

BAB III

ANALISIS & PERANCANGAN SISTEM

3.1 Analisis Masalah

Pariwisata merupakan kegiatan perjalanan dimana orang akan berdiam menuju suatu tempat asing untuk maksud tertentu, dan tidak tinggal menetap untuk selama-lamanya atau sementara waktu (Dr.Huber). Kemajuan dalam hal teknologi telah mengubah cara bepergian dan merencanakan kegiatan pariwisata, wisatawan tidak perlu lagi menggunakan *tool offline* seperti pensil dan kertas untuk mencatat lokasi obyek wisata tersebut. Informasi yang sering ditampilkan adalah informasi tempat-tempat wisata, cerita lokal mengenai tempat tersebut, info akomodasi dan sebagainya. Namun karena keterbatasan mengenai informasi tempat pariwisata dan jarak yang akan ditempuh atau info akomodasi membuat para wisatawan kurang berminat untuk mengunjungi lokasi obyek wisata tersebut. Misalnya, jika seorang wisatawan berada di daerah pusat kota tujuan, dan melihat sebuah obyek wisata seperti bangunan kuno atau museum, wisatawan pasti ingin tahu diskripsi bangunan tersebut terlebih dahulu untuk memastikan informasi obyek wisata yang sedang dikunjungi, wisatawan tidak bisa mendapatkan informasi tersebut tanpa bergabung dengan kelompok tur atau ponsel dengan akses web. Hal ini menyebabkan masyarakat kurang berminat terhadap obyek wisata yang menurut mereka tidak menarik dikunjungi karena kurangnya informasi.

Dari permasalahan diatas maka diperlukan suatu sistem yang mampu membantu aktifitas wisatawan dalam mencari informasi mengenai lokasi

wisatawan, obyek wisata dan rute tujuan lokasi wisata ke obyek wisata, sistem yang mudah dan mendukung mobilitas dapat dibangun sistem navigasi pariwisata pada *smartphone* Android, pada saat mengakses dimanapun berada secara *realtime* menggunakan perangkat *mobile smartphone android* bagi wisatawan, karena *smartphone android* merupakan telepon selular (ponsel) yang mampu selalu terhubung ke internet untuk *sinkronisasi* data secara otomatis atau manual dan dilengkapi dengan fitur GPS untuk sistem navigasi.

Untuk melakukan penentuan lokasi suatu perangkat *mobile* terdapat banyak cara, namun yang kerap digunakan adalah *Cell Identification* (Cell-ID) dan *Global Positioning System* (GPS). Pemanfaatan Cell-ID memiliki kelebihan yaitu tidak membutuhkan perangkat tambahan karena memanfaatkan fasilitas penyedia jaringan yang memungkinkan seorang pelanggan seluler untuk mengetahui posisinya terhadap *Base Transceiver Station* (BTS) terdekat. Namun akurasi dengan teknik Cell-ID ini sangat rendah yaitu berkisar 1-3 kilometer (km). GPS memiliki akurasi yang jauh lebih tinggi sehingga lebih cocok untuk diimplementasikan dalam sistem pelacakan meskipun teknik ini mengharuskan dibutuhkannya perangkat tambahan. Untuk mendapatkan hasil penentuan posisi yang maksimal dan sesuai kebutuhan sistem maka implementasi metode *mobile positioning* menggunakan GPS dikarenakan tingkat akurasinya yang jauh lebih tinggi dibandingkan dengan Cell-ID.

Oleh karena itu, Perangkat yang mendukung mobilitas untuk membangun sistem yang dapat membantu menunjukkan lokasi obyek wisata adalah dengan cara menggunakan handphone yang dilengkapi fitur GPS, karena perangkat komunikasi ini dapat dengan mudah dibawa kemana-mana, sehingga wisatawan

dapat dipandu dalam menentukan lokasi obyek wisata maupun komponen-komponen wisata. Komponen yang dimaksud adalah komponen pendukung wisata seperti kuliner, pompa bensin, mesin ATM, souvenir atau oleh-oleh dan tempat penginapan dimana komponen tersebut dapat menjamin akomodasi perjalanan wisatawan. Sistem ini diharapkan dapat mempermudah wisatawan mendapatkan informasi obyek wisata yang dilengkapi dengan sistem navigasi lokasi obyek wisata dan posisi keberadaan mereka pada peta digital secara *mobile*. Untuk mempermudah terciptanya sistem ini, maka akan memanfaatkan *Google*

Map API untuk menampilkan peta digital dan penunjuk rute tujuan.

3.1.1 Cara menampilkan rute terpendek dengan memanfaatkan google map API pada aplikasi

Pengetahuan yang diperlukan untuk mengembangkan Google Maps API adalah tentang HTML dan JavaScript, sedangkan peta sudah disediakan oleh Google, adapun langkah – langkahnya adalah sebagai berikut :

1. Agar peta dapat ditampilkan ke dalam aplikasi *mobile* , maka diharuskan mempunyai *account* Google, kemudian mendaftarkan diri dahulu di <http://code.google.com/apis/maps/signup.html>. Hal ini bertujuan untuk mendapatkan Google Maps API key. Berikut potongan *source code* untuk memasukkan API key.

```
<com.google.android.maps.MapView
android:id="@+id/mapView"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:enabled="true"
android:clickable="true"
android:apiKey="0ov2h9KP34mRIQV8ia5gKxHUck2noII9KpuL1Yg"/>
```

2. Selanjutnya apabila peta sudah tampil dalam aplikasi kita barulah

membuat *point marker*, *point marker* adalah penanda untuk koordinat asal ke koordinat tujuan. Berikut adalah potongan *source code* untuk membuat *point marker*.

```
List<Overlay> mapOverlays = mapView.getOverlays();
Drawable drawblethis.getResources().getDrawable(R.drawable.marker);
itemizedoverlay.addOverlay(overlayitem);
mapOverlays.add(itemizedoverlay);
```

sebelumnya icon *point marker* harus sudah tersimpan dalam editor program misalnya eclipse.

3. Langkah ketiga barulah kita menampilkan rute terpendek dengan memanfaatkan google map API, perhitungan jarak pada google maps merekomendasikan jalur terpendek dari semua jalur yang mungkin. Salah satu cara menentukan jarak rute terpendek adalah dengan mengetahui *geopoint* dari lokasi awal dan lokasi tujuan, dimana Google Maps akan menentukan *point marker* pada rute yang akan dilewati dan menghitung jarak antara *point marker* tersebut berdasarkan perhitungan jarak antara dua *geopoint*. Logikanya kita click *point marker* yang terdapat pada peta untuk mendapatkan koordinat asal atau koordinat terakhir dan koordinat tujuan, setelah itu baru membuat fungsi *getDirectiondata* untuk menampilkan rute terpendek dengan memanfaatkan google maps API perintah untuk *request direction* pada google maps adalah sebagai berikut:

```
String urlString =
http://maps.google.com/maps?f=d&hl=en&saddr=
+ srcPlace + "&daddr=" + destPlace
+ "&ie=UTF8&0&om=0&output=kml";
```

Didalam *source code* diatas terdapat fungsi "*srcPlace*" perintah tersebut digunakan untuk menampilkan koordinat asal dan untuk fungsi

“*destPlace*” adalah perintah untuk menampilkan koordinat tujuan. Sedangkan output yang dihasilkan dari fungsi tersebut adalah KML (*keyhole markup language*) yang kemudian disimpan dalam database server google map.

4. File KML berisi semua urutan koordinat rute yang dihasilkan oleh *point marker* yang kita pilih, file KML kemudian diproses atau dipecah oleh google map hingga menjadi sebuah informasi berupa rute jalur terpendek dan koordinat yang diambil dalam server google map untuk disimpan dalam aplikasi kita melalui SQLite untuk aplikasi *mobile* android dan MySQL untuk aplikasi websitenya. dengan menggunakan kombinasi dari gambar peta, database, serta obyek-obyek interaktif yang dibuat dengan bahasa pemrograman *HTML dan Javascript*, serta beberapa bahasa pemrograman lainnya. Gambar-gambar yang muncul pada peta merupakan hasil komunikasi dengan database pada *web server Google* untuk menampilkan gabungan dari potongan-potongan gambar yang diminta. Keseluruhan citra yang ada diintegrasikan ke dalam database pada *Google Server*, yang nantinya akan dapat dipanggil sesuai kebutuhan permintaan. Bagian- bagian gambar map merupakan gabungan dari potongan gambar-gambar bertipe *PNG* yang disebut *tile* yang berukuran 256 x 256 pixel seperti pada gambar 3.1.

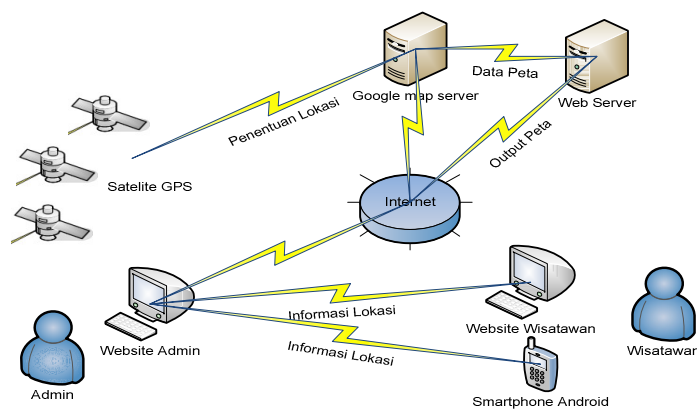


Gambar 3.1 Pembagian gambar peta sebesar 256 x 256 *pixel*

3.1.2 Arsitektur aplikasi

Sistem yang dibangun akan menampilkan informasi yang berkaitan dengan pariwisata, yang mana informasi tersebut didukung oleh visualisasi peta yang diperoleh dari google map. Seorang *User* Wisatawan membutuhkan koneksi internet pada *smartphone* Android ketika pertama kali akan memakai aplikasi, untuk keperluan *download* data dan untuk proses pencarian lokasi terdekat.

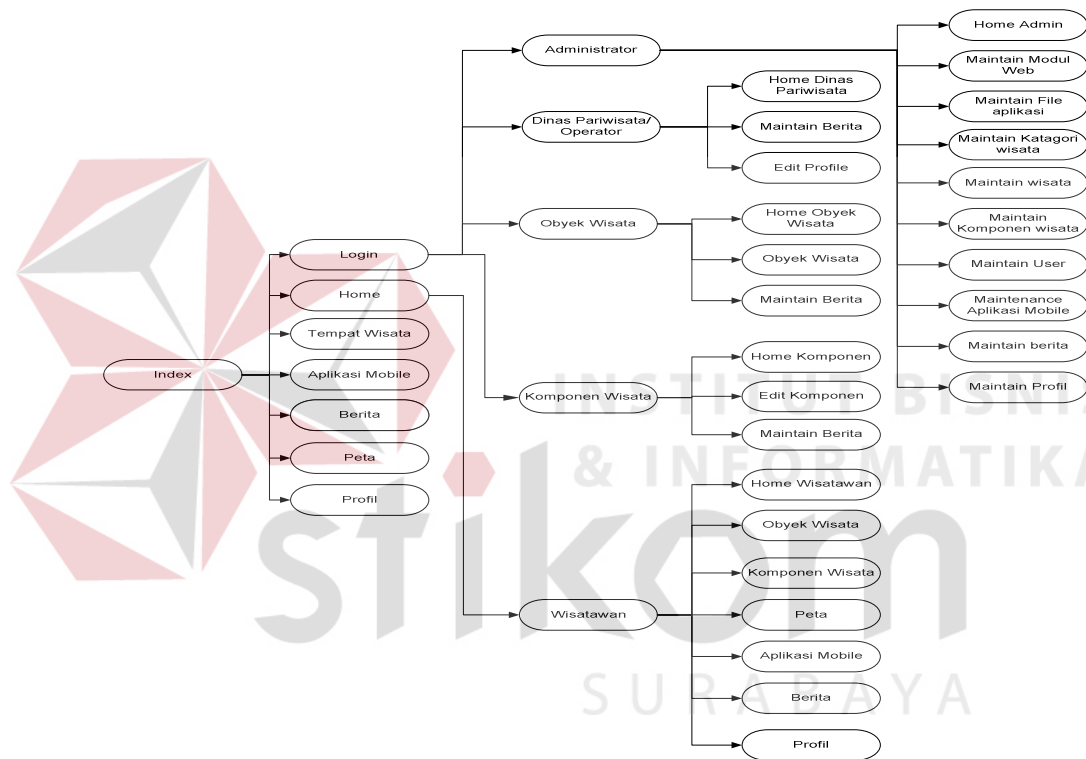
Sistem pencarian informasi pariwisata merupakan aplikasi pada *smartphone* Android yang membutuhkan aplikasi *Web Service*. Dimana data-data yang digunakan pada aplikasi *smartphone* Android ini merupakan data-data yang ada pada situs *web Service* pariwisata.



Gambar 3. 2 Arsitektur *Navigasi* Pariwisata Jawa Timur pada *Smartphone* Android.

3.1.3 Arsitektur aplikasi pada Website

Situs sistem pariwisata Jawa Timur digunakan oleh *User* wisatawan untuk melihat informasi seputar pariwisata di Jawa Timur. Tambahan sistem ini juga digunakan untuk melakukan *maintenance* data yang digunakan untuk aplikasi *smartphone* Android dalam mengolah data yang diperlukan seperti komponen wisata dan informasi lokasi wisata untuk keperluan visualisasi peta.

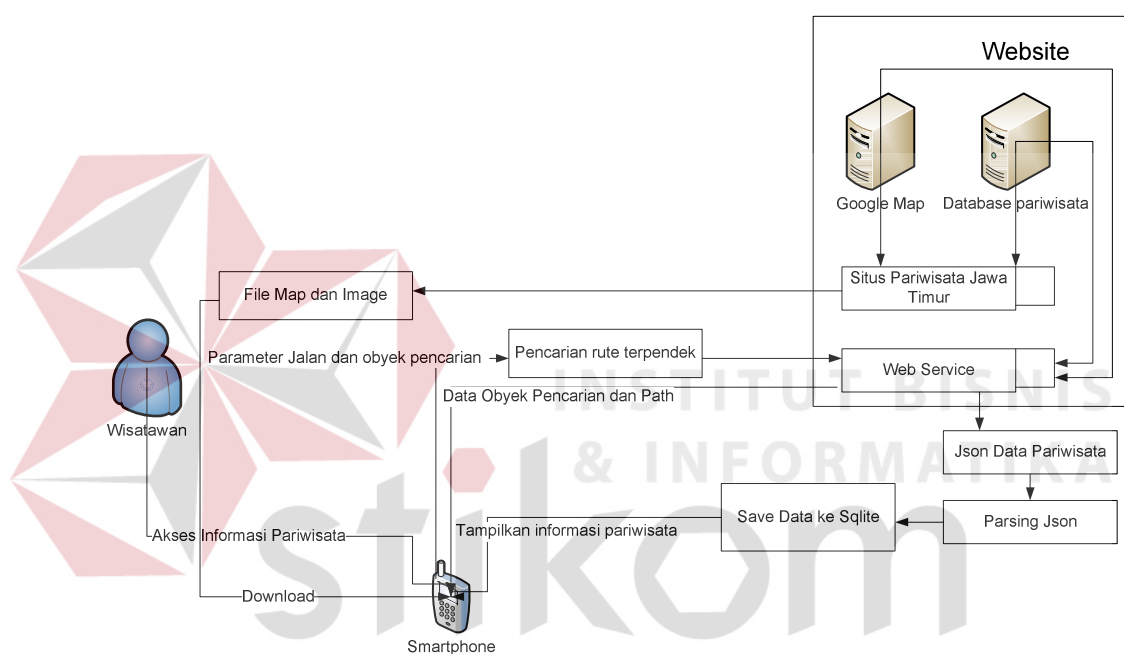


Gambar 3.3 Arsitektur aplikasi *website* pariwisata Jawa Timur

Pada gambar 3.3, Untuk dapat melakukan *maintain* data, seorang *User* admin harus *login* terlebih dahulu ke aplikasi web dengan menggunakan *User* dan password yang dipunyai. Seorang *User* admin dapat melakukan *maintain* data-data yang terkait untuk proses pengolahan informasi lokasi wisata pada *smartphone* Android.

