

## PERTUKARAN DATA PADA APLIKASI ANDROID MENGUNAKAN JAVA SCRIPT OBJECT NOTATION (JSON) DAN REST API DENGAN RETROFIT 2

Ester Lumba<sup>1</sup>

<sup>1</sup>Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis  
e-mail : <sup>1</sup>114242@lecturer.kalbis.ac.id

### ABSTRACT

*Developing Android-based applications has its own challenges. Besides requiring hardware devices with quite high specifications, they must also learn various programming languages. On the client side the programmer must understand XML for layout and Java for programming logic. On the server side must master PHP and SQL for data processing. To exchange data, the developer must also master the commonly used data formats such as Java Script Object Notation (JSON). Developing an Android application for reporting lecturer activities to exchange data from client to server and vice versa, having to handle aspects of the HTTP method. This study aims to examine the process of exchanging data on Android applications using the Java Script Object Notation and REST API with the Retrofit 2 library. The research method used is the Extreme Programming (XP).*

**Keywords:** API, JSON, REST, Retrofit 2

### INTISARI

*Mengembangkan aplikasi berbasis Android memiliki tantangan tersendiri. Selain membutuhkan perangkat hardware dengan spesifikasi yang cukup tinggi, juga harus mempelajari berbagai bahasa pemrograman. Disisi client programmer harus memahami XML untuk layout dan Java untuk logika programingnya. Di sisi server harus menguasai PHP dan SQL untuk pengolahan data. Untuk melakukan pertukaran data maka pengembang juga harus menguasai format data yang umum dipakai seperti Java Script Object Notation (JSON). Mengembangkan aplikasi Android untuk pelaporan kegiatan dosen untuk melakukan pertukaran data dari client ke server dan sebaliknya harus menangani aspek-aspek pada metode HTTP.*

*Penelitian ini bertujuan untuk mengkaji proses pertukaran data pada aplikasi Android menggunakan Java Script Object Notation dan REST API dengan library Retrofit 2. Metode penelitian yang digunakan adalah Extreme Programming (XP).*

**Kata kunci:** API, JSON, REST, Retrofit 2

## 1. PENDAHULUAN

Penggunaan perangkat *mobile* semakin hari semakin meningkat pesat, bagi programmer atau perusahaan yang bergerak dibidang TI, tentu memiliki tantangan tersendiri dalam menyediakan aplikasi-aplikasi dan solusi berbasis *mobile*. Pada laporan khusus situs <https://websindo.com/> dalam topik Indonesia Digital 2019: Mobile menunjukkan bahwa *platform mobile* terus berkembang dan diminati masyarakat. Seiring dengan banyaknya produsen-produsen telepon pintar, pencipta OS *mobile*, dan seluruh developer aplikasi *mobile* di seluruh dunia. Terutama kepada tim pengembang Android, menjadikan ponsel-ponsel kelas menengah ke bawah menjadi smartphone yang handal dan murah. Hal ini sangat berperan dalam penetrasi pengguna teknologi *mobile* di Indonesia. Total jumlah perangkat seluler yang tersebar di Indonesia mencapai 355,5 juta. Jumlah perangkat seluler jauh lebih banyak dari pada jumlah penduduk di Indonesia, yaitu 133%. Dari jumlah pengguna seluler tersebut aktivitasnya adalah sebesar 96% untuk melakukan akses ke internet.

Mengembangkan aplikasi piranti bergerak seperti Android memiliki tantangan tersendiri. Pengembangan membutuhkan perangkat keras dengan spesifikasi yang tinggi, dan harus menggunakan berbagai bahasa pemrograman. Disisi klien programmer harus memahami XML untuk layout dan Java untuk logika programmingnya. Di sisi server harus menguasai PHP dan SQL untuk melakukan pengolahan data.

