Implementation of CAS Server as Authentication Protocol on Single Sign-On (SSO) Network With PHP Programming

Amarudin[*]

[*]*Informatics Engineering Department*
STMIK Teknokrat, Bandarlampung, Indonesia
amarudins@gmail.com

Abstract **— Web application development demanding user applications to manage users and passwords application system as well indirectly, so as not to be abused by not authorized parties. User account security methods have been tried to solve the problem. Including by implementing authentication service on Single Sign On (SSO) network, applying Server Radius as captive portal authentication, and by building a CAS Server as authentication Center and with each others methods by programming languages variety.**

This research was conducted by means of experimenting to make a CAS Server as authentication center that built with PHP programming language. Aside from that, this paper also describes how to test the performance of CAS Server by using tools hacking application (Apache Banch). The tools hacking application can run on Linux Ubuntu operating system. Whereas the methods used in this testing is black box methods (functional testing).

The results showed that CAS Servers have been built as a authentication center in Single Sign On (SSO) network was able to respond requests maximum for users as much as 4.490.000 request, for a total time of request 5.406,568 ms (5 seconds). Thus CAS Server that has been built is very suitable to handle the user request is less than 4.490.000 request.

Key words - Single Sign On (SSO), Authentication, CAS Servers, Hacking.

I. INTRODUCTION

The Internet is a medium of information and communication that is often used these days. In fact, almost all formal and informal institutions, not spared to use the Internet. The usage of Internet is certainly depend on the usage of the website. In fact, almost no formal institutions that do not have a website. Either a local or a website portal that is used as a public information system that is accessible from the outside (Internet). However, along with the development of the web application, indirectly it's require user of application and admin system to can manage the user account and the system applications with good, so the user account don't misuse by unauthorized parties.

Based on these problems, in this study aims to discuss the implementation of the CAS Server as a authentication center in a SSO network system that is built with PHP programming language. And Then tested the ability of the CAS Server is with hacking tools. As for the hacking tools used is Apache Banch (AB).

II. RELATED WORK

According to Gao Zheng, et al. [1], Central Authentication Service (CAS) as the central authentication is used. CAS is a web application design open source Single Sign-On, published by Yale University. Additionally, it's describe about usage the https protocol encryption to encrypt communication as a safeguard, saving the identity of bills in the authentication center, without the browser to save cookies.

Despite this, CAS there are still some deficiencies: 1. The user that in multiple data sources will be difficult to verified caused more than one database holds all registered users of the services. 2. CAS notes and session storage are not very good support for distributed, web flow has some shortcomings for distributed mechanisms. 3. CAS verification through the user and service and allow access, just put the user information into session, but for the user's authentication using attribute cannot.

So, by extending the open-source framework of CAS, enable them to make up for the lack of these points above, which will enable the SSO to complete the more powerful features in addition to certification methode.

Based on research conducted by G. Ramadan [2], the products SSO system based on open source that is commonly used today include CAS, OpenAM (Open Access Manager), and JOSSO (Java Open Single Sign-On). However, in this study suggested the name of the product from the CAS. This is the reason that CAS is a free (open source) and many library that support with client.

III. BASIC THEORY

A. *Introduction of Single Sign On (SSO)*

SSO is a mechanism that makes the user only needs to remember one user name and password that authentic to open multiple services at once. SSO needs to be authenticated once, then authentication will happen automatically when the user opens another website through a session [1].

By used SSO, a user just simply does once the authentication process to gain permits access to all the services contained within the network. An example is the Google account that is integrated with google apps (google drives, google plus, gmail, etc.). With a single login to one of the google apps, then automatically the Google has logging in to other apps.

The SSO Server is actually not different web applications in General, its task is to handle the request and provide a response to the client that access. Just reply SSO server task is 99% of authentifikasi, the processing of session and coockies. In addition, an SSO server must be able to handle the request and response quickly [2]. The basic concept of SSO can be seen in Figure 3.1.

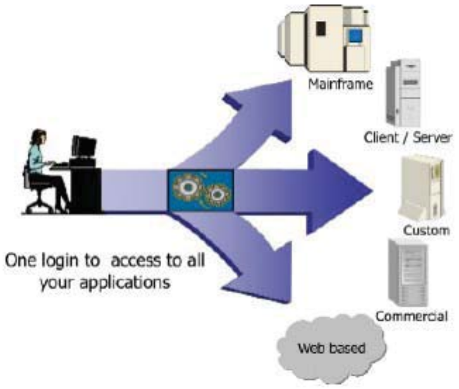


Figure 3.1. Basic concept of Single Sign On (SSO) [3]

B. *Definition of CAS*

The Central Authentication Service (CAS) is an authentication system originally created by Yale University to provide a safe way for an application to authenticate a user. The CAS then implemented as an open source Java server component and a client library support for Java, PHP, Perl, Apache, uPortal, and others. CAS Server is a basic framework that used for SSO network security [4]. To more clearly understand the basic concepts of CAS can be seen in Figure 3.2.

