

# **MASTER TEST PLAN**

**KIRI BANG !**

**Versi 1.0**

**Prototype 1**

**9 April 2016**

**Disusun oleh :**

Anneke

Hendry Gustin

Junior

Mahesa

## DAFTAR ISI

<b>1. Pengantar.....</b>	<b>1</b>
<b>2. Tujuan.....</b>	<b>1</b>
<b>3. Gambaran Proses.....</b>	<b>1</b>
<b>4. Alur Proses Pengujian .....</b>	<b>2</b>
<b>5. Strategi Pengujian .....</b>	<b>3</b>
a. Unit Testing .....	4
i. Pengujian White Box.....	4
ii. Pengujian Black Box.....	4
b. Integration Testing .....	5
c. Function Testing .....	5
<b>6. Penelusuran Bug/ Proses Bug .....</b>	<b>6</b>
<b>7. Pembagian Tanggung Jawab.....</b>	<b>7</b>
7.1 Tim Developer & Bug Master .....	7
7.2 Tim Tester & Documentation.....	7
<b>8. Jadwal Pengujian .....</b>	<b>7</b>
<b>9. Deliverables .....</b>	<b>9</b>

## 1. PENGANTAR

Dokumen ini merupakan gambaran tentang tahapan akhir dari pembuatan aplikasi Kiri Bang !, yang meliputi perencanaan tes sampai dengan pelaksanaannya. Tujuan utamanya adalah untuk mengkomunikasikan tentang prosedur dan standar kualitas pada aplikasi. Proses pengujian yang akan dilakukan pada aplikasi Kiri Bang ! meliputi unit testing, integrasi testing, dan sistem testing. Metode yang akan kami gunakan dalam proses pengujian adalah whitebox dan blackbox. Selama proses pengujian kami akan menerapkan spesifikasi dokumentasi pengujian yang dijelaskan dalam Standar IEEE 829-1983 untuk Software Test Documentation.

## 2. TUJUAN

Tujuan pembuatan dokumen uji coba pada aplikasi ini adalah sebagai berikut:

- Melakukan verifikasi dan validasi untuk menghindari adanya error pada aplikasi.
- Mencari dan menemukan bugs, error, dan kelemahan dari aplikasi
- Mengetahui sejauh mana kualitas dari aplikasi
- Meningkatkan integritas aplikasi
- Memastikan bahwa aplikasi sudah siap untuk digunakan.

Dari referensi lain:

- Mengidentifikasi kebutuhan apa saja yang akan diuji
- Menjelaskan pendekatan pengujian yang akan digunakan
- Mendeskripsikan alur proses pengujian yang harus dilakukan
- Menyediakan batas waktu (timeline) dengan milestones untuk tahap pengujian

## 3. GAMBARAN PROSES

Berikut ini adalah gambaran keseluruhan dari proses pengujian :

- Mengidentifikasi persyaratan untuk diuji
- Mengidentifikasi tes yang akan digunakan untuk menguji setiap modul.
- Meninjau data dan uji kasus untuk memastikan bahwa unit telah sepenuhnya diverifikasi secara tepat.

- Mengidentifikasi hasil yang diharapkan untuk setiap tes.
- Mendokumentasikan konfigurasi uji kasus, data uji, dan hasil yang diharapkan.
- Melakukan pengujian.
- Dokumentasi dari data uji, uji kasus, dan konfigurasi uji digunakan selama proses pengujian.
- Keberhasilan unit testing diperlukan sebelum unit memenuhi syarat untuk komponen pengujian sistem.
- Kegagalan dari pengujian memerlukan Bug Report Form. Dokumen ini harus menguraikan kasus uji, masalah yang dihadapi, kemungkinan penyebabnya, dan urutan kejadian yang menyebabkan masalah. Itu akan digunakan sebagai dasar untuk analisis teknis nanti.
- Dokumen uji dan laporan harus disampaikan. Setiap spesifikasi yang ditinjau, direvisi, atau diperbarui harus ditangani segera.

## 4. ALUR PROSES PENGUJIAN



Gambar 1: Alur Proses Pengujian

Diagram di atas akan menggambarkan secara keseluruhan mengenai proses pengujian yang akan dilaksanakan.

