

PENGEMBANGAN REST API SEBAGAI TEKNOLOGI INTEROPERABILITAS PADA APLIKASI UII TRAINING CENTER

Khairina Afifah¹, Hari Setiaji²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia
e-mail: ¹16523109@students.uui.ac.id, ²hari.setiaji@uui.ac.id

ABSTRAK

Universitas Islam Indonesia (UII) merupakan universitas yang selalu menjaga kredibilitas dalam segala aspek yang salah satunya adalah teknologi. UII saat ini tengah mengembangkan aplikasi berbasis web UII Training Center (UTC). Berbagai macam fakultas dan jurusan di UII tentu memiliki banyak pelatihan. Aplikasi UTC memiliki tujuan sentralisasi semua pelatihan di UII untuk mempermudah peserta dan organisasi dalam penyelenggaraan. Saat ini informasi pada web UTC hanya terbatas web. Belum berkembang aplikasi mobile UTC karena belum terdapat web service di dalamnya untuk sebuah komunikasi data. Jaman kini komunikasi data penting, maka dibangunlah REST API. Salah satu jenis web service untuk menjembatani komunikasi data agar para pengembang aplikasi lain dapat mengakses data web UTC untuk kebutuhan aplikasi mereka. Metode inkremental dan iteratif merupakan metode di mana tiap pengembangan fungsi baru harus melakukan pengulangan dari fase awal. Dengan metode iteratif, menggunakan teks resource JSON, framework Laravel, bahasa PHP dan basis data MySQL berhasil dikembangkan empat method GET dan dua POST sesuai dengan kebutuhan pengembang mobile. Method yang dihasilkan telah berhasil dalam tahap uji dan berhasil diimplementasikan pada aplikasi mobile. Dengan ini para pengembang aplikasi lain dapat dengan mudah mengakses data pada web UTC.

Kata Kunci: Interoperabilitas, JSON, API, REST.

1. PENDAHULUAN

Salah satu tren aplikasi masa kini adalah aplikasi yang memiliki kemampuan untuk bertukar informasi, atau dikenal dengan sebutan interoperabilitas. Interoperabilitas atau kemampuan berbagai jenis aplikasi untuk bertukar informasi merupakan hal yang penting. Aplikasi masa kini tidak bisa hanya terbatas pada satu platform, umumnya platform *web*. Faktanya, tren masa kini sangat berkaitan erat dengan *smartphone*, sebuah *smartphone* merupakan hal yang tidak bisa lepas dari kehidupan modern masa kini. Telah banyak aplikasi yang berubah dari *standalone* menjadi aplikasi yang dapat terhubung dengan aplikasi lain [1]. Dengan kemampuan aplikasi untuk berkomunikasi data, maka aplikasi platform *web* akan dapat diakses oleh aplikasi platform *mobile*. Dalam upaya merealisasikan kemudahan yang ditawarkan dengan adanya interoperabilitas, maka untuk membuka jalan dan memungkinkan kolaborasi data dibutuhkan sebuah *middleware*. *Middleware* merupakan sebuah jembatan untuk jalannya komunikasi data antar aplikasi pada platform yang memiliki antarmuka, sistem operasi, dan arsitektur yang berbeda [2].

Web Service merupakan salah satu *middleware*, yang memiliki fungsi untuk menjembatani hal pada pembahasan sebelumnya. Salah satu jenis dari *Web Service* adalah REST (*Representational State Transfer*). REST merupakan arsitektur di mana tidak memiliki standar notasi yang resmi, dan dalam hal transfer data REST lebih cepat dibanding *Web Service* lainnya [1]. Data yang disimpan berupa JSON atau XML, data ini yang dapat diakses oleh sistem lain walaupun berbeda platform. Hal ini menjadi penting karena aplikasi lintas *platform* merupakan suatu solusi untuk memperpanjang siklus perangkat lunak tersebut, dengan satu kali pengembangan data dapat dijalankan pada *platform* yang berbeda untuk menghemat tenaga para pengembang [3]. Hal ini juga memberikan manfaat bagi pengguna bahwa suatu kebutuhan menjadi lebih mudah terpenuhi dengan aplikasi *mobile*.

