Eksplorasi ABAC dan XACML untuk *Design Access Control* pada *Resource Digital*

Fauzan Natsir

14917123

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Digital Forensik

Program Studi Magister Teknik Informatika

Program Pascasarjana Fakultas Teknologi Industri

Universitas Islam Indonesia

2019

Lembar Pengesahan Pembimbing

Eksplorasi ABAC dan XACML untuk Design Access Control pada Resource Digital

Fauzan Natsir

14917123

Yogyakarta, Februari 2019

Pembimbing

Dr. Imam Riadi, M.Kom

i

Lembar Pengesahan Penguji

Eksplorasi ABAC dan XACML untuk *Design Access Control* pada *Resource Digital*

Fauzan Natsir

14917123

Yogyakarta, Februari 2019

Tim Penguji,

Dr. Imam Riadi, M.Kom
Ketua

Dr. Bambang Sugiantoro
Anggota I

Dr. Bambang Sutiyoso,S.H.,M.Hum
Anggota II

Mengetahui,
Ketua Program Pascasarjana Fakultas Teknologi Industri
Universitas Islam Indonesia

UNIVERSITAS ISLAM INDONESIA
PRODI TEKNIK INFORMATIKA
PROGRAM MAGISTER
FAKULTAS TEKNOLOGI INDUSTRI

Izzati Muhimmah, ST.,M.Sc.,Ph.D

ii

Abstrak

Eksplorasi ABAC dan XACML untuk *Design Access Control* pada *Resource Digital*

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 dan terbatas. Mekanisme access kontrol *policy* yang diyakini adaptif di masa yang akan datang yaitu ABAC (*Attribute Based Access Control*) dengan implementasi model XACML (*Extensible Access Control Modelling Language*). Desain *policy* ABAC disajikan dengan atribut dari salah satu studi kasus *resource digital* dengan sistem *e-Library*. 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 untuk pemberian kinerja akses kontrol terhadap *resource digital*. Pendekatan ABAC dengan model XACML ini menyajikan suatu keamanan sistem dengan model akses kontrol berbasi 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

ABAC and XACML Exploration for Design Access Control in Digital Resources

Digital resources require a mechanism to regulate policy against controls to get access rights into a system. Access control is more flexible than the authorization, authentication or verification approach that is very simple and limited. The mechanism of access control policy that is believed to be adaptive in the future is ABAC (Attribute Based Access Control) with the implementation of the XACML (Extensible Access Control Modeling Language) model. The ABAC policy design is presented with attributes from one of the digital resource case studies with the e-Library system. This research 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 to provide performance control access to digital resources. The ABAC approach with the XACML model presents a system security with attribute-based access control models from policy statements to be a previously created access control model solution and support the access control model relevant for digital resources.

Keywords

Access Control; Digital Resource; ABAC; XACML; ALFA

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, Februari 2019

Fauzan Natsir

Daftar Publikasi

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Author Fauzan Natsir	Mendesain eksperimen (60%) Menulis *paper* (70%)
Author Yudi Prayudi	Mendesain eksperimen (20%) Menulis dan mengedit *paper* (15%)
Author Imam Riadi	Mendesain eksperimen (20%) Menulis dan mengedit *paper* (15%)

Halaman Kontribusi

Tidak ada kontribusi dari pihak lain

Halaman Persembahan

Syukur alhamdulillah kehadirat Allah SWT atas limpahan rahmat, hidayah, serta inayahnya yang tidak pernah berhenti memberikan pertolongan maha pengasih maha penyayang, serta salawat bagi Rasulullah SAW rahmat bagi semesta alam. Kupersembahkan karyaku

untuk Rasulullah,

untuk Islam agamaku,

untuk Bapak dan Ibuku,

untuk Istriku,

untuk Keluargaku,

untuk Sahabatku,

untuk Pembaca,

dan melalui ini juga penulis menghaturkan permohonan maaf yang sebesar-besarnya kepada keluarga dan rekan-rekanku yang selalu aku repotkan.

Saya persembahkan tesis ini kepada semua pembaca. Semoga tesis ini dapat bermanfaat, menambah pengetahuan, dan memberikan inspirasi kepada siapapun yang membacanya. Semua penulis lakukan untuk membanggakan keluarga.

Kata Pengantar

Assalamualaikum Wr. Wb.

Alhamdulillahirabbil ‘alamin, segala puji hanya bagi Allah SWT atas segala rahmat, hidayah, dan inayah-Nya sehingga penulis dapat menyelesaikan penulisan laporan tesis sebagai salah satu syarat menyelesaikan pendidikan dan memperoleh gelar Magister di Program Pascasarjana Magister Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang berjudul “Eksplorasi ABAC dan XACML untuk *Design Access Control* pada *Resource Digital*”.

Sholawat dan salam semoga senantiasa tercurah dan terlimpah kepada junjungan kita Nabi Besar Muhammad SAW, keluarga, sahabat, dan kepada seluruh umatnya sampai dengan akhir zaman nanti.

Dalam penyusunan laporan tesis ini penulis menyadari tidak akan dapat menyelesaikan dengan baik tanpa adanya bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak, ibu, istri, adik, beserta keluarga besar yang selalu memberikan dukungan dan doanya agar penulis senantiasa diberikan kemudahan.
2. Bapak Rektor dan seluruh jajaran Rektorat Universitas Islam Indonesia
3. Ibu Izzati Muhimmah.,S.T., M.Sc., Ph.D. selaku direktur Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan kebijakan selama proses penyelesaian tesis ini.
4. Bapak Dr. Imam Riadi M.Kom dan Bapak Yudi Prayudi, S.Si, M,Kom selaku dosen pembimbing 1 dan 2 yang telah memberikan pengarahan, bimbingan, masukan, serta dorongan semangat selama penulis mengerjakan tesis ini.
5. Bapak Dr. Bambang Sugiantoro, M.T. selaku dosen penguji yang telah memberikan motivasi dan semangat serta bimbingan yang sangat berarti bagi penulis dalam menyelesaikan tesis ini.

6. Bapak Dr. Bambang Sutiyoso, S.H.,Hum. selaku dosen penguji yang telah memberikan motivasi dan semangat serta bimbingan yang sangat berarti bagi penulis dalam menyelesaikan tesis ini.
7. Dosen-dosen Magister Teknik Informatika dan jajaran staf program Pascasarjana, terima kasih atas semua ilmu pengetahuan, saran, motivasi, dan bantuannya.
8. Rekan-rekan konsentrasi Forensika Digital UII, terima kasih atas dukungan dan kerjasamanya.
9. Rekan-rekanku UMS dan UP45, terima kasih atas dukungannya.
10. Keluarga besar Magister Informatika UII, terima kasih atas kerjasamanya.
11. Staf Administrasi dan tata usaha Magister Informatika,Universitas Islam

    Indonesia, yang telah membantu dalam segala urusan administrasi di kampu
12. Semua pihak yang telah memberikan bantuan dan dukungannya yang tidak dapat penulis sebutkan satu per satu.

Semoga segala kebaikan yang semua pihak berikan kepada penulis dibalas oleh Allah SWT dengan kebaikan yang lebih baik. Aamiin.

Penulis menyadari bahwasanya dalam penulisan dan penyusunan laporan tesis ini masih banyak terdapat kekurangan dan masih jauh dari sempurna. Untuk itu penulis memohon maaf yang sebesar-besarnya dan penulis mengharapkan kritik dan saran yang membangun untuk penyempurnaan di masa mendatang.

Akhir kata semoga laporan tesis ini dapat bermanfaat bagi kita semua. Aamiin Allahumma Aamiin.

Wassalamualaikum Wr. Wb.

Yogyakarta, Februari 2019

Fauzan Natsir

Daftar Isi

Daftar Tabel

Daftar Gambar

Glosarium

ABAC	- Attribute Based Access Control
ALFA	- Axiomatics Language for Authorization
XACML	- eXtensible Access Control Markup Language
RBAC	- Rule Based Access Control
ACL	- Access Control List
DAC	- Directory Access Control
MAC	- Mandatory Access Control
RD	- Resource Digital
NGAC	- Next Generation Access Control
SOA	- Service Oriented Architecture
ACPO	- Association of Chief Police Officer
PDP	- Policy Decision Point

BAB 1
Pendahuluan

1.1 Latar Belakang

Era digital saat ini telah menghasilkan berbagai macam *resource digital* yang penting. *Resource digital* didefinisikan sebagai fisik atau informasi elektronik seperti tertulis atau dokumentasi elektronik, komputer *file log*, data, laporan, fisik *hardware*, *software*, disk gambar, dan sebagainya yang dikumpulkan. *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. Pendekatan yang umumnya dilakukan adalah menggunakan skema atau mekansime autentikasi dan otorisasi. Melalui mekanisme ini, proses autentikasi melalui validasi pengguna pada saat memasuki sistem, nama dan password dari pengguna dicek melalui proses yang mengecek langsung ke daftar yang diberikan hak untuk memasuki sistem tersebut. Dalam hal ini autentikasi merupakan sebuah proses identifikasi yang dilakukan oleh pihak yang satu terhadap pihak yang lain ataupun sebaliknya dengan melakukan berbagai proses identifikasi untuk memastikan keaslian dari informasi yang diterima berupa waktu pembuatan informasi, waktu pengiriman informasi, isi informasi, kepastian pengirim ataupun si penerima data.

Salah satu *resource digital* yang penting adalah bukti digital. Bukti digital merupakan salah satu *resource*s yang sangat sensitif. Sehingga perlu ada sebuah mekanisme untuk mengatur *policy* terhadap akses pada *resource bukti digital*. Akses terhadap *resource bukti digital* ternyata tidak sesederhana akses terhadap *resource* yang lain. Dengan demikian, akses terhadap *resource bukti digital* tidak cukup difasilitasi hanya dengan autentikasi dan otorisasi saja tetapi harus dengan pendekatan yang lain.

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 untuk memodelkan akses terhadap *resource digital*. Contohnya, dengan autentikasi hanya dapat melakukan validasi dan konfirmasi pengguna yang berbasis pada kerahasisaan informasi (*password, PIN, digital certificate, private key*) pada saat memasuki suatu sistem. Sedangkan otorisasi hanya dapat melakukan layanan yang bisa dinikmati pengguna yang telah jelas identitasnya (*authenticated user*). Identitas yang telah dibuktikan, diproses melalui autentikasi menjadi dasar untuk menentukan layanan yang berhak dinikmati seorang pengguna. Sementara kebutuhan pembatasan akses terhadap *resource digital* bisa lebih kompleks dari itu, misalnya akses dibatasi oleh sejumlah atribut dibatasi oleh akses waktu, akses ukuran, akses lokasi hingga akses *cybermetric*.

Salah satu solusi untuk mengatasi permasalahan tersebut adalah melalui pendekatan *access control policy*. Pendekatan ini 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*, seperti *Discretionary Access Control* (DAC), *mandatory access control* (MAC), *Access Control List* (ACL), serta *Rule Based Access Control* (RBAC) dan 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. Sejumlah penelitian yang ada antara lain pernah dilakukan oleh (D. Ferraiolo, October 2015) dan (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). Untuk itu maka eksplorasi ABAC dan XACML untuk *design access control policy* pada *resource digital* sangatlah

penting untuk dikaji. Hal ini akan memberikan kontribusi pada ketersediaan kajian tentang ABAC dan XACML pada lingkup bidang forensika digital khususnya dan keamanan komputer pada umumnya.

Mengingat pentingnya eksplorasi tentang ABAC dan XACML terhadap akses *resource digital* maka diperlukan penelitian lebih lanjut untuk melakukan model lebih lanjut untuk mengetahui tentang bagaimana pemodelan ABAC pada akses *resource* bukti digital, implementasi terkait kinerja ABAC dengan XACML dibandingkan dengan pendekatan otorisasi, autentikasi, dan verifikasi yang umumnya dipakai selama ini. 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*.

1.2 Rumusan Masalah

Berdasarkan permasalahan pada latar belakang, maka dapat diambil beberapa rumusan masalah di antaranya :

1. Bagaimana desain *atribut* yang sesuai untuk akses kontrol terhadap *resource digital* ?
2. Bagaimanakah desain ABAC yang sesuai untuk atribut akses kontrol terhadap *resource digital*?
3. Bagaimana implementasi XACML untuk desain ABAC yang dirancang?

1.3 Batasan Masalah

Di dalam melaksanakan kegiatan penelitian ini ada beberapa batasan masalah, yaitu:

1. *Tools* yang digunakan dalam penelitian ini adalah ALFA (*Axiomatics Language for Authorization)* dengan pendekatan ABAC dan XACML sebagai implementasinya.
2. Sistem yang dipakai bersifat *dummy* yang akan diuji keberhasilannya melalui implementasi atribut ABAC.
3. Atribut dan sistem ini diujikan di Perpustakaan SMK Baturjaya 1 Ceper Klaten.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dibuat maka dapat diambil tujuan penelitian ini sebagai berikut:

1. Menghasilkan desain dari atribut-atribut yang relevan pada akses kontrol terhadap resource digital.
2. Menghasilkan desain policy ABAC pada akses kontrol resource digital.
3. Menghasilkan implementasi desain atribut ABAC dengan XACML terhadap kinerja dari sistem akses kontrol terutama pada konsep autentikasi, verifikasi dan otorisasi.

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari hasil penelitian adalah sebagai berikut:

1. Memberikan kemudahan dalam penggunaan *Access Control* untuk penyimpanan *resource digital* yang menggunakan pendekatan ABAC dengan XACML.
2. Mengetahui jenis karakteristik data atribut yang didapat dari kompilasi kebijakan ABAC.
3. Sebagai referensi bagi peneliti lain yang mengambil kajian penelitian yang sama dan sebagai wawasan untuk mengembangkan penelitian selanjutnya.

