

ID: 04

Perancangan Sistem Informasi Pelaporan Peralatan Meteorologi Berbasis Android

Design of Meteorology Instrumentation Reporting Information System Based on Android

Sugiarto^{1*}, Naufal Ananda², Maulana Putra³

^{1,3}Pusat Instrumentasi, Kalibrasi dan Rekayasa, Badan Meteorologi Klimatologi dan Geofisika
Jl. Angkasa I No.2 Kemayoran, Jakarta Pusat, DKI Jakarta, (021) 4246321

²Balai Besar Meteorologi Klimatologi dan Geofisika Wilayah II - Tangerang Selatan
Jl. H. Abdulgani No.05, Cempaka Putih, Ciputat Timur, Tangerang Selatan, Banten, (021) 7402739

sugiarto@bmet.go.id^{1*}, naufal.ananda17@gmail.com², maulanaputrassi@gmail.com

Abstrak – Badan Meteorologi Klimatologi dan Geofisika mengoperasikan sejumlah peralatan elektronik yang digunakan untuk memantau, mengukur dan mencatat perubahan parameter cuaca. Data-data yang tercatat pada peralatan selanjutnya akan dianalisis dan hasilnya akan diinformasikan kepada masyarakat. Jumlah peralatan meteorologi setiap tahunnya bertambah secara kuantitas maupun jenisnya sehingga memerlukan sistem pemantauan kondisi peralatan secara real-time. Saat ini pelaporan kondisi peralatan masih dilakukan secara manual sehingga pada penelitian ini akan dirancang perancangan sistem informasi pelaporan peralatan meteorologi berbasis Android. Sistem terdiri dari front end berupa interface berbasis aplikasi Android, front-end tambahan berupa interface web untuk rekapitulasi, dan back-end berupa web service. Penelitian ini memanfaatkan teknologi web service sebagai penghubung antara beberapa platform dalam mengolah data dari sebuah server database. Penggunaan web service merupakan salah contoh adaptasi terhadap akses data secara terbuka, dimana data tersebut dapat digunakan multi-platform. Hasil penelitian menunjukkan sistem yang dirancang beroperasi dengan baik dimana dari segi fungsionalitas penerapan arsitektur RESTful web service yang diimplementasikan pada server mampu menyediakan resource dengan representasi format data berupa JSON. Method pada HTTP Request dapat bekerja sesuai dengan URL yang di input. Sistem diharapkan dapat diimplementasikan dalam mempermudah pelaporan kondisi peralatan meteorologi.

Kata Kunci: sistem informasi, Android, RESTful, API, web service

Abstract – Indonesia Meteorology, Climatology and Geophysics Agency operates a number of electronic instrumentation to observe, measure and record weather parameters. The data recorded on the instrumentation will be analyzed and informed to the public. The number of meteorological instrumentations increases each year both in quantity and type, thus requiring a real-time instrumentation condition monitoring system. Currently, reporting on equipment condition is still done manually, so this research will design an Android-based meteorological instrumentation reporting information system. The system consists of a front-end Android-based interface, an additional front-end web interface for recapitulation, and a back-end web service. This study utilizes web service technology as a link between several platforms in processing data from a database server. The use of web services is an example of adaptation to open data access, where the data can be used multi-platform. The results showed that the system designed is can be operated normally, where in terms of functionality, RESTful web service architecture implemented on the server is able to provide resources with a data format representation in the form of JSON. The HTTP Request method can work according to the input URL. The system is expected to be implemented to facilitate reporting of meteorological instrumentation conditions.

Keywords: information system, Android, RESTful, API, web service

1. Pendahuluan

Badan Meteorologi Klimatologi dan Geofisika (BMKG) merupakan lembaga pemerintahan non departemen yang memiliki tugas utama dalam mengukur, mengamati, menganalisis dan menginformasikan cuaca, iklim dan geofisika di wilayah Indonesia [1]. Dalam menjalankan tugas khususnya dibidang meteorologi, BMKG mengoperasikan sejumlah peralatan elektronik yang digunakan untuk memantau, mengukur dan mencatat perubahan parameter cuaca. Data-data yang tercatat pada peralatan selanjutnya akan dianalisis dan hasilnya akan diinformasikan kepada masyarakat.

