

Aplikasi *Find Kuliner Nusantara* Berbasis *Google Map* dan *Android Mobile*

Annas Setyo Pratomo¹ dan Aris Puji Widodo²

Departemen Ilmu Komputer/Informatika, Fakultas Sains dan Matematika
Universitas Diponegoro
¹annaspratama7@gmail.com, ²arispuji@gmail.com

Wealth Indonesia culinary is a diversity of culture and archipelago tradition. Besides original culinary, also many culinary influences from India, China, Middle East, and Europe. Each region in Indonesia has typical culinary. But tourists always having trouble for getting culinary recommendations in the visited region. Mobile technology has developed quite rapidly and the culinary recommendation can be easily accessed. One of the most popular mobile operating system for mobile smartphone is android. By using Location Based Services (LBS), users can obtain location an information about any location according to their need. LBS is a combination of location services technology of mobile device, communication network, positioning component, service and application provider, and data and content provider. Find Kuliner Nusantara Application is the solution to know the culinary information based on the shortest distance from the user using the haversine formula to calculate the distance, the latest time, the most popular rating of the average rating given by the member and street view display of culinary locations. This application was developed using the process model of the Unified Process (UP) with client-server architecture. After it being developed, then it was tested by using blackbox and performance testing. This application used GPS technology to determine the user's position and cellular positioning technology if the GPS service is not available. It was able to serve no more than 200 users simultaneously. The distance calculation of this application is almost similar on the site www.movable-type.co.uk with the difference of 0,009 km and on site www.andrew.hedges.name/experiments/haversine with the difference of 0,03 km. The final result in this Tugas Akhir is the Find Kuliner Nusantara application that allowed users or travelers to find the nearest culinary locations of the user in the visited region.

Keywords: *Culinary, Location Based Service, Unified Process, Haversine Formula, Google Map, Android*

Abstrak

Kekayaan kuliner Indonesia merupakan cermin keberagaman budaya dan tradisi nusantara. Selain kuliner asli, banyak pula pengaruh kuliner dari India, China, Timur Tengah, dan Eropa. Setiap daerah di Indonesia memiliki kuliner khasnya masing-masing. Namun wisatawan selalu kesulitan untuk mendapatkan rekomendasi kuliner didaerah yang dikunjungi. Teknologi *mobile* mengalami perkembangan yang cukup pesat, kebutuhan rekomendasi kuliner dapat diakses dengan mudah. Salah satu sistem operasi *mobile smartphone* terpopuler yaitu *android*. Dengan memanfaatkan *Location Based Service* (LBS), pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya. LBS merupakan teknologi layanan lokasi perpaduan antara *mobile device, communication network, positioning component, service and application provider*, dan *data and content provider*. Aplikasi *Find Kuliner Nusantara* merupakan solusi untuk mengetahui informasi kuliner berdasarkan jarak terdekat dari *user* menggunakan *haversine formula* untuk menghitung jarak, waktu terbaru, *rating* terpopuler dari rata-rata *rating* yang diberikan oleh *member* dan tampilan *streetview* dari lokasi kuliner. Aplikasi ini dikembangkan menggunakan model proses *Unified Process* (UP) dengan arsitekur *client-server*. Setelah

selesai dikembangkan, kemudian diuji dengan menggunakan pengujian *blackbox* dan *performance*. Aplikasi ini menggunakan teknologi GPS untuk menentukan posisi *user* dan teknologi *cellular positioning* jika layanan GPS tidak tersedia. Aplikasi tersebut sanggup melayani tidak lebih dari 200 user secara bersamaan. Perhitungan jarak pada aplikasi ini hampir sama pada situs *www.movable-type.co.uk* dengan selisih 0.009 km dan pada situs *www.andrew.hedges.name/experiments/haversine* dengan selisih 0.03 km. Hasil akhir pada Tugas Akhir ini adalah aplikasi *Find Kuliner Nusantara* yang memudahkan *user* atau wisatawan untuk mencari kuliner terdekat dari lokasi *user* di daerah yang dikunjungi.

Kata Kunci: *Kuliner, Location Based Service, Unified Process, Haversine Formula, Google Map, Android*

1. Pendahuluan

Di era otonomi daerah, sektor pariwisata memegang peran penting dalam perekonomian daerah di Indonesia. Pariwisata mampu meningkatkan pemasukan devisa negara, peningkatan pendapatan masyarakat, dan untuk meningkatkan Pendapatan Asli Daerah (PAD) di berbagai kabupaten maupun provinsi di Indonesia. Salah satu daya tarik wisata bagi wisatawan baik dari mancanegara maupun domestik adalah wisata kuliner, karena makan dan minum merupakan kegiatan primer yang mana tidak bisa dilepaskan dari kehidupan manusia.

Di Indonesia, saat ini banyak sekali dikembangkan tempat makan diantaranya warung, restoran, kafe yang menyajikan berbagai jenis makanan dan minuman mulai dari kuliner lokal dari berbagai daerah di Indonesia, sampai kuliner dari mancanegara seperti Eropa, China, Jepang, dan Amerika Serikat. Masalah yang sering muncul adalah wisatawan kesulitan mencari lokasi kuliner apalagi jika tidak ada rekomendasi kuliner di daerah yang dikunjungi oleh wisatawan. Dengan perkembangan teknologi dan informasi yang begitu pesat melalui penggunaan internet dan *mobile smartphone* mempunyai pengaruh yang sangat bagus untuk industri pariwisata. Perpaduan antara internet dan teknologi *Global Positioning System* (GPS) yang dikembangkan memberikan kemudahan para wisatawan untuk mencari tempat-tempat wisata menarik [1].

