Blackbox Testing On E-Commerce System Web-Based Evermos (Feature: Registration Experiment & Revamp)

Ucu Nugraha [1], Tiodor Sianturi[2]

[1]Engineering Faculty, Widyatama University, Indonesia
[2]Engineering Faculty, Widyatama University, Indonesia
ucu.nugraha@widyatama.ac.id [1], tiodor.sianturi@widyatama.ac.id [2]

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstrct: Software Testing/System Testing is a critical element of assurance software quality that represents the main study of specifications, design, and coding. Increased visibility (capabilities) of the software as a system element and the "costs" that arise due to device failure lenient, motivates good planning through its testing thorough.Black-box testing techniques focus on the information domain of the software, by doing a test case by partitioning the input domain of a program in a way that provides in-depth testing coverage. The Black Box itself has several methods in testing. Namely, the test method Graph-based explores the relationship between objects and behavior program. The equivalent partition divides the input domain into data classes it is possible to perform certain software functions. Boundary value analysis check the program's ability to handle data to the extent it can received.Evermos is a platform, a platform, to sell products Indonesian Muslims. As “Everyday Need for Every Moslem”, Evermos was started from a dream, vision and goal to helping small businesses and people individuals to compete with large and existing companies advanced. Testing software using the Black Box method is expected to increase visibility and meet quality requirements.Evermos software itself. The results of testing the software is aimed at finding errors in the category of functions incorrect, interface errors, errors in data structures or access external database, performance errors, initialization and termination errors.Evermos Web software test results is to provide documentation of test results which informs the suitability of the software being tested with predetermined specifications and finds errors on Evermos.
Keywords: software testing, system testing, blackbox, ecommerce

1. Introduction

The registration flow contained in the evermos application at this time, after conducted several research by a team of researchers and sales team several inputs, including:

a. The registration flow for the Evermos app was confusing, the team saw a decline everytime;
b. Error/Bugs often appear in the registration flow for the Evermos application;
c. The Evermos application does not have a feature to collect additional data other than registration data.

The registration process is very important and constitutes

The main feature of the Evermos application, so that the input it has received and analyzed by the team concerned needs to be followed up.

To overcome the points above in the flow and registration UI on the Evermos system.

In this case, only testing the Register feature, and does not do the development process. Testing performed only on web based platforms.

The objectives and benefits to be achieved are checking the requirement feature register.

Testing results can minimize bugs that will be experienced by the user so that major or critical bugs do not occur when it comes to features the new registration is already in the production or live environment.

2. Literature review

Software testing is done to find out whether at a the program or system is in accordance with the expected results. Testing is an integral part of a software. With Over time nowadays many systems or programs were built with the purpose of facilitating activities that run in an agency or organization, so that there needs to be an increase, namely by conducting testing on a software so that the application or system can run properly or the features on the system can be used properly. Importance software testing and its implications refer to the quality of the device soft. Software is a critical element of software quality assurance and represents the main study of specification, design, and coding. Increased visibility of software as a system element and costs arising from software failure, motivates him to do so good planning through careful testing.

Design testing software and other engineering products can be alike challenging with the initial design of the product itself. Based on the objectivity of testing, the importance of doing a test design for the purpose of finding

out frequent errors, with minimum effort and time (B, 2006). Currently, a wide variety of test case design methods have been developed, that is used in software testing. These methods provide to software developers a systematic approach to do the test. And more importantly, these methods provide mechanisms that can help to ensure completeness of testing and provides the highest probability of getting device errors soft (Jatnika & Irwan, 2010; Mogano & Mokoele, 2019).

There are several testing approaches carried out in software, including:

a. Based on the specified function of the product, testing done to show that each function is already fully operational, at the same time finding fault with each

b. function (B, 2006). Based on an internal performance of the product, testing done by making sure all components in the program can run well as it should be (Jatnika & Irwan, 2010).

The first method of testing is called black box testing and the second approach is called white box testing.

