JURNAL RESISTOR
Vol. 1 No 1 – April 2018
p-ISSN 2598-7542 (Print), e-ISSN 2598-9650 (Online)
Available Online at :
http://jurnal.stiki-indonesia.ac.id/index.php/jurnalresistor

RANCANG BANGUN IMPLEMENTASI APLIKASI *SINGLE SIGN ON* PADA SISTEM PEMBELAJARAN DAN SISTEM INFORMASI BERBASIS WEB

I Putu Agus Eka Darma Udayana

Manajemen Sistem Informasi Dan Komputer Universitas Udayana, Fakultas Teknik, Universitas Udayana Denpasar, Indonesia

e-mail: agus.ekadarma@gmail.com

Received : April, 2018	Accepted : April, 2018	Published : April, 2018

Abstrak

Elearning dan sistem informasi berbasis web merupakan media yang digunakan sebagai sarana berkomunikasi dan bertukar informasi untuk kepentingan akademik. Salah satu metode untuk otentikai user yang sering digunakan adalah menggunakan LDAP (Lightweight Directory Access Protocol). LDAP memungkinkan pengguna aplikasi untuk login pada semua aplikasi berbasis web hanya menggunakan satu username dan password. Namun ketika ingin masuk dan mendapatkan hak akses pada setiap aplikasi, user harus mengisi username dan password yang sema secara berulang kali. Untuk mengatasi proses otentikasi dengan melakukan input username dan password berulang kali pada aplikasi yang ingin di akses, dikembangkanlah metode otentikasi Single Sign On (SSO). Dimana metode SSO ini memungkinkan user hanya melalukan satu kali login/otentikasi pada sejumlah aplikasi yng terintegrasi pada sistem Single Sign On. Untuk membangun sistem Single Sign On tersebut digunakan Central Authentication Service (CAS) sebagai titik utama proses otentikasi user dan menggunakan LDAP sebagai pusat manajemen user. Pada peneletian ini, sistem berbasis web yang terintegrasi dengan sistem SSO yaitu Simak, E-Learning dan Blog System telah dapat terintegrasi dengan sistem SSO, dimana user hanya perlu melakukan satu proses otentikasi untuk mendapat hak akses dari semua aplikasi tersebut.

Kata Kunci: *Single Sign On, LDAP, CAS*

Abstract

Elearning and web based information systems is a means to communicate and exchange information for academic purposes. Nowadays lightweight directory access protocol (LDAP) is a state of the art method of choice. With LDAP technologies user only need one username and password to access to multiple web based application, The problem is if the user wanted to do autentification said user had to input their credentials over and over again for each application. To solve that problem single sign on mechanism (SSO) is invented. With SSO user only need login once and they got all the same credentials with them to all intergrated application wthin the campus. To implement the SSO we use Central authentication service (CAS) as a authentifiation central within LDAP structure as a user management. In this reseach we see that single sign on (SSO) system that intergrated into student management system, E-Learning system and Internal blog system both use of database based system or even LDAP based system.

Keywords: *Single Sign On, LDAP, CAS*

1. PENDAHULUAN

Seiring dengan perkembangan zaman dan meluasnya pemakaian media internet, pengguna layanan akan mengakses lebih dari satu layanan pada setiap harinya dengan menggunakan lebih dari satu *username* dan *password* [1]. Pada kasus ini instansi pendidikan seperti halnya Universitas

merupakan instansi yang memiliki lebih dari satu layanan *web service* untuk media mempermudah proses berkomunikasi juga *sharing* informasi demi kepentingan akademis. Layanan *web service* yang biasanya terdapat pada instansi pendidikan diantaranya Simak, *E-Learning* serta layanan berbasis *Blog System.* Dengan pertumbuhan jumlah layanan yang semakin banyak, tentunya sangat tidak efisien jika setiap masuk pada sistem administrasi suatu layanan, pengguna harus melakukan proses *login* dengan menggunakan lebih dari satu *username* dan *password* yang telah mereka miliki.