Location Based Service (LBS) memanfaatkan teknologi GPS dalam pengaplikasiannya. Selain dapat mengetahui posisi pengguna, aplikasi LBS juga dapat menentukan posisi tempat-tempat tertentu. Dan dengan kombinasi ini, aplikasi LBS

akan mencari rute untuk menghubungkan posisi pengguna dengan suatu tempat.

Salah satu sistem operasi untuk *mobile smartphone* yaitu sistem operasi *android*. *android* merupakan sistem operasi yang *open source*. Salah satu keuntungan dari sistem operasi *open source* adalah aplikasi pihak ketiga dapat mengakses seluruh *resource* yang dimiliki *smartphone* tersebut, tanpa membedakannya dengan aplikasi inti dari *smartphone*. *Android* juga memudahkan seorang *developer* dalam mengembangkan aplikasi seperti aplikasi LBS, karena *android* menyediakan akses dan integrasi dengan layanan *google maps* [2].

Beberapa penelitian mengenai aplikasi LBS dengan menggunakan *haversine formula* telah dilakukan sebelumnya, penelitian pertama menghasilkan aplikasi *Chruch Map* yang mampu menampilkan posisi *user* saat ini, gereja-gereja dari posisi *user* berada dan menampilkan rute gereja dengan menggunakan teknologi LBS pada platform *android* dan terhubung dengan MySQL melalui perantara Modul *JSON Parsing* [3]. Penelitian kedua menghasilkan aplikasi untuk mencari lokasi tambal ban dengan memperkirakan jarak antara pengguna dengan lokasi tambal ban dengan *haversine formula* [4]. Kemudian penelitian ketiga menghasilkan aplikasi pencari masjid dan mushola dalam radius 3 kilometer dari lokasi *user* menggunakan *haversine formula* [5].

Dengan memanfaatkan teknologi LBS dan penelitian-penelitian yang ada, maka akan dibangun sebuah aplikasi untuk mencari lokasi kuliner berdasarkan jarak yang terdekat dengan *user* di daerah yang dikunjungi. Aplikasi ini diharapkan memudahkan wisatawan domestik maupun mancanegara dapat memperoleh

informasi tentang rekomendasi kuliner berdasarkan jarak terdekat, waktu, dan popularitas kuliner, peta lokasi kuliner, mengusulkan lokasi kuliner baru dengan persetujuan *admin* dan navigasi menuju lokasi kuliner.

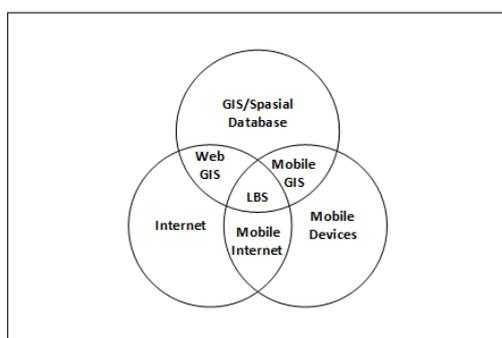
2. Tinjauan Pustaka

Pada bab ini menyajikan teori yang digunakan dalam penelitian yaitu LBS, GPS, *haversine formula* dan UP.

2.1. LBS

LBS merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui tersebut. Dengan berkembangnya teknologi GSM, maka LBS menjadi semakin mudah dan murah, bahkan untuk individu sekalipun [6].

LBS memberikan kemungkinan komunikasi dan interaksi dua arah. Oleh karena itu pengguna memberitahu penyedia layanan untuk mendapatkan informasi yang dia butuhkan, dengan referensi posisi pengguna tersebut. Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : *Geographic Information System*, *Internet Service*, dan *Mobile Devices*, hal ini dapat dilihat pada gambar LBS adalah pertemuan dari tiga teknologi [7].



Gambar 2.1 LBS Sebagai Simpang Teknologi

LBS adalah aplikasi yang bergantung pada pada lokasi tertentu dan didefinisikan pula sebagai layanan informasi dengan memanfaatkan teknologi untuk mengetahui posisi sesuatu. Layanan berbasis lokasi menggunakan teknologi *Positioning System*, teknologi ini memungkinkan

para pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya. LBS termasuk dalam kategori teknologi yang sama dengan *Geographic Information System (GIS)* dan aplikasi GPS, yaitu dikenal dengan teknologi *geospasial*. Teknologi ini terdiri atas perangkat untuk mengumpulkan, menyimpan, menganalisa dan mendistribusikan data yang sesuai dengan kebutuhan pengguna terhadap sistem koordinat bumi. Layanan ini menjadi sangat penting bagi penggunanya karena mampu menghubungkan antara lokasi *geographic* informasi terhadap lokasi penggunanya, hal ini sangat mendukung era mobilitas seperti pada masa ini. Teknologi LBS ini terdiri atas perangkat-perangkat yang digunakan untuk mengumpulkan, menyimpan, menganalisa dan mendistribusikan data dan informasi pada berdasarkan sistem koordinat *geographic* bumi secara *real time*. Identifikasi koordinat pengguna memungkinkan aplikasi LBS untuk menyediakan layanan bagi pengguna perangkat *mobile* [6].

