INSECT: Jurnal Teknik Informatika
INFORMATICS AND SECURITY e-ISSN. 2614-431X

Perancangan Topologi Sistem *Single Sign On* Berbasis CAS dan OpenLDAP

Sugiyatno*[1], Harun Noviar[2]
[1,2] Prodi Informatika , STMIK El Rahma Yogyakarta
E-mail: ***[1]sugiyatno@stmikelrahma.ac.id**, [2]harunnoviar@gmail.com

Abstrak

Metode otentikasi yang mewajibkan pengguna login ke setiap aplikasi dan dengan akun yang berbeda sering membuat lupa password. Masalah ini sering disebabkan oleh ketidakmampuan integrasi otentikasi di sebagian besar aplikasi, sehingga pengguna harus memasukkan username dan password berulang kali. Solusi atas masalah ini dalam penelitian ini adalah dengan merancang dan mengimplementasikan sistem Single Sign-On (SSO) berbasis CAS Apereo dan OpenLDAP di Debian 11 dengan virtualisasi. Sistem dikelola menggunakan aplikasi manajemen pengguna yang dikembangkan dengan PHP dan CodeIgniter 4. Pengujian dibagi menjadi dua bagian. Pertama, pengujian fitur pada dashboard administrator menunjukkan semua fungsi berjalan dengan baik. Kedua, pengujian fungsionalitas SSO menggunakan tiga aplikasi berbeda (CodeIgniter 4, Moodle, dan Drupal) membuktikan bahwa pengguna cukup login sekali, dan sesi autentikasi tetap berlaku pada aplikasi berikutnya. Dengan sistem SSO, waktu yang dibutuhkan oleh aplikasi kedua dan ketiga kurang dari 1,5 detik. Sedangkan waktu saat autentikasi manual pada aplikasi kedua dan ketiga kurang lebih 6-12 detik. Dari hasil pengujian diatas penggunaan SSO lebih cepat 62% dibandingkan tanpa SSO. Hasil penelitian adalah bahwa implementasi SSO ini efektif dalam menyederhanakan proses autentikasi lintas aplikasi, meningkatkan efisiensi, serta kemudahan akses bagi pengguna.

Kata kunci— *SSO, CAS, ldap, directory, codeigniter*

1. PENDAHULUAN

Semakin bertambah banyaknya layanan aplikasi berbasis *internet* saat ini menyebabkan beban yang lebih pada sisi pengguna. Hal ini dikarenakan pengguna diharuskan semakin banyak mengingat atau menghafal akun berupa *username* dan *password* untuk masuk ke layanan aplikasi tersebut. Sementara itu pada sisi layanan aplikasi, hal ini juga tidak aman dan efisien dengan perkembangan layanan aplikasi yang semakin bertambah [1]. Dengan banyaknya aplikasi yang akan diakses, setiap pengguna diharuskan *login* ke tiap-tiap aplikasi dengan cara memasukkan *username* dan *password* yang berbeda sehingga terkadang menghambat produktivitas suatu pekerjaan hanya dikarenakan pengguna lupa akan *username* dan *password* tersebut.

Kerta melakukan penelitian dengan judul Penggunaan *Single Sign On* (SSO) Pada Jaringan Internet Badan Pengkajian Dan Penerapan Teknologi (BPPT). Permasalahan yang dibahas dalam penelitian ini yaitu banyaknya aplikasi yang digunakan BPPT menyebabkan penggunaan autentikasi (*user* dan *password*) yang beragam dan mempersulit pengguna aplikasi. Penyelesainnya dengan cara merancang dan mengimplementasikan sistem *Single Sign On* untuk melakukan autentikasi pada semua aplikasi. Hasil dari penelitian ini adalah sistem *Single Sign On* menggunakan *Lightweight Directory Access Protocol* (LDAP) dan *Remote Authentication Dial-In User Service* (RADIUS) yang memberikan kemudahan bagi pengguna dalam proses autentikasi ke semua aplikasi[2].

Kresnanto melakukan penelitian dengan judul Sistem Penelusuran Sebaran Alumni Menggunakan PHP dan PostgreSQL. Permasalahan yang dibahas dalam penelitian ini yaitu pelacakan data sebaran alumni yang ada di Universitas Negeri Yogyakarta belum dilakukan secara tersistem. Penyelesaiannya dengan cara mengembangkan aplikasi web berbasis PHP dan PostgreSQL untuk mencatat dan mengelola data sebaran alumni di Universitas Negeri Yogyakarata. Hasil dari penelitian ini yaitu berkat penggunaan aplikasi web tersebut data sebaran alumni di Universitas Negeri Yogyakarta dapat dikelola dan dipantau dengan baik [3].

Prestyan melakukan penelitian dengan judul Aplikasi Penggajian Guru Dan Staff SITD Darrussunnah Menggunakan *Framework* Codeigniter 3. Permasalahan yang dibahas dalam penelitian ini yaitu mengenai sistem penggajian staf di SITD Darrussunnah yang masih dilakukan secara manual oleh bendahara. Penyelesainnya dengan cara mengembangkan aplikasi sistem penggajian menggunakan bahasa pemrograman PHP dan *framework* codeigniter 3. Hasil dari penelitian ini adalah STID Darrussunnah memiliki aplikasi sistem penggajian yang dapat menghitung dan mengelola gaji pegawai lebih cepat dan terdokumentasi dengan baik[4].

