**Scenario Test**

< KIRI BANG ! >

**Versi 1.0**

**Prototype 1**

**9 April 2016**

**Disusun oleh :**

Anneke

Hendry Gustin

Junior

Mahesa

# **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.*

1. Integration Testing

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

1. Function Testing

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

# **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 yangi 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:** | | Anneke | |
| **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 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Anneke | |
| **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 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih kembali combo box tempat asal, lalu pilih lagi  salah satu tempat yang berbeda dengan pilihan  pada langkah 2. | | |

### **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:** | | Anneke | |
| **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 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Anneke | |
| **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 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 4 | Pilih kembali combo box tempat tujuan, lalu pilih  salah satu tempat yang berbeda dengan pilihan  sebelumnya. | | |

### **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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Anneke | |
| **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 | 2743 meter | 2743 meter | No |
| **Langkah pengujian** | | | |
| 1 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 4 | Tekan tombol “OK”. | | |

#### **2.1.3.2 White Box Testing**

**Basis Path Testing**

Diagram.png

|  |  |
| --- | --- |
| **Path ID** | **Path** |
| 1.1 | 1 - 2 - 3 - 4 - 5 - 6 |
| 1.2 | 1 - 2 - 6 |

### **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 membagi tarif berdasarkan kriteria jarak tertentu.

#### **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 < 7km.

Test Case 3: Mendapatkan variabel jarak >= 7 km dan jarak < 10 km.

Test Case 4: Mendapatkan variabel jarak > 10 km.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Anneke | |
| **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 3500 | Rp 3500 | No |
| **Langkah pengujian** | | | |
| 1 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 4 | Tekan tombol “OK”. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Penguji:** | | Anneke | |
| **Tipe tes:** | | Unit | |
| **Nomor *test case:*** | | 2 | |
| **Nama *test case:*** | | countFare\_low | |
| **Deskripsi *test case:*** | | Mendapatkan variabel jarak > 7km. | |
| **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 4000 | Rp 4000 | No |
| **Langkah pengujian** | | | |
| 1 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 4 | Tekan tombol “OK”. | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Anneke | |
| **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 5000 | Rp 5000 | No |
| **Langkah pengujian** | | | |
| 1 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 4 | Tekan tombol “OK”. | | |

### **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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Anne | |
| **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 | 02 | 02 | No |
| **Langkah pengujian** | | | |
| 1 | Pilih menu “Navigasi” pada jendela Main Menu. | | |
| 2 | Pilih combo box tempat asal, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 3 | Pilih combo box tempat tujuan, lalu pilih salah satu  tempat yang tersedia dalam daftar. | | |
| 4 | Tekan tombol “OK”. | | |

## **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.

1. Jendela Main Menu

* Button Daftar Angkot, Navigasi, Help, About.

Komponen berikut harus diujikan terlebih dahulu sebelum melakukan pengujian Jendela Main Menu.

1. Jendela Daftar Angkot

* List Daftar Angkot.

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

1. Jendela Navigasi

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

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

1. Jendela Info Angkot

* Label Informasi Angkot, Gambar Peta Trayek.

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

1. Jendela Help

* Label Informasi Help, Button Exit.

