Specification and Visualization of Policy Model in RBAC

Rafif Favian, I Made Wiryana dan Cahyawati Diah Kusumarini

Program Studi Magister Manajemen Sistem Informasi, Pascasarjana, Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
E-mail: rafiffavian584@gmail.com, mwiryana@staff.gunadarma.ac.id, cahyawati@staff.gunadarma.ac.id

Abstrak

Aplikasi Sistem Informasi Aparatur Sipil Negara adalah aplikasi yang mengelola seluruh tahapan manajemen kepagawaian aparatur sipil negara, sejalan dengan perkembangan teknologi dan kebutuhan aparatur sipil negara akan data digital, maka Sistem Informasi Aparatur Sipil negara perlu dilakukan pengembangan. terdapat 18 pengembangan aplikasi salah satunya adalah pada fungsi manajemen user. Adapun permasalahan yang ada pada fungsi tersebut yaitu user memiliki role yang cukup banyak, dimana 1 user bisa memiliki akses untuk beberapa aplikasi. maka dari itu dibutuhkan sebuah manajemen user yang jelas untuk mengatur hal tersebut, yaitu menggunakan akses kontrol dengan pendekatan *RBAC (Role Based Access Control).* Tujuan dari penelitian ini adalah Menghasilkan model role pada fitur manajemen user menggunakan *XACML* dan Menghasilkan desain *policy RBAC* pada akses kontrol manajemen user di aplikasi Sistem Informasi Aparatur Sipil negara. Metode penelitian terdiri dari : (1) *Literature Review*, (2) *Role Modelling Using XACML,* (3) *Result Visualization*, dan (4) *Merge Operation Process*. Penelitian ini telah menghasilkan 6 model role, dimana keenam model tersebut telah di visualisasikan dalam bentuk graph berupa gambar, visualisasi dilakukan guna mendeteksi apabila terjadi kesalahan dalam pendefinisian role pada layanan Sistem Informasi Aparatur Sipil negara. dan yang kedua, penelitian ini telah menghasilkan *policy RBAC* yang ditulis menggunakan skema *XACML* sebagai spesifikasinya.

Kata kunci : Akses Kontrol, *RBAC, XACML*, Sistem Informasi Aparatur Sipil negara, Manajemen User.

Abstract

Aplikasi Sistem Informasi Aparatur Sipil Negara is an application that manages all stages of the civil service management of the state civil apparatus, in line with technological developments and the needs of the State Civil Apparatus for digital data, so the Sistem Informasi Aparatur Sipil Negara to be developed. There are 18 application developments, one of which is the user management function. The problem with this function is that the user has quite a lot of roles, where 1 user can have access to several applications. Therefore, a clear user management is needed to manage this, namely using access control with an RBAC (Role Based Access Control) approach. The purpose of this study is to produce a role model on user management features using XACML and produce an RBAC policy design on user management access control in the state Sistem Informasi Aparatur Sipil Negara. The research method consists of: (1) Literature Review, (2) Role Modeling Using XACML, (3) Result Visualization, and (4) Merge Operation Process. This research has produced 6 role models, where the six models have been visualized in the form of graphs in the form of images, visualizations are carried out to detect if there is an error in defining roles in the State Civil Apparatus Information System service. and secondly, this research has produced an RBAC policy written using the XACML scheme as its specification.

Keywords : Acsses Control, *RBAC, XACML*, Sistem Informasi Aparatur Sipil Negara, Manajemen User

DOI : http://dx.doi.org/10.32409/jikstik.21.2.3057

Introduction

In order to realize a transparent, effective, efficient, and integrated state civil service personnel management process, an application system is needed that is able to meet all the needs of the personnel management business process and integrate the data with other related systems. This is in line with the main tasks and functions of the State Civil Service Agency as the organizer of the National Civil Service Management as stated in the Law on State Civil Apparatus concerning State Civil Apparatus in the fourth part of article 47 which states that one of the functions of the State Civil Service Agency is "guidance and administration of State Civil Apparatus Management, State Personnel Agency is an Indonesian non-ministerial government agency tasked with carrying out government duties in the field of state civil service management. The National Civil Service Agency as the National Civil Service Supervisor as mandated by Law Number 5 of 2014 concerning State Civil Apparatus, in this case, the need for accurate, integrated, and up-to-date Civil Servant data is very important. Civil Service Data is managed by the State Personnel Agency as a national database where the data becomes a reference in the staffing process by all agencies. In line with technological developments and the needs of the State Civil Apparatus for digital data including electronic signatures in accordance with KOMINFO Ministerial Regulation number 11 of 2018 concerning the application of digital signatures on official documents, this strategic issue has also become one of the innovations in the development of the State Civil Apparatus Information System. latest. There are 18 developments of integrated State Civil Apparatus Information System applications, one of which is Admin Services and Support Systems.

In this support service, there is an admin service where there are functions for user management, file management wherein file management is arranged related to document types, workflow repositories, document procedures, validation types, requirements validation, and document templates that will all be used during verification, validation. proposals to print service documents for promotions, pensions, and other services. One of the important functions is user management. The problem behind the user management function is that users in the State Civil Apparatus Information System have quite a lot of roles, where 1 user can have access to several applications. therefore we need clear user management to manage this. There are various ways to do user management, one of which is to use ACL (Access Control List), where ACL can manage access for each user in each application [1]. ACL itself has several approaches, one of which is RBAC (Role-Based Access Control) [2]. It is hoped that this approach can be an appropriate and relevant solution to overcome the problems that exist in the user management feature in the State Civil Apparatus Information System application.

Supporting Theory

Access Control

Access control is one aspect that relates to how to manage access to information. One of the commonly used ways to secure information is to regulate access to information through “authentication” and “access control” mechanisms. Implementation of this mechanism, among others, by using a "password". Access control is usually done by grouping users in "groups". [3]

Access control in principle is a mechanism to limit operations or actions on a computer system to only legitimate users. Furthermore, there are 4 main issues in access control, namely identification, authentication, authorization and access decisions. The brief explanation is as follows [4] :

1. Identification recognizes the party who will be responsible for the access request, it can be in the form of a person or an NPE (non-person entity) such as a computer, or an application.

2. Authentication is an attempt to confirm the truth of a piece of data or an entity. User authentication itself means confirming previously stored user data.

3. Authorization is a process to determine what services are allowed to be used by authenticated users.

4. Access Decision: based on a combination of the three aspects above, a decision is then given whether the request is allowed or rejected by the system.

ACL (Access Control List)

ACL is to establish the access authority list based on files, and the list registers the access user names and the access authority administrative relationships of the object files. By using ACL, it is easy to judge the authorized access of special objects, and which subjects can access the system and which access authorities they have. ACL records the subjects and their operation authorities which can access the object by the form of linked list (seen in Figure 1).

In Figure 1, the subjects who can access the subject F2 respectively are U2, PI, and P2, and the authorities of U2 include reading, writing, adding, and deleting, and the authority of PI is reading, and the authorities of P2 are reading and writing. Because ACL is simple, convenient, and practical, and many commonly used operation systems all adopt ACL to provide the service of access control strategy, such as UNIX and VMS. [5]

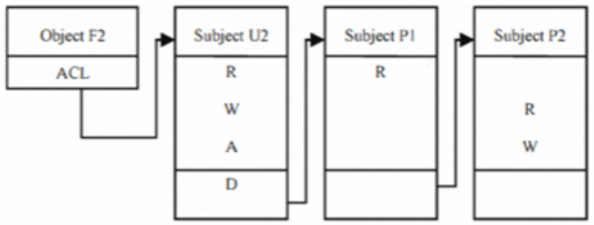

Figure 1: Access Control List

Access Control Methods

Access is the ability to work on a particular computer resource (e.g., use, change, or view). Access control is the means by which this ability is explicitly enabled or disabled through physical or system-based controls. Computer-based access control can prescribe not only which entity has the permission to access a specific system resource, but also the type of access. DAC permits the granting and revoking of access control privileges to be left to the discretion of the individual users. A DAC mechanism allows users to grant or revoke access to any of the objects under their control. MAC, as defined in the Trusted Computer Security Evaluation Criteria (TCSEC), is "a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity". In role-based access control, access decisions are based on the roles that individual users have as part of an organization. [6]

Access Control For XML Data

XML is the standard today for modeling and transmitting data on the Web and one of the key technologies of the Semantic Web. The most important characteristic of XML,that distinguishes it from other markup languages, such as HTML, is the notion of semantic tags, allowing one to structure a document into different portions, called elements, with an associated semantics. An element may also contain attributes, whose purpose is to provide additional information on the element. XML documents may have a nested or hierarchical structure, since elements can be organized into sub-elements; they may be inter-linked, through IDREFs/URI attributes; they may have an associated DTD/XMLSchema, specifying their structures. [7]

eXtensible Access Control Markup Language (XACML)

The eXtensible Access Control Markup Language (XACML) 70 is also an OASIS standard that is intended to specify policies and procedures for access control and authorisation purposes. The standard provides a model for establishing access control decisions, an XML syntax for specifying access control policies as well as algorithms and functions that can be used in policy evaluation and constituting access control decisions. [8]

The main components in the policy-based access control model are a Policy Enforcement Point (PEP) and a Policy Decision Point (PDP) that may be distributed over several nodes of a network. The PEP enforces access control based on policy decisions from the PDP. [9]

Role-Based Access Control

In role based access control access decisions are based on the individual's roles and responsibilities within the cloud environment. It formulates the user's access to the system based on the activities that the user has been executed in the cloud. It requires the identification of roles of users on the system. Role can be set of objects or actions associated with the subject. Role may vary depends on the user's priority. RBAC provides the web based application security. Roles are assigned based on the particular cloud organizational structure with their security policies. Each role in the organization's profile includes all authorized users, commands, transaction and allowable information access. Roles can be assigned based on the least privilege. These identified roles can be transferred and used based on the appropriate procedures and security policies. Roles can be managed centrally. [10]

Design and Analysis

Role Modelling Using XACML

Figure 2 is an XACML flow model that illustrates how the work process of XACML using the RBAC approach, starting from a request for a request from the subject, then the request is processed using XACML Access Control Policies to check whether the request is by the established policies, if it is appropriate then the request will be permitted to the destination resource.

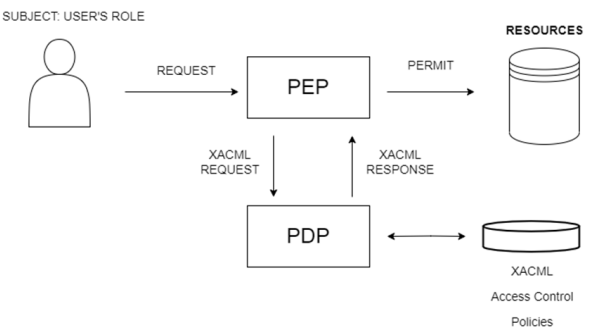

Figure 2: XACML Flow Model

The following is a detailed process of the XACML FLOW MODEL in Figure 2:

- The client sends a request to access the destination resource.
- The request is then filtered by PEP (policy enforcement point), where PEP will convert the request into an XACML request, and forward it to PDP (policy decision point).
- PDP will evaluate XACML requests against policies that have been defined in the XACML access control policies database.
- after the evaluation of the request has been completed, the PDP will send an XACML response to the PEP stating whether access is permitted or not.
- If the response is a permit, then the PEP filter is bypassed, and the client can access the destination resource.

Figure 3 is an implementation of XACML access control policies used by PDP as a policy to evaluate requests from clients. XACML code in Figure 3 is one of the sample XACML policies used in the SI ASN Service. XACML has several elements such as targets, rules, and conditions. there are four targets in XACML in Figure 3, namely:

- url: /layananPeremajaan/monitoringUsulan, dengan label Inbox Usul.
- url: /layananPeremajaan/monitoringPenetapan, dengan label Inbox Penetapan.
- url: /layananPeremajaan/penetapanSK, dengan label Buat Surat Keputusan.
- url: /layananPeremajaan/monitoringSK, dengan label Inbox Surat Keputusan.

where the four targets consist of several URLs and several labels, which can be accessed as long as the attribute role is role:siasn-instansi:peremajaan:operator, if the role is not role:siasn-instansi:peremajaan: operator, then the target cannot be accessed on the Information System service for state civil servants.

Figure 3: XACML Document

Result Visualization

Figure 4 is a visualization stage where the RBAC design modeled using XACML in Figure 3 will be visualized into a graph using Graphviz. The visualization is carried out to obtain information related to the authorization possessed by each role in the Information System service for the state civil apparatus. To visualize the RBAC design into a Graphviz form, XSLT (XSL Transformations) is needed to specify the rules between XML documents that are transformed into other forms, in this case, into dotML (Dot Markup Language) form, which is a language to describe a graph.

The following is a detailed process of XSLT PROCESS in Figure 4:

- Prepare an XML file as a source that contains in this case data from roles that exist in the Information System service for state civil ser-

vants.

- Setting up an XSL Style Sheet with the required transformation template in this case is a dotML form, the language used to create a graph.
- Next, the XSLT processor will take the XML source as well as the XSLT stylesheets, and process them to produce an output document that is dotML.

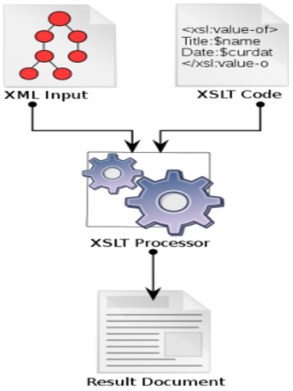

Figure 4: XSLT Mapping Proccess

Merge Operation Process

Figure 5 is a mapping of the merging operation process. The merging operation process is carried out on a role with the condition that there are 2 roles that have the same or identical service in the Information System service for the state civil apparatus. The merging operation process is illustrated in Figure 5.

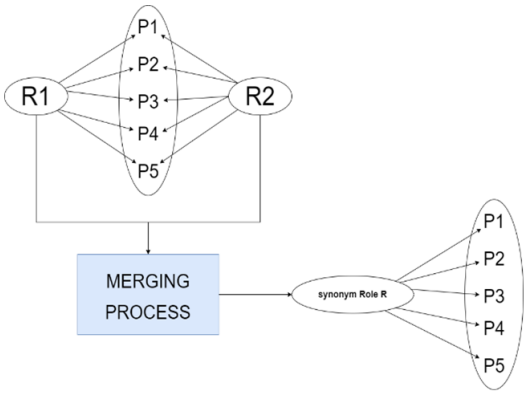

Figure 5: Mapping Of Merging Proccess

The following is a detailed process from Figure 5:

- There are 2 Roles defined with variables R1 and R2, where each of these roles has a permission defined by alphabetical letters, namely:
  - P1: permission 1
  - P2: permission 2
  - P3: permission 3
  - P4: permission 4
  - P5: permission 5
- Both roles have identical or the same permissions, so conditions such as duplication of roles in a particular service must be carried out, therefore a merging operation must be carried out as shown in the figure 5, where 2 roles that have identical services These services are merged to create a new role with the service, so this method can prevent duplication which makes a service inefficient.

Discussion

Figure 6 is the process of the Graphviz transformation where the Dot File document obtained from stage RESULT VISUALIZATION whose details can be seen from the figure 4, is then transformed into the form PNG file with the help of GRAPHVIZ so that the graph visualization on the role of the State Civil Apparatus Information System that is designed can be seen clearly.

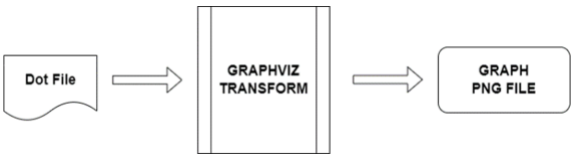

Figure 6: Component Graphviz Diagram

Figure 7 is the implementation flow of the process in the figure 7, where in the dot file there are several nodes and edges are marked with arrows, the arrow here is a sign of a "edge" that connects the role node and permission node, which is then processed by graphviz , and transformed into a graph with PNG extension. so that the dot file containing the code nodes, edges, and arrows becomes a visualization of the original graph in the form of an image. From the results of the graph visualization, several sample roles were formed.

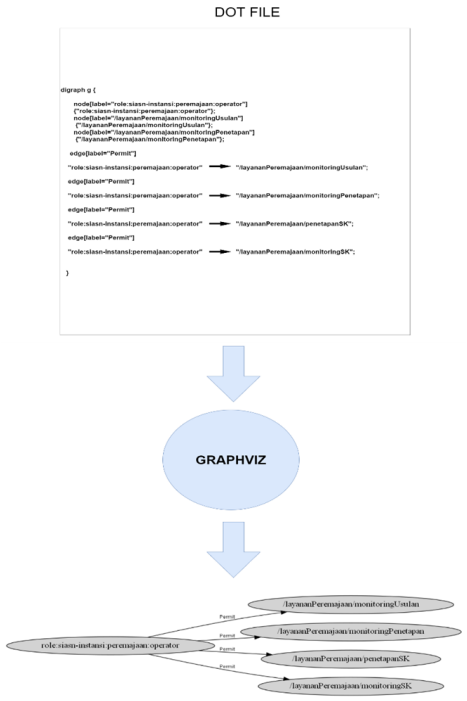

Figure 7: Graphviz Detail Workflow

Roles do not have permissions

From the results of the analysis on roles in the Information System service for state civil servants, it was found a role that did not have any permissions, namely a role named role:siasn-bkn:perencanaan:PP-TTE-Renpegfor. The figure 8 is a dot file transformed from the XSLT process, where the XSLT processor will take the XML source and XSLT stylesheets, and process them to produce an output document, namely dotML. The detailed image of the XSLT process can be seen in Figure 4.

Figure 9 is the output in the form of an image with PNG extension from the dot file in the figure 6. we can look at the dot file figure 8. there is only 1 node, namely role:siasn-bkn:perencanaan:PP-Renpegfor which is one of the role names in the Information System of the State Civil Apparatus service, and there is no arrow which is a sign of an edge on the graph, so it can be concluded that the role does not have permissions , because the node has no edges but only stands alone. then next, the dot file is processed by graphviz , and transformed into a graph with a PNG extension. so that the dot file containing the code becomes a visualization of the original graph in the form of an image like the Figure 9.

```
digraph g{
    compound="true";bgcolor="";fontcolor="";fontname="";fontsize="";label="";margin="";nodesep="";rankdir="LR";ranksep="";ratio="";size="";
    node[label="role:siasn-bkn:perencanaan:PP-Renpegfor",
    color="",fillcolor="",fixedsize="",fontcolor="",fontname="",fontsize="16",height="",shape="",style="filled",URL="",width="",]
    {"role:siasn-bkn:perencanaan:PP-Renpegfor"};
}
```

Figure 8: PP-Renpegfor-Dot

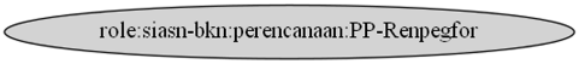

Figure 9: PP-Renpegfor-graph

From the results of the analysis, it was found that 1 role was independent or did not have any permissions at all, this happened because the role had not been assigned by the owner of the system. because basically a role is a reflection or mapping of roles in the real world, so that when the role has been defined by the system owner, it turns out that the role does not yet have its implementation in the Information System service for state civil servants.

Role has only one permission

From the results of the analysis on roles in the Information System service for state civil servants, it was found that roles that only have 1 permission, namely a role named role:siasn-bkn:kp:landing with permissions to access the servcice http://landing-siasn.bkn.go.id. The figure 10 is a dot file transformed from the XSLT process, where the XSLT processor will take the XML source and XSLT stylesheets, and process them to produce an output document, namely dotML. The detailed image of the XSLT process can be seen in the Figure 4.

```
digraph g {
compound="true";bgcolor="";fontcolor="";fontname="";fontsize="";label="";margin="";nodesep="";rankdir="LR";ranksep="";ratio="";size="";
node[label="role:siasn-bkn:kp:landing", color="",fillcolor="",fixedsize="",fontcolor="",fontname="",fontsize="16",height="",shape="",style="filled",URL="",width="",]
{"role:siasn-bkn:kp:landing"};
node[label="http://landing-siasn.bkn.go.id",
color="",fillcolor="",fixedsize="",fontcolor="",fontname="",fontsize="16",height="",shape="",style="filled",URL="",width="",]
{"http://landing-siasn.bkn.go.id"};
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Perm
it",labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhe
ad="",ltail=""]
"role:siasn-bkn:kp:landing" -> "http://landing-siasn.bkn.go.id";
}
```

Figure 10: KP-Landing-Dot

Figure 11 is the output in the form of an image with PNG extension from the dot file in the figure 10. can be seen in the dot file figure 10. there is 1 node, namely role:siasn-bkn:kp:landing which is one of the names of roles in the Information System service for state civil servants, and edges marked with arrows, the arrow here is a sign of an "edge" that connects the node roles and node permissions, it can be seen that in the figure 10 there is only 1

edge which is the permission of role role:siasn-bkn:kp:landing which means role:siasn-bkn:kp:landing can access the servcice http://landing-siasn.bkn.go.id. then next, the dot file is processed by graphviz , and transformed into a graph with a PNG extension. so that the dot file containing the code becomes a visualization of the original graph in the form of an image like the figure 11.

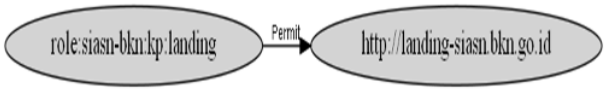

Figure 11: KP-Landing-Graph

Role has more than one permission

From the results of the analysis on roles in the Information System service for state civil servants, it was found that roles that have several permissions, one of which is a role named

role:siasn-instansi:peremajaan:operator that can access several services, include:

- /layananPeremajaan/monitoringUsulan
- /layananPeremajaan/monitoringPenetapan
- /layananPeremajaan/penetapanSK
- /layananPeremajaan/monitoringSK

The Figure 12 is a dot file transformed from the XSLT process, where the XSLT processor will take the XML source and XSLT stylesheets, and process them to produce an output document, namely dotML. The detailed image of the XSLT process can be seen in the Figure 4.

```
digraph g {
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Perm
it",labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhe
ad="",ltail=""]
"role:siasn-instansi:peremajaan:operator" -> "/layananPeremajaan/monitoringPenetapan";
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Perm
it",labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhe
ad="",ltail=""]
"role:siasn-instansi:peremajaan:operator" -> "/layananPeremajaan/penetapanSK";
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Perm
it",labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhe
ad="",ltail=""]
"role:siasn-instansi:peremajaan:operator" -> "/layananPeremajaan/monitoringSK";
}
```

Figure 12: Peremajaan-Operator-Dot

Figure 13 is the output in the form of a PNG image from the dot file in the Figure 12. can be seen in the dot file Figure 12. there is 1 node, namely role:siasn-instansi:peremajaan:operator which is one of the names of roles in the Information System ser-

vice for state civil servants, and edges marked with arrows, the arrow here is a sign of an "edge" that connects the node roles and permission nodes, it can be seen that in the Figure 12 there are 4 edges which are permissions from role:siasn-instansi:peremajaan:operator which means role:siasn-instansi:peremajaan:operator can access several servcices, namely :

- /layananPeremajaan/monitoringUsulan
- /layananPeremajaan/monitoringPenetapan
- /layananPeremajaan/penetapanSK
- /layananPeremajaan/monitoringSK

then , the dot file is processed by graphviz , and transformed into a graph with a PNG extension. so that the dot file containing the code becomes a visualization of the original graph in the form of an image like the figure 13.

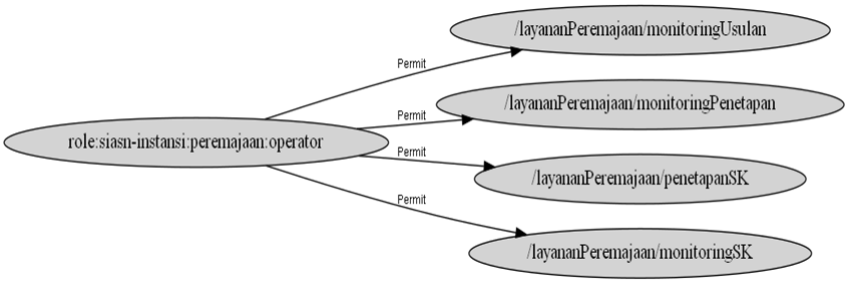

Figure 13: Peremajaan-Operator-Graph

Two Roles have the same 2 permissions

From the results of the analysis on roles in the Information System service for state civil servants, it was found a condition where 2 roles have the same 2 permissions, namely a role named role:siasn-bkn:kp:PP-BKN and role:siasn-bkn:kp:PP-TTE-BKN where the two roles have several permissions including:

- role:siasn-bkn:kp:PP-BKN can access several services, namely:
  - /parafPertek
  - /monitoringParafPertek
  - /parafSK
  - /monitoringParafSK
  - /monitoringProgress
  - /viewProfile
- role:siasn-bkn:kp:PP-TTE-BKN can access several services, namely:
  - /monitoringProgress
  - /viewProfile
  - /monitoringTtdSK
  - /ttdPertek
  - /monitoringTtdPertek
  - /ttdSK

The figure 14 is a dot file transformed from the XSLT process, where the XSLT processor will take the XML source as well as the XSLT stylesheets, and process it to produce an output document, namely dotML. The detailed image of the XSLT process can be seen in the figure 4.

```
digraph g {
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Permit",
labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhead="",lt
ail=""]
"role:siasn-bkn:kp:PP-TTE-BKN" -> "/monitoringTtdPertek";
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Permit",
labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhead="",lt
ail=""]
"role:siasn-bkn:kp:PP-TTE-BKN" -> "/ttdSK";
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Permit",
labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhead="",lt
ail=""]
"role:siasn-bkn:kp:PP-TTE-BKN" -> "/monitoringTtdSK";
}
```

Figure 14: Kanreg-Kemendikbud-Dot

The figure 15 is the output in the form of an image with the extension PNG from the dot file in the figure 14. can be seen in the dot file figure 14 there are 2 nodes namely role:siasn-bkn:kp:PP-BKN and role:siasn-bkn:kp:PP-TTE-BKN which is the name of the role in the service Information system for state civil servants, then there are 10 edges marked with arrows, the arrow here is a sign of an "edge" that connects the role node and the permission node, it can be seen that in Figure 15 there are Cross Relation between role:siasn-bkn:kp:PP-BKN and role:siasn-bkn:kp:PP-TTE-BKN, where there are 2 services owned by the two roles. namely service /monitoringProgress and /viewProfile, thus forming a Cross Relation between the two roles. then next, the dot file is processed by graphviz , and transformed into a graph with a PNG extension. so that the dot file containing the code becomes a visualization of the original graph in the form of an image like the figure 15.

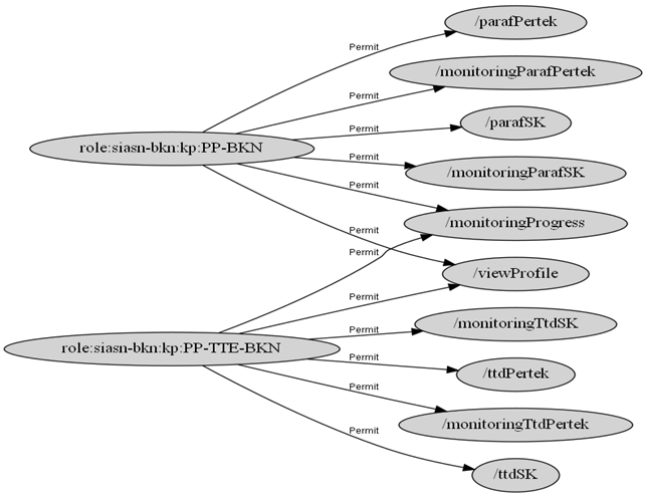

Figure 15: Kanreg-Kemendikbud-Graph

From the results of the analysis, it was found that 2 permissions are the same between the two roles so as to form a cross relation, if we don't visualize the definition of roles first like the figure 15, then we cannot detect if there is an error in defining the role, for example 2 roles should only have the same 2 permissions, because they were not identified beforehand through a visualization like the figure 15, as a result 2 roles could have more than 2 the same permissions so that an error occurred in the definition of roles.

Two roles have different permissions

From the results of the analysis on roles in the Information System service for state civil servants, it was found a condition where 2 roles had different permissions, namely a role named role:siasn-instansi:pemberhentian:paraf and role:siasn-instansi:pemberhentian:ap proval where both roles were has several permissions including:

- role:sias-instansi:pemberhentian:paraf can access several services, namely:
  - /layananPemberhentian/parafSK
  - /layananPemberhentian/inboxParafSK
- role:siasn-instansi:pemberhentian:approval can access several services, namely:
  - /layananPemberhentian/TTDSK
  - /layananPemberhentian/Monitoring TTDSK

The figure 16 is a dot file transformed from the XSLT process, where the XSLT processor will take the XML source and XSLT stylesheets, and process them to produce an output document, namely dotML. The detailed image of the XSLT process can be seen in the figure 4.

```
digraph g {
node[label="/layananPemberhentian/inboxParafSK",
color="",fillcolor="",fixedsize="",fontcolor="",fontname="",fontsize="16",height="",shape="",style="filled",URL="",width="",]
("/layananPemberhentian/inboxParafSK");
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Perm
it",labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhe
ad="",ltail=""]
"role:siasn-instansi:pemberhentian:paraf" -> "/layananPemberhentian/parafSK";
edge[arrowhead="",arrowsize="",arrowtail="",constraint="",color="",decorate="",dir="",fontcolor="",fontname="Arial",fontsize="9",headlabel="",headport="",label="Perm
it",labeldistance="",labelfloat="",labelfontcolor="",labelfontname="",labelfontsize="",minlen="",samehead="",sametail="",style="",taillabel="",tailport="",URL="",lhe
ad="",ltail=""]
"role:siasn-instansi:pemberhentian:paraf" -> "/layananPemberhentian/inboxParafSK";
}
```

Figure 16: Paraf-Approval-Dot

Figure 17 is the output in the form of an image with PNG extension from the dot file in the figure 16. can be seen in the figure 16. there are 2 nodes, namely role:siasn-instansi:pemberhent ian:paraf and role:siasn-instansi:pemberhen tian:approvalwhich is the name of the role in the Information System service for state civil servants, and the edge marked with an arrow, the arrow here is a sign of a The "edge" that connects the node role and the permission node, it can be seen that in the figure 16 there are 4 edges which are the permissions of role:siasn-instansi:pemberhent ian:paraf and role:siasn-instansi:pemberhent ian:approval, where each role has 2 permissions. then next, the dot file is processed by graphviz , and transformed into a graph with a PNG extension. so that the dot file containing the code becomes a visualization of the original graph in the form of an image like the figure 17.

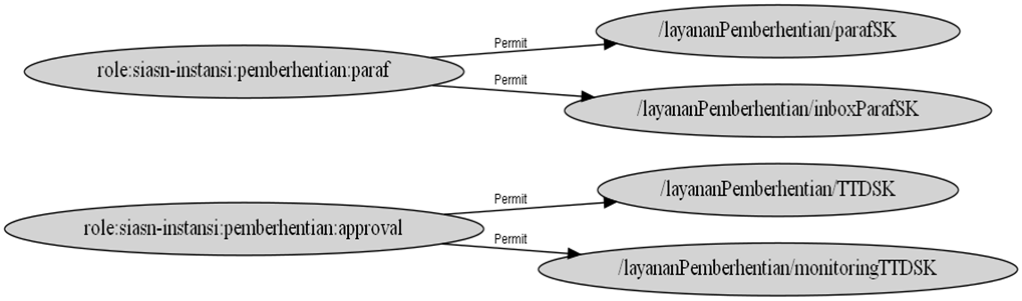

Figure 17: Paraf-Approval-Graph

From the results of the analysis, it was found that 2 roles have different permissions, if we don't visualize the definition of the role first like the figure 17, then we cannot detect if there is an error in defining the role, for example, 2 roles should only be have different permissions, because they were not identified first through visualization such as the figure 17, as a result 2 roles can have the same permissions which makes the two roles have a relationship, so that an error occurs in the definition of roles.

Two Roles have identical permissions

From the results of the analysis on roles in the Information System service for state civil servants, it was found that 2 roles have identical or the same permissions, so that conditions like can create duplication of roles in a certain service, therefore a merging operation process must be carried out as shown in the figure 5 where 2 roles that have identical services are merged so as to give birth to a new role with services so that this method can prevent duplication which makes a service inefficient.

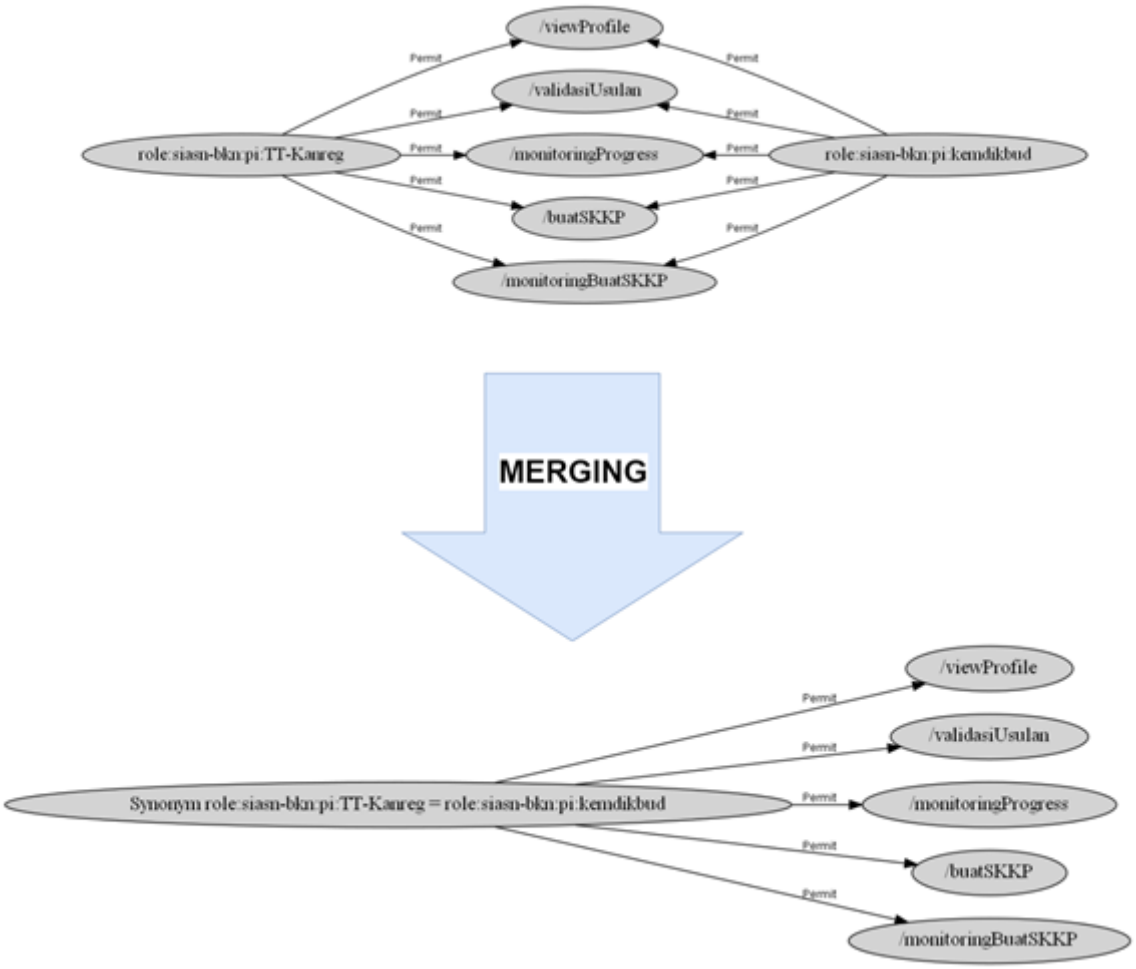

Figure 18: Merging Implementation

- Found 2 roles that have identical or the same services in the Information System service for state civil servants, these roles are:
  - role:siasn-bkn:pi:TT-Kanreg , with 5 permiss ions, namely:
    - /viewProfile
    - /validasiUsulan
    - /monitoringProgress
    - /buatSKKP
    - /monitoringBuatSKKP
  - role:siasn-bkn:pi:TT-Kanreg , with 5 permiss ions, namely:
    - /viewProfile
    - /validasiUsulan
    - /monitoringProgress
    - /buatSKKP
    - /monitoringBuatSKKP
- From these 2 roles, we can see that they both have identical permissions, thus creating a duplication of roles in the SI ASN service, therefore a merging operation must be carried out as shown in the figure 5, where 2 roles have services identical in the merge.
- after the merging process is carried out, it forms a new role, namely "Synonym role:siasn-bkn:pi:TT-Kanreg = role:siasn-bkn:pi:kemdik bud", so that this method can prevent duplication which makes SI ASN services become not efficient.

Coclussion

Based on the analysis carried out on the information system service for state civil servants, 6 role models have been produced, including the Role that does not have any permissions, the Role that only has 1 permission, ,Role has more than 1 permissions, 2 Roles have the same permissions, 2 Roles have different permissions, and lastly 2 Roles have the same permissions. identical. and the six models have been visualized in the form of a graph in the form of images, visualization is carried out to detect if there is an error in defining the role in the information system services for state civil servants. Then an RBAC (Role-Based Access Control) policy has been generated which is written using the XACML (eXtensible Access Control Markup Language) scheme as a specification, where the XACML contains a policy statement in the form of a rule created based

on the role of each user to be able to check the requests given by the user, and this policy design has been applied to the information system service for state civil servants.

As for suggestions for further researchers, namely the first, this XACML still requires testing and validation of the XACML structure design that has been made, and secondly, it has not been made. XACML request as input to test the XACML policy that has been made.

Daftar Pustaka

[1] Ausanka Crues & Ryan, "Methods for access control: advances and limitations", Harvey Mudd College, 2001. [Online]. Available: https://www.cs.hmc.edu/~mike/public_html/courses/security/s06/projects/ryan.pdf

[2] Punithasurya & Priya Jeba, "Analysis of different access control mechanism in cloud", International Journal of Applied Information Systems, Vol.4, No.2, doi: 10.5120/ijais12-450660, 2012.

[3] Erwin Gunadhi & Heru Arranuri, "Pengamanan Basis Data Pengelolaan Hak Akses Dengan Metode Role-Based Access Control", Jurnal Algoritma, vol. 12, no. 1, pp. 166-170, 2015.

[4] Alan H. Karp, H. Haury and M.H. Davis, "From ABAC to ZBAC: The evolution of access control models", ISSA (Information Systems Security Association), Hewlett-Packard Development Company, L.P, 2010.

[5] Bai & Zheng, "Study on the access control model", In Proceedings of 2011 Cross Strait QuadRegional Radio Science and Wireless Technology Conference, 2011..

[6] S. Pinagapani, D. Xu, J. Kong, "A Comparative Study of Access Control Languages", Third IEEE International Conference on Secure Software Integration and Reliability Improvement (SSIRI 2009), Shanghai, Chin, pages 407–412, 2009.

[7] Ferrari & Elena, "Access Control in Data Management Systems: A Visual Querying Perspective", Synthesis Lectures on Data Management, Springer Link, ISBN: 978-3-031-01836-7, 2010..

[8] Fischer Hellmann & Klaus Peter, "Information Flow Based Security Control Beyond RBAC: How to enable fine-grained security policy enforcement in business processes beyond limitations of role-based access control (RBAC)", Springer Science & Business Media, volume 1, 2012.

[9] Overxeer Simon Godik, "EXtensible Access Control Markup Language (XACML) version 1", OASIS Standard, 2005.

[10] David Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli, "Role-based access control", Artech house, ISBN: 1-58053-370-1, 2003.