Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.201961679
Vol. 6, No. 5, Oktober 2019, hlm. 535-542 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

EKSPLORASI ABAC DAN XACML UNTUK *DESIGN ACCESS CONTROL* PADA *RESOURCE DIGITAL*

Fauzan Natsir[1], Imam Riadi [2], Yudi Prayudi [3]

[1,3] Department of Informatics, Universitas Islam Indonesia, Indonesia, [2]Department of Information System, Universitas Ahmad Dahlan Yogyakarta, Indonesia
Email : [1]fauzan.natsir@gmail.com, [2]imam.riadi@is.uad.ac.id, [3] prayudi@uii.ac.id

(Naskah masuk: 21 Januari 2019, diterima untuk diterbitkan: 25 April 2019)

Abstrak

Resource digital memerlukan sebuah mekanisme untuk mengatur *policy* terhadap kontrol untuk mendapatkan hak akes ke dalam suatu sistem. Akses kontrol lebih fleksibel dibanding dengan pendekatan otorisasi, autentikasi ataupun verifikasi yang sangat sederhana. Mekanisme *access control policy* dengan pendekatan atribut diyakini sebagai solusi adaptif yaitu ABAC (*Attribute Based Access Control*) dengan implementasi model XACML (*Extensible Access Control Modelling Language*). Desain *policy* ABAC ini disajikan dengan atribut-atribut dari salah satu studi kasus *resource digital* dengan sistem *e-Library*. *e-Library* merupakan salah satu resource digital dimana proses autentikasinya belum dimodelkan dengan atrubut subjek yang ada. Penelitian ini diawali dari identifikasi atribut dari *rule*, pemodelan ABAC *resource digital*, implementasi XACML, simulasi sistem dan analisis sistem. Hasil dari pengujian akses kontrol menggunakan *ALFA (Axiomatics Language for Authorization)* untuk pemberian kinerja akses kontrol terhadap *resource digital*. Hasil analisis dengan pendekatan ABAC dengan model XACML ini menyajikan suatu keamanan sistem dengan model akses kontrol berbasis atribut dari *policy statement* untuk menjadi solusi model akses kontrol yang dibuat sebelumnya dan mendukung model akses kontrol yang relevan untuk *resource digital*

Kata kunci: *Access Control, Resource Digital, ABAC, XACML, ALFA*

Abstract

Digital resources require a mechanism to regulate policy against controls to get access rights to a system. Access control is more flexible than the very simple approach of authorization, authentication or verification. The access control policy with the attribute approach is believed to be an adaptive solution, namely ABAC (Attribute Based Access Control) with the implementation of the XACML (Extensible Access Control Modeling Language) model. This ABAC policy design is presented with attributes from one of the digital resource case studies with the e-Library system. e-Library is one of the digital resources where the authentication process has not been modeled with the existing subject matter. This study begins with the identification of the attributes of the rule, digital ABAC resource modeling, XACML implementation, system simulation and system analysis. The results of testing access control using ALFA (Axiomatics Language for Authorization) to provide performance control access to digital resources. The results of the analysis using the ABAC approach with the XACML model present a system security with attribute-based access control models from policy statements to be a solution to the previously created access control model and support the access control model relevant for digital resources

Keywords: *Access Control, Digital Resource, ABAC, XACML, ALFA*

1. PENDAHULUAN

Resource digital tidak hanya mencakup atribut atau identitas, namun tidak pula terbatas pada komputer *file* (seperti *file* log atau dihasilkan laporan) dan *file* yang dihasilkan manusia (seperti *spreadsheet*, dokumen, atau pesan *email*). Hal yang harus diperhatikan berikutnya adalah bagaimana menjaga atau mengatur akses terhadap *resource digital* sehingga dapat dijaga dengan baik, Riadi, (2018). Pendekatan yang umumnya dilakukan adalah menggunakan skema atau mekansime autentikasi dan otorisasi Solusi terhadap *resource digital* di antaranya menggunakan otorisasi, autentikasi dan verifikasi yang sangat sederhana dan terbatas. Proses ini bisa menerapkan ketentuan yang lebih fleksibel untuk akses *resource bukti digital*. Dengan autentikasi dan otorisasi, ada keterbatasan dalam pengaksesan untuk mengontrol pemodelan *resource digital*. Salah satu solusi untuk mengatasi permasalahan tersebut melalui pendekatan *access control policy* yang memungkinkan mekanisme akses terhadap *resource*

digital menjadi lebih fleksibel dan lebih kompleks sesuai dengan kebutuhan interaksi yang terjadi. Di antara sekian banyak model untuk *access control policy*, salah satu di antaranya adalah model ABAC(*Attribute Based Access Control*).

