

**PERANCANGAN APLIKASI MOBILE TEMPAT WISATA
(Studi Kasus: WILAYAH BANDUNG)**

Budiman

Fakultas Teknologi dan Informatika, UNIBI

budiman@unibi.ac.id

Abstrak

Penggunaan *smartphone* memberikan manfaat bagi masyarakat, semua informasi dapat diakses dengan mudah melalui *smartphone*. Bandung merupakan kota *Paris Van Java*, Kota Bandung terkenal dengan keindahan dan kesejukan udaranya. Bandung memiliki banyak tempat wisata yang beberapa diantaranya belum diketahui oleh wisatawan yang akan berkunjung ke Bandung diantaranya wisata alam, kuliner, belanja dan hotel. Selain itu wisatawan yang berkunjung kesulitan untuk menuju tempat wisata karena belum dilengkapi dengan *Google Maps*. Metodologi penelitian yang dilakukan terdiri dari beberapa tahap, yaitu tahap perencanaan, tahap analisis kebutuhan dan tahap perancangan aplikasi. Pada setiap tahap terdapat beberapa tugas yang dilakukan supaya sistem yang dirancangan sesuai dengan kebutuhan. Berdasarkan perancangan aplikasi yang dilakukan, maka perancangan aplikasi tempat wisata ini dapat memberikan informasi kepada wisatawan mengenai tempat wisata di wilayah Bandung dan penggunaan fitur *Google Maps* pada aplikasi ini dapat memberikan kemudahan bagi wisatawan untuk berkunjung ke tempat wisata yang akan dikunjungi. Perancangan aplikasi ini dapat di implementasikan pada tempat wisata di wilayah lain.

Kata Kunci : Perancangan, Aplikasi, Wisata, Android

Abstract

The use of smartphones provide benefits to society, all information can be accessed easily through a smartphone. Bandung is the capital of Paris Van Java, Bandung is famous for its beauty and coolness. Bandung has many tourist attractions some of which not yet known by travelers who will pay a visit to Bandung include nature tours, culinary, shopping and hotels. In addition tourists visiting the difficulty to get to tourist attractions because it is not yet equipped with Google Maps. Research methodology undertaken consists of several stages, namely the planning phase, the design phase and needs analysis application. At each stage there are some tasks that are carried out so that the dirancangan system to suit your needs. Based on the design of the application is done, then the design of the application of these sights can provide information to travelers about tourist attractions in the area of Bandung and the use of the Google Maps feature on these applications can give ease for tourists to visit tourist attractions to be visited. The design of this application can be implemented on other area attractions.

Keywords: Design, Application, Tour, Android

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi semakin pesat, terutama penggunaan *Smartphone* yang kini menjadi kebutuhan dan gaya hidup masyarakat saat ini dalam menjalani aktifitas. Penggunaan *smartphone* memberikan manfaat bagi masyarakat, semua informasi dapat diakses dengan mudah melalui *smartphone* yang terhubung dengan jaringan internet mulai dari sarana komunikasi, berita *up to date*, bisnis *online*, pendidikan, permainan, dan lain-lain. Berdasarkan kemanfaatan dan pertumbuhan penggunaan *smartphone* menjadi peluang bagi peneliti, *developer* atau *programmer* dalam mengembangkan aplikasi *mobile*.

Bandung merupakan kota *Paris Van Java*, Kota Bandung terkenal dengan keindahan dan kesejukan udaranya. Selain itu Bandung terkenal dengan tempat wisata, kuliner dan lain-lain. Jumlah wisatawan pun semakin meningkat terutama warga Ibu Kota yang jaraknya tidak begitu jauh dari Bandung. Banyak wisatawan yang menghabiskan libur akhir pekan bersama keluarganya di Bandung.

Seiring berkembangnya teknologi yang menjadikan *Smartphone* sebagai kebutuhan dan meningkatnya jumlah wisatawan yang berkunjung ke Bandung maka penulis akan melakukan penelitian dengan judul Perancangan Aplikasi *Mobile* Tempat Wisata (Studi Kasus : Wilayah Bandung).

