Evaluasi Penerapan Single Sign-On SAML dan OAuth 2.0: Studi pada Perguruan Tinggi Yogyakarta

Salmuasih
18917127

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Sistem Informasi Enterprise
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2023

Lembar Pengesahan Pembimbing

Evaluasi Penerapan Single Sign-On SAML dan OAuth 2.0:
Studi pada Perguruan Tinggi Yogyakarta

Salmuasih
18917127

Yogyakarta, 2023

Pembimbing

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.

Lembar Pengesahan Penguji

Evaluasi Penerapan Single Sign-On SAML dan OAuth 2.0:
Studi pada Perguruan Tinggi Yogyakarta

Salmuasih
18917127

Yogyakarta, 2023

Tim Penguji,

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.
Ketua

Dr. Raden Teduh Dirgahayu, ST., M.Sc.
Anggota I

Irving Vitra Paputungan, ST., M.Sc., Ph.D.
Anggota II

Mengetahui,
Ketua Program Studi Informatika Program Magister
Universitas Islam Indonesia

Irving Vitra Paputungan, ST., M.Sc., Ph.D.

Abstrak

Evaluasi Penerapan Single Sign-On SAML dan OAuth 2.0: Studi pada Perguruan Tinggi Yogyakarta

Digitalisasi proses pendidikan semakin diminati, terlihat dari semakin banyaknya aplikasi dan sumber daya yang dikelola oleh Perguruan Tinggi untuk keperluan dosen, staf, dan mahasiswa. Sebagai contoh aplikasi akademik, *Learning Management System* (LMS), *online meeting*, email, dst. IT Center perguruan tinggi memerlukan manajemen akses yang efektif dan efisien. *Identity Access Management* dan *Single Sign-On* (SSO) dapat menjadi salah satu solusinya. Untuk memutuskan strategi SSO yang efektif, IT Center perlu memahami manfaat SSO, mengidentifikasi kebutuhan spesifik organisasi, dan memilih protokol yang akan memenuhi kebutuhan tersebut.

Penelitian ini menyajikan *business case* protokol SSO SAML dan OAuth 2.0, mengidentifikasi kebutuhan spesifik organisasi, dan mengevaluasi penerapan kedua protokol tersebut di perguruan tinggi Yogyakarta. Penelitian dilakukan dengan metode kualitatif pada 22 responden dari 17 perguruan tinggi. Penelitian dilakukan melalui survei dan wawancara yang merujuk pada pertanyaan kuesioner. Hasil analisis menunjukkan bahwa masih terdapat ketidaksesuaian dalam praktik penerapan SSO SAML dan OAuth 2.0 pada beberapa perguruan tinggi Yogyakarta yang disebabkan oleh berbagai keterbatasan.

Kata kunci

identity access management, oauth 2.0, perguruan tinggi, saml, single sign-on

Abstract

Evaluation of the Implementation of Single Sign-On SAML and OAuth 2.0: Study at Yogyakarta Higher Education

Digitization of the educational process is increasingly in demand, as seen from the increasing number of applications and resources managed by Higher Education for the needs of lecturers, staff and students. For example, academic applications, Learning Management System (LMS), online meetings, email, etc. College IT Center requires effective and efficient access management. Identity Access Management and Single Sign-On (SSO) can be a solution. In order to decide on an effective SSO strategy, IT Centers need to understand the benefits of SSO, identify the specific needs of the organization, and choose a protocol that will meet those needs.

This study presents the business case of the SSO SAML and OAuth 2.0 protocols, identifies the specific needs of organizations, and evaluates the implementation of the two protocols in Higher Education Institution of Yogyakarta. The research was conducted using qualitative methods on 22 respondents from 17 universities. The research was conducted through surveys and interviews referring to survey questions. The results of the analysis show that there are still discrepancies in the practice of implementing SSO SAML and OAuth 2.0 at several Higher Education Institution of Yogyakarta due to various limitations.

Keywords

higher education, identity access management, oauth 2.0, saml, single sign-on

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Januari 2023

Salmuasih

Daftar Publikasi

Tidak ada publikasi yang menjadi bagian dari tesis.

Halaman Kontribusi

Dosen Pembimbing dan Dosen-dosen Penguji memberikan banyak kontribusi berupa arahan, saran, dan masukan perbaikan dalam penulisan penelitian ini.

Halaman Persembahan

Teruntuk sal, sebagai pengingat

Kata Pengantar

Segala puji bagi Allah SWT yang telah memberikan limpahan rahmat dan teriring sholawat serta salam kepada tauladan kita Nabi Muhammad SAW, yang memberikan kekuatan dalam penulisan tesis ini untuk memenuhi salah satu persyaratan dalam memperoleh gelar Master pada Program Studi Magister Informatika, Fakultas Teknik Industri Universitas Islam Indonesia Yogyakarta.

Dalam penulisan tesis ini penulis menyadari masih banyak kekurangan, disebabkan keterbatasan kemampuan dan pengetahuan yang penulis miliki, akan tetapi berkat bimbingan dan bantuan dari berbagai pihak sehingga pada akhirnya penulis dapat menyelesaikan tesis ini.

Dalam tesis ini penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah memberikan bantuan berupa bimbingan, motivasi, arahan dan do'a selama proses penulisan tesis ini. Ucapan terima kasih dan penghargaan penulis sampaikan kepada:

1. Bapak Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D., dosen pembimbing yang telah memberikan bimbingan, arahan, dan motivasi sehingga tesis ini dapat terselesaikan.
2. Bapak-bapak dosen penguji, saya ucapkan terimakasih atas semua arahan dan saran yang membangun demi perbaikan tesis ini.
3. Ibu Izzati Muhimmah, S.T., M.Sc., Ph.D., Ketua Program Studi Magister Informatika periode 2017 s/d 2022 yang begitu perhatian, terima kasih Ibu.
4. Bapak Irving Vitra Paputungan, ST., M.Sc., Ph.D., Ketua Program Studi Magister Informatika yang memfasilitasi dan melancarkan proses terselesaikannya tesis ini.
5. Bapak dan ibu dosen yang banyak memberikan ilmu, membimbing dan memotivasi penulis selama menempuh studi dan Staff akademik maupun non akademik Program Studi Magister Informatika yang senantiasa mengingatkan dan membantu penulis dalam administrasi terkait penulisan tesis.
6. Pusat IT 17 Perguruan Tinggi Yogyakarta yang memberikan izin penelitian dan seluruh responden yang bersedia membantu penulis dengan meluangkan waktunya untuk wawancara dan mengisi kuesioner sehingga tesis dapat terselesaikan.
7. Untuk teman-teman Marson's, Rumah Pelangi, Dilots, TE, SWT, dan SIE18 yang selalu siap membantu dan memotivasi, terima kasih.

8. Semua pihak yang tidak dapat disebutkan satu persatu, atas bantuan dalam penyelesaian tesis ini.

Penulis menyadari bahwa tesis ini masih banyak terdapat kekurangan. Oleh karena itu, penulis dengan senang hati menerima saran dan kritik yang membangun dari semua pihak sehingga dapat bermanfaat bagi kita semua.

Yogyakarta, Januari 2023

Salmuasih

Daftar Isi

Daftar Tabel

Daftar Gambar

BAB 1
Pendahuluan

1.1 Latar Belakang

Menurut *press release* Gartner pada April 2021, selama pandemi Covid-19 migrasi dari *on-premises* ke *cloud* mengalami pertumbuhan yang signifikan. Kebutuhan industri dan akademik terhadap layanan *cloud* terus meningkat, terutama pada model layanan *Software-as-a-Service* (SaaS) yang menjadi segmen pasar terbesar, disusul *Platform-as-a-Service* (PaaS) dan *Infrastructure-as-a-Service* (IaaS). Sebagai contoh di perguruan tinggi, proses belajar mengajar yang semula tatap muka beralih ke platform online, memanfaatkan berbagai aplikasi *Learning Management System* (LMS) dan online meeting seperti Zoom, Google Meet, Microsoft Teams, dan Cisco Webex Meeting (Costello & Rimol, 2021). Selain SaaS, perguruan tinggi juga masih menggunakan aplikasi pada infrastruktur *on-premises*, dimana sebagian besar *monolith* dan menggunakan *role-based access control* (RBAC) (Triartono & Negara, 2019).

Dengan semakin banyaknya aplikasi dan sumber daya yang tersedia untuk pengajar, staf, dan mahasiswa, penyediaan akses yang *seamless* menjadi semakin penting. *Single Sign-On* (SSO) dapat menjadi salah satu solusi untuk mengintegrasikan sistem yang dimiliki perguruan tinggi (Aldosary & Alqahtani, 2021). SSO menyediakan tempat terpusat untuk mengakses semua aplikasi dan sumber daya dengan satu *username* dan *password. Security Assertion Markup Language* (SAML) dan *Open Authentication* (OAuth) 2.0 merupakan protokol SSO yang populer digunakan (Indu et al., 2018). OAuth 2.0 adalah kerangka kerja yang mengontrol otorisasi ke sumber daya yang dilindungi seperti aplikasi atau kumpulan file, sementara SAML merupakan standar industri untuk autentikasi federasi. Artinya, OAuth 2.0 digunakan dalam situasi yang secara fundamental berbeda dari SAML, namun dapat digunakan secara bersamaan (Ito et al., 2013).

Dalam pemilihan protokol SSO yang efektif, perlu mempertimbangkan kebutuhan aplikasi saat ini, kondisi infrastruktur organisasi, seberapa banyak upaya yang diperlukan, dan kebutuhan organisasi dimasa mendatang. Pertimbangan kebutuhan aplikasi artinya kebutuhan sistem atau *resource* yang perlu diintegrasikan dengan SSO. Pertimbangan kondisi infrastruktur organisasi bertujuan agar protokol yang dipilih dapat diimplementasikan dengan infrastruktur yang dimiliki perguruan tinggi. Namun pada

penerapannya, memungkinan adanya gap antara kebutuhan organisasi dengan protokol SSO yang dipilih. Hal ini berdampak pada berkurangnya efektifitas protokol SSO yang diterapkan, bahkan dapat menimbulkan masalah baru lainnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, diperlukan evaluasi untuk mengetahui efektifitas dan kesesuaian protokol SSO yang dipilih/diterapkan terhadap kebutuhan yang dimiliki perguruan tinggi, sehingga setiap fitur dari kedua protokol SAML dan OAuth 2.0 dapat dimanfaatkan dengan optimal.

1.3 Pertanyaan Penelitian

1. Seperti apakah kebutuhan adopsi SSO perguruan tinggi Yogyakarta?
2. Seperti apakah efektivitas implementasi protokol SSO SAML dan OAuth 2.0 pada perguruan tinggi Yogyakarta?
3. Seperti apakah rencana dan kendala penerapan SSO pada perguruan tinggi Yogyakarta yang belum menerapkannya?

1.4 Tujuan Penelitian

1. Untuk menganalisis kebutuhan adopsi SSO perguruan tinggi Yogyakarta.
2. Untuk menganalisis efektivitas implementasi protokol SSO SAML dan OAuth 2.0 pada perguruan tinggi Yogyakarta.
3. Untuk menganalisis rencana dan kendala penerapan SSO pada perguruan tinggi Yogyakarta yang belum menerapkannya

1.5 Manfaat Penelitian

Hasil penelitian yang dilakukan diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat Teoritis

   Secara teoritis diharapkan dapat memberikan sumbangan pemikiran dalam memperkaya wawasan pada ilmu komputer khususnya sistem informasi *enterprise*.

2. Manfaat Praktis

   Secara praktis diharapkan dapat menyumbangkan pemikiran berkaitan dengan implementasi protokol SSO khususnya SAML dan OAuth 2.0. Selanjutnya

hasil penelitian diharapkan dapat membantu perguruan tinggi dalam proses evaluasi dan adopsi SSO.

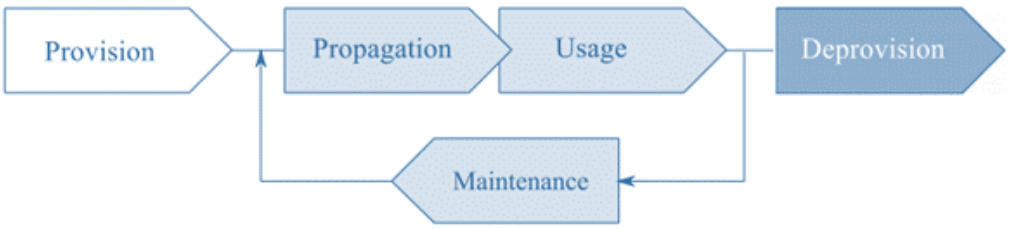
BAB 2
Tinjauan Pustaka

2.1 Landasan Teori

2.1.1 Identity Lifecycle

Aplikasi merupakan layanan atau kelompok layanan yang tersedia untuk pengguna yang terdaftar. Aplikasi ini dapat memanfaatkan internet atau infrastruktur perangkat keras jaringan lainnya untuk menjalankan fungsi yang bermanfaat, seperti berbagi data. *Lifecycle* (siklus hidup) adalah definisi fase untuk mengidentifikasi status objek dalam suatu periode. Oleh karena itu, siklus hidup identitas menentukan status identitas pengguna dalam konteks aplikasi. Pada Gambar 2.1, identitas memiliki kerangka siklus hidup generik yang dapat diterapkan terlepas dari aplikasinya, dan terdiri dari lima fase: *provisioning, propagation, usage, maintenance, dan de-provisioning* (Cremonezi et al., 2020).

Setiap konteks aplikasi mungkin memiliki siklus hidup identitasnya sendiri, dan sangat penting merencanakan setiap fase untuk membangun arsitektur identitas. Secara singkat, identitas mulai disediakan atau dibuat untuk subjek. Setelah dibuat, subjek menjadi pengguna aplikasi tersebut, dan identitasnya disebarluaskan oleh aplikasi yang menggunakan informasi identitas pengguna ini. Setelah disebarkan, aplikasi menggunakan identitas, dan terkadang beberapa perubahan identitas, seperti perubahan kredensial atau penambahan atribut, dapat terjadi. Dalam kasus ini, pembaruan identitas memaksa identitas untuk disebarkan lagi. Akhirnya, ketika identitas ini memenuhi kegunaannya dan tidak lagi dibutuhkan, identitas tersebut akan dinonaktifkan atau dihapus.



Gambar 2.1 *Identity Lifecycle*

Identity Provision, pembuatan identitas untuk subjek, langkah sebelum menjadi pengguna aplikasi. Oleh karena itu, proses *provisioning* identitas membuat identifikasi unik, kredensial, dan catatan atribut subjek. Atribut ini dapat berupa, misalnya, lokasi, email, dan atribut khusus untuk konteks aplikasi. Beberapa aplikasi memerlukan penyediaan identitas eksplisit untuk penggunanya. Dalam kasus tersebut, yang biasanya

adalah orang, pengguna mendaftarkan dirinya ke dalam aplikasi yang mengirimkan atribut dan bukti identitas untuk digunakan sebagai kredensial. Aplikasi memeriksa keaslian, validitas, dan keakuratan atribut ini sebelum membangun hubungan antara subjek dan identitas. Namun, ada kasus dimana identitas subjek tidak ditetapkan secara eksplisit. Dalam kasus tersebut, dimungkinkan untuk membangun identitas digital pengguna, berdasarkan kumpulan berbagai atribut jaringan yang digunakan dalam berbagai konteks. Biasanya subjek tidak menyadari konstruksi identitas digital ini.

Identity Propagation, beberapa aplikasi mengharuskan bagian dari identitas dapat diakses pada sistem lain. Tujuan replikasi ini adalah agar dapat mereplikasi identitas untuk kinerja yang lebih baik, efisiensi biaya, atau sistem pertahanan kegagalan. Aplikasi yang lebih kompleks mungkin memerlukan direktori identitas terpadu dimana identitas yang dibuat oleh beberapa aplikasi dapat digunakan di aplikasi lain. Idealnya, propagasi harus terjadi setelah setiap perubahan identitas, dan propagasi harus terjadi dengan cara yang *reliable* untuk menghindari masalah keamanan dan konsistensi.

Identity Usage, adalah fase utama dari siklus hidup identitas. Selama fase ini, aplikasi dan pengguna menggunakan identitas untuk melakukan verifikasi identitas. Fase ini menentukan apakah identitas pengguna sah dan access control yang dimiliki memungkinkan akses terhadap sumber daya.

Identity Maintenance, umumnya identitas tidak statis, atribut dan kredensial mungkin mengalami beberapa perubahan selama siklus hidup identitas. Misalnya, karakteristik pengguna dapat berubah seiring waktu, dan identitasnya harus mengikuti perubahan ini. Sebagai contoh lain, aplikasi harus mendukung peluang bisnis baru yang membutuhkan perubahan identitas secara menyeluruh dengan menambahkan atribut baru. Terlepas dari faktor yang menyebabkan perubahan ini, perubahan identitas harus dapat diakses pada semua aplikasi yang terpengaruh.