In general, it is known that in a software development cycle there are always four main processes, namely:

1. Plan (prepare a plan)

Defines objectives and determines supportive strategies and methods achievement of objectives.

2. Do (carry out the plan)

Creating the conditions and performance necessary to carry out the plan.

3. Check (check the results)

Checks are carried out to determine whether the work is progressing according to plan, and whether the expected results have been realized.

4. Action (Take action that is important)

If it is found that the work is not in accordance with the plan and the results that have been determined, then a measurement is made of what action will be taken (Perry, 1995).

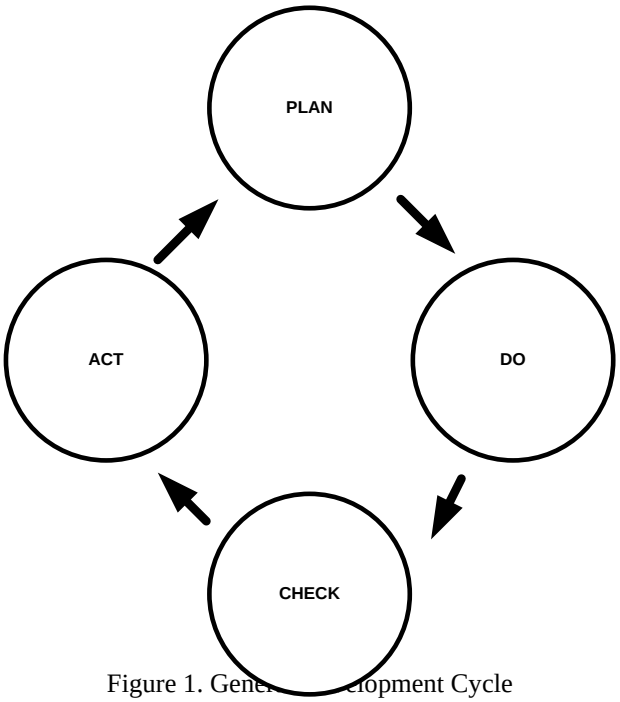

Figure 1. Gene[illegible]lopment Cycle

Software testing is a process to find errors on each software item, record the results, evaluate every aspect on each system component and evaluate all the facilities of the software being developed. There are 2 main things that are done in testing, namely:

1. Verification is the process of evaluating a system / component for determine whether a product is finished after the phase development fulfills the conditions as set out in the beginning development (when specifying) the software. (“Are we building the product right? ”).

2. Validation is the process of evaluating a system or component on end or during the development period to determine whether the product produced has met the needs and requirements specific requests requested by the user. ("Are we building the right product?").

(Perry, 1995).

Black Box testing is a complementary approach to the White Box technique, because black box testing is expected to be able to reveal the class of errors that are wider than the White Box technique. Black Box testing focuses on testing the functional requirements of the software, to get a series input conditions that correspond to the functional requirements of a program (Smirnov, 2002 & Laurie, 2006).

Black Box testing is testing the fundamental aspects of a system without pay attention to the internal logic structure of the software. This method is used to find out if the software is functioning properly. Testing Black Box is a test data design method based on software specifications. The test data is generated, executed in the software and then the output of the software is checked whether it matches that expected. Black Box testing attempts to find errors in categories:

1. Functions that are incorrect or missing.
2. Interface errors.
3. Errors in data structures or external database access.
4. Performance errors.
5. Initialization and termination errors.

(Perry, 1995).

