

PEMBUATAN APLIKASI INFORMASI TAGIHAN LISTRIK BERBASIS ANDROID

Ariza Dyah Kartikasari

Program Studi Sistem Informasi, Fakultas Ilmu Komputer,
Universitas Dian Nuswantoro Semarang

Jl. Nakula I, No. 5-11, Semarang, Kode Pos 50131, Telp (024) 3515261, 3520165

E-mail : girlz_cold@yahoo.com

Abstrak

Ada beberapa pelanggan PLN yang membutuhkan informasi mengenai jumlah tagihan listrik mereka. Untuk menyediakan informasi tersebut, PLN membuat website yang dapat diakses melalui PC maupun mobile. Namun alamat website layanan yang disediakan oleh PLN ini kurang populer. Ditambah lagi dengan masalah kecepatan akses internet. Ada layanan lain yang disediakan, yaitu melalui SMS. Namun untuk mengetahui jumlah tagihan, pelanggan harus mengingat format SMS dan nomor layanan. Dan untuk sekali mendapatkan SMS balasan, pelanggan dikenakan biaya Rp 500. Untuk mengatasi permasalahan di atas, maka dibuatlah sebuah aplikasi yang dapat dijalankan pada perangkat mobile, lebih simpel, cepat, tidak mengkonsumsi banyak bandwidth, dan lebih informatif, dibandingkan dengan layanan yang sudah disediakan oleh PLN. Aplikasi mobile dipilih agar dapat dioperasikan di mana saja dan kapan saja. Aplikasi Informasi Tagihan PLN ini dibuat untuk perangkat mobile yang menggunakan Android sebagai sistem operasinya. Android dipilih karena merupakan sistem operasi mobile yang paling banyak digunakan di Indonesia.

Kata Kunci: PLN, informasi tagihan listrik, aplikasi, mobile, android

Abstract

From 49 million customers of PLN, there are customers who need information about their electric bill in every month. To provide this information, PLN build a website that can be accessed on a PC or mobile. But the website address provided by PLN is not quite popular for the customers. Moreover there is internet access speed issue. It would be a problem for mobile users who use volume-based internet package, because website files are large enough. There is another service provided by PLN for billing information, that is SMS service. But, customers must remember the SMS service format and service number, and it is charged customers for IDR 500 for every reply or SMS from PLN. To overcome these problems, so the author built an application that can be run on mobile device. It is simpler, faster, consumes less bandwidth, and more informative than the service provided by PLN. The author chose a mobile application because it can be operated anywhere and anytime. This application is built for mobile devices which use Android as its operating system. Android is chosen because it is the most popular mobile operating system in Indonesia.

Keywords: PLN, electric bill information, application, mobile, android

1. PENDAHULUAN

1.1 Latar belakang

Pelanggan PLN ada yang membutuhkan informasi mengenai tagihan listrik setiap bulannya dan PLN juga sudah menyediakan *website*. Namun *website* itu kurang populer dan

memakan banyak *bandwidth*. PLN juga menyediakan layanan lewat SMS, namun pelanggan harus mengingat format dan setiap mendapat balasan sms pelanggan dikenai biaya Rp500. Berdasarkan masalah diatas maka dibuatlah aplikasi yang dapat berjalan secara *mobile* dan mempermudah

pelanggan untuk mendapatkan informasi tagihan listrik. Aplikasi informasi tagihan listrik ini dibuat untuk perangkat *mobile* yang memiliki sistem operasi android.

1.2 Perumusan masalah

Bagaimana membangun suatu aplikasi tagihan listrik PLN yang dapat berjalan pada perangkat *mobile* seperti *smartphone* dan komputer tablet yang memiliki sistem operasi android, sehingga informasi tagihan listrik dapat dengan mudah diakses oleh pengguna.

