

# Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android

Amin Rulloh<sup>1</sup>, Dewi Erla Mahmudah<sup>2</sup>, Herman Kabetta<sup>3</sup>

<sup>1,2,3</sup>STMIK Widya Utama

<sup>1</sup>aminrulloh@gmail.com, <sup>2</sup>dewierla@swu.ac.id, <sup>3</sup>hermanka.beta@gmail.com

**Abstract**— Application trends that require developers to create apps that can make content up-to-date whenever accurately makes developers search for methods that can make that happen. With the RESTful web service technology, applications now can connect to the global network. With the notation format support from JSON, result from data will be faster to load in the application and allows developers to recognize the resulting objects. First, creating a database to store content that supports apps. Then create the controller that is used for each API call on each required content. After that the app on the android device is made with the support of several libraries, namely Volley for calling API and HTML-TextView to convert content data in HTML format into TextView format on android device. This research has resulted that REST can be applied in this Guidance Application.

**Keywords**— Android, REST API, Volley, Guidance Application

## 1. PENDAHULUAN

Sebuah *smartphone* sekarang menjadi hal yang tak pernah lepas dari kehidupan masyarakat modern saat ini. Ketergantungan masyarakat pada sebuah *smartphone* tak lepas dari kemudahan-kemudahan yang ditawarkan dari perkembangan sebuah perangkat tersebut. Seiring dengan perkembangan perangkat *smartphone* tersebut, dibarengi pula perkembangan aplikasi-aplikasi yang menunjang penggunaan *smartphone* tersebut dengan fitur-fitur yang ditawarkan. Tren dari sebuah teknologi yang mendukung aplikasi tersebut kini mulai meningkat dalam pengembangannya, terutama pada *smartphone* dengan sistem operasi android, dari sebuah aplikasi *standalone*, kini berubah menjadi aplikasi yang dapat terhubung dengan aplikasi lain melalui jaringan global [1]. Teknologi tersebut kini dikenal dengan istilah *Web Service APIs*.

Salah satu jenis dari *Web Service* adalah REST atau RESTful (*Representational State Transfer*). REST sendiri memungkinkan *system request* dapat mengakses dan memanipulasi teks yang direpresentasikan dari sebuah *Web Service*. *Web Service API* yang menggunakan REST disebut dengan RESTful API. Tidak seperti jenis *Web Service* lainnya, RESTful API tidak memiliki standar yang resmi untuk notasinya dikarenakan REST merupakan sebuah arsitektur [2]. Dengan berbagai implementasi, REST menemukan notasi yang biasa digunakan, seperti HTTP, URI, JSON, dan XML.

Dalam penggunaannya, REST API terbukti lebih cepat dalam transfer data daripada metode lain yang serupa, dalam hal ini SOAP (*Symples Object Access Protocol*) [3]. Tidak seperti SOAP, tidak ada notasi standar resmi untuk RESTful API. Hal ini dikarenakan REST merupakan arsitektur, sedangkan SOAP adalah protokol.

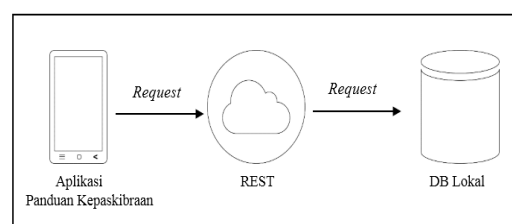
Dalam perjalanannya, pengembangan untuk mengembangkan sebuah RESTful API menemui kendala dalam penamaan sebuah objek. Notasi objek-objek yang ada haruslah memudahkan pengembang dalam menamainya namun juga harus dikenali oleh komputer. Alasan inilah yang membuat pengembang menggunakan JSON (*JavaScript Object Notation*) sebagai notasi untuk REST *Web Service* dalam aplikasi miliknya. Penggunaan JSON untuk menyokong pembuatan aplikasi mobile berbasis REST API juga dianggap lebih baik performanya untuk pengguna mobile [4].

