APLIKASI PENCARIAN LOKASI TERDEKAT BERBASIS *PLATFORM* ANDROID

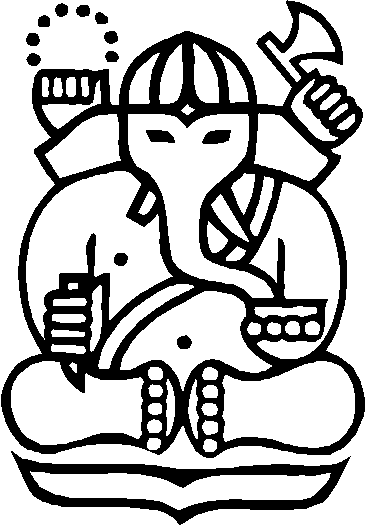
Laporan Tugas Akhir

Oleh

ANANTI SELARAS SUNNY

NIM: 13507009

Disusun sebagai syarat kelulusan tingkat sarjana



PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

Juli 2011

# DAFTAR ISI

[DAFTAR ISI i](#_Toc291095638)

[DAFTAR GAMBAR iii](#_Toc291095639)

[DAFTAR TABEL iv](#_Toc291095640)

[BAB I 1](#_Toc291095641)

[PENDAHULUAN 1](#_Toc291095642)

[1.1 Latar Belakang 1](#_Toc291095643)

[1.2 Rumusan Masalah 2](#_Toc291095644)

[1.3 Tujuan 3](#_Toc291095645)

[1.4 Batasan Masalah 3](#_Toc291095646)

[1.5 Metode Penelitian 3](#_Toc291095647)

[1.6 Sistematika Penulisan 4](#_Toc291095648)

[BAB II 5](#_Toc291095649)

[STUDI LITERATUR 5](#_Toc291095650)

[2.1 Location-based Service 5](#_Toc291095651)

[2.1.1. Definisi 5](#_Toc291095652)

[2.1.2. Arsitektur *location-based service* 6](#_Toc291095653)

[2.2. Penentuan lokasi 7](#_Toc291095654)

[2.2.1. Lokasi 7](#_Toc291095655)

[2.2.2. Teknologi penentuan posisi 8](#_Toc291095656)

[2.2.3. *Global Positioning System* 8](#_Toc291095657)

[2.3. Web Service 9](#_Toc291095658)

[2.4. Web 2.0 11](#_Toc291095659)

[*2.5.* *Android Developers Tools* 12](#_Toc291095660)

[2.6. Implementasi Location-based Service 14](#_Toc291095661)

[2.6.1. Toresto 14](#_Toc291095662)

[2.6.2. Urbanesia 15](#_Toc291095663)

[2.6.3. Foursquare 16](#_Toc291095664)

[BAB III 17](#_Toc291095665)

[DESKRIPSI SOLUSI 17](#_Toc291095666)

[3.1 Analisis Masalah Umum 17](#_Toc291095667)

[3.2 Analisis Fungsional 18](#_Toc291095668)

[3.3 Analisis non-Fungsional 25](#_Toc291095669)

[3.3 Analisis Perangkat Lunak 26](#_Toc291095670)

[3.3.1 Deskripsi Umum Sistem 26](#_Toc291095671)

[3.3.2 Kebutuhan Perangkat Lunak 27](#_Toc291095672)

[3.3.3 Model Use Case 28](#_Toc291095673)

[3.3.4 Model Analisis 31](#_Toc291095674)

[3.3.6 Teknologi 34](#_Toc291095675)

[BAB IV 35](#_Toc291095676)

[PERANCANGAN 35](#_Toc291095677)

[4.1 Perancangan Kelas 35](#_Toc291095678)

[4.1.1 Identifikasi Kelas Perancangan 35](#_Toc291095679)

[4.1.2 Sequence Diagram 36](#_Toc291095680)

[4.1.3 Diagram Kelas Perancangan 37](#_Toc291095681)

[4.1.4 Kelas Perancangan 37](#_Toc291095682)

[4.1.5 Diagram Kelas Keseluruhan 38](#_Toc291095683)

[4.2 Perancangan Antarmuka 39](#_Toc291095684)

[4.2.1 Halaman Main 39](#_Toc291095685)

[4.2.2 Halaman ListPlace 40](#_Toc291095686)

[4.2.3 Halaman DetailPlace 41](#_Toc291095687)

[4.2.4 Halaman Login 41](#_Toc291095688)

[4.2.5 Halaman Register 42](#_Toc291095689)

[4.2.6 Halaman Contribute 43](#_Toc291095690)

[4.2.7 Halaman MapView 44](#_Toc291095691)

# DAFTAR GAMBAR

Gambar 2. 1 Arsitektur location-based service (Brimicombe, 2009) 7

[Gambar 3. 1 Skema Area Pencarian Lokasi 24](#_Toc291142528)

[Gambar 3. 2 Skema Arsitektur Umum 27](#_Toc291142529)

[Gambar 3. 3 Skema Arsitektur Aplikasi 28](#_Toc291142530)

[Gambar 3. 4 Diagram Use Case 30](#_Toc291142531)

[Gambar 3. 5 Sequence Diagram UC-01 34](#_Toc291142532)

[Gambar 3. 6 Diagram Kelas Analisis UC-01 34](#_Toc291142533)

[Gambar 3. 7 Diagram Kelas Keseluruhan 35](#_Toc291142534)

Gambar 4. 1 Sequence Diagram Tahap Perancangan untuk UC-01 40

Gambar 4. 2 Kelas Perancangan UC-01 41

Gambar 4. 3 Diagram Kelas Keseluruhan 43

Gambar 4. 4 Antarmuka Main 44

Gambar 4. 5 Antarmuka ListPlace 45

Gambar 4. 6 Antarmuka DetailPlace 46

Gambar 4. 7 Antarmuka Login 47

Gambar 4. 8 Antarmuka Register 49

Gambar 4. 9 Antarmuka Contribute 50

Gambar 4. 10 Antarmuka MapView 51

# DAFTAR TABEL

Tabel 3. 1 Kebutuhan fungsional 29

Tabel 3. 2 Kebutuhan non-Fungsional 29

Tabel 3. 3 Definisi Aktor 31

Tabel 3. 4 Definisi use case 32

Tabel 3. 5 Skenario use case (UC-01) 32

Tabel 3. 6 Identifikasi Kelas Analisis UC-01 33

Tabel 3. 7 Kelas Analisis 36

Tabel 4. 1 Identifikasi Kelas Perancangan UC-01 39

Tabel 4. 2 Kelas Perancangan 42

Tabel 4. 3 Deskripsi Antarmuka Main 45

Tabel 4. 4 Deskripsi Antarmuka ListPlace 46

Tabel 4. 5 Dekripsi Antarmuka DetailPlace 47

Tabel 4. 6 Deskripsi Antarmuka Login 48

Tabel 4. 7 Deskripsi Antarmuka Register 50

Tabel 4. 8 Deskripsi Antarmuka Contribute 51

Tabel 4. 9 Deskripsi Antarmuka MapView 52

# BAB I

# PENDAHULUAN

## Latar Belakang

Ponsel biasanya hanya digunakan untuk telepon dan mengirimkan SMS (*Short Message Service*). Sekarang, ponsel memiliki fitur-fitur tambahan seperti kamera digital, radio, LCD dengan resolusi tinggi sehingga bisa memainkan video HD (*High Definition*), Wi-Fi, bahkan GPS (*Global Positioning System*) (Cybertrain, 2009). Ponsel dengan fitur lengkap tersebut disebut *smartphone*.

Belakangan ini, banyak *developer* yang mengembangkan aplikasi *location-based service* untuk *smartphone*. Hal ini dipicu oleh ketersediaan fitur GPS pada *smartphone* sehingga aplikasi semacam itu sedang banyak dikembangkan. Fungsionalitas aplikasi yang ada pun bermacam-macam, tergantung pengembangnya.

Salah satu aplikasi *location-based service* yang banyak dipakai sekarang adalah Google Map buatan Google. Aplikasi Google Map ini sudah digunakan hampir untuk semua *platform*. Google Map merupakan aplikasi yang mudah digunakan dan memiliki data yang lengkap dengan cakupan seluruh dunia sehingga tidak heran jika banyak orang menggunakan aplikasi ini. Hanya saja, aplikasi Google Map belum memiliki informasi detil tentang lokasi spesifik seperti hotel, restoran, pusat perbelanjaan (mall), ATM, pom bensin, rumah sakit, dan sebagainya, hal ini menjadi kelemahan Google Map.

Selain aplikasi Google Map, terdapat aplikasi sejenis akan tetapi hanya menyediakan pencarian lokasi tertentu saja seperti restoran atau kuliner dan kebanyakan aplikasinya buatan *programmer* luar yang tentu saja hanya menyediakan *database* lokasi di daerahnya dan aplikasi seperti ini memang belum banyak dikembangkan oleh *developer* lokal. Selain itu jika ada, secara fungsionalitasnya hanya sebatas pencarian restauran berdasarkan nama dan letaknya saja. Belum bisa melakukan pencarian lokasi berdasarkan lokasi terdekat dari pengguna tersebut berada.

Oleh karena itu dibutuhkan suatu aplikasi yang memiliki aspek fungsionalitas dalam pencarian lokasi terdekat berdasarkan lokasi keberadaan pengguna dan menyediakan informasi yang lengkap, seperti pencarian lokasi selain tempat makan yaitu Pom Bensin, Rumah Sakit, ATM, dan sebagainya. Aspek-aspek tersebut diharapkan dapat memberikan nilai tambah bagi aplikasi yang akan dibangun agar dapat memenuhi kebutuhan pengguna dalam pencarian lokasi terdekat. Dengan demikian, pengguna dapat mengefisienkan waktu pencarian lokasi dan mendapatkan lokasi yang akurat. Diharapkan di masa depan, aplikasi ini akan semakin mempermudah setiap aktivitas pengguna.

## 1.2 Rumusan Masalah

Seringkali banyak orang yang mengalami kesulitan dalam mencari suatu lokasi pada tempat yang baru dikenali. Terlebih lagi mencari lokasi tertentu yang berada di sekitarnya. Untuk menyelesaikan permasalah ini maka dibuatlah aplikasi yang bersifat mobile sehingga dapat secara efisien dan efektif membantu dalam pencarian lokasi.

Pada Tugas Akhir ini akan difokuskan pada perancangan dan penerapan aplikasi tersebut. Aspek pengguna yang ingin dicapai pada Tugas Akhir ini antar lain:

* Pencarian lokasi terdekat

Dapat melakukan pencarian lokasi terdekat dari lokasi pengguna.

* Pencarian lokasi sesuai kategori

Dapat melakukan pencarian lokasi terdekat yang sesuai dengan kategori tertentu yang diinginkan pengguna.

* Mendapatkan detil informasi lokasi tertentu

Dapat mengakses dan melihat informasi terhadap suatu lokasi tertentu yang dikehendaki pengguna.

* Mendapatkan posisi lokasi pada peta

Dapat melihat posisi lokasi sebenarnya pada peta sehingga memudahkan dalam pencarian lokasinya.

* Berkontribusi menambahkan lokasi tertentu

Pengguna dapat melakukan kontribusi lokasi tertentu untuk ditambahkan dalam basis data di server.

## 1.3 Tujuan

Tujuan yang ingin dicapai pada Tugas Akhir ini adalah menghasilkan aplikasi *location-based service* yang digunakan dalam pencarian lokasi terdekat dari lokasi pengguna berada dengan menggunakan teknologi GPS pada *smartphone* sehingga dapat mempermudah pengguna dalam pencarian suatu tempat pada daerah tertentu.

## 1.4 Batasan Masalah

Batasan masalah yang akan dikaji dan diselesaikan pada Tugas Akhir ini adalah pada pengguna aplikasi ini, yang dibatasi pada pengguna *smartphone* dengan sistem operasi Android OS dan memiliki GPS *built-in* pada perangkat yang digunakan.

## 1.5 Metode Penelitian

Metode penelitian yang akan digunakan pada Tugas Akhir ini antar lain:

1. Studi Literatur

Studi literature akan digunakan sebagai pembanding dan tolok ukur dalam pengembangan aplikasi ini sehingga aplikasi yang dibuat memiliki nilai lebih secara fungsional dibandingkan terhadap aplikasi yang sudah ada.

1. Pembangunan Perangkat Lunak

Pengembangan aplikasi ini menggunakan metode *waterfall* dengan pemodelan data dengan OOAD.

Tahapan pengembangan yang dilakukan antar lain:

1. *Requirement*

Pada tahapan ini dilakukan pengumpulan spesifikasi dan kebutuhan aplikasi yang akan dibangun.

1. *Analysis*

Pada tahapan ini kebutuhan-kebutuhan yang sudah didapatkan direpresentasikan dalam model-model.

1. *Design*

Pada tahapan ini dilakukan perancangan arsitektur aplikasi berdasarkan aspek fungsionalnya.