Sebagai acuan dalam penelitian penulis melakukan kajian pustaka pada jurnal-jurnal dari beberapa penelitian terkait sebelumnya. Seperti penelitian yang dilakukan oleh Jiri Hradil, Vilém Sklenak dengan judul “Practical Implementation of 10 Rules for Writing REST APIs”. Penelitian ini melakukan implementasi dari 10 aturan REST API yang diterapkan pada aplikasi bisnis kecil yang dapat diakses dari seluruh dunia. API diterapkan dalam format JSON. Tujuan utama pembuatan API adalah untuk mempermudah pengguna aplikasi menerbitkan faktur agar dapat dikirimkan kepada pelanggan (Jiri Hradil, Vilém Sklenak, 2017). Penelitian selanjutnya dilakukan oleh Jaydevi Bhade dan Prof. Himanshu Yadav dengan judul “Evaluation of Android Networking Libraries”. Penelitian ini melakukan analisis pada empat library yaitu *HTTP Client*, *Volley*, *Retrofit* and *Fast Android Networking*. Peneliti melakukan secara komprehensif pada keempat library dibuat dalam bentuk matriks (Jaydevi Bhade, Himanshu Yadav, 2019).

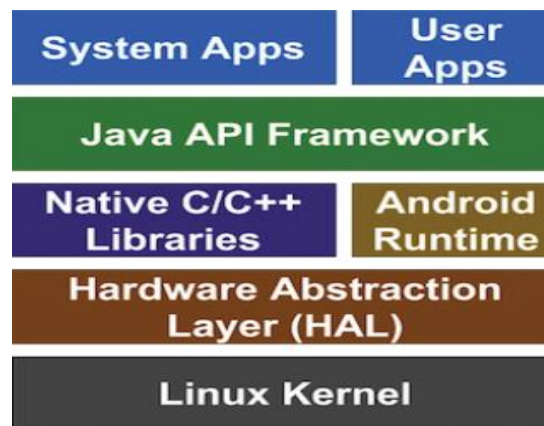
Untuk melakukan pertukaran data maka pengembang juga harus menguasai format data yang umum dipakai seperti *Java Script Object Notation*. Mengembangkan pustaka HTTP sendiri agar aplikasi dapat melakukan pertukaran data dari klien ke server dan dari server ke klien harus menangani banyak hal, misalnya membuat koneksi, *caching*, mengulangi permintaan yang gagal, *parsing respons* dan *threading* serta penanganan kesalahan.

Berdasarkan permasalahan tersebut maka tim peneliti melakukan penelitian tentang pertukaran data pada aplikasi Android menggunakan Java Script Object Notation dan REST API dengan library Retrofit 2. Adapun pertanyaan penelitian ini adalah “Bagaimana melakukan pertukaran data menggunakan JSON dan REST API dengan Retrofit 2 pada Aplikasi Android?”

Tujuan dari penelitian ini adalah membangun aplikasi Android untuk melakukan pertukaran data dari klien ke server menggunakan JSON dan REST API dengan Library Retrofit 2.

Android merupakan sistem operasi *mobile* yang *open-source*. Sistem operasi ini awalnya dibuat atau dikembangkan oleh Android, Inc. Tetapi pada tahun 2005, diakuisisi dan dikembangkan oleh Google (Dimarzio, 2017). OS Android digunakan untuk komputer tablet dan *smartphone*. Di Palo Alto, Andy Rubin, Rich Miner, Nick Sears, dan Chris White mendirikan Android pada bulan Oktober 2003. Google mengakuisisi Android Inc. pada 17 Agustus 2005, akan tetapi Rubin, White dan Miner tetap bekerja di perusahaan tersebut.

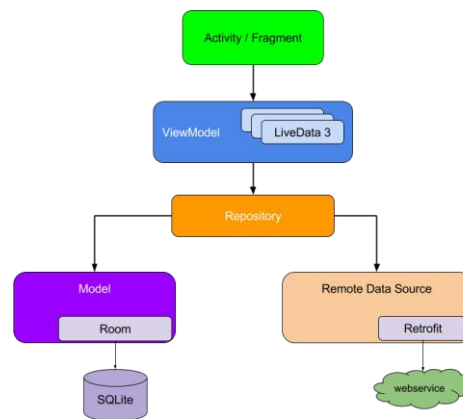
Android dapat diibaratkan sebagai tumpukan perangkat lunak. Pada setiap lapisan dari tumpukan, terdiri dari program-program yang mendukung fungsi spesifik dari suatu sistem operasi (Dimarzio, 2017). Gambar 1 menunjukkan arsitektur dari Android.