1.3 Pembatasan masalah

Dari permasalahan yang telah disebutkan di atas, maka batasan-batasan dalam tugas akhir ini adalah :

1. Pembuatan aplikasi *mobile* berbasis android.
2. Menggunakan Eclipse IDE dan ADT sebagai *development tool*.
3. Penyajian informasi tagihan listrik berupa teks berdasarkan tanggal (bulan).
4. Penyajian informasi tagihan listrik bulan dan bulan-bulan sebelumnya dalam bentuk grafik.
5. Hanya menampilkan tagihan listrik PLN Pasca Bayar.
6. Tidak menyediakan fasilitas pembayaran tagihan.
7. Tidak mengeluarkan *file report* dalam format .pdf ataupun format lainnya.

2. LANDASAN TEORI

2.1 Aplikasi

Pengertian Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas.[1]

Pada pengertian umumnya, aplikasi adalah alat terapan yang difungsikan

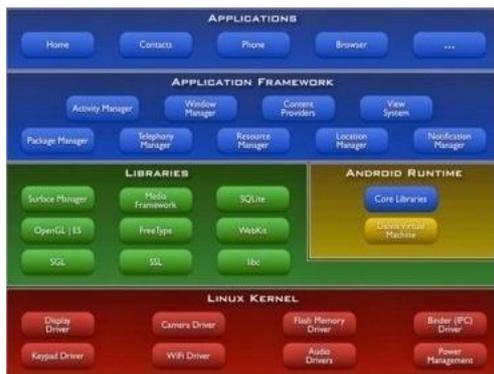
secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. [2]

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.[2]

2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat *software* untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [3]

2.2.1 Arsitektur Android



Gambar 1. Arsitektur Android

Arsitektur android terdiri dari *Applications*, *Application Framework*, *Android Runtime*, *Libraries*, *Linux Kernel*.

2.2.2 Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di dalam DVM bukan di *Java Virtual Machine* (JVM). DVM adalah “*register based*” sementara JVM adalah “*stack based*”. DVM, *Dalvik Virtual Machine*, menggunakan *kernel* Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk menjalankan aplikasi. [4]

2.2.3 Android SDK (Software Development Tools)

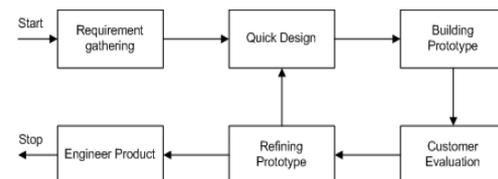
Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google.

2.2.4 Android Development Tools (ADT)

Android Development Tools (ADT) adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse, akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga dapat melakukan *running* aplikasi menggunakan Android SDK melalui Eclipse.[4]

2.3 Prototype

Prototype adalah salah satu metode pendekatan dalam proses pengembangan *software*, yaitu pembuatan *prototype* atau rancangan dari *software* yang akan dikembangkan.[5]



Gambar 2. Arsitektur Metode Prototype

2.3.1 Tahapan Prototype

Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut:

- Pengumpulan kebutuhan (*Requirement Gathering*)
- Desain cepat (*Quick Desain*)
- Membangun *prototype* (*Building Prototype*)
- Evaluasi *prototype* oleh pelanggan (*Customer Evaluation*)
- Perbaikan *prototype* (*Refining Prototype*)
- Produk rekayasa (*Engineer Product*)

2.3.2 Prinsip Dasar Prototype

- Merupakan metode lama yang membuat rancangan lengkap tetapi hanya menjelaskan bagian-bagian penting dari pengembangan *software*.
- Upaya untuk mengurangi resiko dalam *project*, yaitu dengan memecahnya menjadi bagian-bagian yang lebih kecil, sehingga dapat memberikan kemudahan untuk melakukan perubahan-perubahan selama proses pengembangan aplikasi.
- User* terlibat dalam proses pengembangan, sehingga dapat memperbesar peluang berhasilnya implementasi akhir.
- Memberikan gambaran kepada *user* tentang rancangan sistem yang dibuat, yang mengikuti modifikasi-modifikasi yang diinginkan hingga rancangan tersebut memenuhi kebutuhan *user*.
- Sebagian besar rancangan dibuat dengan pemikiran bahwa rancangan tersebut akan ditolak oleh *user*, hal ini bisa terjadi dalam beberapa kasus untuk memberi semangat dalam menjadikan rancangan tersebut sebagai sebuah sistem yang dapat dioperasikan.
- Dibutuhkan pemahaman terhadap dasar-dasar *business process* dari sistem yang akan dikembangkan untuk mnghindari kesalahan dalam proses pengembangan.

