

Aplikasi Toko Tas Online Berbasis Android

Poundra Kartika Putra, Petrus Santoso

Jurusan Teknik Elektro, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya

E-mail: m23407014@john.petra.ac.id ; petrus@petra.ac.id

Abstrak—Internet tidak hanya dapat digunakan pada komputer saja melainkan sudah merambah ke perangkat komunikasi seluler yang saat ini sedang meledak perkembangannya yaitu *smartphone* berbasis *Android*. Pengembangan aplikasi pada *smartphone Android* dapat digunakan sebagai peluang usaha guna memanfaatkan tren perdagangan yaitu aplikasi toko online. Metode yang digunakan untuk membangun aplikasi toko online adalah membangun database, XML parse dan memanfaatkan SMTP Gmail. Database berfungsi untuk menyimpan data barang ke server. XML digunakan untuk menampilkan informasi yang ada pada database server dalam bentuk dokumen XML. Selanjutnya dokumen XML diurai menggunakan metode DOM parser untuk mendapatkan nilai dari dokumen tersebut. Nilai tersebut akan ditampilkan di aplikasi toko online dalam bentuk listview. Notifikasi barang yang dipesan memanfaatkan SMTP. Berdasarkan hasil pengujian, aplikasi *Android* yang dibuat dapat menampilkan product dari database server.

Kata Kunci— *Android, Database, Java, Store, XML*

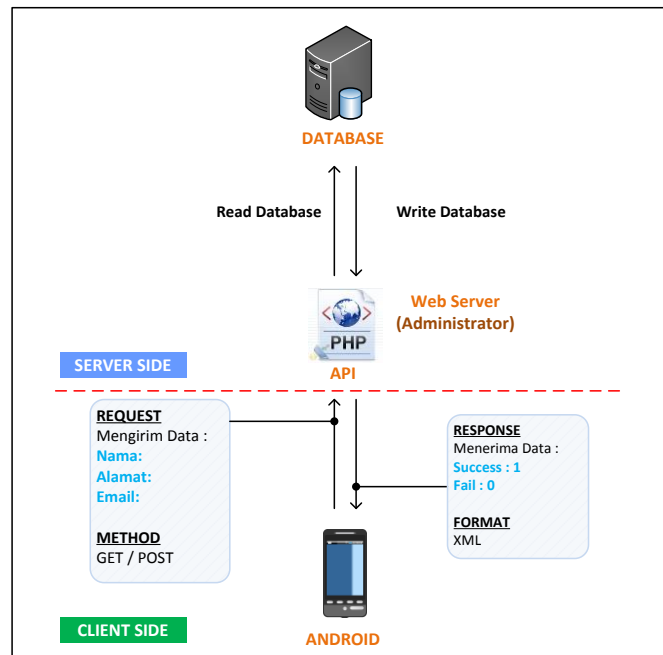
I. PENDAHULUAN

TREN perdagangan yang semakin meledak saat ini yaitu perdagangan *online*. Hal tersebut tidak akan terjadi apabila teknologi yang mendukung tidak ada. Salah satu teknologi yang dimaksud adalah teknologi yang berkaitan dengan telekomunikasi dan informasi baik perkembangan teknologi di bidang *internet* maupun *mobile device*. Saat ini banyak orang menggunakan *smartphone* sebagai *mobile device* dimana perangkat komunikasi tersebut memiliki kelebihan dapat terhubung dengan *internet* untuk berkomunikasi antara satu dengan yang lainnya. Salah satu *smartphone* yang beredar luas di pasaran yaitu berbasis *Android*. Kelebihan dari sistem operasi *Android* adalah variasi merek ponsel yang beredar di pasaran, aplikasi-aplikasi yang tersedia, dan pengembangan aplikasi untuk segala macam keperluan seperti aplikasi toko *online*. Suatu kesempatan bagi para wirausaha sebagai media untuk memperdagangkan barang melalui pemanfaatan teknologi tersebut sebagai suatu peluang bisnis yang menguntungkan secara finansial. Peluang tersebut akan diwujudkan dalam tugas akhir pembuatan aplikasi toko *online* yang akan diadopsi oleh salah satu toko tas yang terletak di salah satu tempat pusat perbelanjaan.

