# SIMPLIFIKASI *LIST* ANDROID DENGAN PENGGUNAAN *HASHMAP*

**Argo Wibowo**
Universitas Kristen Duta Wacana; Jl. Dr. Wahidin Sudiro Husodo No. 5 – 25
Yogyakarta 55224, Telp. 0274 – 563929 Fax. 0274 – 513235
Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Yogyakarta
e-mail: argo@staff.ukdw.ac.id

***Abstrak***

*Dalam pemrograman android banyak sekali cara yang dapat digunakan untuk mencapai sebuah luaran. Salah satunya adalah membuat list. Dalam membuat list banyak metode yang bisa digunakan, antara lain membuat model, atau membuat adapter sendiri. Dalam tulisan yang saya buat kali ini akan dibahas pembuatan list dengan menggunakan hashmap. Hashmap adalah sebuah bentuk abstrak dari sebuah key dan value, dari key dan value yang disediakan dapat dicocokkan dalam sebuah list.Metode yang akan dibandingkan adalah dengan menggunakan model. Hasil yang didapatkan adalah sebuah list yang sama, namun dengan penggunaan kode yang lebih minimalis dan lebih ringan dalam penggunaan memori. Penggunaan Hashmap lebih fleksibel, karena tidak perlu membuat model dan kelas baru, ataupun adapter baru.*

***Kata kunci***—Android, Model, HashMap, Fleksibel

***Abstract***

*In many ways android programming that can be used to achieve an outcome. One of them is to make a list. In making the list are many methods that can be used, among other things create a model, or create their own adapter. In the paper that I made this time will be discussed making a list by using hashmap. Hashmap is an abstract form of a key and value, of the key and value are provided can be matched in a list.Metode to be compared is to use the model. The result is a list that is similar, but with the use of the code more minimalist and lighter in memory usage. Hashmap the use of more flexible, because it does not need to create a model and a new class, or a new adapter.*

***Keywords***—Android, Model, HashMap, Flexible

## I. PENDAHULUAN

Pemrograman adalah suatu bentuk penulisan program yang tertulis rapi dan terstruktur. Pemrograman adalah bahasa yang bisa dipelajari untuk membuat sebuah program. Ada banyak bahasa pemrograman, di antaranya adalah C, C++, C#, Java, PHP, dan masih banyak bahasa lainnya.Bahasa pemrograman yang akan dibahas dalam penulisan ini adalah Java. Pemrograman Java adalah pemrograman yang bersifat terbuka, sehingga banyak sekali cara untuk membuat program menggunakan Java. Pemrograman java bersifat OOP, sehingga dalam membuat programnya banyak membuat objek atau kelas. Dengan kelas dan objek, diharapkan semua data dapat dimodelkan dengan baik.

Salah satu cabang pemrograman Java adalah Java Android. Dengan adanya Java Android, maka pemrograman Java menjadi lebih luas cakupannya, karena sudah mencakup pemrograman mobile. Dengan adanya java Android pula, aplikasi yang bisa dikembangkan semakin meluas dari aplikasi server, desktop, web, lalu mobile semua bisa menggunakan bahasa Java.Di dalam pemrograman Java, khususnya java Android sendiri sudah mendukung konsep *Object Oriented Programming* (OOP)yang artinya semuaaspek yang terdapat di Java adalah Objek[1]. Namun apakah semua masalah selesai dengan menggunakan kelas dan objek? Menggunakan kelas dan objek tidak selamanya efisien. Selain membutuhkan kelas baru, dengan memakai kelas dan objek yang berlebihan juga bisa membebani jumlah memori yang digunakan dalam sebuah aplikasi.

