

Implementasi Web Service Pada Aplikasi Berbasis Web Untuk Mendukung Sistem Penilaian Siswa di SMP Indriasana Palembang

Implementation of Web Services in Web-Based Applications to Support the Student Assessment System at SMP Indriasana Palembang

Hendrik Fery Herdiytmoko

Universitas Katolik Musi Charitas , jl. Bangau no 60, (0711)366326
Jurusan Informatika, Fakultas Sains Dan Teknologi UKMC, Palembang
e-mail: hendrik@ukmc.ac.id

Abstrak

Teknologi Web muncul sebagai platform favorit untuk mengirimkan data dan layanan. Teknologi web digunakan pada gaya arsitektur REST, dimana REST merupakan arsitektur komunikasi client-server berbasis web. REST server menyediakan data berupa URL untuk diakses oleh client dalam bentuk JSON. Pada papper ini akan dijelaskan mengenai pembuatan web service yang ada di Sistem Penilaian Siswa. Menggunakan teknologi RESTful pada pertukaran data dengan bentuk JSON melalui URI (Uniform Resource Identifier). Teknologi restful atau biasa disingkat dengan rest saja menggunakan empat request method dari http method yaitu GET untuk mengambil data, POST untuk memasukkan data, PUT untuk mengubah data dan DELETE untuk menghapus data. Tujuan dari penelitian ini adalah untuk mengeksplorasi desain web service menggunakan metodologi WSIL (Web Service Implementation Lifecycle) yang di dalamnya terdapat tahapan pengembangan web service mulai dari analisis, design, coding, testing dan deployment. Hasil dari penelitian ini dengan menggunakan metode pengembangan sistem Web Service Implementation Methodology (WSIM) telah berhasil dilakukan pada implementasi web service. Dengan metode ini terdapat paradigma baru dalam metode pengembangan perangkat lunak selain metode yang umum lainnya.

Kata kunci— WSIM, web service, restful

Abstract

Web technologies are emerging as a favorite platform for delivering data and services. Web technology is used in the REST architectural style, where REST is a web-based client-server communication architecture. REST server provides data in the form of a URL to be accessed by the client in the form of JSON. In this paper, it will be explained about the creation of a web service in the Student Assessment System. Using RESTful technology to exchange data in JSON form via URI (Uniform Resource Identifier). Restful technology or commonly abbreviated as rest only uses four request methods from the http method, namely GET to retrieve data, POST to enter data, PUT to change data and DELETE to delete data. The purpose of this study is to explore web service design using the WSIL (Web Service Implementation Lifecycle) methodology which includes the stages of web service development starting from analysis, design, coding, testing and deployment. The results of this study using the Web Service Implementation Methodology (WSIM) system development method have been

successfully carried out on the implementation of web services. With this method there is a new paradigm in software development methods in addition to other common methods.

Keywords— *WSIM, web service, restful*

1. PENDAHULUAN

Web 3.0 adalah generasi ketiga dari layanan internet berbasis web. Konsep Web 3.0 diperkenalkan oleh Tim Berners-Lee, penemu *World Wide Web*. Web semantik dan *Web service* adalah unsur utama dari Web 3.0 [1]. Web 3.0 mendesentralisasikan layanan pencarian, sosial media, dan aplikasi *chat* yang bergantung pada satu fungsi atau layanan pada arsitektur *web service* [2][3], yang didalamnya mengandung REST API, secara spesifik, menggunakan protokol HTTP yang menggunakan JSON sebagai format pertukaran data [4].

Aplikasi Penilaian Siswa di SMP Indriasana Palembang pada mulanya adalah aplikasi berbasis web yang dibangun menggunakan Framework Codeigniter [5]. Namun karena tuntutan interoperabilitas yang tinggi maka aplikasi ini dikembangkan lagi menggunakan teknologi *web service* [6]. Tujuannya adalah agar pihak sekolah yaitu guru dan wali kelas dapat menginput nilai menggunakan platform web berbasis *web service*, sedangkan pihak stakeholder (orang tua/wali murid) dapat memantau perkembangan belajar siswa menggunakan aplikasi client *web service*.

Dalam pengembangan aplikasi, dibutuhkan suatu metode pengembangan sistem yang dikhususkan untuk aplikasi berbasis *web service*. Metode pengembangan sistem yang umum seperti Waterfall, Agile, Scrum, RAD, Prototype belum memadai pengembangan sistem khusus untuk aplikasi berbasis *web service*. Untuk itu dibutuhkan suatu metode pengembangan *web service* yang spesifik.