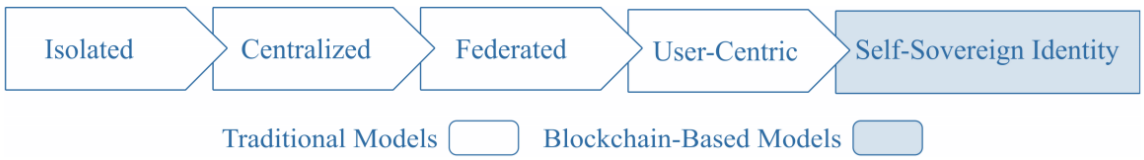
Identity Deprovision, menghapus identitas di akhir siklus hidupnya sama pentingnya dengan memberikan identitas tersebut. *Deprovisioning* adalah proses yang memungkinkan aplikasi mengetahui pengguna mana yang tidak lagi valid. Pendekatan yang paling mudah untuk *Identity Deprovision* hadir dengan penghapusan total identitas pengguna. Namun, menghapus identitas berarti menghapus semua informasi terkait, yang mungkin masih diperlukan untuk audit. Dengan demikian, aplikasi memilih untuk menonaktifkan daripada menghapus identitas. Dalam kasus tersebut, aplikasi mencabut kredensial identitas, artinya identitas tersebut masih ada tetapi tidak lagi ditautkan ke pengguna dan tidak memiliki hak akses apa pun yang terkait dengannya. Oleh karena itu,

aplikasi meminimalkan hilangnya informasi dan tetap aman; namun, biaya untuk memelihara informasi identitas yang dinonaktifkan mungkin terbebani. Untuk alasan ini, beberapa aplikasi menggunakan pendekatan hybrid. Mereka menerapkan mekanisme yang menonaktifkan identitas terlebih dahulu tetapi menghapusnya hanya setelah selang waktu tertentu. Oleh karena itu, pendekatan ini bertujuan untuk menggabungkan manfaat dari kedua pendekatan yang telah disebutkan sebelumnya; namun, itu menyimpan kekurangan yang pertama setelah penghapusan.

2.1.2 *Identity and Access Management* (IAM)

Identity Management juga dikenal sebagai *Identity and Access Management* (IAM) adalah sistem untuk mengelola siklus hidup identitas digital pengguna. Tujuan utama dari IAM adalah untuk memastikan bahwa hanya pengguna yang diautentikasi yang memiliki akses ke layanan tertentu, terdiri dari empat modul utama yaitu *Authentication, Authorization, User Management* dan *Central User Repository* (Joshi et al., 2018). IAM mengacu pada serangkaian proses yang dapat diterapkan untuk memberikan hak akses kepada orang yang tepat dalam lingkungan perusahaan manapun. Bisnis mengimplementasikan solusi IAM untuk mengelola ID dan *password* pengguna serta memecahkan masalah terkait tantangan pengelolaan akses dan izin dari beberapa ID pengguna. Solusi IAM juga memberikan kemampuan audit yang memungkinkan para manajer dan pejabat tinggi dalam organisasi untuk melacak berbagai akses yang dimiliki karyawan. Karyawan yang diberhentikan namun masih memiliki akses ke sumber daya organisasi dapat menyebabkan kerusakan tertentu. Sistem IAM dapat mengurangi risiko terkait saat menghapus atau menambahkan pengguna secara manual ke sistem yang berbeda (Joshi et al., 2018).

IAM terus dikembangkan untuk mengikuti perubahan teknologi. Pada IAM generasi pertama, tidak ada pemisahan antara entitas yang menawarkan layanan atau informasi identitas; oleh karena itu, setiap layanan mengelola identitas penggunanya masing-masing (*isolated*). Seiring waktu muncul model baru yang lebih fleksibel. Terdapat lima model IAM yaitu model *isolated, centralized, federated, user-centric*, dan *self-sovereign*. Dalam model *isolated, centralized, federated, user-centric*, pengguna selalu bergantung pada penyedia identitas pihak ketiga untuk menyimpan dan berbagi data. Disisi lain, model *self-sovereign* memungkinkan pengguna untuk menyimpan dan mengelola semua data identitas digital secara mandiri, memungkinkan pengguna untuk berbagi informasinya dengan SP yang dipilih secara selektif. Gambar 2.2 menunjukkan evolusi model IAM dari waktu ke waktu sejak model *isolated* hingga *self-sovereign*.



Gambar 2.2 Model *Identity and Access Management*

1. Model *Isolated*

Model ini telah berkembang pesat dengan komputasi terpusat. Pada model ini, pengguna yang ingin mengakses layanan SP harus terlebih dahulu membuktikan identitasnya. Identitas ini memiliki atribut pengguna, dan SP mengelola atribut ini secara terpisah. Oleh karena itu, karakteristik utama dari model ini adalah bahwa SP memikul tanggung jawab IdP, mengelola dan menyimpan semua identitas dan atribut penggunanya. Sebagai ilustrasi, seorang pengguna mendaftarkan dirinya di dua SP untuk mendapatkan layanan. Pada setiap SP, pengguna memiliki identitas unik dengan atributnya, dan setiap identitas memiliki kredensial unik. Setelah tidak ada kerjasama antara kedua SP ini, setiap SP mengambil peran IdP sendiri, mengelola identitas pengguna dalam isolasi total dari yang lain.

Kesederhanaan model ini hadir dengan beberapa kekurangan. Mengelola identitas dalam lingkup lokal sangatlah mudah, yang memungkinkan autentikasi, otorisasi, dan akuntansi sederhana. Namun, skalabilitas menjadi jelas seiring bertambahnya jumlah identitas. Setelah pengguna memiliki identitas yang berbeda pada setiap SP yang terdaftar, model ini dapat menyebabkan kredensial yang digunakan kembali atau membuat beberapa di antaranya dilupakan. Selain itu, model ini dapat mencederai privasi pengguna terdaftar, karena setiap layanan memiliki identitasnya dengan semua atribut yang diperlukan.

2. Centralized

Model ini tidak membatasi ruang lingkup identitas ke satu layanan. Oleh karena itu, model ini memperkenalkan IdP pusat, yang merupakan otoritas identitas, yang memusatkan IAM. Setelah menetapkan identitas, pengguna dapat menggunakan SP yang dilampirkan ke IdP tersebut tanpa harus terlibat dalam proses autentikasi secara eksplisit. Konsep ini - dikenal sebagai Single Sign-On (SSO) - memungkinkan pengguna dengan satu identitas unik mengakses beberapa layanan. Dua SP setuju untuk mengirimkan tugas ini ke IdP terpusat alih-alih menjadi IdP mereka sendiri. Oleh karena itu, saat pengguna mendaftarkan diri di IdP pusat tersebut, dia memiliki identitas unik yang memberinya akses ke semua SP bergantung pada IdP yang sama.

Kemudian, berbeda dengan model *isolated*, pengguna dapat memiliki akses ke kedua SP menggunakan identitas yang sama. Dengan sentralisasi ini, pengguna hanya perlu menghafal satu identitas dan kredensialnya, bukan beberapa identitas dan kredensial. Namun, sentralisasi ini merupakan keuntungan yang diragukan, karena jika satu pengidentifikasi dengan kredensial terkait dikompromikan, semua layanan yang dapat diakses oleh identitas juga dikompromikan. Selain itu, aspek terpusat dari model ini tidak menyelesaikan masalah skalabilitas sejumlah besar pengguna atau layanan.

3. Federated

Konsep federasi merepresentasikan hubungan antara dua atau lebih organisasi yang memiliki kapabilitas infrastruktur identitas. Dalam model ini, sekelompok IdP dan SP terikat bersama untuk membentuk federasi. Dalam hal ini, diatur oleh seperangkat perjanjian komersial dan platform teknologi standar, pengguna yang berpartisipasi dalam satu organisasi dapat langsung mengakses SP di organisasi lain. Hasilnya adalah pengguna yang berpartisipasi dalam federasi memiliki perpanjangan layanan tanpa perlu mengelola identitasnya di organisasi lain. Dengan kata lain, model ini memungkinkan beberapa otoritas identitas untuk berbagi kekuatan. Sama seperti model terpusat, SSO juga diperbolehkan di federasi. Dalam hal ini, pengguna dapat mengautentikasi dirinya sendiri satu kali dengan satu IdP, dan semua IdP federasi menganggap pengguna tersebut diautentikasi, memungkinkan pengguna tersebut mengakses semua SP yang juga merupakan anggota federasi. Dalam bentuknya yang murni, dalam federasi identitas, pengguna hanya perlu memiliki satu profile identitas di organisasi asalnya. Namun, sebuah identitas dapat dijangkau di beberapa IdP yang berpartisipasi dalam federasi. Rentang identitas ini terjadi karena, meskipun menghindari redudansi merupakan inti dari desain federasi, terkadang IdP masih perlu mereplikasi identitas untuk manajemen internalnya atau bahkan untuk kinerja yang lebih baik atau untuk mengurangi biaya dan risiko kegagalan.

Begitu identitas dapat membawa informasi berharga, ada beberapa peraturan tentang perlindungan privasi dan pengungkapan identitas. Oleh karena itu, melakukan replikasi identitas secara sembarangan dapat menimbulkan masalah keamanan dan privasi. Dengan demikian, penting untuk menginformasikan tujuan replikasi identitas. Salah satu pendekatan untuk memenuhi persyaratan privasi dan keamanan adalah melalui penggunaan identitas parsial dengan nama samaran (*pseudonyms*). Dalam pendekatan ini, identitas pengguna direplikasi hanya untuk membawa atribut identitas asli yang diperlukan. Nama samaran adalah pengidentifikasi baru dari identitas yang

direplikasi ini, yang membantu mencapai sesuatu yang mendekati anonimitas dalam identitas baru ini. Agar ini berhasil, perlu untuk menegakkan manajemen yang tepat dari nama samaran ini untuk menjaga hubungan pribadi antara identitas dan nama samaran. Namun, model federasi saat ini tidak memiliki mekanisme yang efektif untuk menjaga konsistensi informasi pengguna selama modifikasi atau pencabutan identitas.

Sebagai ilustrasi model federasi tipikal, saat dua organisasi membentuk federasi, berarti kedua organisasi ini mewujudkan serangkaian kesepakatan, standar, dan teknologi untuk memungkinkan keduanya mengenali identitas dari organisasi lain. Dengan demikian, identitas pengguna, yang sebelumnya didaftarkan oleh organisasi 1, sekarang diterima oleh organisasi 2. Oleh karena itu, pengguna sekarang dapat mengautentikasi dirinya sendiri satu kali dengan IdP mereka dan memberikan akses ke semua SP yang menjadi anggota federasi. Dalam contoh ini, jika pengguna ingin meminta layanan dari SP di organisasi 2, IdP organisasi 1 mengautentikasinya dan mengirimkan pesan mirip klaim ke IdP organisasi 2: "Saya adalah IdP organisasi 1, dan saya mengautentikasi pengguna itu". Dengan demikian, IdP organisasi 1 membuat pengenal nama samaran yang ditautkan ke identitas sebenarnya dan membagikan pengenal ini dengan IdP dari organisasi 2 untuk memastikan bahwa pengguna dapat menggunakan SP dari organisasi 2 tanpa mengungkapkan identitas aslinya. Melalui nama samaran ini, kedua IdP setuju bahwa mereka mengacu pada pengguna yang sama. Namun, hanya IdP dari organisasi 1, yang diberi nama samaran, yang menyimpan identitas asli pengguna dengan semua atribut yang terkait dengannya. Model federasi memperkenalkan gagasan untuk menawarkan satu set SP kepada pengguna hanya dengan satu identitas. Namun, tidak realistis untuk menganggap federasi global mencakup semua SP. Dengan demikian, jumlah identitas yang perlu dikelola pengguna terus meningkat karena pengguna mungkin memiliki identitas di beberapa federasi.

4. User-centric

Model ini adalah yang pertama memperkenalkan sistem yang mendukung manajemen identitas di sisi pengguna. Alih-alih mengelola beberapa identitas, pengguna memiliki perangkat anti-perusakan pribadi yang menyimpan beberapa pengenal dan kredensial, bersama dengan IdP yang menyediakannya. Perangkat ini bertindak seperti pemilih IdP dan berisi portofolio pengidentifikasi dan kredensial dari berbagai IdP. Pendekatan ini membuka kemungkinan bagi pengguna untuk hanya perlu mengelola identitasnya dengan pemilih IdP pribadi mereka. Setelah diautentikasi

dengan pemilih IdP, pengguna mengizinkan pemilih IdP menangani autentikasi dengan IdP eksternal. Dalam model ini, pada setiap penggunaan identitasnya, pengguna harus menyetujuinya secara eksplisit, artinya tidak mungkin untuk mengungkapkan informasi tersebut kepada pihak ketiga tanpa izinnya. Secara umum, model *centralized, federation, dan user-centric* menaruh kepercayaan pada IdP, mentransfer kendali atas identitas kepada mereka. Oleh karena itu, IdP menjadi penyimpanan data informasi pribadi yang besar, menyimpan semua jenis data tentang pengguna.

5. *Self-sovereign*

IAM *self-sovereign* mengandalkan *distributed ledger technology* (DLT), yang pada dasarnya adalah infrastruktur dan protokol teknologi yang memungkinkan perekaman dan berbagi data di seluruh jaringan terdistribusi dari peserta yang berbeda. Dimungkinkan untuk merekam, berbagi, dan menyinkronkan data ini dengan cara yang tidak dapat diubah di seluruh jaringan, tanpa memerlukan koordinator pusat. Singkatnya, DLT memberikan kendali atas evolusi data antara pengguna melalui jaringan peer-to-peer, biasanya menggunakan algoritma konsensus untuk memastikan replikasi di antara node jaringan. Struktur data DLT memungkinkan pembuatan buku besar transaksi yang dapat dirusak, dan buku besar tetap sama melalui jaringan. Singkatnya, semua peserta dapat melihat semua data yang direkam pada buku besar yang terdiri dari "Blok" yang terhubung secara kriptografis, yang merupakan potongan informasi digital. Keamanan DLT datang dengan fakta bahwa setelah membuat dan menambahkan blok ke blockchain, tidak mungkin mengubah atau mengembalikan transaksi di blok itu. Awalnya terfokus pada sektor keuangan, DLT dengan cepat menyebar di beberapa bidang, secara inklusif berakhir sebagai kunci dari IAM *self-sovereign*.

Model ini muncul dari konsep yang memungkinkan pengguna untuk menyimpan data identitas, menghilangkan kontrol terpusat dari otoritas identitas. Jadi, alih-alih bergantung pada IdP, pengguna menjadi IdP mereka sendiri, artinya mereka menyimpan dan mengelola atributnya. Dalam mode *self-sovereign*, pengguna mengendalikan identitas mereka, tidak bergantung pada otoritas pusat untuk tujuan ini. Agar ini berfungsi, informasi identitas harus disediakan secara efisien untuk layanan yang perlu memvalidasinya, harus berada di lingkungan tepercaya, dan tidak boleh dimiliki atau dikendalikan oleh siapa pun. Untuk alasan keamanan dan privasi, meletakkan data pribadi apa pun di buku besar bukanlah pendekatan terbaik karena buku besar tidak dapat diubah. Dengan demikian, tidak mungkin mengubah atau

menghapus data apa pun yang ditulis ke buku besar. Oleh karena itu, alih-alih berbagi atribut saat ini, model ini menggunakan DLT untuk berbagi sekumpulan klaim, bukti, dan pengesahan. Model ini beroperasi melalui metode *Zero-Knowledge Proof*, yang memungkinkan satu pengguna untuk membuktikan kepada pengguna lain bahwa mereka mengetahui informasi tertentu atau memenuhi persyaratan tertentu tanpa mengungkapkan informasi aktual yang mendukung bukti tersebut.

Terdapat tiga entitas dalam model *self-sovereign*: *Identity Owner, Identity Proofer,* dan *Identity Verifier*. *Identity Owner* adalah pengguna dengan kontrol penuh atas identitasnya. Ketika *Identity Owner* ingin berbagi beberapa data dengan orang lain, informasi yang dimilikinya menjadi publik. *Identity Proofer* bertanggung jawab untuk membuktikan validitas data yang diklaim oleh *Identity Owner*. Penting untuk menunjukkan bahwa klaim yang dibuat oleh pemilik identitas dapat ditegaskan sendiri atau ditegaskan oleh entitas lain yang keasliannya dapat diverifikasi secara independen oleh pihak yang mengandalkan. Melalui data yang sudah terverifikasi, *Identity Owner* dan *Identity Verifier* mengalami cara yang lebih aman untuk memeriksa atribut karena *Identity Owner* tersebut tidak membagikan data yang tidak diperlukan, dan *Verifier* tidak menyimpan data sensitif.

2.1.3 *Authentication*

Authentication atau autentikasi adalah proses menyetujui suatu entitas untuk melalui entitas lain. Autentikasi digunakan untuk memastikan apakah orang atau aplikasi memenuhi syarat untuk mengakses atau mengklaim. Proses autentikasi biasanya dilakukan oleh sebuah perangkat lunak atau bagian dari perangkat lunak. Metode autentikasi umum dalam lingkungan jaringan adalah kredensial *login, multifactor authentication, third-party authentication, password* teks sederhana, *password* objek 3D, *password grafis, biometric authentication,* dan autentikasi perangkat digital. Sistem *cloud* mengikuti salah satu atau kombinasi dari mekanisme autentikasi tersebut di atas. Saat ini, izin akses *cloud* diberikan melalui sistem manajemen identitas (Indu et al., 2018). Berikut merupakan mekanisme autentikasi:

1. Mekanisme keamanan fisik

   Mekanisme keamanan fisik seperti kartu akses dan biometrik memastikan keamanan sumber daya dan fasilitas *cloud* dengan menolak akses tidak sah melalui autentikasi. Pusat data cloud memusatkan semua server, jaringan, dan aplikasi sehingga pengguna dapat mengakses data kapan pun dan dari lokasi mana pun.

Sebagai bagian dari keamanan pusat data, kartu akses dan otentikasi biometrik seperti pengenalan iris atau retina, pengenalan sidik jari, pengenalan wajah dan pengenalan telapak tangan dapat digunakan. Untuk mencegah kebocoran data dari orang dalam atau akses tidak sah ke pusat data, diperlukan keamanan fisik bersama dengan kebijakan penggunaan dan tata kelola tertentu. Mekanisme keamanan fisik yang saat ini digunakan adalah kontrol akses biometrik dan perangkat digital untuk autentikasi (Indu et al., 2018).

2. Mekanisme keamanan digital

Mekanisme kemananan digital terdiri dari kredensial dan *secure shell keys*, *multifactor authentication* (MFA), *Chip* & PIN.

