DEVELOPMENT OF BACKEND ON SOLIDARE SOFTWARE

Adam Mukharil Bachtiar[1], Boby Indra Pratama[2]

[1]Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
[2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : adammb@outlook.com[1], boby.indr@gmail.com[2]

ABSTRACT

Solidare is an application that aims to help children dropout of school, previously this aplication devoted to the social community called “Gerakan Bantu Seribu Anak Asuh (GBSA)”. The annual report of UNESCO Institute of Statistics (UIS) in June 2013 showed that there are 126 million children around the world are dropout of school and 44,4 milion of them live in poverty. Based on data from this report, solidare is previously only devote at GBSA community, turned the concept into a global donation portal that can work together with charities in the world. However based on change this concept, backed system at solidare aplication can not support with new concept. Beside that there is no security at web service solidare currently running.

Therefore, the proposed solution is to develop the software backend solidare to support the system with the concept of change and build a more secure web service at software solidare. The research methodology used is the method applied research. Development is done on the backend include databases, web services and admin website. On the web service performed cryptographic algorithm AES (Advanced Encryption Standard)-128 and the website admin to do development on donation confirmation menu and report foster children. And additional new menu such a charity,donors and recomendation foster children.

After the development and testing of the black box method can be concluded that the backend system developed from the database, web service and website administrators have been able to support the existing system software solidare with the new concept asa donation portal. And web service which had been built already safer with the implementation of a cryptographic algorithm AES-128 into a web service that is built.

Keywords : *Solidare, donation, web service, crypthography, Advance Encryption Standard (AES)-128.*

1. INTRODUCTION

In this section will explain about main of problem, goals and objectives, and the basic of theory.

1.1 Background Problem

Solidare is a software that aims to help children who are dropping out of school. Software solidare previously named ”bantu anak asuh” devoted to a community called “Gerakan Bantu Seribu Anak Asuh (GBSA)”.

The annual report of UNESCO Institute of Statistics (UIS) in June 2013 showed that there are 126 million children around the world are dropout of school and 44,4 milion of them live in poverty [1]. Based on data from this report, solidare turned the concept into a global donation portal that can work together with charities in the world. Therefore Solidare is not only reserved for GBSA, but it can work together with charities in the world. And backend at solidare aplication can not support the existing system software solidare. Another problem is web service is not secure, That is because web service can be accessed easily by everyone. Therefore, it is need for development at the backend application to support the application solidare solidare in the changes concept.

Based on that problem, obtained a backend solution is to develop the application solidare by the database and website development and build a more secure web service and can be support software solidare to change the concept.

1.2 Purpose and objectives

The intent of this research is to develop the software backend solidare. The purpose of this study are as below:

1. Develop a database on software solidare order to support the existing system on the application solidare.
2. Build a new web service that is safer and can support the existing system on the application solidare.

3. Develop a confirmation menu donation and report foster children and adding charities menu,donors and recomendation foster children on the admin website solidare to support the system in solidare application.

1.3 Web Service

Web Service is a set of application logic, objects and methods of its located on one server connected to the Internet [2]. The purpose of this technology is to facilitate aplications or components to interact with another aplications within an organization and outside the organization using platform-neutral and language-neutral. This is because the use of the XML standard that is supported by many large companies in the world, which is used to exchange data. Beside XML, there are also other data exchange format that is JSON.

1.4 JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write It is easy for machines to parse and generate [3]. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

1.5 Cryptography

Cryptography is defined as the science and writing lessons for confidential consideration that communication and data can be encoded to prevent spies or others who want to know its contents, dengan menggunakan kode-kode using codes and certain rules and other methods so that only people who are entitled to be able to know the actual message content. Cryptography has two types of ways to perform encryption and decryption are symmetric and asymmetric. Algorithm symmetric called classical algorithms, because it uses the same key for encryption and decryption activities. whereas asymmetric algorithm often also called public key algorithm, meaning the keywords that are used to perform different encryption and decryption [4].

