Master Test Plan

< KIRI BANG ! >
Versi 1.0

9 April 2016

Disusun oleh :
Anneke
Hendry Gustin
Junior
Mahesa

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:

- Menginditifikasi 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

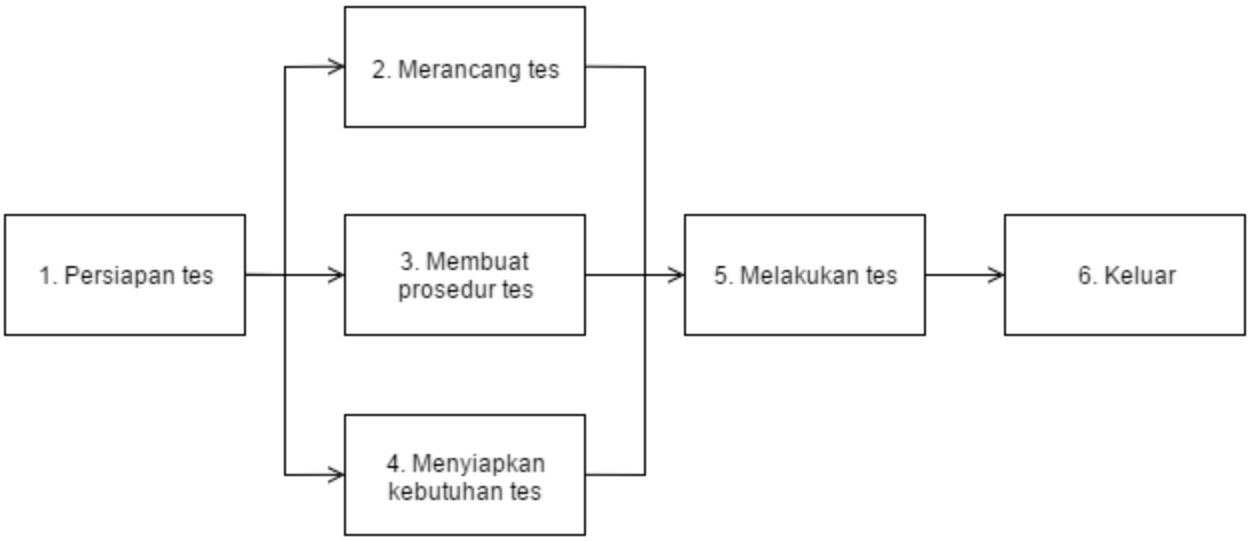
3. Gambaran Proses

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. Informasi ini disampaikan melalui Unit/System Test Report (STR).
- 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 sistem. Termasuk apa yang akan diuji, spesifik (use case) yang menentukan bagaimana pengujian yang dilakukan akan dijelaskan lebih rinci pada Dokumen Perencanaan Tes. 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

5.1. 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. Pada unit testing ini, kami akan menguji setiap fungsi, modul, dan class pada aplikasi Kiri Bang ! dengan menggunakan whitebox dan blackbox.

5.1.1. 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.

Setiap fungsi di dalam aplikasi ini dijalankan secara independen, oleh karena ini, berikut ini adalah daftar fungsi-fungsi yang akan dilakukan pengujian menggunakan metode white box:

- A
- A
- A
- A

5.1.2. 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. Kami menggunakan Equivalence Partitioning dan Boundary Value Analysis untuk aplikasi Kiri Bang!.

5.1.2.1 Equivalance Partitioning

Berdasarkan input yang dapat diterima oleh aplikasi kami, Eh ini blackboxnya kita kan ga ada input" angka begini.. Kalo blackboxnya kyk skripsi aja gmn? MAKSUDNYA GIMANA?

5.2. Integration Testing

Integration Testing merupakan pengujian apakah gabungan dari bagian (fungsi) dari sebuah aplikasi atau sistem dapat bekerja sama dengan benar.