1. *Implementation*

Pada tahapan ini dilakukan pembangunan aplikasi berdasarkan rancangan arsitektur yang sudah dibuat.

1. *Testing*

Pada tahapan ini dilakukan pengujian aplikasi sesuai dengan aspek-aspek fungsi yang ingin dicapai.

## 1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

1. **Bab I Pendahuluan**

Pada Bab ini diberikan penjelasan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan yang digunakan untuk menyusun laporan Tugas Akhir.

1. **Bab II Studi Literatur**

Pada Bab ini berisi tentang studi literature dan landasan teori yang akan digunakan dalam penentuan *requirement*, *analysis*, *design*, *implementation*, dan *testing* pada Tugas Akhir.

1. **Bab III Deskripsi Solusi**

Pada Bab ini akan dijelaskan tentang kebutuhan dan spesifikasi aplikasi kemudian kebutuhan tersebut dianalisis sebagai dasar dari tahap pembuatan aplikasi.

1. **Bab IV Perancangan**

Pada Bab ini akan dijelaskan tentang tahapan perancangan yang menjadi dasar untuk implementasi perangkat lunak pada tahap selanjutnya.

1. **Bab V Evaluasi**

Pada Bab ini akan dijelaskan proses dan hasil evaluasi, khususnya melaporkan sejauh mana solusi yang ditawarkan.

1. **Bab V Penutup**

Pada Bab ini berisi kesimpulan dan saran yang dihasilkan dari pembuatan Tugas Akhir.

# BAB II

# STUDI LITERATUR

Pada Bab ini akan dipaparkan mengenai studi literatur maupun kelengkapan lain yang akan dipakai dalam Tugas Akhir ini. Studi literature ini mencakup penjelasan tentang *location-based service*, penentuan lokasi pengguna, web service, *Android Developers Tools*, implementasi LBS pada *mobile*. Pembahasan mengenai *Android Developers Tools* karena pada Tugas Akhir ini akan menggunakan API android tersebut.

## 2.1 Location-based Service

Pada bagian ini akan dibahas mengenai *location-based service* yang menjadi ide utama dalam pengerjaan Tugas Akhir ini.

* + 1. **Definisi**

*Location-based services (LBS) are the delivery of data and information services where the content of those services is tailored to the current or some projected location and context of a mobile user* (Brimicombe, 2009)*.*

Jadi *location-based service* adalah pengiriman data dan informasi yang berisi tentang data lokasi pengguna menggunakan perangkat *mobile*. *LBS* selalu merupakan *context-aware service*, karena lokasi adalah kasus khusus dalam konteks informasi.

*LBS* dapat diklasifikasikan menjadi dua, antara lain:

1. *Reactive LBS*

*Reactive LBS* selalu secara eksplisit diaktifkan oleh pengguna. Misalnya pengguna melakukan pencarian lokasi restrauran. *Request/*response akan diulang beberapa kali sebelum *session-*nya selesai. Reactive *LBS* memiliki karakteristik dengan sinkronisasi interaksi antara user dan service.

1. *Proactive LBS*

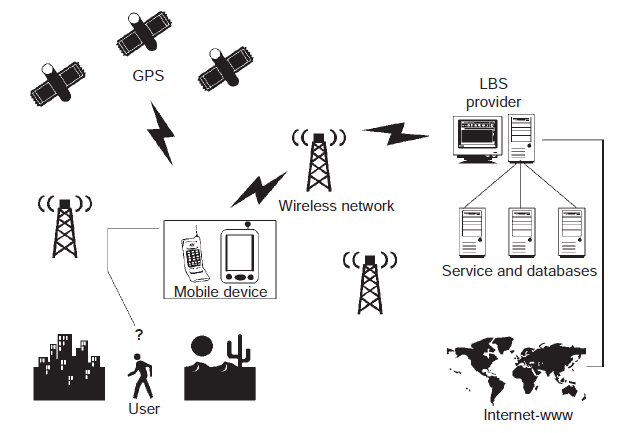
*Proactive LBS* secara otomatis akan diaktifkan segera ketika pengguna memasuki area lokasi yang sebelumnya sudah didefinisikan. Biasanya digunakan dalam pemandu wisata dalam suatu daerah, dengan mengirimkan notifikasi menggunakan *SMS* atau fasilitas lainnya. Maka *Proactive LBS* melakukan pengawasan permanen terhadap pengguna untuk mendeteksi kejadian di lokasi tertentu.

* + 1. **Arsitektur *location-based service***

*Location-based service* merupakan teknologi yang heterogen sehingga memiliki banyak sub-arsitektur untuk mendapatkan lokasi dengan wireless, internet, GPS, dan lain-lain. Pada bagian ini akan dibahas mengenai komponen utama dari arsitektur *location-based service* (Brimicombe, 2009).

Ada lima komponen dalam arsitektur yang diperlukan untuk membuat *LBS* yang efektif, antara lain:

* Informasi tentang dunia nyata yang melibatkan pengguna, dapat diakses dalam basis data dan melalui internet yang dapat memproyeksikan keberadaan pengguna.
* Perangkat *mobile* yang digunakan untuk mengakses *LBS*. Akses dapat dilakukan baik secara pasif atau aktif, bergantung dengan jenis *LBS*  yang digunakan.
* Sistem penempatan memperbolehkan lokasi geografis pengguna untuk diketahui.
* *Wireless network* dapat mengkomunikasikan antar perangkat *mobile* dan server *LBS* untuk bertukar data.
* Penyedia *LBS*, termasuk perangkat lunak dan layanan yang tersebar dan komponen yang digunakan untuk merespon query yang dikirimkan oleh pengguna.



Gambar 2. 1 Arsitektur location-based service (Brimicombe, 2009)

* 1. **Penentuan lokasi**

Aspek esensial dari *LBS* adalah dengan mendapatkan lokasi pengguna secara tepat menggunakan perangkat *mobile*. Dalam bagian ini akan dijelaskan mengenai lokasi, teknologi yang umum dalam penentuan posisi, dan tentang GPS.

* + 1. **Lokasi**

Kategori dari lokasi fisik dibagi menjadi tiga subkategori yang relevan dengan *LBS* (Kupper, 2005), antara lain:

1. ***Descriptive*** **locations**

Lokasi yang selalu memiliki hubungan geografi secara alami seperti gunung, danau, jalan, atau bangunan. Strukturnya adalah referens dari deskripsi nama, *identifier*, atau angka. Maka *descriptive locations* merupakan landasan dari konsep kehidupan sehari-hari, yang sering digunakan orang dalam mendefinisikan tempat untuk pertemuan, navigasi, atau pengantaran barang.

1. ***Spatial*** ***locations***

Lokasi yang merepresentasikan sebuah titik dalan Euclidean space. Biasanya diekspresikan dengan dua atau tiga dimensi koordinat, yang diberikan sebagai vector, setiap dari *spatial location* memperbaiki posisi satu dimensi. Berbeda dari *descriptive location*, *spatial location* tidak digunakan pada kehidupan sehari-hari. Konsep dari *spatial location* ini menyediakan basis untuk survey dan pemetaan dari *descriptive locations.*

1. ***Network locations***

*Network* *location* berhubungan dengan topologi dari komunikasi jaringan misalnya, internet atau sistem selular seperti GSM atau UMTS. Jaringan ini dibuat dari banyak jaringan lokal, yang berhubungan satu dengan yang lain dengan hirarki topologi. Layanannya dengan mengasumsikan lokasi perangkat pengguna berada dalam jaringan topologi yang diketahui. Hal ini untuk mendapatkan alamat jaringannya yang berisi informasi *routing* dan kombinasi layanan *directory*, untuk pemetaan nomor, *identifier*, atau nama skema, misalnya saja IP address.

* + 1. **Teknologi penentuan posisi**

Ada banyak teknologi yang digunakan untuk menentukan posisi dalam kombinasi solusi *hybrid*. Metode dalam penentuan posisi secara umum bisa dikategorikan dalam *network-based, device-based,* dan *hybrid-based* (Kupper, 2005). Bisa juga dibedakan dengan *integrated* atau *stand-alone positioning* *infrastructure* dan juga bisa bergantung terhadap *satellite-based, network-based,* atau *indoor* *infrastructures*. *Device-based* biasanya menggunakan GPS yang berdasarkan *satellite-based* dan *stand-alone positioning* *infrastructure*. *Network-based* digunakan untuk komunikasi dan pengiriman data dengan gateway ke jaringan dan sistem yang lain. Sebagai tambahan, teknologi yang menggunakan jarak bisa juga digunakan untuk menentukan lokasi pengguna dalam area jarak yang ditentukan, contohnya penggunaan RFID.

* + 1. ***Global Positioning System***

GPS adalah *Satellite-based* sistem navigasi global radio. GPS dikembangkan sebagai sistem pertahanan dari Departemen Pertahanan Amerika Serikat.

Prinsip desain GPS (Brimicombe, 2009) secara umum antar lain:

* Cocok untuk semua platform baik yang berupa pesawat, kapal, bahkan kendaran darat.
* Menyediakan posisi secara *real-time* bersamaan dengan kemampuan untuk menentukan waktu dan kecepatan.
* Mereferensikan semua posisi ke dalam sebuah *single global geodetic datum*.
* Mampu menyediakan keakuratan data.
* Mampu melayani penggunanya secara tidak terbatas di seluruh dunia.
* Memiliki biaya akses yang murah (kenyataannya gratis) dengan menggunakan daya yang rendah.

Banyak aplikasi LBS yang didukung oleh GPS. Penentuan lokasi pengguna menggunakan GPS adalah metode *device-based positioning*. Perangkat dengan *built-in* GPS bisa menawarkan akurasi lebih tinggi antar 3 – 5 meter. Tetapi akurasi tersebut mungkin saja tidak sesuai dengan keseluruhan lingkungan. Tetapi terdapat kelemahan dengan penggunaan GPS yang ada di perangkat *mobile*, seperti penggunaannya membutuhkan daya yang lebih tinggi dan meningkatkan biaya untuk manufaktur perangkatnya.

* 1. **Web Service**

Web service dapat mengkonversi suatu aplikasi menjadi aplikasi web. Dasar dari web service platform adalah XML dan HTTP.

Prinsip dari web service (W3School, 2010), antar lain:

* Web service adalah komponen dari aplikasi.
* Web service berkomunikasi menggunakan protokol yang terbuka.
* Web service merupakan *self-contained* dan *self-describing*.
* Web service bisa ditemukan menggunkan UDDI.
* Web service bisa digunakan dengan aplikasi lain.
* XML merupakan basis dari web service.

Dasar dari web service adalah XML dan HTTP. XML menyediakan bahasa yang bisa digunakan antar platform yang berbeda dan bahasa pemrograman yang berbeda dan lagi dapat mengekspresikan pesan dan fungsi yang rumit.

Web service platform elemen, antara lain:

* SOAP (*Simple Object Access Protocol*)

SOAP adalah protokol berbasiskan XML yang membiarkan aplikasi mempertukarkan informasi melalui HTTP. SOAP merupakan komunikasi protokol dan format yang digunakan untuk mengirimkan pesan. SOAP dirancang untuk komunikasi menggunakan internet, sebuat platform yang berdiri sendiri dengan bahasa yang berdiri sendiri. SOAP bentuknya sederhana dan dapat di-extent. SOAP juga bisa menembus firewall dan merupakan standar W3C.

* UDDI (*Universal Description, Discovery and Integration*)

UDDI adalah *directory* untuk menyimpan informasi tentang web service dan merupakan *directory* antarmuka web service yang mendeskripsikan WSDL. Komunikasi dilakukan menggunakan SOAP.

* WSDL (*Web Service Description Language*)

WSDL adalah sebuah XML format untuk mendeskripsikan layanan jaringan sebagai satu set operasi *endpoint* pada pesan. Operasi dan pesan dideskripsikan secara abstrak dan terikat pada protokol jaringan yang konkrit dan pesan format untuk mendefinisikan *endpoint*. WSDL dapat di-extent untuk memperbolehkan deskripsi dari *endpoint* dan pesan yang tanpa memperhatikan format pesan atau protokol jaringan yang digunakan untuk berkomunikasi, tetapi hanya mengikat deskripsi dalam dokumen yang mendesripsikan cara penggunaan WSDL bersamaan dengan SOAP 1.1, HTTP, dan MIME.

Struktur dokumen WSDL terdiri dari elemen types, message, portType, dan binding.

Web service memiliki interoperability dengan prioritas yang tinggi. Ketika sebagian besar platform bisa mengakses web menggunakan web browser, platform yang lain bisa berinteraksi. Web service dapat mengangkat aplikasi pada level selanjutnya. Dengan menggunakan web service, aplikasi dapat di-*publish* fungsi atau pesannya. Web service menggunakan XML untuk struktur kodenya dan SOAP untuk mentranportasikan XML menggunakan protokol terbuka. Dengan web service, dua atau lebih aplikasi bisa terhubung dan mengakses satu sama lain (W3C, 2001).