Salah satu aplikasi *web* baru di Universitas Islam Indonesia bernama UII Training Center (UTC), merupakan aplikasi dengan tujuan sentralisasi berbagai macam *training*/pelatihan yang berada di seluruh Fakultas Universitas Islam Indonesia. Dengan pengembangan yang terbilang masih baru, UTC belum sampai ke tahap pengembangan pada *platform mobile*. Dalam hal memperluas informasi maka perlu dilakukan pengembangan platform lain yaitu *mobile*. Informasi yang terdapat dalam *web* agar dapat diakses oleh platform lain membutuhkan jembatan, sebuah REST API sebagai sumber daya didapatnya informasi tersebut.

2. TINJAUAN PUSTAKA

2.1 Interoperabilitas

Suatu kemampuan berbagai jenis aplikasi, sistem informasi, jaringan untuk bertukar informasi dengan cara yang bermanfaat merupakan arti dari interoperabilitas berdasarkan KBBI. Masa kini dibutuhkan kemampuan sistem informasi yang andal untuk memenuhi kebutuhan layanan sistem elektronik. Di Indonesia peraturan interoperabilitas

secara khusus tertuang dalam PP 82 Tahun 2012, Pasal 23 yang berbunyi: Penyelenggaraan Sistem Elektronik harus menjamin berfungsinya Sistem Elektronik sesuai dengan peruntukannya, dengan tetap memperhatikan interoperabilitas dan kompatibilitas dengan Sistem Elektronik sebelumnya dan/atau Sistem Elektronik yang terkait [4]. Hal ini merupakan bukti serius bahwa di Indonesia sedang terjadi peningkatan sistem informasi yang lebih baik lagi. Keadaan ini tidak hanya terjadi di sektor pemerintahan namun, para pengembang aplikasi lain juga turut mengembangkan teknologi untuk mengatasi interoperabilitas pada aplikasi mereka [5].

2.2 JSON

JavaScript Object Notation (JSON) adalah salah satu format pertukaran data yang ringan. JSON bersifat mudah untuk dibaca dan ditulis oleh manusia, diterjemahkan dan dibuat (*generate*) oleh komputer. Format teks JSON tidak bergantung pada suatu bahasa pemrograman tertentu [6]. Penerapan JSON banyak digunakan pada basis data dan *web service*.

2.3 API

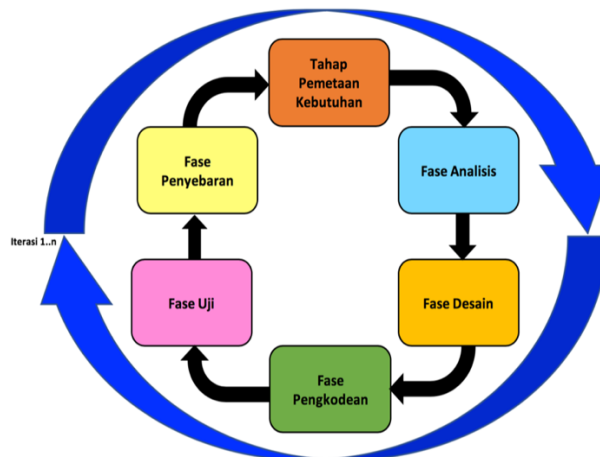
Application Programming Interface (API) merupakan kumpulan fungsi yang dapat dijalankan oleh program lain. API dapat dirancang dengan banyak cara dan teknologi namun, dalam konteks aplikasi *web* maka perlu melalui protokol HTTP yang memiliki mekanisme yang mudah diimplementasikan, di *maintenance*, dan distandardisasi [7]. Di sinilah REST menjadi jalan keluar, REST adalah gaya desain untuk sistem *hypermedia* terdistribusi [8].