Model ini berbasiskan pada verifikasi atribut dan diyakini akan menjadi model *access control* yang adaptif terhadap kebutuhan *access policy* terhadap berbagai *resource digital* di masa yang akan datang. Sementara itu untuk kepentingan implementasi dari *access control policy* dari ABAC dikembangkan bahasa pemodelan XACML (*Extensible Access Control Modelling Language*). Sejauh ini penerapan ABAC dan XACML sebagai sebuah sistem untuk akses terhadap *resource digital* masih sangat terbatas

Penerapan ABAC dan XACML sebagai sebuah sistem untuk akses terhadap *resource digital* masih sangat terbatas. Sejumlah penelitian yang ada antara lain pernah dilakukan oleh Varadharajan, (2015). Namun pada penelitian tersebut model yang diterapkan adalah model *Next Generation Access Control* (NGAC) dengan implementasi pada XML serta penelitian selanjutnya model yang diterapkan adalah model *Role Based Access Control* (RBAC) dengan implementasi pada *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE). Penelitian yang dilakukan oleh Panende, (2018) mendukung metode ABAC menjadi solusi atas permasalahan *acces control* sebelumnya.

Eksplorasi tentang ABAC dan XACML terhadap akses *resource digital* memerlukan penelitian pemodelan ABAC pada akses *resource digital* dengan implementasi kinerja ABAC dan XACML dibandingkan dengan pendekatan otorisasi, autentikasi, dan verifikasi yang umumnya dipakai selama ini, Subektingsih, (2018). Pendekatan ini sangat cocok dengan menggunakan pendekatan atribut akses kontrol atau ABAC. Oleh karena itu, maka belum ada kajian tentang bagaimana penerapan ABAC dengan implementasi XACML terhadap *resource digital* sehingga lebih perlu dikaji lebih lanjut harapannya menguatkan sistem keamanan terhadap *resource digital*.

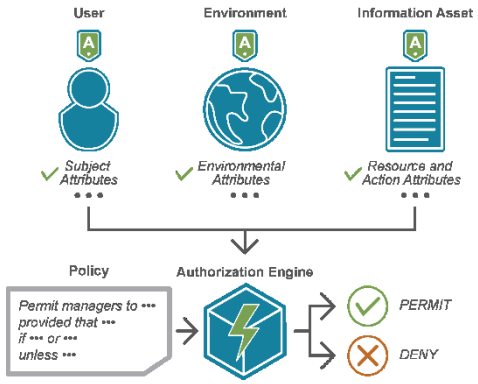
Saxena (2009) menjelaskan bahwa jenis-jenis *resource digital* elektronik sangat beragam, yaitu mencakup buku elektronik (*e-books*), *database* elektronik (e-*databases*), penerbitan elektronik dalam CD-ROM, POD (*Print On Demand*), *content digital*, dan tinta elektronik (*e-ink*). Selanjutnya, Wikoff (2011) menyebutkan bahwa yang disebut dengan sumber-sumber digital adalah, "*databases, e-joumal collection, e-book, and some mention linking technologies and e- resources management systems*".

Berdasarkan sumbernya, Marshall (2008), *resource digital* terbagi menjadi 2 kategori yaitu *closed system* dan *open system*. *Closed system* merupakan sistem yang pernah terkoneksi internet. Menurut Kurniawan (2017), berbeda dengan *closed system, open system* merupakan sistem yang terhubung dengan internet meskipun sistem tersebut tidak terhubung dengan sistem pada komputer lain, contohnya ketika seseorang menghubungkan laptop pada *WiFi*.

Menurut Stallings (2015), *access control* adalah merupakan *central* dari keamanan komputer. Selanjutnya menurut Stallings, (2015), *access control* didasarkan pada fungsi utama dari keamanan komputer itu sendiri yaitu tercapainya tiga hal, yaitu mencegah pengguna yang tidak sah dari akses ke *resource*, mencegah pengguna yang sah dari mengakses *resource* secara tidak sah, dan memungkinkan pengguna yang sah untuk mengakses sumber daya secara resmi.

