

# Pembangunan *Web Service* Data Masyarakat Menggunakan REST API dengan *Access Token*

Muhammad Iqbal Perkasa<sup>1</sup>, Eko Budi Setiawan<sup>2</sup>

Program Studi Teknik Informatika, Universitas Komputer Indonesia, Jalan Dipatiukur 112 Bandung  
mail@iqbalperkasa.com  
eko@email.unikom.ac.id

Diterima 23 April 2018

Disetujui 8 Juni 2018

**Abstract**— Data is one of the most important things in this information and information technology era that evolving now. Currently, the government still has not used the public data maximally for administrative purposes. Utilization of this big population data is the creation of a web service application system with REST API where this data will be open and accessible to those who have access. One of the institutions that use this service is the Manpower and Transmigration Service where this system can make the Dinas staff more efficient to create and register job search cards using available community data. This application is able to provide and facilitate many parties, such as data administrators to monitor data usage, registration employee in input data, and people able to register independently.

**Index Terms**—Web service, API, Rest api, People data

## I. PENDAHULUAN

Pada tahun 2010, di salah satu Kabupaten di Jawa Barat tercatat memiliki 1.443.156 jiwa yang berusia di atas 15 tahun. Data tersebut mencakup nama lengkap, jenis kelamin, agama, golongan darah, tanggal dan tempat lahir, status perkawinan, alamat, dan kewarganegaraan, serta informasi detail lainnya. Namun banyaknya data yang tercatat di *server* hingga saat ini belum dimanfaatkan secara maksimal dan belum adanya halaman statistik untuk memonitor penggunaan data kependudukan tersebut.

Pemanfaatan data penduduk yang terolah salah satunya adalah digunakan oleh Dinas Ketenagakerjaan dan Transmigrasi dalam pengisian baris data pencari pekerjaan. Salah satu tugas dari dinas ini adalah pembuatan atau penerbitan Kartu Pencari Kerja untuk masyarakat. Setiap harinya petugas dinas mampu memberikan pelayanan dengan menerbitkan kartu tersebut sebanyak lebih dari 100 kartu.

Setelah dianalisis, Dinas Ketenagakerjaan dan Transmigrasi dapat menggunakan data yang tercatat di *server database* Dinas Kependudukan dan Catatan Sipil yang diharapkan mampu jauh mengurangi beban kerja dari petugas Dinas Ketenagakerjaan dan Transmigrasi. Data yang disediakan bersifat terbuka

dan transparan, meskipun begitu, pihak *developer* harus memiliki *access token* untuk mengakses data di *server*. Berdasarkan pemaparan di atas, maka sebuah *web service* untuk mengambil data masyarakat diharapkan mampu mempercepat dalam penulisan atau pengisian data masyarakat tersebut.

Pemanfaatan *web service* sudah banyak dilakukan pada beberapa sektor. Warsito [1] mengimplementasikan *web service* pada sektor pendidikan di lingkungan perguruan tinggi, Fatoni [2] menggunakannya dalam segi sektor bisnis pembelian tiket pesawat, serta penelitian Kurniawan [3] yang mengimplementasikan *rest web service* untuk proses *sales order* dan *sales tracking*.

Beberapa penelitian yang pernah dilakukan sebelumnya masih belum ada yang menggunakan fitur untuk pengelolaan *access token*. Penelitian ini menggunakan fitur untuk *monitoring* akses ke dalam *database* di *server*, sehingga seorang Administrator dapat dengan mudah memonitor kondisi *traffic* data terutama *request* akses data ke *server*.

## II. TINJAUAN PUSTAKA

### A. Web Service

Menurut W3C yang merupakan lembaga internasional yang bergerak dalam membangun standarisasi *web service* adalah sistem perangkat lunak yang didesain dapat dioperasikan mesin ke mesin melalui jaringan. Arsitektur *web service* menurut Chen, et al [4] memodelkan interaksi antara tiga peran yaitu penyedia layanan, konsumen layanan, dan pendaftar layanan. Penggunaan *web service* juga digunakan oleh Pruter et al [5] untuk mengkoneksikan secara dinamis dari perangkat-perangkat yang sudah diketahui maupun tidak diketahui di dalam satu jaringan komputer. Dengan demikian, *web service* memungkinkan sarana operasi antar perangkat lunak menjadi standar pada berbagai *platform* yang berbeda.

### B. Access Token

*Access token* merupakan sebuah *string unique* dan acak yang berisi angka dan huruf. Seperti yang dibahas pada penelitian Ahmadi [6] bahwa *access token* adalah sebuah kunci yang dibutuhkan oleh pemohon layanan yang harus digunakan untuk mengidentifikasi dan memverifikasi apakah pemohon memiliki hak dalam mengakses *web service*. Masing-masing pemilik akun pada layanan *web service* memiliki sebuah *access token* yang bersifat unik. *Access token* ini berfungsi untuk menambah tingkat keamanan pada *web service*, sehingga *web service* hanya dapat diakses oleh pihak yang berwenang saja.

