

Jurani

Master Test Plan

Version 1.0

*Angellica Octavianzy Sabrina, Kenneth Mohammad Albani Hakim,
Mahendra Sunt Servanda, Sandy Febriant Eka Putra*

March 31, 2016

Table of Contents

1. Introduction

- 1.1 Purpose - 3
- 1.2 Test Scope - 3
- 1.3 Testing Overview - 4

2. Test Strategy

- 2.1 Function Test - 4
- 2.2 Unit Test - 5
- 2.3 Device Test - 5
- 2.4 User Acceptance Test - 6

3. Target Test Items

- 3.1 Features for Function Test - 7
- 3.2 Pseudocode for Unit Test - 7
- 3.3 Device Target for Device Test - 8
- 3.4 Parameter of User Acceptance Test - 9

4. Test Milestones - 9

5. Test Roles and Responsibilities - 10

6. Document Tracking - 10

1. Introduction

Jurani merupakan sebuah aplikasi permainan *truth or dare* berbasis android. Tantangan yang disediakan sangat beraneka ragam yang dikemas semenarik mungkin sehingga para penggunanya tidak merasa bosan saat berkumpul bersama. Dokumen ini merupakan fase terakhir dalam *project* ini yang berisi tentang implementasi beserta paparan segala bentuk pengujian yang digunakan selama proses pengembangan aplikasi. Adapun isi dari dokumen *test plan* ini meliputi cakupan dari testing, fitur yang diuji, strategi testing, skenario pengujian, dan juga penanggung jawab untuk setiap pengujiannya.

1.1 Purpose

Tujuan utama dari pembuatan *master test plan* ini ialah menggali segala informasi yang dibutuhkan untuk perencanaan pengujian yang dijadikan sebagai acuan pada saat melakukan uji coba eksekusi aplikasi Jurani.

Adapun tujuan dari pembuatan *test plan* adalah sebagai berikut:

1. Mengidentifikasi fitur-fitur yang akan dilakukan uji coba pada beberapa *target device*
2. Menentukan *test types* apa saja yang harus diujikan pada aplikasi
3. Menjelaskan alur proses pengujian beserta input dan *output*-nya yang disajikan dalam bentuk *test scenario*
4. Membuat *timeline* pelaksanaan pengujian untuk setiap *milestones test*-nya

1.2 Test Scope

Test plan ini akan digunakan untuk melakukan proses uji coba kebutuhan yang diberikan oleh aplikasi. Pengujian pertama yaitu *Functional Test* dengan menerapkan teknik *blackbox* untuk mengetahui segala fungsi yang ada di dalam aplikasi dapat berjalan sesuai dengan ekspektasi. Selanjutnya *Unit Test* dengan teknik *whitebox*, yaitu untuk menelusuri algoritma program. Kemudian dilakukan *Device Test* yang bertujuan untuk mengetahui apakah aplikasi dapat berjalan dengan baik di berbagai macam perangkat dengan spesifikasi yang berbeda-beda. Selain itu dilakukan UAT (*User Acceptance Test*) untuk memastikan bahwa aplikasi dapat diterima oleh *user*. Adapun *test scenario* untuk setiap pengujiannya juga

dipaparkan agar para penguji mengerti bagaimana alur beserta *input* dan *output* yang diinginkan para pengembang aplikasi.

1.3 Testing Overview

Pengujian harus dilakukan sesuai dengan langkah-langkah yang disepakati sebelumnya. Adapun langkah-langkah dalam pengujian aplikasi ini adalah sebagai berikut:

1. Melakukan identifikasi kebutuhan apa saja yang akan diuji
2. Menentukan jenis dan metode yang dipakai di dalam pengujian
3. Menentukan data dan skenario apa saja yang akan dipakai dalam pengujian untuk memastikan aplikasi berfungsi dengan baik
4. Menggambarkan perkiraan hasil yang didapatkan berdasarkan pengujian yang telah dilakukan
5. Melakukan pengaturan pengujian
6. Melaksanakan pengujian
7. Hasil pengujian yang gagal harus dibuatkan dokumen yang merekam kegagalan apa saja yang terjadi serta bagaimana cara mengatasinya
8. Dokumen pengujian diberikan kepada pihak yang berkepentingan dan jika ada perubahan akan segera diperbaiki

2. Test Strategy

Pada bagian ini dijelaskan jenis-jenis pengujian apa saja yang akan dilaksanakan beserta deskripsi dan contoh *test scenario*-nya. Pengujian yang dilakukan kepada aplikasi ini menggunakan jenis pengujian *Function Test*, *Unit Test*, *Device Test*, dan *User Acceptance Test*., Adapun penjelasannya akan dijelaskan di bagian selanjutnya.

2.1. Function Test

Function testing merupakan sebuah pengujian terhadap fitur-fitur yang telah dibuat di dalam aplikasi. Fitur yang diuji harus sama dan sesuai dengan yang ada di dalam *system requirements document* tepatnya berada di *use case scenario*. Pengujian ini bertujuan untuk memastikan bahwa semua fitur yang telah dibuat di aplikasi sesuai dengan yang telah

direncanakan sebelumnya dan dapat berjalan dengan baik (sesuai dengan *input* dan *output*-nya). Di dalam pengujian *Function Test* ini menerapkan metode *blackbox*. Berikut ini merupakan contoh *scenario test* yang akan dibuat.