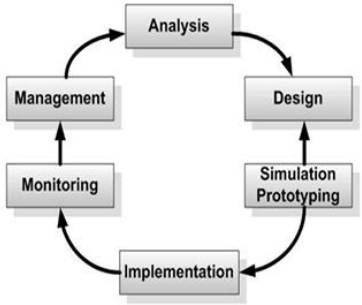
Putro melakukan penelitian dengan judul Implementasi SAML Pada Layanan *UIIGateway* Dengan Metode *Single Sign On* (SSO) Berbasis Web Portal (Studi Kasus: Badan Sistem Informasi UII). Permasalahan yang dibahas dalam penelitian ini yaitu dibutuhkannya sebuah sistem proses autentikasi dan otorisasi pengguna layanan aplikasi yang aman dan terpadu di UII. Penyelesainnya dengan cara mengimplementasikan sistem *Single Sign On* pada layanan sistem di BSI UII (*UIIGateway*). Hasil yang diperoleh dari penelitian ini adalah sistem *Single Sign On UIIGateway* yang mampu memudahkan pengguna dalam mengakses setiap layanan aplikasi[5].

Single Sign On (SSO) adalah proses mengakses ke banyak sumber daya aplikasi yang aman dan disediakan oleh sebuah organisasi dengan satu *login*. Pengguna tidak perlu lagi memasukkan nama pengguna (*username*), kata sandi (*password*), dan akun masuk lainnya untuk setiap aplikasi. Dengan satu autentikasi tersebut, pengguna dapat mengakses ke setiap aplikasi yang diijinkan [6].

Berdasarkan dari permasalahan tersebut maka dibuatlah sebuah rancangan topologi sistem *Single Sign On* yang bertujuan agar setiap layanan aplikasi dapat diakses dengan menggunakan satu akun *username* dan *password* saja. Pengguna layanan aplikasi tidak perlu direpotkan dengan menghafalkan satu persatu akun *login* untuk masing-masing aplikasi. Selain itu, dalam rancangan sistem *Single Sign On* ini terdapat aplikasi manajemen akun yang bertujuan untuk memudahkan *administrator* dalam mengelola dan mengkategorikan tiap-tiap akun yang ada.

2. METODE PENELITIAN

Metode penelitian digunakan agar langkah-langkah yang dilakukan dalam penelitian dapat terstruktur secara baik dan sesuai dengan permasalahan. Metode pengembangan yang digunakan pada penelitian ini yaitu Network Development Life Cycle yang memiliki skema dan tahapan kegiatan berikut ini.



Gambar 1 Kerangka Kerja *Network Development Life Cycle* [7]

Berdasarkan skema gambar 1 dapat dijelaskan langkah-langkah penelitian sebagai berikut ini.

1. Analisis Permasalahan

Tahap pertama dalam penelitian ini adalah melakukan analisis mendalam mengenai permasalahan yang dihadapi di STMIK EL RAHMA dalam penggunaan sistem informasi. Fokus utama pada tahap ini adalah mengidentifikasi masalah utama, mencari tahu penyebabnya, serta merumuskan solusi yang tepat. Selain itu, analisis juga mencakup pemeriksaan desain dan topologi jaringan yang ada, untuk memahami kekurangan dan area yang perlu diperbaiki dalam infrastruktur jaringan perusahaan.

2. Perancangan Solusi

Berdasarkan temuan dari tahap analisis, langkah selanjutnya adalah merancang solusi yang sesuai dengan kebutuhan STMIK EL RAHMA. Desain ini meliputi pengusulan topologi jaringan yang lebih optimal serta prosedur konfigurasi jaringan yang efektif. Selain itu, pada fase ini juga dilakukan perancangan sistem web portal yang akan digunakan sebagai media Single Sign-On (SSO) menggunakan bahasa pemrograman codeigniter.

3. Simulasi Prototipe

Sebelum implementasi langsung, dilakukan simulasi prototipe untuk memastikan bahwa desain dan konfigurasi yang telah direncanakan berjalan dengan baik. Tujuan dari simulasi adalah untuk memverifikasi kinerja jaringan dan mengidentifikasi potensi masalah sebelum diterapkan di dunia nyata, sehingga memungkinkan untuk melakukan perbaikan lebih awal dan meminimalkan kesalahan yang dapat terjadi pada tahap implementasi.

4. Implementasi Jaringan

Setelah prototipe diuji dan hasilnya sesuai dengan harapan, tahap berikutnya adalah implementasi jaringan. Pada fase ini, topologi jaringan yang telah dirancang dan prosedur konfigurasi yang telah disimulasikan diterapkan langsung pada perangkat jaringan. Dari tahap implementasi ini dapat diketahui unjuk kerja dan perbedaan ketika menggunakan SSO dan tidak menggunakan SSO.

5. Pemantauan dan Pemeliharaan

Setelah implementasi selesai, jaringan yang baru diatur perlu dipantau secara terus-menerus untuk memastikan performa dan kestabilannya tetap terjaga. Pemantauan dilakukan untuk mengidentifikasi potensi masalah yang mungkin muncul dan melakukan perbaikan atau penyesuaian jika diperlukan.

2.1 Single Sign On (SSO)

Single Sign On biasa disingkat dengan SSO adalah proses mengakses ke banyak sumber daya aplikasi yang aman dan disediakan oleh sebuah organisasi dengan satu login. Pengguna tidak perlu lagi memasukkan nama pengguna (username), kata sandi (password), dan akun masuk lainnya untuk setiap aplikasi. Dengan satu autentikasi tersebut, pengguna dapat mengakses ke setiap aplikasi yang diijinkan [6].

Terdapat beberapa tipe arsitektur SSO, dengan properti dan infrastruktur yang berbeda. Tipe-tipe arsitektur SSO tersebut yaitu Secure Client-Side Credential Caching, Secure Server-Side Credential Caching, SSO dengan Single Set Credentials, SSO berbasis Public Key Infrastructure, dan SSO berbasis Token[8].