1. Persiapan tes, meliputi penyusunan rencana tes, pendekatan dan penjadwalan tes serta menetapkan tanggung jawab dari setiap anggota.
2. Merancang tes, meliputi penentuan siklus tes, Test Cases, kriteria masuk & keluar, hasil yang diinginkan, dsb. Secara umum, hasil yang diinginkan akan ditentukan oleh tim tes dibantu oleh tim pengembang. Tim tes kemudian akan

mencari Test Cases dan data yang dibutuhkan. Test Conditions didapatkan dari dokumen spesifikasi program.

3. Membuat prosedur tes, meliputi pembuatan berbagai macam prosedur seperti sistem manajemen error dan pelaporan status.
4. Menyiapkan kebutuhan tes, meliputi permintaan dan pembangunan hardware, software dan set data yang dibutuhkan.
5. Melakukan tes, semua tes yang ada di dalam prosedur tes akan dilaksanakan. Semua hasil didokumentasikan dan setiap bug akan dicatat lalu diserahkan kepada tim pengembang apabila dibutuhkan.
6. Keluar, proses testing selesai apabila semua kriteria keluar telah terpenuhi.

## 5. STRATEGI PENGUJIAN

Berikut adalah bentuk dari jenis pengujian yang akan dilakukan untuk pengujian unit, pengujian integrasi, dan pengujian fungsi. Apa yang akan diuji, bagaimana pengujian yang dilakukan akan dijelaskan lebih rinci pada dokumen Scenario Test. Template yang akan digunakan untuk merancang kasus penggunaan ditunjukkan pada Gambar 2.

Tested By:		
Test Type		
Test Case Number		
Test Case Name		
Test Case Description		
Item(s) to be tested		
1		
2		
Specifications		
Input		Expected Output/Result
Procedural Steps		
1		
2		
3		
4		
5		
6		
7		

Gambar 2. Template Kasus Uji

## A. UNIT TESTING

Unit Testing merupakan pengujian bagian terkecil dari sebuah *code*, seperti sebuah fungsi, modul, dan class dari sistem. Hal ini dilakukan untuk menguji validitas program, menghindari terjadinya kesalahan sintaks, logika, dan fungsi yang terdapat pada sistem. Pengujian ini bertujuan untuk memastikan dapat berjalan sesuai yang diharapkan. Setiap fungsi di dalam aplikasi ini dijalankan secara independen, oleh karena ini, berikut ini adalah daftar fungsi-fungsi yang akan dilakukan pengujian:

- Fungsi `getFromLocation` (ambil lokasi asal)
- Fungsi `getToLocation` (ambil lokasi tujuan)
- Fungsi `countDistance` (hitung jarak)
- Fungsi `countFare` (hitung ongkos)
- Fungsi `generateNavigation` (cari navigasi)

Pada unit testing ini, setiap fungsi di atas diujikan dengan menggunakan whitebox dan blackbox:

---

### I. PENGUJIAN WHITE BOX

Dalam White Box, pengujian input dan output langsung diujikan pada tingkat kode dan hasilnya dibandingkan terhadap spesifikasi dari aplikasi Kiri Bang!. Pengujian ini akan mengabaikan fungsi dari program yang diuji dan fokus hanya pada kode dan struktur kode itu. Para desainer pengujian diharuskan untuk menghasilkan *case* yang menyebabkan masing-masing kondisi di dalam program untuk dieksekusi minimal sekali. Hal ini dapat terjadi dengan menerapkan *Branch Testing*. Metode ini layak untuk diterapkan karena fungsi dari aplikasi Kiri Bang! ini adalah relatif sederhana.

---

### II. PENGUJIAN BLACK BOX

Pengujian secara Black Box akan menggunakan segala macam kemungkinan input yang dapat diterima oleh aplikasi untuk mengklarifikasi apakah output dari aplikasi adalah benar sebagaimana mestinya digunakan oleh pengguna umum.