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka penulis mengidentifikasi permasalahan sebagai berikut:

1. Bandung memiliki banyak tempat wisata yang beberapa diantaranya belum diketahui oleh wisatawan yang akan berkunjung ke Bandung.
2. Wisatawan yang berkunjung kesulitan untuk menuju tempat wisata karena belum dilengkapi dengan Google Maps.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain:

1. Merancang aplikasi yang dapat memberikan kemudahan bagi wisatawan yang akan berkunjung ke Bandung dengan memperoleh informasi mengenai tempat wisata di wilayah Bandung.
2. Memberikan kemudahan wisatawan untuk menuju tempat wisata yang terintegrasi dengan *Google Maps*

1.4 Batasan Masalah

Untuk menghindari pembahasan yang melebar, maka masalah yang dibahas pada penelitian ini adalah sebagai berikut:

1. Aplikasi yang dirancang hanya membahas info tempat wisata wilayah Bandung diantaranya wisata kuliner, wisata hotel, wisata alam dan wisata belanja.
2. Tempat wisata wilayah Bandung mencakup Kota Bandung, Kabupaten Bandung dan Kabupaten Bandung Barat.

2. LANDASAN TEORI

2.1 Aplikasi

Menurut (Yuhefizar, 2012) Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Jadi Aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan.

Sedangkan menurut (Nazrudin Safaat H 2012 : 9) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

2.2 Android

Menurut (Kasman, 2015) Android merupakan sebuah sistem operasi telepon seluler dan *computer table* layar sentuh (*touch*

screen) yang berbasis Linux. Namun seiring dengan perkembangannya, Android berubah menjadi *platform* yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakangnya, yaitu Google. Google yang telah mengakuisisi Android, kemudian membuatkan sebuah *platform*.

Platform Android terdiri dari Sistem Operasi berbasis Linux, sebuah GUI (*Graphic User Interface*), sebuah *web browser* dan Aplikasi *End-User* yang dapat di *download* dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat.

2.3 Konsep Object Oriented Programming

Menurut (Widodo dan Herlawati, 2001) Bahasa C++, Java, VB.NET dan sejenisnya kita kenal sebagai bahasa pemrograman berorientasi objek. Tetapi hanya dengan menggunakan bahasa tersebut bukan berarti kita sudah pasti membuat program yang berorientasi objek. Perla diketahui bahwa kelas adalah tempat berkumpulnya objek yang merupakan ciri khas pemrograman berorientasi objek.

1. Analisis dan Desain Berorientasi Objek (*Object-Oriented Analysis and Design Process*)

Pemrograman berorientasi objek bekerja dengan baik ketika digabungkan dengan *Object-oriented Analysis and Design Process* (OOAD). (Wampler, 2001: 2) mengatakan jika kita membuat program berorientasi objek tanpa OOAD, seperti membangun rumah tanpa terlebih dahulu menganalisa apa saja yang dibutuhkan oleh rumah tersebut, tanpa perencanaan, tanpa blueprint, tanpa menganalisis ruangan apa saja yang diperlukan, berapa besar rumah yang akan dibangun dan sebagainya.

2. Objek (Object)

Menurut (Widodo dan Herlawati, 2001) orientasi objek merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan titik kulminasi dalam

menemukan cara yang efektif dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak saat ini. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek. Inti dari konsep ini adalah objek yang merupakan model dari sistem nyata.

Menurut (Douglas, 2004) objek adalah entitas yang memiliki atribut, karakter (*behaviour*) dan kadangkala disertai kondisi (*state*). Objek merepresentasikan sesuatu sistem nyata seperti siswa, sistem kontrol permukaan sayap pesawat, sensor atau mesin. Objek juga merepresentasikan sesuatu dalam bentuk konsep seperti nasabah bank, merek dagang, pernikahan atau sekedar *listing*. Bahkan bisa juga menyatakan visualisasi seperti, bentuk huruf (*font*), histogram, poligon, garis atau lingkaran. Semuanya memiliki fitur atribut (untuk data), *behaviour* (*operation* atau *method*), keadaan (memori), identitas dan tanggung jawab. Proses menjabarkan sistem nyata menjadi objek dinamakan abstraksi (*abstraction*). Abstraksi, mengeliminir aspek yang tidak perlu dalam suatu objek.