Access control pada prinsipnya adalah sebuah mekanisme untuk membatasi operasi atau aksi terhadap sistem komputer hanya pada *legitimate* pengguna saja oleh Sandhu, (2010). Selanjutnya menurut Karp (2009), terdapat 4 isu utama dalam *access control*, yaitu *identification, authentication, authorization* dan *access decisions.*

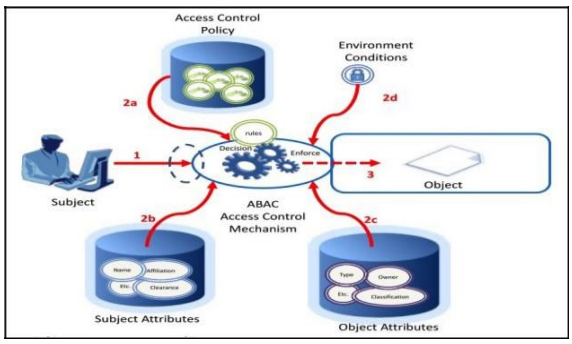
Attribute Based Access control (ABAC) sebagai sebuah model untuk menerapkan *access control policy*, oleh Jin, (2014) diprediksi menjelang tahun 2020, ABAC ini akan menjadi standar dan akan lebih banyak diterapkan oleh industri. Untuk itulah sejumlah peneliti seperti halnya Burmester, (2013) lebih banyak menggunakan pendekatan ABAC ini dalam menyelesaikan sejumlah permasalahan seputar *access control policy*. Beberapa contoh penerapan dalam dunia nyata dari penggunaan ABAC serta berbagai kelebihan dari penerapannya dapat dilihat dari laporan yang dibuat oleh Cavoukian, (2015). Bahasa spesifikasi ABAC yang saat ini ada, menyediakan berbagai pendekatan berbeda untuk menspesifikasikan fungsi *access control* dengan menggunakan *rule*. *Access control* melindungi sistem dan sumber daya dari akses yang tidak berhak dan umumnya menentukan tingkat otorisasi setelah prosedur autentikasi berhasil dilengkapi seperti yang ada di gambar 1.



Gambar 1. Attribute Absed Acccess Control System

XACML (*Extensible Access Control Markup Language*) adalah standar dari OASIS untuk menspesifikasikan ABAC *policy* menggunakan format XML. Terdapat 4 atribut *predefined* yaitu *subject, resource, action* dan *environment*. Namun

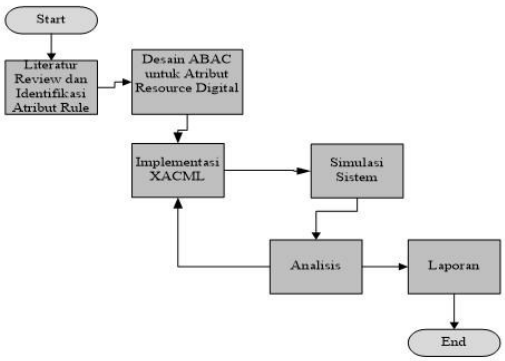
tipe pengguna *attribute* dapat juga diterapkan untuk aplikasi tertentu. XACML mendukung berbagai tipe data, tipe nama serta *path expression* untuk atribut misalnya *string, integer, internet-based names, regular expression* dan XPATH. Dalam hal penggunaan atribut, tipe data lebih utama dispesifikasikan dibandingkan dengan domain, Kannan (2013). Gambar 2 menunjukkan cara kerja pemodelan dengan sistem ABAC.



Gambar 2. Gambaran Umum Cara Kerja ABAC

2. METODOLOGI PENELITIAN

Skema penelitian ini dilakukan untuk memberikan rincian tentang alur sistematis dan menyelesaikan masalah serta membuat analisis terhadap hasil penelitian. Gambar 3 menjelaskan tentang skema awal penelitian ini dibuat.



Gambar 3. Alur Metodologi Penelitian

2.1 Literatur Review dan Identifikasi Atribut

Literatur review dilakukan untuk mendapatkan informasi mengenai topik-topik yang akan diteliti yang dapat diperoleh dari buku, dokumen, artikel, atau bahan tertulis lainnya yang berupa buku laporan, teori, maupun penemuan lainya yang bersifat *online* maupun *offline* yang bertujuan memberikan informasi.