### C. REST

*REST* menurut Sinha et al [7] dan Zhou et al [8] merupakan gaya arsitektur dalam mendesain sebuah *web service* di mana desain *REST* memiliki *resource* yang dapat diakses melalui sebuah alamat *HTTP URL* yang *unique*. *REST* juga memungkinkan klien dapat melakukan request melalui protokol *HTTP* dengan mudah menggunakan *URI* seperti pada penelitian Kurniawan [9]. Masing-masing alamat *URL* mengacu kepada kumpulan program yang akan dieksekusi dan akan mengembalikan pesan kepada pengirim perintah.

*REST* mengirimkan perintah yang akan dikerjakan oleh *server* menggunakan metode-metode *HTTP request method* yang disebut *verb*. Mengacu pada penelitian Lee [10] dan Rahman [11] terdapat delapan *HTTP request method*, yaitu *GET*, *POST*, *PUT*, *DELETE*, *OPTIONS*, *HEAD*, *TRACE*, dan *CONNECT*. Dalam penggunaan *API REST* hanya menggunakan empat dari metode-metode tersebut, yaitu: *GET*, *POST*, *PUT*, dan *DELETE*.

Pesan yang diterima dari *server* berupa kode *HTTP* berhasil atau gagal di dalam *header* dan isi pesan hasil pengolahan program itu sendiri. Berikut adalah kode-kode *HTTP* yang sering digunakan dalam penggunaan *REST API* [12]:

- a. 200 *OK*  
Perintah yang dikirim ke *server* benar dan berhasil dijalankan.
- b. 400 *Bad Request*  
Perintah yang dikirim ke *server* berisi isian yang salah.
- c. 401 *Unauthorized*  
Pengirim perintah mengirimkan kode kunci yang salah.
- d. 403 *Forbidden*  
Pengirim perintah tidak memiliki hak akses ke dalam *resource* yang dituju.
- e. 404 *Not Found*  
*Resource* yang dituju tidak ditemukan dalam *server*.
- f. 429 *Too Many Requests*

Pengirim perintah mengakses mencapai/melebihi dari limit yang telah ditentukan dari batas waktu tertentu.

### g. 500 *Internal Server Error*

*Server* atau potongan program dalam *resource* mengalami kesalahan.

### D. Apache JMeter

Aplikasi yang dibuat oleh *Apache Software Foundation* ini adalah berfungsi untuk melakukan pengujian terhadap *server*, bagaimana *server* mampu mengimbangi banyaknya permintaan yang datang. Aplikasi ini dapat digunakan untuk pengujian basis data, *FTP*, *LDP*, *Web Service*, *HTTP*, dan koneksi *TCP*.

### E. JSON

*JSON* adalah sebuah format pesan balikan yang berukuran kecil yang mudah dibaca dan ditulis oleh manusia, dan juga mudah ditulis dan diurai oleh mesin. *JSON* merupakan salah satu pesan balikan yang dapat digunakan dalam sebuah *REST API*.

*JSON* terbagi dalam dua struktur, yaitu:

1. Gabungan *name/value*, atau yang biasa dikenal *object* atau *record* dalam bahasa pemrograman.
2. List *value*, atau yang biasa dikenal larik dalam bahasa pemrograman.

### F. PHP

*PHP* adalah sebuah bahasa pemrograman sumber terbuka yang ditujukan untuk pemrograman *web* dan dapat diaplikasikan ke *HTML*. *PHP* merupakan bahasa skrip yang ditanam dalam *HTML*. Ini berarti bahwa kita dapat menggabungkan kode *PHP* dan *HTML* dalam berkas yang sama [13]. *PHP* merupakan bahasa pemrograman *server-side* yang di mana program akan dijalankan di *server* dan hasilnya akan diintegrasikan ke dalam kode sumber *HTML*.

### G. 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.

## III. HASIL DAN PEMBAHASAN

Pada tahap ini, akan dijelaskan mengenai tahapan-tahapan dari penelitian yang dilakukan, yaitu adalah analisis masalah yang menjelaskan mengenai masalah yang ditemukan; analisis pembentukan *access token*, bagaimana proses *access token* dibentuk; analisis alur validasi, menjelaskan validasi permintaan masing-masing request sebelum data diberikan; analisis request dan response dari permintaan *web service*;

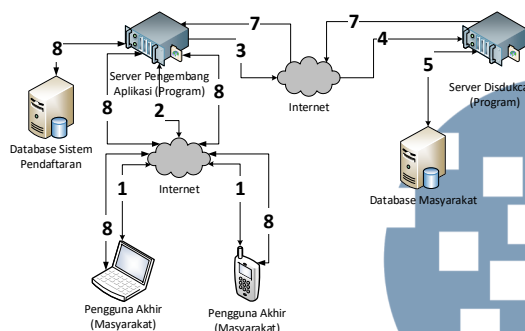
analisis infrastruktur; *use case diagram*; implementasi sistem; pengujian; dan terakhir adalah wawancara ke administrator data penduduk.