Web Service Implementation Methodology adalah metode pengembangan perangkat lunak *web service* yang menggunakan pendekatan sistematis untuk pengembangan *web service* [7]. *Web service*, tidak seperti sistem *client/server* tradisional, seperti sistem browser/server web, tidak dimaksudkan untuk digunakan pengguna secara langsung. Sebaliknya, *web service* adalah bagian dari logika bisnis, yang memiliki antarmuka terprogram dan melalui antarmuka inilah pengembang dapat membuat sistem aplikasi baru.

2. METODE PENELITIAN

2.1. *Web Service Implementation Methodology (WSIM)*

Metodologi Implementasi *web service* mendefinisikan pendekatan sistematis untuk pengembangan *web service* dengan memanfaatkan metodologi pengembangan perangkat lunak yang gegas dan memperluas metodologi tersebut dengan menentukan aktivitas spesifik *web service* dan peran serta produk kerja yang sesuai yang dihasilkan dalam proses

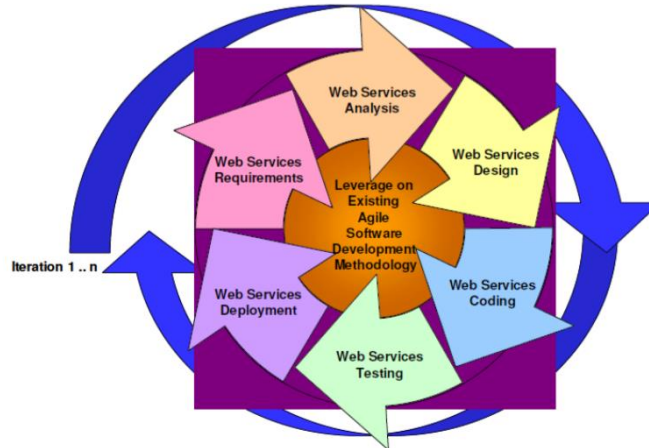
2.2. *Web Service Implementation Lifecycle*

Siklus hidup implementasi *web service* mengacu pada fase untuk mengembangkan *web service* dari *requirement* sampai ke tahap *deployment*.

Siklus hidup implementasi *web service* biasanya mencakup fase berikut:

- a. *Requirements Phase*
- b. *Analysis Phase*
- c. *Design Phase*
- d. *Coding Phase*
- e. *Test Phase*
- f. *Deployment Phase*

Transisi pada fase-fase ini prosesnya cenderung berulang dan bersifat inkremental dan harus cukup cepat untuk mengakomodasi perubahan dalam situasi dimana ruang lingkup tidak dapat sepenuhnya ditentukan sebelumnya.



Gambar 1. *Web Service Implementation Lifecycle* (sumber : *Web Service Implementation Methodology* [7])

Berikut penjelasan fase-fase siklus hidup implementasi *web service* :

a. Fase *Requirement*

Pada fase *requirement* ini tujuannya untuk memahami kebutuhan bisnis dan menerjemahkannya menjadi kebutuhan *web service* dalam hal fitur, fungsional dan kebutuhan *non* fungsional dan kendala

b. Fase *Analysis*

Fase ke dua ini, *requirement web service* lebih disempurnakan dan diterjemahkan ke dalam model konseptual dimana tim pengembangan teknis dapat memahami. Pada fase ini juga dapat dilakukan analisis arsitektur untuk mendefinisikan struktur *high-level* dan mengidentifikasi antarmuka dari *web service*.

c. Fase *Design*

Pada fase *design* ini, desain rinci *web service* dilakukan. Dalam fase ini, para designer *web service* harus mendefinisikan kontrak antarmuka *web service* yang telah diidentifikasi dalam tahap analisis. Pendefinisian antarmuka *web service* harus mengidentifikasi unsur – unsur dan jenis data yang sesuai dengan jenis interaksi antara *web service* dan klien

d. Fase *Coding*

Tujuan dari fase *coding* dan *debugging* adalah untuk mengimplementasikan *web service*, pada dasarnya sangat mirip dengan fase *coding* dan *debugging* pada perangkat lunak berbasis komponen.

e. Fase *Testing*

Untuk pengujian *web service*, selain pengujian untuk kebenaran dan kelengkapan fungsional, Selain itu, pengujian kinerja harus dilakukan untuk memastikan bahwa *web service* mampu menahan beban maksimum sebagaimana ditentukan dalam *requirement non* fungsional.