Identifikasi atribut dari aktor dan sistem dilakukan untuk tujuan dilakukannya penelitian yang terkait dengan atribut-atribut yang terkait untuk sarana pendukung bahwa akses terhadap *resource digital* itu tidak sesederhana autentikasi dan otorisasi, Riadi (2017) berikut juga metode yang digunakan agar dapat menunjang tujuan akhir dalam penelitian ini

Adapun aspek atribut-atribut yang dimasukkan ke dalam ABAC, yaitu :

1. Subjek adalah pengguna manusia ataupun non human (misalnya *device* ataupun komponen software) yang meminta *request access*. Contoh dari atribut untuk subjek adalah nama, tanggal lahir, alamat rumah, pekerjaan.
2. *Resource* adalah sesuatu target yang diproteksi seperti halnya device, *file*s, *record*, *table*, proses, program, dan jaringan.
3. *Operation* adalah eksekusi dari suatu fungsi pada saat melakukan *request* dari sebuah subjek terhadap *resource*. Sebagai contoh, operation terhadap *file* data akan melibatkan *creation*, *modification* dan *deletion*.
4. *Environment* atribut adalah karakteristik dari operational ataupun situasional seperti misalnya *current time, current temperature, IP address*

2.2 Atribut Bukti Digital dan Pemodelan Interaksi Akses Kontrol

Tahap perancangan desain atribut pada sistem ini denngan menambahkan desain yang sebelumnya belum terstruktur oleh atribut. Desain sistem yang terdapat pada *resource digital e-library* ini digunakan sebagai objek penelitian. Salah satu contoh yang ada di tabel 1 menunjukkan contoh atribut yang tersemat di dalam atribut subject.

Tabel 1. Daftar Atribut Subject

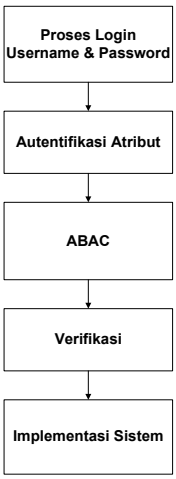
No	User	Atribut				*Action*
		A 1	A 2	A 3	A4	
1	P *1*	√	√	√	√	*Permit*
2	P *2*	√	√	√	√	*Permit*
3	P *3*	√	√	√	√	*Permit*
4	P *4*	√	√	√	√	*Permit*
5	P *5*	√	√	√	√	*Permit*

Atribut yang diberikan pada setiap aktor berdasarkan daftar atribut *subject* merupakan atribut yang berkaitan dengan identitas diri setiap aktor hal ini untuk dapat memastikan bahwa akun yang melakukan *login* adalah benar-benar pemilik akun, pada daftar atribut *subject* ini atribut yang diberikan pada setiap aktor yaitu berupa biodata pengguna seperti, nama, tanggal lahir, alamat rumah, pekerjaan.

3. PERANCANGAN SISTEM

Implementasi XACML dengan melengkapi paket XACML dengan jenis *predefined attribute-matching predicates* (misalnya: *name-match and string-equal*) yang mendukung *attribute types and expressions*. Hal ini memungkinkan dibangunnya *user-defined predicates* dari kondisi *predefined and* pengguna-*defined functions*. *Rule* untuk kombinasi algoritma dapat digunakan untuk mengatasi terjadinya konflik dari *rule* untuk *policy* yang sama. Kondisi kombinasi yang mungkin terjadi adalah

deny-overrides, permit-overrides, first-applicable, ordered-deny -overrides, ordered-permit-overrides, deny unless- permit, and permit-unless-deny. Selain itu dapat pula diterapkan *Policy combining algorithms* untuk mengatasi terjadinya *conflict policies* pada himpunan *policy* yang sama. Kondisi yang mungkin diterapkan adalah *deny - overrides, permitoverrides, first-applicable, only-one-applicable-policy, ordered-deny -overrides, ordered-permit-overrides, deny unless-permit, and permit-unless-deny*. Pada XACML, *Access decisions (or answers to access requests*) tidak hanya terbatas pada *permit* dan *deny* saja namun juga termasuk *intermediate* dan *Not Applicable*. *Hierarchical attribute* diterapkan melalui profil yang terpisah melalui gambar 4.