Sub Bidang Instrumentasi dan Rekayasa Peralatan Meteorologi (SIRM) merupakan sub bidang yang ditugaskan oleh BMKG untuk melakukan proses perawatan dan perbaikan peralatan elektronik yang dimiliki BMKG diantaranya adalah *Automated Weather Observing System (AWOS)*, *Ground Radiosone*, Digitalisasi Taman Alat, dan Radar Cuaca [2].

Jumlah peralatan meteorologi yang berada di bawah kendali operasi SIRM setiap tahun bertambah secara kuantitas maupun jenisnya. Lokasinya pun tersebar di seluruh Indonesia. Hal ini menambah kesulitan dalam mengelola peralatan tersebut, salah satunya adalah masalah pemantauan kondisi peralatan. Peralatan-peralatan tersebut belum terhubung dengan internet yang memungkinkan untuk pelaporan otomatis dari peralatan, sehingga memerlukan pelaporan manual. Pelaporan kondisi peralatan sangat diperlukan untuk menindaklanjuti permasalahan yang terjadi peralatan agar kondisi kerusakan dapat ditangani secepat mungkin. Kondisi peralatan yang optimal sangat berpengaruh terhadap kualitas layanan informasi cuaca untuk berbagai sektor seperti penerbangan dan informasi cuaca publik.

Saat ini sistem pelaporan peralatan masih dilakukan secara manual menggunakan aplikasi WhatsApp. Kepala Unit Pelaksana Teknis (UPT) Stasiun Meteorologi di Indonesia melaporkan kondisi peralatan secara manual pada WhatsApp dan selanjutnya dilakukann perekapan manual oleh admin dari SIRM. Sistem pelaporan tersebut memungkinkan data tidak direkap dengan baik karena keterbatasan admin dalam memilah informasi yang begitu banyak. Oleh karena itu ,diperlukan aplikasi sistem informasi untuk pelaporan kondisi peralatan yang terpusat dan mudah diakses. Sistem informasi sendiri adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manjerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [3]. Secara umum sistem informasi dapat didefinisikan sebagai suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunaanya [4], dalam hal ini informasi kondisi peralatan meteorologi yang ada di BMKG.

Pada penelitian ini akan dirancang sistem informasi pelaporan kondisi peralatan berbasis Android. Sistem dibagi menjadi dua *platform* dimana untuk pelaporan informasi dirancang menggunakan *react native* untuk aplikasi Android dan hasil rekapan dari pelaporan dibangun berbasis web menggunakan *Javascript*. REST API dibangun menggunakan teknologi Node.JS karena fleksibilitas dan kemudahannya dalam berkomunikasi satu sama lain sehingga API ini dapat diintegrasikan dan diolah dengan data lain di *multi-platform* serta sistem ini menggunakan *database* MySQL sebagai media penyimpanan *database*. Implementasi sistem tersebut diharapkan dapat mempermudah *monitoring* kondisi peralatan meteorologi di Indonesia.

2. Metodologi Penelitian

Penelitian ini memanfaatkan teknologi *web service* sebagai penghubung antara beberapa platform dalam mengolah data dari sebuah server database. Penggunaan *web service* merupakan salah contoh adaptasi terhadap akses data secara terbuka, dimana data tersebut dapat digunakan *multi-platform* yaitu *mobile,desktop, dan aplikasi web*.