3. Kelas (*Class*)

Menurut (Widodo dan Herlawati, 2001) kelas adalah penggambaran satu set objek yang memiliki atribut dan *behaviour* yang sama. Kelas mirip tipe data pada pemrograman non objek, tapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kita dapat membentuk kelas baru yang lebih spesifik dari kelas general-nya. Kelas dan objek merupakan jantung dari pemrograman berorientasi objek. Untuk menghasilkan program jenis ini sangat penting untuk selalu berfikir.

4. Pembungkusan (*Encapsulation*)

(Nugroho, 2005: 6) mengartikan pembungkusan sebagai penggabungan potongan-potongan informasi dan perilaku perilaku spesifik yang bekerja pada informasi tersebut, kemudian mengemasnya menjadi apa yang disebut sebagai objek. Dalam perbankan kita mengenal objek rekening yang memiliki perilaku-perilaku misalnya buka, tutup, penarikan, penyimpanan, ubah nama, ubah alamat dan sebagainya. Akibatnya, perubahan-perubahan pada sistem perbankan yang

berkaitan dengan rekening-rekening dapat secara sederhana diimplementasikan satu kali saja pada objek rekening. Keuntungan lainnya adalah membatasi efek-efek perubahan pada sistem. Misalnya, saat manajemen bank menentukan jika seseorang memiliki rekening pinjaman di bank yang bersangkutan, rekening pinjaman itu harus dapat juga digunakan sebagai sarana bagi penarikan rekening.

5. Pewarisan (*Inheritance*) dan Generalisasi/Spesialisasi

Menurut (Whitten, 2004:411), konsep di mana metode dan atau atribut yang ditentukan di dalam sebuah objek kelas dapat diwariskan atau digunakan lagi oleh objek kelas lainnya. Sedangkan generalisasi atau spesialisasi merupakan teknik dimana atribut dan perilaku yang umum pada beberapa tipe kelas objek, dikelompokkan (atau diabstraksi) ke dalam kelasnya sendiri disebut *supertype*. Atribut dan metode kelas objek *supertype* kemudian diwariskan oleh kelas objek tersebut yang disebut *subtype*.

6. Polimorfisme

Menurut (Widodo dan Herlawati, 2001) *polimorfisme* berarti suatu fungsionalitas yang diimplementasikan dengan berbagai cara yang berbeda. Pada terminologi berorientasi objek, ini berarti kita dapat memiliki berbagai implementasi untuk sebagian fungsionalitas tertentu. Sebagai contoh, misalkan kita akan mengembangkan sistem berbasis gratis. Saat pengguna mau menggambar sesuatu, misalnya garis atau lingkaran, sistem akan memunculkan perintah gambar. Sistem akan mengenali berbagai bentuk gambar, masing masing dengan prilakunya sendiri-sendiri. Manfaat dari *polimorfisme* adalah kemudahan pemeliharannya. Jika perlu menambahkan gambar baru (misalnya segitiga), maka cukup menambahkan fungsi baru (fungsi menggambar segitiga) sedangkan fungsi umumnya (fungsi gambar) tidak mengalami perubahan (Nugroho, 2005: 10).

2.4 Unified Modelling Language

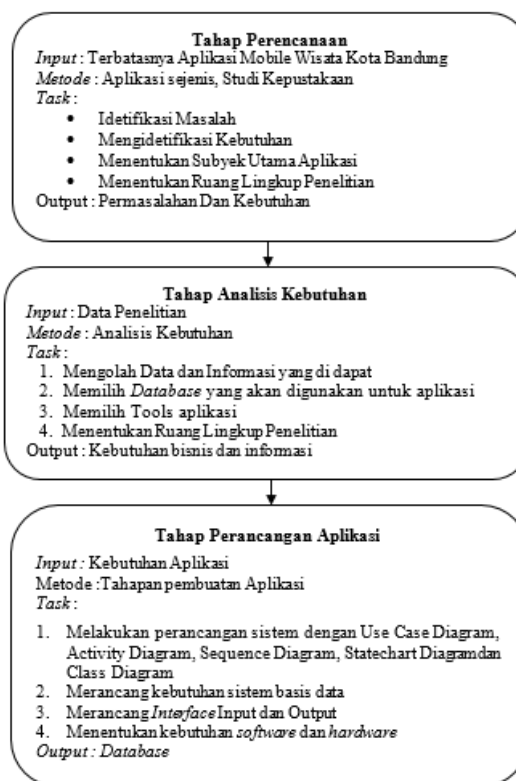
UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003: bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Menurut (Widodo dan Herlawati, 2001) berikut ini merupakan jenis diagram UML :

