

PERANCANGAN DAN IMPLEMENTASI WEB SERVICE SEBAGAI MEDIA PERTUKARAN DATA PADA APLIKASI PERMAINAN

Gilang Arsyia Priyatna Zaman

Program Studi Teknik Informatika Jurusan Teknik Informatika dan Komputer

Politeknik Negeri Jakarta

email: gilang.zaman.tik13@mhs.wpnj.ac.id

Abstract

Data storage in internal smartphones for an applications game that can be played by many users certainly is not something that is recommended. Base on the problem, it is required a useful application as a provider of the data / messages for the application game. Then the web service is one solution, that can be serves as a bridge between applications and the database. So with the web service will ease the loads on the application, because the exchange of data has been provided by the web service. This paper focuses on implementing web service as a media of exchange data on the game application and the testing using application postman. Keywords: web service, client-server, postman, JSON.

1. Pendahuluan

1.1 Latar Belakang

Smartphone dimasa modern ini merupakan sebuah kebutuhan yang wajib dimiliki oleh setiap orang. Kegunaannya yang sangat banyak dapat membantu setiap pekerjaan manusia. Salah satu fungsi dari *smartphone* saat ini yaitu sebagai media hiburan. Adanya permainan pada *smartphone* membuat manusia menjadi nyaman berlama-lama menggunakan *smartphone*.

Memfaatkan semakin banyaknya pengguna *smartphone* maka dikembangkanlah sebuah permainan untuk *smartphone* android, perangkat bergerak yang dalam memainkannya membutuhkan koneksi internet. Serta akan dimainkan oleh ribuan bahkan ratusan ribu user secara bersamaan. Permainan ini tentu saja membutuhkan basis data untuk menampung data-data yang akan digunakan. Tentu saja sangat tidak disarankan untuk menyimpan data-datanya secara lokal di perangkat android. Demi mementingkan kepuasan user yang memainkannya, tentu saja aplikasi yang ringan menjadi sebuah keharusan.

Untuk mengatasi hal tersebut, maka dibutuhkanlah sebuah aplikasi yang berguna menyediakan data/pesan yang dibutuhkan oleh permainan. Maka *web service* merupakan salah satu solusinya, yang berfungsi sebagai jembatan penghubung antara aplikasi dengan basis data. Sehingga dengan adanya *web service* ini akan meringankan beban aplikasi karena semua pertukaran data telah disediakan olehnya.

1.2 Tujuan dan Manfaat

Tujuan dari makalah ini adalah:

Mengimplementasikan *web service* sebagai media pertukaran data dan informasi pada aplikasi permainan.

Manfaat dari adanya *web service*:

- a. Hadirnya *web service* sebagai media pertukaran data antara *server* dengan *client*.

- b. *Web service* sebagai suatu fasilitas yang disediakan oleh suatu sistem untuk menyediakan sebuah layanan kepada sistem lain.

2. Tinjauan Pustaka

2.1 Web Service

Web Service adalah aplikasi yang dibuat agar dapat dipanggil dan diakses oleh aplikasi lain melalui internet dengan menggunakan format pertukaran data sebagai format pengiriman pesan, menurut Kasman(2016). Web service digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada sistem lain, sehingga dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan. Web service menyimpan data informasi dalam format JSON atau XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, dan bahasa pemrograman.

Sebuah web service dapat dipanggil oleh aplikasi lain dengan menggunakan bantuan HTTP (HyperText Transfer Protocol). Karena web service menggunakan protokol HTTP, tentu PHP sebagai bahasa pemrograman web menjadi salah satu kekuatan dalam bahasa pemrograman yang mengelola web services.

2.2 JSON

JSON (Javascript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh computer, menurut Json Org. JSON(2005). JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

2.3 Arsitektur Client-Server

Arsitektur client – server merupakan suatu desain dari sebuah aplikasi yang terdiri dari client dan server di mana saling berkomunikasi satu sama lain ketika mengakses server dalam satu jaringan, menurut Eka(2015). Client sendiri merupakan mesin pengguna yang menyediakan user interface dan local processing di mana melakukan suatu permintaan data atau layanan ke server. Sedangkan server merupakan mesin yang menyediakan data atau layanan yang diminta oleh client.