Kredensial adalah bukti otoritas, status, hak akses, dan kepemilikan untuk memberikan bukti bahwa pengguna tertentu berhak atau pantas memanfaatkan sumber daya dan layanan. Penggunaan kredensial seperti kata sandi satu kali, pola, dan captcha adalah cara tradisional untuk mengamankan sistem dari aktivitas berbahaya. Mekanisme yang paling umum digunakan untuk mengelola kredensial akses untuk lingkungan cloud adalah teknologi *Lightweight Directory Access Protocol* (LDAP) dan *Microsoft Active Directory* (AD). Penting untuk menambah, menonaktifkan, memodifikasi, atau menghapus akun setiap kali karyawan meninggalkan atau memasuki organisasi. Dalam mengelola kredensial di sisi penyedia, kerentanan *reset* kredensial yang lemah direpresentasikan saat menggunakan mekanisme pemulihan kata sandi yang lemah. Peretas dapat memantau atau memanipulasi data di *cloud* bersama dengan *malicious redirect* setiap kali kredensial disusupi (Indu et al., 2018).

Secure Shell* (SSH) *key membantu mengidentifikasi server SSH melalui kriptografi *public-key* atau autentikasi *challenge-response*. Keuntungan utama SSH *key* adalah otentikasi ke server dilakukan tanpa meneruskan kata sandi melalui jaringan. Ini mencegah penyadapan atau peretasan kata sandi. Upaya menebak kredensial melalui serangan *brute force* selama otentikasi dapat dihilangkan dengan SSH *key*. Agen SSH membantu membangun koneksi dengan server tanpa menggunakan kata sandi terpisah untuk setiap sistem. Agen SSH *key* menyimpan kunci pribadi dan memberikannya ke program klien SSH. Kunci pribadi ini dienkripsi dengan *passphrase* dan *passphrase* disediakan selama setiap upaya untuk terhubung dengan server. Dalam setiap pemanggilan SSH, *passphrase* diperlukan untuk mendekripsi kunci privat sebelum melanjutkan ke fase

autentikasi. *Passphrase* hanya digunakan selama proses penambahan kunci privat ke penyimpanan agen. Upaya ini mendukung perangkat komunikasi yang sering membuat koneksi SSH. Agen SSH berjalan secara otomatis setelah login dimulai dan bertahan selama durasi sesi. Perhatian utama dari SSH *key* adalah keamanannya tidak lebih baik dari kredensial jika kunci privat tidak terlindungi dengan baik. Kredensial statis dan mekanisme SSH *key* umumnya digunakan untuk autentikasi layanan web *cloud* (Indu et al., 2018).

***Multifactor authentication* (MFA)** adalah metode lain untuk mengamankan aset digital dan transaksi melalui Internet. Biasanya, *One-Time Password* (OTP), *Captcha* atau *Pattern* digunakan sebagai mekanisme otentikasi sekunder bersama dengan kredensial. MFA memberikan lapisan keamanan tambahan atas autentikasi berbasis kredensial tradisional. Umumnya, transaksi *online* diautentikasi menggunakan OTP. Dalam transaksi keuangan melalui *online*, server menghasilkan *one-time password* menggunakan algoritma khusus berdasarkan konfigurasinya dan OTP yang dihasilkan dikirim ke pengguna baik melalui nomor ponsel yang terdaftar atau melalui email. Jenis lain dari OTP dihasilkan dengan bantuan pembuat token perangkat keras/perangkat lunak yang dilindungi oleh *Personal Identification Number* (PIN). Kata sandi ini dapat digunakan sekali dan memiliki batas waktu penggunaan tertentu. *Captcha* biasanya digunakan untuk mengamankan aplikasi web dari serangan oleh program *malware*. *Captcha* bisa berupa kombinasi alfanumerik, persamaan matematika, atau gambar dan dengan ketentuan *refresh*. Pattern adalah bentuk otentikasi lain yang memiliki format berbeda. Pola titik-titik banyak digunakan dalam aplikasi seluler sementara pemilihan gambar yang cocok sebagian besar digunakan dalam aplikasi web. Penggunaan pertanyaan keamanan merupakan metode alternatif untuk mengamankan aset digital yang merupakan bentuk rahasia bersama. Dalam metode ini, pengguna memilih pertanyaan keamanan dari daftar yang telah ditentukan dan menentukan jawabannya. Pada saat autentikasi, pertanyaan keamanan yang telah ditentukan muncul di layar login dan dengan memberikan jawaban yang ditentukan bersama dengan kredensial memungkinkan pengguna untuk diautentikasi (Indu et al., 2018).

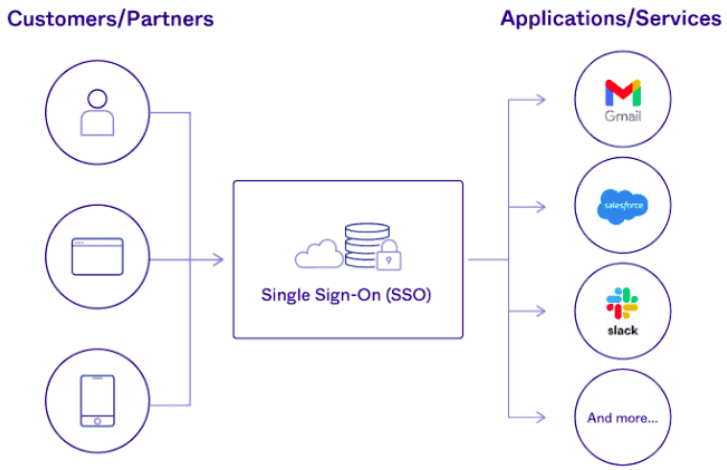
***Chip* dan *Personal Identification Number* (PIN)** adalah metode autentikasi konvensional untuk transaksi keuangan. Ini juga dapat digunakan untuk autentikasi ke mesin/layanan dalam jaringan organisasi. Teknologi enkripsi asimetris

menggunakan kunci publik dan pribadi untuk mengenkripsi dan mendekripsi data yang digunakan dalam mekanisme *chip* & PIN. *Chip* mikroprosesor menyimpan data pengguna dan kunci keamanan melalui pembuatan data transaksi unik untuk melindungi dari penipuan. Komunikasi antara klien/terminal dengan server autentikasi dienkripsi dan ditandatangani dengan bantuan *security key* yang disimpan didalam *chip*. Server memverifikasi tanda tangan dan mendekripsi komunikasi dengan bantuan kunci pasangan yang disimpan di server. PIN digunakan untuk mengautentikasi klien/terminal untuk mengakses data pengguna dan kunci dari *chip* (Indu et al., 2018).

2.1.4 Single Sign-On

IAM membantu organisasi mengelola semua aspek akses pengguna, sedangkan SSO adalah salah satu bagian dari lanskap identitas yang lebih luas tersebut. SSO sangat penting untuk memverifikasi identitas pengguna dan memberikan tingkat izin yang tepat, terintegrasi dengan log aktivitas, *access control tools,* dan proses yang memantau perilaku pengguna (Joshi et al., 2018).

Single Sign-On (SSO) telah banyak digunakan dalam keseharian. SSO adalah alat autentikasi pengguna yang memungkinkan pengguna untuk mengakses beberapa aplikasi dan layanan secara aman hanya dengan menggunakan satu set kredensial. Lingkungan kerja saat ini pada umumnya menggunakan berbagai aplikasi seperti Slack, Asana, Google Workspace, Zoom, atau aplikasi enterprise lainnya. SSO memberikan *widget pop-up* atau halaman *login* hanya dengan satu kata sandi yang memberi akses ke setiap aplikasi terintegrasi. Alih-alih dua belas kata sandi dalam sehari, SSO dengan aman memastikan pengguna hanya memerlukan satu kata sandi. SSO mengakhiri hari-hari mengingat dan memasukkan banyak kata sandi, dan menghilangkan rasa frustrasi karena harus mengatur ulang kata sandi yang terlupakan. Pengguna juga dapat mengakses berbagai platform dan aplikasi tanpa harus login setiap saat (Indu et al., 2018). Gambar 2.3 menunjukkan ilustrasi SSO.



Gambar 2.3 *Single Sign-On*[1]

Teknologi SSO berawal dari alat identitas lokal yang membantu menghubungkan komputer, jaringan, dan server secara aman pada pertengahan hingga akhir 1990-an. Saat ini, organisasi mulai mengelola identitas pengguna melalui sistem khusus seperti *Microsoft Active Directory* (AD) dan *Lightweight Directory Access Protocol* (LDAP), kemudian mengamankan akses melalui alat SSO atau *Web Access Management* (WAM) lokal. Teknologi Informasi (TI) yang terus berkembang dengan berpindah ke cloud, tersebar di beberapa perangkat, dan menghadapi ancaman dunia maya yang lebih canggih, alat manajemen identitas tradisional ini berjuang untuk mengimbanginya. Tim TI saat ini membutuhkan solusi akses SSO yang cepat dan aman ke aplikasi atau layanan (Joshi et al., 2018).

Ada tiga varian utama SSO: Web SSO, *Legacy* Web SSO, dan *Federated* SSO. Deskripsi singkat tentang masing-masing varian tersebut di bawah ini (Bazaz & Khalique, 2016):

a. **Web *Single Sign On***: Web *Single Sign On* biasa disebut sebagai manajemen akses web. Web SSO akan menerbitkan *trust relationship* atau memberikan akses ke *resource* yang diizinkan kepada pengguna, hanya setelah pengguna berhasil menyelesaikan proses autentikasi.

b. ***Legacy* Web *Single Sign On***: *Legacy* SSO disebut juga sebagai *Enterprise* SSO. Setelah proses autentikasi berhasil, *Legacy* SSO mengelola login ke beberapa aplikasi. Struktur Web SSO and *Legacy* SSO hampir identik. Perbedaannya terletak pada fakta bahwa Web SSO hanya mengelola layanan berbasis web, sedangkan

[1] Sumber: https://net.cloudinfrastructureservices.co.uk/wp-content/uploads/2021/10/what-is-sso.png

Legacy SSO memperluas fungsionalitas SSO ke aplikasi *legacy* tradisional dan *resource* jaringan, biasanya dalam jaringan internal perusahaan.

c. ***Federated Single Sign On***: *Federated* SSO memiliki konsep yang jauh lebih luas daripada Web SSO. *Federated* SSO menggunakan *Simple Object Access Protocol* (SOAP) dan *Security Assertion Markup Language* (SAML) untuk memungkinkan pengguna masuk menjadi anggota grup organisasi yang terafiliasi, dan selanjutnya mengakses semua situs web dalam federasi tepercaya tersebut. *Federated* SSO memperluas fungsionalitas SSO dari domain asal pengguna ke domain lainnya. Fungsi ini menjadi keunggulan *Federated* SSO. Organisasi yang menggunakan *Federated* SSO diizinkan untuk mengelola sendiri layanan lokal dan menunjukkan resource ke kelas pengguna yang lebih besar tanpa administrasi langsung perusahaan.

Ada berbagai protokol dan standar yang harus diperhatikan saat mengidentifikasi dan bekerja dengan SSO. *Security Assertion Markup Language* (SAML), *Open Authentication* (OAuth) 2.0 dan OpenID Connect menyediakan fasilitas SSO dengan mengizinkan Penyedia Identitas (*Identity Provider* - IdP) untuk berbagi informasi autentikasi dan otorisasi dengan Penyedia Layanan (*Service Provider* - SP) (Naik & Jenkins, 2016).

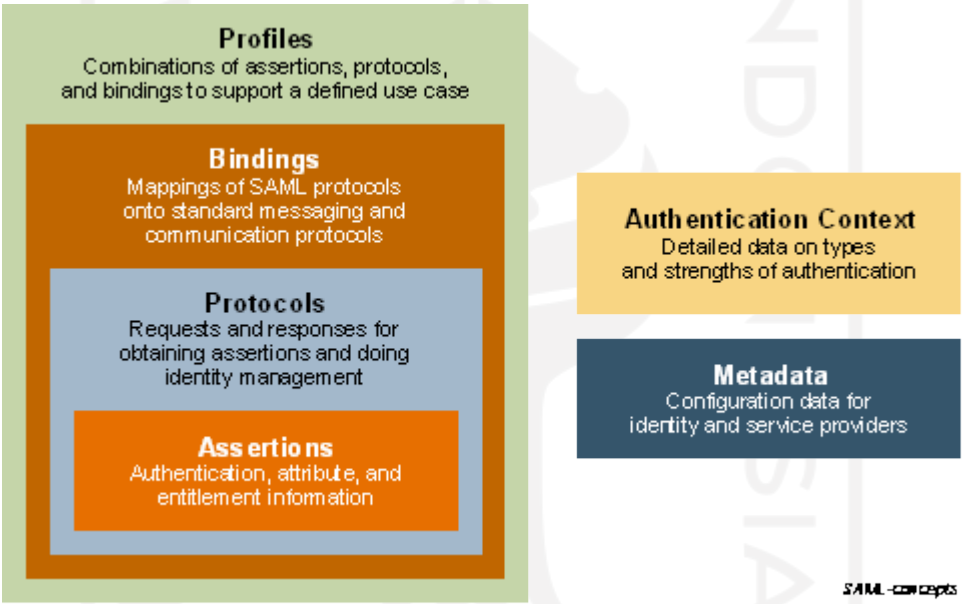
2.1.5 *Security Assertion Markup Language (*SAML)

Security Assertion Mark-up Language (SAML) 1.0 dirilis pertama kali pada tahun 2002 oleh *Organization for the Advancement of Structured Information Standards* (OASIS) (Aldosary & Alqahtani, 2021). SAML 1.1 mengikuti pada bulan September 2003 dan terlihat keberhasilan yang signifikan dalam layanan keuangan, pendidikan tinggi, pemerintah, dan segmen industri lainnya. SAML telah diterapkan secara luas oleh vendor besar manajemen akses web dan keamanan. Berdasarkan kesuksesan tersebut, OASIS menyetujui standar SAML 2.0 pada Maret 2005.

SAML adalah kerangka kerja berorientasi XML untuk mentransmisikan autentikasi pengguna, hak, dan informasi atribut lainnya (Naik & Jenkins, 2017). Hal ini memungkinkan dua mitra federasi untuk memilih dan berbagi atribut identitas yang diinginkan pada *payload* SAML *assertion (message)* selama atribut tersebut dapat direpresentasikan dalam XML. SAML mengandalkan tiga peran utama dalam setiap transaksi: *Identity Provider* (IdP), *Service Provider* (SP), dan Pengguna. *SAML assertion (security token)* adalah konsep inti dari SAML. SAML *assertion* adalah klaim, pernyataan,

atau deklarasi identitas yang terdiri dari IdP dan dipercaya oleh SP. IdP dan SP biasanya menyepakati terlebih dahulu informasi apa yang dibutuhkan SP. Namun, tetap ada mekanisme negosiasi ulang jika ada informasi tambahan (Naik & Jenkins, 2016). Kerangka kerja protokol SAML menggabungkan enam komponen utama, yaitu (OASIS, 2008):

- Security Assertions,
- Protocols,
- Bindings,
- Profiles,
- Metadata and,
- Authentication Context



Gambar 2.4 *Basic* SAML *Concept*

1. Security Assertions

Pertukaran informasi identitas dalam bentuk pernyataan keamanan (*security assertions*). *Assertion* (penegas) ini dengan sintaks yang ditentukan sebelumnya adalah kumpulan pernyataan tentang entitas. Ada tiga set pernyataan yang menyimpan pertukaran informasi, seperti otentikasi pengguna, hak, dan atribut. Pernyataan otentikasi pengguna menunjukkan apakah pengguna diautentikasi atau tidak. Jika autentikasi berhasil, ia harus memberikan detail metode autentikasi yang digunakan dan waktu autentikasi dari pihak SP dan IdP. Pernyataan hak menunjukkan apakah pengguna yang diautentikasi harus melakukan tindakan yang diinginkan. Keputusan tindakan ini disediakan melalui hak kontrol akses. Pernyataan atribut menunjukkan

apakah atribut yang ditentukan misalnya nama, usia, pekerjaan dan sebagainya, digunakan dalam pengambilan keputusan *access control* antara pihak yang terlibat.

2. Protocols

Protokol digunakan untuk mengeluarkan dan menukar pernyataan keamanan diatas dalam bentuk paket antara SP dan IdP. Protokol tersebut adalah *authentication request*, *request-response*, *assertion query*, *artifact resolution*, dan sebagainya. Setiap spesifikasi paket bervariasi dengan sekumpulan aturan dan elemen SAML seperti elemen permintaan dan respons yang harus diikuti oleh IdP atau SP.

3. Bindings

SAML *binding* adalah pemetaan pesan protokol diatas dengan format pesan standar dan/atau protokol komunikasi. Beberapa *binding* ini adalah SAML SOAP, HTTP *Redirect* (GET), HTTP *Post*, SAML URI, dan sebagainya.

4. Profiles

Profil SAML adalah deskripsi teknis tentang bagaimana pernyataan, protokol, dan SAML binding digabungkan untuk menangani kasus penggunaan tertentu dari suatu proyek atau skenario penerapan. Beberapa dari profil ini adalah SSO web browser, *single logout profile*, *enhanced client or proxy* (ECP), profil penemuan IdP, profil atribut SAML, dan sebagainya. Profil yang paling penting dan banyak digunakan adalah profil SSO Web browser. Misalnya, jika Profil SSO Web digunakan untuk autentikasi SAML, Profil ini merinci bagaimana pernyataan autentikasi dipertukarkan antarentitas (termasuk SAML *protocol* dan *binding* apa yang digunakan)

5. Metadata

Metadata SAML menentukan data konfigurasi dalam format skema XML dan menukar data ini antarentitas. Metadata mendefinisikan data seperti layanan apa yang tersedia, alamat, peran operasional (IdP atau SP), *binding*, sertifikat dengan informasi kunci untuk enkripsi dan penandatanganan, dan sebagainya. SP menggunakan metadata untuk mengetahui cara berkomunikasi dengan IdP dan sebaliknya

6. Authentication Context

SP meminta IdP untuk menggunakan beberapa jenis konteks autentikasi untuk pengguna. Informasi ini mencakup jenis dan kekuatan autentikasi. Hubungan antara komponen-komponen ini seperti blok bangunan seperti yang ditunjukkan pada Gambar di bawah ini. Ketika disatukan, memberikan fleksibilitas dan ekstensibilitas untuk mengimplementasikan sistem dan mendukung berbagai *business use case*.