2.4 REST

REST adalah singkatan dari Representational State Transfer merupakan desain arsitektur API yang diperkenalkan oleh Fielding pada tahun 2000 [8]. Cara kerja REST berkomunikasi antar platform, *client* mengirim perintah atau *request* melalui *HTTP Request* kemudian *server* merespon melalui *HTTP Response*. Resource yang dihasilkan berupa teks dalam format yang bermacam-macam, namun umumnya adalah JSON atau XML. Terdapat empat metode *HTTP Request* yang bisa digunakan pada REST yaitu GET, POST, PUT dan DELETE [9].

3. METODE PENELITIAN

Proses pengembangan *web service* cenderung dilakukan secara berulang dan bersifat inkremental di mana fase-fase yang dikerjakan tidak berupa proses *single-pass*. Dalam setiap iterasi, *web service* akan melewati semua fase. Terdapat enam fase siklus implementasi *web service* (Gambar 1) yang pertama tahap pemetaan kebutuhan, fase analisis, fase desain, fase pengkodean, fase uji, dan fase penyebaran [10].



Gambar 1. Siklus Implementasi Web Service.

Implementasi metodologi *web service* bersifat iteratif dan inkremental. Iterasi terjadi pada semua fase [10]. Ketika satu fungsi telah berhasil dikembangkan hingga dapat diimplementasikan kepada pengembang aplikasi *mobile* (fase penyebaran) maka pengembangan fungsi selanjutnya akan dimulai dari fase awal yaitu fase pemetaan. Apabila pada pengembangan fungsi pertama fase pemetaan telah dilakukan untuk semua fungsi, maka dilakukan pengecekan ulang apabila terdapat perubahan pada basis data aplikasi. Ketika terdapat perubahan pada basis data maka akan berpengaruh pada semua fase setelahnya. Hal ini yang menyebabkan iterasi harus terjadi pada semua fase.

4. HASIL DAN PEMBAHASAN

Pembuatan fungsi secara inkremental menghasilkan empat GET dan dua POST pada aplikasi *web* UTC seperti pada Gambar 2. Fungsi yang dibangun akan diuji dengan aplikasi Postman. Apabila berhasil maka

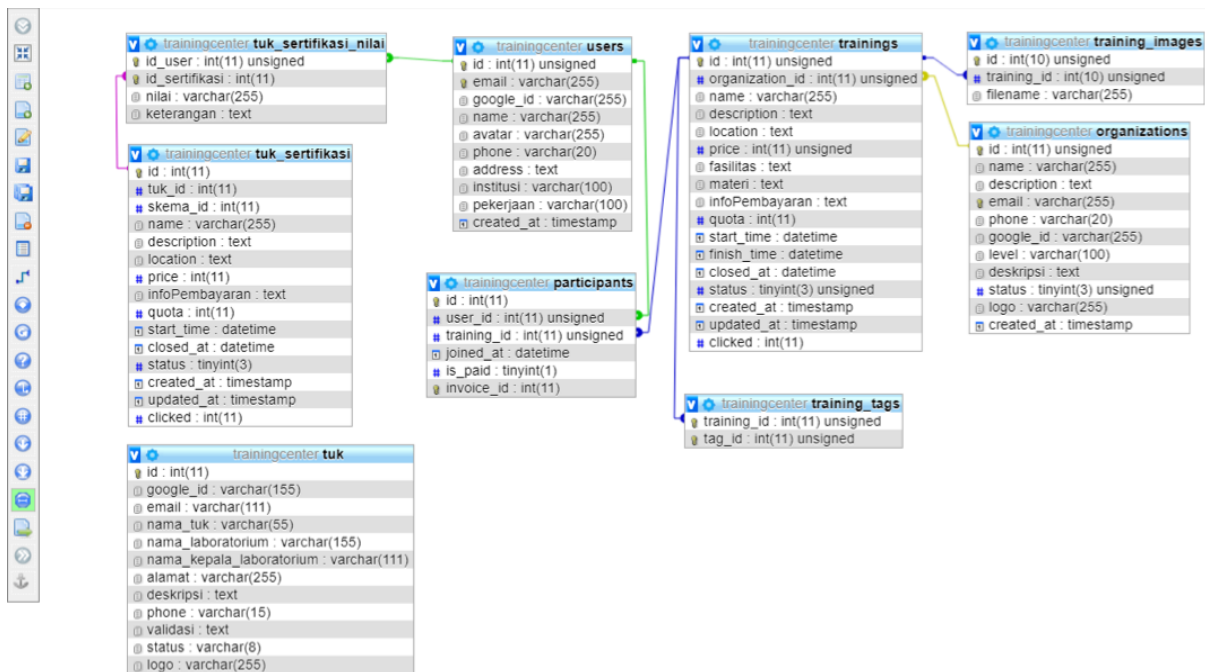
selanjutnya akan didokumentasikan pada sebuah katalog REST API. Sebuah katalog dibangun dengan Laravel akan berisi fungsi-fungsi dari REST API yang telah berhasil dikembangkan. Dengan ini pengembang aplikasi *mobile* dapat mengakses katalog untuk mengambil fungsi yang dibutuhkan.

```
Route::get('/trainings', 'TrainingController@index');
Route::get('/trainings/{id}', 'TrainingController@show');
Route::get('/users', 'UsersController@index');
Route::get('/users/{id}', 'UsersController@show');
Route::post('/users', 'UsersController@createUser');
Route::post('/participants', 'ParticipantsController@create');
```