Oleh karena itu, peneliti tertarik untuk meneliti implementasi REST API pada sebuah *prototype* aplikasi panduan kepaskibraaan. Penelitian serupa juga pernah dilakukan oleh Hanindya [5] dengan menerapkan REST API pada aplikasi panduan turis di daerah Karimunjawa.

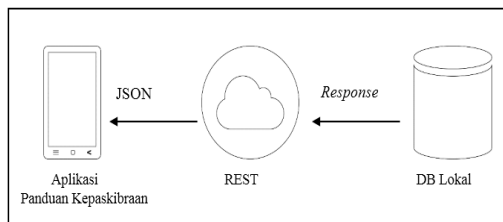
## 2. METODE PENELITIAN

RESTful *Web Service* dibangun menggunakan bahasa pemrograman PHP dan basis data MySQL untuk menyimpan konten-konten penunjang aplikasi dan disimulasikan menggunakan server lokal.

RESTful API memanfaatkan pemanggilan dan respon setiap konten sehingga memudahkan ketika pemanggilan dan pergantian konten yang akan ditampilkan pada aplikasi mobile. Skema untuk cara kerja REST dalam Aplikasi Panduan Kepaskibraan ditampilkan dalam gambar 1 dan gambar 2 berikut.



Gambar 1 Proses *request* Aplikasi dan REST

Gambar 2 Proses *response* Aplikasi dan REST

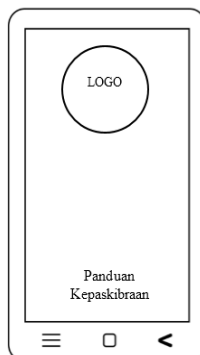
REST dibagi menjadi dua pengelolaan, yaitu mengelola akses aman pada web dan mengelola akses ke aplikasi pada android.

Setiap *request* dan *response* yang terdapat pada objek JSON disesuaikan dengan materi apa yang akan ditampilkan pada aplikasi dan web, seperti contoh pada tabel 1 berikut.

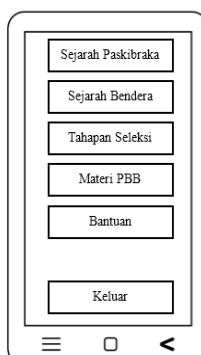
Tabel 1 Contoh operasi API

<b>GET</b> /materi	Mengambil list materi
<b>GET atau POST</b> /materi/PBB	Materi PBB
<b>GET</b> /materi/PBB/langkah-tegap	Materi PBB untuk langkah tegap

Pada gambar 3 dapat dilihat bahwa tampilan awal saat membuka aplikasi (*splash screen*) hanya menampilkan logo aplikasi dan judul aplikasi pada bagian bawah.

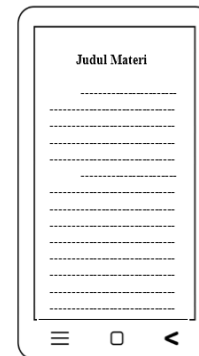
Gambar 3 *Splash Screen*

Setelah *splash screen* menghilang, kemudian ditampilkan menu dengan isi : sejarah paskibra, sejarah bendera, tahapan seleksi menjadi paskibra, materi PBB yang digunakan dalam pengibaran bendera, menu bantuan dan keluar aplikasi seperti yang terdapat pada gambar 4.



Gambar 4 Tampilan Menu

Tampilan saat menampilkan isi materi dari menu yang di pilih sebelumnya akan tampak seperti pada gambar 5 yang berisi judul dan isi materi.



Gambar 5 Tampilan Isi

### 3. HASIL DAN PEMBAHASAN

#### 3.1 REST API Method

Hal utama dalam penggunaan REST *web service* adalah pembuatan *controller* yang akan digunakan untuk pemanggilan *web service*.