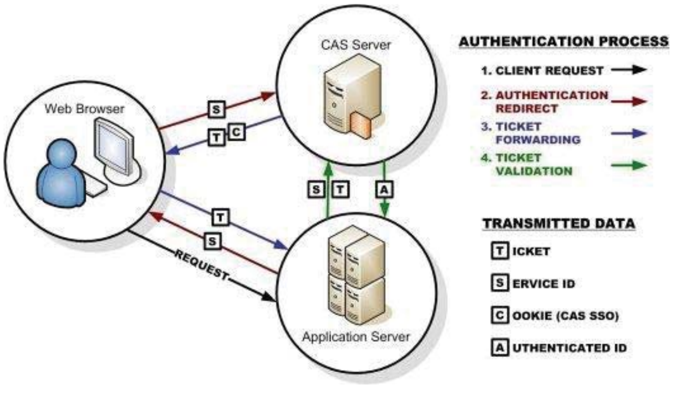


Figure 3.2. CAS Authentication Concepts [5]

C. *Hacking and Sniffing Tools*

Hacking (attack) is a computer program belonging to break through the activities of people/other party. While hackers are people who do hacking and usually have the expertise to create and read a particular program, and obsessed with observing his security [6]. While Sniffing is eavesdropping on data traffic on a computer network that is carried out by sniffer [6]. In the test system used in this study is Apache Banch (AB). In addition, there are kind of attack that called Spoofing. It is one of more form attack that pernicious in the networking system [7].

D. *Black Box Testing*

Black Box testing (functional testing) is one of the conditions of the test developed by the program or system function. Where tester (examiner) requires information about the data input and output to be observed, but did not know how to program or system is working. As someone who does not have to know how a car works internally to drive it, do not need to know the internal structure of the program work. The tester function testing focuses on the specifics of the program. With black-box testing, the tester saw the program as a black box and really do not care about the internal structure of a program or system. Some examples in this category include: decision tables, equivalence partitioning, range testing, boundary value testing, database integrity testing, graphing cause effect, orthogonal array testing, array and table testing, exception testing, limit testing, and random testing [8].

IV. DESAIN SYSTEM

SSO network system design that is constructed in this study were divided into two groups of systems. That CAS Server as the primary system is used as the authentication center. And the next group is the CAS Client systems that function as activators session on the application on the client side. To understand the SSO network system design that is built in this study, can be seen in the description of the SSO network system design and SSO authentication data flow scenario the following:

A. *SSO Network Systems Design*

The following is an explanation of the design of a network system consisting of an Information Systems Server Web and a CAS Server. The architecture of network system Single Sign On (SSO) as a whole in this study are as described in Figure 4.1.

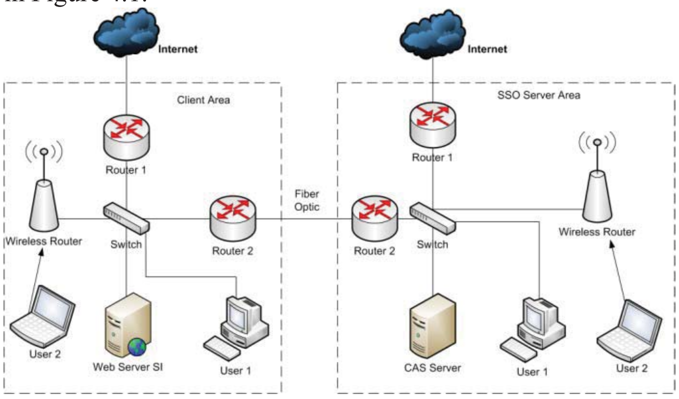


Figure 4.1. SSO Network Systems Design

Each user was located in client area that will login to access a Web Server SI, then immediately directed (redirect) to the CAS Server that was located in the SSO Server Area with a service address to authenticate by checking whether the user is already registered in the CAS Server database. Once successfully authenticated, then from CAS Server redirected back to address service (CAS Client) and then directed to a Web Server SI (Client Area).