Web service memiliki dua tipe penggunaannya, antar lain:

* Penggunaan kembali komponen aplikasi

Seringkali aplikasi memerlukan suatu fungsi yang digunakan berulang-ulang. Maka web service menawarkan komponen aplikasi seperti, konversi *currency*, laporan cuaca, bahkan translasi bahasa sebagai layanan.

* Berhubungan menggunakan perangkat lunak yang sudah ada.
* Web service bisa menolong untuk menyelesaikan masalah interoperabilitas dengan memberikan aplikasi ini jalan lain untuk berhubungan dengan datanya. Dengan menggunakan web service bisa mempertukarkan data antar aplikasi yang berbeda dan platform yang berbeda.
  1. **Web 2.0**

Web 2.0 dihubungkan, dengan aplikasi web yang memfasilitasi persebaran dan penyaluran informasi, *interoperability*, desain yang berpusat pada pengguna, dan kolaborasi pada World Wide Web. Web 2.0 memperbolehkan pengguna untuk berinteraksi dan berkolaborasi satu sama lain. Contoh Web 2.0 adalah web yang berkaitan dengan *social networking* seperti twitter atau facebook, blog, wikis, dan video sharing.

Secara konsep dalam Tugas Akhir ini, menggunakan konsep yang dimiliki oleh Web 2.0, yaitu berbagi *content*. Setiap orang dapat berbagi lokasi atau menyumbangkan lokasi yang diketahui, tidak hanya sekedar mendapatkan informasi lokasi saja.

Konsep dan karakteristik Web 2.0 (O’Reilly, 2005) yang digunakan dalam Tugas Akhir, yaitu:

* *Lightweight Programming Model*

Lightweight programming itu berkaitan dengan penggunaan SOAP web service atau menyediakan XML data melalui HTTP yang biasa disebut REST (Representational State Transfer). Contoh lightweight programming adalah RSS yang sudah sering digunakan, data yang didapatkan dengan menggunakan RSS adalah berupa XML. Amazon.com juga menggunakan lightweight programming dengan menyediakan akses web service yang bisa dimanfaatkan oleh pihak lain.

Pada Tugas Akhir ini menggunakan lightweight programming dengan memanfaatkan web service dalam penyaluran data. Penggunaan web service ini dinilai sederhana dan efisien karena mentransferkan data berupa XML.

* *Rich User Experiences*

Web 2.0 menawarkan user experience yang kaya seperti konten yang dinamik dan memiliki interaksi dengan pengguna. Contohnya search engine seperti google yang menggunakan AJAX untuk load data dan konten yang diinginkan pengguna serta memberikan pilihan pada pengguna saat melakukan pencarian. Jika dikaitkan dengan Tugas Akhir, konten yang disediakan juga bersifat dinamik. Saat pengguna terhubung dengan fasilitas internet maka data-data akan ter-*update* secara otomatis.

* *Users Add Value*

Kunci keuntungan kompetitif dalam aplikasi internet adalah dengan menyediakannya fasilitas kepada pengguna untuk menambahkan data yang dikehendaki. Memasukkan data pengguna dapat menambahkan nilai lebih dari aplikasi yang dibuat. Dalam Tugas Akhir ini juga memiliki fasilitas untuk menambahkan data yang berasal dari pengguna, yaitu data lokasi pada konteks aplikasi ini. Data lokasi tersebut cukup detail dan dapat dilihat dan direview oleh pengguna lainnya sehingga fitur content-sharing dapat berjalan dengan baik.

* 1. ***Android Developers Tools***

Dalam pengembangan Tugas Akhir ini akan menggunakan *Android Developers Tools* sebagai framework-nya (Android, 2011). Karena aplikasi akan dijalankan pada sistem operasi Android.

Dalam implementasinya akan menggunakan android SDK untuk pemrogramannya dan untuk visualisasi petanya menggunakan Map dari google API. Dengan menggunakan Map dari google API akan memudahkan dalam intrepetasi ke dalam petanya. Dan untuk basis data pada internal aplikasi akan menggunakan SQLite yang sudah tersedia dalam android SDK.

Penggunaan sistem operasi Android ini disebabkan sedang populernya pengembangan aplikasi untuk sistem operasi Android dan mendapat dukungan penuh untuk menggunakan google API yang memudahkan dalam pengembangan aplikasi.

Pemilihan sistem operasi Android sebagai sistem operasi yang akan menjalankan aplikasi yang dihasilkan pada Tugas Akhir, alasannya antara lain:

1. *Developer*-*kit* mudah didapatkan.
2. *Developer*-*kit* dapat digunakan pada berbagai platform, seperti windows, mac os, dan linux.
3. *Developer*-*kit* lengkap mencakup semua aspek pengembangan aplikasi.
4. Referensinya lengkap dan banyak ditemukan.
5. Terdapat GPS emulator untuk pembuatan aplikasi *location-based service*.
6. Didukung oleh google API.
7. Dokumentasi setiap *library* lengkap dan jelas.
8. Pembuatan antarmuka pada android lebih mudah dibandingkan menggunakan J2ME.

Untuk implementasi aplikasi *location-based service* dengan memanfaatkan teknologi *platform* android, dibutuhkan beberapa *library* *framework*, yaitu:

1. Android SDK dan ADT

Android SDK dan ADT digunakan untuk menjalankan emulator. Emulator android ini berperan seperti layaknya Android OS. Saat *start-up* awal, emulator akan menjalankan Android OS dan *loading* emulatornya agak lama sampai siap untuk menjalankan aplikasinya. Jika aplikasi dapat berjalan dengan baik pada emulator maka aplikasi dijamin dapat dipakai di android *mobile*.

1. DrawDroid

DrawDroid adalah aplikasi untuk memudahkan pembuat antarmuka pada aplikasi android. Aplikasi ini berbasiskan java dan cukup dijalankan jar-nya saja. Dengan menggunakan DrowDroid, membuat antarmuka cukup *drag-drop* saja, lalu *generate* xml-nya. Penting sekali untuk melakukan *generate* xml, karena antarmuka aplikasi android itu berbasiskan xml.

1. *Library* KSOAP2

*Library* KSOAP2 digunakan untuk melakukan koneksi web service dengan cara mengirimkan SOAP *request* dan mengambil SOAP *response*. Jika sudah mendapatkan SOAP *response*, langkah selanjutnya adalah mem-*parsing* data yang didapat.

1. Google Map API

Untuk tampilan peta yang diperlihatkan ke layar menggunakan Google Map API. *Platform* Android didukung penuh oleh Google API.

5. SQLite

SQLite digunakan untuk melakukan konektivitas dan penyimpan data pada aplikasi android. SQLite adalah *flat database* yang ringan dan cocok untuk dipakai pada aplikasi *mobile*. SQLite ini sudah termasuk di dalam android ADT jadi tidak memerlukan *library* yang terpisah.

* 1. **Implementasi Location-based Service**

Pada bagian ini akan dipaparkan mengenai implementasi *location-based service* yang sudah dibuat. Hal ini dilakukan untuk perbandingan dengan aplikasi yang akan dibuat pada Tugas Akhir.

* + 1. **Toresto**

Toresto adalah aplikasi LBS android buatan GITS Indonesia, salah satu developer Indonesia yang menghasilkan aplikasi android. Toresto ini adalah aplikasi yang memudahkan pengguna dalam mencari lokasi restoran atau tempat makan yang ada di Indonesia. Toresto dapat melakukan pencarian tempat makan yang terdekat dari lokasi pengguna dan menampilkannya pada aplikasi tersebut. Hasil restoran atau tempat makan yang didapatkan dapat diurutkan sesuai dengan nama tempat atau berdasarkan jarak yang terdekat dari lokasi pengguna. Pada aplikasi ini juga dapat dilakukan pencarian lokasi tempat makan menggunakan nama.

Toresto merupakan aplikasi yang bersifat gratis, karena pengguna dapat men-*download* aplikasi tanpa membayar pada pihak developer. Toresto juga menyediakan fitur promo untuk memfasilitasi restoran atau tempat makan yang sedang mengadakan promo. Kedepannya Toresto bisa menjadi sarana promosi yang bersifat bisnis.

Dari segi antarmukanya, Toresto memiliki tampilan yang bagus dan sederhana. Simbol dan icon yang digunakan mudah dimengerti oleh pengguna. Informasi yang berkaitan dengan tempat makan tersebut ditampilkan dengan antarmuka yang menarik dan informasi yang lengkap. Informasi yang ditampilkan antara lain alamat restoran, nomor telpon, harga makanan, deskripsi singkat restoran, fasilitas yang dimiliki restoran, jarak restoran dari lokasi pengguna, daftar menu yang dilengkapi dengan gambar, dan promo jika restoran sedang mengadakan promo penjualan. Informasi yang disediakan oleh Toresto sudah cukup lengkap.

Pada aplikasi ini juga disediakan fitur menambahkan lokasi yang pengguna inginkan dengan cara *login* melalui aplikasi tersebut. Login bisa menggunakan akun Facebook atau membuat akun Toresto dengan cara *register* terlebih dahulu melalui aplikasinya. Sebenarnya aplikasi Toresto ini masih dalam tahap pengembangan jadi fitur yang ada sekarang masih bisa bertambah, contoh fitur yang belum diimplementasikan adalah fitur menambahkan rating pengguna yang masih belum diimplementasikan.

* + 1. **Urbanesia**

Urbanesia adalah aplikasi web buatan developer Indonesia. Urbanesia merupakan sebuah *Lifestyle City Online Directory* yang menyajikan informasi mengenai gaya hidup dan tempat-tempat bisnis yang ada di kota-kota besar seluruh Indonesia. Urbanesia memungkinkan pengguna untuk mencari bermacam-macam jenis lokasi. Pada aplikasi ini terdapat fitur promo yang membuat pengguna tetap ter-*up-to-date* tentang informasi tersebut. Selain promo ada juga fitur event, fitur ini membantu pengguna untuk mengetahui event yang sedang berlangsung di Jakarta.

Kelebihan dari Urbanesia adalah banyaknya pengelompokan kategorinya. Terdapat berbagai macam kategori yaitu kuliner, hiburan, pusat perbelanjaan, sampai fasilitas pendidikan. Saat pengguna mengunjungi tempat tersebut bisa berbagi tempat tersebut dengan cara berbagi informasi melalui twitter. Banyak sekali fitur pengguna yang disediakan oleh Urbanesia. Pengguna dapat menulis artikel, meng-*update* *event* terkini, menyebarkan promo menarik, me-review tempat-tempat favorit, dan dapat meng-upload foto dalam review yang dituliskan pengguna.

Urbanesia juga merupakan bentuk dari peluang bisnis secara *online*. Para pemilik tempat dapat mendaftarkan tempatnya pada Urbanesia secara gratis sehingga mereka dapat mempromosikan tempat yang mereka miliki dengan mudah, efektif, dan tanpa biaya. Urbanesia juga menyediakan map dari Google Map yang digunakan untuk *tagging* lokasi tempat sehingga memudahkan pengguna untuk mencari tempatnya. Terdapat semacam cpanel untuk mengatur data dan informasi yang akan digunakan. Urbanesia juga menyediakan interaksi yang baik dengan pelanggan dengan fasilitas *make friends*.

Saat ini database Urbanesia yang lengkap di daerah Jakarta. Urbanesia juga masih dalam tahap pengembangan jadi banyak fitur yang masih belum ditambahkan, padahal fitur yang disediakan sekarang sudah cukup lengkap. Fitur yang akan ditambahkan pada Urbanesia adalah *shopping* *cart*, *report*, *ticket box*, dan masih banyak lagi yang lainnya.

Fitur utama yang diunggulkan oleh Urbanesia adalah pencarian lokasi, event, promo, dan pengguna, dengan cukup mengetikan nama yang ingin dicari. Pencarian juga bisa dioptimasi dengan mencari berdasarkan lokasi. Akses penggunaan fitur pencarian bisa digunakan tanpa harus login terlebih dahulu. Keperluan login dibutuhkan saat pengguna akan membuat review tempat favorit, membuat artikel, membuat promo, dan membuat event. Fasilitas login dapat menggunakan akun Facebook, Twitter, atau register terlebih dahulu untuk mendapatkan akun Urbanesia.

* + 1. **Foursquare**

Foursquare adalah *location-based mobile platform* yang mudah digunakan dan menjadikan eksplorasi menarik. Caranya dengan checking in menggunakan aplikasi *smartphone* atau SMS, pengguna dapat berbagi lokasinya kepada temannya ketika mengumpulkan poin dan *virtual badges*. Foursquare memandu pengalaman dengan memperbolehkan pengguna untuk menandai informasi tentang jalan yang akan dikunjungi dan memunculkan pilihan relevan dekat dengan jalan tersebut.

