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 Software Center is a required application in all operating system, both conventional or mobile devices. BlankOn as the local operating system should have this application but in reality this application is not yet available. HTML5 is a trend of programming because this technology easy to adapted for desktop utilization. This research tries to make software center HMTL5-based as solution for BlankOn. System development methods used in this study is evolutionary prototyping. It possible to develop application base on prototype of software. Next step is implement prototype to language programming. Every progress will be correction to get better features. Progress will be finished if prototype had all features what needs. Stages of research is divided into several steps. first analysis similar applications, second design system, third implementation and testing of the system. Based on the results of functionality testing software center, the percentage obtained 100% agree the system can run well. From the results usability testing software center, it is concluded that the respondents strongly agreed that as much as 80% and 20% agree. It can be concluded that the software center application can run well and smoothly.**

Keywords : Vala; Webkit; HTML5; Software Center; BlankOn

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. 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 [1]. 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[2].

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.[3][4]

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 [5].

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 [6].

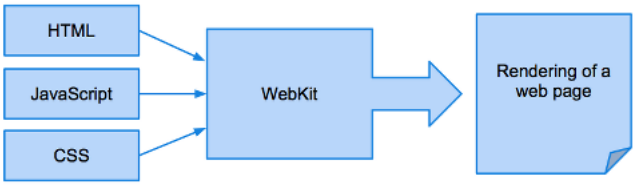

Fig. 1: How Webkit Work

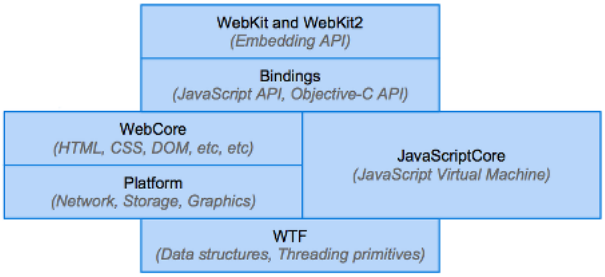

Fig. 2: Webkit Major Components

B. WarSi (Software Center)

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 [7].

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

WarSi design base on HTML5, AngularJS for front-end. Back-end use Vala programming. Vala chosen because it can communicate with operating system like run specific apt command use Vala API. Another reason Vala can run simultanly with Manokwari Desktop base on GNOME 3.

Figure 3 show how design of WarSi.

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 [8].

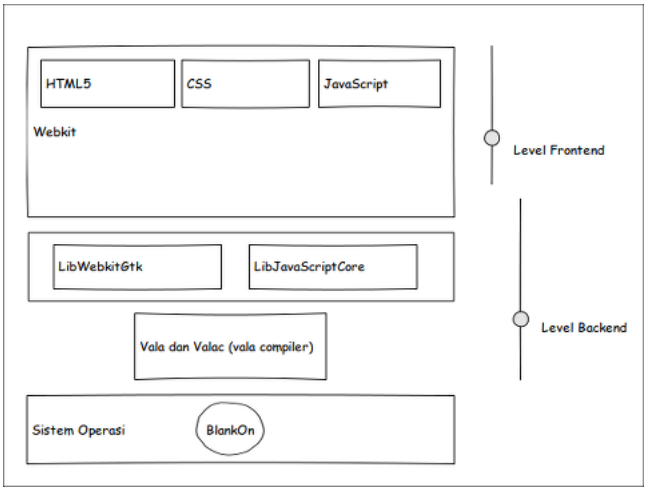

Fig. 3: Design System

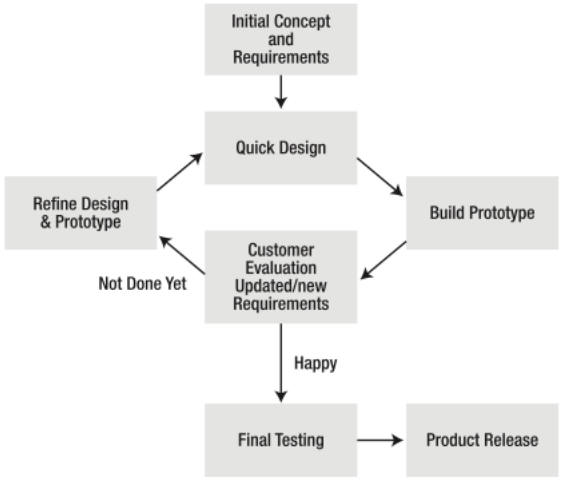

Fig. 4: Evolutionary Prototyping Process Model

Evolutionary prototyping provides other benefits including[9]: 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: At this stage will be determined what will be in software center includes system requirements, criteria, the function to be achieved. This stage is important that detailed description of the application can be explained from the beginning.