3. RESULTS AND DISCUSSION

Test Case

Here are some test cases for the modules:

a. Test Case Modul [Web] Registration Form

Table 1. Test Case Modul [Web] Registrasi Form

TC-ID	DESKRIPSI TEST CASE
EVM520-TC001	verify copywriting and design
EVM520-	when config firebase is ON **evm_registration_experiment_web**
EVM520-TC003	when config firebase is OFF **evm_registration_experiment_web**
EVM520-	verify config link for
EVM520-	verify the password
EVM520-	verify the phone number
EVM520-	verify when phone number
EVM520-	verify when phone number
EVM520-	verify when phone number already registered in EVM and has
EVM520-TC010	when filling all mandatory fields
EVM520-TC011	when user checked the conditions and checkbox conditions
EVM520-	when user click masuk
EVM520-	verify when click the icon
EVM520-	when using promo free
EVM520-	when using promo half free
EVM520-	when using referral

TC-ID	DESKRIPSI TEST CASE
EVM520-TC017	when registration via referral link https://evermos.com/?linkType=referral&timeout=5
EVM520-TC018	when registration using promo link https://evermos.com/verify?tv
EVM520-	verify the tracker

b. Test Case Modul OTP Verification

Table 2. Test Case Modul OTP Verification

TC-ID	DESKRIPSI TEST CASE
EVM521-	verify copywriting and design
EVM521-	when sending OTP via
EVM521-	when sending OTP via SMS
EVM521-	when user click verifikasi by
EVM521-	verify configuration
EVM521-	when input the invalid OTP
EVM521-	when the timer is end
EVM521-	when resend the OTP
EVM521-	when already send OTP 3 times
EVM521-	verify the tracker

c. Test Case Modul Welcome Message

Table 3. Test Case Modul Welcome Message

TC-ID	DESKRIPSI TEST CASE
EVM522-	verify design and
EVM522-	when click lihat video
EVM522-	when click Yes, understand
EVM522-TC004	verify the video configuration in mimin **evm_welcome_youtube_li**
EVM522-	verify the tracker

d. Test Case Modul Payment Premium

Table 4. Test Case Modul Payment Premium

TC-ID	DESKRIPSI TEST CASE
EVM523-	verify design and
EVM523-TC002	when selecting the payment method
EVM523-	when click pay now
EVM523-	when click exit account

EVM523-TC005	when click check payment
EVM523-TC006	verify the tracker

e. Test Case Modul Notification Dialog

Table 5. Test Case Modul Notification Dialog

TC-ID	DESKRIPSI TEST CASE
EVM524-TC001	verify the design and copywriting
EVM524-TC002	verify the tracker
EVM524-TC003	verify when close the notif the go to the katalog page again
EVM524-TC004	verify the queue notification dialogue
EVM524-TC005	when user click action
EVM524-TC006	verify the configuration
EVM524-TC007	verify each notification dialogue type

f. Test Case Modul Not-paid-yet User LLogin

Table 6. Test Case Modul Not-paid-yet user login

TC-ID	DESKRIPSI TEST CASE
EVM527-TC001	when the user has been selected the payment method
EVM527-TC002	when the user has not yet selected the payment method
EVM527-TC003	when user already paid but the status still pending
EVM527-TC004	verify the design and copywriting
EVM527-TC005	verify the tracker

g. Test Case Modul Save url of Registration

Table 7. Test Case Modul Save url of registration

TC-ID	DESKRIPSI TEST CASE
EVM529-TC001	verify the URL was saved after finish registration
EVM529-TC002	verify the data ads for testing

h. Test Case Modul Target Market Survey

Table 8. Test Case Modul Target market survey

TC-ID	DESKRIPSI TEST CASE
EVM530-TC001	When user see the target market notification
EVM530-TC002	When user open target market survey in web
EVM530-TC003	When user open target market survey in android
EVM530-TC004	When fill all mandatory question fields and click save
EVM530-TC005	When user didn't fill one mandatory question fields and click save
EVM530-TC006	When click back on the device function after not finish answer the target market question
EVM530-TC007	When click back on the device function after finish answer the target market question
EVM530-TC008	When click back on the icon after not finish answer the target market question
EVM530-TC009	When click back on the icon after finish answer the target market question
EVM530-TC010	Verify the rule for each question
EVM530-	Verify design and
EVM530-	Verify tracker

i. Test Case Modul Starter Kit Address

Table 9. Test Case Modul Starter Kit Address

TC-ID	DESKRIPSI TEST CASE
EVM531-TC001	verify the design and copywriting
EVM531-	verify the tracker
EVM531-TC003	when input all mandatory fields
EVM531-	when click back
EVM531-TC005	when checked the main address checkbox
EVM531-TC006	when unchecked the main address checkbox
EVM531-TC007	when already have primary address then input the form
EVM531-	verify the phone number