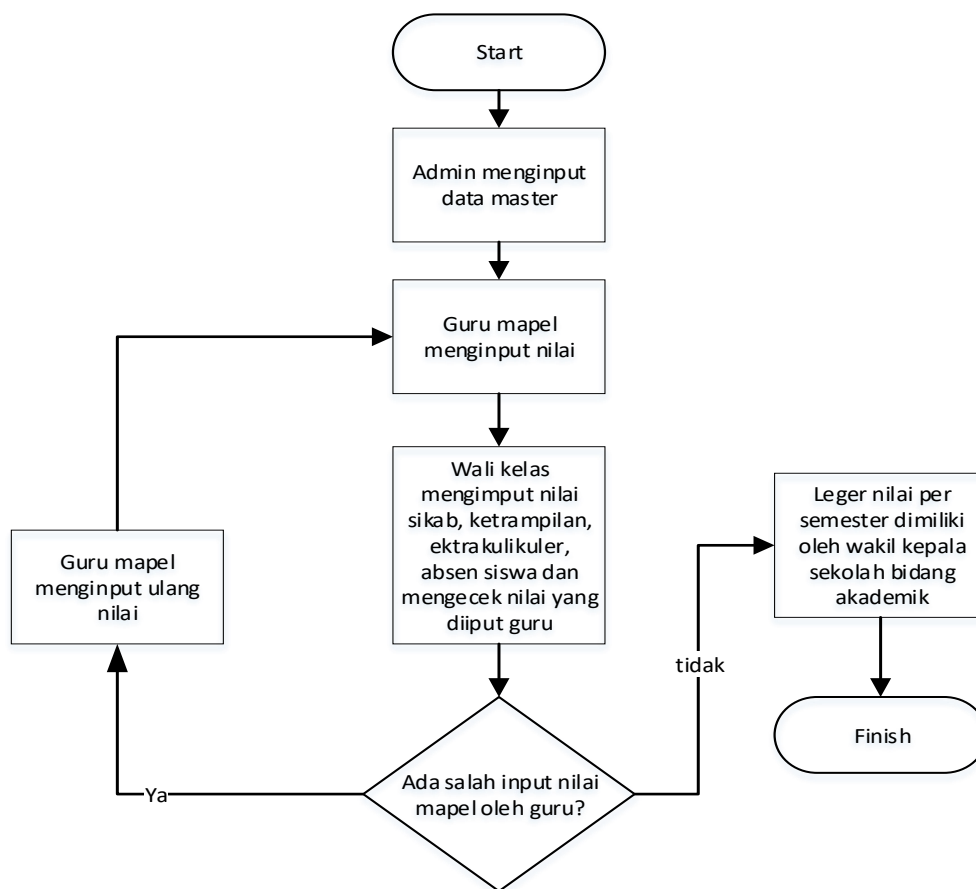
f. Fase *Deployment*

Tujuan dari fase *deployment* adalah untuk memastikan bahwa *web service* digunakan dengan benar. Fase ini dilaksanakan setelah dilakukan fase pengujian. *Deployment web service* dilakukan pada *platform* tertentu. Titik akhir dari layanan *web service* adalah menentukan dimana layanan ini digunakan dan menyesuaikan identifikasi dan konfigurasi. Tugas utama pengembang adalah untuk memastikan bahwa *web service* telah dikonfigurasi dan dikelola dengan benar, dan menjalankan tes pasca *deployment* untuk memastikan bahwa *web service* memang siap untuk digunakan