**Gambar 1.** Lapisan arsitektur aplikasi Android (Team G. D., 2016)

Linux Kernel merupakan fondasi dari platform Android dan berada di tumpukan paling bawah pada arsitektur Android. Untuk membangun sistem Android, Google menggunakan kernel Linux versi 2.6. Kernel ini mencakup *memory management*, *power management*, *security setting* dan beberapa dari *driver hardware*. Lapisan selanjutnya adalah Hardware Abstraction Layer (HAL). Lapisan ini terdiri dari beberapa pustaka untuk penerapan antarmuka komponen perangkat keras seperti modul kamera atau bluetooth. Selanjutnya adalah lapisan Native C/C++ dan *Android Runtime*. Setiap aplikasi di Android akan berjalan dalam proses sendiri-sendiri dan dengan instance Android Runtime. Layanan dan komponen sistem Android dibangun dari pustaka asli yang ditulis dalam C dan C++. Lapisan Java API *Framework* merupakan tool dasar bagi developer yang ditulis menggunakan bahasa Java. Developer tidak perlu mengetahui secara detail mengenai API untuk keperluan pengembangan aplikasi Android. API ini digunakan untuk membangun UI dari aplikasi seperti tombol, menu dan komponen UI lainnya. API juga digunakan untuk alokasi *resource* dari telepon pintar, mengatur pergantian antar proses dan mengatur aksesibilitas data pada aplikasi lain. Lapisan bagian atas terdiri dari dua bagian yaitu *System Apps* dan *User Apps*. *System Apps* ini merupakan aplikasi bawaan dari *smartphone* seperti aplikasi email, SMS, kalender, browser internet dan kontak. Sedangkan *User Apps* adalah aplikasi yang dibangun oleh para pengembang aplikasi Android.

Retrofit 2 merupakan pustaka klien HTTP yang aman untuk aplikasi Android dan Java. Retrofit digunakan sebagai REST klien. Dengan retrofit tidak perlu membuat metode-metode sendiri untuk terhubung ke layanan web REST. GSON adalah pustaka untuk mengubah objek Java ke dalam format JSON dan sebaliknya dari format JSON ke objek Java. Retrofit 2 mempermudah proses pertukaran data dari format JSON atau XML, lalu diurai menjadi Plain Old Java Objects (POJOs) (JournalDev, 2020). Proses permintaan berupa metode POST, GET, PUT, DELETE dan PATCH pada arsitektur REST ini dapat dieksekusi melalui pustaka retrofit. Retrofit 2 dibangun dari beberapa pustaka-pustaka dan alat bantu lainnya. Pustaka Retrofit 2 ini memanfaatkan protokol OkHttp yang dikembangkan oleh developer <https://squareup.com/> untuk menangani proses transmisi data (permintaan) melalui jaringan (*network*). Arsitektur aplikasi Android yang memanfaatkan pustaka Retrofit ditunjukkan pada Gambar 2 berikut ini:

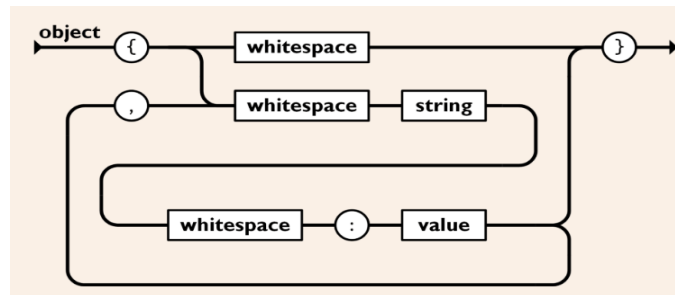


**Gambar 2.** Arsitektur aplikasi Android dengan library Retrofit (Team D. , 2021)

JSON merupakan singkatan dari *JavaScript Object Notation* digunakan sebagai sarana untuk menyimpan dan mentransfer data secara terstruktur melalui media jaringan. JSON memiliki struktur data yang sederhana, mudah dibaca, dipahami dan mudah di-generate oleh program komputer. Kemudahan yang ditawarkan oleh JSON ini menjadi alasan mengapa sering digunakan pada pembuatan API. Format JSON ini merupakan subset dari Pemrograman JavaScript, menurut Standar ECMA-262 Edisi ke-3 - Desember 1999 (International, 2021). Format data JSON tidak bergantung pada bahasa pemrograman apapun seperti C/C++, Java, Python C# dan sebagainya. JSON memiliki dua struktur yaitu kumpulan pasangan *key/value* dan daftar nilai yang berurutan seperti pada array (Smith, 2015). Objek dimulai dengan dengan kurung kurawal buka ({} dan pada bagian akhir ditutup dengan tanda kurung kurawal penutup (}). Berikut contoh representasi dari format data JSON untuk mendeskripsikan objek dosen.

```
{
  "nidn": "12345",
  "nama": "Blessy",
  "mengajar": {
    "kodeMtk": "IF-001",
    "namaMtk": "Pemrograman Mobile",
    "sks": "3"
  },
  "mobilePhone": [
    "0813 4443-3110",
    "0819 7778-8118"
  ]
}
```

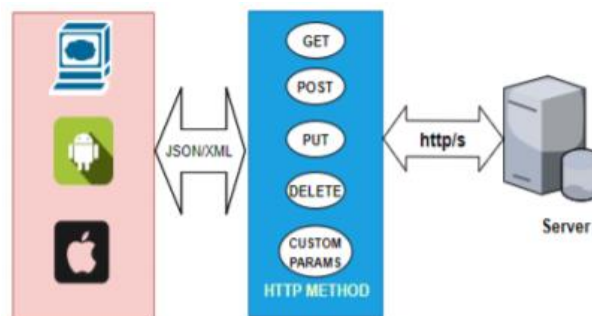
Pada struktur data JSON tersebut setiap nama diikuti dengan tanda titik dua (:) dan setiap pasangan *key/value* dipisahkan dengan tanda koma yaitu (,). Bentuk umum dari struktur data objek JSON ditunjukkan pada Gambar 3.



Gambar 3. Representasi struktur objek JSON (Developer, 2021)

Representational State Transfer atau biasa disingkat dengan REST merupakan standar dari arsitektur berbasis web. REST menggunakan protokol HTTP untuk komunikasi data.

Pada arsitektur REST, di sisi server menyediakan *resources* atau data dan di sisi klien mengakses dan menampilkan sumber daya tersebut. Pada setiap *resource* diidentifikasi melalui URIs (*Universal Resource Identifiers*) atau global ID (Doglio, 2015). JSON atau XML digunakan untuk merepresentasikan sumber daya pada REST. Gambar 4 menunjukkan cara kerja REST API pada aplikasi web.



Gambar 4. Cara kerja REST (Anshu Soni, Virender Ranga, 2019)

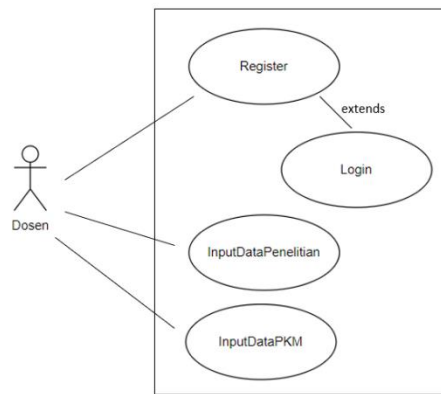
Secara umum metode HTTP yang dipakai dalam REST API ditunjukkan pada Gambar 4 di atas. Metode GET biasanya digunakan untuk membaca sumber daya dari REST server. Metode POST, biasanya digunakan untuk membuat sumber daya baru pada REST server. Sedangkan metode PUT, umumnya digunakan untuk memperbaharui sumber daya pada REST server. Kemudian metode DELETE, digunakan untuk menghapus sumber daya dari REST server. Yang terakhir adalah metode OPTIONS (CUSTOM PARAMS), biasanya digunakan untuk mendapatkan operasi yang didukung oleh sumber daya di REST server.