Salah satu metode manajemen *user* otentikasi yang dapat diimplementasikan pada kasus ini adalah LDAP. LDAP atau yang dikenal dengan *Lighweight Directory Access Protocol,* merupakan suatu metode manajemen *user* yang mirip dengan relasional *database* sebagai media untuk penyimpanan dan mengambil informasi [2]. LDAP dan relasional *databse* walaupun sekilas terlihat sama namun terdapat perbedaan dari kedua metode tersebut, dimana LPAP memiliki layanan query yang dapat dikatakan lebih cepat dibandingkan layanan dari relasional *database*. Hal ini yang membuat metode LDAP sangat terlihat mirip jika dilihat pada hirarki sebenarnya yang pada organisasi. Penggunaan metode ini dapat tentunya akan mampu menyelesaikan permasalahan penggunaan lebih dari satu *username* dan *password* serta memberikan kenyamanan lebih untuk para pengguna aplikasi berbasisa web [3].

Permasalahan yang masih timbul dari penggunaan metode ini adalah pengguna masih harus menginputkan *username* dan *password* ketika akan melakukan otentikasi pada setiap layanan aplikasi. Penggunaan LDAP tersebut masih memiliki kelemahan lain, kelemahan tersebut adalah memerlukan proses login berulang kali untuk setiap aplikasi jika *user* menginginkan masuk pada sistem admin layanan berbasis *web* tersebut. Dengan masih terdapatnya otentikasi yang dilakukan secara berulang *user* harus masih menginpukan *username* dan *password* pada setiap masing-masing aplikasi. Proses ini secara tidak langsung akan membuat *user/* pengguna merasa jenuh, karena harus melakukan proses otentikasi secara berulang-ulang terlebih jika *user* tersebut menggunakan lebih dari satu layanan aplikasi. Cara yang dapat digunakan untuk mengatasi masalah tersebut adalah mengembangkanlah suatu mekanisme otentikasi yang disebut *Single Sign On* (SSO).

SSO tersebut merupakan salah satu metode yang membuat pengguna hanya perlu melakukan sekali proses *login* atau otentikasi untuk dapat mendapat hak mengakses pada semua layanan yang dibutuhkan oleh pengguna. Hal ini dikarena pada hakikatnya proses *login/* otentikasi *Single Sign On* (SSO) hanyan menggunakan *credential user* untuk dapat *login* pada semua layanan aplikasi berbasis *web* setelah terlebih dahulu melakukan otentikasi pada salah satu/ sebuah aplikasi layanan berbasis web yang sudah terintegrasi dengan sistem *Single Sign On* (SSO). Pada peneletian ini untuk menerapkan sistem SSO akan digunakan *Central Authentication Service* (CAS) sebagai proses otentikasi terpusat pada aplikasi Simak, *E-Learning* dan *Blog System*. Dengan menerapkan CAS, maka semua layanan aplikasi yang sudah terintegrasi dalam sistem SSO tidak memerlukan lagi melakukan proses *login/*otentikasi dengan tersendiri, namun metode CAS yang melakukan manajemen *login/*otentikasi *user* disetiap layanan demi menghindari otentikasi yang berulang dan demi menjamin keamanan informasi pada prose otentikasi, dimana proses transmisi data yang terjadi sistem tersebut digunakan metode SSL protokol enkripsi [4]. Penggunaan metode *Single Sign On* (SSO) dengan menggunakan CAS ini, didasarkan bahwa CAS adalah metode SSO yang mendukung *library* untuk *client* pada aplikasi layanan berbasis *web*.