Teknologi Single sign on (sering disingkat menjadi SSO) adalah teknologi yang mengizinkan pengguna jaringan agar dapat mengakses sumber daya dalam jaringan hanya dengan menggunakan satu akun pengguna saja [9]. Dengan menggunakan SSO, seorang pengguna hanya cukup melakukan proses autentikasi sekali saja untuk mendapatkan izin akses terhadap

semua layanan yang terdapat didalam jaringan.Produk-produk sistem SSO yang berbasis open sourceyang umum digunakan saat ini seperti CAS (CentralAuthentication Service), OpenAM (Open Access Manager), dan JOSSO (Java Open SingleSign-On) [10]

Menurut Techtarget, *Single Sign On* adalah sebuah layanan autentikasi sesi dan pengguna yang memungkinkan pengguna menggunakan satu jenis akun login contohnya username dan password untuk mengakses beberapa sistem atau aplikasi. SSO dapat digunakan oleh berbagai kalangan mulai dari individu hingga perusahaan kecil dan besar untuk memudahkan manajemen dari akun yang beragam [11].

2.2 Cara Kerja Single Sign On

Konsep dasar *Single Sign On* adalah mengurangi kerumitan yang biasanya terjadi ketika pengguna harus mengelola atau mengingat beberapa autentikasi akun yang dimiliki untuk mengakses berbagai aplikasi dan layanan. Dalam mengaplikasikan SSO diperlukan *username* dan *password* yang disimpan pada sebuah *server* aplikasi *Directory Service* misalnya seperti *Windows Active Directory*, *Novell eDirectory*, *Netscape Directory*, dan *Open Lightweight Directory Access Protocol* (OpenLDAP). Semua aplikasi *Directory Service* menggunakan protokol standard *Lightweight Directory Access Protocol* (LDAP) untuk mengakses dan mengelola informasi seperti data yang tersimpan di dalamnya [12].
SSO pada umumnya diimplementasikan menggunakan berbagai protokol autentikasi antara lain sebagai berikut [13].

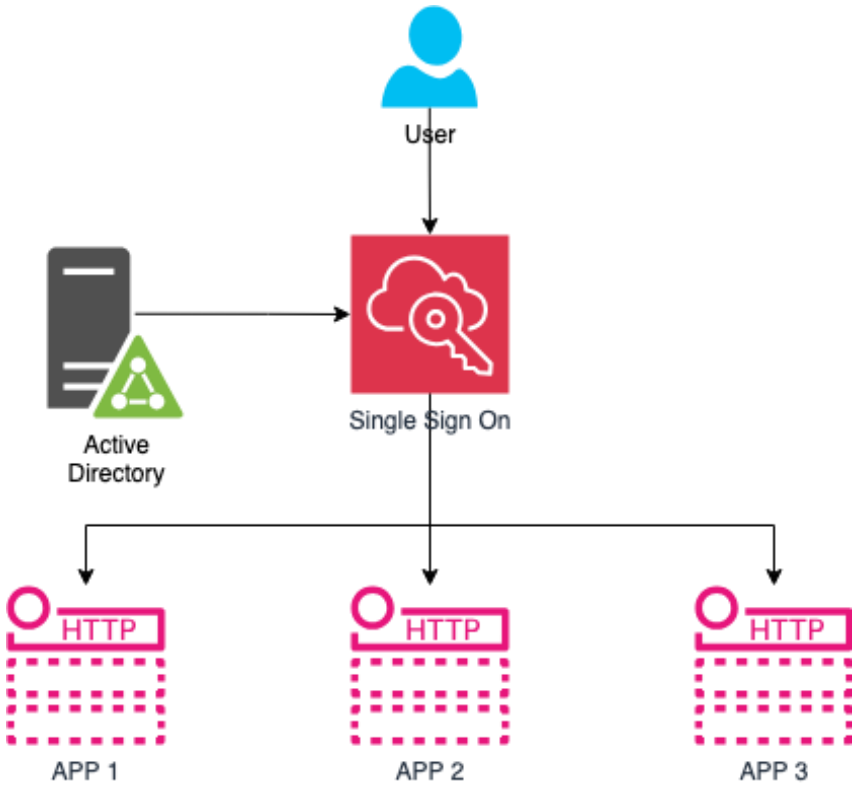
a. *Security Assertion Markup Languange* (SAML)
   SAML merupakan protokol atau kumpulan aturan yang digunakan aplikasi untuk bertukar informasi autentikasi dengan layanan SSO menggunakan bahasa markup XML.
b. *Open Authorization* (Oauth)
   Oauth merupakan standar terbuka yang memungkinkan aplikasi untuk secara aman mengakses informasi pengguna dari situs web lain tanpa memberinya kata sandi. Aplikasi menggunakan Oauth untuk mendapatkan izin pengguna agar dapat mengakses data yang dilindungi kata sandi. Oauth membangun kepercayaan antar aplikasi melalui API, yang memungkinkan aplikasi mengirim dan merespons permintaan autentikasi dalam kerangka kerja yang telah ditetapkan.
c. *OpenID Connect* (OIDC)
   OIDC merupakan kombinasi antara Oauth dan SAML yang digunakan untuk autentikasi pengguna. OIDC memungkinkan pengguna untuk masuk ke berbagai aplikasi dengan satu set *username* dan *password* melalui protokol Oauth.
d. Kerberos
   Kerberos merupakan sistem autentikasi berbasis tiket yang memungkinkan dua atau lebih pihak saling memverifikasi identitas mereka di jaringan. Kerberos menggunakan kriptografi keamanan untuk mencegah akses tidak sah ke informasi identifikasi yang dikirimkan di antara *server*, *client*, dan *Key Distribution Center*.
e. *Central Authentication Service* (CAS)
   CAS merupakan protokol SSO *Open Source* yang digunakan untuk mengijinkan pengguna mengakses beberapa aplikasi layanan dengan satu kali memasukkan akunnya. Protokol CAS mengimplementasikan autentikasi tunggal berbasis token.

