# IMPLEMENTASI REST API WEB SERVICE DALAM MEMBANGUN APLIKASI MULTIPLATFORM UNTUK USAHA JASA

*By* Ahmat Adil

WORD COUNT 3805

TIME SUBMITTED 06-JUL-2022 09:38AM

PAPER ID 88066571

# IMPLEMENTASI *REST* API *WEB SERVICE* DALAM MEMBANGUN APLIKASI *MULTIPLATFORM* UNTUK USAHA JASA

**Romi Choirudin[1], Ahmat Adil[2]**
[1,2]Program Studi Teknik Informatika
[1,2] STMIK Bumigora Mataram JL. Ismail Marzuki Mataram
Telp (0370) 634498, Fax (0370) 638369
[1]Email : romi@stmikbumigora.ac.id, [2]Email : adilahmat@gmail.com

## ABSTRAK

Perkembangan kegiatan pariwisata di pulau Lombok, menyebabkan banyak macam profesi yang ditekuni atau dimanfaatkan oleh masyarakat khusunya dibindang jasa. Dengan keamjuan teknologi masyarakat membutuhkan berbagai macam kemudahan dalam menjalankan usaha dan aktivitasnya. Salah satunya adalah dalam mencari jasa tukang termasuk harga atau biaya jasa, harga material dan lain sebagainya. Aplikasi multiplatform (lebih dari satu platform) dapat memberikan kemudahan pengguna dalam memilih platform yang akan digunakan. Dengan arsitektur web service sebagai sebuah entitas komputasi yang dapat di akses melalui jaringan internet maupun intranet dan standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen. Dalam melakukan penelitian, metode pengembangan sistem yang digunakan penulis adalah metode waterfall. Metode waterfall merupakan metode pengembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode (Implementasi), test (Pengujian), dan pemeliharaan. Aplikasi yang dibangun telah diuji kepada beberapa tukang. Serta dari hasil pengujian responden tukang memiliki jawaban sangat setuju sebesar 51.4%, dikarenakan 51.4% merupakan hasil tertinggi pada respon penilaian jawaban maka dapat disimpulkan aplikasi telah memenuhi kebutuhan tukang.

**Kata kunci** : *Rest Api Web Service, Aplikasi Multi platfoem, Usaha Jasa.*

## *ABSTRACT*

The development of tourism activities on the island of Lombok has caused many types of professions to be occupied or utilized by the community, especially those who are services. With the technological advancements, people need various kinds of facilities in carrying out their businesses and activities. One of them is in looking for artisan services including price or service fees, material prices and so on. Multiplatform applications (more than one platform) can make it easy for users to choose the platform to be used. With web service architecture as a computational entity that can be accessed through the internet and intranet networks and certain protocol standards in the platform and interface of an independent programming language. In conducting research, the system development method used by the author is the waterfall method. Waterfall method is a systematic and sequential method of software development that starts at the level and progress of the system up to the analysis, design, code (implementation), test (testing), and maintenance. The application that was built has been tested to several builders. As well as from the results of testing the artisan respondents have a very agreeable answer of 51.4%, because 51.4% is the highest result in the response to the response of the answer, it can be concluded that the application has met the needs of the mason

***Keywords:*** *Rest API Web Service, Multi-platform application, Service Business.*

## I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi menyebabkan manusia terus berinovasi untuk memenuhi kebutuhannya. Teknologi informasi merupakan alat bantu manusia untuk memproses kegiatan yang menghasilkan informasi [1].

Pulau Lombok saat ini sedang berkembang dalam bidang pariwisata dan pendidikan menjadi faktor utama menyebabkan banyak pendatang berkunjung ke Lombok termasuk kota Mataram, baik wisatan manca negara maupun wisatawan nusantara. Karena perkembangan tersebut pula, maka banyak macam profesi yang ditekuni atau dimanfaatkan oleh masyarakat khusunya dibindang jasa. Salah satu profesi jasa yang terus berkembangan seiring perkembangan kota adalah adalah jasa tukang dalam membangun infrastruktur. Tukang merupakan orang atau sekelompok orang yang mempunyai kepandaian dalam suatu pekerjaan tangan (dengan alat atau bahan yang tertentu)[2].

Dengan banyaknya orang yang berdatanganke pulau Lombok untuk berinvestasi dalam

membangun infrastruktur baik pariwisata dan bidang lainnya, maka makin banyak pula membutuhkan jasa tukang untuk membangun. Kesulitan para pendatang adalah dalam mencari jasa tukang termasuk harga atau biaya jasa, harga material dan lain sebagainya. Hal ini menjadi kendala bagi calon pemesan untuk melakukan proses pemesanan jasa tukang. Untuk mengatasi masalah tersebut, para calon pemesan jasa tukang biasanya melakukan *survey* terhadap beberapa tukang dengan mendatagi beberapa lokasitertentu untuk mencari jasa tukang dan harga jasa. Cara survey seperti ini dapat membutuhkan waktu, biaya dan tenaga yang terkadang tidaksedikit[2].