Similarly, for each user who is in the SSO Server Area, which will access the Web Server SI, authenticating to CAS Server first and then once authenticated redirected back to the Web Server SI originating in accordance with the submitted address service.

Information specification and selection of the components shown in Figure 4.1 with the following explanation:

1. CAS server has the specs Processor Intel Pentium Core 2 Duo, 4 GB RAM, 500 GB Hard Disk, with Centos Linux OS version 6.4 (64-bit).
2. Web Server SI with specifications Processor Intel Pentium Dual Core 1.8 GHz, 3 GB RAM, 320 GB Hard Disk, with dual boot OS are Linux OpenSUSE, and Windows 7 Ultimate, which serves as a computer attacker and Information System Web Server Papyrus (replica).
3. Switch: TP-LINK TL-SF1016D. Port Number: 16 Port 10/100 Mbps. Media interfaces: RJ-45: 100 ohm, UTP / STP cable, 10/100 Base - EIA/TIA Categories 3 or 5 cable, and LED.
4. WiFi: 300Mbps Wireless-N Access Point TP-Link TL-WA801ND.

B. SSO Authentication Data Flow Scenario

Design of network systems that is integrated with SSO to authenticate on CAS Server has scenario as described in the Figure 4.2.

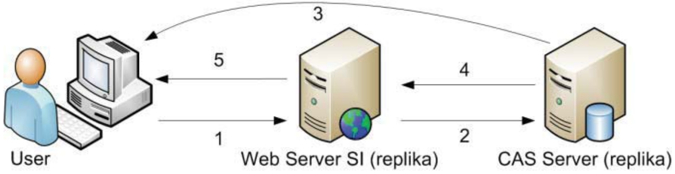


Figure 4.2. Sequence Data Flow Scenario SSO Authentication

The scenario in Figure 4.2 is marked with the serial number one to five can be explained as follows:

1. Users access the Web Server System Information (SI) through the browser on the user PC with the address http://www.amarudin.com
2. From the Server SI then redirected to the CAS Server with a service address to authenticate whether the user account is registered in the data base by the CAS Server.
3. If the user account has not been registered in the data base of the CAS Server, then the user can not access the Web Server System Information directly, but returned again to the user without going through the Information System Web Server.
4. If the user account is registered in the data base of the CAS Server, then from the CAS Server is directed to the CAS Client Web Server System are in accordance with the address information service that has been delivered.
5. From the Web Server Information Systems were directed to the user's PC in a state already authenticated. If the user opens another application that is integrated with SSO, then the application system does not need to authenticate again to the CAS Server, but just uses a User Name that is in session on the client browser.

By usage of authentication session is expected authentication process to be faster than without using the session that always logged into the CAS Server.

C. SSO Authentication Flowchart

The authentication process on the SSO network can be described in the flowchart that shown in Figure 4.3.

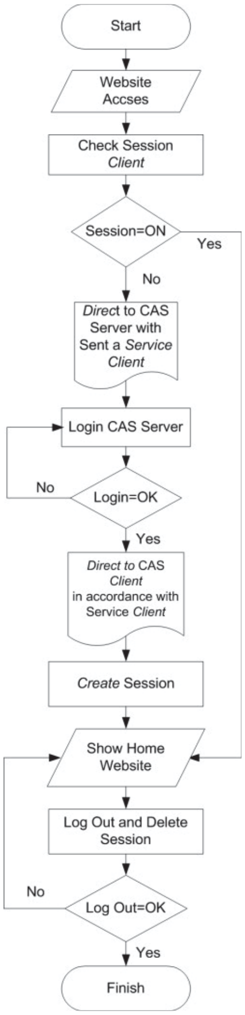


Figure 4.3. SSO Authentication Flowchart

D. CAS Server Application Design

The design of CAS Server application in this study using the PHP programming language and the MySQL database. The CAS Server login page display is shown in Figure 4.4.