Dalam layanan berbasis lokasi terdapat lima komponen penting yaitu meliputi:

1. *Mobile Devices*

Suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan teks.

2. *Communication Network*

Komponen kedua adalah jaringan komunikasi yang mengirim data pengguna dan informasi yang diminta dari *mobile terminal* ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan seluler (GSM, CDMA), *Wireless Local Area Network (WLAN)*, atau *Wireless Wide Area Network (WWAN)*.

3. *Positioning Component*

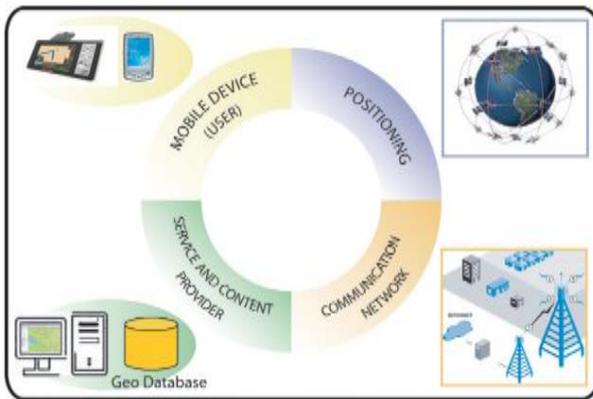
Untuk memproses suatu layanan maka posisi pengguna harus diketahui. Untuk mengetahuinya dengan alat bernama GPS.

4. *Service and Application Provider*

Penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.

5. *Data and Content Provider*

Penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, data dapat diminta dari data dan *content provider*.



Gambar 2.2 Komponen Dasar LBS

Selanjutnya *Service and Application Provider* mengirim informasi yang telah diolah melalui jaringan internet dan jaringan komunikasi. Pada akhirnya pengguna dapat menerima informasi yang diinginkan.

2.2. GPS

Perkembangan teknologi navigasi yang pesat saat ini menghasilkan suatu sistem navigasi yang sangat canggih yang dapat digunakan untuk mengetahui posisi suatu objek di permukaan bumi. Sistem ini dikenal dengan nama GPS, secara garis besar GPS dibagi menjadi 3 bagian, yaitu:

1. *Space Segment*

Bagian dari GPS yang terdiri dari beberapa satelit yang mengorbit di sekeliling bumi.

2. *Control Segment*

Beberapa stasiun yang berada di bumi dan bertugas untuk mengontrol dan melakukan koreksi terhadap orbit satelit.

3. *User Segment*

Perangkat *GPS receiver* yang dijual di pasaran dan berfungsi untuk menerima data dari satelit, menerjemahkannya ke dalam satuan posisi koordinat bumi.

Dengan menggunakan GPS yang terintegrasi dengan sistem telekomunikasi, maka suatu objek dapat dipantau keberadaannya secara *continue* dan data-data posisi objek tersebut dapat disimpan ke dalam suatu memori. Selain dapat

digunakan sebagai *data logging*, GPS juga dapat digunakan untuk melakukan pemantauan secara berkala terhadap objek yang bergerak sehingga keberadaan objek tersebut dapat diketahui dengan mudah di mana posisinya saat ini [8].

GPS tidak selalu akurat di semua lingkungan apalagi di dalam ruangan yang dipenuhi bangunan. Dengan demikian, dibutuhkan teknik untuk menentukan lokasi *user* yang tidak tergantung pada satelit GPS. Salah satu teknik yang telah digunakan adalah dengan menggunakan *Cellular Positioning*. *Cellular Positioning* atau *Cell-ID* merupakan teknologi untuk menemukan posisi lokasi *user* dengan menggunakan *Base Transceiver Station* (BTS). Keakuratan dengan teknik tersebut tergantung pada ukuran *cell*. Dalam banyak kasus menjadi kurang karena diameter dari *Global System for Mobile Communication* (GSM) *cell* antara 2-20 km [9].

2.3. *Haversine Formula*

Haversine formula adalah persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan *latitude* dan *longitude*. *Haversine formula* memberikan jarak minimum antara dua titik pada tubuh bola dengan menggunakan *latitude* dan *longitude* [10].

$$d = 2r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2 \left(\frac{\delta_2 - \delta_1}{2} \right)} \right)$$

Keterangan:

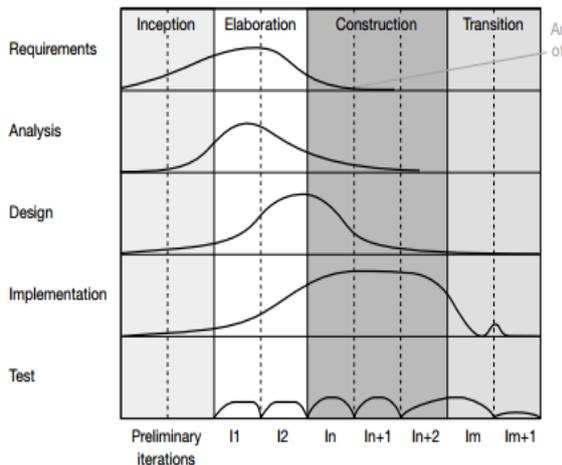
- d = jarak (km)
- δ_2 = *longitude* destinasi
- ϕ_1 = *latitude* asal
- r = jari-jari bumi
- ϕ_2 = *latitude* destinasi
- δ_1 = *longitude* asal

Sudut pada rumus menggunakan radian dengan 1 derajat adalah 0.017453 radian. Penggunaan *haversine formula* untuk melakukan perhitungan jarak antara lokasi *user* dengan lokasi kuliner, kemudian ditentukan lokasi terdekat sampai terjauh dengan *user* dengan radius 10 kilometer. Selain itu, *haversine formula* juga digunakan untuk membatasi jumlah lokasi

kuliner didalam radius 10 kilometer didalam peta kuliner.