3. DESAIN SISTEM

3.1 Arsitektur Sistem

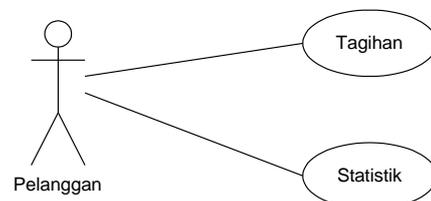


Gambar 3. Arsitektur Sistem

Aplikasi Informasi Tagihan Listrik Berbasis Android ini mempunyai sebuah arsitektur yang dapat dijabarkan seperti gambar di atas. Pada gambar 2, pelanggan yang menggunakan *smartphone* android melakukan permintaan data ke DB SQLite, begitupun sebaliknya.

Pelanggan android pun juga melakukan permintaan HTTP Request ke website resmi PLN yaitu <http://layanannya.pln.co.id>, dan sebaliknya yaitu website resmi PLN tersebut memberikan HTTP Response kepada pelanggan.

3.2 Diagram Usecase



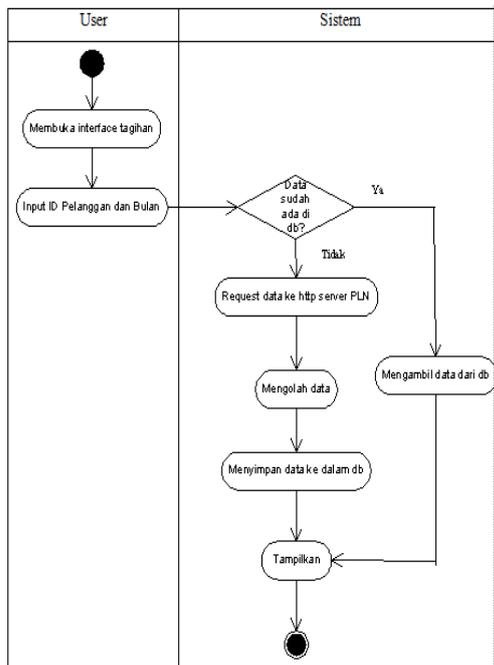
Gambar 4. Usecase Diagram

Pada diagram *usecase* di atas, hanya terdapat satu aktor, yaitu pelanggan. Pada aplikasi informasi tagihan listrik, terdapat dua menu, yaitu Tagihan dan Statistik.

Pelanggan dapat mengetahui jumlah tagihan pada *interface* tagihan dan pemakaian KWH selama 6 bulan terakhir pada *interface* statistik. Begitu aplikasi terbuka, pelanggan langsung dihadapkan pada menu Tagihan. Dan untuk mengakses menu Statistik, pelanggan harus memasukkan ID Pelanggan dahulu di menu Tagihan.

