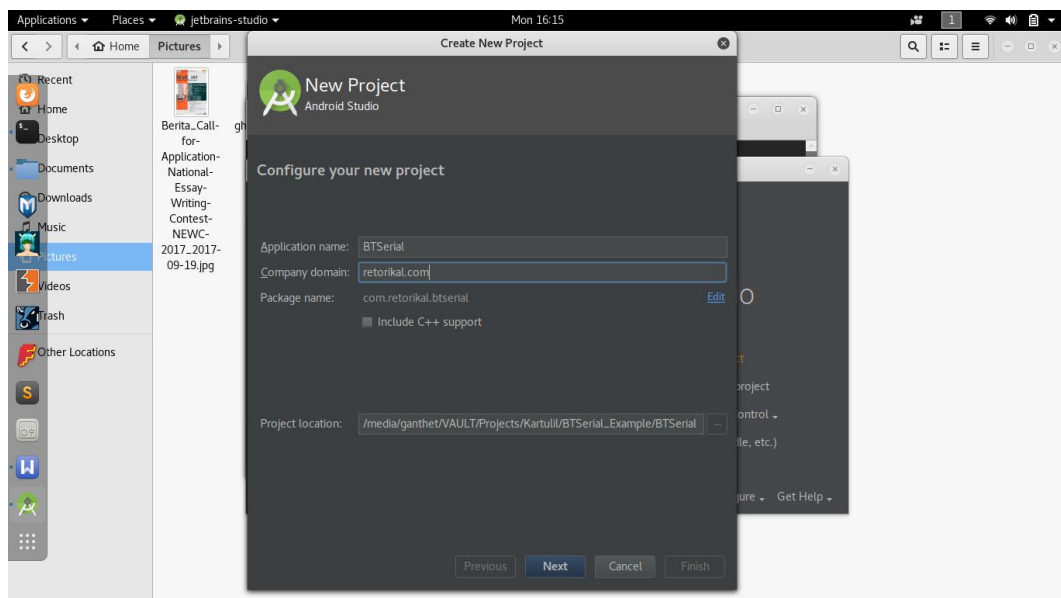
Jimbo. 2012. Serial Communication. <https://learn.sparkfun.com/tutorials/serial-communication> (Diakses 22:36/16/10/2017)