3.1.4 Arsitektur aplikasi Pada *Smartphone* Android

Dalam pembuatan aplikasi *Smartphone* Android dibutuhkan koneksi yang menghubungkan *smartphone* dengan web atau lebih tepatnya adalah mengakses halaman web. Dalam hal ini, koneksi yang digunakan adalah *HTTP Connection*. Untuk penggunaan *HTTP Connection*, layanan *GPRS/3G* pada *smartphone* harus sudah aktif.



Gambar 3.4 Garis besar aplikasi pada *Smartphone* Android

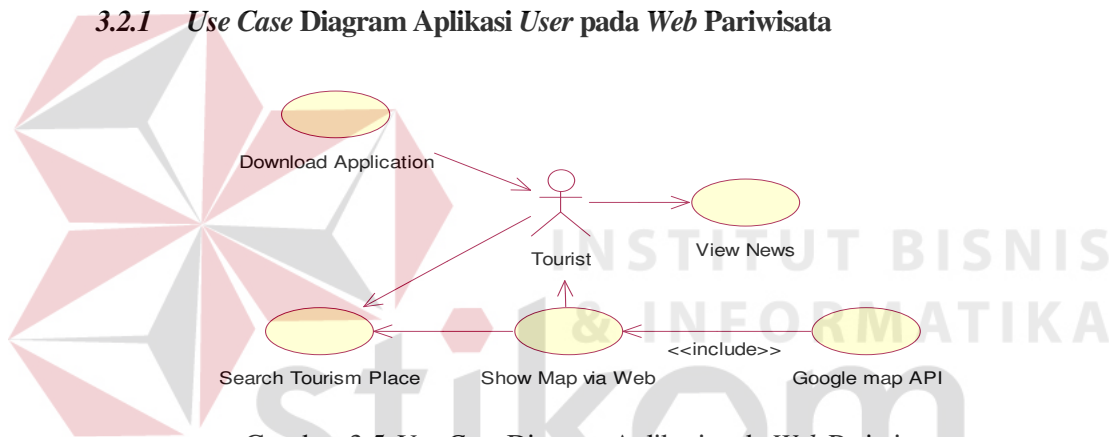
Untuk dapat menggunakan aplikasi Sistem Navigasi Pariwisata di Jawa Timur pada *Smartphone* Android, langkah pertama yang harus dilakukan oleh *User* wisatawan adalah melakukan *download* aplikasi Sistem Navigasi yang telah disediakan di situs *web* pariwisata. Setelah aplikasi diinstall pada *smartphone* Android kemudian jalankan, *User* wisatawan membutuhkan koneksi internet agar aplikasi ini dapat mengakses *Web Service* untuk mengambil data-data pariwisata. Setelah data-data pariwisata sukses *download*, baru seorang *User* wisatawan

dapat menggunakan aplikasi sistem navigasi bersifat *offline*. *User* wisatawan membutuhkan koneksi bersifat *online* ketika *User* wisatawan menggunakan menu pencarian lokasi obyek wisata, posisi *User* wisatawan dan rute tujuan.

3.2 Use Case Diagram

Use case diagram digunakan untuk menspesifikan apa yang dapat dilakukan oleh sistem atau untuk menspesifikan kebutuhan fungsional utama dari sistem. Berikut akan dijelaskan *use case diagram* untuk masing-masing sistem.

3.2.1 Use Case Diagram Aplikasi User pada Web Pariwisata



Gambar 3.5 Use Case Diagram Aplikasi pada Web Pariwisata

Use Case ini dimulai dari pertama kali wisatawan membuka aplikasi *website* yang telah disediakan tanpa harus melakukan proses *login* terlebih dahulu, kemudian aktor wisatawan bisa melakukan *download* aplikasi *mobile* pada *website* khusus untuk aktor wisatawan. wisatawan juga bisa mengakses peta yang ada pada *website* untuk mencari lokasi obyek wisata tanpa harus *download* aplikasi *mobile* terlebih dahulu yaitu dengan mengakses peta yang ada pada *website* yang dipadukan dengan *google map API* untuk arah rute tujuan dan penentuan lokasi. Sehingga informasi yang ditampilkan tidak hanya melalui

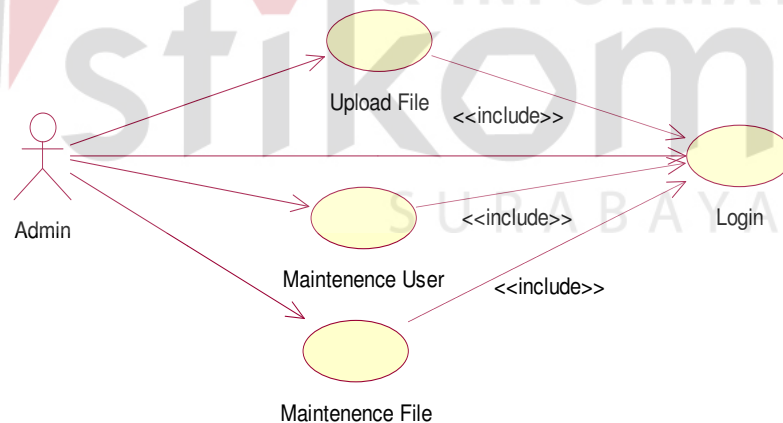
aplikasi *Mobile* tetapi juga bisa ditampilkan melalui *website* dengan koneksi internet.

Berikut adalah Keterangan singkat dari masing-masing *use case* yang dimiliki oleh aplikasi *web site* untuk *user* wisatawan:

Tabel 3.1 Keterangan singkat *use case diagram* aplikasi *web site* untuk *user*

Nama Use Case	Deskripsi
<i>Download Application</i>	Proses yang digunakan untuk melakukan unduh aplikasi dari <i>web site</i>
<i>Search Tourism Place</i>	Proses yang digunakan untuk melakukan pencarian lokasi obyek pariwisata
<i>Show Map Via Web</i>	Proses yang digunakan untuk menampilkan peta melalui <i>website</i>
<i>Google Map AP</i>	Proses yang menangani rute terpendek dan maintenance pada map
<i>View News</i>	Proses yang menangani penampilan berita yang akan di upload pada web pariwisata.

3.2.2 Use Case Diagram Aplikasi Admin pada Web Service Pariwisata



Gambar 3.6 Use Case Diagram Aplikasi pada Web Service Pariwisata.

Use Case ini dimulai dari pertama kali membuka aplikasi *web service*, untuk mengakses semua fasilitas yang ada pada *web service* aktor admin harus melakukan proses *login*. Fasilitas yang dimaksud adalah fasilitas *Upload File*,

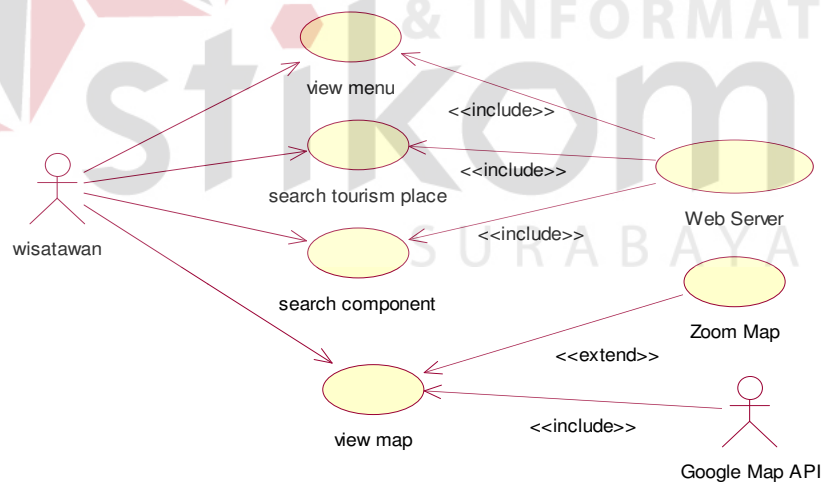
fasilitas *Maintenance User*, fasilitas *Maintenance User* yang merupakan proses turunan (*include*) dari proses login.

Berikut adalah keterangan singkat dari masing-masing *use case* yang dimiliki oleh aplikasi *web site* untuk admin:

Tabel 3.2 keterangan singkat *use case diagram* aplikasi *web site* untuk admin

Nama Use Case	Deskripsi
<i>Login</i>	Proses yang menangani <i>login</i> untuk admin
<i>Maintenance User</i>	Proses yang menangani pemeliharaan data akun pengguna
<i>Upload File</i>	Proses yang menangani <i>upload file</i> ke server
<i>Maintenance File</i>	Proses yang menangani pemeliharaan <i>file</i> aplikasi

3.2.3 Use Case Diagram Aplikasi pada Smartphone Android



Gambar 3.7 Use Case Diagram Aplikasi pada Smartphone Android

Use Case ini dimulai dari pertama kali membuka aplikasi *mobile* yang sebelumnya sudah di *download* pada *website* pariwisata, aktor wisatawan di dalam *use case* ini dapat mengakses informasi berupa *menu* yang ada pada

tampilan *form* utama aplikasi *mobile* dan aktor wisatawan juga dapat mengakses pencarian rute lokasi obyek wisata dan obyek komponen wisata yang semua pengolahan datanya merupakan proses turunan dari (*include*) *web server* pariwisata yang dipadukan dengan *server google map API* untuk penentuan arah rute tujuan dan penentuan lokasi. Dalam *use case* ini aktor wisatawan tidak perlu terus menerus membutuhkan koneksi internet, koneksi internet hanya dibutuhkan pada saat penentuan arah rute tujuan dan penentuan lokasi aktor wisatawan

Berikut adalah keterangan singkat dari masing-masing *use case* yang dimiliki oleh aplikasi *Sistem Navigasi Pariwisata pada Smartphone Android*:

Tabel 3.3 keterangan singkat *use case diagram* aplikasi *Smartphone Android*

Nama Use Case	Deskripsi
<i>View Menu</i>	Proses yang menangani <i>Menu</i> pada aplikasi untuk mengakses Lokasi, Setting, Berita, Petunjuk aplikasi, tandai lokasi, list tanda.
<i>Search Tourism Place</i>	Proses yang menangani pencarian obyek wisata beserta informasinya.
<i>Search Component</i>	Proses yang menangani pencarian komponen wisata beserta informasinya seperti pompa bensin, hotel, ATM dan sebagainya.
<i>View Map</i>	Proses yang digunakan untuk melihat peta yang diambil dari <i>Google Map AP</i> untuk menentukan posisi dan rute tujuan ke obyek wisata.

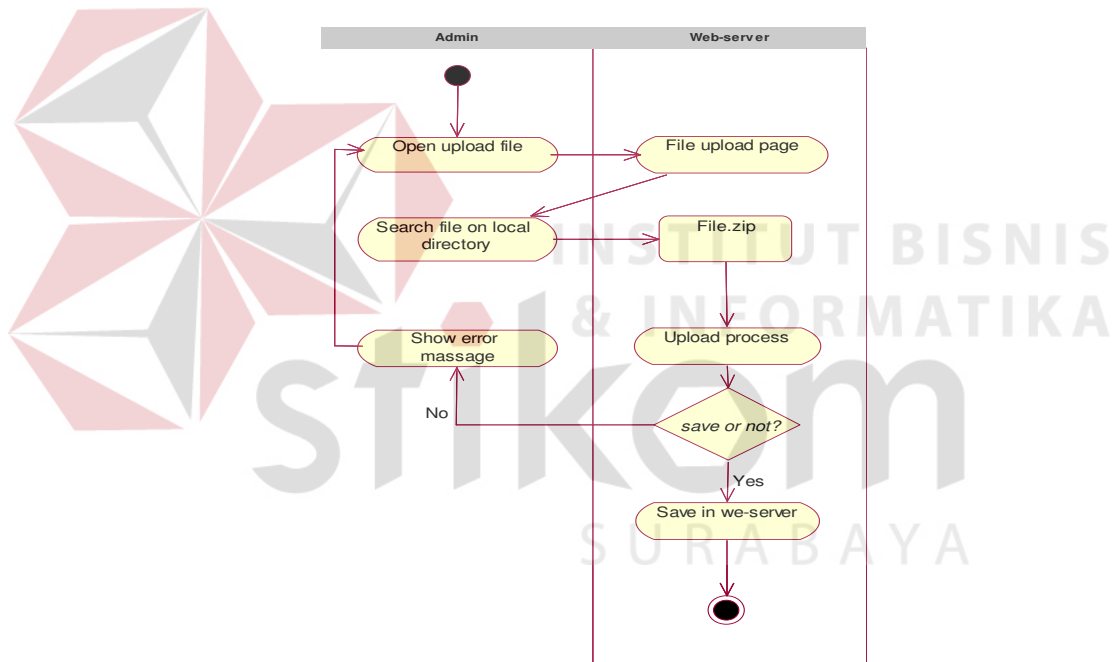
3.3 Activity Diagram

Dari *Use Case* yang ada, dibutuhkan *Activity Diagram* untuk menjelaskan proses/aliran yang terjadi pada tiap *Use Case*. *Activity Diagram* adalah salah satu bentuk diagram UML yang paling mudah dimengerti dikarenakan diagram ini memiliki simbol yang menyerupai simbol *flowchart*, yang sangat berguna untuk menerangkan langkah-langkah proses ke pihak lain.

3.3.1 Activity diagram Web Service Pariwisata

A. Activity Diagram untuk "Upload File"

Proses dimulai dari *Useradmin* membuka *web site* dan menu *upload*. Setelah tampil halaman *upload* admin harus mencari *file* yang ingin di *upload* menggunakan tombol *browse*, Pada saat proses *upload* berhasil maka *file* disimpan dalam *web server*, namun jika gagal maka akan muncul pesan error dan mengulangi lagi proses *upload*. Untuk lebih jelasnya dapat dilihat pada gambar 3.8.