2.3. Analisis Sistem Aplikasi Penilaian Siswa

a. Analisis Kebutuhan Sistem

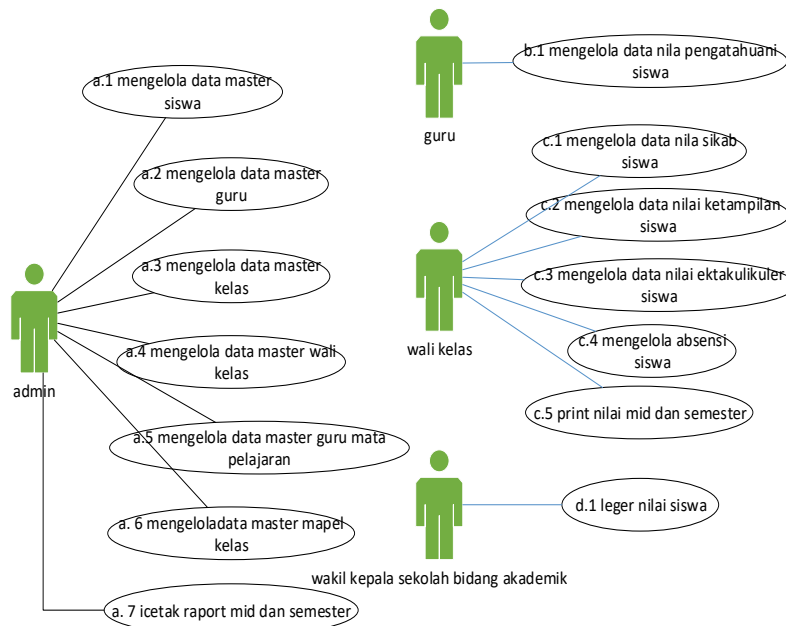
Analisis sistem adalah mengidentifikasi kebutuhan sistem dan alur proses data yang berjalan. Proses utama dari sistem penilaian siswa adalah proses input nilai siswa. Tahap pertama input nilai adalah admin menginput data master, yaitu data siswa, data guru, data guru mata pelajaran, data mata pelajaran, data kelas, data tahun akademik, data wali kelas ke dalam sistem pada tahap input data master tidak ditoleransi input data salah karena akan berakibat pada proses data input nilai. Tahap kedua yaitu guru mapel menginput data nilai sesuai dengan mata pelajaran yang diampunya. Tahap ketiga adalah wali kelas memasukkan data nilai sikab, ketrampilan, nilai ekstrakurikuler, absen siswa. Tahap ke empat adalah leger nilai siswa yang dapat dilihat oleh user wakil kepala sekolah bidang akademik. Gambar 2 menjelaskan proses utama input nilai pada sistem.



Gambar 2. Proses utama memasukkan data nilai siswa

a. Use Case Diagram

Use case diagram [8] merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. Use case diagram bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case diagram juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem. Komponen tersebut kemudian menjelaskan komunikasi antara aktor dengan sistem yang ada. Gambar 3 adalah interaksi aktor dan sistem.



Gambar 3. Use Case Diagram Aplikasi Penilaian Siswa

3. HASIL DAN PEMBAHASAN

b. Fase Analisis

Pada fase analisis ini akan ditentukan arsitektur dari *web service* yang akan dibangun, memilih platform yang akan digunakan, menentukan kandidat *web service*, menentukan detail *web service*.

Kandidat *web service*

Kandidat *web service* merupakan objek yang akan dijadikan *web service*. Kandidat *web service* yang akan dibangun ditentukan berdasarkan kebutuhan fungsional *web service* yang telah ditentukan sebelumnya pada tahap analisis. Untuk lebih rincinya kandidat *web service* bisa dilihat pada Tabel 1 berikut:

Tabel 1. Kandidat *web service*

No	Use Case	Service	Keterangan
1	a.1	mengelola data master siswa	Menambah, mengubah, menghapus data master siswa
2	a.2	mengelola data master guru	Menambah, mengubah, menghapus data master guru
3	a.3	mengelola data master kelas	Menambah, mengubah, menghapus data master kelas
4	a.4	mengelola data master wali kelas	Menambah, mengubah, menghapus data master wali kelas
5	a.5	mengelola data master mata pelajaran	Menambah, mengubah, menghapus data master mata pelajaran
6	a.6	mengelola data master guru mapel	Menambah, mengubah, menghapus data master guru mapel

Identifikasi *web service interface*

Identifikasi *Web service interface* berfungsi untuk menentukan antarmuka dari *web service* yang akan dibangun. Tahap ini dilakukan dengan menentukan HTTP Method yang digunakan berdasarkan operasi yang dilakukan oleh *service* tersebut. Tabel 2 berikut ini mendeskripsikan mengenai identifikasi *web service interface*.