Test Case	Kondisi utama dari proses yang akan dijalankan
Test Case Description	Penjelasan proses yang akan dijalankan
Input (Step to produce test)	Langkah yang harus dilalui untuk mencapai kondisinya
Output (Expected Results)	Hasil yang diharapkan setelah kondisi utama dijalankan

2.2. Unit Test

Unit test merupakan pengujian yang bertujuan untuk menguji semua bagian yang berbeda dari aplikasi yang telah dibuat. Pengujian unit dilakukan pada tingkat *source code* untuk menemukan *error* atau *bug* pada program, *syntax error*, *logic error*, atau menguji suatu unit modul pada program. Adapun metode pengujian yang biasa digunakan dalam *unit testing* adalah *whitebox* dan *blackbox*. Namun di dalam pengujian aplikasi Jurani, tim pengembang hanya berfokus pada pengujian menggunakan metode *whitebox* pada suatu unit modul. Dalam pengujian *whitebox* ini, *source code* suatu fungsi diuji apakah berjalan sesuai dengan spesifikasi yang telah ditentukan. Unit modul yang diuji adalah proses pengacakan tantangan beserta objek tantangan. Kedua bagian ini terdapat pada satu *activity class* yang sama.

Code	Branch
<i>Source code</i> algoritma proses pengacakan tantangan dan target tantangan.	<i>Flowgraph</i> dari pseudocode yang akan diuji jalurnya.

2.3. Device Test

Device testing merupakan sebuah pengujian yang bertujuan untuk menguji aplikasi dengan cara melakukan instalasi dan pengoperasian di beberapa *device*. Parameter-parameter

keberhasilan *device testing* telah didefinisikan terlebih dahulu sebelum melakukan pengujian. Parameter yang dibuat dalam pengujian berdasarkan *minimum system requirements* yang telah ditentukan. Adapun *test scenario* untuk pengujian ini adalah sebagai berikut.

Input (Step to produce test)	Output (Expected Results)	
Langkah yang harus diikuti untuk mencapai kondisinya	Hasil yang diharapkan setelah kondisi utama dijalankan	
Element	Color Palette	Position
Elemen-elemen yang dianalisis	Kode hexagonal dari palette warna	Posisi yang diharapkan

2.4. User Acceptance Test

User Acceptance Test adalah pengujian penerimaan pengguna. UAT merupakan langkah terakhir sebelum meluncurkan aplikasi. Biasanya pengguna akhir (*end user*) yang akan menggunakan aplikasi menguji aplikasi sebelum ‘menerima’ aplikasi. Jenis pengujian ini memberikan pengguna akhir keyakinan bahwa aplikasi yang disampaikan kepada mereka memenuhi persyaratan mereka. Tes ini juga membantu menemukan *bug* yang berhubungan dengan kegunaan dari aplikasi.

Pembuatan UAT ini dibuat menggunakan *tools* Google Form, dengan jenis pertanyaan dalam bentuk *scale* serta satu pertanyaan untuk saran terhadap aplikasi Jurani.

3. Target Test Items

Sebelum menjabarkan elemen-elemen apa saja yang dijadikan sebagai target uji coba, berikut ini merupakan spesifikasi kebutuhan yang diharuskan untuk menjalankan aplikasi Jurani:

- Minimum OS Android 4.0.4, API 15 (Ice Cream Sandwich)
- Layar 5.0 inch untuk tampilan yang optimal

Pada penjelasan di awal bab telah dipaparkan *test types* apa saja yang dijadikan sebagai tolak ukur pengujian. Di bawah ini adalah item-item untuk target uji coba untuk setiap *test types*-nya.

3.1 Features for Function Test

Adapun fitur-fitur yang akan dilakukan pengujian adalah sebagai berikut:

1. Memasukkan jumlah pemain.
2. Memasukkan nama pemain.
3. Mengacak nama pemain yang akan mendapatkan tantangan.
4. Memilih tantangan jujur atau berani.
5. Mengacak tantangan jujur atau berani yang dipilih sebelumnya.
6. Mengacak nama pemain sebagai objek tertantang.
7. Melanjutkan permainan
8. Melakukan *reset* permainan.
9. Melihat menu *help*.
10. Melihat menu *Development*.

3.2 Pseudocode for Unit Test

Berikut adalah pseudocode dari modul pengacakan tantangan dan objek tantangan. Untuk melakukan proses pengujian dibuat *branch* dan analisis jalur (*path*).

```
i      <- 0
task   <- getTask
truth  <- [array pertanyaan jujur]
dare   <- [array tantangan berani]
while (i < dare.length) {
  switch (task) {
    case "truth":
      text1 <- truth[i]
      if (i == truth.length - 1) {
        i <- 0
      } else {
        i++
      }
    case "dare" :
      text1 <- dare[i]
      if (i == dare.length - 1) {
        i <- 0
      } else {
        i++
      }
  }
```

```

    }
  }
  if interrupted (button clicked), break from loop.
}
print text1.

j      <- 0
target <- [array pemain sisa]
while (j < target.length) {
  text2 <- target[j]
  if (j == target.length - 1) {
    j <- 0
  } else {
    j++
  }
  if interrupted (button clicked), break from loop.
}
print text2.

