BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penilitian

Jawa Barat dikenal sebagai provinsi yang memiliki kekayaan pariwisata yang beraneka ragam dan beberapa diantaranya memiliki daya tarik yang sangat tinggi. Sumber daya kebudayaan yang dimiliki seperti bahasa, sastra, dan aksara daerah, kesenian, kepurbakalaan, kesejarahan, nilai tradisional, dan museum, masih tumbuh dan berkembang serta keberadaannya dapat diandalkan untuk pembangunan jati diri bangsa.

Pemerintah menempatkan kepariwisataan menjadi salah satu sektor strategis sebagai salah satu sumber pendapatan asli daerah yang potensial. Berbagai upaya telah banyak dilakukan pemerintah daerah untuk meningkatkan sektor kepariwisataan terhadap pembangunan Jawa Barat secara keseluruhan. Dimulai dari upaya untuk meningkatkan jumlah dan lama kunjungan wisatawan, pelayanan di bidang jasa kepariwisataan, kerja sama dengan pihak luar negeri, promosi, hingga meningkatkan daya tarik kepariwisataan daerah.

Dalam strategi pembangunan kepariwisataan khususnya dalam upaya meningkatkan daya tarik, sektor kebudayaan menjadi unsur andalan yang kemudian sangat mempengaruhi daya beli suatu daerah. Hal ini menempatkan

kebudayaan sebagai sektor potensial bagi pengembangan kepariwisataan sehingga dapat berperan menjadi pendukung dan mitra utama kepariwisataan.

Dengan demikian, kebudayaan dan kepariwisataan merupakan dua sektor yang hendaknya dapat dikembangkan secara sinergis sebagai satu aset dan peluang bagi pengembangan dan pembangunan daerah secara keseluruhan.

3.2 Metode Penelitian

Metode penelitian adalah suatu teknis atau cara mencari, memperoleh, mengumpulkan atau mencatat data, baik yang berupa data primer maupun data sekunder yang digunakan untuk keperluan menyusun suatu karya ilmiah dan kemudian menganalisa faktor-faktor yang berhubungan dengan pokok-pokok permasalahan sehingga akan terdapat suatu kebenaran data-data yang akan *diperoleh.*

3.2.1 Desain Penelitian

Penelitian ini ditujukan untuk membuat rancangan aplikasi media sosial traveler berbasis gamification menggunakan google map api v3. Sejalan dengan tujuan yang dilakukan maka metode yang digunakan adalah metode deskriptif.

Penelitian dengan metode deskriptif adalah suatu metode dengan cara mengumpulkan data kemudian disusun dan dianalisis untuk memperoleh gambaran mengenai masalah yang dihadapi pada saat penelitian.

3.2.2 Jenis dan Metode Pengumpulan Data

Dalam proses pengumpulan data tentu diperlukan sebuah alat atau

instrument pengumpul data. Pengumpulan data terdapat dua sumber yaitu sumber data primer dan sumber data sekunder

3.2.2.1 Sumber Data Primer

Data primer adalah data asli yang dikumpulkan sendiri oleh peneliti. Metode Pengumpulan data primer yang dilakukan diantaranya :

a. Observasi

Observasi sebagai suatu cara untuk memproleh data, dalam hal ini penulis terjun langsung ke lokasi objek penelitian yakni kebeberapa tempat wisata yang ada dijawabarat.

b. Wawancara

Wawancara adalah suatu teknik pengumpulan data dengan tanya jawab secara lisan terhadap para traveler yang sedang melakukan aktivitas traveling ataupun orang yang menyukai kegiatan aktivitas traveling.

3.2.2.2 Sumber Data Sekunder

Data Sekunder adalah data yang diperoleh dari literatur-literatur yang terkait topik penelitian.

a. Studi dokumentasi

Yaitu mengumpulkan data yang berkaitan dalam penyusunan laporan ini secara langsung dari Dinas Pariwisata Provinsi Jawa Barat dan internet, seperti dokumen mengenai obejek keparawisataan yang ada di provinsi Jawa Barat.

b. Studi Literatur

Yaitu pengumpulan data dengan cara mempelajari buku-buku, makalah- makalah, dan skripsi untuk memperoleh informasi yang berhubungan dengan teori dan konsep-konsep yang dibahas

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Dalam sub bab ini akan dijelaskan mengenai metode pendekatan sistem metode pengembangan sistem dan alat bantu analisis pengembangan sistem.