2. PENELITIAN PENDAHULU

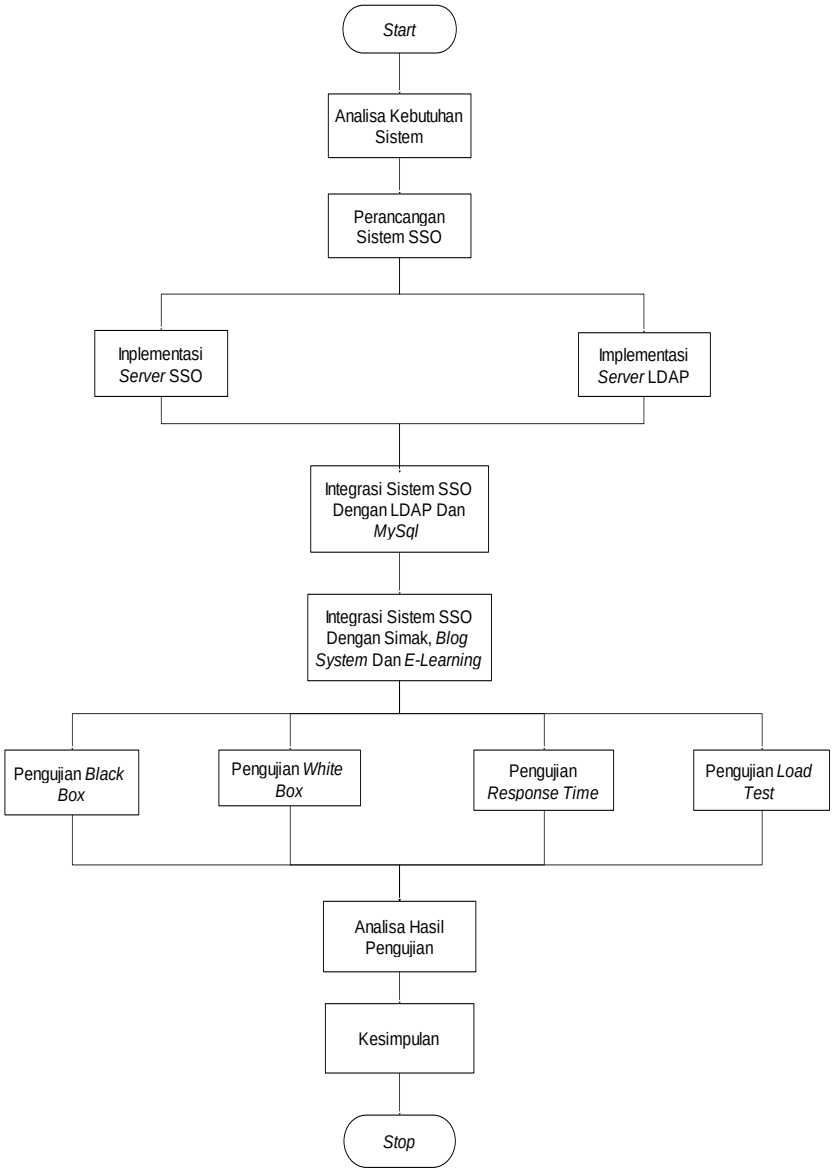
Pada proses penulisan penelitian ini terdapat beberapa penelitian-penelitian terdahulu yang mejadi latar belakang dan pertimbangan penulis untuk mengangkat pengembangan penelitian dengan topik SSO. Penelitian sebelumnya yang melatarbelakangi penulis tersebut mengangkat bagaimana proses penerapan sistem SSO dengan menggunakan CAS pada jaringan LDAP [5]. Hasil yang diperoleh dari penelitiannya adalah sistem SSO telah berhasil digunakan untuk halaman *login* terpusat bagi layanan berbasis *web*. Keberhasilan otentikasi akan ditentukan dari aktifitas *login* dan *logout* pada salah satu aplikasi. Ketika salah satu aplikasi sudah *login* atau *logout* maka secara otomatis aplikasi lain akan *login* atau *logout* dengan sendirinya[6].

Pada kasus penelitian yang penulis review dipaparkan bagaimana cara mengintegrasikan sistem SSO pada manajemen *user* LDAP. Pada penelitian tersebut sistem SSO diimplemetasikan dan diintegrasikan hanya pada manajemen *user* berbasis LDAP [3]. Hasil dari penelitian yang telah dilakukan tersebut adalah pengguna layanan terbantu dengan adanya sistem SSO memudahkan pengguna layanan sebab tidak perlu lagi menggunakan lebih dari satu *username* juga menghafal banyak *password* unuk melakukan *login* pada berbagai layanan aplikasi. Pengimplementasian sistem SSO tersebut membantu pada pengorganisasian pengguna sebab diimplementasikannya manajemen *user* LDAP sebagai metode *single* manajemen data pengguna layanan. Pada penelitian pendahulu tersebut sistem SSO dimplementasikan hanya menggunakan basis LDAP sehingga penguna harus tetap melakukan proses *login* berulang meskipun menggunakan *username* dan *password* yang sama. Pada penelitian yang dilakukan oleh penulis, sistem SSO akan diimplementasikan menggunakan metode otentikasi CAS dengan mengadopsi LDAP untuk sentralisasi manajemen *user*, dengan tujuan *user* tidak akan perlu lagi melakukan proses *login/* otentikasi secara berulang kali. Pada penelitian terdahulu berikutnya, yaitu penelitian pendahulu ketiga dipaparkan tentang implementasi *cross-domain* SSO terhadap *municipal portal*. Pada penelitian tersebut dihasilkan kesimpulan dengan penggunaan metode CAS, semua data ataupun informasi otentikasi pengguna tersebut disimpan pada *server* CAS, yang nantinya akan menghasilkan akses *cross-domain* pada layanan aplikasi yang terintegrasi dengan CAS [4]. Pada penelitian yang akan dilakukan oleh peneliti, peneliti akan melakukan pengimplentasian metode CAS terhadap *database* manajemen user yang berbeda, sesuai dengan review belum ada peneliti yang melakukan hal tersebut, sehingga akan menjadi novelti untuk implementasi sistem SSO.

