

Implementation of Web Service Host to Host Payment for Pajak Bumi dan Bangunan PBB Pekanbaru City

1st Wawan Kurniawan
Departement of Computer
Politeknik Caltex Riau
Pekanbaru, Indonesia
wedy20s2tk@mahasiswa.pcr.ac.id

2nd Dadang Syarif Sihabudin Sahid
Departement of Computer
Politeknik Caltex Riau
Pekanbaru, Indonesia
dadang@pcr.ac.id

Abstract—Web service is an application of a collection of data (database), software (software) or part of software that can be accessed remotely by various devices with a certain intermediary. Web service can also be interpreted as a method of exchanging data, regardless of where a database is embedded, in what language an application that consumes data is made, and on what platform a data is consumed. Web services are able to support interoperability. So that the web service is able to become a bridge between the various existing systems.

Keywords— *Web Service, Host to Host, Json, Rest API, HTTP.*

I. INTRODUCTION

Information systems and computer technology are developing very rapidly in line with the magnitude of the need for information. Changes and dynamics of society that are getting faster along with the times and technology demand quality information that is fast, precise and accurate. Information systems are one example of a rapidly growing technology product that can assist humans in processing data and presenting quality information so that currently almost all fields of work use information systems.

Every company agency and government certainly needs an information system in carrying out activities its work so that it is more organized and directed with a more efficient time. Likewise with the city of Pekanbaru, into order realize the vision of a civilized Smart City. Where one of the indicators is Smart Governance (transparent, informative and responsive government) information systems are a necessity for every Regional Apparatus Organization.

Pekanbaru City Regional Revenue Agency is one of the Regional Apparatus Organizations which one of its duties is to collect Land and Building Tax in the Pekanbaru City area. The PBB payment process by involving banks has made it easier to process financial data for the Government. But because the bank payment data directly access the database without using the method web service. This is a problem because the bank should not have direct access to the database because the database is data that is very risky to be accessed by the public.

Several studies to integrate information systems have been carried out, one of which uses web service-based technology [1], [2], [3], [4], [5], [6]. The architecture of the REST method is the configuration with the best latency value to be implemented in the data integration process [7]. Apart from that, REST is also better than SOAP in response time and response data size [8].

Based on this background, in this study, technology was REST-based web service applied to integrate the PBB payment information system managed by banking with the PBB information system managed by the Pekanbaru City Regional Revenue Agency. With this technology, data communication is more secure because the mechanism is not a combination between systems but the provision of specific data access services (Data on Land and Building Tax Payments).

II. RELATED RESEARCH

A. Web Service

Studies related to web services have been carried out. Rohmat Gunawan, Alam Rahmatulloh in a paper using a REST-based web service method, from the results of his research that results in an analysis of the test results is that the Web service exchanges data using XML format over a network that utilizes the standard internet or intranet protocol, namely HTTP [9].

Other research related to payments using the web service that Randi Rizala, Nature Rahmatulloh journal conducts research using methods Rest API, while the conclusions obtained from this study is the format of data exchange between systems using JSON format stateless (stateless), making it easier to be accessed by any programming language, architecture or different operating system [10].

B. Payment Host to Host

A study related to host to host payments, namely Iis Pradesan, this study aims to design and implement a Web Service using the H2H method from banking, while the research method used is descriptive with the RUP information system development methodology [11].

III. THEORETICAL BACKGROUND

A. Web Service

W3C defines a web service as a software system designed to support communication and interaction between machine to machine (Machine to Machine) through a network (network). Web Services also include Web APIs that can be accessed over a network such as the internet and executed via a remote system according to the requested service. The definition of Web Service according to the W3C also includes many different systems, but in general it is more concerned with the client and server communicating using XML that meets the SOAP (Simple Object Access Protocol) standard. The general assumption is that in terminology there is a description of the machine whose services are provided by the server, or the same as the concept of WSDL. WSDL is not a standard of SOAP but is an absolute requirement for automatic client-side in Java and .NET SOAP frameworks. Some industry organizations such as WS-I claim both SOAP and WSDL as the definition of Web Service essence.

B. API

API is a software interface that consists of a collection of instructions stored in the form of a library and describes how a piece of software can interact with other software. This explanation can be exemplified by analogy if a house is to be built. By hiring a contractor who can handle different parts, the homeowner can give the contractor the tasks the contractor needs to do without having to know how the contractor gets the job done. From this analogy, the house is the software that will be made, and the contractor is the API that works on certain parts of the software without having to know the procedure for doing the work [15].

