Developing Software Center Using Evolutionary Prototyping Based-on HTML5

Imam Riadi
Information Systems Study Program, Faculty of Mathematics and Natural Sciences,
Ahmad Dahlan University, Yogyakarta, Indonesia

Estu Fardani
Informatics Department, Science and Technology Faculty,
Sunan Kalijaga State Islamic University,
Yogyakarta, Indonesia

***Abstract*—The abstract goes here.**

I. Introduction

Software Center has a function to management software, which adds, removes and update software. Its existence is necessary in all operating systems both desktop and mobile devices. This application use on OpenSUSE, Ubuntu, Mac OS X, Android, iOS, BlackBerry and Windows. However, these applications are not in BlankOn Linux as Indonesian local Linux distribution. Installation of applications in BlankOn Linux currently can be done in the traditional way: first from the console (terminal), this way can be done if you know the name of the application you want to install the package. The second way to use the Synaptic Package Manager a management application packages. Despite using a GUI (graphic user interface) but still only be done if you know the name of the application you want to install the package and limited its use. Making it difficult for an ordinary user to install the application.

Similar applications that have been there is the Ubuntu Software Center for Ubuntu distribution. This application allowing licensed OpenSource applied to BlankOn Linux. However, since these applications have become a trademark of Ubuntu, needs further modification process. This process takes time and is not guaranteed to run well so, what ideas arise if made to start from scratch.

HTML5 is a trend of programming because this technology easy to adapted, can run in many platforms up to desktop utilization [2]. Manokwari Desktop a desktop environment for BlankOn Linux is base on HTML5 and GNOME 3. Development software center base on HTML5 can run consistent with development of BlankOn Linux [3]. This research tries to make software center HMTL5-based as solution for BlankOn Linux.

II. Similar Software Center

YaST (Yet another Setup Tool) Software center for openSUSE. It use to manage all of user need for use openSUSE. Example to install application, package, network setting etc. YaST use ruby and YCP for language programming and Qt for user interface[3].

Ubuntu Software Center Software management for Ubuntu by Canonical. It is free software written in Python, PyGTK/PyGObject based on GTK+ and the further development of the GNOME application, gnome-app-install.[4][5]

III. HTML5 App in Desktop

A. Webkit and WebkitGTK+

WebKit is a layout engine software component for rendering web pages in web browsers. It powers Apple's Safari web browser and was previously used in Google Chrome web browser. Figure 1 and 2 shows how Webkit work.

WebKit is also used as the basis for the experimental browser included with the Amazon Kindle e-book reader, as well as the default browser in the Apple iOS, Android, BlackBerry 10, and Tizen mobile operating systems. WebKit's C++ application programming interface provides a set of classes to display web content in windows, and implements browser features such as following links when clicked by the user, managing a back-forward list, and managing a history of pages recently visited.[6] bart

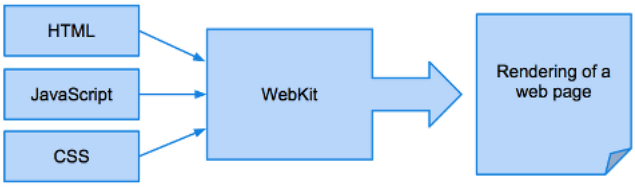

Fig. 1: How Webkit Work

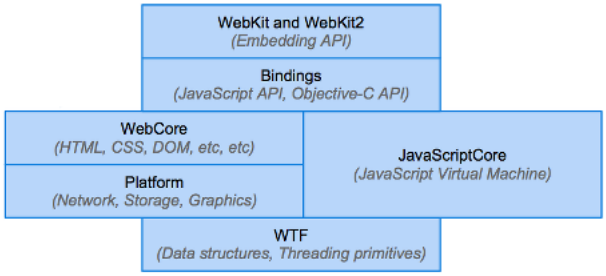

Fig. 2: Webkit Major Components

WebKitGTK+ is the GNOME platform port of the WebKit rendering engine. Offering WebKits full functionality through a set of GObject-based APIs, it is suitable for projects requiring any kind of web integration, from hybrid HTML/CSS applications to full-fledged web browsers, like Epiphany and Midori. Its useful in a wide range of systems from desktop computers to embedded systems like phones, tablets, and

televisions. WebKitGTK+ is made by a lively community of developers and designers, who hope to bring the web platform to everyone.[7] webkitgtk.org

B. WarSi

WarSi (Warung aplikasi) is the name of software center run in BlankOn Linux. It job to help user management applications by make Graphic User Interface. To do management WarSi need link to apt. its use vala programming. Object oriented programming base on c and GObject.

