BAB 2

TINJAUAN PUSTAKA

2.1 Profil Dinas Budaya Dan Pariwisata Kota Bandung

Dinas Kebudayaan Dan Pariwisata Kota Bandung yang beralamat di Jl. Ahmad Yani No. 277 Kota Bandung, Jawa Barat. Aktivitas utama dinas budaya dan pariwisata adalah Pembinaan dan pelaksanaan tugas di bidang kebudayaan dan pariwista yang meliputi kebudayaan dan kesenian, sarana wisata, objek wisata dan pemasaran wisata. Pada pembangunan aplikasi pemandu wisata di kota Bandung ini data-data tempat wisata didapat dari dinas budaya dan pariwisata sebagai penndukung penelitian ini.

2.1.1 Sejarah Dinas Budaya Dan Pariwisata Kota Bandung

Dinas Kebudayaan dan Pariwisata yaitu jabatan segala sesuatu yang berhubungan dengan turisem atau perpelancongan. Pemerintah pusat dalam mengantur kepariwisataan di seluruh Indonesia mengeluarkan keputusan Presiden No. 30 Tahun 1986 yang berisi tentang pembentukan dewan Pertimbangan Kepariwisataan Nasional tersebut mempunyai tugas utama yaitu membantu Presiden dalam memetapkan kebijaksanaan umum dibidang kepariwisataan nasional. Di dalam keputusan ini dicantumkan pula tentang pembentukkan Direktorat Jendral Pariwisata dalam suatu Struktur Organisasi Departement Perhubungan. Masih pada tahun 1969, pemerintah mengeluarkan intruksi Presiden No. 9 /1969 yang berisi tentang pembentukan Badan Pariwisata Nasional (BAPAPARNAS) dalam rangka untuk menjamin pembinaan dan pembangunan yang efektif dan berlanjut dalam pelaksanaan yang diusahakan oleh pihak pemerintah dan swasta. BAPAPARNAS bertugas untuk membantu Menteri Perhubungan dan Dirjen Pariwisata [1].

Kemudian berdasarkan surat keputusan Menteri Perhubungan No. SK. 71/ 11/ 1969, dibentuklah Dinas Pariwisata di tingkat daerah (Dinas Pariwisata Daerah/

DISPARDA). Kemudian Pemerintah Daerah Tingkat Jawa Barat membentuk Dinas Pariwisata Tingkat I Jawa Barat, setelah itu dibentuklah Dinas Pariwisata Kotamadya Tingkat II Bandung berdasarkan peraturan daerah No. II/ 1969, pada tahun 1971 sampai dengan 1987 tercatat dengan nama Kantor Dinas Pariwisata (KAPARDA). Sejak keluarnya peraturan daerah No. II/ 1985 tentang pembentukan Dinas Pariwisata Kotamaya Daerah Tingakat II Bandung, maka sejak tahun 1987 KAPARDA diganti menjadi Dinas Pariwisata Kotamadya Daerah Tingakat II Bandung. Sejalan dengan adanya otonomi daerah Indonesia, seluruh tingkat Pemerintahan di Indonesia baik di pusat, propinsi, maupun kota dan kabupaten mengalami beberapa perubahan.

Bersarkan UU No. 22/ 1999, Dinas Pariwisata daerah Tingkat II Kota mengalami perubahan di antaranya yaitu :

1. Dinas Pariwisata daerah Tingkat II Kotamadya berubah namanya menjadi Dinas Pariwisata Kota Bandung.
2. Dinas Kota Bandung mendapatkan penambahan kewenangan yang lebih luas dalam urusan kepariwisataan di Kota Bandung, terhadap urusan Biro Perjalanan dan Hotel Berbintang.

Sehubungan dengan hal tersebut di atas, dalam upaya memenuhi kewenangan yang iberikan berdasarkan UU No. 22/ 1999 struktur organisasi dan tata kerja Dinas Pariwisata Kota Bandung pun mengalami beberapa Perubahan yang diatur dalm Peraturan Daerah No. 5 tanggal 7 maret 2001 mengenai struktur organisasi Dinas Pariwisata Kota Bandung.

2.1.2 Visi Dan Misi Dinas Bubudaya Dan Pariwisata Kota Bandung

2.1.2.1 Visi

Visi Dinas Kebudayaan dan Pariwisata Kota Bandung Tahun 2009 – 2013 adalah **”MEMANTAPKAN KOTA BANDUNG SEBAGAI KOTA SENI BUDAYA DAN TUJUAN WISATA TAHUN 2013”** Visi mengandung pengertian bahwa Kota Bandung sebagai Kota Seni Budaya (Puseur Budaya) dan

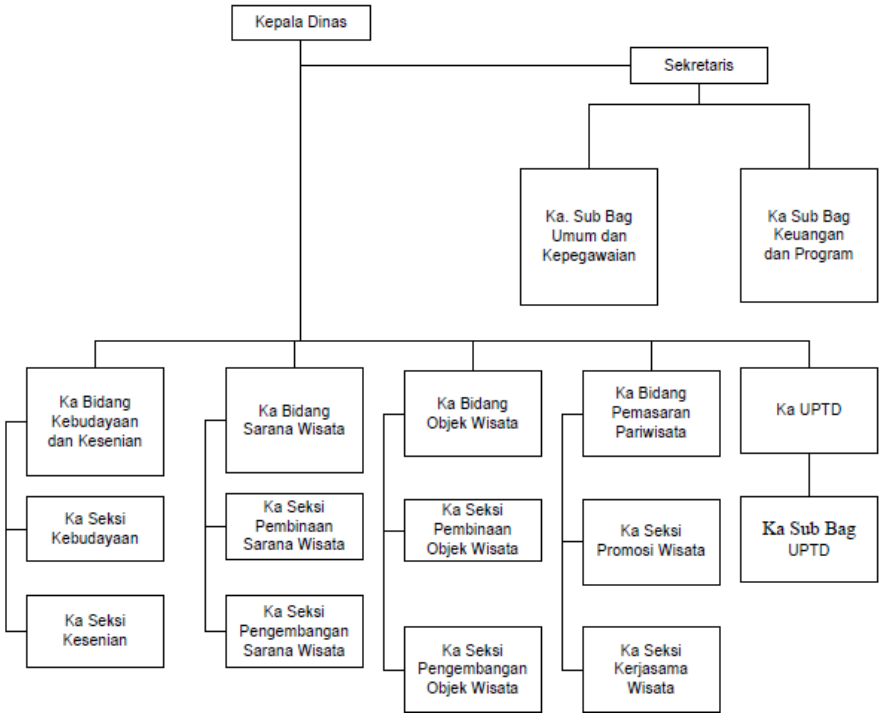
Kota Tujuan Wisata saat ini merupakan upaya yang secara *intens* di lakukan pada periode 2004 – 2008 akan dan perlu dimantapkan keberadaannya pada periode 2009 – 2013. Sehingga Kota Bandung ke depan benar-benar menjadi Kota Seni Budaya dan Tujuan Wisata yang berdaya saing tinggi sejajar dengan kota-kota lain di dalam dan luar negeri yang selama ini telah menunjukkan kiprahnya di bidang Budaya dan Pariwisata.