Adapun cara kerja dari *Single Sign On* adalah sebagai berikut [14].

a. Pengguna melakukan autentikasi dengan memasukkan akun berupa *username* dan *password* ke aplikasi yang mendukung sistem SSO. Aplikasi yang mendukung sistem SSO sendiri disebut sebagai *Service Provider* (SP).
b. *Service Provider* kemudian melakukan pengecekan *username* dan *password* ke server SSO yang menggunakan LDAP sebagai *Identity Provider* (IdP).
c. Apabila autentikasi berhasil, IdP mengembalikan token autentikasi atau sesi ke aplikasi SSO/ *Service Provider*.

d. *Service Provider* kemudian menggunakan token autentikasi untuk mengizinkan pengguna mengakses aplikasi atau layanan lain yang terintegrasi dengan SSO tanpa perlu memasukkan *username* dan *password* lagi.

Diagram dari Single Sign On dapat dilihat dalam gambar 2.



Gambar *2* Diagram *Sistem Single Sign On*

2.3 Directory Service

Directory adalah *database* khusus yang dirancang secara spesifik untuk pencarian dan penelusuran dengan tambahan dukungan pencarian dasar (*basic lookup*) dan fungsi pembaruan (*update function*). *Directory* cenderung berisi informasi deskriptif berbasis atribut, dan mendukung kemampuan penyaringan data secara canggih. Pada umumnya, *Directory* tidak mendukung transaksi yang rumit atau skema pemulihan seperti pada sistem manajemen basis data yang dirancang untuk menangani pembaruan kompleks dan besar.

Directory Service adalah *database* untuk menyimpan dan memelihara informasi tentang pengguna dan sumber daya lainnya. *Directory Service* biasa dikenal juga sebagai direktori, penyimpanan pengguna, penyimpanan indentitas, direktori LDAP, yang menyimpan informasi seperti nama pengguna, kata sandi, preferensi pengguna, informasi perangkat, dan banyak lagi[15].

Active Directory (AD) adalah implementasi LDAP directory services yang digunakan dalam lingkungan Microsoft. AD sendiri sebenarnya merupakan suatu database terdistibusi yang dapat mereplikasi ke semua Domain Controller (DC) pada suatu jaringan. Isi dari database tersebut menyimpan informasi berupa: User, Computer, Group, Policy, Aplikasi, Printer, Object directory lain[16].

2.4 OpenLDAP

OpenLDAP adalah sebuah *software* bersifat *open source* yang mengimplementasikan protokol *Lightweight Directory Access Protocol* dan tersedia pada sistem operasi berbasis linux. Di dalam OpenLDAP sendiri terdiri dari dua *service* utama yaitu "*slapd"* dan "*lloadd"*. *Slapd* merupakan OpenLDAP *daemon service* yang memiliki fungsi melayani permintaan *query* dari klien dan berkomunikasi dengan *backend database*. Sedangkan *lloadd* adalah *daemon service* yang menyediakan penyeimbang beban (*load balancer*) LDAPv3 dan bertanggung jawab mendistribusikan permintaan ke serangkaian *slapd instance* [17]

Lightweight Directory Access Protocol (LDAP), yang dikembangkan oleh University of Michigan, merupakan metode akses direktori yang diterima secara luas [6] dan merupakan standar industri terbuka. LDAP merupakan basis data khusus, yang datanya diatur dalam direktori, dan direktori tersebut terdiri dari objek-objek, yang memiliki informasi atribut. Atribut direktori pada dasarnya adalah pasangan kunci-nilai, yang merupakan cara untuk menyimpan data di direktori. LDAP menulis data secara lambat, dan operasi modifikasi hanya diimplementasikan menggunakan mekanisme penguncian sederhana[18].

LDAP juga sebagai sebuah protokol yang mengatur mekanisme dalam mengkases layanan direktori (Directory Service) yang bisa digunakan untuk mendeskripsikan banyak informasi layaknya informasi tentang orang-orang, organisasi, aturan, layanan dan banyak entitas lainnya [19].

2.5 CAS Apereo

CAS Apereo (*Central Authentication Service*) adalah sebuah *framework* autentikasi *Single Sign On open source* berbasis web yang dirancang untuk mengizinkan pengguna mengakses beberapa aplikasi dengan hanya sekali memasukkan akunnya seperti *username* dan *password*. Dengan protokol CAS memungkinkan aplikasi web mengautentikasi pengguna tanpa perlu mengakses langsung akun pengguna seperti *password*. Penggunaan istilah CAS juga mengacu pada paket *software* yang mengimplementasikan protokol ini.

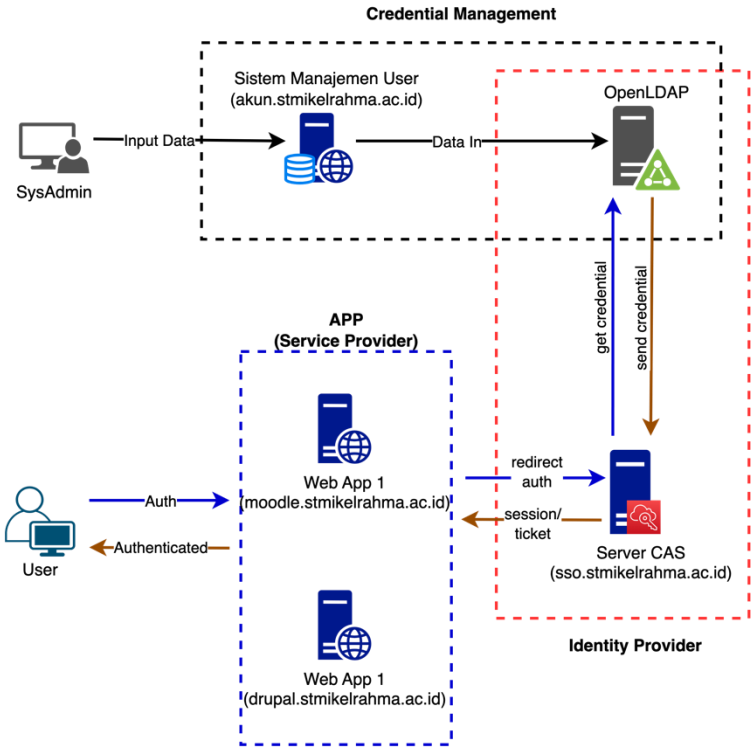
CAS Apereo adalah sebuah paket *software* solusi *single sign on* multi bahasa dan penyedia identitas berbasis web yang berupaya menjadi platform komprehensif untuk kebutuhan autentikasi dan otorisasi. Implementasi utama dari CAS Apereo adalah komponen *open source java server* dengan dukungan sejumlah protokol dan fitur tambahan di dalamnya.
Fitur dan teknologi yang teradapat pada CAS Apereo antara lain sebagai berikut [20].