2.4 Postman

Postman merupakan sebuah tool *REST client* berbasis web yang tersedia dalam bentuk ekstensi pada Google Chrome. Berguna untuk melakukan pengujian dari *web service* yang sedang dikembangkan.

2.5 ERD

ERD (Entity Relationship Diagram) adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan field-field di dalamnya pada suatu database sistem. Sebuah database memuat minimal sebuah tabel dengan sebuah atau beberapa buah field (kolom) di dalamnya, menurut Rosa (2014). Namun pada kenyataannya, database lebih sering memiliki lebih dari satu buah tabel (dengan beberapa field di dalamnya). Setiap tabel umumnya memiliki keterkaitan hubungan. Keterkaitan antar tabel ini biasa disebut dengan relasi. Terdapat tiga buah jenis relasi antar tabel di dalam bagan ERD. Ketiga relasi tersebut yaitu:

- a. One to One (Satu ke Satu) Relasi ini menggambarkan hubungan satu field pada tabel pertama ke satu field pada tabel kedua. Relasi ini paling sederhana.

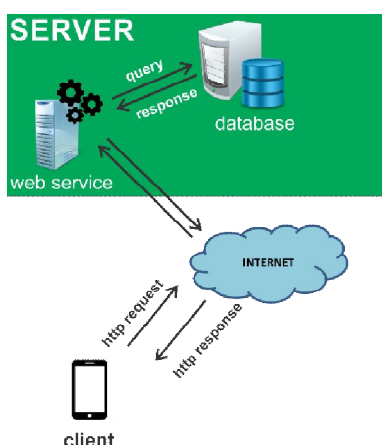
- b. to Many (Satu ke Banyak) Relasi ini menggambarkan hubungan satu field pada tabel pertama ke dua atau beberapa buah field di tabel kedua.
- c. Many to Many (Banyak ke Banyak) Relasi ini berupa N ke N, yang artinya satu atau lebih field pada tabel pertama dapat dihubungkan dengan satu atau lebih field di tabel berikutnya.

3. HASIL DAN PEMBAHASAN

Saat ini *web service* telah banyak dikembangkan untuk menunjang efektifitas pertukaran informasi oleh beberapa pengembang aplikasi, salah satunya aplikasi berbasis *smartphone*. Penerapan *web service* dalam pertukaran informasi bedaplatform antara *website* yang dikembangkan menggunakan pemrograman PHP dan aplikasi Android yang dikembangkan menggunakan pemrograman Java. Tentu saja hadirnya *web service* untuk mengatasi hal tersebut, selain itu *web service* juga dapat membuat aplikasi pada *smartphone* menjadi ringan, karena semua data yang dibutuhkan aplikasi tersimpan di dalam *server* bukan pada *smartphone*. Jadi apabila aplikasi membutuhkan data hanya melakukan *request* data menggunakan *link* yang disediakan oleh *web service*.

3.1 Model Arsitektur

Model arsitektur yang akan dibuat dapat dilihat pada gambar 1 berikut ini.



Gambar 1 Model Arsitektur

Proses dimulai dari *Client*, yaitu aplikasi android meminta data ke server melalui internet ke *web service*, kemudian *web service* pun melakukan perintah yang diminta *client* untuk berkomunikasi dengan *database*, *database* pun mengirimkan data yang diminta. Sukses atau tidaknya permintaan data dari *client*, akan dikirimkan balikan oleh *web service* untuk memberitahu *client*.