2.1.2.2 Misi

Mengembangkan Sumber Daya Manusia dan Kelembagaan Kepariwisataan yang Profesional, Berkarakteristik Sunda Dan Berwawasan Global. Meningkatkan pelestarian, pemberdayaan, pengembangan dan pemanfaatan kebudayaan dan kesenian. Mengembangkan industri pariwisata yang kreatif, inovatif dengan memperhatikan terlaksananya sapta pesona. Meningkatkan destinasi pariwisata kota yang berdaya saing tinggi baik pada tingkat regional, nasional maupun internasional. Meningkatkan pemasaran melalui kemitraan dan kerjasama budaya dan pariwisata dengan pemangku kepentingan dan atau Kab, kota atau negara lain.

2.1.3 Struktur Organisasi Dinas Budaya Dan Pariwisata Kota Bandung

Berdasarkan Peraturan Daerah Kota Bandung Nomor 13 Tahun 2007 tentang Pembentukan dan Susunan Organisasi Dinas Daerah, Struktur Organisasi Dinas Kebudayaan dan Pariwisata Kota Bandung adalah sebagai berikut: [1]



Gambar 2. 1 Struktur Organisasi Dinas Kebudayaan dan Pariwisata Kota Bandung

Berikut ini tugas dan wewenang petugas Dinas Kebudayaan dan Pariwisata Kota Bandung :

1. Kepala Dinas

   Kepala Dinas Kebudayaan dan Pariwisata mempunyai tugas pokok melaksanakan urusan pemerintah di bidang kebudayaan dan pariwisata berdasarkan asas otonomi dan pembantuan.

2. Sekretariat

   Sekretariat mempunyai tugas pokok melaksanakan sebagian tugas Dinas Kebudayaan dan Pariwisata lingkup kesekretariatan.

3. Sub Bagian Umum dan Kepegawaian

   Sub Bagian Umum dan Kepegawaian mempunyai tugas pokok melaksanakan sebagian tugas Sekretariat lingkup umum dan kepegawaian.

4. Sub Bagian Keuangan dan Program

   Sub Bagian Keuangan dan Program mempunyai tugas pokok melaksanan sebagian tugas Sekretariat lingkup keuangan dan program.

5. Bidang Kebudayaan dan Kesenian

   Bidang Kebudayaan dan Kesenian mempunyai tugas pokok melaksanakan sebagian tugas Dinas lingkup kebudayaan dan kesenian.

6. Seksi Kebudayaan

   Seksi Kebudayaan mempunyai tugas pokok melaksanakan sebagian tugas Bidang Kebudayaan dan Kesenian lingkup kebudayaan.

7. Seksi Kesenian

   Seksi Kesenian mempunya tugas pokok melaksanakan sebagian tugas Bidang Kebudayaan dan Kesenian lingkup kesenian.

8. Bidang Sarana Wisata

   Bidang Sarana Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Pengendalian lingkup Sarana Wisata.

9. Seksi Pembinaan Sarana Wisata

   Seksi Pembinaan Sarana Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Sarana Wisata lingkup Pembinaan Sarana Wisata.

10. Seksi Pengembangan Sarana Wisata

    Seksi Pengembangan Sarana Wisata mempunyai tugas pokok melaksanakansebagian tugas Bidang Sarana Wisata lingkup pengembangan sarana wisata.

11. Bidang Objek Wisata

    Bidang Objek Wisata mempunyai tugas pokok melaksanakan sebagian tugas Dinas lingkup Objek Wisata.

12. Seksi Pembinaan Objek Wisata

    Seksi Pembinaan Objek Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Objek Wisata lingkup pembinaan objek wisata.

13. Seksi Pengembangan Objek Wisata

Seksi Pengembangan Objek Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Objek Wisata lingkup Pengembangan Objek Wisata.

14. Bidang Pemasaran wisata
    Bidang Pemasaran Pariwisata mempunyai tugas pokok melaksanakan sebagian tugas Dinas lingkup pengwasan dan yustisi pemasaran pariwisata.
15. Seksi Promosi Pariwisata
    Seksi Promosi Pariwisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Pemasaran Pariwisata lingkup promosi pariwisata.
16. Seksi Kerjasama Wisata
    Seksi Kerjasama Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Pemasaran Pariwisata lingkup kerjasama wisata

2.2 Landasan Teori

Landasan Teori merupakan definisi, konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan pembangunan aplikasi Bandung *near you* untuk pemberitahuan tempat – tempat wisata di kota bandung dengan memanfaatkan teknologi *geofence* studi kasus DISBUDPAR kota Bandung, meliputi Wisata, pemandu wisata, tahahapan perencanaan paket wisata, Aplikasi, Pengertian Aplikasi, Klasifikasi Aplikasi, Pengertian *Mobile*, Pengertian Aplikasi *Mobile*, *Android*, *Geofence*, Jaringan Internet, GPS, LBS, *Google Maps*, HTML5, PHP7, *Javascript*, JSON, XAMPP, Database MySQL, *Unified Modeling Language* (UML) dan *Web Service*.

2.3 Wisata

Pengertian wisata menurut Intosh dan Goeldner (1984) mengartikan wisata sebagai ilmu, seni, dan bisnis yang berhubungan dengan menarik, memindahkan, mengakomodasikan secara ramah untuk memenuhi kebutuhan dan keinginan para pengunjung. Wisata merupakan segala sesuatu yang dapat menarik minat wisatawan untuk dating, melihat, dan melakukan aktivitas wisata. Sumber daya

terdapat dalam ruang atau disebut *geosfer* [5]. *Geosfer* merupakan objek wisata yang dapat dijabarkan sebagai berikut ;

1. Iklim dan cuaca, yaitu berupa panas, sejuk dan dingin. Masing-masing cuaca tersebut memiliki daya tarik sendiri dan dapat mempengaruhi aktivitas wisata yang dilakukan.
2. Bentuk lahan, seperti perbukitan, daratan, pegunungan, dan gunung api. Bentuk lahan ini berkorelasi dengan cuaca, sehingga menghasilkan berbagai argowisata, wisata gunung berapi, air panas, gua dan sebagainya, termasuk dalam kategori wisata alam.
3. Tata air, yaitu berupa danau, sungai dan laut, termasuk dalam kategori wisata alam.
4. Flora dan fauna menghasilkan objek wisata yang berupa kebun binatang, taman kota, termasuk dalam kategori wisata edukasi.
5. Kehidupan manusia yang akan menghasilkan objek wisata social dan budaya, berupa adat, kesenian, bangunan bersejarah, hasil teknologi modern seperti dunia fantasi, termasuk dalam kategori wisata sejarah dan budaya.

2.3.1 Pemandu Wisata

Pramuwisata adalah profesi di bidang kepariwisataan. Pramuwisata disebut juga Pemandu Wisata atau *Tour Guide* dalam Bahasa Inggris. Di Indonesia, secara nasional telah dibentuk organisasi yang mewadahi profesi ini, yaitu Himpunan Pramuwisata Indonesia atau HPI. Organisasi ini telah memiliki jaringan ke seluruh provinsi di Indonesia. Di beberapa daerah juga terbentuk sejumlah organisasi serupa yang bersifat lokal. Dalam pembangunan aplikasi pemandu wisata, diharapkan aplikasi yang akan dibuat bisa memberikan rekomendasi tempat wisata dan memberikan informasi detail tentang tempat wisata tersebut.