## 2. METODE PENELITIAN

Metode penelitian diawali dengan melakukan studi pustaka baik menggunakan buku-buku maupun jurnal yang terkait dengan judul penelitian dan observasi langsung pada objek penelitian. Metode pengembangan aplikasi menggunakan *Extreme Programming (XP)*. Studi kasus yang digunakan adalah aplikasi pelaporan kegiatan dosen yaitu pengabdian dan penelitian yang dilakukan setiap semester atau setiap tahun. XP adalah salah satu metode dalam pengembangan perangkat lunak. Metode XP memiliki ciri-ciri yaitu cepat, efisien, beresiko rendah, fleksibel,

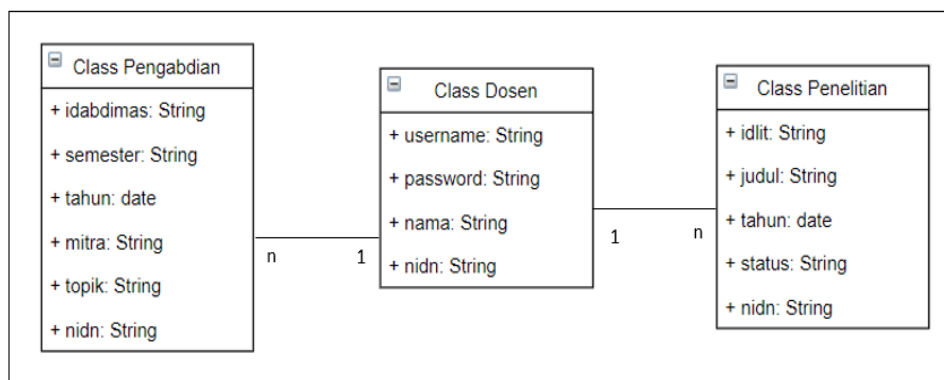
terprediksi dan *scientific* (Kent Beck, 2004). Model ini menggunakan pendekatan *Object-Oriented*. Pada metode *extreme programming* dapat membentuk tim dengan ukuran kecil dan sedang. Tujuannya untuk menangani persyaratan-persyaratan yang tidak jelas dan jika terjadi perubahan-perubahan persyaratan-persyaratan yang sangat cepat. Pada metode *extreme programming* ini terdapat 5 tahapan yang harus dilakukan oleh pengembang.

Pada tahap pertama yaitu *planning*, tahap ini peneliti membuat *use case diagram* untuk aplikasi pelaporan kegiatan dosen. Dalam hal ini peneliti hanya menggunakan dua kegiatan yaitu Pengabdian Kepada Masyarakat (PKM) dan Penelitian dosen. Interaksi dosen dengan aplikasi ditunjukkan pada *use case diagram* seperti Gambar 5 berikut ini:



Gambar 5. Use Case Diagram

Tahap selanjutnya adalah perancangan (*design*). Pada tahap ini peneliti membuat *class diagram* dan juga sekaligus representasi database yang digunakan pada penelitian ini. *Use case diagram* dapat dilihat pada Gambar 6.



Gambar 6. Use Case Diagram

Tahap ketiga dari *extreme programming* adalah pengkodean (*coding*). Pada tahap ini tim peneliti membuat kode program menggunakan bahasa Java untuk logika program dan XML sebagai bahasa desain layout. Peran library retrofit 2 dilakukan pada tahap ini. Peneliti membuat sebuah interface BaseApiService untuk