## LAMPIRAN

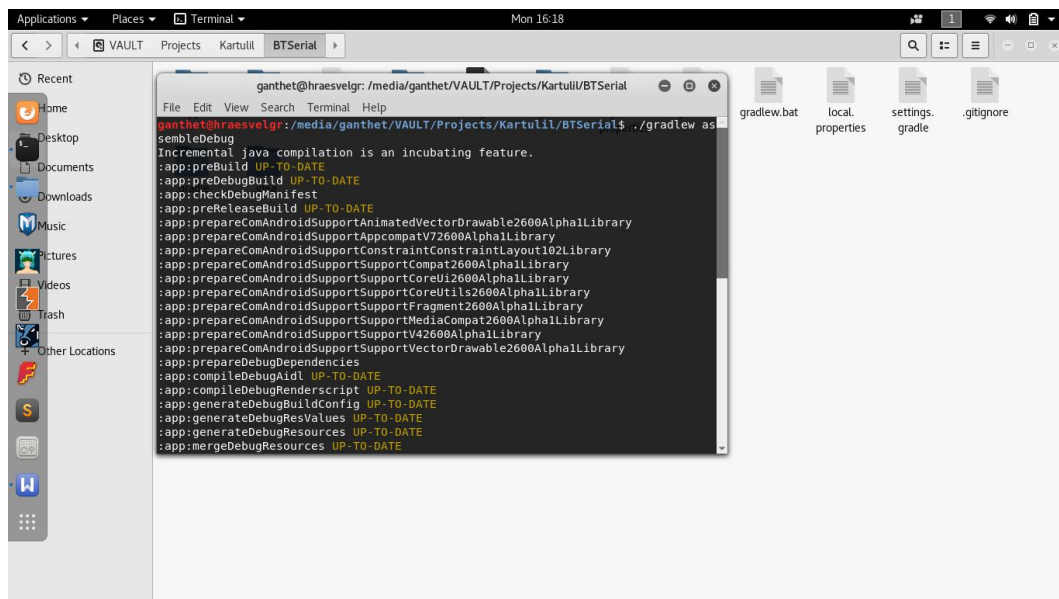
### A. Dokumentasi



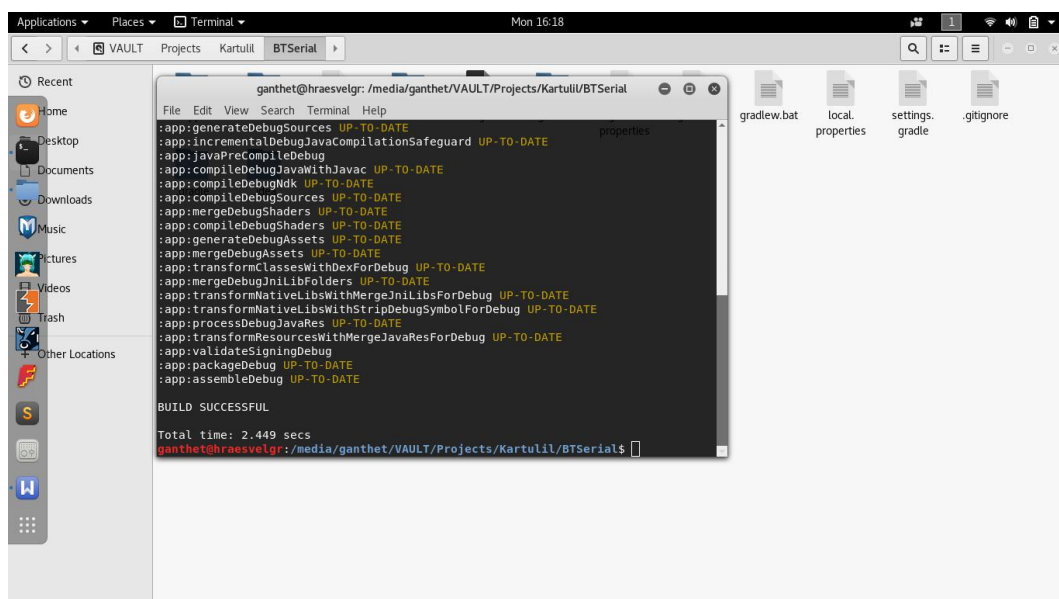
Tampilan Android Studio



Membuat *project* baru *BTSerial*

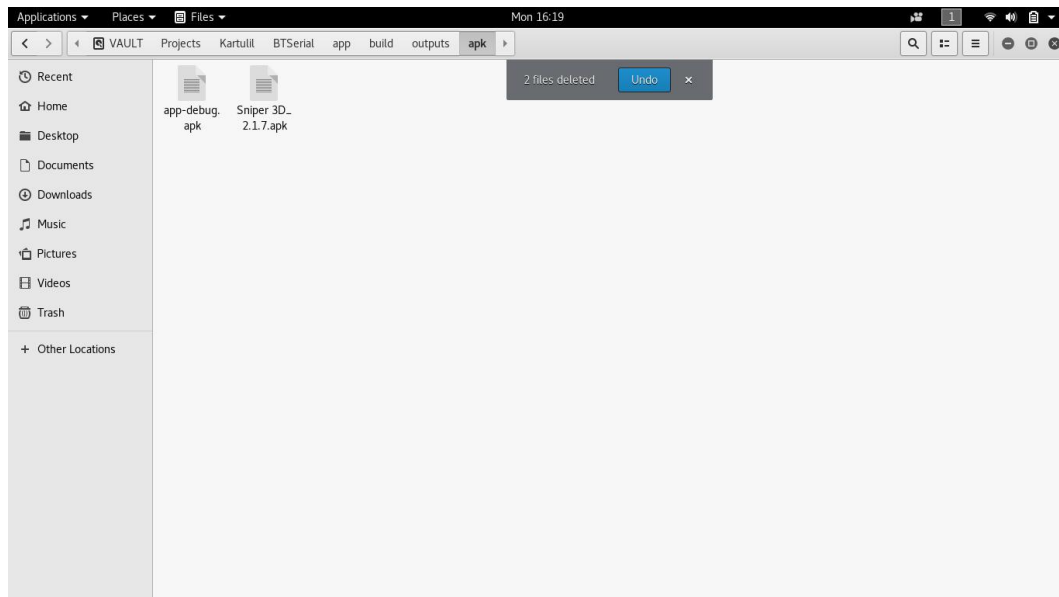