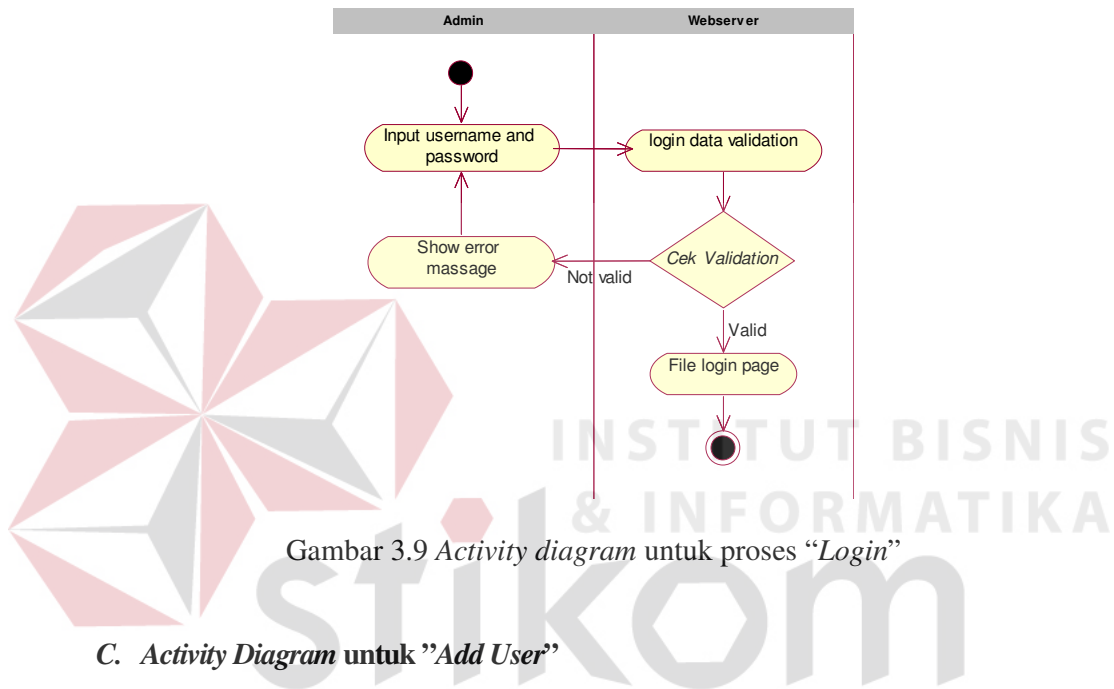


Gambar 3.8 Activity diagram untuk proses "Upload File"

B. Activity Diagram untuk "Login"

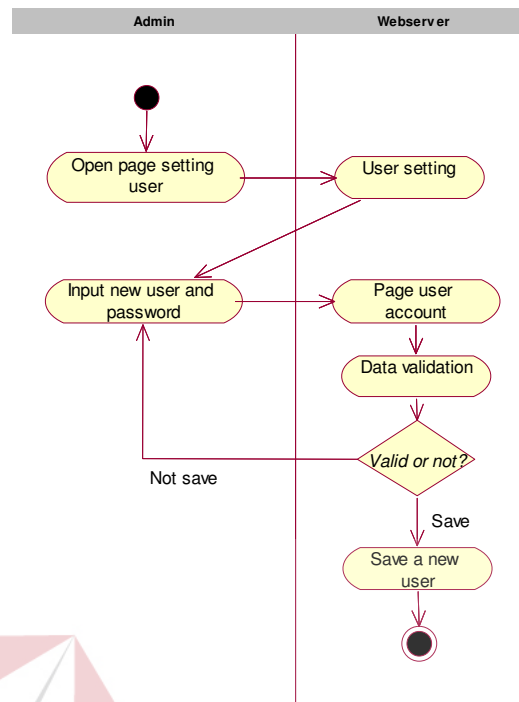
Proses dimulai ketika *UserAdmin* memasukkan *username* dan *password*. Setelah semua data terisi, data kemudian dikirim ke *web server* via internet untuk dilakukan proses validasi *login*. Proses validasi *login* mencocokkan data yang dimasukkan *UserAdmin* tersebut dengan tabel *member* yang ada di *database*, jika

proses validasi salah maka ditampilkan pesan error dan *UserAdmin* memasukkan kembali *username* dan *password* yang benar, namun jika proses validasi berhasil maka data *login* tersebut akan disimpan dalam *database* lokal untuk proses merekam siapa saja yang sudah *login* pada web. Untuk lebih jelas bisa dilihat pada gambar 3.9.



C. Activity Diagram untuk "Add User"

Proses dimulai dari *UserAdmin* membuka *web site* dan *menu setting*, setelah muncul halaman setting *User* dapat mengubah *password* yang digunakan yaitu dengan cara memasukkan *password* lama dan *password* baru. Setelah itu di cek apakah *password* lama sesuai dengan *database*, jika benar maka *password* baru akan disimpan, namun jika tidak muncul pesan error dan mengulangi memasukkan *password* lama dan baru. Untuk lebih jelasnya dapat dilihat pada gambar 3.10.



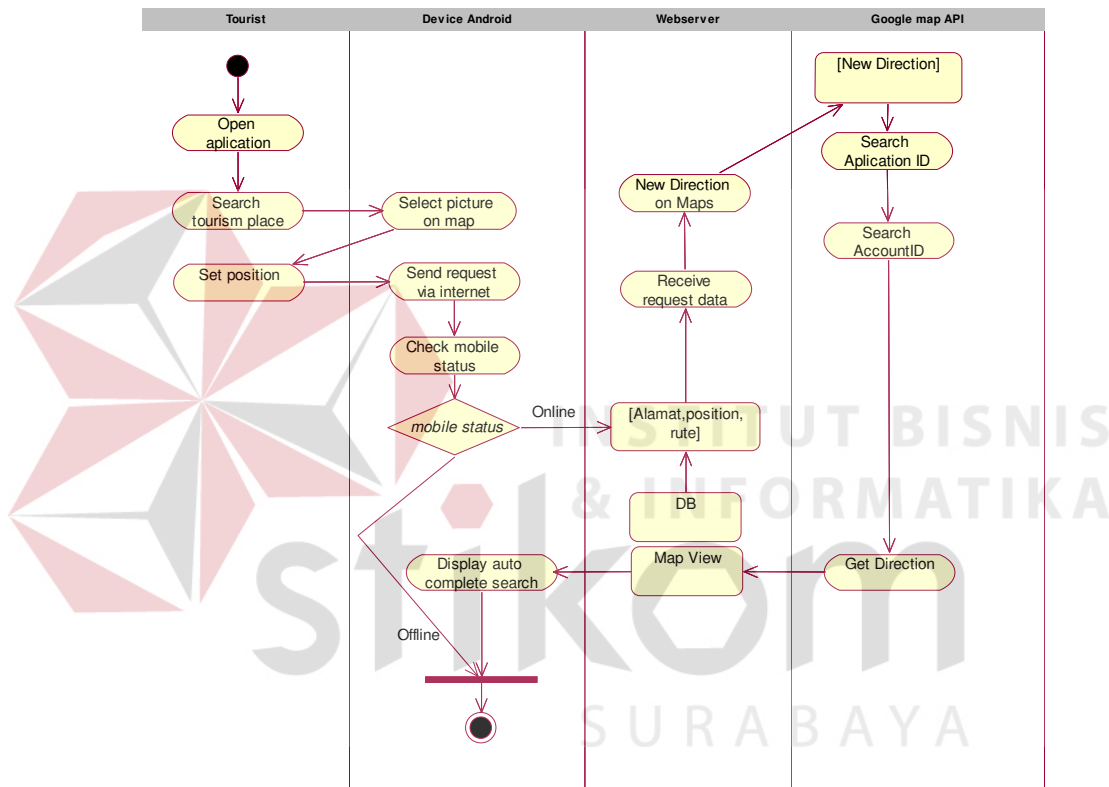
Gambar 3.10 Activity diagram untuk proses “Add User”

3.3.2 Activity diagram Smartphone Android

A. Activity Diagram untuk Search Tourism Place

Proses ini dimulai dari *Mobile Application* yang telah menerima permintaan *Userwisatawan* mengenai obyek wisata yang akan dikunjungi dari aplikasi Sistem Navigasi Pariwisata yang ada pada *smartphone android* dalam hal ini *device* harus bersifat *online*, dan selanjutnya aplikasi navigasi memberitahukan pada *server Google Map AP* bahwa ada *request* baru yang harus diberitahukan pada ponsel *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan *request* baru pada *Userwisatawan*, informasi baru tersebut merupakan keseluruhan informasi tentang peta, rute tujuan, dan posisi wisatawan. Data tentang obyek wisata diambil dari *web server* Pariwisata menggunakan media internet, *smartphone android request* data tentang obyek

wisata melalui media internet ke dalam *web server*, kemudian *web server* mengirim informasi mengenai obyek wisata ke dalam aplikasi *mobile*. *Web server* segera memproses permintaan dan menghasilkan informasi detail obyek wisata dan mengirimkannya kembali pada *smartphone*. Untuk lebih jelasnya dapat dilihat pada gambar 3.11.

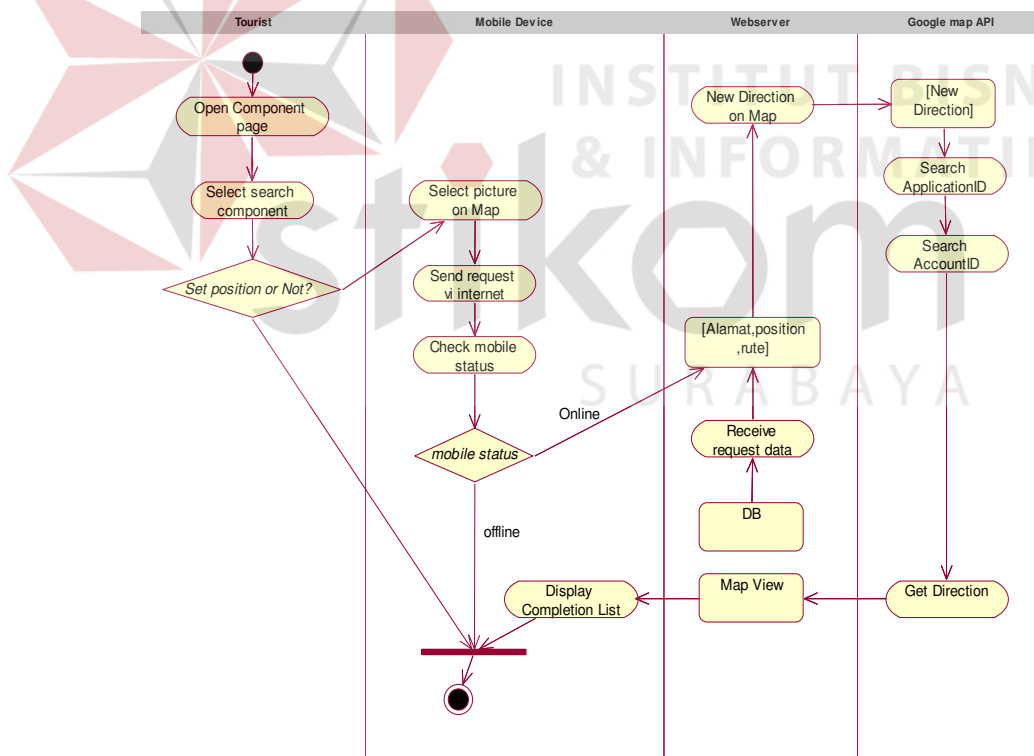


Gambar 3.11 Activity diagram untuk proses “Search Tourism Place”

B. Activity Diagram untuk Use Case Search Component

Proses ini dimulai dari *Mobile Application* yang telah menerima permintaan *Userwisatawan* mengenai komponen obyek wisata yang akan dikunjungi dari aplikasi Sistem Navigasi Pariwisata yang ada pada *smartphone android* dalam hal ini *device* harus bersifat *online*, dan selanjutnya aplikasi navigasi memberitahukan pada *server Google Map AP* bahwa ada *request* baru yang harus diberitahukan

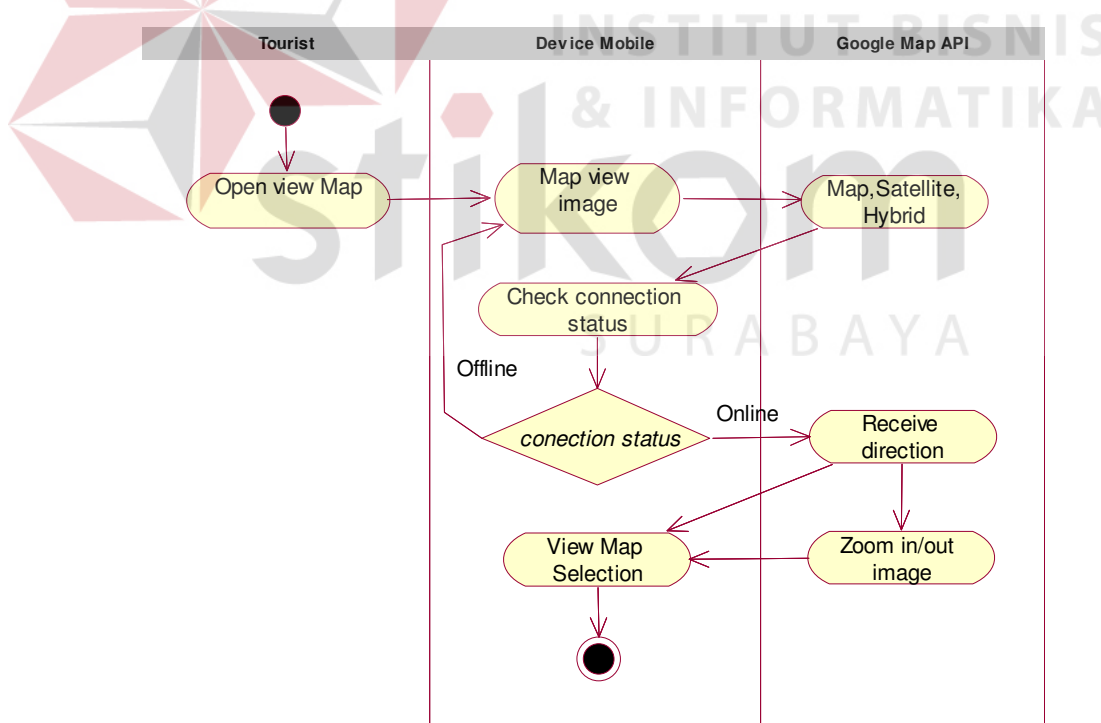
pada ponsel *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan *request* baru pada *Userwisatawan*, informasi baru tersebut merupakan keseluruhan informasi tentang peta, rute tujuan, dan posisi wisatawan. Data tentang komponen obyek wisata diambil dari *web server* Pariwisata menggunakan media internet, *smartphone android request* data tentang obyek wisata melalui media internet ke dalam *web server*, kemudian *web server* mengirim informasi mengenai obyek wisata ke dalam aplikasi *mobile*. *Web server* segera memproses permintaan dan menghasilkan informasi detail komponen obyek wisata dan mengirimkannya kembali pada *smartphone*. Untuk lebih jelasnya dapat dilihat pada gambar 3.12.



Gambar 3.12 Activity diagram untuk proses “Search Component”

C. Activity Diagram untuk proses View Map

Proses ini dimulai dari *Mobile Application* yang telah menerima permintaan *User* wisatawan mengenai obyek wisata yang akan dikunjungi dari aplikasi Sistem Navigasi Pariwisata yang ada pada *smartphone android* dalam hal ini *device* harus bersifat *online*, dan selanjutnya aplikasi navigasi memberitahukan pada *server Google Map AP* bahwa ada *request* baru yang harus diberitahukan pada ponsel *smartphone android*, *smartphone android* yang menerima *request* tersebut akan menampilkan *request* baru pada *User* wisatawan, informasi baru tersebut merupakan keseluruhan informasi tentang pilihan gambar peta yang akan ditampilkan, *zoom in/out*, dan posisi wisatawan. Untuk lebih jelasnya dapat dilihat pada gambar 3.13.