2.3.2 Tahapan Perencanaan Paket Wisata

Perencanaan tahapan paket wisata yang baik bukanlah sekadar khayalan tanpa dasar. Perencanaan merupakan pemikiran terhadap kegiatan di masa

mendatang yang didasarkan pada pertimbangan-pertimbangan rasional dan data-data yang akurat. Untuk itu dalam melakukan perencanaan paket wisata perlu dipahami tahapan -tahapan seperti tampak pada bagan berikut ini [6]:



Gambar 2. 2 Tahapan perencanaan paket wisata

Dari bagian di atas dapat dijelaskan sebagai berikut :

1. Pencarian Gagasan

   Sumber utama gagasan-gagasan produk paket wisata adalah pasar. Gagasan pasar merupakan kebutuhan dan keinginan para wisatawan yang belum terpenuhi.
2. Merumuskan Tujuan Wisata

   Tujuan yang dirumuskan itu harus dapat menjawab pertanyaan -pertanyaan yang sudah umum dikenal dengan rumusan 5W2H, yaitu: apa (*What*), wisata apa yang akan disusun; mengapa (*Why*), mengapa wisata itu disusun; siapa (*Who*), siapa saja yang akan terlibat dalam

wisata tersebut; dimana (*Where*), dimana wisata itu diselenggarakan; kapan (*When*), kapan wisata tersebut diselenggarakan; bagaimana (*How*), bagaimana wisata itu diselenggarakan; dan berapa banyak (*How Much*), berapa besar biaya yang dikeluarkan oleh wisatawan.

3. Observasi dan Pengumpulan Data

   Observasi pada dasarnya adalah pengejah wantahan tujuan yang telah dirumuskan dan menghubung-hubungkan antara hipotesis dengan kenyataan di lapangan. Hal -hal yang diobservasi adalah seluruh masalah yang dipertanyakan dalam rumusan tujuan wisata. Komponen-komponen pokok paket wisata yang wajib mendapat perhatian pada saat melakukan observasi dan pengumpulan data adalah *tour leader* (pengatur wisata), transportasi, akomodasi, restoran, obyek dan atraksi wisata dan toko cinderamata.

4. Analisis Data

   Data yang telah diperoleh dalam kegiatan observasi diolah dan dianalisis. Analisis data dimaksudkan untuk: menentukan strategi pencapaian tujuan; mengidentifkasi kendala yang mungkin timbul dalam proses pencapaian tujuan; dan mencari alternatif -alternatif yang mungkin dapat ditempuh.

5. Desain Terakhir

   Hasil dari evaluasi paket wisata yang telah diujikan atau dioperasionalkan akan dipakai sebagai rujukan untuk memperbaiki kekurangan atau kelemahan paket wisata tersebut dengan melakukan modifikasi -modifikasi. Selanjutnya hasil dari modifikasi tersebut dijadikan desain akhir untuk segera dipasarkan sesuai dengan pasar -pasar yang dituju. Dalam tahap ini pula penetapan standar dan prosedur pelayanan ditentukan.

2.4 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer

dapat memproses *input* menjadi *output*. Menurut Kamus Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Aplikasi dibagi menjadi tiga katergori adalah sebagai berikut ;

1. Aplikasi Berbasis *Web*

   Aplikasi berbasis web atau disebut juga *web base*, merupakan jenis aplikasi berbasis teknologi *web browser*. Artinya, aplikasi ini berjalan jika ada *browser* dan koneksi internet. Selama ada *browser* dan jaringan internet, aplikasi berbasis *web* dapat diakses dengan mudah di sistem operasi apapun.
2. Aplikasi Berbasis *desktop*

   Aplikasi *desktop* merupakan aplikasi yang dioperasikan dengan cara menginstalnya terlebih dulu di sistem operasi. Sesuai namanya, aplikasi ini mengacu pada program yang di instal pada perangkat desktop (PC dan laptop) mampu beroperasi secara *offline*, tetapi kita harus menginstalnya sendiri pada laptop atau komputer.
3. Aplikasi *Mobile*

   Pada perkembangan selanjutnya, ketika perangkat *smartphone* populer, maka berkembang pula aplikasi ke arah baru, yakni aplikasi berbasis *mobile*. Aplikasi berbasis *mobile* ini sedikit berbeda dengan aplikasi *web*. Aplikasi berbasis *mobile* umumnya ditujukan untuk perangkat *mobile* seperti *smartphone*. Karena bentuk *smartphone* yang lebih kecil daripada *desktop*, maka otomatis tampilan pada aplikasi *mobile* disesuaikan dengan kebutuhan dengan pengguna *smartphone*.

Aplikasi memungkinkan kita mempermudah pekerjaan menjadi lebih mudah dan efisien dibandingkan dengan proses yang belum terkompurisasikan seperti pencatatan, pencarian dokumen dll. Aplikasi yang akan dibangun dengan *platform*

mobile dan sistem operasinya adalah *android*, dikarenakan saat ini *android* memiliki *market share* sebesar 87,7% dari seluruh *smartphone* yang diaktifkan. Berdasarkan dari uraian masalah yang dipaparkan diatas, maka aplikasi menjadi sebuah media yang akan dibangun dengan tujuan untuk memenuhi kebutuhan wisatawan dalam menentukan tujuan wisatanya[8].

2.5 Aplikasi *Mobile*

aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, dan penggunaan. Secara istilah aplikasi adalah program siap pakai yang melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Kata mobile mempunyai arti bergerak atau berpindah sehingga aplikasi mobile merupakan sebutan untuk aplikasi yang berjalan di *mobile device*.

Aplikasi berbasis *mobile* sendiri terbagi menjadi dua, aplikasi berbasis *mobile web* dan *native apps*, penjelasannya adalah sebagai berikut ;

1. Aplikasi berbasis *mobile web (Hybird)*

   Aplikasi *hybrid* adalah aplikasi *web* yang ditransformasikan menjadi kode native pada platform seperti iOS atau *Android*. Aplikasi *hybrid* biasanya menggunakan *browser* untuk mengijinkan aplikasi *web* mengakses berbagai fitur di *device mobile* seperti *Push Notification, Contacts,* atau *Offline Data Storage*. Beberapa *tools* untuk mengembangkan aplikasi *hybrid* antara lain *Phonegap*, *Rubymotion* dan lain-lain.

2. *Native* Aplikasi

   Aplikasi *native* adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk *platform* tertentu. Contoh populernya yakni penggunaan bahasa pemrograman *Objective*-C atau *Swift* untuk platform iOS (*Apple*). Adapun *platform android* yang menggunakan bahasa pemrograman *Java*.

Aplikasi *mobile* digunakan Pada pembangunan aplikasi pemandu wisata di kota Bandung ini, pada era maju seperti ini banyak orang menggunakan aplikasi *mobile* karena lebih mudah untuk digunakan, tampilan aplikasi *mobile* lebih nyaman dan dapat mendapatkan informasi lebih cepat. Jenis aplikasi yang dipakai adalah *hybird* aplikasi.