Implementasi Web Service Pada Aplikasi Berbasis Web Untuk Mendukung Sistem Penilaian Siswa di SMP Indriasana Palembang

Tabel 2. Identifikasi web service inteface

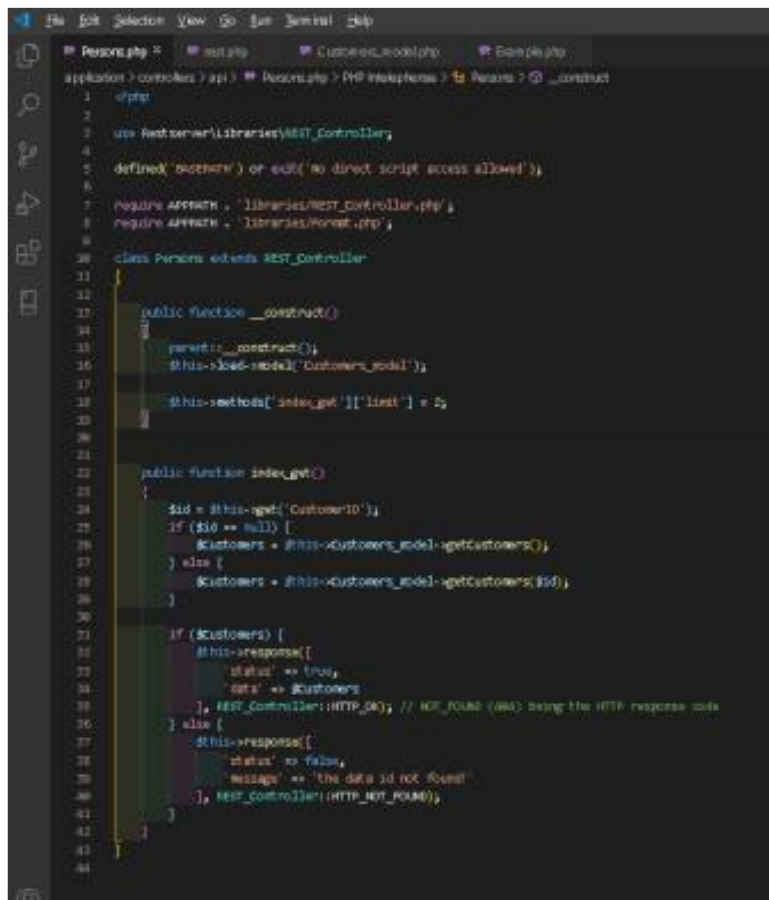
No	Service	http method
1	Input data master siswa	POST
2	mengelola data master siswa	PUT, GET, DELETE
3	Menambah data master kelas	POST
	mengelola data master kelas	PUT, GET, DELETE, POST
4	Menambah data master wali kelas	POST
5	mengelola data master wali kelas	PUT, GET, DELETE, POST
6	Menambah data master mata pelajaran	POST
7	mengelola data master mata pelajaran	PUT, GET, DELETE, POST
8	Menambah data master guru mapel	POST
9	Mengelola data master guru mapel	PUT, GET, DELETE, POST

c. Fase Desain

Pada tahap desain dilakukan langkah *Web service design* dilakukan dengan cara mendesain *interface web service* dari sisi klien.

d. Fase Coding

Sebelum melakukan konfigurasi otentikasi REST Server, perlu dilakukan konfigurasi *endpoint* yang merupakan bagian dari RESTful API. *Pseudocode* pada Gambar 3 berikut adalah *endpoint* dari *method* GET.



```
1 </php>
2
3 use Restserver\Libraries\REST_Controller;
4
5 defined('BASEPATH') or exit('no direct script access allowed');
6
7 require APPPATH . 'libraries/REST_Controller.php';
8 require APPPATH . 'libraries/Person.php';
9
10 class Person extends REST_Controller
11 {
12     public function __construct()
13     {
14         parent::__construct();
15         $this->load_model('Customers_model');
16         $this->methods['index_get'] = ['list' => 2];
17     }
18
19     public function index_get()
20     {
21         $id = $this->get('CustomerID');
22         if ($id == null) {
23             $customers = $this->customers_model->getcustomers();
24         } else {
25             $customers = $this->customers_model->getcustomers($id);
26         }
27
28         if ($customers) {
29             $this->response([
30                 'status' => true,
31                 'data' => $customers
32             ], REST_Controller::HTTP_OK); // HTTP_OK (200) being the HTTP response code
33         } else {
34             $this->response([
35                 'status' => false,
36                 'message' => 'the data id not found'
37             ], REST_Controller::HTTP_NOT_FOUND);
38         }
39     }
40 }
41
42
43
44
```

Gambar 4. Pseudocode method GET

Konfigurasi autoload.php

```
$autoload['libraries'] = array('database');
/*
|-----
| Auto-load Drivers
|-----
|| Prototype:
|| $autoload['drivers'] = array('cache');
|| You can also supply an alternative property name to be assigned in
|| the controller:
|| $autoload['drivers'] = array('cache' => 'cch');
*/
$autoload['drivers'] = array();
/*|-----
| Auto-load Helper Files
|-----
| Prototype:
|| $autoload['helper'] = array('url', 'file');
*/
$autoload['helper'] = array('url');
```