Valac, the Vala compiler, is a self-hosting compiler that translates Vala source code into C source and header files. It uses the GObject type system to create classes and interfaces declared in the Vala source code. The syntax of Vala is similar to C#, modified to better fit the GObject type system. Vala supports modern language features as the following:

- Interfaces
- Properties
- Signals
- Foreach
- Lambda expressions
- Type inference for local variables

Vala is designed to allow access to existing C libraries, especially GObject-based libraries, without the need for runtime bindings. All that is needed to use a library with Vala is an API file, containing the class and method declarations in Vala syntax. Vala currently comes with experimental bindings for GLib and GTK+. It's planned to provide generated bindings for the full GNOME Platform at a later stage.

IV. Development WarSi

A. Design System

Figure 3 show how design of WarSi.

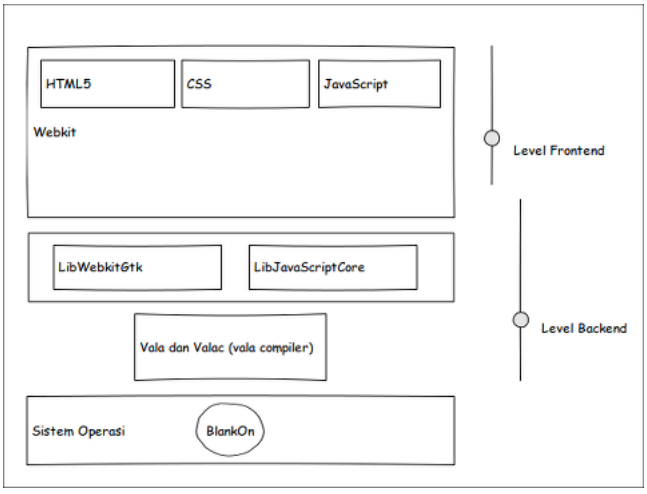

Fig. 3: Design System

B. Development Method

System development methods used in this study is the method of Evolutionary Prototyping. Evolutionary prototyping method has the steps as shown in Figure 4.

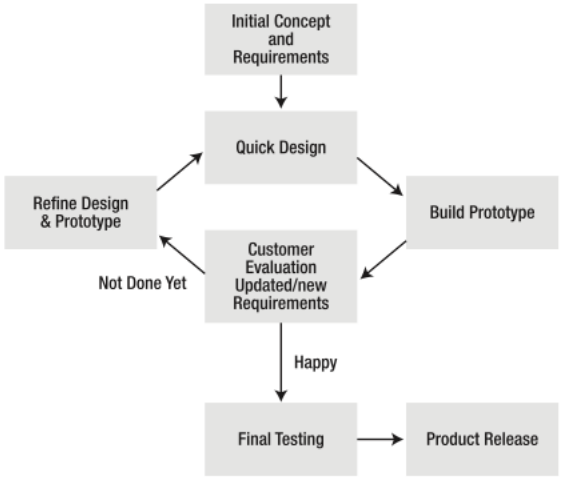

Fig. 4: Evolutionary Prototyping Process Model

Evolutionary prototyping provides other benefits including:

- the clarification of management and user requirements;
- the ability to uncover missing or previously unknown requirements;
- the flexibility to meet changing constraints for software systems;
- the provision of a method whereby users, management, and developers can communicate about systems;
- the easing of maintenance tasks;
- the creation of better user interfaces;
- prototyping with quality; and
- the ability for developers to reflect on lessons learned during system development.

Steps of Evolutionary Prototyping:

1) Initial Concept and Requirement:

2) Quick Design:

3) Build Prototype:

4) Costumer Evaluation, Update:

5) If not done yet, Refine design and prototype:

6) If happy, final testing and product release:

C. UML (Unified Modelling Language

1) Use Case Diagram: Figure 5 show the use case diagram for software center.

This is a description of the process in the use case diagram:

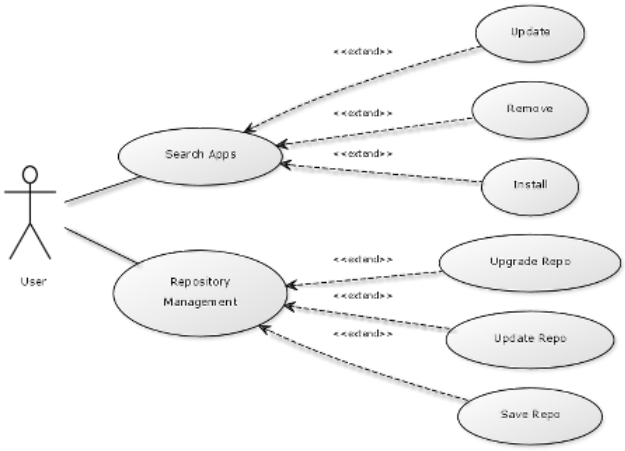

Fig. 5: Use case Diagram

- Search Application
  This process is performed to find the application as the user desires. Once the search is complete the user can install / remove applications and application upgrades.
- Repository Management
  This funtions make user can change source repository, save, update and upgrade system.

