



ITS
Institut
Teknologi
Sepuluh Nopember

TESIS C12541
**EFISIENSI TRACKING MULTI TARGET DENGAN MODEL
INTERAKSI PUBLISH-SUBSCRIBE DAN DETEKSI
AKTIFITAS PADA LINGKUNGAN BERGERAK**

DIAN HANIFUDIN SUBHI
5111201002

DOSEN PEMBIMBING
WASKITHO WIBISONO S.Kom, M.Eng, Ph.D
NIP. 197410222000031001

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI BERBASIS JARINGAN
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2013

LEMBAR PENGESAHAN PROPOSAL TESIS

Judul : Efisiensi Tracking Multi Target dengan Model Interaksi Publish-Subscribe dan Deteksi Aktivitas Pada Lingkungan Bergerak
Oleh : Dian Hanifudin Subhi
Nrp. : 5111201002

Telah diseminarkan pada:

Hari : Jum'at
Tanggal : 10-Mei-2013
Tempat : Ruang Rapat TU

Mengetahui/menyetujui:

Dosen Penguji:

1.

Prof. Ir. Supeno Djanali, M.Sc, Ph.D

NIP: 194806191973011001

2.

Ir. Muchammad Husni M.Kom

NIP: 196002211984031001

3.

Dosen Pembimbing:

1.

Waskitho Wibisono, S.Kom, M.Eng, Ph.D

NIP: 197410222000031001

Henning Titi Ciptaningtyas S.Kom, M.Kom

NIP: 198407082010122004

Efisiensi Tracking Multi Target dengan Model Interaksi Publish-Subscribe dan Deteksi Aktifitas Pada Lingkungan Bergerak

Nama Mahasiswa : Dian Hanifudin Subhi
NRP : 5111201002
Pembimbing : Waskitho Wibisono, S.Kom, M.Eng, Ph.D

ABSTRAK

Perkembangan teknologi informasi dan komunikasi mempengaruhi cara orang berinteraksi dengan obyek yang terkait dengan dirinya. Salah satu proses interaksi yang dibutuhkan dalam lingkungan bergerak adalah proses *tracking*. Secara umum, proses *tracking* adalah proses mengamati orang atau benda yang bergerak secara kontinyu dimana obyek-obyek yang diamati terus dimonitor baik posisi maupun aktifitasnya. Proses *tracking* yang ideal dapat mengirimkan perubahan lokasi secara terus-menerus dalam kondisi yang berubah-ubah. Namun demikian sistem *tracking* seperti ini umumnya kurang efisien karena dapat menghabiskan *resource* baik daya maupun kebutuhan *bandwidth* jaringan sehingga membutuhkan proses yang lebih efisien.

Sistem *tracking* tradisional kurang efisien untuk dikembangkan menjadi infrastruktur *tracking* multi target pada perangkat bergerak dimana baik pengamat ataupun obyek yang diamati lebih dari satu. Dibutuhkan mekanisme komunikasi yang *loosely coupled*. *Publish-Subscribe* memiliki kelebihan *decoupling* pada waktu, ruang dan sinkronisasi. Kelebihan interaksi seperti ini, menjadikan sistem *publish-subscribe* ideal dalam komunikasi skala besar yang dinamis. Efisiensi lain dapat dilakukan dengan melakukan *tracking* secara adaptif dan bersifat *context awareness*. Pembaruan lokasi berdasarkan pada kondisi *tracking* yang diberikan oleh pengamat serta kondisi objek pengamatan secara adaptif.

Kata kunci: *Tracking* Multi Target, *Publish-Subscribe*, *Context Awareness*

Efficiency of Multi Target Tracking with Publish-Subscribe Interaction Model and Activity Detection on Mobile Environment

Name : Dian Hanifudin Subhi
Student Identity Number : 5111201002
Supervisor : Waskitho Wibisono, S.Kom, M.Eng, Ph.D

ABSTRACT

The development of information and communication technology affect the way people to interact with objects that associated. One of the processess of interaction needed in mobile environment is the process of tracking. Tracking is the process of observing people or moving objects continuously in which the objects were observed continuously monitored both positions and activities. The ideal tracking process can send the location changes constantly in the changeable conditions. However, such tracking systems are generally less efficient because it can spend resource of power and bandwidth requiring a more efficient process.

Traditional tracking system less efficient to be developed into multi targets tracking infrastructure on mobile devices where either tracker or tracked object is more than one. It takes a loosely coupled communication mechanism. Publish-Subscribe have the advantages of decoupling in time, space and synchronization. This interaction makes the publish-subscribe is ideal on a dynamic large scale communication. Other efficiencies can conducted by doing the tracking adaptively with context awareness. Location updates based on conditions of tracking that given by tracker and tracked object conditions adaptively.

Keywords: Multi Target Tracking, Publish-Subscribe, Context Awareness

Daftar Isi

Lembar Pengesahan	ii
Abstrak	iii
Daftar Isi	v
Daftar Gambar	vi
Daftar Tabel	vii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Penelitian	3
1.4 Tujuan dan Manfaat Penelitian	4
1.5 Kontribusi Penelitian	4
2 KAJIAN PUSTAKA	5
2.1 <i>Tracking</i>	5
2.1.1 <i>Tracking</i> Pada Lingkungan Bergerak	5
2.1.2 <i>Update Protocol</i>	5
2.2 Location Model	8
2.3 <i>Publish-Subscribe</i>	9
2.3.1 Dasar Skema Interaksi <i>Publish-Subscribe</i>	10
2.3.2 Jenis-jenis <i>Publish-Subscribe</i>	11
2.3.3 Protokol MQTT	12
2.4 <i>Context Aware</i>	12
2.4.1 <i>Accelerometer</i>	13
2.4.2 <i>Activity Recognition</i>	13
2.5 Location API Sistem Operasi Android	14
3 METODE PENELITIAN	16
3.1 Tahapan Penelitian	16
3.2 Langkah-langkah Penelitian	17
3.3 Rancangan Sistem	18
3.4 Location Model	20
3.5 <i>Sampling</i> Deteksi Aktifitas	22
3.6 Uji Coba dan Evaluasi	23
3.7 Jadwal Kegiatan Penelitian	24
Daftar Pustaka	25

Daftar Gambar

2.1	Klasifikasi <i>Update Protocol</i> (Leonhardi dkk, 2001)	6
2.2	Perbandingan <i>location model</i> (Hinske, 2006)	9
2.3	Arsitektur dasar <i>publish-subscribe</i> (Eugster dkk, 2003)	10
2.4	Jenis-jenis <i>decoupling</i> pada arsitektur <i>publish-subscribe</i> (Eugster dkk, 2003)	11
2.5	Diagram alir untuk mendeteksi pergerakan (Kjaergaard dkk, 2010) .	14
3.1	Diagram Alir Tahapan Penelitian	16
3.2	Langkah-langkah penelitian secara umum	18
3.3	Diagram alir sistem <i>tracking</i>	19
3.4	Arsitektur rancangan sistem	20
3.5	<i>Location model</i>	22
3.6	<i>Windows Sampling</i> dan <i>Overlapping</i>	22
3.7	Denah lokasi kampus Institut Teknologi Sepuluh Nopember	23

Daftar Tabel

2.1	Perbedaan beberapa protokol (Leonhardi dkk, 2001)	7
2.2	Jenis-jenis <i>LocationProvider</i>	15
3.1	Data Model	21
3.2	Level Hak Akses	21
3.3	Jadwal Kegiatan	24

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang cepat, terutama dalam internet dan telepon seluler telah mengubah cara orang berinteraksi dengan lingkungannya. Telepon seluler secara revolusioner memungkinkan orang untuk berkomunikasi kapan saja dan dimana saja. Adanya hubungan ini menyebabkan perlu adanya konsep berbagi informasi. Informasi-informasi ini dapat berupa data profil pribadi, berkas, lokasi, suhu lingkungan yang didapatkan melalui sensor dan lain-lain.

Fitur perangkat bergerak sudah dilengkapi dengan kemampuan untuk penentuan lokasi memanfaatkan GPS (*Global Positioning System*). Fitur ini tidak hanya digunakan secara lokal pada perangkat tersebut, namun dapat juga dimanfaatkan secara *remote* seperti halnya dalam *GPS tracking*. Agar penggunaan menjadi bermanfaat, proses *tracking* haruslah ramah energi untuk mencegah konsumsi energi berlebih pada perangkat bergerak. Optimasi efisiensi pada perangkat bergerak menjadi isu yang hangat dengan berbagai macam pendekatan. Seperti halnya optimasi menurunkan konsumsi energi dalam protokol jaringan serta optimasi pada level sistem operasi.

Suatu proses *tracking* dikatakan ideal, apabila dapat mengirimkan perubahan lokasi secara *robust* dalam kondisi yang berubah-ubah. Beberapa penelitian yang ada (Civilis dkk, 2005) (Leonhardi dkk, 2001) (Leonhardi dkk, 2002) telah mengusulkan adanya *tracking* secara dinamis untuk memperbaharui lokasi. *tracking* secara dinamis meminimalkan beban pada server dengan menurunkan jumlah pembaharuan, tetapi masih mempunyai nilai akurasi yang relevan. Leonhardi dkk (Leonhardi dkk, 2001) mempelajari *tracking* berbasis waktu serta jarak yang membutuhkan akurasi pada benda bergerak. Simulasi dilakukan dengan melakukan sejumlah pembaharuan pada beberapa teknik *tracking* untuk mengukur akurasinya.