2.4. UP

Unified Software Development Process (USDP) adalah suatu standar industri SEP dari UML. USDP disebut sebagai *Unified Process*. UML dimaksudkan untuk menyediakan bahasa visual dan proses rekayasa perangkat lunak. UP didasarkan pada proses kerja yang dilakukan di *Ericsson* (*The Ericsson Approach*, 1967), di *Rational* (*The Rational Objectory Process*, 1996-1997) dan sumber-sumber lain dari praktik terbaik. Dengan demikian, UP adalah metode pragmatis dan diuji untuk mengembangkan perangkat lunak yang menggabungkan praktik terbaik dari pendahulunya [11].



Gambar 2.3 Model *Unified Process*

Lifecycle project didalam UP dibagi menjadi empat fase yang masing-masing berakhir dengan tonggak utama. Dalam setiap tahap dapat memiliki satu atau lebih iterasi, dan setiap iterasi menjalankan lima *core workflows* dan setiap *workflows* tambahan. Jumlah pasti iterasi per fase tergantung pada ukuran proyek:

1. *Inception phase* merupakan fase dengan penekanan utama dalam *inception* adalah pada *requirements* dan *analysis workflow*.
2. *Elaboration phase* merupakan fase yang berfokus untuk melengkapi *requirements*, serta *analysis* dan *design workflows*, dengan *implementation* juga dilakukan tetapi hanya sedikit.
3. *Construction phase* merupakan fase yang berfokus pada *implementation workflow*.

Dilakukan tambahan *design* dan dilakukan sedikit *test*.

4. *Transition phase* merupakan fase dengan penekanan *implementation* dan *test workflow*.

Didalam UP terdapat lima *core workflows*, yaitu:

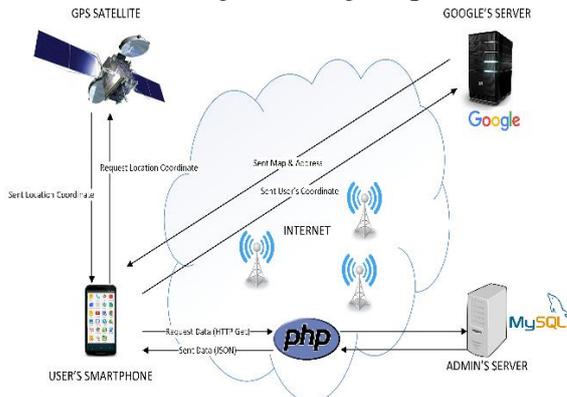
1. *Requirement*
Requirement merupakan dasar dari sistem. Kebanyakan pekerjaan pada *requirement workflow* terjadi sepanjang fase *inception* dan *elaboration* saat awal *project lifecycle*.
2. *Analysis*
Analysis berfungsi untuk menyempurnakan dan menstrukturkan *requirements*. Pekerjaan utama dalam *analysis* dimulai menjelang akhir fase *inception* dan merupakan fokus utama dari fase *elaboration*.
3. *Design*
Design adalah merealisasikan *requirements* ke dalam arsitektur sistem. *Design workflow* adalah aktivitas utama selama bagian akhir dari fase *elaboration* dan paruh pertama fase *construction*.
4. *Implementation*
Implementation adalah tahapan dimana mengubah *design model* ke dalam bentuk *code*.
5. *Test*
Test yaitu dilakukan pengujian yang bertujuan untuk menemukan *bug* sistem sehingga dapat segera dilakukan perbaikan terhadap sistem.

3. Requirement

Requirement terjadi pada *inception* dan *elaboration phase*. *Requirement* menentukan proses bisnis dari aplikasi *Find Kuliner Nusantara*. Kemudian memaparkan *functional* dan *non-functional requirement*. Pada pengembangan aplikasi *Find Kuliner Nusantara* hanya dibahas untuk aplikasi utama, yaitu aplikasi *android*.

Aplikasi *Find Kuliner Nusantara* dapat digunakan *user* di daerah yang dikunjungi. Aplikasi akan menampilkan lokasi kuliner jika didalam radius jarak lokasi *user* terdapat data lokasi kuliner. Jika tidak terdapat data lokasi kuliner didalam radius jarak lokasi *user* maka aplikasi menampilkan *message* bahwa data lokasi kuliner tidak ada disekitar *user*. Untuk

menentukan lokasi *user*, aplikasi menggunakan teknologi GPS dan teknologi *Cellular Positioning* jika layanan GPS tidak tersedia. Salah satu keunggulan aplikasi ini adalah *user* dapat melihat *streetview* dari lokasi kuliner disekitar *user*. Aplikasi tersebut terdiri dari *client-side* berbasis *android* yang digunakan oleh *user* untuk mencari lokasi kuliner terdekat dan *server-side* berbasis web yang digunakan oleh *admin* untuk menyetujui atau menolak lokasi kuliner baru atas usulan dari *user* yang sudah mendaftar (*member*). Dalam penerapan perhitungan dan penentuan batas radius jarak, aplikasi ini menggunakan perhitungan dengan *haversine formula*. Batas radius yang digunakan adalah 10 kilometer. Gambaran arsitektur aplikasi *Find Kuliner Nusantara* dapat dilihat pada gambar 3.1.