#### A. Analisis Masalah

Analisis masalah yang ditemukan setelah proses identifikasi di Dinas Kependudukan dan Catatan Sipil adalah data masyarakat yang dimiliki oleh Dinas Kependudukan dan Catatan Sipil belum dimanfaatkan dengan baik dan administrator data di Dinas Kependudukan dan Catatan Sipil belum memiliki aplikasi untuk memonitor penggunaan data masyarakat.

#### B. Gambaran Umum

Berikut adalah gambaran umum sistem API masyarakat yang dapat dilihat pada Gambar 1:



Gambar 1. Gambaran umum sistem API

Penjelasan dari Gambar 1 adalah sebagai berikut, yang diterangkan dengan bentuk poin per poin bagaimana tahapan dari pengambilan data dari server:

1. Pengguna akhir yaitu masyarakat mengakses web untuk mendaftarkan kartu pencari kerja pada aplikasi yang telah dikembangkan oleh pengembang aplikasi.
2. Permintaan akan masuk ke dalam server pengembang aplikasi. Program akan membaca permintaan dari pengguna akhir, yaitu meminta data masyarakat berdasarkan NIK.
3. Server pengembang aplikasi harus terkoneksi ke internet untuk dapat mengakses server Dinas Kependudukan dan Catatan Sipil pada alamat domain tertentu.
4. Server pengembang aplikasi mengakses server Dinas Kependudukan dan Catatan Sipil pada alamat endpoint URL API yang telah dibuat.
5. Setelah berhasil melakukan validasi permintaan yang *valid* dari pengembang aplikasi, program akan terhubung ke basis data masyarakat pada server database dan mengambil data masyarakat berdasarkan NIK yang dimaksud.
6. Setelah berhasil mengambil data yang dimaksud, program akan menuliskan catatan *history* di server database aplikasi API dan

mengurangi limit pemakaian *access token* pengembang yang bersangkutan.

7. Server Dinas Kependudukan dan Catatan Sipil mengembalikan balikan berupa data masyarakat berformat *JSON* ke server pengembang aplikasi.
8. Data yang telah didapat akan dikembalikan ke aplikasi yang digunakan oleh pengguna akhir, yaitu masyarakat.

#### C. Analisis Pembentukan Access Token

Pembentukan *access token* merupakan asumsi dari peneliti. Token ini didapat dari kombinasi alamat email penanggung jawab pemilik *access token* dan waktu *timestamp* pembuatan *access token* yang dienkripsi menggunakan metode SHA256. Contoh dari pembentukan *access token* yang dapat dilihat pada Tabel 1.

Tabel 1. Contoh pembentukan *access token*

Alamat Email	email@domain.com
Timestamp Pembuatan Token	1492035660 (12 April 2017 22:20)
Fungsi SHA256	sha256('email@domain.com1492035660')
Access Token yang Didapat	91b37dee294eb66ce67c9d8bbbe6f71f0d9baf8f9a697c37f35e4108bbccf9be

Berdasarkan data pada tabel 1 maka diperoleh dari pembangkitan *access token* tersebut adalah berupa string 91b37dee294eb66ce67c9d8bbbe6f71f0d9baf8f9a697c37f35e4108bbccf9be. String inilah yang kemudian harus disisipkan di setiap permintaan data API oleh akun email@domain.com.

#### D. Analisis Alur Validasi

Untuk memastikan permintaan berasal dari sumber yang jelas dan untuk memastikan keamanan data, maka perlu dibuatnya tahapan-tahapan validasi dari sumber permintaan. Masing-masing tahap harus dipenuhi untuk mendapatkan data yang dibutuhkan. Terdapat delapan tahap validasi yang harus dipenuhi, yaitu:

1. Validasi *access token* kosong. Sistem akan memeriksa apakah sumber permintaan membawa *access token* atau tidak. Apabila parameter *access token* kosong, maka sistem akan mengembalikan kode error 403 *Forbidden* dan pesan error "Harap masukkan *access token*".
2. Validasi format *access token*. Sistem akan memeriksa apakah sumber permintaan membawa *access token* dengan format yang valid atau tidak. *Access token* yang valid adalah berupa string dengan panjang string adalah enam puluh empat (64) karakter. Apabila format yang dikirimkan salah, maka sistem akan mengembalikan kode error 400