3.3 Activity Diagram

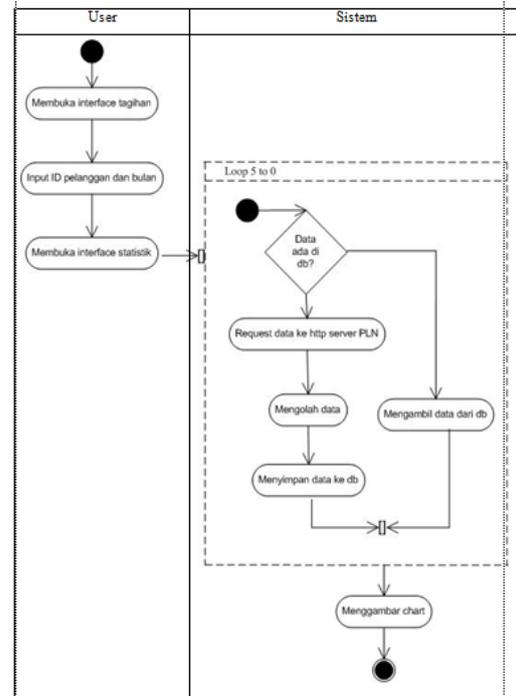
3.3.1 Activity Diagram Tagihan



Gambar 5. Diagram Activity Interface Tagihan

Untuk menjalankan aplikasi tagihan listrik ini, pertama-tama pelanggan membuka interface tagihan, kemudian menuliskan id pelanggan serta bulan dan tahun yang ingin ditampilkan. Setelah itu di dalam sistem dilakukan pengecekan di dalam database, yaitu apakah data yang diinginkan sudah ada di dalam database. Apabila tidak ada maka sistem melakukan request data ke HTTP server PLN. Setelah data didapatkan, sistem melakukan pengolahan data dan kemudian disimpan ke dalam database, kemudian data tersebut ditampilkan. Namun, apabila data sudah terdapat di dalam database maka sistem langsung mengambil data tersebut dari database kemudian ditampilkan.

3.3.2 Activity Diagram Statistik

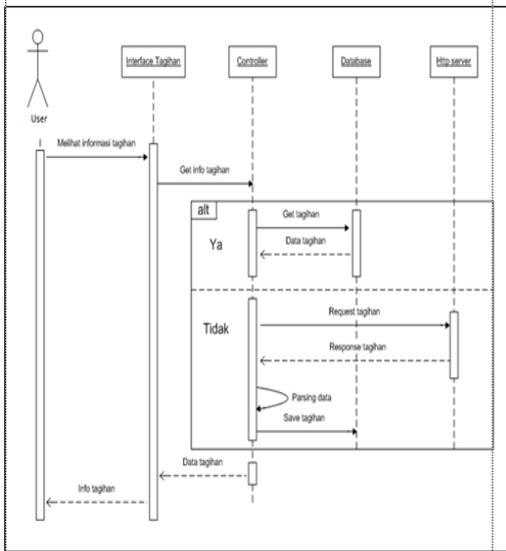


Gambar 6. Diagram Activity Interface Statistik

Untuk mengakses menu statistik, pelanggan pertama-tama membuka *interface* tagihan, lalu menuliskan nomor id pelanggan, bulan serta tahun yang ingin ditampilkan. Setelah itu baru pelanggan membuka *interface* statistik. Di dalam sistem statistik, dilakukan pengecekan di dalam *database*, apakah data yang diminta oleh pelanggan sudah ada di *database* atau belum. Apabila data sudah ada maka sistem langsung mengambil data dari *database*. Namun, apabila data belum ada di *database*, maka sistem melakukan *request* data ke HTTP server PLN. Setelah data didapat, data diolah dan kemudian disimpan di dalam *database*. Proses yang terjadi di dalam sistem ini dilakukan secara berulang sebanyak 6 kali, yaitu 6 bulan terakhir. Setelah semua data didapatkan dari hasil perulangan, kemudian data ditampilkan berupa chart.