## B. INTEGRATION TESTING

Integration Testing merupakan pengujian apakah gabungan dari bagian (fungsi) dari sebuah aplikasi atau sistem dapat bekerja sama dengan benar. Selama pengujian integrasi, kami akan menguji komponen secara terpisah, dan kemudian mengintegrasikan setiap komponen menjadi satu lalu dilakukan pengujian kembali. Karena keterbatasan waktu, tidak semua kasus tes integrasi kami lakukan. Di bawah ini adalah daftar tes integrasi yang harus dilakukan, didokumentasikan dan diuji:

- Jendela Splash
- Jendela Main Menu
- Jendela Daftar Angkot
- Jendela Navigasi
- Jendela Info Angkot
- Jendela Help
- Jendela About

## C. FUNCTION TESTING

Pengujian fungsi terdiri dari pengujian semua spesifikasi dan kebutuhan perangkat lunak, sesuai dengan dokumen SRS. Pada intinya, semua fungsi yang diujikan akan mengikuti semua use case yang ada pada dokumen SRS. Pengujian fungsi sangatlah penting, oleh karena itu kami telah menyediakan dokumentasi dan penjelasan rinci untuk semua fungsi. Berikut adalah daftar fungsi yang diuji:

- Melihat daftar dan informasi angkot
- Melihat trayek/rute setiap angkot
- Memasukkan tempat asal
- Memasukkan tempat tujuan
- Mendapatkan prakiraan tarif, jarak, dan navigasi dari tempat asal ke tujuan
- Melihat petunjuk penggunaan aplikasi
- Melihat *About*

## 6. PENELUSURAN BUG/ PROSES BUG

Selama pengujian, tim penguji biasanya menghadapi aktivitas-aktivitas yang tidak sesuai dengan design requirement produk. Aktivitas-aktivitas ini kemudian didokumentasikan kemudian mengembalikannya kembali pada tim pengembang untuk diperbaiki.

Ekspektasi terhadap sebuah bug:

- Menelusuri jenis versi aplikasi yang ditemukan bug di dalamnya.
- Menentukan apakah bug sudah didokumentasikan sebelumnya
- Mengindikasi langkah-langkah untuk memperbaiki bug, mencatat rincian yang cukup jelas sehingga dapat dipahami oleh orang lain dan dapat menduplikasi bug tersebut
- Hasil yang aktual, lebih fokus pada apa yang telah ditemukan
- Hasil yang diharapkan, bagaimana aplikasi yang sesuai dengan kebutuhan yang telah disebutkan sebelumnya
- Implikasi, bagaimana kekeliruan dapat mempengaruhi kualitas suatu aplikasi

Tabel berikut menjelaskan tingkatan-tingkatan impact yang biasanya digunakan ketika menemukan suatu bug.

Impact	Definisi
<b>1- Fatal</b>	<b>Tes Stopper:</b> Jika Anda tidak dapat mengakses fungsi dan perlu segera memperbaiki bug. Cacat mencegah QA dari menguji area fitur, sub-wilayah atau fungsi dari fitur.
<b>2- Serious</b>	<b>Beta Stopper:</b> Ini adalah bug yang pengguna akan alami seperti: data rusak, kesalahan perhitungan, data yang tidak benar, UE dan sistem crash pada skenario pengguna umum, risiko QA signifikan, dan cacat UI utama.
<b>3- Minor</b>	<b>Live Release:</b> Sebuah bug yang harus diperbaiki sebelum produk resmi selesai, UE atau crash, konten, dan UI dan perubahan grafis diperlukan untuk rilis.



## 7. PEMBAGIAN TANGGUNG JAWAB

### 7.1 TIM DEVELOPER & BUG MASTER

Tim pengembang umumnya terdiri dari programmer dan bertugas untuk mengembangkan dan menyelesaikan aplikasi Kiri Bang! sesuai dengan metode pengembangan yang dipilih. Setelah aplikasi selesai maka tugas tim ini adalah untuk memelihara dan memperbaiki bug. Setelah bug telah diperbaiki, anggota tim ini kemudian mencatat dan mendokumentasikan bug tersebut. Untuk peran ini akan dijalankan oleh:

- Hendry Gustin
- Anneke Annassia P

### 7.2 TIM TESTER & DOCUMENTATION

Tugas dari tim adalah untuk melakukan segala pekerjaan yang bersifat dokumen dan menjalankan upaya pengujian, yang berarti bahwa setiap tester dari tim pengujian mencoba untuk menemukan bug. Peran ini sangatlah penting untuk menjaga kualitas dari aplikasi yang dihasilkan. Jika sebuah bug ditemukan dia mengisi laporan bug dan mengirimkannya ke tim Bug Master untuk diperbaiki. Untuk peran ini akan dijalankan oleh:

- Junior Lie
- Mahesa Sunt Servanda

## 8. JADWAL PENGUJIAN

Jadwal pengujian merupakan penjadwalan proyek secara keseluruhan, yang terdiri dari fase-fase dan milestones yang berkaitan dengan *quality assurance*. Jadwal pengujian ini membahas tujuan dari pengujian yang dilakukan serta standar-standar yang diharapkan tercapai pada setiap fase/milestone yang kemudian akan dikembangkan. Tanggal-tanggal penting untuk keseluruhan pengembangan Aplikasi Kiri Bang! Beserta dengan pengujiannya terdapat pada outline di bawah ini.

Kiri Bang! Milestone	Tanggal Dilakukan	Catatan	QA deliverables
Analisis dan Perencanaan		Pada Milestone ini, perencanaan tingkat tinggi harus diselesaikan, diantaranya adalah Rencana Proyek, spesifikasi fungsi Program.	Kegiatan perencanaan pengujian tingkat tinggi, yang meliputi pengembangan awal Rencana Master QA (dokumen ini, dan jadwal QA)
Perancangan		Persyaratan dan inisiatif ditetapkan lebih lanjut dan solusi diselesaikan. Deliverable untuk tahap ini adalah source code Program dan dokumen desain terkait lainnya.	Pengembangan dan pengujian berpartisipasi aktif dalam desain fitur dengan memeriksa dan meninjau persyaratan dan dokumen desain. Setelah dokumen desain selesai, pengujian didorong untuk mulai bekerja pada dokumen Rencana Test dan uji perencanaan desain
Penyelesaian Kode - Infrastruktur		Milestone ini adalah ketika semua pembangunan infrastruktur dan fungsi harus lengkap. Tim pengujian seharusnya melakukan pengujian Unit & Integrasi sebelum memeriksa kode program	Pengujian harus telah menyelesaikan atau dalam tahap akhir dari awal Rencana Infrastruktur Uji mereka, kasus uji dan dokumen QA lain yang terkait untuk menguji eksekusi untuk setiap fitur atau komponen seperti skenario pengujian, hasil yang diharapkan, set data, prosedur pengujian, script dan berlaku alat pengujian.
Penyelesaian Kode - Fungsi		Milestone ini termasuk unit testing dan review kode masing-masing komponen fungsi sebelum memeriksa kode ke tahap uji coba. Deliverable termasuk spesifikasi sistem pengujian, spesifikasi pengujian Unit, rencana Integrasi	Pengujian harus menyediakan Kode Uji Penilaian Lengkap untuk Insinyur Pengembangan. Pengujian juga harus telah menyelesaikan atau dalam tahap akhir dari mereka Rencana Uji pendahuluan White Box, kasus uji dan dokumen QA lain yang terkait untuk menguji eksekusi untuk setiap fitur atau komponen seperti skenario pengujian, hasil yang

			diharapkan, set data, prosedur pengujian, script dan alat pengujian yang berlaku.
Penyelesaian Fitur		Fase ini memungkinkan untuk pembersihan fitur untuk memverifikasi sisa perbaikan bug	Semua bug diverifikasi dan dokumentasi QA diselesaikan
Ship / Live		Produk keluar	Sebuah dokumen Pengujian yang belum selesai harus lengkap

## 9. DELIVERABLES

- Spesifikasi fungsi program
- Program source code
- User manual
- Software Requirement Specification
- Master Test Plan - berisi tentang tujuan pengujian, kriteria, jadwal, dan standar pengujian.
  - Rencana Unit Testing
  - Rencana Integration Testing
  - Rencana Function Testing
- Scenario Test dan Hasil :
  - Unit white-box test – batasan kriteria White Box Testing, metode, kasus uji dan hasil uji
  - Unit black-box test – batasan kriteria BlackBox Testing, metode, kasus uji dan hasil uji
  - Integration Test report – batasan kriteria Integration Testing, metode, kasus uji dan hasil uji
  - Function Test report – batasan kriteria Function Testing, metode, kasus uji, dan hasil uji