- Bad Request* dan pesan error “Format *access token* salah”.
- Validasi kebenaran *access token*. Sistem akan memeriksa apakah sumber permintaan membawa *access token* yang valid atau tidak. Sistem akan membandingkan *access token* yang dibawa oleh sumber dengan daftar *access token* di basis data. Apabila *access token* tidak ditemukan, maka sistem akan mengembalikan kode error 401 *Unauthorized* dan pesan error “Access token salah”.
  - Validasi *access token* dengan domain terdaftar. Sistem akan memeriksa apakah *domain* sumber permintaan berasal dari *domain* yang terdaftar pada *access token* yang bersangkutan. Sistem akan membandingkan *domain* dari sumber permintaan dengan daftar *domain access token* yang bersangkutan di basis data. Apabila *domain* tidak ditemukan, maka sistem akan mengembalikan kode error 403 *Forbidden* dan pesan error “Sumber *domain* tidak terdaftar pada *access token* ini”. Validasi ini penting untuk memblokir permintaan AJAX dari *domain* yang tidak terdaftar.
  - Validasi batas *limit*. Sistem akan memeriksa apakah *access token* yang bersangkutan memiliki limit akses data atau tidak. Sistem akan memeriksa jumlah permintaan dari *access token* yang bersangkutan dari batas waktu tertentu di basis data. Apabila limit *access token* telah dicapai, maka sistem akan mengembalikan kode error 429 *Too Many* dan pesan error “Access token salah”.
  - Validasi NIK kosong. Sistem akan memeriksa apakah sumber permintaan membawa NIK atau tidak. Apabila parameter NIK kosong, maka sistem akan mengembalikan kode error 404 *Not Found* dan pesan error “Harap masukkan NIK”.
  - Validasi format NIK. Sistem akan memeriksa apakah sumber permintaan membawa NIK dengan format yang valid atau tidak. NIK yang valid adalah berupa kumpulan angka dengan panjang numerik adalah enam belas (16) karakter. Apabila format yang dikirimkan salah, maka sistem akan mengembalikan kode error 400 *Bad Request* dan pesan error “Format NIK salah”.
  - Validasi kebenaran NIK. Sistem akan memeriksa apakah sumber permintaan membawa NIK yang valid atau tidak. Sistem akan membandingkan NIK yang dibawa oleh sumber dengan daftar NIK di basis data Dinas Kependudukan dan Catatan Sipil. Apabila NIK tidak ditemukan, maka sistem akan mengembalikan kode error 404 *Unauthorized* dan pesan error “NIK salah”.

#### E. Analisis Request-Response

Setelah permintaan yang masuk ke dalam *server* divalidasi dan diterima dengan benar, maka *server* akan memproses permintaan menghubungkan dengan basis data.

##### 1. URL Endpoint.

Semua permintaan diakses melalui sebuah endpoint URL yang beralamat di `http://11x.97.187.1xx:8080/api/get_data_by_nik`.

##### 2. Parameter

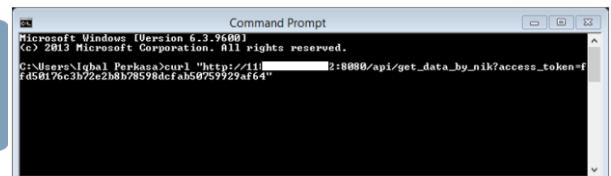
Terdapat dua parameter yang dibutuhkan dalam setiap proses. Kedua parameter harus valid yang dalam setiap permintaan *server* akan memvalidasi dan akan mengembalikan balikan *error* apabila terjadi kesalahan. Parameter-parameter tersebut adalah *access token*, kunci *access token* yang dimiliki oleh masing-masing akun yang valid (terverifikasi, berstatus aktif, dan memiliki sisa limit); dan *nik*, NIK masyarakat yang akan diambil datanya.

##### 3. HTTP Verb

HTTP Verb yang digunakan untuk sistem API masyarakat ini adalah *GET*.

##### 4. Otentikasi

Setiap permintaan, sistem akan memvalidasi apakah permintaan membawa *access token* yang valid atau tidak. Access token dikirim melalui parameter pada URL.



Gambar 2. Otentikasi *access token* dikirim melalui parameter url

##### 5. Limiting

Masing-masing *access token* memiliki batas limit permintaan data sebanyak 60 permintaan setiap menit. Tiap menit limit ini akan kembali di-reset dan mengulang hitungan. Administrator memiliki hak untuk mengubah jumlah batas limit.