- a. Diagram Kelas. Bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
- b. Diagram Paket atau *Package Diagram*. Bersifat statis, diagram ini memperlihatkan kumpulan kelas-kelas. merupakan bagian dari diagram komponen.
- c. Diagram Use-Case. Bersifat statis, diagram ini memperlihatkan himpunan use-case dan aktor-aktor. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
- d. Diagram interaksi dan *Sequence*. Bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
- e. Diagram Komunikasi atau *Communication Diagram*. Bersifat dinamis, diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
- f. Diagram *Statechart* atau *Statechart Diagram*. Bersifat dinamis, diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status atau *state*, transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari Antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

- g. Diagram Aktivitas atau *Activity Diagram*. Bersifat dinamis, diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.
- h. Diagram Komponen (*Component Diagram*). Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
- i. Diagram *Deployment* (*Deployment Diagram*). Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

3. METODE PENELITIAN

Metode penelitian merupakan rancangan penelitian yang digunakan sebagai pedoman dalam melakukan penelitian. Adapun tahap yang dilakukan mulai dari tahap perencanaan, analisis kebutuhan, dan perancangan aplikasi. Berikut ini merupakan gambar metodologi penelitian yang dilakukan:

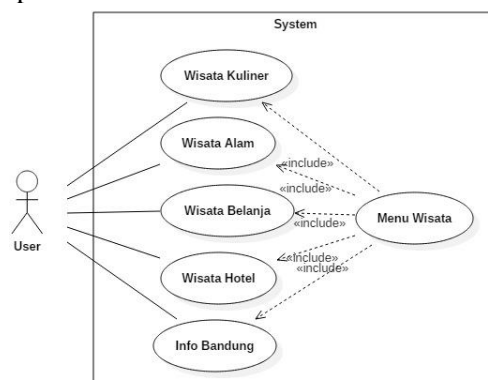


Gambar 3.1 Metode Penelitian

4. PERANCANGAN

4.1 Use Case Diagram

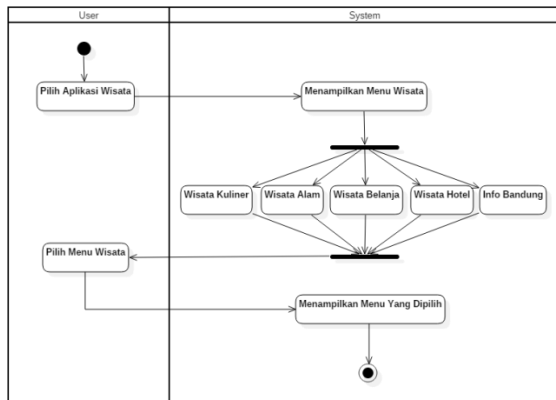
Pada perancangan aplikasi ini, *use case* menjelaskan tentang hubungan antara sistem dengan aktor. Aktor merupakan pengguna aplikasi, sedangkan sistem merupakan aplikasi wisata kota bandung. Hubungan yang terjadi dapat berupa input aktor ke sistem ataupun output dari sistem ke aktor.