Salah satu penerapan kelas dan objek dalam pemrograman Java Android adalah pada List. Sebelum dikenalkan secara luas oleh pemrograman Java, list hanyalah sebuah list biasa [2].Sekarang List adalah salah satu

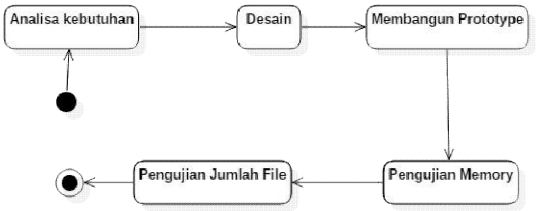
komponen dalam antarmuka Android, dan biasa digunakan untuk menampung dan menampilkan data yang diambil dari basis data. Data yang diambil dari basis data lalu ditampung dalam objek. Objek tersebut adalah hasil dari sebuah kelas.Konsep ini sudah cukup lama digunakan, karena hampir semua basis data direpresentasikan sebagai sebuah kelas atau objek dalam pemrograman. Dalam kelas tersebut, semua tipe data harus didefinisikan dengan baik sesuai dengan tipe data yang terdapat dalam basis data. Jika sebuah list sudah tidak standar, atau dibutuhkan sebuah kustomisasi, maka diperlukan sebuah kelas baru dan sebuah adapter baru. Adapter dapat menggunakan kelas adapter lama, namun ditambahkan fungsi baru tanpa memodifikasi kelas adapter lama [3]. Kelas Adapter berbeda dengan kelas kontroller [4], sehingga harus dibuat terpisah. Penambahan kelas baru dan adapter baru akan menambah memori pada aplikasi, sehingga sebisa mungkin tidak terlalu banyak menambah kelas atau adapter untuk sebuah antarmuka.

Salah satu alternatif yang bisa digunakan selain kelas dan objek adalah HashMap. HashMap adalah sebuah konsep abstrak dari sebuah kelas. HashMap memetakan atribut dan value dalam sebuah array. Secara teknis HashMap menyimpan pemetaan antara teks dan teks terenkrip, dalam bentuk KEY : VALUE, di mana KEY adalah teks, dan VALUE adalah nilai dari teks yang terenkrip[5]. Dengan mendefinisikan KEY dan VALUE, tidak diperlukan lagi kelas atau objek untuk memasukkan data ke dalam sebuah list. Tidak perlu untuk menulis tipe data, karena yang perlu diperhatikan adalah KEY dan VALUE saja. Hal ini memberi dampak positif karena tidak perlu lagi membuat kelas tambahan. Dalam list yang tidak standar pun penggunaan HashMap bisa dilakukan. Dalam tulisan ini didapatkan bahwa dengan menggunakan HashMap ukuran file installer untuk Android menjadi lebih kecil. Selain itu dalam penulisan kode juga lebih sederhana karena tidak memerlukan adapter dan kelas baru.

Penelitian ini bertujuan untuk mencari tahu apakah penggunaan HashMap dapat menjadi alternatif pengganti kelas. Jika bisa digunakan pada sebuah antarmuka, maka tidak perlu membuat banyak kelas baru. Dengan tidak banyak membuat kelas baru untuk tiap antarmuka, maka diharapkan dapat memberi kode yang lebih sederhana, namun efisien. Efisien dalam segi jumlah file, memori, dan ukuran aplikasi.