Tukang juga melakukan promosi secara konvensional, seperti menggunakan spanduk, pamflet, kartu nama. Cara promosi seperti ini kurang efektif dalam meningkatkan pendapatan, karena selain membutuhkan biaya, waktu tidak sedikit, cakupan luas area promosi yang terbatas. Untuk mengoptimalkan pemesanan jasa tukang. Untuk itu dibutuhkan sistem yang dapat digunakan untuk media promosi tukang serta dapat berinteraksi antara calon pemesan jasa tukang dengan tukang tersebut.

Aplikasi *multiplatform* (lebih dari satu *platform*) dapat memberikan kemudahan pengguna dalam memilih *platform* yang akan digunakan. Pengguna tidak akan merasa dituntut untuk menggunakan jenis komputer tertentu, baik itu desktop ataupun mobile. Sebagai sebuah entitas komputasi arsitektur *web service* dapat diakses melalui jaringan internet maupun intranet, dengan standar *protocol* tertentu dengan *platform dan interface* bahasa pemrograman yang independen [3]. Arsitektur *web service* dikembangkan sebagai jembatan komunikasi antar program, sehingga antara aplikasi terdapat jaringan yang sama dengan standar protocol yang sama yang ditetapkan oleh *web service*. Dengan menggunakan arsitektur *web service* sistem dapat terintegrasi walaupun dengan *platform* lain.

Dengan adanya aplikasi *mulitplatfrom* selain pengguna dapat terintegrasi meskipun menggunakan *platform* yang berbeda, tukang maupun calon pemesan tukang dapat saling berinteraksi dengan sistem sehingga proses pemesanan jasa tukang serta pencarian dapat menjadi lebih mudah.

**1. Arsitektur Web Service**

Web Service adalah mekanisme komunikasi yang didefinisikan antara sistem komputer yang berbeda. Tanpa web service, komunikasi peer- to-peer kustom menjadi rumit dan spesifik platform. Ini seperti ratusan jenis hal yang perlu dipahami dan ditafsirkan oleh web. Jika sistem komputer sejajar dengan protokol yang dapat dipahami web dengan mudah, ini sangatmembantu [4].

Tiga komponen standar Web *Service* adalah sebagai berikut [4] :

a. *Simple Object Access Protocol* (SOAP), sebagai protokol yang menangani pertukaran informasi pada jaringan terditribusi.
b. *Web Service Definition Lmanguage* (WSDL), mendefinisikan kehadiran *web service* dalam suatu jaringan sekaligus merupakan dokumen standar yang dituliskan dalam format XML.
c. *Universal Deskription, Discovery, and Integration* (UDDI), dituliskan berbasis XML dan dapat diakses oleh entitas yang berada di dalam jaringan, dan juga menunjukan lokasi direktori yang berisi *service* dan bersifat bebas *platfrom*.

**2. RESTful Web Service**

Nama Representational state transfer (REST) diciptakan oleh Roy Fielding dari University of California. Ini adalah layanan web yang sangat sederhana dan ringan dibandingkan dengan SOAP. Kinerja, skalabilitas, kesederhanaan, portabilitas, dan kemampuan modifikasi adalah prinsip utama di balik desain REST[5].

REST API memungkinkan berbagai sistem untuk berkomunikasi dan mengirim / menerima data dengan cara yang sangat sederhana. Setiap dan setiap panggilan API REST memiliki hubungan antara kata kerja HTTP dan URL. Sumber daya dalam database dalam suatu aplikasi dapat dipetakan dengan titik akhir API di REST[6].

Ketika menggunakan aplikasi seluler di ponsel Anda, ponsel Anda mungkin secara diam-diam berbicara dengan banyak layanan cloud untuk mengambil, memperbarui, atau menghapus data Anda. Layanan REST memiliki dampak besar pada kehidupan kita sehari-hari [7] .

Layanan Web RESTful adalah layanan yang antarmuka dan mekanisme

pengaksesannya selaras dengan prinsip REST, dimana REST merupakan satu jenis *web service* yang menerapkan konsep perpindahan antar *state* [7]. Cara kerja *web service* yaitu dengan bernavigasi melalui *link-link* yang disediakan seperti mengganti *state* dari halaman web. Sementara REST bekerja dengan bernavigasi melalui *link-link* HTTP dalam aktivitas tertentu, sehingga seperti terjadi perpindahan antara *state*.

Beberapa perintah HTTP antara lain fungsi GET, POST, PUT, UPDATE atau DELETE. Supaya informasi yang diterima lebih mudah dibaca dan di-*parsing* disisi *client, maka* balasan yang dikirimkan oleh API *Server* dalam bentuk JSON[8].

RESTful *services* adalah teknologi yang bekerja menggunakan *resource* untuk membangun sistem terdistribusi, sehingga didesain dengan menekankan pada kesederhanaan, skalabilitas, serta kegunaan. *Web service* yang berorientasi pada *resource* dalam implementasinya banyak menggunakan REST, dengan menyediakan *resource-resource* sebagai layanan dan bukankumpulan dari aktifitas yang mengolah *resource*. RESTful Service digunakan karena pada aplikasi web *client* dan android *client* akan mengolah *resource-resource* tersebut[8].Gaya arsitektur REST didasarkan pada permintaan dan pesan tanggapan yang ditransfer antara klien dan server tanpa simpul yang berpartisipasi melacak keadaan sesi sebelumnya.