C. Rest API

REST (Representational State Transfer) is an architectural method of communication that is often applied in the development of web-based services. REST architecture, which is generally run via HTTP (Hypertext Transfer Protocol), involves reading a specific web page that contains an XML or JSON file. This file describes and contains the content to be presented. After going through a certain definition process, consumers will be able to access the intended application interface. The specialty of REST lies in the interaction between the client and server which is facilitated by a number of operational types (verbs) and Universal Resource Identifiers (URIs) that are unique to each resource. Each of the GET, POST, PUT and DELETE verbs has a special operational meaning to avoid ambiguity. REST is often used in mobile applications, social networking websites, mashup tools, and automated business processes. The decoupled REST architecture and the light communication load between producers and consumers make it popular in the world of cloud-based APIs, such as those provided by Amazon, Microsoft, and Google. Web-based services that use such a REST architecture are called RESTful APIs (Applications). Programming Interfaces) or REST APIs [12][13].

D. Json

JSON (JavaScript Object Notation) is a lightweight data interchange format, readable and writable by humans, as well as easily translated and made (*generated*) by the computer. This format is based on part of the JavaScript Programming

Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that does not depend on language programming and because it uses a language style commonly used by C family programmers including C, C++, C#, Java, JavaScript, Perl, Python etc. Because of these properties, it makes JSON ideal as a data-exchange language.

JSON is made up of two structures:

- A collection of name/value pairs. In some languages, this is represented as an object (*object*), record (*record*), structure (*struct*), dictionary (*dictionary*), hash table (*hash table*), keyed list (*keyed list*), or *associative array*.
- An(ordered list of values an ordered list of values). In most languages, this is expressed as an array (*array*), vector (*Vector*), lists (*list*), or sequences (*sequences*).[14]

IV. THE PROPOSED MODEL

There are four stages carried out in this study, namely: data collection, system analysis and software requirements, design, implementation and testing as shown in Figure 1



Fig. 1. Stages Carried Out in Study

A. Data Collection

At this stage data collection related to the system to be integrated. An overview of data related to the academic fee payment system is shown in Table I.

TABLE I. DATA AVAILABLE IN THE PBB SYSTEM

No	Name Data	Description
1	SPPT	Determination of Taxpayers
2	Payment of SPPT	Tax Payment List
3	Proof of Payment	List of Payment Proofs

B. Analysis of System and Equipment Needs Software

Based on the analysis of the academic fee payment process carried out by taxpayers, there are 3 entities involved, namely: taxpayers, finance and banking departments. The business process for paying PBB payments is shown in Figure 2.

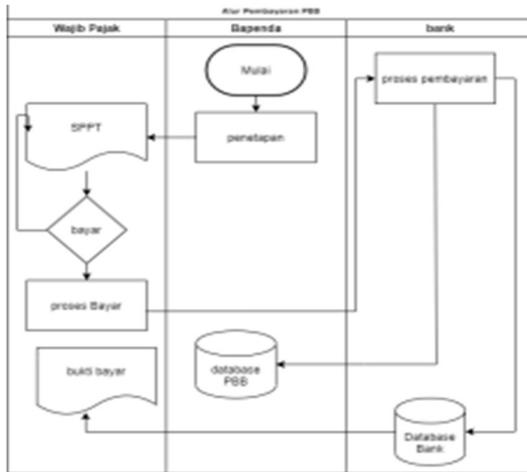


Fig. 2. Flowchart of the ongoing Land and Building Tax Payment Process

In Figure 2, the stages of the current PBB payment process are shown. The explanation of these stages is as follows:

1. The Regional Revenue Agency issues SPPT for taxpayers to make payments immediately.
2. Payment slips or SPPT received by taxpayers can make payments through banks.
3. Banks receive PBB payment fees and provide proof of payment to taxpayers.
4. The bank flags payments directly to the PBB database.

From Figure 2, it can be seen that there are deficiencies in the current system, namely the bank should not be allowed to access the database directly.

C. Design

a) System Architecture Design

There are several main components involved in the system designed, including: user, server, database, service and connection. This system is designed to be able to integrate PBB payment systems stored in banks and integrated host to host. The system architecture designed to support system integration is shown in Figure 3.