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

1. 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 1** | | Memanggil Jendela Splash | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih ikon Kiri Bang! pada smartphone. | |
| **Hasil *test case:*** | | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 1** | | Memanggil Jendela Main Menu | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih ikon Kiri Bang! pada smartphone, lalu menunggu jendela Splash hingga berakhir. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 2** | | Menambahkan Button Daftar Angkot | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 3** | | Menambahkan Button Navigasi | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 4** | | Menambahkan Button Help | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 5** | | Menambahkan Button About | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Main Menu dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 6** | | Memanggil Jendela Daftar Angkot | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat memanggil jendela Daftar Angkot, dengan cara memilih button Daftar Angkot. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 7** | | Memanggil Jendela Navigasi | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat memanggil jendela Navigasi, dengan cara memilih button Navigasi. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 8** | | Memanggil Jendela Help | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat memanggil jendela Help, dengan cara memilih button Help. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 9** | | Memanggil Jendela About | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat memanggil jendela About, dengan cara memilih button About. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 10** | | Menutup Jendela Main Menu | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button “Back” pada smartphone. | |
| **Hasil *test case:*** | | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 1** | | Memanggil Jendela Daftar Angkot | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button Daftar Angkot. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 2** | | Menambahkan List Daftar Angkot | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Daftar Angkot dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 3** | | Memanggil Jendela Informasi Angkot | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat memanggil Jendela Informasi Angkot, dengan cara memilih salah satu dari list angkot yang ada. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 4** | | Menutup Daftar Angkot | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button “Back” pada smartphone. | |
| **Hasil *test case:*** | | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 1** | | Memanggil Jendela Navigasi | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button Navigasi. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 2** | | Menambahkan Combo Box Tempat Asal | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Navigasi dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 3** | | Menambahkan Combo Box Tempat Tujuan | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Navigasi dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 4** | | Menambahkan Button OK | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Navigasi dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Junior Lie | |
| **Test Case 5** | | Menutup Jendela Navigasi | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button “Back” pada smartphone. | |
| **Hasil *test case:*** | | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 1** | | Memanggil Jendela Info Angkot | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih salah satu angkot dari list yang ada pada jendela Daftar Angkot. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 2** | | Menambahkan Label Informasi Angkot | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Info Angkot dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 3** | | Menambahkan Gambar Trayek Angkot | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Info Angkot dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 4** | | Menutup Jendela Info Angkot | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button “Back” pada smartphone. | |
| **Hasil *test case:*** | | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 1** | | Memanggil Jendela Help | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button Help. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 2** | | Menambahkan Label Help | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Help dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 3** | | Menambahkan Button Exit | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela Help dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 4** | | Menutup Jendela Help | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button Exit. | |
| **Hasil *test case:*** | | 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 1** | | Memanggil Jendela About | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat dipanggil, dengan cara memilih button About. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 2** | | Menambahkan Label About | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela About dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 3** | | Menambahkan Button Exit | |
| **Deskripsi *test case:*** | | Pengujian ini akan dijalankan secara otomatis ketika jendela About dipanggil. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Hendry | |
| **Test Case 4** | | Menutup Jendela About | |
| **Deskripsi *test case:*** | | Melakukan pengujian apakah jendela ini dapat ditutup, dengan baca memilih button Exit. | |
| **Hasil *test case:*** | | Ok. | |

## **2.3 Function Testing**

Bagian ini akan memastikan bahwa perangkat lunak yang diujikan memenuhi persyaratan dan diterima dengan baik oleh pengguna. Setiap persyaratan akan dikaitkan dengan beberapa *test case*, menggunakan beberapa dengan data yang valid, dan beberapa dengan data yang tidak valid. Meskipun waktu yang ada sangat terbatas, kami telah mencantumkan semua fungsi utama serta *test case* untuk setiap fungsi tersebut. Salah satu manfaat dari pengujian ini adalah tim implementasi dapat memastikan bahwa setiap fungsi telah diimplementasikan dengan benar dan bahwa perangkat lunak telah bekerja baik dalam keadaan normal atau tidak normal.