Aplikasi Foursquare tersedia dalam berbagai platform seperti iPhone, BlackBerry, Android, Palm, OVI Nokia, dan perangkat lain. Check in melalui perangkat mobile, semakin sering check in maka akan mendapatkan badges tertentu. Saat teman melakukan check in bisa disinkronisasi ke twitter sehingga persebaran informasinya menjadi lebih luas. Hanya saja Foursquare tidak memberikan data dan informasi yang detil tentang lokasi tempat check in tersebut dan tidak dapat memberikan komentar atau rating. Jadi hanya sebatas memberi tau orang lain, keberadaan orang lain sekarang.

Pengguna Foursquare sudah mencapai 6.5 juta orang di seluruh dunia. Pengguna Foursquare juga cukup banyak di Indonesia. Dengan melihat banyaknya pengguna di seluruh dunia ini, Foursquare dapat dijadikan lahan bisnis dan promosi toko atau restoran. Sebenarnya saat pengguna berbagi saat check ini di suatu tempat, mereka sudah mengajak pengguna lain untuk tertarik check ini ditempat tersebut.

# BAB III

# DESKRIPSI SOLUSI

Pada Bab ini akan dipaparkan tentang hasil analisis Tugas Akhir yang lebih mendalam dan menguraikan pendekatan dan detil solusi yang dipakai untuk menyelesaikan persoalan Tugas Akhir. Bagian awal akan membahas analisis masalah global dan perangkat lunak. Bagian selanjutnya akan dibahas tentang solusi yang diterapkan untuk menyelesaikan permasalahan.

## 3.1 Analisis Masalah Umum

Sejak perangkat ponsel ditanamkan fitur GPS-A maka memetakan lokasi setiap orang bukanlah hal yang sulit. Dengan mengetahui posisi lokasi sekarang banyak hal yang dapat dilakukan, salah satu hal yang sering dilakukan adalah navigasi saat berkendaraan. Lokasi yang didapatkan dari GPS-A kemudian dipetakan pada peta dan sistem dapat memperkirakan perjalanan lokasi selanjutnya.

Pertukaran data informasi lokasi dan posisi seseorang merupakan konsep dari sistem Location-based Service. Location-based Service yang biasa disebut LBS ini erat kaitannya dengan GPS dalam mendapatkan lokasi seseorang. Sebenarnya untuk mendapatkan lokasi seseorang tidak hanya menggunakan GPS saja, tetapi bisa juga menggunakan Wifi atau mobile network. Meskipun terdapat banyak cara untuk mengakses dan mendapatkan lokasi seseorang, GPS memiliki tingkat keakuratan paling tinggi sehingga didapatkan koordinat lokasi yang cukup tepat pada koordinat yang sesungguhnya.

Dengan dukungan fitur GPS ini aplikasi yang menerapkan konsep LBS sedang banyak dikembangkan. Jenis aplikasi yang menggunakan konsep ini ada banyak sekali, salah satu aplikasi yang paling banyak dipakai bahkan hampir ada di semua platform adalah aplikasi Google Map. Dengan aplikasi Google Map ini dapat melakukan *tracking* perjalanan dari lokasi sekarang menuju lokasi yang akan dituju. Adanya fitur seperti itu maka sangat memudahkan seseorang yang tidak begitu hafal dengan suatu daerah untuk mencapai daerah itu tanpa perlu bertanya kepada orang. Cukup mengikuti setiap petunjuk dan arahan untuk mencapai lokasi yang dimaksud.

Seringkali di saat yang dibutuhkan, pencarian suatu lokasi terdekat dari posisi terdekat pengguna menjadi sangat penting. Saat berjalan-jalan ke suatu daerah yang cenderung asing didatangi, lalu tiba-tiba ingin mengisi bensin, padahal tidak begitu mengerti daerah tersebut. Jika dalam keadaan yang terdesak seperti itu, diharuskan mendapatkan lokasi pom bensin yang terdekat. Bahkan Google Map belum memiliki fitur seperti itu.

Mungkin untuk saat ini terdapat beberapa aplikasi yang dapat mencari lokasi terdekat dari pengguna, tetapi pencarian lokasi biasanya untuk jenis lokasi seperti restoran atau tempat makan saja. Belum ada aplikasi yang memfasilitasi pencarian yang lebih banyak jenis kategori pencariannya. Kalangan developer lokal juga belum ada yang mengembangkan aplikasi LBS semacam ini. Oleh karena itu dibutuhkan aplikasi yang dapat melakukan pencarian lokasi terdekat dengan kategori pencarian yang bermacam-macam.

## 3.2 Analisis Fungsional

Sebelumnya sudah dibahas tentang analisis masalahnya secara umum. Untuk menjawab masalah kesulitan dalam melakukan pencarian lokasi yang terdekat dari lokasi sekarang, dibuatlah suatu aplikasi yang dapat melakukan pencarian lokasi terdekat menggunakan teknologi GPS dan memanfaatkan konsep Location-based Service. Untuk merealisasikan aplikasi tersebut diperlukan fungsionalitas yang dapat mendukung agar aplikasi berjalan dengan baik dan memiliki nilai guna yang tinggi.

* Aplikasi bisa mendapatkan lokasi pengguna

Dapat melakukan pencarian lokasi terdekat merupakan fungsionalitas utama dari aplikasi ini. Untuk dapat melakukan pencarian lokasi terdekat maka diperlukan akses lokasi pengguna sekarang. Sebelumnya sudah disinggung untuk mendapatkan lokasi pengguna dapat menggunakan tiga cara, yaitu:

1. GPS (Global Positioning System)

Tidak semua perangkat smartphone atau ponsel memiliki fitur GPS. Fitur GPS merupakan fitur tambahan. GPS mulai dimasukkan ke dalam perangkat ponsel baru-baru ini saja, karena sebelumnya fitur GPS ini tidak hadir dalam fungsionalitas mobile phone. GPS memiliki keakuratan yang paling tinggi, karena memanfaatkan sistem satelit melalui hubungan radio gelombang GPS. GPS menyediakan posisi secara real time dan kecepatannya tinggi. Saat melakukan akses lokasi menggunakan GPS, tidak dipungut biaya akses.

Meskipun GPS terkenal unggul dalam keakuratan data lokasi yang dihasilkan, ternyata terdapat juga kelemahan fitur ini. GPS yang sudah paling akurat saat ini pun sebenarnya masih belum mencapai performa yang 100% untuk keakuratan data lokasi yang dihasilkan. Noise bisa membuat error 1 meter sampai 10 meter dari lokasi yang sebenarnya. Objek seperti gunung dan gedung tinggi juga menyumbakan penurunan keakuratan GPS berkurang menjadi 30 meter dari keakuratan pada kondisi normal (map-gps-info, 2011). Selain itu. GPS pada mobile phone menggunakan daya yang tinggi sehingga boros batere. Belum banyak pengguna smartphone yang terdapat fasililtas GPS karena harganya tergolong mahal.

1. Wifi Network

Seperti halnya GPS, perangkat Wifi yang ada pada mobile phone merupakan fitur tambahan. Tidak semua smartphone memiliki fitur ini. Wifi bisa digunakan untuk menentukan posisi pengguna dengan menyambungkan perangkat mobile ke *access-point* terdekat. Penentuan lokasi pengguna menggunakan Wifi terdapat metode yang disebut Wifi Positioning System atau yang disingkat dengan WPS.

WPS atau yang biasa dikenal dengan *fake* GPS ini dikembangkan oleh Skyhook (Evans, 2008). Cara kerjanya dengan menggunakan sinyal yang dipancarkan oleh jaringan Wifi lokal dan *public wireless hotspot* untuk menelusuri lokasi pengguna dalam jarak 20 meter. Lalu, ketika GSM triangulasi mendapatkan informasi kota tempat pengguna berada, WPS menjadi lebih akurat. WPS dapat mendapatkan informasi yang lebih detil tentang lokasi pengguna sekarang, seperti nama jalan. Walaupun WPS memiliki galat sampai 20 meter, hal ini cukup bagus terlebih lagi dibandingkan dengan GPS hanya berbeda 10 meter dalam hal keakuratan data lokasinya.

WPS bergantung dengan database yang besar dengan menyediakan daftar lokasi lebih dari 23 juta Wi-Fi access point. Untuk mendapatkan informasi ini, Skyhook mengembangkan kendaraan khusus untuk melakukan survey jalan di kota-kota. Dengan metode ini dapat men-*scan* access point dan memetakan lokasinya secara geografi. Sekarang ini area yang sudah didukung WPS adalah kota-kota yang memiliki populasi tinggi seperti London, Birmingham, Bristol, Liverpool, Manchester, Sheffield, dan Leeds(Evans, 2008).

Keakuratan lokasi yang didapatkan menggunakan WPS memang lebih rendah dari GPS tetapi sudah cukup untuk memetakan lokasi keberadaan pengguna. WPS memiliki kelemahan yang sangat fatal yaitu area yang mendukung teknologi WPS hanya sedikit, untuk negara seperti Indonesia bahkan tidak terdapat dukungan teknologi tersebut. Penggunaan teknologi WPS tidak dapat digunakan oleh pengguna di Indonesia.

1. Mobile Network

Penentuan posisi pengguna menggunakan mobile network adalah dengan memanfaatkan BTS operator GSM terdekat. Tingkat keakuratan menggunakan mobile network tergolong yang paling rendah dibandingkan dengan menggunakan Wifi atau GPS. Keakuratan data lokasi yang didapatkan hanya sebatas lokasi kota pengguna berada. Mobile network tidak dapat mendeteksi secara pasti keberadaan pengguna yang sebenarnya dan sering memberikan data lokasi yang salah, tergantung BTS terdekat dari pengguna sekarang. Karena *coverage-area* mobile network terlalu luas sehingga tidak bisa mendapatkan data lokasi yang akurat seperti GPS.

Setelah menganalisis keakuratan yang didapatkan dengan tiga metode diatas maka bisa diambil simpulan bahwa untuk mendapatkan data lokasi secara akurat menggunakan GPS. Jika GPS tidak dapat digunakan, pengguna sedang berada di gedung yang tinggi sehingga menimbulkan penurunan keakuratan, untuk itu digunakan gabungan metode Wifi dan mobile network untuk mencari posisi pengguna agar lebih efisien dan efektif.

* Aplikasi dapat melakukan pencarian lokasi sesuai dengan kategori tertentu

Biasanya lingkup pencarian lokasi kategorinya kurang beragama. Hanya terbatas pada kategori lokasi tertentu saja. Contoh yang paling banyak adalah melakukan pencarian lokasi restoran atau tempat makan saja. Hal ini dinilai masih kurang, karena kebutuhan pencarian lokasi tidak hanya sebatas tempat makan saja. Ada orang yang membutuhkan ATM atau rumah sakit terdekat dari lokasi sekarang.

Dalam keadaan yang sedang terdesak dan membutuhkan informasi yang cepat dan lengkap, maka tingkat kegunaan pencarian lokasi ini menjadi tinggi. Untuk semakin memudahkan dan mengefisienkan waktu pencarian, maka pencarian menggunakan kategori lokasi akan menghasilkan hasil yang lebih akurat dan tepat guna. Kategori lokasi yang akan dicari bersifat dinamik, karena datanya kategori dapat berubah dalam artian bertambah jumlahnya sesuai dengan data kategori pada database server. Pengguna dapat menambahkan data kategori ke database di server. Data kategori dibuat dinamik karena kategori lokasi jumlahnya akan bertambah seiring dengan bertambahnya lokasi.

Solusi pencarian dengan menggunakan kategori tertentu dinilai lebih efisien dalam hal transaksi data. Misalnya saat pengguna ingin mencari data lokasi rumah sakit terdekat dari posisi sekarang, maka data tentang restoran atau kategori lokasi lainnya tidak perlu diambil juga. Untuk mendapatkan data tersebut menggunakan koneksi internet dengan mengambil data dari server, jika mengambil data yang tidak ingin dicari membuat pencarian tidak efisien dan cenderung sia-sia. Untuk menghindari hal tersebut lingkup pencarian dibatasi per kategori yang terdefinisi pada server.

* Aplikasi dapat mencari lokasi terdekat

Setelah mendapatkan lokasi pengguna berada dan kategori lokasi yang ingin dicari, selanjutnya dapat memetakan lokasi-lokasi terdekat dari pengguna. Pada database server terdapat data koordinat lokasi yang berupa *latitude* dan *longitude*. Disimpan dalam bentuk type data double. Database yang digunakan tidak mengusung konsep spatial database. Karena setelah berbagai pertimbangan, diputuskan untuk tidak memakai spatial database.