1.6 Review Penelitian

Dalam bidang *resource* bukti digital, (Hsu, 2011) mencoba untuk menerapkan solusi hierarchical *access control methods* sebagai solusi penanganan bukti digital serta akses control terhadapnya. Lingkungan lab forensik digital dijadikan sebagai salah satu contoh dari penerapan mekanisme *access control* dalam solusi tersebut. Walaupun kemudian tidak banyak peneliti yang membahas lebih lanjut tentang penanganan *access control* untuk *resource digital*, namun bila nilai dan fungsi digital tersebut sama halnya dengan sebuah medical *record/health system record* yang juga sifatnya spesifik, maka terdapat beberapa kajian seputar *access control* untuk *medical/health system record*. Issuenya adalah bagaimana menjaga data yang sifatnya privacy dari kemungkinan akses seseorang yang tidak memiliki authorisasi terhadapnya. Berdasarkan kajian literatur tersebut, maka terlihat

bahwa isu seputar akses kontrol terhadap bukti digital sebenarnya adalah sebuah open problem dalam bidang forensik digital, hanya saja belum banyak dikaji oleh para peneliti dalam bidang ini. Model awal yang digunakan menggunakan *hierarchical access control methods* dapat dijadikan sebagai acuan awal untuk memberikan solusi tentang issue *access control* pada bukti digital.

Attribute Based Access control (ABAC) sebagai sebuah model untuk menerapkan *access control policy* , diprediksi menjelang tahun 2020 akan menjadi standar dan akan lebih banyak diterapkan oleh industri (Jin 2014). Untuk itulah sejumlah peneliti seperti halnya (Burmester et al., 2013; Jin, 2014; Yang dan Jia, 2012; Smari et al., 2014) lebih banyak menggunakan pendekatan ABAC ini dalam menyelesaikan sejumlah permasalahan seputar *access control policy* .

Access control terhadap bukti digital tidak cukup hanya ditangani oleh mekanisme autentikasi dan otorisasi pengguna saja. Autentikasi, otorisasi dan *access control* memiliki fungsi dan tujuan yang berbeda walaupun pada implementasinya terlihat seolah-olah sebuah proses tunggal. Menurut Younis et al. (2013), autentikasi fokusnya pada verifikasi terhadap klaim identitas pengguna, otorisasi fokusnya pada pemberian hak akses terhadap *resource*, sementara *access control* fokusnya pada proteksi keamanan dari *resource* tertentu, yaitu sebuah mekanisme untuk memastikan bahwa pengguna tidak melakukan tindakan tertentu yang tidak sesuai dengan *policy* umum keamanan yang diterapkan. *Access control* melindungi sistem dan sumber daya dari akses yang tidak berhak dan menentukan tingkat otorisasi setelah prosedur autentikasi berhasil dilengkapi. (Stallings, 2015) Keberadaan barang bukti sangat penting dalam investigasi kasus-kasus *computer crime* maupun *computer-related crime* karena dengan barang bukti inilah investigator dan *forensic analyst* dapat mengungkap kasus-kasus tersebut dengan kronologis yang lengkap, untuk kemudian melacak keberadaan pelaku dan menangkapnya. Oleh karena posisi barang bukti ini sangat strategis, investigator dan f*orensic analyst* harus paham jenis-jenis barang bukti.

Tabel 2. 1 Review Penelitian

No	Paper Utama	Rumusan Masalah	Solusi	Metode	Hasil
1	(Burmester, 2013)	Model Akses kontrol berbasis atribut waktu (T-ABAC)	Pembuatan akses kontrol berbasis waktu *real-time* (T-ABAC)	1.Peninjauan kembali model akses kontrol yaitu *Traditional Access Control Model* terutama berfokus pada kerahasiaan dan integritas	Menyajikan model ABAC, dimana attribute *real-time* dapat mempertimbangkan prioritas akses
2	(Hsu, 2011)	Model akses control bersifat spesifik	Pembuatan akses kontrol berbasis	Access control untuk medical/health system record	*Hierarchical access control methods* dijadikan sebagai acuan awal untuk memberikan solusi tentang issue access control
3	(Dan & Yuan, 2012)	Model akses kontrol lintas domain	Pembuatan akses kontrol lintas domain berbasis ABAC	Pendekatan *Service Oriented Architecture* (SOA), dan pendekatan Atribute	Menyajikan sistem akses kontrol lintas domain berbasis ABAC, bersama dengan domain keamanan sebagai atribut dengan subjek, objek, otoritas atribut lingkungan sebagai dasar akses

No	Paper Utama	Rumusan Masalah	Solusi	Metode	Hasil
				Based Access Control (ABAC) karena di anggap yang lebih halus dalam pengendalian akses.	pengambilan keputusan, menghilangkan hambatan integrasi untuk kerangka kerja SOA
4	(Longstaff, 2013)	model acces control yang diberi nama Tees Confidentiality Model (TCM)	skema yang diberi nama Ciphertext-*Policy* Attribute-Based Signcryption (CP-ABSC), yaitu pengunaan digital signature dan enkripsi untuk menjamin terpenuhi sifat confidentiality	Pendekatan RBAC dan ABAC (Attribute Based Access Control)	Model yang dikembangkan dibangun dan diuji menggunakan tools B-Methods, sebuah tools untuk membangun dan menguji spesifikasi metode formal
5	(Son Ha et al., 2016)	Model akses kontrol bebasis solusi	Pembuatan akses kontrol berbasis	Pendekatan yang digunakan dalam	Model akses kontrol bebasis solusi dilingkungan perubahan kebijakan, yaitu *modification on each*

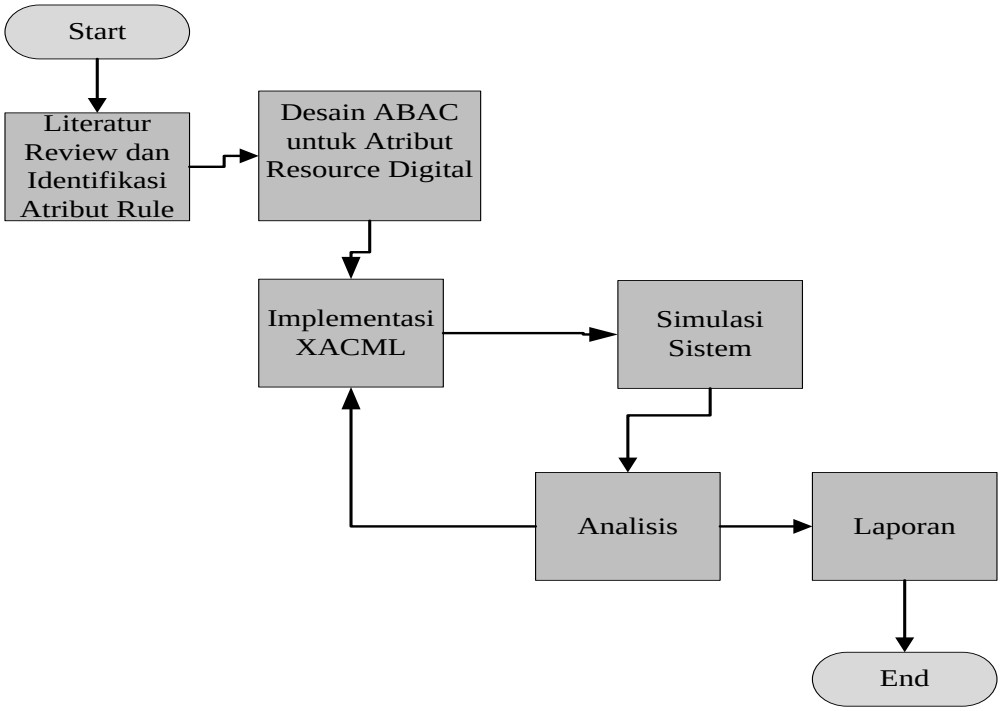
No	Paper Utama	Rumusan Masalah	Solusi	Metode	Hasil
			solusi Rew-XAC	penelitian ini adalah: Rew-XAC yaitu model akses kontrol yang dapat menerapkan proses penulisan ulang.	*request* dengan *Not Applicable* yang artinya,respon berdasrkan kebijakan sistem, lebih tepatnya model tersebut mencegat respon "*Not Applicable*" dan menghitung nilai kebijakan *fuzzy* berdasarkan kebutuhan pemilik permintaan.
6	(Riad et al., 2015)	Model akses kontrol berbasis atribut untuk Cloud Computing	Membangun sebuah model akses kontrol berbasis atribut baru yang mendukung atribut aturan untuk Cloud Computing	*Attribute-Rule* atau (AR-ABAC) untuk menentukan kesepakatan tentang jenis atribut apa yang harus digunakan dan berapa banyak atribut yang harus diperimbangkan dan memperhitungkan keputusan akses	Menyajikan usulan model akses kontrol AR-ABAC

No	Paper Utama	Rumusan Masalah	Solusi	Metode	Hasil
7	(Qi et al., 2016)	Model akses kontrol berdasarkan peran dan atribut	Merancang model akses kontrol bebasis peran dan atribut serta implementasinya	RABAC untuk memfasilitasi pengolahan peraturan akses kontrol oleh administrator	Akses kontrol dapat mengoptimalkan pilihan dalam memisahkan kode kebijakan, dengan demikian mempermudah pengolahan kebijakan.

	Sampai dengan saat ini belum adanya rancangan sebuah model akses kontrol yang secara khusus diperuntukan untuk sebuah system dummy berdasarkan atribut *policy* . Model akses kontrol yang ada pada *resource digital* saat ini hanya berupa proses autentifikasi sederhana yang pada umumnya digunakan pada sebuah sistem, hal ini akan berakibat pada *resource digital* yang ada dapat dimanipulasi oleh pihak-pihak yang tidak bertanggung jawab. Solusi dengan pendekatan akses kontrol berbasis atribut *policy* ini diterapkan karena karakteristik pengguna sistem dan interaksinya tidak sederhana akses untuk pengguna biasa. Apabila pengguna biasa, maka akses terhadap sistem cukup difasilitasi dengan model autentikasi sederhana. Namun model bisnis, khususnya untuk aktivitas investigasi digital sangat komplek sehingga membutuhkan pengaturan dalam bentuk *access control policy* .

1.7 Metode Penelitian

Susunan laporan penelitian ini perlu metodologi penyelesaian secara sistematis, penelitian ini menggunakan beberapa tahap berupa:



Gambar 1. 1 Metodologi Penelitian

1. Literatur Review dan Identifikasi Atribut *Rule*

   Literatur review dan identifikasi atribut dilakukan untuk mendapatkan informasi mengenai topik penelitian yang dapat bersumber dari dokumen, buku, artikel, atau bahan tertulis lainya yang berupa teori, laporan penelitian, atau penemuan sebelumnya, baik bersifat *online* maupun *offline source*. Pengidentifikasian atribut dan aktor dari sistem yang akan dibuat.

2. Desain ABAC untuk Atribut *Resource digital*

   Merupakan tahap perancangan desain interaksi dari atribut yang terdapat pada *resource digital* yang akan digunakan sebagai objek penelitian.

3. Implementasi XACML

   Tahap implemntasi XACML dimulai dari penerapan metode ABAC ke dalam *output* XACML.

4. Simulasi Sistem

   Merupakan tahap dilakukannya simulasi desain atribut *resource digital* menggunakan metode atribut, serta diimplementasikan ke dalam XACML.

5. Analisis

   Tahap ini dilakukan untuk melakukan investigasi dalam menemukan metode yang membantu untuk sistem keamanan dari suatu bukti digital

6. Laporan

   Tahap ini dilakukan untuk mereport semua data yang telah dianalisis yang digunakan sebagai *resource* bukti digital yang sah dan dapat diterima secara umum.

1.8 Sistematika Penulisan

Untuk mempermudah proses pembahasan dalam penelitian, maka dibuat sistematika penulisan pada penelitian ini:

BAB I PENDAHULUAN

Pendahuluan merupakan pengantar terhadap permasalahan yang akan dibahas. Di dalamnya menguraikan tentang gambaran suatu penelitian yang terdiri dari latar belakang, rumusan masalah, batasan masalah, manfaat penelitian, tujuan penelitian, metodologi penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini menjelaskan teori-teori yang digunakan untuk memecahkan masalah dalam penelitian ini. Teori yang dibahas pada bagian ini merupakan teori yang berhubungan dengan *access control.*

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langlah-langkah penelitian dan gambaran umum langkah penyelesaian.

BAB IV PEMBAHASAN

Hasil dan pembahasan, berisi tentang pembahasan penyelesaian masalah yang diangkat yaitu dengan melakukan analisis dan uji coba.

BAB V KESIMPULAN DAN SARAN

Kesimpulan dan saran, memuat kesimpulan-kesimpulan dari hasil penelitian dan saran-saran yang perlu diperhatikan berdasar keterbatasan yang ditemukan serta asumsi-asumsi yang dibuat selama melakukan penelitian dan juga rekomendasi yang dibuat untuk pengembangan penelitian selanjutnya.

BAB 2
Tinjauan Pustaka

2.1 Resource digital

Sebelum menjelaskan bagaimana melakukan pengelolaan sumber-sumber informasi elektronik, terlebih dahulu akan dikemukakan mengenai pengertian *resource digital*. Secara umum, *resource digital* adalah sumber-sumber informasi yang dikemas atau disimpan dalam bentuk elektronik atau digital. Sumber-sumber informasi digital dapat merupakan hasil alih bentuk dari format lain yang dikenal dengan reproduksi atau digitalisasi, dan dapat pula merupakan terbitan yang sengaja dikemas dalam format elektronik atau digital *(digital born)* sebagai bentuk suatu penerbitan.

Lebih lanjut Saxena (2009) menjelaskan bahwa jenis-jenis *resource digital* elektronik sangat beragam, yaitu mencakup buku elektronik (e-books), *database* elektronik (e-*database*s), 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*".

