Laporan

Interface Testing

for

Delivery Order Yeski Cafe

Prepared by

Product Owner Taufan Dardiry (5215100154)

Scrum Master Husain Panatas (5215100123)

Development Team Ahsanul Marom (5215100031)

Alfian Fakhruddin F (5215100082)

Fajar Febrianto (5215100083)

Jurusan Sistem Informasi

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

2016

## INTERFACE TESTING

Dilakukan jika modul-modul dan sub-sistem terintegrasi dan membentuk sistem yang lebih besar. Tujuannya untuk medeteksi fault terhadap kesalahan interface atau asumsi yang tidak valid. Interface testing terdiri dari beberapa tipe, yaitu :

* Parameter interfaces
* Shared memory interfaces
* Procedural interfaces
* Message passing interfaces

**Interface Errors**

* Interface misuse

Komponen pemanggil memanggil komponen lainnya dan membuat suatu kesalahan dalam penggunaan interfacenya.

* Interface misunderstanding

Komponen pemanggil salah dalam mengasumsikan behaviour komponen yang dipanggil.

* Timing errors

Komponen yang memanggil dan yang dipanggil beroperasi pada kecepatan yang berbeda.

**Petunjuk Melakukan Interface Testing**

1. Merancang test dimana parameter ke prosedur yang dipanggil berada pada nilai batas extrim
2. Test menggunakan null pointer
3. Perancangan test sehingga komponen yang di test akan fail
4. Menggunakan stress testing pada message passing
5. Pada shared memory systems, variasikan urutan dimana komponen diaktifkan.

**Stress Testing**

Menguji sistem dengan nilai yang melebihi maksimum load. Stressing suatu sistem menyebabkan tidak mudah kerusakan dengan mencek kehilangan service yang tidak diduga ataupun data yang hilang.

**Percobaan Interface Testing Pada Software Yeski Cafe**

Dalam pengetesan Interface Testing pada software Delivery Order Yeski Cafe ini kami menggunakan software yang bernama **Espresso**, sehingga pengetesan berjalan secara otomatis dengan meng-run source code sebagai berikut :

**Test 1 :**

package com.example.user.myorderyeski.UITesting;

import android.support.test.espresso.Espresso;

import android.support.test.espresso.matcher.ViewMatchers;

import android.support.test.rule.ActivityTestRule;

import android.support.test.runner.AndroidJUnit4;

import com.example.user.myorderyeski.MainPage;

import com.example.user.myorderyeski.R;

import org.junit.Rule;

import org.junit.Test;

import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;

import static android.support.test.espresso.action.ViewActions.click;

import static android.support.test.espresso.matcher.ViewMatchers.withId;

@RunWith(AndroidJUnit4.class)

public class MainPageEspressoTest {

@Rule

public ActivityTestRule<MainPage> mActivityRule = newActivityTestRule<>(MainPage.class);

@Test

public void ensureButtonWork() {

onView(ViewMatchers.withId(R.id.belisekarang)).perform(click());

ensureButtonAboutWork();

ensureButtonFaqWork();

ensureButtonOrderWork();

ensureButtonMenuWork();

ensureButtonPromoWork();

}

public void ensureButtonAboutWork() {

onView(withId(R.id.about)).perform(click());

Espresso.pressBack();

}

public void ensureButtonFaqWork() {

onView(withId(R.id.faq)).perform(click());

Espresso.pressBack();

}

public void ensureButtonPromoWork() {

onView(withId(R.id.promo)).perform(click());

Espresso.closeSoftKeyboard();

Espresso.pressBack();

}

public void ensureButtonOrderWork() {

onView(withId(R.id.order)).perform(click());

Espresso.pressBack();

}

public void ensureButtonMenuWork() {

onView(withId(R.id.menu)).perform(click());