2.1.6 *OAuth 2.0*

OAuth 2.0 adalah standar otorisasi, dimana kerangka otorisasi inti OAuth 2.0 dijelaskan oleh IETF di RFC-6749 (Hardt, 2012) bersama dengan beberapa spesifikasi dan profil lainnya. OAuth 2.0 memungkinkan delegasi akses, misalnya, jika seseorang menginginkan aplikasi lain mengakses fotonya di Google. OAuth 2.0 tidak seharusnya digunakan untuk kepentingan autentikasi. Namun, OAuth 2.0 dapat digunakan untuk autentikasi dengan beberapa fitur tambahan (seperti analogi lemari es dengan *freezer* tambahan). OpenID Connect (OIDC) adalah spesifikasi dari fitur ini. Jika sebuah aplikasi menggunakan OAuth 2.0 untuk SSO, biasanya yang dimaksud adalah aliran kode otorisasi OAuth 2.0 dengan OpenID Connect. Masih ada beberapa sistem *legacy* yang menggunakan OAuth 2.0 tanpa OIDC untuk *pseudo-authentication flow*, tetapi ini adalah hal yang sama sekali berbeda (Naik & Jenkins, 2017).

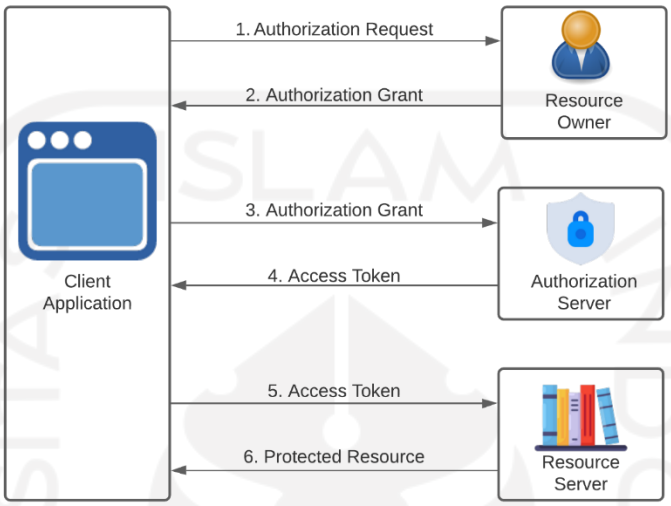
OpenID *Connect* digunakan secara luas tidak hanya di organisasi tetapi juga dalam skenario sehari-hari seperti "*Log in* dengan Google" atau "*Log in* dengan Facebook". Alur OAuth sangat berbeda dari aliran SAML. Perbedaan utamanya adalah bahwa beberapa tindakan terjadi di *browser* pengguna (*front channel*), dan beberapa terjadi secara langsung antara Aplikasi *Client* dan *Identity Provider* (*back channel*). Hal ini meminimalisir potensi serangan permukaan, karena *request* yang paling penting dikirim melalui *back channel* (Fett et al., 2016).

Terdapat empat peran (role) yang terlibat dalam OAuth 2.0 flow, yaitu (Hardt, 2012):

1. ***Resource Owner***, adalah pemilik *resource* (entitas atau orang) pada *resource server* yang mampu memberikan akses ke sumber daya yang dilindungi. Jika pemilik sumber daya adalah orang, maka disebut *end-user*.
2. ***Resource Server***, adalah server tempat *resource owner* menyimpan sumber dayanya yang dilindungi.
3. ***Client Application*** (aplikasi klien), adalah aplikasi pihak ketiga yang ingin mengakses sumber daya yang dilindungi di *resource server* atas nama *resource owner*. *Resource owner* harus memberikan izin yang diperlukan kepada aplikasi pihak ketiga (aplikasi klien) untuk mengakses sumber daya yang dilindungi. Spesifikasi OAuth 2.0 telah dirancang untuk profil klien berupa aplikasi web, aplikasi *user-agent-based*, dan aplikasi *native*.
4. ***Authorization Server***, adalah server yang mengautentikasi *resource owner* menggunakan kredensial mereka. *Authorization server* mengeluarkan *access token*

ke aplikasi klien setelah berhasil mengautentikasi *resource owner*. *Access token* ini adalah kredensial yang diberikan kepada aplikasi klien untuk mengakses sumber daya yang dilindungi *resource owner* di *resource server*.

Abstract flow pada Gambar 2.5 berikut menjelaskan interaksi antarperan dan membantu memahami cara kerja OAuth 2.0.



Gambar 2.5 OAuth 2.0 *Abstract Flow*[2]

1. *Client Application* mengirimkan permintaan otorisasi kepada *resource owner* (pengguna akhir) untuk mendapatkan akses ke sumber daya yang dilindungi pengguna.
2. Jika *resource owner* belum diautentikasi dengan *authorization server*, dia akan diminta untuk masuk dan mengautentikasi dengan *authorization server*. Setelah autentikasi berhasil, pengguna harus mengotorisasi permintaan *client application*. Jika pengguna menyetujui permintaan otorisasi, *client application* menerima *authorization grant*.
3. Kemudian *client application* mengautentikasi dengan *authorization server* dan meminta token akses dari *authorization server* dengan memberikan *authorization grant*, yang diterima pada langkah terakhir.
4. *Authorization Server* memvalidasi *client application* dan pemberian otorisasi dan kemudian mengirimkan *access token* ke *client application.*
5. Kemudian *client application* meminta sumber daya dari *resource server* dengan mengirimkan *access token* yang diperoleh pada langkah sebelumnya. *Client*

[2] https://miro.medium.com/max/720/1*MvP9jc_0KvhU4ymMF2B4rw.webp

application dapat menggunakan *access token* ini beberapa kali hingga token kedaluwarsa.

6. Terakhir, *resource owner* memvalidasi *access token* yang diterima dan jika *access token* tersebut valid, mengembalikan sumber daya yang diminta ke *client application*.

Pada tahap *authorization grant*, yaitu kredensial yang digunakan oleh client untuk memperoleh *access token*, OAuth 2.0 mendefinisikan empat *grant type*, yaitu (Hardt, 2012):

1. *Authorization Code*

   Authorization code diperoleh dengan menggunakan server otorisasi sebagai perantara antara klien dan *resource owner*. Alih-alih meminta otorisasi langsung dari *resource owner*, klien mengarahkan *resource owner* ke server otorisasi yang pada gilirannya mengarahkan *resource owner* kembali ke klien dengan *authorization code*. Sebelum mengarahkan *resource owner* kembali ke klien dengan *authorization code*, server otorisasi mengautentikasi *resource owner* dan mendapatkan otorisasi. Karena *resource owner* hanya mengautentikasi dengan server otorisasi, kredensial *resource owner* tidak pernah dibagikan dengan klien.

   Authorization code memberikan beberapa manfaat keamanan yang penting, seperti kemampuan untuk mengautentikasi klien, serta transmisi *access token* langsung ke klien tanpa meneruskannya melalui agen pengguna *resource owner* dan berpotensi menyebarkannya kepada orang lain, termasuk *resource owner*.

2. *Implicit*

   Implicit grant adalah simplifikasi dari *authorization code* yang dioptimalkan untuk klien yang diterapkan di browser menggunakan bahasa seperti *JavaScript*. Dalam *flow implicit*, alih-alih mengeluarkan kode otorisasi kepada klien, klien diberikan *access token* secara langsung (sebagai hasil dari otorisasi *resource owner*). *Grant type* ini bersifat implisit, karena tidak ada kredensial perantara (seperti kode otorisasi) yang dikeluarkan (dan nantinya digunakan untuk mendapatkan *access token*). Saat mengeluarkan *access token* selama *flow implicit grant*, server otorisasi tidak mengautentikasi klien. Dalam beberapa kasus, identitas klien dapat diverifikasi melalui *redirect* URI yang digunakan untuk mengirimkan *access token* ke klien. *Access token* dapat diekspos ke *resource owner* atau aplikasi lain dengan akses ke agen pengguna *resource owner*.

Implicit grant meningkatkan daya tanggap dan efisiensi beberapa klien karena mengurangi jumlah perjalanan bolak-balik yang diperlukan untuk mendapatkan *access token*. Namun, kenyamanan ini harus dipertimbangkan terhadap implikasi keamanan dari penggunaan *implicit grant*, seperti kemungkinan manipulasi *redirect* URI dan modifikasi *access token*, khususnya ketika *grant type authorization code* tersedia.

3. *Resource Owner Password Credentials*

Kredensial kata sandi *resource owner* (yaitu, nama pengguna dan kata sandi) dapat digunakan secara langsung sebagai pemberian otorisasi untuk mendapatkan *access token*. Kredensial hanya boleh digunakan bila ada tingkat kepercayaan yang tinggi antara *resource owner* dan klien (misal klien adalah bagian dari sistem operasi perangkat atau *highly privileged application*), dan bila jenis pemberian otorisasi lainnya tidak tersedia (seperti kode otorisasi). Meskipun *grant type* ini memerlukan akses klien langsung ke kredensial *resource owner*, kredensial *resource owner* digunakan untuk satu permintaan dan ditukar dengan *access token*. *Grant type* ini dapat menghilangkan kebutuhan klien untuk menyimpan kredensial *resource owner* untuk penggunaan di masa mendatang, dengan menukar kredensial dengan *access token* atau *refresh token* yang berumur panjang.

4. *Client Credentials*

Kredensial klien (atau bentuk autentikasi klien lainnya) dapat digunakan sebagai pemberian otorisasi saat *scope* otorisasi terbatas pada sumber daya yang dilindungi di bawah kendali klien, atau pada sumber daya yang dilindungi yang sebelumnya diatur dengan *authorization server*. Kredensial klien digunakan sebagai pemberian otorisasi biasanya ketika klien bertindak atas namanya sendiri (klien juga merupakan *resource owner*) atau meminta akses ke sumber daya yang dilindungi berdasarkan otorisasi yang sebelumnya diatur dengan *authorization server*.

2.1.7 *Authorization*

Jika autentikasi mengonfirmasi bahwa pengguna yang masuk sesuai dengan yang diklaim, *authorization* atau otorisasi memberikan izin kepada pengguna tersebut untuk mengakses sumber daya tertentu. Meskipun autentikasi dan otorisasi mungkin terdengar mirip, keduanya adalah proses keamanan yang berbeda pada IAM. Proses otorisasi memutuskan pengguna mana atau aplikasi apa yang diizinkan untuk tampil pada sistem dan informasi identitas pengguna/aplikasi yang digunakan untuk memenuhi keputusan tersebut. Pada

sebagian besar sistem, tidak semua pengguna memiliki hak yang sama untuk melakukan tindakan tertentu (Cremonezi et al., 2020).

Jika otorisasi melibatkan penentuan kebijakan, *access control* membuat kebijakan tersebut berfungsi. Kedua fungsi ini tidak dapat dipertukarkan, namun saling melengkapi. Setelah menyelesaikan proses otorisasi, sistem mengetahui peran pengguna dan apa yang boleh dilihat. Sistem *access control* membuka akses terhadap aset, sehingga pengguna dapat melakukan pekerjaan yang perlu dilakukan. Mekanisme *access control* sangat penting untuk melindungi layanan tertentu dari akses yang tidak sah. Mekanisme *access control* menentukan pengguna mana dan dalam kondisi/kebijakan mana yang dapat mengakses layanan. Mekanisme *access control* terpusat menguntungkan organisasi dalam mengamankan informasi sensitif, mengurangi beberapa tugas manajemen dan keamanan. Beberapa mekanisme *access control* yang diusulkan diantaranya (Cremonezi et al., 2020):

1. *Discretionary Access control* (DAC) adalah mekanisme *access control* yang awalnya dikembangkan untuk sistem operasi, dan kemudian dialihkan untuk konteks IoT. Di DAC, pemilik perangkat IoT menentukan pengguna mana yang dapat mengaksesnya dan menentukan beberapa aturan akses, seperti operasi mana yang valid, dan jam berapa pengguna lain dapat mengaksesnya. Beberapa pendekatan telah diusulkan untuk mengimplementasikan DAC: matriks akses, tabel otorisasi, dan *Access Control List* (ACL). Secara umum, untuk satu perangkat, pendekatan ini memberi pemilik kendali penuh, mengidentifikasi siapa yang dapat mengaksesnya dan dalam kondisi apa operasi dapat diakses oleh mereka. Namun, jika pengguna adalah pemilik sejumlah besar perangkat, kurangnya administrasi terpusat dapat mengubah desain kondisi akses menjadi rumit dan proses audit menjadi rumit.
2. *Mandatory Access Control* (MAC) adalah model *access control* berdasarkan klasifikasi semua entitas di IAM. Dalam model ini, setiap pengguna (baik manusia maupun non-manusia) dan layanan memiliki label keamanan, yang mencerminkan sensitivitas informasi yang dapat mereka akses atau hasilkan. Label keamanan mencerminkan kepercayaan pengguna untuk tidak mengungkapkan informasi sensitif. Agar berfungsi dengan benar, model MAC menempatkan beberapa batasan untuk membatasi perubahan label, yang memungkinkan hanya sekelompok terbatas pengguna manusia untuk memodifikasi label keamanan objek. Untuk alasan ini, model MAC sulit dan mahal untuk diimplementasikan dan dipelihara, terutama dalam skenario dinamis yang membutuhkan lebih banyak fleksibilitas, misalnya, dalam keadaan darurat pasien, aplikasi layanan kesehatan harus menurunkan keamanan data pengguna untuk

memberikan respons yang lebih cepat. Namun, jika aplikasi menggunakan MAC, hanya sedikit orang yang dapat mengubah label keamanan, yang dapat membahayakan nyawa orang tersebut, karena profesional kesehatan tidak dapat menerima informasi ini tepat waktu.

3. *Role-Based Access Control* (RBAC) adalah salah satu model *access control* yang paling banyak digunakan. Setiap pengguna aplikasi memiliki peran, dan peran ini menentukan layanan dan operasi mana yang dapat dia akses. Dengan demikian, pengguna ditugaskan ke peran dan mewarisi izin yang diberikan ke peran itu. Peran juga dapat diatur dalam hierarki peran, yang menentukan bahwa suatu peran dapat mewarisi izin dari peran lain. Secara umum, model ini menyediakan manajemen akses yang efektif, tetapi hanya mendefinisikan peran statis yang telah ditentukan sebelumnya. Definisi peran ini sangat bergantung pada entitas terpusat, dan bergantung pada kerumitan aplikasi, jumlah peran dapat meningkat dengan cepat, menjadikan metode ini tidak layak dan berpotensi merepotkan bagi IAM

Access control Berbasis Atribut (ABAC) mirip dengan RBAC, namun lebih fleksibel. Alih-alih mendefinisikan peran, sekumpulan kebijakan menguji kondisi atribut, mengizinkan atau tidak mengakses beberapa layanan. Strategi ini menyediakan model *fine-grained access control*. Namun, ada pertanyaan seputar jumlah ideal kebijakan dan evaluasinya. Pertanyaan-pertanyaan ini menjadi lebih rumit ketika mereka berasumsi bahwa atribut ini dapat diberikan dari banyak sumber daya (misalnya, akses bergantung pada dua identitas pengguna) dan dapat berubah seiring waktu, yang menyebabkan masalah keamanan dan konsistensi.

2.1.8 Organisasi Akademik

Organisasi akademik yang akan menjadi subjek pada penelitian ini adalah perguruan tinggi di Yogyakarta. Pada 2022, terdapat 11 PTN dan 103 PTS di Yogyakarta dengan berbagai bentuk (LLDIKTI V). Berdasarkan Peraturan Menteri Pendidikan dan Kebudayaan Nomor 7 Tahun 2020 tentang Pendirian, Perubahan, Pembubaran Perguruan Tinggi Negeri (PTN), dan Pendirian, Perubahan, Pencabutan Izin Perguruan Tinggi Swasta (PTS), disebutkan bahwa PTN atau PTS dapat berbentuk:

1. Universitas;
2. Institut;
3. Sekolah tinggi;
4. Politeknik;

5. Akademi; atau
6. Akademi komunitas.

2.2 Penelitian Terdahulu

Penelitian terkait SAML dan OAuth 2.0 pada perguruan tinggi yang telah ada sebelumnya telah banyak dilakukan. Penggunaan SAML pada perguruan tinggi lebih difokuskan untuk kebutuhan integrasi aplikasi web dan federasi. Dalam hal ini perguruan tinggi memiliki kebutuhan kolaborasi antarinstitusi pendidikan maupun penelitian untuk saling berbagi akses terhadap *resource*. Penggunaan OAuth 2.0 pada perguruan tinggi lebih banyak untuk mengintegrasikan sistem IoT dan perangkat *mobile*. Tabel 2.1 menunjukkan penelitian terkait penggunaan SAML dan OAuth 2.0 di perguruan tinggi.

2.2.1 *Business case* SAML pada perguruan tinggi

Bagian ini akan menjelaskan pemanfaatan SAML pada tiga universitas, yaitu Universitas Kyushu, Universitas Aristotle Thessaloniki (AUTh), dan West University of Timișoara (WUT). Universitas Kyushu telah memiliki Identity Management berbasis LDAP sejak 2009, yang kemudian dikembangkan dengan *multifactor authentication* (MFA) Shibboleth IdP. Universitas Kyushu memanfaatkan Shibboleh yang berbasis protokol SAML untuk mencapai misi *security* (keamanan) dan *usability* (kenyamanan dan keandalan) layanan Web SSO, serta memanfaatkannya untuk mengintegrasikan layanan web *legacy*. Universitas Kyushu juga berpartisipasi dalam GakuNin yang juga menggunakan Shibboleth (Federasi akademik yang melibatkan universitas dan penerbit, sama seperti US InCommon dan UK Federation), sehingga mahasiswa dan anggota staf dapat menggunakan layanan federasi seperti E-Journal (Ito et al., 2013).