3. PERANCANGAN SISTEM

Pada tahap ini, SSO yang dikembangkan untuk sistem informasi akademik memiliki beberapa proses untuk mengacu agar sistem yang dikembangkan dapat berjalan searah dengan harapan yang telah direncanakan. Dimana tahapan yang ada pada perancangan sistem SSO tersebut dapat dilihat pada Gambar 1.

Terlihat seperti pada Gambar 1, beberapa tahapan yang digunakan untuk membangun serta memastikas implementasi sistem single sign on yang telah dikembangkan telah dapat berjalan sesuai dengan rancangan. Metode yang digunakan penulis untuk mengembangkan sistem SSO adalah Central Authentication Service (CAS). CAS adalah salah satu metode authentication yang focus utamanya adalah mengutamakan keamanan informasi pengguna layanan [7]. Sesuai dengan konsep Central Authentication *Service*, yang fokus utamanya melakukan pengamanan tersebut terdapat dua aspek yang diperhatikan oleh CAS. Adapaun aspek tersebut adalah proses *authentication user* dan *accessing a protected web resource when authenticated*. Pada *fase authenticating user*, pada saat username dan password dari pengguna sesuai dengan data yang ada pada database, secara otomatis server CAS akan mengirim cookie dengan bentuk TGC *(Ticket Granting Cookie)* ke browser pengguna layanan. TGC dapat dikatakan sebagai paspor user terhadap server CAS. Batas waktu dari penggunaan TGC tersebut akan tergantung definisi saat proses implementasi pada server CAS. Fungsi dari TGC ini adalah untuk menghilangkan re-authenticate ketika pengguna mengakses aplikasi lain yang telah terintegrasi dengan sistem SSO. Untuk *accessing a protected web resource when authenticated*, pada saat user melakukan akses aplikasi yang terintegrasi dengan CAS, server CAS secara otomatis akan terlebih dahulu mengkonfirmasi paspor yang telah dimiliki oleh *browser* pengguna layanan. Pada saat TGC tersebut cocok, maka server sistem CAS memberikan ST (*Service Ticket*) kepada pengguna layanan. ST adalah *opaque ticket*, yang mana tiket tersebut tidak berguna menyimpan informasi dari pengguna dan hanya dapat digunakan untuk satu layangan saja, sehingga dapat mengoptimalkan keamanan identitas pengguna saat terjadinya proses *authentication*.



Gambar 1. Flowchart Analisa Dan Perancangan Sistem

ST yang telah dimiliki tersebut nantinya di validasi oleh CAS *Client* yang telah terpasang pada setiap aplikasi layanan yang terintegrasi, pada valid pengguna dapat mengakses aplikasi layanan yang telah disediakan oleh layanan terintegrasi dengan sistem SSO tersebut. Selain pemaparan tentang sistem CAS, terdapat beberapa proses atau tahapan utama yang digunakan menghasilkan sistem SSO yang mampu menghasilkan sistem sesuai dengan harapan instansi, berikut ini adalah pemaparan dari beberapa tahapan udari pembangunan sistem SSO tersebut :