Test Result
Here are the test results:

a. Testing Module [Web] Registration Form

Tabel 10. Testing Results Module [Web] Registration Form

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM520-TC001	verify copywriting and design	*PASSED*
EVM520-TC002	when config firebase is ON **evm_registration_experiment_web**	*PASSED*
EVM520-TC003	when config firebase is OFF **evm_registration_experiment_web**	*PASSED*
EVM520-TC004	verify config link for **evm_product_before_register**	*PASSED*
EVM520-TC005	verify the password validation	*PASSED*
EVM520-TC006	verify the phone number validation	*PASSED*
EVM520-TC007	verify when phone number already registered in EVM	*PASSED*
EVM520-TC008	verify when phone number already registered in co-branding	*PASSED*
EVM520-TC009	verify when phone number already registered in EVM and has status pending payment → updated db	*PASSED*
EVM520-TC010	when filling all mandatory fields	PASSED
EVM520-TC010	when filling all mandatory fields	*PASSED*
EVM520-TC011	when user checked the terms and conditions checkbox	*PASSED*
EVM520-TC012	when user click enter	*PASSED*

EVM520-TC013	verify when click the icon information in promo	*PASSED*
EVM520-TC014	when using promo free	*PASSED*
EVM520-TC015	when using promo half free	*PASSED*
EVM520-TC016	when using referral	*PASSED*
EVM520-TC017	when registration via referral link https ://evermos.com/? link Type= referral& timeout =5000&referralId= toko.croco.40f	*PASSED*

b. Welcome Message Module Testing

Tabel 11. Welcome Message Module Testing Results

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM522-TC001	verify design and copywriting	*PASSED*
EVM522-TC002	when click lihat video	*PASSED*
EVM522-TC003	when click Yes, understand	*PASSED*
EVM522-TC004	verify the video configuration in mimin **evm_welcome_youtube_link**	*PASSED*
EVM522-TC005	verify the tracker	*PASSED*

c. Premium Payment Module Testing

Tabel 12. Premium Payment Module Testing Results

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM523-TC001	verify design and copywriting	*PASSED*
EVM523-TC002	when selecting the payment method	*PASSED*
EVM523-TC003	when click pay now	*PASSED*

EVM523-TC004	when click exit account	*PASSED*
EVM523-TC005	when click check payment	*PASSED*
EVM523-TC006	verify the tracker	*PASSED*

d. Testing the Notification Dialog Module

Tabel 13. Test Results of the Notification Dialog Module

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM524-TC001	verify the design and copywriting	*PASSED*
EVM524-TC002	verify the tracker	*PASSED*
EVM524-TC003	verify when close the notif the go to the katalog page again	*PASSED*
EVM524-TC004	verify the queue notification dialogue	*PASSED*
EVM524-TC005	when user click action	*PASSED*
EVM524-TC006	verify the configuration	*PASSED*
EVM524-TC007	verify each notification dialogue type	*PASSED*

e. Testing the Not-Paid-Yet User Login Module

Tabel 14. Test Results for the Not-Paid-Yet User Login Module

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM527-TC001	when the user has been selected the payment method	*PASSED*
EVM527-TC002	when the user has not yet selected the payment method	*PASSED*
EVM527-TC003	when user already paid but the status still pending	*PASSED*
EVM527-TC004	verify the design and copywriting	*PASSED*
EVM527-TC005	verify the tracker	*PASSED*