EnTracked (Kjaergaard dkk, 2010), adalah sebuah sistem *tracking* yang efisien pada lingkungan perangkat bergerak untuk target tunggal. Sistem ini dapat mendeteksi kapan diperlukan pembaharuan lokasi memanfaatkan GPS atau tidak secara adaptif. Deteksi pergerakan memanfaatkan sensor *accelerometer* yang tertanam pada perangkat bergerak. Sensor GPS digunakan pada kondisi bergerak atau telah melewati batas waktu yang ditentukan melalui *error model*. Selebihnya lokasi akan diestimasi menggunakan estimasi kecepatan pergerakan. Hasil dari *track-*

ing ini akan disimpan pada *EnTracked Server*, untuk kemudian dikirimkan ke klien yang membutuhkan.

Mekanisme seperti ini, kurang efisien untuk menjadi infrastruktur *tracking* multi target pada perangkat bergerak. Komunikasi *point-to-point* dan *synchronous* menyebabkan aplikasi menjadi kaku dan statis, serta membuat pengembangan skala besar yang dinamis menjadi rumit (Eugster dkk, 2003). Untuk mengurangi beban pada aplikasi, diperlukan skema komunikasi yang bersifat *loosely coupled*. *Publish-Subscribe* memiliki kelebihan *decoupling* pada waktu, ruang dan sinkronisasi. Interaksi seperti ini, menjadikan sistem *publish-subscribe* ideal dalam komunikasi skala besar yang dinamis.

Publish-Subscribe merupakan sebuah paradigma interaksi dimana *subscriber* mempunyai suatu ketertarikan berupa suatu *event*, untuk setiap *event* yang cocok dengan ketertarikan akan dinotifikasi oleh *publisher* (Eugster dkk, 2003). Banyak skema untuk menentukan suatu *event* yang menjadi ketertarikan *subscriber*. Dua skema yang banyak digunakan adalah *topic-based publish-subscribe* serta *content-based publish-subscribe*. Skema *topic-based publish-subscribe* bersifat statis dan primitif, dapat diimplementasikan secara efisien. Lain halnya dengan *content-based publish-subscribe* yang bersifat ekspresif, tetapi sebagai konsekuensinya dibutuhkan suatu komponen *filtering* yang mempunyai kompleksitas yang tinggi.

Adaptasi *publish-subscribe* pada lingkungan perangkat bergerak dipaparkan oleh Huang dkk (Huang dkk, 2004). Dalam penelitiannya dipaparkan beberapa arsitektur *publish-subscribe* dari yang paling sederhana hingga terdistribusi. Sistem *publish-subscribe* dapat secara dinamis mengirimkan *event* kepada *subscriber*. Dibutuhkan *event queues* serta *broker handoff protocol* untuk membangun konsep *publish-subscribe* dalam lingkungan perangkat bergerak. Penelitian ini mendasari *Mobile XSiena*, suatu *platform* yang merupakan pengembangan *XSiena publish-subscribe* berbasis konten (Salvador dkk, 2010).

Salah satu fitur utama dari *publish-subscribe* dapat melakukan *content-filtering*. Dengan adanya ini pengguna hanya mendapatkan pembaharuan yang diinginkan sesuai dengan ketertarikan. Beban pada sisi klien dapat berkurang, karena klien tidak perlu melakukan mekanisme *polling* ke server pada waktu tertentu. Pada sistem *publish-subscribe* tradisional, *publisher* akan tetap mengirimkan *event* kepada server, walaupun tidak ada *subscription* terhadap *event* tersebut.

Penentuan metode pengambilan lokasi pada *tracking*, mempengaruhi dalam efisiensi energi. Metode dengan sensor GPS, memberikan nilai presisi lebih tinggi dari metode lain serta membutuhkan energi yang lebih tinggi juga. Oleh karena itu dibutuhkan sistem *tracking* yang memperhatikan *context aware*. Con-

text berisi informasi yang dapat digunakan untuk menggambarkan situasi dari entitas. Entitas dapat berupa orang, tempat atau obyek yang dianggap relevan untuk interaksi antara pengguna dan aplikasi (Hong dkk, 2009). Pada sistem EnTracked, penentuan lokasi menggunakan sensor GPS hanya digunakan ketika obyek bergerak.

Dari uraian di atas, dapat diketahui bahwa kelebihan *publish-subscribe* dapat dimanfaatkan dalam proses pembaharuan *tracking* pada lingkungan skala besar yang dinamis. Dengan adanya ini sisi client cukup mendaftarkan ketertarikan sebagai *event* kepada server. Pengiriman informasi lokasi pada *tracking* secara sekuensial, menyebabkan perlu adanya penentuan lokasi yang efisien tetapi tetap akurat. Dengan adanya sistem yang bersifat *context aware*, sistem dapat secara adaptif menentukan kebutuhan tingkat presisi informasi. Karakteristik *publisher* yang tetap mengirimkan *event* kepada server dapat membebani sistem, terutama dalam proses *content-filtering*. Oleh karena itu, dalam penelitian ini diajukan suatu metode untuk mengoptimalkan proses *tracking* dengan sifat *context aware* pada arsitektur *publish-subscribe*. Integrasi sistem *tracking* akan dilakukan pada perangkat bergerak yang memiliki sensor GPS di dalamnya.

1.2 Perumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini, yaitu:

1. Bagaimana memodelkan skema *tracking* multi target dengan model komunikasi *publish-subscribe* dalam lingkungan bergerak.
2. Bagaimana menangani *query tracking* multi target dalam lingkungan perangkat bergerak yang bersifat *unreliable*.
3. Bagaimana melakukan optimasi *update protocol* pada sistem *tracking* multi target secara adaptif untuk efisiensi trafik komunikasi.

1.3 Batasan Penelitian

Dalam penelitian ini, batasan masalah yang dibahas diuraikan sebagai berikut:

1. Walaupun masalah keamanan menjadi masalah utama dalam komunikasi jaringan, hal ini tidak ditekankan pada penelitian ini.
2. Sistem *Publish-Subscribe* menggunakan arsitektur tunggal (terpusat).
3. Pemodelan lokasi *tracking* hanya dilakukan untuk wilayah kampus Insitut Teknologi Sepuluh Nopember yang terbagi menjadi tiga level hirarki.

1.4 Tujuan dan Manfaat Penelitian

Tujuan penelitian yang dapat dicapai dalam penelitian ini dapat diuraikan sebagai berikut:

1. Membangun sistem *tracking* multi target berbasis *publish-subscribe* dalam lingkungan bergerak.
2. Membangun *update protocol* yang bersifat adaptif berdasarkan *context* pengguna pada sistem *tracking* multi target.
3. Mengimplementasikan dan menguji secara nyata sistem *tracking* multi target dalam lingkungan bergerak.

Sedangkan manfaat penelitian ini terciptanya sistem *tracking* multi target secara lebih efisien pada lingkungan bergerak. Efisiensi dalam lingkungan bergerak menjadi hal utama dikarenakan keterbatasan *resource* baik daya maupun kebutuhan jaringan.

1.5 Kontribusi Penelitian

Kontribusi penelitian ini adalah pembangunan suatu mekanisme interaksi yang memfasilitasi *tracking* multi target untuk multi user dengan efisiensi pada proses *update protocol* secara adaptif.

BAB 2

KAJIAN PUSTAKA

Dalam bab ini, diuraikan dasar teori mengenai *tracking* serta sistem *publish-subscribe*. Dasar teori tersebut digunakan untuk membangun sistem *tracking* multi target yang efisien dengan arsitektur *publish-subscribe*.

2.1 Tracking

Pada umumnya proses *tracking* adalah proses mengamati orang atau benda yang bergerak dan secara sekuensial mengirimkan data lokasi beserta informasi-informasi lainnya. Proses ini erat kaitannya dengan GPS (*Global Positioning System*) maupun sensor lokasi lainnya. Karena sifatnya yang sekuensial, dibutuhkan proses *tracking* yang efisien namun tetap akurat.

2.1.1 Tracking Pada Lingkungan Bergerak

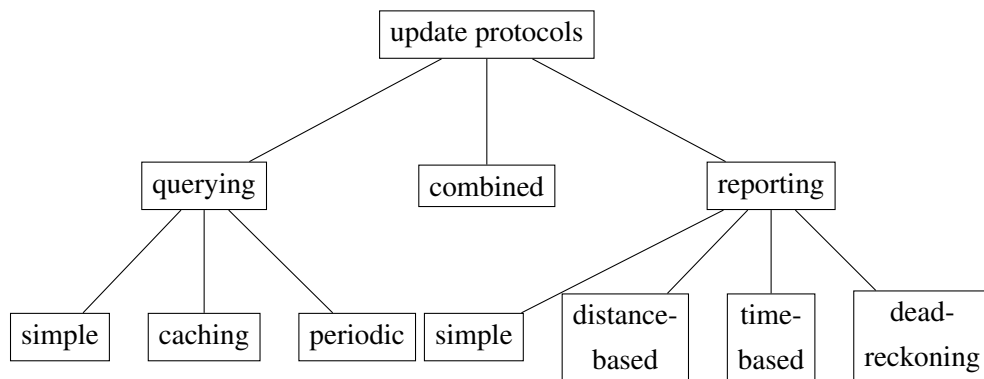
Proses *tracking* dikatakan ideal, apabila dapat mengirimkan perubahan lokasi secara *robust* dalam kondisi yang berubah-ubah. Beberapa penelitian sebelumnya (Civilis dkk, 2005) (Leonhardi dkk, 2001) (Leonhardi dkk, 2002) telah mengusulkan adanya *tracking* secara dinamis untuk memperbarui lokasi. Dalam sistem EnTracked (Kjaergaard dkk, 2010) sebuah sistem *tracking* yang efisien pada lingkungan perangkat bergerak untuk target tunggal. Sistem ini dapat secara adaptif menentukan kebutuhan presisi penentuan lokasi pada lingkungan bergerak. Penentuan lokasi dapat memanfaatkan sensor GPS atau estimasi kecepatan obyek.