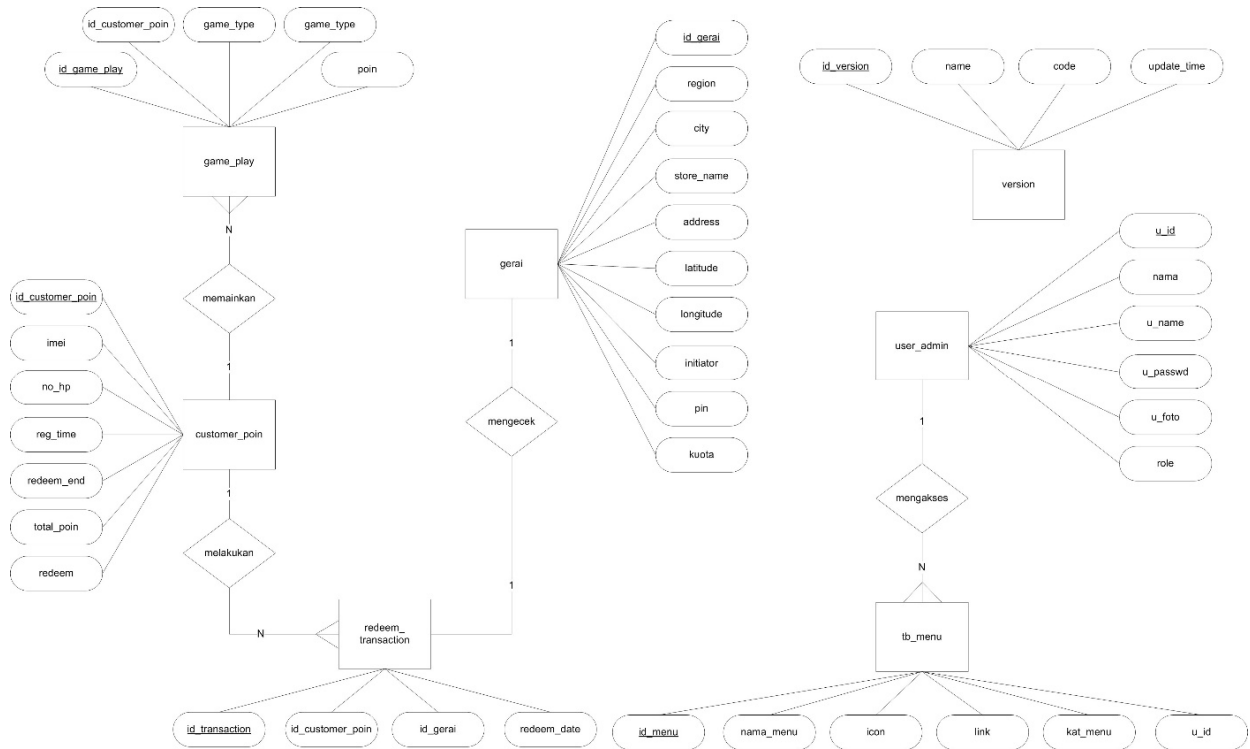
Adapun modul-modul yang akan dibuat dari *web service* ini adalah sebagai berikut:

- a. Modul Register dan login
- b. Modul Poin
- c. Modul *redeem*/tukar pulsa

3.2 ERD

Entity Relationship Diagram pada web service ini dapat dilihat di gambar 2

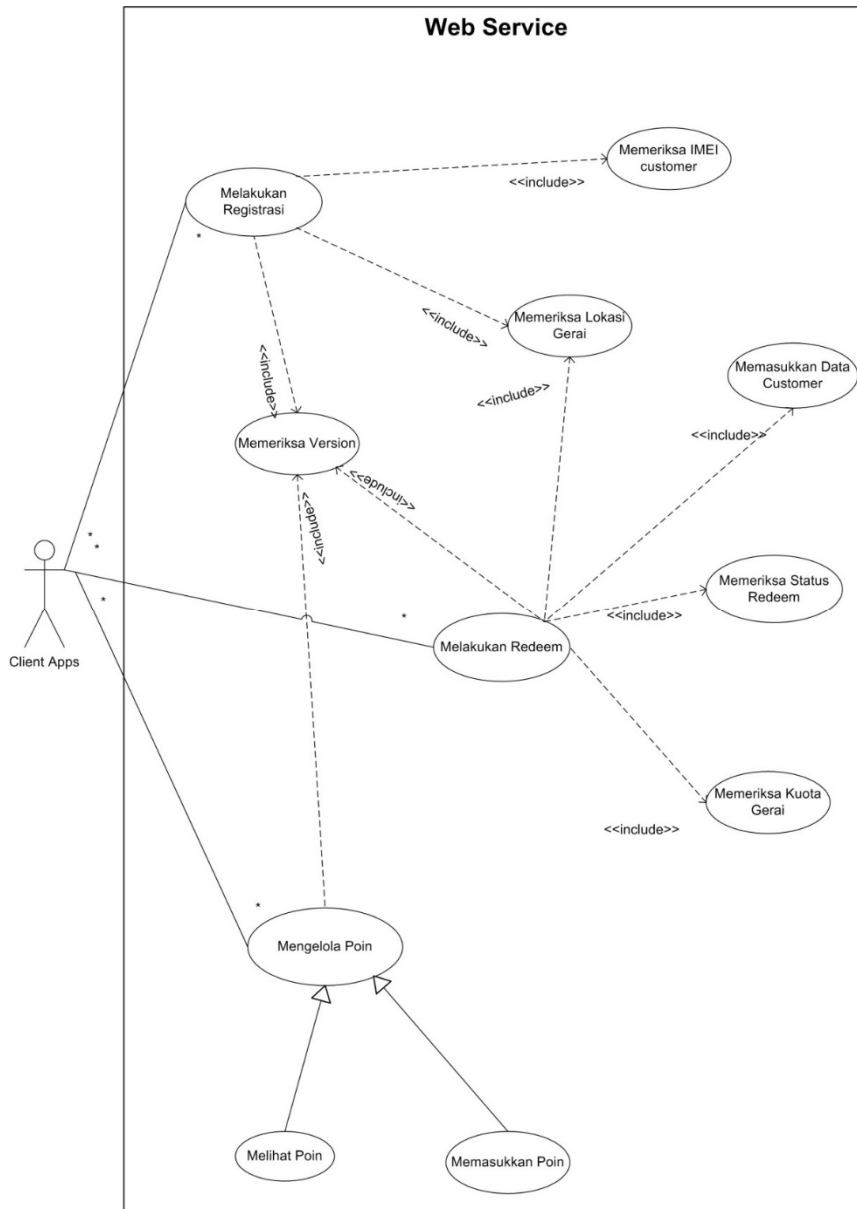
berikut ini.



Gambar 2. Entity Relationship Diagram Web Service

3.3 Use Case

Adapun *use case diagram* untuk pembuatan *web service* ini dapat dilihat pada gambar 3 berikut ini.



Gambar 3. Use Case Diagram Web Service

Tabel 1. Penjelasan Use Case Diagram Web Service

Aktor	Description
Client Apps	Dapat melakukan registrasi, dapat bermain game yang poinnya disimpan ke dalam database, dapat melakukan redeem transaction, yaitu menukar poin dengan pulsa. Semua proses tersebut melalui web service terlebih dahulu.