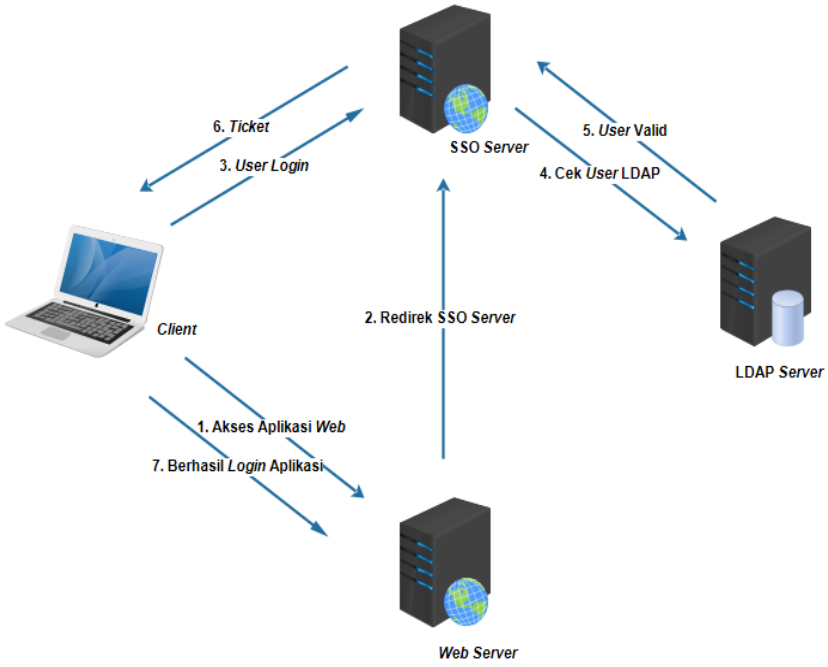
3.1 Analisis Kebutuhan

Belakangan ini instansi pendidikan pasti memiliki bebrapa layanan sistem informasi berbasis *web.* Untuk mengintegrasikan sistem SSO pada sistem informasi akademik tentunya diperlukan berbagai proses analisis terlebih dahulu, sehingga ketika diimplementasikan sistem SSO tersebut bisa terintegrasi dengan sesuai harapan pada sistem informasi yang sudah berjalan sebelumnya di isntansi pendidikan tersebut. Adapun fase proses analisis ini dibagi menjadi beberapa bagian diantaranya *baselining* dan *needs analysis*.

Pada proses analisis *baselining,* layanan aplikasi berbasis *web* yang telah siap diintegrasikan

dengan sistem SSO adalah Simak, *E-Learning* dan *Blog System*. Simak itu sendiri adalah sistem yang digunakan untuk mengolah data dan melakukan manjemen kegiatan akademik dengan melibatkan antara mahasiswa, dosen dan administrasi akademik. Aplikasi berbasis web yang kedua adalah *E-Learning,* aplikasi ini adalah media pembelajaran berbasis *web* yang digunakan dan diterapkan oleh instansi pendidikan untuk membuat mahasiswa dapat masuk atau bergabung ke dalam ruangan kelas digital. Pada ruangan kelas digital tersebut mahasiswa dapat mengakses materi, diskusi, quis, UTS dan UAS melalui aplikasi tersebut. Selain mahasiswa, pengajar juga dapat memanipulasi data materi ajar, soal dan tugas pada aplikasi *E-Learning* tersebut dan mahasiswa menentukan kursus yang telah disediakan sesuai dengan bidang minatnya masing-masing. *Blog System* adalah salah satu wadah yang digunakan sebagai media berbagi informasi yang dimilikinya oleh civitas akademik dalam hal berupa ilmu baru, penelitian amaupun informasi akademik yang dimiliki oleh pengguna *blog*. Dilihat dari sisi *needs analysis* semua aplikasi layanan web yang telah dipaparkan tadi memiliki proses *login* atau otentikasi dan juga memiliki basis data berbeda sebagai media manajemen ataupun penyimpanan *user*. Sebagi sarana untuk mempermudah *user* melakukan proses *login* atau otentikasi, proses otentikasi dari ketiga aplikasi pendukung proses akademik tersebut tersebut akan diintegrasikan menggunakan sistem SSO.

Perancangan gambaran Arsitektur Sistem Single Sign On SSO merupakan layanan yang memungkinkan pengguna untuk mengakses ke semua sistem aplikasi yang terintegrasi oleh sistem SSO dengan hanya melakukan satu kali proses *login* [8] [9] . Berikut adalah rancangan arsitektur dari sistem SSO yang diimplementasikan oleh penulis.