II. DESAIN DAN IMPLEMENTASI SISTEM

Aplikasi toko tas *online* yang dibuat terdiri atas dua bagian. Pertama, sisi *server* yang terdiri atas pembuatan *website administrator* dan *database*. Kedua, sisi *client* yang merupakan

aplikasi berbasis *Android*. Gambaran sistem aplikasi secara umum dapat dilihat pada gambar 1.



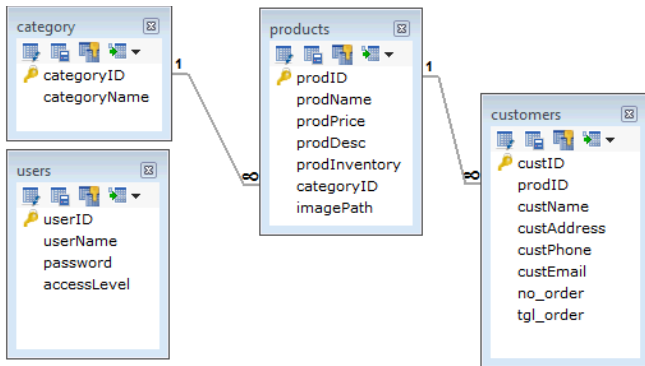
Gbr. 1 Perancangan sistem aplikasi *Android*

A. Desain Server

Pada gambar 1, *server* terdiri atas dua bagian yaitu *web server* dan *database server*. *Web server* berfungsi sebagai antar muka pemilik toko *online* (*administrator*) yang memiliki tugas untuk memanipulasi *database*. Sedangkan *database server* berfungsi sebagai tempat penyimpanan data untuk keperluan *administrator* dan *client*.

1) *Web Server* : *Web server* memiliki fungsi untuk memanipulasi data pada *database* dan menghasilkan dokumen yang memiliki format XML sebagai penghubung *client Android* dan *database* sebagai media untuk menyimpan informasi. Dokumen kode program yang membentuk *web server* merupakan gabungan dari bahasa pemrograman HTML, PHP, dan MySQL. Dokumen yang bertugas untuk memanipulasi *database* dikelompokkan berdasarkan dua kategori yaitu kategori *products* dan *customer*.

1) *Database Server* : *Database server* pada aplikasi ini berfungsi sebagai tempat penyimpanan data *products*, *customers*, *category*, dan *users*. *Entity Relationship Diagram* pada *database* dapat dilihat pada gambar 2.