5.2.1 Incremental Testing

Dua modul utama yang harus digabungkan antara lain adalah modul GUI (Graphic User Interface) dan modul Tree Repository (back-end). Kedua komponen ini, ketika tergabung akan membentuk Aplikasi Kiri Bang!. Berikut ini akan dijelaskan mengenai kedua modul dan langkah-langkah untuk melakukan integrasi yang lengkap, dengan menggunakan strategi *Incremental Testing.*

Modul 1 - Graphic User Interface

Modul ini menyediakan sebuah tampilan GUI sederhana dimana penggunanya dapat melakukan berbagai aksi atau fungsi. Modul ini akan diujikan secara terpisah dari back-end untuk memastikan apakah setiap antarmuka (contohnya, button Help) bekerja sesuai dengan fungsinya masing-masing. Secara umum, setiap fungsi yang menggunakan mouse harus berjalan dengan benar. Pengujian akan dilakukan dengan menuliskan rintisan dari setiap elemen yang ada di antarmuka.

Modul 2 - Tree Repository Backend Module

GW GA NGERTI YANG INI HIKS, KYKNYA INI BERHUBUNGAN DENGAN ALGORITMA YANG DIPAKE PROGRAMNYA GAK SIH

Setelah GUI digabungkan dengan modul back-end, maka aplikasi Kiri Bang! Kini telah lengkap. Untuk mencapai integrasi yang sempurna maka setiap elemen di dalam GUI perlu diujikan dengan cara menggantikan setiap rintisan yang ada dengan fungsi back-end. Hasilnya akan ditampilkan menggunakan GUI dan menggunakan *incremental*

method. Setiap rintisan akan diujikan satu per satu sampai semua rintisan telah tergantikan dengan fungsi back-end.

5.3. System Testing

System Testing merupakan pengujian dari keseluruhan sIstem yang ada. Secara umum, System Testing akan memperhitungkan aspek performa, keamanan, validasi, load/stress, dan sensitivas program terhadap konfigurasi. Tapi kali ini kami hanya memfokuskan kepada validasi fungsi dan performa dari aplikasi, dengan menggunakan metode pengujian black-box.

5.3.1 Function Validation Testing

Aplikasi Kiri Bang! akan diuji berdasarkan persyaratan untuk memastikan bahwa kami membangun aplikasi yang tepat. Kami akan mencoba untuk menemukan kesalahan dalam input dan output, oleh karena itu kami akan menguji setiap fungsi untuk memastikan bahwa itu benar mengimplementasikan algoritma NAVIGASINYA APA. Peran dan algoritma dari masing-masing fungsi dapat dilihat pada Spesifikasi Software Program.

5.3.2 Performance Testing

Berhubungan dengan database? Atau navigasi?
Yg performance testing, kalo ttg db susah test nya, atau navigasi aja kayanya.

6. Entry and Exit Criteria

Bagian ini menjelaskan kriteria umum dimana pengujian dimulai, dihentikan, dilanjutkan dan diselesaikan dalam setiap tahap pengujian. Beberapa fitur atau komponen mungkin memiliki sedikit variasi dari kriteria mereka, dalam hal ini, harus dimasukkan dalam *feature test plan*.

6.1 Unit Testing

Unit Testing dilakukan pada tingkat kode pemrograman seperti sintaks yang buruk, kesalahan logika, atau untuk menguji fungsi atau modul tertentu. Unit Test Cases harus dirancang untuk menguji tingkat validitas program.

6.1.1 Black Box Phase

Pengujian Black-Box biasanya menelusuri setiap input yang memungkinkan, untuk memverifikasi bahwa hasil output tepat sesuai dengan sebagaimana mestinya digunakan oleh pengguna akhir. Kami akan menggunakan ????????? dalam rangka pada banyaknya menentukan berapa banyak kasus uji yang diperlukan untuk mencakup kode program secara maksimal.

6.1.1.1 Black Box Entry Criteria

Kriteria Masuk Black Box akan bergantung pada spesifikasi komponen, dan kebutuhan user interface. Hal-hal yang harus dilakukan pada masuk ke tahap Black Box:

- Semua fungsi baik di dalam kode program atau stubs harus sudah dibuat.
- Jenis metode pengujian Black Box akan ditentukan pada saat dimulai. Kami akan menggunakan ?????????

6.1.1.2 Black Box Exit Criteria

Black Box Exit Criteria di bawah menjelaskan apa yang perlu diselesaikan untuk keluar dari fase Black Box. Untuk keluar dari fase ini, tingkat keberhasilan Black Box 100% harus dicapai. Hal-hal yang harus dilakukan saat keluar dari fase Black Box:

- Semua bug yang ditemukan telah dikoreksi.