2.6 Android

Pada Bulan Oktober 2003 *Android*, *Inc*. Didirikan di Palo, California, oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White untuk mengembangkan "perangkat *smartphone* yang lebih sadar akan lokasi dan preferensi penggunanya". Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), *konsorsium* dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia

Setiap versi yang dirilis dinamakan Secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula berikut versi lengkapnya, *Cupcake* (1.5), *Donut* (1.6), *Éclair* (2.0 - 2.1), *Froyo* (2.2 - 2.2.3) , *Gingerbread* (2.3 – 2.3.7), *Honeycomb* (3.0 – 3.2.6), *Ice Cream Sandwich* (4.0 – 4.0.4), *jelly Bean* (4.1 – 4.3), *Kitkat* (4.4), *Lollipop* (5.0), *Marshmallow* (6.0) dan *Nougat* (7.0) [3].

Pada pembangun aplikasi pemandu wisata kota bandung, sistem operasi yang digunakan adalah adalah *Android* versi 4.4 (*Kitkat*) karena masih banyak dari pengguna *smartphone android* yang sistem operasi nya dibawah 5.0 (lolipop). Sehingga saat aplikasi yang di bangun bisa dipakai oleh semua pengguna[7].

2.6.1 Android Life Cycle

Aplikasi *android* terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen *android* yang

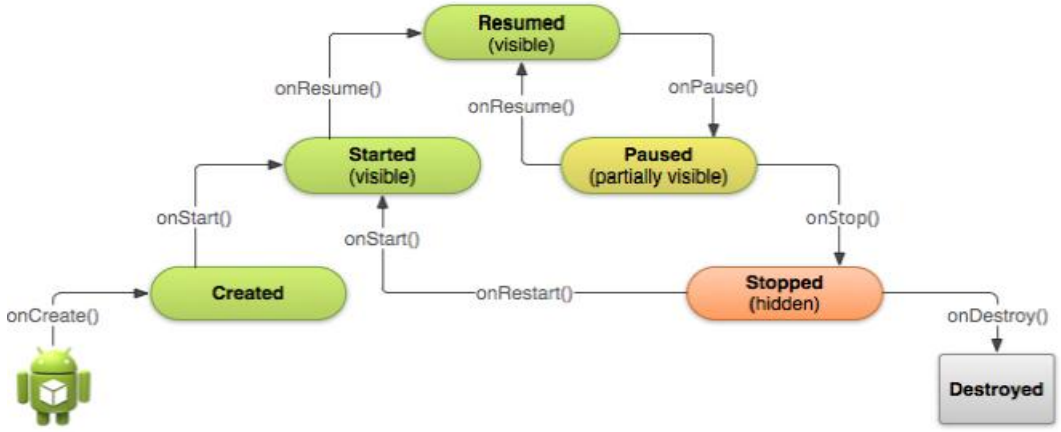
berbeda seperti ditunjukkan pada , klasifikasi tersebut berdasarkan kelas- kelas dasar *java* yang digunakan.

Tabel 2. 1 Komponen Aplikasi Android

Functionality	Java Base Class	Examples
Focused thing a user can do	Activity	Edit a note, play a game
Background process	Service	Play music, update weather icon
Receive messages	BroadcastReceiver	Trigger alarm upon event
Store and retrieve data	ContentProvider	Open a phone contact

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service, BroadcastReceiver. ContentProvider* [7].

android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem *android* menjalankan kode dalam *method Activity* dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi *desktop* atau *web*. Hal ini dikarenakan aplikasi *mobile* memiliki tingkat *interupsi* proses yang lumayan tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna .



Gambar 2. 3 Siklus Hidup *Android*

Terdapat beberapa *state* dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.3, Siklus Hidup *Android*, akan tetapi hanya beberapa dari *state* tersebut yang menjadi statis diantaranya:

1. *Resumed*

   Resumed terjadi ketika aplikasi berjalan setelah *state paused*. *State* ini akan menjalankan perintah program yang ditulis pada *method onResume*() (*Google Inc*,2014).

2. *Paused*

   Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam *state* ini (*Google Inc*, 2014).

3. *Stopped*

   Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service* di *background* (*Google Inc*, 2014).

State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cycle callback* berikutnya. Artinya, setelah sistem *OnCreate*() dipanggil, dengan cepat sistem akan memanggil *method OnStart*(), kemudian diikuti oleh *onResume*() (Schwarz, Dutson, Stelee, & To, 2013) [10].

2.7 *Global Positioning System* (GPS)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang *mikro* ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS [11], seperti:

1. GPS *Receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. GPS memberikan informasi lokasi dalam tiga dimensi yaitu, *Latitude, Longitude* dan *Elevasi.*
3. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. GPS dapat menunjukkan arah tujuan atau arah perjalanan.
5. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
6. Komulasi data. GPS *receiver* dapat menyimpan informasi *track*, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya.

GPS dapat melakukan perhitungan dan menentukan posisi user dan menampilkan dalam *maps*. Jika sudah dapat menyimpan posisi user selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah tuju, rute, tujuan perjalanan serta jarak tujuan. Dalam pembangunan aplikasi pemandu wisata kota

bandung teknologi GPS digunakan untuk perhitungan waktu yang akurat sangat menentukan informasi lokasi pengguna.

2.8 *Location Based Service* (LBS)

Location Based Service adalah service yang berfungsi untuk mencari dengan teknologi *Global Positioning Service* (GPS) dan *Google's cell-based location.* Map dan layanan berbasis lokasi menggunakan lintang dan bujur untuk menentukan lokasi geografis, namun sebagai user kita membutuhkan alamat atau posisi realtime kita bukan nilai lintang dan bujur [10], Dua unsur utama LBS adalah:

1. *Location Manager* (API Maps)

   Menyediakan *tools* atau *resource* untuk LBS, *Application Programming Interface* (API) *Maps* menyediakan fasilitas untuk menampilkan, memanipulasi *maps* atau peta beserta fitur - fitur lainnya seperti tampilan satelit, *street*(jalan), maupun gabungannya. Paket ini berada pada *com.google.android.maps*.

2. *Location Providers* (API *Location*)

   Menyediakan teknologi pencarian lokasi yang digunakan oleh *device* atau perangkat. API *Location* berhubungan dengan data GPS dan data lokasi *real-time*. API *Location* berada pada paket *Android* yaitu dalam paket *android location*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, gerakan atau perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

Teknologi ini dipakai dalam pembangunan aplikasi *mobile* pemandu wisata kota bandung, untuk menentukan lokasi geografis tempat wisata, Selain dapat mengetahui posisi pengguna, aplikasi LBS juga dapat menentukan posisi tempat-tempat tertentu. Dan dengan kombinasi ini, aplikasi LBS akan mencari rute untuk menghubungkan posisi pengguna dengan suatu tempat.

2.8.1 *Geofence*

Geofence adalah parameter *virtual* untuk wilayah geografis di dunia nyata. Parameter *virtual* yang ditetapkan ini bisa menjadi statis ataupun dinamis. Salah satu contoh khas dari *geofence* dinamis adalah radius sekitar toko atau titik lokasi. *geofence* statis dapat di atur dari batas yang telah ditetapkan, seperti zona kehadiran di sekolah atau batas lingkungan. Kostum digital *geofence* juga dapat digunakan, jika perangkat yang menyediakan layanan berbasis lokasi (LBS) pengguna masuk atau keluar *geofence*, perangkat akan menerima pesan *notifikasi* [12].

geofence adalah parameter virtual untuk wilayah geografis di dunia nyata. Parameter virtual yang ditetapkan ini bisa menjadi statis ataupun dinamis. Salah satu contoh dari *geofence* dinamis adalah radius sekitar *took* atau titik lokasi suatu objek tempat. *Geofence* dapat diatur dari batas yang telah ditetapkan, seperti zona kehadiran di sekolah atau batas lingkungan. Kostum digital *geofence* juga dapat digunakan, jika perangkat yang menyediakan layanan berbasis lokasi (LBS) pengguna masuk atau keluar dari *geofence*, perangkat akan menerima pesan *notifikasi*. *Notifikasi* ini berisi informasi tentang lokasi objek tertentu yang sudah ditandai sebelumnya.