Alasan tidak memakai spatial database dan memilih database standar, yaitu:

1. Terlalu kompleks dan rumit jika menggunakan native spatial database

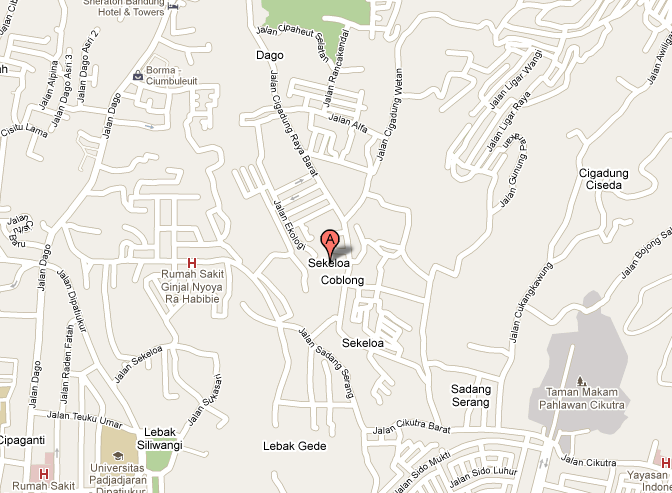
Menggunakan native spatial database menjadi sangat merepotkan dalam membangun server tersebut. Fungsi dan fitur yang ditawarkan oleh spatial database sangat lengkap, tetapi dinilai terlalu kompleks untuk spesifikasi server yang dibutuhkan dalam Tugas Akhir ini. Spatial database dinilai terlalu berlebihan jika dipakai untuk fungsionalitas yang diperlukan dalam pembangunan sistem pada Tugas Akhir ini. Karena yang dibutuhkan hanya untuk penyimpanan data latitude dan longitude, cukup disimpan dalam bentuk tipe data double.

1. Query data menggunakan sintaks SQL biasa lebih mudah dibandingkan menggunakan sintaks spatial database
2. Inti permasalahan pada Tugas Akhir ini membutuhkan solusi tingkat aplikasi mobile tidak sampai dalam pembangunan server.

Fokus dalam pemecahan masalah Tugas Akhir ini terletak dalam pengembangan aplikasi mobile. Pembangunan server hanya untuk memberikan penyedia data sehingga tidak fokus ke dalam pembangunan spatial database. Terlebih lagi query pencarian lokasi bisa menggunakan query SQL biasa dan performansinya cukup bagus.

Pencarian lokasi terdekat yang berada di sekitar lokasi pengguna berada, menggunakan query data yang dikirimkan melalui web service ke database tempat penyimpanan data lokasi. Urutan eksekusi pencarian lokasi terdekat, yaitu:

1. Mendapatkan latitude dan longitude lokasi pengguna berada
2. Mengirimkan data lokasi dan kategori pencarian melalui web service ke server
3. Dilakukan perhitungan data menggunakan query SQL untuk menentukan area terdekat dari lokasi pengguna
4. Setelah didapatkan lokasi-lokasi yang masuk ke dalam area tersebut lalu dikirimkan datanya melalui web service ke pengguna.



Gambar 3. 1 Skema Area Pencarian Lokasi

Pada gambar di atas adalah contoh lokasi pengguna berada, yaitu daerah bernama Sekeloa, yang ditandai dengan huruf A. Kemudian lingkup area untuk pencariannya adalah area dengan *outline* berwarna merah. Semua lokasi yang memiliki kategori pencarian pengguna akan diambil datanya dan dikirimkan ke pengguna.

Pembentukan area ber-outline merah itu menggunakan query SQL. Dengan menggunakan koordinat lokasi pengguna, dibuat empat buat titik dengan jarak 0.02 derajat dari posisi horizontal dan vertikal pengguna. Satu derajat itu bernilai 111.322 Km, berarti 0.02 derajat = 2,22644 Km (Mustopa, 2011). Jadi, data-data yang akan dikembalikan pada pengguna adalah semua data lokasi yang terletak radius 2.5 Km dari pengguna. Jika dalam jarak tersebut belum ditemukan lokasi yang dicari maka akan dilakukan query kembali dengan memperluas area pencarian menjadi 5 Km dan seterusnya ditambahkan jarak 2.5 Km setiap penambahan area pencarian.

* Aplikasi bisa mendapatkan detil informasi lokasi tertentu

Setelah mendapatkan daftar lokasi terdekat dari lokasi pengguna berada, daftar lokasi tersebut diurutkan berdasarkan jarak terdekat sampai terjauh dari pengguna. Pengguna dapat memilih lokasi yang dianggap sesuai dari daftar lokasi yang ada. Setelah memilih lokasi yang sesuai, maka pengguna dapat melihat detil informasi yang dimiliki oleh lokasi yang dipilih tadi.

Detil informasi yang dapat dilihat adalah nama lokasi, jarak lokasi dari pengguna, deskripsi lokasi, alamat lokasi, dan nomor telpon. Dengan informasi yang diberikan, pengguna dapat leluasa menghubungi nomor telpon pemilik tempat yang dimaksud atau langsung menuju tempat tersebut berdasarkan alamat yang tertera. Data informasi diharapkan dapat memperjelas keterangan dari lokasi tersebut.

* Aplikasi dapat menampilkan lokasi suatu tempat pada peta

Setiap lokasi yang disimpan pada database memiliki atribut latitude dan longitude yang dapat digunakan untuk menunjukkan lokasi tersebut pada peta. Setelah mendapatkan detil informasi lokasi, maka pengguna dapat juga melihat letak lokasi di peta. Tampilan peta yang digunakan adalah milik Google Map API.

Dengan menampilkan lokasi sebenarnya pada peta, diharapkan dapat memperjelas tentang lokasi tempat itu sebenarnya berada. Visualisasi pada peta semakin memudahkan pengguna untuk menemukan lokasinya. Semakin cepat menemukan lokasinya, maka semakin efisien aplikasi ini untuk keperluan yang cepat.

* Aplikasi dapat menambahkan lokasi yang dikehendaki pengguna

Konsep yang ingin diusung dalam Tugas Akhir ini adalah Web 2.0. Konsep inti dari Web 2.0 adalah content-sharing. Jadi setiap orang dapat menyumbangkan tempat yang pernah dikunjunginya ke dalam database server. Lokasi yang ditambahkan pada server dapat diakses datanya oleh pengguna lainnya. Selain dapat menambahkan lokasi, pengguna juga dapat menambahkan kategori lokasinya jika belum ada kategori yang dimaksud tersebut.

Penambahan data ini dapat membawa dampak positif, yaitu semakin banyak orang yang menyumbangkan lokasi berarti semakin banyak pemakai aplikasinya. Selain itu, menambah database lokasi yang ada tanpa perlu harus survey terpisah. Ada juga kelemahan dari penambahan data yang bersifat publik ini. Kualitas data bisa menurun, konten data tidak sesuai atau lokasinya secara *real* tidak ada.

Untuk menyumbangkan data lokasi ke database di server, pengguna diwajibkan login terlebih dahulu dengan akun khusus. Jika belum memiliki akun tersebut, pengguna dapat melakukan registrasi melalui aplikasi tersebut dan bisa langsung menggunakan akun tersebut setelah registrasi.

Data-data yang diisikan pada form penambahan lokasi berupa nama lokasi, latitude, longitude, deskripsi lokasi, alamat lokasi, dan nomor telpon, serta kelengkapan web service jika memiliki web service. Hanya saja belum tersedia halaman untuk meng-edit data lokasi yang ditambahkan ke server. Jadi saat melakukan penambahan data lokasi diharapkan data yang dimasukkan tidak ada yang salah. Kedepannya diharapkan terdapat halaman mengedit data.

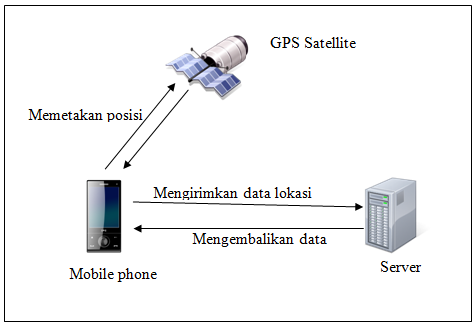
## Analisis non-Fungsional

Fitur utama dari aplikasi ini adalah pencarian lokasi terdekat dari lokasi pengguna berada, maka keakuratan data lokasi yang didapatkan sebaiknya tinggi. Data yang dikirimkan ke pengguna harus dikontrol agar data yang sampai ke pengguna adalah data yang penting saja. Hal ini sangat penting untuk efisiensi. Pertukaran data menggunakan web service harus terhubung dengan internet. Jika data yang dikirimkan ada yang sebenarnya tidak perlu dikirimkan, hanya akan menghabiskan pulsa. *User Interface* dibuat sederhana sehingga memudahkan pengguna saat menggunakannya.

## 3.3 Analisis Perangkat Lunak

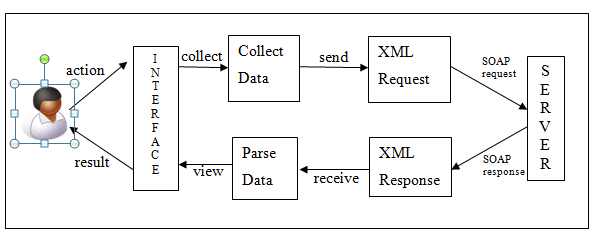
### 3.3.1 Deskripsi Umum Sistem

Aplikasi berbasis Location-based Service merupakan jawaban dari permasalah pada bab sebelumnya. Aplikasi ini dibuat dalam lingkungan sistem operasi Android. Skema sistem secara global adalah sejenis dengan client-server. Aplikasi LBS ini merupakan client yang akan berhubungan dengan server untuk mengambil dan memproses data. Pertukaran data antara aplikasi dan server melalui web service dan menggunakan jaringan internet.



Gambar 3. 2 Skema Arsitektur Umum

Pada gambar di atas merupakan skema sistem secara umum. *Mobile phone* mewakili aplikasi yang akan dibuat pada Tugas Akhir. Mobile phone berhubungan dengan GPS Satellite untuk mendapatkan posisi pengguna dalam koordinat latitude dan longitude. Setelah mendapatkan koordinat, mobile phone mengirimkan data lokasi ke server menggunakan web service yang terhubung dengan internet pada perangkat mobilenya. Setelah server menerima data tersebut lalu diproses dan menghasilkan data yang akan dikirim kembali ke perangkat mobile dan melalui web service. Data yang dikirimkan oleh server sampai pada perangkat mobile dan ditampilkan.



Gambar 3. 3 Skema Arsitektur Aplikasi

Pada gambar di atas merupakan skema aplikasi yang ada di dalam perangkat mobile. Setelah pengguna manjalankan aksi, terjadi pengumpulan data-data yang dibutuhkan untuk melaksanakan aksi tersebut. Data yang dibutuh telah terkumpul dan siap untuk dikirimkan menggunakan web service berupa XML dan melalui protokol SOAP, biasanya disebut request. Web service menghubungkan dengan server dan dilakukan perhitungan. Hasil pemrosesan oleh server dikirimkan kembali melalui SOAP dalam bentuk XML yang biasa disebut response. Data diterima oleh aplikasi. Kemudian data tersebut diparsing agar siap untuk ditampilkan pada pengguna. Selesai diparsing data ditampilkan ke pengguna.

### 3.3.2 Kebutuhan Perangkat Lunak

Kebutuhan fungsional dapat dilihat pada Tabel 3.1 di bawah ini.

|  |  |  |
| --- | --- | --- |
| **Nomor** | **Nama** | **Deskripsi** |
| SRS-F-01 | Mendapatkan lokasi pengguna | Sistem bisa mendapatkan koordinal lokasi pengguna dengan parameter latitude dan longitude |
| SRS-F-02 | Pencarian lokasi sesuai kategori | Sistem dapat melakukan pencarian lokasi sesuai dengan kategori tertentu |
| SRS-F-03 | Pencarian lokasi terdekat | Sistem dapat mencari lokasi terdekat dari lokasi pengguna berada. |
| SRS-F-04 | Mendapatkan detil informasi | Sistem bisa mendapatkan detil informasi lokasi tertentu dan menampilkannya di peta |
| SRS-F-05 | Menambahkan lokasi | Sistem dapat menambahkan lokasi tertentu yang belum ada pada database. |

Tabel 3. 1 Kebutuhan fungsional

Sesuai dengan analisis kebutuhannya pada subbab sebelumnya. Kebutuhan fungsional dapat didefinisikan menjadi lima, mendapatkan lokasi pengguna, pencarian lokasi sesuai dengan kategori, mendapatkan detil informasi, dan menambahkan lokasi. Daftar pada Tabel 3.1 di atas tersebut merupakan kebutuhan fungsional utama yang akan menjadi dasar dalam pembuatan use case.

Untuk kebutuhan non-Fungsional dapat dilihat pada Table 3.2 di bawah ini.

|  |  |  |
| --- | --- | --- |
| **Nomor** | **Nama** | **Deskripsi** |
| SRS-NF-01 | *Accuracy* | Keakuratan data harus dijaga untuk menghasilkan kualitas data yang baik. |
| SRS-NF-02 | *Interface* | Antarmuka dibuat sederhana dan mudah digunakan. |

Tabel 3. 2 Kebutuhan non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang bukan bersifat fitur dari perangkat lunak atau yang berhubungan dengan fungsionalitas utama. Kebutuhan non-fungsional dibuat untuk pelengkap yang akan menambah nilai tambah perangkat lunak. Kebutuhan non-fungsional didefinisikan menjadi dua buah, yaitu accuracy yang berhubungan dengan keakuratan data yang dihasilkan harus dijaga dengan baik. Jika tidak akurat maka pengguna tidak akan menggunakan perangkat lunak. Kebutuhan non-fungsional yang kedua adalah interface atau antarmuka. Antarmuka perangkat lunak seharusnya dibuat yang rapi dan menarik agar pengguna betah menggunakan perangkat lunak tersebut. Selain itu, antarmuka juga harus mudah digunakan dan memiliki *user experience* yang bagus dalam pemberian *feedback* ke pengguna.

