

Scenario Test

< KIRI BANG ! >

Versi 1.0

9 April 2016

Disusun oleh :

Anneke

Hendry Gustin

Junior

Mahesa

1. Pengantar

Dokumen ini meliputi semua obyektif, fokus, dan juga strategi yang digunakan dalam pengujian kami. Selain itu, struktur dan peralatan yang digunakan untuk pengujian White Box dan pengujian tingkat fungsional juga dibahas. Hasil dan perencanaan yang terlibat dalam pengujian kami dikerjakan secara manual.

1.1 Tujuan

Tujuan utama untuk pengujian kami adalah untuk mengenal proses setiap kasus pengujian dan menghasilkan data uji. Sementara tujuan lain dari pengujian ini tentunya untuk mencari dan menemukan bugs, error, dan kelemahan dari aplikasi, mengetahui sejauh mana kualitas dari aplikasi, serta memastikan bahwa aplikasi sudah siap untuk digunakan.

1.2 Gambaran Proses Pengujian

Tim kami akan melakukan beberapa pengujian:

1. Unit Testing

Pengujian ini akan dilakukan dengan kotak hitam dan pengujian kotak putih, yang akan menggunakan *Branch Testing*.

2. Integration Testing

Pengujian ini memungkinkan pengujian dari semua unit yang diuji secara individual untuk diujikan secara keseluruhan.

3. Function Testing

Pengujian ini akan memastikan bahwa semua *use case* telah dilaksanakan dengan benar dengan memverifikasi jika mereka hadir di dalam aplikasi.

2. Skenario dan Hasil Tes

Bagian ini akan menggambarkan strategi yang direkomendasikan untuk merancang dan menerapkan tes yang diperlukan. Kami akan menjelaskan rincian tes yang perlu dilakukan untuk setiap item tes yang telah diidentifikasi sebelumnya. Tes yang akan dilakukan sebagai berikut :

- Unit Testing
- Integration Testing

- Function Testing

Selain itu, untuk masing-masing tes ini, setiap *test case* akan dijelaskan secara rinci. Untuk setiap *test case*, kami akan memberikan deskripsi, input atau langkah-langkah untuk melakukan *test case* tersebut, dan output atau hasil yang diharapkan.

2.1 Unit Testing

Unit testing akan menguji komponen beserta dengan fungsinya secara individu dengan fungsi. Ini pengujian tingkat rendah yang mencakup pengujian Black Box dan pengujian White Box. Dalam pengujian kotak hitam, batas-batas fungsi ini akan diuji untuk melihat apakah terjadi kesalahan di sana. pengujian kotak putih akan memverifikasi bahwa semua jalur dalam fungsi yang benar melalui basis Branch Testing.

2.1.1 Fungsi *getFromLocation*

Tes akan dilakukan pada fungsi untuk mengambil lokasi asal berdasarkan pilihan user pada *combo box* yang telah tersedia. Lokasi tersebut akan digunakan lebih lanjut untuk menghitung jarak, tarif dan pencarian navigasi.

2.1.1.1 Black Box Testing

Setiap test case akan diujikan ketika combo box dalam keadaan tertutup.

Test Case 1: Memilih salah satu pilihan combo box ketika pilihan awal masih kosong.

Test Case 2: Memilih salah satu pilihan combo box ketika pilihan awal sudah ada.

Nama penguji:		Junior Lie
Tipe tes:		Unit
Nomor test case:		1
Nama test case:		getFromLocation_empty
Deskripsi test case:		Memilih salah satu pilihan combo box ketika pilihan awal masih kosong.
Item yang diujikan		
1	Melihat apakah hasil yang didapatkan sesuai dengan pilihan combo box.	

Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Stasiun Bogor	Stasiun Bogor	Stasiun Bogor	No
Langkah pengujian			
1			
2			
3			

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor <i>test case</i> :		2	
Nama <i>test case</i> :		getFromLocation_unempty	
Deskripsi <i>test case</i> :		Memilih salah satu pilihan combo box ketika pilihan awal sudah ada.	
Item yang diujikan			
1	Melihat apakah hasil yang didapatkan sesuai dengan pilihan combo box.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Botani Square	Botani Square	Botani Square	No
Langkah pengujian			
1			
2			
3			

2.1.1.2 White Box Testing

Basis Path Testing

[ISI DISINI]

2.1.2 Fungsi *getToLocation*

Tes akan dilakukan pada fungsi untuk mengambil lokasi tujuan berdasarkan pilihan user pada *combo box* yang telah tersedia. Lokasi tersebut akan digunakan lebih lanjut untuk menghitung jarak, tarif dan pencarian navigasi.

2.1.2.1 Black Box Testing

Setiap test case akan diujikan ketika *combo box* dalam keadaan tertutup.

Test Case 1: Memilih salah satu pilihan *combo box* ketika pilihan awal masih kosong.

Test Case 2: Memilih salah satu pilihan *combo box* ketika pilihan awal sudah ada.

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		1	
Nama test case:		getToLocation_empty	
Deskripsi test case:		Memilih salah satu pilihan combo box ketika pilihan awal masih kosong.	
Item yang diujikan			
1	Melihat apakah hasil yang didapatkan sesuai dengan pilihan combo box.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Stasiun Bogor	Stasiun Bogor	Stasiun Bogor	No
Langkah pengujian			
1			
2			
3			

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		2	
Nama test case:		getToLocation_unempty	
Deskripsi test case:		Memilih salah satu pilihan combo box ketika pilihan awal sudah ada.	
Item yang diujikan			
1	Melihat apakah hasil yang didapatkan sesuai dengan pilihan combo box.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Botani Square	Botani Square	Botani Square	No
Langkah pengujian			
1			
2			
3			

2.1.2.2 White Box Testing

Basis Path Testing

[ISI DISINI]

2.1.3 Fungsi *countDistance*

Tes akan dilakukan pada fungsi untuk melakukan perhitungan jarak berdasarkan kedua input lokasi yang telah diterima sebelumnya. Lokasi tersebut adalah lokasi awal dan lokasi tujuan berupa variabel string.

2.1.3.1 Black Box Testing

Pengujian akan dilakukan ketika dua buah input variabel lokasi telah diterima, lalu fungsi *countDistance* akan melakukan perhitungan jarak antar kedua variabel tersebut

menggunakan data yang diambil dari database dengan perhitungan matematika sederhana.

Test Case 1: Mendapatkan dua buah variabel lokasi yang berada di dalam satu jalur angkot yang sama.

Test Case 2: Mendapatkan dua buah variabel lokasi yang tidak berada di dalam satu jalur angkot yang sama.

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		1	
Nama test case:		countDistance_samePath	
Deskripsi test case:		Mendapatkan dua buah variabel lokasi yang berada di dalam satu jalur angkot yang sama.	
Item yang diujikan			
1	Melihat apakah hasil perhitungan jarak yang didapatkan sesuai atau mendekati dengan perhitungan jarak sesungguhnya.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Stasiun Bogor, Botani Square	[ISI DISINI] km	[ISI DISINI] km	No
Langkah pengujian			
1			
2			
3			

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor <i>test case</i>:		2	

Nama <i>test case</i> :		countDistance_differentPath	
Deskripsi <i>test case</i> :		Mendapatkan dua buah variabel lokasi yang tidak berada di dalam satu jalur angkot yang sama.	
Item yang diujikan			
1	Melihat apakah hasil perhitungan jarak yang didapatkan sesuai atau mendekati dengan perhitungan jarak sesungguhnya.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Peppino House, Pusat Grosir Bogor	[ISI DISINI] km	[ISI DISINI] km	No
Langkah pengujian			
1			
2			
3			

2.1.3.2 White Box Testing

Basis Path Testing

[ISI DISINI]

2.1.4 Fungsi countFare

Tes akan dilakukan pada fungsi untuk melakukan perhitungan tarif berdasarkan variabel jarak yang telah diterima sebelumnya. Perhitungan tarif yang dilakukan cukup sederhana dengan cara [ISI DISINI].