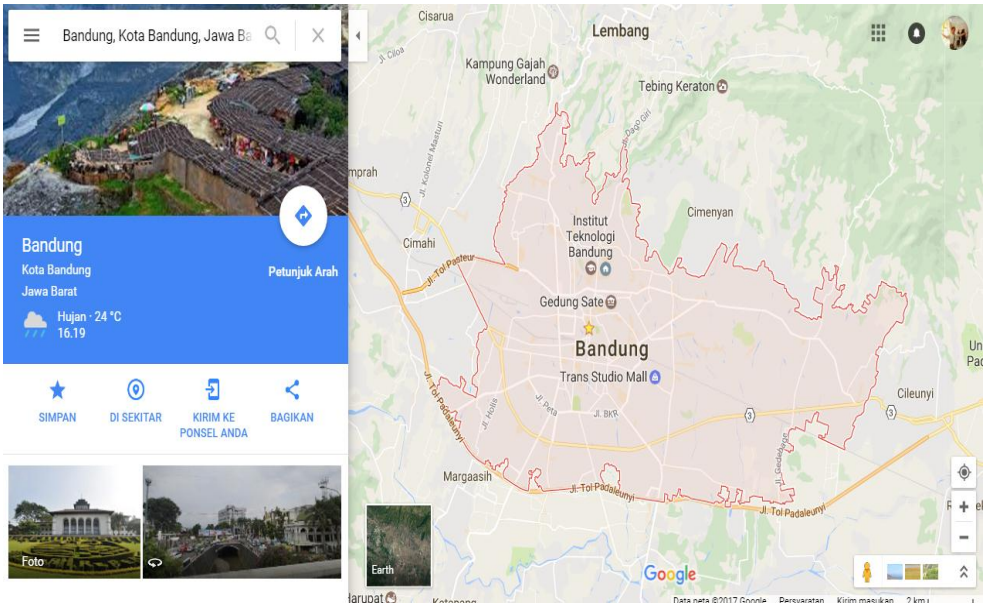
geofence menggabungkan kesadaran lokasi pengguna dengan kesadaran kedekatan pengguna pada lokasi *geofence*. Untuk menandai lokasi, di perlukan untuk menentukan lintang dan bujur. Pada pembangunan aplikasi pemandu wisata kota bandung ini teknologi *geofence* dipakai untuk mengetahui tempat-tempat wisata di sekitar area pengguna.

2.9 *Google Maps* API

Google maps adalah layanan gratis yang diberikan oleh *Google* dan sangat popular. *Google maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Cara membuat *google maps* untuk ditampilkan pada suatu *web* atau *blog* sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi Internet yang sangat stabil. Dengan menggunakan *Google maps* API, kita dapat menghemat waktu dan biaya untuk

membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan [10].

Google maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *Google maps* dalam *web* yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *google maps* API. *Google maps* API adalah suatu library yang berbentuk *JavaScript*. Layananan *Google maps* digunakan untuk menampilkan tempat wisata yang direkomendasikan dalam pembangunan aplikasi pemandu wisata kota bandung .



Gambar 2. 4 Tampilan Peta Pada Google Maps

2.10 *Unified Modeling Language* (UML)

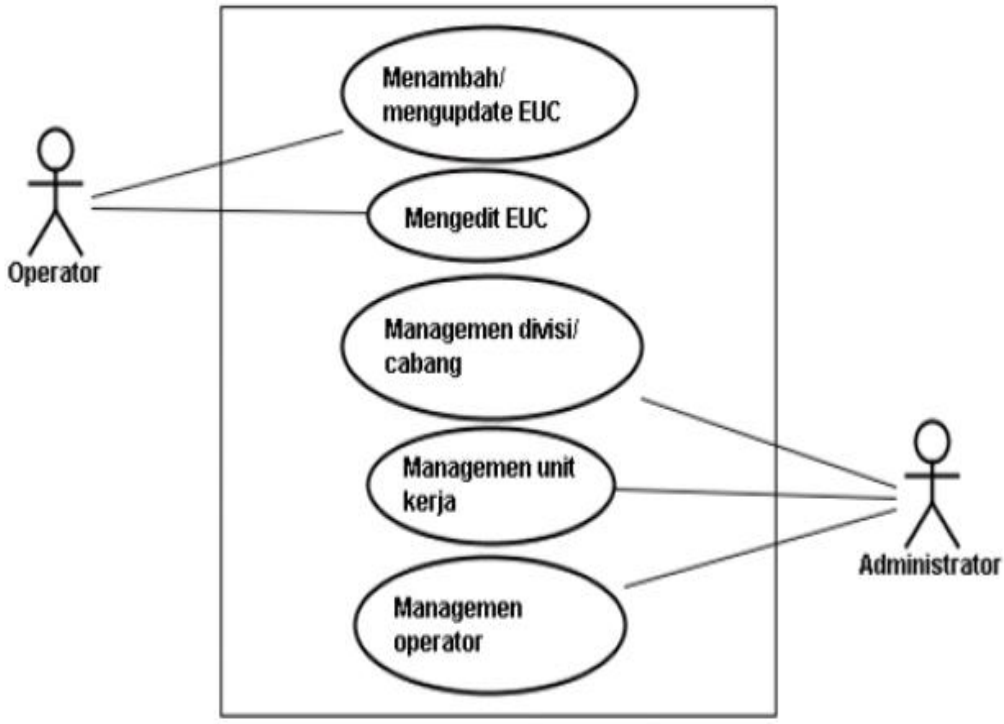
Metode analisis yang memerika *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. *Unified Modeling Language* (UML) adalah bahasa untuk *mengspesifikasi*, *memvisualisasi*, membangun dan mendokumentasi *artefacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak), seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML merupakan bahasa standar untuk penulisan

blueprint software yang digunakan untuk *visualisasi*, *spesifikasi*, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Berikut ini merupakan beberapa bagian dari UML adalah sebagai berikut :