In this research the type of cryptographic algorithm used is symmetric cryptographic algorithm with the algorithm used is AES (Advance Encryption Standard). AES algorithm itself has three types with different key lengths, including AES-128, AES-192 and AES-256 [5]. Encryption and decryption in AES algorithm can be seen in Figure 1.

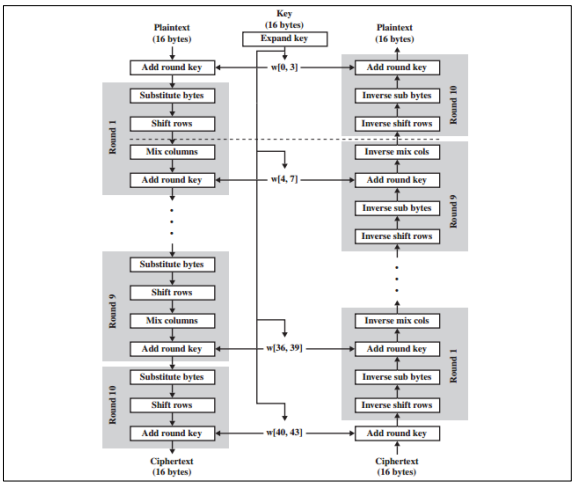


Figure 1 Cryptography on AES Algorithm

1.6 Blackbox Testing

Blackbox testing methods focus on the functional needs of the software, based on software requirements specification. Tests using blackbox trying to find fault in several categories, including:

1. The functions are incorrect or missing
2. Errors interface
3. Errors in data structures or external database access
4. Errors performance
5. The initialization and termination errors

Blackbox has a lot of testing techniques including Equivalence Partitioning, Boundary Value Analysis/Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing. One of the techniques that will be discussed is the Equivalence Partitioning. *Equivalence Partitioning* are black box testing method that divides the input domain of a program into several classes datadari generated test cases [5].

2. Research Contents

2.1 Analysis and Design System

The following will be presented regarding the analysis and design of the system being developed.

2.1.1 Problem Analyze

Solidare is an application that aims to help children dropout of school, previously this aplication devoted to the social community called “Gerakan Bantu Seribu Anak Asuh (GBSA)”. solidare turned the concept into a global donation portal that can work together with charities in the world due to changes in the concept, there are problems on solidare software. these problems are:

1. Database software solidare current operation is not optimal to support the system in solidare application.

2. Web service that is running on the software solidare not currently able to support the existing system on solidare applicationsand there are no security on web service solidare.
3. The menu confirmation and reports that exist on the website solidare current operation is not optimal to support the system solidare, as well as the need for the addition of menu management of charitable institutions, active donors and recommendations on the website solidare foster care in order to support the system in the application as an application portal solidare more global donation.

2.1.2 Cryptographic Algorithm Analysis

Analysis of cryptographic algorithms to explain how the cryptographic algorithm implemented into the existing system software solidare. Cryptographic algorithms used in this software is a cryptographic algorithm AES (Advanced Ecryption Standart) - 128. In the cryptographic algorithms used, there are two processes, processes of encryption and decryption process.

1. Encryption

Text encryption process that is implemented into the system can be seen in Figure 2.