2) Activity Diagram: Figure 6 show activity diagram generally.

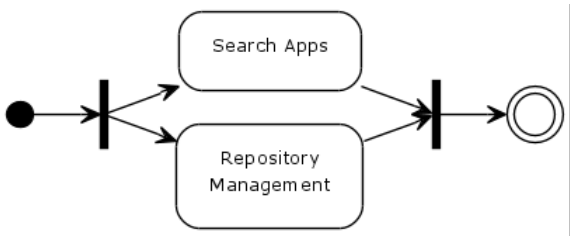

Fig. 6: Activity Diagram

Activity Diagram Search Applications

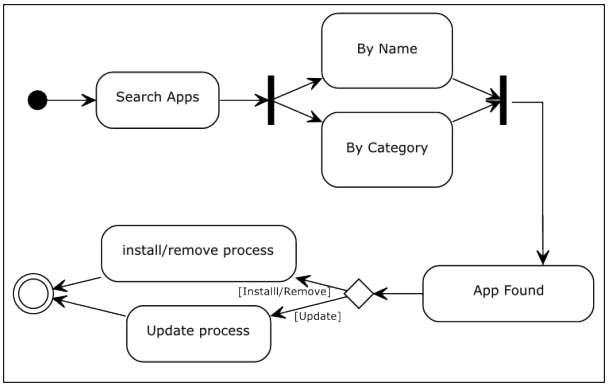

Fig. 7: Activity Diagram Search Application

Figure 7 is an activity diagram for a search application. Search Application is divided into two, namely by entering a keyword in the search field and the search for applications by category. Process for Figure 6 are:

- Users select search for applications by name or by category of applications,
- If you select by name, users enter a keyword or the name of the application you are looking for in the search field to find the application,
- If you select a category based application, the user selects a category that existed until the application is found,
- Application is found, the user chooses to install/remove or update the application,
- Installation process, and the process is complete.

Activity Diagram Repository Management

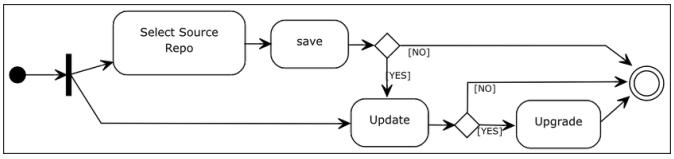

Fig. 8: Activity Diagram Repository Management

Process for Figure 8 are:

- User select a menu from the main menu next repository, system will display the page repository, there are two options, choose source repository or directly perform the update,
- User can make changes to the repository by selecting the list repository that is already available,
- When the user makes changes to the repository, the system will saved the changes,
- When the user performs an update to the update repository the repository will be made and the system will display the process update,
- After update user can upgrade system if available, then the system will display the process upgrade.
- Process is completed.

3) Sequence Diagram: (gambar ulang)

D. Result Development

diisini apa ini?

E. Testing

Black box testing (also called functional testing) is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.[11]

With black box testing, the software tester does not (or should not) have access to the source code itself. The code is considered to be a big black box to the tester who cannot

see inside the box. The tester knows only that information can be input into to the black box, and the black box will send something back out.

What the test : Functional include: (list of testing)

after testing to 15 people the result is (diagram)

V. Conclusion

Based on the research that have been carried out during development Software Center using Evolutionary Prototyping HTML5-based, it can be concluded as follows:

- Analysis and development Software Center that can help users to manage applications in BlankOn Linux has been successfully carried out,
- In this study the HTML5 programming language can be used for development desktop application,
- Based on the test results showed that the Software Center can be run on BlankOn Linux.

.

Acknowledgment

Thank you to Ahmad Dahlan University for funding this research

References

[1] Duncan Mac-Vicar P., *What you should know about YaST*, Novell, Inc, 2008.

[2] Mathew Paul Thomas, *Ubuntu Software Center*, http://wiki.ubuntu.com/SoftwareCenter.

[3] Lounpad Ubuntu

[4] Wiki BlankOn,

[5] John Dooley, *Software Development and Professional Practice*, Springer Science Business Media, Inc, 2011.

[6] Carter, Ryan A.,Anton, Annie I.,Dagnino, Aldo., Williams, Laurie, *Evolving Beyond Requirements Creep:A Risk-Based Evolutionary Prototyping Model*, North Carolina State University, 2001.

[7] IEEE, *IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*, 1990.