Penyusunan varian *debug* aplikasi *BTSerial*

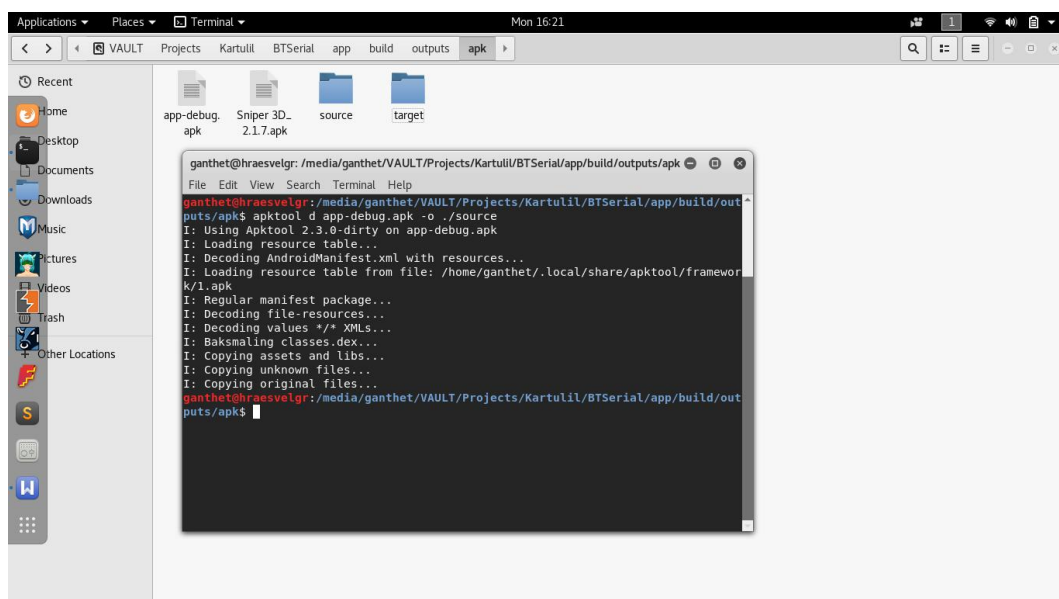


Penyusunan selesai

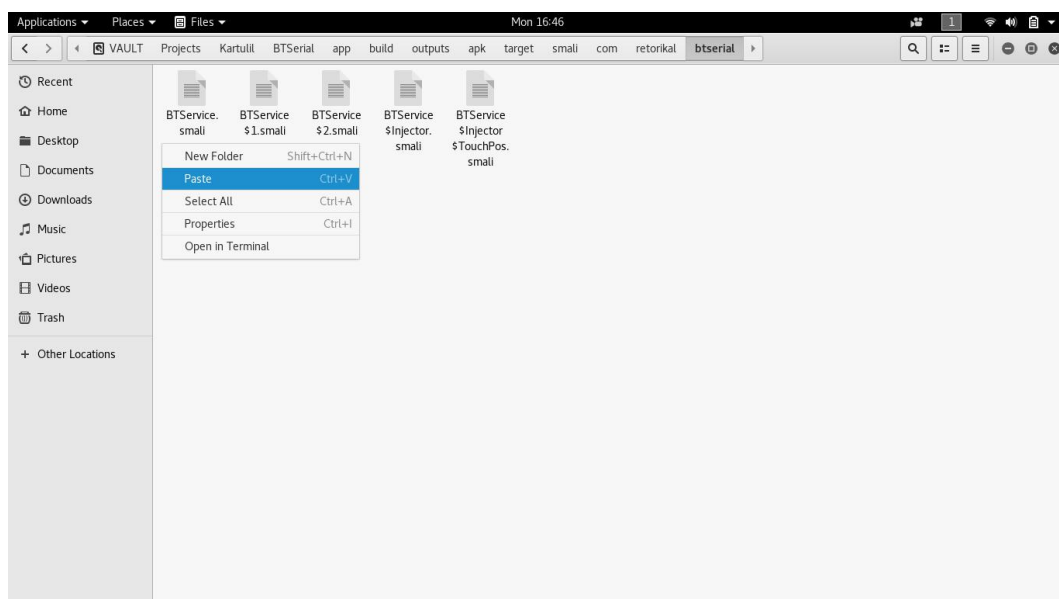
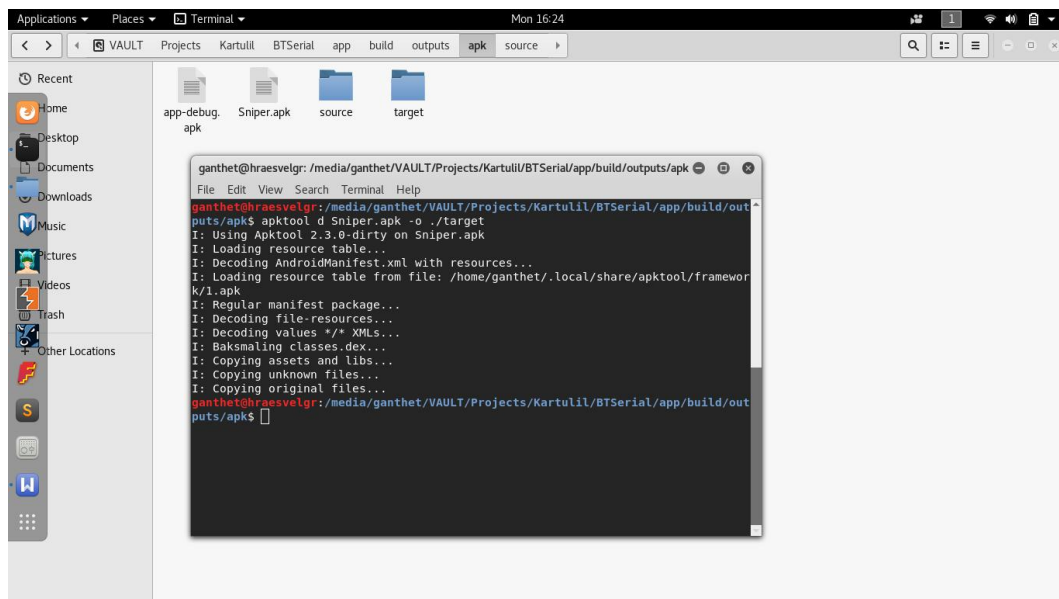