2.1. Web service

Web services merupakan sebuah sistem terdistribusi memiliki komponen yang dapat di-deploy dan diakses menggunakan protokol HTTP (*Hyper Text Transport Protocol*) maupun HTTPS (*HTTP Secure*). Layanan *web* dapat di program dalam berbagai bahasa pemograman yang ada. Pada *web services* sekurang-kurangnya terdapat sebuah *web server* (jaringan penyedia

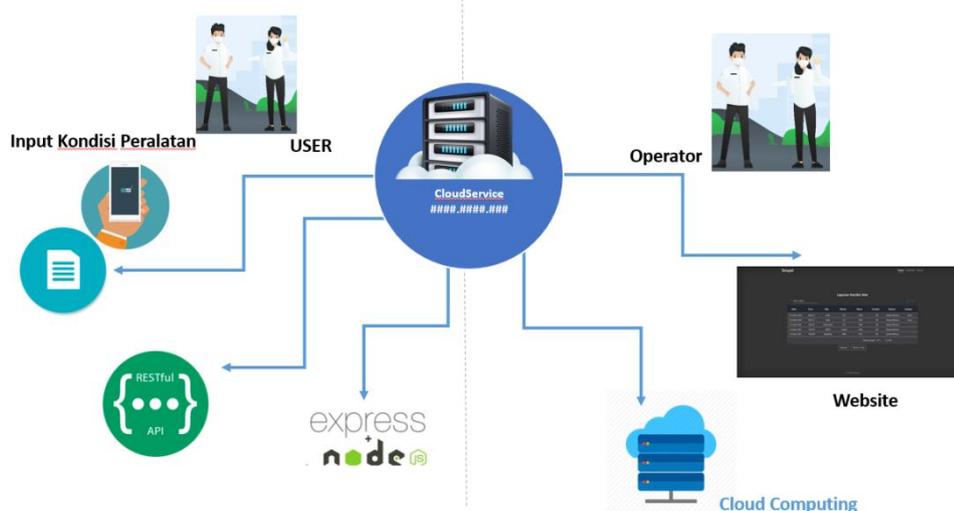
layanan) dan sebuah klien. Klien meminta layanan yang ditawarkan oleh *web server* bisa melalui desktop/PC maupun *mobile*. Layanan web memiliki dua teknik populer, yaitu SOAP (*Simple Object Access Protocol*) dan RESTfull. Dalam layanan berbasis SOAP, klien mengirimkan pesan SOAP kelayanan dan layanan merespon dalam jenis pesan SOAP juga. RESTfull adalah salah satu yang menggunakan HTTP tidak hanya sebagai infrastruktur pertukaran informasi, tetapi juga sebagai sebuah pedoman untuk merancang permintaan layanan dan tanggapan dari layanan. Dalam layanan RESTfull, HTTP itu sendiri dapat diketahui sebagai suatu API. SOAP memiliki standar, *toolkit*, dan perpustakaan *software* yang melimpah. REST tidak memiliki standard resmi, relatif sedikit *toolkit*, dan pustaka *software* yang tidak merata antara bahasa pemrograman, namun ada dukungan yang terus ditingkatkan untuk layanan REST di seluruh bahasa pemograman [5].

2.2. API

API merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan *software* yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari *software* tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut. [6].

2.3. REST

Representational State Transfer yang disingkat REST merupakan salah satu jenis arsitektur untuk penerapan *web service* yang menerapkan konsep perpindahan antar state [7]. State disini dapat digambarkan seperti peramban meminta suatu halaman situs, di sisi *server* akan mengirimkan *state* halaman situs yang sekarang ke peramban. Navigasi melalui URL yang disediakan sama halnya dengan mengganti *state* dari halaman situs. Sama seperti REST bekerja, dengan bernavigasi melalui link HTTP untuk melakukan aktivitas tertentu. Seakan-akan terjadi perpindahan *state* antara satu dengan yang lain. Perintah HTTP yang bisa digunakan dalam REST adalah fungsi GET, POST, PUT atau DELETE. Dalam pengaplikasiannya, REST lebih banyak digunakan pada *web service* yang berorientasi data sumber daya. Sebutan untuk *web service* yang menerapkan arsitektur REST adalah RESTful *web service*.