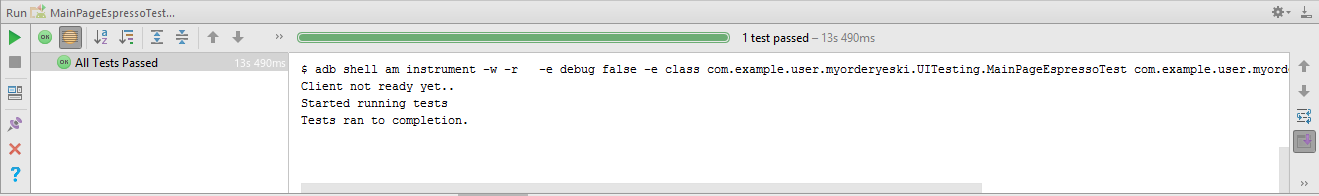
Espresso.closeSoftKeyboard();

Espresso.pressBack();

}

}

**Hasil Run :**



**Test 2 :**

package com.example.user.myorderyeski.UITesting;

import android.support.test.espresso.Espresso;

import android.support.test.rule.ActivityTestRule;

import android.support.test.runner.AndroidJUnit4;

import com.example.user.myorderyeski.MenuMuka;

import com.example.user.myorderyeski.R;

import org.junit.Rule;

import org.junit.Test;

import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;

import static android.support.test.espresso.action.ViewActions.click;

import static android.support.test.espresso.action.ViewActions.typeText;

import static android.support.test.espresso.matcher.ViewMatchers.withId;

@RunWith(AndroidJUnit4.class)

public class MenuMukaEspressoTest {

@Rule

public ActivityTestRule<MenuMuka> mActivityRule = new ActivityTestRule<>(MenuMuka.class);

@Test

public void ensureButtonAboutWork() {

onView(withId(R.id.about)).perform(click());

onView(withId(R.id.BtnMaps)).perform(click());

Espresso.pressBack();

}

@Test

public void ensureButtonFaqWork() {

onView(withId(R.id.faq)).perform(click());

Espresso.pressBack();

}

@Test

public void ensureButtonPromoWork() {

onView(withId(R.id.promo)).perform(click());

onView(withId(R.id.et)).perform(typeText("diskon"));

onView(withId(R.id.btn)).perform(click());

Espresso.closeSoftKeyboard();

Espresso.pressBack();

}

@Test

public void ensureButtonOrderWork() {

onView(withId(R.id.order)).perform(click());

onView(withId(R.id.buttonform)).perform(click());

onView(withId(R.id.formnama)).perform(typeText("aaa"));

Espresso.closeSoftKeyboard();

onView(withId(R.id.formnotelp)).perform(typeText("000"));

Espresso.closeSoftKeyboard();

onView(withId(R.id.formalamat)).perform(typeText("abcdef"));

Espresso.closeSoftKeyboard();

onView(withId(R.id.buttonform)).perform(click());

onView(withId(R.id.formnama)).perform(typeText("Espresso Test"));

Espresso.closeSoftKeyboard();

onView(withId(R.id.formnotelp)).perform(typeText("123456789"));

Espresso.closeSoftKeyboard();

onView(withId(R.id.formalamat)).perform(typeText("Ini hanya percobaan"));

Espresso.closeSoftKeyboard();

onView(withId(R.id.buttonform)).perform(click());

onView(withId(R.id.btnNol)).perform(click());

onView(withId(R.id.btnkonfirm)).perform(click());

onView(withId(R.id.buttonkonfirmasi)).perform(click());

onView(withId(R.id.buttonkonfirmasi)).perform(click());

Espresso.pressBack();

}

@Test

public void ensureButtonMenuWork() {

onView(withId(R.id.menu)).perform(click());

onView(withId(R.id.et)).perform(typeText("nasi"));

onView(withId(R.id.btn)).perform(click());

Espresso.closeSoftKeyboard();

onView(withId(R.id.et)).perform(typeText("cumi"));

onView(withId(R.id.btn)).perform(click());

Espresso.closeSoftKeyboard();

onView(withId(R.id.et)).perform(typeText(""));

onView(withId(R.id.btn)).perform(click());