Gambar 3.13 Activity diagram untuk proses “View Map”

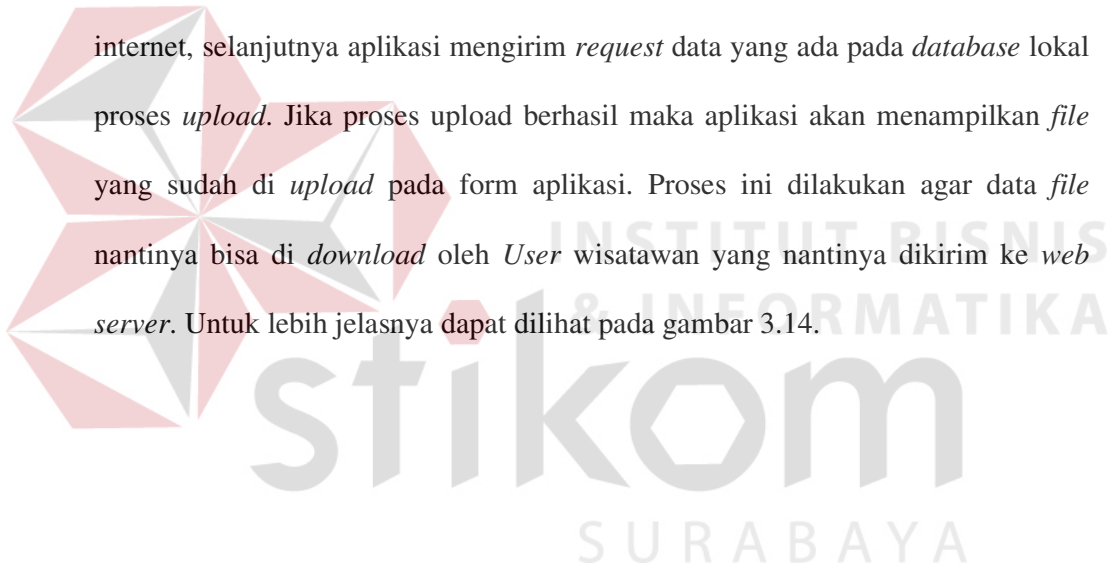
3.4 Sequence Diagram

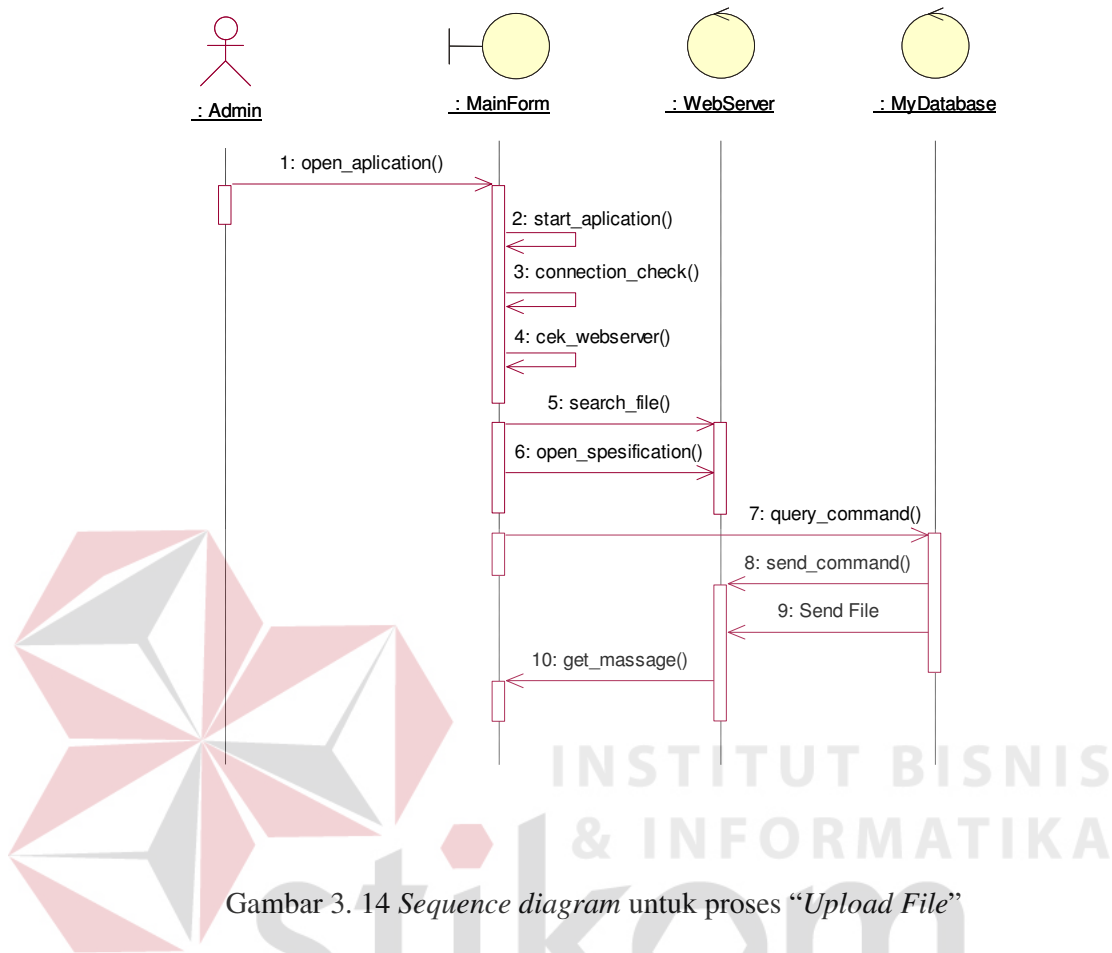
Sequence diagram digunakan untuk menggambarkan interaksi antar objek berdasarkan urutan waktu yang digambarkan dari atas kebawah.

3.4.1 Web Server

A. Upload File

Proses dimulai ketika aplikasi dijalankan, proses *upload* dilakukan dengan mengambil data yang ada pada *database* lokal, aplikasi menghubungi *web server* terlebih dahulu untuk mengetahui apakah ada koneksi yang terhubung melalui internet, selanjutnya aplikasi mengirim *request* data yang ada pada *database* lokal proses *upload*. Jika proses upload berhasil maka aplikasi akan menampilkan *file* yang sudah di *upload* pada form aplikasi. Proses ini dilakukan agar data *file* nantinya bisa di *download* oleh *User* wisatawan yang nantinya dikirim ke *web server*. Untuk lebih jelasnya dapat dilihat pada gambar 3.14.

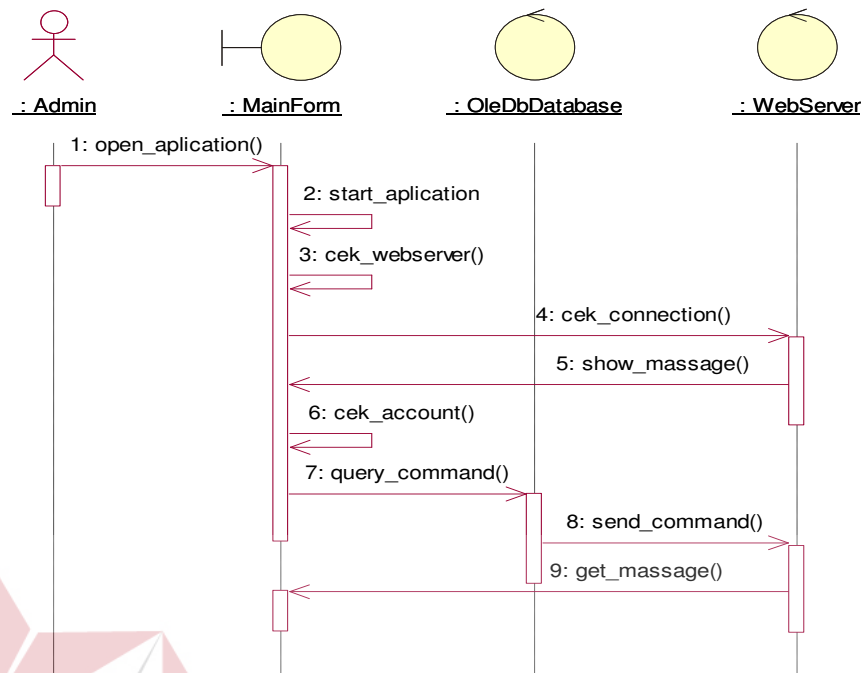




Gambar 3. 14 Sequence diagram untuk proses “Upload File”

B. Login

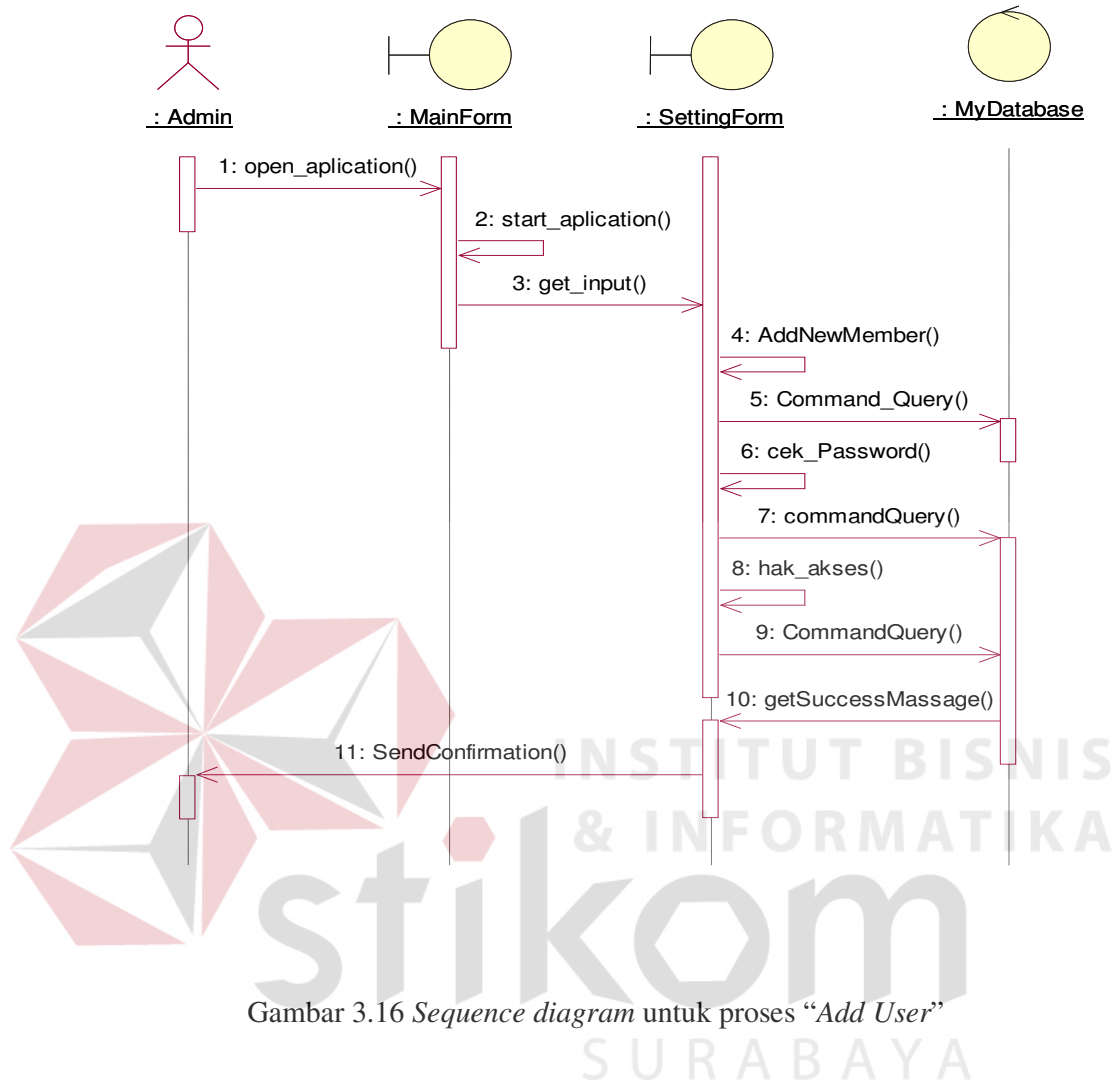
Proses dimulai ketika aplikasi dijalankan, proses *login* dilakukan dengan menginputkan data *Useraccount* dan *password* yang ada pada *database* lokal, aplikasi menghubungi *web server* terlebih dahulu untuk mengetahui apakah ada koneksi yang terhubung melalui internet, selanjutnya aplikasi mengirim *username* dan *password* yang ada pada *database* lokal proses autentikasi. Jika proses autentikasi berhasil maka aplikasi sudah terhubung ke *web server* dan *login* sebagai *User* tertentu. Proses ini dilakukan agar data yang nantinya dikirim ke *web server* semuanya ditampilkan untuk *User* tertentu dengan hak akses masing-masing. Untuk lebih jelasnya dapat dilihat pada gambar 3.15.



Gambar 3.15 *Sequence diagram* untuk proses “Login”

C. Add User

Proses dimulai dari admin membuka *web site* dan masuk pada halaman setting, *User* admin dapat mengganti hak akses yang dimiliki oleh masing-masing *user*, admin juga dapat mengubah password dengan cara memasukkan *password* lama dan *password* baru, dan, jika *password* lama valid maka *password* baru akan disimpan. Untuk lebih jelasnya dapat dilihat pada gambar 3.16.

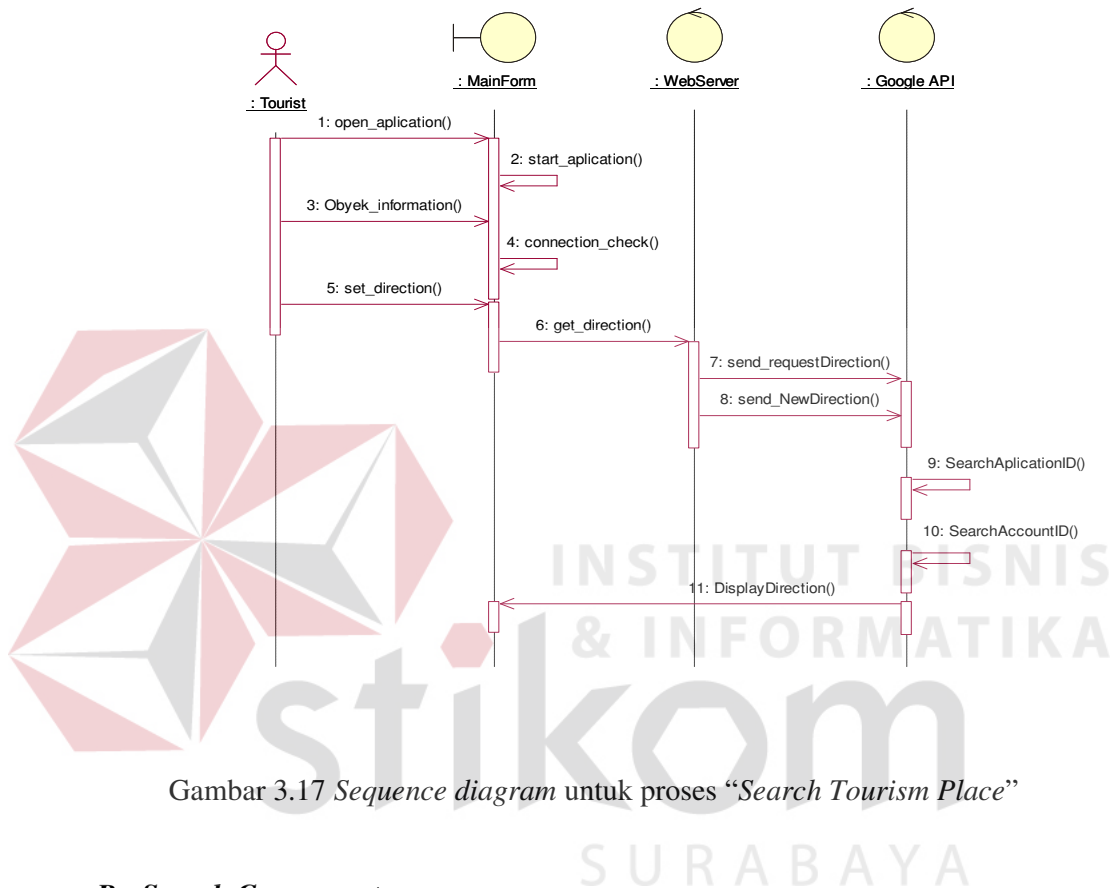


3.4.2 Mobile

A. Search Tourist Place

Proses dimulai dari *web server* yang mendapatkan informasi *request* terbaru dari aplikasi navigasi pariwisata. Selanjutnya *web server* menyuruh *server Google Map AP* untuk menghubungi ponsel *android*, proses yang terjadi pada *server Google Map AP* secara umum adalah mencari *Application ID* untuk mengetahui aplikasi apa yang membutuhkan layanan *Google Map AP* dan *Account ID* untuk mengetahui kepada siapa *server Google Map AP* menghubungi

ponsel *Android*. Selanjutnya informasi dikirimkan via *internet* melalui *smartphone android* dan ditampilkan pada ponsel *Android*. Untuk lebih jelasnya dapat dilihat pada gambar 3.17.

