

Test Plan about OPEN CART

1. Gambaran Proyek

- Sebagai bahan pengerjaan maka pada proyek kali ini saya akan menguji beberapa fungsi dari aplikasi <https://demo.opencart.com/>.
- Tujuan penulisan dokumen *Test Plan* ini adalah sebagai pengujian dengan memperhatikan tahap – tahap berikut ini
 - Lingkup Sekitar Proyek
 - Strategi pengujian yang akan diterapkan
 - Jadwal pengujian
 - Kebutuhan akan *Resource*
 - *Test Deliverable and schedule*

2. Ruang Lingkup

Ruang lingkup yang menjadi perhatian dalam melakukan pengujian “<https://demo.opencart.com/>” mencakup fitur fitur berikut ini

Register	Home Page
Login & Logout	Checkout Page
Forgot Password	My Account Page
Search	Order History Page
Product Compare	Downloads Page
Product Display Page	Contact Us Page
Add to Cart	Menu Options
Wish List	Footer Options
Shopping Cart	Category Pages
Currencies	

3. Test Environmet

Aplikasi dijalankan di “Google Chrome”

4. Pengecualian Pengujian

- a) Semua fitur kecuali yang disebutkan pada poin 2
- b) Pengujian dengan *automation*

5. Test Strategy

Untuk melakukan pengujian saya akan menerapkan Langkah – Langkah berikut ini

- a) Langkah pertama (membuat scenario dari kasus uji untuk fitur yang telah dituliskan pada poin 2)

i. Teknik *design test* yang akan dilakukan saat membuat test case

- ***Equivalence Class Partition***

Metode dari **black box** yang membagi domain input dari program menjadi beberapa partisi

- ***Boundary Value Analysis***

Metode yang akan melengkapi equivalence partitioning dengan menentukan nilai tepat batas bawah dan batas atas.

- ***Decision Table Testing***

Metode yang digunakan untuk membangun satu set lengkap kasus uji tanpa menggunakan struktur internal dari program tersebut.

Dalam menciptakan kerangka *test case* saya menggunakan *table* untuk melaporkan nilai *input* dan *output* dari sebuah program.

- ***State Transition Testing***

Teknik **black box** yang digunakan pada software yang mendefinisikan state (keadaan) dan transition (perubahan) berdasarkan aturan/ requirement sistem pada setiap form yang ada pada sistem.

- ***Use Case Testing***

Pengujian dari **black box** dengan menguji use case yang telah dibuat

b) Langkah kedua (proses pengujian yang akan saya lakukan)

i. Saya akan melakukan ***smoke testing*** untuk memeriksa apakah fungsi-

fungsi utama/kritis dari sistem ini bekerja sesuai dengan yang diharapkan.

ii. Saya akan menolak build jika smoke testing gagal dan akan menunggu

sampai stabil sebelum saya melakukan pengujian mendalam dari fungsionalitas aplikasi ini.

iii. Setelah menerima build yang sudah stabil (lulus smoke testing), saya akan melakukan ujian secara mendalam menggunakan ***test case*** yang telah saya buat

iv. Saya set up environment yang mendukung

v. Saya akan membuat laporan bug pada alat pelacak bug

- vi. Sebagai bagian dari pengujian maka saya akan melakukan jenis pengujian *o Smoke Testing and Sanity Testing*
sanity testing dilakukan untuk menentukan apakah versi terbaru dari software tersebut dapat bekerja dengan baik atau tidak, jika tidak maka tidak akan dilakukan pengujian lebih lanjut

o Regression Testing and Retesting

Regression testing dilakukan untuk memastikan tidak ada perubahan yang dilakukan selama proses pembangunan yang menyebabkan bug baru

o Usability Testing, Functionality & UI Testing.

- c) Langkah ketiga (menerapkan praktik berikut untuk menjadikan pengujian yang lebih baik)
 - i. **Context Driving Testing**, Saya akan melakukan pengujian sesuai dengan konteksnya dari aplikasi yang diberikan
 - ii. **Shift Left Testing**, Saya akan memulai pengujian tersebut dari tahap awal pengembangan itu sendiri alih – alih menunggu stable build.
 - iii. **Exploratory Testing**, Saya akan melakukan eksploratory pengujian, selain dari mengeksekusi kasus uji normal
 - iv. **End to end flow testing**, Saya akan menguji scenario end to end yang melibatkan beberapa fungsi untuk mensimulasikan *end users flow*.

6. Prosedur Bug Report

Selama proses **test execution** yang akan saya lakukan adalah

- i. Setiap penyimpangan dari perilaku yang diharapkan oleh aplikasi akan dicatat, jika hal tersebut tidak dapat dilaporkan sebagai cacat, maka akan dilaporkan sebagai pengamatan/masalah atau diajukan sebagai pertanyaan
- ii. Setiap masalah kegunaan juga akan dicatat
- iii. Setelah ditemukan cacat, selanjutnya akan diuji ulang untuk memverifikasi reproduktifitas dari cacat. Tangkapan layar dengan langkah-langkah mereproduksi akan didokumentasikan.

- iv. Diakhir pelaksanaan tes, cacat yang ditemui akan dicatat bersama dengan observasi.

Note: ** ➔ Cacat akan didokumentasikan di excel

➔ Skenario uji dan kasus uji akan didokumentasikan dalam dokumen excel

7. Jadwal test

Jadwal tes yang telah saya rencanakan adalah sebagai berikut ini

Task	Time Duration
Creating Test Plan	Start Date to End Date
Test case creation	Start Date to End Date
Test Case Execution	Start Date to End Date
Summary Reports Submission	Date

8. Hasil Pengujian

Saya akan mencatat hasil dari pengujian sebagai berikut!

Pengujian	Deskripsi	Tanggal penyelesaian target
Test Plan	Detail tentang ruang lingkup proyek, strategi pengujian, jadwal pengujian, persyaratan sumber daya, hasil pengujian dan jadwal	Date
Functional test cases	Test case dibuat untuk ruang lingkup yang telah ditentukan	Date
Defect reports	Deskripsi terperinci tentang cacat yang diidentifikasi dan cara acara mengidentifikasinya	Date
Summary reports	Summary reports	Date

	<ul style="list-style-type: none">- Bugs by bug,- Bugs by functional area and- Bugs by priority	
--	---	--