3.2.3.1 Metode Pendekatan Sistem

Metode pendekatan sistem yang di gunakan adalah pendekatan dengan Object Oriented yang menggunakan OOA(*Object Oriented Analysis*) di visualisasikan dengan UML dan di antara nya adalah sebagai berikut : *Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, Collaboration Diagram, Component Diagram dan Deployment Diagram.*

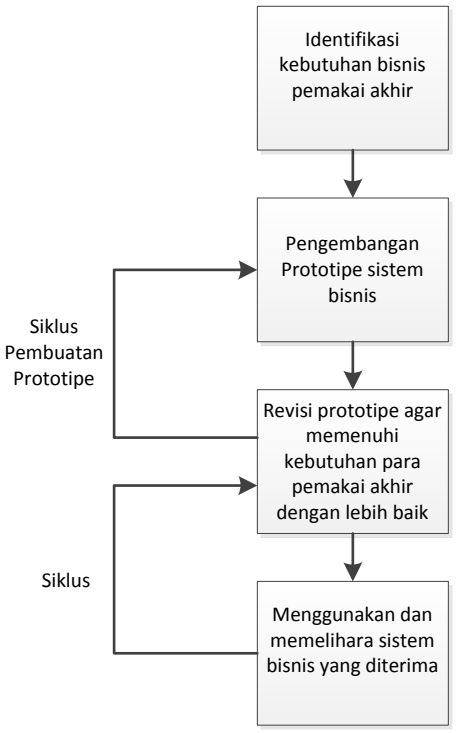
3.2.3.2 Metode Pengembangan Sistem

Pengembangan sistem dapat juga didefinisikan sebagai kumpulan kegiatan dari para analis sistem, perancang, dan pemakai sistem yang mengembangkan danmengimplementasikan sistem informasi. Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode model *prototype*

Prototype merupakan model kerja dari sebuah sistem informasi penjualan yang belum lengkap. Para pengembang sistem informasi melakukan pertemuan-pertemuan intensif dengan *user* untuk menampung informasi yang akan dijadikan dasar dalam disajikan kelak. *Prototype* yang dihasilkan kemuadian

dipresentasikan kepada user dan *user* diberikan kesempatan untuk memberikan masukan-masukan sehingga system informasi penjualaln yang dihasilkan betul-betul sesuai dengan keinginan dan kebutuhan *user*. Perubahan dan presentasi prototype ini dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk sistem informasi penjulan yang akan diterapkan.

Metode prototype dirancang agar dapat menerima perubahan-perubahan dlam rangka menyempurnakan protitipe yang sudah ada sehingga pada akhirnya dapat menghasilkan sistem informasi penjualan yang dapat diterima dan perubahan perubahan yang terjadi dianggap dapat merupakan sebagian dari proses pengembangan itu sendiri.



Gambar 3. 1 Metode Pengembangan Prototyping

Penjelasan tahapan – tahapan *protoyp*

1. Identifikasi kebutuhan *end user*

   Pada tahap ini para pemakai akhir mengidentifikasi kebutuhan bisnis mereka dan menilai kelayakan beberapa alternatif solusi sistem informasi. Pengguna sistem informasi dan vendor mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat beserta alternatif solusi sistem.

2. Membangun prototipe business system

   Pada tahap ini para pemakai akhir atau pakar sistem informasi menggunakan alat pengembangan aplikasi untuk secara interaktif mendesain dan menguji prototipe berbagai komponen sistem informasi yang memenuhi kebutuhan para pemakai akhir. Membangun prototyping aplikasi pengembangan dengan membuat model sebagai uji coba yang mewakili kebutuhan pengguna secara garis besar.

3. *Revisi* prototipe kedalam bentuk yang mendekati kebutuhan *end user*

   Model sistem bisnis diuji coba, dievaluasi dan dimodifikasi berulang-ulang hingga dapat diterima oleh pengguna dan dirasakan oleh pengguna telah sesuai dengan kebutuhan.

4. Menggunakan dan memelihara *business* system yang telah diterima

   Dalam tahap ini sistem bisnis yang telah disepakati antara pengguna dan vendor dapat dimodifikasi dengan mudah karena sebagian besar dokumentasi dari sistem telah tersimpan.

3.2.3.3 Alat Bantu Analisis dan Perancangan

a. *Use case Diagram*

use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang di perbuat system, dan bukan "bagaimana". Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu misalnya login sistem, meng-*create* sebuah daftar belanja dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan – pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan klien, dan merancang test case untuk semua fitur yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di include akan di panggil setiap kali use case yang meng *included* eksekusi secara normal.