##### 6. Response

Apabila sistem berhasil memvalidasi permintaan yang masuk, maka sistem akan mengembalikan permintaan yang dibutuhkan. Balikan yang diberikan berupa format *JSON* dan informasi-informasi *header*. Contoh balikan dari permintaan yang valid dapat dilihat pada gambar 3.



x-limit-limit → 999999

x-limit-remaining → 999998

```

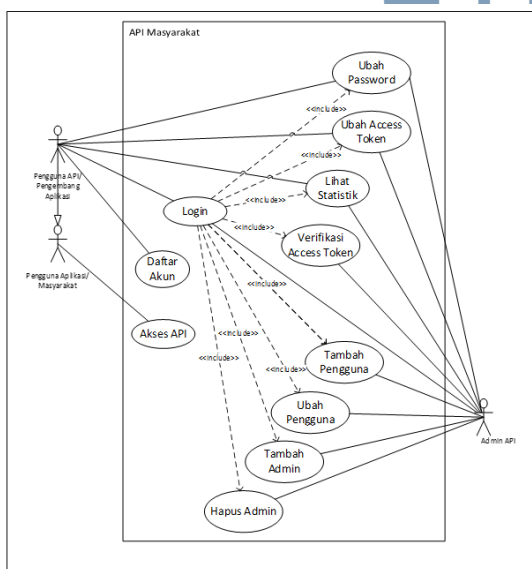
1 {
2   "SUCCESS": true,
3   "MESSAGE": [],
4   "DATA": {
5     "NIK": "0918",
6     "NAMA_LGKP": "CIANJUR",
7     "TMPT_LHR": "JAWA BARAT",
8     "NAMA_PROP": "CIANJUR",
9     "NAMA_KAB": "SUKALUVU",
10    "NAMA_KEC": "SELAJAMBE",
11    "NO_PROP": "32",
12    "NO_KAB": "3",
13    "NO_KEC": "9",
14    "NO_KEL": "2001",
15    "AGAMA": "ISLAM",
16    "JENIS_KLMIN": "PEREMPUAN",
17    "GOL_DRH": "TIDAK TAHU",
18    "STAT_KWN": "BELUM KAWIN",
19    "TGL_LHR": "09.09.1998"
20  }
21 }

```

Gambar 3. Hasil balikan data masyarakat berupa JSON

#### F. Use Case Diagram

*Use Case Diagram* adalah diagram yang menggambarkan kegiatan dan cakupan yang dikerjakan oleh masing-masing aktor. *Use case diagram* pada aplikasi yang akan dibangun yang dapat dilihat pada gambar 4.



Gambar 4. Use case diagram

#### G. Implementasi Sistem

Tahap implementasi menjelaskan mengenai implementasi yang dilakukan dari pembahasan dan analisis pada tahapan-tahapan sebelumnya.

##### G.1. Implementasi Perangkat Lunak

Pada tahap ini akan dijelaskan mengenai implementasi perangkat lunak yang terdapat di lapangan yang dapat dilihat pada tabel 2.

Tabel 2. Implementasi perangkat lunak

No.	Komponen	Spesifikasi
1.	Arsitektur sistem operasi	32 bit
2.	Apache Server	versi 2.4.4
3.	PHP	versi 5.5.3
4.	MySQL/MariaDB	versi 5.6.11
5.	Browser	Tersedia

##### G.2. Implementasi Server

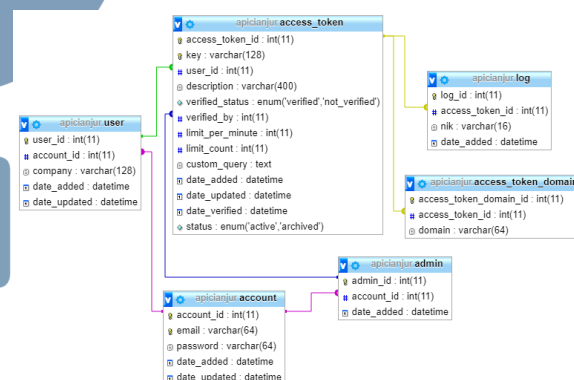
Pada tahap ini akan dijelaskan mengenai implementasi server yang terdapat di lapangan yang dapat dilihat pada Tabel 3.

Tabel 3. Implementasi server

No.	Komponen	Spesifikasi
1.	CPU	Intel Xeon X5650 @ 2.67 GHz
2.	Media penyimpanan (HDD/SSD)	1 TB
3.	Memori (RAM)	12 GB
4.	Keyboard	Tersedia
5.	Mouse	Tersedia
6.	Koneksi Internet	Tersedia

##### G.3. Implementasi Basis Data

Implementasi basis data merupakan implementasi tabel-tabel basis data yang digunakan untuk menyimpan seluruh data sistem yang dapat dilihat pada gambar 5.