Gambar 1. REST Web service

2.4. React Native

React Native merupakan *framework open source* yang berfungsi dalam membangun aplikasi *multi-platform* dengan menggunakan bahasa javascript [8]. *Framework* ini dibangun oleh facebook setelah facebook merilis react.js. Terdapat beberapa perusahaan besar yang telah menggunakan *framework* ini, seperti Tesla, Walmart, Instagram, dan lain-lain. Untuk mengembangkan sebuah aplikasi *mobile* dengan tampilan seperti pengembangan sistem secara *native*, *framework React Native* memiliki beberapa komponen yang dapat digunakan pada *platform* iOS dan Android. Sintaks *React Native* adalah perpaduan antara XML dan JavaScript yang biasa disebut JSX. Dengan menggunakan *framework* ini, pengembang dapat merender *user interface* untuk *platform* iOS yang akan di *compile* di Objective-C dan Android yang akan di *compile* di Java dalam satu kode saja.

2.5. Javascript

JavaScript merupakan salah satu bahasa *script website* yang paling banyak digunakan untuk menambah manipulasi *script* HTML dan CSS pada sisi *client/browser*. JavaScript mampu memberikan fungsionalitas lebih pada *website*, seperti validasi form, berkomunikasi dengan *server* serta membuat *website* lebih interaktif dan animatif. JavaScript digunakan pada banyak peramban seperti Internet Explorer, Firefox, Chrome, Opera, Safari dan lain sebagainya. Hampir seluruh peramban mendukung JavaScript sehingga tidak perlu khawatir kode JavaScript yang digunakan pada *website* tidak berfungsi [9].

2.6. JSON

JSON (*JavaScript Object Notation*) adalah modal pertukaran data yang ringan, mudah dibaca, dan mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum seperti C, C++, C#, Java, JavaScript, Perl, dan Python. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur yaitu kumpulan pasangan nama atau nilai dan daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

2.7. NodeJS

Node.js merupakan perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis *web* dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi *client / browser* saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi *server*, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Berbeda dengan bahasa pemrograman sisi *server* pada umumnya yang bersifat sinkronis atau *blocking*, Node.js bersifat asinkronis atau *non- blocking* sebagaimana halnya JavaScript bekerja. Node.js berjalan dengan basis event (event-driven). Maksud dari *blocking* secara sederhana adalah bahwa suatu kode program akan dijalankan hingga selesai, baru kemudian beralih ke kode program selanjutnya [10].

2.8. MySQL

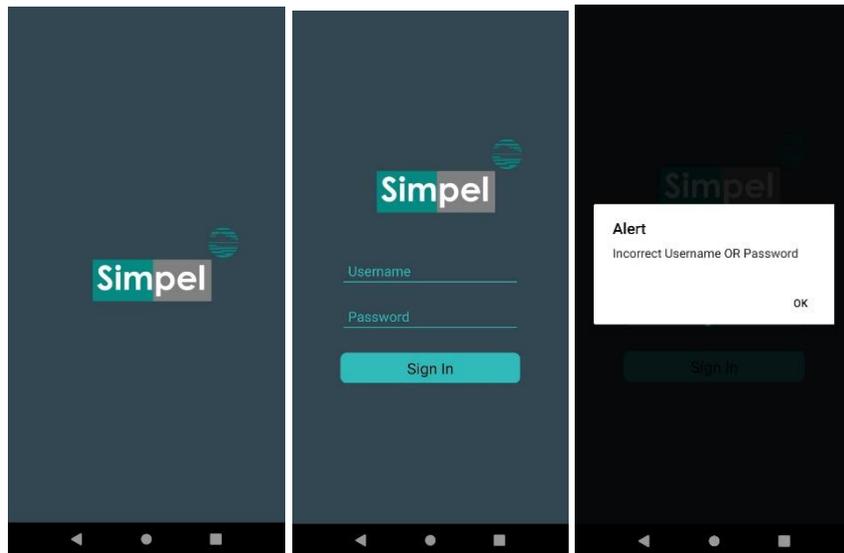
MySQL adalah sebuah aplikasi *server* basis data yang dijalankan di *server*, MySQL menggunakan sintak SQL standar dalam penggunaannya. MySQL bersifat relasional yang memungkinkan satu tabel dapat berelasi dengan tabel lainnya, sehingga memungkinkan dilakukannya normalisasi untuk mencegah redundansi dalam pemakaian data.[11]