a. Komponen server menggunakan Java Spring Webflow/ Spring Boot.
b. Bermacam-macam pilihan dukungan autentikasi (LDAP, Database, X.509, SPNEGO, JAAS, JWT, RADIUS, MongoDb).
c. Mendukung banyak protokol (CAS, SAML v1, SAML v2, WS-Federation, Oauth2, OpenID, OpenID Connect, REST).
d. Mendukung autentikasi multifaktor melalui beberapa penyedia (Duo Security, FIDO U2F, YubiKey, FIDO2 WebAuthN, Google Authenticator, Authy, Acceptto, Inwebo).
e. Mendukung autentikasi ke penyedia identitas eksternal seperti ADFS, Facebook, Twitter, SAML2 IdPs, OIDC Ops.
f. Mendukung fitur bawaan berupa manajemen kata sandi, notifikasi, dan ketentuan penggunaan.
g. Mendukung rilis atribut termasuk izin penggunaan.
h. Memantau dan melacak kondisi aplikasi dan sistem, statistik, dan matrik secara *real-time*.
i. Mengelola dan memantau *log* secara terpusat, serta mempublikasikan data ke berbagai layanan *downstream* lainnya.
j. Mendukung klien lintas platform (Java, .NET, PHP, Perl, Apache, dsb).
k. Mendukung integrasi dengan InCommon, Box, Office365, ServiceNow, Salesforce, Workday, WebAdvisor, Drupal, Blackboard, Moodle, Google Apps, SCIM, reCAPTCHA, Swagger.

3. HASIL DAN PEMBAHASAN

3.1 Rancangan Topologi

Rancangan Topologi adalah rancangan desain sistem topologi yang akan dikembangkan dengan menggunakan SSO, secara garis besar desain topologi dapat dilihat pada gambar 3.



Gambar 3 Rancangan Topologi Sistem SSO

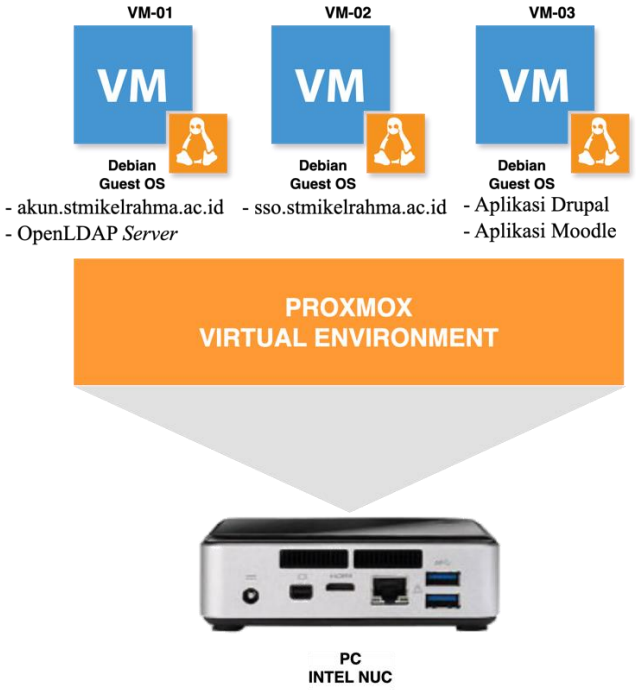
Dari gambar 3 terdapat lima komponen penyusun sistem yaitu *SysAdmin, Credential Management, User, Service Provider,* dan *Identity Provider.* Dari kelima komponen tersebut masing-masing memiliki definisi dan tugas sebagai berikut.

a. *SysAdmin* merupakan orang atau petugas yang diberi wewenang melakukan manajemen akun *username* dan *password* pengguna. Pengelolaannya meliputi pembuatan akun baru ataupun pembaruan akun yang sudah ada. Proses pembuatan akun dapat dilakukan dengan dua cara yaitu memasukkan data secara manual satu persatu atau dengan melakukan impor data dari *template* csv.
b. *Credential Management* merupakan aplikasi yang berfungsi sebagai manajemen akun *username* dan *password* pengguna, terdiri dari dua sub komponen yaitu aplikasi manajemen dan tempat penyimpanan data pengguna (*user datastore*). Aplikasi manajemen pengguna dikembangkan dengan menggunakan bahasa pemrograman PHP, *framework* Codeigniter 4, PostgreSQL *database*, dan *webserver Nginx*. Sedangkan untuk *user datastore* dikembangkan menggunakan *software* OpenLDAP.
c. *User* merupakan pengguna yang melakukan autentikasi ke aplikasi-aplikasi yang mendukung sistem SSO (*Service Provider - SP*).
d. *Service Provider* (SP) merupakan aplikasi-aplikasi yang mendukung sistem autentikasi dengan metode SSO. Dari SP inilah pengguna pertama kali melakukan autentikasi

dengan memasukkan *username* dan *password* yang dimilikinya. Kemudian SP akan melakukan *redirect* ke halaman *Identity Provider*.

e. *Identity Provider* (IdP) merupakan aplikasi yang bertugas menyimpan dan melakukan pengecekan kesesuaian *username* dan *password*. Dari IdP ini apabila autentikasi *username* dan *password* berhasil dan sesuai maka akan dikembalikan dengan nilai *session* dan *ticket* ke SP, sehingga *user* memiliki status telah terautentikasi.