Gambar 2. *Arsitektur Sistem SSO Berbasis CAS*

Gambar 2 adalah ilustrasi umum dari sistem SSO yang dikembangkan oleh penulis, pada gambar 2 terlihat terdapat beberapa komponen utama yang membangun sistem tersebut, dimana komponen terdiri dari CAS *Server*, LDAP *Server* dan Aplikasi *Web Server*. CAS *Server* adalah komponen terpenting dari sistem ini yang merupakan tempat ditanamkannya sistem SSO dengan mengimplementasikan metode *Central Authentication Service* (CAS). Komponen berikutnya adalah LDAP *Server*, komponen ini adalah tempat menanamkan *database* pengguna dari semua layanan aplikasi berbasis

web yang terintegrasi dengan sistem SSO dan *Web Server* adalah tempat aplikasi berbasis *web* yang diintegrasikan pada sistem SSO. Adapaun alur proses kerja dari sistem SSO yang dibangun oleh penulis dapat dipaparkan sebagai berikut :

1. Pertama *Client* akan melakukan akses terhadap aplikasi berbasis *web;*
2. Pada saat *client* menekan proses *login*, aplikasi *web* akan melakukan *redirect browser* menuju SSO *Server*.
3. Proses selanjutnya *Client* akan memasukkan *username* dan *password* untuk otentikasi.
4. Langkah selanjutnya adalah *username* dan *password client* akan dicocokkan pada manajemen data LDAP.
5. Setelah melakukan proses pengecekan informasi data *client* akan dikirim kembali menuju SSO *Server.*
6. Langkah keenam adalah *client* melewati proses otentikasi dan kembali ke aplikasi berbasis *web* dengan sebuah tiket.
7. Proses terakhir adalah *client* yang sah dapat mengakses informasi yang terdapat pada aplikasi berbasis *web*.

3.2 Implementasi dan Konfigurasi

Untuk membangun sistem SSO, komponen utama yang harus diimplemntasikan dari arsitektur sistem SSO tersebut adalah CAS *Server*. CAS *Server* memiliki tugas untuk memberikan layanan *Single Sign On* terhadap semua layanan aplikasi berbasis *web* yang kedepannya akan terintegrasi dengan sistem SSO. Selain itu CAS *Server* tersebut memiliki tugas untuk meneruskan otentikasi ke halaman *web service* dan memberikan tiket atau paspor sebagai sebagai tanda bukti hak akses pengguna terhadap layanan aplikasi berbasis *web*. Untuk dapat melalukan implementasi sistem *Single Sign On* otentikasi pengguna atau *user* CAS akan menggunakan manajemen data LDAP. Langkah pertama yang harus disiapkan adalah *server* Ubuntu *Server* 12.04 LTS beserta *Apache2*, *Tomcat Java Server* dan aplikasi *Apache Maven* serta SSL. Lalu dilanjutkan dengan menanam aplikasi CAS *Server*.

Untuk membuat agar CAS *Server* otentikasinya menggunakan LDAP maka perlu melakukan editing pada berkas *deployerConfigContext.xml*, *cas-servlet.xml* dan *pom.xml*. Karena aplikasi *web* yang diintegrasikan berbasis PHP, maka perlu diinstall *apache web server* dan *php5*. *Apache web server* bertugas menerima dan membalas permintaan situs yang datang dari klien di *web server*. Php5 berfungsi mendukung bahasa pemrograman php pada *web server*.

Untuk membangun aplikasi berbasis *web* yang mendukung SSO perlu dilakukan modifikasi pada berkas yang berbeda pada masing–masing aplikasi *web*. Tombol *login* maupun *logout* pada aplikasi *web* diarahakan menuju halaman *login* dan *logout* CAS *Server*. Halaman pemrosesan *login* harus memuat fungsi GET tiket yang dikirimkan CAS *Server* dengan validasi *login* berupa *username*. Halaman *logout* tidak perlu memuat fungsi khusus selain fungsi pemusnahan *session* atau *cookie* lokal aplikasi *web*.