Sebuah use case dapat di include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng *extend* use case

lain dengan behaviornya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

b. *Activity Diagram*

Activity Diagram atau yang di sebut Diagram Aktivitas digunakan untuk mendokumentasikan alur kerja pada sebuah system, yang di mulai dari pandangan business level hingga ke operational level, adapaun penjelasanya antara lain :

1. Activity diagram merupakan cara lain memodelkan aliran kejadian.
2. Kita gunakan activity diagram untuk menggambarkan workflow/aliran kerja proses bisnis.
3. Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang di rancang bagaimana masing – masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.
4. Activity diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

c. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan – pekerjaan tertentu.

Use case diagram dapat sangat membantu bula kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua fitur yang ada pada sistem. Sebuah use case dapat meng include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di *include* akan di panggil setiap kali *use case* yang meng *include* dieksekusi secara normal. Sebuah *use case* dapat di include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng *extend use case* lain dengan *behavior* sendiri.

Sementara hubungan generalisasi antar *use case* menunjukan bahwa use case yang satu merupakan spesialisasi dari yang lain.

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class* diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan antar class.

d. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence*

diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men- *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.

e. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message.

Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

f. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code

maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

g. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal . Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

3.2.4 Pengujian *Software*

Pengujian *Software* (perangkat Lunak) adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Metode pengujian yang digunakan oleh penulis adalah *black-box testing*. *Black Box Testing* digunakan untuk menguji fungsi-fungsi dari perangkat lunak yang dirancang. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujuan *black box* memungkinkan perekayasa perangkat lunak mendapat serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian

black box merupakan pendekatan komlementer yang kemungkinan besar mampu mengungkap kelas kesalahan. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi yang tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja dan Inisialisasi kesalahan terminasi

3.3 Analisis Aplikasi *Traveling*

Traveling dewasa ini sudah menjadi bagian dari gaya hidup seseorang yang sangat minati oleh berbagai kalangan. Berbicara *Traveling* maka juga berbicara media informasi. Media informasi merupakan suatu media yang sangat dibutuhkan oleh para *traveler* untuk mencari informasi sekaligus sebagai tempat referensi untuk memilih tempat liburan. Terlebih lagi setelah internet telah melakukan revolusi di dunia teknologi informasi, kemudahan bagi pengguna untuk memperoleh suatu informasi dengan cepat dan praktis.