Gambar 3.1 Arsitektur Sistem

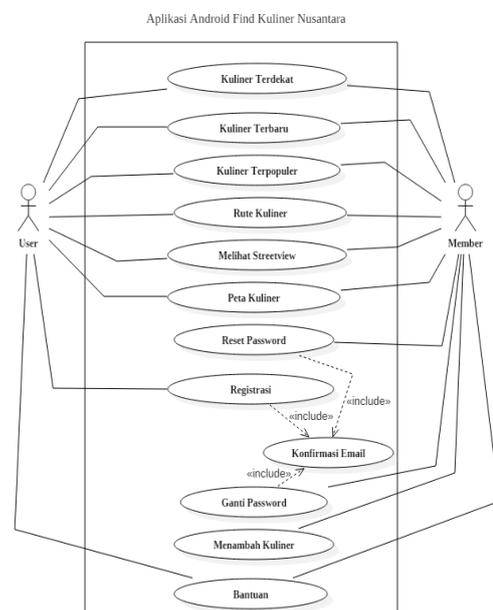
Dari analisa observasi terhadap aplikasi *android* pencarian kuliner terdekat dari *user* yang terpopuler dengan *rating* antara 4 sampai dengan 5 didapat *requirement* aplikasi *Find Kuliner Nusantara* sebagai berikut:

1. Menampilkan kuliner berdasarkan jarak terdekat.
2. Menampilkan kuliner terdekat berdasarkan waktu terbaru persetujuan *admin*.
3. Menampilkan kuliner terdekat berdasarkan rata-rata *rating* populer dari *member*.
4. Memberikan rute dan navigasi menuju lokasi kuliner.
5. Menampilkan peta kuliner terdekat.
6. Menampilkan *streetview*.
7. Melakukan registrasi.
8. Melakukan *reset password*.
9. Melakukan ganti *password*.
10. Menambah lokasi kuliner baru.
11. Menampilkan bantuan.

Sedangkan *non-functional requirement* aplikasi *Find Kuliner Nusantara* sebagai berikut:

1. Aplikasi harus mampu memberikan permintaan layanan sekurang-kurangnya 100 *user* dalam satu waktu.
2. *Password member* harus terenkripsi.

Setelah mendefinisikan *requirement* kemudian memodelkan *requirement* tersebut pada *use case diagram*. *Use case diagram android* dapat dilihat pada gambar 3.2



Gambar 3.2 Use Case Diagram Aplikasi Android

4. Analysis

Analysis terjadi pada akhir *inception phase* dan berfokus pada *elaboration phase*. Pada *workflow* tersebut menghasilkan *analysis classes*, *use case realizations* pada tahap *analysis workflow*, dan *sequence diagram*.

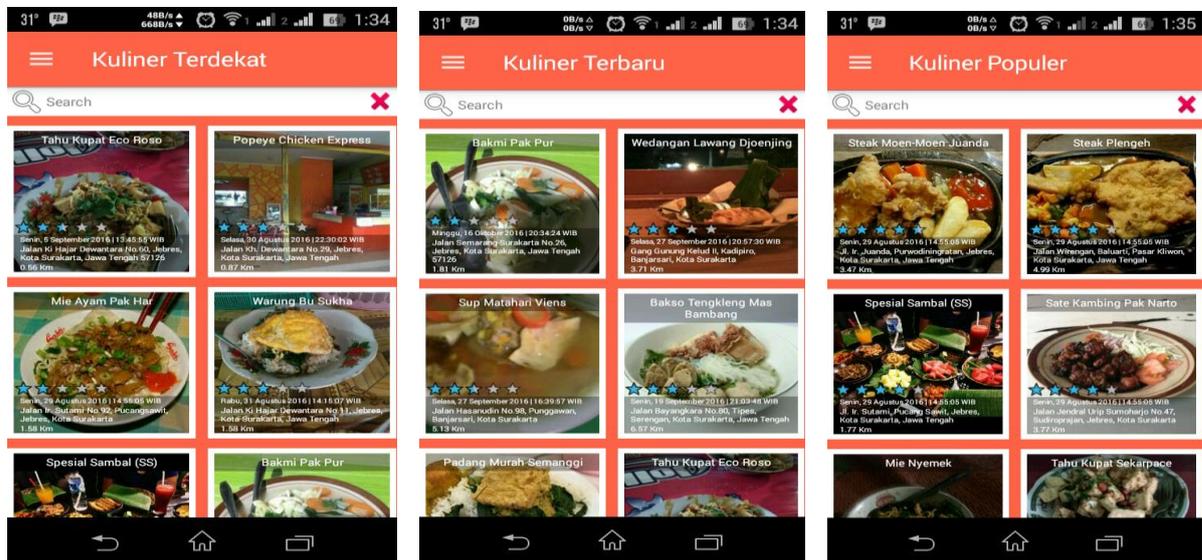
Pada tahap *analysis classes* menjelaskan tentang identifikasi *class* dari masing-masing *use case* yang telah didefinisikan sebelumnya seperti *boundary*, *control*, dan *entity* dan bagaimana saling berinteraksi. *Analysis class* pada salah satu *use case*, yaitu *use case* menampilkan kuliner terdekat dapat dilihat pada gambar 4.1.