3.4 Sequence Diagram

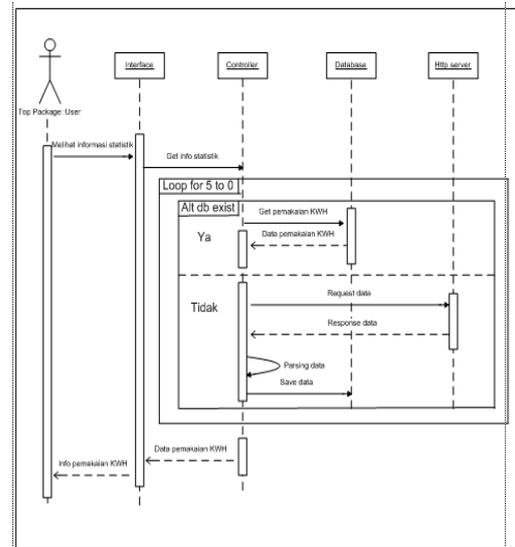
3.4.1 Sequence Diagram Tagihan



Gambar 7. Diagram Sequence Interface Tagihan

User melihat informasi tagihan pada *interface* tagihan. *Interface* tagihan kemudian mengirimkan pesan get info tagihan kepada *controller*. Di dalam *controller* terjadi pengecekan, apakah data sudah tersimpan di dalam *database*. Apabila data sudah ada, maka *controller* mengirimkan pesan *get* tagihan kepada *database*, kemudian *database* mengirimkan data tagihan. Namun, apabila data belum ada di *database*, *controller* mengirimkan *request* tagihan kepada *http server* dan *http server* memberikan *response* tagihan kepada *controller*. Di dalam *controller*, hasil dari *response* tagihan tersebut kemudian *diparsing*, selanjutnya dilakukan penyimpanan di dalam *database*. Setelah itu *controller* memberikan data tagihan kepada *interface* dan *interface* meneruskan info tagihan kepada *user*.

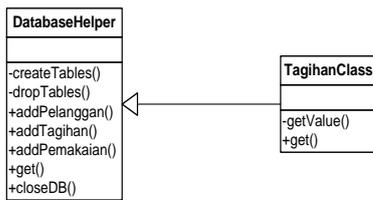
3.4.2 Sequence Diagram Statistik



Gambar 8. Diagram Sequence Interface Statistik

User melihat informasi statistik pada *interface*. *Interface* kemudian mengirimkan pesan get info statistic kepada *controller*. Di dalam *controller* terjadi perulangan sebanyak 6 kali, yaitu dari indeks ke 5 sampai indeks ke 0. Di dalam perulangan juag terjadi pengecekan, apakah data sudah tersimpan di dalam *database*. Apabila data sudah ada, maka *controller* mengirimkan pesan *get* pemakaian KWH kepada *database*, kemudian *database* mengirimkan data pemakaian KWH. Namun, apabila data belum ada di *database*, *controller* mengirimkan *request* data kepada *http server* dan *http server* memberikan *response* data kepada *controller*. Di dalam *controller*, hasil dari *response* data tersebut kemudian *diparsing*, selanjutnya dilakukan penyimpanan di dalam *database*. Selanjutnya pengecekan dilakukan kembali sebanyak 6 kali. Apabila perulangan tersebut telah selesai, *controller* memberikan data pemakaian KWH kepada *interface* dan *interface* meneruskan info pemakaian KWH kepada *user*.

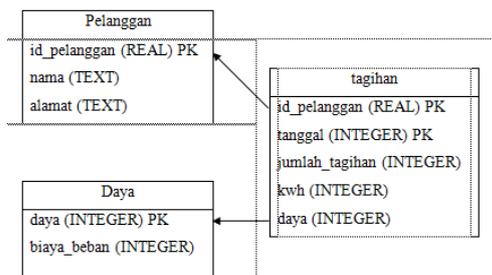
3.5 Class Diagram



Gambar 9. Class Diagram

Ada dua kelas dalam aplikasi tagihan listrik ini, kelas DatabaseHelper dan kelas TagihanClass. Pada kelas DatabaseHelper terdapat operation createTables(), dropTables(), addPelanggan(), addTagihan(), addPemakaian(), get(), dan closeDB(). Operation createTables() dan dropTables() bersifat *private*, sedangkan yang lain bersifat *public*. Sedangkan pada TagihanClass terdapat operation getValue() yang bersifat *private* dan operation get() yang bersifat *public*. Kelas DatabaseHelper menjadi *superclass* untuk kelas TagihanClass.