6.1.2 White Box Phase

Kriteria White Box berlaku untuk tujuan yang berfokus pada struktur internal program dan menemukan semua kesalahan internal program. Setiap kesalahan akan dikategorikan dan kualitas produk akan dinilai.

6.1.2.1 White Box Entry Criteria

Kriteria masuk White Box akan bergantung pada para *Quality Assurance* memverifikasi bahwa fitur, desain dan antarmuka yang stabil. Hal-hal yang harus dilakukan pada masuk ke tahap White Box:

- Semua fungsi baik di dalam kode program atau stubs harus sudah dibuat.
- Jenis metode pengujian White Box akan ditentukan pada saat dimulai. Kami akan menggunakan ?????????
- Pengujian Black Box harus dalam tahap akhir.

Setelah kriteria White Box telah dipenuhi, produk memasuki tahap White Box. Selama tahap White Box dijalankan maka para pengembang akan memperbaiki produk dan memperbaiki cacat yang ditemukan.

6.1.2.2 White Box Exit Criteria

Pengujian White Box terus dijalankan sampai Black Box atau kriteria keluarnya terpenuhi. Untuk keluar dari fase White Box, tingkat keberhasilan 100% harus dicapai. Berikut ini menggambarkan keadaan produk setelah keluar dari fase White Box:

- Semua fungsi telah diimplementasikan, bisa dijalankan dan telah diujikan.
- Antarmuka grafis telah ditinjau, stabil dan mampu memuaskan QA, tidak ada perubahan lebih lanjut untuk kotak dialog atau elemen antarmuka lainnya yang direncanakan. Perubahan kecil dapat diterima, tetapi harus diatur dengan tim pengembang dan pengujian.
- Semua bug yang ditemukan telah dikoreksi.

6.2 Integration Testing

Ada dua modul yang akan diintegrasikan untuk *Integration Testing*. Dua modul adalah modul GUI dan modul ????? (back-end). Berikut ini adalah kriteria masuk dan keluar dari *Integration Testing*.

6.2.1 Integration Test Entry Criteria

Kriteria masuk/mulai *Integration Test* mengandalkan kedua modul untuk telah beroperasi. Desain algoritma dan antarmuka harus mencapai tingkat yang stabil. Hal-hal yang harus dilakukan untuk masuk ke tahap Integration Test:

- Semua fungsi baik di dalam kode program atau stubs harus sudah dibuat.
- GUI harus sudah diciptakan baik dalam kode program atau stubs.
- Antarmuka dan interaksi antara ??? Modul dan Graphical User Interface harus sudah operasional.
- Sebuah strategi *bottom-up Integration Testing* akan dilakukan.
- Pengujian Black Box sudah dalam tahap akhir atau selesai.
- Pengujian White Box harus sudah dimulai.

6.2.2 Integration Test Exit Criteria

Kriteria keluar dari *Integration Test* mengandalkan kedua modul untuk telah beroperasi. Desain algoritma dan antarmuka harus mencapai tingkat yang stabil. Untuk keluar dari fase tingkat keberhasilan Integrasi Pengujian 100% harus dicapai. Hal-hal yang harus dilakukan untuk keluar dari tahap Integration Test:

- Semua bug yang ditemukan telah dikoreksi.
- Kedua modul telah berinteraksi dengan akurasi yang baik, sesuai dengan *System Specification Design*. Semua perbedaan harus dikoreksi.

- Kedua modul siap untuk *System Testing*. Stubs dan driver akan diganti dengan kode yang berfungsi penuh.
- Black Box Testing selesai.
- White Box Testing harus sudah dalam tahap akhir atau selesai.

6.3 System Testing

System Testing diujikan untuk menemukan dan mengkategorikan setiap kekurangan dan menilai tingkat kualitas produk. Semua elemen dari kedua modul yang telah menyatu akan diuji secara keseluruhan. *System Testing* berfokus pada fungsi dan kinerja, keandalan, perilaku selama kondisi khusus, dan *stress testing*.

6.3.1 System Test Entry Criteria

Kriteria masuk harus sudah disetujui oleh para pengembang sebelum *System Test* dapat dimulai. Oleh karena itu, *System Test* dapat dimulai jika para pengembang dan penguji berada dalam persetujuan bahwa risiko dari pengujian dapat dikelola dengan baik.