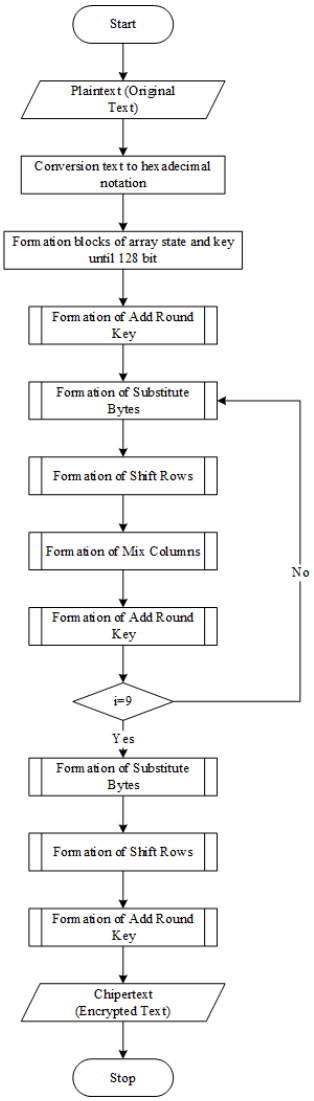


Figure 2 Flowchart of Encryption Text

Explanation of the existing encryption process in Figure 2:

1. Changing the text into hexadecimal notation

This Process is standardized as an initial stage before text is encrypted by system where the text will be converted first into hexadecimal notation.

2. Formation of the blocks array of state and along the 128-bit key

The text that has been transformed into hexadecimal form will then be formed into the blocks along the 128-bit and put in an array called arraystate.

In the process of encryption the text requires a security key that is used as text. Key length will be tailored to the type of AES are used. In this research used AES algorithm is AES-128, which means it has a key length of 16 bytes with the calculation of 128/8 = 16. The key will be transformed into hexadecimal notation as text on the previous stage.

3. Formation Add Round Key

This stage is the initial state XOR process between the key state array. The point is to XOR the first column in an array of key state with the first column of up to N times throughout the text is discharged is formed into an array of state.

4. Formation SubstituteBytes

This process aims to map or swap each byte of the state array using the substitution table S-Box.

5. Formation Shift Rows

This process aims to shift the last three lines in an array of state. Shifting line is the line to r = 1 by 1 byte, row r = 2 by 2 bytes and lines to r = 3 by 3 bytes.

6. Formation Mix Columns

This process aims to multiply each column of the array of state with a polynomial a (x) mod (x4+1). This transformation is expressed as a matrix multiplication:

$$\begin{bmatrix} S'0c \\ S'1c \\ S'2c \\ S'3c \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} S0,c \\ S1,c \\ S2,c \\ S3,c \end{bmatrix}$$

7. Formation Add Round key

Before performing this process Add Round Key, the next step is to expand the key or form a new key to the process of getting key Add Round Key.Setelah expansion, then the next step is to perform calculations Add Round Key.

8. Formation SubstituteBytes

This process will map the results back from Add Round Key to the table S-Box.

9. Formation Shift Rows

Rows Shift process will once again shift the last line on the array of state. r = 1 will be shifted 1 byte, r = 2 will be shifted 2 bytes, and r = 3 will be shifted 3 bytes.

10. Formation Add Round key

Add Round Key process is the last stage or the 10th round of the encryption process an array of state along the 128 bit, then the next key is to expand or establish new key for the Add Round Key. After the expansion key is obtained, the next step is the calculation Add Round Key. After the encryption process is done, the text still exists on blockarray state. Then the system will be transformed back to be encrypted text.

2. Decryption

Text decryption processes are implemented into the system can be seen in Figure 3.

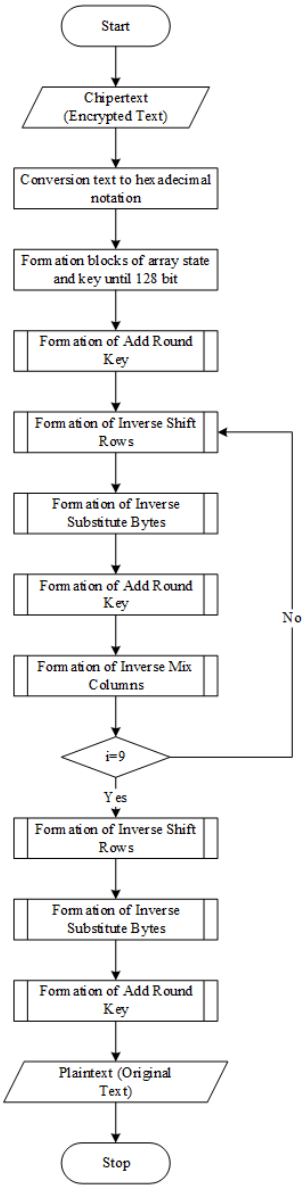


Figure 3 Flowchart of Decryption Text

Explanation of existing decryption process in Figure 3 is as follows:

1. Changing the text into hexadecimal notation

The text will be decrypted to be converted first into hexadecimal notation by the system.