Target *tracking* dapat melibatkan satu obyek maupun lebih. Semakin banyak obyek yang di-*tracking* maka semakin besar energi yang dibutuhkan dan pembaharuan posisi yang terjadi. Perbedaan arah pergerakan, variasi kecepatan serta konektifitas pada jaringan tak handal menjadi faktor utama dalam *tracking* banyak obyek. (Ashfaque dkk, 2010) mengajukan metode target kinematika dalam *tracking* pada lingkungan WSN (*Wireless Sensor Networks*). Proses *tracking* hanya aktif ketika sebuah obyek memasuki area tertentu. Hanya pada area *tracking*, node sensor WSN yang aktif. Hal ini dapat mengurangi penggunaan energi dan pengurangan *latency* pada jaringan.

2.1.2 Update Protocol

Terdapat beberapa metode *update protocol* yang dikenalkan dan mempunyai karakteristik yang berbeda. Suatu protokol diasumsikan untuk memberbarui salinan informasi dari obyek bergerak berdasarkan informasi obyek utama. Tujuannya adalah untuk menjamin akurasi informasi yang diberikan. *Update protocol*

dapat diklasifikasikan dalam tiga jenis, yaitu: *querying*, *reporting* dan *combined*.



Gambar 2.1: Klasifikasi *Update Protocol* (Leonhardi dkk, 2001)

1. *Querying Protocol*

Sebuah protokol disebut *querying protocol* jika permintaan informasi lokasi dari sumber ditentukan oleh *server*. Sumber tidak perlu menyimpan informasi yang luas mengenai keadaan atau logika yang rumit. Jika informasi lokasi jarang diperlukan, protokol ini lebih efisien dibandingkan dengan protokol lain yang dipaparkan dalam Gambar 2.1. Terdapat tiga jenis protokol yang masuk dalam kategori ini yaitu: *simple*, *cached* dan *periodic*.

Simple, *server* meminta informasi lokasi dari sumber setiap aplikasi memberikan *request*. Protokol ini memberikan akurasi informasi lokasi yang tinggi, tetapi juga memberikan jumlah pesan yang besar karena frekuensi dari *query*. Selain itu waktu respon *server* juga relatif besar, karena *server* harus menghubungi sumber untuk setiap *query*.

Cached, protokol ini merupakan optimasi dari protokol *simple* dimana *server* menyimpan salinan informasi lokasi terakhir. *Server* melakukan estimasi apakah informasi terakhir masih cukup akurat, jika tidak *server* akan mengirimkan protokol *simple*.

Periodic, jika sisi *server* mengirimkan *query* informasi lokasi secara periodik ke sumber dengan rentang waktu, protokol seperti ini disebut protokol *periodic*. Protokol ini memiliki kemiripan dengan jenis *time-based* pada protokol *reporting* yang akan dijelaskan selanjutnya.

2. *Reporting Protocol*

Perbedaan mendasar dengan *querying protocol* adalah pihak mana yang melakukan inisiasi. Pada *reporting protocol* pihak sumber akan memberikan pembaharuan jika telah mencapai suatu *threshold*. Pada umumnya *reporting protocol* memberikan akurasi yang lebih efisien dibandingkan protokol sebelumnya (lihat

Tabel 2.1). Protokol ini dibagi menjadi tiga jenis, yaitu: *simple*, *time-based* dan *distance-based*.

Pada protokol *simple*, pengiriman informasi lokasi terjadi setiap ada perubahan informasi, dikarenakan sebuah sensor mendapatkan informasi lokasi baru. Pada kasus ini, banyaknya pesan tergantung dari tingkat *update* dari sensor dan biasanya sangat tinggi.

Sedangkan pada protokol *time-based*, informasi lokasi dikirimkan secara periodik setiap batas waktu T terlewati. Jumlah tingkat *update* bersifat tetap, tidak bergantung pada perilaku obyek bergerak. Jika obyek bergerak lambat, sedikit atau tidak ada perbedaan dalam informasi lokasi yang dikirimkan ke *server*. Dan jika obyek bergerak cepat, tidak cukup pesan yang dikirimkan untuk mencapai akurasi yang diinginkan.

Tabel 2.1: Perbedaan beberapa protokol (Leonhardi dkk, 2001)

	Message rate depending on mobility	Message rate depending on queries	Upper bound for uncertainty of results	Applications may specify desired accuracy	Can detect disconnections
Querying:					
Simple		x	(x)	x	x
Cached		x	(x)	x	x
Periodic					x
Reporting:					
Simple			(x)		(x)
Time-based					x
Distance-based	x		x		
Dead reckoning	x		x		x
Combined	x	x	x	x	(x)

Protokol *distance-based* mengirimkan pembaharuan informasi lokasi ketika jarak antara lokasi dari obyek bergerak dan lokasi terakhir lebih besar dari *threshold* yang diberikan. Protokol ini akan memberikan lebih banyak pesan jika obyek bergerak cepat dan lebih sedikit jika bergerak pelan. Dimungkinkan juga untuk menggabungkan protokol *time-based* dan *distance-based*.

Optimasi dari protokol *distance-based* adalah *dead-reckoning*. Server melakukan estimasi posisi sekarang dengan melihat posisi lama serta kecepatan dan arah pergerakan dari obyek. Sisi sumber juga melakukan estimasi lokasi dan mengirimkan pembaharuan jika informasi lokasi telah berbeda dari *threshold*.

3. Combined Protocol

Querying protocol tidak dapat menyesuaikan karakteristik pergerakan yang

berbeda dari obyek bergerak, sedangkan *reporting protocol* tidak mempertimbangkan tingkat permintaan dan akurasi yang diminta oleh aplikasi. Dengan *combined protocol* yang mengintegrasikan *distance-based* dan *cached*, kedua kebutuhan di atas dapat terpenuhi. Jika informasi lokasi yang tersimpan pada *server* kurang akurat, server akan meminta informasi lokasi dari sumber. Sedangkan sisi sumber mempunyai perilaku yang sama dengan *distance-based*. Dengan gabungan protokol ini, waktu respon bergantung pada seberapa akurat informasi yang tersimpan pada sisi server.

2.2 Location Model

Dalam proses *tracking* diperlukan adanya yang disebut *location model*. *Location model* adalah sebuah representasi yang ekspresif, fleksibel dan efisien dari informasi lokasi (Leonhardi, 1998). Informasi lokasi tak lepas dengan koordinat, yaitu penanda yang menentukan posisi sebuah obyek sehubungan dengan referensi sistem koordinat yang diberikan (Becker, 2005). Sistem koordinat terbagi menjadi dua, yaitu:

1. Koordinat geometris yang mengacu pada titik geometris atau ruang multi dimensi seperti pada koordinat kartesius maupun koordinat geografis (*latitude* dan *longitude*).
2. Koordinat simbolis, mendefinisikan lokasi dalam bentuk simbol abstrak atau nama. Contoh: “Jakarta”, “Surabaya”, “Ruang IF 105” dan lain-lain.

Dari kedua sistem koordinat diatas, *location model* dibagi menjadi tiga macam, yaitu: *Geometric*, *Symbolic* dan *Hybrid*.

1. *Geometric*

Geometric location model berbasiskan referensi koordinat sistem geometris. Sebuah lokasi dapat direpresentasikan dalam suatu titik, area atau volume pada sistem koordinat. Relasi jarak dapat mudah didapatkan dengan penurunan geometris menggunakan model matematika.

2. *Symbolic*








Symbolic location model berbasiskan koordinat simbolis. Sebuah *symbolic location* adalah nama yang bersifat simbol yang kehilangan referensi geometri. Kumpulan dari beberapa koordinat dapat dikelompokkan dalam sebuah *symbolic location*, misalnya: Gedung Jurusan Informatika. Beberapa pengembangan dari *location model* ini antara lain: *set-based*, *hierarchical*, *graph-based* dan *combined symbolic*.

3. *Hybrid*

Perpaduan antara *symbolic location model* yang melampirkan koordinat

dalam sistem referensi. Terdapat dua pendekatan yaitu, *subspaces* dan *partial subspaces*. Perbedaan utama kedua pendekatan diatas adalah jumlah informasi geometris yang disimpan. *Partial subspaces* hanya menyimpan informasi geometris untuk beberapa lokasi saja.