Sama halnya dengan Universitas Kyushu, AUTh dan WUT memanfaatkan SAML untuk mengintegrasikan berbagai aplikasi web. AUTh melakukan transisi ke eLearning kombinasi software komersial dan opensource (terutama Zoom dan Moodle). Moodle sebagai platform Learning Management System (LMS), terutama untuk distance learning, disesuaikan dengan kebutuhan pedagogis dengan menambahkan berbagai plugin. Namun masih ada keterbatasan penggunaan oleh user. Untuk pembelajaran *synchronous*, Zoom meeting dinilai lebih efektif dan *user-friendly*. Dengan SSO, semua kredensial sensitif seperti email akademik, nama pengguna, dan kata sandi dienkripsi, tidak ada yang disimpan di server Zoom tetapi di pusat data universitas (LDAP). Selain itu integrasi SSO

berdasarkan SAML digunakan untuk melindungi data akademik dari pencurian, kehilangan, pengumpulan tanpa izin, penggunaan, pengungkapan, penyalinan, modifikasi, pembuangan, dll (Kalfa et al., 2021).

Aplikasi pembelajaran yang digunakan di AUTh tidak jauh berbeda dengan di WUT. Sampai dengan 2021 WUT telah menggunakan Moodle selama 15 tahun. Moodle eksisting adalah versi 3.11.2, telah diupdate dengan 676 plugin dan 240 plugin tambahan agar kompatibel dengan versi Moodle saat ini. Selain Moodle, muncul berbagai teknologi yang mendukung pengajaran dan kurikulum digital, diantaranya Google Workspace, Office 365, h5p, Cisco Webex, dan anti-plagiarism platform Turnitin. Integrasi teknologi tersebut dilakukan menggunakan SAML SSO, sehingga user hanya cukup login sekali menggunakan akun resmi universitas untuk mengakses semua teknologi SaaS yang tersedia dan diizinkan (Iordan et al., 2021).

Berdasarkan pengalaman implementasi pada ketiga universitas diatas, SAML sangat lumrah digunakan pada skema web SSO dan federasi (Ito et al., 2013). SAML dapat mengintegrasikan aplikasi web, aplikasi web *legacy*, maupun SaaS dengan konfigurasi yang mudah (Iordan et al., 2021).

2.2.2 *Business case* OAuth 2.0 pada perguruan tinggi

Layanan pembelajaran seperti Moodle berbasis *cloud* dapat diotorisasi menggunakan OAuth 2.0 via jejaring sosial seperti Twitter, Facebook, dan Yahoo (Kumar & Sharma, 2016). Instituto Tecnológico Superi or Ibarra juga melakukan hal yang sama untuk mengintegrasikan *Virtual Learning Environment* (VLE) Moodle ke jejaring sosial Facebook dan Twitter (Juma et al., 2019). Hal ini memudahkan pengajar, administrator, dan mahasiswa untuk berkolaborasi tanpa perlu membuat akun baru pada LMS.

OAuth 2.0 menyajikan konsep *trust agent* untuk otorisasi aplikasi *native mobile* yang transparan dan aman ke ekosistem teknologi perguruan tinggi. Selain untuk meningkatkan pelindungan data dan privasi, juga untuk menciptakan pengalaman belajar yang *seamless* dan generalisasi autentikasi pada aplikasi *native* (Glahn & Mazza, 2018).

Variasi perangkat yang saling terhubung terus bertambah di bidang pembelajaran akademik berbanding lurus dengan ancaman keamanan dan kerentanan yang ditimbulkan pada IoT di universitas. Menurut Mawgoud et al. (2020), manajemen identitas adalah salah satu masalah IoT karena praktik keamanan yang buruk sering diterapkan. Sebagai contoh, penggunaan *machine-to-machine* (M2M) dan *clean textual content/Base64 encoded*

IDs/passwords with gadgets adalah kesalahan umum yang harus diganti dengan *controlled token* seperti *JSON web Tokens* (JWT) yang digunakan oleh kerangka otorisasi OAuth 2.0.

Jumlah perangkat IoT yang digunakan di perguruan tinggi terus bertambah, karena mobilitas, skalabilitas, dan kemudahan penggunaan (Arina, 2021). Sebagai contoh pada National Chiao Tung University Taiwan, memiliki IoT yang terdiri dari mesin *laundry*, mesin pengering, dan *air conditioner* (AC) yang terinstal pada asrama kampus. Khusus subsistem IoT AC, terdiri dari 1 IoT Gateway, 100 Access Point, dan 200 AC. OAuth 2.0 digunakan sebagai *framework access control* untuk mengintegrasikan sistem *remote* kontrol IoT dengan sistem autentikasi perguruan tinggi. Fokusnya adalah *secure access control* (memastikan keamanan koneksi perangkat pengguna dengan perangkat IoT) dan perlindungan privasi (Shieh et al., 2020). Adanya kampus pintar, laboratorium inovatif yang memungkinkan simulasi proses teknologi, juga kelas pintar yang menggunakan campuran teknologi untuk pendidikan profesional di Industri 4.0, mewakili masa depan pendidikan. Itulah mengapa analisis keamanan sistem IoT sangat penting untuk memastikan kerahasiaan, integritas, dan ketersediaan data akademik. Otorisasi OAuth 2.0 direkomendasikan sebagai salah satu mitigasi pada layer perangkat untuk membatasi akses ke perangkat IoT (Arina, 2021).

Tabel 2.1 *Business case* pemanfaatan SAML dan OAuth 2.0 pada perguruan tinggi

Judul Referensi	**Protokol**	**Penggunaan**	**Sumber**
Implementation and Operation of the Kyushu University Authentication System	SAML (Shibboleth)	Universitas Kyushu memasang server IdP Shibboleth pada tahun 2011, dan berpartisipasi dalam GakuNin (Federasi akademik yang melibatkan universitas dan penerbit, sama seperti US InCommon dan UK Federation), agar mahasiswa dan anggota staf dapat menggunakan layanan federasi seperti E-Journal. Untuk mengintegrasikan layanan web *legacy* dan mewujudkan multifaktor autentikasi, universitas memasang sistem SSO untuk meningkatkan *security* dan *usability*.	(Ito et al., 2013)
Creating Collaborative and Convenient Learning Environment Using Cloud-Based Moodle LMS:An Instructor and Administrator Perspective	OAuth2	Otorisasi OAuth2 via jejaring sosial seperti Twitter, Facebook, dan Yahoo dimanfaatkan untuk mengakses LMS berbasis cloud seperti Moodle. Hal ini memudahkan pengajar, administrator, dan mahasiswa untuk berkolaborasi tanpa perlu membuat akun baru pada LMS.	(Kumar & Sharma, 2016)
Integrating Native Mobile Apps into Institutional Educational-technology Ecosystems	OAuth2	Menyajikan konsep *trust agent* untuk akses *native mobile apps* yang transparan dan *secure* ke layanan federasi perguruan tinggi. Selain untuk meningkatkan pelindungan data dan privasi, juga untuk menciptakan *seamless learning experience* dan generalisasi autentikasi pada aplikasi *native*.	(Glahn & Mazza, 2018)
Integration and Evaluation of Social	OAuth2	Melakukan integrasi Virtual Learning Environment (VLE) Moodle ke jejaring sosial Facebook dan Twitter	(Juma et al., 2019)

Networks in Virtual Learning Environments: A Case Study		untuk meningkatkan komunikasi virtual antara dosen, mahasiswa dan tenaga administrasi pada " Instituto Tecnológico Superi or Ibarra " (Ecuador)	
Security Threats of Social Internet of Things in the Higher Education Environment	OAuth2	Manajemen identitas adalah salah satu masalah IoT karena praktik keamanan yang buruk sering diterapkan. Sebagai contoh, penggunaan *clean textual content/Base64 encoded IDs/passwords with gadgets* dan *machine-to-machine* (M2M) adalah kesalahan umum yang harus diganti dengan *controlled token* seperti JSON web Tokens (JWT) yang digunakan oleh kerangka otentikasi dan otorisasi OAuth/OAuth2.	(Mawgoud et al., 2020)
Analysis of IoT security issues used in Higher Education Institutions	OAuth2	Jumlah perangkat IoT yang digunakan di perguruan tinggi terus bertambah, karena mobilitas, skalabilitas, dan kemudahan penggunaan. Penggunaan IoT pada sektor akademik meliputi: - *IoT-based Smart classroom* - *IoT-based Smart lab* - *IoT-based Smart Campus*	(Arina, 2021)
OAuth-Based Access Control Framework for IoT Systems	OAuth2	OAuth2 digunakan sebagai *framework access control* untuk mengintegrasikan sistem *remote* kontrol IoT dengan sistem autentikasi perguruan tinggi National Chiao Tung University Taiwan. IoT terdiri dari *laundry machines, drying machines, air conditioners* yang terinstal pada asrama kampus. Khusus subsistem IoT AC, terdiri dari 1 IoT Gateway, 100 Access Point, 200 AC. Fokusnya adalah *secure access control* (memastikan keamanan koneksi perangkat pengguna dengan perangkat IoT) dan perlindungan privasi.	(Hammann et al., 2020)
Coping with the COVID-19 challenges in a comprehensive university: learning tools and procedures adopted by Aristotle University of Thessaloniki	SAML	Universitas Aristotle sudah menggunakan SSO berbasis SAML untuk mengautentikasi user pada berbagai aplikasi web. Dengan SSO, semua kredensial sensitif seperti email akademik, nama pengguna, dan kata sandi dienkripsi, tidak ada yang disimpan di server Zoom tetapi di pusat data universitas (LDAP). Dengan konfigurasi manual integrasi yang mudah, SAML juga digunakan untuk melindungi data akademik dari pencurian, kehilangan, pengumpulan tanpa izin, penggunaan, pengungkapan, penyalinan, modifikasi, pembuangan, dll.	(Kalfa et al., 2021)
Moodle Platform' Support in Digitizing the Academic Process. Case Study West University of Timișoara	SAML	Untuk mendukung kurikulum digital, muncul berbagai teknologi yang mendukung pengajaran, diantaranya Google Workspace, Office 365, h5p, Cisco Webex, anti-plagiarism *platform* Turnitin, dan Moodle. Integrasi teknologi tersebut dilakukan menggunakan SAML SSO, sehingga *user* hanya cukup login sekali menggunakan akun resmi universitas untuk mengakses semua teknologi SaaS yang tersedia dan diizinkan.	(Iordan et al., 2021)

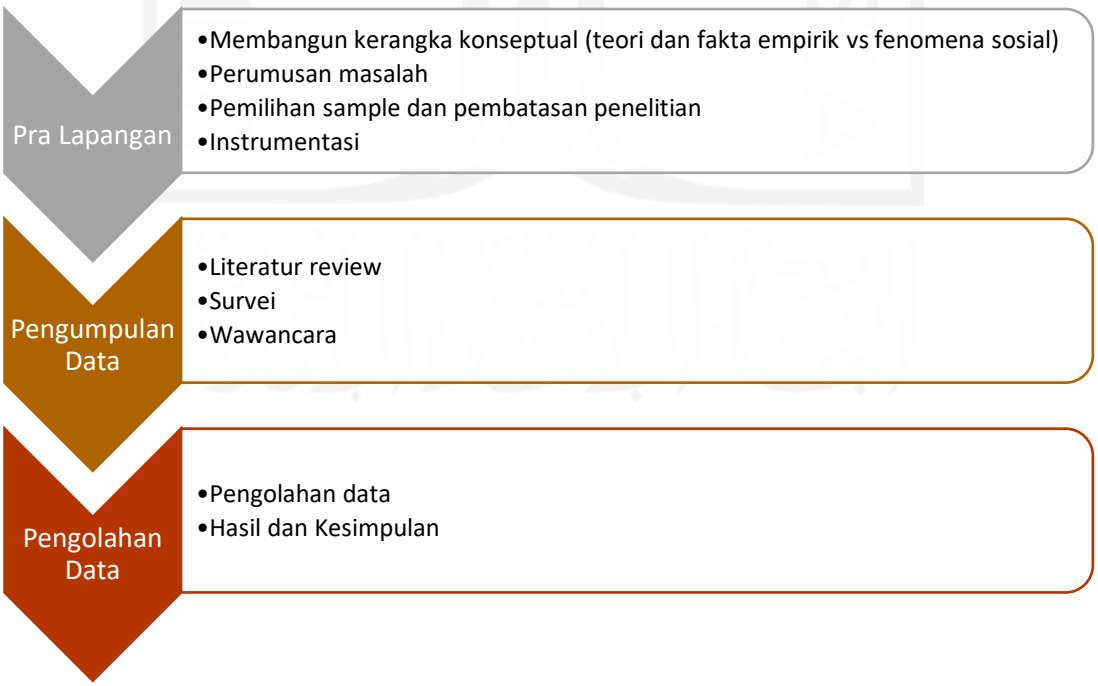
BAB 3

Metodologi

3.1 Pendekatan Penelitian

Untuk memahami tantangan yang dihadapi oleh perguruan tinggi saat mengimplementasikan solusi IAM dan SSO, penting untuk berinteraksi, memahami, dan menganalisis wawasan orang yang bekerja di platform tersebut. Berdasarkan penelitian sebelumnya terkait IAM dan SSO, memberikan wawasan tentang metode dan analisis pengumpulan data. Penelitian ini akan menggunakan metode kualitatif untuk pengumpulan data. Metode pengumpulan data kualitatif melibatkan literatur review, survei, dan wawancara individu (Creswell, 2013). Survei dan wawancara akan dilakukan secara online dengan narasumber personil Pusat IT perguruan tinggi di Yogyakarta.

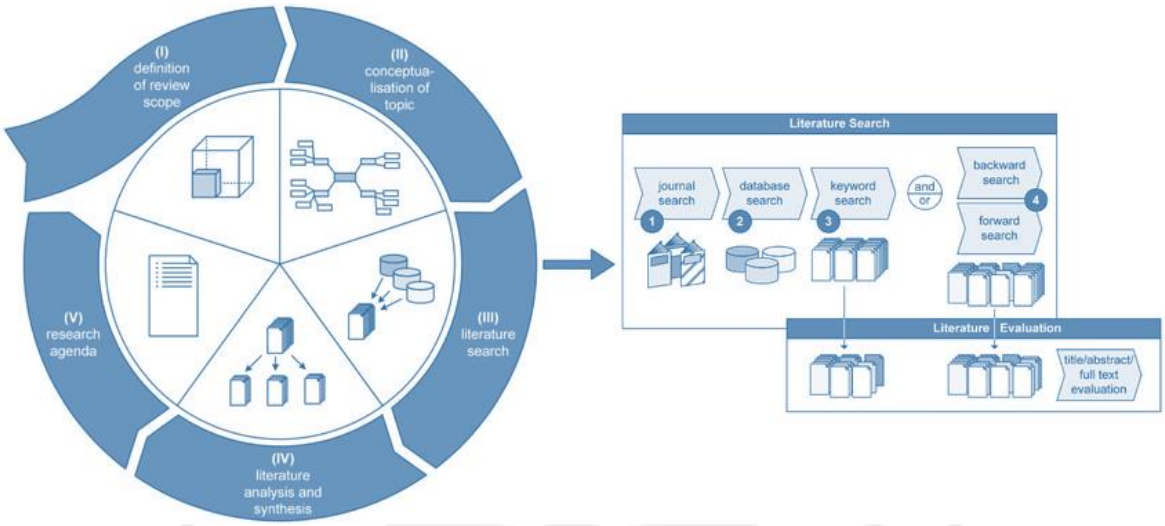
Selain survei dan wawancara, penulis juga meninjau berbagai penelitian yang telah dilakukan sebelum penelitian ini dan menganalisis hasil dan keluaran dari penelitian tersebut. Survei dan wawancara dilakukan pada perguruan tinggi yang sudah mengimplementasikan solusi IAM dan SSO, maupun yang belum atau berencana menerapkan beberapa bentuk solusi SSO dan IAM. Hal ini dimaksudkan untuk memberikan lebih banyak wawasan tentang tantangan yang harus dilalui perguruan tinggi dalam memilih solusi yang tepat.



Gambar 3.1 Tahapan penelitian

3.2 Literatur Review

Kerangka metodologi tinjauan pustaka ini dimotivasi dan diadaptasi dari karya (vom Brocke et al., 2009). Kerangka kerja ini dilakukan dalam rangkaian 5 fase yang bersifat siklik. Tahapannya adalah sebagai berikut: definisi ruang lingkup review, konseptualisasi topik penelitian, pencarian literatur, analisis literatur yang diperoleh dan penentuan agenda penelitian. Penjelasan rinci tentang bagaimana tinjauan literatur dilakukan pada setiap fase dijelaskan pada sub-bagian berikut.



Gambar 3.2 Framework literatur review Vom Brocke et al.

3.2.1 Definisi ruang lingkup review

Ruang lingkup kajian pustaka adalah:

1. Untuk mempelajari protokol yang populer digunakan dalam sistem manajemen identitas (SAML, OAuth 2.0, OpenID Connect) dan membuat analisis komparatif untuk menjelaskan perbedaan fiturnya.
2. Untuk mempelajari adopsi SAML dan OAuth 2.0 pada perguruan tinggi.

3.2.2 Konseptualisasi topik penelitian

Untuk memulai topik penelitian, dilakukan pencarian literatur untuk menemukan artikel dengan menggunakan istilah-istilah kunci yang relevan. Istilah kunci awal yang digunakan untuk pencarian adalah sebagai berikut: SAML, OAuth2, *Identity and Access Management, Single Sign-On*, *Authentication*, *Authorization*, RBAC. Studi pendahuluan dari artikel ini digunakan untuk membantu menurunkan istilah kunci yang lebih luas seperti adopsi SAML pada perguruan tinggi, adopsi OAuth2 pada perguruan tinggi, IMS,

lingkungan multi/hibrid, autentikasi SAML, autentikasi OIDC, *on-premises*, implementasi SAML, implementasi OAuth2, integrasi aplikasi RBAC dengan SSO.