2. The formation of the blocks array of state and along the 128-bit key

Of the process that has been carried out the previous stage, it will then be reshaped into blocks along the 128-bit and put into an array of state. Cipherkey used was the same with the encryption key when the text on the final lap encryption process.

3. Formation Add Round Key

After Add Round Key process is finished, it will produce an array of state whose value is equal to the Shift Rows. That is, this process aims to restore the value by XOR or rollback.

4. Formation Inverse Shift Rows

This process aims to slide back end of the array state row to the starting position.

5. Formation Inverse Subtitute Bytes

Substitute Bytes Inverse process will be remapped array of state results from the Inverse Shift Rows with tables S-Box.

6. Formation Add Round Key

Add Round Key process will restore the value of the array state by conducting the process XOR the results of Inverse Substitute Bytes with a key that has been in previous expansions.

7. Formation Inverse Mix Columns

This process aims to restore the result of the multiplication of each column of the array state that has been done in the mix column at the time of encryption.

8. Formation Inverse Shift Rows

Inverse Shift Rows process will again shifted one byte at the end of the line in an array of state.

9. Formation Inverse Subtitute Bytes

Substitute Bytes Inverse process will be re-charted every state array value with the S-Box table to return any value in the state array into the initial conditions before entering the Add Round Key first round when encryption.

10. Formation Add Round Key

Add Round Key process is the final process of a series of steps the decryption process of the text. This phase will be conducted between the expansion process XOR encryption key first round when the Add Round Key to the results of the Inverse Substitute Bytes. Add Round Key Once the process is completed, the data will be returned in accordance with the text of the initial conditions before going through the encryption process is done. However, the text must be transformed first as a result of the decryption is still in the form of hexadecimal notation. Before transformed, hexadecimal notation is formed into blocks as 128bit state array to array the last state in the N position. Once this process is completed, a new transformation by the system to be restored to the original text or original text.

2.1.3 Data analysis

Analysis of the data which is used in software development are the documents that relate to software solidare and other objects related .Entity relational diagram (ERD) and cardinality can be seen in Figure 4.

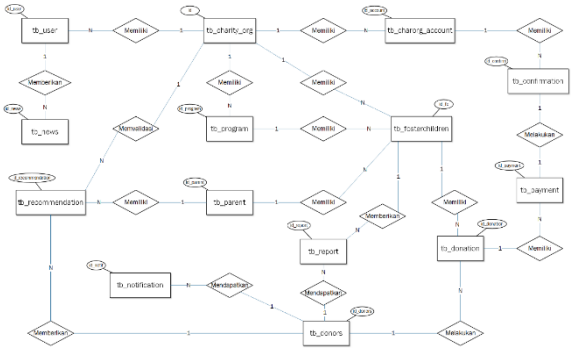


Figure 4 ERD and Cardinality

Another concept of data analysis used in this study is the concept of Object Relational Mapping (ORM). Object relational mapping do the mapping to tables in a relational database with an existing entity class in object-oriented programming language.

ORM mapping is done would require a bridge as a medium of exchange data. The concept of exchange of the data used in this study of the JSON data format that can connect to the database ORM physically exist on the server. The structure of the JSON data format used in this study is divided into three object data, the data array and null data.

1. Object Data

The data format when the data received from the server is a single data or in the form of an object can be seen in Table 1.

Table 1 Structure Data Object

Structure Data Objects
{ "key" : "value", "key" : "value", "key" : "value" }

2. Data Array

The data format when the data received from the server is a data array can be seen in Table 2.