6. Implementation

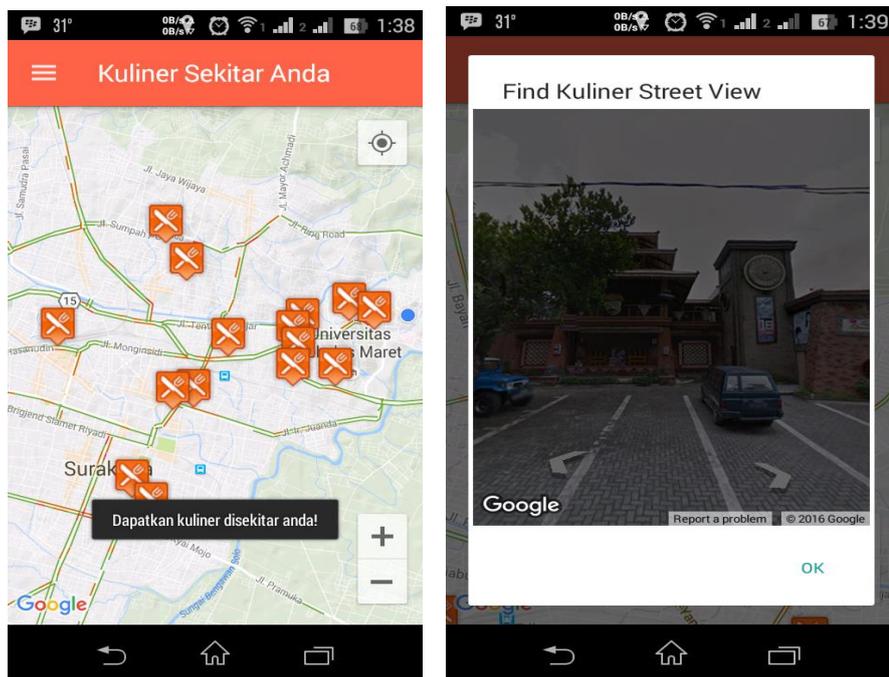
Implementation workflow merupakan transformasi dari model *design* ke *executable code*. *Workflow* tersebut terjadi pada *construction phase* dan merupakan fokus utama dari *phase* tersebut. Tujuan dari *workflow* ini adalah untuk menghasilkan produk perangkat lunak. Pada *implementation workflow* membahas tentang implementasi *class*, implementasi *database*, dan implementasi *interface* dari masing-masing *use case*.

Implementasi *class* diperoleh dari hasil *design classes* pada *design workflow*, implementasi

database diperoleh dari *design database* pada *design workflow*, dan implementasi *interface* diperoleh dari *interface design* pada *design workflow*. Implementasi *interface* kuliner terdekat, terbaru, dan terpopuler aplikasi *Find Kuliner Nusantara* dapat dilihat pada gambar 6.1. Sedangkan implementasi *interface* peta kuliner terdekat dan *streetview* lokasi kuliner aplikasi *Find Kuliner Nusantara* dapat dilihat pada gambar 6.2.



Gambar 6.1 Implementasi *Interface* Kuliner Terdekat, Terbaru, dan Terpopuler



Gambar 6.2 Implementasi *Interface* Peta Kuliner dan *Streetview*

7. Test

Test workflow merupakan tahap pengujian terhadap perangkat lunak yang telah dikembangkan. *Test workflow* dilakukan pada *construction* dan *transition phase*. Pengujian aplikasi *Find Kuliner* Nusantara dilakukan dengan menggunakan metode *blackbox test* dan *performance test*. Materi uji yang dilakukan dengan *blackbox test* yaitu dengan menguji fungsional sistem pada tiap *use case*. Jika masing-masing fungsi sistem berjalan sesuai *requirement* yang telah diidentifikasi, maka pengujian dapat diterima. Sedangkan materi uji yang dilakukan dengan *performance test* yaitu dengan menguji aplikasi saat diakses *user* secara bersama-sama dalam jumlah banyak. Pengujian tersebut dilakukan untuk mengetahui performa aplikasi dalam menerima banyak *request* dalam satu waktu. Perangkat lunak yang digunakan untuk *performance test* adalah *Apache JMeter*. *Apache JMeter* merupakan sebuah *tool* yang digunakan untuk melakukan *performance test* pada sebuah perangkat lunak. Perangkat lunak tersebut dapat memberikan *request* dalam jumlah banyak secara bersamaan dalam satu waktu pada *server*.

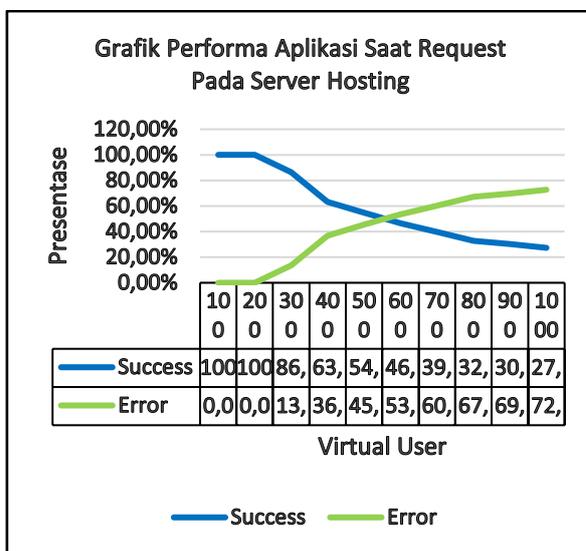
Berikut merupakan spesifikasi *server hosting* yang digunakan untuk melakukan *performance test*.