Gambar 2. GET dan POST.

4.1 Tahap Pemetaan Kebutuhan

Tahap ini dilakukan untuk mengetahui dan memahami kebutuhan bisnis yang selanjutnya akan diterjemahkan ke dalam *web service*. Pemetaan kebutuhan atau seleksi persyaratan pada tahap ini dilakukan dengan metode pemetaan terhadap basis data *web* UTC. Pemetaan bertujuan agar fungsi REST API yang akan dikembangkan tidak lepas dari kebutuhan pengembang aplikasi *mobile*. Apabila memungkinkan, kebutuhan yang telah dipetakan dapat disentralisasikan dan disimpan agar memudahkan pengembang API selanjutnya [10]. Terdapat tiga fitur pada *web* UTC, yaitu Log In, Pendaftaran Pelatihan, dan LSP. Tidak semua fungsi pada ketiga fitur membutuhkan API maka dilakukan pemetaan fungsi berdasarkan skema basis data *web* pada Gambar 3 dan ditentukan prioritas pada fase berikutnya.



Gambar 3. Skema Basis Data.

4.2 Fase Analisis

Analisis bertujuan memahami lebih dalam mengenai fungsi yang telah diidentifikasi pada proses sebelumnya dan membantu dalam menentukan prioritas pengembangan fungsi. Pada tahap ini dilakukan kerja sama dengan salah satu pengembang UTC *mobile* untuk menentukan fungsi yang terpenting untuk dijadikan prioritas pada pengembangan API. Analisis juga mengidentifikasi, membuat konsep awal, dan merasionalkan proses sebagai rangkaian fungsi yang saling berinteraksi [11]. Pada fase ini fungsi yang telah selanjutnya disempurnakan dan diterjemahkan ke model konseptual. Juga dalam fase ini dilakukan analisis arsitektur dan mengidentifikasi antarmuka *web service* [10]. Tabel 1 memuat detail fitur beserta fungsi pada *web* UTC dan prioritas pengembangan.

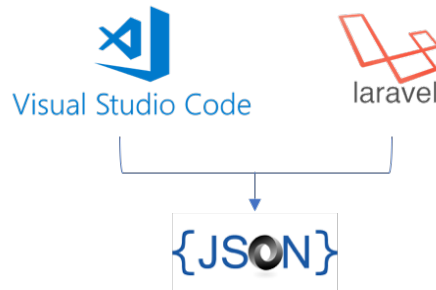
Tabel 1. Pemetaan Fungsi pada Fitur.