### **2.3.1 Melihat daftar dan informasi angkot**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Melihat Daftar Angkot | |
| **Deskripsi *test case:*** | | Melihat Daftar Angkot yang tersedia di Bogor. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Daftar Angkot” pada jendela Main Menu. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan daftar angkot yang tersedia di Bogor. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Melihat Informasi Angkot | |
| **Deskripsi *test case:*** | | Melihat Informasi Angkot yang dipilih oleh pengguna. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Daftar Angkot” pada jendela Main Menu. 2. Pilih salah satu angkot yang ada pada daftar. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan informasi angkot yang dipilih pengguna. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Daftar Angkot | |
| **Deskripsi *test case:*** | | Kembali ke Daftar Angkot dari jendela Informasi Angkot. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Daftar Angkot” pada jendela Main Menu. 2. Pilih salah satu angkot yang ada pada daftar. 3. Tekan tombol “Back” pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali daftar angkot yang tersedia di Bogor. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Main Menu | |
| **Deskripsi *test case:*** | | Kembali ke Main Menu dari jendela Daftar Angkot. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Daftar Angkot” pada jendela Main Menu. 2. Pilih salah satu angkot yang ada pada daftar. 3. Tekan tombol “Back” sebanyak 2 kali pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali Main Menu dari Kiri Bang!. | |
| **Hasil *test case:*** | | Ok. | |

### **2.3.2 Melihat trayek/rute setiap angkot**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Melihat Peta Trayek Angkot | |
| **Deskripsi *test case:*** | | Melihat Peta Trayek Angkot yang dipilih oleh pengguna. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Daftar Angkot” pada jendela Main Menu. 2. Pilih salah satu angkot yang ada pada daftar. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan peta trayek angkot yang dipilih pengguna. | |
| **Hasil *test case:*** | | Ok. | |

### **2.3.3 Memasukkan tempat asal**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Memilih Tempat Asal | |
| **Deskripsi *test case:*** | | Melihat dan memilih Tempat Asal untuk input navigasi angkot. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan daftar tempat asal yang ada di Bogor. * Pilihan pengguna tersimpan sebagai input. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Mengganti Tempat Asal | |
| **Deskripsi *test case:*** | | Melihat dan mengganti Tempat Asal yang sudah terisi sebelumnya. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Pilih kembali combo box tempat asal, lalu pilih lagi salah satu tempat yang berbeda dengan pilihan pada langkah 2. | |
| **Output (Hasil yang diharapkan)** | | * Pilihan pengguna yang baru tersimpan sebagai input. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Main Menu | |
| **Deskripsi *test case:*** | | Kembali ke Main Menu dari jendela Navigasi. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Tekan tombol “Back” pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali Main Menu dari Kiri Bang!. | |
| **Hasil *test case:*** | | Ok. | |

### **2.3.4 Memasukkan tempat tujuan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Memilih Tempat Tujuan | |
| **Deskripsi *test case:*** | | Melihat dan memilih Tempat Tujuan untuk input navigasi angkot. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Pilih combo box tempat tujuan, lalu pilih salah satu tempat yang tersedia dalam daftar. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan daftar tempat tujuan yang ada di Bogor. * Pilihan pengguna tersimpan sebagai input. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Mengganti Tempat Tujuan | |
| **Deskripsi *test case:*** | | Melihat dan mengganti Tempat Tujuan yang sudah terisi sebelumnya. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Pilih combo box tempat tujuan, lalu pilih salah satu tempat yang tersedia dalam daftar. 4. Pilih kembali combo box tempat tujuan, lalu pilih salah satu tempat yang berbeda dengan pilihan sebelumnya. | |
| **Output (Hasil yang diharapkan)** | | * Pilihan pengguna yang baru tersimpan sebagai input. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Main Menu | |
| **Deskripsi *test case:*** | | Kembali ke Main Menu dari jendela Navigasi. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Pilih combo box tempat tujuan, lalu pilih salah satu tempat yang tersedia dalam daftar. 4. Tekan tombol “Back” pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali Main Menu dari Kiri Bang!. | |
| **Hasil *test case:*** | | Ok. | |