1. *Server Name* : *Titania*
2. *Disk Space* : *500 MB*
3. *Architecture* : *x86_64*
4. *Operating System* : *Linux*
5. *Server Location* : *Indonesia*
6. *Connection* : *1 Gbps*

Rencana pengujian menggunakan *blackbox test* yaitu melakukan pengujian terhadap fungsional sistem pada masing-masing *use case*. Rencana pengujian menggunakan *performance test* yaitu melakukan pengujian yang dilakukan saat pertama kali aplikasi *android* melakukan *request* data ke *server hosting* ketika aplikasi dibuka yaitu pada menu kuliner terdekat. *Ramp-up period* yang digunakan adalah 1 detik. *Ramp-up period* merupakan waktu *delay* antara *user* satu dengan yang lain saat mengakses *server hosting*. Sedangkan materi uji perbandingan jarak dengan membandingkan hasil perhitungan jarak dengan situs-situs perhitungan jarak dengan *haversine formula*.

Pelaksanaan pengujian *blackbox* dilakukan dengan merealisasikan materi uji yang telah dibuat pada rencana pengujian sebelumnya secara *online* pada *server hosting*. Perangkat pengujian aplikasi *android* dalam kondisi GPS hidup dan terhubung ke jaringan Internet, sedangkan aplikasi *web* dalam kondisi terhubung ke jaringan

Internet. Sedangkan pelaksanaan pengujian *performance* dilakukan dengan merealisasikan materi uji pada rencana pengujian *performance* dan pada saat pengujian jumlah data kuliner berjumlah 32. Kecepatan *modem* Internet yang digunakan saat pengujian *performance* adalah 3.6 *Mbps* untuk *upload* dan 7.2 *Mbps* untuk *download*. Pengujian dilakukan pada *server hosting* yang ditujukan pada alamat *server hosting* saat aplikasi *android* dibuka pertama kali oleh *user*. Sedangkan ringkasan hasil uji dalam bentuk grafik dapat dilihat pada gambar 7.1.



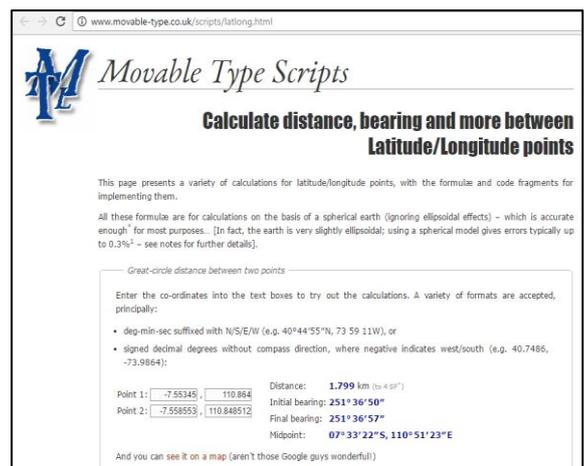
Gambar 7.1 Grafik Performa Aplikasi Saat Request Pada Server Hosting

Uji perbandingan jarak *haversine formula* dilakukan untuk mengetahui apakah hasil perhitungan jarak pada aplikasi sama dengan hasil perhitungan dengan alat ukur lainnya. Pengujian dilakukan dengan menentukan koordinat awal dengan koordinat *latitude* -7.55345 dan *longitude* 110.864 kemudian menentukan lokasi destinasi dengan koordinat *latitude* -7.558553 dan *longitude* 110.848512. Hasil perhitungan jarak aplikasi *Find Kuliner Nusantara* dengan hasil yang ditunjukkan pada gambar 5.34 dengan hasil 1.77 km dapat dilihat pada gambar 7.2. Kemudian dibandingkan dengan alat ukur lain, yaitu situs www.movable-type.co.uk dan situs www.andrew.hedges.name/experiments/haversine/ yang menghitung jarak dengan

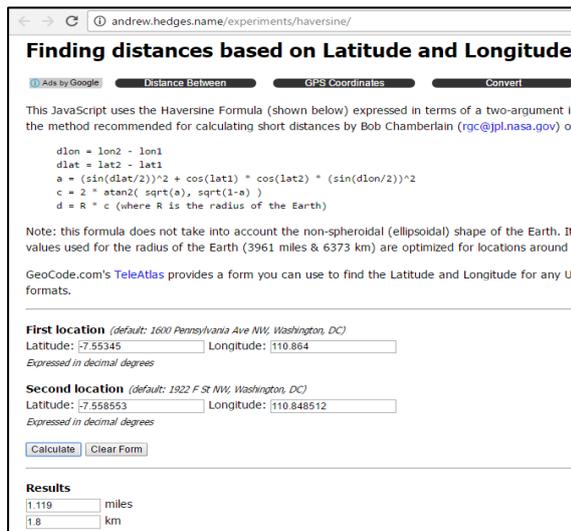
menggunakan *haversine formula*. Hasil perhitungan *movable-type* ditunjukkan pada gambar 7.3 dengan hasil 1.779 km. Sedangkan hasil perhitungan *andrew.hedges.name* ditunjukkan pada gambar 7.4 dengan hasil 1.8 km.