## II. METODE PENELITIAN

Metodologi penelitian berisi tentang metode pendekatan dan pengembangan yang digunakan yaitu metode komparatif untuk menguji efisiensi penggunaan HashMap dan Kelas atau Objek, lalu metode prototipe untuk pembangunan aplikasinya.Sebuah prototipe adalah model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem[6].Pendekatan dasar komparatif melibatkan kegiatan peneliti yang diawali dari mengidentifikasi pengaruh variabel satu terhadap variabel lainnya. Metode komparatif adalah teknik untuk mempelajari perkembangan bahasa pemrograman dengan melakukan perbandingan fitur-fitur[7], dalam hal ini adalah HashMap dan Kelas. Variabel hashMap dan Kelas memiliki perbedaan dalam segi inisialisasi, dan pembentukan kelasnya. Sehingga dengan memakai kedua variabel ini tentunya akan membuat perbedaan dalam menghasilkan sebuah aplikasi, walaupun dalam luaran terlihat sama.Kemudian untuk metode pengembangan aplikasi dipilih metode prototipe dalam penelitian kali ini karena sangat baik digunakan untuk pengujian aplikasi ini, di mana metode prototipe dapat memberikan definisi program secara umum atau khusus pada suatu fungsi saja. Dalam penelitian kali ini yang dibuat adalah fungsi untuk memasang data pada sebuah list dalam android.Dengan menggunakan prototipe, peneliti tidak perlu membuat keseluruhan aplikasi, cukup membuat fungsi tersebut agar aplikasi bisa segera digunakan [8] dan diuji.Hal ini memberi manfaat bagi peneliti, dengan lebih cepat mengetahui apa yang dibuat dan apa yang bisa mereka gunakan dan mereka uji. Metode ini selalu menggunakan contoh atau sampel aplikasi sehingga peneliti bisa selalu mencoba aplikasi atau sistem yang sedang dikembangkan. Berikut adalah metode yang digunakan dalam pengembangan aplikasi ini ditunjukkan oleh gambar 1.




**Gambar 1** Metode Pengembangan dan Penelitian Aplikasi

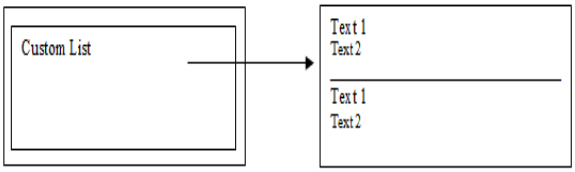
*2.1 Analisa Kebutuhan*

Dalam tahap ini adalah melakukan analisa kebutuhan untuk pengujian aplikasi.Fase perencanaan syarat kebutuhan sistem berhubungan dengan pengumpulansemua data atau variabel [9]. Kebutuhan dalam penelitian kali ini adalah kebutuhan fungsional, karena yang akan diuji adalah fungsinya. Kebutuhan fungsional menggambarkankebutuhan sistem secara fungsi[10].Berikut ini adalah beberapa variabel yang dibutuhkan dalam penelitian kali ini, adalah sebagai berikut:

1. Sebuah list, dengan berisi minimal 2 data teks seperti pada desain.
2. Data JSON, bisa menggunakan data online ataupun offline. JavaScript Object Notation(JSON) merepresentasikan bentuk sebuah data [11].Dalam penelitian kali ini akan digunakan data offline saja, agar ujicoba bisa berjalan lebih cepat. Data json yang digunakan kurang lebih 100 data, sehingga akan terlihat lebih jelas penggunaan memori yang lebih efisien pada aplikasi A atau aplikasi B
3. Kelas yang merepresentasikan data yang akan diambil. Kelas digunakan pada aplikasi A.
4. Adapter. Karena list pada Android pada dasarnya menggunakan sebuah adapter, dan pada penelitian kali ini aplikasi A membutuhkan adapter tambahan karena menggunakan sebuah kelas baru.
5. HashMap, untuk menggantikan kelas di aplikasi B.

*2.2 Desain*

Dalam desain tidak terlalu banyak hal yang dilakukan. Karena aplikasi menguji list, maka cukup diperlukan sebuah antarmuka dengan list di dalamnya. List yang digunakan adalah custom list, maka perlu dibuat 1 antarmuka tambahan sebagai detil dari list tersebut. Desain list dapat dilihat pada gambar 2. Gambar 2 menunjukkan sebuah antarmuka utama di sebelah kiri, lalu di sebelah kanan merupakan antarmuka detilnya. Di dalam antarmuka detil list, terdapat 2 text dalam masing-masing baris. Hal ini yang menjadi perhatian dalam penelitian ini, karena pada dasarnya sebuah list dalam android hanya bisa 1 teks saja. Untuk bisa menjadi 2 teks seperti pada desain membutuhkan sebuah list khusus, dan memerlukan kelas dan adapter untuk bisa mengaksesnya. Salah satu alternatif lain yang bisa diterapkan adalah dengan menggunakan HashMap.