Gambar 4.1. Use Case Diagram

4.2 Activity Diagram

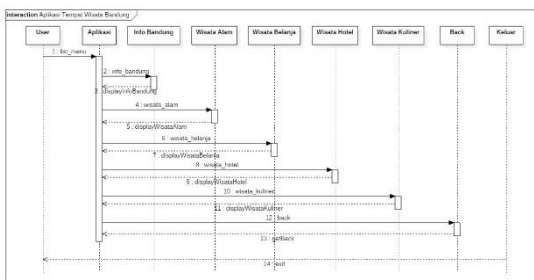
Untuk memodelkan *workflow* atau aliran kerja, aktifitas dan operasi. Dimodelkan dalam *activity diagram* yang disertai uraian tekstual dan menggambarkan aliran dari suatu aktivitas ke aktifitas lain yang terdapat pada sistem, berikut :



Gambar 4.2 Activity Diagram

4.3 Sequence Diagram

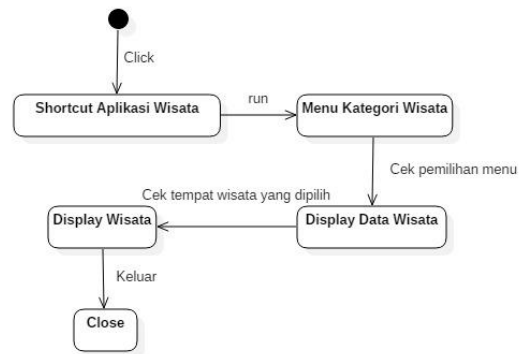
Menggambarkan aspek keterurutan waktu dari pesan yang disampaikan, dan/atau menggambarkan aspek struktur organisasi objek yang mengirim dan menerima pesan. Dimodelkan dengan *Sequence diagram* beserta uraian tekstual.



Gambar 4.3. Sequence Diagram

4.4 Statechart Diagram

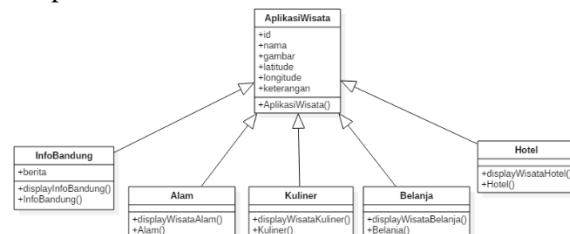
Diagram ini menggambarkan perilaku sistem aplikasi yang dibuat dan perilaku kelas, subsistem dan seluruh aplikasi. Diagram ini juga bermanfaat untuk menyediakan cara yang baik dalam memodelkan komunikasi yang terjadi dengan entitas luar melalui protokol.



Gambar 4.4. Statechart Diagram

4.5 Class Diagram

Class diagram atau diagram kelas merupakan inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* merupakan proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model. Pada *class diagram* terdapat atribut dan operasi yang menggambarkan perilaku suatu kelas. Di bawah ini merupakan *class diagram* aplikasi tempat wisata.



Gambar 4.5. Class Diagram

4.6 Perancangan Antar Muka

Perancangan antar muka merupakan bagian yang paling penting dalam merancang sistem. Pada bagian ini akan menjelaskan proses penggambaran bagaimana sebuah gambaran sistem dibentuk.



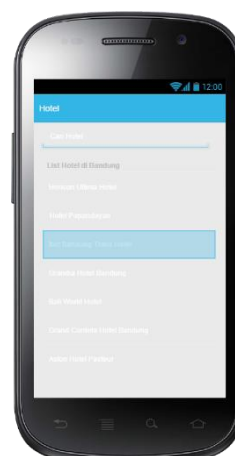
Gambar 4.6
Rancangan Antar Muka *Splash Screen*

Gambar di atas merupakan rancangan antar muka sebagai tampilan aplikasi pertama sebelum masuk ke menu utama.



Gambar 4.7.
Rancangan Menu Aplikasi

Rancangan menu aplikasi ini memberikan kemudahan bagi pengguna untuk memilih menu atau fasilitas yang disediakan oleh aplikasi dengan menggunakan *widget GridView* dan *Library Picasso* untuk membaca *resource image*.



Gambar 4.8.
Rancangan Antar Muka Daftar Hotel

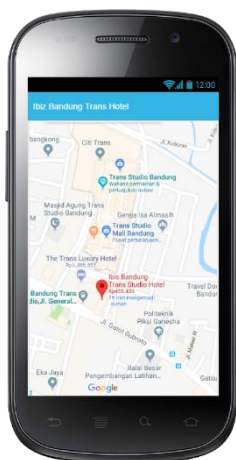
Rancangan antar muka diatas merupakan fitur dari aplikasi untuk menyediakan daftar hotel yang berada di bandung. Data hotel tersimpan di dalam database dan di tampilkan pada *widget ListView* yang dilengkapi dengan pencarian data hotel. Gambar rancangan di atas juga dapat digunakan untuk daftar Wisata Kuliner, daftar Wisata Alam, dan daftar Wisata Belanja.