3. Hasil dan Pembahasan

Sistem informasi peralatan meteorologi dirancang dengan keluaran berupa *interface user side*, *interface admin side*, dan *back end* berupa *web service*. Hasil keluaran yang telah diperoleh dari penelitian ini berdasarkan perancangan sistem sebagai berikut :

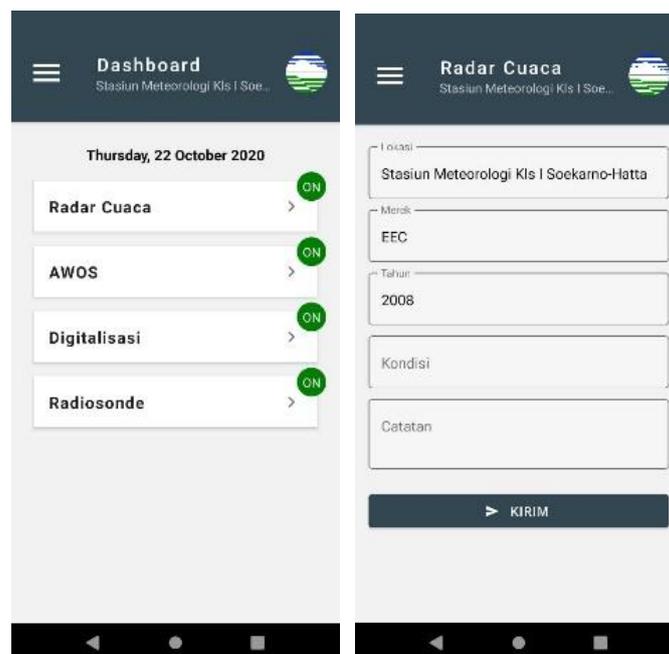
3.1. Interface (Front End) User- Side

Interface user-side dibuat dalam bentuk aplikasi berbasis Android. Pada aplikasi ini akan dilakukan input data oleh Kepala UPT untuk melaporkan status kondisi peralatan meteorologi setiap hari.



Gambar 2. Halaman Utama

Sistem tampilan berbasis aplikasi Android menggunakan *framework React Native*. Pada Gambar 2 merupakan tampilan *splashscreen* serta halaman *Login* dimana *user* yaitu Kepala UPT Stasiun Meteorologi memasukkan *username* serta *password* yang telah diberikan sehingga *user* dapat mengakses dan melakukan pelaporan peralatan utama yang ada di stasiun.