- Kedua modul harus sudah berfungsi penuh.
- Semua kode program harus melalui *Unit Testing*. *Unit Testing* harus diselesaikan dan ditandatangani oleh tim pengembang.
- Semua kebutuhan hardware dan pengujian harus sudah tersedia dan siap digunakan untuk *System Test*.
- Semua pengujian Black Box harus lengkap dan setiap bug sudah diperbaiki.
- Semua pengujian White Box harus lengkap dan setiap bug sudah diperbaiki.
- Integration Test harus lengkap dan setiap bug sudah diperbaiki
- Function Validation Testing adalah metode yang dapat digunakan untuk melakukan pengujian semua fungsi. Graphical User Interface akan menjadi wadah untuk berinteraksi dengan sistem, sehingga GUI juga akan diujikan secara menyeluruh.
- Tim pengembang dan pengujian setuju bahwa *Function Validation Testing* akan meliputi aspek kinerja fungsi, keandalan, *stress* dan *load testing*.

6.3.2 System Exit Criteria

Kriteria keluar harus memenuhi semua kriteria yang tercantum di bawah ini. Hal ini membuktikan bahwa semua elemen dari proyek ini adalah benar. Hal ini untuk memastikan bahwa semua fungsi sistem sudah sesuai dengan *System Specification Document.*

- *Function Validation Testing* memiliki tingkat keberhasilan 100 persen. Pengujian untuk semua fungsi memiliki akurasi yang baik.
- Tidak ada degradasi kinerja sistem di platform yang berbeda dari sistem operasi Windows yang dijalankan. (Windows 95 atau di atas dapat diterima)
- The Graphical User Interface memenuhi *System Specification Requirements.*
- Semua properti ???? disajikan dengan benar melalui Graphical User Interface.
- Semua bagian penerimaan input dari Graphical User Interface bekerja dengan benar.
- Semua error dengan prioritas yang tinggi dari *System Testing* harus diperbaiki dan diuji.
- Jika ada error dengan prioritas yang sedang atau rendah yang tidak biasa, maka tim pengembang dan manajer pengujian harus menyatakan bahwa risiko implementasi dapat diterima.

6.4 Shipping or Live Release

KALO DI TEMPLATE KAN ADA REGRESSION TESTING GITU, KITA ENGGAK. TRUS GIMANA

6.4.1
6.4.2

7. 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**
1- Fatal	

8. Pembagian Tugas ?

8.1 Tim Pengembang

Pengembangan Kode - Hendry Gustin

Anneke Annassia P

8.2 Tim Penguji

Junior Lie

Mahesa Sunt Servanda

9. 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, Usability Testing, Code Complete Acceptance, Beta Testing, Integration Testing, Regression Testing, System Testing.

Tanggal-tanggal penting untuk keseluruhan pengembangan Binary Tree dan Pengujian terdapat pada outline di bawah ini. Penjadwalan yang lebih rinci merujuk pada Jadwal Binary Tree Project (dokumen ini) sedangkan Engineering QA deliverables merujuk pada dokumen perencanaan pengujian.

TABEL

10. Deliverables

- Spesifikasi fungsi program
- Program source code
- Dokumen perencanaan pengujian - dokumen ini sebaiknya berisi tujuan pengujian, kriteria, standar, penjadwalan dan penugasan, serta alat bantu uji.
    - Perencanaan untuk pengujian unit
    - Perencanaan untuk integrasi
    - Perencanaan untuk pengujian sistem
- Dokumen Desain Pengujian
    - Desain untuk unit white-box, meliputi kriteria pengujian white box, metode dan kasus uji
    - Desain untuk unit black-box, meliputi kriteria pengujian black box, metode dan kasus uji
    - Desain untuk pengujian sistem, meliputi kriteria pengujian sistem, metode dan kasus uji

- Dokumen Laporan Pengujian
    - Laporan unit pengujian white box, meliputi hasil pengujian unit white box, masalah, kesimpulan dan analisis
    - Laporan unit pengujian black box, meliputi hasil pengujian unit black box, masalah, kesimpulan dan analisis
    - Laporan pengujian sistem, meliputi hasil pengujian sistem, masalah, kesimpulan dan analisis