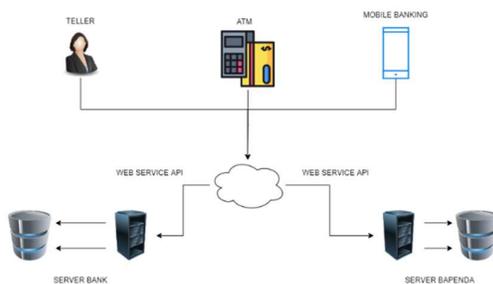


Fig. 3. Web Service API Architecture

In Figure 3 it can be seen, the service used for system integration is placed between the *Server* (BANK) and the *Server* (Bapenda) which is connected by connection. The next stage is the design *service* that will be implemented.

b) Service Design

There are two *services* main that will be built to perform system integration. The *service* is designed to be able to *query data*, *Payments*.

1) Query Data

TABLE II. QUERY DATA

URI Format	:	http://{web}/API-PBB/inquiry_Tagihan
Request Method	:	(GET)
parameter	:	{ "nop": "1471100004001xxxx", "year": "2020" }

Description

- a. Web: Is the web address where the API stored
- b. Body: Parameter requested

2) Payment

TABLE III. PAYMENT

Format URI	:	http://{web}/API-PBB/inquiry_Tagihan
Request Method	:	(GET)
parameter	:	{ "nop": "1471100004001xxxx", "year": "2020", "noref": "xxxx112204", "merchant": "xxxx", "receive_date": "20201203", "ship_date": "20201203" }

Description

- a. Web: Is the web address where the API is stored
- b. Body: Parameters requested, from the bank

V. IMPLEMENTATION AND TESTING

At this stage the implementation of the design *service* that has been done previously is carried out. The Design is *service* implemented into programming using the programming PHP language.

A. Implementation

Implementation of design that has been done in the previous stage implementation (*coding*) into the PHP programming language and data exchange using format *JavaScript Object Notation* (JSON). Pieces *source code* of *web service module* API is shown in Figure 4.

```

<?php
include_once('konfig.php');
include_once('db.php');
date_default_timezone_set('Asia/Jakarta');
$konfig = new konfigurasi();
$tanggal = date("Y-m-d");
$jam = date("H:i:s");
$upat = date("Y-m-d H:i:s");
$getip = $_SERVER['REMOTE_ADDR'];
$tagangalbayar_log = date("Y-m-d H:i:s");
$getip = $_SERVER['REMOTE_ADDR'];

$arr = json_decode(file_get_contents("php://input"));
if (empty($arr)) { // cek array request
    $m_status['status'] = '12';
    $m_status['message'] = "Array tidak dimasukkan";
    $response['Result'] = $m_status;
} else {
    $headers = apache_request_headers();
    if (!isset($headers['Authorization'])) { //cek authorization header jika tidak ada
        $m_status['status'] = '220402';
        $m_status['message'] = "Authorization is null";
        $response['Result'] = $m_status;
    } else { //cek authorization header jika ada
        $specah = explode(" ", $headers['Authorization']);
        $saatanggal = $specah[1];
        $base64UrlPayload = str_replace(['+', '/', '='], ['-', '_', ''], base64_decode($saatanggal));
        $ctanggal_jwt = substr($base64UrlPayload, 12, -2);
        if ($ctanggal_jwt == $ctanggal) { // cek tanggal header jika sama
            $nop = $arr->nop;
            $tahun = $arr->tahun;
            $noref = $arr->noref;
            $merchant = $arr->merchant;
            $tanggal_kerima = $arr->tanggal_kerima;
            $tanggal_kirim = $arr->tanggal_kirim;
            $tnop = $nop;
            $kasirbayar = "BRK000000000000000000";
            $kd_propinsi = substr($tnop, 0, 2);
            $kd_kabupaten = substr($tnop, 2, 3);
            $kd_kecamatan = substr($tnop, 4, 5);
            $kd_kelurahan = substr($tnop, 6, 10);
            $kd_pokok = substr($tnop, 11, 14);
            $kd_denda = substr($tnop, 15, 17);
            $kd_total = substr($tnop, 18, 24);
            $data = [
                "nop" => $nop,
                "tahun" => $tahun,
                "nama" => "MURSAL",
                "alamat" => "KO VILLA INDAH PAUS",
                "tempo" => "2021-08-31 12:00:00",
                "keLurahan" => "TANGKERANG TENGAH",
                "kecamatan" => null,
                "kabupaten" => "PEKANBARU",
                "luas_tanah" => "315",
                "luas_bangunan" => "114",
                "pokok" => 145465,
                "denda" => "0",
                "total" => 145465
            ];
        }
    }
}