REST menggunakan kata benda dan kata kerja untuk keterbacaan. Sumber diidentifikasi dalam permintaan.

Representasi sumber daya yang dikirim ke klien tergantung pada permintaan dan bagaimana server mengirim data[10].

Ada empat prinsip utama teknologi yang mendasari Metode REST[10] yaitu:

1. *Resource identifier* melalui *Uniform Resource Identifier* (URI), yaitu sekumpulan sumber daya yang interaksi antar *client* diindetifikasi oleh REST *Web service*.
2. *Uniform interface*, menggunakan operasi PUT, GET, POST dan DELETE untuk memanipulasi manipulasi CRUD (*Create, Read, Update, Delete*).
3. *Self-descriptive messages*, disamping Metadata dapat digunakan, sumber daya informasi tidak terikat, juga berbagai format konten (HTML, XML, PDF, JPEG, *Plain Text* dan lainnya) dapat diakses..
4. *Stateful interactions*, dapat berinteraksi dengan suatu sumber daya bersifat *stateless*, dengan *request messages* tergantung jenis kontennya melalui *hyperlink*.

## III. METODOLOGI

Dalam melakukan penelitian, metode pengembangan sistem yang digunakan penulis adalah metode *waterfall*. Metode *waterfall* merupakan metode pengembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode (Implementasi), test (Pengujian), dan pemeliharaan[11]. Tetapi di dalam penelitian ini, penulis hanya sampai pada tahap pengujian.

Berikut tahapan-tahapan dalam *waterfall.:*

1. Analisa kebutuhan.

Pada tahap ini penulis melakukan analisa kebutuhan yaitu meliputi analisa data dan informasi, serta indentifikasi masalah. Penulis melakukan analisa melalui wawancara. Analisa dilakukan dengan tujuan menetapkan alternatif solusi dan spesifikasi kebutuhan sistem.

a. Analisis Data dan Informasi

Sebelum melakukan perancangan sistem baru, diperlukan in*form*asi yang jelas mengenai sistem lama yang biasa dilakukan oleh tukang, untuk memperoleh gambaran umum mengenai prosedur pemesanan jasa tukang. Hal ini dilakukan untuk mengetahui bagian mana dari prosedur tersebut yang perlu ditingkatkan atau diganti untuk menghasilkan sistem yang dapat berjalan pada *multiplatform*.

Teknik pengumpulan data dilakukan dengan melakukan wawancara terhadap beberapa orang dengan jasa tukang. Berikut ini adalah rincian hasil pengumpulan data dan in*form*asi yang dilakukan dengan teknik tersebut.

b. Indetifikasi Masalah

Berdasarkan hasil analisa yang telahdijabarkan di atas, ditemukan beberapa hal yang dianggap sebagai kendala dari sistem yaitu, antara lain:

- Media Promosi
- Kecepatan Pencarian
- Pemilihan Tukang
- Aplikasi Yang Dapat Membantu

- Pemilihan Platform Pengguna

c. Alternatif Solusi

Berdasarkan hasil identifikasi masalah yang telah dijelaskan di atas, maka solusi yang dapat dilakukan untuk mengatasi masing- masing permasalahan antara lain:

- Diperlukan sebuah sistem yang dapat digunakan sebagai media promosi yang mudah dan murah untuk tukang.
- Diperlukan sebuah sistem yang dapat melakukan pencarian berdasarkan lokasi penyedia jasa tukang.
- Diperlukan sebuah sistem yang dapat melakukan interaksi secara tak langsung dengan pemesan dan jasa tukang.
- Diperlukan pembuatan sistem dengan arsitektur REST API *Web Service* sehingga *platform* lain dapat dibangun.
- Mengimplementasikan API *Web Service* dalam *platform* web untuk memberikan user interface yang lebih baik pada web *client*.
- Mengimplementasikan API *Web Service* dalam *platform* android, khususnya aplikasi native yang dapat memberikan *user interface* serta memiliki kecepatan akses yang cepat.

d. Analisa Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk perkiraan kebutuhan sistem baru. Adapun analisis kebutuhan sistem meliputi:

- Perangkat Keras

Untuk menjalankan sistem ini diperlukan beberapa perangkat keras yaitu:

- Komputer dengan RAM minimal 2GB.
- Komputer dengan *Hard disk* minimal 500GB.
- Komputer dengan CPU minimal *Core* 2 Duo.
- Smartphone dengan sistem operasi android.
- Smartphone dengan RAM minimal 512 MB.
- Smartphone dengan penyimpanan *internal* minimal 2GB.

- Perangkat Lunak

Untuk menjalankan sistem ini diperlukan beberapa perangkat lunak yaitu:

- Sistem Operasi Linux/Windows sebagai *server* API *Web service* dan *server* aplikasi *Web Client*.
- Xampp sebagai *Web Server*.
- MySQL sebagai DBMS.
- *Web* Browser untuk mengakses aplikasi *Web Client*.
- Sistem operasi Android dengan versi minimal 4.1 (*JellyBean*).