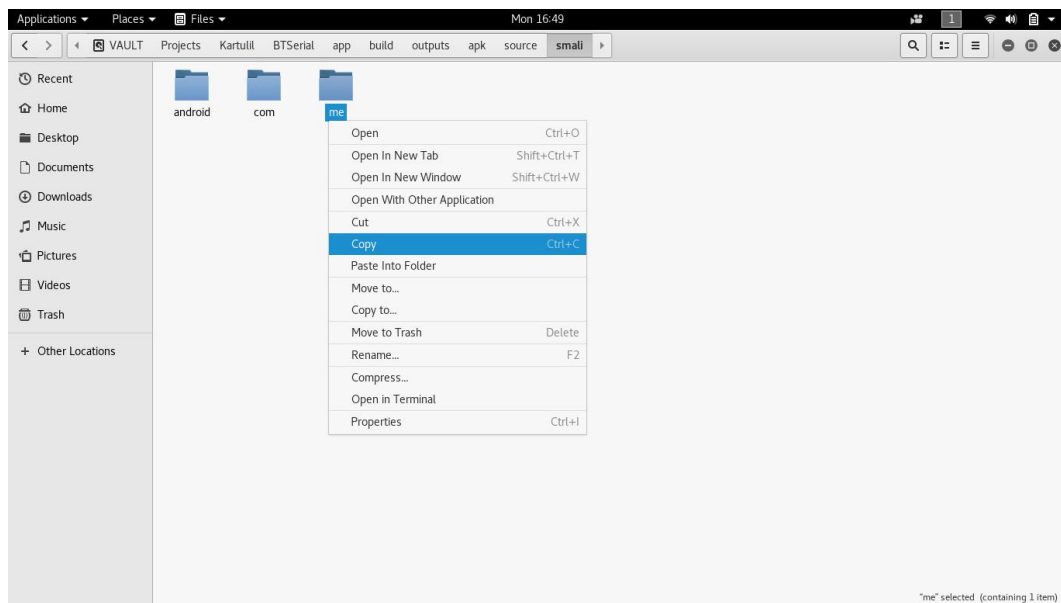


Aplikasi *BTSerial* dan target, dalam direktori kerja

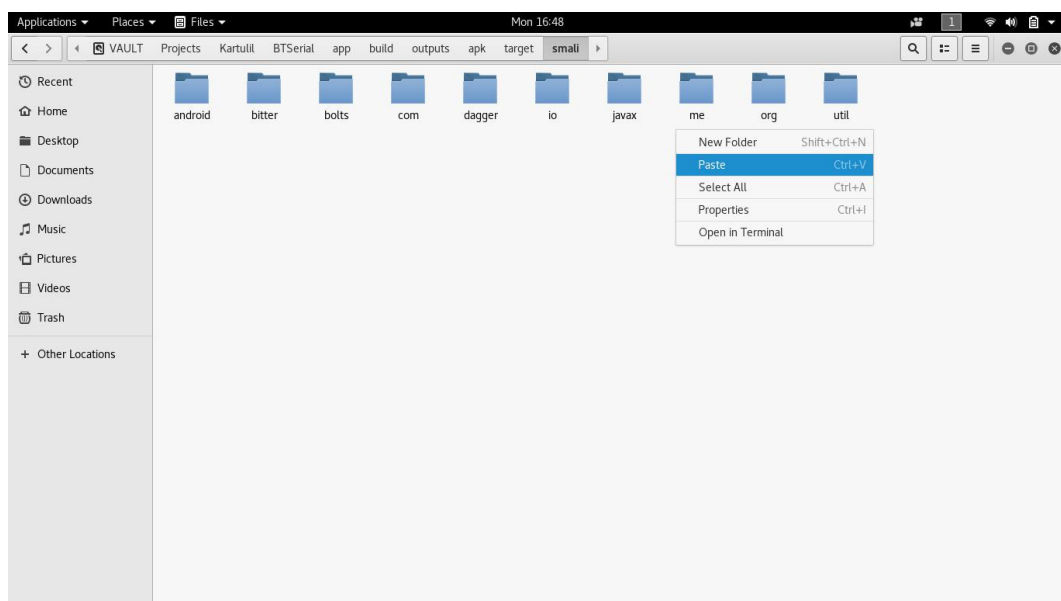


Dekompilasi aplikasi *BTSerial*