Untuk menentukan keberhasilan dari implementasi sistem SSO tersebut, tentunya diperlukan beberapa proses pengujian. Dalam penelitan ini terdapat dua jenis pengujian yang dilakukan oleh penulis, adapun jenis pengujian yang digunakan adalah pengujian *black box* dan pengujian *white box*. Untuk lebih jelasnya mengenai jenis pengujian tersebut akan dibahas lebih lengkap pada pembahasan berikut:

1. Pengujian *black box* yang digunakan dalam penelitian ini adalah pengujian *input* dan *output* dari sistem *Single Sign On*. Pengujian ini ditujukan untuk mengetahui kebenaran *output* dari perintah yang dimasukkan
2. Pengujian *white box* yang digunakan dalam penelitian ini adalah pengujian basis *path*. Notasi yang digunakan untuk menggambarkan jalur eksekusi adalah notasi diagram alir (grafik program), yang menggunakan notasi lingkaran (*node*) dan anak panah (*link*). Pengujian basis *path* pada penelitian ini difokuskan pada proses *login*.

4. HASIL DAN PEMBAHASAN

Hasil dari setiap jenis pengujian sistem SSO tentunya akan menghasilkan sebuah informasi yang sangat dibutuhkan untuk mengevaluasi sistem SSO tersebut. Adapun hasil dari setiap jenis pengujian yang telah dilakukan adalah :

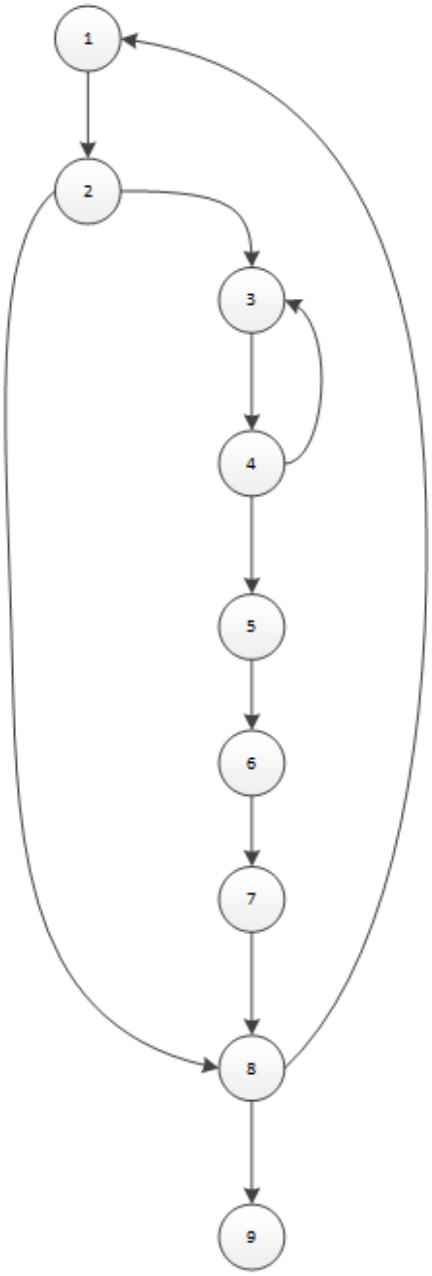
4.1 Pengujian Black Box

Dari hasil pengujian *black box* yang telah dilakukan dalam penelitian ini sistem SSO telah dapat memenuhi harapan *output* yang diharapkan pada saat melakukan pengujian. Dimana ketika *user* mengakses pada aplikasi Simak, *output* yang dihasilkan berupa sistem redirek menuju SSO *server* dan melakukan

proses *login* hingga terotentikasi. Setelah terotentikasi pada Simak*, user* selanjutnya mengakses aplikasi *E-Learning*, *output* yang dihasilkan berupa *user* secara otomatis terotentikasi pada aplikasi *E-Learning*. Hal ini juga berlaku pada *Blog System* setelah melakukan *login* pada salah satu aplikasi terintegrasi, maka *user* tidak perlu lagi melakukan proses *login* ketika mengakses aplikasi lainnya.