3.3.1 Aplikasi Traveling

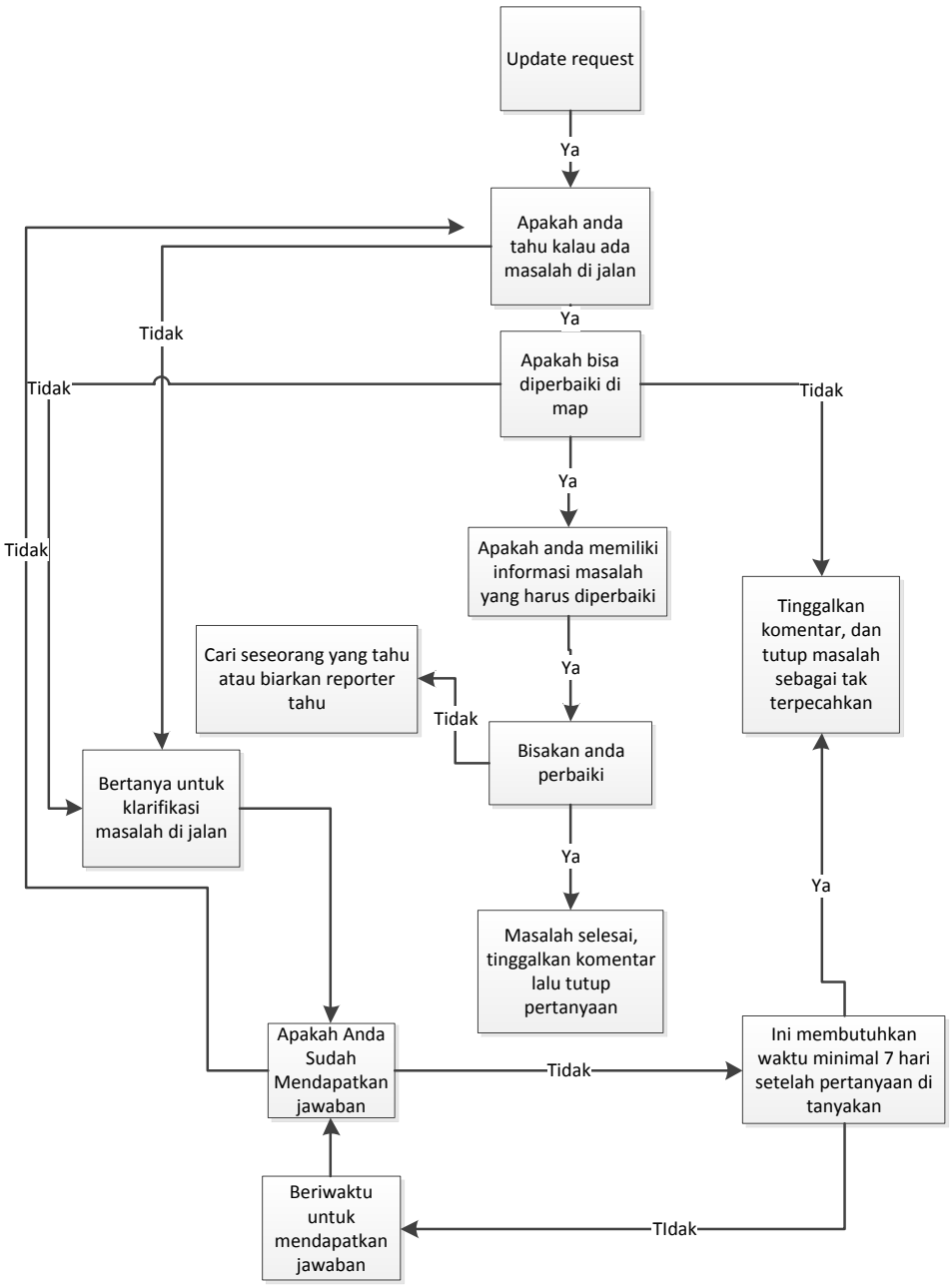
Dalam segi bisnis *traveling* merupakan sebuah lahan bisnis yang sangat menggiurkan. Baik untuk para pengelola hotel ataupun pihak *travel* agen yang menawarkan paket liburan yang menyenangkan. Sehingga para pengusaaha dan para *developer* berlomba – lomba dalam membuat aplikasi untuk memudahkan aktivitas *traveling*, baik dalam proses pemesanan tiket pesawat, hotel atau paket liburan. Berikut beberapa hasil analisa aplikasi traveling.

3.3.1.1 *Waze*

Waze adalah sebuah piranti lunak navigasi gratis untuk perangkat telepon

genggam dan Tablet PC yang memiliki GPS. Saat ini *Waze* mendukung perangkat dengan iOS (*iPhone*/*iPad*), *Android*, *Windows Mobile*, *Symbian* dan *BlackBerry*.

Berbeda dengan piranti lunak navigasi umumnya, *Waze* memberikan informasi dan peta berdasarkan masukan komunitas pemakainya. Informasi mengenai kecelakaan, kemacetan jalan, polisi, bahaya berdasarkan kondisi nyata yang dilaporkan para penggunanya. Pengguna *Waze* yang juga disebut wazers juga bisa melakukan pemutakhiran peta, pemberian nomor rumah/bangunan, penandaan lokasi secara pribadi dan langsung.



Gambar 3. 2 Flowchart *Update Request Waze* App

(Sumber :https://wiki.waze.com/wiki/[13])