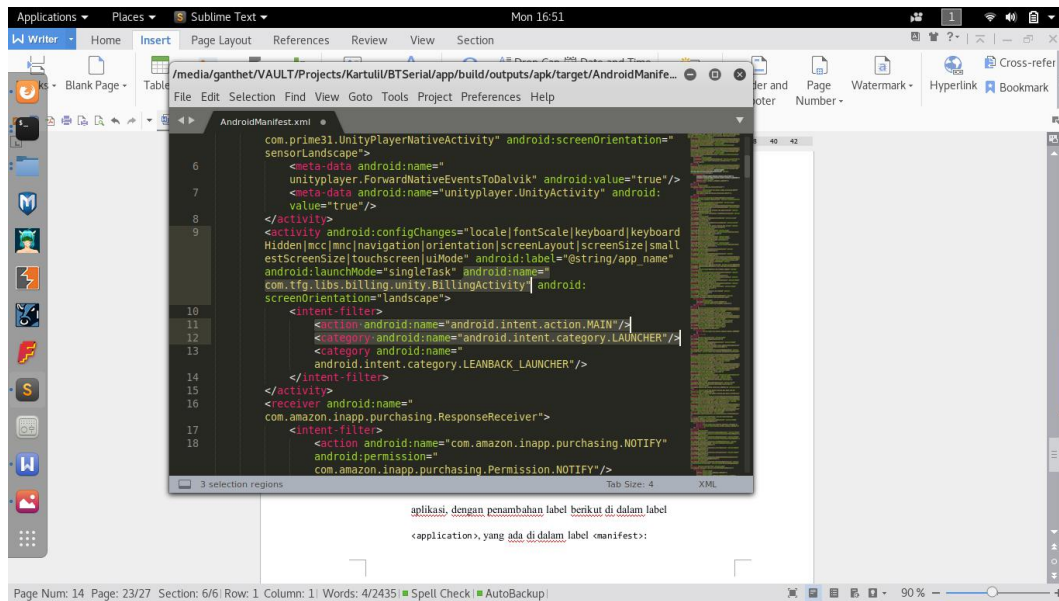




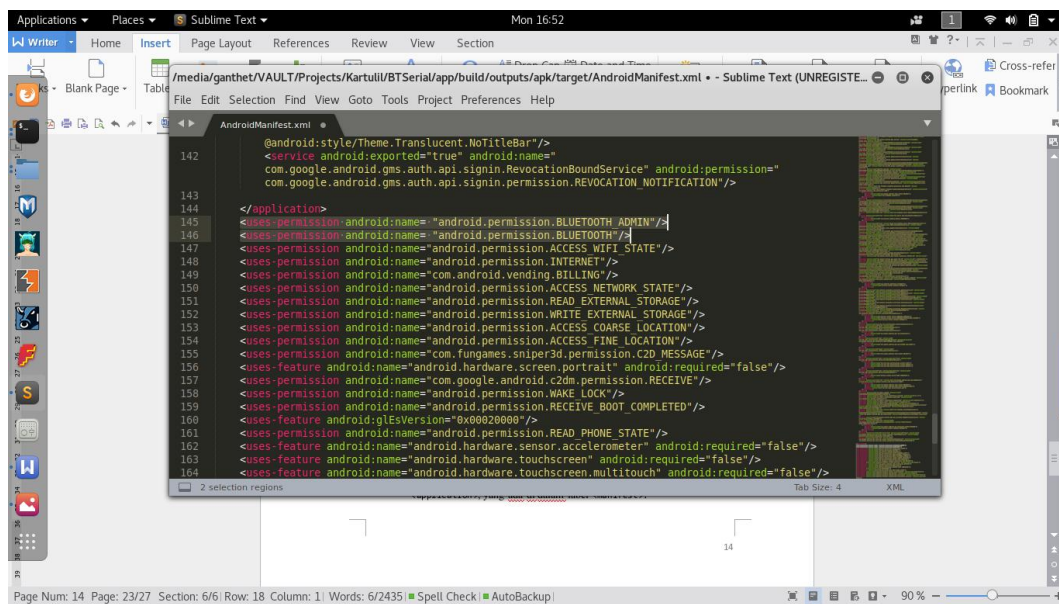
Penyalinan struktur direktori untuk berkas *BTSerial*