Fungsi yang Dipetakan	Fungsi Prioritas	Fungsi yang Telah Dikembangkan
-----------------------	------------------	--------------------------------

Fungsi yang Dipetakan	Fungsi Prioritas	Fungsi yang Telah Dikembangkan
1. LSP -> menampilkan daftar sertifikasi.	1. Trainings -> menampilkan daftar pelatihan dan melakukan pendaftaran pelatihan, menampilkan pelatihan sesuai dengan id.	1. Trainings -> menampilkan daftar pelatihan, melakukan pendaftaran pelatihan, menampilkan pelatihan sesuai id.
2. Trainings -> menampilkan daftar pelatihan, melakukan pendaftaran pelatihan, menampilkan pelatihan sesuai dengan id, organisasi menambahkan pelatihan.	2. Users -> melakukan <i>log in</i> , mengisi identitas diri, menampilkan keseluruhan <i>user</i> dan berdasarkan id.	2. Users -> melakukan <i>login</i> , menampilkan keseluruhan <i>user</i> dan berdasarkan id.
3. Users -> melakukan <i>log in</i> , mengisi identitas diri, menampilkan keseluruhan <i>user</i> dan berdasarkan id.	3. LSP -> menampilkan daftar sertifikasi.	

4.3 Fase Desain

Sebelum membangun *web service*, pengembang harus menentukan spesifikasi pengembangan untuk fungsi yang telah diidentifikasi pada fase analisis [11]. Spesifikasi *framework*, gaya pemrograman, dan spesifikasi *tools* yang digunakan. Selain itu diperlukan juga desain untuk pendokumentasian fungsi yang telah berhasil dikembangkan. Pada aplikasi UTC tahap desain dilakukan dengan menentukan *framework* yang digunakan adalah Laravel, *tools* Visual Studio Code (VSC), dan format teks *resource* yang dihasilkan berupa JSON dengan visualisasi pada Gambar 4.



Gambar 4. Desain.

4.4 Fase Pengkodean

Pada dasarnya fase ini sama seperti fase pengembangan aplikasi pada umumnya. Satu perbedaan sederhana terletak pada penciptaan antarmuka, di mana *web service* tidak memiliki *Graphical User Interface (GUI)* seperti aplikasi pada umumnya. Tahap pengkodean dikerjakan sesuai dengan kriteria menggunakan bahasa pemrograman, basis data, *framework* yang telah didesain pada tahap sebelumnya. Sejatinya, pengkodean dapat dilakukan dengan bahasa yang beragam namun tetap harus mempertimbangkan tipe data pada bahasa tersebut [10]. Tahap ini sesuai dengan yang telah didesain dengan menggunakan bahasa PHP (Gambar 5).

```

class TrainingResource extends Resource
{
    /**
     * Transform the resource into an array.
     *
     * @param \Illuminate\Http\Request $request
     * @return array
     */
    public function toArray($request)
    {
        return [
            'id' => $this->id,
            'name' => $this->name,
            'description' => $this->description,
            'location' => $this->location,
            'price' => $this->price,
            'fasilitas' => $this->fasilitas,
            'materi' => $this->materi,
            'quota' => $this->quota,
            'start_time' => $this->start_time,
            'finish_time' => $this->finish_time,
            'closed_at' => $this->closed_at,
            'status' => $this->status,
            'created_at' => $this->created_at->format('d F Y H:i:s'),
            'updated_at' => $this->updated_at->format('d F Y H:i:s'),
            'filename' => $this->training_images->filename,
            'organization' => [
                'name' => $this->organization->name,
                'id' => $this->organization->id
            ]
        ];
    }
}

class UsersResource extends Resource
{
    /**
     * Transform the resource into an array.
     *
     * @param \Illuminate\Http\Request $request
     * @return array
     */
    public function toArray($request)
    {
        return [
            'id' => $this->id,
            'email' => $this->email,
            'google_id' => $this->google_id,
            'nama' => $this->name,
            'avatar' => $this->avatar,
            'telepon' => $this->phone,
            'alamat' => $this->address,
            'institusi' => $this->institusi,
            'pekerjaan' => $this->pekerjaan,
            'created_at' => $this->created_at
        ];
    }
}
    
```

Gambar 5. Kode Program Users dan Training.

