

# **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.

<b>Nama penguji:</b>		Junior Lie
<b>Tipe tes:</b>		Unit
<b>Nomor test case:</b>		1
<b>Nama test case:</b>		getFromLocation_empty
<b>Deskripsi test case:</b>		Memilih salah satu pilihan combo box ketika pilihan awal masih kosong.
<b>Item yang diujikan</b>		
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:		getFromLocation_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			

#### 2.1.1.2 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			

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

Nama test case:		countDistance_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 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

Integration Testing adalah jenis pengujian dimana perangkat lunak dikombinasikan dan diuji untuk melihat apakah mereka berinteraksi sesuai dengan fungsi dan kebutuhan mereka. Pengujian ini dapat dilakukan secara progresif sampai seluruh sistem telah terintegrasi.

### 2.2.1 Urutan Pengujian

Integration Testing dimulai setelah setiap unit telah diuji secara individual. Untuk menghemat waktu, kami melakukan pengujian ini dengan urutan sebagai berikut:

1. Pertama, kita membagi integrasi menjadi beberapa bagian, lalu masing-masing akan diujikan.
2. Kedua, untuk setiap tes, beberapa *test case* akan dirancang. Dalam setiap *test case*, harus ada tepatnya satu komponen baru yang dianalisis.
3. Ketiga, untuk setiap komponen yang berkaitan kami akan mengujinya terlebih dahulu.

#### 2.2.1.1 Urutan komponen yang diujikan

Dengan mempertimbangkan Integration dan Unit Testing, maka kami membagi pengujian ini ke dalam 7 komponen berikut:

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

#### 2.2.1.2 Prasyarat Unit Testing untuk setiap komponen

Untuk setiap komponen Integration Testing, kami telah membuat daftar prasyarat Unit Testing yang harus dilakukan terlebih dahulu, yaitu:

1. Jendela Splash  
Untuk jendela Splash hanya konstruktor yang perlu diujikan.
2. Jendela Main Menu
  - Button Daftar Angkot, Navigasi, Help, About.Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela Main Menu.
3. Jendela Daftar Angkot
  - List Daftar Angkot.Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela Daftar Angkot.

#### 4. Jendela Navigasi

- Combo Box Tempat Asal, Tempat Tujuan, Button OK, Label Informasi Navigasi.

Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela Navigasi.

#### 5. Jendela Info Angkot

- Label Informasi Angkot, Gambar Peta Trayek.

Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela Info Angkot.

#### 6. Jendela Help

- Label Informasi Help, Button Exit.

Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela Help.

#### 7. Jendela About

- Label Informasi About, Button Exit.

Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela About.

### 2.2.2 Metode Pengujian

Integration Testing akan dilakukan dengan menggunakan metode Sandwich karena metode ini merupakan kombinasi antara pengujian bottom-up dan top-down.

Kita bisa mulai integrasi sedini mungkin dalam tahap pengembangan perangkat lunak. Untuk Integration Testing, satu-satunya cara untuk menguji integrasi setiap *test case* adalah untuk menambahkan satu komponen dan mengujinya untuk melihat apakah Komponen tersebut bekerja dengan komponen lainnya.

### 2.2.3 Jendela Splash

Hanya terdapat satu *test case* untuk jendela Splash.

<b>Test Case 1</b>	Memanggil Jendela Splash
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih ikon Kiri Bang! pada smartphone.
<b>Hasil test case:</b>	Ok.

### 2.2.4 Jendela Main Menu

Ada beberapa *test case* untuk jendela Main Menu, yaitu untuk melakukan penambahan, pengujian beberapa button yang ada pada jendela ini, serta untuk keluar dari jendela ini.

<b>Test Case 1</b>	Memanggil Jendela Main Menu
--------------------	-----------------------------

<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih ikon Kiri Bang! pada smartphone, lalu menunggu jendela Splash hingga berakhir.
<b>Hasil test case:</b>	Ok.

<b>Test Case 2</b>	Menambahkan Button Daftar Angkot
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 3</b>	Menambahkan Button Navigasi
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 4</b>	Menambahkan Button Help
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 5</b>	Menambahkan Button About
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 6</b>	Memanggil Jendela Daftar Angkot
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat memanggil jendela Daftar Angkot, dengan cara memilih button Daftar Angkot.
<b>Hasil test case:</b>	Ok.

<b>Test Case 7</b>	Memanggil Jendela Navigasi
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat memanggil jendela Navigasi, dengan cara memilih button Navigasi.
<b>Hasil test case:</b>	Ok.

<b>Test Case 8</b>	Memanggil Jendela Help
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat memanggil jendela Help, dengan cara memilih button Help.
<b>Hasil test case:</b>	Ok.