Gambar 5. Konfigurasi file autoload.php

Fungsi dari konfigurasi file autoload.php di dalam folder application-config adalah untuk mengatur spesifik sistem yang akan di-load secara otomatis. Terdapat beberapa hal yang diatur di autoload.php ini, yaitu :

- a. Packages
- b. Libraries
- c. Drivers
- d. Helper files
- e. Custom config files
- f. Language files
- g. Models

Konfigurasi config.php

Konfigurasi file config.php di dalam folder application-config memiliki tujuan mengatur base_url, yaitu url utama yang akan dipanggil sistem secara otomatis, kemudian menghilangkan index.php agar url tampil lebih rapih. Berikut beberapa konfigurasi dari file config.php tersebut.

Implementasi Web Service Pada Aplikasi Berbasis Web Untuk Mendukung Sistem Penilaian Siswa di SMP Indriasana Palembang

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

/*
|-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
| http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|
|*/
$config['base_url'] = 'http://localhost/RestServer21/';
/*
|-----
| Index File
|-----
|
| Typically this will be your index.php file, unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
|*/
```

Gambar 6. Konfigurasi file config.php

Memanggil library ke dalam Controller

Tujuan dari memanggil library ini agar library dapat dikenali oleh controller codeigniter 3 dan dapat digunakan untuk membuat rest server. Berikut potongan program memanggil library ke dalam controller utama.

```
<?php

use Restserver\Libraries\REST_Controller;

defined('BASEPATH') or exit('No direct script access allowed');

require APPPATH . 'libraries/REST_Controller.php';
require APPPATH . 'libraries/Format.php';
```


Gambar 7. Potongan program memanggil library ke dalam controller utama*Konfigurasi method Post*

Konfigurasi method post dilakukan di controller utama *method post*. Fungsi *method* ini untuk menjalankan *http request* POST dari *client*. Berikut potongan program dari method post.

```
public function index_post()
{
    $data = [
        'CustomerName' => $this->post('CustomerName'),
        'ContactName' => $this->post('ContactName'),
        'Address' => $this->post('Address'),
        'City' => $this->post('City'),
        'PostalCode' => $this->post('PostalCode'),
        'Country' => $this->post('Country')
    ];