2.1.4.1 Black Box Testing

Pengujian akan dilakukan ketika input variabel jarak telah diterima, lalu fungsi countFare akan melakukan perhitungan sederhana untuk menentukan perkiraan tarif yang harus dibayarkan untuk mencapai tempat tujuan.

Test Case 1: Mendapatkan variabel jarak = 0.

Test Case 2: Mendapatkan variabel jarak > 5km.

Test Case 3: Mendapatkan variabel jarak > 10km.

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		3	
Nama test case:		countFare_zero	
Deskripsi test case:		Mendapatkan variabel jarak = 0.	
Item yang diujikan			
1	Melihat apakah hasil perkiraan tarif yang didapatkan sesuai atau mendekati dengan tarif sesungguhnya.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
0km	Rp [ISI DISINI]	Rp [ISI DISINI]	No
Langkah pengujian			
1			
2			
3			

		Junior Lie
Tipe tes:		Unit
Nomor test case:		2
Nama test case:		countFare_low
Deskripsi test case:		Mendapatkan variabel jarak > 5km.
Item yang diujikan		
1	Melihat apakah hasil perkiraan tarif yang didapatkan sesuai atau mendekati dengan tarif sesungguhnya.	

Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
8km	Rp [ISI DISINI]	Rp [ISI DISINI]	No
Langkah pengujian			
1			
2			
3			

		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		3	
Nama test case:		countFare_high	
Deskripsi test case:		Mendapatkan variabel jarak > 10km.	
Item yang diujikan			
1	Melihat apakah hasil perkiraan tarif yang didapatkan sesuai atau mendekati dengan tarif sesungguhnya.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
12km	Rp [ISI DISINI]	Rp [ISI DISINI]	No
Langkah pengujian			
1			
2			
3			

2.1.4.2 White Box Testing

Basis Path Testing

[ISI DISINI]

2.2.5 Fungsi generateNavigation

Tes akan dilakukan pada fungsi untuk melakukan perhitungan jarak berdasarkan kedua input lokasi yang telah diterima sebelumnya. Lokasi tersebut adalah lokasi awal dan lokasi tujuan berupa variabel string.

2.1.3.1 Black Box Testing

Pengujian akan dilakukan ketika dua buah input variabel lokasi telah diterima, lalu fungsi generateNavigation akan melakukan pencarian navigasi tersebut menggunakan data yang diambil dari database dengan algoritma sederhana.

Test Case 1: Mendapatkan dua buah variabel lokasi yang berada di dalam satu jalur angkot yang sama.

Test Case 2: Mendapatkan dua buah variabel lokasi yang tidak berada di dalam satu jalur angkot yang sama.

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		1	
Nama test case:		generateNavigation_samePath	
Deskripsi test case:		Mendapatkan dua buah variabel lokasi yang berada di dalam satu jalur angkot yang sama.	
Item yang diujikan			
1	Melihat apakah hasil navigasi yang didapatkan sesuai dengan yang seharusnya ditempuh.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Stasiun Bogor, Botani Square	[ISI DISINI]	[ISI DISINI]	No
Langkah pengujian			
1			
2			

3	
---	--

Nama penguji:		Junior Lie	
Tipe tes:		Unit	
Nomor test case:		2	
Nama test case:		generateNavigation_differentPath	
Deskripsi test case:		Mendapatkan dua buah variabel lokasi yang tidak berada di dalam satu jalur angkot yang sama.	
Item yang diujikan			
1	Melihat apakah hasil navigasi yang didapatkan sesuai dengan yang seharusnya ditempuh.		
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Peppino House, Pusat Grosir Bogor	[ISI DISINI]	[ISI DISINI]	No
Langkah pengujian			
1			
2			
3			

2.1.5.2 White Box Testing

Basis Path Testing

[ISI DISINI]

2.2 Integration Testing

2.3 Function Testing

TEMPLATE :

Nama penguji:			
Tipe tes:			
Nomor <i>test case</i> :			
Nama <i>test case</i> :			
Deskripsi <i>test case</i> :			
Item yang diujikan			
1			
2			
Spesifikasi			
Input	Expected Output	Actual Output	Bug found?
Langkah pengujian			
1			
2			
3			