Setiap desain *location model* mempunyai kelebihan dan kekurangan masing-masing, hal ini dipaparkan pada Gambar 2.2. Tanda + menunjukkan dukungan akan fungsi tersebut, semakin banyak tanda + menunjukkan kualitas yang lebih baik. Sedangkan tanda - bersifat sebaliknya. Adapun untuk kategori *High*, *Medium*, dan *Very High* menunjukkan kompleksitas pemodelan *location model*. Suatu desain *location model* yang baik, desain yang dapat memenuhi kebutuhan serta kompleksitas desain secara seimbang.

Location model	Type	Position	Nearest neighbor	Navigation	Range	Modeling effort
Geometric 	geo	+	+	-	++	High
Set-based 	symb	+	-	-	+	High
Hierarchical 	symb	+	-	-	++	Medium
Graph-based 	symb	+	+	++	-	Medium
Combined symbolic 	symb	+	+	+	+/++	High
Subspaces (Hybrid) 	symb/ geo	++	++	++	++	Very high
Partial subspaces (Hybrid) 	symb/ geo	++	+/++	+/++	+/++	High

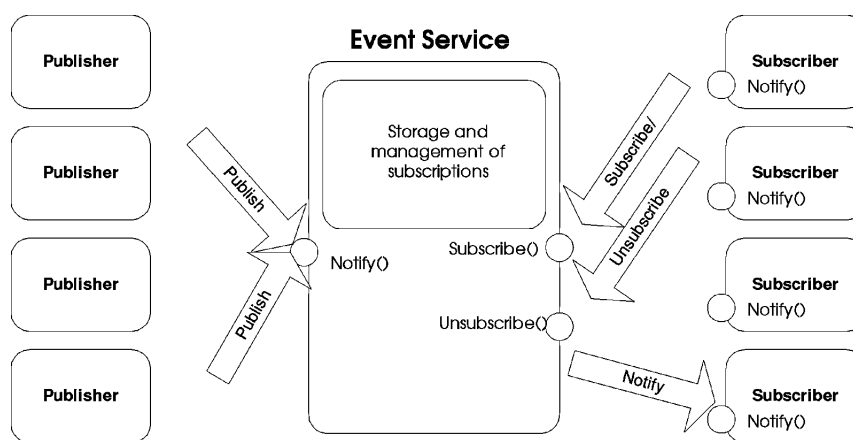
Gambar 2.2: Perbandingan *location model* (Hinske, 2006)

2.3 Publish-Subscribe

Skema interaksi *publish-subscribe* mendapat perhatian yang cukup meningkat terutama dalam interaksi skala besar. *Subscriber* dapat mendaftarkan ketertarikan suatu *event publish-subscribe* atau pola *event publish-subscribe* yang dikirimkan oleh *publisher*. Kelebihan dari *event-based interaction* ini adanya sifat *decoupling* pada waktu, ruang serta sinkronisasi antara *publisher* dan *subscriber*.

Dasar dari model arsitektur pada interaksi *publish-subscribe* (Gambar 2.3)

bergantung pada *event notification service* yang menyediakan penyimpanan dan manajemen untuk *subscription* dan pengiriman *event publish-subscribe* secara efisien. *Subscriber* mendaftarkan ketertarikan mereka yaitu *event publish-subscribe* dengan memanggil operasi *subscribe()* pada *event service*. Informasi *subscription* ini akan disimpan pada *event service* tanpa disalurkan kepada *publisher*. Operasi *unsubscribe()* menghapus *subscription* pada *event service*. Untuk mengirimkan suatu *event publish-subscribe*, *publisher* memanggil operasi *publish()*. Setiap *subscriber* akan mendapat notifikasi setiap *event publish-subscribe* yang menjadi ketertarikannya.



Gambar 2.3: Arsitektur dasar *publish-subscribe* (Eugster dkk, 2003)

2.3.1 Dasar Skema Interaksi *Publish-Subscribe*

Interaksi dalam sistem *publish-subscribe* antara pihak *publisher* dengan *subscriber* yang disediakan oleh *event service* terdapat *decoupling*. Adanya *decoupling* ini memberikan kelebihan pada sistem interaksi *publish-subscribe* yang dapat didekomposisikan menjadi tiga dimensi. Untuk lebih jelasnya dapat dilihat pada Gambar 2.4.

1. *Space decoupling*

Antara pihak yang berinteraksi tidak perlu saling mengetahui satu dengan yang lain. Para *publisher* mempublikasikan suatu *event publish-subscribe* melalui *event service* dan para *subscriber* mendapatkannya secara tidak langsung melalui *event service*. Dalam interaksi ini antara *publisher* dan *subscriber* tidak saling mengetahui berapa jumlah yang terlibat.

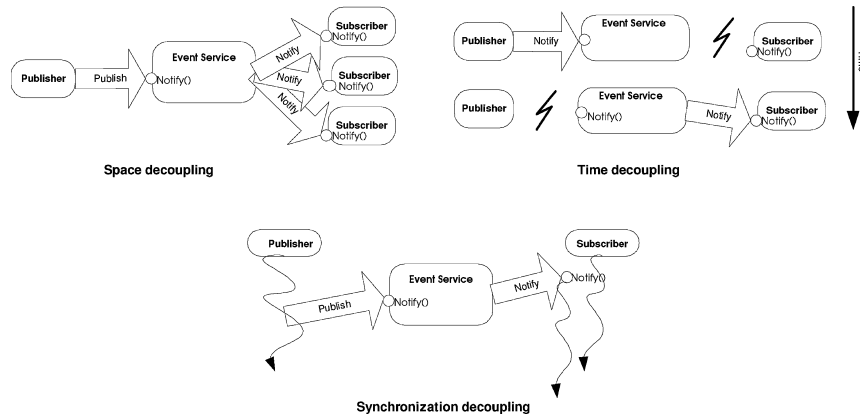
2. *Time decoupling*

Para pihak yang berinteraksi tidak perlu secara aktif berpartisipasi dalam interaksi pada waktu yang bersamaan. Pihak *publisher* mungkin mem-publish beberapa *event* saat *subscriber* terputus. Dan sebaliknya, *subscriber* mungkin akan

mendapatkan notifikasi sementara *publisher* sedang terputus dari interaksi.

3. *Synchronization decoupling*

Dalam interaksi antara pihak *publisher* dan *subscriber*, *publisher* tidak diblok ketika mengirimkan suatu *event*, dan *subscriber* mendapatkan pemberitahuan secara *asynchronous*. Proses interaksi *event* tidak terjadi pada aliran kontrol utama antara *publisher* dan *subscriber* serta tidak terjadi secara sinkron.



Gambar 2.4: Jenis-jenis *decoupling* pada arsitektur *publish-subscribe* (Eugster dkk, 2003)

2.3.2 Jenis-jenis *Publish-Subscribe*

Umumnya *subscriber* hanya tertarik pada *event publish-subscribe* tertentu, bukan semua *event publish-subscribe*. Ada berbagai macam cara yang dapat digunakan dalam menentukan hal ini. Dua skema yang paling banyak digunakan yaitu, *topic-based publish-subscribe* serta *content-based publish-subscribe*.

1. *Topic-Based Publish-Subscribe*

Merupakan jenis skema *publish-subscribe* pertama yang berdasarkan topik atau subyek. Ketertarikan *subscriber* diidentifikasi dengan kata kunci. Pengembangan dari skema jenis ini dengan adanya hirarkis topik. Ketertarikan pada suatu topik termasuk sub-topik yang berada di bawahnya. Selain itu ketertarikan pada suatu topik dapat juga berupa *wildcard*, hal ini pertama kali diperkenalkan pada TIBCO *Rendezvous*. Nama topik pada umumnya dinotasikan dalam bentuk seperti URL, misalnya: *Eurecom/Courses/DSMWare*. Kelemahan skema ini bersifat statis, kriteria topik sudah didefinisikan.

2. *Content-Based Publish-Subscribe*

Skema *subscription* pada jenis ini, ketertarikan didasarkan pada isi dari suatu *event publish-subscribe*. Pada umumnya properti nya berisi atribut dari data struktur. Skema jenis ini lebih dinamis dari *topic-based publish-subscribe*. Dalam skema ini terdapat suatu bahasa dalam menentukan *subscription*, contohnya:

“*Course=DSMWare and Grade<10*”.

3. Type-Based Publish/Subscribe

Suatu topik biasanya dikelompokkan tidak hanya berdasarkan kemiripan yang muncul, tetapi juga dalam suatu struktur. Hal ini yang mendasari adanya klasifikasi yang menyaring suatu *event publish-subscribe* berdasarkan jenisnya.

2.3.3 Protokol MQTT

Protokol MQTT (MQ *Telemetry Transport*) adalah sebuah protokol *publish-subscribe* yang didesain sangat sederhana dan ringan. Protokol ini desain untuk perangkat dengan *resource* terbatas, *low-bandwidth*, *high-latency* atau jaringan yang tak handal. Protokol ini ditemukan oleh Dr Andy Stanford-Clark dari IBM dan Arlen Nipper dari Arcom pada tahun 1999. Hunkeler dkk (Hunkeler dkk, 2008) melakukan adaptasi protokol MQTT pada lingkungan WSN (*Wireless Sensor Network*). Lingkungan WSN membutuhkan suatu protokol yang ramah energi serta *low-bandwidth*. Beberapa kelebihan protokol ini dalam lingkungan WSN, komunikasi data berdasarkan ketertarikan bukan dari suatu alamat perangkat serta dapat menyembunyikan topologi jaringan. Protokol ini memungkinkan adanya komunikasi antara WSN dan jaringan terdahulu maupun antar WSN yang berbeda. Karakteristik lingkungan WSN dapat disejajarkan dengan lingkungan bergerak, sehingga dalam penelitian ini akan digunakan protokol MQTT.