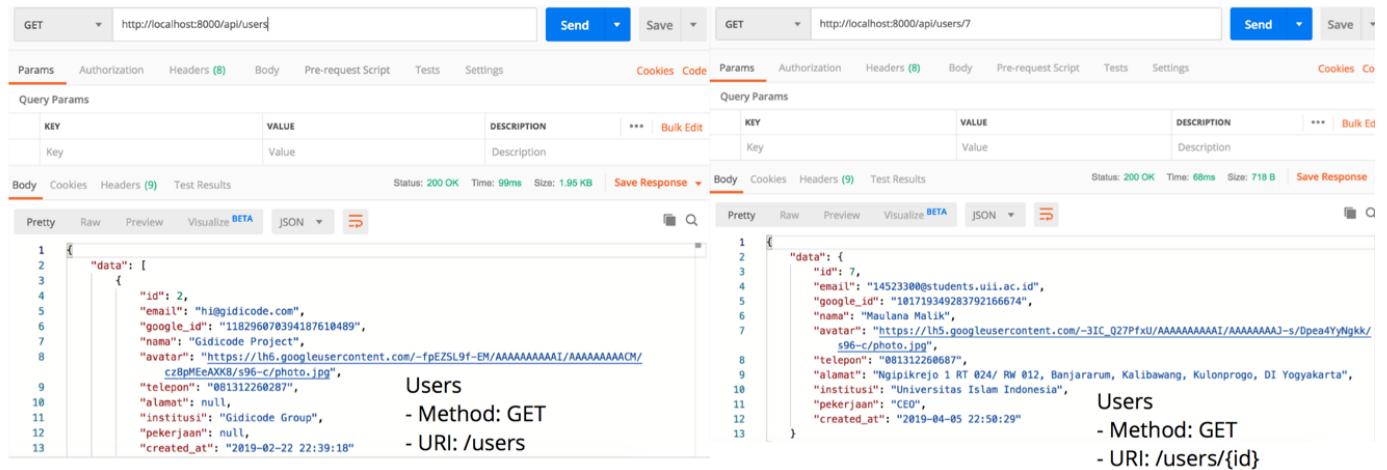
4.5 Fase Uji

Pengujian melibatkan pelaksanaan rangkaian tugas dan membandingkan hasil dengan *output* yang diharapkan. Menerapkan tes fungsional di awal fase pengembangan berguna untuk mempercepat pengembangan, meningkatkan kualitas, dan mengurangi risiko pada akhir fase [12]. Pengujian menggunakan aplikasi Postman dengan memasukkan dua *method* yang dibangun yaitu GET pada Gambar 6 dan Gambar 7 dan POST pada Gambar 8.

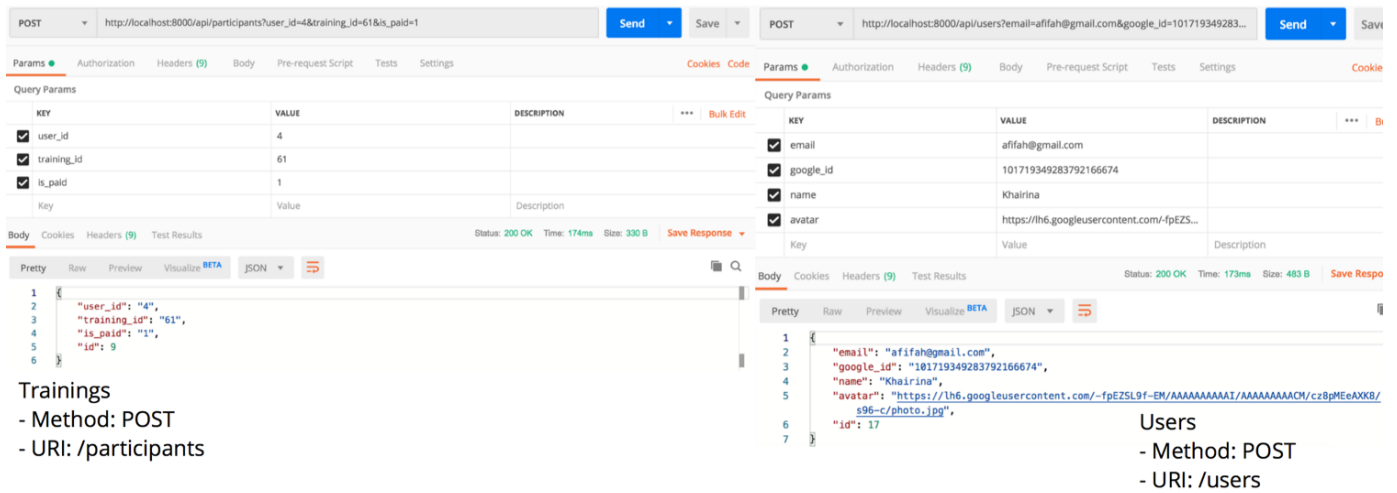
Trainings
- Method: GET
- URI: /trainings