1. *Use Case Diagram*

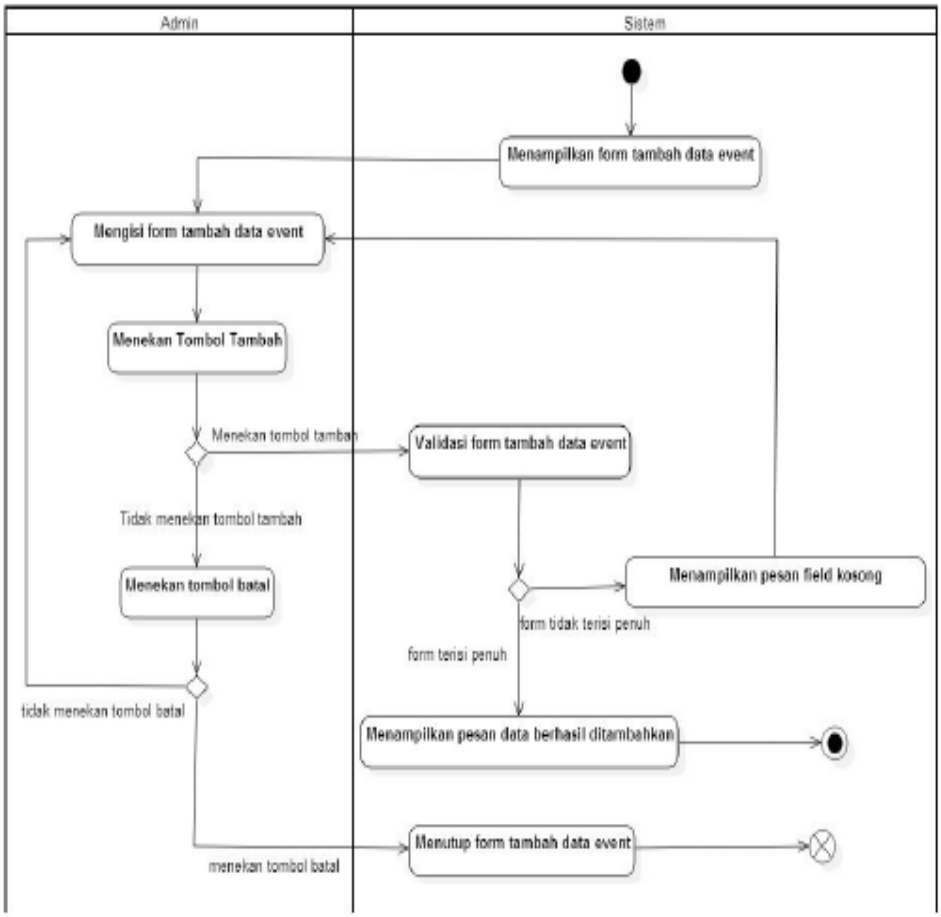
   Use case diagram menggambarkan sejumlah eksternal aktor dan hubungannya ke *use case* yang diberikan oleh sistem.



Gambar 2. 5 Contoh *Use Case Diagram*

2. *Activity Diagram*

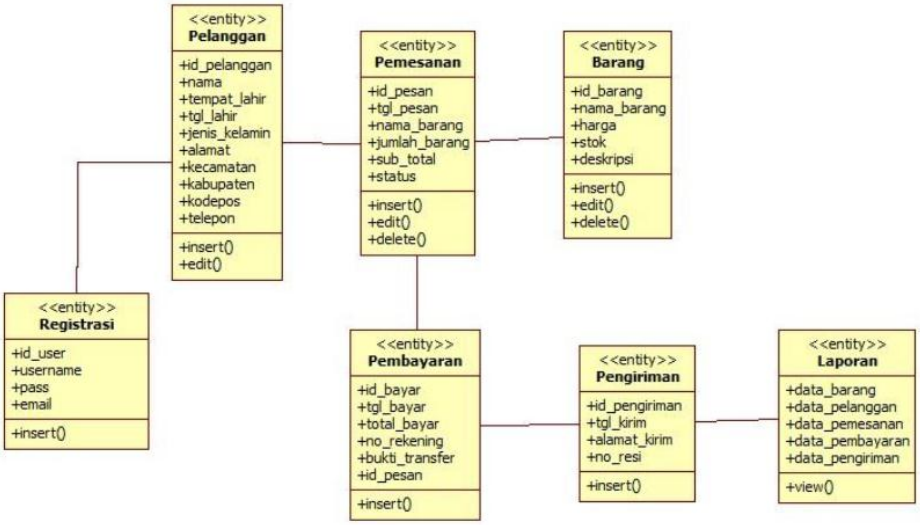
   Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti *use case*.



Gambar 2. 6 Contoh *Activity Diagram*

3. *Class Diagram*

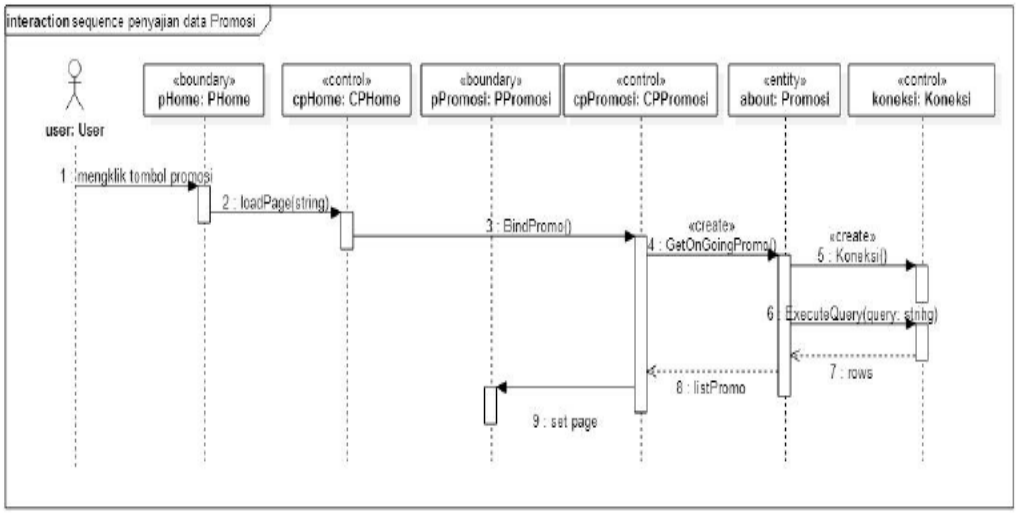
   Class diagram menggambarkan struktur dan deskripsi class, *package* (paket) dan objek beserta hubungan satu sama lain seperti *containment* (penahanan), pewarisan, asosiasi dan lain-lain. Dapat dilihat pada gambit 2. 7 Contoh gambar *class diagram.*



Gambar 2. 7 Contoh Gambar *Class Diagram*

4. *Sequence Diagram*

   Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



Gambar 2. 8 Contoh *Squence Diagram*

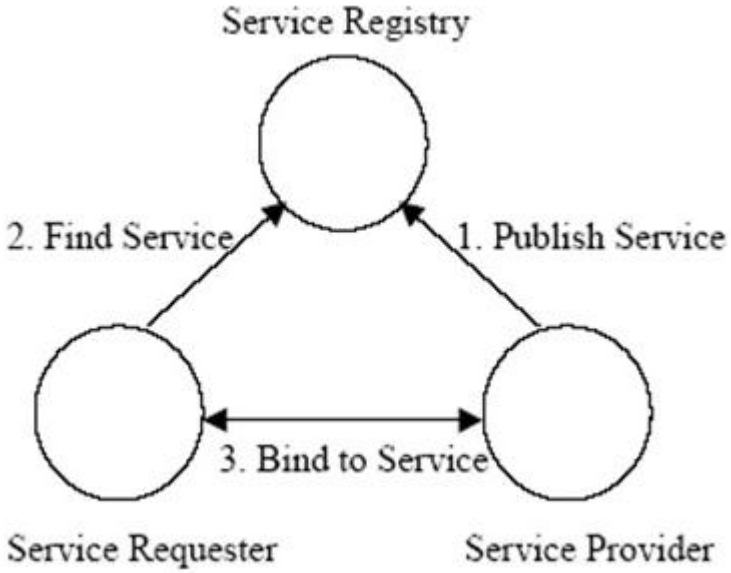
2.11 *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *web site* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan web *service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa compiler [16].