ActiveMQ (sebuah *messaging queue framework*) yang dirilis oleh Apache menyediakan kemudahan dalam membangun sistem *publish-subscribe*. ActiveMQ mendukung bermacam client bahasa pemrograman dan protokol. Selain dukungan di atas, *framework* ini telah teruji pada server J2EE yang umum dikenal. ActiveMQ dibangun menggunakan bahasa Java, sehingga mewarisi kelebihan dari bahasa pemrograman Java. Dan selain itu, protokol MQTT sudah didukung oleh *framework* ini.

2.4 Context Aware

Untuk membangun sistem yang adaptif diperlukan adanya *context aware*. *Context* didefinisikan sebagai informasi yang dapat digunakan untuk menggambarkan situasi dari entitas (Hong dkk, 2009). Entitas disini dapat berupa orang, tempat atau obyek yang dianggap relevan untuk interaksi antara pengguna dengan aplikasi termasuk lokasi, waktu, aktivitas dan preferensi. Sebuah sistem dapat dikategorikan dalam *context aware* jika sistem dapat mengekstrak serta menterjemahkan menggunakan informasi dari *context* serta terdapat fungsi yang dapat beradaptasi pada *context* yang digunakan. Dalam pengertian yang lebih sederhana, sebuah sistem disebut *context aware* jika sistem menggunakan *context* untuk memberikan in-

formasi yang relevan kepada pengguna, dimana relevansi tergantung pada kegiatan pengguna.

2.4.1 Accelerometer

Sensor *accelerometer* adalah sebuah alat yang berfungsi untuk mengukur percepatan dan getaran akibat gravitasi bumi. Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu dimana terjadi perubahan kecepatan yang semakin bertambah daripada kecepatan sebelumnya. Hal ini disebut dengan *acceleration*. Percepatan juga bergantung pada arah/orientasi karena merupakan penurunan kecepatan yang merupakan besaran vektor. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula. Pembagian sumbu pada *accelerometer* dibagi menjadi sumbu x , y dan z (*triaxial accelerometer*). Aktivitas pengguna yang berbeda menghasilkan nilai yang berbeda pula, dipengaruhi interaksi antara pengguna dan sensor.

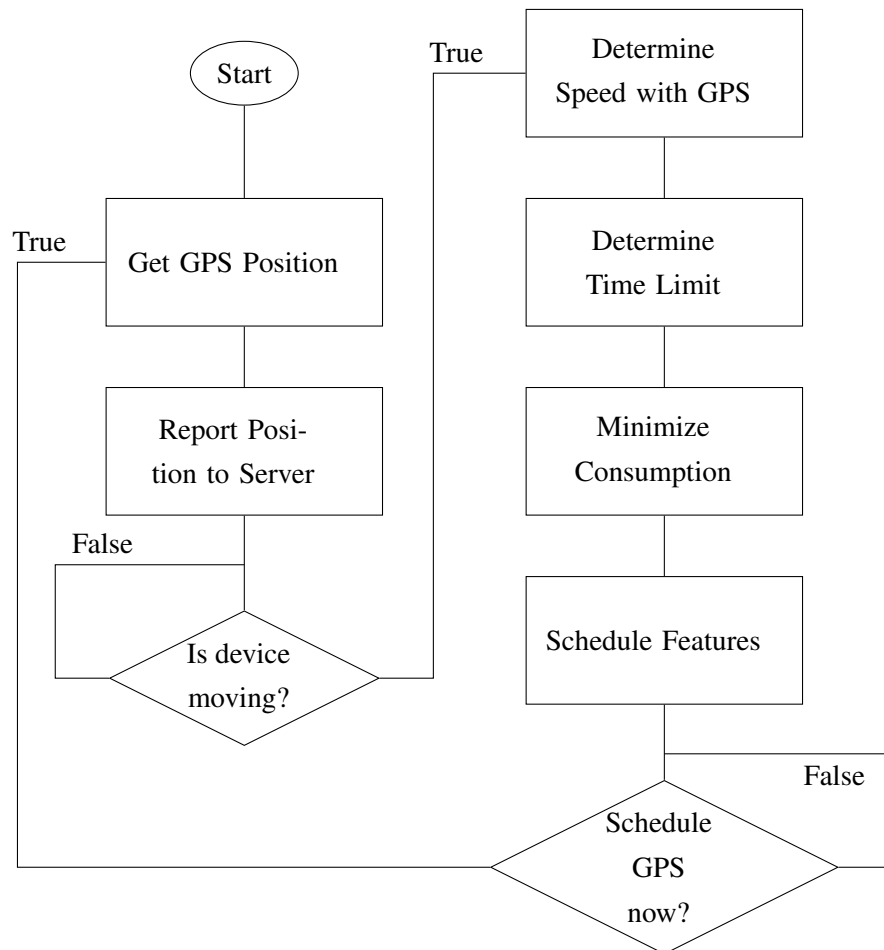
Dengan memanfaatkan sensor *accelerometer* dapat dideteksi aktivitas seorang pengguna. Randell dkk (Randell dkk, 2000) mengumpulkan data sampel sumbu x dan y pada *accelerometer*. Data sampel dari 10 orang, berisi tentang aktivitas pengguna dalam berbagai aktivitas: berjalan, berlari, duduk, berjalan naik tangga, turun tangga dan berdiri. Dari data yang dikumpulkan, dianalisa untuk mengetahui pola aktivitas menggunakan algoritme *clustering*.

2.4.2 Activity Recognition

Activity recognition merupakan teknik yang digunakan dalam proses klasifikasi aktivitas fisik pengguna (berjalan, duduk atau berlari). Dalam proses ini terbagi menjadi beberapa tahapan, yaitu: *data collection*, *feature extraction* dan *data interpretation*. Untuk melakukan sebuah klasifikasi diperlukan adanya data. Data yang diperoleh disesuaikan dengan *context* yang digunakan. Pada umumnya pengambilan data dilakukan dengan menggunakan sebuah alat yang berupa sensor. Dari data yang didapatkan melalui sensor, proses selanjutnya adalah ekstraksi data. Data yang akan diklasifikasi diolah sedemikian rupa sehingga data tidak terlalu konvergen atau divergen. Kemudian tahapan terakhir yaitu klasifikasi, tahapan yang terpenting dalam pengenalan aktivitas pengguna.

Mengacu pada konsep *tracking* yang ideal, maka diperlukan mekanisme penentuan lokasi yang ramah energi. Suatu mekanisme yang dapat secara adaptif menentukan tingkat kebutuhan presisi suatu lokasi. Dalam kondisi bergerak, diperlukan ketelitian posisi yang lebih baik. Pergerakan dapat dideteksi memanfaatkan sensor *accelerometer* yang telah diajukan oleh You dkk (You dkk, 2008).

Mekanisme ini kemudian diadopsi pada sistem EnTracked (Kjaergaard dkk, 2009). Sistem EnTracked hanya mendeteksi dua kondisi pergerakan, yaitu: diam berdiri atau bergerak. Ketika terjadi pergerakan yang terdeteksi, perubahan lokasi ditentukan menggunakan sensor GPS. Diagram alir untuk deteksi pergerakan *tracking* dipaparkan pada Gambar 2.5.



Gambar 2.5: Diagram alir untuk mendeteksi pergerakan (Kjaergaard dkk, 2010)

2.5 Location API Sistem Operasi Android

Saat ini, *smartphone* seperti Android sudah dilengkapi dengan GPS untuk menentukan lokasi. Dalam menentukan lokasi, tidak hanya memanfaatkan GPS saja, tetapi dapat juga dengan memanfaatkan Cell-Id atau WiFi. Sistem operasi Android sudah menyediakan *Location API*, sebuah API (*Application Programming Interface*) yang dapat membantu dalam menentukan lokasi. API ini terdapat dalam *package android.location*. Terdapat dua kelas utama yang berperan dalam penentuan lokasi yaitu *LocationManager* dan *LocationProvider*. *Class LocationManager* menyediakan akses layanan lokasi pada Android. Layanan ini memungkinkan

untuk mengakses penyedia lokasi. Sedangkan *class LocationProvider* menentukan metode apa yang digunakan untuk memperoleh lokasi. Beberapa metode yang tersedia dijabarkan pada Tabel 2.2.

Tabel 2.2: Jenis-jenis *LocationProvider*

<i>LocationProvider</i>	Deskripsi
network	<i>Provider</i> ini menentukan lokasi berdasarkan ketersediaan menara seluler dan akses WiFi. Mungkin memiliki presisi lebih tinggi dibandingkan dengan GPS dalam ruang tertutup.
gps	<i>Provider</i> ini menentukan lokasi menggunakan satelit. Tergantung pada kondisi, <i>provider</i> ini mungkin memerlukan waktu untuk mendapatkan lokasi yang tepat. Pada umumnya memiliki presisi lebih baik dibandingkan dengan <i>network</i> .
passive	<i>Provider</i> ini dapat digunakan secara pasif dalam menerima perubahan lokasi di saat aplikasi atau layanan lain meminta perubahan posisi. <i>Provider</i> ini mengembalikan lokasi yang dihasilkan oleh <i>provider</i> lain.