Table 2 Structure Data Array

Structure Data Array
{ "result" : "value", "item" : [{ "key" : "value", "key" : "value", "key" : "value" }, { "key" : "value", "key" : "value", "key" : "value" }] }

3. Data Null

The data format when the data received from the server is null can be seen in Table 3.

Table 3 Structure Data Null

Structure Data Null

{
 "result" : "value",
 "message" : "value",
 "status_code" : "value"
 }

2.1.4 Functional Needs Analysis

Here is a diagram of the context of the development of this software can be seen in Figure 5.

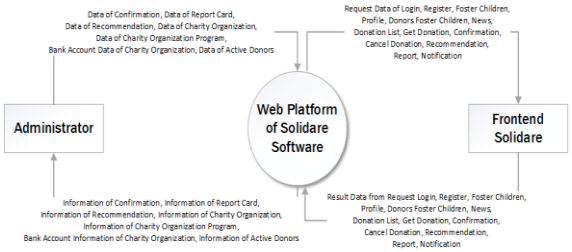


Figure 5 Context Diagram

Here is the DFD level 1 of the development of this software can be seen in Figure 6.

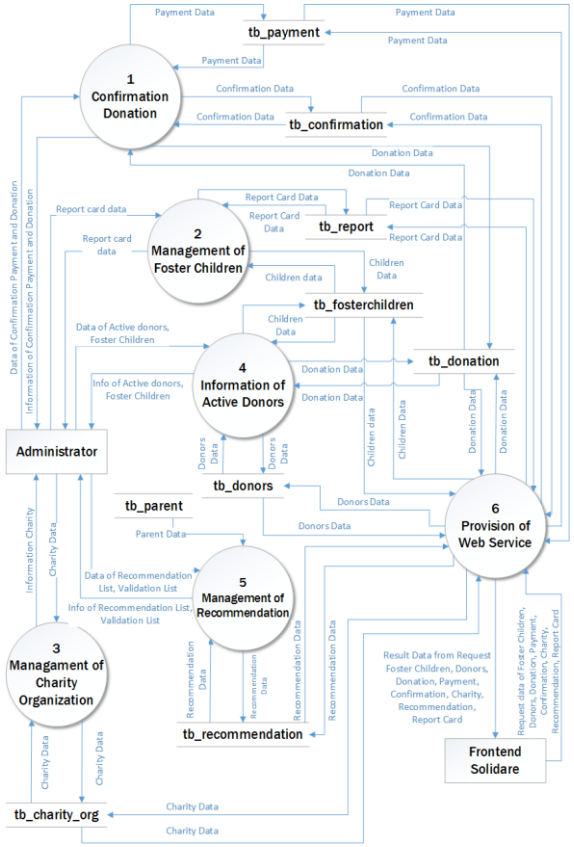


Figure 6 DFD Level 1

2.2 Implementation and Testing System

The following will be presented on the implementation and testing of software systems developed.

2.2.1 Implementation Environment

Implementation environment is a hardware and software specifications in which these systems will be installed and accessed. Here is the implementation environment:

1. Hardware Environment

Here are the hardware specifications in the development of this software.

Table 4 Hardware Environment Of Software Development

No	Item	Spesification
1	Processor	Intel(R) Core i7-2640M @2.80Ghz
2	Harddisk	750 GB
3	Memory	4 GB
4	VGA	Intel HD Graphics 3000
5	Monitor	13.3 inch
6	Mouse	Optical Logitech
7	Keyboard	Qwerty Sony S Series
8	USB Port	3 buah
9	Internet connection	Available

2. Software Environment

The software used in the development of this software are:

a. Operating System Windows 8.1 Professional 64-bit.
b. Programming language PHP.
c. DBMS : MySql 5.6.14.
d. Web Browser : Google Chrome 43.0.2357.130 mdan Mozilla Firefox 38.5.
e. Code Editor used Sublime text.
f. Web Server : XAMPP.
g. The exchange of data is done using JSON.