Trainings
- Method: GET
- URI: /trainings/{id}

Gambar 6. GET Trainings.



Gambar 7. GET Users.



Gambar 8. POST Participants dan Users.

4.6 Fase Penyebaran

Fase penyebaran merupakan fase diujinya sebuah fungsi pada salah satu aplikasi UTC basis *mobile* seperti Gambar 9, Gambar 10, dan Gambar 11. Tujuannya adalah untuk memastikan bahwa fungsi yang telah diuji dengan Postman dapat digunakan sesuai kebutuhan pengembang *mobile*. Fungsi yang disebar pada *mobile* ini masih bersifat *private* dengan salah satu pengembang *mobile* yang bekerja sama dengan pengembang REST API. Ketika semua fungsi telah berhasil diimplementasikan pada aplikasi UTC *mobile* maka fungsi tersebut akan didokumentasikan pada sebuah sistem berbentuk katalog yang kemudian akan disebar untuk dapat digunakan para pengembang lain [10].

```
public interface ApiInterface {
    @GET("trainings")
    Call<TrainingRespon> getTrainings();

    @FormUrlEncoded
    @POST("users")
    Call<Users> saveUsers(
        @Field("email") String email,
        @Field("google_id") String google_id,
        @Field("name") String name,
        @Field("avatar") Uri avatar
    );
}
```

Gambar 9. Kode Program ApiInterface pada *Android Mobile* untuk Membuat Perintah *Get Data* dan *Insert Data*.

```
private void parseJson() {
    ApiInterface api = ApiService.getClient().create(ApiInterface.class);
    Call<TrainingRespon> getDataTraining = api.getTrainings();

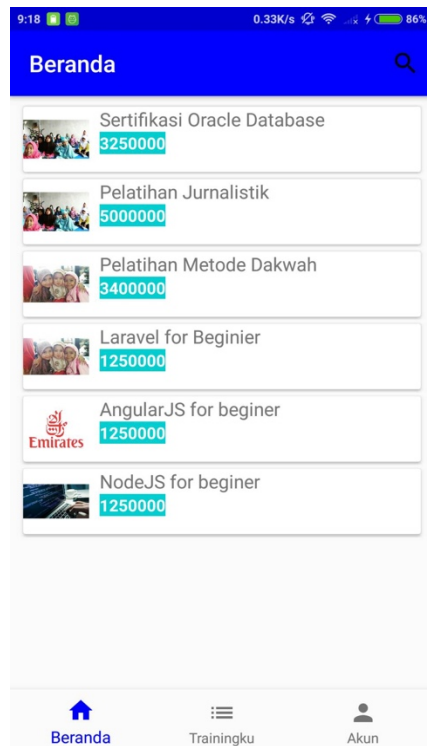
    getDataTraining.enqueue(new Callback<TrainingRespon>() {
        @Override
        public void onResponse(Call<TrainingRespon> call, Response<TrainingRespon> response) {
            lstTraining = response.body().getTrainingList();
            mTrainingAdapter = new TrainingAdapter(lstTraining, getContext());
            mRecyclerView.setAdapter(mTrainingAdapter);

            mTrainingAdapter.setOnItemClickListener(BerandaFragment.this);
        }

        @Override
        public void onFailure(Call<TrainingRespon> call, Throwable t) {

        }
    });
}
```

Gambar 10. Kode Program pada *Android Mobile* untuk Menampilkan *Training*.