Pada definisi lain, dalam guidelines yang dikeluarkan oleh Library of Congress (2008), disebutkan mengenai digital *resource*s sebagai berikut: *An "digital resource" is defined as* any *work encoded and made available for* access *through the use of a computer. It* includes *electronic data available by (1)* remote *access and (2) direct access fixed media).* In *other words: Remote access (electronic resources) refers to the use of electronic resources via computer cartridges) designed* to *be inserted into a computerized device or* its *auxiliary equipment.*

Selanjutnya Sharon Jhonson (2012) juga menjelaskan bahwa yang dimaksud dengan *digital resources* adalah sebagai berikut: "*those materials that require computer access, whether through a personal* computer, *mainframe, or handheld mobile device. They may either be accessed remotelyvia the Internet or locally. Some of the most frequently encountered types are: e-journals, e-books full-text (aggregated) databases, indexing and abstracting databases, reference databases (biographies, dictionaries, directories, encyclopaedias, etc.), numeric and statistical databases, e- images, e-audio/visual resources".*

Sedangkan definisi bukti digital menurut (Harbawi & Varol, 2017) adalah jejak yang dinginkan maupun tidak diinginkan yang berasal dari perubahan data digital pada perangkat elektronik. Berdasarkan sumbernya, bukti digital terbagi menjadi 2 kategori (Marshall, 2008), yaitu *closed system* dan *open system*. *Closed system* merupakan sistem yang pernah terkoneksi internet. Artinya, sistem tersebut sangat terisolasi dan hanya terhubung dengan sistem pada komputer yang lain. 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 ACPO (*Association of Chief Police Officer*) terdapat 4 prinsip dalam penanganan bukti digital. Berikut adalah 4 prinsip penanganan bukti digital menurut ACPO antara lain :

1. Prinsip 1, seorang penegak hukum tidak diperbolehkan untuk mengubah data yang terdapat pada komputer atau media penyimpanan karena hal ini akan dipertanggungjawabkan di pengadilan.
2. Prinsip 2, dalam situasi tertentu dan jika memang diharuskan, seseorang diperbolehkan untuk mengakses data yang asli, namun orang tersebut harus kompeten dan ia harus dapat menjelaskan tentang relevansi terhadap barang bukti serta implikasi terhadap kegiatan yang dilakukan terhadap barang bukti tersebut.
3. Prinsip 3, catatan dan audit yang berisi semua proses dalam penanganan barang bukti elektronik harus dibuat dan ketika pihak ketiga memeriksa catatan dan audit tersebut, hasilnya harus sama dengan yang dimiliki oleh pihak investigator.

4. Prinsip 4, orang yang bertanggung jawab dalam investigasi ini harus memastikan bahwa hukum dan semua prinsip ini dipatuhi oleh orang-orang yang terlibat.

2.2 Access Control

Menurut (Stallings, 2015), *access control* adalah merupakan *central* dari keamanan komputer. Selanjutnya menurut (Stallings, 2015), didasarkan pada fungsi tujuan utama dari keamanan komputer itu sendiri yaitu tercapainya tiga hal, yaitu mencegah pengguna yang tidak sah dari mendapatkan akses ke *resource*, mencegah pengguna yang sah dari mengakses *resource* secara tidak sah, dan untuk 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. (Sandhu, Security Models : Past , Present and Future San Antonio, TX, USA, 2010). Selanjutnya menurut (Karp, 2009), terdapat 4 isu utama dalam *access control*, yaitu *identification, authentication, authorization* dan *access decisions*. Penjelasan singkatnya adalah sebagai berikut:

1. *Identification* mengenali pihak yang akan bertanggung jawab terhadap *request access*, bisa berwujud orang ataupun NPE (*non person entity)* seperti halnya komputer, ataupun aplikasi.
2. *Authentication* adalah upaya untuk melakukan konfirmasi kebenaran suatu bagian dari data atau suatu *entity*. Pengguna *authentication* sendiri berarti melakukan konfirmasi data pengguna yang sebelumnya sudah tersimpan.
3. *Authorization* merupakan proses untuk menentukan layanan apa saja yang diperbolehkan untuk bisa digunakan oleh pengguna yang sudah jelas identitasnya (*authenticated* pengguna).
4. *Access Decision*: berdasarkan kombinasi dari tiga aspek diatas maka kemudian diberikan keputusan apakah *request* tersebut diijinkan ataukah ditolak oleh sistem.

Pada prinsipnya, *access control* merupakan fitur keamanan yang mengontrol bagaimana pengguna dan sistem berkomunikasi dan berinteraksi dengan sistem dan sumber daya lainnya. *Access control* melindungi sistem dan sumber daya dari akses yang

tidak berhak dan umumnya menentukan tingkat otorisasi setelah prosedur autentikasi berhasil dilengkapi.

Gambar 2. 1 ***Access control Systems***

2.3 Security Policy Statement

Sebuah *secure environment* sangat dipengaruhi oleh bagaimana penerapan dari *security policy* , *security model* serta *trust management system*. Secara umum *security policy* adalah kumpulan *statement* dan *requirement* dari *system behavior* sehingga akan terjamin wujudnya sebuah *secure system*. Sementara menurut Bishop (2004) dalam (Taylor, 2007), *security policy* adalah sebuah *statement* yang secara jelas menspesifikasikan apa yang boleh dan apa yang tidak boleh dalam lingkup keamanan. Pada level yang lebih rendah *security policy* akan berisi kumpulan kebijakan tentang otorisasi dan *secure states*.

Selanjutnya, menurut Clark dan Wilson (1987) dalam (Taylor, 2007) disebutkan bahwa untuk lingkungan penegak hukum, *security policy* juga harus memuat kebijakan tentang *confidentiality of classified data*. Dalam hal ini, semua *classified data/information* harus terproteksi dan hanya pengguna dengan level tertentu saja yang memiliki hak akses terhadap data dan informasi tersebut. Selain itu juga harus ada peraturan dan kewajiban yang mengikat untuk setiap pengguna yang akan memanfaatkan *classified data* tersebut.

Security Model merupakan sebuah abtraksi yang menyediakan bahasa konseptual yang akan digunakan oleh administrator untuk mengimplementasikan *security policy*. Umumnya *security model* akan mendefinisikan hierarki dari akses atau modifikasi hak yang dapat dimiliki oleh pengguna dari institusi. Sementara itu, *Trust Management System* adalah sebuah *framework* untuk menentukan apakah *security policy* yang diekspresikan lewat logika dan abstraksi serta diimplementasikan lewat pemrograman atau setting sistem

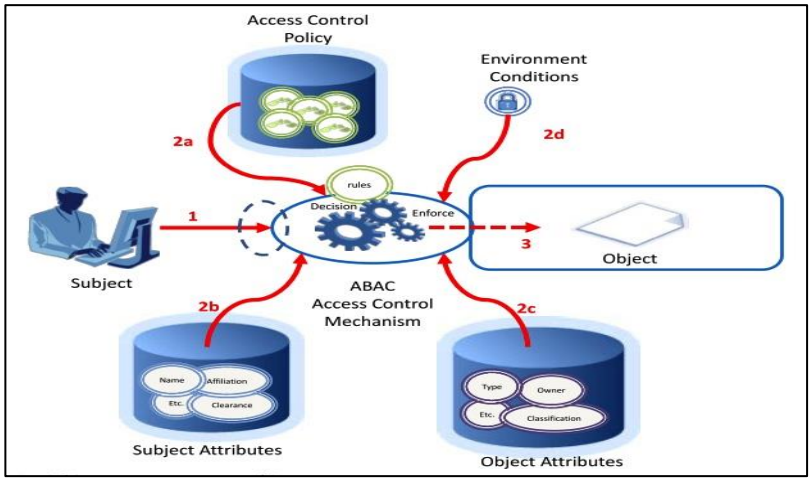
telah benar-benar memenuhi ketentuan *policy* yang seharusnya diikuti. *Trust management system* umumnya diterapkan melalui *policy language* dan *compliance checker*.

2.4 Attribute Based Acccess Control (ABAC)

Attribute Based Access control (ABAC) sebagai sebuah model untuk menerapkan *access control policy*, diprediksi menjelang tahun 2020, ABAC ini akan menjadi standar dan akan lebih banyak diterapkan oleh industri (Jin, 2014). Menurut (Hu, 2015), salah satu pertimbangan mengapa ABAC akan lebih banyak digunakan adalah dalam hal fleksibilitas dalam penerapan atribut terhadap pengguna, *object* dan kondisi lingkungan sehingga akan semakin banyak *rule* dan *policy* yang dapat diekspresikan untuk menggambarkan *access control policy* . Untuk itulah sejumlah peneliti seperti halnya (Burmester et al. 2013; Jin 2014; Yang dan Jia 2012; Smari et al. 2014) 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).

Selanjutnya menurut Sandhu (2010), terdapat 4 aspek *attribute* 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. Sementara itu *request access* dapat menggunakan atribut individual dari subjek atau kombinasinya untuk menunjukkan identitas yang unik.
2. *Resource* adalah sesuatu target yang diproteksi seperti halnya *device, files, record,* tabel, 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 attribute* adalah karakteristik dari operasional ataupun situasional seperti misalnya *current time, current temperature, IP address*.



Gambar 2. 2 **Gambaran Umum Cara Kerja ABAC**

Sebagai sebuah generasi baru dari *access control model*, menurut (Xu, 2014), ABAC memiliki sejumlah *feature* yang lebih baik dari model pada generasi sebelumnya. Di antara *feature* tersebut adalah :

1. ABAC memungkinkan pemberian *grant access control* melalui kombinasi dari sejumlah attribute dari elemen otorisasi seperti *subject, resource, action* dan *environment* menjadi satu keputusan *access control*. Teknik ini juga memungkinkan untuk seluas mungkin cakupan subjek untuk mengakses seluas mungkin cakukan *resource* tanpa adanya hubungan individual antara setiap subjek dengan setiap *resource*.

2. ABAC memfasilitasi *collaborative policy administration* di dalam sebuah organisasi besar ataupun antar organisasi yang berbeda. Individual *policy* dapat disusun oleh seorang pembuat *policy* yang berasal dari berbagai *department* ataupun organisasi yang berbeda. Dalam sebuah perusahaan besar, elemen dari otorisasi *policy* dapat dikelola oleh *department* yang berbeda.

3. ABAC memfasilitasi proses *decoupling access control* dari logika bisnis dari sebuah aplikasi tertentu. Hal ini akan menyebabkan meningkatnya sifat dinamis dari *access control*. Bila keputusan *access control* terpisah dari kode aplikasi, perubahan terhadap *access control policy* akan menyebabkan munculnya modifikasi minimal dari kode aplikasi. Mekanisme *decoupling* juga membuat lebih nyaman untuk melakukan verifikasi apakah akan melakukan akses kontrol dan kebutuhan fungsional adalah

sesuai ataukah tidak. Lebih lanjut ABAC bersifat kompatibel dengan konsep tradisional *access control* sebelumnya seperti DAC, MAC, ACL dan RBAC.

Mengingat ABAC memungkinkan sistem otorisasi bersinggungan dengan banyak sekali atribut yang berbeda satu sama lain, maka sistem ABAC akan menjadi semakin komplek untuk dikelola. Karena itu sangatlah diperlukan sebuah bahasa spesifikasi (*specification language*) untuk menjelaskan *policy* ABAC yang demikian kompleks. Namun pada sisi yang lain, semakin kompleksnya ABAC juga akan menimbulkan meningkatnya kemungkinan munculnya *defect*. Karena itu salah satu tantangannya adalah bagaimanakah cara untuk menjamin bahwa *policy* ABAC telah sesuai dengan spesikasi dan bersifat *correctly*.

Sebuah *policy* ABAC adalah merupakan representasi dari fungsi yang menentukan apakah permintaan akses dibolehkan berdasarkan nilai *attribute* yang diberikan. Secara formal sebuah *policy* ABAC akan memuat triple (X,Y,f). Dimana

1. X adalah himpunan finite dari atribut dengan domain D1…Dn
2. Y adalah himpunan finite dari *access control decision* (misalnya : *permit, deny , undefined*)
3. F := D1 x D2 x … Dn → Y ; inilah fungsi akses *control*

Sebuah *policy* ABAC dikatakan sebagai *complete* bila dan hanya bila f adalah fungsi total, dimana untuk nilai yang diberikan dari setiap atribut maka f selalu menghasilkan sebuah *deterministic decision*. Dalam hal ini sistem ABAC yang berbeda akan menggunakan himpunan keputusan *access control* yang berbeda, misalnya: {*permit, deny , undefined*} atau {*permit, deny , NotApplicable, Intermediate*}.

Bahasa spesifikasi ABAC yang saat ini ada, menyediakan berbagai pendekatan berbeda untuk menspesifikasikan fungsi *access control* dengan menggunakan *rule*. *Completeness* biasanya didapat dengan menggunakan *default decision* (misalnya : *deny*) untuk kondisi *unspecified situations*. Penggunaan *rule* juga memunculkan isu konflik atau inkonsistensi, yaitu sebuah *rule* menghasilkan *decision* yang berbeda untuk nilai atribut yang sama. *Conflict of rule* atau *policies* dapat diatasi secara eksplisit dengan menggabungkan algoritma atau secara implisit dengan menggunakan prioritas.

2.5 Extended Access Control Markup Language (XACML)

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 (Abd El-Aziz dan Kannan 2013).

XACML menyediakan mekanisme modularisasi untuk mengatasi *policy* yang kompleks. Pada satu sisi, spesifikasi XACML adalah bersifat *hierarchical*, memuat satu atau lebih himpunan *policy* dan setiap himpunan *policy* memiliki daftar *rules*. Penggunaan *multiple policy* dimungkinkan untuk mendukung *collaborative policy administration* (misalnya pada beberapa *department* yang berbeda) dan dapat pula dibagi dalam sub *policy* yang lebih kecil untuk mengurai kompleksitas *policy*. Sementara pada sisi yang lain komponen target yang direpresentasikan oleh *atribut matching condition* menyediakan level abtraksi dan meningkatkan *performance* dari *attribute matching*.