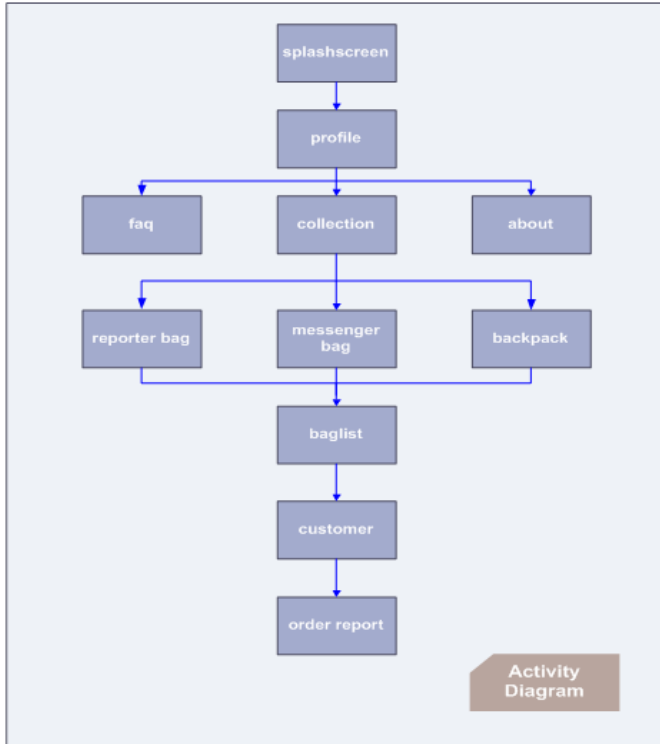


Gbr. 2 Entity relationship diagram

B. Desain Client Android

Desain *client Android* terdiri atas banyak *activity*. *Activity* tersebut saling berhubungan antara satu dengan lainnya. *Activity collection* memiliki *single listview activity* yang berfungsi menampilkan tiga kategori tas. *Activity* tersebut menampilkan berbagai macam tas berdasarkan kategori tas yang dipilih dalam bentuk *listview* di *activity baglist*.

Activity customer didesain untuk mengirimkan data pembeli ke *database server*. Sedangkan *activity order report* didesain untuk mengirim konfirmasi pesanan melalui *email*.

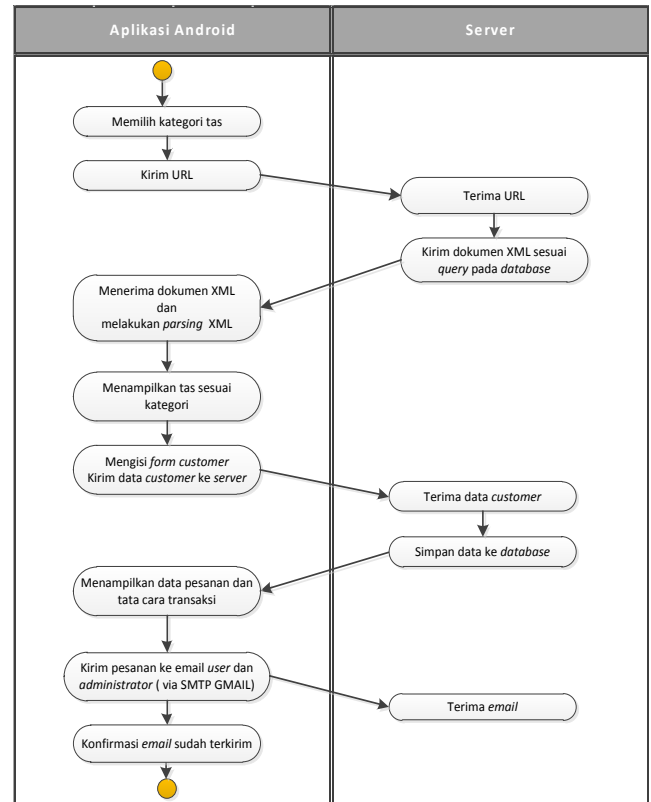


Gbr. 3 Activity diagram

C. Implementasi Sistem

Implementasi sistem dapat dirangkum dalam beberapa proses sebagai berikut :

1) *Proses komunikasi client-server*: Proses komunikasi antara *client Android* dengan *server* dapat dilihat pada gambar 4 dibawah ini :



Gbr. 4 Proses komunikasi client Android dengan server

Proses komunikasi antara *client Android* dengan *server* pada gambar 4 secara garis besar yaitu aplikasi akan meminta URL kepada *server*. *Server* akan melakukan *query* sesuai dengan kategori tas. Kemudian oleh *server*, hasil *query* tersebut dirubah ke dalam bentuk dokumen XML. Dokumen XML akan dikirim ke aplikasi. Aplikasi *Android* melakukan proses *parsing* pada dokumen XML. Proses ini bertujuan untuk mengambil nilai-nilai *element* dimana nilai *element* tersebut merupakan isi *database*. Tahap selanjutnya aplikasi akan memproses hasil *parsing* XML untuk ditampilkan kepada *user* dalam bentuk *list*. Kemudian *client Android* mengisi *form* identitas dengan tujuan *administrator* (*server*) dapat mengetahui informasi *client* yang melakukan pemesanan barang. Informasi tersebut digunakan sebagai wadah dari *administrator* untuk mengirim konfirmasi dan pengecekan transaksi. Selanjutnya *client* melakukan konfirmasi pesanan. Konfirmasi tersebut berwujud *email* yang dikirim ke *administrator* dan *client*. Aplikasi menggunakan SMTP GMAIL sebagai media pengirim *email*.