3.2 Rancangan Infrastruktur Server



Gambar 4 Rancangan Infrastruktur Server

Sistem *Single Sign On* akan dirancang dan dibangun pada infrastruktur *hardware* menggunakan PC Desktop Intel NUC D54250WYK Core i5-4250U dengan desain seperti terlihat pada gambar 4. PC Intel NUC tersebut di dalamnya di-*install*-kan *software* virtualisasi *Proxmox Virtual Environment* (PVE) versi 6.4.4. Di atas virtualisasi PVE dibuat tiga *virtual machine* (VM) dengan *guest* OS-nya menggunakan Debian 11. Ketiga VM tersebut memiliki spesifikasi *software* dan fungsinya masing-masing sebagai berikut.

a. **VM-01** difungsikan sebagai *host* dari aplikasi Sistem Manajemen Pengguna (akun.stmikelrahma.ac.id) dan *datastore* berupa aplikasi OpenLDAP *Server*. Aplikasi sistem manajemen *user* sendiri membutuhkan *webserver Nginx* dan *database* PostgreSQL *server*. OpenLDAP *server* menjadi bagian dari *Identity provider* (IdP) yang terhubung dengan Aplikasi *CAS Apereo*.
b. **VM-02** difungsikan sebagai *host* dari aplikasi CAS *Apereo Server* (sso.stmikelrahma.ac.id) yang terkoneksi ke OpenLDAP *server* dan bersama-sama berjalan sebagai *Identity Provider* (IdP).
c. **VM-03** difungsikan sebagai *host* dari dua aplikasi yang bertugas sebagai *Service Provider* (SP). Aplikasi tersebut diantaranya adalah LMS berbasis Moodle dan portal *web* berbasis CMS Drupal. Fungsi dari kedua aplikasi tersebut yaitu sebagai paramater aplikasi yang diuji

menggunakan sistem autentikasi SSO. Di dalam VM ini juga disematkan aplikasi *webserver Nginx* dan *database* MySQL.

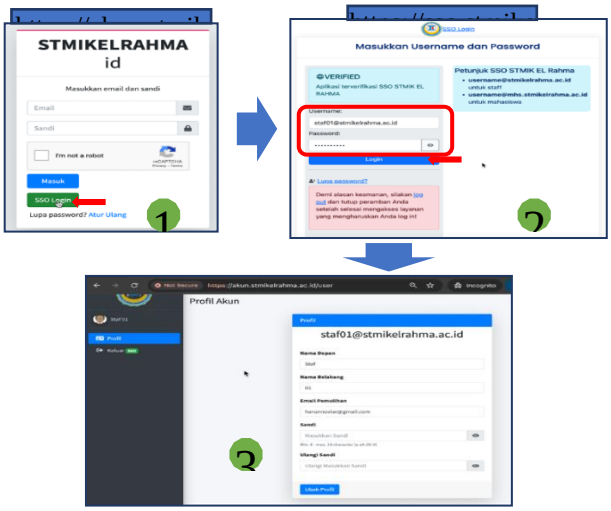
3.3 Pengujian Aplikasi SSO

Pengujian sistem merupakan fase krusial dalam melakukan pengembangan sistem SSO dengan tujuan untuk memastikan aplikasi atau sistem yang dibangun ini dapat berjalan dan berfungsi sesuai spesifikasi yang telah ditentukan. Pengujian sistem SSO ini meliputi dua aspek yaitu pengujian fungsionalitas sistem SSO dan pengujian perbandingan dengan tanpa SSO. Adapun secara lebih lengkap tahapan pengujian tersebut dilakukan dengan langkah-langkah sebagai berikut.

1. Pengujian Fungsionalitas Sistem SSO

Pengujian fungsionalitas sistem SSO menggunakan tiga paramater aplikasi yang berbeda, yaitu dari aplikasi "akun.stmikelrahma.ac.id", "moodle.stmikelrahma.ac.id", dan "drupal.stmikelrahma.ac.id". Dari ketiga *domain* tersebut masing-masing mewakili aplikasi yang memiliki karakteristik ataupun *framework coding* berbeda yaitu Codeigniter, LMS Moodle, dan CMS Drupal. Hal ini sekaligus untuk membuktikan bahwa sistem SSO yang dibangun memang mendukung jenis aplikasi dengan berbagai macam bahasa pemrograman ataupun *framework* yang digunakan.

Pertama kali pengguna mengakses halaman aplikasi https://akun.stmikelrahma.ac.id sebagai salah satu *Service Provider* (SP) kemudian memilih tombol "**SSO Login**". Selanjutnya pengguna akan diarahkan ke halaman *Identity Provider* (IdP) yaitu https://sso.stmikelrahma.ac.id dan memasukkan *username* dan *password* yang sebelumnya sudah dibuatkan dari halaman *administrator* aplikasi manajemen akun. Setelah berhasil melakukan autentikasi SSO maka pengguna akan diarahkan kembali ke halaman "Profil" pada aplikasi "akun". Proses yang dilakukan dalam autentikasi pengguna pada aplikasi "akun.stmikelrahma.ac.id". Proses awal autentikasi tersebut hanya membutuhkan tiga langkah utama saja seperti terlihat dalam Gambar 5.