Gambar 4 Prinsip Kerja Sistem

3.1. Simulasi Sistem

Skenario rancangan akses kontrol yang akan dibangun yaitu bagaimana seorang aktor melakukan proses *login* pada aplikasi, pada saat memasukan *username* dan *password* sistem dengan otomatis akan melakukan autentifikasi apakah benar *username* dan *password* yang dimasukan merupakan identitas dari salah satu aktor yang telah didaftarkan, selanjutnya jika *username* dan *password* yang dimasukan merupakan identitas dari salah satu aktor yang ada, maka sistem akan mencari tahu siapa aktor yang mempunyai *username* dan *password* tersebut, jika sudah diketahui pemiliknya maka masuk pada proses pengecekan atribut pemilik identitas tersebut apakah sudah sesuai dengan kebijakan yang diberikan atau tidak, verifikasi atribut meliputi kecocokan atribut berdasarkan *subject, resource, operation,* dan *environment.* Jika identitas tersebut memenuhi persyaratan kebijakan yang telah diberikan maka *action* yang akan diterima yaitu *permit* atau pengguna diijinkan untuk masuk pada sistem. *Action deny* akan terjadi pada dua kemungkinan yaitu pertama jika *username* dan *password* yang dimasukan tidak terdaftar dalam sistem, kedua jika saat proses verifikasi berjalan tidak dapat memenuhi persyaratan kebijakan salah satu atribut yang ada, maka proses *login* dinyatakan *deny* atau tidak bisa masuk pada sistem *dummy*.

Pada struktur pembuatan rancangan ABAC ini adalah bagaimana ABAC menerapkan aturan-aturan (*rule policy*) setiap pengguna yang masuk ke sistem. Untuk mempermudah klarifikasi kondisi *resource digital* dan aktor yang mengakses akan dikelompokkan menggunakan data analisis yang terinci seperti di Tabel 2.

Tabel 2. Tabel skema pengujian untuk *Subject*

Pengguna	Atribut					Detail
	(1)	(2)	(3)	(4)	(5)	
Subject 1	√	√	√	√	√	*Yes*
Subject 1	-	√	√	√	√	*No*
Subject 1	-	-	√	√	√	*No*
Subject 1	-	-	-	√	√	*No*
Subject 1	-	-	-	-	√	*No*

4. HASIL DAN ANALISIS

Langkah-langkah penelitian dilakukan dari proses analisis hingga mendapatkan hasil dari penelitian ini. Pembahasan ini meliputi tahap studi identifikasi sistem yang digunakan dari atribut aktor-aktor dalam sistem dilanjutkan dengan pembuatan desain ABAC untuk atribut *resource digital* untuk diimplementasikan dalam output XACML. Studi kasus dilaksanakan di Perpustakaan SMK Baturjaya Ceper, Klaten.

4.1 Identifikasi Atribut

Sebuah atribut dapat dispesifikasikan melalui sebuah *identifier* (variabel), *type* data dan sebuah domain dimana sebuah himpunan finite yang memuat nilai *type* data yang diberikan. *Type* data dari atribut dapat berupa *type* data yang umumnya dipakai dalam sistem komputer seperti *integer*, *string* dan *boolean*. Beberapa usulan *Rule* dan Atribut yang digunakan dalam *resource* digital dengan studi kasus yang dikembangkan yaitu sistem perpustakaan dengan atribut yang di antaranya terdapat di tabel 3.

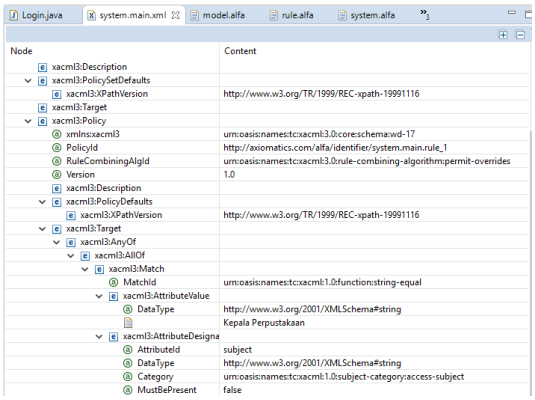
Penggunaan *rule* juga memunculkan isu konflik atau inkonsistensi, yaitu sebuah *rule* menghasilkan *decision* yang berbeda untuk nilai atribut yang sama.

4.2 *Implementasi XACML*

Struktur XACML yang diterapkan di dalam sistem ini disusun berdasarkan *request* yang telah diusulkan sebelumnya. Salah satu *tool* yang digunakan untuk mengimplementasikan sistem ini adalah *Axiomatics Language for Authorization* (ALFA) yang ada di dalam sistem Java Eclipse. Pembuatan atribut dari *policy statement*-nya diawali dengan pembuatan struktur *policy* dengan XACML yang terlihat pada gambar 5.