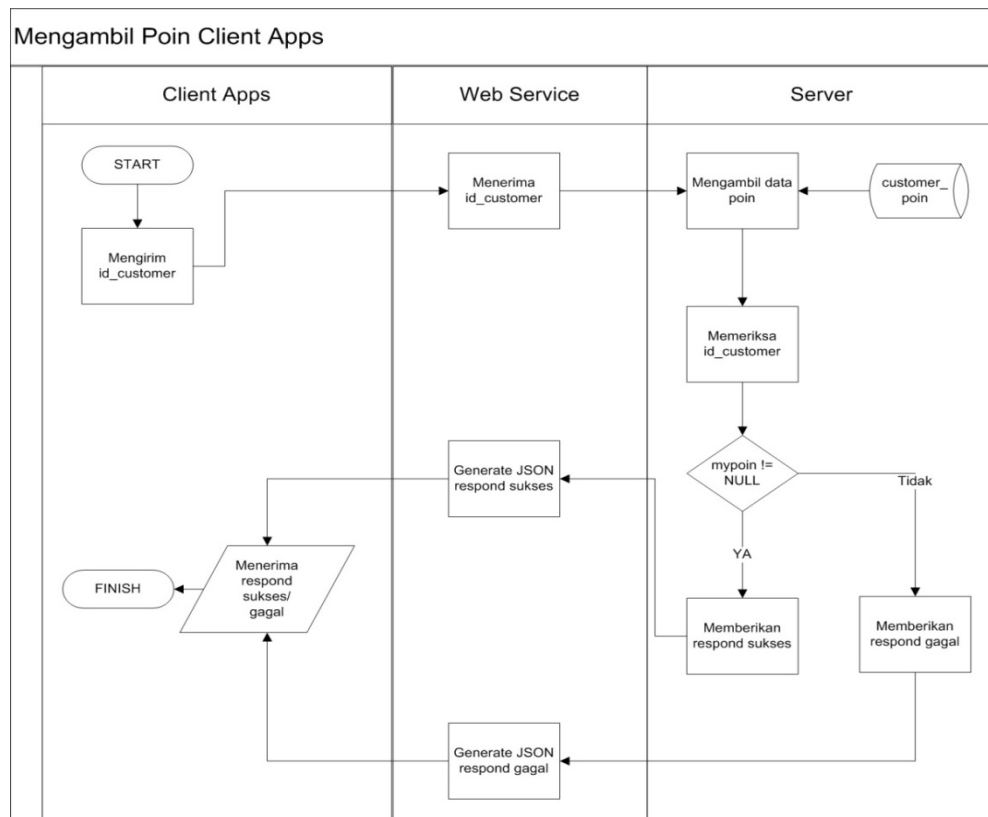
3.4 Flowchart

3.4.1 Flowchart Modul Poin Client Apps

Pada makalah ini membahas untuk fungsi poin, yaitu mengambil data poin *client apps* ke server melalui *web service* yang tersedia, kemudian di fungsi

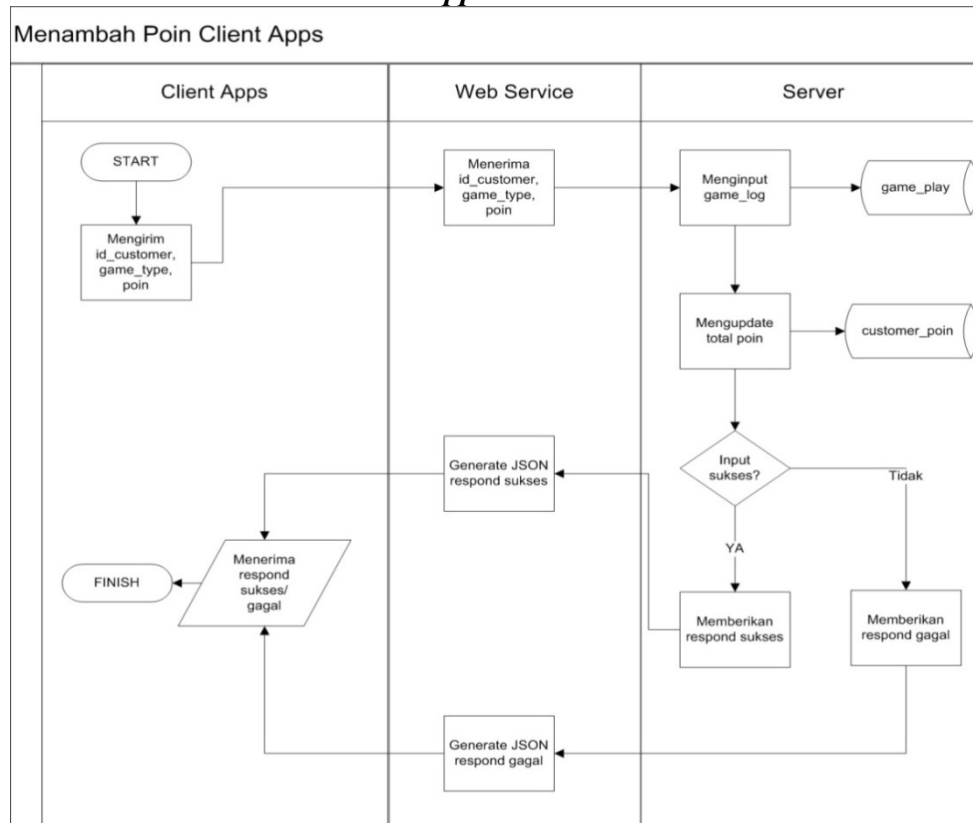
ini juga penginputan poin *client apps* dilakukan, poin yang didapat dari setelah memainkan permainan akan disimpan ke dalam database.

a. Mengambil Poin *Client Apps*



Gambar 4.Flowchart Poin Client Apps

b. Menambah Poin Client Apps



Gambar 5. Flowchart Menambah Poin Client Apps

4. PENGUJIAN

4.1 Pengujian Modul Poin Client Apps

Pengujian dilakukan menggunakan tools Postman yang dapat diunduh di market *browser chrome*.

a. Mengambil Poin Client Apps

Tabel 2. Hasil Pengujian Mengambil Poin Client Apps

HTTP Request			
1	Verb/Method	:	POST
2	URI	:	http://namadomain/get_poin/get
3	HTTP Version	:	HTTP v1.1.
4	Request Header	:	-
5	Request Body	:	id_customer
HTTP Response			
1	Status/Response Code	:	200 response OK Status: 200 OK Time: 346 ms
2	HTTP Version	:	HTTP v1.1.