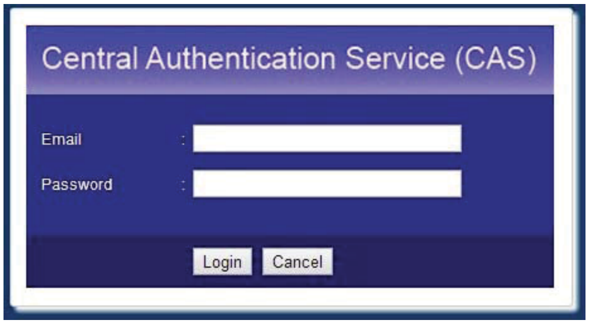
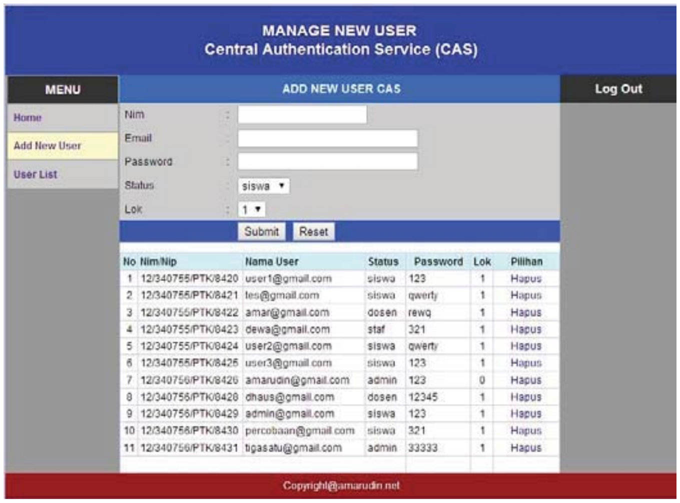

Central Authentication Service (CAS)

Figure 4.4. CAS Server Login Page

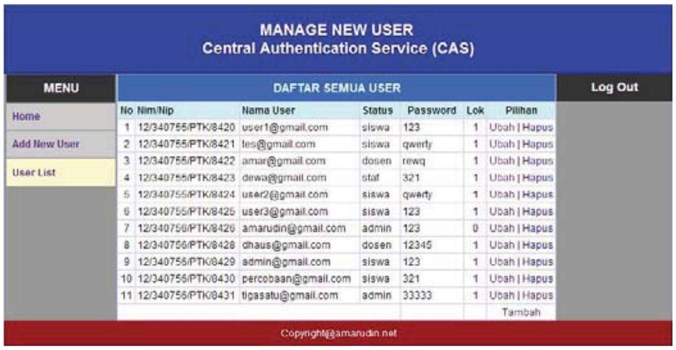
The display form for filling and processing user data is shown in Figure 4.5.



No	Nim/Nip	Nama User	Status	Password	Lok	Pilihan
1	12/340755/PTK/8420	user1@gmail.com	siswa	123	1	Hapus
2	12/340755/PTK/8421	tes@gmail.com	siswa	qwerty	1	Hapus
3	12/340755/PTK/8422	amar@gmail.com	dosen	rewq	1	Hapus
4	12/340755/PTK/8423	dewa@gmail.com	staf	321	1	Hapus
5	12/340755/PTK/8424	user2@gmail.com	siswa	qwerty	1	Hapus
6	12/340755/PTK/8425	user3@gmail.com	siswa	123	1	Hapus
7	12/340756/PTK/8426	amarudin@gmail.com	admin	123	0	Hapus
8	12/340756/PTK/8428	dhaus@gmail.com	dosen	12345	1	Hapus
9	12/340756/PTK/8429	admin@gmail.com	siswa	123	1	Hapus
10	12/340756/PTK/8430	percobaan@gmail.com	siswa	321	1	Hapus
11	12/340756/PTK/8431	tigasatu@gmail.com	admin	33333	1	Hapus

Figure 4.5. Add New User Page

Meanwhile, to see a list of all user accounts and also to be able to modify and delete user accounts can be seen via the User menu list as shown in Figure 4.6.



No	Nim/Nip	Nama User	Status	Password	Lok	Pilihan
1	12/340755/PTK/8420	user1@gmail.com	siswa	123	1	Ubah | Hapus
2	12/340755/PTK/8421	tes@gmail.com	siswa	qwerty	1	Ubah | Hapus
3	12/340755/PTK/8422	amar@gmail.com	dosen	rewq	1	Ubah | Hapus
4	12/340755/PTK/8423	dewa@gmail.com	staf	321	1	Ubah | Hapus
5	12/340755/PTK/8424	user2@gmail.com	siswa	qwerty	1	Ubah | Hapus
6	12/340755/PTK/8425	user3@gmail.com	siswa	123	1	Ubah | Hapus
7	12/340756/PTK/8426	amarudin@gmail.com	admin	123	0	Ubah | Hapus
8	12/340756/PTK/8428	dhaus@gmail.com	dosen	12345	1	Ubah | Hapus
9	12/340756/PTK/8429	admin@gmail.com	siswa	123	1	Ubah | Hapus
10	12/340756/PTK/8430	percobaan@gmail.com	siswa	321	1	Ubah | Hapus
11	12/340756/PTK/8431	tigasatu@gmail.com	admin	33333	1	Ubah | Hapus
						Tambah