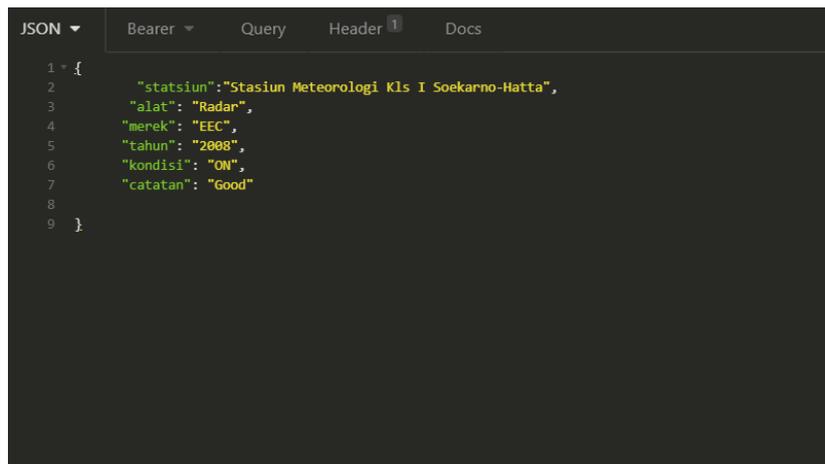


Gambar 3. Halama Dashboard dan Pelaporan

Setelah *user* memasukkan *username* dan *password* yang sesuai, kemudian akan tampil tampilan *dashboard* yang berisi peralatan di stasiun. *User* juga dapat melihat riwayat terakhir data pelaporan. Data *username* dan data peralatan dirancang berelasi pada *database server*. Pada Gambar 3, *username* dapat melakukan pelaporan kondisi peralatan dan catatan terhadap peralatan di stasiun. Data inputan yang dimasukkan *user* melalui aplikasi akan otomatis masuk ke *database server* menggunakan metode Rest API POST.

3.2. Web Service (Back End)

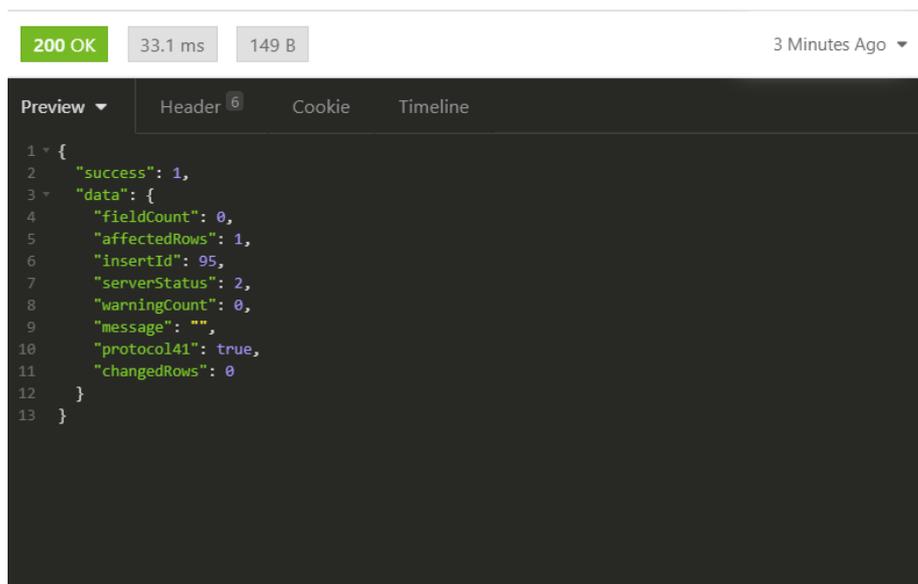
Berikut hasil pengujian *Back End* menggunakan aplikasi Insomnia dengan melakukan Rest API Method Post dan Get.



```
JSON ▾  Bearer ▾  Query  Header 1  Docs
1 {
2   "stasiun": "Stasiun Meteorologi KIs I Soekarno-Hatta",
3   "alat": "Radar",
4   "merek": "EEC",
5   "tahun": "2008",
6   "kondisi": "ON",
7   "catatan": "Good"
8
9 }
```