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya.

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan *registrasi* khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan *otorisasi*.
3. *Web service* berjalan di *port* 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

Web Service memliliki tiga entitas dalam arsitekturnya yaitu, *service requester* (peminta layanan), *service provider* (penyedia layanan) dan *service registry* (daftar layanan) berikut gambaran dan penjelasan dari arsitektur web *service*; [13].



Gambar 2. 9 Arsitektur *Web Service*

1. *Service provider*, berfungsi untuk menyediakan layanan dan mengolah sebuah *registry* agar layanan tersebut dapat tersedia
2. *Service resgistry*, berfungsi sebagai lokasi tengah yang mendeskripsikan semua layanan yang telah di register.
3. *Service requestor*, berfungsi sebagai peminta layanan yang menacri dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2.12 *Representational State Transfer (REST)*

REST adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula *REST* bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET, POST, PUT* atau *DELETE*. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan di *parsing* disisi *Client* [14].

Dalam pengaplikasiannya, *REST* lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktifitas yang mengolah *resource* itu. Alasan mengapa *REST* tidak digunakan dalam skripsi ini karena orientasi pada resourcenya itu, sedangkan aplikasi *event* calendar membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan resource event. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *web service* yang menggunakan *REST* adalah: *Flickr* API (Application *ProgramInterface*), YouTube API, Amazon API.

Pada pembangunan aplikasi pemandu wisata kota bandung untuk pembuatan backend digunakan *Rest API* digunakan untuk memasukkan data – data pendukung wisata kota Bandung dan sebagai perantara penampung data untuk aplikasi pemandu wisata data, dan *REST* dapat mengeluarkan data berupa json, xml dan text format.

2.13 *Hypertext Prepocessor* (PHP)

PHP singkatan dari *Personal Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua script PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP pertama kali di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, namun sekarang di ambil oleh oleh *The PHP Group*. Pada awalnya PHP adalah singkatan dari *Personal Home Page*, namun dalam perkembangannya, di ubah menjadi PHP*: Hypertext Preprocessor*, sebuah kepanjangan *rekursif* PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *ServerSideScripting*.

Artinya bahwa dalam setiap menjalankan PHP, wajib adanya web server.PHP ini bersifat *opensource* sehingga dapat dipakai secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *Windows* maupun *Linux* [15].

PHP juga dibangun sebagai modul pada web apache dan sebagai binary yang dapat berjalan sebagai CGI. Pada pembangunan aplikasi pemandu wisata Kota Bandung khususnya halaman *backend* menggunakan *script* PHP untuk membuat halaman pelaporan. Berikut contoh skrip dasar PHP.

```
<html>
<head>
<title>Contoh Kode PHP </title>
</head>
<body>
<?php
echo ("Ini adalah Kode PHP yang diletakan di antara kode HTML");
?>
</body>
</html>
```

Gambar 2. 10 Contoh Skrip PHP

2.14 HTML 5

HTML5 (*Hypertext Markup Language version* 5) adalah sebuah bahasa markah yang menstrukturkan isi dari *World Wide Web*, sebuah teknologi utama pada internet. Standar HTML5 menyempurnakan elemen-elemen lama yang terdapat pada standar sebelumnya, menambahkan elemen-elemen yang lebih semantik dan menambahkan fitur-fitur baru untuk mendukung pembuatan aplikasi *web* yang lebih kompleks [16].

HTML5 merupakan salah satu karya *Konsortium* Waring Wera Wanua (*World Wide Web Consortium*, W3C) untuk mendefinisikan sebuah bahasa markah tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak pembuat *web*.

Pada pembangunan aplikasi pemandu wisata kota Bandung khususnya halaman *backend* menggunakan *script* HTML5 untuk membuat halaman *administrator*. Karena sudah *responsive* atau bisa menyesuaikan bentuk halaman sa Berikut contoh skrip struktur umum HTML5:

Skrip HTML

```
<!DOCTYPE html>
<html>
    <head>
        <title>Web Warung Tegal</title>
        <style type="text/css">
        </style>
        <!--[if lt IE 9]>
        <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
        <![endif]-->
    </head>
    <body>
        <div class="wrapper">
        </div><!-- .wrapper -->
    </body>
</html>
```

Gambar 2. 11 Contoh Skrip HTML 5

2.15 *Sublime Text* 3

Sublime text adalah *teks editor* berbasis *Python*, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan *developer* (pengembang), penulis dan *desainer*. Para programmer biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang ia kerjakan. Sampai saat ini *sublime text* sudah mencapai versi 3. *support* terhadap banyak bahasa pemrograman mulai dari ASP, C, C#, C++, PHP, HTML, dan masih banyak lagi [17].

Pada pembangunan aplikasi pemandu wisata, menggunakan *Tool Sublime text* 3, karna bahasa pemogramannya adalah *javascript*. *Sublime text* mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*.

1. *Multiple Selection*

   Multiple Selection mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. *multiple selection* ini juga merupakan salah satu fitur unggulan dari *Sublime Text* 3. Kita dapat meletakkan kursor pada kode yang akan di

ubah, lalu tekan Ctrl+klik atau blok kode yang akan diubah kemudian Ctrl+D setelah itu kita dapat merubah kode secara bersamaan.

2. *Command Pallete*

   Command Pallete mempunyai fungsi yang berguna untuk mengakses *file shortcut* dengan mudah. Untuk mencari *file* tersebut kita dapat tekan Ctrl+Shift+P, kemudian cari perintah yang kita inginkan.

3. *Distraction Free Mode*

   Fitur ini mempunyai fungsi untuk merubah tampilan layar menjadi penuh dengan menekan SHIFT + F11. Fitur ini sangat dibutuhkan ketika pengguna ingin fokus pada pekerjaan yang sedang dikerjakannya.

4. *Find Profject*

   Fitur ini kita dapat mencari dan membuka file di dalam sebuah *project* dengan cepat dan mudah. Hanya dengan menekan Ctrl+P anda dapat mencari *file* yang diinginkan.

5. *Pluggin API Switch*

   Sublime Text mempunyai keunggulan dengan plugin yang berbasis *Python Plugin* API. Teks editor ini juga mempunyai plugin yang sangat beragam, dan ini dapat memudahkan pengguna dalam mengembangkan *softwarenya*.

6. *Drag and Drop*

   Dalam *text* editor ini pengguna dapat menyeret dan melepas *file text* ke dalam editor yang akan membuka tab baru secara otomatis.

7. *Split Editing*

   Di dalam fitur ini pengguna dapat merubah *file* secara berdampingan dengan klik *File->New menu into file.*

8. *Multi Platform*

   Sublime Text juga mempunyai keunggulan dalam berbagai *platform*. *Sublime text* sendiri sudah tersedia dalam berbagai *platform* sistem operasi, yaitu Windows, Linux, dan MacOS.