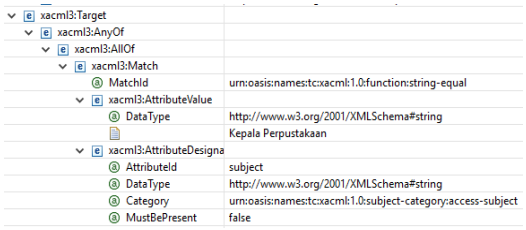
Tabel 3. Usulan Atribut *Rule* Sistem Perpustakaan

Rule	*Subject*	*Resource*	*Action*	*Environment*
Rule 1	Kepala Perpus	*Upload Digital Book* *View Statistic* *Create Session*	*Upload* *View* *Create*	*IP Address* *Mac Address* *Time Access*
Rule 2	Pustakawan	*Download Book* *Upload New Book* *Complete Data*	*Download* *Upload* *Complete*	*IP Address* *Mac Address* *Time Access*
Rule 3	Petugas IT	*Delete Inventory* *Change Password* *Validate Book* *Upload Foto* *Validate Data*	*Delete* *Change* *Validate* *Upload* *Validate*	*IP Address* *Mac Address* *Time Access*



Gambar 5. Struktur *Policy* Atribut ABAC

Penggunaan 1 *policy* dan 1 buah *rule* dikarenakan kebutuhan yang ada pada sistem tidak mengharuskan untuk menggunakan lebih dari 1 *policy* dan *rule* yang menampung keseluruhan atribut yang disematkan pada *user*. Gambar 5 menjelaskan contoh atribut subject yang letak fleksibilitas ABAC-nya terdapat pada aturan *policy* dimana sebuah *resource* itu dioperasikan lebih dari satu permintaan akses.



Gambar 6. Atribut *Subject*

Perancangan yang selanjutnya, *policy* yang dibangun dengan menambahkan *rule* diberi nama *rules* yang dimana berisi aturan kebijakan yang diberikan kepada setiap pengguna yang terdiri dari elemen *subject*, *resource*, *action*, dan *environment*. *Rules* tersebut diberi nilai *effect*: *permit* yang artinya pengguna tersebut akan diizinkan mengakses sistem apabila dianggap memenuhi kebijakan yang diberikan, selanjutnya menentukan target yang menjadi kebijakan pada masing-masing pengguna dengan kebijakan yang diberikan terbagi menjadi 4 bagian atau 4 komponen elemen utama yang menjadi landasan perancangan *policy* ini.

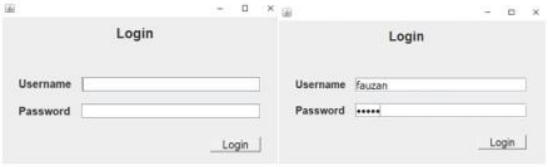
4.3. *Output XACML*

Output XACML *policy* yang telah dibuat berupa tahapan awal pada target *root* elemen *policy* dimana jumlah *subject* sebanyak 3 buah, dan jumlah *resource* sebanyak 13 buah. Hanya ada 3 pengguna yang berhak melakukan akses pada sistem ini. Masing-masing dari pengguna itu adalah kepala perpustakaan, pustakawan dan admin yang berhak mengakses *resource* sesuai dengan kebijakan *access* yang telah diberikan seperti ditunjukkan di Gambar 7 yang berupa output XACML.

Gambar 7. Tampilan Output XACML

4.4. *Simulasi Sistem*

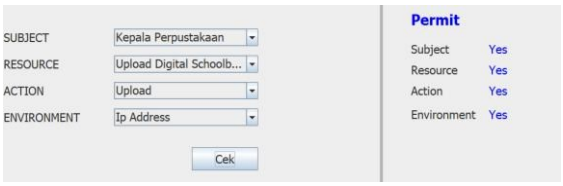
Rancangan *Attribute Based Access Control* (ABAC) dengan struktur XACML pada *resource digital* dengan studi kasus sistem perpustakaan ini, akses kontrol yang dibuat menggunakan stuktur XACML serta bahasa pemrograman Java dan *windows builder* sebagai *complier*. Penggunaan struktur XACML sangat mengukung untuk memenuhi semua kebutuhan yang digunakan saat merancang akses kontrol *policy* seperti *Attribute Based Access Control* (ABAC). Tampilan autentikasi berada pada sistem *login* yang tertera di Gambar 8.