**Gambar 2** Desain Layout List pada Aplikasi A dan B

*Membangun Prototipe*

Setelah ditetapkan desain list yang akan digunakan, maka proses selanjutnya adalah pembuatan prototype. Prototipe mengacu pada desain yang sudah dibuat sebelumnya. Ada 2 antarmuka yang dibuat, dalam bentuk xml karena dalam Android antarmuka didesain dengan menggunakan xml. Aplikasi yang dibuat ada 2 macam, aplikasi A dan B. Aplikasi A akan menggunakan kelas dan adapter tambahan, sedangkan aplikasi B menggunakan HashMap. Namun untuk antarmuka kedua aplikasi sama. Fungsi yang dibuat cukup fungsi untuk memanggil data, lalu dipasang pada list. Tidak perlu membuat fungsi lain, karena yang diuji adalah efisiensi pada list.

*Pengujian Memori*

Dalam tahap ini aplikasi akan diuji dalam beberapa aspek. Yang perlu diperhatikan dan diuji dalam tahap ini adalah:

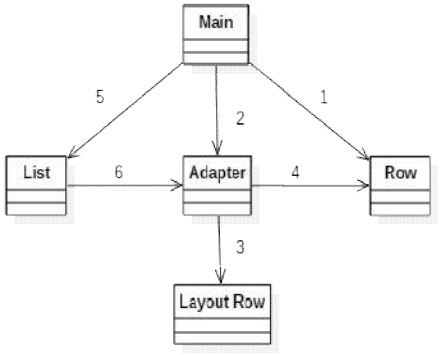
1. Hasil *installer* aplikasi. Karena ada perbedaan metode dalam pembuatan aplikasi A dan B tentunya akan menghasilkan ukuran aplikasi yang berbeda pula. Dengan library yang sama, fungsi yang sama, layout yang sama, namun dengan kelas dan adapter berbeda diharapkan dapat menemukan hasil yang berbeda di antara 2 aplikasi tersebut.
2. Hasil memori yang digunakan oleh aplikasi. Setelah dijalankan akan dicek jumlah memori yang dikonsumsi oleh kedua aplikasi.

*Pengujian Jumlah File*

Dalam tahap ini akan diuji jumlah file yang digunakan dalam pembuatan aplikasi. Metode pembuatan aplikasi A dan B berbeda dalam segi pembentukan kelas, tentu saja jumlah file yang digunakan pun berbeda. Pada aplikasi A dibutuhkan sebuah kelas dan adapter baru, sehingga pada aplikasi A minimal akan lebih banyak 2 file dibandingkan aplikasi B.

## III. HASIL DAN PEMBAHASAN

Dari percobaan membuat 2 aplikasi dengan metode berbeda, maka didapatkan alur kerja sistem seperti gambar 3. Gambar 3 menunjukkan alur kerja dari aplikasi A, yaitu aplikasi yang menggunakan kelas dan adapter.




**Gambar 3** Alur kerja Aplikasi A

Berikut ini adalah penjelasan dari alur kerja dari aplikasi A:

1. Main, Main class akan memanggil kelas Row, untuk mengumpulkan data digunakan list of object Row.
2. Adapter, di sini kelas adapter akan membentuk objeck baru
3. Layout Row xml akan dipanggil oleh adapter, untuk menentukan bentuk layout row
4. Adapter mengambil data yang sudah dikumpulkan oleh list of object Row
5. Main class akan membentuk objek dari kelas layout list.
6. Objek list akan menggunakan adapter yang sudah berisikan layout row dan list of object row

Hasil di atas adalah alur kerja dari aplikasi A. Kelas Adapter akan mengacu pada kelas BaseAdapter seperti di bawah ini:

```
public class CustomListAdapter extends
BaseAdapter
```