```

Fig. 4. Chunks of Source Code the Bill Inquiry Web Service API

```

<?php
ini_set('max_execution_time', 600); //300 seconds = 5 minutes
include_once('konfig.php');
include_once('db.php');

function curl($url, $data) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $output = curl_exec($ch);
    curl_close($ch);
    return $output;
}

$alamatapinya = "https://monitoring-pbb.pekanbaru.go.id/ws/postbayer";

date_default_timezone_set("Asia/Jakarta");
$konfig = new konfigurasi();
$tanggal = date("Y-m-d");
$jam = date("H:i:s");
$upat = date("Y-m-d H:i:s");
$getip = $_SERVER['REMOTE_ADDR'];
$tagangalbayar_log = date("Y-m-d H:i:s");
$getip = $_SERVER['REMOTE_ADDR'];

$arr = json_decode(file_get_contents("php://input"));
if (empty($arr)) { // cek array request
    $m_status['status'] = '12';
    $m_status['message'] = "Array tidak dimasukkan";
    $response['Result'] = $m_status;
} else {
    $headers = apache_request_headers();
    if (!isset($headers['Authorization'])) { //cek authorization header jika tidak ada
        $m_status['status'] = '220402';
        $m_status['message'] = "Authorization is null";
        $response['Result'] = $m_status;
    } else { //cek authorization header jika ada
        $specah = explode(" ", $headers['Authorization']);
        $saatanggal = $specah[1];
        $base64UrlPayload = str_replace(['+', '/', '='], ['-', '_', ''], base64_decode($saatanggal));
        $ctanggal_jwt = substr($base64UrlPayload, 12, -2);
        if ($ctanggal_jwt == $ctanggal) { // cek tanggal header jika sama
            $nop = $arr->nop;
            $tahun = $arr->tahun;
            $noref = $arr->noref;
            $merchant = $arr->merchant;
            $tanggal_kerima = $arr->tanggal_kerima;
            $tanggal_kirim = $arr->tanggal_kirim;
            $tnop = $nop;
            $kasirbayar = "BRK000000000000000000";
            $kd_propinsi = substr($tnop, 0, 2);
            $kd_kabupaten = substr($tnop, 2, 3);
            $kd_kecamatan = substr($tnop, 4, 5);
            $kd_kelurahan = substr($tnop, 6, 10);
            $kd_pokok = substr($tnop, 11, 14);
            $kd_denda = substr($tnop, 15, 17);
            $kd_total = substr($tnop, 18, 24);
            $data = [
                "nop" => $nop,
                "tahun" => $tahun,
                "nama" => "MURSAL",
                "alamat" => "KO VILLA INDAH PAUS",
                "tempo" => "2021-08-31 12:00:00",
                "keLurahan" => "TANGKERANG TENGAH",
                "kecamatan" => null,
                "kabupaten" => "PEKANBARU",
                "luas_tanah" => "315",
                "luas_bangunan" => "114",
                "pokok" => 145465,
                "denda" => "0",
                "total" => 145465
            ];
        }
    }
}

```

Fig. 5. Chunks Of Source Code Modules API Web Service Payments

B. Test

```

POST http://50.10.10.3/API-PBB/inquiry_Token
{
  "username": "bjb",
  "password": "bjb"
}

```

Fig. 6. Generate Token Inquiry UN data

```

pbb-dev / inquiry Tagihan
POST http://192.168.0.6/API-PBB/inquiry_Tagihan...
{
  "nop": "147109000300102210",
  "tahun": "2021"
}
{
  "Result": {
    "status": "00",
    "message": "Request Berhasil",
    "data": {
      "nop": "147109000300102210",
      "tahun": "2021",
      "nama": "MURSAL",
      "alamat": "KO VILLA INDAH PAUS",
      "tempo": "2021-08-31 12:00:00",
      "keLurahan": "TANGKERANG TENGAH",
      "kecamatan": null,
      "kabupaten": "PEKANBARU",
      "luas_tanah": "315",
      "luas_bangunan": "114",
      "pokok": 145465,
      "denda": "0",
      "total": 145465
    }
  }
}

```

Fig. 7. Generate Token Inquiry UN data

C. Analyst Testing Results

After testing, The following is an analysis of the results of the implementation and tests that have been carried out.