2) *Proses menampilkan produk dari database*: Aplikasi *Android* mengakses *database products* melalui perantara API yaitu dokumen XML. Dokumen XML ini berisi *database*

products. Proses pembuatan dokumen XML ini menggunakan *query* untuk memanggil semua *products* sesuai dengan kategori. Hasil *query* yang dirubah kedalam format XML terdapat pada gambar 5 dibawah ini :

```
<polarisbagshop>
<bag>
<prodID>29</prodID>
<prodName>Graves</prodName>
<prodPrice>175000</prodPrice>
<prodDesc>Tas untuk pria
kasual, energik dan percaya
diri.</prodDesc>
<prodInventory>In Stock</
prodInventory>
<categoryName>Messenger bag</
categoryName>
<imagePath>
http://203.189.123.200/
~m23407014/images/myproducts/
mb7.png
</imagePath>
</bag>
```

Gbr. 5 Hasil *query* yang dirubah kedalam format XML

Kemudian dari dokumen XML pada gambar 5 diurai (*parse*) agar nilainya dapat diambil untuk ditampilkan pada *Listview*. Proses penguraian dokumen XML menggunakan DOM (*Document Object Model*). DOM merupakan objek yang terdapat dalam halaman *website* berupa *element-element* tag, seperti tag HTML. *DOM Parser* akan mendapatkan *element-element* dari dokumen XML. Nilai *element-element* tersebut akan diambil dengan melakukan *passing* sesuai dengan *node element*. Selanjutnya nilai *element* yang sudah didapat akan dipetakan menggunakan *hashmap*. Cuplikan kode untuk memetakan hasil *parsing* XML terdapat pada gambar 6 sebagai berikut :

```
final ArrayList<HashMap<String, String>> baglist = new
ArrayList<HashMap<String, String>>();

XMLParser parser = new XMLParser();
String xml = parser.getXmlFromUrl(URL);
Document doc = parser.getDomElement(xml);
NodeList nl = doc.getElementsByTagName(KEY_BAG);

for (int i = 0; i < nl.getLength(); i++)
{
HashMap<String, String> map = new HashMap<String,
String>();
Element e = (Element) nl.item(i);

map.put(KEY_ID, parser.getValue(e, KEY_ID));
map.put(KEY_NAME, parser.getValue(e, KEY_NAME));
map.put(KEY_PRICE, "IDR. " + parser.getValue(e,
KEY_PRICE));
map.put(KEY_INV, parser.getValue(e, KEY_INV));
map.put(KEY_DESC, parser.getValue(e, KEY_DESC));
map.put(KEY_PATH, parser.getValue(e, KEY_PATH));
baglist.add(map); }
```

Gbr. 6 Potongan kode untuk memetakan hasil *parsing* XML

Tahap selanjutnya nilai *hashmap* ditanamkan pada setiap *widget* di *Listview*. *Listview* ini akan menampilkan nilai dari setiap nilai kunci *hashmap* sesuai dengan *widget* yang mewakili nilai tersebut. Potongan kode untuk menanamkan

nilai *hashmap* ke *widget* dapat dilihat pada gambar 7 dibawah ini.

```
prodName.setText(bag.get(BagListView.KEY_NAME));
prodPrice.setText(bag.get(BagListView.KEY_PRICE));
prodInventory.setText(bag.get(BagListView.KEY_INV));
prodDesc.setText(bag.get(BagListView.KEY_DESC));
imageLoader.DisplayImage(bag.get(BagListView.KEY_PATH), thumb_image);
```