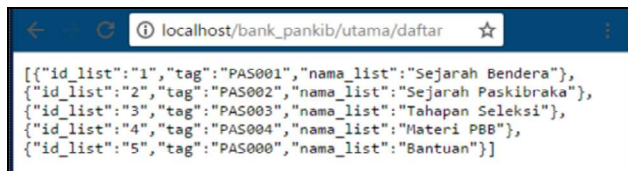
Dalam penelitian ini, *web service* yang digunakan adalah dari CodeIgniter versi 3.1.2. *Controller* berperan untuk menentukan informasi yang dikirim ketika pengguna mengakses *web service* yang dibuat. Dengan adanya arsitektur REST, maka *method-method* yang digunakan tentu yang didukung oleh protokol HTTP, antara lain : GET, POST, PUT, dan DELETE.

```
// show data daftar
function daftar_get() {
    $id_list = $this->get('id_list');
    if ($id_list == '') {
        $daftar = $this->db->get('tb_list')->result();
    } else {
        $this->db->where('id_list', $tag);
        $daftar = $this->db->get('tb_list')->result();
    }
    $this->response($daftar, 200);
}
```

Gambar 6 Contoh *method* GET pada *controller*

Pada gambar 6 diatas, dapat dilihat penggunaan *method* GET untuk menampilkan daftar isi yang diakses menggunakan URL tertentu. Contoh URL yang digunakan untuk mengakses bagian tersebut adalah [http://localhost/bank\\_pankib/utama/daftar](http://localhost/bank_pankib/utama/daftar). Bagian awal merupakan nama domain, dikarenakan penelitian ini menggunakan *local storage*, maka dapat ditulis "localhost" atau bisa juga alamat IP dimana *web service* tersebut berada.

Pada bagian berikutnya adalah direktori, kemudian bagian berikutnya merupakan *controller* yang akan diakses. Dan pada bagian terakhir merupakan informasi apa yang akan dimunculkan sesuai dengan yang terdapat dalam *controller*. Berikut adalah hasil pemanggilan dari URL diatas.



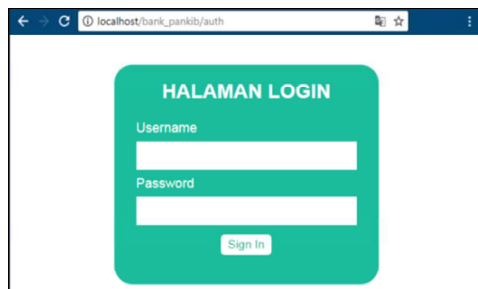
Gambar 7 Contoh hasil akses API

Pada gambar 7 diatas merupakan hasil dari mengakses *web service* pada URL yang telah ditentukan sebelumnya. Hasil *output*-nya dalam format JSON array yang nantinya akan diolah lagi agar memudahkan dalam pengelolaan data-datanya.

### 3.2 REST API Backend

Pada penelitian ini dibangun sebuah aplikasi backend berbasis web untuk memudahkan dalam mengelola konten REST API yang nantinya akan diakses oleh aplikasi *client* pada perangkat android.

Untuk dapat mengelola seluruh konten pada REST API Backend, admin diwajibkan untuk login terlebih dahulu seperti ditunjukkan pada gambar 8.



Gambar 8 Tampilan login

Halaman daftar isi akan muncul setelah admin berhasil login menggunakan username dan password yang benar. Halaman ini menampilkan daftar isi atau menu dari Aplikasi Panduan Kepaskibraan.



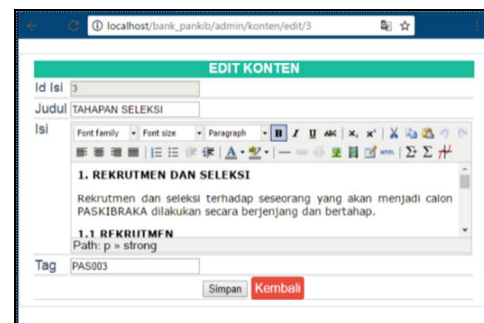
Gambar 9 Tampilan daftar isi

Selain halaman daftar isi, terdapat juga halaman konten yang digunakan untuk mengelola konten berdasarkan menu / daftar isi dari Aplikasi Panduan Kepaskibraan.



Gambar 10 Tampilan web konten

Tampilan untuk mengedit data konten ditunjukkan pada gambar 11.