Gambar 4.9.
Rancangan Antar Muka *Display Hotel*

Rancangan antar muka *display* hotel merupakan rancangan untuk menampilkan data hotel yang dipilih oleh pengguna. Rancangan di atas juga dapat digunakan untuk rancangan *Display Wisata Kuliner*, *Display Wisata Alam*, dan *Display Wisata Belanja* Melalui fitur ini pelanggan dapat mengunjungi *website* dan melihat peta lokasi hotel tersebut yang

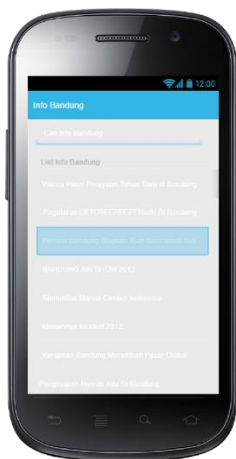
terintegrasi dengan *Google Maps* yang dapat dilihat pada gambar berikut:



Gambar 4.10

Rancangan Antar Muka Peta Lokasi

Rancangan antar muka di atas merupakan rancangan aplikasi yang terintegrasi dengan *Google Maps*. Pengguna aplikasi akan mendapatkan informasi lokasi tempat wisata di wilayah Bandung.



Gambar 4.11

Rancangan Antar Muka Daftar Info Bandung

Rancangan antar muka daftar info Bandung merupakan rancangan aplikasi untuk menyediakan daftar info seputar Bandung dengan menggunakan RSS Reader. Aplikasi ini dapat membaca *input stream* berupa XML yang disediakan oleh situs berita.



Gambar 4.12.

Rancangan Antar Muka Display Info Bandung

Gambar diatas merupakan rancangan antar muka *display* info Bandung berdasarkan daftar info Bandung yang dipilih oleh pengguna.

5. KESIMPULAN

Berdasarkan hasil perancangan dari penelitian ini, maka dapat disimpulkan sebagai berikut:

1. Perancangan Aplikasi Tempat Wisata ini dapat memberikan informasi kepada wisatawan mengenai tempat wisata di wilayah Bandung.
2. Penggunaan fitur *Google Maps* pada aplikasi ini dapat memberikan kemudahan bagi wisatawan untuk berkunjung ke tempat wisata yang dituju.
3. Perancangan aplikasi ini dapat digunakan untuk tempat wisata di wilayah lain selain Bandung.

6. REFERENSI

Chonoles, M.J., James A. Schardt. 2003. UML 2 for Dummies. Willey Publishing, Inc. New York.

Douglas. Bruce P. 2004. Real Time UML : Advance in The UML for Real – Time System, Third Edition. Pearson Education, Inc. Kanada.

Kasman, Akhmad Dharma. 2015. Trik Kolaborasi Android dengan PHP & MySQL. Lokomedia. Yogyakarta.

- Nazruddin Safaat H. 2012 (Edisi Revisi). Pemograman Aplikasi Mobile. Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung.
- Nugroho, Adi. 2005. Rational Rose untuk Pemodelan Berorientasi Objek. Informatika. Bandung.
- Tim EMS, Pemrograman Android dalam Sehari. 2015. PT Elex Media Komputindo. Jakarta.
- Wampler, Bruce E. 2003. The Essence of Object Oriented Programming with Java and UML. Butterworth-heinemann. United Kingdom.
- Widodo, Prabowo Pudjo. Herliawati. 2011. Menggunakan UML. Informatika. Bandung.
- Whitten, Jeffrey L., et al. 2004. Metode Desain dan Analisis Sistem (terjemahan). Andi Offset. Yogyakarta.
- Yuhefizar . (2012) . CMM Website Interaktif MCMS Joomla (CMS). Jakarta . Gramedia.