menjembatani antara aplikasi Android dengan database. Pada library Retrofit 2 dilakukan pertukaran data menggunakan JSON. Interface ini yang mendefinisikan metode POST dan form URL encode milik protokol http dari library Retrofit. Interface BaseAPIService ditunjukkan pada Gambar 7 berikut ini:

```
4 import okhttp3.ResponseBody;
5 import retrofit2.http.Field;
6 import retrofit2.http.FormUrlEncoded;
7 import retrofit2.http.GET;
8 import retrofit2.http.POST;
9 import retrofit2.Call;
10 import retrofit2.http.Query;
11
12
13 public interface BaseAPIService {
14     @FormUrlEncoded
15     @POST("login.php")
16     Call<ResponseBody> loginRequest(@Field("username") String username,
17                                   @Field("password") String password);
18
19     @FormUrlEncoded
20     @POST("register.php")
21     Call<ResponseBody> registerRequest(@Field("username") String username,
22                                      @Field("password") String password,
23                                      @Field("nama") String nama,
24                                      @Field("noinduk") String noinduk);
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47     @GET("pengabdian.php")
48     Call<ResponseBody> tampilpengabdianRequest(@Query("username") String username);
49
50
51     @GET("pengabdianadmin.php")
52     Call<ResponseBody> tampilpengabdianRequestAdmin();
53
54     @GET("penelitian.php")
55     Call<ResponseBody> tampilpenelitianRequest(@Query("username") String username);
```

Gambar 7. Interface BaseAPIService daftar, login dan mengambil data profil

Tahap terakhir adalah pada XP adalah pengujian (*testing*). Pada tahap ini peneliti melakukan pengujian fungsional dengan metode *blackbox*. Hasil pengujian secara keseluruhan untuk input data dan tampil data telah berfungsi dengan baik.

### 3. HASIL DAN PEMBAHASAN

Aplikasi Android yang digunakan untuk menguji pertukaran data adalah aplikasi pelaporan kegiatan dosen dalam hal ini adalah kegiatan Pengabdian Kepada Masyarakat (PKM) dan penelitian dosen. Di sisi *client* bahasa yang digunakan adalah Java dan XML untuk desain layoutnya. Jika dijalankan akan menghasilkan tampilan seperti Gambar 8. berikut ini:



Gambar 8. Activity Login

Sebelum bisa melakukan *login* maka dosen harus membuat akun atau harus mendaftar terlebih dahulu. Untuk itu dosen harus membuka *activity Register* melalui link yang telah disediakan. Setelah *activity Register* aktif dosen harus memasukkan data-data seperti yang ditunjukkan pada Gambar 9. Setelah dosen memasukkan data tekan tombol *Register*.