Gbr. 7 Kode untuk menanamkan nilai *hashmap* ke *listview*

3) *Proses kirim data client ke server*: Proses ini mengirim data yang diisi oleh *client* Android ke *database server*. Dalam mengirim *data client* ke *server* dibutuhkan beberapa variabel. variabel tersebut akan berisi *string* dari inputan yang diketik oleh *client*. Tahap berikutnya membangun parameter dari aplikasi Android menuju tabel *customer*. Selanjutnya program akan melakukan proses *HTTP Request*. *HTTP Request* ini kemudian memanggil *script* di *server* dengan membawa data-data dari aplikasi. *Server* kemudian menjalankan pula serangkaian fungsi-fungsi *Server-Side Script* (PHP) sebagai respon terhadap panggilan *HTTP Request*. Respon tersebut berupa memasukkan data-data ke tabel *customer*. *Script* untuk mengirim data *customer* ke *server* dapat dilihat pada gambar 8 sebagai berikut :

```
List<NameValuePair> params = new ArrayList<NameValuePair>();

params.add(new BasicNameValuePair("custName", custName));
params.add(new BasicNameValuePair("custAddress", custAddress));
params.add(new BasicNameValuePair("custPhone", custPhone));
params.add(new BasicNameValuePair("custEmail", custEmail));
params.add(new BasicNameValuePair("no_order", no_order));

JSONObject json =
jsonParser.makeHttpRequest(add_customer_url,"POST", params);
```

Gbr. 8 *Script* untuk mengirim data *customer* ke API

Selanjutnya nilai yang dikirim oleh aplikasi akan diterima oleh variabel (*server script*). Variabel tersebut menggunakan metode *POST*. Kemudian program akan melakukan *query* pada *database* untuk memasukkan nilai yang berasal dari variabel. Kode untuk memasukkan data ke *database* tabel *customer* dapat dilihat pada gambar 9.

```
$cNama      =$_POST['custName'];
$cAlamat    =$_POST['custAddress'];
$cTelepon   =$_POST['custPhone'];
$cEmail     =$_POST['custEmail'];

$result = mysql_query ( "INSERT INTO customers
(custName,custAddress,custPhone,custEmail)
VALUES
('$cNama', '$cAlamat', '$cTelepon', '$cEmail') " );
```

Gbr. 9 *Script* untuk memasukkan data ke *database* tabel *customer*

4) *Proses konfirmasi melalui SMTP Gmail* : Proses ini mengirim *email* ke *client* Android dan administrator

menggunakan SMTP Gmail. Script untuk mengirim *email* dapat dilihat pada gambar 10 sebagai berikut:

```

Mail mail = new Mail("polarisbagshop@gmail.com",
"*****");

String[] toArr = {email_customer,
"polarisbagshop@gmail.com"}; mail.setTo(toArr);
mail.setFrom("polarisbagshop@gmail.com");
mail.setSubject("Order Polaris Bagshop dengan " +
nomorpesanan_customer);
mail.setBody(isiEmail);

try {

if(mail.send()) {
Toast.makeText(OrderReport.this, "Pesanan Anda
Telah Dikonfirmasi.", Toast.LENGTH_LONG).show();
} else {
Toast.makeText(OrderReport.this, "Konfirmasi
Pesanan Gagal.", Toast.LENGTH_LONG).show();
}
} catch(Exception e) {
Log.e("Order Report", "Email tidak dapat dikirim.",
e);
}
}
    
```

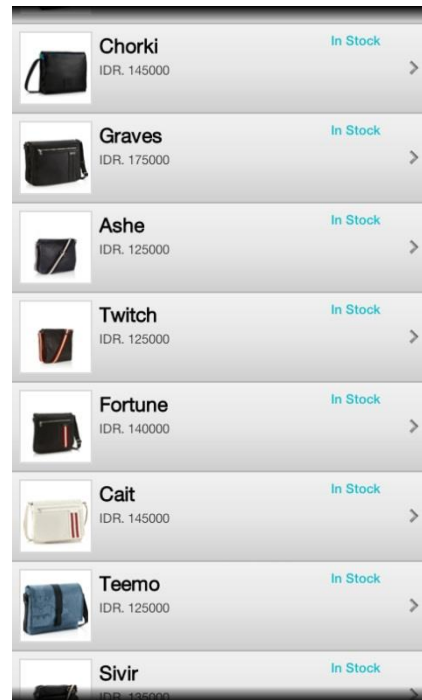
Gbr. 10 Script untuk mengirim email

III. PENGUJIAN SISTEM

Berikut merupakan hasil pengujian dari aplikasi yang dibuat.:

A. Pengujian proses menampilkan produk dari database

Pengujian ini dilakukan untuk mengetahui apakah *activity* ini berjalan di aplikasi *Android*. *Activity* bekerja dengan menampilkan hasil *parsing* dokumen XML untuk mengambil informasi yang ada pada *database products* dalam bentuk *Listview*. Hasil pengujian dapat dilihat pada gambar 11 sebagai berikut :



Gbr. 11 Tampilan produk pada aplikasi *Android*

B. Pengujian proses kirim data client ke server

Pengujian ini dilakukan untuk mengetahui apakah *activity Customer* dapat berjalan di aplikasi *Android*. *Activity* ini menampilkan *form Customer* dengan tujuan *client Android* yang memesan tas mencantumkan identitas dirinya sebagai media komunikasi bagi *administrator*. Informasi tersebut digunakan sebagai data *administrator* untuk mengecek transaksi dan konfirmasi yang dilakukan oleh *client Android*. Tampilan *form Customer* pada aplikasi dapat dilihat pada gambar 12 dibawah ini :

| | | | |
|------------------|---------------------|----------------------|---------------------|
| Nama | poundra | Email | poundra_9@yahoo.com |
| Alamat | SWK Permai J5 | ORDER | |
| Telepon | 08121219999 | ID Produk | 36 |
| Email | poundra_9@yahoo.com | Item | adminEDIT |
| ORDER | | Harga | IDR. 12340000 |
| ID Produk | 36 | Nomor Pesanan | 72186666 |
| Item | | | Next |

Gbr. 12 Tampilan *form customer* pada aplikasi *Android*

Kemudian nilai dari *form* tersebut akan dikirim ke *database*. Hasil pengiriman data ke *database* dapat dilihat pada gambar 13 dibawah ini:

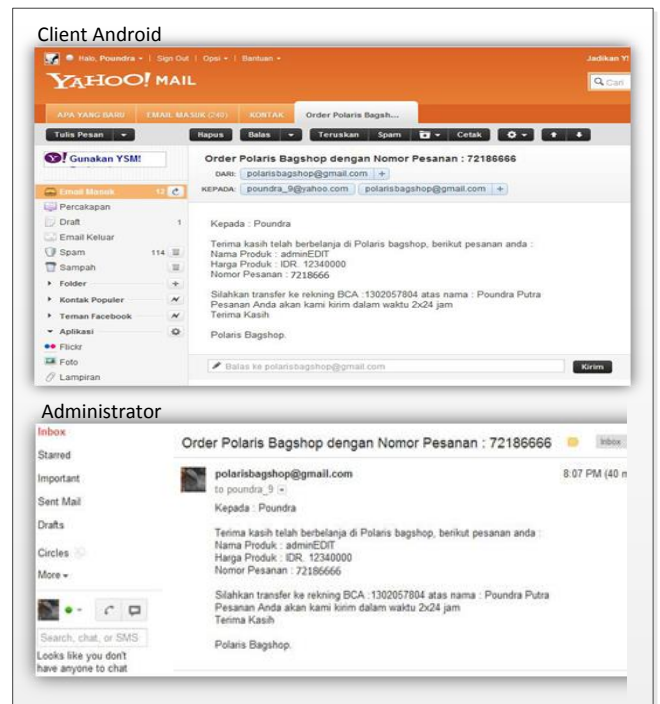
| custID | prodID | custName | custAddress | custPhone | custEmail | no_order | tgl_order |
|--------|--------|----------|---------------|-------------|---------------------|----------|---------------------|
| 24 | 36 | poundra | SWK Permai J5 | 08121219999 | poundra_9@yahoo.com | 72186666 | 2012-11-29 10:31:02 |

Gbr. 13 Hasil dari pengiriman data *activity Customer* ke *database*

C. Pengujian proses konfirmasi melalui SMTP Gmail

Pengujian ini dilakukan untuk mengetahui apakah *activity OrderReport* dapat berjalan di aplikasi *Android*. *Activity* ini berfungsi untuk menampilkan data pesanan dari *client Android* dan perihal yang dibutuhkan dalam transaksi. *Activity* ini terdapat tombol “*I confirm my order*” yang berfungsi sebagai tanda bahwa *client Android* melakukan konfirmasi tas yang dipesan. Wujud konfirmasi adalah *email* yang dikirimkan oleh aplikasi ke *email client Android* dan *email administrator*.