### 3.3.3 Model Use Case

Pada Subbab ini akan digambarkan model *use case* yang akan menjadi dasar pembuatan aplikasi dalam Tugas Akhir. Use case ini dibuat berdasarkan fungsionalitas aplikasi yang sudah didefinisikan sebelumnya.

#### 3.3.3.1 Diagram Use Case



Gambar 3. 4 Diagram Use Case

#### 3.3.3.2 Definisi Aktor

Pada subbab sebelumnya sudah digambarkan mengenai skema use case yang akan menjadi dasar pembuatan aplikasi secara fungsional. Pada skema use case tersebut terdapat tiga aktor yang akan berinteraksi dengan sistem. Aktor yang menjalankan akitifitas fungsionalnya ada dua, yaitu *user* dan *contributor*. User adalah generalisasi dari contributor. User dapat register untuk mendapatkan akun yang berfungsi sebagai contributor.

Pada Tabel 3.3 diberikan definisi aktor yang digunakan pada diagram use case.

|  |  |  |
| --- | --- | --- |
| **No** | **Actor** | **Deskripsi** |
| AC-01 | user | User memiliki fungsional paling banyak diantara aktor lain. User dapat mencari lokasi terdekat dari data posisi keberadaannya, mendapatkan daftar kategori, mendapatkan detil informasi dari lokasi tertentu, mencari lokasi sesuai kategori tertentu, melihat letak lokasi sebenarnya pada peta, dan dapat melakukan registrasi untuk mendapatkan akun akses sebagai contributor. |
| AC-02 | contributor | Contributor dapat melakukan login pada aplikasi, setelah login contributor dapat menyumbangkan lokasi pada server, dan mengisikan detil informasi lokasi tersebut. Contributor merupakan spesialisasi dari user. |
| AC-03 | server | Server dikenai semua aksi yang dapat dilakukan oleh user dan contributor. Mulai dari pencarian lokasi terdekat per kategori dan menambahkan lokasi baru pada database. |

Tabel 3. 3 Definisi Aktor

#### 3.3.3.3 Definisi Use Case

Pada Tabel 3.4 di bawah ini dipaparkan mengenai definisi tiap use case yang ada pada diagram use case.

|  |  |  |
| --- | --- | --- |
| **No** | **Use Case** | **Deskripsi** |
| UC-01 | *Searching nearest location from current positon by category* | Sistem dapat melakukan pencarian lokasi terdekat dari perhitungan posisi user berada pada saat itu lalu mengembalikan daftar lokasi ke user. |
| UC-02 | *Gaining category list* | Sistem mendapatkan dan menampilkan daftar kategori yang bersifat dinamik tergantung data yang terdapat database server. |
| UC-03 | *Gaining place's information* | Sistem bisa mendapatkan dan menampilkan data informasi lokasi tertentu secara detil. |
| UC-04 | *Registering an account* | Sistem dapat melakukan registrasi yang merupakan layanan untuk user yang ingin berkontribus dalam penambahan lokasi pada database sehingga memerlukan akun untuk penambahan data ini. |
| UC-05 | *See the location on map* | Sistem dapat memperlihatkan lokasi tertentu pada peta. |
| UC-06 | *Contributing a location* | Sistem dapat menambahkan lokasi baru yang belum ada pada database dan menambahkan detil informasi untuk lokasi tersebut. |
| UC-07 | *Login to application* | Sistem dapat melakukan login yang berfungsi untuk melakukan penambahan lokasi dengan akun yang sebelumnya diregistrasikan terlebih dahulu. |

Tabel 3. 4 Definisi use case

#### 3.3.3.4 Skenario Use Case

Skenario use case yang dipaparkan dibawah ini adalah untuk use case *Searching nearest location from current positon by category* (UC-01). Use case yang lainnya dapat dilihat pada Lampiran A.

Pada Tabel 3.5 di bawah ini merupakan skenario use case *Searching nearest location from current positon by category* (UC-01).

|  |  |
| --- | --- |
| **Aksi Actor** | **Reaksi Sistem** |
| Skenario Normal |  |
| 1. Mengklik salah satu kategori |  |
|  | 2. Mendapatkan latitude dan longitude user |
|  | 3. Mengirimkan data koordinat dan lokasi |
|  | 4. Mengambil data lokasi |
|  | 5. Mengembalikan data lokasi |
|  | 6. Menampilkan data lokasi |

Tabel 3. 5 Skenario use case (UC-01)

### 3.3.4 Model Analisis

Model analisis yang dipaparkan hanya untuk use case *Searching nearest location from current positon by category* (UC-01). Model analisis untuk use case lain terdapat di Lampiran A.

#### 3.3.4.1 Identifikasi Kelas Analisis

Dalam identifikasi kelas analisis ini terdapat tiga jenis kelas yaitu interface, control, dan entity. Untuk use case *Searching nearest location from current positon by category* (UC-01) terdapat dua kelas interface yaitu Main dan ListPlace. Main terdapat bagian menu dan ListPlace untuk menampilkan daftar lokasi hasil pencarian. SearchPlaceControl berfungsi untuk menghubungkan kelas interface dengan kelas entity. DataPlace merupakan kelas entity yang berkaitan dengan data lokasi.

Pada Tabel 3.6 di bawah ini merupakan identifikasi kelas analisis untuk use case *Searching nearest location from current positon by category* (UC-01).

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas** | **Jenis Kelas**  **(Interface, Control, Entity)** |
| 1 | Main | Interface |
| 2 | ListPlace | Interface |
| 3 | SearchPlaceControl | Control |
| 4 | DataPlace | Entity |

Tabel 3. 6 Identifikasi Kelas Analisis UC-01

#### 3.3.4.2 Sequence Diagram

Sequence diagram dibuat berdasarkan skenario *guideline* tentang cara kerja sistem dengan iteraksi setiap aktor, kelas interface,control, dan entity.

User mengklik salah satu kategori dari daftar kategori yang ditampilkan. Sistem mendapatkan koordinat lokasi user, lalu mengirimkan ke server dan mendapatkan data lokasi terdekat sesuai dengan kategori yang dicari oleh user. Sistem menampilkan daftar lokasi.



Gambar 3. 5 Sequence Diagram UC-01

#### 3.3.4.3 Diagram Kelas Analisis

Diagram Kelas Analisis menggambarkan interaksi antar kelas yang ada pada use case. Gambar 3.6 di bawah ini adalah gambar diagram kelas analisis untuk use case *Searching nearest location from current positon by category* (UC-01).



Gambar 3. 6 Diagram Kelas Analisis UC-01

#### 3.3.4.4 Diagram Kelas Keseluruhan

Menggambarkan hubungan seluruh kelas dari seluruh use case yang ada pada tahap analisis.



Gambar 3. 7 Diagram Kelas Keseluruhan

#### 3.3.4.5 Kelas Analisis

Memaparkan semua jenis kelas yang dipakai pada model analisis ini. Terdapat tiga jenis kelas yaitu interface, control, dan entity. Daftar lengkapnya dapat dilihat pada Tabel 3.7.

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas** | **Jenis** |
| 1 | Main | Interface |
| 2 | ListPlace | Interface |
| 3 | ListPlace | Interface |
| 4 | PlaceDetail | Inteface |
| 5 | Login | Interface |
| 6 | Register | Interface |
| 7 | MapView | Interface |
| 8 | Contribute | Interface |
| 9 | SearchPlaceControl | Control |
| 10 | CategoryControl | Control |
| 11 | PlaceInfoControl | Control |
| 12 | RegisterControl | Control |
| 13 | MapControl | Control |
| 14 | ContributeControl | Control |
| 15 | LoginControl | Control |
| 16 | DataLokasi | Entity |
| 17 | DataKategori | Entity |
| 18 | DataAkun | Entity |

Tabel 3. 7 Kelas Analisis

### 3.3.6 Teknologi

Perangkat lunak yang akan dibuat ini adalah aplikasi berbasis platform Android, maka digunakan bahasa java dalam pengembangannya. Dalam *testing running* aplikasinya membutuhkan emulator. Emulator didapatkan dalam SDK-nya yang diintegrasikan dengan IDE Eclipse. Saat pembuatan aplikasi dibutuhkan library ADT android yang juga diintegrasikan dengan IDE Eclipse. Sebelum mulai pembuatan aplikasi, diharuskan men-*download* framework android yang akan digunakan misal Android 2.1 dan Google API-nya. Sebelum running aplikasi, dibuat avd untuk emulatornya. Karena menggunakan java maka dibutuhkan dukungan jre/jdk.

Teknologi yang digunakan pada pembuatan aplikasi ini adalah menggunakan *framework* Android 2.1, karena dinilai pengguna aplikasi Android masih ada yang menggunakan versi 2.1. SDK yang digunakan adalah SDK\_v3.6.1. ADT yang digunakan adalah versi 9.0.0. IDE Eclipse yang digunakan adalah Eclipse Helios. Database yang digunakan untuk server adalah MySQL. Web service yang diakses oleh aplikasi dibuat menggunakan bahasa java yang menggunakan dukungan glassfish. Akses web service oleh aplikasi menggunakan library ksoap2.

# BAB IV

# PERANCANGAN

Pada Bab ini merupakan bahasan lanjut dari analisis yang sudah dipaparkan sebelumnya. Analisis perangkat lunak yang ada dikembangkan menjadi perancangan perangkat lunak. Perancangan lebih dekat ke arah implementasi perangkat lunak. Perancangan perangkat lunak yang akan dibahas dibagi menjadi dua bagian yaitu perancangan kelas dan perancangan antarmuka.

## 4.1 Perancangan Kelas

Pada Subbab ini akan dipaparkan mengenai perancangan kelas yang dibuat berdasarkan model analisis perangkat lunak sebelumnya. Perancangan ini dibuat lebih detil dan mendekati implementasi perangkat lunak, karena dilengkapi dengan pendefinisian atribut dan method yang akan digunakan untuk tiap kelas.

Perancangan Kelas yang akan dibahas adalah use case *Searching nearest location from current positon by category* (UC-01). Perancangan kelas use case lain beserta kelengkapannya bisa dilihat pada Lampiran B.

### 4.1.1 Identifikasi Kelas Perancangan

Berdasarkan kelas analisis yang terkait sebelumnya, terdapat empat kelas analisis, yaitu kelas Main, ListPlace, SearchPlaceControl, dan DataPlace. Kelas Main dan ListPlace merupakan kelas interface yang berinteraksi langsung atau menerima aksi dari user. Kelas SearchPlaceControl merupakan kelas control yang berfungsi menghubungkan dua kelas yaitu interface dan entity agar bisa saling berinteraksi dalam alur program dan pertukaran data. Kelas DataPlace merupakan kelas entity, yang biasanya dikaitkan dengan data atau database penyimpanan.

Pada Tabel 4.1 merupakan identifikasi kelas perancangan use case *Searching nearest location from current positon by category* (UC-01).

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Nama Kelas Analisis Terkait** |
| 1 | Main | Main |
| 2 | ListPlace | ListPlace |
| 3 | Webservice | SearchPlaceControl |
| 4 | Place | DataPlace |

Tabel 4. 1 Identifikasi Kelas Perancangan UC-01

### 4.1.2 Sequence Diagram

Sequence diagram menggambarkan alur kerja sistem. Jika pada analisis, sequence diagramnya masih sangat mentah hanya alur kerja sistem secara definisi, untuk perancangan ini berbeda. Sequence diagram pada perancangan ini sudah lebih mengarah ke arah yang teknis, seperti pemakaian method untuk setiap alur kelas yang dilalui.

Saat user diperlihatkan tampilan kelas Main yang merupakan kelas interface, maka saat itu kelas Main memanggil method onCreat() sebagai inisiasi program. Kemudian untuk melakukan interasksi antar kelas interface menggunakan method intent(). Pada kelas interface ListPlace memanggil method getCoordinate() untuk mendapatkan koordinat lokasi user yang berupa latitude dan longitude. Koordinat posisi tersebut diteruskan pada Webservice yang merupakan kelas control. Oleh Webservice diteruskan ke server dengan menggunakan method searchPlace(). Server mendapatkan data yang lokasi dan dikirimkan ke Webservice. Sesampainnya di Webservice, data lokasi yang masih raw tadi diparsing agar siap ditampilkan. Setelah diparsing, data daftar lokasi ditampilkan pada kelas interface ListPlace.