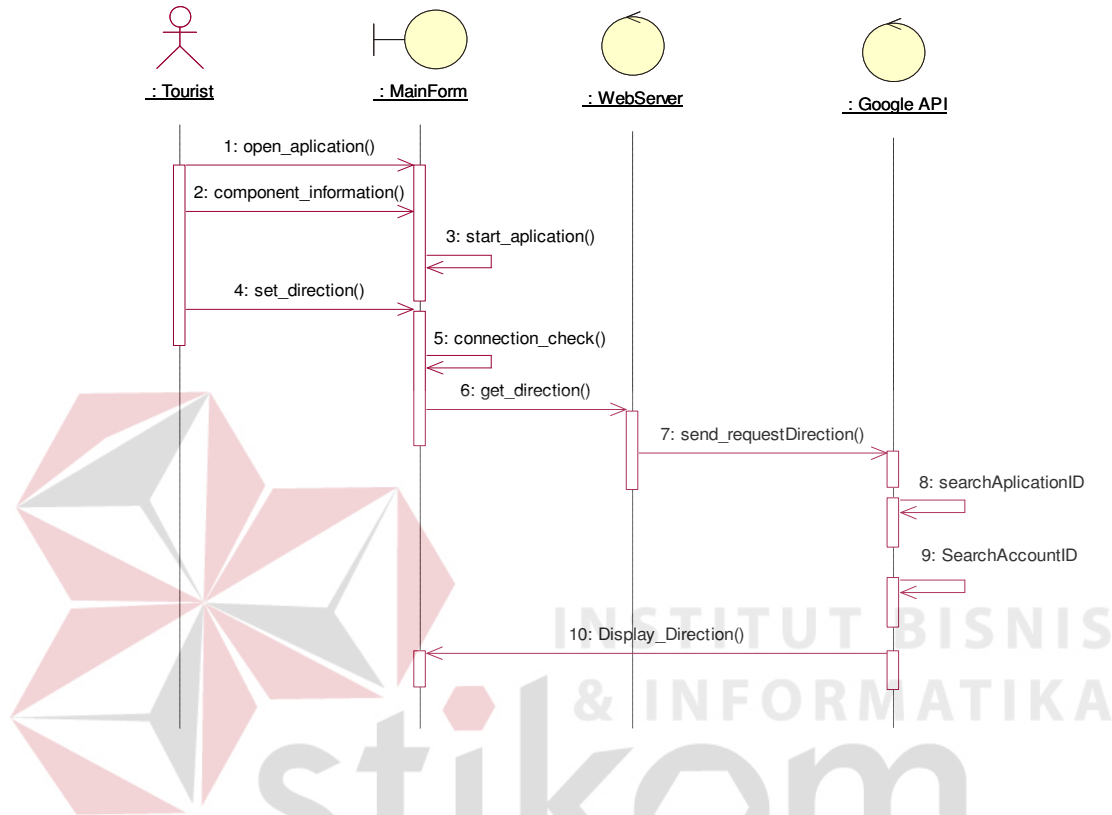


Gambar 3.17 Sequence diagram untuk proses “Search Tourism Place”

B. Search Component

Proses dimulai dari *web server* yang mendapatkan informasi *request* terbaru dari aplikasi navigasi pariwisata. Selanjutnya *web server* menyuruh *server Google Map AP* untuk menghubungi ponsel *android*, proses yang terjadi pada *server Google Map AP* secara umum adalah mencari *Application ID* untuk mengetahui aplikasi apa yang membutuhkan layanan *Google Map AP* dan *Account ID* untuk mengetahui kepada siapa *server Google Map AP* menghubungi ponsel *Android*. Selanjutnya informasi dikirimkan via *internet* melalui

smartphone android dan ditampilkan pada ponsel *Android*. Untuk lebih jelasnya dapat dilihat pada gambar 3.18.

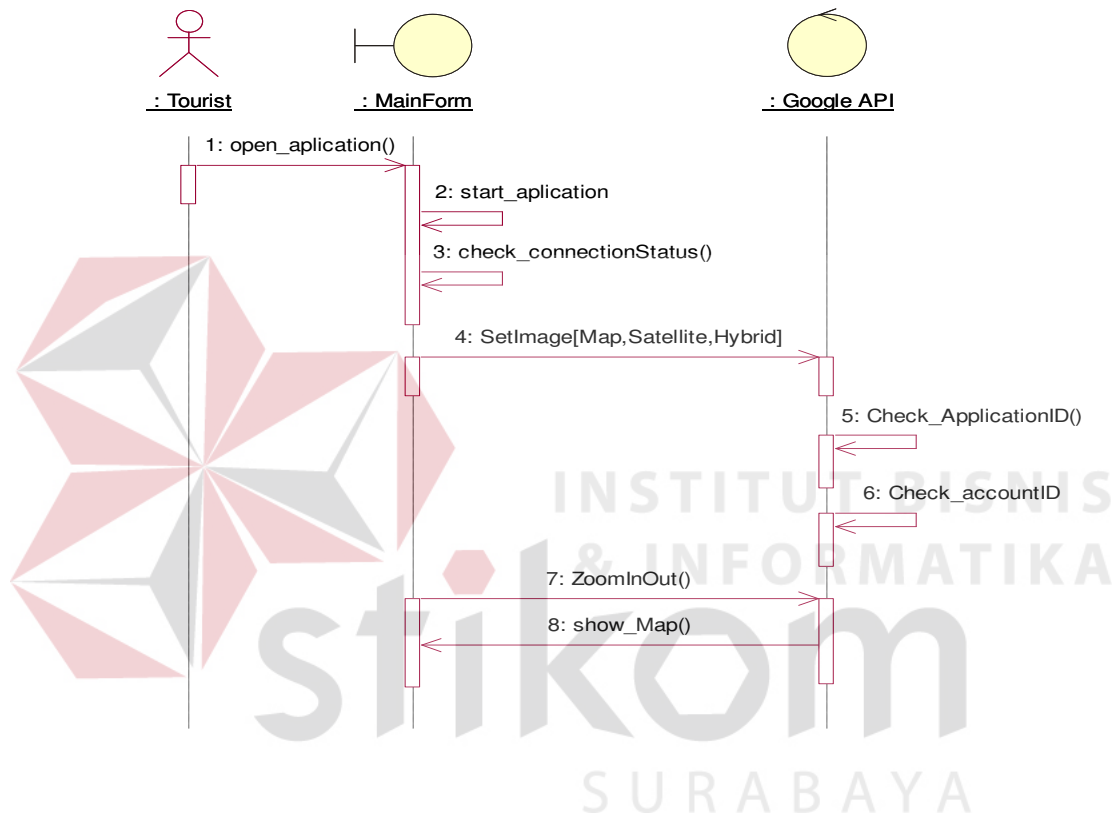


Gambar 3.18 *Sequence diagram* untuk proses “*Search Component*”

C. *View Map*

Proses dimulai dari *User* wisatawan yang membuka *form* pencarian, *User* wisatawan memasukkan *request* yang ingin dicari, pada saat aplikasi dijalankan aplikasi akan mencari data pada *server Google Map AP* menggunakan internet, jika data ditemukan selanjutnya *server Google Map AP* mengirimkan kembali informasi tersebut ke aplikasi menggunakan internet. proses yang terjadi pada *server Google Map AP* secara umum adalah mencari *Application ID* untuk mengetahui aplikasi apa yang membutuhkan layanan *Google Map AP* dan

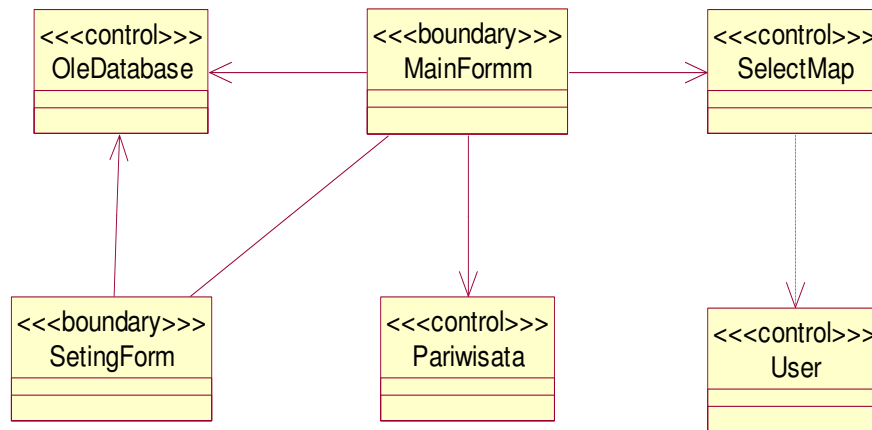
Account ID untuk mengetahui kepada siapa *server Google Map AP* menghubungi ponsel *Android*. Aplikasi menampilkan data peta yang dapat dilihat dengan memilih menggunakan pencitraan *Map*, *Satellite*, atau *Hybrid*. Untuk lebih jelasnya dapat dilihat pada gambar 3.19.



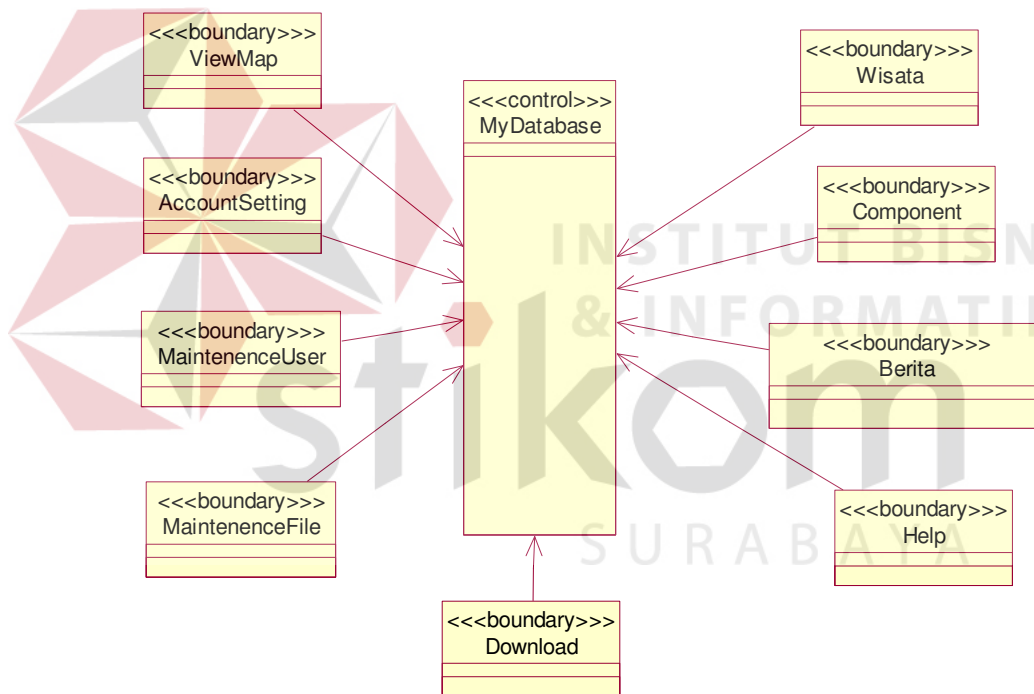
Gambar 3.19 *Sequence diagram* untuk proses “View Map”

3.5 Class Diagram

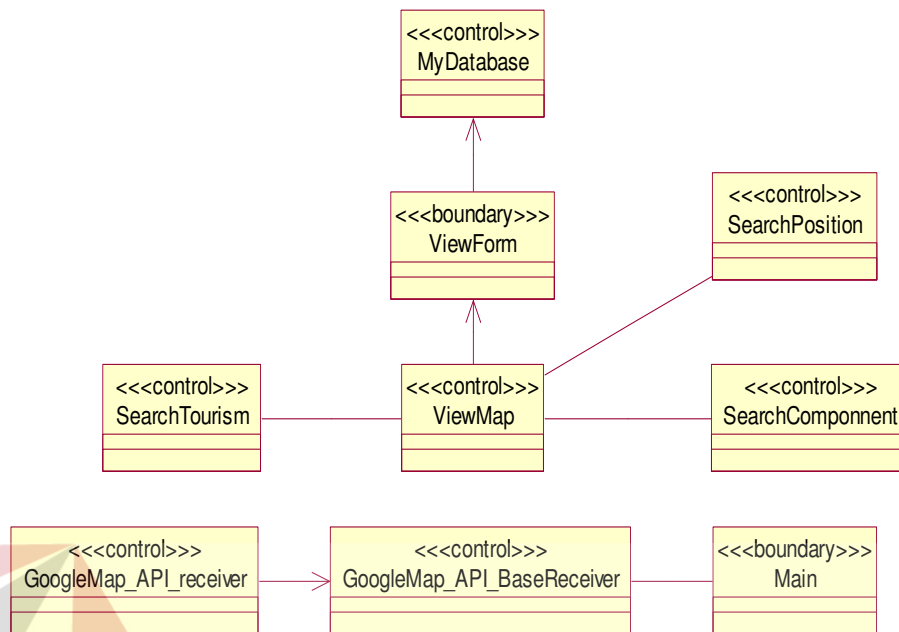
Class Diagram digunakan untuk menampilkan kelas-kelas atau paket-paket didalam sistem dan relasi antar kelas tersebut (menunjukkan interaksi antar kelas di dalam aplikasi).



Gambar 3. 20 Class diagram aplikasi Pariwisata



Gambar 3. 21 Class diagram aplikasi *website*

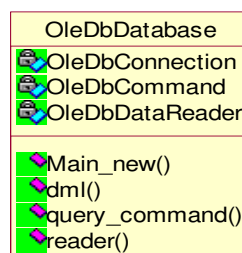


Gambar 3. 22 Class diagram aplikasi *mobile*

Gambar 3.20, gambar 3.21, dan gambar 3.22 menampilkan gambar kelas-kelas yang digunakan oleh aplikasi pariwisata, *website*, dan aplikasi *mobile*, berikut adalah penjelasan singkat dari kelas-kelas yang digambarkan:

3.5.1 kelas “*OleDbDatabase*”

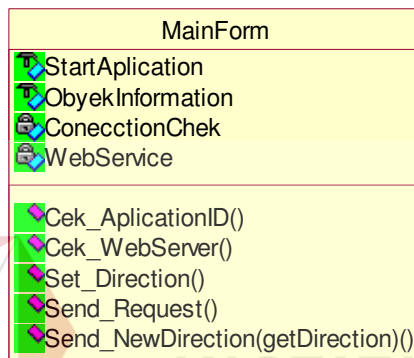
Kelas “*OleDbDatabase* ” digunakan untuk mengontrol koneksi aplikasi dengan *database* lokal. Kelas ini berisi atribut-atribut yang digunakan untuk melakukan koneksi dan operasi-operasi yang digunakan untuk melakukan *query* dan *read* database. Untuk lebih jelasnya dapat dilihat pada gambar 3.23.