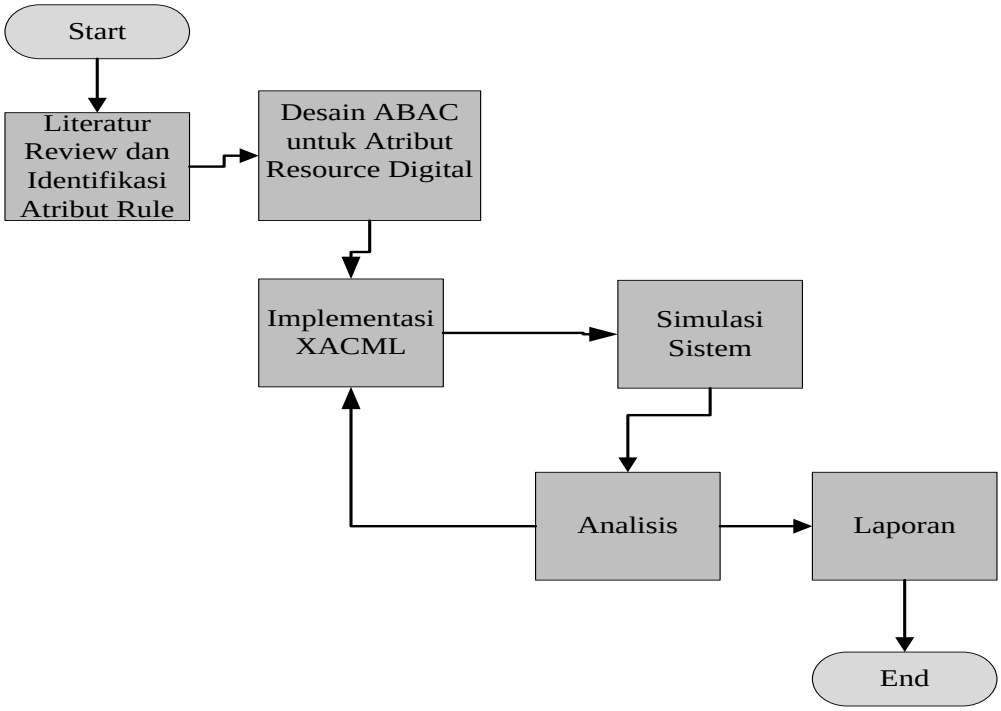
Menurut Aqib dan Ahmed Shaikh (2014), secara umum terdapat 2 masalah yang dihadapi dalam menerapkan solusi *access control* yaitu *inconsistency* dan *incompleteness*. Penjelasan singkatnya adalah sebagai berikut :

1. *Inconsistency* adalah kondisi dimana terdapat dua *rule* yang memberikan hasil yang kontradiksi. Bila S, O dan A adalah *Subject*, *Object* dan *Actions*. Bila diberikan a∈ A, s∈S, o∈O, kemudian diberikan d∈D yaitu himpunan Decision D = { *permit*ted, denied, undefined} serta r ∈R berupa *three tuple rule* (s,o,a) □ d. Sebuah *policy* dikatakan sebagai *inconsistency* bila untuk setiap dua buah *rule* r_(i) dan r_j ∈R, dimana i ≠j maka untuk r_(i)→d_i dan r_j→d_j dimana dimana i ≠j maka r_i dan r_j akan memberikan hasil *decision* yang kontradiksi.
2. *Incompleteness* adalah kondisi dimana terdapat *rule* yang belum terakomodasi dalam himpunan *rule* yang sudah didefinisikan sebelumnya. Yaitu terdapat r untuk satu keadaan dimana r ∉R.

Untuk melakukan validasi terhadap *access control policy* , yaitu memastikan tidak terjadi masalah *inconsistency* dan *incompleteness* dari *policy* yang telah dibangun maka dilakukan pengujian melalui sejumlah pendekatan. Dalam hal ini menurut Aqib dan Ahmed Shaikh (2014) terdapat 5 pendekatan umum untuk melakukan validasi acces control *policy* yaitu : *Mining technique, model checking technique, formal methods, matrix based approaches, mutation testing* dan *other technique.*

BAB 3
Metodologi Penelitian

Bab ini menjelaskan bagaimana skema penelitian ini dilakukan, sehingga dapat memberikan rincian tentang alur atau langkah-langkah yang dibuat secara sistematis serta dapat digunakan dengan jelas dalam menyelesaikan masalah, membuat analisis terhadap hasil penelitian. Adapun skema penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3. 1 Alur Metodologi Penelitian

3.1 Literatur Review dan Identifikasi Atribut *Rule*

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.

Sedangkan identifikasi atribut dari aktor dan sistem dilakukan untuk tujuan terhadap dilakukannya penelitian yang terkait dengan atribut-atribut yang terkait untuk sarana pendukung bahwa akses terhadap *resource digital* itu tidak sesederhana autentikasi dan

otorisasi, 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. Sementara itu *request access* dapat menggunakan atribut individual dari subjek atau kombinasinya untuk menunjukkan identitas yang unik.
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*.

3.2 Desain Atribut Bukti Digital dan Memodelkan Interaksi Akses Kontrol

Pada tahap ini akan dilakukan tahap perancangan desain atribut yang terdapat pada *resource digital* yang akan digunakan sebagai objek penelitian. Terdiri dari beberapa komponen berupa:

Tabel 3. 1 Daftar Atribut Subject

No	Pengguna	Atribut					*Action*
		Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5	
1	Pengguna *1*	√	√	√	√	√	*Permit*
2	Pengguna *2*	√	√	√	√	√	*Permit*
3	Pengguna *3*	√	√	√	√	√	*Permit*
4	Pengguna *4*	√	√	√	√	√	*Permit*
5	Pengguna *5*	√	√	√	√	√	*Permit*

Tabel 3.1 menjelaskan bahwa 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.

Tabel 3. 2 Daftar Atribut *Resource*

No	*Resource*	Atribut					*Action*
		Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5	
1	RD 1	√	√	√	√	√	*Permit*
2	RD 2	√	√	√	√	√	*Permit*
3	RD 3	√	√	√	√	√	*Permit*
4	RD 4	√	√	√	√	√	*Permit*

Tabel 3.2 menjelaskan bahwa atribut dari daftar atribut *resource* yang diberikan pada setiap aktor merupakan atribut yang berkaitan dengan perangkat yang digunakan oleh aktor, penandaan perangkat yang digunakan aktor saat melakukan akses ini akan menjadi salah satu identitas yang menandakan bahwa akun tersebut digunakan oleh aktor pada perangkat yang sudah didaftarkan sebelumnya, sebagaimana yang terlihat pada tabel di atas. Atribut yang diberikan berupa *device*, *file*s, *record*, dan tabel.

Tabel 3. 3 Daftar Atribut Operation

No	*Operation*	Atribut					*Action*
		Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5	
1	*Creation*	√	√	-		√	*Permit*
2	*Modification*	-	-	√	-	√	*Permit*
3	*Deletion*	-	-	√	√	-	*Permit*

Tabel 3.3 menjelaskan bahwa atribut yang diberikan pada aktor melalui daftar atribut *operation* ini adalah atribut yang berkaitan dengan hal-hal apa saja yang dapat dilakukan aktor ketika sudah diberikan izin akses. Proses ini merupakan pemetaan hak akses yang akan diberikan pada setiap aktor. Atribut yang diberikan akan langusung menentukan kegiatan yang dilakukan sebagai mana yang terlihat pada tabel di atas. Atribut yang diberikan berupa *creation*, *modification* dan *deletion*.

Tabel 3. 4 Tabel Daftar Atribut *Environment*

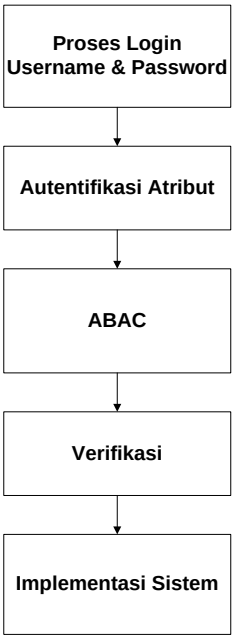
No	*1.3 Environtment*	Atribut					*Action*
		Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5	
1	*Current time*	√	√	√	√	√	*Permit*
2	*Current location*	√	√	√	√	√	*Permit*
3	*Current Date*	√	√	√	√	√	*Permit*

Tabel 3.4 menjelaskan bahwa daftar atribut *environment* ini yaitu atribut yang berkaitan dengan kondisi saat dilakukannya akses oleh masing-masing aktor, atribut ini bertujuan untuk dapat memastikan bahwa saat melakukan akses telah sesuai dengan ketentuan yang diberikan, setiap aktor diberikan waktu akses yang berbeda-beda antara satu dengan yang lain, waktu akses yang diberikan berdasarkan jabatan serta tugas pokok yang sudah ditentukan. Atribut yang diberikan berupa, *current time, current temperature, IP address.*

3.3 Implementasi XACML

Implementasi paket XACML yang menyediakan berbagai jenis *predefined attribute-matching predicates* (misalnya: *name-match and string-equal*) untuk mendukung *attribute types and expressions*. Hal ini memungkinkan dibangunnya pengguna-*defined predicates* dari kondisi *predefined and user-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..

Untuk memudahkan dalam memahami sistem implementasi XACML pada *resource bukti digital* dapat dijelaskan sebagai berikut:



Gambar 3. 2 Prinsip Kerja Sistem

3.4 Simulasi Sistem

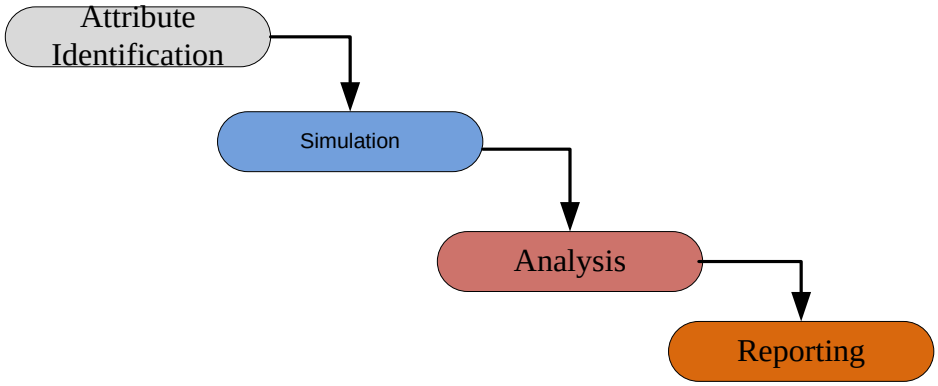
Merupakan tahap dilakukannya simulasi sistem untuk mencoba skenario dalam mengimplementasikan ABAC dalam mendeteksi *resource digital* untuk memberi informasi dengan implementasi XACML. Simulasi sistem bertujuan untuk melakukan pengujian terhadap timestamp *resource digital* dalam menemukan metode system keamanan dari *resource digital*. 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 siapa 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. Seperti proses *login* pada sistem perpustakaan, yang akan melalui *rule policy* pada saat pengguna memasukan *username* dan *password*, secara otomatis sistem akan memverifikasi *rule policy* yang telah dimasukan sesuai dengan ketentuan yang ada.

3.5 Analisis

Tahap pada bagian analisis merupakan tahap ketika akses kontrol dianalisis dan dinyatakan lulus uji. Akses kontrol dinyatakan lulus uji dengan melihat hasil tabel pengujian aktivitas. Pada tahap ini, pembuatan rancangan akses kontrol sistem perpustakaan menggunakan bahasa pemrograman Java, XACML 5.0, ALFA, dan *windows builder* sebagai *compiler*. Setelah dinyakatan lulus uji, langkah berikutnya adalah penjelasan tentang dan kesimpulan dari keberhasilan sistem yang telah dibuat.



Gambar 3. 3 Model Proses Implementasi

Dari model proses forensik tersebut maka pengujian yang dilakukan pada adalah pengujian fungsionalitas. Adapun pengujian fungsionalitas meliputi kemampuan menerapkan *rule policy* untuk proses *login* pada sistem, mampu mengidentifikasi setiap pengguna yang melakukan akses pada aplikasi. Akses kontrol yang dibangun harus berjalan dengan baik ketika diuji. Selain itu, sistem juga harus mampu memilah pengguna

yang masuk berdasarkan atribut yang ada ketika pengguna *name* dan *password* yang dimasukan. Hal ini bertujuan untuk meningkatkan keamanan sistem digital yang diakses oleh orang yang tidak memiliki kepentingan. Untuk mempermudah klarifikasi kondisi *resource digital* dan aktor yang mengakses akan dikelompokkan menggunakan data analisis sebagai berikut:

Tabel 3. 5 Rancangan Pengujian untuk *Subject*

No	***Pengguna***	***Atribut***						***Detail***
		(1)	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	
1	*Subject 1*	*√*	*√*	*√*	*√*	*√*	*√*	*Yes*
	Subject 1	-	*√*	*√*	*√*	*√*	*√*	*No*
	Subject 1	-	-	*√*	*√*	*√*	*√*	*No*
	Subject 1	-	-	-	*√*	*√*	*√*	*No*
	Subject 1	-	-	-	-	*√*	*√*	*No*
	Subject 1	-	-	-	-	-	*√*	*No*

Tabel 3. 6 Tabel skema Pengujian untuk *Resource*

No	**Daftar**	**Atribut**						**Detail**
		(1)	(2)	(3)	(4)	(5)	(6)	
1	*Resource 1*	√	√	√	√	√	√	*Yes*
	Resource 1	-	√	√	√	√	√	*No*
	Resource 1	-	-	√	√	√	√	*No*
	Resource 1	-	-	-	√	√	√	*No*
	Resource 1	-	-	-	-	√	√	*No*
	Resource 1	-	-	-	-	-	√	*No*

Tabel 3. 7 Tabel Skema Pengujian untuk *Operation*

No	**Daftar**	**Atribut**						**Detail**
		(1)	(2)	(3)	(4)	(5)	(6)	
1	*Operation 1*	√	√	√	√	√	√	*Yes*
	Operation 1	-	√	√	√	√	√	*No*
	Operation 1	-	-	√	√	√	√	*No*
	Operation 1	-	-	-	√	√	√	*No*
	Operation 1	-	-	-	-	√	√	*No*
	Operation 1	-	-	-	-	-	√	*No*

Tabel 3. 8 Skema Pengujian untuk *Environment*

No	**Daftar**	**Atribut**						**Detail**
		(1)	(2)	(3)	(4)	(5)	(6)	
1	*Environment 1*	√	√	√	√	√	√	*Yes*
	Environment 1	-	√	√	√	√	√	*No*
	Environment 1	-	-	√	√	√	√	*No*
	Environment 1	-	-	-	√	√	√	*No*
	Environment 1	-	-	-	-	√	√	*No*
	Environment 1	-	-	-	-	-	√	*No*

Dari data tersebut nantinya akan ditemukan atribut-atribut tempat dimana alamat, jenis, waktu yang digunakan, darimana asalnya, dan bagaimana itu terjadi yang berguna untuk membuktikan, mengumpulkan dan menganalisis serta melakukan proses investigasi

dari segala aktivitas dari *history*, serta dapat mengetahui dengan grafik diagram yang paling dilakukan pada sistem.

3.6 Laporan

Merupakan tahap pembuatan laporan dan hasil pembuktian implementasi desain atribut pada ABAC dengan XACML terhadap kinerja dari sistem akses kontrol terutama pada konsep otorisasi dan verifikasi serta dapat mendapatkan informasi akses terhadap *resource digital* itu tidak sesederhana autentikasi dan otorisasi agar dapat mengurangi tingkat kerentanan terhadap akses kontrol. Laporan berisi mengenai pendahuluan, literatur review, metodologi penelitian, hasil dan pembahasan, serta penutup.

BAB 4
PEMBAHASAN

Bab ini menjelaskan bagaimana langkah-langkah penelitian yang dilakukan, analisis dan hasil yang didapatkan dari penelitian ini. Pembahasan dalam bab 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.

4.1 *Literatur Review Policy Statement* dan Identifikasi Atribut dari Aktor

4.1.1 Literatur review Policy Statement

Literatur review Policy Statement 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 ataupun teori. Sedangkan identifikasi atribut dari aktor dan sistem dilakukan untuk tujuan terhadap dilakukannya penelitian yang terkait dengan atribut-atribut yang terkait untuk sarana pendukung bahwa akses terhadap *resource digital* itu tidak sesederhana autentikasi dan otorisasi, berikut juga metode yang digunakan agar dapat menunjang tujuan akhir dalam penelitian ini. Beberapa penelitian menggunakan berbagai *resource digital* dan menggunakan banyak metode, di antaranya

1. Membangkitkan XACML *Request* Menggunakan Framework X-CREATE

Pendekatan yang dilakukan dalam sistem tersebut adalah pengujian XACML *policy* dengan menggunakan XACML *request* sebagai masukan dari PDP. Pendekatan yang digunakan adalah XACML *Request* dari PDP yang ditulis secara manual dalam format xml. X-CREATE dijalankan dengan melengkapi properti dari bagian tersebut dengan menghasilkan folder *temp* sebagai tempat penyimpanan XACML *request* dan tabel yang digunakan di dalam *database*. Berdasarkan penelitian tersbut didapatkan beberapa masukkan berupa file XACML *policy* dan XACML *request*. Beberapa atribut tertera di dalam Tabel 4.1.

Tabel 4. 1 Database Policy XACML Request

AttributeId	**Data*Type***	**Attribute*Value***
*Subject*Set		
Role	String	professor
		researcher
		staff
***subject*-id**	String	Julius
*Resource*Set		
***resource*-id**	anyURI	*http://library.com/record*
***resource*-id**	anyURI	*http://library.com/record/iournals*
*Action*Set		
***action*-id**	String	read
***action*-id**	String	write

Pengujian XACML *policy* dapat dilakukan dengan memeriksa respon yang diberikan oleh PDP. Pengujian dilakukan dengan menggunakan file beberapa XACML *policy* serta menggunakan beberapa pendekatan di antaranya *simple combinatorial strategy, XPT strategy dan multiple combinatorial strategy* dengan pengontrolan akses menggunakan XACML *policy.*

2. Menggunakan UMU Editor pada sistem LPBD

Pendekatan yang digunakan dalam sistem lemari penyimpanan bukti digital (LPBD) ini adalah menggunakan model ABAC dengan pembuatan konsep XACML *policy*. Akses kontrol yang dirancang menggunakan *tools* UMU Editor dan implementasi dengan program *Python*. Skenario yang diujikan berjalan dengan baik dan berfungsi sebagaimana mestinya yang diharapkan. *Policy Decision Point* (PDP) yangada di dalam sistem ini berfungsi untuk mengevaluasi XACML *policies* yang berada pada *Policy Administration Point* (PAP) yang berfungsi sebagai yang mengolah XACML *policy* tersebut. Pendekatan dengan metode tersebut menjadi solusi keamanan sistem LPBD khususnya dalam hal identifikasi pengguna. Rancangan ABAC pada LPBD ini dapat meningkatkan tingkat keamanan sistem LPBD yang berisi bukti digital yang harus tetap terjaga keasliannya.

4.1.2 Identifikasi Atribut dari Rule

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*. *Type* data atau domain dari atribut pada ABAC dapat dispesifikasikan secara eksplisit ataupun implisit. *Policy* dan *rule* merupakan 2 komponen yang saling terintegrasi dalam perancangan ABAC dan nilai yang akan ditempelkan pada setiap atribut yang ada pada elemen. Atribut berfungsi sebagai aturan *policy* pada saat *request* dilakukan. Beberapa usulan *Rule* dan Atribut yang sering digunakan dengan studi kasus perpustakaan teruraikan di Tabel 4.2.

Tabel 4. 2 Usulan Atribut *Rule* Sistem Perpustakaan

No	*Rule*	*Subject*	*Resource*	*Actions*	*Environment*
1	*Rule* 1	Kepala Perpustakaan	*Upload Digital Book*	*Upload*	*IP Address* *Mac Address* *Time Access*
			View Statistic	*View*	
			Create Session	*Create*	
2	*Rule*2	Pustakawan	*Download Digital Book*	*Download*	*IP Address* *Mac Address* *Time Access*
			Upload New Book Arrived	*Upload*	
			Complete Data Book	*Complete*	
3	*Rule*3	Petugas IT/Admin	*Delete Old Inventory*	*Delete*	*IP Address* *Mac Address* *Time Access*
			Change Password Pengguna	*Change*	
			Validate Digital Book	*Validate*	
			Upload Foto	*Upload*	
			Validate Data Pengguna	*Validate*	

Tabel di atas merupakan penjelasan *policy statement* yang direpresentatifkan ke dalam beberapa *rule* untuk studi kasus *resource digital* dalam akses ke sistem perpustakaan. Masing-masing *rule* mempunyai aturan kontrol sendiri untuk menjadi aturan pengaksesan setiap *request* yang ada di dalam sistem tersebut. Bahasa spesifikasi ABAC yang saat ini ada, menyediakan berbagai pendekatan berbeda untuk menspesifikasikan fungsi *access control* dengan menggunakan *rule*. *Completeness* biasanya didapat dengan menggunakan *default decision* (misalnya : *deny*) untuk kondisi *unspecified situations*. Penggunaan *rule* juga memunculkan isu konflik atau inkonsistensi, yaitu sebuah *rule* menghasilkan *decision* yang berbeda untuk nilai atribut yang sama. Beberapa usulan atribut *rule* dapat dipadankan sebagai berikut

1. Kepala Perpustakaan

*Rule*1 dengan identitas Kepala Perpustakaan yang dapat melakukan aktivitas setelah login pada system dengan *policy* yang diberikan. *Policy statement* yang disematkan ke dalam *rule* tersebut di antaranya, *subject* yang berisi atribut identitas pengguna dan *resource* yang berisi atribut *upload digital book, view statistic*, dan *create session* dengan *action* yang bisa dilakukan adalah *upload, view* dan *create* serta *environment* berupa *ip address, mac address*, dan *time access*. Setelah semua *policy* terpenuhi maka dapat *permit* untuk mengakses sistem tersebut. Ketika Kepala Perpustakaan sebagai atribut dari subjet *rule*1 melakukan *request* akses kontrol terhadap sistem maka atribut *resource* yang diberikan hak akses *permit* adalah *upload digital book, view statistic*, dan *create session*. Sedangkan atribut *action* yang diperbolehkan diakses adalah *upload, view* dan *create* dengan kondisi *environment* yang diberikan adalah *ip address, mac address*, dan *time access*.

2. Pustakawan

*Rule*2 dengan identitas Pustakawan yang dapat melakukan aktivitas setelah login pada system dengan *policy* yang diberikan di antaranya, *subject* yang berisi identitas pengguna dan *resource* yang berisi *download digital book, upload new book arrived*, dan *complete book data* dengan *action* yang bisa dilakukan adalah download, upload dan complete serta *environment* berupa *ip address, mac address*, dan *time access*. Setelah semua *policy* terpenuhi maka *rule*2 dapat diberikan control *permit* untuk mengakses system tersebut. Ketika Pustakawan sebagai atribut dari subjet *rule*2 melakukan *request* akses kontrol terhadap sistem maka atribut *resource* yang diberikan hak akses *permit* adalah download

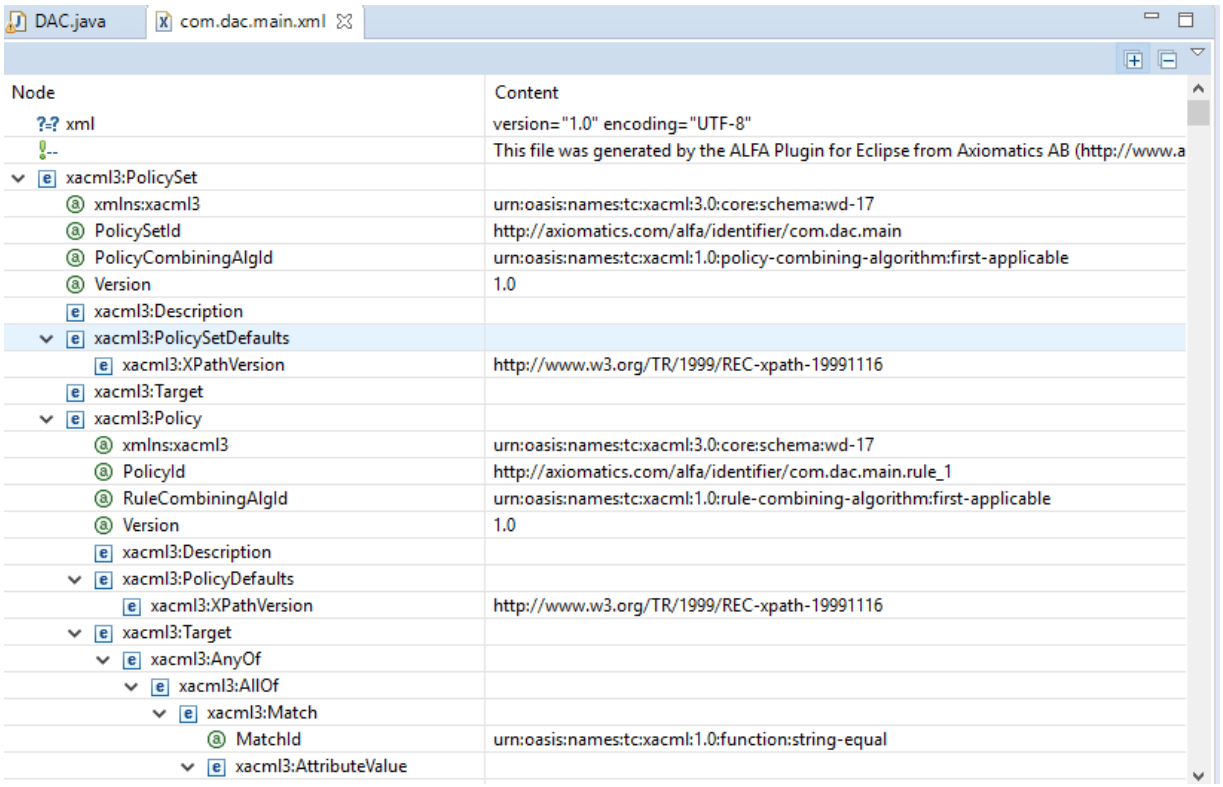
digital book, upload new book arrived, dan *complete book data* sedangkan yang lain akan dikenai akses *deny* . Sedangkan atribut *action* yang diperbolehkan diakses oleh *rule*2 adalah download, upload dan complete dengan kondisi *environment* yang diberikan adalah *ip address, mac address,* dan *time access*.

3. Petugas IT/Admin

*Rule*3 dengan identitas Petugas IT/Admin yang dapat melakukan aktivitas setelah login pada system dengan *policy* yang diberikan di antaranya, *subject* yang berisi identitas pengguna dan *resource* yang berisi *delete old inventory, change password* pengguna, validasi dengan *action* yang bisa dilakukan adalah upload, view dan create serta *environment* berupa *ip address, mac address,* dan *time access*. Setelah semua *policy* terpenuhi maka dapat *permit* untuk mengakses system tersebut. Setelah semua *policy* terpenuhi maka *rule*3 dapat diberikan control *permit* untuk mengakses system tersebut. Ketika Petugas IT/Admin sebagai atribut dari subjet *rule*3 melakukan *request* akses kontrol terhadap sistem maka atribut *resource* yang diberikan hak akses *permit delete old inventory, change password* pengguna, dan *validasi* sedangkan yang lain akan dikenai akses *deny* . Sedangkan atribut *action* yang diperbolehkan diakses oleh *rule*3 adalah *upload, view dan create* dengan kondisi *environment* yang diberikan adalah *ip address, mac address,* dan *time access.*

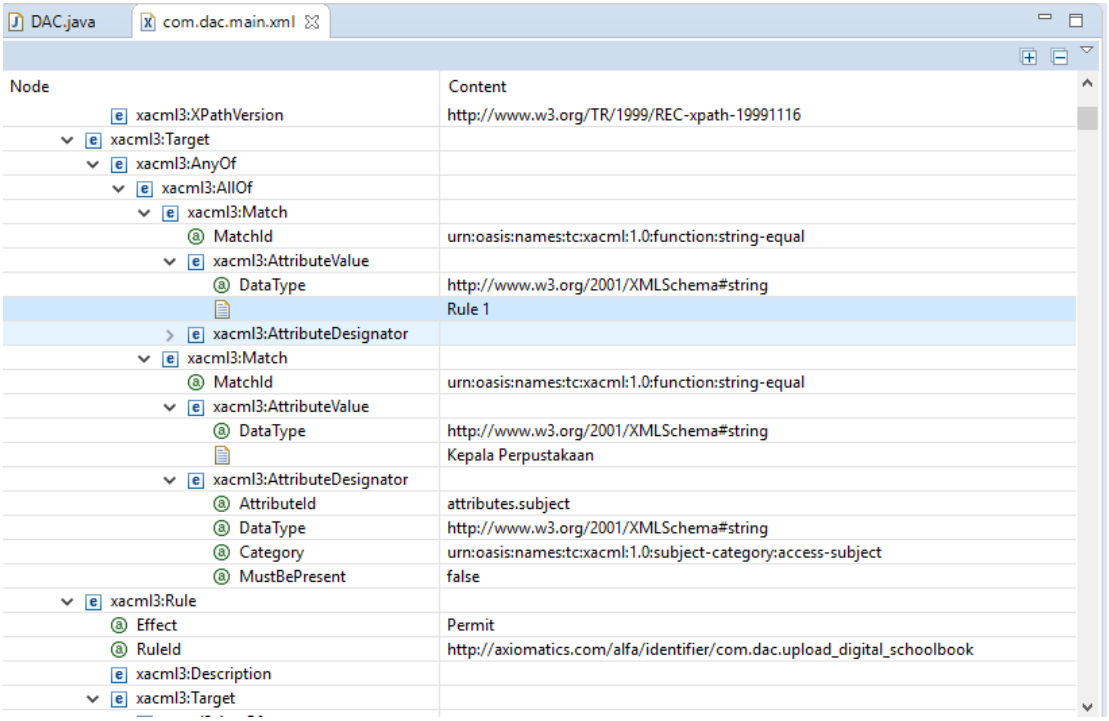
4.2 *Implementasi XACML*

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. Penggunaan ALFA sangat ideal karena fitur yang ada di dalam *Eclipse* seperti pengecekan sintaks dan fungsi *auto-complete*-nya sangat mudah untuk membuat *policy statement*. *Axiomatics* juga telah mengusulkan ALFA sebagai profil XACML dengan julukan "*Abbreviated Language for Authorization*". Pembuatan atribut dari *policy statement*-nya diawali dengan pembuatan struktur *policy* dengan XACML yang terlihat pada Gambar 4.1.



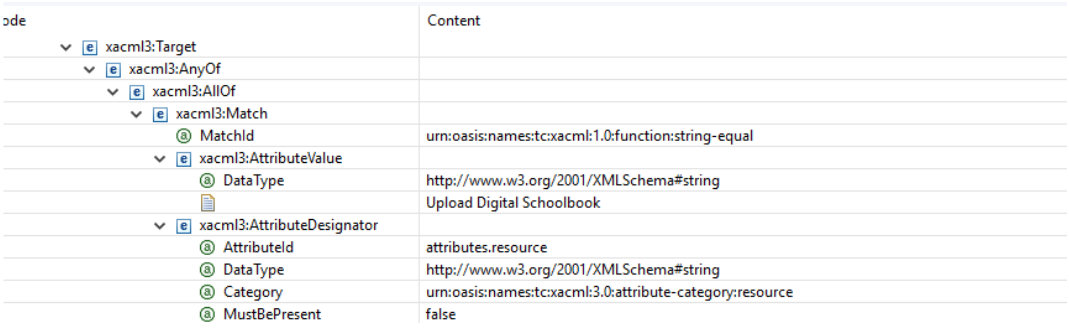
Gambar 4. 1 Struktur *Policy* Atribut ABAC.

Gambar 4.1 menjelaskan bahwa letak fleksibilitas ABAC itu terdapat pada aturan *policy* dimana sebuah *resource* dapat dioperasikan lebih dari satu permintaan akses (Lorch, Proctor, Lepro, Kafura, & Shah, 2003). 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 pengguna. Langkah pertama yang akan dilakukan yaitu menentukan nilai pada *policy* target yang juga merupakan root elemen pada rancangan *access control policy* ini, *policy*Id diberi nama **com.dac.main** dan *rule combining algorithm* diisi dengan nilai *first applicable* hal ini bertujuan karena *policy* mempunyai permintaan lebih dari 1 *rule*. Selanjutnya menjelaskan isi dari target *policy*, sabagaimana yang terlihat pada gambar di atas bahwa target *policy* ini berisi 3 *subject* yang artinya bahwa ke 3 pengguna tersebut merupakan jumlah keseluruhan pengguna yang diizinkan melakukan akses pada sistem di antaranya yaitu kepala perpustakaan, pustakawan dan admin. Selain *subject*, target *policy* juga berisi *resource* yang artinya merupakan keseluruhan *resource* yang ada dan bisa diakses oleh pengguna dengan ketentuan aturan yang ada pada *rule* yang ada di dalam Gambar 4.2.



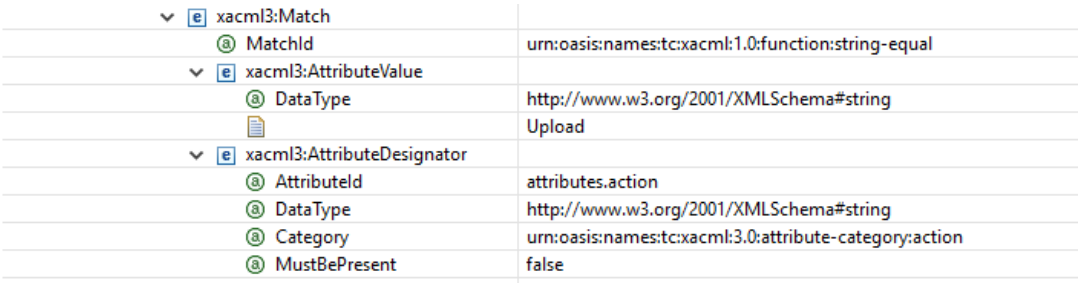
Gambar 4. 2 Atribut *Subject.*

Rule subject yang terdapat di Gambar 4.2 memperlihatkan semua atribut yang tersemat di dalam *rule subject* dari *MatchId, DataType, AttributeId, Category*, dan *MustBePresent.* Perancangan yang selanjutnya, *policy* yang dibangun dengan menambahkan *rule* diberi nama *rule*s yang dimana berisi aturan kebijakan yang diberikan kepada setiap pengguna yang terdiri dari elemen *subject*, *resource*, *action*, dan *environment*. *Rule*s 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. Elemen *subject* yang berisi nilai atribut yang dimana nilai atribut tersebut mengandung informasi yang ada pada pengguna. *Subject* matchId yang pertama diisi dengan nilai string dan mengisi nilai *attribute value* memiliki data *type*:string-equal, selanjutnya mengisi *subject attribute designator* nilai *attributeId* diisi dengan nilai *subject* nilai *type* data diberi nilai *string* dan pada *category* diberi nama Kepala Perpustakaan sebagai access *subject* yang menjadi jabatan dari pengguna yang ada di sistem perpustakaan yang atributnya tertera di Gambar 4.3



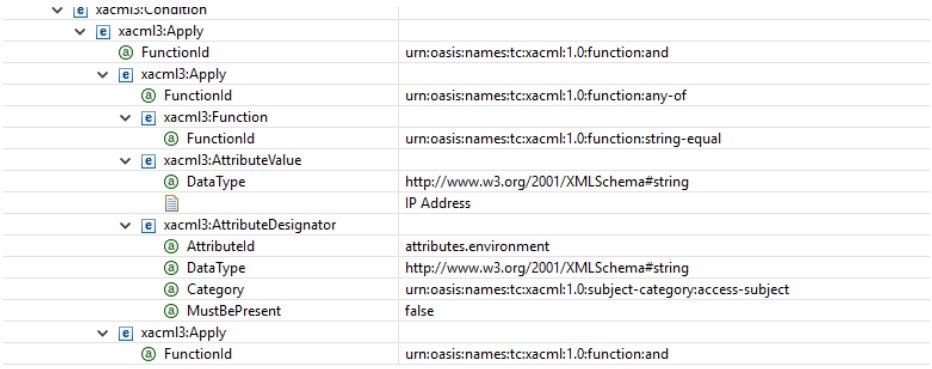
Gambar 4. 3 Atribut *Resource*.

Penentuan nilai *resource* match yang pertama diberi nilai function dari MatchId string-equal dan nilai attribute *value type* data: string dan *value* diberi nama upload digital schoolbook. *Resource* attribute designator mempunyai attributeId yang diberi nilai *resource*Id *type* data: string serta issuer diberi nama *upload digital schoolbook. new bag* dan *resource attribute designator* mempunyai *attributeId* diberi nilai attribute *value* : *upload* yang tertera di Gambar 4.4



Gambar 4. 4 Atribut *Action*.

Perancangan elemen *action*s yang bertujuan untuk menentukan *action*s yang diberikan pada pengguna, elemen *action*s ini berfungsi menentukan operasi yang dilakukan pada *resource*. Diawali dengan menentukan nilai *action* match diberi nilai *matchId*; string-equal dan form attribute *value* memiliki *type* data: *string* dan *value* diberi nama upload. Action attribute designator memiliki *attributeId* diberi nilai *action*Id *type* data: *string* serta *category* diberi nama *upload* seperti Gambar 4.5.



Gambar 4. 5 Atribut *Environment.*

Perancangan elemen *environment* pada pengguna *first responder* ini bertujuan untuk menentukan kondisi lingkungan saat *request* dilakukan dan berfungsi mengumpulkan informasi mengenai konteks akses yang dapat digunakan dalam membuat keputusan akses. Diawali dengan menentukan nilai *environment match* yang pertama dengan diberi nilai *function*: *string-equal*, *attribute value type* data: *string* dan nilai *action*: *enableRole*. *Environment attribute designator* diberi nilai *subject*: *string: ip address type data: string* serta *issuer* diisi dengan IP Address, *must be present* diberi nilai *true*.

4.3 *Output XACML*

XACML *policy* ini merupakan struktur *policy* yang sesuai dengan perancangan awal dan akan berfungsi sebagai aturan *policy* pada sistem *resource digital* pada sistem perpustakaan ini. Output hasil rancangan XACML *policy* yang pertama dijelaskan yaitu target dari root elemen yaitu elemen *policy*. Output XACML ini menggunakan format .xml dengan metode acces control yang bisa digunakan kembali untuk penelitian yang selanjutnya. Pada hasil ini menjelaskan bahwa hasil output XACML *policy* ini merupakan bagian dari beberapa sample dari keseluruhan jumlah elemen *subject* maupun *resource*. 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 yang artinya bahwa hanya ada 3 pengguna yang berhak melakukan akses pada sistem ini dan masing-masing dari pengguna tersebut adalah kepala perpsutakaan, pustakawan dan admin serta berhak mengakses *resource* yang ada sesuai dengan kebijakan *access* yang telah diberikan dan disemeatkan sebelumnya.

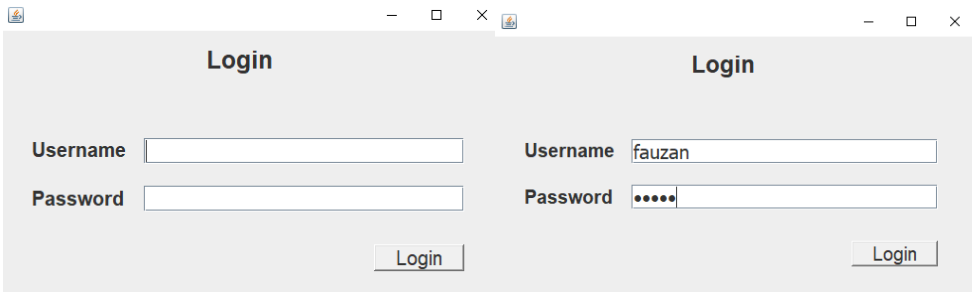
Resource yang akan diakses yaitu *upload digital schoolbook, view statistic, create, download digital book, upload new book arrived, complete the book data, delete old inventory, change password user, validate pinjam buku user, upload foto user,* dan *validate data user*. Rangkaian tersebut ditampilkan seperti Gambar 4.6.

```
<xacml3:PolicySetDefaults>
    <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
</xacml3:PolicySetDefaults>
<xacml3:Target />
<xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
    PolicyId="http://axiomatics.com/alfa/identifier/com.dac.main.rule_1"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
    Version="1.0">
    <xacml3:Description />
    <xacml3:PolicyDefaults>
        <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
    </xacml3:PolicyDefaults>
    <xacml3:Target>
        <xacml3:AnyOf>
            <xacml3:AllOf>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">Rule 1</xacml3:AttributeValue>
                    <xacml3:AttributeDesignator
                        AttributeId="role"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                        MustBePresent="false"
                        />
                </xacml3:Match>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">Kepala Perpustakaan</xacml3:Attribute
                    <xacml3:AttributeDesignator
                        AttributeId="attributes.subject"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                        MustBePresent="false"
```

Gambar 4. 6 Tampilan Output XACML.

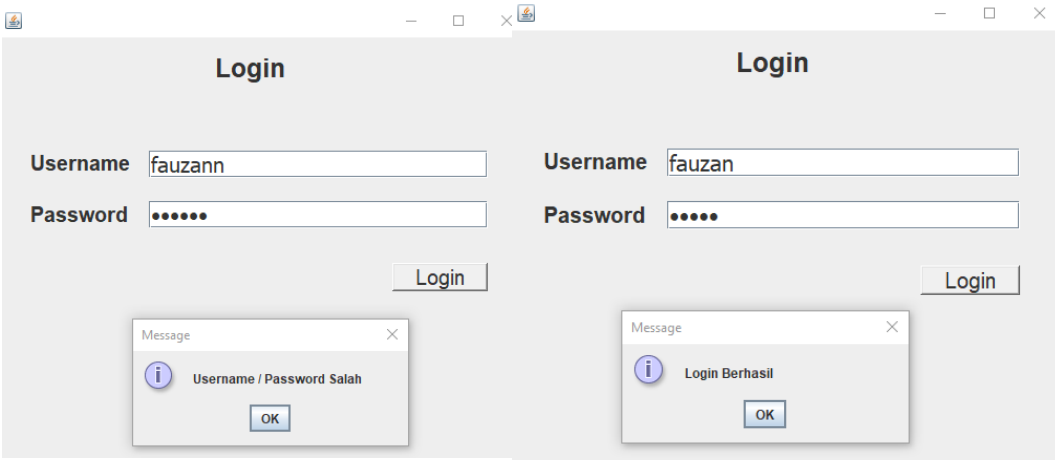
4.4 *Simulasi Sistem*

Sub bab ini menjelaskan tentang implementasi sistem yang dilakukan yakni dengan membuat rancangan *Attribute Based Access Control* (ABAC) dengan struktur XACML pada *resource digital* dengan studi kasus sistem akses kontrol ini dibuat menggunakan stuktur XACML serta bahasa pemrograman Java dan *windows builder* sebagai *complier*. *Complier* ini mendukung dengan bahasa pemrograman *Java* dan bisa *diexport* ke dalam ALFA. Penggunaan struktur XACML sangat mengukung untuk memenuhi semua kebutuhan yang digunakan saat merancang akses kontrol *policy* seperti *Attribute Based Access Control* (ABAC).



Gambar 4. 7 Halaman Login.

Gambar 4.7 menunjukkan halaman login untuk masuk ke dalam akses kontrol sistem *resource digital* pada sistem perpustakaan dengan mengisikan nama pengguna di kolom pengguna *username* dan isian kata kunci di kolom *password*.



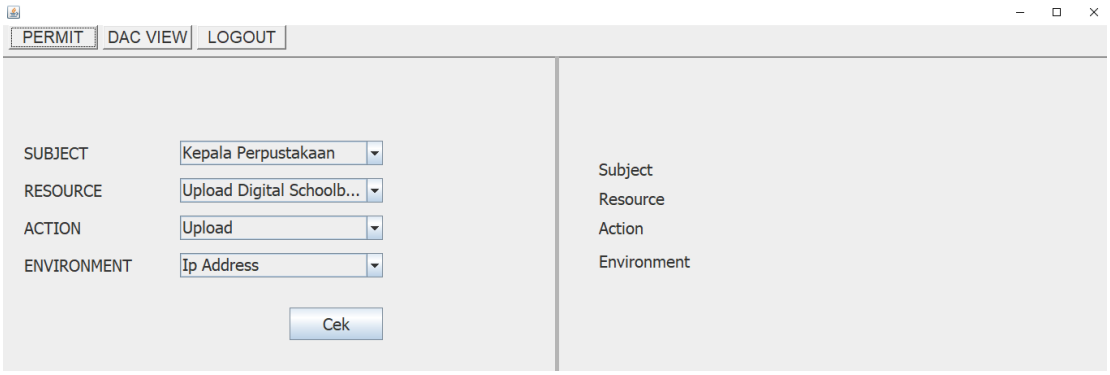
Gambar 4. 8 Halaman tampilan login gagal dan berhasil.

Gambar 4.8 menunjukkan tampilan *javascript* jika akses kontrol yang diberikan sesuai dan benar sehingga proses masuk ke sistem dianggap *succcess*. Sedangkan jika akses yang diberikan tidak sesuai maka akan keluar tampilan *javascript* akses yang diberikan salah sehingga tidak dapat masuk ke dalam sistem .

Rangkaian pengujian akses kontrol pada sistem ini akan dikategorikan baik ketika kondisi yang dihasilkan adalah *permit*, begitupun sebaliknya jika kondisi yang dihasilkan adalah *deny* maka akan dikategorikan tidak baik. Adapun rangkaian pengujian akses kontrol yang dilakukan yaitu menggunakan 2 *sample* data kondisi *permit* dan 2 *sample* data kondisi *deny* yang diuji coba menggunakan 3 pengguna yaitu kepala perpustakaan,

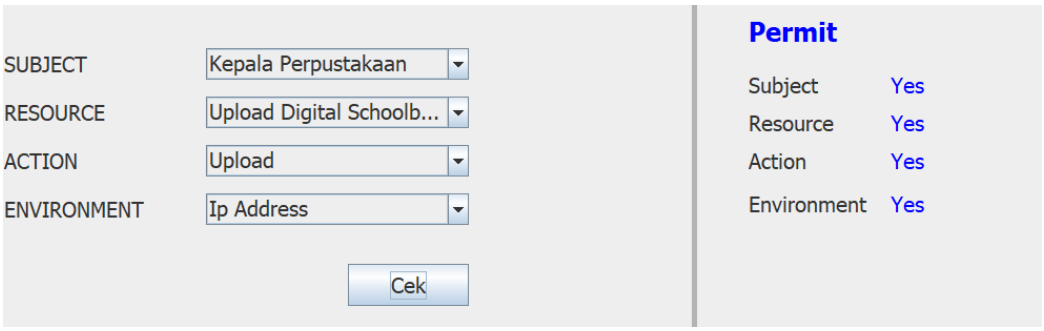
pustakawan dan admin. Pengujian yang dilakukan berdasar pada penjabaran yang ada pada simulasi dan skenario. Pengujian ini juga memiliki keterbatasan pada saat melakukan perubahan terhadap atribut *policy* yang berkaitan dengan perubahan kebijakan akses, pengujian ini juga hanya dapat dilakukan menggunakan desain awal *policy*, jika sewaktu-waktu kebijakan yang ada pada *policy* akan diubah, maka proses yang akan dilakukan yaitu mengubah struktur XACML *policy* sebagai proses awal melalui ALFA (*Axiomatics Language for Authorization*), kemudian disimpan dalam bentuk file dengan format xml dan membandingkan kembali dengan *source code* yang sudah disiapkan pada tools pengujian akses kontrol ini dalam bentuk format *file .xml*.

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*.



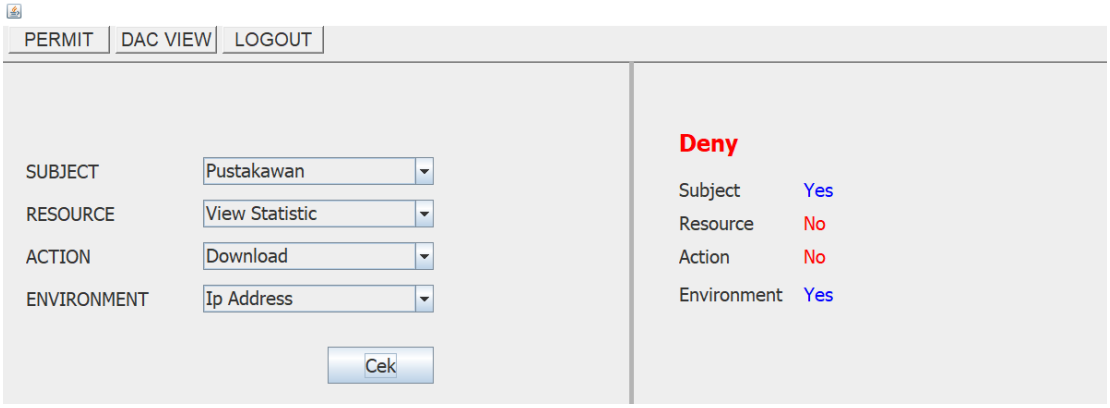
Gambar 4. 9 Halaman Pengujian.

Pengujian sample *permit* dilakukan oleh setiap *rule* dari beberapa atribut yang disematkan dengan masing-masing *subject* seperti Gambar 4.10



Gambar 4. 10 Halaman *Permit*.

Gambar di atas menjelaskan bahwa pada pengujian kondisi *permit* yang pertama ketika melakukan klik tombol "cek" maka kondisi yang dihasilkan yaitu *permit* dikarenakan semua atribut yang dimasukan benar seperti yang terlihat pada halaman info *permit* ***subject*:yes *resource*: yes *action*: yes dan *environment*: yes** atribut yang dimasukan diatas merupakan atribut yang mewakili kesuluruhan atribut yang ada pada *rule*s. Atribut yang telah disematkan pada *rule*1, *attribute* yang dimasukan yaitu *subject*: kepala perpustakaan, *resource*: *upload digital schoolbook*, *action*s: upload, dan *environment*: ip address. Selanjutnya melakukan pengujian sample kondisi *permit* yang oleh *rule* yang lain. Pengujian sample *deny* dilakukan oleh setiap *rule* dengan masing-masing *subject* seperti Gambar 4.11



Gambar 4. 11 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*

Berdasarkan studi kasus yang terdapat pada penelitian ini bahwa terdapat 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,

4.5.1. Konsep Model Akses Kontrol

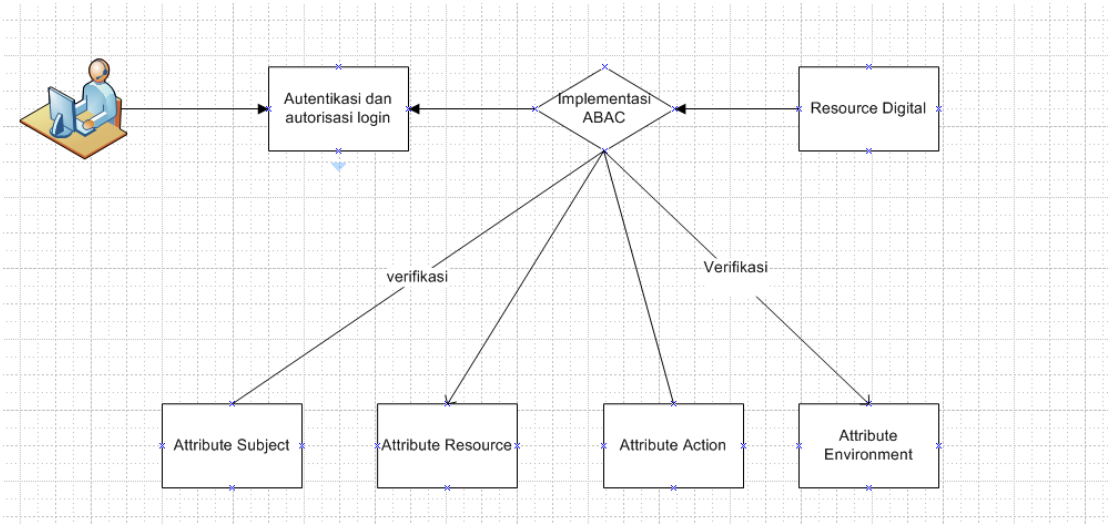
Pemodelan akses kontrol dari sistem yang dibuat hanya dengan mekanisme authentikasi, otorisasi dan verifikasi pengguna saja, tidak adanya parameter lain yang mendukung proses authetifikasi dan otorisasi yang lebih kompleks. Konsep tradisional *access control* sebelumnya seperti DAC, MAC, ACL dan RBAC masih belum kompatibel dibandingkan dengan ABAC karena mekanisme ABAC membuat lebih nyaman dalam melakukan verifikasi apakah akan melakukan akses kontrol dan kebutuhan fungsional sudah sesuai atau belum. Sistem ini dibuat berdasarkan pada permasalahan-permasalahan tentang manajemen *resource digital* saja, akan tetapi komponen-komponen penting lainnya seperti pada pengaturan akses kontrolnya. Perbandingan komponen dan metode *access control* yang bisa dijadikan usulan dalam penelitian ini tertera di Tabel 4.3

Tabel 4. 3 Perbandingan Metode Akses Kontrol dengan Metode yang Diusulkan

Component	**Methods of Access Control**		**Keterangan**
	Old	ABAC	
Username	√	√	
Password	√	√	
Autentifikasi	√	√	
Otorisasi	√	√	
Verifikasi	√	√	
Rule Policy		√	
Subject Attribute		√	Policy yang disematkan ke subject yang berupa identitas

Resource Attribute		√	Atribut yang disematkan ke resource yang berupa identitas
Action Attribute		√	Atribut yang disematkan ke action
Environment Attribute		√	Identitas device dan waktu pembagian waktu akses

Hal ini karena sistem ini menyimpan *resource digital* yang tidak sembarang orang dapat mengaksesnya. Metode yang digunakan dalam *resource digital* ini telah memenuhi aturan baku dari ABAC yaitu memakai atribut dari *policy* subjet, *resource*, *action*, dan *environment*. Rancangan yang dibuat dari sistem ini dipastikan bahwa keamanan dari *resource digital* akan benar-benar aman yang dibuktikan dari pengguna yang tidak bisa melakukan tindakan yang tidak sesuai denga *policy* yang telah ditetapkan. Tampilan model yang dirancang seperti di Gambar 4.12



Gambar 4. 12 Halaman *Model Perancangan.*

4.5.2. Konsep XACML

Perancangan struktur XACML *policy* yang telah dijabarkan sebelumnya bahwa salah satu fleksibilitas terhadap penggunaan ABAC sebagai *access control* model adalah adanya standard bahasa XML untuk mendukung desain *access control policy* dan penggunaan ABAC sebagai access control model, yaitu XACML (Abd El-Aziz & Kannan, 2013). Hasil uji fungsional XACML yang dibuat telah sesuai dengan kebutuhan akses kontrol yang yang diterapkan pada sistem ini, disebabkan struktur XACML yang dibuat telah dapat diimplementasikan menjadi sebuah rangkaian pengaturan akses berupa sebuah *form* log yang tertera di Tabel 4.4.

Tabel 4. 4 Analisis XACML Policy

XACML log	**ALFA log**	**Function**	
		Yes	No
xmlns	*urn:oasis:names:tc:xacml:3.0:core:schema:wd-17*	√	
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*	√	
AttributeValue	*DataType;*	√	

AttributeDesignator	*AttributeId; DataType; Category; MustBePresent*	√	
Environment Match	*MatchId*	√	
AttributeValue	*DataType;*	√	
AttributeDesignator	*AttributeId; DataType; Category; MustBePresent*	√	

XACML *log* yang dirancang di sistem perpustakaan ini menggunakan 1 *policy* dan 1 *rule*, untuk penggunaan aturan *firt applicable* pada *rule combining applicable* dikarenakan untuk dapat 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.

4.5.3. Pengujian dengan *Policy Statement*

Sebuah policy statement yang secara jelas menspesifikasikan apa yang boleh dan apa yang tidak boleh diakses dalam lingkup keamanan. Untuk itu *policy statement* yang diusulkan pada *access control* sistem perpustakaan ini yaitu menyesuaikan antara kebutuhan akses kontrol dan kebutuhan yang ada pada *resource* digital itu sendiri, pengujian *policy statement* dikatakan berfungsi dengan baik ketika semua komponen kebutuhan akses kontrol dan sistem telah terpenuhi. Kebutuhan-kebutuhan tersebut telah diimplementasikan dan dilakukan pengujian melalui rangkaian simulasi kasus dan pengujian akses kontrol. Pengujian dilakukan dua kali dengan pendekatan internal dan pengujian sampling eksternal di salah satu perpustakaan yang sudah ditentukan yaitu di Perpustakaan SMK Batur Jaya 1 Ceper Klaten.

Proses pengujian dilakukan pada logika internal untuk memastikan semua pernyataan sudah diuji. Pengujian eksternal fungsional untuk menemukan kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan. *Policy statement* yang digunakan dalam sistem perpustakaan ini telah diuji dan

dinyatakan sesuai dengan apa yang dibutuhkan oleh sistem sebagaimana yang terlihat pada tabel 4.5

Tabel 4. 5 Daftar Pengujian Eksternal

No	***Subject***	***Resource***	***Actions***	***Environment***	***Output Testing***
1	Kepala Perpustakaan (Wahyu Tri Mulyandari, M.Pd)	*Upload Digital Book*	*Upload*	*IP Address Mac Address Time Access*	*OK*
		View Statistic	*View*		*OK*
		Create Session	*Create*		*OK*
2	Pustakawan (Hasta A.F,S.I.Pust)	*Download Digital Book*	*Download*	*IP Address Mac Address Time Access*	*OK*
		Upload New Book Arrived	*Upload*		*OK*
		Complete Data Book	*Complete*		*OK*
3	Petugas IT/Admin (Nur Muhammad Sholahudin, S.I.Pust)	*Delete Old Inventory*	*Delete*	*IP Address Mac Address Time Access*	*OK*
		Change Password Pengguna	*Change*		*OK*
		Validate Digital Book	*Validate*		*OK*
		Upload Foto	*Upload*		*OK*
		Validate Data Pengguna	*Validate*		*OK*

Tabel 4.5 menjelaskan bahwa semua *policy statement* yang diusulkan sebelumnya telah berjalan dan berfungsi dengan baik sebagaimana mestinya yang diharapkan bahwa *policy statement* yang di usulkan dapat digunakan sebagai nilai dari atribut yang disematkan.

BAB 5
Kesimpulan dan Saran

5.1 *Kesimpulan*

Desain atribut yang diterapkan di salah satu resource digital dengan studi kasus sistem perpustakaan ini menunjukkan bahwa pendekatan dengan pendekatan model atribut menjadi solusi yang tepat dan relevan dalam mendukung proses untuk mendukungnya tingkat keamanan sistem *resource digital* secara general khususnya dalam hal identifikasi, otorisasi dan autentikasi pengguna. Dari studi kasus dengan objek sistem perpustakaan ini, diharapkan pendekatan ABAC dapat digunakan di berbagai contoh *resource* digital yang lain.

Salah satu metode untuk mengamankan data yang berada pada *resource digital* dengan cara metode akses kontrol dengan desain pendekatan atribut atau *attribute based access control (ABAC)*. Desain policy yang dibuat di sistem perpustakaan merupakan salah satu contoh *resource* digital yang akses kontrolnya dikembangkan dengan atribut-atribut dalam metode ABAC. Perancangan yang dibuat sesuai dengan struktur XACML dan validasi terhadap rancangan pengujian struktur XACML.Perancangan model ABAC diawali dengan pembuatan *policy statement* untuk dapat dengan memeriksa respon yang diberikan oleh *rule* yang diinput, serta diimplementasikan dalam bentuk model halaman login sebagai XACML *request*. Sample yang digunakan dalam sistem ini adalah sistem perpustakaan yang berada di SMK Batur Jaya 1 Ceper Klaten sebagai pengujian eksternal.

Berdasarkan hasil pengujian implementasi ABAC dengan sampel data uji melalui rangkaian skenario,simulasi dan pengujian kinerja akses kontrol menggunakan tools yang dibuat khusus untuk menguji kinerja ABAC pada sistem ini, didapatkan hasil bahwa akses kontrol yang dibuat telah berjalan dengan baik dan berfungsi sebagaimana mestinya yang diharapkan.

5.2 *Saran*

Adapun saran bagi peneliti selanjutnya yang berkenan mengembangkan akses kontrol yang telah dibuat ini, perlu memperhatikan beberapa faktor berikut ini yaitu:

1. Akses kontrol ini belum dilengkapi dengan uji coba dari protokol *Policy Decision Point* (PDP) *response* untuk memproses kebijakan *request response*.
2. Selain melakukan pengujian skema struktur XACML,masih diperlukan validasi terhadap rancangan strukur XACML yang telah dibuat.
3. 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 dan harus transparan di tampilan *backend*.

Daftar Pustaka

Ade Kurniawan, I. R. (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.

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

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

Henim, S. R. (2016). Membangkitkan XACML Request Menggunakan Framework X-CREATE.

Hsu, C.-l. a.-l. (2011). A Digital Evidence Protection Method with Hierarchical Access Control Mechanisms. *IEEE* (pp. 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.

Imam Riadi, S. &. (2017). Forensic Investigation Technique on Android's Blackberry Messenger using NIST Framework. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 198-205.

Imam Riadi, U. R. (2018.). Experimental Investigation of Frozen Solid State Drive on Digital Evidence with Static Forensic Methods. , . *Lontar Komputer*, 169-181.

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). 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.

Khalid Zaman Bijon, R. K. (n.d.). Constraint Specification in Attribute Based Access Control. 2015.

M Fadly Panende, Y. P. (2018). Konsep Attribute Based Access Control (ABAC) Pada Lemari Penyimpanan Bukti Digital (LPBD). *Jurnal Teknik Informatika Vol. 11 No. 1*, 85-94.

Mike Burmester, E. M. (2013). T-ABAC: An attribute-based access control model for realtime availability in highly dynamic systems. . *Symposium on Computers and*.

Nergaard, H. a.-m. (2015). A Scratch-Based Graphical Policy Editor for XACML. *Proceedings of the 1st International Conference on Information Systems*.

Ni dan, S. H. (2012). Attribute Based Access Control (ABAC)-Based Cross-Domain Access Control in Service-Oriented Architecture. *International Conference on Computer*.

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.

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, Y. P. (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.

X, R. N. (2014). XACML-Based Access Control for Decetralized OnlineSocial Networks. . *International Conference on Utility and Cloud Computing.*

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-C 2014*, 41–49. doi:10.1109/SERE.

(X, 2014) (Mike Burmester, 2013) (Ni dan, 2012) (Khalid Zaman Bijon)

LAMPIRAN

Lampiran Hasil Pengujian

No	***Pengguna***	***Atribut***						***Detail***	***Status***
		(1)	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*		
1	*Subject 1*	*√*	*√*	*√*	*√*	*√*	*√*	*Yes*	*Permit*
	Subject 1	*-*	*√*	*√*	*√*	*√*	*√*	*No*	*Deny*
	Subject 1	*-*	*-*	*√*	*√*	*√*	*√*	*No*	*Deny*
	Subject 1	*-*	*-*	*-*	*√*	*√*	*√*	*No*	*Deny*
	Subject 1	*-*	*-*	*-*	*-*	*√*	*√*	*No*	*Deny*
	Subject 1	*-*	*-*	*-*	*-*	*-*	*√*	*No*	*Deny*
2	*Subject 2*	*√*	*√*	*√*	*√*	*√*	*√*	*Yes*	*Permit*
	Subject 2	*-*	*√*	*√*	*√*	*√*	*√*	*No*	*Deny*
	Subject 2	*-*	*-*	*√*	*√*	*√*	*√*	*No*	*Deny*
	Subject 2	*-*	*-*	*-*	*√*	*√*	*√*	*No*	*Deny*
	Subject 2	*-*	*-*	*-*	*-*	*√*	*√*	*No*	*Deny*
	Subject 2	*-*	*-*	*-*	*-*	*-*	*√*	*No*	*Deny*
3	*Subject 3*	*√*	*√*	*√*	*√*	*√*	*√*	*Yes*	*Permit*
	Subject 3	*-*	*√*	*√*	*√*	*√*	*√*	*No*	*Deny*
	Subject 3	*-*	*-*	*√*	*√*	*√*	*√*	*No*	*Deny*
	Subject 3	*-*	*-*	*-*	*√*	*√*	*√*	*No*	*Deny*

	Subject 3	-	-	-	-	√	√	*No*	*Deny*
	Subject 3	-	-	-	-	-	√	*No*	*Deny*
4	*Subject 4*	√	√	√	√	√	√	*Yes*	*Permit*
	Subject 4	-	√	√	√	√	√	*No*	*Deny*
	Subject 4	-	-	√	√	√	√	*No*	*Deny*
	Subject 4	-	-	-	√	√	-	*No*	*Deny*
	Subject 4	-	-	-	-	√	√	*No*	*Deny*
	Subject 4	-	-	-	-	-	√	*No*	*Deny*

No	**Daftar**	**Atribut**						**Detail**	**Status**
		(1)	(2)	(3)	(4)	(5)	(6)		
1	*Resource 1*	√	√	√	√	√	√	*Yes*	*Permit*
	Resource 1	-	√	√	√	√	√	*No*	*Deny*
	Resource 1	-	-	√	√	√	√	*No*	*Deny*
	Resource 1	-	-	-	√	√	√	*No*	*Deny*
	Resource 1	-	-	-	-	√	√	*No*	*Deny*
	Resource 1	-	-	-	-	-	√	*No*	*Deny*
2	*Resource 2*	√	√	√	√	√	√	*Yes*	*Permit*
	Resource 2	-	√	√	√	√	√	*No*	*Deny*
	Resource 2	-	-	√	√	√	√	*No*	*Deny*
	Resource 2	-	-	-	√	√	√	*No*	*Deny*

	Resource 2	-	-	-	-	√	√	*No*	*Deny*
	Resource 2	-	-	-	-	-	√	*No*	*Deny*
3	*Resource 3*	√	√	√	√	√	√	*Yes*	*Permit*
	Resource 3	-	√	√	√	√	√	*No*	*Deny*
	Resource 3	-	-	√	√	√	√	*No*	*Deny*
	Resource 3	-	-	-	√	√	√	*No*	*Deny*
	Resource 3	-	-	-	-	√	√	*No*	*Deny*
	Resource 3	-	-	-	-	-	√	*No*	*Deny*
4	*Resource 4*	√	√	√	√	√	√	*Yes*	*Permit*
	Resource 4	-	√	√	√	√	√	*No*	*Deny*
	Resource 4	-	-	√	√	√	√	*No*	*Deny*
	Resource 4	-	-	-	√	√	-	*No*	*Deny*
	Resource 4	-	-	-	-	√	√	*No*	*Deny*
	Resource 4	-	-	-	-	-	√	*No*	*Deny*

No	**Daftar**	**Atribut**						**Detail**	**Status**
		(1)	(2)	(3)	(4)	(5)	(6)		
1	*Operation 1*	√	√	√	√	√	√	*Yes*	*Permit*
	Operation 1	-	√	√	√	√	√	*No*	*Deny*
	Operation 1	-	-	√	√	√	√	*No*	*Deny*
	Operation 1	-	-	-	√	√	√	*No*	*Deny*

	Operation 1	-	-	-	-	√	√	*No*	*Deny*
	Operation 1	-	-	-	-	-	√	*No*	*Deny*
2	*Operation 2*	√	√	√	√	√	√	*Yes*	*Permit*
	Operation 2	-	√	√	√	√	√	*No*	*Deny*
	Operation 2	-	-	√	√	√	√	*No*	*Deny*
	Operation 2	-	-	-	√	√	√	*No*	*Deny*
	Operation 2	-	-	-	-	√	√	*No*	*Deny*
	Operation 2	-	-	-	-	-	√	*No*	*Deny*
3	*Operation 3*	√	√	√	√	√	√	*Yes*	*Permit*
	Operation 3	-	√	√	√	√	√	*No*	*Deny*
	Operation 3	-	-	√	√	√	√	*No*	*Deny*
	Operation 3	-	-	-	√	√	√	*No*	*Deny*
	Operation 3	-	-	-	-	√	√	*No*	*Deny*
	Operation 3	-	-	-	-	-	√	*No*	*Deny*
4	*Operation 4*	√	√	√	√	√	√	*Yes*	*Permit*
	Operation 4	-	√	√	√	√	√	*No*	*Deny*
	Operation 4	-	-	√	√	√	√	*No*	*Deny*
	Operation 4	-	-	-	√	√	-	*No*	*Deny*
	Operation 4	-	-	-	-	√	√	*No*	*Deny*
	Operation 4	-	-	-	-	-	√	*No*	*Deny*

No	Daftar	Atribut						Detail	Status
		(1)	(2)	(3)	(4)	(5)	(6)		
1	*Environment 1*	√	√	√	√	√	√	*Yes*	*Permit*
	Environment 1	-	√	√	√	√	√	*No*	*Deny*
	Environment 1	-	-	√	√	√	√	*No*	*Deny*
	Environment 1	-	-	-	√	√	√	*No*	*Deny*
	Environment 1	-	-	-	-	√	√	*No*	*Deny*
	Environment 1	-	-	-	-	-	√	*No*	*Deny*
2	*Environment 2*	√	√	√	√	√	√	*Yes*	*Permit*
	Environment 2	-	√	√	√	√	√	*No*	*Deny*
	Environment 2	-	-	√	√	√	√	*No*	*Deny*
	Environment 2	-	-	-	√	√	√	*No*	*Deny*
	Environment 2	-	-	-	-	√	√	*No*	*Deny*
	Environment 2	-	-	-	-	-	√	*No*	*Deny*
3	*Environment 3*	√	√	√	√	√	√	*Yes*	*Permit*
	Environment 3	-	√	√	√	√	√	*No*	*Deny*
	Environment 3	-	-	√	√	√	√	*No*	*Deny*
	Environment 3	-	-	-	√	√	√	*No*	*Deny*
	Environment 3	-	-	-	-	√	√	*No*	*Deny*
	Environment 3	-	-	-	-	-	√	*No*	*Deny*
4	*Environment 4*	√	√	√	√	√	√	*Yes*	*Permit*

	Environment 4	-	√	√	√	√	√	*No*	*Deny*
	Environment 4	-	-	√	√	√	√	*No*	*Deny*
	Environment 4	-	-	-	√	√	-	*No*	*Deny*
	Environment 4	-	-	-	-	√	√	*No*	*Deny*
	Environment 4	-	-	-	-	-	√	*No*	*Deny*