3.2.3 Pencarian literatur

Kata kunci di atas dicari pada *database* yang diterbitkan, menggunakan kriteria pencarian khusus untuk menemukan artikel yang relevan, ditunjukkan pada Tabel 3.1.

Tabel 3.1 Kriteria pencarian dan *database* yang digunakan

Kriteria Pencarian	**Database**
Artikel yang diterbitkan pada:	Google Scholar
- Jurnal, Conference, website dan buku	ACM digital library
- Rentang waktu antara 2016-2022	Science Direct

Artikel hasil pencarian kemudian dipelajari berdasarkan judul, abstrak dan kesimpulan. Jika artikel cukup sesuai, maka dilakukan pencarian lebih lanjut berdasarkan kutipan dan referensi yang digunakan dalam artikel tersebut.

Penelitian terkait implementasi protokol SSO SAML dan OAuth 2.0 pada perguruan tinggi terdiri dari: (1) kolaborasi dan federasi, (2) integrasi dengan sistem berbasis web pada perguruan tinggi, (3) integrasi dengan perangkat IoT dan aplikasi *native*, (4) perbandingan fitur. Penelitian terkait RBAC lebih banyak membahas modifikasi untuk meningkatkan performa dan mengatasi keterbatasan model *access control* ini. Tabel 3.2 menunjukkan jumlah artikel yang dicari dan digunakan dalam penelitian.

Tabel 3.2 Topik dan jumlah artikel yang digunakan

Topik	Jumlah artikel
Perbandingan fitur SAML dan OAuth 2.0	6
Implementasi SAML pada perguruan tinggi	6
Implementasi OAuth 2.0 pada perguruan tinggi	7
Role-based access control (RBAC)	7

3.2.4 Analisis literatur yang diperoleh

1. Perbandingan fitur SAML dan OAuth 2.0

Integrasi pada lingkungan domain yang berbeda memerlukan protokol autentikasi standar untuk membangun komunikasi yang aman antara pihak yang terlibat. Bagian ini menyajikan analisis komparasi fitur dari tiga protokol SSO yaitu SAML, OAuth 2.0, dan OpenID Connect (OIDC) (Naik & Jenkins, 2017) seperti yang ditunjukkan pada Tabel 3.3.

Dari analisis ini disimpulkan bahwa SAML memiliki keterbatasan pada perangkat mobile dan IoT. OAuth 2.0 merupakan protokol otorisasi yang mengizinkan delegasi akses, tidak untuk kebutuhan autentikasi. OAuth 2.0 dapat digunakan untuk autentikasi dengan beberapa fitur tambahan (seperti analogi lemari es dengan *freezer* tambahan). OIDC adalah spesifikasi dari fitur ini. OIDC telah dikembangkan untuk memberikan layanan untuk web, *cloud*, perangkat *mobile*, dan IoT.

Tabel 3.3 Perbandingan fitur SAML dan OAuth 2.0

Fitur	SAML	OAuth 2.0	Sumber
Authentication	✓		(Naik & Jenkins, 2016) (Naik & Jenkins, 2017) (Indu et al., 2018) (Carretero et al., 2018) (Aldosary & Alqahtani, 2021)
Authorization	✓	✓	
Access Delegation		✓	
Support Web App	✓	✓	
Support Native Mobile App		✓	
Mobile Standard (IoT)		✓	
Flexibility (ease of use)		✓	
Interoperability	✓	✓	
Single Sign-On	✓		
Single Logout	✓	✓	
Attribute/ Claims	✓		
Federation	✓		
Extensibility	✓	✓	
Open Standard	✓	✓	

2. Implementasi SAML pada perguruan tinggi

Dalam pengaturan paling dasar (dan paling populer), SAML memungkinkan pertukaran informasi tentang pengguna antara IdP dan SP melalui *browser* pengguna (User Agent). Hal ini tidak memerlukan koneksi langsung antara IdP dan SP – semuanya terjadi melalui *redirect browser*. Pengguna meminta dokumen yang ditandatangani yang mengonfirmasi identitas mereka dari IdP, diautentikasi, dan kemudian mengirimkan dokumen yang ditandatangani ke SP untuk login. Dokumen yang ditandatangani juga dapat menyertakan informasi tentang *privileges* pengguna, grup, dll (Naik & Jenkins, 2017).

SAML digunakan untuk menyediakan pusat manajemen identitas dan mengintegrasikan *resource* pada perguruan tinggi, terutama yang berbasis web. Universitas Aristotle sudah

menggunakan SSO berbasis SAML untuk mengautentikasi *user* pada berbagai aplikasi web, seperti Moodle dan Zoom (Kalfa et al., 2021). West University of Timișoara juga mengintegrasikan teknologi SaaS yang terdiri dari Google Workspace, Office 365, h5p, Cisco Webex, anti-plagiarism platform Turnitin, dan Moodle untuk mendukung kurikulum digital (Iordan et al., 2021).

Kolaborasi ilmiah menyatukan peneliti dan infrastruktur lintas institusi akademik, penyedia layanan *cloud*, dan batas internasional. Meski demikian, pada prakteknya sering terbatas pada mekanisme autentikasi dan otorisasi. Ketersediaan federasi identitas, memungkinkan peneliti mengakses infrastruktur siber menggunakan kredensial yang telah dimiliki, tanpa harus membuat kredensial baru (Basney et al., 2019). SAML banyak digunakan pada perguruan tinggi untuk berkolaborasi antarinstitusi di berbagai negara. Universitas Kyushu menggunakan Shibboleth untuk berkolaborasi dalam federasi akademik GakuNin (Ito et al., 2013). CILogon dibangun di atas perangkat lunak Shibboleth dan COmanage *open source* menyediakan platform IAM terintegrasi untuk sains, yang digabungkan ke seluruh dunia melalui eduGAIN. CILogon melayani kebutuhan unik kolaborasi penelitian, yaitu untuk secara dinamis membentuk grup kolaborasi lintas organisasi dan negara (Basney et al., 2019).

3. Implementasi OAuth 2.0 pada perguruan tinggi

OAuth biasanya digunakan untuk mendelegasikan akses ke sesuatu. Kita dapat memperbolehkan seseorang untuk "bertindak" sebagai kita. Pada kasus otorisasi antaraplikasi, OAuth digunakan untuk memberikan akses ke API yang dapat melakukan sesuatu sebagai akun kita (Naik & Jenkins, 2017). Manajemen identitas adalah salah satu masalah IoT karena praktik keamanan yang buruk sering diterapkan. Sebagai contoh, penggunaan *clean textual content*/Base64 *encoded* IDs/*passwords with gadgets* dan *machine-to-machine* (M2M) adalah kesalahan umum yang harus diganti dengan *controlled token* seperti JSON Web Tokens (JWT) (Mawgoud et al., 2020).

Perguruan tinggi memiliki berbagai macam *resource* yang sebagian ataupun keseluruhannya diintegrasikan menggunakan OAuth 2.0 (Ueda & Ikeda, 2017) (Glahn & Mazza, 2018) (Juma et al., 2019), karena dinilai lebih fleksibel (Naik & Jenkins, 2017) dan mendukung integrasi aplikasi *native* (Glahn & Mazza, 2018) serta perangkat IoT (Ueda & Ikeda, 2017) (Arina, 2021) (Hammann et al., 2020). Perangkat IoT yang digunakan pada perguruan tinggi terus bertambah, karena mobilitas, skalabilitas, dan kemudahan penggunaan. IoT pada sektor akademik diantaranya IoT-*based Smart classroom*, IoT-

based Smart lab, dan IoT-*based Smart Campus* (Arina, 2021). Studi kasus kampus digital di Jepang memiliki *platform* yang terdiri dari tablet PC untuk setiap mahasiswa, akses ke jaringan Wi-Fi, LMS, groupware, SNS, data *opensource* terbaru, berbagai kursus *online* gratis yang tersedia secara global, jam tangan, dan kacamata pintar. Akses ke semua platform pendidikan dan kesehatan tersebut dilakukan menggunakan otorisasi OAuth 2.0 (Ueda & Ikeda, 2017).

4. *Role-based access control* (RBAC)

Selama 30 tahun sejak konsep RBAC diperkenalkan oleh Ferrailo & Khun pada 1992, telah banyak modifikasi yang dilakukan oleh para peneliti untuk meningkatkan performa dan mengatasi keterbatasan model *access control* ini. Modifikasi dilakukan dengan mengembangkan RBAC tradisional, maupun menggabungkan dengan model lain (*hybrid*). Xu et al. mengusulkan skema kombinasi *identity-based cryptosystem* (IBC) dan *upgrade* RBAC untuk *chipertext* pada penyimpanan *cloud*, tujuannya untuk mendukung perubahan kebijakan *access control* yang dinamis dengan biaya komputasi rendah (Xu et al., 2021). Rao et al. mengusulkan model rekomendasi role pada RBAC untuk mengoptimalkan *user-role assignment* berdasarkan pola perilaku pengguna, model ini digunakan pada *cloud Role-Assignment-as-a-Service* guna optimalisasi biaya *built-in role* (Rao et al., 2021).

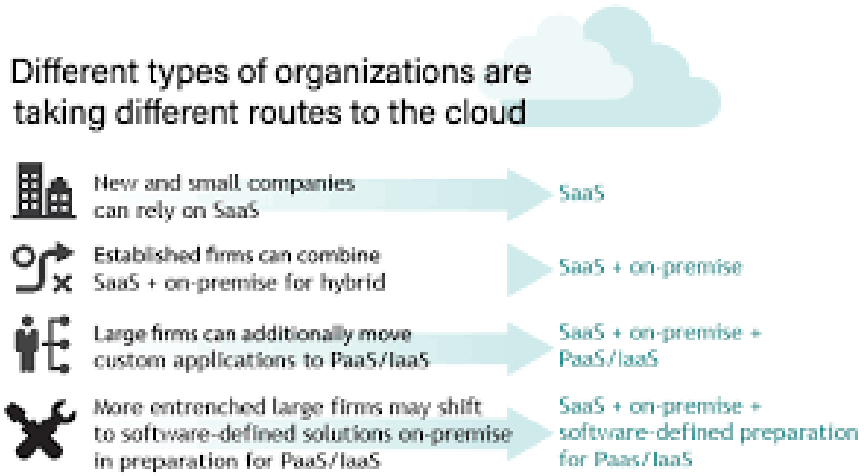
RBAC merupakan *access control* yang diterapkan secara luas di banyak *platform cloud*. Namun saat jumlah *role* meningkat, kompleksitasnya pun meningkat (Soni & Kumar, 2019). Keterbatasan utama RBAC adalah bahwa *role* yang ditetapkan dapat berubah dari waktu ke waktu dan perubahan tersebut harus divalidasi secara *real-time* (Indu et al., 2018). Untuk menutupi keterbatasan tersebut, beberapa peneliti (Aftab et al., 2020) (Aftab, Qin, Hundera, et al., 2019) (Aftab, Qin, Quadri, et al., 2019) mengusulkan model *hybrid access control* yang menggabungkan RBAC dan ABAC yang memiliki lebih banyak fleksibilitas.

RBAC direkomendasikan sebagai mekanisme otorisasi untuk organisasi, memastikan onboarding karyawan ke struktur organisasi yang menghemat biaya dan waktu orientasi. Untuk integrasinya dengan SSO, Triartono dkk mengusulkan model untuk mengimplementasikan RBAC pada OAuth 2.0 menggunakan *framework* Laravel, kemudian menguji kinerja sistem berikut dengan waktu respons, transfer data, dan *throughput* (Triartono & Negara, 2019). Penelitian terkait lainnya ada Se-Ra Oh yang berfokus pada interoperabilitas kerangka kerja OAuth 2.0 eksisting. Makalah ini mengusulkan lapisan otorisasi tambahan untuk OAuth 2.0 yang dapat digunakan sebagai

dasar untuk memisahkan antara otorisasi dan layanan dalam arsitektur *microservices* (Oh & Kim, 2019).

5. Implementasi *Identity Access Management* (IAM) dan SSO pada organisasi secara umum

Menurut IDC OPINION, semakin besar organisasi dan semakin teregulasi industrinya, semakin besar kemungkinan perusahaan memiliki sistem IAM *on-premises*. Singkatnya, pengguna organisasi berada pada tahap adopsi *cloud* dan implementasi IAM yang berbeda, seperti pada Gambar 3.3. Banyak yang memiliki sistem dan *software on-premises*, dalam waktu yang bersamaan juga menggunakan SaaS, PaaS atau IaaS. Beberapa terhubung dengan mitra dan pelanggan dalam skenario B2B dan B2C. Organisasi lainnya memiliki lingkungan SaaS murni, namun ingin menambahkan autentikasi yang kuat, SSO, dan perlindungan data, termasuk perluasan bisnis ke *social identity* (Christiansen & Stuart, 2016).



Gambar 3.3 Pemanfaatan cloud pada berbagai level organisasi

Organisasi diatas mungkin berada pada berbagai tahap penerapan solusi IAM sebagai berikut:

a. Tidak memiliki solusi IAM terintegrasi dan bergantung pada staf internal untuk menambahkan dan menghapus pengguna pada aplikasi perusahaan.
b. Perusahaan bahkan mungkin tidak memiliki kemampuan autentikasi untuk memverifikasi identitas pengguna yang mengakses aplikasi dan data perusahaan.
c. Organisasi mungkin belum mengimplementasikan SSO, agar pengguna dapat mengakses aplikasi SaaS dan aplikasi organisasi dengan satu *username* dan *password*.

d. Organisasi mungkin memiliki manual, sertifikasi akses dan alur persetujuan, *segregation of duties* (SOD) yang mendetail, atau bahkan mungkin tidak memiliki sama sekali.
e. Seringkali organisasi mungkin tidak memiliki kemampuan khusus untuk memastikan identitas dan kontrol pengguna "*privileged*", seperti administrator organisasi dan staf TI yang perlu memiliki level akses yang lebih tinggi.

Beberapa organisasi mungkin telah menginstal solusi IAM *on-premises* atau berbasis SaaS (IDaaS) yang cukup komprehensif.

3.2.5 Penetapan agenda penelitian

Berdasarkan analisis tinjauan pustaka, dapat dirangkum beberapa poin penting dalam implementasi SAML dan OAuth 2.0 di lingkungan *on-premises* dan *cloud*. Penelitian-penelitian tersebut mencakup analisis komparasi fitur dari tiga protokol SSO yaitu SAML, OAuth 2.0, dan OIDC, serta teknis integrasi dengan aplikasi RBAC *on-premises* dan integrasi dengan aplikasi SaaS. Terkait implementasinya pada perguruan tinggi, SAML dan OAuth 2.0 banyak digunakan untuk mengintegrasikan berbagai *resource* yang dimiliki perguruan tinggi agar aksesnya *seamless*, meskipun digunakan pada perangkat yang beragam. Fokus utamanya ada pada *usability*, baru kemudian *security*.

Penelitian ini akan mengevaluasi implementasi SSO protokol SAML dan OAuth 2.0 pada perguruan tinggi di Yogyakarta dan mengidentifikasi keterbatasan yang dimiliki oleh perguruan tinggi yang belum memanfaatkan SSO.

3.3 Survei dan Wawancara

Survei dan wawancara adalah sumber data utama untuk penelitian yang diusulkan. Selain untuk membandingkan implementasi protokol SAML dan OAuth 2.0, survei dan wawancara dapat menjadi pedoman yang dapat dipertimbangkan saat menerapkan IAM dan SSO. Pertanyaan survei dan wawancara lebih difokuskan pada kebutuhan organisasi, langkah yang telah dilakukan, dan tantangan yang dihadapi dengan solusi tersebut. Pertanyaan-pertanyaan ini dimaksudkan untuk menganalisis keadaan yang berbeda di dalam perguruan tinggi sebelum membuat kesimpulan.

Survei dan wawancara yang akan dilakukan juga akan memiliki pertanyaan yang terkait infrastruktur organisasi serta keefektifan solusi IAM dan SSO untuk menyediakan solusi atas permasalahan organisasi. Tanggapan yang diperoleh dari survei dan wawancara

dibandingkan dengan berbagai konsep dan *use case* SSO di perguruan tinggi pada penelitian yang lalu.

Proses survei dan wawancara melibatkan 20 orang yang bekerja di 17 Pusat IT perguruan tinggi Yogyakarta, khususnya IAM dan SSO. Analisis didasarkan pada umpan balik survei dan wawancara serta tinjauan literatur. Data dikumpulkan berdasarkan survei dan wawancara yang dilakukan dan dievaluasi berdasarkan klasifikasi perguruan tinggi yang telah disebutkan pada bagian sebelumnya.