Figure 4.6. User Accounts list on the CAS Server

E. Listing Program of CAS Server

CAS Server must provide user data to be used as the authentication center. So the CAS Server must also provide the address to be accessed by the client. Here are a few links that provided by CAS Server:

```
Host         : https://10.55.1.246
Port         : 443
Context      : /cass/
Login URL    : https://10.55.1.246/cass/login
Logout URL   : https://10.55.1.246/cass/logout
```

Because of so many listing program that used on the CAS Server application, then in this paper include the main program listing used. The program listing as follows:

```
<?php
$serviceHidden = $_REQUEST['txtServiceHidden'];
$username   = $_REQUEST['TxtUserName'];
$password   = $_REQUEST['TxtPassword'];
if((trim($username)=="")and(trim($password)=="")){
             include_once
"librari/inc.koneksidb.php";
      $cek=mysql_query("SELECT * FROM tb_user
      WHERE username='$username'
      AND password='$password'");
   if(mysql_num_rows($cek)==1)
      {//jika berhasil akan bernilai 1
      $c = mysql_fetch_array($cek);
      $user= $c['username'];
      //simpan user pada session server
      session_start();
      $_SESSION['username']= $c['username'];
      $_SESSION['status']  = $c['status'];
      //kembali ke alamat asal.
      echo "<meta http-equiv='refresh'
      content='0;
      url=$serviceHidden?username=$user'>";
      //buka LOK pada tabel user
      $sql  = "UPDATE tb_user SET lok='0'
      WHERE username='$username'";
      mysql_query($sql, $koneksi) or die
      ("SQL Error".mysql_error());
      }
     }
?>
```

V. System Testing Method

In the CAS Server system testing, is done by the method of black box testing model (functional testing). The testing is done with performance testing. The test load and response time. Architecture of the testing Server on the CAS Single Sign On (SSO) network can be seen in Figure 5.1.

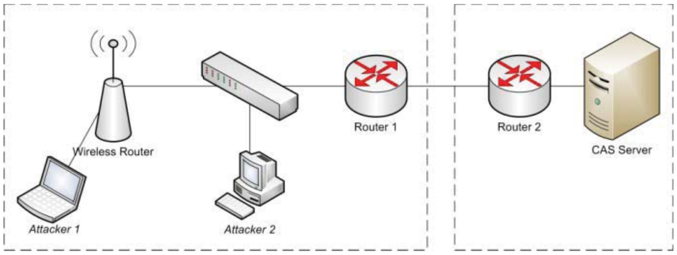


Figure 5.1. CAS Server Architecture Testing

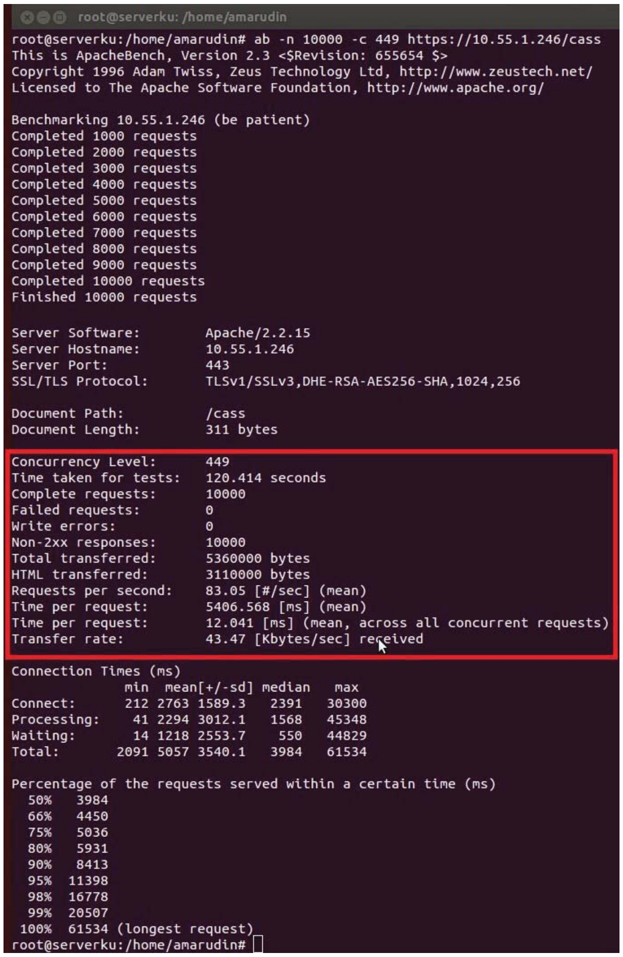
VI. Results And Discussion

The results of testing the CAS Servers in this performance testing was using Apache Banch (AB) tools. The results of system performance testing conducted CAS Server obtained the results as shown in TABLE 1.