- Pengguna

Agar perangkat lunak yang dibangun dapat berjalan dengan baik, maka diperlukan beberapa pengguna antara lain:

- *Admin*, yaitu pengguna yang menangani kebutuhan aplikasi *server control*. Kemampuan yang harus dimiliki *admin* yaitu dapat mengoperasikan komputer dan perangkat lainnya dengan baik, dapat bekerja pada sistem operasi Windows maupun Linux, dan dapat mengoprasikan aplikasi *server control* dengan baik dan benar.
- Tukang dan Pemesan, yaitu pengguna yang akan menggunakan aplikasi ini. Kemampuan yang harus dimiliki antara lain, dapat mengoperasikan komputer dengan baik, dan dapat mengoperasikan perangkat lunak *web browser* dengan baik bagi pengguna web *client*, dapat mengoperasikan smartphone dengan baik, bagi pengguna android *client*.

2. Desain sistem

Perancangan sistem untuk mendapatkan solusi pemecahan masalah adalah dengan menggunakan perangkat pemodelan sistem seperti relasi table. Desain sistem dibuat dengan menggunakan UML, meliputi: *Use Case Diagram, Class Diagram, Activity Digram, dan Sequence Diagram*. *Unfied Modeling Language* (UML) adalah notasi grafis untuk membantu pendeskripsian dan mendesain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek[8].

3. Implementasi

Pada tahap impelementasi, penulis membangun sistem baru dengan mengimplementasi hasil desain sistem yang telah di lakukan. Penulis membangun sistem baru berupa *API Webservice* dengan bahasa pemrograman php, dan *web client* dengan javascript serta *android client* dengan java native.

4. Uji coba

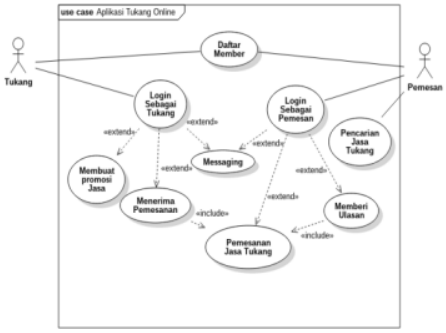
Pada tahap ini, penulis melakukan pengujian terhadap sistem baru yang telah dibangun. Proses pengujian dilakukan dengan metode *blackbox*. Metode ini melibatkan pengguna

dengan kuisioner sebagai alat ukur untuk menilai apakah sistem yang baru sudah dapat memenuhi kebutuhan pengguna atau belum.

## IV. HASIL DAN PEMBAHASAN

a. Use case diagram

*Use case diagram* digunakan untuk menggambarkan fungsionalitas aplikasi. *Use Case Diagram* terdiri dari *actor-actor* dan fungsi-fungsi apa saja yang bisa dilakukan di dalam sistem.

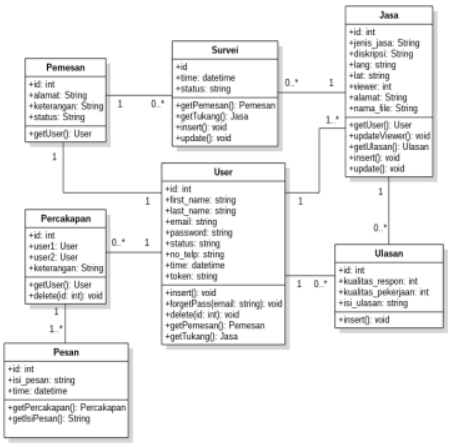



**Gambar 1.** Usecase diagram

*Use Case Diagram* di atas menggambarkan kegiatan apa saja yang bisa dilakukan oleh tukang dan pemesan di dalam sistem. tukang dan pemesan harus melakukan *login* untuk bisa mengakses sistem. Tukang dapat membuat promosi jasa, menerima pemesanan serta melakukan *messaging* dengan pemesan. Semua action ini dapat dilakukan setelah melakukan action *login*. pemesan dapat melakukan pencarian jasa tukang, *messaging*, melakukan pemesanan jasa tukang, serta memberi ulasan jika telah melakukan pemesanan.

b. Class Diagram

Class Diagram digunakan untuk menampilkan kelas-kelas yang jika diinisiasikan akan menjadi objek di dalam sistem.



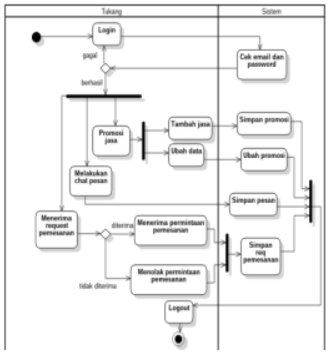

**Gambar 2.** Class diagram

Pada *Class* diagram di atas, Class Pemesan dan class Jasa meng-*extend* class *User*, sehingga ini menjelaskan *User* dapat menjadi salah satu Pemesan atau jasa Tukang. Class Survei merupakan objek yang menggambarkan antara transaksi pada Pemesan dan Tukang. Class Survei meng-*extend* class Pemesan dan Jasa. Class ulasan meng-*extend* dari class jasa, sehingga dalam satu objek jasa tukang memiliki banyak ulasan. Class yang terakhir adalah class Percakapan yang meng-*extend* dari class user yang menjelaskan pengirim dan penerima, serta meng-*extend* class Pesan yang menjelaskan isipesan percakapan antara *User*.

c. Activity

*activity diagram* digunakan untuk menggambarkan urutan aktivitas yang terjadi di dalam aplikasi**.**