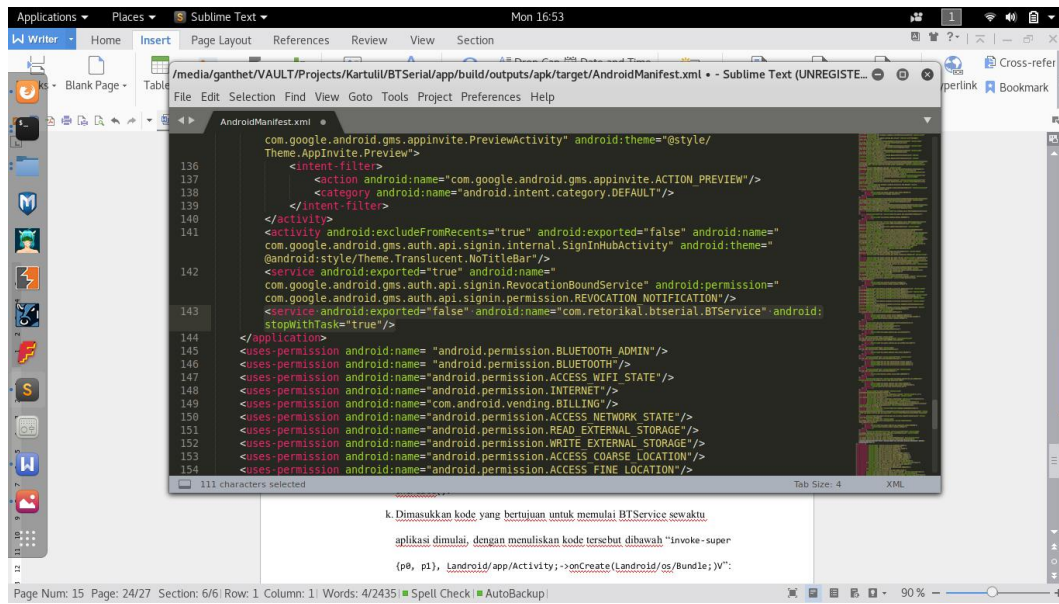
Penempelan struktur direktori untuk berkas *BTSerial*



## Analisa rantai panggilan aktivitas utama



## Deklarasi penggunaan dukungan perangkat keras



```
com.google.android.gms.appinvite.PreviewActivity" android:theme="@style/
Theme.AppInvite.Preview">
136     <intent-filter>
137     <action android:name="com.google.android.gms.appinvite.ACTION_PREVIEW"/>
138     <category android:name="android.intent.category.DEFAULT"/>
139     </intent-filter>
140 </activity>
141 <activity android:excludeFromRecents="true" android:exported="false" android:name="
com.google.android.gms.auth.api.signin.internal.SignInHubActivity" android:theme="
@android:style/Theme.Translucent.NoTitleBar"/>
142 <service android:exported="true" android:name="
com.google.android.gms.auth.api.signin.RevocationBoundService" android:permission="
com.google.android.gms.auth.api.signin.permission.REVOCATION_NOTIFICATION"/>
143 <service android:exported="false" android:name="com.retorikal.btserial.BTService" android:
stopWithTask="true"/>
144 </application>
145 <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
146 <uses-permission android:name="android.permission.BLUETOOTH"/>
147 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
148 <uses-permission android:name="android.permission.INTERNET"/>
149 <uses-permission android:name="com.android.vending.BILLING"/>
150 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
151 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
152 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
153 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
154 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

k. Dimasukkan kode yang bertujuan untuk memulai BTService sewaktu
aplikasi dimulai, dengan menuliskan kode tersebut dibawah "Invoke - super
(p0, p1), Landroid/app/Activity;.>onCreate(Landroid/os/Bundle;)V":
```