Tampilan sebelum mendaftarkan akun



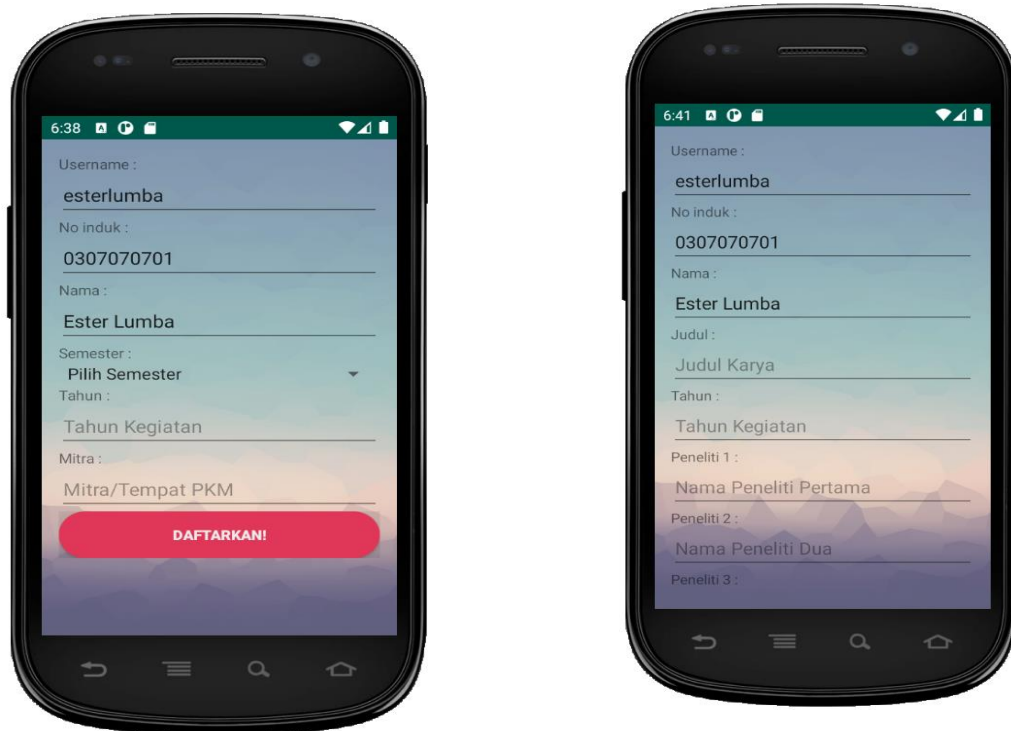
Tampilan input data akun

Gambar 9. Activity Daftar

Isi form pendaftaran tersebut akan dikirim ke server melalui *Interface BaseAPIService*. *Interface* ini yang mendefinisikan metode POST dan form URL encode milik protokol http dari library Retrofit.



Setelah berhasil mendaftar maka data pendaftaran akan tersimpan di *database*. Selanjutnya pengguna bisa melakukan *login*. Jika *username* dan *password* yang dimasukan salah atau tidak terdapat di *database* maka *login* gagal. Namun jika *username* dan *password* yang dimasukkan benar maka akan masuk ke aplikasi. Setelah melakukan *login* maka pada aplikasi dapat menampilkan *activity* kegiatan dosen di mana profil dosen ditampilkan di setiap *activity* untuk *input* kegiatan dosen seperti Gambar 10 berikut ini:



**Gambar 10.** Profil ditampilkan pada setiap *activity* kegiatan dosen

Data profil dosen akan ditampilkan pada setiap dosen menginput kegiatan. Jadi yang diinput hanya kegiatan saja, sedangkan profil dosen telah ditampilkan sesuai dengan data yang didaftarkan sebelumnya.

#### 4. KESIMPULAN

Kesimpulan dari penelitian ini adalah dengan menggunakan library Retrofit 2 maka dapat memudahkan pertukaran data pada aplikasi Android. Dari aplikasi Android format data akan diubah ke dalam format JSON oleh library Retrofit 2 lalu dikirim ke database. Dan sebaliknya ketika mengambil data dari database format data diubah ke dalam format JSON oleh library Retrofit 2 lalu dikirim ke aplikasi Android.

#### DAFTAR PUSTAKA

Anshu Soni, Virender Ranga. (2019). API Features Individualizing of Web Services: REST and SOAP.

*International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 664.

Developer, T. (2021). *Introducing JSON*. Diakses dari <https://www.json.org/json-en.html>

Dimarzio, J. F. (2017). *Beginning Android Programming with Android Studio*. John Wiley & Sons, Inc.

Doglio, F. (2015). *Pro REST API Development with Node.js*. Apress.

International, E. (2021). *The JSON Data Interchange Syntax*. Diakses dari ECMA International:  
<https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>

Jaydevi Bhade, Himanshu Yadav. (2019). Evaluation of Android Networking Libraries. *International Journal of Scientific Research & Engineering Trends*, 1374.

Jiri Hradil, Vilém Sklenak. (2017). Practical Implementation of 10 Rules for Writing REST APIs. *Journal of Systems Integration DOI: 10.20470/jsi.v8i1.290*, 45.

JournalDev. (2020). *Retrofit Android Example Tutorial*. Diakses dari JournalDev:  
<https://www.journaldev.com/13639/retrofit-android-example-tutorial>

Kent Beck, C. A. (2004). *Extreme Programming Explained*. Addison-Wesley Professional.

Smith, B. (2015). *Beginning JSON*. Apress.

Team, D. (2021). *Guide to App Architecture*. Diakses dari Android Developer:  
<https://developer.android.com/jetpack/guide>

Team, G. D. (2016). *Android Developer Fundamental Course Learn to Develop Android Application: Concept Reference*.