2.16 Android Studio

Android Studio adalah IDE resmi untuk membangun aplikasi Android berdasarkan IntelliJ IDEA. IntelliJ IDEA sendiri adalah *Java Integrated Development Environtment* (IDE) yang dikembangkan oleh JetBrains, untuk mengembangkan perangkat lunak komputer. IntelliJ IDE berfungsi dalam membantu anda di dunia percodingan baik dari segi navigasi, penyokong produktivitas, hingga code editor yang cerdas[19]. Berdasarkan website http://developer.android.com, android studio memiliki keunggulan:

1. Android studio didasarkan pada intellij DEA yang memberikan kemungkinan tercepat dalam melakukan coding dan *running workflow* sehingga meng-*coding* dan melakukan iterasi dapat dilakukan dengan cepat.
2. Struktur proyek Android Studio dan pembangunannya yang berbasis *gradle* memberikan fleksibilitas yang dibutuhkan untuk menghasilkan APK untuk semua jenis *device* sehingga konfigurasi pemangunan dapat dlakukan tanpa batas.
3. Android Studio menyediakan fitur-fitur yang membuat pengembang menjadi percaya diri dan dapat membuat kode terbaik.
4. *Android Studio* mengetahui bahwa tidak seluruh kode ditulis dengan bahasa java dan tidak semua kode dapat berjalan di *device* pengguna. Dengan fitur-fitur yang dimiliki *Android Studio* maka aplikasi menjadi kaya dan terkoneksi satu sama lain.
5. *Android Studio* menyediakan alat GUI untuk mempermudah pengembang dalam merancang tampilan aplikasi sehingga pengembang tidak lelah.

Dengan menggunakan *android studio* pengembang aplikasi dapat mengerjakan aplikasi yang akan dibangunya. Pada pembangunan aplikasi pemandu wisata Kota Bandung *android studio* digunakan untuk menjalankan atau *debugging* program yang dibangun untuk mencari dan mengurangi bug, atau kerusakan didalam sebuah program, sehingga aplikasi dapat bekerja sesuai dengan harapan.

2.17 *Javascript*

JavaScript pada awalnya diciptakan untuk menangani validasi dari sebuah formulir dalam *website* yang sebelumnya ditugaskan pada bahasa pemograman *server* seperti *Perl*. Namun dengan berkembangnya zaman *JavaScript* menjadi pemain penting dalam perkembangan teknologi *website. JavaScript* adalah bahasa *scripting* yang dibuat untuk berinteraksi dengan halaman *website* dan dibuat dengan 3 bagian [16] :

1. *ECMAScript*, yang didefinisikan dalam ECMA-262 dan menjadi bagian utama dalam menjalankan fungsionalitas.
2. *Document Object Model* (DOM), yang menyediakan *method* dan *interface* untuk bekerja dengan isi dari halaman *website*.
3. *Browser Object Model* (BOM), yang menyediakan *method* dan *interface* untuk berinteraksi dengan *browser*.

2.18 *Javascript Object Notation* (JSON)

JavaScript Object Notation (JSON) pertama kali dikenalkan oleh Douglas Crockford adalah bagian dari *JavaScript*, dibuat dengan menggunakan pola dari JavaScript untuk merepresentasikan data terstruktur. Crockford membuat JSON sebagai alternatif dari *(eXtended Markup Language*) XML untuk mengakses data terstruktur dari *JavaScript* dengan menggunakan *method eval()* dan tidak membutuhkan pembuatan dari DOM. Bagian terpenting dari JSON adalah format data bukan bahasa pemrogramannya. JSON bukan bagian dari *JavaScript* walaupun mereka saling berbagi *syntax*, JSON juga tidak semata-mata digunakan oleh *JavaScript*. JSON merupakan data format, jadi JSON memiliki *parsers* dan *serializers* yang ada dalam banyak bahasa pemograman. JSON terbuat dari dua struktur [14]:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative* array.

2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data *universal*. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini.

2.19 Database MySql

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang *multithread*, dan *multi-user*. MySQL adalah implementasi dari system manajemen basisdata *relasional* (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. Dalam MYSQL ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu:

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris [19]. Dalam MYSQL ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu:

1. *Create*

   Create adalah operasi penambahan data baru kedalam table. Querynya adalah sebagai berikut :

INSERT INTO table_name(column1,column2,column3,...)

VALUES (value1,value2,value3,...);

2. *Read*

   Read adalah operasi untuk menampilkan semua atau sebagian data yang berada di dalam database. Querynya adalah sebagai berikut:

   SELECT column_name,column_name
   FROM table_name;

3. *Update*

   Update adalah operasi untuk mengubah data yang ada di dalam database. Querynya adalah sebagai berikut:

   UPDATE table_name SET column1=value1,column2=value2,...

   WHERE some_column=some_value;

4. *Delete*

   Delete adalah operasi untuk menghapus data yang ada di dalam table. Querynya adalah sebagai berikut:

   DELETE FROM table_name

   WHERE some_column=some_value;

2.20 XAMPP

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia database *server* mysql dan *support* php *programming*. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache* Web Server, MySQL Database Server,

PHP *Support* (PHP 4 dan PHP 5) dan beberapa *module* lainnya. Hanya bedanya kalau yang versi untuk Windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah *server* secara *grafis*, sedangkan Linux masih berupa perintah-perintah di dalam console. Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan [20].

Dalam XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman web XAMPP berfungsi sebagai *server*

offline yang berdiri sendiri (seringkali disebut sebagai *localhost*). XAMPP terdiri dari beberapa program yaitu *Apache* HTTP Server, MYSQL *Database*, PHP, dan Pearl XAMPP juga dilengkapi fitur manajemen *database PHPMyAdmin* seperti pada *server hosting* sungguhan, sehingga pengembang *web* dapat mengembangkan aplikasi *web* berbasis database secara mudah.

2.21 Metode Pengujian Sistem

Metode pengujian sistem terdiri dari Pengujian *black box* dan *beta* yang dilakukan untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan.

2.21.1 Pengujian *Black Box*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianalogikan seperti melihat suatu kotak hitam yang hanya bisa dilihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. *Black box* testing melakukan evaluasi untuk menemukan kesalahan dalam kategori berikut[21]:

1. Fungsi tidak benar atau hilang.
2. Kesalahan *interface* atau antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja atau perilaku dan kesalahan inisialisasi dan terminasi.

2.21.2 *Skala Likert*

Skala likert adalah suatu skala *psikometrik* yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama *skala Likert*. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena [22].

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan *negative* yang berfungsi untuk mengukur sikap *negative* objek. Skor pernyataan positif dimulai dari 1 untuk sangat tidak setuju (TS), 2 untuk tidak setuju (S), 3 untuk raguragu (CS), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (CS), 4 untuk tidak setuju (KS), dan 5 untuk sangat tidak setuju (TS). Beberapa menghilangkan option "Ragu-ragu" dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. *Skala Likert* digunakan untuk mengukur kesetujuan dan ketidak setujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Kekurangan *Skala Likert*:

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai error dari respons yang terjadi.

Kelebihan *Skala Likert*:

1. Mudah dibuat dan di terapkan.
2. Skala Likert lebih mudah membuatnya dibanding lain sepertti skala *Thurstone*.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.
4. Jawaban suatu item dapat berupa *alternative*, sehingga informasi mengenai item tersebut diperjelas.
5. Reliabilitas pengukuran bisa diperoleh dengan jumlah item tersebut diperjelas.