Gambar 5 Proses Autentikasi Pada Aplikasi "akun"

Pengujian kedua yaitu dengan melakukan proses autentikasi aplikasi moodle melalui alamat https://moodle.stmikelrahma.ac.id/login/index.php. Pengguna memilih tombol "Login SSO" selanjutnya akan langsung diarahkan ke halaman profil pengguna di aplikasi moodle. Pengguna tidak perlu lagi melewati tahapan memasukkan *username* dan *password* pada IdP di halaman https://sso.stmikelrahma.ac.id dikarenakan sistem SSO mendeteksi *username* yang

bersangkutan memiliki status *session* masih berlaku, sehingga IdP tinggal mengirimkan kembali *Service Ticket* (ST) ke aplikasi SP tersebut.

Pengujian ketiga yaitu dengan mengakses CMS drupal pada alamat https://drupal.stmikelrahma.ac.id/user/login. Saat pengguna melakukan klik pada menu "**CAS Login**", sebenarnya pengguna juga diarahkan ke halaman https://sso.stmikelrahma.ac.id, akan tetapi proses ini dilakukan secara background. Proses yang terlihat adalah pengguna sama sekali tidak perlu memasukkan kembali *username password* dan langsung ditampilkan halaman "drupal" dengan status berhasil login ("**CAS Sukses Login**"). Proses tersebut sama dengan saat pengguna melakukan autentikasi pada aplikasi "moodle". IdP mendeteksi kalau *username* tersebut masih memiliki *session* sehingga mengirimkan informasi balikan berupa *Service Ticket* ke aplikasi "drupal".

2. Pengujian Unjuk Kerja Sistem

Tahapan pengujian selanjutnya ini adalah membandingkan beberapa parameter aplikasi-aplikasi untuk pengujian SSO sebelumnya tanpa menggunakan atau mengintegrasikan sistem tersebut dengan SSO. Pengujian ini menggunakan kredensial *login* masing-masing yang sudah dipersiapkan sebelumnya pada aplikasi tersebut.

Tabel 1 Perbandingan Waktu Autentikasi SSO dan Tanpa SSO

Akun	Codeigniter		Moodle		Drupal		Rata-Rata
	Waktu (detik)						
	SSO	Non SSO	SSO	Non SSO	SSO	Non SSO	
staf01@stmikelrahma.ac.id	13,86	15,45	1,46	13,7	1,2	6,72	
staf02@stmikelrahma.ac.id	13,26	12,69	1,39	12,59	1,02	6,31	
staf03@stmikelrahma.ac.id	11,07	12,01	1,66	11,14	0,82	6,57	
staf04@stmikelrahma.ac.id	12,45	12,89	1,54	12,23	0,87	6,54	
staf05@stmikelrahma.ac.id	11,02	13,26	1,33	11,59	0,67	6,51	
mhs01@mhs.stmikelrahma.ac.id	13,01	15,26	1,53	12,15	0,6	6,3	
mhs02@mhs.stmikelrahma.ac.id	13,45	15,12	1,34	13,43	0,79	6,75	
mhs03@mhs.stmikelrahma.ac.id	13,22	14,96	1,62	12,85	0,71	6,31	
mhs04@mhs.stmikelrahma.ac.id	13,69	15,44	1,58	13,37	0,63	6,8	
mhs05@mhs.stmikelrahma.ac.id	14,03	15,22	1,54	12,46	0,68	6.83	
Rata-rata	12,91	14,23	1,5	12,55	0,8	6,57	
Selisih Waktu (detik)		**1,32**		**11,05**		**5,77**	**6,05**
Prosentase Waktu		**9%**		**88%**		**88%**	**62%**

Tabel 1 di atas merupakan hasil ujicoba perbandingan waktu (dalam detik) yang dibutuhkan oleh ketiga aplikasi (codeigniter, moodle, drupal) dalam memproses autentikasi menggunakan sistem SSO login dan manual login (tanpa SSO). Ujicoba ini dilakukan dengan parameter pada kondisi ideal, di mana timer dihitung mulai dari pengguna menekan tombol "SSO Login" pada saat autentikasi SSO dan pengguna mengisi kredensial login pada saat autentikasi manual. Dari tabel hasil ujicoba di atas dapat terlihat bahwa saat aplikasi pertama melakukan autentikasi baik SSO atau manual, waktu yang diperlukan kurang lebih 12-14 detik. Perbedaan waktu antara SSO dan manual terlihat signifikan saat autentikasi dilakukan pada aplikasi kedua maupun ketiga. Dengan sistem SSO, waktu yang dibutuhkan oleh aplikasi kedua dan ketiga kurang dari 1,5 detik. Sedangkan waktu saat autentikasi manual pada aplikasi kedua dan ketiga kurang lebih 6-12 detik. Dari hasil pengujian diatas penggunaan SSO lebih cepat 62% dibandingkan tanpa SSO.

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem Single Sign-On (SSO). Sistem SSO yang telah dikembangkan memberikan kemudahan bagi pengguna dalam

melakukan autentikasi ke berbagai aplikasi yang mendukung sistem ini. Dengan hanya menghafalkan satu akun berupa username dan password, pengguna tidak perlu lagi memasukkan kredensialnya secara berulang untuk mengakses layanan yang terintegrasi. Hal ini tidak hanya meningkatkan kenyamanan pengguna tetapi juga mengurangi risiko lupa password serta menghemat waktu dalam proses login.

Dari sisi pengembangan, sistem ini memberikan manfaat bagi programmer aplikasi dengan menyederhanakan manajemen akun. Dengan tidak perlu menyimpan data akun di setiap aplikasi, potensi kerentanan keamanan dapat diminimalisir, sehingga sistem menjadi lebih aman dan efisien. Selain itu, aplikasi manajemen pengguna yang dibangun memiliki fitur menu yang berjalan sesuai dengan tujuan perancangan, memungkinkan administrator untuk mengelola akun dengan lebih mudah dan tanpa kendala.