Gambar 11 Tampilan web edit konten

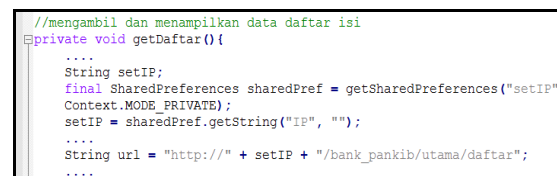
### 3.3 Implementasi REST API

Pada gambar 13 berikut merupakan tampilan awal atau splashscreen dari Aplikasi Panduan Kepaskibraan yang berisi logo Paskibraka dan nama aplikasi.



Gambar 12 Tampilan SplashScreen

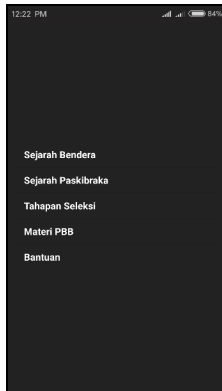
Pada gambar 14 adalah baris kode untuk memanggil API untuk data menu/daftar isi dan untuk men-decode-kan data yang semula JSON menjadi data string yang diletakkan pada masing-masing text pada ListView. Untuk men-decode-kan data JSON menjadi data string yang dapat ditampilkan pada komponen-komponen aplikasi android digunakan *library* Volley.



Gambar 14 Coding untuk data daftar isi

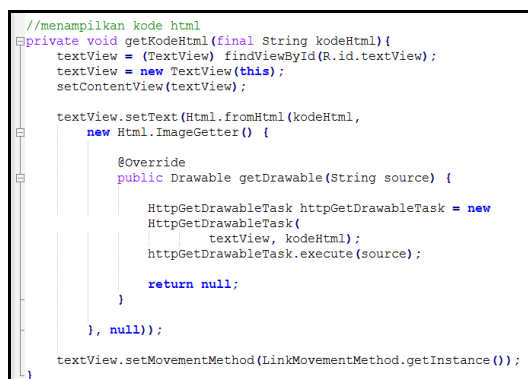
Seperti yang terlihat pada gambar 14 diatas, pada baris awal kode digunakan untuk mendapatkan data IP yang telah tersimpan pada *cache* aplikasi android menggunakan *SharedPreferences*. Alasan digunakannya IP yang dinamis dikarenakan pada penelitian ini menggunakan simulasi *client-server* lokal yang IP-nya diset menurut IP DHCP dari router yang digunakan. Kemudian pada baris selanjutnya merupakan alamat API untuk data daftar isi yang disimpan pada variabel "url".

Berikut adalah hasil dari tampilan daftar menu yang *berisi* : Sejarah Bendera, Sejarah Paskibraka, Tahapan Seleksi, Materi PBB, dan Bantuan aplikasi.



Gambar 15 Tampilan Menu

Cara yang sama dilakukan untuk menampilkan konten, *komponen* yang digunakan adalah TextView. Gambar 16 adalah baris kode yang digunakan untuk *me-request* konten pada server REST.



Gambar 16 Kode untuk request konten

Pada gambar 16 berikut merupakan tampilan dari isi / konten materi dari menu yang dipilih, contohnya tentang sejarah bendera. Pada bagian ini, terdapat isi materi beserta gambar penunjang materi.



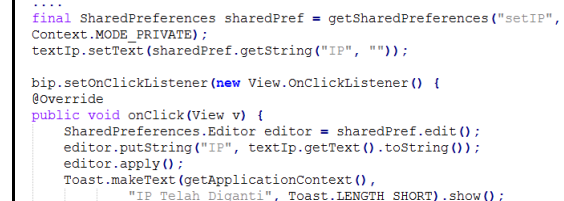
Gambar 16 Tampilan Materi

Halaman 404 akan muncul ketika aplikasi pada perangkat android tidak dapat mengakses server dikarenakan perbedaan IP yang digunakan.



Gambar 17 Tampilan 404

Terlihat pada gambar 17, aplikasi menyediakan setelan tambahan untuk mengganti IP server REST yang nantinya disimpan ke dalam *cache* aplikasi, contohnya 192.168.100.1. Untuk menyimpan IP yang baru, tekan tombol Ganti IP dan akan muncul *floating message* "IP telah diganti". Berikut adalah baris kodenya.