2) Quick Design: The stage design is done in standard applications with minimal functionality to answer the important points that have been described in the previous step. Design prototype mock-up done in the form of the initial design and

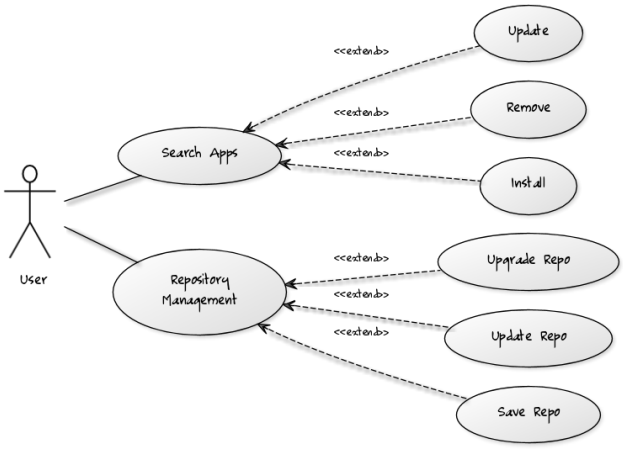

Fig. 5: Use case Diagram

the functions that can run applications.

3) Build Prototype: From the results of the initial prototype design to be started in the form of program code. Starting from the design of the interface using HTML5, CSS3, javascript. The design is achieved not have to be exactly the same as designed. The next thing is to combine the Vala programming, Webkit as the back-end Software Center. Installation of container is done at this stage that some of the functions of the application can run. After this initial prototype Software Center obtained. This process also includes whether the application can be run in BlankOn Linux.

4) Costumer Evaluation, Update: Prototype has been done early discussion back to the client for review and evaluate, both design, functional, and usability. Any corrections and recommendation will be collected.

5) If not done yet, Refine design and prototype: If client not satisfied, prototype must be correction base on recommendation which have been collect in previously.

6) If happy, final testing and product release: Once the prototype is approved client, the application entered the stage of release and make documentation of software center.

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:

- 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.

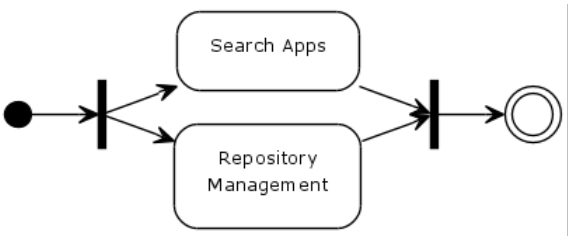

Fig. 6: Activity Diagram

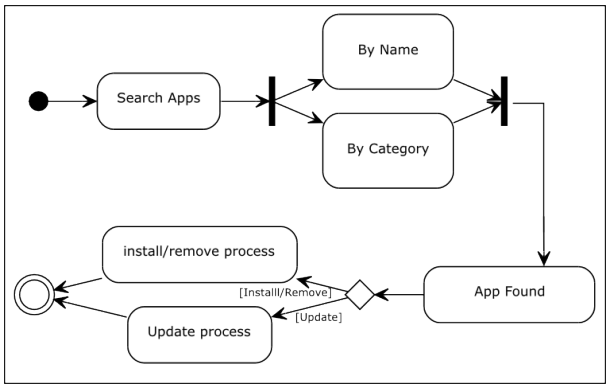

Fig. 7: Activity Diagram Search Application

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

Activity Diagram Search Applications. 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 7 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 chose to install/remove or update the application,
- Installation process, and the process is complete.