Secara keseluruhan, implementasi sistem SSO ini berhasil meningkatkan efisiensi autentikasi, keamanan data, serta kemudahan pengelolaan akun bagi pengguna, pengembang, dan administrator di lingkungan STMIK El Rahma Yogyakarta. Dengan sistem SSO, waktu yang dibutuhkan untuk proses autentikasi lebih cepat 62% dibandingkan tanpa SSO.

5. SARAN

Dalam melakukan pengujian, penulis mengajukan saran yang dapat dikembangkan dari penelitian ini antara lain sebagai berikut.

1. Server CAS Apereo dapat diimplementasikan menggunakan *load balancer* sehingga dapat memiliki *high availabilty* yang handal untuk meminimalisir apabila server fisik ataupun VM mengalami *down*.
2. Aplikasi manajemen pengguna dapat ditambahkan fitur yang menampilkan status *history log* dari masing-masing akun saat proses autentikasi dengan server CAS.
3. Melakukan integrasi sistem SSO pada aplikasi dengan bahasa pemrograman ataupun *framework* lain semisal *frontend* nextjs dan *backend* laravel yang memudahkan pemrograman karena didukung paket modul-modul lebih lengkap.

DAFTAR PUSTAKA

[1] J. Wang, G. Wang, and W. Susilo, “Secure single sign-on schemes constructed from nominative signatures,” 2013. [Online]. Available: https://ro.uow.edu.au/eispapers

[2] J. M. Kerta, P. Adiprabowo, E. Kusmiyati, and S. A. W. Rahardjo, “Penggunaan Single Sign On (SSO) pada Jaringan Internet Badan Pengkajian dan Penerapan Teknologi (BPPT),” *ComTech: Computer, Mathematics and Engineering Applications*, vol. 2, no. 2, pp. 880–886, 2011.

[3] K. Agus, “Sistem Penelusuran Sebaran Alumni Menggunakan Php Dan Postgresql,” *naskah publikasi KP*, 2022.

[4] D. W. I. F. PRESTYAN, “Aplikasi Penggajian Guru Dan Staff Sitd Darrussunnah Menggunakan Framework Codeigniter 3,” *Naskah Publikasi KP*, 2022.

[5] I. A. N. Fathony, “Kontainerisasi Shibboleth Idp Sebagai Akses Manajemen Single Sign On Pada Arsitektur Microservice Sistem Enterprise Dengan Metode Load Balancing,” 2022.

[6] A. Dey and S. Suriya, “Single Sign on,” 2016. [Online]. Available: http://www.ServiceProvider.com/MyApp?ticket....

[7] A. Kurnianto, D. H. Sulaksono, and A. Rachman, “Penerapan Single Sign on (Sso) Pada Keamanan Jaringan Dengan Metode Lightweight Directory Access Protocol (Ldap) Di Pt Unichem,” *KERNEL: Jurnal Riset Inovasi Bidang Informatika dan Pendidikan Informatika*, vol. 3, no. 1, pp. 20–26, 2022.

[8] I. K. D. Senapartha, “Implementasi Single Sign-On Menggunakan Google Identity, REST dan OAuth 2.0 Berbasis Scrum,” *Jurnal Teknik Informatika Dan Sistem Informasi*, vol. 7, no. 2, pp. 307–320, 2021.

[9] A. H. Kurniawan, “Konsep Altmetrics Dalam Mengukur Faktor Dampak Artikel Melalui Academic Social Media Dan Non-Academic Social Media,” *UNILIB: Jurnal Perpustakaan*, pp. 43–49, 2020.

[10] M. G. An'ars and A. Kurniawan, "Sistem Informasi Manajemen Berbasis Key Performance Indicator (KPI) dalam Mengukur Kinerja Guru," *Jurnal Data Mining Dan Sistem Informasi*, vol. 3, no. 1, pp. 8–18, 2022.
[11] Techtarget.com, "Single Sign On." Accessed: Oct. 17, 2023. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/single-sign-on
[12] E. Prasetyo, "Otentikasi Web Terpusat Menggunakan Metode Single Sign On Berbasiskan Lightweight Directory Access Protocol (LDAP)," 2011.
[13] Amazon.com, "What is SSO (Single-Sign-On)?" Accessed: Oct. 17, 2023. [Online]. Available: https://aws.amazon.com/what-is/sso
[14] Cloudflare.com, "What is SSO? How single sign-on works." Accessed: Oct. 19, 2023. [Online]. Available: https://www.cloudflare.com/en-gb/learning/access-management/what-is-sso
[15] Openldap.org, "OpenLDAP Software 2.4 Administrator's Guide." Accessed: Oct. 21, 2023. [Online]. Available: https://www.openldap.org/doc/admin24/OpenLDAP-Admin-Guide.pdf
[16] S. Farizy, "Implementasi Single Sign-On Berbasis Active Directory Sebagai Basis Data dan Layanan Direktori," *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, vol. 28, no. 1, 2018.
[17] Panji999999, "Lightweight Directory Access Protocol," https://id.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol.
[18] Z. Wu and Y. Cheng, "Research and Implementation of High Available Digital Campus Unified Identity Authentication System Based on LDAP," *Academic Journal of Computing & Information Science*, vol. 6, no. 7, pp. 8–14, 2023.
[19] K. A. Hafizd, "Perancangan Single Sign On (SSO) Pada Aplikasi Web Menggunakan Cloud Identity," *Antivirus: Jurnal Ilmiah Teknik Informatika*, vol. 15, no. 2, pp. 242–251, 2021.
[20] Apereo.com, "Apero CAS Documentation." Accessed: Oct. 21, 2023. [Online]. Available: https://apereo.github.io/cas/6.6.x/index.html