Gambar 3. 23 Kelas “*OleDbDatabase*” aplikasi Pariwisata

3.5.2 kelas “MainForm”

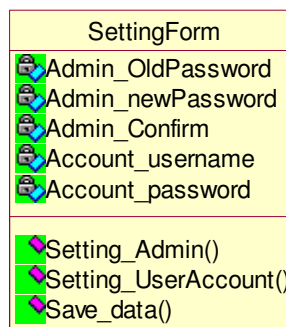
Kelas “MainForm” digunakan sebagai *form* utama aplikasi, di dalam kelas ini terdapat atribut dan operasi untuk memanggil kelas lain yang berhubungan dengan *Map*, *Search Tourism Place*, dan *Search Component Place*. Dengan kata lain kelas ini digunakan sebagai penghubung dengan kelas-kelas yang lain. Untuk lebih jelasnya dapat dilihat pada gambar 3.24.



Gambar 3. 24 “MainForm” aplikasi Pariwisata

3.5.3 kelas “Setting Form”

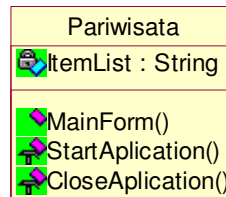
Kelas “SettingForm” digunakan untuk mengatur tampilan modul yang ditampilkan pada web service. Didalam kelas ini Admin dapat mengatur hak *privelledge* yang dimiliki oleh masing-masing *User* untuk mengakses modul yang sudah disediakan. Untuk lebih jelasnya dapat dilihat pada gambar 3.25.



Gambar 3. 25 Kelas “Setting Form” aplikasi Pariwisata

3.5.4 kelas “Pariwisata”

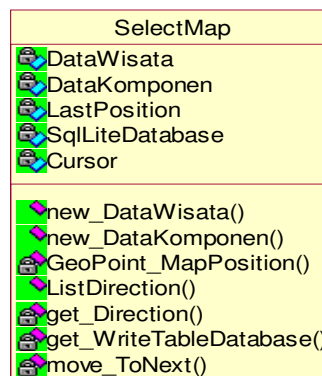
Pada kelas “Pariwisata” digunakan untuk menampilkan semua informasi mengenai obyek wisata. Data obyek wisata yang disimpan nantinya akan ditampilkan kedalam peta. Untuk lebih jelasnya dapat dilihat pada gambar 3.26.



Gambar 3. 26 kelas “Pariwisata” aplikasi Pariwisata

3.5.5 kelas “Select Map”

Pada kelas “*Select Map*” digunakan untuk memilih peta dengan cara menginputkan posisi terakhir dilanjutkan dengan memilih obyek wisata atau komponen sehingga kita dapat mendapatkan rute yang kita inginkan. Didalam kelas ini terdapat operasi *GeopointGP()* untuk merekam posisi terakhir *User* dan *getDirection()* untuk menampilkan rute tujuan. Selain itu terdapat operasi *setOnLongClickListener()* digunakan untuk memilih obyek wisata atau cari komponen. Untuk lebih jelasnya dapat dilihat pada gambar 3.27.



Gambar 3. 27 kelas “Select Map” aplikasi Pariwisata

3.5.6 kelas “User”

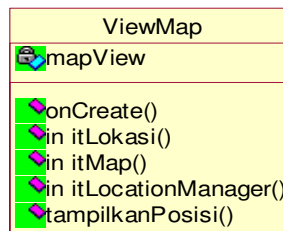
Pada kelas “User” digunakan untuk mengakses *file* berdasarkan hak privilege yang dimiliki oleh masing-masing *User* dengan cara menginputkan *username* dan *password* terlebih dahulu. Dalam kelas ini terdapat operasi *levelUser()* yang digunakan untuk mengakses modul sesuai dengan *level User* tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 3.28.



Gambar 3. 28 kelas “User” aplikasi Pariwisata

3.5.7 kelas “View Map”

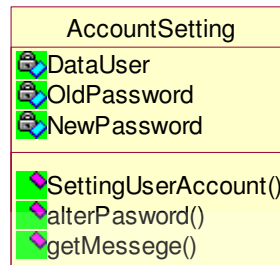
Pada kelas “view Map” digunakan untuk menampilkan peta dengan cara menginputkan posisi terakhir dilanjutkan dengan memilih obyek wisata atau komponen sehingga kita dapat mendapatkan rute yang kita inginkan. Didalam kelas ini terdapat operasi *GeopointGP()* untuk merekam posisi terakhir *User* dan *getDirection()* untuk menampilkan rute tujuan. Selain itu terdapat operasi *setOnLongClickListener()* digunakan untuk memilih obyek wisata atau cari komponen. Untuk lebih jelasnya dapat dilihat pada gambar 3.29.



Gambar 3. 29 kelas “View Map” aplikasi Website

3.5.8 kelas “Account Setting”

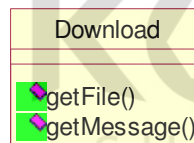
Kelas “AccountSetting” digunakan untuk mengatur informasi akun pengguna. Didalam kelas ini terdapat operasi *alterPassword* (*email,oldpassword,newpassword*) digunakan untuk mengubah *password*. Untuk lebih jelasnya dapat dilihat pada gambar 3.30.



Gambar 3. 30 kelas “AccountSetting” aplikasi Website

3.5.9 kelas “Download”

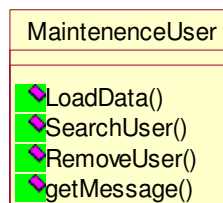
Kelas “Download” digunakan untuk mengatur proses *download* bagi pengguna. Untuk lebih jelasnya dapat dilihat pada gambar 3.31.



Gambar 3. 31 kelas “Download” aplikasi Website

3.5.10 kelas “Maintenance User”

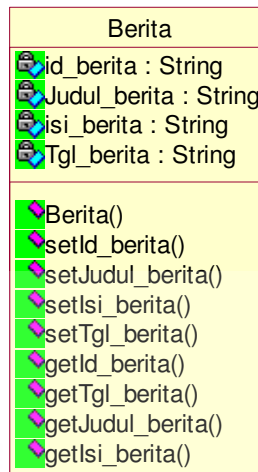
Kelas “MaintenanceUser” digunakan untuk mengatur proses menambah atau menghapus *user*. Untuk lebih jelasnya dapat dilihat pada gambar 3.32.



Gambar 3. 32 kelas “MaintenanceUser” aplikasi Website

3.5.11 kelas “Berita”

Kelas “Berita” digunakan untuk mengatur proses menambah atau menghapus berita yang akan disampaikan kepada wisatawan sebagai tambahan informasi. Untuk lebih jelasnya dapat dilihat pada gambar 3.33.



Gambar 3. 33 kelas “Berita” aplikasi *Website*

3.5.12 kelas “Kategori Komponen”

Pada kelas “KategoriKomponen” digunakan untuk menampilkan komponen obyek wisata. Komponen yang dimaksud adalah komponen pendukung wisata seperti kuliner, pompa bensin, mesin ATM, souvenir atau oleh-oleh dan tempat penginapan dimana komponen tersebut dapat menjamin akomodasi perjalanan wisatawan. Untuk lebih jelasnya dapat dilihat pada gambar 3.34.

KatagoriKomponen	
id_komponen	: String
nama_komponen	: String
alamat_komponen	: String
keterangan_komponen	: String
koordinat_komponen	
id_kategori	
nama_kategori	: String
setId_komponen()	
setNama_komponen()	
setAlamat_komponen()	
setKeterangan_komponen()	
setKoordinat_komponen()	
setId_Kategori()	
setNama_Katagori()	
getId_komponen()	
getNama_komponen()	
getAlamat_komponen()	
getKeterangan_komponen()	
getKoordinat_komponen()	
getIdKomponen()	
getNama_komponen()	

Gambar 3. 34 kelas “KategoriKomponen” aplikasi *Website*

3.5.13 kelas “Kategori Wisata”

Pada kelas “KategoriWisata” digunakan untuk menampilkan obyek wisata. Kategori obyek wisata yang dimaksud adalah wisata budaya, wisata kesehatan, wisata religi, wisata sosial, wisata politik, dan wisata olah raga.. Untuk lebih jelasnya dapat dilihat pada gambar 3.35.

KategoriWisata	
id_wisata	: String
id_kategori	: String
nama_kategori	: String
nama_wisata	: String
alamat_wisata	: String
keterangan_wisata	: String
koordinat_wisata	
setId_wisata()	
setNama_wisata()	
setAlamat_wisata()	
setKeterangan()	
setKoordinat_wisata()	
setId_kategori()	
setNama_katagori()	
getId_wisata()	
getNama_wisata()	
getAlamat_wisata()	
getKeterangan_wisata()	
getKoordinat_wisata()	
getId_kategori()	
getNama_kategori()	

Gambar 3. 35 kelas “KategoriWisata” aplikasi *Website*

3.5.14 kelas “Petunjuk”

Help	
Id_petunjuk	
Isi_petunjuk	
setContain()	
UpdateModul()	

Gambar 3. 36 kelas “Help” aplikasi *Website*

3.5.15 kelas “My Database”

Kelas “MyDatabase” digunakan untuk melakukan koneksi aplikasi dengan *database*. Didalam kelas ini terdapat atribut nama *host*, *username*, *password*, nama *database* dan *driver database*. Sedangkan operasi yang digunakan yaitu *SetConenction()*, *GetConnection()*, *CommandQuery(query)* dan *DMLCommand(query)*. Untuk lebih jelasnya dapat dilihat pada gambar 3.37.

MyDatabase	
HostName	
UserName	
Password	
DatabaseName	
SetConnection()	
GetConnection()	
CommandQuery()	
DMLCommand()	

Gambar 3. 37 kelas “MyDatabase” aplikasi Website

3.5.16 kelas “GoogleMap API Receiver”

Kelas “GoogleMap API Receiver” adalah kelas turunan dari kelas *GoogleMap API BaseReceiver* yang menangani penerimaan *direction* dari server *Google Map API*. untuk lebih jelasnya dapat dilihat pada gambar 3.38.

GoogleMap_API_Receiver	
GoogleMap_API_Receiver()	
onRegistered()	
onUnregistered()	
onError()	
onMessage()	

Gambar 3. 38 kelas “GoogleMap API Receiver” aplikasi mobile

3.5.17 kelas “GoogleMap API BaseReceiver”

Kelas “GoogleMap API BaseReceiver” digunakan untuk menangani penerimaan *direction* dari server *Google Map API*. untuk lebih jelasnya dapat dilihat pada gambar 3.39.

GoogleMap_API_BaseReceiver	
GoogleMap_API_BaseReceiver	
Registration	
Unregistered	
Error	
RegistrationID	
Service_Not_Available	
Account_Missing	
Authentication_Failed	
Invalid_Parameter	
Invalid_Sender	
GoogleMap_API_BaseReceiver()	
onMessage()	
onError()	
onRegistered()	
onUnregistered()	

Gambar 3. 39 kelas “*GoogleMap_API_BaseReceiver*” aplikasi mobile

3.5.18 kelas “*Set Position*”

Pada kelas “*SetPosition*” digunakan untuk menampilkan posisi terakhir, kelas ini terdapat operasi *GeopointGP()* untuk merekam posisi terakhir *User*. Untuk lebih jelasnya dapat dilihat pada gambar 3.40.

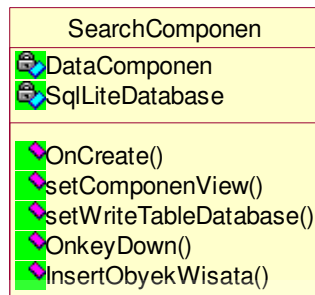
SetPosition	
StartingPoint	
NavigateThePath	
EndPoint	
NewGeoPoint()	
Start_GeoPointGP()	
list_direction()	
setNavigationButton()	
getDirection()	

Gambar 3. 40 kelas “*SetPosition*” aplikasi mobile

3.5.19 kelas “*SearchComponent*”

Pada kelas “*SearchComponent*” digunakan untuk menampilkan lokasi komponen pariwisata dengan cara menginputkan posisi terakhir dilanjutkan dengan memilih komponen wisata sehingga kita dapat mendapatkan rute yang kita

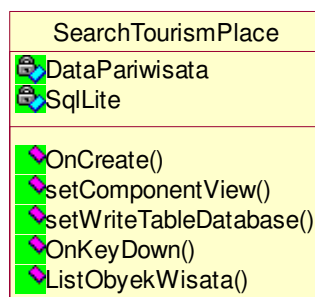
inginkan. Didalam kelas ini terdapat operasi *GeopointGP()* untuk merekam posisi terakhir *User* dan *getDirection()* untuk menampilkan rute tujuan. Selain itu terdapat operasi *setOnLongClickListener()* digunakan untuk memilih obyek wisata atau cari komponen. Untuk lebih jelasnya dapat dilihat pada gambar 3.41.



Gambar 3. 41 kelas “*SearchComponen*” *aplikasi mobile*

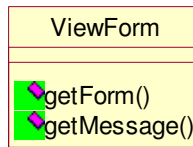
3.5.20 kelas “*Search Tourism Place*”

Pada kelas “*Search Tourism Place*” digunakan untuk menampilkan lokasi obyek pariwisata dengan cara menginputkan posisi terakhir dilanjutkan dengan memilih obyek wisata sehingga kita dapat mendapatkan rute yang kita inginkan. Didalam kelas ini terdapat operasi *GeopointGP()* untuk merekam posisi terakhir *User* dan *getDirection()* untuk menampilkan rute tujuan. Selain itu terdapat operasi *setOnLongClickListener()* digunakan untuk memilih obyek wisata atau cari komponen. Untuk lebih jelasnya dapat dilihat pada gambar 3.42.



Gambar 3. 42 kelas “*SearchTourismPlace*” *aplikasi mobile*

3.5.21 kelas “View Form”



Gambar 3. 43 kelas “ViewForm” aplikasi mobile

3.6 Component Diagram

Component Diagram atau diagram komponen adalah diagram UML yang menampilkan komponen dalam sistem dan hubungan antara mereka. Hanya ada satu tipe relasi dalam diagram ini yaitu relasi dependensi yang berarti suatu komponen yang lain atau satu komponen harus dikompilasi sebelum komponen lain yang bergantung padanya dikompilasi.

Di dalam Tugas Akhir ini, komponen-komponen dibagi menjadi dua komponen utama yaitu *KomponenGUI* yang menangani antarmuka pengguna, *KomponenKontrol* yang berisi semua kelas kontrol.

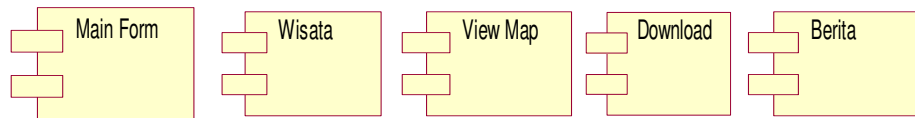


Gambar 3. 44 Hubungan antar komponen

3.6.1 *Package spesification* dalam aplikasi Web Pariwisata

A. *Package specification* Komponen *KomponenGUI*

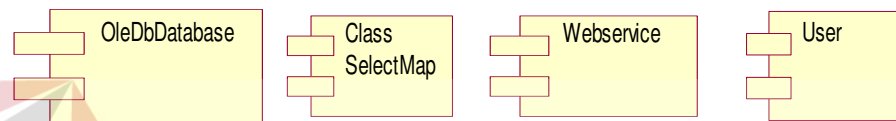
Package Spesification yang terdapat dalam komponen ini dapat dilihat pada gambar 3.45.



Gambar 3. 45 Package specification Komponen KomponenGUI

B. *Package specification* Komponen KomponenKontrol

Package Spesification yang terdapat dalam komponen ini dapat dilihat pada gambar 3.46.