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,

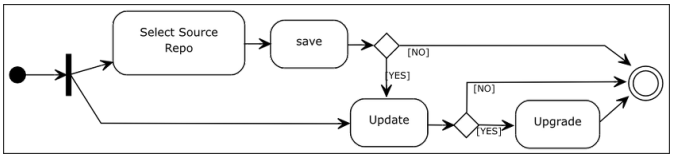

Fig. 8: Activity Diagram Repository Management

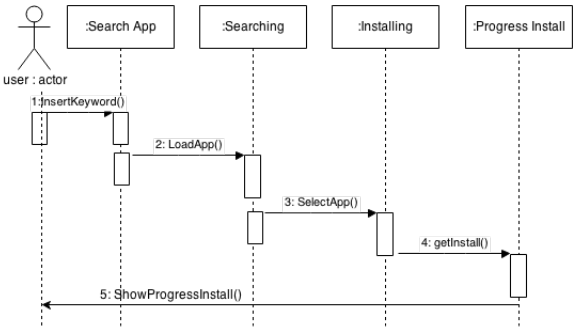

Fig. 9: Sequence Diagram Search Application by Name

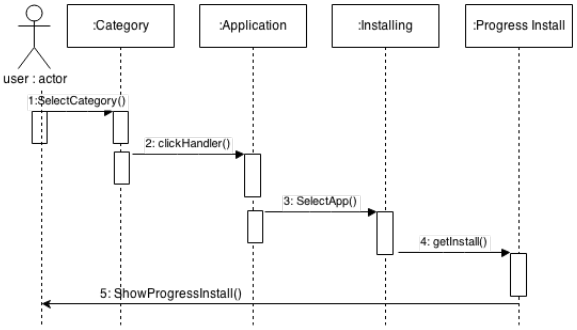

Fig. 10: Sequence Diagram Search Application by Category

- 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: Figure 9 show sequence diagram for search application by name.

D. Result Development

After development finished, the result is prototype software center for BlankOn Linux. Base on HTML5 for front-end. Screenshoot WarSi can be found in Figure 12 and Figure 13.

Figure 14 show page to search application.

Figure 15 show page repository management.

E. Testing

Black box testing (also called functional testing) is testing that ignores the internal mechanism of a system or component

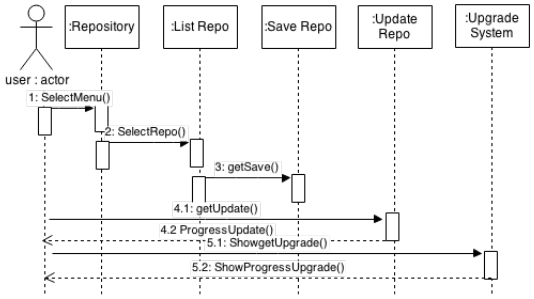

Fig. 11: Sequence Diagram Management Repository

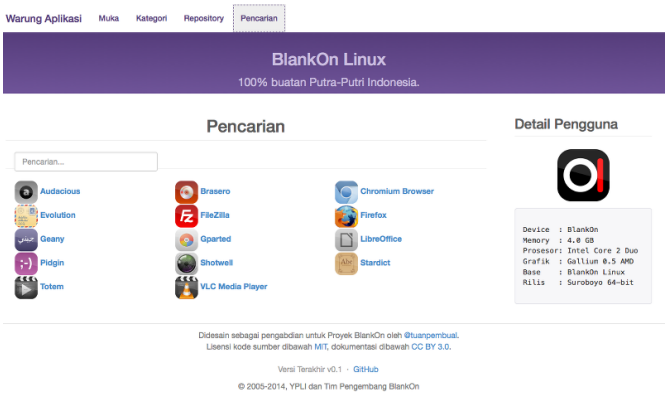

Fig. 12: Implement Search Page

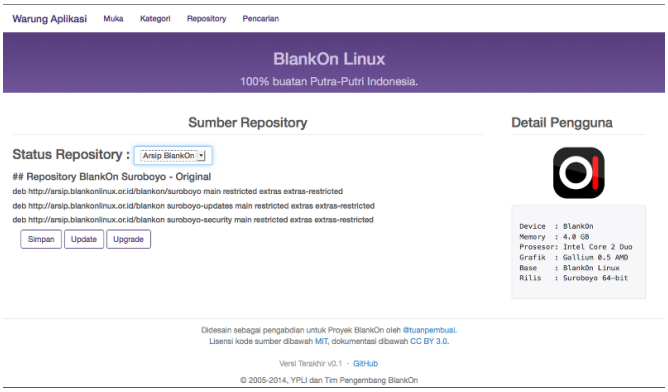

Fig. 13: Implement Repository Management

and focuses solely on the outputs generated in response to selected inputs and execution conditions.[10]

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.