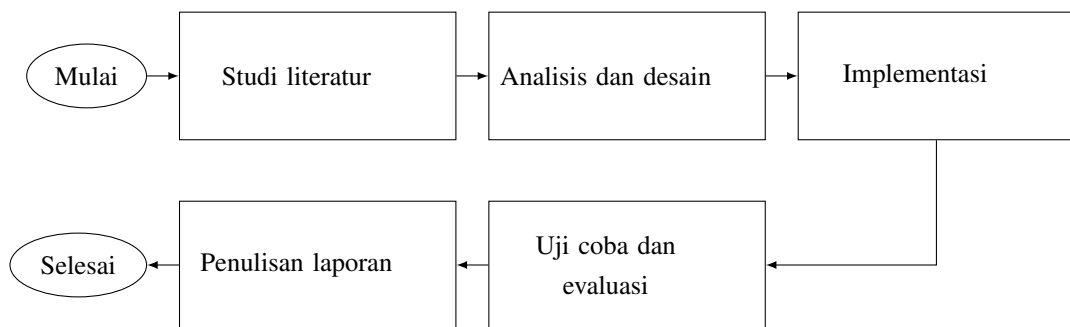
Selain metode-metode *provider* di atas, penentuan lokasi dapat dilakukan secara fleksibel dengan menggunakan *provider* terbaik menggunakan *Criteria*. Dalam penentuan lokasi *provider* membutuhkan akses dari sistem pada android, yang diatur dalam ACCESS_COARSE_LOCATION serta ACCESS_FINE_LOCATION (*provider* gps). Ketika akses ACCESS_FINE_LOCATION digunakan, akses ACCESS_COARSE_LOCATION tidak diperlukan.

BAB 3

METODE PENELITIAN

3.1 Tahapan Penelitian

Pada penelitian ini akan dilakukan dalam beberapa tahapan. Tahap yang pertama adalah studi literatur untuk mempelajari dan memahami materi-materi terkait dengan penelitian, seperti *tracking* serta arsitektur *publish-subscribe*. Tahapan yang kedua yaitu melakukan analisa. Tahap ketiga adalah mengimplementasikan hasil analisa. Selanjutnya, tahap keempat yaitu melakukan ujicoba hasil implementasi pada lingkungan ujicoba yang telah disiapkan. Dan tahap yang terakhir adalah melakukan evaluasi hasil dari uji coba dan membuat laporan akhir. Skema tahapan penelitian dapat dilihat pada Gambar 3.1. Tahapan-tahapan ini akan dikerjakan sesuai dengan jadwal yang terlampir pada Tabel 3.3 pada halaman 24.



Gambar 3.1: Diagram Alir Tahapan Penelitian

1. Studi Literatur

Dalam studi literatur, dikaji berbagai referensi mengenai konsep *tracking* dan arsitektur *publish-subscribe* serta karakteristik *context aware*. Dari hasil studi literatur ini, dapat ditemukan kekurangan dan kelebihan dari masing-masing topik. *Tracking* yang efisien harus bersifat adaptif dan efisien dalam energi karena proses pembaharuan yang bersifat sekuensial. Pada sistem *publish-subscribe* mempunyai kelebihan yang dapat diadopsi pada proses *tracking*, terutama dalam proses *tracking* multi target.

2. Analisis dan Desain

Tahap awal analisis dan desain adalah merumuskan kontribusi utama penelitian ini. Dari studi literatur yang telah dilakukan, dapat diketahui bahwa sistem *tracking* memerlukan mekanisme yang efisien dalam energi dan pembaharuan. Sistem *tracking* adaptif terhadap kebutuhan tingkat presisi suatu lokasi. Dalam tahap desain, dijelaskan langkah-langkah dalam proses pengerjaan penelitian. Rancangan

telah dibagi dalam modul-modul sehingga akan mempermudah dalam tahap implementasi.

3. Implementasi

Dalam tahap ini, diimplementasikan rancangan yang telah dibuat pada proses sebelumnya. Bahasa pemrograman yang digunakan adalah bahasa Java pada sisi client maupun server. Implementasi dilakukan pada *smartphone* bersistem operasi android pada sisi klien, dan pada sisi server dikembangkan dengan bantuan ActiveMQ sebagai sistem *publish-subscribe*. ActiveMQ mendukung protokol MQTT (MQ Telemetry Transport), sebuah protokol yang ringan dalam sistem *publish-subscribe* sehingga cocok digunakan dalam komunikasi pada jaringan bergerak.

4. Uji Coba dan Evaluasi

Untuk menguji apakah kontribusi yang diajukan dapat berjalan dengan baik, perlu dilakukan uji coba. Uji coba dilakukan pada *smartphone* dalam suatu area yang telah ditentukan dengan berbagai kondisi. Skenario uji coba dilakukan dengan memberikan berbagai beban kerja pada proses *tracking*. Hasil uji coba akan dievaluasi dan dapat dilihat kinerja metode yang diajukan.

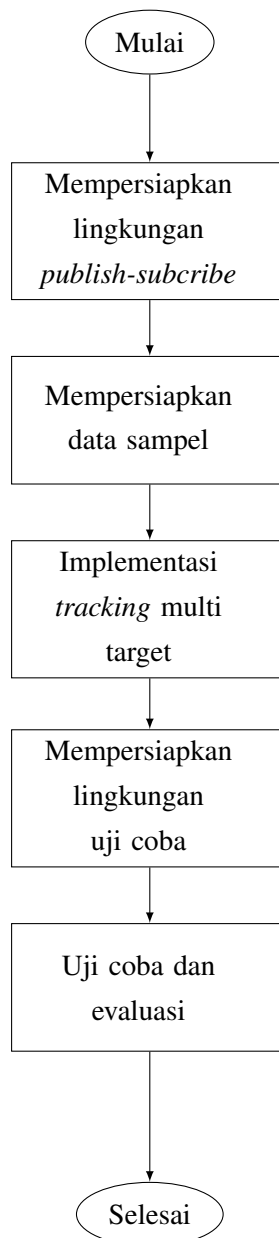
5. Penulisan Laporan Penelitian

Tahap ini adalah tahap untuk menyusun laporan penelitian. Setiap kegiatan yang dilakukan dalam penelitian akan didokumentasikan. Laporan berisi penjelasan mulai dari tahap studi literatur, analisa, tahap ujicoba serta kesimpulan dan saran. Laporan penelitian ditulis sesuai dengan ketentuan yang berlaku.

3.2 Langkah-langkah Penelitian

Langkah-Langkah penelitian secara umum disajikan dalam Gambar 3.2. Langkah pertama adalah mempersiapkan lingkungan *publish-subscribe*, yaitu instalasi *ActiveMQ*. Selanjutnya, mempersiapkan data sampel yang digunakan dalam *activity recognition*. Langkah selanjutnya adalah mengimplementasikan *tracking* multi target, implementasi dilakukan pada sisi *client* serta *server*. Kemudian, dilakukan uji coba dan evaluasi untuk mengetahui kinerja dari metode yang diajukan pada lingkungan uji coba yang telah disiapkan dalam penelitian ini.

Dalam mempersiapkan lingkungan *publish-subscribe* diperlukan suatu *middleware*, yaitu *ActiveMQ*. *Middleware* ini dibangun menggunakan bahasa pemrograman Java. Oleh karena itu sebelum konfigurasi *ActiveMQ* dilakukan, diperlukan pengaturan lingkungan *JAVA_HOME* pada sistem operasi. Sistem operasi yang dipilih adalah Linux, dikarenakan kemudahan dalam konfigurasi serta kehandalannya. Banyak metode dalam mengatur konfigurasi *ActiveMQ*, tetapi dalam penelitian ini akan dilakukan dengan konfigurasi berkas XML sebagai parameter.



Gambar 3.2: Langkah-langkah penelitian secara umum

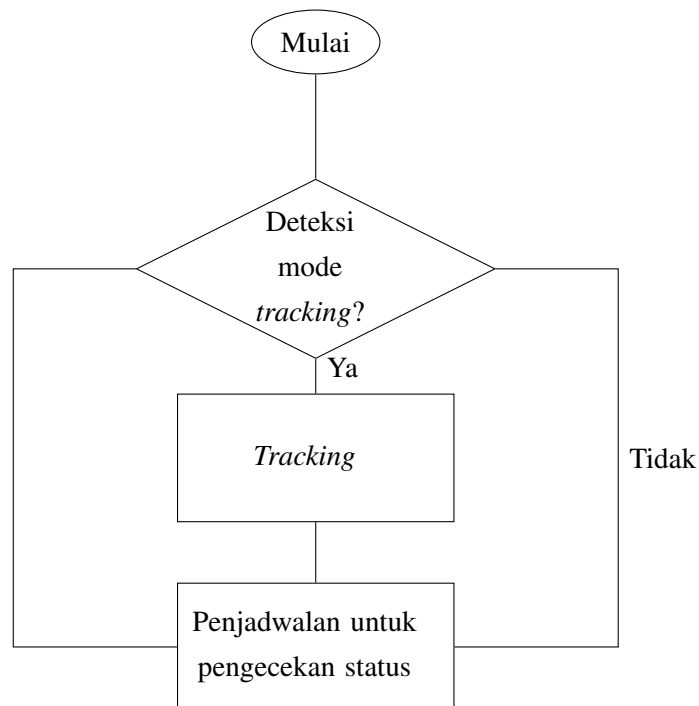
3.3 Rancangan Sistem

Pada penelitian sebelumnya (Kjaergaard dkk, 2009) telah diimplementasikan sistem *tracking* yang efisien untuk perangkat bergerak. Sayangnya sistem *tracking* ini kurang mengakomodir untuk proses *tracking* multi target. Sistem *publish-subscribe* ideal pada lingkungan jaringan tak handal seperti WSN (Hunkeler dkk, 2008). Karena karakter lingkungan WSN yang hampir sama dengan jaringan bergerak, hal ini dapat diadopsi.