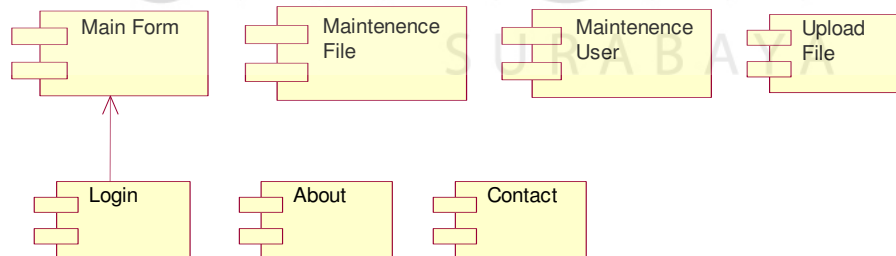


Gambar 3. 46 Package specification Komponen KomponenKontrol

3.6.2 *Package specification* dalam aplikasi Web Service

A. *Package specification* Komponen KomponenGUI

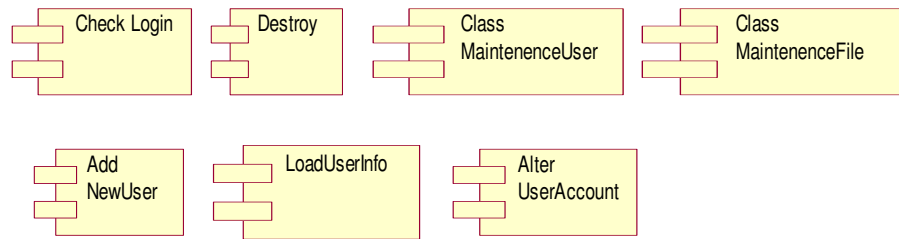
Package Spesification yang terdapat dalam komponen ini dapat dilihat pada gambar 3.47.



Gambar 3. 47 *Package specification* Komponen KomponenGUI

B. *Package specification* Komponen KomponenKontrol

Package Spesification yang terdapat dalam komponen ini dapat dilihat pada gambar 3.48

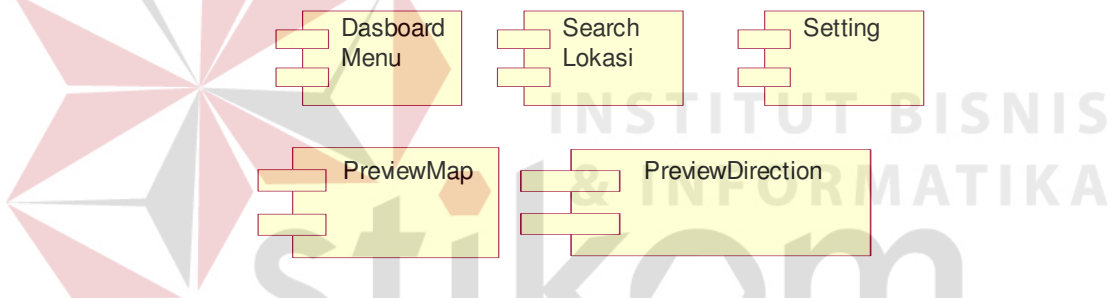


Gambar 3. 48 *Package specification* Komponen KomponenKontrol

3.6.3 *Package specification* dalam aplikasi aplikasi *Mobile*

A. *Package specification* Komponen KomponenGUI

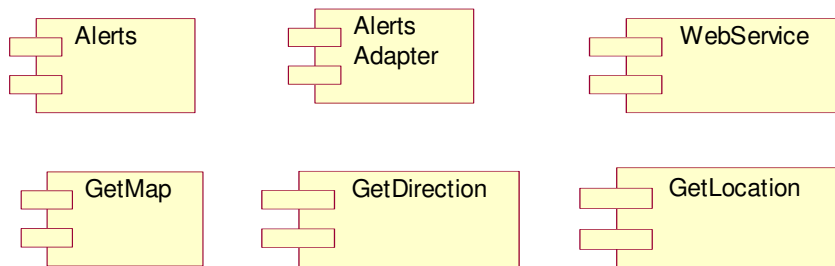
Package Spesification yang terdapat dalam komponen ini dapat dilihat pada gambar 3.49.



Gambar 3. 49 *Package specification* Komponen KomponenGUI

B. *Package specification* Komponen KomponenKontrol

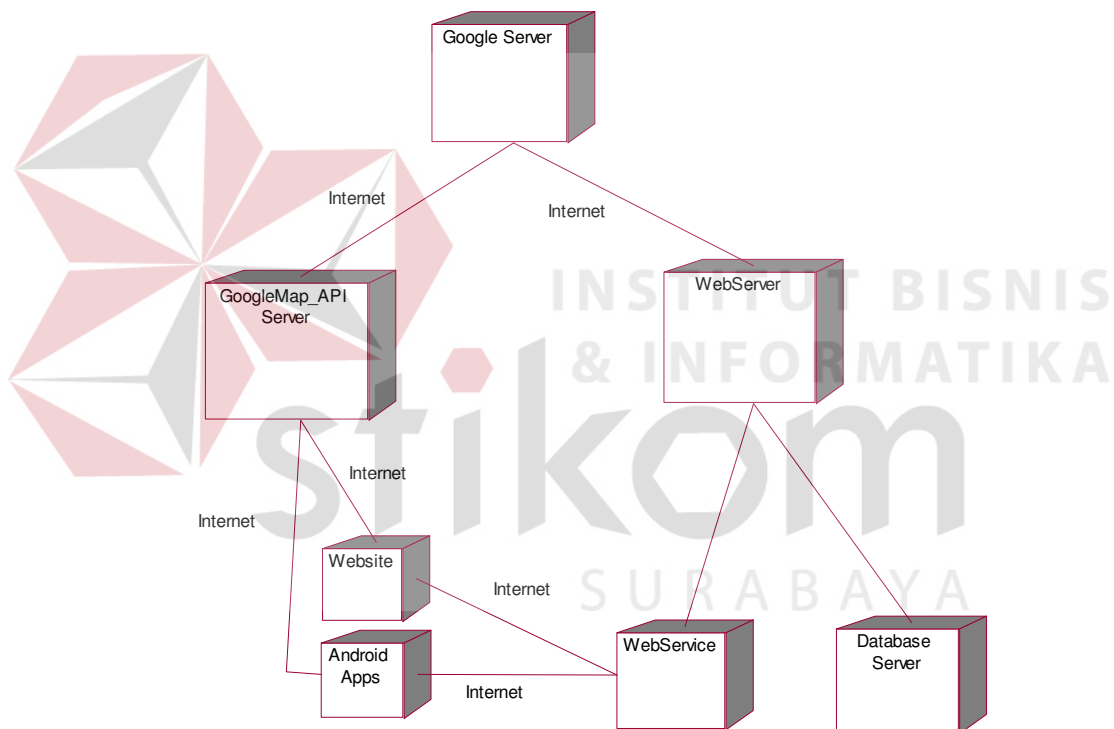
Package Spesification yang terdapat dalam komponen ini dapat dilihat pada gambar 3.50.



Gambar 3. 50 *Package specification* Komponen KomponenKontrol

3.7 Deployment Diagram

Deployment Diagram menunjukkan pandangan secara fisik dari suatu sistem dan menunjukkan bagaimana sistem diimplementasikan di perangkat nyata. *Deployment diagram* menampilkan semua *node* dalam suatu jaringan dan hubungan diantara mereka. *Node* adalah perangkat keras yang dapat menjadi *host* suatu aplikasi. *Deployment diagram* pada tugas akhir ini digambarkan oleh gambar 3.51.

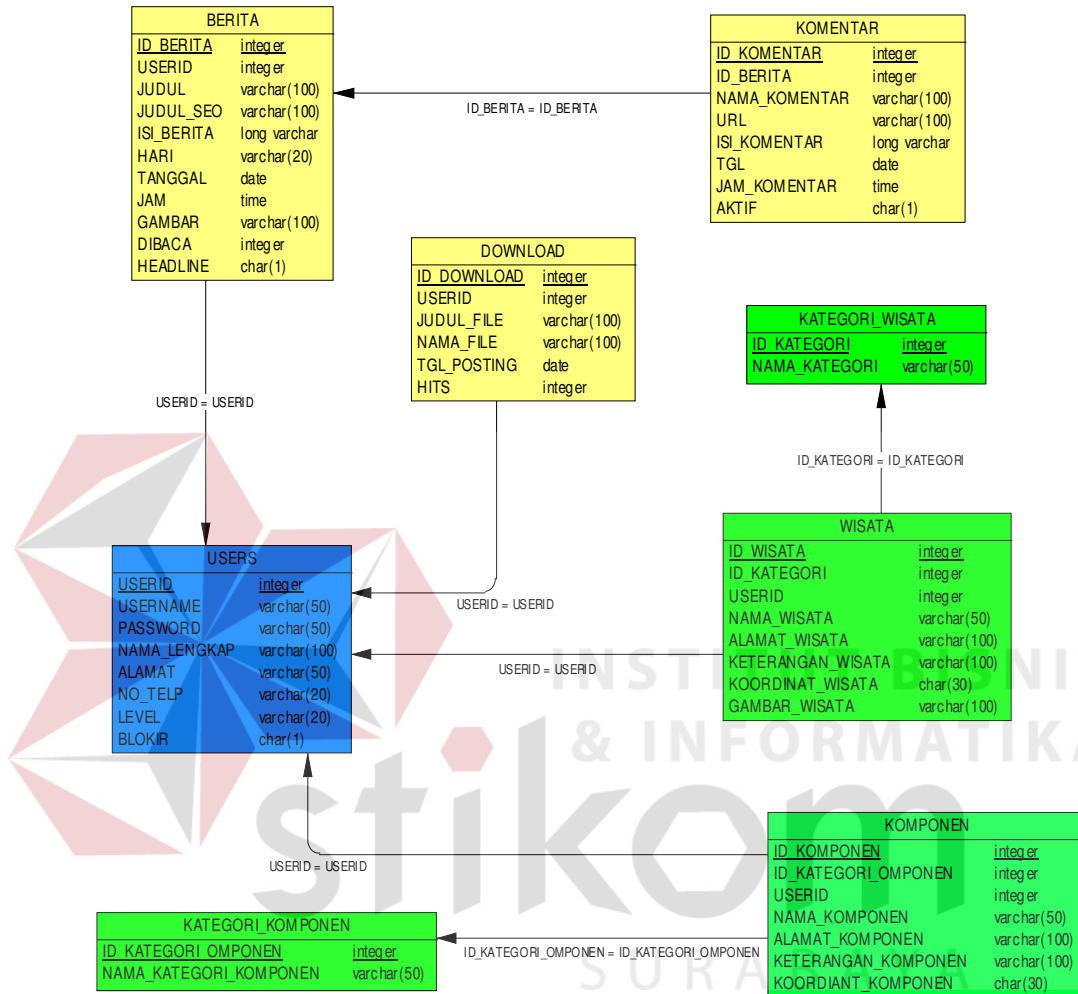


Gambar 3. 51 *Deployment diagram*

3.8 Desain Strukur Tabel

Pada gambar 3.52 merupakan desain struktur tabel dari sistem yang akan dibuat. Terdapat delapan buah tabel yang digunakan yaitu satu tabel master yang diberi warna biru, empat tabel detail yang diberi warna hijau dan tiga tabel

sementara yang di beri warna kuning. Untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 3. 52 Desain struktur table

3.9 Desain Input Output

Desain *input output* memvisualisasikan tampilan dari aplikasi. Desain *input output* Sistem Navigasi Pariwisata di Jawa Timur pada *Smartphone* Android dibagi menjadi beberapa bagian sebagai berikut:

3.9.1 Desain aplikasi *Web Server*

A. *Form Login*

Rancangan *Form* ini digunakan sebagai validasi untuk masuk pada aplikasi *web server*. *User admin* harus memasukkan *password* yang sudah diatur sebelumnya dan menekan tombol *Enter*. Rancangan menu *login* dapat dilihat pada Gambar 3.53.



Gambar 3. 53 Form login

B. *Form Home*

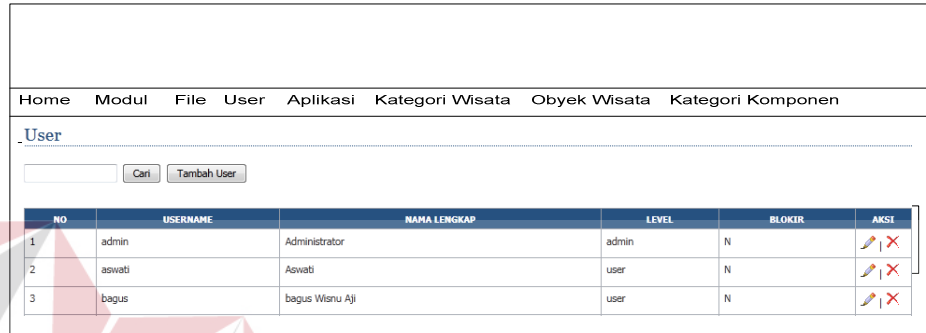
Rancangan halaman menu utama (*Web Service*) merupakan rancangan tampilan awal aplikasi *Web Service* saat *user* berhasil melakukan *login*. Halaman menu utama terdiri dari 11 menu yaitu halaman *User*, file, Modul, Aplikasi, tempat wisata, *kategori wisata*, komponen wisata, kategori komponen wisata, petunjuk, berita, dan keluar. Rancangan menu utama dapat dilihat pada Gambar 3.54.









Gambar 3. 54 *Form Home*

C. Form User

Rancangan *Form User* merupakan rancangan tampilan saat administrator melakukan *maintenance user* dan pendaftaran *user* (operator) baru agar dapat melakukan *login*. *Form* ini hanya dapat diakses oleh admin. Rancangan *form user* dapat dilihat pada Gambar 3.55.



NO	USERNAME	NAMA LENGKAP	LEVEL	BLOKIR	AKSI
1	admin	Administrator	admin	N	 
2	aswati	Aswati	user	N	 
3	bagus	bagus Wisnu Aji	user	N	 

Gambar 3. 55 *Form User*

D. Form Edit User

Rancangan *Form edit User* merupakan rancangan tampilan saat administrator melakukan *maintenance user* (operator) agar dapat melakukan *login* dan memberi hak akses yang dimiliki oleh masing – masing *user* (operator). *Form* ini hanya dapat diakses oleh admin. Rancangan *form edit user* dapat dilihat pada Gambar 3.56.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen	
Edit User	
Username	: admin
Password	: <input type="password"/> *)
Nama Lengkap	: Administrator
alamat	: <input type="text"/>
No.Telp/HP	: 0823892352
Blokir	: <input type="radio"/> Y <input checked="" type="radio"/> N
*) Apabila password tidak diubah, dikosongkan saja.	
<input type="button" value="Update"/> <input type="button" value="Batal"/>	

Gambar 3. 56 Form edit user

E. Form Tambah User

Rancangan *Form* tambah *User* merupakan rancangan tampilan saat administrator melakukan *maintenance user* (operator) agar dapat melakukan *login* dan menambahkan data *user* (operator) ke dalam *database*. *Form* ini hanya dapat diakses oleh admin. Rancangan *form* tambah *user* dapat dilihat pada Gambar 3.57.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen	
Tambah User	
Username	: <input type="text"/>
Password	: <input type="password"/>
Nama Lengkap	: <input type="text"/>
alamat	: <input type="text"/>
No.Telp	: <input type="text"/>
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>	

Gambar 3. 57 Form tambah user

F. Form Upload File

Rancangan *Form file* merupakan rancangan tampilan saat administrator melakukan *maintenance file* untuk di *upload* ke dalam *web service* agar dapat diakses oleh *user* wisatawan pada *website* pariwisata. Rancangan *form file* dapat dilihat pada Gambar 3.58.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen					
File					
<input type="text"/> <input type="button" value="Cari"/> <input type="button" value="Upload File"/>					
NO	JUDUL	NAMA FILE	TGL. POSTING	AKSI	
1	aaa	803001.JPG	07 Maret 2012	Edit	Hapus
2	koala	Koala.jpg	08 November 2011	Edit	Hapus
3	andjtrack3.1	AndjTrack3.1.apk	24 Oktober 2011	Edit	Hapus
4	kompas	Compass.apk	26 September 2011	Edit	Hapus
5	andjtrack2.0	andjtrack2.0.apk	26 September 2011	Edit	Hapus
6	andjtrack1.2	AndjT.apk	22 September 2011	Edit	Hapus
7	andjtrack1.1	andjtrack.zip	21 September 2011	Edit	Hapus
8	gambar	kabab.jpg	12 September 2011	Edit	Hapus

Gambar 3. 58 Form file

G. Form Modul

Rancangan *Form modul* merupakan rancangan tampilan saat administrator melakukan *maintenance modul* apa saja yang dapat diakses oleh *user* (operator) pada *website* (operator). Rancangan *form modul* dapat dilihat pada Gambar 3.59.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen					
Modul					
<input type="button" value="Tambah Modul"/>					
NO	NAMA MODUL	LINK	AKTIF	STATUS AKSES	AKSI
1	Modul	?module=modul	Y	admin	Edit Hapus
2	File	?module=file	Y	admin	Edit Hapus
3	Kategori Wisata	?module=kategoriwisata	Y	admin	Edit Hapus
4	Tempat Wisata	?module=wisata	Y	user	Edit Hapus
5	User	?module=user	Y	user	Edit Hapus

Gambar 3. 59 Form modul

H. Form Aplikasi Mobile

Rancangan *Form* aplikasi *mobile* merupakan rancangan tampilan saat administrator atau operator melakukan *maintenance* aplikasi atau menambahkan informasi mengenai informasi aplikasi yang akan di *upload* (operator) agar dapat agar dapat diakses atau di *download* oleh *user* wisatawan pada *website* pariwisata. Rancangan *form* aplikasi *mobile* dapat dilihat pada Gambar 3.60.