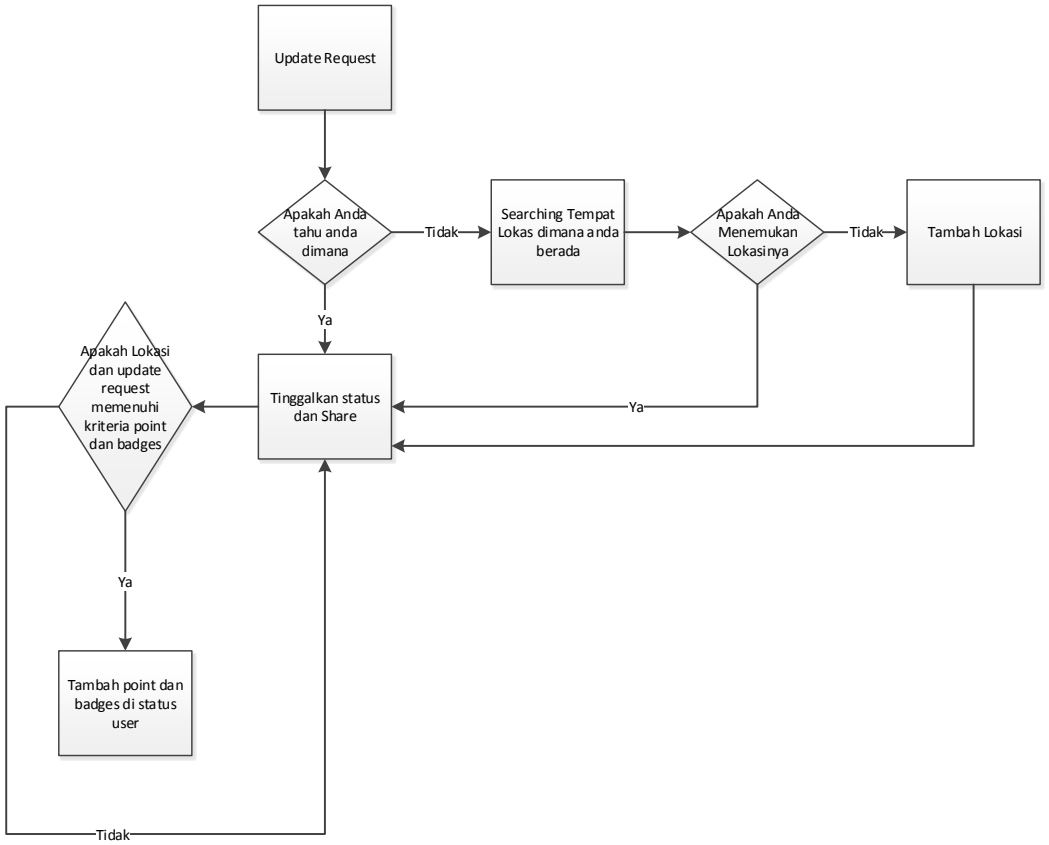
3.3.1.2 *Foursquare*

Foursquare adalah sebuah situs web dan aplikasi dari telepon selular yang memungkinkan pengguna untuk berinteraksi dengan teman mereka dan secara

langsung *check-in* lokasi terbaru mereka. *Foursquare* hanya bisa digunakan pada *Iphone*, *Android* , *Blackberry*, *Palm Pre* dan telepon selular lain yang menawarkan aplikasi *mobile* web.

Check-in di dalam *foursquare* adalah ketika pengguna memberitahu foursquare dimanakah dia berada . *Check-in* bisa dilakukan dimana saja, di restoran, museum, taman kota, atau di universitas. Pada saat anda melakukan check-in maka anda akan diberikan sebuah poin. Poin juga bisa didapatkan dengan menambah lokasi baru yang belum ada di dalam situs web *Foursquare*. Pengguna foursquare juga dapat *check-in* lokasi terbaru mereka sekaligus juga memberikan komentar tentang lokasi tersebut .

Foursquare bekerja sama dengan *Facebook* dan juga *Twitter*, jadi jika pengguna *check-in* maka akan otomatis terdapat pada account *Facebook* dan *Twitter*. Pada *Iphone* versi 1.3, pengguna *Foursquare* dapat mengetahui *check-in* dari teman mereka yang dikenal dengan nama "*Ping*". Pengguna juga bisa mendapatkan sejumlah badges yang bisa diartikan sebagai pangkat dengan *check-in* di tempat-tempat yang direkomendasikan dari situs atau mengunjungi sebuah lokasi dengan intensitas yang tinggi. Dan jika pengguna lain telah mengalahkan intensitas anda dalam check-in di lokasi tersebut maka dialah yang akan dinobatkan sebagai seorang Mayor yang baru. Banyak penawaran yang didapat bila anda adalah seorang mayor dari sebuah lokasi.



Gambar 3. 3 Flowchat Alur *Update Request fourquare*

3.3.1.3 Hasil Evaluasi

Foursquare dan *waze* merupakan sebuah aplikasi yang banyak digunakan dikalangan *traveler*. Akan tetapi masih bisa dikatakan kurang interaktif kecuali *foursquare*. *Waze* merupakan aplikasi untuk memberikan informasi yang ada dijalan sedangkan *foursquare* merupakan aplikasi berbasis *check location* menggunakan passif point dan badges dan keduanya merupakan aplikasi media sosial.

Kedua aplikasi ini tidak memberikan informasi keparawisataan yang ada, hanya memberikan sebatas informasi yang di bagikan oleh orang lain.