Deklarasi penggunaan service *BTSerial*

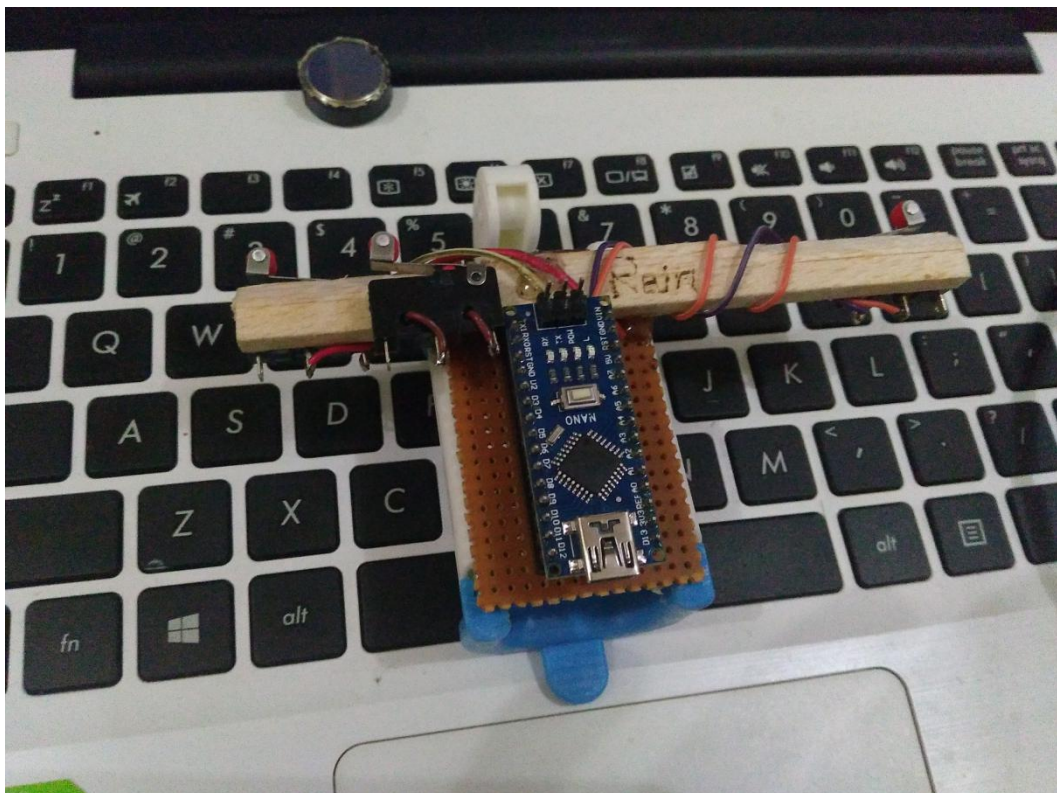


Alat yang terpasang ke *smartphone*





Tampak belakang alat, diletakkan di *keyboard* dari *laptop* perangkat kerja



Tampak depan alat



Aplikasi uji: *Pixel Fury*, dengan *pointer location screen overlay*



Aplikasi uji: *Sniper 3D*, dengan *pointer location screen overlay*





*Cutscene* detik-detik terbunuhnya *high value target/HVT* pada pengujian dengan *Sniper 3D*. Titik merah di pojok bawah kiri (dalam lingkaran biru) adalah *feedback* dari *pointer location screen overlay* hasil *touch event injection* di posisi tombol tembak



Selesai misi dengan perangkat yang dibuat dengan aplikasi uji *Sniper 3D*



## B. Kode sumber untuk unit kontroler

```
/*
 *Kode unit kontroler
 *By Reinard Rahardian
 *https://github.com/retorikal/
 */

float x[3];
float y[3];
int input[3];
boolean inputPressing[3];

void setup(){
  Serial.begin(9600);
  for(int i=0; i<3; i++){
    x[i] = 3.50f;
    y[i] = 95.00f;
    input[i] = i+2;
    inputPressing[i] = false;
    pinMode(input[i], INPUT_PULLUP);
  }
  x[1] = 81.42f;
  y[1] = 79.16f;
}

void loop(){
  for(int i=0; i<3; i++){
    if((digitalRead(input[i])==LOW) && inputPressing[i] ==
false){
      inputPressing[i]=true;
      send(i);
    }
    else if(digitalRead(input[i])==HIGH){
      inputPressing[i]=false; }
  }
}

void send(int ord){
  Serial.print(x[ord]);
  Serial.print(' ');
  Serial.println(y[ord]);
}
```

## C. Kode sumber untuk penerima *bluetooth*

```
package com.retorikal.btserial;

/*
```

```

*Kode unit kontroler
*By Reinard Rahardian
*https://github.com/retorikal/
*
*/

import android.app.Instrumentation;
import android.app.Service;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.res.Resources;
import android.content.Intent;
import android.os.Handler;
import android.os.IBinder;
import android.os.Looper;
import android.os.SystemClock;
import android.view.MotionEvent;
import android.widget.Toast;

import java.util.List;

import me.aflak.bluetooth.Bluetooth;

public class BTService extends Service implements
Bluetooth.CommunicationCallback {
    private Bluetooth bt;
    private BluetoothDevice device;
    private Injector injector;

    @Override
    public int onStartCommand(Intent intent, int flags, int
startId) {
        bt = new Bluetooth(this);
        bt.enableBluetooth();
        bt.setCommunicationCallback(this);
        injector = new Injector();

        notifyMsg("Implements \"Injektor\" by Ganthet");
        device = getDeviceByName("HC-05");

        Thread delayer = new Thread(new Runnable() {
            @Override
            public void run() {
                try {Thread.sleep(3000);}
                catch (InterruptedException e)
{e.printStackTrace();}
                if (device != null);
                bt.connectToDevice(device);
            }
        });
        delayer.start();
    }