2.2.2 Implementation Interface

Here is an implementation of the interface of the website developed:



Figure 7 Management of Confirmation Donation

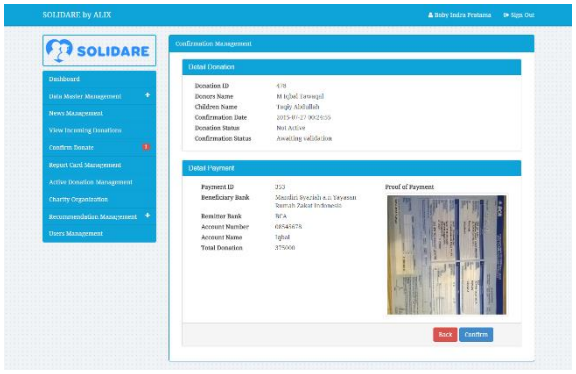


Figure 8 Detail Confirmation Donation

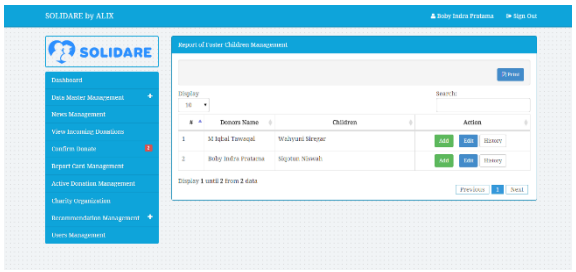


Figure 9 Report Card Management of Foster Children

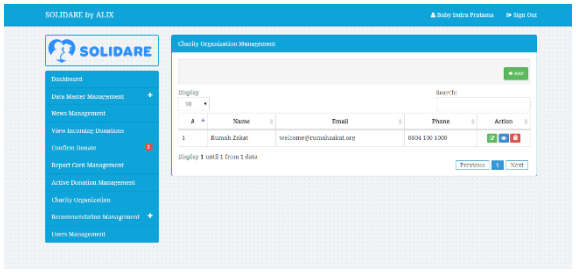


Figure 10 Management Charity Organization

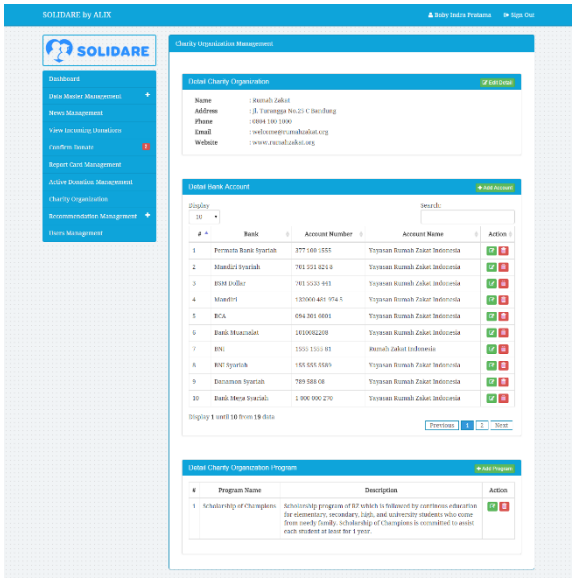


Figure 11 Detail Charity Organization

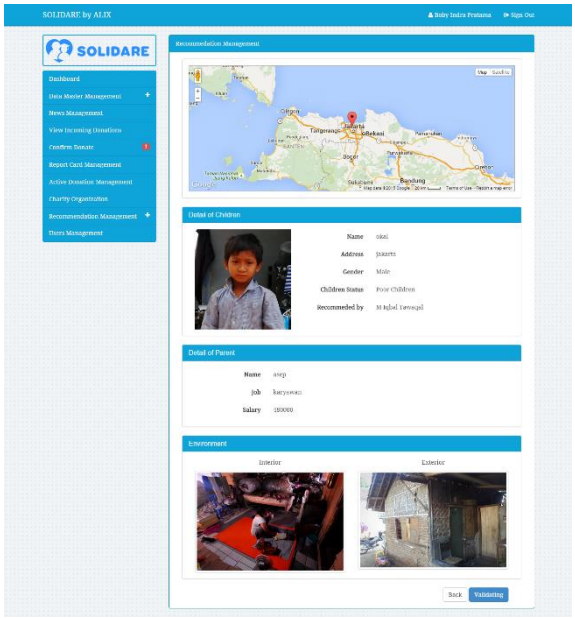


Figure 12 Detail of Recommendation

2.2.3 Testing

Testing of the system consists of two stages of testing alpha and beta testing. The method used in the testing of this system is to use black box testing methods where this test focuses on the functional requirements of software.

1. Plan Testing

Plan testing to be performed on the backend software solidare is functional testing with the kind of black box testing.

2. Scenario Testing

Scenario functional testing is done by testing the website solidare scenario testing sequence that can be seen in Table 5

Tabel 1Table Scenario Testing Website Admin Solidare

No	Class Of Testing	Point Of Testing	Type Of Testing
1	Search data report foster children	Filling the search keywords	*Equivalence Partitioning*
2	Add data report foster children	Filling data report	*Equivalence Partitioning*
		Validation report field	*Equivalence Partitioning*
		Push the button save data	*Equivalence Partitioning*
3	Data retrieval charity	Filling the search keywords	*Equivalence Partitioning*
4	Add data Charity	Fill data Charity	*Equivalence Partitioning*
		Validasicharity field	*Equivalence Partitioning*
		Push the button save data	*Equivalence Partitioning*
5	Searches charity bank account data	Filling the search keywords	*Equivalence Partitioning*
6	Add data bank account of	Filling data bank account of charity	*Equivalence Partitioning*

No	Class Of Testing	Point Of Testing	Type Of Testing
	charity	Validation bank account of charity field	*Equivalence Partitioning*
		Push the button save data	*Equivalence Partitioning*