Gambar 8 Halaman *Login*

Beberapa pengujian dengan kondisi *permit* dapat dibuktikan dengan memberikan inputan *subject*, *resource*, *action* dan *environment* sesuai dengan akses kontrol yang sudah diterapkan. Berikut merupakan tampilan pengujian akses kontrol yang berbasis *java*. Pengujian sample *permit* dilakukan

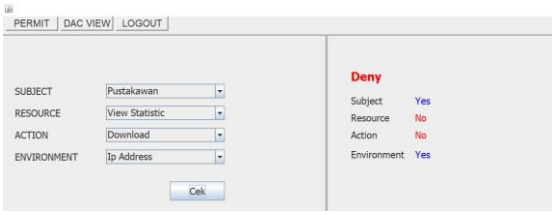
oleh setiap *rule* dari beberapa atribut yang disematkan dengan masing-masing *subject* seperti gambar 9.



Gambar 9. Halaman Permit

Pengujian kondisi *permit* yang pertama ketika melakukan klik tombol "cek" merupakan kondisi yang dihasilkan yaitu *permit* dikarenakan semua atribut yang dimasukkan benar seperti yang terlihat pada halaman info *permit subject*: yes; *resource*: yes; *action*: yes; dan *environment*: yes atribut yang dimasukkan di atas merupakan atribut yang mewakili keseluruhan atribut yang ada pada *rule*s. Atribut yang telah disematkan pada *rule*1,yaitu *subject*: kepala perpustakaan, *resource*: *upload digital schoolbook*, *action*s: upload, dan *environment*: ip address

Pengujian sample *deny* dilakukan dengan memasukkan atribut yang tidak sesuai oleh setiap *rule* dengan masing-masing *atribute* seperti yang ada di gambar 10.



Gambar 10. Halaman Deny

Sampel pengujian kondisi *deny* yang pertama yang dilakukan menggunakan pengguna pustakawan, ketika melakukan klik tombol "cek" kondisi yang dihasilkan adalah *deny* sebagaimana yang terlihat pada halaman info bahwa *subject*: yes, *resource*: no, *action*s no dan *environment*: yes terdapat 2 kesalahan yaitu pertama terdapat pada inputan *resource* yang diisi dengan *view statistic* dan kesalahan kedua ditemukan pada inputan *action* yang diisi dengan *download*. Hal ini disebabkan bahwa atribut *resource* dan *action* yang dimasukan bukan merupakan atribut dari pustakawan.

4.5. *Analisis*

Studi kasus yang terdapat pada penelitian ini terdapat permasalahan-permasalahan pada metode model akses kontrol sebelumnya, sehingga dalam proses analisis ini akan menjabarkan beberapa penyelesaian berdasarkan studi kasus yang diangkat di dalam salah satu *resource digital* yaitu sistem perpustakaan. Beberapa penyelesaian menjadi solusi di antaranya berdasarkan perbandingan metode akses kontrol yang terdahulu dengan metode akses kontrol yang digunakan penelitian yang dirangkum di tabel 4.

Tabel 4. Perbandingan Metode Akses Kontrol dengan Metode yang Diusulkan

XACML log	ALFA log	Function	
		Y	N
Xmlns	*urn:oasis:names:tc:xacml:3.0:core:schema:*	√	
PolicyId	*system.main.rule_1*	√	
RuleCombiningAlgId	*rule-combining-algorithm:permit-overrides*	√	
Rule	*Effect & RuleId*	√	
Subject Match	*MatchId*	√	
AttributeValue	*DataType;*	√	
AttributeDesignator	*AttributeId; DataType; Category; MustBePresent*	√	
Resource Match	*MatchId*	√	
AttributeValue	*DataType;*	√	
AttributeDesignator	*AttributeId; DataType; Category; MustBePresent*	√	
Action Match	*MatchId*	√	

XACML *log* yang dirancang di sistem perpustakaan ini menggunakan 1 *policy* dan 1 *rule*, untuk penggunaan aturan *firt applicable* pada *rule combining applicable* untuk mengatasi terjadinya konflik antar elemen pada 1 himpunan *policy* dan *rule* yang sama. Penggunaan 1 *rule* pada 1 *policy* bertujan agar dapat mempermudah jika sewaktu-waktu dilakukan penambahan jumlah pengguna pada 1 jabatan yang sama serta dilengkapi dengan 4 atribut *predefined* yaitu *subject*, *resource*, *action* dan *environment* yang berguna untuk menentukan atribut yang digunakan pada akses kontrol yang ada di sistem ini.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Desain akses kontrol ABAC yang diterapkan di sistem perpustakaan ini menunjukkan bahwa pendekatan dengan metode ABAC menjadi solusi yang tepat dan relevan dalam mendukung proses mendukungnya tingkat keamanan khususnya dalam hal identifikasi, otorisasi dan autentikasi pengguna. Dari studi kasus objek sistem perpustakaan ini, hasil pengujian implementasi dengan sampel data uji berupa skenario dan simulasi dan pengujian menggunakan *tools* yang dibuat khusus didapatkan hasul bahwa akses kontrol yang dibuat telah berjalan dengan baik dan berfungsi sebagaimana mestinya dengan pendekatan ABAC yang dapat digunakan di berbagai contoh *resource* digital yang lain seperti *ejournal* atau *edetailing.*