1. The PBB payment application on abasis *Host to Host* by implementing technology *web service* has been running according to system requirements.
2. The test system is done in 2 stages: The first stage of performed *testing* on a *web service* that has been built. The second stage is *testing the* application that accesses the *service*, through *interface* the application *Postman*.
3. *Web services* exchange data using JSON format over a network that utilizes the standard internet or intranet protocol, namely HTTP.

VI. CONCLUSION

A system has been designed that can integrate the PBB payment information system managed by banking with the Land and Building Tax information system managed by the Pekanbaru City Revenue Agency by applying web service technology. Data exchange between systems using the format *JavaScript Object Notation (JSON)*. The security of data exchange on the host-to-host system is better than the previous system, because the bank no longer has direct access to the Land and Building Tax database of the Regional Revenue Agency. Meanwhile, to improve the security aspect when exchanging data, an authentication process can be added by applying data encryption and integrating json data with the ISO8583 method.

REFERENCES

[1] Ratih Titi Komala Sari "Applied PBB Electronic Payment With Host To Host Iso 8583 In Local Government", 2017

- [2] IP & BS Officer, "Web Service Implementation In Academic Data Integration With Dikti Database Replication," *Web Service Implementation. On Integr. Contract Data. With Dikti Database Replica*, vol. 40, no. 1, p. 11, 2017.
- [3] R. Hidayat and A. Ashari, "Application of Web Service Technology for Integration of Health Center and Hospital Services," *Berk. MIPA*, (23)1, pp. 64–77, 2013.
- [4] D. Irawati, A. Ashari, and K. Mustofa, "Utilization of Web Services in Data Integration Prototypes of Vocational High School Students Recipients of Student Special Assistance Funds," in *the 2013 National Informatics Seminar (semnasIF 2013)*, 2013, pp. 12–19.
- [5] A. Adi, "Utilization of Web Service as Pharmacy Data Integration at RSU Banyumas (Web Service Useness as a Pharmacy Data Integration in RSU Banyumas)," vol. II, pp. 231–238, 2013.
- [6] B. Santosa, DB Prasetyo, and Y. Pungki, "Integration of online stores using technology web service," *Telematics*, vol. Vol. 8 No., pp. 33–42, 2011.
- [7] Yogiswara, Wiyono, and HS Dahlan, "Web Service Performance in Data Integration Processes," *Eeccis*, vol. 1, no. 1, pp. 73–78, 2014.
- [8] RK Sungkur and S. Daiboo, "SOREST, A Novel Framework Combining SOAP and REST for Implementing Web Services," *SOREST, A Nov. Frames. Comb. SOAP REST Implements. WebServer.*, pp. 22–34, 2015.
- [9] Rohmat Gunawan, Alam Rahmatulloh "Implementation of Web Services in the Host-To-Host System for Payment of Academic Fees", 2018 p-ISSN: 2301-4652 / e-ISSN: 2503-068X
- [10] Randi Rizala, Alam Rahmatulloh "RESTful Web Service for Integration of Academic Systems and Libraries of the University of Struggle' 2019 Scientific Journal of Informatics
- [11] Iis Pradesan, "Design and Implementation of Payment Information Systems in Higher Education Using Web Services Method H2H from Banking" 2017 Jatisi, Vol. 4 No. 1 September 2017
- [12] <https://ekajogja.com/>
- [13] Ali Firdaus 1, Slamet Widodo 2, Adi Sutrisman3, Sutan Gading Fadillah Nasution 4, Rina Mardiana 5 "DESIGN AND BUILD A LIBRARY INFORMATION SYSTEM USING WEB SERVICE IN POLSRI COMPUTER ENGINEERING DEPARTMENT", 2019 Journal of Informatics, Volume 5 No.2, July-December 2019 ISSN:2407-1730
- [14] <https://www.json.org/json-id.html>
- [15] J. Webber, S. Parastatidis, and I. Robinson. 2010. REST in Practice. UnitedStates: O'Reilly Media, Inc.