4.2 Pengujian *White Box*

Pengujian tersebut ditujukan untuk melihat *path-path* yang dilalui saat program dijalankan. Pengujian basis *path* pada penelitian ini difokuskan pada proses *login*. Gambar 3 adalah penggambarn *flowgraph* pada pengujian proses *login* :



Keterangan
1 : Pengaksesan aplikasi *web*
2 : Cek tiket
3 : Proses *login* SSO *server*
4 : Validasi *username* dan *password*
5 : Redirek aplikasi *web*
6 : Validasi tiket
7 : *Get username*
8 : Validasi *username* pada *database* aplikasi
9 : *Login* pada aplikasi *web*

Gambar 3. *Flowgraph Login Aplikasi*

V(G) = E – N + 2

V(G) = 11 – 9 + 2

V(G) = 4

Jalur pengujian :

1) Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 - 9
2) Jalur 2 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 - 1
3) Jalur 3 : 1 – 2 – 8 – 9
4) Jalur 4 : 1 – 2 – 8 – 1

Nilai V(G) yang didapat adalah 4, dimana terdapat 4 jalur pengujian yang didapat berdasar perhitungan dari *flowgraph* tersebut. Dari pengujian tersebut nilai *cyclomatic complexity* yang dimiliki adalah 4. Dimana *cyclomatic complexity* tersebut merupakan suatu nilai yang menentukan jumlah jalur dalam basis suatu program dan memberikan batas atas jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.

Dari hasil pengujian *white box* yang telah dilakukan, sistem SSO yang telah dibangun dapat diintegrasikan dengan aplikasi yang menggunakan basis data selain LDAP. Hal ini dilakukan dengan mengambil nilai tiket yang didapatkan setelah melakukan otentikasi pada *server* SSO. *Username* dari tiket yang telah dibuat pada *server* SSO tersebut akan diambil pada node 7 dan dilakukan pencocokan pada node 8, ketika *username* pada tiket sesuai dengan *username* pada aplikasi yang terintegrasi maka *user* tersebut dapat menikmati layanan yang disediakan pada aplikasi web tersebut.

5. KESIMPULAN

Berdasarkan pada hasil pengujian yang telah dilakukan oleh penulis, maka dapat diambil kesimpulan bahwa implementasi sistem *Single Sign On* (SSO) yang dikembangan dengan *Central Authentication Service* (CAS) dan dintegrasikan dengan layanan Simak, *E-Learning* dan *Blog System* telah mampu menangani otentikasi secara terpusat. Sesuai dengan latar belakang yang dipaparkan sebelumnya, sistem SSO ini juga telah mampu diintegrasikan pada aplikasi yang tidak menggunakan *Lighweight Directory Access Protocol* (LDAP) sebagai manajemen *user*.

DAFTAR PUSTAKA

[1] J. Wang, G. Wang, and W. Susilo, “Secure Single Sign-On Schemes Constructed from Nominative Signatures,” in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013, pp. 620–627.

[2] J. Hu, Q. Sun, and H. Chen, “Application of Single sign-on (SSO) in Digital Campus,” in *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, 2010, pp. 725–727.

[3] N. Dian, N. K. Yesi, and S. Rusmala, “Single Sign On (Sso) Dengan Menggunakan Lightweight Directory Access Protocol (Ldap) Studi Kasus Di Universitas Bina Darma,” *J. Mhs. TI S1*, no. JURNAL-TIS1-UBD-DIAN, 2013.

[4] H. Hu and Z. Guo, “The application of cross-domain single sign-on in municipal portal,” in *TENCON 2013 - 2013 IEEE Region 10 Conference (31194)*, 2013, pp. 1–4.

[5] K. I. S. Muhammad Yanuar Ary . S., “Implementasi Sistem Single Sign On / Single Sign Out Berbasis Central Authentication Service Protocol Pada Jaringan Lightweight Directory Access Protocol Universitas Diponegoro,” vol. 1, no. 3, 2012.

[6] B. Gueye, I. Niang, B. Gueye, M. O. Deye, and Y. Slimani, “Constraints-based response time for efficient QoS in Web Services Composition,” in *2011 7th International Conference on Next Generation Web Services Practices (NWeSP)*, 2011, pp. 141–146.

[7] I. P. A. E. D. Udayana and L. Jasa, “Implementasi Dan Analisis Single Sign On Pada Sistem Informasi Universitas Udayana”, Seminar Nasional Teknologi Informasi dan Multimedia, 2016.

[8] T. Yulin and Z. Feng, “The analysis and design for single sign-on in the mobile Application Data Center,” in *2012 3rd International Conference on System*

Science, Engineering Design and Manufacturing Informatization (ICSEM), 2012, vol. 2, pp. 258–260.

[9] Z. Jiang and A. Hassan, "A Survey on Load Testing of Large-Scale Software Systems," IEEE Trans. Softw. Eng., vol. PP, no. 99, pp. 1–1, 2015.