Untuk constructor memanggil arrayList yang berisi objek dua string:

```
public CustomListAdapter(Activity
context,ArrayList<DuaString>pairs)
```

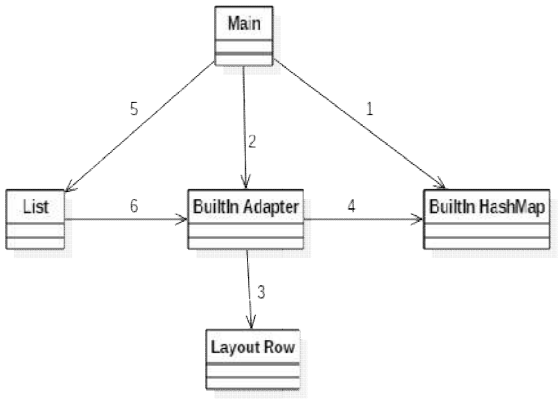
Untuk kelas DuaString sebagai model dari objek data yang akan dipasang pada adapter:

```
public DuaString(String satu, String
dua)
```

Setelah semua kelas sudah siap, lalu dipanggil pada kelas Main. Penggunaan kode di kelas Main adalah sebagai berikut:

```
ArrayList<DuaString> duaStringsList =
new ArrayList<>();
DuaString duaStrings = new
DuaString("Argo", "argo@staff.ukdw.ac.id");
duaStringsList.add(duaStrings);
ListView listView =
(ListView)findViewById(R.id.listViewMenu);
CustomListAdapter customListAdapter
= new CustomListAdapter(MainActivity.this,
duaStringsList);
listView.setAdapter(customListAdapter)
;
```

Lalu menuju aplikasi B, didapatkan alur kerja seperti gambar 4. Terdapat HashMap menggantikan kelas Row pada aplikasi A.




**Gambar 4** Alur kerja aplikasi B

Berikut ini adalah penjelasan dari alur kerja dari aplikasi B:

1. Main, Main class akan memanggil kelas HashMap, untuk mengumpulkan data digunakan array of object HashMap. HashMap di sini adalah kelas yang sudah disediakan oleh Java Android, jadi tidak perlu membuat kelas tersendiri


ISSN 1858-4144

2. Adapter, di sini kelas adapter akan membentuk objek baru. Adapter di sini menggunakan simpleAdapter, yaitu adapter yang sudah disediakan oleh java Android. Tidak perlu membuat adapter khusu.
3. Layout Row xml akan dipanggil oleh adapter, untuk menentukan bentuk layout row
4. Adapter mengambil data yang sudah dikumpulkan oleh array of object HashMap
5. Main class akan membentuk objek dari kelas layout list.
6. Objek list akan menggunakan adapter yang sudah berisikan layout row dan array of object HashMap

Penggunaan HashMap dalam kode program cukup mudah, bisa dilihat pada penggalan kode di bawah ini:

```
HashMap<String, String> hashMap =
new HashMap<String, String>();
ArrayList<HashMap<String, String>>
Timetablelist = new
ArrayList<HashMap<String,
String>>();

hashMap.put("name","Argo");
hashMap.put("email","argo@staff.ukdw
.ac.id");
Timetablelist.add(hashMap);
```

Cukup membuat object HashMap, lalu membuat array. Kemudian tentukan key dan value nya di dalam perintah PUT. Key ini perlu diingat, karena nanti akan berpasangan dengan id pada layout row. Berikut perintah untuk memasang data HashMap dengan Layout Row pada adapter:

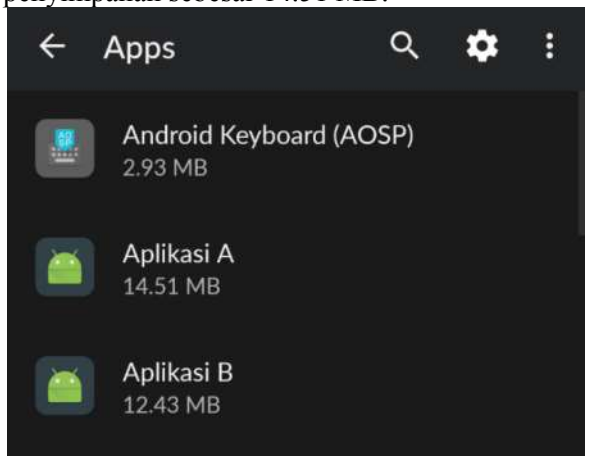
```
ListAdapter adapter = new
SimpleAdapter(MainActivity.this,Timet
ablelist,R.layout.rows,new String[]
{"name","email"},new int[]
{R.id.txtName,R.id.txtEmail});

ListView lv =
(ListView)findViewById(R.id.listViewJSON);
lv.setAdapter(adapter);
```

Dalam penggalan kode di atas dapat dilihat bahwa urutan key NAME dan EMAIL disesuaikan dengan id pada layout, yaitu R.id.txtName dan R.id.txtEmail. Gambar 5 dan 6 di bawah ini menunjukkan hasil pada aplikasi A dan B. Kedua aplikasi menunjukkan hasil yang sama. Data yang diujicobakan ada 100 data, menggunakan perulangan.

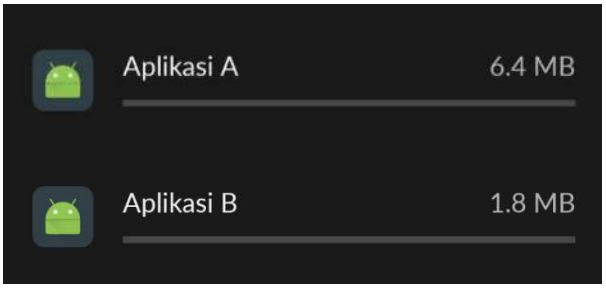

**Gambar 5** Aplikasi A **Gambar 6**Aplikasi B

Hasil pengujian ukuran aplikasi setelah diinstal menunjukkan perbedaan. Gambar 7 menunjukkan perbedaan ukuran aplikasi setelah diinstal pada ponsel. Hasil menunjukkan aplikasi B yang menggunakan HashMap memiliki ukuran file jauh lebih kecil dari aplikasi A, yaitu aplikasi B menyita ruang penyimpanan sebesar 12.43 MB, sedangkan aplikasi B menyita ruang penyimpanan sebesar 14.51 MB.



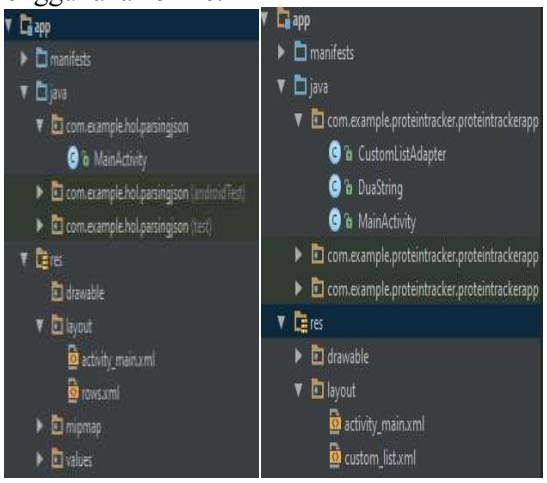

**Gambar 7** Hasil Pengujian Memori Penyimpanan Aplikasi A dan B

Hasil pengujian efisiensi memory RAM menunjukkan hasil yang mirip dengan pengujian ukuran aplikasi. Berikut gambar 8 menunjukkan hasil pengujian RAM dengan menggunakan ponsel pada saat aplikasi dijalankan pertama kali.