    if ($this->Customers_model->createCustomers($data) > 0) {
        $this->set_response([
            'status' => true,
            'message' => 'new data customers has been created'
        ], REST_Controller::HTTP_CREATED);
    } else {
        $this->set_response([
            'status' => false,
            'message' => 'failed create new data'
        ], REST_Controller::HTTP_BAD_REQUEST);
    }
}
```

Gambar 8. Potongan program dengan *request method* POST

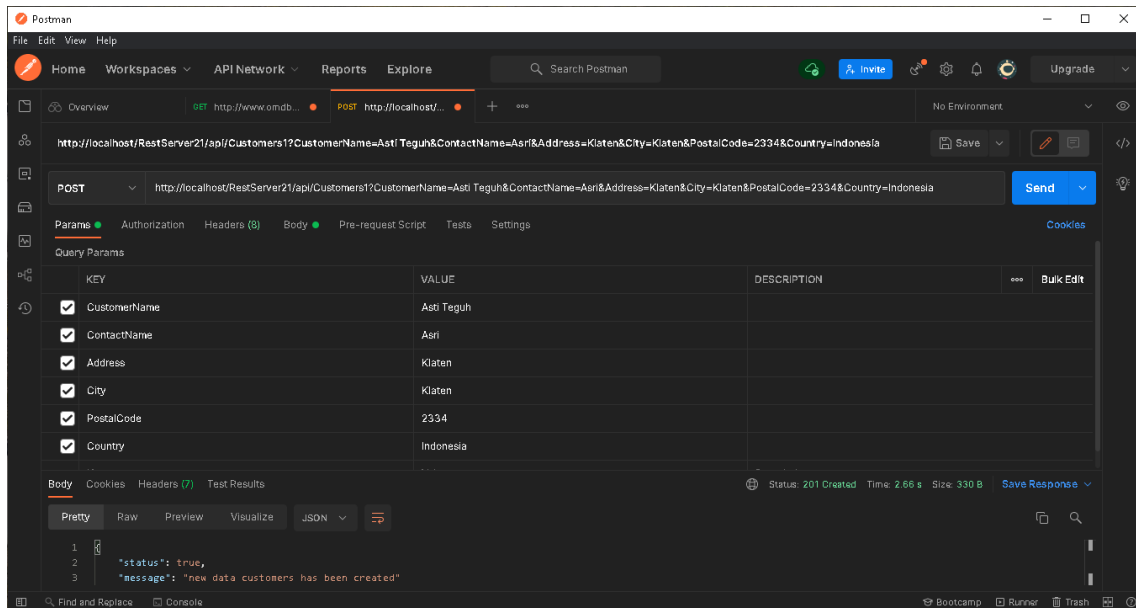
e. Fase Testing

Langkah pertama dari fase *testing* adalah tahap otentikasi REST Server. Menggunakan tiga fase otentikasi yaitu otentikasi dengan key, otentikasi dengan *limit access* dan ketiga otentikasi dengan login.

Testing request method POST

Tujuan utama dari proses test ini adalah mendapatkan respon dari server apakah *request method post* sudah mengembalikan data berupa JSON atau belum. Untuk mendapatkan respon JSON dari rest server request dilakukan di body bagian x-www-form-urlencoded dan isikan key dan value yang diminta. Berikut hasil request method POST dengan aplikasi Postman.

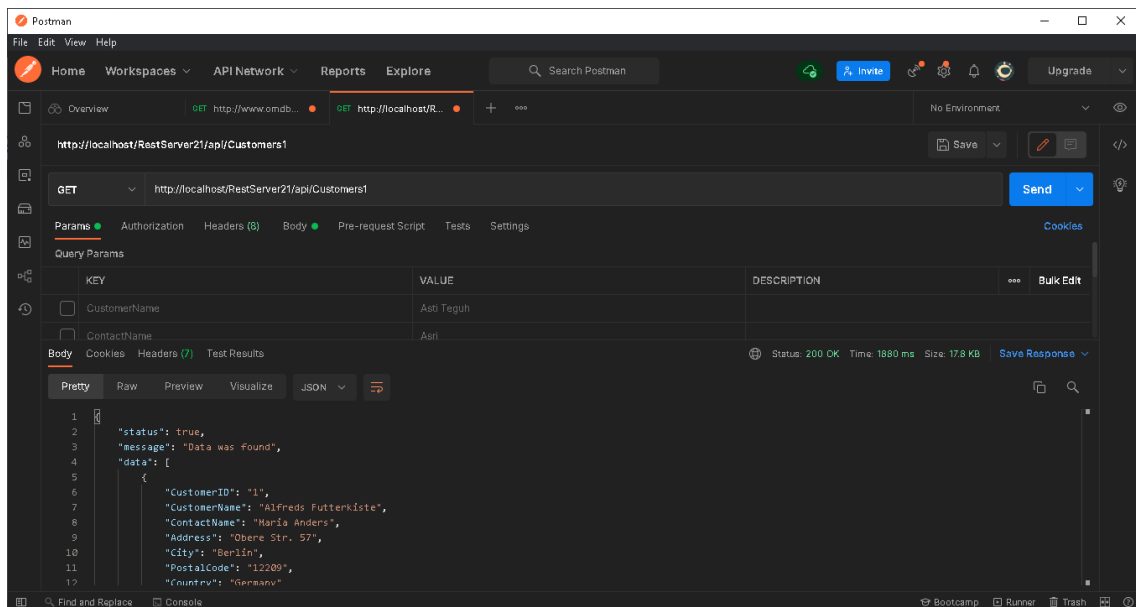
Implementasi Web Service Pada Aplikasi Berbasis Web Untuk Mendukung Sistem Penilaian Siswa di SMP Indriasana Palembang



Gambar 9. Respon rest server untuk *request method* POST

Testing Request method GET

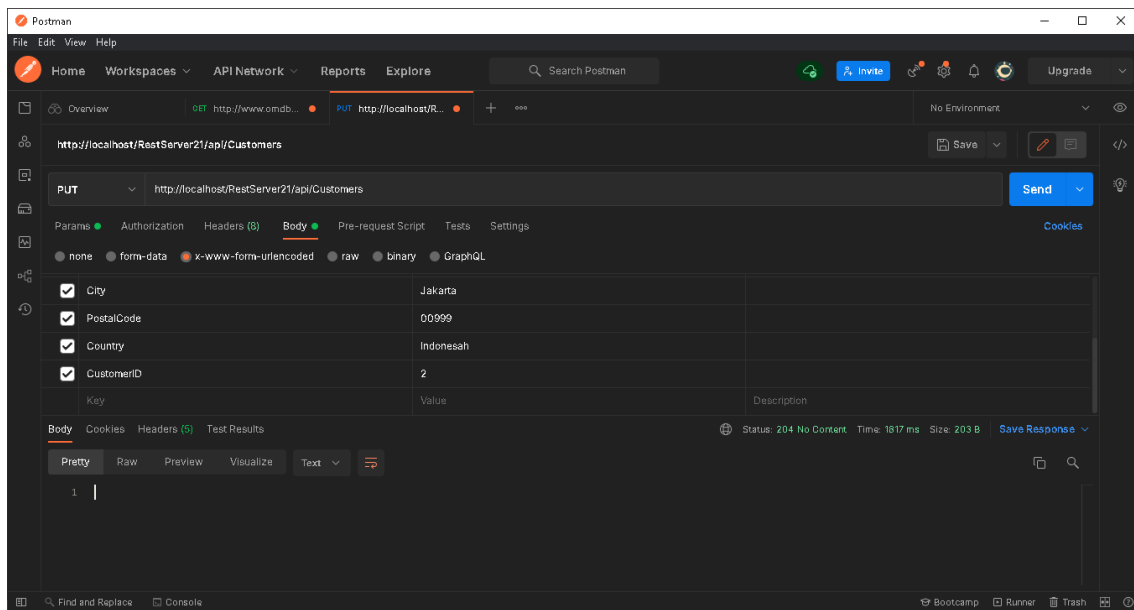
Request method GET digunakan untuk mendapatkan atau mengambil data keseluruhan atau sebagian berdasar kunci tertentu. Respon dari rest server berupa JSON. Khusus untuk *request method* GET dilakukan di param bukan di body pada aplikasi Post man. Berikut contoh *request method* get dari rest server.



Gambar 10. Request method GET dari rest server dengan aplikasi Postman

Testing request method PUT

Tujuan dari request method PUT adalah untuk mengubah data sama dengan perintah SQL UPDATE. Berikut contoh respon dari request method PUT di dalam aplikasi Postman.