Gambar 11. Tampilan pada *Android Mobile*.

5. KESIMPULAN

Metode implementasi *web service* secara inkremental dan iteratif yang terjadi di tiap fungsi memungkinkan untuk hasil yang lebih baik. Hasil yang diperoleh meliputi empat GET yang terdiri dari :

- /trainings : menampilkan pelatihan secara keseluruhan.
- /trainings/{id} : menampilkan pelatihan sesuai id.
- /users : menampilkan identitas peserta secara keseluruhan.
- /users/{id} : menampilkan identitas peserta sesuai id.

Terdapat juga dua POST, yaitu :

- /users : melakukan log in.

- /participants : melakukan pendaftaran pelatihan.

Hasil tersebut telah berhasil diuji dan telah diimplementasikan pada salah satu aplikasi *mobile* UTC dengan tujuan mendapatkan hasil maksimal dan sesuai kebutuhan. Apabila terdapat kekurangan maka akan segera dilakukan tahap perbaikan sebelum dilakukan penyebaran secara umum. Metode inkremental dan iteratif memiliki kelebihan mendapatkan hasil yang lebih baik dan bertujuan menghindari kesalahan fatal, karena fase yang dikerjakan tidak berupa proses *single-pass*. Dalam setiap iterasi, *web service* akan melewati semua fase. Namun metode tersebut juga memiliki kekurangan di mana waktu yang dibutuhkan lebih lama karena tiap fase yang dikerjakan harus mengulang dari fase awal. Dengan segala hasil dan keberhasilan yang telah dicapai, pengembangan REST API pada *web* UTC akan terus berlanjut seiring dengan perkembangan dari *web* UTC itu sendiri.

DAFTAR PUSTAKA

- [1] Rulloh dkk., “Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android,” *Tek. Vol. I No. 2*, vol. 1, no. 2, pp. 85–89, 2017.
- [2] A. Farahzadi, P. Shams, J. Rezazadeh, and R. Farahbakhsh, “Middleware technologies for cloud of things: a survey,” *Digit. Commun. Networks*, vol. 4, no. 3, pp. 176–188, 2018.
- [3] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, “Taxonomy of Cross-Platform Mobile Applications Development Approaches,” *Ain Shams Eng. J.*, vol. 8, no. 2, pp. 163–190, 2017.
- [4] PeraturanPemerintah, “PERATURAN PEMERINTAH REPUBLIK INDONESIA NOMOR 82 TAHUN 2012 TENTANG PENYELENGGARAAN SISTEM DAN TRANSAKSI ELEKTRONIK,” 2012. [Online]. Available: <http://www.jdih.kemenkeu.go.id/fullText/2012/82TAHUN2012PP.HTM>.
- [5] Gamatechno, “Pentingnya Penerapan Interoperabilitas dalam Pemerintahan,” 2019. [Online]. Available: <https://blog.gamatechno.com/penerapan-interoperabilitas-dalam-pemerintahan/>.
- [6] D. Crockford, “Pengenalan JSON - How JavaScript Works.” [Online]. Available: <https://www.json.org/json-id.html>.
- [7] J. Hradil, “Practical Implementation of 10 Rules for Writing REST APIs,” pp. 45–54, 2017.
- [8] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” 2000.
- [9] M. I. Perkasa and E. B. Setiawan, “Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token,” vol. X, no. 1, pp. 19–26, 2018.
- [10] P. R. Draft, “Web Service Implementation Methodology,” no. September, pp. 1–35, 2005.
- [11] M. P. Papazoglou, “Chapter 15 Web Services Development Lifecycle,” 2008.
- [12] AppPerfectCorporation, “WEB SERVICES FUNCTIONAL TESTING.” [Online]. Available: <http://www.appperfect.com/services/web-testing/web-services-functional-testing.php>.