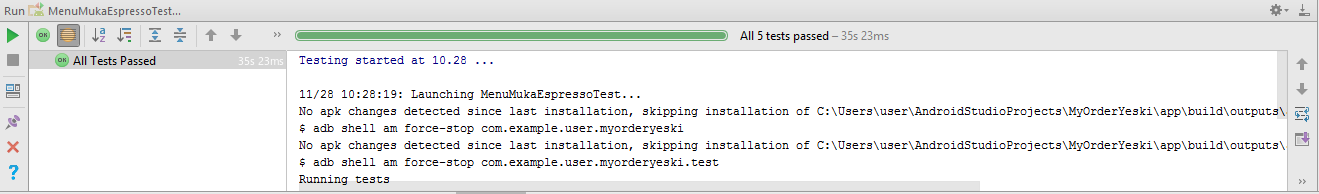
Espresso.closeSoftKeyboard();

Espresso.pressBack();

}

}

**Hasil Run :**



## REQUIREMENT YANG TELAH TERPENUHI

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | As | I Want to... | So That... | Definition of “Done” | Priority | Status | Estimation (day) |
| 1 | User | Mendapat interface yang mudah dipahami dan menarik | Mempermudah penggunaan aplikasi baik pengoperasian maupun pemesanan | Terdapat tampilan aplikasi yang menarik dan mudah dipahami (UI dan UX) | Medium | Done | 6 |
| 2 | User | Melihat tampilan makanan/minuman yang ditampilkan | Mengetahui detail makanan yang akan dipesan | Terdapat tampilan makanan dan minuman di seluruh daftar menu | Low | Done | 3 |
| 3 | User | Ingin bisa mengetahui harga makanan/minuman secara akurat | Mengetahui perkiraan harga yang akan dibayarkan | Terdapat harga pada setiap makanan dan minuman | Medium | Done | 3 |
| 4 | User | Bisa memilih makanan dan minuman yang ada | Melakukan pemesanan pada aplikasi secara langsung | Terdapat checkbox pada tampilan makanan dan minuman | High | Done | 5 |
| 5 | User | Bisa memilih multiple items(makanan/minuman) dalam sekali pemesanan | Melakukan pemesanan beberapa makanan sekaligus | Terdapat multi checkbox pada tampilan makanan dan minuman | Medium | Done | 5 |
| 6 | User | Mengisi form pemesanan | Melakukan pemesanan dengan cara pendataan melalui form yang tersedia | Terdapat formulir pengsisian | High | Done | 7 |
| 7 | User | Menghapus beberapa makanan/minuman dalam pemesanan sekaligus | Membatalkan makanan atau minuman yang masuk *cart* apabila tidak sengaja tertekan | Terdapat menu remove makanan/minuman pada daftar pesanan | Medium | Done | 5 |
| 8 | User | Menghapus pemesenan yang telah di pilih | Membatalkan pemesanan secara total | Terdapat menu remove all form | Low | Done | 3 |
| 9 | User | Bisa memesan makanan dan minuman | Melakukan order dan pemesanan | Terdapat tombol order pada formulir | High | Done | 5 |
| 10 | User | Melihat total pembayaran dari pemesanan | Mengetahui jumlah yang harus dibayarkan | Terdapat total jumlah harga pembayaran tiap pesanan | High | Done | 4 |
| 11 | User | Membatalkan pesanan jika belum menyelesaikan order | Melakukan pembatalan total apabila berubah pikiran | Terdapat tombol cancel order | Low | Done | 3 |
| 12 | Client | Bisa melihat pemesanan dari user | Mengetahui pemesanan dari user | Owner menerima pemesanan dari user | High | Done | 5 |
| 13 | Client | Membuat user mengetahui lokasi Yeski cafe | Lokasi Yeski Cafe diketahui user | Terdapat peta Yeski Cafe | Low | ToDo | 3 |
| 14 | Client | Melakukan promo | Menyebarluaskan promo | Terdapat menu promo | Medium | ToDo | 3 |