**Gambar 8** Hasil pengujian efisiensi memori RAM Aplikasi A dan B

Hasil pengujian jumlah file menunjukkan file yang digunakan oleh aplikasi A lebih banyak dibandingkan aplikasi B. Gambar 9 menunjukkan aplikasi B menggunakan 3 file, sedangkan pada aplikasi A menggunakan 5 file.




**Gambar 9** Perbandingan File aplikasi A (kiri) dan aplikasi B (kanan)

## IV. KESIMPULAN dan SARAN

Ada beberapa kesimpulan yang bisa didapatkan dalam penelitian ini, yaitu:

1. Penggunaan HashMap bisa mengurangi beban ukuran installer aplikasi.
2. Penggunaan HashMap bisa mengurangi penggunaan memori ram pada ponsel ketika digunakan.
3. Penggunaan HashMap membutuhkan file dalam jumlah yang lebih sedikit.

Berikut adalah beberapa saran untuk kekurangan dan kelanjutan penelitian ini adalah:

1. Penggunaan HashMap bisa di mana saja, untuk itu bisa dilakukan dan diterapkan pada antarmuka lain selain list.
2. Tidak semua antarmuka harus menggunakan HashMap atau kelas, bisa dilihat dari kebutuhannya saja. Jika sudah terlalu kompleks, bisa dipertimbangkan menggunakan HashMap diabndingkan membuat kelas baru.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak Universitas Kristen Duta Wacana, khususnya pada Fakultas Teknologi Informasi yang selalu memberi dukungan terhadap peneliti untuk selalu melakukan penelitian dan publikasi ilmiah.

## VI. DAFTAR PUSTAKA

[1] Warno, "Pembelajaran Pemrograman Bahasa JAVA dan Arti Keyword," *Jurnal Komputer,* vol. 8, no. 1, p. 40, 2012.

[2] M. Torgersen, E. Ernst, C. Plesner Hansen, P. von der Ah´e, G. Bracha dan N. Gafter, "Adding Wildcards to the Java Programming Language," *Journal of Object Technology,* vol. 3, no. 11, p. 97, 2004.

[3] S. "A Study on Inheritance Using Object Oriented Programming with C++," *International Journal of Advance Research in Computer Science and Management Studies,* vol. 1, no. 2, p. 10, 2013.

[4] A. Singh, S. Sharma dan S. Singh, "Android Application Development using Android Studio and PHP Framework," *International Journal of Computer Applications,* vol. 0975, no. 8887, p. 5, 2016.

[5] M. Alhanjouri dan A. M. A. Derawi, "A New Method of Query over Encrypted Data in Database using Hash Map," *International Journal of Computer Applications,* vol. 41, no. 4, p. 46, 2012.

[6] P. M. Ogedebe dan B. P. Jacob, "Software Prototyping A Strategy to use WHen User Lacks Data Processing Experience," *ARPN Journal os Systems and Software,* vol. 2, no. 6, p. 219, 2012.

[7] P. K. Mudholkar dan M. Mudholkar, "A New Paradigm of Study of Object Oriented Programming: A Acomparative Approach," *International Journal of Advances in Engineering Research,* vol. 3, no. 5, p. 47, 2012.

[8] R. G. Sabale dan A. Dani, "Comparative Study of Prototype Model For Software Engineering With System Development Life Cycle," *IOSR Journal of Engineering*

*(IOSRJEN),* vol. 2, no. 7, p. 21, 2012.

[9] S. Kosasi, "Penerapan Rapid Application Development Dalam Sistem Perniagaan Elektronik Furniture," *Citec Journal,* vol. 2, no. 4, p. 265, 2015.

[10] P. "Analisis dan Perancangan Sistem Informasi Penjualan Buku Dengan Konsinyasi Berbasis Client Server," *Jurnal Informatika,* vol. 12, no. 2, p. 118, 2012.

[11] B. N. Rupa, G. K. Mohan, J. S. Babu dan T.-H. Kim, "Test Report Generation Using JSON," *International Journal of Software Engineering and Its Applications,* vol. 9, no. 6, p. 63, 2015.