Gambar 7.2 Hasil Perhitungan *Haversine Formula* Pada Aplikasi



Gambar 7.3 Hasil Perhitungan Jarak *Haversine Movable-type*



Gambar 7.4 Hasil Perhitungan Jarak *Haversine Andrew.hedges.name*

Pengujian *blackbox* aplikasi *Find Kuliner Nusantara* telah dilakukan dengan implementasi *database* dan aplikasi *web* pada *server hosting*. Semua pengujian terkait fungsionalitas aplikasi dapat diterima. Semua fungsionalitas aplikasi *android* sudah berjalan sesuai yang diharapkan begitu juga fungsionalitas aplikasi *web*.

Pada pengujian *performance* berdasarkan tabel 5.6, performa aplikasi saat pertama kali melakukan *request* ke *server hosting* adalah 100% sukses saat *virtual user* berjumlah 200. Kemudian menurun menjadi 86.33%, 63.25%, 54.60%, 46.50%, 39.71%, 32.87%, 30.33%, dan 27.30% pada saat *user virtual* bertambah dengan kelipatan 100. Sebaliknya *user virtual* yang mengalami *error* saat melakukan *request* pada *server hosting* adalah 0% saat *user virtual* berjumlah 200. Kemudian *user virtual* bertambah dengan kelipatan 100, performa aplikasi terus mengalami *error* dengan kenaikan menjadi 13.67%, 36.75%, 45.40%, 53.50%, 60.29%, 67.13%, 69.67%, dan 72.70%.

Pada uji perbandingan jarak menggunakan *haversine formula* yang dibandingkan dengan dua alat ukur berbeda, aplikasi *Find Kuliner Nusantara* menghasilkan jarak 1.77 km, sedangkan pada *movable-type* menghasilkan jarak dengan nilai 1.779 km dan pada situs *andrew.hedges.name* menghasilkan jarak dengan nilai 1.8 km. Dapat dilihat bahwa hasil perhitungan aplikasi *Find Kuliner Nusantara*

hampir sama dengan kedua situs tersebut dengan selisih 0.009 km pada *movable-type* dan selisih 0.03 km pada *andrew.hedges.name*.

8. Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian ini adalah :

- Menghasilkan aplikasi *Find Kuliner Nusantara* yang dikembangkan dengan menggunakan model proses UP.
- Aplikasi *Find Kuliner Nusantara* menggunakan teknologi GPS untuk menentukan posisi *user* dan teknologi *cellular positioning* jika layanan GPS tidak tersedia.
- Hasil perhitungan jarak dengan menggunakan *haversine formula* aplikasi *Find Kuliner Nusantara* hampir sama pada situs *www.movable-type.co.uk* dengan selisih 0.009 km dan pada situs *www.andrew.hedges.name/experiments/haversine* dengan selisih 0.03 km.
- Server hosting* aplikasi *Find Kuliner Nusantara* sanggup melayani *request* tidak lebih dari 200 *user* dalam satu waktu.

DAFTAR PUSTAKA

- J. Sharma, "Location Based Information Delivery in Tourism," *International Journal of Computing Science and Communication Technologies*, Vol. %1 dari %24, No. 2, 2012.
- B. Rompas, "Aplikasi Location Based Service Pencarian Tempat di Kota Manado Berbasis Android," *E-journal Teknik Elektro dan Komputer Unsrat*, Vol. %1 dari %21, No. 2, 2012.
- D. Prasetyo, "Penerapan Haversine Formula Pada Aplikasi Pencarian Lokasi dan Informasi Gereja Kristen Di Semarang Berbasis Mobile," *E-Journal Udinus Repository*, 2015.

- [4] B. Sumarsono, "Perancangan Aplikasi Mobile Tambal Ban Terdekat di Kabupaten Sleman Menggunakan Location Based Service Pada Platform Android," *Amikom OJS Journal*, 2014.
- [5] M. Sholeh dan S. Pradhityo, "Aplikasi Mobile Pencari Masjid dan Mushola di Wilayah Daerah Istimewa Yogyakarta Dengan Google Maps," *Prosiding SNST FT Universitas Wahid Hasyim*, Vol. %1 dari %21, No. 1, 2014.
- [6] B. Anwar, "Implementasi Location Based Service Berbasis Android untuk Mengetahui Posisi User," *Jurnal Ilmiah Sains dan Komputer*, Vol. %1 dari %213, No. 2, 2014.
- [7] J. Imaniar, "Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android," *Jurnal Proyek Akhir Politeknik Elektronika Negeri Surabaya (PENS)*, 2011.
- [8] F. Harijanto dan H. Widyantara, "Frekuensi Sistem Pemantauan Posisi Mobil Menggunakan Global Positioning System (GPS) Berbasis Radio," *Jurnal Teknik Komputer Stikom Surabaya*, Vol. %1 dari %210, No. 1, 2008.
- [9] M. Mohammadi, E. Molaei dan A. Naserasadi, "A Survey on Location based Services and Positioning Techniques," *International Journal of Computer Applications*, vol. 24 No. 5, 2011.
- [10] N. Chopde dan M. Nichat, "Landmark Based Shortest Path Detection by Using A* and Haversine Formula," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, pp. 298-302, 2013.
- [11] J. Arlow dan I. Neustadt, *UML and The Unified Process Practical Object-Oriented Analysis & Design*, London: Addison Wesley Professional, 2002.