*Activity Diagram* untuk tukang




**Gambar 3.** Activity diagram pada tukang

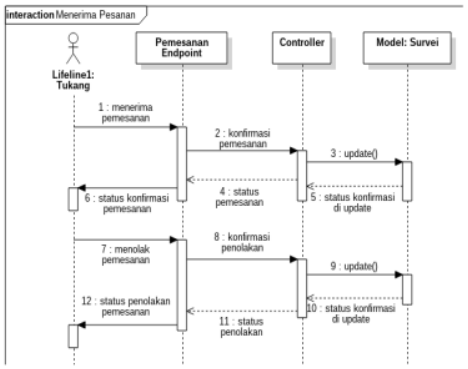
*Activity Diagram* untuk Tukang, menggambarkan urutan aktivitas yang

dilakukan tukang di dalam sistem. Untuk mengakses sistem, tukang harus melakukan *login* terlebih dahulu. Beberapa aktifitas yang bisa dilakukan tukang di dalam sistem adalah melakukan melakukan promosi jasa, menerima pemesanan jasa, serta berkirim pesan.

d. Sequence Diagram
*Sequence diagram* digunakan untuk menggambarkan interaksi antara sejumlah objek dalam urutan waktu.

*Sequence Diagram* yang menggambarkan proses pemesanan jasa tukang.

- *Sequence diagram* proses menerima pemesanan jasa




**Gambar 5.** Sequence diagram proses menerima pemesanan jasa

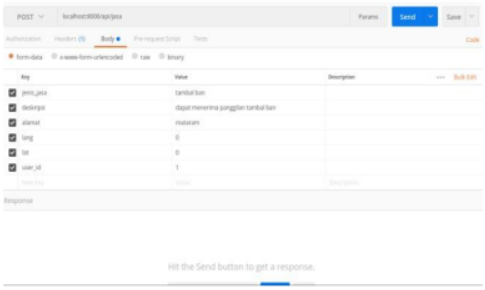
*Sequence Diagram* yang menggambarkan proses penerimaan pemesanan jasa tukang.

Tampilan Antarmuka

Bahasa pemrograman yang digunakan penulis untuk membangun *API Web service* ini adalah PHP dan framework Laravel. Setelah *API* Web *service* selesai dibuat barulah dilakukan pembuatan tampilan antarmuka aplikasi berbasis *web* menggunakan bahasa pemrograman *Javascript* dengan framework *VueJS* sertaaplikasi berbasis android dengan menggunakan java *native*. Aplikasi berbasis *web* dan android inilah sebagai tampilan antarmuka yang mengakses data-data dari *endpoint API Webservice*.

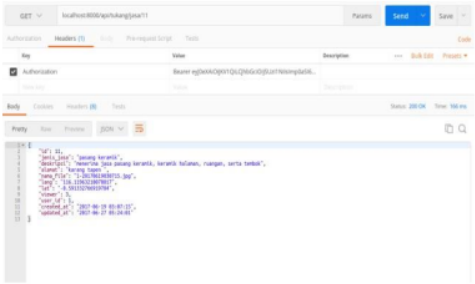
Berikut merupakan hasil uji coba pada URI /api/find /{id} dengan method GET. Pada URI ini menghasilkan respon berupa data json jasa dengan parameter data pencarian.

a) URI /api/jasa (POST)



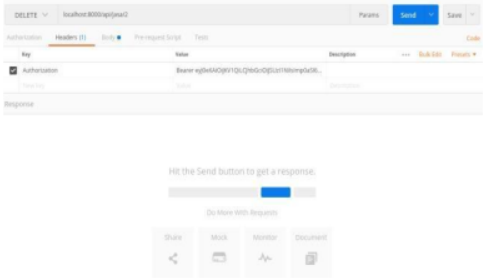
**Gambar 6.** URI /api /jasa (POST) Berikut merupakan hasil uji coba pada URI /api/jasa dengan method POST. Pada URI ini membutuhkan form-data berupa field-field yang akan di simpan di database serta menghasilkan respon json berupa jasa yang baru diposting.

b) URI /api/tukang/jasa/{id} (GET)



**Gambar 7.** URI /api /tukang/jasa/{id} (GET) Berikut merupakan hasil uji coba pada URI /api/tukang/jasa/{id} dengan method GET. Pada URI ini menghasilkan respon berupa data json jasa dengan parameter id jasa.

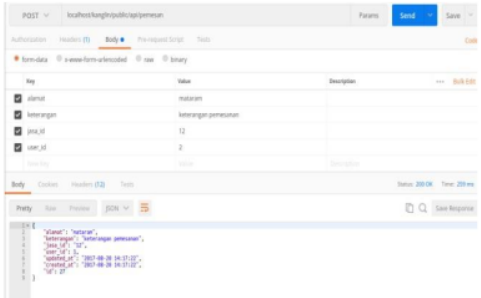
c) URI /api/jasa/{id} (DELETE)



**Gambar 8.** URI /api/jasa/{id} (DELETE)

Berikut merupakan hasil uji coba pada URI /api/jasa/{id} dengan method DELETE. Pada URI ini digunakan untuk menghapus data jasa pada database menggunakan id jasa.