```

3.3 Device Target for Device Test

Berikut ini merupakan *list device* yang akan dijadikan sebagai target pengujian.

Product	OS Version	Code Name	Display Type	Display Size	Density	CPU	Memory Size (Internal)
Sony Xperia ST23i	4.0.4	Ice Cream Sandwich	TFT, capacitive touchscreen, 16M colors	320 x 480 pixels	3.5 inches (~165 ppi pixel density)	800 MHz Cortex-A5	4 GB (2.2 GB user available), 512 MB RAM
Sony Xperia Z3 D6653	6.0	Marshmallow	TFT, capacitive touchscreen, 16M colors	1920 x 1080 pixels	5.2 inches (~442 ppi pixel density)	Quad-core 2.5 GHz Krait 400	16 GB, 3 GB of RAM
Asus Fonepad 7 k012	5.0	Lollipop	Capacitive touchscreen, 16M colors	600 x 1024 pixels	7.0 inches (~65.3% screen-to-body ratio)	Dual-core 1.2 GHz	8 GB, 1 GB RAM
Samsung Galaxy Duos i8262	4.3	Jellybean	TFT capacitive touchscreen, 16M colors	480 x 800 pixels (~217 ppi pixel density)	4.3 inches (~60.2% screen-to-body ratio)	Dual-core 1.2 GHz Cortex-A5	8 GB (4.5 GB user available), 1 GB RAM

Adapun halaman-halaman yang akan dilakukan uji coba sama dengan halaman yang akan diakses saat *Function Test* dijalankan.

3.4 Parameter of User Acceptance Test

Aspek-aspek dari aplikasi yang akan diuji oleh responden adalah sebagai berikut:

1. Kemudahan penggunaan dan pengaksesan aplikasi
2. Rancangan antarmuka/tampilan aplikasi
3. Kualitas konten aplikasi
4. Kepuasan dan kesetiaan pengguna terhadap aplikasi

4. Test Milestones

Milestones ini akan dijadikan tolak ukur waktu dalam menangani suatu aktivitas di dalam penyelesaian *project*. Dalam hal ini, manajemen yang dilakukan ialah terhadap masing-masing pengerjaan *test types* yang sudah ditentukan sebelumnya.

Milestone	Date	Responsible Person
Functional Test	15/4/2016	Mahendra Sunt Servanda
Unit Test	26/4/2016	Angellica Octavianzy S.
Device Test	23/4/2016	Kenneth M. Albani Hakim
User Acceptance Test	25/4/2016	Sandy Febriant Eka Putra

5. Test Roles and Responsibilities

Berikut adalah daftar peran dan tanggung jawab pada proses pembuatan dokumentasi *Master Test Plan* ini.

Resource	Role	Responsibility
Mahendra Sunt Servanda	Test Manager/Project Manager	<ul style="list-style-type: none">• Mengatur jalannya proses pengujian.• Memastikan proses pengujian berjalan dengan baik.• Memberikan tugas pengujian kepada penguji.
Sandy Febriant Eka Putra	Software Analyst	Menganalisis kekurangan dan kelebihan dari aplikasi dilihat dari kebutuhan apakah sudah terpenuhi atau belum.
Kenneth M. Albani Hakim	UI/UX tester	Melakukan proses pengujian terhadap UI dan UX dari aplikasi.
Angellica Octavianzy S.	Source code tester	Melakukan proses pengujian terhadap source code.

6. Document Tracking

Bagian ini untuk mencatat seluruh *log* pengerjaan dokumentasi *Master Test Plan*.

Date	Action Taken	By Whom
1 April 2016	Created master template	Mahendra Sunt Servanda
7 April 2016	<ul style="list-style-type: none">- Wrote description for some sections- Specified test roles and responsibilities	Angellica Octavianzy S.
10 April 2016	<ul style="list-style-type: none">- Determined test features and the responsible party	Mahendra Sunt Servanda Sandy Febriant Eka Putra

	<ul style="list-style-type: none"> - Detailed inputs and outputs of the process to be tested - Decided all the test strategy that will be implemented 	
11 April 2016	Retouched the documents before being presented	Mahendra Sunt Servanda
14 April 2016	<ul style="list-style-type: none"> - Recreated a new master template by adding more additional section - Described the introduction of master test plan - Gave the explanation of test strategy 	Mahendra Sunt Servanda
23 April 2016	Described the purpose of implementing User Acceptance Test	Kenneth M. Albani Hakim
24 April 2016	Added some explanation for the Introduction section and made the document more structured	Sandy Febriant Eka Putra
25 April 2016	Cleared all the strategy test and target test items sessions	Sandy Febriant Eka Putra
27 April 2016	Finished all MTP documentations	Mahendra Sunt Servanda Sandy Febriant Eka Putra Angellica Octavianzy S. Kenneth M. Albani Hakim