BAB 4
Hasil dan Pembahasan

4.1 Klasifikasi Perguruan Tinggi

Klasifikasi perguruan tinggi ditentukan berdasarkan infrastruktur organisasi. Praktik yang direkomendasikan untuk diikuti dapat bervariasi berdasarkan klasifikasi dan kebutuhan perguruan tinggi. Klasifikasi perguruan tinggi dikelompokkan menjadi kecil, sedang, dan besar berdasarkan jumlah aplikasi yang dimiliki (Joshi et al., 2018):

a. Kecil, jumlah aplikasi kurang dari 50
b. Sedang, jumlah aplikasi kurang dari 150 dan lebih dari 50
c. Besar, jumlah aplikasi lebih dari 150

Arsitektur dalam suatu organisasi dapat dikategorikan berdasarkan apakah aplikasi mereka diinstall di *cloud, hibrid* atau pada jaringan sendiri yang juga dikenal sebagai *on-premises*.

a. *On-premises*
b. *Cloud*
c. Hibrid

Arsitektur tersebut meningkatkan kompleksitas dalam mengelola pengguna dan memberikan akses kepada pengguna karena meningkatnya risiko privasi dan keamanan. Berdasarkan kompleksitas pengelolaannya dikelompokkan berdasarkan variasi *resource* yang dikelola, termasuk jika terdapat aplikasi *legacy* dan SaaS yang dikelola akan menambah kompleksitas arsitektur perguruan tinggi. Kompleksitas pengelolaan *resource* perguruan tinggi pada penelitian ini terbagi menjadi sederhana, sedang, dan kompleks:

a. Sederhana, jika mengelola minimal 1 sampai 2 jenis *resource*
b. Sedang, jika mengelola 3 sampai 4 jenis *resource*
c. Kompleks, jika mengelola lebih dari 4 jenis *resource*

4.2 Proses Analisis Data

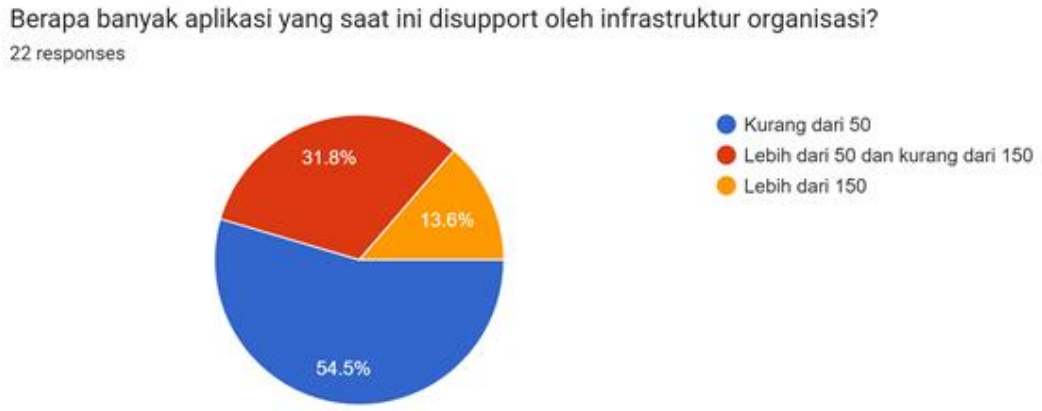
Kuisioner terdiri dari 37 pertanyaan yang dibagi menjadi 3 (tiga) kategori, yaitu (1) Identifikasi Infrastruktur dan Kebutuhan Organisasi; (2) Implementasi SSO Eksisting; (3) Roadmap/ Rencana Implementasi SSO. Proses analisis data seperti yang dijelaskan di bawah ini:

a. Memahami klasifikasi dan kebutuhan perguruan tinggi yang disurvei saat ini
b. Menganalisis kondisi eksisting dan isu-isu yang dihadapi oleh perguruan tinggi yang telah menerapkan solusi IAM dan SSO
c. Menganalisis kemungkinan kendala isu yang dihadapi oleh perguruan tinggi saat hendak merealisasikan solusi IAM dan SSO
d. Membandingkan umpan balik dari survei dengan *business case* penelitian terdahulu

4.3 Hasil Penelitian

Hasil penelitian didasarkan pada survei dan wawancara yang dilakukan terhadap 22 personil Pusat IT dari 17 perguruan tinggi Yogyakarta. Terdapat 3 (tiga) perguruan tinggi dengan jumlah responden lebih dari satu. Proses survei tidak membatasi jumlah responden per perguruan tinggi, sehingga memungkinkan terdapat lebih dari satu responden dalam satu perguruan tinggi. Hal ini memungkinkan adanya perbedaan jawaban pada beberapa pertanyaan, dikarenakan perbedaan pemahaman/ persepsi responden. Hasil penelitian yang disajikan di bawah ini telah dianalisis dan dibandingkan dengan penelitian sebelumnya.

4.3.1 Infrastruktur dan kebutuhan organisasi



Gambar 4.1 Jumlah aplikasi yang dikelola

Berdasarkan survei yang dilakukan dapat disimpulkan bahwa sebagian besar perguruan tinggi masuk dalam klasifikasi kecil dengan mengelola kurang dari 50 aplikasi, yaitu sebesar 54.5% (Gambar 4.1). Hanya 13.6% responden dari 2 perguruan tinggi yang masuk dalam klasifikasi besar dengan mengelola lebih dari 150 aplikasi.

Aplikasi diatas merupakan bagian dari *resource* yang dikelola oleh perguruan tinggi. Penelitian ini mengelompokkan *resource* menjadi lima jenis, yaitu jaringan/Wifi, email, IoT, aplikasi *web-based*, dan aplikasi *native*. Gambar 4.2 menunjukkan *resource* yang saat ini dikelola oleh perguruan tinggi yang disurvei. Berdasarkan kompleksitasnya, 81.8% perguruan tinggi memiliki kompleksitas sedang dengan mengelola 3 sampai dengan 4 jenis *resource*, selebihnya yaitu 18.2% termasuk klasifikasi kompleks dengan mengelola lebih dari 4 *resource*. Perguruan tinggi dengan klasifikasi kompleks tersebut juga mengelola aplikasi *legacy* dan SaaS yang diintegrasikan dengan SSO SAML.

Gambar 4.2 Jenis *resource* yang dikelola

Menurut respon survei, berbagai *resource* yang dimiliki perguruan tinggi di Yogyakarta keseluruhannya dikelola secara terpusat pada arsitektur *on-premises* (50%), hibrid (45.5%), dan *cloud* (4.5%). Domain yang dikelola pun beragam, meliputi:

a. Administasi perguruan tinggi (Sistem Akuntansi, dll)
b. Layanan kemahasiswaan (KRS, KHS, dll)
c. Layanan Pembelajaran (Moodle, Google Classroom, dll)
d. Online Meeting (Zoom, Webex, Microsoft Teams, dll)

11 dari 22 responden tercatat mengelola seluruh domain aplikasi diatas. Meskipun ada yang hanya mengelola satu atau dua kategori, namun Layanan Kemahasiswaan selalu ada. Terkait kebutuhan fitur SSO, 45.5% responden memerlukan pendefinisian atribut disisi pengguna. Ini merupakan salah satu fitur SAML untuk memudahkan admin IT disaat terdapat perubahan atribut pengguna. Admin IT hanya perlu menyesuaikan atribut pengguna, karena akses ke aplikasi akan mengikuti atribut yang melekat pada pengguna. Namun hal ini tidak sejalan dengan kebutuhan kolaborasi, dimana terdapat 13 responden

(59.1%) yang memiliki rencana untuk berkolaborasi dengan perguruan tinggi lain. Saat perguruan tinggi berkolaborasi dengan perguruan tinggi lain, akan ada *resource* atau data yang dipertukarkan. Penggunaan protokol yang mendukung prinsip federasi sangat disarankan.

4.3.2 Implementasi SSO

Gambar 4.3 Perguruan tinggi sudah menerapkan SSO

Penelitian ini menganalisa implementasi SSO pada 10 dari 22 responden (Gambar 4.3) berdasarkan umpan balik survei dan wawancara dari 20 pertanyaan. Hasilnya dapat disimpulkan sebagai berikut:

a. 7 perguruan tinggi memiliki rencana untuk berkolaborasi dengan perguruan tinggi lain. Dari ketujuh perguruan tinggi tersebut, dua diantaranya masih menggunakan OAuth 2.0 dan sudah memiliki rencana untuk melakukan evaluasi. Satu dari dua perguruan tinggi tersebut sudah melakukan riset dan sudah menentukan akan menggunakan kombinasi SAML dan OAuth 2.0 sesuai kasus yang ada.
b. Dilihat dari protokol yang diterapkan, 6 perguruan tinggi melakukan riset terlebih dahulu untuk menentukan yang paling sesuai dengan kebutuhan organisasi; 3 perguruan tinggi menerapkan protokol yng dianggap populer dilingkungan perguruan tinggi; dan 1 perguruan tinggi sisanya mendapatkan rekomendasi dari pihak ketiga. Terlepas dari motivasi dan upaya penerapannya, berdasarkan hasil observasi kesepuluh perguruan tinggi ini memiliki tim IT yang

memadai untuk mengelola, memelihara, dan terus mengembangkan *tools* yang telah diterapkan.

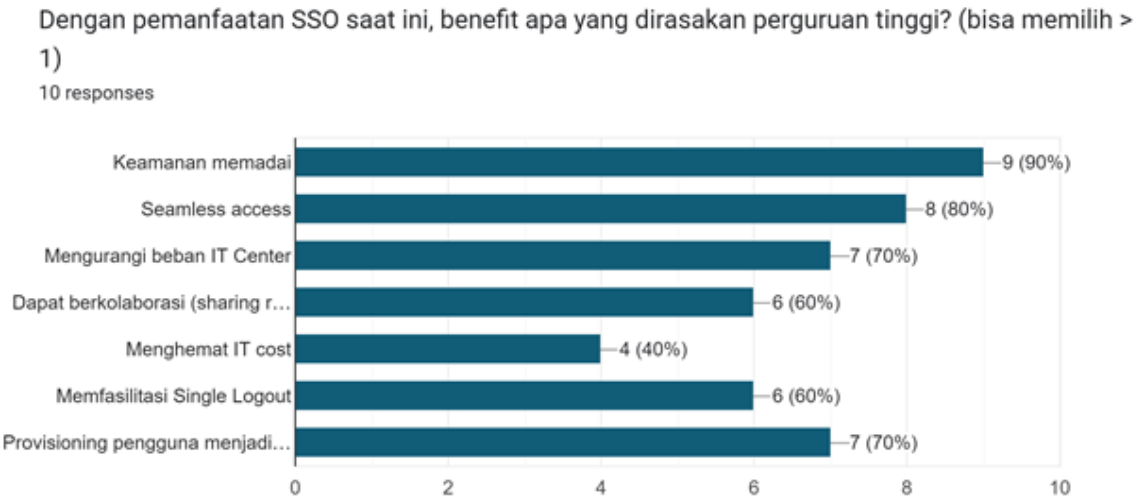
c. Dari sisi pengembangan, semua perguruan tinggi yang melakukan riset sebelum memilih protokol yang hendak digunakan, membangun layanan SSO nya secara mandiri karena menilai bahwa implementasinya cukup mudah. Meskipun masih terdapat 4 perguruan tinggi yang belum sepenuhnya mengintegrasikan semua resource dengan SSO.
d. Diantara domain aplikasi yang diintegrasikan ke SSO, domain Layanan Kemahasiswaan menjadi prioritas perguruan tinggi. Disusul oleh Layanan Administrasi perguruan tinggi, dan Layanan Pembelajaran.
e. Perguruan tinggi klasifikasi besar selalu memiliki arsitektur yang kompleks, memiliki infrastruktur *on-premises,* dan menggunakan protokol SAML untuk mengintegrasikan *resource*-nya. Berdasarkan hasil survei, perguruan tinggi ini memiliki *resource* yang beragam, masih memiliki aplikasi *legacy* dan juga memanfaatkan layanan SaaS. Namun perguruan tinggi ini juga tidak mengalami kesulitan saat harus mengintegrasikan kedua jenis aplikasi tersebut ke SSO.
f. Pada perguruan tinggi klasifikasi sedang dan kecil, terdapat 2 responden yang menggunakan layanan SaaS mengintegrasikannya dengan OAuth 2.0. Berdasarkan penelitian sebelumnya, baik SAML maupun OAuth 2.0 sangat memungkinan untuk mengintegrasikan aplikasi *legacy* dan aplikasi SaaS.
g. Terdapat 7 perguruan tinggi yang mengintegrasikan aplikasi *native* (*desktop-based/mobile-based*) dan IoT menggunakan SAML, hal ini tidak sejalan dengan *business case* SAML.

1. Keuntungan memanfaatkan SSO

   Diantara keuntungan pemanfaatan SSO yaitu:

   a. Keamanan memadai (Indu et al., 2018)
   b. *Seamless access* (Indu et al., 2018)
   c. Mengurangi beban IT *Center* (Joshi et al., 2018)
   d. Dapat berkolaborasi (*sharing resource*) dengan perguruan tinggi lain (Indu et al., 2018) (Cremonezi et al., 2020)
   e. Menghemat IT *cost* (Joshi et al., 2018)
   f. Memfasilitasi *Single Logout* (Aldosary & Alqahtani, 2021)
   g. Provisioning pengguna menjadi lebih cepat (Joshi et al., 2018)

Gambar 4.4 berikut merupakan hasil survei terhadap 9 perguruan tinggi yang telah memanfaatkan SSO. Keamanan, kecepatan *provisioning*, dan *seamless access* menjadi hal yang paling dirasakan.



Gambar 4.4 Keuntungan memanfaatkan SSO

2. Kendala yang masih dihadapi dengan implementasi SSO eksisting

Teknis implementasi SSO di lapangan sangat bervariasi. Pada perguruan tinggi yang telah menerapkan SSO sesuai spesifikasi protokol dan *best practice*, lebih banyak keuntungan yang di-*checklist* daripada kendala, bahkan dapat dikatakan tidak ada kendala berarti. Namun disamping manfaat yang dirasakan dari penerapan SSO, beberapa perguruan tinggi memiliki beberapa kendala yang masih dialami. Sebagai contoh pada 5 perguruan tinggi yang sudah mengintegrasikan semua domain aplikasi ke SSO, 3 diantaranya masih membuka fitur login aplikasi bawaan. Hal ini menyebabkan pengguna memiliki kecenderungan melakukan login dari fitur login aplikasi bawaan dan pemanfaatan SSO menjadi tidak maksimal. Dari sisi pengguna, SSO tidak meringkas jumlah kredensial yang digunakan, sebaliknya malah menambah jumlah kredensial. Dampaknya, permintaan *reset password* tidak berkurang. Berikut beberapa kendala yang masih dialami oleh perguruan tinggi yang telah memanfaatkan SSO, sebagian diambil dari referensi, sebagian merupakan kendala hasil observasi yang dialami di lapangan disebabkan implementasi yang tidak sesuai *best practice*. Prosentase kendala hasil survei dapat dilihat pada Gambar 4.5.

a. Masih banyak permintaan *reset password* dikarenakan lupa *password* akun aplikasi maupun akun SSO
b. Pengguna masih cenderung menggunakan fitur *login* masing-masing aplikasi, karena masih terdapat pilihan login menggunakan SSO dan login bawaan aplikasi
c. Pengelolaan hak akses belum dilakukan secara terpusat, masih dilakukan pada masing-masing aplikasi
d. Manajemen data pengguna belum terpusat (*Provisioning* akun pengguna memakan waktu lama karena harus memberikan/menghapus akses pada masing-masing aplikasi terkait sesuai *role*-nya.) (Joshi et al., 2018)
e. Masih rentan terjadi kebocoran data
f. Kesulitan saat mengintegrasikan dengan aplikasi *on-premises* (aplikasi *legacy*)
g. Implementasinya cukup kompleks
h. Tidak dapat berkolaborasi/ sharing *resource* dengan perguruan tinggi lain (Indu et al., 2018)
i. Tidak dapat diintegrasikan dengan aplikasi *native* (*mobile-based*/*desktop-based*) dan IoT, kalaupun bisa akan membutuhkan banyak kustomisasi yang tidak sesuai *best practice* (Joshi et al., 2018)(OASIS, 2008)
j. Tidak mendukung *Single Logout* (Joshi et al., 2018)
k. Tidak ada kendala berarti

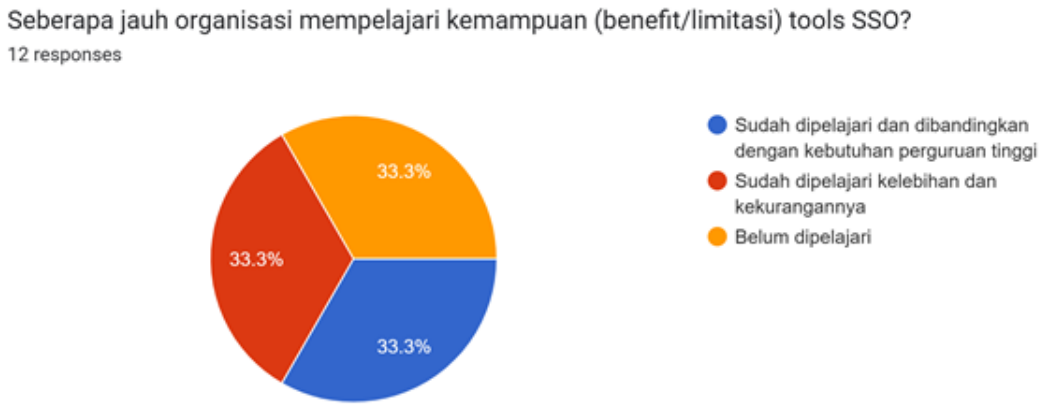
Gambar 4.5 Kendala SSO eksisting

Dari kendala-kendala yang masih dialami pada pemanfaatan SSO eksisting, 70% perguruan tinggi memiliki rencana untuk mengevaluasi dan menyesuaikan protokol SSO yang digunakan. Namun hanya 40% saja yang telah menentukan hendak menggunakan protokol apa, selebihnya masih akan melakukan riset kembali. Gambar 4.6 menunjukkan prosentase tersebut.

Gambar 4.6 Rencana penggunaan protokol

4.3.3 Rencana pengembangan SSO (*Roadmap*)

Selain perguruan tinggi yang telah menggunakan SSO, survei ini juga ditujukan untuk perguruan tinggi yang belum menerapkan SSO. Terdapat 12 responden yang 58.3% diantaranya telah memiliki roadmap implementasi SSO, namun 75% diantaranya belum menentukan protokol apa yang hendak digunakan. Gambar 4.7 menunjukkan ragam tahap yang telah dilakukan untuk mengimplementasikan protokol SSO.