d) URI /api/pemesan (POST)

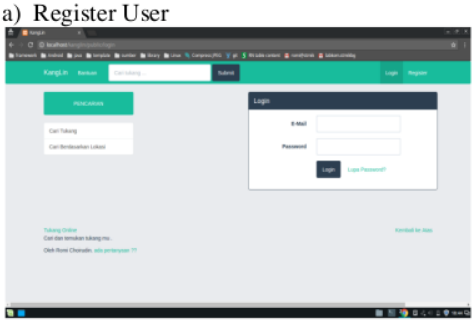


**Gambar 10.** URI /api/pemesan (POST)

Berikut merupakan hasil uji coba pada URI /api/pemesan dengan method POST. Pada URI ini membutuhkan form-data berupa field-field yang akan di simpan di database serta menghasilkan respon json berupa data pemesanan yang baru diposting.

Berikut adalah tampilan aplikasi yang dibangun:

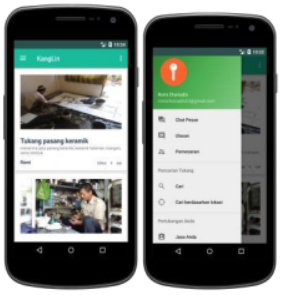
a) Register User



**Gambar 12.** Halaman register

Halaman Register digunakan untuk melakukan pendaftaran user baru ke dalam aplikasi, padafitur register aplikasi hanya dibangun di web.

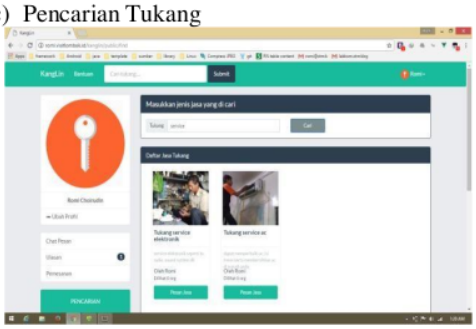
b) Home



**Gambar 13.** Halaman Home

Halaman home merupakan halaman yang muncul setelah melakukan proses login. Pada halaman ini secara otomatis sistem akan menampilkan beberapa jasa secara *random*. Jika *user* ingin melakukan beberapa fungsi pada sistem maka user akan menggunakan navigasi yang telah dibuat pada halaman awal

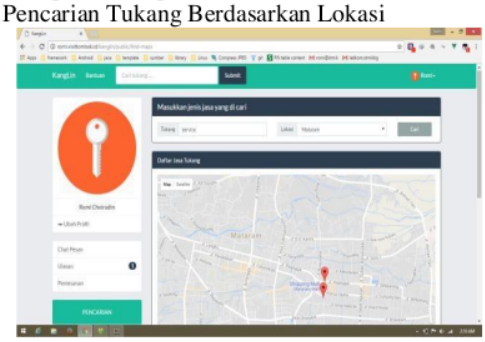
c) Pencarian Tukang



**Gambar 14.** Halaman Pencarian Tukang

Halaman pencarian digunakan untuk melakukan pencarian jasa tukang yang akan dicari oleh Pemesan. Pada halaman ini pemesan dapat memasukkan kata kunci tertentu guna mempermudah pencarian.

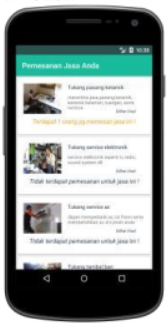
d) Pencarian Tukang Berdasarkan Lokasi



**Gambar 15.** Halaman Pencarian Tukang Berdasarkan Lokasi

Halaman pencarian tukang berdasarkan lokasi digunakan untuk melakukan pencarian jasa tukang yang akan dicari oleh Pemesan. Pada halaman ini pemesan dapat memasukkan kata kunci serta lokasi tertentu guna mempermudah pencarian.

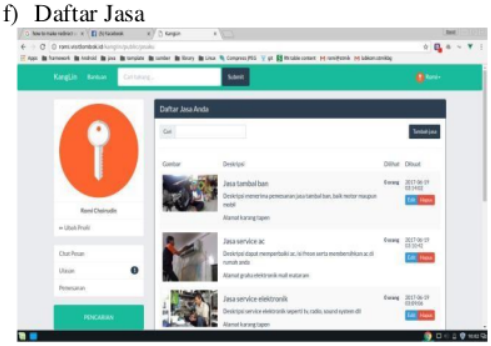
e) Daftar Pemesan Jasa



**Gambar 16.** Halaman Pemesanan Jasa

Halaman berfungsi menampilkan pemesanan yang telah ada yang di *request* oleh pemesan. Setelah ditampilkan tukang dapat menerima atau menolak pemesan yang di request oleh pemesan.

f) Daftar Jasa



**Gambar 18.** Halaman Daftar Jasa Anda

Halaman ini menampilkan daftar jasa yang telah diupload oleh user tukang.

g) Konfirmasi Pemesanan

**Gambar 19.** Halaman Konfirmasi Pemesanan

Halaman konfirmasi pemesan merupakan halaman yang berfungsi melakukan konfirmasi pemesan terhadap *request* pemesanan.