Gambar 3. 60 *form* aplikasi *mobile*

I. Form Wisata

Rancangan *Form* wisata merupakan rancangan tampilan saat administrator melakukan *maintenance* mengenai obyek wisata yang akan di informasikan kepada *user* wisatawan pada *website* pariwisata. Rancangan *form* wisata dapat dilihat pada Gambar 3.61.

ID	NAMA WISATA	ALAMAT_WISATA	KATEGORI	Aksi
1	Bukit Bromo	sukapura	wisata alam	
2	Batu Asongan	asongan, magetan	wisata alam	
3	Batu Ambar	kedungrejo	wisata bahari	
4	Museum Bumi Agung	malang	wisata sosial	
5	Garis Putih Bendo	subanda	wisata bahari	
6	Gua Akbar	tulung	wisata alam	
7	Batu Merah	pasuruan	wisata bahari	

Gambar 3. 61 *form* wisata

J. *Form* Tambah Wisata

Rancangan *Form* tambah wisata merupakan rancangan tampilan saat administrator menambahkan obyek wisata informasi mengenai data apa saja yang perlu di *upload* agar dapat di infomasi kepada *user* wisatawan, agar dapat diakses oleh *user* wisatawan pada *website* pariwisata. Rancangan *form* tambah wisata dapat dilihat pada Gambar 3.62.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen

Cari Lokasi

nama_wisata :

alamat :

Keterangan Wisata

koordinat_wisata :

















kategori : - pilih kategori -

Gambar :

Gambar 3. 62 form tambah wisata

K. Form Kategori Wisata

Rancangan *Form* kategori wisata merupakan rancangan tampilan saat administrator mengelompokkan katagori wisata, kategori wisata yang dimaksud adalah wisata budaya, wisata kesehatan, wisata religi, wisata sosial, wisata politik, dan wisata olah raga. Agar dapat diakses oleh *user* dengan mengelompokan obyek wisata pada *website* pariwisata. Rancangan *form* kategori wisata dapat dilihat pada Gambar 3.63.

kategori wisata		
Tambah Kategori		
NO	NAMA KATEGORI	AKSI
1	wisata bahari	 
2	wisata alam	 
3	wisata budaya	 
4	wisata kesehatan	 
5	wisata religi	 
6	wisata sosial	 
7	wisata politik	 
8	wisata olah raga	 

Gambar 3. 63 form kategori wisata

L. Form Tambah Komponen Wisata

Rancangan *Form* tambah komponen wisata merupakan rancangan tampilan saat administrator menambah atau menghapus komponen wisata, komponen wisata yang dimaksud adalah komponen pendukung wisata seperti kuliner, pompa bensin, mesin ATM, souvenir atau oleh-oleh dan tempat penginapan dimana komponen tersebut dapat menjamin akomodasi perjalanan wisatawan. wisata budaya, wisata kesehatan, wisata religi, wisata sosial, wisata politik, dan wisata olah raga. Rancangan *form* tambah komponen wisata dapat dilihat pada Gambar 3.64.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen	
Cari Lokasi <input type="text"/> <input type="button" value="Cari"/>	
nama_komponen :	<input type="text"/>
alamat :	<input type="text"/>
keterangan_komponen :	<input type="text"/>
koordinat_komponen :	<input type="text"/>
kategori :	- pilih kategori -
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>	

Gambar 3. 64 form tambah komponen wisata

M. Form Help

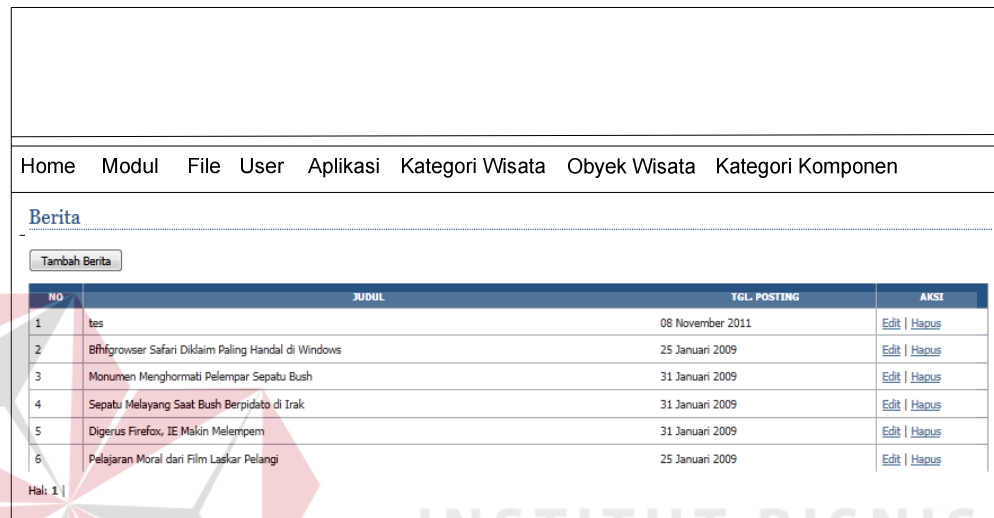
Form ini digunakan untuk menampilkan bantuan tentang cara penggunaan aplikasi.

Home Modul File User Aplikasi Kategori Wisata Obyek Wisata Kategori Komponen	
<u>HELP</u>	

Gambar 3. 65 form help

N. Form Berita

Rancangan *Form* berita merupakan rancangan tampilan saat administrator atau operator akan menambahkan informasi mengenai kejadian yang sedang terjadi. Rancangan *form* berita dapat dilihat pada Gambar 3.66.



NO	JUDUL	TGL. POSTING	AKSI
1	tes	08 November 2011	Edit Hapus
2	BHfgrowser Safari Diklaim Paling Handal di Windows	25 Januari 2009	Edit Hapus
3	Monumen Menghormati Pelempar Sepatu Bush	31 Januari 2009	Edit Hapus
4	Sepatu Melayang Saat Bush Berpidato di Irak	31 Januari 2009	Edit Hapus
5	Digerus Firefox, IE Makin Melempem	31 Januari 2009	Edit Hapus
6	Pelajaran Moral dari Film Laskar Pelangi	25 Januari 2009	Edit Hapus

Gambar 3. 66 *form* berita

3.9.2 Desain Aplikasi Website

A. Form Home Aplikasi Website

Rancangan halaman menu utama (*Website*) merupakan rancangan tampilan awal aplikasi *Website* saat *user* wisatawan berhasil mengakses *website* pariwisata. Halaman *menu* utama terdiri dari 6 menu yaitu halaman *user* wisatawan, aplikasi *mobile*, obyek wisata, berita, *help*, dan *contact Us*. Rancangan *menu* utama dapat dilihat pada Gambar 3.67.

Home	Aplikasi Mobile	Obyek Wisata	Berita	Help	Contact Us
MAP			Aplikasi Mobile		
			Kategori Wisata		
			List Berita		

Gambar 3. 67 form home aplikasi website

B. Form *Aplikasi* Aplikasi Website

Rancangan halaman aplikasi (*Website*) merupakan rancangan tampilan website untuk download aplikasi yang telah di upload oleh administrator untuk di download oleh *user* wisatwan pada website pariwisata. Rancangan *form* aplikasi utama dapat dilihat pada Gambar 3.68.

HOME	Tempat wisata	Aplikasi Mobile	Berita	Help	Contact Us
Download Aplikasi			<div style="border: 1px solid black; width: 100%; height: 40px; margin-bottom: 5px;"></div> <div style="border: 1px solid black; width: 100%; height: 40px;"></div>		

Gambar 3. 68 form *aplikasi* aplikasi website

C. Form Tempat Wisata

Rancangan halaman tempat wisata merupakan rancangan tampilan awal aplikasi *Website* saat *user* mengakses obyek wisata atau kategori obyek wisata maka *user* wisatawan akan mendapat informasi mengenai *list* obyek wisata yang berhasil di upload oleh *user* . Rancangan *form* tempat wisata dapat dilihat pada Gambar 3.69.

HOME	Tempat wisata	Aplikasi Mobile	Berita	Help	Contact Us
List Wisata					

Gambar 3. 69 Form Tempat Wisata

D. Form Berita

Rancangan *Form* berita merupakan rancangan tampilan saat *user* wisatawan mengakses form berita untuk mendapatkan informasi tentang kejadian yang sedang terjadi. Rancangan *form* berita dapat dilihat pada Gambar 3.70.

HOME	Tempat wisata	Aplikasi Mobile	Berita	Help	Contact Us
List Berita <div style="border: 1px solid black; padding: 5px;"> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> </div>			<div style="border: 1px solid black; padding: 5px; height: 40px;"></div> <div style="border: 1px solid black; padding: 5px; height: 40px;"></div>		

Gambar 3. 70 form berita

E. Form Help

Form ini digunakan untuk menampilkan bantuan tentang cara penggunaan aplikasi.

HOME	Tempat wisata	Aplikasi Mobile	Berita	Help	Contact Us
Help <div style="border: 1px solid black; padding: 5px;"> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> </div>			<div style="border: 1px solid black; padding: 5px; height: 40px;"></div> <div style="border: 1px solid black; padding: 5px; height: 40px;"></div>		

Gambar 3. 71 form help

F. Form Contact Us

Form ini digunakan untuk menampilkan bantuan tentang cara menghubungi instansi yang terkait.

HOME	Tempat wisata	Aplikasi Mobile	Berita	Help	Contact Us
------	---------------	-----------------	--------	------	------------

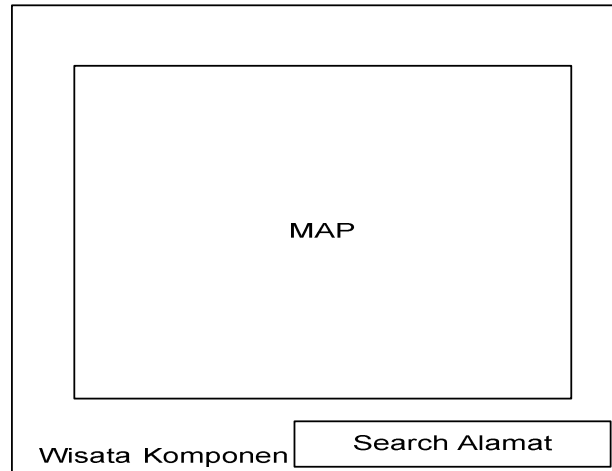
Contact Us

Gambar 3. 72 *form contact us*

3.9.3 Desain Aplikasi *Android*

A. Form Home

Rancangan *form home* merupakan rancangan untuk menampilkan peta digital pada aplikasi *mobile*. Pada peta ini kita dapat melihat posisi kita, obyek wisata beserta komponennya, dan juga catatan yang telah kita buat untuk menandai suatu tempat. Fitur lain yang ada pada halaman ini adalah kita bisa mencari lokasi berdasarkan alamat yang kita ketikkan, selain itu kita juga dapat memperoleh arah rute tujuan sesuai dengan yang kita inginkan. Kedua fitur tersebut memanfaatkan fitur yang telah disediakan oleh *Google Map*. Rancangan *form home* dapat dilihat pada Gambar 3.73.



Gambar 3. 73 form home

B. Form Kategori Wisata

Rancangan halaman kategori wisata merupakan rancangan tampilan aplikasi android saat *user* wisatawan mengakses kategori obyek wisata maka *user* wisatawan akan mendapat informasi mengenai *list kategori* obyek wisata untuk mempermudah *user* wisatawan dalam melakukan pencarian obyek wisata dengan cara pengelompokan berdasarkan jenis wisata. Rancangan *form* kategori wisata dapat dilihat pada Gambar 3.74.



Gambar 3. 74 form kategori wisata

C. Form Detail Wisata

Rancangan halaman detail wisata merupakan rancangan tampilan aplikasi android untuk menampilkan detail obyek wisata yang akan dikunjungi. Rancangan *form* detail wisata dapat dilihat pada Gambar 3.75.



Detail Wisata
Nama Wisata
Alamat Wisata
Keterangan Wisata
Kategori Wisata
Koordinat
<input type="button" value="OK"/>

Gambar 3. 75 *form* detail wisata

D. Form Kategori Komponen

Rancangan halaman kategori komponen wisata merupakan rancangan tampilan aplikasi android saat *user* mengakses kategori komponen wisata maka *user* wisatawan akan mendapat informasi mengenai *list* kategori komponen wisata untuk mempermudah *user* wisatawan dalam melakukan pencarian obyek komponen wisata dengan cara pengelompokan berdasarkan jenis komponen wisata. Rancangan *form* kategori komponen wisata dapat dilihat pada Gambar 3.76.

Kategori Komponen Wisata

List Kategori Komponen Wisata

- Hotel
- Mesin ATM
- Rumah Sakit
- Pompa Bensin
- Restoran
- Handycraft
- Art
- Transportasi

Wisata | Komponen

Gambar 3. 76 *form list* kategori komponen wisata

E. Form Detail Komponen

Rancangan halaman detail komponen wisata merupakan rancangan tampilan aplikasi android untuk menampilkan detail komponen wisata yang akan dikunjungi. Rancangan *form* detail wisata dapat dilihat pada Gambar 3.77.

Detail Komponen Wisata

- Nama Komponen
- Alamat Komponen
- Keterangan Komponen
- Kategori Komponen
- Koordinat Komponen

Gambar 3. 77 *form* detail komponen wisata

F. Form Menu Setting

Rancangan halaman *menu setting* merupakan rancangan tampilan aplikasi android untuk menampilkan *setting* data, *setting* data yang dimaksud adalah *auto update* yang dilakukan secara otomatis pada saat aplikasi pertama kali dijalankan

dalam pengambilan data mengenai pariwisata, sedangkan *manual update* dilakukan secara *manual* dan tidak *update* pada saat aplikasi pertama kali dijalankan melainkan dengan *mengupdate* sendiri pada saat *user* wisatawan menjalankan aplikasi *mobile*. Rancangan *form* setting wisata dapat dilihat pada Gambar 3.78.

Menu Setting

Setting

Auto Update
Update Data Otomatis

Manual Update
Update Data Secara Manual

Proses

Gambar 3. 78 *form* menu setting