Gambar 4. Insert Laporan Kondisi Peralatan

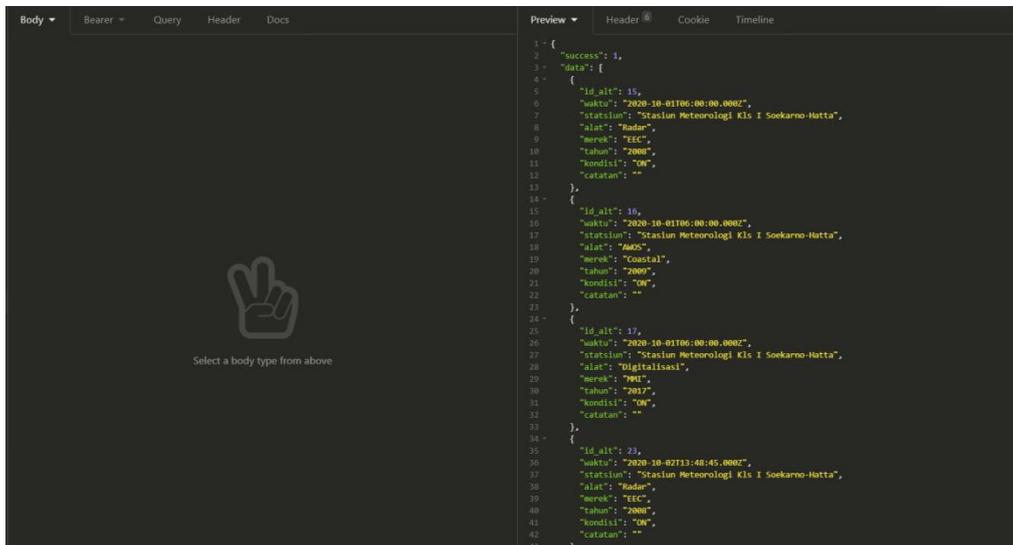
Gambar 4 menyajikan hasil dari pengujian HTTP *Request* POST pada *insert* laporan kondisi peralatan meteorologi. Pengujian dilakukan dengan mengetik alamat URL pada *text box* bagian atas dan memilih *method* POST pada *list* disamping *text box*. Untuk melakukan penambahan data dibutuhkan list data yang akan di input sesuai dengan nama kolom yang terdapat pada tabel dalam *database*.



```
200 OK  33.1 ms  149 B  3 Minutes Ago ▾
Preview ▾  Header 6  Cookie  Timeline
1 {
2   "success": 1,
3   "data": {
4     "fieldCount": 0,
5     "affectedRows": 1,
6     "insertId": 95,
7     "serverStatus": 2,
8     "warningCount": 0,
9     "message": "",
10    "protocol41": true,
11    "changedRows": 0
12  }
13 }
```

Gambar 5. Respon Status Success

Gambar 5 menyajikan respon dari HTTP *request* POST. Respon tersebut menyajikan status *message success* dan status 200 OK pada bagian *header*.

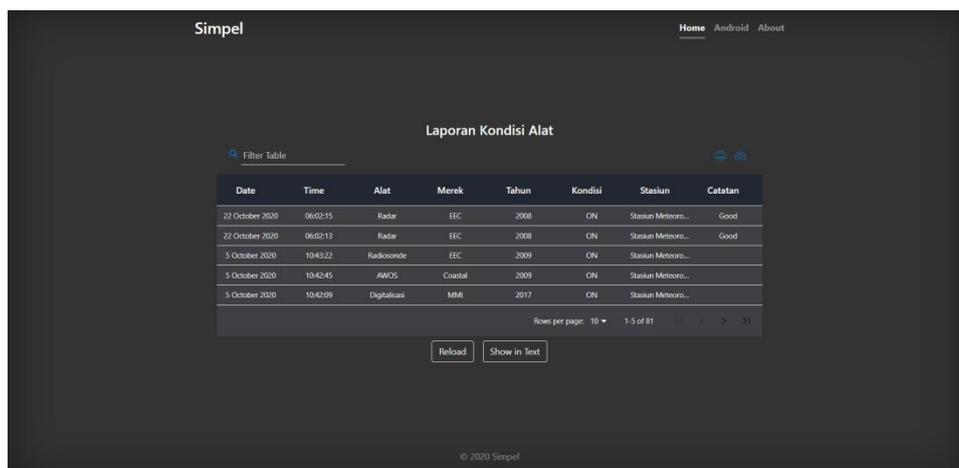


Gambar 6. View Service Laporan Kondisi Peralatan

Gambar 6 menyajikan hasil pengujian *web service* pada HTTP *request* GET data hasil pelaporan kondisi alat yang telah diinput oleh *user*. Terdapat kolom *raw* yang menampilkan keluaran data mentah JSON yang membuktikan bahwa keluaran sesuai dengan *request* yang dikirim. Terdapat status 200 OK pada bagian *header body* yang menyatakan bahwa *request* berhasil dieksekusi.