Gambar 4. 1 Sequence Diagram Tahap Perancangan untuk UC-01

### 4.1.3 Diagram Kelas Perancangan

Pada Subbab ini digambarkan diagram kelas perancangan masing-masing kelas yang berperan pada use case *Searching nearest location from current positon by category* (UC-01). Setiap kelas diberikan gambaran atribut dan method yang akan digunakan saat implementasi. Atribut dan method yang ada adalah perkiraan yang dilakukan untuk mempersiapkan tahap implementasi.

Sesuai dengan identifikasi kelas perancangan pada subbab sebelumnya, kelas perancangan untuk UC-01 terdapat empat kelas, yaitu kelas interface Main dan ListPlace, kelas control Webservice, dan kelas entity Place. Kelas interface Main dan ListPlace terhubung dengan kelas control Webservice. Kelas control Webservice terhubung dengan kelas entity Place.



Gambar 4. 2 Kelas Perancangan UC-01

### 4.1.4 Kelas Perancangan

Pada Subbab ini diberikan tabel daftar semua kelas perancangan yang dipakai pada perancangan perangkat lunak dan keterkaitannya dengan kelas analisis yang sebelumnya dibuat. Pada kelas analisis setiap use case memiliki kelas control masing-masing, tetapi pada tahap perancangan ini semua control diwakili oleh satu control saja yaitu kelas Webservice. Pusat control dari aplikasi ini ada pada kelas Webservice.

Kelas Webservice ini melakukan hubungan dengan server untuk mendapatkan data yang diminta oleh kelas interface. Kemudian server akan mengeksekusi perintah dengan mengambilkan data yang dimaksud. Lalu mengirimkan data mentah tersebut pada kelas Webservice. Sesampainya data pada kelas Webservice, data diparsing agar siap ditampilkan pada kelas interface. Semua service control dijadikan satu pada kelas Webservice, untuk non-service ada satu kelas control lagi yaitu MapController, yang berfungsi untuk menampilkan map pada aplikasi.

Pada Tabel 4.2 ini dipetakan kelas perancangan dengan kelas analisis terkait yang sudah dibahas pada bab sebelumnya.

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas Perancangan** | **Nama Kelas Analisis Terkait** |
| 1 | Main | Main |
| 2 | ListPlace | ListPlace |
| 3 | Webservice | SearchPlaceControl |
| CategoryControl |
| PlaceInfoControl |
| RegisterControl |
| ContributeControl |
| LoginControl |
| 4 | Place | DataPlace |
| 5 | Category | DataKategori |
| 6 | DetailPlace | PlaceDetail |
| 7 | Login | Login |
| 8 | Register | Register |
| 9 | Account | DataAkun |
| 10 | MapView | MapView |
| 11 | MapController | MapControl |
| 12 | Contribute | Contribute |

Tabel 4. 2 Kelas Perancangan

### 4.1.5 Diagram Kelas Keseluruhan

Pada Gambar 4.3 tersebut dapat dilihat jelas bahwa kelas control Webservice menjadi pusat hampir semua control dari kelas interface dan kelas entity. Semua kelas yang digunakan untuk semua use case tergambar pada gambar tersebut lengkap dengan rancangan atribut dan method yang akan digunakan.

Diagram kelas keseluruhan dapat dilihat pada Gambar 4.3.



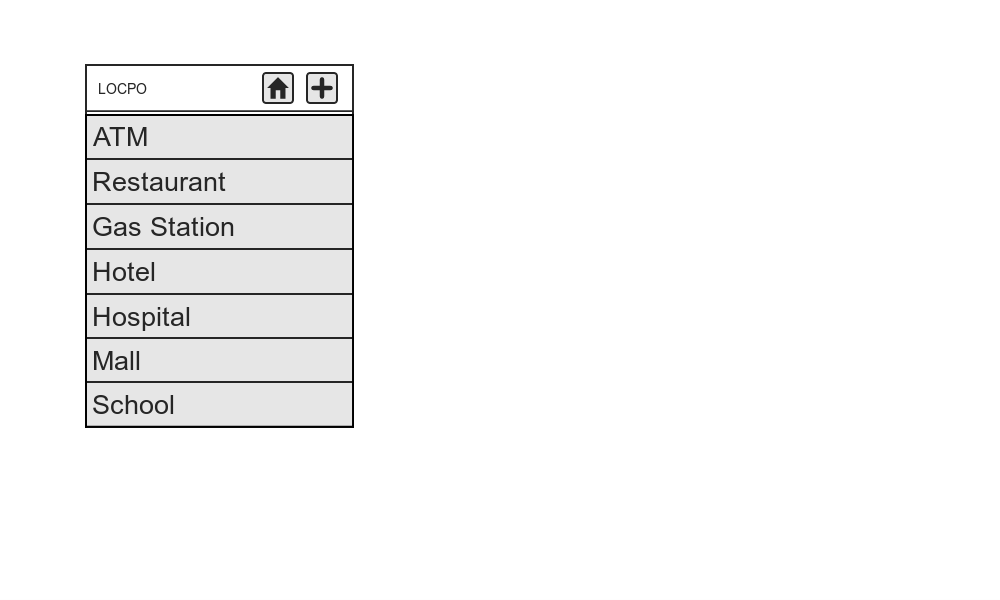
Gambar 4. 3 Diagram Kelas Keseluruhan

Semua kelas interface memiliki method onCreate(). Method onCreate ini berfungsi untuk memberikan inisialisasi layout halaman. Layout tersebut berupa file XML yang dimasukkan pada halaman interface. Setiap halaman memiliki keterkaitan dengan halaman lain. Hubungan antar halaman diatur menggunakan method intent() yang berfungsi untuk link dengan halaman lainnya. Terkadang setiap perpindahan halaman ingin disertai perpindahan data yang ada pada halaman sebelumnya untuk melengkapi hal ini dapat menggunakan method bundle(). Pada kelas control Webservice terjadi pengiriman dan peneriman data. Data yang dikirimkan berupa string atau integer, sedangkan data yang diterima berupa string. Raw string data yang diterima harus diolah dulu sebelum digunakan menggunakan parser yang terdapat pada kelas Webservice.

## 4.2 Perancangan Antarmuka

Pada bagian ini akan dipaparkan perancangan antarmuka yang nantinya akan digunakan pada tahap implementasi perangkat lunak. Perancangan antarmuka pada subbab ini masih tergolong *low fidelity* tetapi cukup menggambarkan fungsionalitas perangkat lunak.

### 4.2.1 Halaman Main



Gambar 4. 4 Antarmuka Main

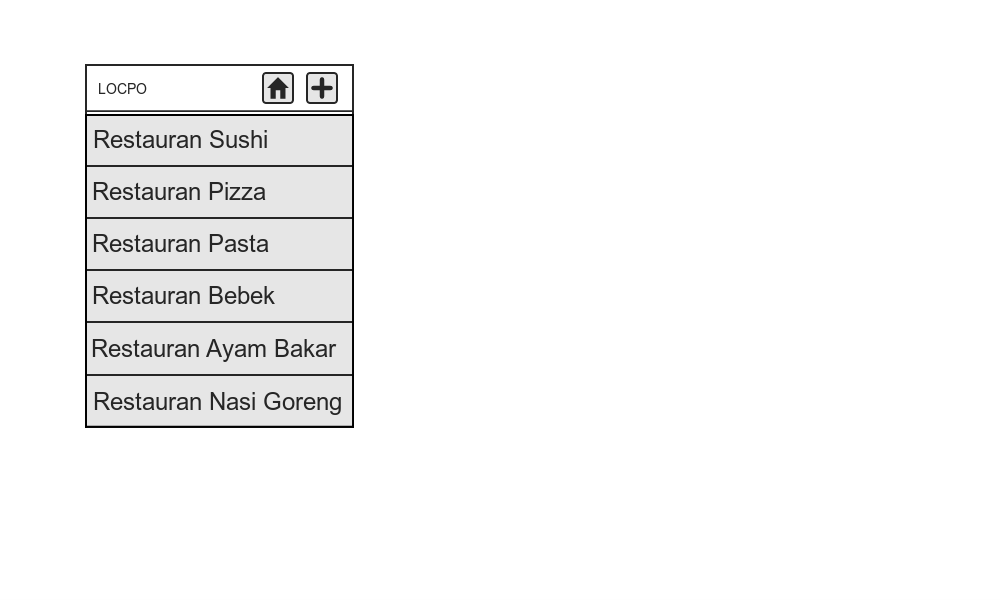
Pada Gambar 4.4 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Daftar kategori lokasi berupa ListView yang diambil dari server datanya dan data bersifat dinamik. Setiap item yang ada pada ListView tersebut dapat diklik dan terhubung dengan halaman ListPlace dan menampilkan hasil pencarian.

Pada Tabel 4.3 di bawah ini merupakan definisi antarmuka Main.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| lvCategory | ListView | Category | Listview berisi daftar nama kategori dan ditampilkan pada halaman Main |

Tabel 4. 3 Deskripsi Antarmuka Main

### 4.2.2 Halaman ListPlace



Gambar 4. 5 Antarmuka ListPlace

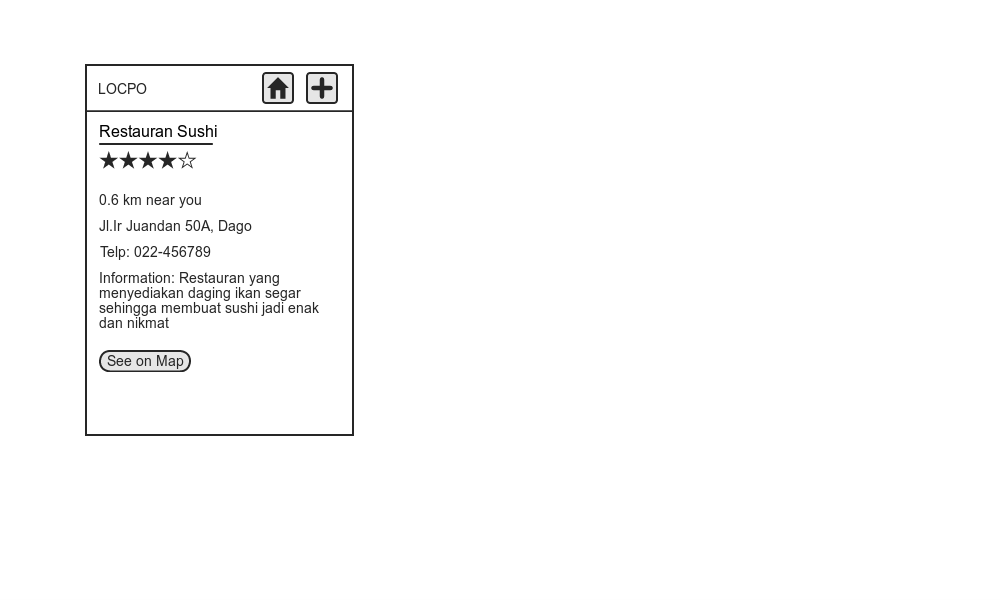
Pada Gambar 4.5 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Pada antarmuka ListPlace ini menampilkan daftar hasil pencarian lokasi terdekat dari pengguna per kategori yang dipilih. Kategori yang dipilih dari halaman Main adalah kategori Restauran, maka pada halaman ListPlace ditampilkan daftar Restauran yang terdekat dari pengguna. Daftar restauran tersebut dibuat menggunakan ListView. Setiap item dapat diklik dan terhubung dengan halaman DetailPlace.

Pada Tabel 4.4 di bawah ini merupakan definisi antarmuka ListPlace.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| lvPlace | ListView | ListPlace | Listview berisi daftar lokasi hasil pencarian |

Tabel 4. 4 Deskripsi Antarmuka ListPlace

### 4.2.3 Halaman DetailPlace



Gambar 4. 6 Antarmuka DetailPlace

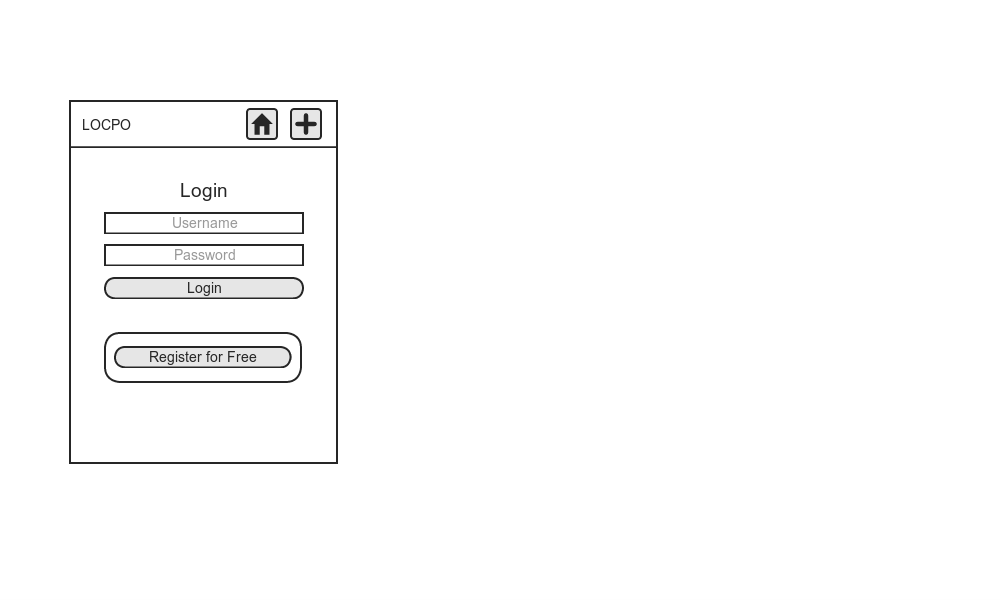
Pada Gambar 4.6 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Pada halaman DetailPlace ini ditampilkan daftar informasi yang terkait dengan suatu lokasi. Informasi lokasi yang ditampilkan tergantung lokasi yang dipilih pada halaman ListPlace. Informasi ditampilkan menggunakan ListView agar mudah pengorganisasiannya dan rapi. Terdapat tombol “see on map” yang menuju ke halaman MapView untuk menampilkan lokasi pada peta.