Gambar 18 Coding untuk menyimpan cache IP

Sebenarnya terdapat berbagai macam metode untuk menyimpan *cache* ke dalam aplikasi, namun karena kebutuhan penyimpanan yang sederhana dan tidak menggunakan banyak penyimpanan maka digunakanlah *SharedPreferences*.

#### 4. KESIMPULAN DAN SARAN

Dengan digunakannya REST API dalam pembuatan Aplikasi Panduan Kepaskibraan dan dengan adanya dukungan notasi JSON maka memudahkan dalam proses penerapan teknologi pada aplikasi yang dibangun.

Diawali dengan membangun *web service*, REST yang terbukti handal. Membuat beberapa *controller* yang diperlukan untuk akses aplikasi pada perangkat android maupun pengelolaan konten pada web admin. Kemudian proses *coding* untuk aplikasi android dimulai dengan pemanggilan API menggunakan *library Volley* dari *web service* yang awalnya berupa data dengan format JSON kemudian dikonversi ke dalam format TextView.

Dalam akses konten materi kepaskibraan, terjadi konversi data dengan format JSON dan kode HTML kedalam format TextView dengan menggunakan *library HTML-TextView* yang juga memuat gambar yang mendukung materi yang akan ditampilkan.

Adapun saran dari peneliti guna meningkatkan hasil pada penelitian selanjutnya yaitu :

1. Penggunaan REST API untuk sinkronisasi data pada *local storage* pada perangkat android dan data pada server.
2. Penggunaan metode lain agar memungkinkan perangkat dapat memuat video.

### DAFTAR PUSTAKA

- [1] Deitel, P., Deitel, H., dan Deitel, A. 2014. *Android How to Program*. Prentice Hall, New Jersey.
- [2] Doglio, F. 2014. *Pro REST API Development with Node.js*. Apress, Canelones.
- [3] Dudhe, A., dan Sharekar, S. S. 2014. Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment. *Second National Conference on Recent Trends in Information Security*. Nagpur, India, Januari 2014. 1-4.
- [4] Raval, R., dan Gonsai, A. 2015. Performance Analysis and Design of a Mobile Web Services on Cloud Servers. *International Journal of Emerging Technology and Advanced Engineering*. 5(9): 104-113.
- [5] Hanindya, G. M., dan Somya, R. 2014. Perancangan dan Implementasi Aplikasi Mobile Tourist Guide pada Platform Android. *Jurnal UKSW*. 1-18.
- [6] JSON. Pengenalan JSON. <http://www.json.org/json-id.html>. Diakses pada 28 Oktober 2016.
- [7] Kadir, A. 2014. *Pengenalan Sistem Informasi Edisi Revisi*. Penerbit Andi, Yogyakarta.
- [8] Lagace, N. 2016. Improving E-Book Management, Discovery, and Access: Two New NISO Initiatives Get Working. *The Serial Librarian*. 1-17.
- [9] Peraturan Menteri Pemuda dan Olahraga Republik Indonesia Nomor 0065 Tahun 2015 Tentang Penyelenggaraan Kegiatan Pasukan Pengibar Bendera Pusaka(PASKIBRAKA). <http://www.kemenpora.go.id/index/perundangan>. Diakses pada 24 Oktober 2016.
- [10] Peraturan Panglima Tentara Nasional Indonesia Nomor 46 Tahun 2014 Tentang Peraturan Baris Berbaris Tentara Nasional Indonesia.
- [11] PHP. PHP: PHP Manual. <http://php.net/manual/en/>. Diakses pada 25 Juli 2017.
- [12] Undang-Undang Republik Indonesia Nomor 24 Tahun 2009 Tentang Bendera, Bahasa, Dan Lambang Negara, Serta Lagu Kebangsaan. [http://badanbahasa.kemdikbud.go.id/lamanbahasa/site/s/default/files/UU\\_2009\\_24.pdf](http://badanbahasa.kemdikbud.go.id/lamanbahasa/site/s/default/files/UU_2009_24.pdf). Diakses pada 24 Oktober 2016.