3.3. Interface (Front End) Admin Side

Interface admin side dibuat dengan tujuan untuk melihat hasil rekapan dari Kepala UPT. *Interface admin side* dibuat secara *realtime* untuk menunjukkan status pelaporan setiap Kepala UPT.



Gambar 7. Halaman Web Admin

Gambar 7 merupakan halaman admin yang dirancang berbasis *web* menggunakan *framework* javascript. Data Pelaporan yang telah diinputkan oleh *user*, secara otomatis akan masuk di *database server* dan diintegrasikan dengan halan *web* admin. Halaman Admin ini berfungsi agar admin dapat melakukan *memonitoring* dan merekapitulasi data kondisi peralatan yang telah dimasukkan oleh user melalui aplikasi Android.

4. Kesimpulan

Berdasarkan penelitian, pengujian dan pembuatan dokumentasi telah dilakukan, dapat ditarik beberapa kesimpulan bahwa sistem pelaporan peralatan meteorologi berbasis Android telah dirancang dengan baik dimana dari segi *interface* dan fungsionalitas dapat berjalan dengan lancar dan penerapan arsitektur *restful web service* yang diimplementasikan pada *server* mampu menyediakan *resource* dengan representasi format data berupa JSON sehingga data yang inputkan dapat diolah diberbagai *platform*. Pengujian menggunakan Aplikasi Insomnia yang menghasilkan keluaran status 200 OK pada status *header* membuktikan bahwa *method* pada HTTP *Request* dapat bekerja sesuai dengan URL yang diinput. *RESTful web service* juga dapat memberi kemudahan bagi *developer* dalam melakukan pengembangan aplikasi kedepannya.

Referensi

- [1] Sasongko, A. (2014). Perancangan Aplikasi Rekam Data Cuaca Hasil Pengamatan Observer Stasiun Meteorologi BMKG Berbasis Website (Studi Kasus: Stasiun Meteorologi Supadio Pontianak). *Jurnal Khatulistiwa Informatika*, 2(2).
- [2] Wardani, D., Sulisty, S., & Mustika, I. W. (2018, November). The Blueprint of AWOS Implementation for Aviation Services at BMKG. In *Conference SENATIK STT Adisutjipto Yogyakarta* (Vol. 4, pp. 157-166).
- [3] Hendrianto, D. E. (2013). Pembuatan Sistem Informasi Perpustakaan Berbasis Website Pada Sekolah Menengah Pertama Negeri 1 Donorojo Kabupaten Pacitan. *IJNS-Indonesian Journal on Networking and Security*, 4(3).
- [4] Athoillah, M., & Irawan, M. I. (2013). Perancangan sistem informasi mobile berbasis Android untuk kontrol persediaan barang di gudang. *Jurnal Sains dan Seni Pomits*, 1(1), 1-6.
- [5] Kalin, M. (2013). *Java Web services: Up and Running: A Quick, Practical, and Thorough Introduction*. " O'Reilly Media, Inc."
- [6] Webber, J., Parastatidis, S., & Robinson, I. (2010). *REST in practice: Hypermedia and systems architecture*. " O'Reilly Media, Inc."
- [7] Culesshabrur, "Trend Teknologi," 31 January 2013. [Online]. Available: <http://s4nba0.blogspot.com/2013/01/rest-api.html>. [Accessed 20 Oktober 2020].
- [8] Hasyim, A., 2016. Memulai Pengembangan Android dengan React Native di Windows. [Online] Available at: <https://www.codepolitan.com/memulai-pengembangan-Android-dengan-react-native-di-windows-57b85678b26a9-17960> [Accessed 20 Oktober 2020]
- [9] Komputer, W. (2012). Paling Dicari! JavaScript Source Code.
- [10] Sheba, Lusfekar. 2017. Panduan Node.js 8.X dalam 14 BAB.
- [11] Perkasa, M. I., & Setiawan, E. B. (2018). Pembangunan *Web service* Data Masyarakat Menggunakan REST API dengan Access Token. *Ultima Computing: Jurnal Sistem Komputer*, 10(1), 19-26.