<b>Test Case 9</b>	Memanggil Jendela About
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat memanggil jendela About, dengan cara memilih button About.
<b>Hasil test case:</b>	Ok.

<b>Test Case 10</b>	Menutup Jendela Main Menu
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat ditutup, dengan cara memilih button “Back” pada smartphone.
<b>Hasil test case:</b>	Ok.

#### 2.2.5 Jendela Daftar Angkot

Ada beberapa *test case* untuk jendela Daftar Angkot, yaitu untuk melakukan penambahan, pengujian list yang ada pada jendela ini, serta untuk keluar dari jendela ini.

<b>Test Case 1</b>	Memanggil Jendela Daftar Angkot
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button Daftar Angkot.
<b>Hasil test case:</b>	Ok.

<b>Test Case 2</b>	Menambahkan List Daftar Angkot
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela



	Daftar Angkot dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 3</b>	Memanggil Jendela Informasi Angkot
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat memanggil Jendela Informasi Angkot, dengan cara memilih salah satu dari list angkot yang ada.
<b>Hasil test case:</b>	Ok.

<b>Test Case 4</b>	Menutup Daftar Angkot
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat ditutup, dengan cara memilih button “Back” pada smartphone.
<b>Hasil test case:</b>	Ok.

#### 2.2.6 Jendela Navigasi

Ada beberapa *test case* untuk jendela Navigasi, yaitu untuk melakukan penambahan, pengujian beberapa combo box dan button yang ada pada jendela ini, serta untuk keluar dari jendela ini.

<b>Test Case 1</b>	Memanggil Jendela Navigasi
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button Navigasi.
<b>Hasil test case:</b>	Ok.

<b>Test Case 2</b>	Menambahkan Combo Box Tempat Asal
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Navigasi dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 3</b>	Menambahkan Combo Box Tempat Tujuan
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Navigasi dipanggil.

<b>Hasil test case:</b>	Ok.
-------------------------	-----

<b>Test Case 4</b>	Menambahkan Button OK
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Navigasi dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 5</b>	Menutup Jendela Navigasi
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat ditutup, dengan cara memilih button “Back” pada smartphone.
<b>Hasil test case:</b>	Ok.

#### 2.2.7 Jendela Info Angkot

Ada beberapa *test case* untuk jendela Info Angkot, yaitu untuk melakukan penambahan, pengujian beberapa label dan gambar yang ada pada jendela ini, serta untuk keluar dari jendela ini.

<b>Test Case 1</b>	Memanggil Jendela Info Angkot
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih salah satu angkot dari list yang ada pada jendela Daftar Angkot.
<b>Hasil test case:</b>	Ok.

<b>Test Case 2</b>	Menambahkan Label Informasi Angkot
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Info Angkot dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 3</b>	Menambahkan Gambar Trayek Angkot
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Info Angkot dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 4</b>	Menutup Jendela Info Angkot
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button “Back” pada smartphone.
<b>Hasil test case:</b>	Ok.

#### 2.2.8 Jendela Help

Ada beberapa *test case* untuk jendela Help, yaitu untuk melakukan penambahan, pengujian label dan button yang ada pada jendela ini, serta untuk keluar dari jendela ini.

<b>Test Case 1</b>	Memanggil Jendela Help
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button Help.
<b>Hasil test case:</b>	Ok.

<b>Test Case 2</b>	Menambahkan Label Help
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Help dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 3</b>	Menambahkan Button Exit
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela Help dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 4</b>	Menutup Jendela Help
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button Exit.
<b>Hasil test case:</b>	Ok.

#### 2.2.9 Jendela About

Ada beberapa *test case* untuk jendela About, yaitu untuk melakukan penambahan, pengujian label dan button yang ada pada jendela ini, serta untuk keluar dari jendela ini.

<b>Test Case 1</b>	Memanggil Jendela About
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button About.
<b>Hasil test case:</b>	Ok.

<b>Test Case 2</b>	Menambahkan Label About
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela About dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 3</b>	Menambahkan Button Exit
<b>Deskripsi test case:</b>	Pengujian ini akan dijalankan secara otomatis ketika jendela About dipanggil.
<b>Hasil test case:</b>	Ok.

<b>Test Case 4</b>	Menutup Jendela About
<b>Deskripsi test case:</b>	Melakukan pengujian apakah jendela ini dapat ditutup, dengan cara memilih button Exit.
<b>Hasil test case:</b>	Ok.

## 2.3 Function Testing

TEMPLATE :

<b>Nama penguji:</b>	
<b>Tipe tes:</b>	
<b>Nomor test case:</b>	
<b>Nama test case:</b>	

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