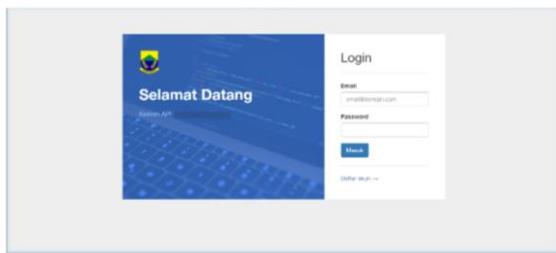
Gambar 5. Skema implementasi basis data

#### H. Implementasi Antarmuka

Tahap ini akan memaparkan implementasi antarmuka dari halaman web untuk memonitor penggunaan data yang telah digunakan dari pengakses. Antarmuka meliputi tampilan untuk melihat penggunaan data dalam bentuk grafik dan tabel dan form pengelolaan data.

##### 1. Login

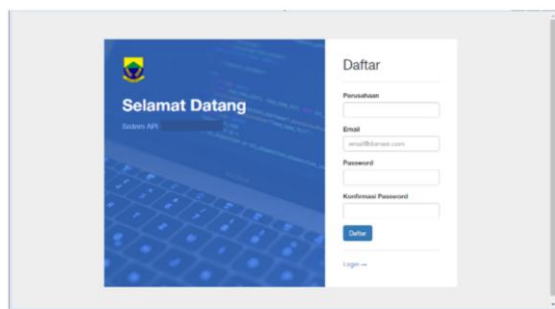
Antarmuka untuk menampilkan form login yang digunakan untuk masuk ke dalam sistem. Dibutuhkan dua masukan, yaitu email dan password dari pengguna yang bersangkutan.



Gambar 6. Antarmuka login

## 2. Daftar Akun

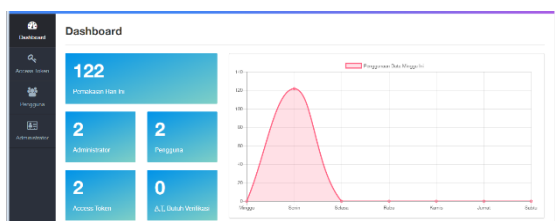
Antarmuka untuk menampilkan form pendaftaran akun yang digunakan untuk mendaftar ke sistem. Setiap pendaftar harus memasukkan detail data instansi atau perusahaan mereka. Data yang harus dimasukkan adalah nama perusahaan, email, dan password.



Gambar 7. Antarmuka daftar akun

## 3. Dashboard

Antarmuka untuk menampilkan statistik penggunaan data masyarakat dari keseluruhan pengguna di sistem. Grafik menampilkan penggunaan data pada minggu tersebut. Di sebelah kiri dashboard terdapat jumlah dari data yang dimiliki di sistem, yaitu data jumlah access token, data jumlah pengguna, dan data jumlah administrator. Tampilan antarmuka dashboard dapat dilihat pada gambar 8.



Gambar 8. Antarmuka dashboard

## 4. List Access Token

Antarmuka untuk menampilkan list access token yang terdaftar di sistem. Terdapat tiga tombol untuk

melakukan aksi dari masing-masing baris, yaitu lihat/ubah access token, melihat deskripsi dari penggunaan access token, dan lihat statistik. Tampilan list access token dapat dilihat pada gambar 9.

No	Key	Perusahaan	Email	Level	Tanggal
1	5d5b8b0c-4c5d3926d807b9f6a115d52bc2a670b6daed07eaa797ab61c	lgbl Perkasa, Personal	lgbl@perkasa.com	Admin	21 Agustus 2017
2	15d52b0c-4c5d3926d807b9f6a115d52bc2a670b6daed07eaa797ab61c	lgbl Perkasa, Personal	lgbl@perkasa.com	Admin	21 Agustus 2017

Gambar 9. Antarmuka list access token

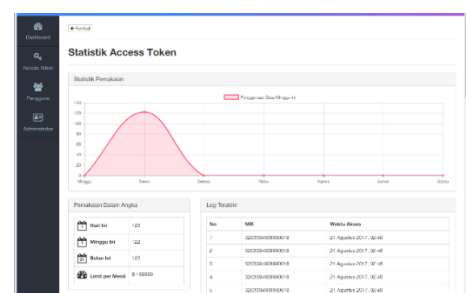
## 5. Form Ubah Access Token

Antarmuka untuk menampilkan form ubah access token yang digunakan untuk mengubah data dari masing-masing access token. Terdapat juga tombol untuk meregenerasi access token yang baru. Tampilan form ubah access token dapat dilihat pada gambar 10.