7	Add the program data of Charity	Filling program data of charity	*Equivalence Partitioning*
		Validation program of charity *field*	*Equivalence Partitioning*
		Push the button save data	*Equivalence Partitioning*
8	Search data active donor	Filling the search keywords	*Equivalence Partitioning*
9	Search data recommendation foster children	Filling the search keywords	*Equivalence Partitioning*
10	Seacrh data validation recommendation foster children	Filling the search keywords	*Equivalence Partitioning*

3. Evaluasion Of Testing

Based on the results of functional testing has been done it can be concluded that the website that has been developed has been running as expected, both in terms of process validation and error handling.

3. Conclusion

The conclusion of the backend software development solidare are:

1. The database has been developed to support the existing system software solidare.
2. Web service that has been built has been more secure by the implementation of a cryptographic algorithm AES-128 and web service that has been built can be able to support the existing system software solidare.
3. Menu confirmation report donations and foster children has been developed by making changes to the UI (user interface) and functional additions. Menu charity, active donors and recommendations foster children have been added to the website solidare. All menus that have been developed and added to already be able to support the existing system software solidare and has been running as expected.

REFERENCES

[1] UNESCO INSTITUTE FOR STATISTICS, Fixing The Broken Promise Of Education For All, UNICEF, 2014.

[2] J. Simarmata, Rekayasa Web, Yogyakarta: ANDI, 2010.

[3] j. org, "json org," [Online]. Available: http://json.org/json-id. [Diakses 25 April 2014].

[4] B. Schneier, Applied Cryptography, India: Wiley, 2007.

[5] W. Stalling, Cryptography and Network Security Principles and Practice, Pearson, 2014.

[6] M. E. Khan dan F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *International Journal of Advanced Computer Science and Applications,* vol. 3, no. 6, pp. 12-15, 2012.