Pada Tabel 4.5 di bawah ini merupakan definisi antarmuka DetailPlace.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| lvPlaceDetail | ListView | DetailPlace | Listview berisi daftar lokasi hasil pencarian |
| placeRating | Rating | PlaceRating | Tombol yang digunakan untuk merating lokasi |
| bMap | Button | See on Map | Tombol yang menghubungkan dengan halaman MapView |

Tabel 4. 5 Dekripsi Antarmuka DetailPlace

### 4.2.4 Halaman Login



Gambar 4. 7 Antarmuka Login

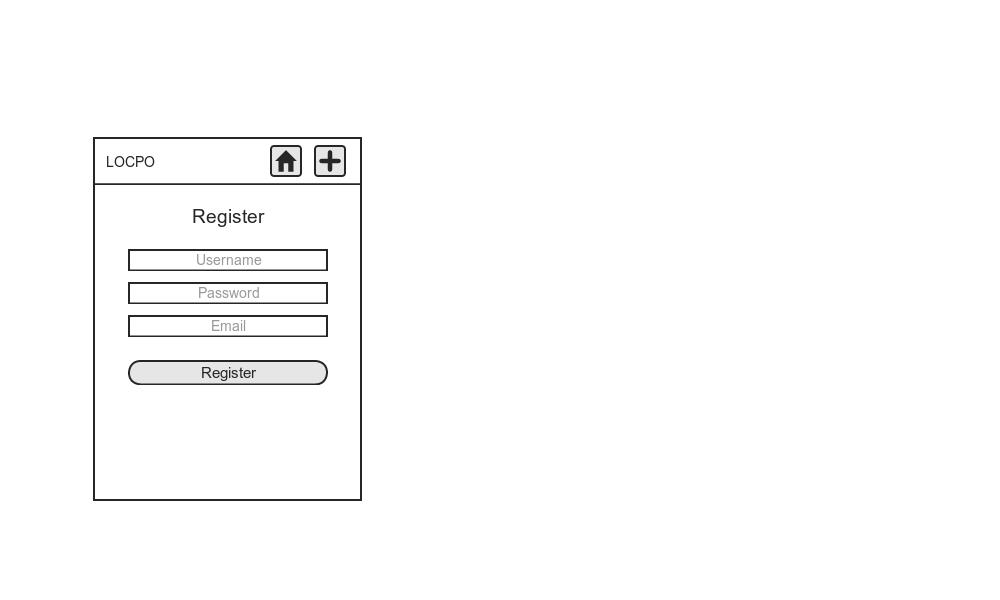
Pada Gambar 4.7 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Pada halaman Login ini terdapat field untuk mengisikan username dan password pengguna yang akan login. Tombol login akan megirimkan data username dan password lalu kan membawa pengguna ke halaman Contribute jika sukses login dan memberikan alert jika autentifikasi gagal. Tombol regiter menghubungkan halaman register.

Pada Tabel 4.6 di bawah ini merupakan definisi antarmuka Login.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| txtUsername | EditText | Username | Field yang diisikan username untuk login |
| txtPassword | EditText | Password | Field yang diisikan password untuk login |
| bLogin | Button | Login | Tombol yang digunakan untuk mengirimkan data username dan password |
| bRegister | Button | Register For Free | Tombol yang menghubungkan dengan halaman Register |

Tabel 4. 6 Deskripsi Antarmuka Login

### 4.2.5 Halaman Register



Gambar 4. 8 Antarmuka Register

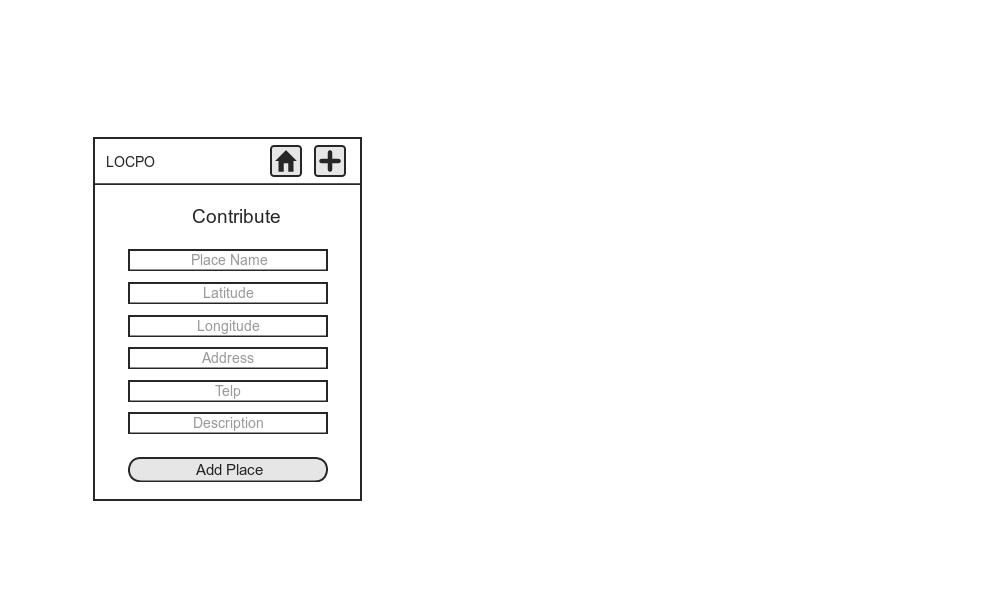
Pada Gambar 4.8 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Pada halaman Register ini pengguna dapat melakukan pendaftaran untuk mendapatkan akun. Akun ini berfungsi untuk login dan pengguna bisa berkontribusi melakukan penambahan lokasi. Tombol register mengirimkan data dan jika berhasil proses registrasinya akan pesan berhasil serta sebaliknya.

Pada Tabel 4.7 di bawah ini merupakan definisi antarmuka Register.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| txtUsername | EditText | Username | Field yang diisi username untuk register |
| txtPassword | EditText | Password | Field yang diisi password untuk register |
| txtEmail | EditText | Email | Field yang diisi alamat email untuk register |
| bRegister | Button | Register For Free | Tombol yang digunakan untuk mengirimkan data registrasi ke server |

Tabel 4. 7 Deskripsi Antarmuka Register

### 4.2.6 Halaman Contribute



Gambar 4. 9 Antarmuka Contribute

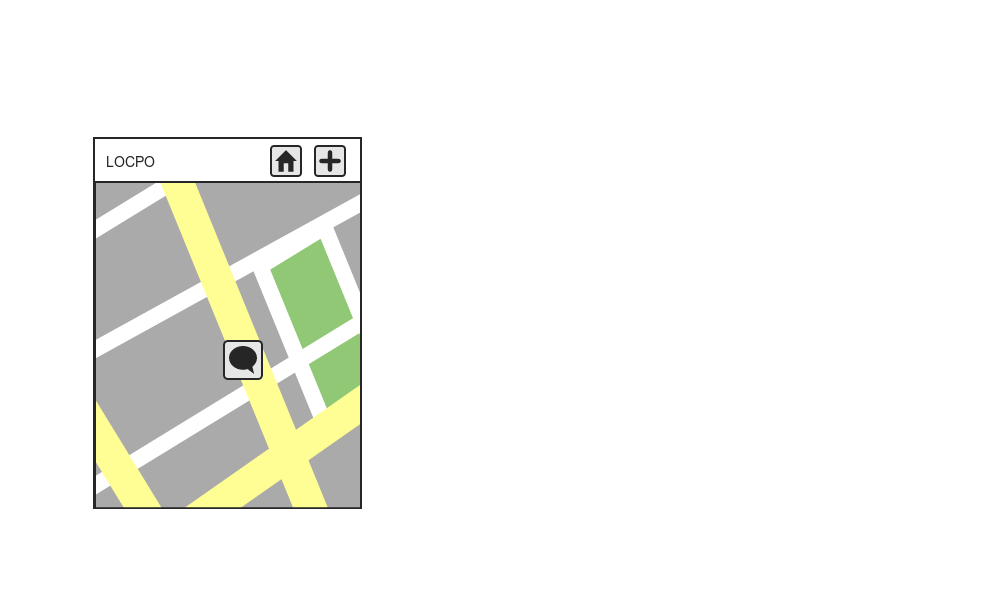
Pada Gambar 4.9 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Pada halam Contribute ini terdapat field Place Name, Latitude, Longitude, Address, Telp, dan Description yang merupakan detil informasi untuk suatu lokasi. Terdapat tombol add yang akan mengirimkan data, jika penambahan berhasil dilakukan akan menhasilkan alert berhasil dan sebaliknya.

Pada Tabel 4.8 di bawah ini merupakan definisi antarmuka Contribute.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| txtPlaceName | EditText | Place Name | Field yang berisi nama lokasi |
| txtLatitude | EditText | Latitude | Field yang berisi nilai latitude lokasi |
| txtLongitude | EditText | Longitude | Field yang berisi nilai longitude lokasi |
| txtAddress | EditText | Address | Field yang berisi alamat lokasi |
| txtTelp | EditText | Telp | Field yang berisi nomor telpon |
| txtDesc | EditText | Description | Field yang berisi deskripsi lokasi |
| bAddPlace | Button | Add Place | Tombol yang digunakan untuk mengirimkan data penambahan lokasi ke server |

Tabel 4. 8 Deskripsi Antarmuka Contribute

### 4.2.7 Halaman MapView



Gambar 4. 10 Antarmuka MapView

Pada Gambar 4.10 di atas digambarkan perancangan antarmuka Main dengan header yang terdapat tombol home dan add. Tombol home untuk kembali ke halaman main dan tombol add untuk menambahkan lokasi yang merupakan kontribusi pengguna menggunakan aplikasi ini. Pada halaman MapView ditampilkan lokasi pada peta. Peta yang digunakan adalah Google Map API.

Pada Tabel 4.9 di bawah ini merupakan definisi antarmuka MapView.

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| bHome | Button | Home | Tombol yang menghubungkan dengan halaman Main |
| bAdd | Button | Add | Tombol uang menghubungkan dengan halaman Login |
| locpoMap | Map | Map | Menampilkan peta lokasi |
| pin | Image | Pin location | Menandai lokasi berada pada peta |

Tabel 4. 9 Deskripsi Antarmuka MapView

# DAFTAR REFERENSI

Android Developers.[http://developer.android.com](http://developer.android.com/) (diakses tanggal 7 November 2010)

Brimicombe, A., & Li, C.(2009).*Location Based Service and Geo-Information Engineering*.West Sussex:Wiley-Blackwell.

Cybertrain.<http://cybertrain.net76.net/index.php?option=com_content&task=view&id=23&Itemid=40> (diakses tanggal 5 November 2010)

Kupper, Axel.(2005).*Location-Based Services (Fundamentals and Operation)*.West Sussex:Wiley.

W3C, Web Service Definition Language (WSDL).<http://www.w3.org/TR/wsdl>(Diakses tanggal 7 November 2010)

W3School. Web Service.<http://www.w3schools.com/webservices/default.asp>

(diakses tanggal 7 November 2010)

O’Reilly.(2005). http://oreilly.com/web2/archive/what-is-web-20.html

(diakses tanggal 17 April 2011)

Evans, D.(2008). <http://www.techradar.com/blogs/article/how-the-iphone-s-wi-fi-location-works-201138> (diakses tanggal 17 April 2011)

Map-GPS-Info.(2011). <http://www.maps-gps-info.com/gps-accuracy.html>

(diakses tanggal 17 April 2011)

Mustopa.(2011). <http://sistiminformasigeografi.blogspot.com/2011/03/menghitung-jarak-panjang-pada-peta.html> (diakses tanggal 17 April)