Parameter testing functionally of WarSI include: can it run?; can it install application; can it remove application; can it search application base on name; can it search application base on category; can it change source repository and save it; can it update and upgrade system. Parameter testing usability

TABLE I: Result Functionally of Testing WarSi

Parameter	Yes	No
WarSi run	15	0
Can install application	15	0
Can remove application	15	0
Can search application by name	15	0
Can search application by category	15	0
Can change repository source	15	0
Can update and upgrade system	15	0

TABLE II: Result Usability of Testing WarSi

Parameter	VA	A	N	D	VD
WarSi easy to use	15	0	0	0	0
Will use WarSi again	9	6	0	0	0
Graphic User Interface interactive	10	5	0	0	0
WarSi helped to software management	10	5	0	0	0

Notes: VA=Very Agree A=Agree; N=Neutral; D=Disagree; VD=Very Disagree

of WarSi include: can user use it without trouble (easy to use); will user use it again; how about interface; are users be helped with this application to repository management.

WarSi tested to 15 random correspondents. Table I and Table II show the result of functionally and usability system.

Percentage result functional WarSi:

- Yes : (75/75) x 100 = 100%,
- No : (75/75) x 100 = 0%,

Percentage result usability WarSi:

- Very Agree : (44/60) x 100 = 73%,
- Agree : (16/60 x 100 = 27%,
- Neutral : (0/60) x 100 = 0%,
- Disagree : (0/60) x 100 = 0%,
- Very Disagree : (0/60) x 100 = 0%,

Based on test results involving 15 users, the most of the them give good score to WarSi center software, then the obtained test results showing that 10% of users say functionality of the system has been running well and 0% respondents expressed a functional system is not running properly.

Under the terms of usability testing software center, it is concluded that the majority of respondents are satisfied with the application made. Data usability test results that the respondents strongly agreed by 80%, 20% agree, neutral 0%, which is expressed as 0% disagree and strongly disagree that states as much as 0%. Based on these test results, it can be concluded that the software center that has made this feasible for use. However, the need for further development of the application to obtain optimal application. Figure 15 and Figure 16 show percent of functionally and satisfied user using WarSi.

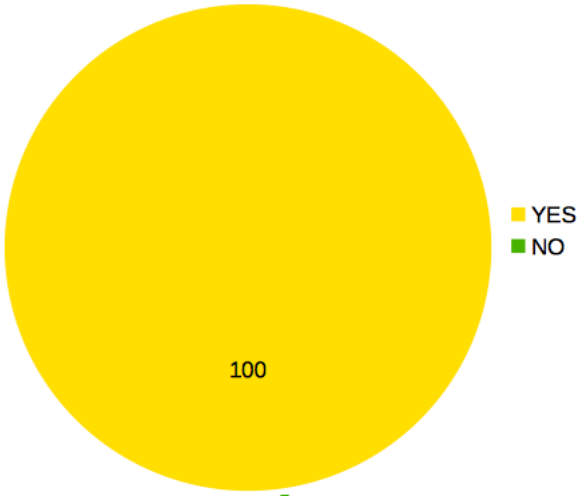

Fig. 14: Result Functionally Testing

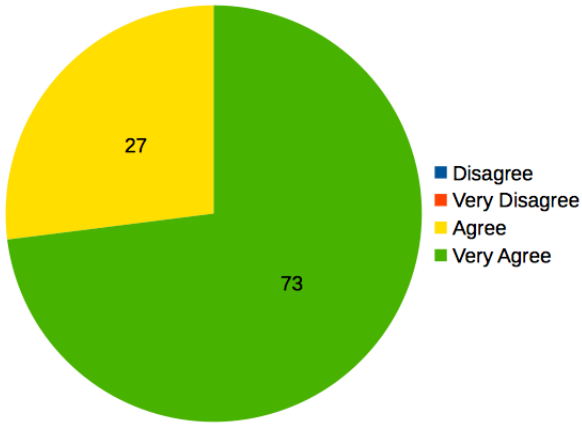

Fig. 15: Result Usability Testing

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.

.

References

[1] *Manokwari, Desktop shell for GNOME 3 with Gtk+ and HTML5 frontend* available at http://manokwari.blankonlinux.or.id/

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

[3] Mathew Paul Thomas, *Ubuntu Software Center*, http://wiki.ubuntu.com/SoftwareCenter, 2005.

[4] *Ubuntu Software Center* available at https://launchpad.net/software-center

[5] Adam Bart, *How WebKit Works*, webkit.org, 2010.

[6] *WebkitGTK+* available at http://webkitgtk.org/

[7] Muhammad Anwari, *GNOME 3 Application Development Beginner*, Birmingham: Packt Publishing Ltd, 2013.

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

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

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

[11] Laurie Williams, *Testing Overview and Black-Box Testing Techniques*, 2006