Pertama aplikasi akan menampilkan data pesanan dari *client Android* dan perihal tata cara transaksi dan konfirmasi pemesanan barang. Selanjutnya *client Android* melakukan konfirmasi pemesanan barang dengan menekan tombol “*I confirm my order*”. Tampilan *OrderReport* dapat dilihat pada gambar 14 sebagai berikut :



Gbr. 14 Hasil pengiriman *email* untuk konfirmasi pesanan



Gbr. 14 Tampilan *OrderReport*

Kemudian aplikasi akan mengirim *email* yang berisi barang yang dipesan oleh *client Android*. Email tersebut akan diterima baik oleh *client Android* maupun *administrator*. Hasil pengiriman *email* dapat dilihat pada gambar 15 sebagai berikut:

IV. KESIMPULAN

Kesimpulan dari pengerjaan tugas akhir ini adalah sebagai berikut :

- 1) Secara keseluruhan aplikasi berjalan dengan baik apabila jaringan internet pada *smartphone* juga baik.
- 2) Aplikasi tidak menyimpan data produk di dalam *database smartphone* sehingga tidak mengambil kapasitas *memory smartphone* dalam jumlah yang besar.
- 3) Aplikasi mengadopsi *Asynchronous Task* agar aplikasi berjalan lebih responsif.
- 4) Kekurangan aplikasi terletak pada *client Android* hanya dapat membeli satu tas saja dalam satu aktivitas pembelian dan gambar tas yang terletak di *activity Detail* tidak bisa diperbesar.
- 5) Saran untuk pengembangan aplikasi adalah mengadopsi *login* untuk *customer* agar proses pemesanan tas lebih dari satu *item*.
- 6) Saran berikutnya gambar pada *activity Detail* mengadopsi “*zoom on touch*” dengan tujuan gambar yang disentuh akan mengalami proses perbesaran skala gambar

DAFTAR PUSTAKA

[1] Tittel, E., & Dykes, L. *XML for Dummies* (4th ed.). Indianapolis: Wiley Publishing, Inc, 2005.

[2] “Application Fundamentals.” *Android Developers*. 2012. Android, Inc. 22 September 2012. <<http://developer.android.com/guide/topics/fundamentals.html>>.

[3] “What is API level.” *Android Developers*. 2012. Android, Inc. 28 September 2012. <<http://developer.android.com/guide/appendix/api-levels.html>>.

[4] “Layout.” *Android Developers*. 2012. Android, Inc. 20 September 2012. <<http://developer.android.com/guide/topics/ui/declaring-layout.html>>.

- [5] "DOM." *Android Developers*. 2012. Android, Inc. 21 September 2012. <<http://developer.android.com/reference/org/w3c/dom/package-summary.html>>.
- [6] "Caching Bitmaps." *Android Developers*. 2012. Android, Inc. 24 September 2012. <<http://developer.android.com/training/displaying-bitmaps/cache-bitmap.html>>.
- [7] "Touch Feedback." *Android Developers*. 2012. Android, Inc. 22 September 2012. <<http://developer.android.com/design/style/touch-feedback.html>>.
- [8] "List View." *Android Developers*. 2012. Android, Inc. 23 September 2012. <<http://developer.android.com/reference/android/widget/ListView.html>>.
- [9] "Intent." *Android Developers*. 2012. Android, Inc. 21 September 2012. <<http://developer.android.com/reference/android/content/Intent.html>>.
- [10] "JSON Object." *Android Developers*. 2012. Android, Inc. 29 September 2012. <<http://developer.android.com/reference/org/json/JSONObject.html>>.