TABLE 1
System Performance Test Results

No	Tested Components	Testing Result	Information
1.	Total Connections (n)	10.000	Total Connections to CAS Server
2.	Concurrency Level (c)	449 request	On request to 449 already not normal.
3.	Request per Second	83,05 request/s	The number of requests in one second
4.	Time per Request	12,041 ms	The time it takes in a single request
5.	Transfer Rate	43,47 kb/s	The average transfer times
6.	Total Time Request	5.406,568 ms	Entire user request time

To view the capture performance test results on CAS Server at this stage can be seen in Figure 6.1.



```
root@serverku:/home/amarudin# ab -n 10000 -c 449 https://10.55.1.246/cass
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 10.55.1.246 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:        Apache/2.2.15
Server Hostname:        10.55.1.246
Server Port:            443
SSL/TLS Protocol:       TLSv1/SSLv3,DHE-RSA-AES256-SHA,1024,256

Document Path:          /cass
Document Length:        311 bytes

Concurrency Level:      449
Time taken for tests:   120.414 seconds
Complete requests:      10000
Failed requests:        0
Write errors:           0
Non-2xx responses:      10000
Total transferred:      5360000 bytes
HTML transferred:       3110000 bytes
Requests per second:    83.05 [#/sec] (mean)
Time per request:       5406.568 [ms] (mean)
Time per request:       12.041 [ms] (mean, across all concurrent requests)
Transfer rate:          43.47 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:      212 2763 1589.3   2391   30300
Processing:    41 2294 3012.1   1568   45348
Waiting:       14 1218 2553.7    550   44829
Total:       2091 5057 3540.1   3984   61534

Percentage of the requests served within a certain time (ms)
  50%   3984
  66%   4450
  75%   5036
  80%   5931
  90%   8413
  95%  11398
  98%  16778
  99%  20507
 100%  61534 (longest request)
root@serverku:/home/amarudin# 
```

Figure 6.1. The test results with a Concurrency Level of 449

VII. CONCLUSION

Based on the research and the results of the analysis system has been done, some of the conclusions obtained as follows:

1. Implementation of CAS Server and CAS Client are constructed by PHP programming language can function well.
2. Exploiting session on the CAS Server can not be used directly by the browser if without of CAS Client. Resulting in building a network of SSO, must involve both the CAS (CAS Server and CAS Client).
3. Based on test performance (response time and load test) conducted with tools Apache Benchmark (AB), that the CAS Server maximum can respond to user 4.490.000 user with as much Total Time Request = 5.406,568 ms (5 seconds).

VIII. FUTURE WORK

The future work for further research is need for further testing in the design of the CAS Servers that are built with different programming languages and different authentication methods.

REFERENCES

[1] P. P. Nugroho, "Pengembangan Model Single Sign-On untuk layanan Internet dan Proxy IPB," S1, Ilmu Komputer Institut Pertanian Bogor, 2012.

[2] Ahsan, "Membangun SSO Server Untuk Implementasi Single Sign ON," ed, 2011.

[3] S. Xiong, "*Web Single Sign-On System For WRL Company*," 2005, p. 79.

[4] K. Aaslund, "Central Authentication Service (CAS)," 2007.

[5] A. Pavlenko and I. Chicago. (2013, 28 MAret 2014). *concept of CAS authentication - Penelusuran Google*. Available: http://www.nobigo.com/central-authentication-services-explained/casdiagram_large/#.UzTsYfmSy8Q

[6] O. Priawadi. (2012). *Definisi atau pengertian dari Spamming, Hacking, Malcious Software (Malware), Snooping, Sniffing, Spoofing, Pharming, Defacing, Phising dan Jamming*. Available: http://priawadi.blogspot.com/2012/05/pengertian-spamming-snooping-spoofing.html

[7] H. Yinghui and L. Guanyu, "A Semantic Analysis for Internet of Things," in *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, 2010, pp. 336-339.

[8] W. E.Lewis, *Software Testing and Continuous Quality Improvement*: Auerbach Publication, 2005.