Gambar 10. Antarmuka form ubah access token

## 6. Statistik Access Token

Antarmuka untuk menampilkan statistik penggunaan data dari access token yang bersangkutan. Statistik ditampilkan dalam bentuk grafik dan tabel. Terdapat juga log dari 50 penggunaan data terakhir serta jumlah penggunaan dari access token di menit tersebut. Tampilan statistik access token dapat dilihat pada gambar 11.



Gambar 11. Antarmuka statistik *access token*

### I. Pengujian Web Service

Pengujian pertama yang dilakukan adalah dengan menghitung jumlah kartu pencari kerja sebelum dan sesudah menggunakan *web service*. Didapat dari perhitungan menggunakan *query count SQL*, jumlah kartu pencari kerja sebelum adalah berkisar 300 kartu dan setelah menggunakan *web service*, didapat jumlah kartu pencari kerja adalah berkisar 900 kartu.

Pengujian selanjutnya adalah pengujian menggunakan aplikasi *Apache JMeter*. Dengan aplikasi ini kita mampu melihat bagaimana ketahanan *server* akan permintaan *request* yang dilakukan. Pengujian akan dilakukan dengan *sample* permintaan 100 kali dalam 1 detik. Hasil dari pengujian yang dilakukan yang akan disampaikan dalam bentuk tabel pada tabel 4 dan grafik yang disajikan pada gambar 12.

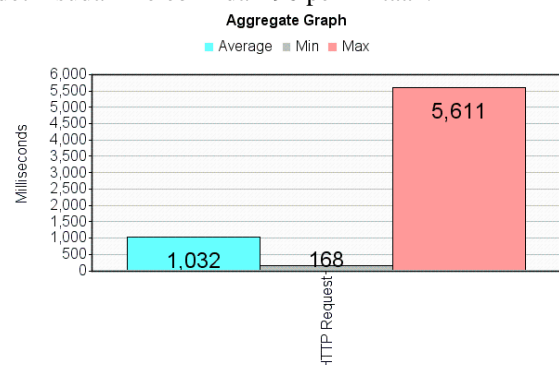
Tabel 4. Hasil pengujian 100 *sample* permintaan

Sample #	Sample Time (ms)	Sample #	Sample Time (ms)
1	232	51	1198
2	212	52	1169
3	265	53	1198
4	293	54	1226
5	241	55	1196
6	291	56	1449
7	203	57	1469
8	215	58	1245
9	204	59	1256
10	198	60	1243
11	241	61	1524
12	231	62	1503
13	210	63	1261
14	216	64	1248
15	246	65	1322
16	235	66	1623
17	237	67	1281
18	279	68	1524
19	168	69	1235
20	214	70	1215
21	236	71	1240
22	207	72	1225
23	248	73	1237
24	195	74	1234
25	266	75	1508
26	246	76	1300
27	186	77	1293
28	195	78	1284
29	194	79	1256
30	215	80	1554
31	217	81	1341
32	231	82	1603
33	227	83	1259
34	227	84	1642

Sample #	Sample Time (ms)	Sample #	Sample Time (ms)
35	212	85	1360
36	212	86	1330
37	177	87	1292
38	212	88	1356
39	223	89	1326
40	214	90	1361
41	173	91	1428
42	230	92	1409
43	219	93	1360
44	1201	94	1362
45	1233	95	1403
46	1219	96	2288
47	1267	97	5579
48	1259	98	5588
49	1298	99	5605
50	1196	100	5611

Kita bisa lihat bahwa pada *sample* percobaan ke 1 sampai 43, *server* mampu menangani permintaan dengan cepat, yaitu dalam rentang waktu 168 – 293 ms dengan rata-rata 223.093 ms, kemudian *sample* 44 – 96 menunjukkan kenaikan waktu namun tidak berarti yaitu dalam rentang waktu 1169 – 2288 ms dengan rata-rata 1345.453 ms. Kenaikan yang cukup berarti adalah ketika *server* menangani lebih dari 96 permintaan, waktu yang dibutuhkan adalah lebih dari 5500 ms.

Dari hasil 100 *sample* percobaan yang dilakukan dalam 1 detik, empat *sample* terakhir baru menunjukkan indikasi *server* mengalami beban, maka dapat diambil kesimpulan bahwa *server* mampu menangani permintaan data masyarakat ini adalah ~96 permintaan dalam 1 detik. Hasil ini dapat menjadi masukan bagi pihak Dinas untuk menambah *resource* pada *server* apabila rata-rata pengaksesan dalam satu detik sudah melebihi dari 96 permintaan.

Gambar 12. Hasil percobaan 100 *sample* permintaan

Gambar 12 menunjukkan grafik waktu permintaan dari 100 permintaan dalam 1 detik. Gambar tersebut menunjukkan rata-rata waktu adalah 1.032 ms, minimal waktu adalah 168 ms, dan maksimal waktu adalah 5.611 ms.

#### J. Wawancara Administrator

Wawancara dilakukan untuk mengetahui penilaian dari administrator basis data selaku administrator dari sistem API Masyarakat. Setelah dilakukan wawancara dengan narasumber pada hari Rabu tanggal 26 Juli 2017, didapatkan kesimpulan wawancara sebagai berikut:

1. Data-data yang ditampilkan dalam aplikasi ini sudah baik yang di mana ditampilkan berupa grafik dan tabel.
2. Kantor-kantor sebagai pengembang dan masyarakat dapat dimudahkan dengan adanya *web service* ini untuk memudahkan dalam pengambilan data masyarakat.
3. Tampilan dan menu dari aplikasi ini sudah baik dan mudah digunakan.
4. Aplikasi sudah bagus dan memudahkan untuk melihat data-datanya, pun dengan tampilan sudah baik dan tepat.

#### IV. SIMPULAN

Adapun kesimpulan dan saran yang didapat dari penelitian ini adalah sebagai berikut:

##### A. Kesimpulan

Dari hasil yang didapat dari tahap-tahap yang telah dikerjakan melalui proses perancangan, implementasi, pengujian, dan wawancara dengan administrator, didapat kesimpulan sebagai berikut:

1. *Web service* ini mampu dan dapat mempercepat pendaftaran dengan keandalan *server* yang baik.
2. Administrator data penduduk dapat dimudahkan dalam memonitor penggunaan data penduduk dan mengatur hak akses masing-masing *access token*.

##### B. Saran

Adapun beberapa saran yang dapat disampaikan adalah sebagai berikut:

1. Memberikan statistik data penggunaan data masyarakat secara *real-time*, untuk memberikan *user experience* yang lebih baik untuk administrator data.
2. Memberikan fitur *cache* pada *web service* untuk penggunaan data yang lebih efisien.

#### DAFTAR PUSTAKA

- [1] Warsito, A.B., Ananda, A. and Triyanjaya, D. Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service. *Technomedia Journal*, 2(1), pp.26-35, 2017.
- [2] Fatoni, A., Rancang Bangun Aplikasi PembelianTiket Pesawat Online Berbasis Mobile Pada platform IOS dengan Bahasa Pemrograman Swift memanfaatkan API Tiket. com. *Jurnal Manajemen Informatika*, 5(2), 2016.
- [3] Kurniawan, E. Implementasi Rest Web Service Untuk Sales Order dan Sales Tracking Berbasis Mobile. *Jurnal Eksplorasi Karya Sistem Informasi dan Sains*, 7(1), 2015.
- [4] Chen, M., Zhang, D. and Zhou, L, Providing web services to mobile users: the architecture design of an m-service portal. *International Journal of Mobile Communications*, 3(1), pp.1-18, 2005.
- [5] Prüter, S., Moritz, G., Zeeb, E., Salomon, R., Golatowski, F. and Timmermann, D., Applicability of web service technologies to reach real time capabilities. In *Object Oriented Real-Time Distributed Computing (ISORC)*, 2008 11th IEEE International Symposium on (pp. 229-233). IEEE, 2008.
- [6] Ahmadi, R., Heidari, E. and Zand, M, Security Enhancementfor Restful Web Services. *Journal of Fundamental and Applied Sciences*, 8(2S), pp.2804-2817, 2016.
- [7] Sinha, R., Khatkar, M. and Gupta, S.C., Design & Development of a REST based Web Service Platform for Applications Integration on Cloud. *International Journal of Innovative Science, Engineering and Technology*, 1(7), pp.385-389, 2014.
- [8] Zhou, W., Li, L., Luo, M. and Chou, W., REST API design patterns for SDN northbound API. In *Advanced Information Networking and Applications Workshops (WAINA)*, 2014 28th International Conference on (pp. 358-365). IEEE, 2014.
- [9] Kurniawan, Y.K., Oslan, Y. and Kristanto, H., Implementasi Rest-API Untuk Portal Akademik Ukdw Berbasis Android. *Jurnal Eksplorasi Karya Sistem Informasi dan Sains*, 6(2), 2015.
- [10] Lee, H.M. and Mehta, M.R., Defense against REST-based web service attacks for enterprise systems. *Communications of the IIMA*, 13(1), p.5, 2013.
- [11] Rahman, M.A., Kuswardayan, I. and Hariadi, R.R., Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android. *Teknik Informatika ITS*, 1(2), 2013
- [12] T. Fredrich, "RESTful Service Best Practices," 2012.
- [13] R. Sianipar, *Membangun Web PHP dan MySQL*, Informatika : Bandung, 2015.