5.2 Saran

Adapun saran bagi peneliti selanjutnya yang mengembangkan akses kontrol, perlu memperhatikan faktor dalam pengujian skema struktur XACML dan masih diperlukan validasi terhadap rancangan strukur XACML yang telah dibuat. Selain itu, sistem ALFA

belum menerapkan manajemen otorisasi yang dinamis untuk mengelola XACML *policy*, sehingga idealnya *request access* diatur dalam manajemen otorisasi tidak sekedar pemanggilan metode dalam *script* baris kode.

DAFTAR PUSTAKA

CAVOUKIAN, A .2015. The Importance of ABAC Attribute-Based Access Control to Big Data : Privacy and Context. http://www.ryerson.ca/pbdi/.

D. FERRAIOLO, S. G. 2015. Policy Machine: Features, Architecture, and Specification. *National Institute of Standards and Technology (NIST) IR-7987 Revision 1*, 23-28.

HSU, C.-L. A.-L. 2011. A Digital Evidence Protection Method with Hierarchical Access Control Mechanisms. *IEEE* (hal. 1–9). Barcelona: IEEE International Carnahan Conference on Security Technology (ICCST).

HU, V. C. 2015. Attribute-Based Access Control. *Computer*, doi:10.1109/MC.2015.33.

JIN, X. 2014. Attribute-Based Access Control Models and Implementation in Cloud Infrastructure as A Service . *The University of Texas at San Antonio*, doi:10.1007/s13398-014-0173-7.2.

KARP, A. H. 2009. [Online] From ABAC to ZBAC : The Evolution of Access Control Models From ABAC to ZBAC. *The Evolution of Access Control Models*, http://www.hpl.hp.com/techreports/2009/HPL-2009-30.pdf.

KURNIAWAN A, RIADI, I, & LUTHFI, A. 2017. Forensic Analysis And Prevent Of Cross Site Scripting In Single Victim Attack Using Open Web Application Security Project (OWASP) Framework. Journal of Theoretical and Applied Information Technology, 1363-1371

PANENDE,M.F, PRAYUDI,Y, & RIADI, I . 2018. Konsep Attribute Based Access Control (ABAC) Pada Lemari Penyimpanan Bukti Digital (LPBD). Jurnal Teknik Informatika Vol. 11 No. 1, 85-94

RIADI, I, SUNARDI, & FIRDONSYAH, A. 2017. Forensic Investigation Technique on Android's Blackberry Messenger using NIST Framework. International Journal of Cyber-Security and Digital Forensics (IJCSDF), 198-205

RIADI, I, UMAR R, & NASRULLOH I M, 2018. Experimental Investigation of Frozen Solid State Drive on Digital Evidence with Static Forensic Methods. Lontar Komputer, 169-181

SANDHU, R. 2010. Security Models : Past , Present and Future San Antonio, TX, USA. *Institute for Cyber Security, UTSA USA.* , http://profsandhu.com/miscppt/utsa100831.pdf.

STALLINGS, W. A. (2015). *Computer Security : Principles and Practice. 3rd Editio.* USA: Pearson Education International.

SUBEKTINGSIH, PRAYUDI,Y, & RIADI, I. 2018. Digital Forensics Workflow as A Mapping Model for People, Evidence, and Process in Digital Investigation. Journal of Cyber-Security and Digital Forensics, 294-304

TAYLOR, C. B.-P. 2007. Specifying Digital Forensics: A Forensics Policy Approach. *Digital Investigation 4 (September)*, 101–104. doi:10.1016/j.diin.2007.06.006.

VARADHARAJAN, V. 2015. Policy Based Role Centric Attribute Based Access Control Model Policy RC-ABAC. *Conference on Computing and Network Communications (CoCoNet'15)*, 12-17.

XU, D. A. 2014. Specification and Analysis of Attribute-Based Access Control Policies: An Overview. *Proceedings - 8th International Conference on Software Security and Reliability - Companion SERE-2014*, 41–49. doi:10.1109/SERE.

Halaman ini sengaja dikosongkan