f. Testing the Save url of Registration Module

Tabel 15. Test Results of the Save url of Registration Module

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM529-TC001	verify the URL was saved after finish registration	*PASSED*
EVM529-TC002	verify the data ads for testing	*PASSED*

g. Testing the Target Market Survey Module

Tabel 16. Test Results of the Target Market Survey Module

TC-ID	DESKRIPSI TEST CASE	STATUS
EVM530-TC001	When user see the target market notification	*PASSED*
EVM530-TC002	When user open target market survey in web	*PASSED*
EVM530-TC003	When user open target market survey in android	*PASSED*
EVM530-TC004	When fill all mandatory question fields and click save	*PASSED*
EVM530-TC005	When user didn't fill one mandatory question fields and click save	*PASSED*
EVM530-TC006	When click back on the device function after not finish answer the target market question	*PASSED*
EVM530-TC007	When click back on the device function after finish answer the target market question	*PASSED*
EVM530-TC008	When click back on the icon after not finish answer the target market question	*PASSED*

EVM530-TC009	When click back on the icon after finish answer the target market question	*PASSED*
EVM530-TC010	Verify the rule for each question	*PASSED*
EVM530-TC011	Verify design and copywriting	*PASSED*
EVM530-TC012	Verify tracker	*PASSED*

h. Testing Module Starter Kit Address

Tabel 16. Test Results of the Starter Kit Address Module

TC-ID	DESKRIPSI TEST	STATUS
EVM531-TC001	verify the design and copywriting	*PASSED*
EVM531-TC002	verify the tracker	*PASSED*
EVM531-TC003	when input all mandatory fields	*PASSED*
EVM531-TC004	when click back	*PASSED*
EVM531-TC005	when checked the alamat utama checkbox	*PASSED*
EVM531-TC006	when unchecked the alamat utama checkbox	*PASSED*
EVM531-TC007	when already have alamat utama then input the form	*PASSED*
EVM531-TC008	verify the phone number rules	*PASSED*

4. Conclusions

Evermos is a platform, a platform, to sell products Indonesian Muslims. As “Everyday Need for Every Moslem”, Evermos was started from a dream, vision and goal to helping small businesses and people individuals to compete with large and existing companies advanced. Testing software using the Black Box method is expected to increase visibility and meet quality requirements.

Evermos software itself. The results of testing the software is aimed at finding errors in the category of functions incorrect, interface errors, errors in data structures or access external database, performance errors, initialization and termination errors.

Evermos Web software test results is to provide documentation of test results which informs the suitability of the software being tested with predetermined specifications and finds errors on Evermos.

References

1. Behm, Barry 1990, Software Risk Management, New York: IEEE Computer Society. 1, 12 – 43.
2. Kadir, Abdul, 2003, Pemrograman eb : Mencakup HTML, CSS, Javascript & PHP. Yogyakarta :Andi Offset.

3. Sergey, Smirnov, 2002, *Software Testing: Black-Box Techniques*, 1 – 4.
4. Summerville, Ian. 2003, Software Enginering :Rekayasa Perangkat Lunak. Edisi 6 Jilid 1. Jakarta : Erlangga.
5. Prakash, G., Darbandi, M., Gafar, N., Jabarullah, N. H., & Jalali, M. R. (2019). A New Design of 2-Bit Universal Shift Register Using Rotated Majority Gate Based on Quantum-Dot Cellular Automata Technology. *International Journal of Theoretical Physics*, 58(9), 3006-3024.
6. William, Laurie. 2006, *Testing Overview and Black-Box Testing Techniques,35-59.*
7. William, Perry, 1995, *Effective Methods for Software Testing*, 1-5**,** 3-430.
8. Mogano, P., & Mokoele, N. (2019). SOUTH AFRICAN CLIMATE CHANGE ADAPTATION POLITICS: URBAN GOVERNANCE PROSPECTS. The International Journal of Social Sciences and Humanity Studies, 11(1), 68-83.