3.6 Relasi Tabel



Gambar 10. Relasi Tabel

Pada aplikasi ini penulis membuat tiga buah tabel yang saling berhubungan. Id_pelanggan pada tabel tagihan mengacu pada id_pelanggan tabel pelanggan. Daya pada tabel tagihan mengacu pada daya di tabel daya.

4. IMPLEMENTASI SISTEM

4.1 Halaman Tagihan



Gambar 11. Tampilan Form Tagihan

Pada gambar 4 terdapat ID Pelanggan, bulan, dan tahun sebagai inputan. Pelanggan hanya perlu memasukan ID Pelanggan dan memilih bulan serta tahun untuk dapat mengetahui informasi tagihan listrik.



Gambar 12. Tampilan Informasi Tagihan

Gambar 5 menunjukkan hasil informasi tagihan listrik pada bulan Mei 2014 dengan ID Pelanggan seperti pada gambar. Pada tampilan informasi tagihan listrik juga terdapat informasi lainnya, seperti nama pelanggan, alamat, daya, pemakaian kwh, biaya beban, dan jumlah tagihan.

4.2 Halaman Statistik



Gambar 13. Tampilan Informasi Statistik

Pada menu statistik pelanggan dapat mengetahui grafik pemakaian listrik selama 6 bulan terakhir dalam satuan kwh seperti pada gambar 6. Untuk mendapatkan informasi pemakaian kwh, pelanggan harus memasukkan ID Pelanggan terlebih dahulu pada menu Tagihan.

5. PENGUJIAN

5.1 Whitebox

Dalam pengujian *white box* terdapat dua cara pengujian yaitu berbasis jalur dan kontrol terstruktur. Untuk pengujian sistem ini penulis menggunakan teknik dengan cara pengujian berbasis jalur. Berikut adalah langkah-langkahnya:

- a. Listing Program

- b. Diagram Alir
- c. Kompleksitas Siklomatis
- d. Basis Set
- e. Value Test

5.2 Blackbox

Pengujian ini dilakukan sebagai tahapan evaluasi yang dilakukan oleh pelanggan terhadap aplikasi tagihan listrik ini, apakah sesuai dengan tujuan atau belum. Evaluasi ini memastikan bahwa *user interface* aplikasi ini telah sesuai tujuan dari pelanggan dan dapat beroperasi dengan baik.

6. KESIMPULAN

Aplikasi Informasi Tagihan Listrik ini hanya membutuhkan id pelanggan untuk mengetahui informasi yang dibutuhkan pelanggan. Pelanggan hanya perlu memasukkan id pelanggan.

Aplikasi ini juga memberikan informasi yang lebih lengkap dibandingkan dengan web maupun sms. Aplikasi ini, selain memberikan jumlah tagihan setiap bulan, juga memberikan informasi lain seperti jumlah pemakaian kwh selama 6 bulan terakhir.

7. DAFTAR PUSTAKA

[1] Astuti, P. D. (2013). *Sistem Informasi Penjualan Obat Pada Apotek Jati Farma Arjosari*. Speed-Indonesian Journal on Computer Science, 10(1).

[2] <http://id.wikipedia.org/wiki/Aplikasi> (9 April 2014)

[3] Safaat, Nazruddin. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.

[4] Setiawan, I., Andjarwirawan, J., & Handojo, A. (2013). *Aplikasi*

Makassar Tourism Pada Kota Makassar Berbasis Android. Jurnal Infra, 1(2), pp-156.

[5] Maheshwari, Ms. Shikha. Jain, Dinesh Ch. (2012) "A Comparative Analysis of Different types of Models in Software Development Live Cycle". *International Journal od Advance Research in Computer Science and Software Engineering. Vol 2, Issue 5, Mei 2012, Hal 285-289.*