```

```

        return START_NOT_STICKY;
    }

    public BluetoothDevice getDeviceByName(String name) {
        List<BluetoothDevice> paired = bt.getPairedDevices();

        for (BluetoothDevice dVice : paired) {
            if (name.equals(dVice.getName())) {
                notifyMsg("Connect to: " + dVice.getName());
                return dVice;
            }
        }
        return null;
    }

    public void notifyMsg(final String msg) {
        Handler handler = new Handler(Looper.getMainLooper());
        handler.post(new Runnable() {
            @Override
            public void run() {
                Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public void onConnect(BluetoothDevice device) {
        notifyMsg("Connected!");
    }

    @Override
    public void onDisconnect(BluetoothDevice device, String
message) {}

    @Override
    public void onMessage(String message) {
        float xCrd, yCrd;
        message.replaceAll("\n", " ");
        String tmpMsg[] = message.split(" ");
        xCrd = Float.parseFloat(tmpMsg[0]);
        yCrd = Float.parseFloat(tmpMsg[1]);
        injector.injectTouch(xCrd, yCrd);
    }

    @Override
    public void onError(String message) {}

    @Override
    public void onConnectError(BluetoothDevice device, String
message) {}

```

```

@Override
public IBinder onBind(Intent intent) {
    // TODO: Return the communication channel to the service.
    throw new UnsupportedOperationException("Not yet
implemented");
}

public static void start(Context ctxt) {
    ctxt.startService(new Intent(ctxt, BTService.class));
}

private class Injector extends Thread {
    private Instrumentation instrument;
    private int scrWdth;
    private int scrHght;
    private Boolean waiting = true;
    private final Object lock = new Object();
    private TouchPos tPos = new TouchPos();

    Injector() {
        scrWdth =
Resources.getSystem().getDisplayMetrics().widthPixels;
        scrHght =
Resources.getSystem().getDisplayMetrics().heightPixels;
        instrument = new Instrumentation();
        this.start();
    }

    public void run() {
        while (true) {
            synchronized (lock) {
                try {
                    if (waiting) {
                        lock.wait();
                    }
                } catch (InterruptedException e)
{e.printStackTrace();}

                instrument.sendPointerSync(MotionEvent.obtain(SystemClock.upti
meMillis(),SystemClock.uptimeMillis(),MotionEvent.ACTION_DOWN,tP
os.getXPos(),tPos.getyPos(), 0));

                instrument.sendPointerSync(MotionEvent.obtain(SystemClock.upti
meMillis(),SystemClock.uptimeMillis(),MotionEvent.ACTION_UP,tPos.
getXPos(),tPos.getyPos(), 0));

                waiting = true;
            }
        }
    }
}

```

```

public void injectTouch(float xPos, float yPos) {
    synchronized (lock) {
        tPos.setRelativePos(xPos, yPos);
        if (waiting) {
            waiting = false;
            lock.notify();
        }
    }
}

private class TouchPos {
    private float xPos = 360;
    private float yPos = 200;

    void setRelativePos(float x, float y) {
        xPos = scrWdth * (x / 100f);
        yPos = scrHght * (y / 100f);
    }

    float getXPos() {
        return xPos;
    }

    float getYPos() {
        return yPos;
    }
}
}
}

```

## BIOGRAFI PENULIS



Reinard Rahardian A.S. lahir di Semarang, pada tanggal 25 Desember 2000. Awal pendidikan dari *playgroup* Satya Wacana Salatiga, dilanjutkan ke TK di Satya Wacana Salatiga hingga 2006. Kemudian mengenyam pendidikan SD di SD Kristen Satya Wacana dan lulus pada tahun 2012, kemudian melanjutkan ke SMP Kristen Satya Wacana, dan pada saat itu berhasil mengikuti OSN pada tahun 2014, dan lolos sebagai finalis tingkat nasional. Kemudian melanjutkan pendidikan ke SMA Pangudi Luhur Van Lith Muntilan, dan membuat sebuah karya tulis berjudul “Kontroler Game Android berbasis *Event Injection*” sebagai salah satu syarat kelulusan. Menyukai hal-hal yang berbau komputer.