Interaksi antara *publisher* (entitas yang menjadi target *tracking*) dengan

subscriber (entitas yang melakukan *tracking* atau *tracker*) dapat dipaparkan pada Gambar 3.4. *Publisher* akan mengirimkan *event* kepada sistem *publish-subscribe* berupa informasi *tracking*. Sedangkan *subscriber* mengirimkan *subscription* yang diatur dalam *subscription language* kepada sistem *publish-subscribe*, *subscriber* hanya akan dinotifikasi sesuai dengan ketertarikannya melalui proses *content-filtering*. Notifikasi ke entitas *subscriber*, akan ditangani oleh *event notification*. Entitas *subscriber* dapat berwujud *mobile client* maupun *web client*. Sistem *publish-subscribe* mengatur masa *subscription* entitas *subscriber*, jika telah melewati masa kadaluarsa maka *publisher* akan dinotifikasi untuk menghentikan proses *publish*.

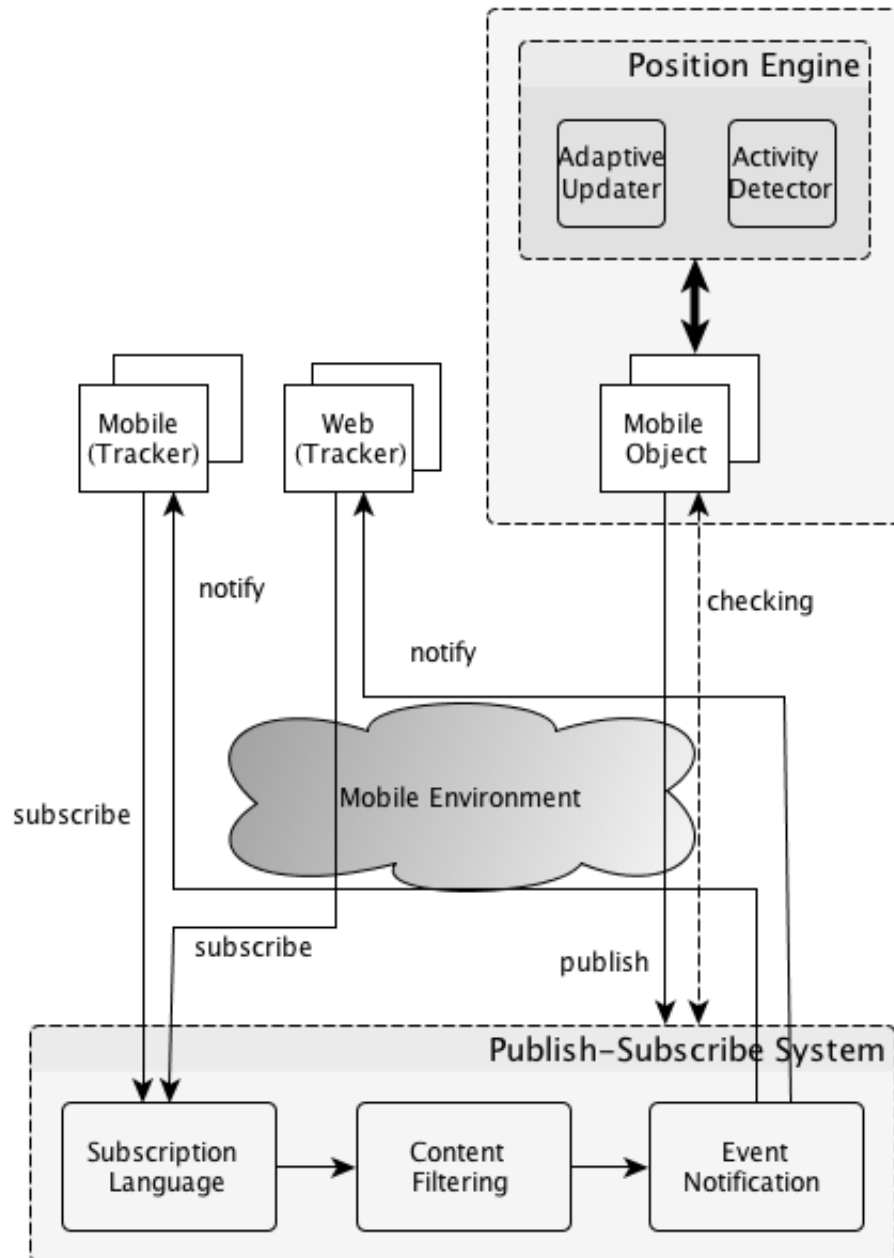
Dalam perangkat bergerak, ditanamkan sebuah modul adaptif dalam penentuan lokasi. Modul ini akan mendeteksi aktifitas obyek yang diamati, dari aktifitas ini menjadi dasar klasifikasi kebutuhan pembaruan serta presisi lokasi. Deteksi aktifitas didasarkan pada data *sampling* yang direkam menggunakan sensor *accelerometer*.



Gambar 3.3: Diagram alir sistem *tracking*

Interaksi pada sistem *publish-subscribe* tradisional, walaupun informasi dari *publisher* tidak dibutuhkan, pihak *publisher* akan tetap mem-*publish* informasi ke server. Untuk mengurangi hal ini, diperlukan mekanisme untuk menangani masalah ini seperti terlampir pada Gambar 3.3. Sedangkan untuk proses *tracking* akan mengadopsi diagram dari sistem EnTracked seperti yang tertera pada Gambar 2.5. Da-

lam komunikasi data pada proses *tracking* akan digantikan dengan skema *publish-subscribe* dengan protokol MQTT (MQ Telemetry Transport).



Gambar 3.4: Arsitektur rancangan sistem

3.4 Location Model

Pada proses komunikasi antara *tracker* dengan obyek yang menjadi target *tracking* terjadi pertukaran data. Dengan ini diperlukan pemodelan data yang dikirimkan. Dalam Tabel 3.1 dipaparkan pemodelan data yang dimaksud. Setiap target

tracking mempunyai atribut ID, Name, Role, Lat, Long dan Time. Deskripsi setiap atribut dijelaskan pada kolom Keterangan.

Tabel 3.1: Data Model

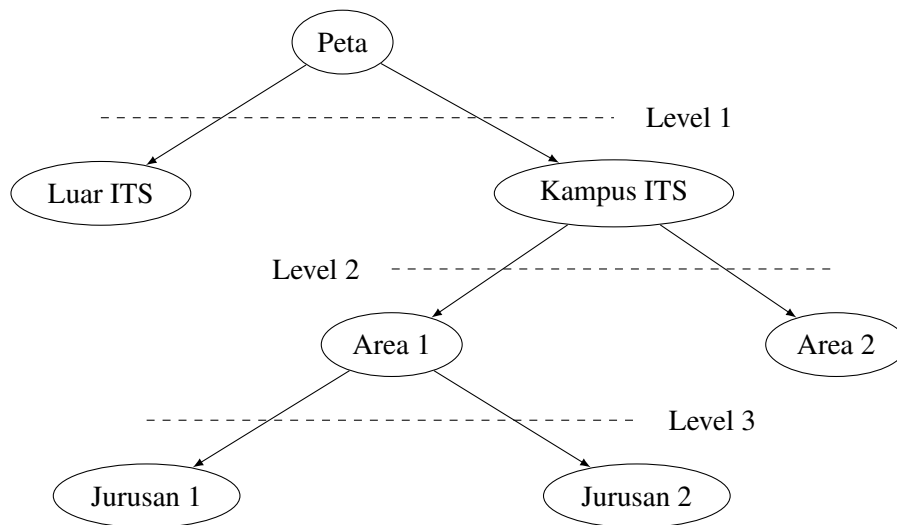
Atribut	Tipe	Keterangan
ID	String	Informasi identitas obyek yang di- <i>track</i>
Name	String	Nama dari obyek yang di- <i>track</i>
Role	String	Informasi hak akses dari obyek yang di- <i>track</i>
Lat	Float	Nilai <i>latitude</i>
Long	Float	Nilai <i>longitude</i>
Time	Date Time	Informasi mengenai waktu data dikirimkan

Dalam pemodelan data terdapat atribut Role, yang berfungsi untuk membedakan level hak akses seperti yang dipaparkan pada Tabel 3.2. Atribut Role dibagi menjadi tiga level hak akses, yaitu: Role 1, Role 2 dan Role 3. Perbedaan level ini mempengaruhi tingkat resolusi spasial yang berkaitan dengan privasi target *tracking*. Semakin tinggi level suatu *Role*, mempunyai hak akses yang lebih luas, contoh: Role 1 mempunyai resolusi spasial sampai level 3, sedangkan Role 3 hanya mempunyai sampai level 1.