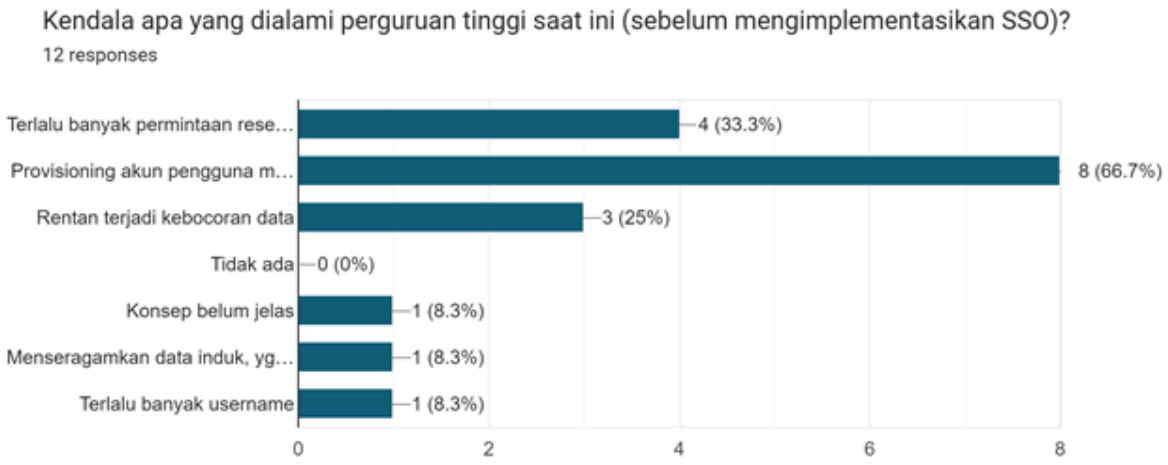


Gambar 4.7 Tahap menuju implementasi SSO

Selama belum memanfaat SSO, berikut beberapa kendala yang teridentifikasi:

1. Terlalu banyak permintaan *reset password*
2. Provisioning akun pengguna memakan waktu lama karena harus memberikan akses pada masing-masing aplikasi terkait sesuai *role*-nya.
3. Rentan terjadi kebocoran data

Kendala yang paling banyak dialami tim IT adalah terkait provisioning akun pengguna yaitu sebesar 66.7% seperti yang terlihat pada Gambar 4.8.



Gambar 4.8 Kendala pengelolaan IT

Meski demikian, sebagian perguruan tinggi belum mengimplementasikan SSO karena berbagai kendala. Kendala terbesar adalah keterbatasan SDM, baik karena minim SDM yang capable untuk mengimplementasikan SSO (75%) maupun keterbatasan update pengetahuan (66.7%). Kendala adopsi SSO ditunjukkan pada Gambar 4.9.

Gambar 4.9 Kendala pemanfaatan SSO

4.3.4 Tantangan Implementasi IAM dan SSO

Berikut merupakan tantangan yang dihadapi terkait implementasi solusi IAM dan SSO berdasarkan survei yang dilakukan dan penelitian sebelumnya:

1. Kurangnya penelitian yang dilakukan untuk mempelajari apakah solusi yang hendak digunakan dapat memenuhi kebutuhan/ kriteria organisasi (Joshi et al., 2018)
2. Kompleksitas arsitektur dan aplikasi yang perlu diintegrasikan dengan solusi IAM dan SSO (Indu et al., 2018)
3. Memahami kelemahan masing-masing solusi IAM dan SSO, untuk mengetahui pengaruhnya terhadap infrastruktur organisasi (Indu et al., 2018)
4. SDM yang berwawasan dan *resource* yang memadai untuk mendukung implementasi *tools* (Joshi et al., 2018)

4.3.5 *Best Practice* penerapan SSO

Setelah menganalisis hasil survei dan wawancara, beberapa *best practice* yang dapat diikuti selama penerapan IAM dan SSO adalah sebagai berikut:

1. Perguruan tinggi harus memahami kondisi infrastruktur dan kebutuhan organisasi dan kapasitas SDM, agar dapat digunakan sebagai pertimbangan dalam pemilihan protokol yang sesuai (Joshi et al., 2018)
2. Perguruan tinggi harus melakukan penelitian terhadap protokol yang diterapkan dan memahami kompleksitasnya (Indu et al., 2018)

3. Perguruan tinggi harus memiliki sumber daya yang memadai dan terlatih untuk mengelola solusi SSO (Joshi et al., 2018)
4. Perguruan tinggi harus memahami kerentanan yang ada pada protokol SSO dan memiliki tim serta sumber daya untuk mengurangi kerentanan pada tools tersebut (Indu et al., 2018)

BAB 5
Kesimpulan dan Saran

5.1 Kesimpulan

Penelitian ini menyajikan perbandingan fitur SAML dan OAuth 2.0 sebagai protokol SSO yang populer digunakan pada berbagai organisasi, termasuk perguruan tinggi. Penelitian ini juga menganalisis *business case* SAML dan OAuth 2.0 di perguruan tinggi, sebagai dasar untuk menganalisis penerapannya pada perguruan tinggi Yogyakarta. Penelitian dilakukan melalui survei dan wawancara untuk mengidentifikasi infrastruktur, kebutuhan, dan praktik penerapan SSO yang dilakukan oleh perguruan tinggi tersebut.

Praktik penerapan SSO SAML dan OAuth 2.0 terhadap kebutuhan organisasi pada perguruan tinggi Yogyakarta, berdasarkan hasil survei dan wawancara dapat disimpulkan sebagai berikut:

1. Protokol SSO yang digunakan beragam, ada yang menerapkan SAML, OAuth 2.0, dan kombinasi keduanya.
2. Terdapat ketidaksesuaian penerapan pada 7 perguruan tinggi yang mengintegrasikan aplikasi *native* (*desktop-based/mobile-based*) dan IoT menggunakan SAML, hal ini tidak sejalan dengan *business case* dan spesifikasi SAML yang tidak direkomendasikan untuk aplikasi *native* dan IoT.
3. Terdapat ketidaksesuaian penerapan protokol yaitu OAuth 2.0 pada 2 perguruan tinggi yang memiliki kebutuhan kolaborasi dengan perguruan tinggi lain, hal ini tidak sejalan dengan spesifikasi OAuth 2.0 yang tidak mendukung fitur tersebut. Namun kedua perguruan tinggi tersebut sudah memiliki rencana evaluasi terhadap protokol yang digunakan.
4. Terdapat keterbatasan praktik implementasi SSO yang menyebabkan masih munculnya kendala-kendala yang seharusnya dapat teratasi dengan adanya penerapan SSO.
5. SAML dan OAuth 2.0 telah berhasil digunakan untuk mengintegrasikan berbagai *resource* perguruan tinggi, termasuk aplikasi SaaS dan aplikasi *legacy*.
6. Ditemukan jawaban yang berbeda dari responden perguruan tinggi yang sama pada kasus responden perguruan tinggi lebih dari satu. Berdasarkan konfirmasi, perbedaan pemahaman kondisi IT perguruan tinggi menjadi salah satu penyebabnya. Hal ini

disebabkan terdapat responden yang bukan merupakan tokoh inti dalam penerapan SSO. Selain itu, tidak semua personil pada Pusat IT difokuskan untuk mengelola IT perguruan tinggi, melainkan hanya diperbantukan dan masih memiliki tanggung jawab utama pada unit lain.

7. Tantangan penerapan SSO beserta rekomendasi *best practice* pada penelitian terdahulu telah disampaikan.

Penelitian tidak hanya dibatasi pada perguruan tinggi yang telah menerapkan SSO, namun juga melibatkan perguruan tinggi yang belum menerapkannya. Tujuannya untuk mengidentifikasi infrastruktur, kebutuhan, langkah yang telah dilakukan untuk menuju penerapan SSO, dan kendala yang dialami dalam prosesnya. Berdasarkan hasil survei dan wawancara, terlihat bahwa lebih banyak perguruan tinggi yang belum menerapkan SSO (10:12) karena berbagai kendala, termasuk 3 responden yang mengklaim belum adanya kebutuhan SSO. Terlepas dari kendala yang ada, 9 responden lainnya telah menyadari permasalahan yang dihadapi dan mengusahakan solusi dengan menyusun roadmap terkait rencana penerapan SSO. Meskipun hanya 3 dari 9 responden yang telah menentukan protokol yang hendak digunakan. Hal ini sudah cukup untuk menggambarkan antusiasme perguruan tinggi Yogyakarta dalam pengembangan teknologi untuk menuju iklim kolaborasi.

5.2 Saran

Penelitian ini memiliki keterbatasan dikarenakan metode survei dan wawancara dilakukan secara online, keterbatasan tersebut antara lain:

1. Perbedaan persepsi atas pertanyaan yang disajikan juga berpengaruh terhadap tanggapan yang diterima.
2. Para peserta hanya diwawancarai berdasarkan pertanyaan survei dan tidak ada umpan balik atau pertanyaan lain yang mungkin dilakukan selain dari pertanyaan survei, namun demikian penulis masih dapat mengkonfirmasi atas respon yang diterima.
3. Penelitian terbatas pada 22 responden dari 17 perguruan tinggi, masih terdapat 97 perguruan tinggi lainnya yang belum terjangkau pada penelitian ini.

Oleh karena itu, pada penelitian selanjutnya dapat dikembangkan pada 97 perguruan tinggi lainnya dan dilakukan dengan metode yang lebih intensif, sehingga dapat menggali informasi yang lebih lengkap dan tanggapan yang dirasa kurang sesuai dapat terkonfirmasi dengan baik. Selain itu, dengan metode semacam ini, penulis dapat memberikan pertanyaan lain yang mendukung diluar pertanyaan yang telah disusun.

Daftar Pustaka

Aftab, M. U., Munir, Y., Oluwasanmi, A., Qin, Z., Aziz, M. H., Zakria, Son, N. T., & Tran, V. D. (2020). A Hybrid Access Control Model with Dynamic COI for Secure Localization of Satellite and IoT-Based Vehicles. *IEEE Access*, *8*, 24196–24208. https://doi.org/10.1109/ACCESS.2020.2969715

Aftab, M. U., Qin, Z., Hundera, N. W., Ariyo, O., Zakria, Son, N. T., & van Dinh, T. (2019). Permission-based separation of duty in dynamic role-based access control model. *Symmetry*, *11*(5). https://doi.org/10.3390/sym11050669

Aftab, M. U., Qin, Z., Quadri, S. F., Zakria, Javed, A., & Nie, X. (2019). Role-based ABAC model for implementing least privileges. *ACM International Conference Proceeding Series, Part F1479*, 467–471. https://doi.org/10.1145/3316615.3316667

Aldosary, M., & Alqahtani, N. (2021). A Survey on Federated Identity Management Systems Limitation and Solutions. *International Journal of Network Security & Its Applications*, *13*(03), 43–59. https://doi.org/10.5121/ijnsa.2021.13304

Arina, A. (2021). Analysis of IoT security issues used in Higher Education Institutions. *International Journal of Mathematics and Computer Research*, *09*(05). https://doi.org/10.47191/ijmcr/v9i5.01

Basney, J., Flanagan, H., Fleury, T., Gaynor, J., Koranda, S., Oshrin, B., Sinica, A., & Taipei, T. (2019). *CILogon: Enabling Federated Identity and Access Management for Scientific Collaborations*. https://www.shibboleth.net/

Bazaz, T., & Khalique, A. (2016). A Review on Single Sign on Enabling Technologies and Protocols. *International Journal of Computer Applications*, *151*(11), 975–8887. https://doi.org/10.5120/ijca2016911938

Carretero, J., Izquierdo-Moreno, G., Vasile-Cabezas, M., & Garcia-Blas, J. (2018). Federated identity architecture of the European eID System. *IEEE Access*, *6*, 75302–75326. https://doi.org/10.1109/ACCESS.2018.2882870

Christiansen, C. A., & Stuart, L. (2016). *Identity as a Service on the Journey to the Cloud IDC OPINION*.

Costello, K., & Rimol, M. (2021, April 21). *Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 23% in 2021*. Https://Www.Gartner.Com/En/Newsroom/Press-Releases/2021-04-21-Gartner-Forecasts-Worldwide-Public-Cloud-End-User-Spending-to-Grow-23-Percent-in-2021.

Cremonezi, B., Vieira, A., Nacif, J. A., & Nogueira, M. (2020). *Survey on Identity and Access Management for Internet of Things*. https://doi.org/10.21203/rs.3.rs-66793/v1

Creswell, J. W. (2013). *John W. Creswell - Research Design_ Qualitative, Quantitative, and Mixed Method Approaches-SAGE Publications (2013)*.

Fett, D., Kuesters, R., & Schmitz, G. (2016). *A Comprehensive Formal Security Analysis of OAuth 2.0*. http://arxiv.org/abs/1601.01229

Glahn, C., & Mazza, R. (2018). Integrating Native Mobile Apps into Institutional Educational-technology Ecosystems. *17th World Conference on Mobile and Contextual Learning*, 77–83.

Hammann, S., Sasse, R., & Basin, D. (2020). Privacy-Preserving OpenID Connect. *ASIA CCS '20: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. https://doi.org/https://doi.org/10.1145/3320269.3384724

Hardt, D. Ed. (2012). The OAuth 2.0 Authorization Framework. *RFC 6749*. http://www.rfc-editor.org/info/rfc6749.

Indu, I., Anand, P. M. R., & Bhaskar, V. (2018). Identity and access management in cloud environment: Mechanisms and challenges. *Engineering Science and Technology, an International Journal*, *21*(4), 574–588. https://doi.org/10.1016/j.jestch.2018.05.010

Iordan, M., Bîzoi, G., Bîzoi, A.-C., & Herman, C. (2021). *MOODLE PLATFORM' SUPPORT IN DIGITIZING THE ACADEMIC PROCESS. CASE STUDY WEST UNIVERSITY OF TIMIȘOARA*. https://www.researchgate.net/publication/361440775

Ito, E., Kasahara, Y., & Fujimura, N. (2013). Implementation and operation of the kyushu university authentication system. *Proceedings ACM SIGUCCS User Services Conference*, 137–142. https://doi.org/10.1145/2504776.2504788

Joshi, U., Cha, S., & Esmaili-Sardari, S. (2018). Towards adoption of authentication and authorization in identity management and single sign on. *Advances in Science, Technology and Engineering Systems*, *3*(5), 492–500. https://doi.org/10.25046/aj030556

Juma, A., Rodríguez, J., Caraguay, J., Naranjo, M., Quiña-Mera, A., & García-Santillán, I. (2019). Integration and evaluation of social networks in virtual learning environments: A case study. *Communications in Computer and Information Science*, *895*, 245–258. https://doi.org/10.1007/978-3-030-05532-5_18

Kalfa, V., Roussos, G., Charidimou, D., & Agorogianni, A. (2021). Coping with the COVID-19 challenges in a comprehensive university: learning tools and procedures adopted by Aristotle University of Thessaloniki. *Proceedings of the European*

University Information Systems Conference 2021. https://doi.org/https://doi.org/10.29007/hhvq

Kumar, V., & Sharma, D. (2016). Creating Collaborative and Convenient Learning Environment Using Cloud-Based Moodle LMS: An Instructor and Administrator Perspective. *International Journal of Web-Based Learning and Teaching Technologies*, *11*(1), 35–50. https://doi.org/10.4018/IJWLTT.2016010103

Mawgoud, A. A., Taha, M. H. N., & Khalifa, N. E. M. (2020). Security Threats of Social Internet of Things in the Higher Education Environment. In *Studies in Computational Intelligence* (Vol. 846, pp. 151–171). Springer Verlag. https://doi.org/10.1007/978-3-030-24513-9_9

Naik, N., & Jenkins, P. (2016). An Analysis of Open Standard Identity Protocols in Cloud Computing Security Paradigm. *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 85–85. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2016.85

Naik, N., & Jenkins, P. (2017). Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect. *2017 11th International Conference on Research Challenges in Information Science (RCIS)*. https://doi.org/10.1109/RCIS.2017.7956534

OASIS. (2008). *Security Assertion Markup Language (SAML) V2.0 Technical Overview*. http://docs. oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

Oh, S.-R., & Kim, Y.-G. (2019). Interoperable OAuth 2.0 Framework. *International Conference on Platform Technology and Service (PlatCon)*.

Rao, K. R., Nayak, A., Ray, I. G., Rahulamathavan, Y., & Rajarajan, M. (2021). Role recommender-RBAC: Optimizing user-role assignments in RBAC. *Computer Communications*, *166*(February 2020), 140–153. https://doi.org/10.1016/j.comcom.2020.12.006

Shieh, M.-Z., Liu, J.-C., Kao, Y.-C., Tsai, S.-C., & Lin, Y.-B. (2020). OAuth-Based Access Control Framework for IoT Systems. *4th EAI International Conference, SGIoT 2020*, *354*, 208–219. http://www.springer.com/series/8197

Soni, K., & Kumar, S. (2019). Comparison of RBAC and ABAC Security Models for Private Cloud. *Proceedings of the International Conference on Machine Learning,*

Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019, 584–587. https://doi.org/10.1109/COMITCon.2019.8862220

Triartono, Z., & Negara, R. M. (2019). Implementation of Role-Based Access Control on OAuth 2.0 as Authentication and Authorization System. *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. https://doi.org/https://doi.org/10.23919/EECSI48112.2019.8977061

Ueda, T., & Ikeda, Y. (2017). Socio-economics and educational case study with cost-effective IOT campus by the use of wearable, tablet, cloud and open E-learning services. *Japan Society for the Promotion of Science KAKENHI*.

vom Brocke, J., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Brocke, J., Plattfaut, R., & Cleven, A. (2009). Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. *17th European Conference on Information Systems (ECIS)*. https://aisel.aisnet.org/ecis2009/161

Xu, J., Yu, Y., Meng, Q., Wu, Q., & Zhou, F. (2021). Role-Based Access Control Model for Cloud Storage Using Identity-Based Cryptosystem. In *Mobile Networks and Applications* (Vol. 26, Issue 4, pp. 1475–1492). https://doi.org/10.1007/s11036-019-01484-4