Gambar 11. Request method PUT

f. Fase Deployment

Titik akhir dari layanan *web service* adalah menentukan dimana layanan ini digunakan dan menyesuaikan identifikasi dan konfigurasi. Tugas utama pengembang adalah untuk memastikan bahwa *web service* telah dikonfigurasi dan dikelola dengan benar, dan menjalankan tes pasca deployment untuk memastikan bahwa *web service* memang siap untuk digunakan.

4. KESIMPULAN

Metode pengembangan sistem menggunakan *Web Service Implementation Methodology* (WSIM) telah berhasil dilakukan pada implementasi *web service* penilaian siswa. dengan metode ini terdapat paradigma baru dalam metode pengembangan perangkat lunak selain metode yang umum lainnya

5. SARAN

Untuk penelitian lebih lanjut dapat digunakan *Rest client* pada user siswa dengan berbasis android, dengan fasilitas siswa atau orang tua siswa dapat melihat nilai siswa selama siswa tersebut aktif.

DAFTAR PUSTAKA

- [1] J. Luis, "Web 3.0 Emerging."
- [2] Nova Spivack, "No Title," *KurzweilAI.net*, 2006. [Online]. Available: <https://www.kurzweilai.net/the-third-generation-web-is-coming>.
- [3] S. P. Ong *et al.*, "The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles," *Comput. Mater. Sci.*, vol. 97, pp. 209–215, 2015.
- [4] R. T. Fielding, "*Architectural Styles and the Design of Network-based Software Architectures*," pp. 130–139, 2000.
- [5] Codeigniter 3, "No Title," 2015. [Online]. Available: <https://api.github.com/repos/bcit-ci/CodeIgniter/zipball/3.1.11>. [Accessed: 20-Aug-2020].
- [6] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, "Restful API Architecture Based on Laravel Framework," *J. Phys. Conf. Ser.*, vol. 910, no. 1, 2017.

- [7] "Web Service Implementation Methodology." [Online]. Available: <https://www.oasis-open.org/committees/download.php/13420/fwsi-im-1.0-guidlines-doc-wd-publicReviewDraft.htm>.
- [8] "Learn UML Simply Easy Learning." [Online]. Available: <https://www.tutorialspoint.com/uml/index.htm>. [Accessed: 15-Mar-2019].