Tabel 3.2: Level Hak Akses

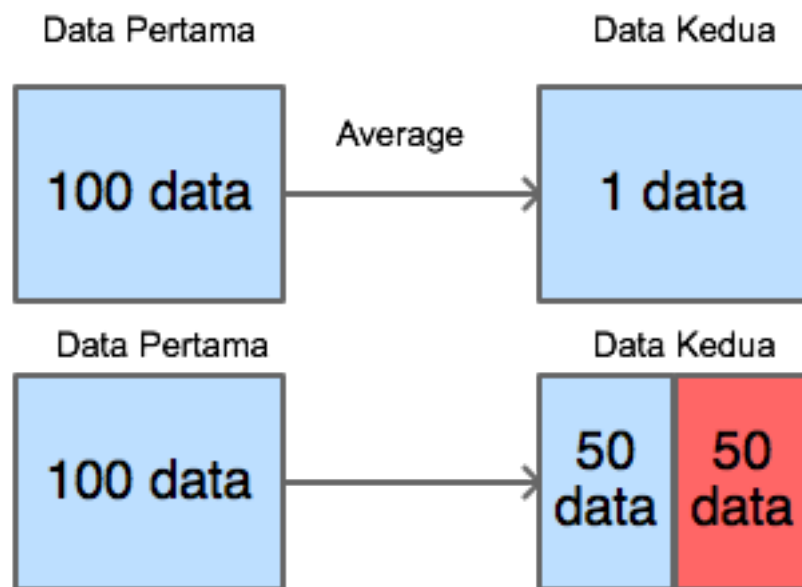
Role	Hak Akses
Role 1	Level 1 - Level 3
Role 2	Level 1 - Level 2
Role 3	Level 1

Pemodelan lokasi yang digunakan merupakan gabungan antara *Geometric* dan *Symbolic* yang bersifat hirarkis. Resolusi suatu informasi lokasi dipengaruhi oleh level hak akses, *tracker* yang mempunyai *Role* lebih tinggi atau selevel dengan target *tracking* memiliki resolusi informasi yang lebih detil. Pada Gambar 3.5 lingkungan Kampus Institut Teknologi Sepuluh Nopember (ITS) dibagi menjadi tiga tingkat level. Seorang *tracker* yang mempunyai hak akses level 1 hanya dapat mengetahui informasi sebatas target *tracking* sedang berada di wilayah kampus ITS atau tidak. Sedangkan untuk *tracker* yang mempunyai hak akses lebih tinggi dapat memperoleh informasi sampai level gedung/bangunan.



Gambar 3.5: Location model

3.5 Sampling Deteksi Aktivitas



Gambar 3.6: Windows Sampling dan Overlapping

Dalam proses *activity recognition* diperlukan data sampel. Keakuratan sebuah data merupakan hal yang sangat penting sebagai langkah awal dalam klasifikasi. Agar kebutuhan tersebut terpenuhi maka diperlukan suatu teknik interpretasi data yang khusus. Salah satu cara mengekstraksi data dengan *windows sampling*, yaitu suatu teknik mengekstraksi data dengan *sampling* data, setiap *windows* terdiri dari kumpulan data yang akan merepresentasikan satu buah data.

Cara pengambilan *sampling* data dapat dipaparkan pada Gambar 3.6. Dalam pengambilan data X dengan menggunakan sensor Y, misalnya setiap satu detik sensor Y menghasilkan seratus perubahan data. 100 data tersebut dapat direpresentasikan dalam satu data menggunakan nilai rata-rata yang ditunjukkan pada Gambar 3.6. *Overlapping* digunakan untuk menjaga agar data tetap konsisten sehingga data tidak terlalu divergen. Untuk pengambilan data berikutnya (kedua dan seterusnya), setengah dari *windows* diambil dari data sebelumnya dan setengah lagi dari perubahan sensor berikutnya.

3.6 Uji Coba dan Evaluasi

Pada bagian ini dipaparkan mengenai lingkungan yang dijadikan uji coba dan evaluasi. Pada sistem *tracking* multi target ini terdapat dua komponen utama, yaitu client dan server. Client merupakan entitas yang menjadi target *tracking* atau yang melakukan *tracking*. Client yang menjadi target *tracking* berupa perangkat *smartphone*, sedangkan pelaku *tracking* dapat berupa *smartphone* maupun web. Pada sisi server terdapat sebuah sistem *publish-subscribe* tunggal, yang memiliki *IP public*. Sistem *publish-subscribe* dibangun menggunakan *ActiveMQ*, dan untuk protokol komunikasi dengan client menggunakan protokol MQTT.



Gambar 3.7: Denah lokasi kampus Institut Teknologi Sepuluh Nopember

Area *tracking* yang digunakan adalah area kampus Insitut Teknologi Sepuluh Nopember (Gambar 3.7) yang sudah dilakukan pemodelan lokasi. Pembatasan area ini untuk memudahkan pengujian dan analisa. Proses *tracking* dilakukan pada orang yang bergerak dengan berjalan kaki (pedestrian) tanpa bantuan kendaraan.

Uji coba dilakukan dengan menghitung banyaknya paket yang dikirimkan, serta banyaknya pengurangan paket setelah modul adaptif diaktifkan dengan parameter yang berbeda pada lingkungan uji coba yang telah disiapkan. Parameter itu berupa jumlah target *tracking*, jarak interval pembaruan lokasi dengan variasi jumlah *publisher* dengan *subscriber*. Jumlah *publisher* dan *subscriber* yang direncanakan dalam pengujian adalah 10. Dari perhitungan ini, didapatkan seberapa signifikan efisiensi sistem *tracking* multi target.

3.7 Jadwal Kegiatan Penelitian

Pada Tabel 3.3 diuraikan mengenai jadwal kegiatan penelitian selama empat bulan. Jadwal akan dipetakan perminggu dalam empat bulan pengerjaan.

Tabel 3.3: Jadwal Kegiatan

Kegiatan	Bulan I 2013				Bulan II 2013				Bulan III 2013				Bulan IV 2013			
Studi Literatur																
Analisa dan desain																
Implementasi																
Uji coba																
Penulisan Laporan																

Daftar Pustaka

- Randell, C., Muller, H., *Context awareness by analysing accelerometer data*. The Fourth International Symposium on Wearable Computers IEEE Computer Society (pp. 175-176). 2000.
- Cugola, G., Murphy, L., A., Picco, G., P., *Content-based dispatching in a mobile environment*. Workshop su sistemi Distribuiti., 2000.
- Leonhardi, A., Rothermel, K., *A comparison of protocols for updating location information*. Cluster Computing, 2001.
- Leonhardi A., Nicu C., Rothermel K., *A map-based dead-reckoning protocol for updating location information*. Proceedings of 16th Int. Parallel and Distributed Processing Symposium, 2002.
- Kjaergaard, M., B., Langdal, J., Godsk, T., Toftkjaer, T., *EnTracked: Energy-efficient robust position tracking for mobile devices*.
- Eugster, P., Felber, P., A., Guerraoui, R., *The many faces of publish/subscribe*. ACM Computing Survey., 2003.
- Huang, Y., Garcia-Molina, H., *Publish/Subscribe in a Mobile Environment*., MobiDe'01: ACM Int. Workshop on Data engineering for Wireless and Mobile access., 2004.
- Civilis, A., Jensen, C. S., Pakalnis, S., *Techniques for efficient road network-network-based tracking of moving objects*. IEEE Trans. Data Eng., 2005.
- Becker, C., Durr F., *On Location Models for Ubiquitous Computing*. Personal and Ubiquitous Computing., 2005.
- Hinske, Steve., *Location Models*. Distributed Systems seminar., 2006.
- Hunkeler, U., Truong, L., *MQTT-S - A publish/subscribe protocol for wireless sensor networks*. IEEE Conference on COMSWARE., 2008.
- Frank, H., P., Charaf, H., *Mobile peer to peer (P2P) - A tutorial guide*., John Wiley & Sons, Ltd., 2009.
- Hong, J., Suh, E., Kim, S., J., *Context-aware systems: A literature review and classification*. POSMIS Lab, Industrial Management Engineering Building, Pohang University of Science Technology., 2009.
- Ashfaq, A., Naznin, M., Islam, A., *Energy-efficient multiple targets tracking using target kinematics in wireless sensor networks*. IEEE Conference., 2010.
- Salvador, Z., Alzua, A., Larrea, M., Lafuente, A., *A: Mobile XSiena: towards mobile publish/subscribe*., ACM International Conference on Distributed Event-Based Systems., 2010.
- Locations and Maps*. Dipetik April 21, 2013, dari Developer Android: <http://www.developer.android.com/guide/topics/location/index.html>

Android Location API - Tutorial. Dipetik April 21, 2013, dari Vogella:
<http://www.vogella.com/artivles/AndroidLocationAPI/article.html>
ActiveMQ. Dipetik April 21, 2013, dari ActiveMQ: <http://www.activemq.com>.