3.4 Uji Coba

Uji coba dilakukan dengan menguji aplikasi kepada 20 responden yang terdiri dari 5 orang tukang dan 15 orang pemesan. Berdasarkan hasil pengujian kuesioner, 45,7% mengatakan setuju dan 51% menyatakan sangat setuju bahwa aplikasi ini membantu dalam melakukan promosijasa serta pencarian jasa tukang.

**3.4.1. Uji coba dengan responden Tukang**

Hasil uji coba dengan responden dapat dilihat pada tabel 4.1. berikut :

Tabel 4.1 Hasil Pengujian Kuesioner Tukang

| No | Pertanyaan | Pilihan Jawaban | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1. | Apakah aplikasi ini memiliki tampilan yang menarik ? | 0 | 0 | 0 | 2 | 3 |
| 2. | Apakah aplikasi ini mempermudah anda dalam mempromosikan jasa tukang ? | 0 | 0 | 0 | 0 | 5 |
| 3. | Apakah aplikasi ini dapat menjadi media promosi pilihan yang | 0 | 0 | 1 | 4 | 0 |
| 4. | Apakah penggunaan aplikasi ini mudah ? | 0 | 0 | 0 | 4 | 1 |
| 5. | Apakah dengan penggunaan aplikasi ini promosi jasa anda menjadi maksimal ? | 0 | 0 | 0 | 3 | 2 |
| 6. | Apakah fitur di aplikasi ini sudah dapat membantu anda dalam melakukan | 0 | 0 | 0 | 2 | 3 |
| 7. | Apakah aplikasi ini dapat terintegrasi *multiplatform* (Web | 0 | 0 | 0 | 1 | 4 |
| **Jumlah** | | **0** | **0** | **1** | **16** | **18** |
| **Persentase (%)** | | **0** | **0** | **2.** | **45.7** | **51.4** |

Keterangan
1 = Sangat Tidak Setuju
2 = Tidak Setuju
3 = Moderat
4 = Setuju
5 = Sangat Setuju

Perhitungan untuk masing-masing jawaban sebagai barikut:

1. Sangat tidak setuju $= \frac{\Sigma}{(q*p)} * 100$

$= 0/(7*5)*100$
$= 0/(35)*100$

2. Tidak setuju $= \frac{\Sigma}{(q*p)} * 100$

$= 0/(7*5)*100$
$= 0/(35)*100$
$= 0$

3. Moderat $= \frac{\Sigma}{(q*p)} * 100$

$= 1/(7*5)*100$

= 1/(35)*100
= 2.9

4. Setuju = $\frac{\Sigma}{(q*p)} * 100$

= 16/(7*5)*100
= 16/(35)*100
= 45.7

5. Sangat Setuju = $\frac{\Sigma}{(q*p)} * 100$

= 18/(7*5)*100
= 18/(35)*100
= 51.4

Berdasarkan hasil dari pengujian tabulasi kuesioner diatas maka didapatkan hasil perhitungan persentase sebanyak 0% responden menjawab sangat tidak setuju (1), sebanyak0% responden menjawab tidak setuju (2),sebanyak 2.9% responden menjawab moderat (3), sebanyak 45.7% responden menjawab setuju
(4), dan sebanyak 51.4% responden menjawab sangat setuju (5). Sehingga dapat diketahui nilai tertinggi dari jawaban responden di atas adalah sangat setuju (5) sebesar 51.4% sehingga dapat disimpulkan bahwa aplikasi ini memenuhi kebutuhan pengguna.

### 4.4.2. Uji coba dengan responden Pemesan

Adapun hasil pengujian dengan kuesioner kepada pemesan adalah sebagai berikut:

Tabel 4.2 Hasil Pengujian pemesan

| No | Pertanyaan | Pilihan Jawaban | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1. | Apakah aplikasi ini memiliki tampilan yang | 1 | 0 | 0 | 10 | 4 |
| 2. | Apakah aplikasi ini mempermudah anda dalam mempromosikan | 0 | 0 | 0 | 6 | 9 |
| 3. | Apakah aplikasi ini dapat menjadi media promosi | 0 | 0 | 3 | 6 | 6 |
| 4. | Apakah penggunaan | 0 | 0 | 2 | 7 | 6 |
| 5. | Apakah dengan penggunaan aplikasi ini promosi jasa anda menjadi | 0 | 0 | 3 | 6 | 6 |
| 6. | Apakah fitur di aplikasi ini sudah dapat membantu anda dalam | 0 | 0 | 0 | 7 | 8 |
| 7. | Apakah aplikasi ini dapat terintegrasi *multiplatform* (Web | 0 | 0 | 1 | 4 | 10 |
| **Jumlah** | | **1** | **0** | **9** | **4** | **4** |
| **Persentase (%)** | | **0.9** | **0** | **8.6** | **43.** | **46.** |

Keterangan
1 = Sangat Tidak Setuju
2 = Tidak Setuju
3 = Moderat
4 = Setuju
5 = Sangat Setuju
Perhitungan untuk masing-masing jawaban sebagai barikut:

1. Sangat tidak setuju = $\frac{\Sigma}{(q*p)} * 100$

= 1/(7*15)*100
= 1/(105)*100
= 0.9

2. Tidak setuju = $\frac{\Sigma}{(q*p)} * 100$

= 0/(7*15)*100
= 0/(105)*100
= 0

6. Moderat = $\frac{\Sigma}{(q*p)} * 100$

= 9/(7*5)*100
= 9/(105)*100
= 8.6

7. Setuju = $\frac{\Sigma}{(q*p)} * 100$

= 46/(7*5)*100
= 46/(105)*100
= 43.8

8. Sangat Setuju = $\frac{\Sigma}{(q*p)} * 100$

= 49/(7*5)*100
= 49/(105)*100
= 46.7

Berdasarkan hasil dari pengujian tabulasi kuesioner diatas maka didapatkan hasil perhitungan persentase sebanyak 0.9% responden menjawab sangat tidak setuju (1), sebanyak 0%

responden menjawab tidak setuju (2), sebanyak 8.6% responden menjawab moderat (3), sebanyak 43.8% responden menjawab setuju (4), dan sebanyak 46.7% responden menjawab sangat setuju (5). Sehingga dapat diketahui nilai tertinggi dari jawaban responden di atas adalah sangat setuju (5) sebesar 46.7% sehingga dapat disimpulkan bahwa aplikasi ini memenuhi kebutuhan pengguna.

## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Dari pembahasan dapat diambil kesimpulan bahwa *API WebService* dengan arsitektur REST yang telah dibuat berhasil mencapai tujuan dan sasaran dimana dapat digunakan untuk membuat aplikasi *multiplatform* dengan data yang terintegrasi.

Aplikasi yang dibangun telah diuji kepada beberapa tukang. Serta dari hasil pengujian responden tukang memiliki jawaban sangat setuju sebesar 51.4%, dikarenakan 51.4% merupakan hasil tertinggi pada respon penilaian jawaban maka dapat disimpulkan aplikasi telah memenuhi kebutuhan tukang.

Aplikasi yang dibangun telah diuji kepada beberapa pemesan jasa tukang. Serta dari hasil pengujian responden tukang memiliki jawaban sangat setuju sebesar 46.7%, dikarenakan ini merupakan hasil tertinggi pada respon penilaian jawaban maka dapat disimpulkan aplikasi telah memenuhi kebutuhan

### 5.2 Saran

Sistem memerlukan konfigurasi *web server* secara khusus guna mempercepat akses data. Sistem API yang dapat dikembangkan dengan arsitektur web *socket* untuk membuat sistem dengan *realtime database*.

Menambahkan fitur *realtime location* khususnya untuk perangkat *platform mobile* agar dapat melakukan pencarian lebih akurat. Menambahkan fitur *realtime call* agar seorang pemesan dan tukang dapat saling berinteraksi melalui suara secara *online* melalui aplikasi. Sistem dapat dikembangkan dengan menambahkan teknologi sms *gateway* guna mempermudah notifikasi jika terdapat pemesanan baru.

## REFERENSI


[1]. Haag, & Keen.. *Information Technology. Tomorrow's Advantage Today*. Hammond: Mcgraw-Hill College, 1996

[2]. *KBBI*.. Retrieved January 11, 2017, from http://kbbi.web.id/tukang: http://kbbi.web.id/tukang. 2016

[3]. K. Peranginangin,. *Aplikasi Web dengan PHP dan MySQL*. Yogyakarta: Andi.. 2006

[4]. G. C. Hillar, "Django RESTful Web Services," 2018.

[5] O. and/or its Affiliates, *Building RESTful Web Services with JAX-RS*. 2013.

[6] K. Rizki and A. Adil, "Implementasi Google Maps API Berbasis Android untuk Lokasi Fasilitas Umum di Kabupaten Sumbawa," *J. MATRIK*, vol. 17, no. 2, pp. 34–44, 2018.

[7] I. Aece, *ASP . NET Web API*. .

[8] N. E. T. Web *et al*., "Table of Contents."

[9] U. Cei and P. Lucidi, *Alfresco 3 Web Services*. 2010.

[10] M. E. Indrawan and A. Adil, "Implementasi Restful Web Service One Chip Multi-Client Untuk Mengoptimalkan Penjualan Pulsa All Operator," *J. Matrik*, vol. 15, no. 2, p. 11, 2018.

*[11]* I. M.Siregar, , & J. Purba, *Membongkar Teknologi Pemrograman Web Service*. Yogyakarta: Gava Media. 2012

[12]. D.Tidwell,. *Programming Web Services with SOAP*. Sebastopol: O'Reilly Media. 2001

[13]. B. Raharjo,. *Pemrograman Web dengan PHP + Oracle*. Bandung: Informatika Bandung. 2011

[14]. M. Fowler,. *UML Distilled Edisi 3 Panduan Singkat Bahasa Pemodelan Objek Standar*. Yogyakarta: Penerbit ANDI, 2004

# IMPLEMENTASI REST API WEB SERVICE DALAM MEMBANGUN APLIKASI MULTIPLATFORM UNTUK USAHA JASA

ORIGINALITY REPORT

0%

SIMILARITY INDEX

PRIMARY SOURCES

| | | | |
|---|---|---|---|
| EXCLUDE QUOTES | ON | EXCLUDE SOURCES | OFF |
| EXCLUDE BIBLIOGRAPHY | OFF | EXCLUDE MATCHES | OFF |