### **2.3.5 Mendapatkan prakiraan tarif, jarak, dan navigasi dari tempat asal ke tujuan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Mendapatkan prakiraan tarif, jarak dan navigasi | |
| **Deskripsi *test case:*** | | Mendapatkan prakiraan tarif, jarak dan navigasi berdasarkan tempat asal dan tempat tujuan yang dipilih pada jendela Navigasi | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Pilih combo box tempat tujuan, lalu pilih salah satu tempat yang tersedia dalam daftar. 4. Tekan tombol “OK”. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan prakiraan tarif, jarak dan navigasi untuk pengguna. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Main Menu | |
| **Deskripsi *test case:*** | | Kembali ke Main Menu dari jendela Navigasi. | |
| **Input (Langkah Pengujian)** | | 1. Pilih menu “Navigasi” pada jendela Main Menu. 2. Pilih combo box tempat asal, lalu pilih salah satu tempat yang tersedia dalam daftar. 3. Pilih combo box tempat tujuan, lalu pilih salah satu tempat yang tersedia dalam daftar. 4. Tekan tombol “OK”. 5. Tekan tombol “Back” pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali Main Menu dari Kiri Bang!. | |
| **Hasil *test case:*** | | Ok. | |

### **2.3.6 Melihat petunjuk pengunaan aplikasi (Help)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Melihat Jendela Help | |
| **Deskripsi *test case:*** | | Melihat petunjuk pengunaan aplikasi Kiri Bang! pada Jendela Help | |
| **Input (Langkah Pengujian)** | | 1. Pilih button “Help” pada jendela Main Menu. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan jendela Help. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Main Menu | |
| **Deskripsi *test case:*** | | Kembali ke Main Menu dari jendela Help. | |
| **Input (Langkah Pengujian)** | | 1. Pilih button “Help” pada jendela Main Menu. 2. Tekan tombol “Back” pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali Main Menu dari Kiri Bang!. | |
| **Hasil *test case:*** | | Ok. | |

### **2.3.7 Melihat *About***

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Melihat Jendela About | |
| **Deskripsi *test case:*** | | Melihat informasi mengenai pengembang aplikasi Kiri Bang! pada Jendela About | |
| **Input (Langkah Pengujian)** | | 1. Pilih button “About” pada jendela Main Menu. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan jendela About. | |
| **Hasil *test case:*** | | Ok. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | | Mahesa | |
| **Test Case** | | Kembali ke Main Menu | |
| **Deskripsi *test case:*** | | Kembali ke Main Menu dari jendela About. | |
| **Input (Langkah Pengujian)** | | 1. Pilih button “About” pada jendela Main Menu. 2. Tekan tombol “Back” pada smartphone. | |
| **Output (Hasil yang diharapkan)** | | * Menampilkan kembali Main Menu dari Kiri Bang!. | |
| **Hasil *test case:*** | | Ok. | |

TEMPLATE :

Unit Testing:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | |  | |
| **Tipe tes:** | |  | |
| **Nomor *test case:*** | |  | |
| **Nama *test case:*** | |  | |
| **Deskripsi *test case:*** | |  | |
| **Item yang diujikan** | | | |
| 1 |  | | |
| 2 |  | | |
| **Spesifikasi** | | | |
| **Input** | **Expected Output** | **Actual Output** | **Bug found?** |
|  |  |  |  |
| **Langkah pengujian** | | | |
| 1 |  | | |
| 2 |  | | |
| 3 |  | | |

Integration Testing:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | |  | |
| **Test Case 4** | |  | |
| **Deskripsi *test case:*** | |  | |
| **Hasil *test case:*** | |  | |

Function Testing:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama penguji:** | |  | |
| **Test Case** | |  | |
| **Deskripsi *test case:*** | |  | |
| **Input (Langkah Pengujian)** | |  | |
| **Output (Hasil yang diharapkan)** | |  | |
| **Hasil *test case:*** | |  | |

**FLOW GRAPH**

1. Menentukan jarak dari lokasi asal menuju lokasi tujuan:

int temp\_tarif = 0;

if (jarak < 7000) {

temp\_tarif = 3500;

} else if ((jarak >=7000) && (jarak < 10000)){

temp\_tarif = 4000;

} else {

temp\_tarif = 5000;

}

1. Menampilkan navigasi result: