# Implementation of Web Service for Integrated Data Students on Information System Academic and Library

Aris Susanto, Rahmat Inggi
Information System
STMIK Bina Bangsa Kendari
Jln. Jend. A.H. Nasution Kendari, Indonesia

**Abstract:- The research aims to integrate data between the academic management information system and the library information system in the Kendari STMIK Bina Bangsa environment. Integrated data is student data sourced from an academic management information system. For the data integration process, RESTFull Web Service is used, the programming language used is PHP Codeigniter and the database used is MySQL. Data access is done through a web service by sending data in JSON format, namely student data in the form of NIM, student name, department, class and gender. The results of this study reveal that student data is the type of data most needed by other information systems in the STMIK Bina Bangsa Kendari environment.**

***Keywords:-*** *Web Service, Integrated Data, Information System.*

## I. INTRODUCTION

STMIK Bina Bangsa Kendari is an educational institution that continues improve the application of information and communication technology. There are several units in the STMIK Bina Bangsa Kendari environment. To improve the services of each unit, each unit has an information system to support administrative activities, data processing, and information services. In its development, until now STMIK Bina Bangsa Kendari has developed several information system applications, including Academic Management Information System (SIMAK), Digital Data Base (PDD), Library Information System, Student Payment System.

The above application runs using a separate database. This of course makes the application run ineffective and the possibility of data asynchronization. This condition gives a big influence in decreasing the performance of each unit, because there are several databases that are used separately and different applications. Each application has its own basic data (master) such as student data. As a result of the application not being integrated, the same data input will be performed in each application. This causes prone to wrong data input. For example the names for the same students differ between information systems.

One of the important problems that needs to be overcome related to the development of information systems applications at STMIK Bina Bangsa is how to synchronize data between applications, so there is a need for special studies on how to synchronize data between applications as a basis for developing solutions.

One solution to overcome the problem is to do data integration. In this study, data integration was carried out between the Library Information System and the Academic Information System. There are many ways to do data integration. In this research, REST API is used for data integration process. The selected data is student data in the Library Information System application which is integrated with student data from the Academic Management Information System.

## II. LITERATURE REVIEW

### *A. Web Service*

Published in the journal Erick Kurniawan (2014), in the opinion of Ethan Cerami [1], Web Service is a service available on the Internet. Web Service uses the standard XML format for sending messages. Web Services are also not tied to specific programming languages or operating systems.

Web Services are interfaces that describe collections that can be accessed on the network using the standard XML format for message exchange. Web Services do specific tasks. Web Services are described using a standard XML notation format called services description [2].

Web service technically has a mechanism of interaction between systems to support interoperability, both in the form of aggregation (collection) and syndication (unification). Web services have open services for the benefit of data integration and information collaboration that can be accessed via the internet by various parties using technology owned by each user. Although similar to web-based Application Programming Interface (API), web service is superior because it can be called remotely via the internet. Web service invocation can use any programming language and in any platform, while the API can only be used on certain platforms [3].

Web service can be understood as Remote Procedure Call (RPC) which is able to process functions defined in a web application and expose an API or User Interface (UI) through the web. The advantages of web services are: 1) cross platform, 2) language independent, 3) bridge connecting with the database without the need for a database driver and not having to know the type of DBMS, 4) simplifying the process of exchanging data, and 5) reusing application components [3].

### *B. REST (Representational State Transfer)*

The Reresentational State Transfer (REST) method was developed by Fielding [4]. REST is a software architecture style that contains guidelines and best practices for creating scalable web services. REST is a coordinated set of restrictions that is applied to the design of components in a distributed hypermedia system that can make architecture easier to maintain. REST efficiently uses HTTP [5]. REST is based on four technological principles, namely:

- ➢ Resource identifier through uniform resource identifier (URI).
- ➢ Uniform interface (CRUD resources use PUT, GET, POST, and DELETE operations)
- ➢ Self-descriptive messages (unbound resources so that they can access HTML, XML, PDF, JPEG, plain text, meta data, etc.) content.
- ➢ Stateful interactions through hyperlinks (stateless) [6].

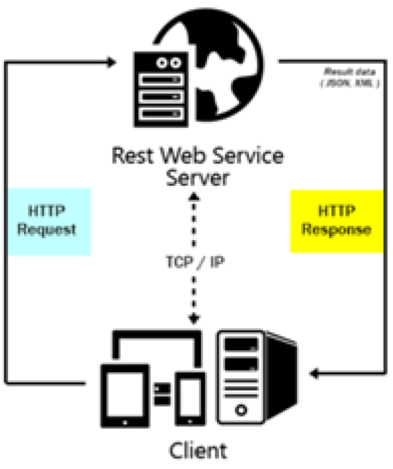
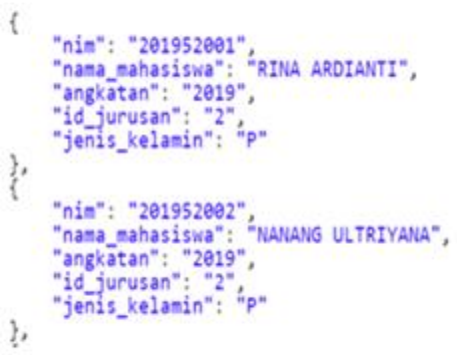


Fig 1:- REST Web Service Interaction

Picture above explains how Rest Web Service makes a request to the server then the server replies with the result in the form of json. The following is the result of json format from the server.



```
{
    "nim": "201952001",
    "nama_mahasiswa": "RINA ARDIANTI",
    "angkatan": "2019",
    "id_jurusan": "2",
    "jenis_kelamin": "P"
},
{
    "nim": "201952002",
    "nama_mahasiswa": "NANANG ULTRIYANA",
    "angkatan": "2019",
    "id_jurusan": "2",
    "jenis_kelamin": "P"
},
```

Fig 2:- Example of JSON Format

### *C. API (Application Programming Interface)*

Application Programming Interface (API) is a technology that facilitates the exchange of information or data between two or more software applications. API is a virtual interface between two software functions that work together, such as between a word processor and a spreadsheet [7]. An API defines how programmers utilize certain features of a computer.

### *D. JSON (Java Script Object Notation)*

JSON (Java Script Object Notation) is a lightweight data exchange method composed by Douglas Crockford. JSON is designed aiming to simplify the process of exchanging web-based data and also at the same time is the result of expansion of JavaScript functions.

Loaded in the journal Aidina Ristyawan, Dwi Harini (2018), JSON is a simple data type format, does not require storage space, and does not require large resources in transferring good data only using slow network connections, does not require processor resources that are is great on Android-based smart mobile phones to decode JSON format. This is because JSON has a fairly small and simple capacity [8]. The structure of the JSON format code is in the object and array format, as an illustration of the format code structure shown in Figure 3.

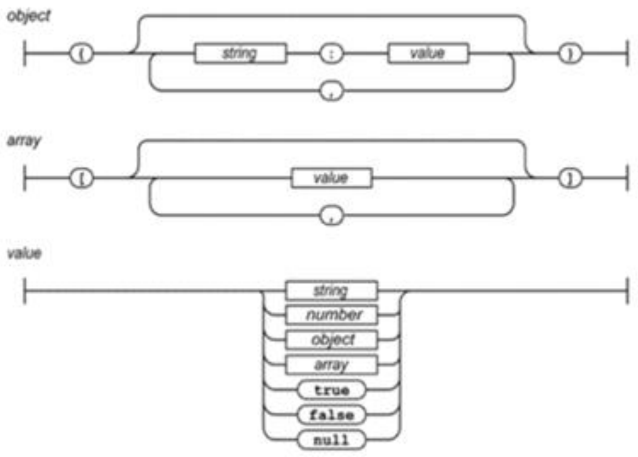


Fig 3:- JSON Data Structure

## III. METHOD RESEARCH

### *A. System Requirements Analysis*

Requirements are divided into 3 parts, namely, data, hardware and software requirements. The data needed is student data, hardware and software requirements, namely the server computer used to run both applications to be integrated, the required software is web server, MySQL and PHP databases. Web service is used for data integration between the Academic Management Information System server and the Library Information System server. Internet network is used for data communication between server and client computers.

### B. *System Architecture*

Data integration architecture design can be illustrated in the following figure 4:

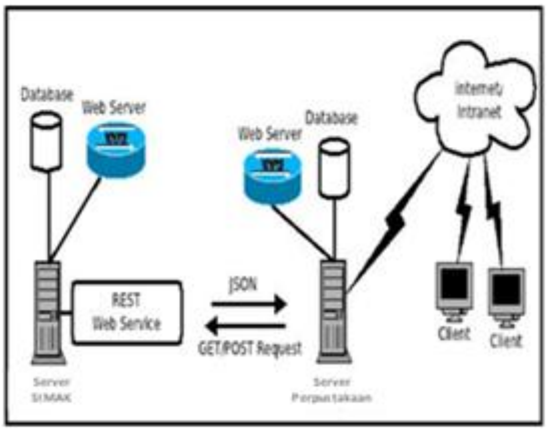


Fig 3: System Integration Architecture

There are two servers namely the academic management information system server and the library information system server. Each server uses a MySQL database. On an academic management information system server a web service is built to serve student data requests. Web service is connected to the academic management information system database. Web service is built using the RESTFull API library with the PHP Framework Codeigniter language, the output of the web service is a JSON string. While on the library server a client service is built to make data requests and retrieve data output from the web service server used by the library information system application for data processing. The library and webservice client application are built using the PHP Framework Codeigniter programming language.

### C. *Designing the API*

Designing the API uses the concept of Representational State Transfer. REST allows clients to make requests via the HTTP protocol easily using URIs. The following is the URI format for making requests to the API:

```
http://{nama_domain}/{nama_fungsi}?{parameter_1}... &{parameter_n}
```

Note:

- ➢ {domain_name} is the domain name where an API is placed;
- ➢ {function_name} is the name of the function to be accessed;
- ➢ {parameter_1} ... & {parameter_n} are the parameters that are sent.

### D. *Development*

At the development stage, the web service coding process uses the RESTFull API library available with the PHP Framework Codeigniter programming language. The coding process is divided into two parts, namely the web service provider coding and requestor. Web service provider aims to provide data services to be accessed by the Library Information System server. Web services using REST are made using the PHP programming language. The request data contains the student ID number, the response data contains the student biodata. Web service request and response data uses the JSON format.

## IV. RESULTS AND DISCUSSION

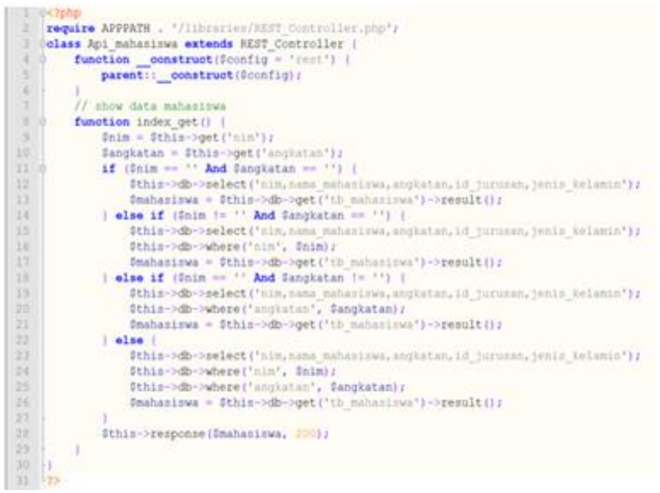
### A. *Web Service Specifications*

Web services are made to make student data requests. Student data is obtained by accessing the academic management information system database server via web service. The web service specifications used are:

1. URL: http://simak.stmikbinsa.ac.id/api_mahasiswa
2. Request method: GET
3. Request format: JSON
4. Response format: JSON
5. Request parameters: nim, force.

### B. *Web Service Provider*

Web service providers are created using the RESTFull API library with the PHP Framework Codeigniter programming language that serves to serve student data requests. Request data is sent in JSON format. Following is the profider web service script.



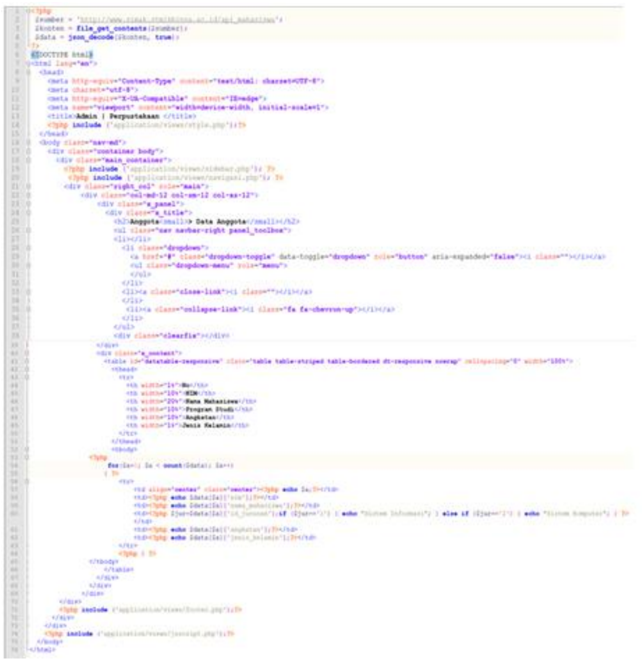
```php
<?php
require APPPATH . '/libraries/REST_Controller.php';
class Api_mahasiswa extends REST_Controller {
    function __construct($config = 'rest') {
        parent::__construct($config);
    }
    // show data mahasiswa
    function index_get() {
        $nim = $this->get('nim');
        $angkatan = $this->get('angkatan');
        if ($nim == '' And $angkatan == '') {
            $this->db->select('nim,nama_mahasiswa,angkatan,id_jurusan,jenis_kelamin');
            $mahasiswa = $this->db->get('tb_mahasiswa')->result();
        } else if ($nim != '' And $angkatan == '') {
            $this->db->select('nim,nama_mahasiswa,angkatan,id_jurusan,jenis_kelamin');
            $this->db->where('nim', $nim);
            $mahasiswa = $this->db->get('tb_mahasiswa')->result();
        } else if ($nim == '' And $angkatan != '') {
            $this->db->select('nim,nama_mahasiswa,angkatan,id_jurusan,jenis_kelamin');
            $this->db->where('angkatan', $angkatam);
            $mahasiswa = $this->db->get('tb_mahasiswa')->result();
        } else {
            $this->db->select('nim,nama_mahasiswa,angkatan,id_jurusan,jenis_kelamin');
            $this->db->where('nim', $nim);
            $this->db->where('angkatan', $angkatam);
            $mahasiswa = $this->db->get('tb_mahasiswa')->result();
        }
        $this->response($mahasiswa, 200);
    }
}
?>
```

Fig 4:- Web Service Server Script Profider

Web service processes by retrieving student data in a database based on data requests. If the request uses parameters, student data will be displayed based on parameter values, if the request does not use parameters, student data will be displayed in its entirety. The student data fields displayed are in the form of student number, student name, department, class, and gender.

### C. *Access Web Service Profider*

Access To access a web service is done by sending data requests to the web service. The following is a script for accessing web service profider:



```
<?php
  $number = 'http://www.simak.stmikbinsa.ac.id/api_mahasiswa';
  $konten = file_get_contents($number);
  $data = json_decode($konten, true);
?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Admin | Perpustakaan </title>
    <?php include ('application/views/style.php');?>
  </head>
  <body class="nav-md">
    <div class="container body">
      <div class="main_container">
        <?php include ('application/views/sidebar.php'); ?>
         <?php include ('application/views/navigasi.php'); ?>
        <div class="right_col" role="main">
            <div class="col-md-12 col-sm-12 col-xs-12">
                <div class="x_panel">
                  <div class="x_title">
                    <h2>Anggota<small>> Data Anggota</small></h2>
                    <ul class="nav navbar-right panel_toolbox">
                    <li></li>
                      <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false"><i class=""></i></a>
                        <ul class="dropdown-menu" role="menu">
                        </ul>
                      </li>
                      <li><a class="close-link"><i class=""></i></a>
                      </li>
                       <li><a class="collapse-link"><i class="fa fa-chevron-up"></i></a>
                      </li>
                    </ul>
                    <div class="clearfix"></div>
                </div>
                <div class="x_content">
                  <table id="datatable-responsive" class="table table-striped table-bordered dt-responsive nowrap" cellspacing="0" width="100%">
                    <thead>
                      <tr>
                        <th width="1%">No</th>
                        <th width="10%">NIM</th>
                        <th width="20%">Nama Mahasiswa</th>
                        <th width="10%">Program Studi</th>
                        <th width="10%">Angkatan</th>
                        <th width="1%">Jenis Kelamin</th>
                      </tr>
                    </thead>
                    <tbody>
                <?php
                    for($a=1; $a < count($data); $a++)
                    { ?>
                      <tr>
                          <td align="center" class="center"><?php echo $a;?></td>
                          <td><?php echo $data[$a]['nim'];?></td>
                          <td><?php echo $data[$a]['nama_mahasiswa'];?></td>
                          <td><?php $jur=$data[$a]['id_jurusan'];if ($jur=='1') { echo "Sistem Informasi"; } else if ($jur=='2') { echo "Sistem Komputer"; } ?>
                          </td>
                          <td><?php echo $data[$a]['angkatan'];?></td>
                          <td><?php echo $data[$a]['jenis_kelamin'];?></td>
                      </tr>
                      <?php } ?>
                </tbody>
                    </table>
                  </div>
                </div>
              </div>
        </div>
        <?php include ('application/views/footer.php');?>
      </div>
    </div>
    <?php include ('application/views/jsscript.php');?>
  </body>
</html>
```

Fig 5:- Web Service Server Access Script

Pada gambar 4.2 merupakan script untuk mengakses web service yang digunakan untuk melakukan request ke server Sistem Informasi Manajemen Akademik (SIMAK), untuk melakukan sinkronisasi data mahasiswa di Sistem Informasi Perpustakaan. Data yang digunakan memiliki format JSON yang terdiri parameter nim dan angkatan.

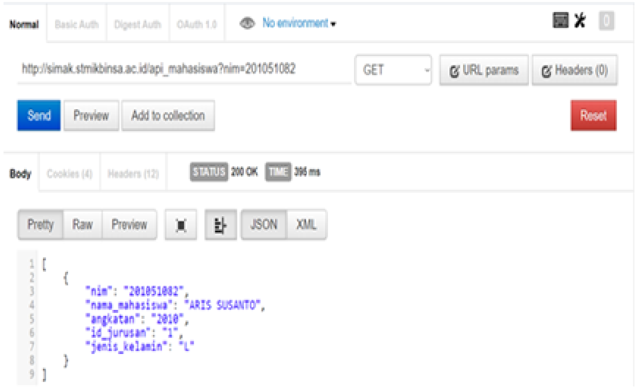
## *D. Testing*

➢ *Trial API*

Testing is carried out to find out whether the web service request process can run well. To test the API used a URI to request student data with nim 201051082. Tools used for testing the API are POSTMAN - Request Client.

| http://simak.stmikbinsa.ac.id/api_mahasiswa?nim=201051082 |
|---|

The URI is sent using the getNIM() function and gets a JSON response like in the following figure:



http://simak.stmikbinsa.ac.id/api_mahasiswa?nim=201051082 GET URL params Headers (0)

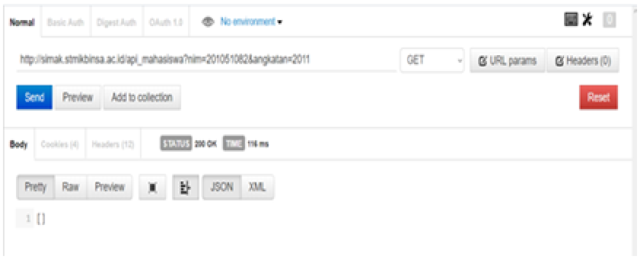
STATUS 200 OK TIME 396 ms

```
[
    {
        "nim": "201051082",
        "nama_mahasiswa": "ARIS SUSANTO",
        "angkatan": "2010",
        "id_jurusan": "1",
        "jenis_kelamin": "L"
    }
]
```

Fig 6:- URI Request Parameter Results

In Figure 6 is a request using the getNIM () parameter which successfully displays student data that matches the parameter values stored in the database server academic management information system.

API testing using the getNIM () parameter with 201051082 and getAngkatan () value with 2011 value then the response given by the web service is not displaying data, this is because the student data records stored in the database do not match the value of the parameter sent. The results are shown in Figure 7:



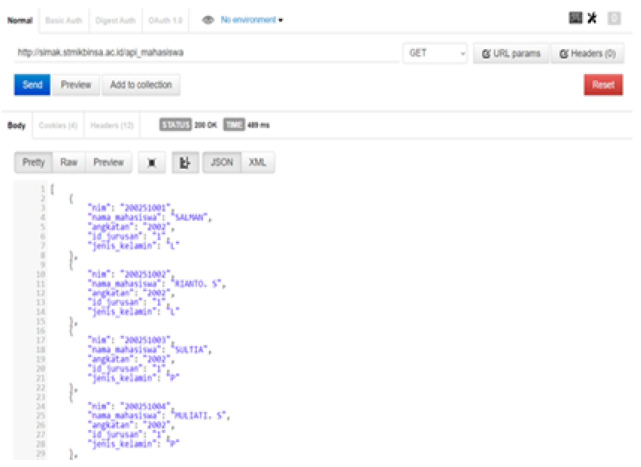
http://simak.stmikbinsa.ac.id/api_mahasiswa?nim=201051082&angkatan=2011 GET URL params Headers (0)

STATUS 200 OK TIME 116 ms

```
[]
```

Fig 7:- Invalid URI Request Results

Furthermore, the API trial does not use parameters so that the student data is displayed as a whole. The results are displayed as in Figure 8:



http://simak.stmikbinsa.ac.id/api_mahasiswa GET URL params Headers (0)

STATUS 200 OK TIME 489 ms

```
[
    {
        "nim": "200251001",
        "nama_mahasiswa": "SALMAN",
        "angkatan": "2002",
        "id_jurusan": "1",
        "jenis_kelamin": "L"
    },
    {
        "nim": "200251002",
        "nama_mahasiswa": "RIANTO. S",
        "angkatan": "2002",
        "id_jurusan": "1",
        "jenis_kelamin": "L"
    },
    {
        "nim": "200251003",
        "nama_mahasiswa": "SULTIA",
        "angkatan": "2002",
        "id_jurusan": "1",
        "jenis_kelamin": "P"
    },
    {
        "nim": "200251004",
        "nama_mahasiswa": "MULIATI. S",
        "angkatan": "2002",
        "id_jurusan": "1",
        "jenis_kelamin": "P"
    },
```

Fig 8:- URI Request Results Do Not Use Parameters

➢ *Application*

Testing by requesting student data through the Academic Management Information System web service with URI http://simak.stmikbinsa.ac.id/api_mahasiswa. The results of student data requests are shown as in Figure 9:

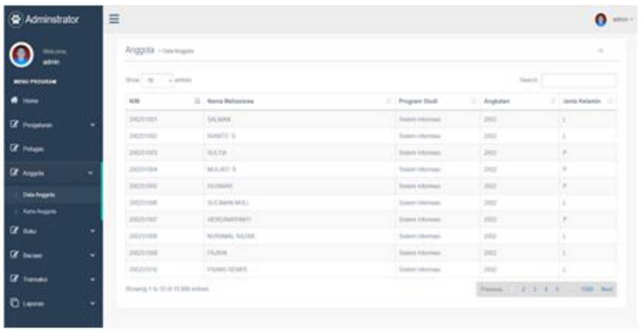


Fig 9:- Results of Request Data in Web Service

## V. CONCLUSION

The application of REST API web service with JSON format as a backend on this system is very suitable for use because with the small JSON format the downloading data from web services is faster. The REST API is easy to access with URIs via the HTTP protocol. But this convenience can be a weakness. URIs can be stolen, as can data. For this reason, an authorization process is needed to prevent invalid requests from entering.

From the experiments that have been done, the data obtained is in accordance with the data sent and student data in the Academic Management Information System database, so it can be concluded that the data obtained is valid and accurate.

This research has succeeded in building a data integration system between information systems namely academic management information systems and library information systems at STMIK Bina Bangsa Kendari. Integrated data is student data sourced from an academic management information system. Data integration is built using the RESTFull web service, the PHP Codeigniter programming language and the MySQL database. Data access is used through a web service by sending data in JSON format, namely student data in the form of student number, student name, department, class and gender.

The results of the study in this paper reveal two things, namely, the student database is the type of data most needed by other information systems in the STMIK Bina Bangsa Kendari environment.

From the REST API Web Service, an academic management information system can be developed to integrate other information systems in the STMIK Bina Bangsa Kendari environment. To improve the security of data web services, the next research can be developed the use of data encryption systems to send data requests and response data. For security, an authorization process is needed to prevent invalid requests from entering.

## REFERENCES

[1]. Cerami, E. (2002). Web Services Essential. O'Reilly.
[2]. [2] Gottschalk, K. (2002), Introduction to Web Services Architecture. IBM System Journal, Vol 41, No 2.
[3]. Lucky, 2008, XML Web services: Aplikasi Desktop, Internet & Handphone, Jasakom, Jakarta.
[4]. [6] Fielding R.T., 200, Architectureal Style & Design of Network-Based Software Architectures, Ph.D. Thesis, Department of Information & Computer Science, UCLA, Irvine.
[5]. Kumari, V., 2015, Web services Protocol : SOAP vs REST, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 5.
[6]. Pautasso, C., 2008, REST vs SOAP Making the Right Architectural Decision, SOA Symposium, Amsterdam.
[7]. M. Bray, Application Programming Interface. The Softwere Engineering Interface Institute, 1997.
[8]. L. J. Mitchell, API's for the Web Modern - PHP Web Services, First Edit. Gravenstein Highway North: O'Reilly Media, 2013.