3	Response Header	:	<p>Connection → keep-alive</p> <p>Content-Type → application/json</p> <p>Date → Thu, 08 Dec 2016 04:35:11 GMT</p> <p>Server → nginx</p> <p>Transfer-Encoding → chunked</p>
4	Response Body	:	<pre>{ "kode": 200, "pesan": "sukses", "mypoin": { "id_customer_poin": "1", "total_poin": "218540" } }</pre>

Tabel diatas merupakan hasil simulasi dari pengimplementasian *web service* untuk memberikan total poin user ke aplikasi. Ketika aplikasi meminta poin user yang sedang menggunakan aplikasi, maka *web service* akan memberikan response berupa total poin dari user tersebut.

b. Menambah Poin *Client Apps*

Tabel 3.Hasil Pengujian Menambah Poin *Client Apps*

HTTP Request			
1	Verb/Method	:	POST
2	URI	:	http://namadomain/post_game_play/post
3	HTTP Version	:	HTTP v1.1.
4	Request Header	:	-
5	Request Body	:	id_customer game_type poin
HTTP Response			
1	Status/Response Code	:	200 response OK Status: 200 OK Time: 108 ms
2	HTTP Version	:	HTTP v1.1.

3	Response Header	:	<p>Connection → keep-alive</p> <p>Content-Type → application/json</p> <p>Date → Thu, 08 Dec 2016 04:35:11 GMT</p> <p>Server → nginx</p> <p>Transfer-Encoding → chunked</p>
4	Response Body	:	<pre>{ "kode": 200, "pesan": "sukses" }</pre>

Tabel diatas merupakan hasil simulasi dari pengimplementasian *web service* untuk menyimpan poin yang diperoleh user setelah memainkan permainan pada aplikasi.

5. KESIMPULAN

Dari hasil penelitian yang sudah dilakukan, maka penulis dapat menarik simpulan bahwa implementasi *web service* serbagai media pertukaran data pada aplikasi permainan berhasil diimplementasikan, proses pertukaran data berjalan dengan baik sesuai dengan hasil simulasi yang dilakukan melalui aplikasi *postman*.

6. REFERENSI

- a. Anhar, ST. 2010. Panduan Menguasai PHP & MySQL Secara Otodidak. Jakarta. Mediakita
- b. A.S, Rosa, dan M.Shalahuddin. 2014. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.
- c. Eka, I Putu Agus Pratama. 2015. Sistem Informasi dan Implementasinya. Bandung. Informatika.
- d. Fielding, Roy Thomas. 2000. Architectural Styles and the Design of Network-based Software Architectures. Disertasi Doktoral, Irvine.
- e. Github Gist #1 Code example from JSON: What It Is, How It Works, & How to Use It [Online]. Url: <https://gist.github.com/2760279>. [20 November 2016]
- f. Github Gist #2 Code example from JSON: What It Is, How It Works, & How to Use It [Online]. Url: <https://gist.github.com/2760312>. [20 November 2016]
- g. Json Org. JSON [Online]. Url: <http://json.org/json-id.html>. [20 November 2016]
- h. Kasman, Akhmad Dharma. 2015. Trik Kolaborasi ANDROID dengan PHP dan MySQL. Yogyakarta. Lokomedia.
- i. Sibero, Alexander F.K. 2011. Kitab Suci Web Programming. Yogyakarta. Mediakom.
- j. Smartdraw. Flowchart Symbols. <http://www.smartdraw.com/flowchart/flowchart-symbols.htm> . [26 November 2016]
- k. Visual-paradigm. Use Case Diagram. <http://www.visual-paradigm.com/VPGallery/diagrams/UseCase.html> . [02 Desember 2016]